# FMP NOTEBOOKS: EDUCATIONAL MATERIAL FOR TEACHING AND LEARNING FUNDAMENTALS OF MUSIC PROCESSING

**Meinard Müller, Frank Zalkow**
International Audio Laboratories Erlangen, Germany
{meinard.mueller, frank.zalkow}@audiolabs-erlangen.de

## ABSTRACT

In this paper, we introduce a novel collection of educational material for teaching and learning fundamentals of music processing (FMP) with a particular focus on the audio domain. This collection, referred to as FMP notebooks, discusses well-established topics in Music Information Retrieval (MIR) as motivating application scenarios. The FMP notebooks provide detailed textbook-like explanations of central techniques and algorithms in combination with Python code examples that illustrate how to implement the theory. All components including the introductions of MIR scenarios, illustrations, sound examples, technical concepts, mathematical details, and code examples are integrated into a consistent and comprehensive framework based on Jupyter notebooks. The FMP notebooks are suited for studying the theory and practice, for generating educational material for lectures, as well as for providing baseline implementations for many MIR tasks, thus addressing students, teachers, and researchers.

## 1. INTRODUCTION

Music information retrieval (MIR) is an exciting and challenging area of research. Music not only connects people but also relates to many different research disciplines including signal processing, information retrieval, machine learning, musicology, and psychoacoustics. In its beginnings, research in MIR has borrowed many ideas and concepts from more established disciplines such as speech processing or computer linguistics. After twenty years, the MIR field has matured to an independent research area that has many things to offer to signal processing and other research disciplines [16]. In particular, thanks to the rich and challenging domain of music, there are many MIR tasks that can serve as motivation application scenarios for introducing, explaining, and studying techniques for audio processing, time-series analysis, and information retrieval.

In this paper, we introduce the **FMP notebooks**, which provide educational material for teaching and learning fundamentals of music processing. One primary goal of these

| Part | Title | Notions, Techniques & Algorithms | HTML | IPYNB |
|---|---|---|---|---|
| B | Basics | Basic information on Python, Jupyter notebooks, Anaconda package management system, Python environments, visualizations, and other topics | [html] | [ipynb] |
| 0 | Overview | Overview of the notebooks (https://www.audiolabs-erlangen.de/FMP) | [html] | [ipynb] |
| 1 | Music Representations | Music notation, MIDI, audio signal, waveform, pitch, loudness, timbre | [html] | [ipynb] |
| 2 | Fourier Analysis of Signals | Discrete/analog signal, sinusoid, exponential, Fourier transform, Fourier representation, DFT, FFT, STFT | [html] | [ipynb] |
| 3 | Music Synchronization | Chroma feature, dynamic programming, dynamic time warping (DTW), alignment, user interface | [html] | [ipynb] |
| 4 | Music Structure Analysis | Similarity matrix, repetition, thumbnail, homogeneity, novelty, evaluation, precision, recall, F-measure, visualization, scape plot | [html] | [ipynb] |
| 5 | Chord Recognition | Harmony, music theory, chords, scales, templates, hidden Markov model (HMM), evaluation | [html] | [ipynb] |
| 6 | Tempo and Beat Tracking | Onset, novelty, tempo, tempogram, beat, periodicity, Fourier analysis, autocorrelation | [html] | [ipynb] |
| 7 | Content-Based Audio Retrieval | Identification, fingerprint, indexing, inverted list, matching, version, cover song | [html] | [ipynb] |
| 8 | Musically Informed Audio Decomposition | Harmonic/percussive separation, signal reconstruction, instantaneous frequency, fundamental frequency (F0), trajectory, nonnegative matrix factorization (NMF) | [html] | [ipynb] |

**Figure 1**. Overview of FMP notebooks.

notebooks is to give an exciting and easy-to-understand introduction to MIR with a particular focus on audio-related analysis and retrieval tasks. Following the textbook [13], the notebooks treat many well-established MIR tasks as summarized in Figure 1. Within each MIR task, fundamental algorithmic approaches and techniques are discussed in detail. Going beyond and complementing traditional toolboxes, the FMP notebooks closely combine textbook-like explanations with Python code examples. Interleaving technical concepts, mathematical details, code examples, illustrations, and sound examples within a unifying and interactive Jupyter notebook framework helps to bridge the gap between theory and practice. Furthermore, the notebooks can be easily adapted to generate educational material (such as figures and sound examples) for lectures and to realize baseline approaches for many MIR tasks. The FMP notebooks (as well as HTML exports) are accessible under a Creative Commons license at: `https://www.audiolabs-erlangen.de/FMP`

There are some excellent software toolboxes such as `essentia` [2], `librosa` [12], `madmom` [1], Marsyas [20], or the MIRtoolbox [7], which provide open source software for MIR and music processing applications. We will give an overview of related toolboxes in Section 2. While such toolboxes aim at implementing a wide range of MIR functionalities, the main goal of the FMP notebooks is to promote the understanding of MIR concepts. Therefore, rather than providing compact and efficient code, the programming style used in the FMP code examples is simple and explicit with a flat functional hierarchy (at the cost of having some redundancy). The mathematical notation and the naming conventions used in the FMP notebooks are carefully matched to establish a close relationship between theory and practice. Furthermore, the notebooks allow a user to generate appealing multimedia objects such as figures and sound examples, which may be useful for lectures and scientific publications. In summary, educational and didactic considerations are the main guide in the development of the FMP notebooks. As such, we hope that these notebooks nicely complement existing open source toolboxes, fostering education and research in MIR.

The remainder of the paper is organized as follows. In Section 2, we review related software frameworks and toolboxes for audio and music processing. Then, in Section 3, we deal with the structure, content, and implementation of the FMP notebooks. In Section 4, we give some concrete examples of how the FMP notebooks can be used for learning and teaching music processing and MIR. Conclusions can be found in Section 5.

## 2. RELATED WORK

As said before, the main aim of the FMP notebooks is to help users to gain a deeper understanding of essential MIR techniques. Providing explicit and simple code examples (by consciously introducing redundancies), the notebooks are not intended to form a toolbox in a stricter sense. Instead, the FMP notebooks reimplement, integrate, and apply various functions that are also provided by existing toolboxes. In the following, we give a summary of open-source toolboxes that have been specifically designed for supporting MIR research.

There are a number of comprehensive and well-document toolboxes that provide modular source code for processing and analyzing music and audio signals. Prominent examples are the Marsyas toolbox [20], the MIRtoolbox [7], the jAudio toolbox [10], and the essentia library [2]. All these collections provide code for audio feature extraction as well as for MIR applications including music classification, melody extraction, beat tracking, and structure analysis.

There are also various toolboxes that focus on specific MIR applications such as the Chroma Toolbox [14] for chroma feature extraction, the Constant-Q Toolbox [19] for computing time-frequency transforms, the TSM Toolbox [3] for time-scale modification, the Tempogram Toolbox [5] for tempo and pulse tracking, and the SM Toolbox [15] as well as the MSAF toolbox [17] for audio structure analysis. While most of these toolboxes cover more traditional MIR techniques, the recent Python library madmom [1] also offers code for MIR approaches that employ deep learning techniques. Other useful toolboxes provide code for the evaluation of MIR approaches such as the mir_eval library [18] or for data augmentation such as the Audio Degradation Toolbox [9] or the muda library [11]. Other useful sources are the MIR notebooks [1] provided by Steve Tjoa as well as the companion website [2] of the textbook [8] on audio content analysis, which offers code for hands-on experience in audio and music processing. Furthermore, Xambó et al. [22] introduce a browser-based learning environment for teaching MIR and programming in high-schools.

In particular, we want to draw attention to the Python package librosa [12], which provides basic functions as well as advanced processing pipelines for several music and audio analysis tasks. librosa also comprises a gallery of advanced examples [3], which nicely illustrate how to use the package for approaching MIR tasks such as onset detection, music synchronization, harmonic-percussive separation, and audio structure analysis. The FMP notebooks are inspired by librosa and integrate, extend, and complement elements offered by this package. While librosa is designed to be an easy-to-use toolbox with convenient presets, the emphasis of the FMP notebooks is on the educational side providing detailed explanations of theoretical and practical aspects. We hope that the FMP notebooks serve as a good basis for carrying on with more advanced techniques as provided by powerful toolboxes such as librosa, essentia, or madmom.

## 3. STRUCTURE OF NOTEBOOKS

The FMP notebooks are structured in ten parts as shown by the table of Figure 1. **Part 0** (also containing this table) is the starting notebook, which is opened when calling https://www.audiolabs-erlangen.de/FMP. Besides giving an overview, this notebook also provides information on the license (Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License), the main contributors, and some links to related toolboxes. **Part B** provides **basic** introductions to the Jupyter notebook framework, the Python programming language, and other technical concepts underlying these notebooks (see Section 3.1). The main body of the FMP notebooks, which covers different music processing and MIR scenarios, consists of **Part 1** to **Part 8** (closely following the eight chapters of the textbook [13]). These parts are described in Section 3.2.

### 3.1 Technical Framework

The notebooks of **Part B** serve different purposes. First, these notebooks describe the main tools used for developing the FMP notebooks. Second, they give short intro-

[1] https://musicinformationretrieval.com/
[2] https://www.AudioContentAnalysis.org
[3] https://librosa.github.io/librosa/advanced.html

ductions of the relevant technical concepts while providing links to more detailed tutorials. Third, the notebooks give examples for best practices in programming as well as for generating and using code, figures, and sound elements. In the following, we describe the technical framework underlying the FMP notebooks while summarizing the content of **Part B**.

#### *3.1.1 Jupyter Notebook*

The FMP notebooks are based on the Jupyter notebook framework. This open-source web application allows users to create documents that contain live code, text-based information, mathematical formulas, plots, images, sound examples, and videos. Jupyter notebooks are often used as a publishing format for reproducible computational workflows [6]. They can be exported to a static HTML format, which makes it possible to generate web applications that can be accessed through standard web browsers with no specific technical requirements. **Part B** introduces some relevant elements of the Jupyter framework including practical aspects such as the most important Jupyter operators and keyboard shortcuts.

#### *3.1.2 Installation*

To run the FMP notebooks, one needs to install Python, Jupyter, and additional Python packages. In **Part B**, we introduce the Anaconda Python distribution with its package and environment manager, which allows for quickly installing, running, and updating the required software packages. Furthermore, we provide a file which specifies an environment called `FMP`. This environment comprises all packages (specified by name and version number) needed for the FMP notebooks. Giving a step-by-step description, we explain how to use Anaconda to set up this environment.

#### *3.1.3 Multimedia*

One notebook of **Part B** gives a short overview of how to integrate multimedia objects (in particular, audio, image, and video objects) into a Jupyter notebook. Rather than being comprehensive, we only give a selection of possibilities as used in the other parts of the FMP notebooks. In particular, we discuss two alternatives: a direct integration of images, video, and audio elements using HTML tags as well as an integration using the Python module `IPython.display`.

#### *3.1.4 Python*

In the FMP notebooks, we use Python as the programming language. The reason for this choice is that Python is a open-source general-purpose language, which is widely used in scientific computing and offers plenty of resources in data sciences and machine learning. Furthermore, being a beginner-friendly language, it suits the didactic orientation of the FMP notebooks well. **Part B** contains a short introduction to Python summarizing the most important data types, control structures, and functions as occurring in later parts of the FMP notebooks. One of our design principles is to keep the required programming skills at an elementary level. Furthermore, one finds code examples that illustrate how to create appealing figures, process audio files, and program interactive plots.

#### *3.1.5 Numba*

As one side topic, we also give a short introduction to the Python package `Numba`, which offers an open source just-in-time (JIT) compiler that translates a subset of Python code into fast machine code. Even though not crucial from a functionality point of view, this package can be used to significantly speed up (sometimes a factor of 100) some of the implementations offered by the FMP notebooks.

#### *3.1.6 Further Topics and Summary*

Further topics covered by **Part B** are descriptions of relevant Python libraries, some basic information of the version control system Git, and links to tools that are helpful for music processing and MIR.

In summary, with having the notebooks of **Part B**, our goal is to make the FMP notebooks self-contained (at least, to a high degree). Rather than trying to be comprehensive, we give useful and instructive code examples that become relevant in the other parts. Furthermore, **Part B** also motivates and documents how the FMP notebooks were created.

### 3.2 Music Processing and MIR Scenarios

The main music processing and MIR topics covered by the FMP notebooks are organized in eight parts, which follow the eight chapters of the textbook on Fundamentals of Music Processing [13]. The notebooks include introductions for each MIR task, provide important mathematical definitions, and describe computational approaches in detail. One primary purpose of the FMP notebooks is to provide audio-visual material as well as Python code examples that implement the computational approaches described before. Additionally, the FMP notebooks provide code that allows a user to experiment with parameters and to gain an understanding of the computed results by suitable visualizations and sonifications. These functionalities also make it easily possible to input different music examples and to generate figures and illustrations that can be used in lectures and scientific articles. This way, the FMP notebooks complement and go beyond the textbook [13], where one finds a more mathematically oriented approach to MIR. In the following, we summarize the main content of the music processing and MIR scenarios covered by the FMP notebooks.

**Part 1 (Music Representations).** Musical information can be represented in many different ways. In this part, we cover three widely used music representations: sheet music, symbolic, and audio representations. Besides introducing basic terminology that is used throughout the following FMP notebooks, we provide Python code to study musical and acoustic properties of audio signals including aspects such as frequency, pitch, dynamics, and timbre. For example, there are code snippets for comparing different tuning systems (equal-tempered, Pythagorean, harmonic series) and for generating Shepard tones.

**Part 2 (Fourier Analysis of Signals).** In this part, we approach the Fourier transform (used as the main signal processing tool in these notebooks) from various perspectives. We provide code to better understand complex numbers and exponential functions, which form the basis for the discrete Fourier transform (DFT). Also, the fast Fourier transform (FFT)—an algorithm of great beauty and high practical relevance—is covered in theory and practice. As another important topic, we discuss the short-time Fourier transform (STFT). In this context, we address issues such as sampling, padding, and axis conventions—issues that are often neglected in theory—from a practical perspective.

**Part 3 (Music Synchronization).** The objective of music synchronization is to temporally align different versions of the same underlying piece of music. Considering this scenario, we provide code examples for generating chroma-based music features, which capture properties that are related to harmony and melody. In this context, we also address issues of high practical relevance including tuning, logarithmic compression, as well as spectral and temporal resolution—aspects that have a significant influence on the features' properties. Furthermore, we study an alignment technique known as dynamic time warping (DTW), a concept that is applicable for the analysis of general time series. For its efficient computation, we discuss an algorithm based on dynamic programming—a widely used method for solving a complex problem by breaking it down into a collection of simpler subproblems.

**Part 4 (Music Structure Analysis).** In this part, we address a central and well-researched area within MIR known as music structure analysis. Given a music recording, the objective is to identify critical structural elements and to segment the recording according to these elements. Within this scenario, we discuss fundamental segmentation principles based on repetitions, homogeneity, and novelty—principles that also apply to other types of multimedia beyond music. In particular, we provide code for generating, visualizing, and understanding self-similarity matrices and for modifying their structural properties using a variety of enhancement strategies. The notebooks also cover classical approaches for novelty detection and audio thumbnailing. Finally, we introduce scape plot representations and demonstrate how this concept can be used to generate beautiful visualizations of time-dependent properties in a compact and hierarchical way.

**Part 5 (Chord Recognition).** Another essential and long-studied MIR task is the analysis of harmonic properties of a piece of music by determining an explicit progression of chords from a given audio recording—a task often referred to as automatic chord recognition. Within this scenario, we first discuss some basic theory of harmony including concepts such as intervals, chords, and scales. To better understand these musical concepts, the notebooks provide code for generating and interacting with suitable sound examples. Furthermore, we introduce a simple baseline system for chord recognition based on a template-based matching procedure. This system is then extended by hidden Markov models (HMMs)—a concept of central importance for the analysis of temporal patterns in time-dependent data streams including speech, gestures, and music. Besides algorithmic aspects and their implementation, we use the chord recognition scenario to illustrate the importance of feature design choices and the effect of temporal smoothing strategies. Such issues become of crucial importance when comparing, understanding, and exploring the potential of more involved chord recognition systems (e. g., based on deep learning).

**Part 6 (Tempo and Beat Tracking).** Tempo and beat are fundamental properties of music. In this part, we introduce the basic ideas on how to extract tempo-related information from audio recordings. A first task, known as onset detection, aims at locating note onset information by detecting changes in energy and spectral content. The notebooks not only introduce the theory but also provide code for implementing and comparing different onset detectors. To derive tempo and beat information, note onset candidates are analyzed concerning quasiperiodic patterns. This second step leads us to the study of general methods for local periodicity analysis of time series. In particular, we introduce two conceptually different methods: one based on Fourier analysis and the other one based on autocorrelation. Furthermore, the notebooks provide code for visualizing time-tempo representations, which deepen the understanding of musical and algorithmic aspects. Finally, the FMP notebooks cover fundamental procedures for predominant local pulse estimation and global beat tracking.

**Part 7 (Content-Based Audio Retrieval).** A central topic in MIR is concerned with the development of search engines that enable users to explore music collections in a flexible and intuitive way. In this part, we discuss audio retrieval strategies that follow the query-by-example paradigm: given an audio query, the task is to retrieve all documents that are somehow similar or related to the query. Within this scenario, we discuss the issue of specificity, which refers to the degree of similarity between the query and the database documents. First, we deal with the problem of audio identification (a retrieval task of high specificity), where the objective is to identify the particular audio recording that is the source of the query. In particular, we introduce the main ideas of an audio identification system based on spectral peaks—a technique used in many commercial applications such as Shazam [21]. Then, the notebooks cover two related retrieval tasks of lower specificity referred to as audio matching and version identification, where the goal is to identify recordings with performance variations and other versions (e. g., cover songs). The main goals of the notebooks are to provide code for baseline systems and for gaining a better understanding of

$$\mathrm{DFT}_N = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \sigma_N & \sigma_N^2 & \dots & \sigma_N^{N-1} \\ 1 & \sigma_N^2 & \sigma_N^4 & \dots & \sigma_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_N^{N-1} & \sigma_N^{2(N-1)} & \dots & \sigma_N^{(N-1)(N-1)} \end{pmatrix}$$

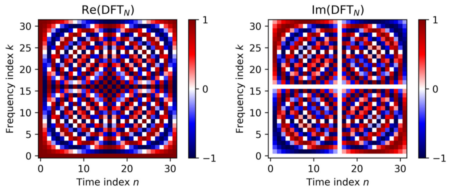


**Figure 2**. The matrix $\mathrm{DFT}_N$ and a visualization of its real and imaginary parts for the case $N = 32$.

the practical requirements of the different retrieval tasks.

**Part 8 (Musically Informed Audio Decomposition).** In the final part on audio decomposition, the notebooks cover challenging research directions that are related to source separation. Within this wide research area, we consider three subproblems: harmonic–percussive separation, main melody extraction, and score-informed audio decomposition. Within these scenarios, the notebooks offer detailed explanations and implementations of essential techniques including instantaneous frequency estimation, fundamental frequency (F0) estimation, spectrogram inversion, and nonnegative matrix factorization (NMF). These techniques are useful for a variety of general multimedia processing tasks beyond source separation and music processing. Besides algorithmic and computational aspects, we again encounter in this part of the notebooks a variety of acoustic and musical properties of audio recordings. Providing tools and instructive scenarios for gaining a good understanding of such properties is a central and overarching objective of the FMP notebooks.

## 4. EXAMPLES

In this section, we give some short examples that illustrate some of the educational aspects of the FMP notebooks.

We start with a classical signal processing topic. Given a discrete signal $x = (x(0), x(1), \dots, x(N-1))^\top \in \mathbb{R}^N$ of length $N$, the discrete Fourier transform (DFT) is defined by $X(k) := \sum_{n=0}^{N-1} x(n)\exp(-2\pi ikn/N)$ for $k \in [0 : N-1]$. The vector $X \in \mathbb{C}^N$ can be interpreted as frequency representation of the time-domain signal $x$. The FMP notebooks approach the DFT in various ways, including the usage of inner products and their geometric interpretation. Defining the complex number $\sigma_N := \exp(-2\pi i/N)$, Figure 2 shows the matrix $\mathrm{DFT}_N$ (given by $\mathrm{DFT}_N(n,k) = \sigma_N^{nk}$ for $n,k \in [0 : N-1]$) along with a visualization of its real and imaginary parts. Furthermore, the notebooks explain how to evaluate the DFT efficiently using the fast Fourier transform (FFT). The general idea of the FMP notebooks is not to shy away from mathematics. Instead, the notebooks provide rigorous introductions to the theory, which are interleaved with code examples that further explain, implement, and visualize abstract concepts.

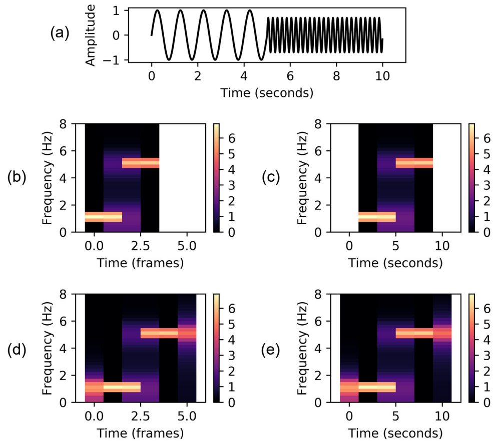


**Figure 3**. Time-domain signal (a) and its STFT without padding (b/c) and with zero-padding (d/e).

To recover time information hidden in the Fourier domain, the main idea of the short-time Fourier transform (STFT) is to consider only small sections of the signal. These sections are obtained by multiplying shifted versions of a window function with the original signal and by computing a Fourier transform for each of the resulting windowed signals. This results in a sequence of spectral vectors, also called frames. In practice, the correct physical interpretation of discrete objects such as samples, frames, and spectral coefficients can be tricky. Also, there are many different conventions when applying windowing. Figure 3 shows a signal (two subsequent sinusoids of 1 Hz and 5 Hz, respectively) and its STFT. In the FMP notebooks, we explain how to correctly interpret discrete parameters, discuss different windowing conventions (including padding), and show how to correctly visualize feature representations (taking a centric view).

For Western music, one often uses a twelve-tone equal-tempered scale, where the 12 pitch classes correspond to the twelve chroma values $\{\mathrm{C}, \mathrm{C}^\sharp, \mathrm{D}, \mathrm{D}^\sharp, \dots, \mathrm{B}\}$. Aggregating all spectral information that relates to a given pitch class into a single coefficient, a spectrogram can be transformed in a chromagram, see Figure 4. This example also demonstrates how to generate accurate and visually appealing figures, which can be a tricky and time-consuming effort. In the FMP notebooks, we give numerous examples on how to enhance, place, and align figure elements. In Figure 4, for example, a waveform (given in samples) and a chromagram (given in frames) are visually aligned using a common time axis (given in seconds). Furthermore, using an adapted colormap enhances essential structures in the chromagram. Finally, the size and the placement of the three subplots are controlled using a grid structure.

Such chromagram representations, which particularly capture harmonic and melodic properties of an audio recording, have turned out to be a powerful tool for various MIR tasks. One such task is music synchronization, where the objective is to automatically link different versions of the same piece of music. Figure 5 shows a synchronization result when aligning two different recordings of the be-

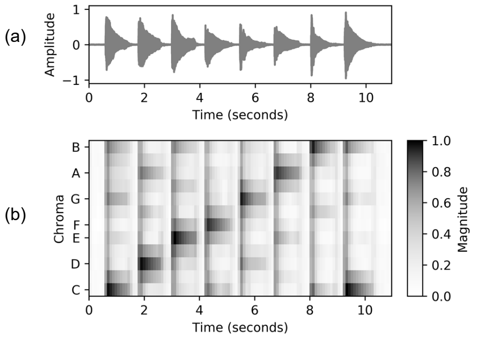


**Figure 4**. Waveform (a) of a C-major scale and resulting chromagram (b).

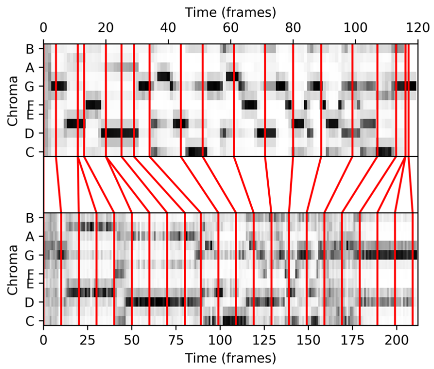


**Figure 5**. Chromagram representations of two different recordings of the beginning of Beethoven's Fifth Symphony and resulting synchronization result (aligned time positions are indicated by red lines).

ginning of Beethoven's Fifth Symphony. Using a chroma-based alignment procedure based on dynamic time warping as an example approach, the FMP notebooks provide a detailed treatment of music synchronization including a musically informed motivation, algorithmic descriptions, implementation details, graphical representations, and application scenarios.

Chromagrams are also commonly used representations for tasks such as structure analysis and chord recognition. Figure 6 shows the annotated score as well as a chord recognition result obtained from an audio recording of the The Beatles' song "Let It Be." The FMP notebooks not only describe and implement computational approaches, but also discuss the results in a musically informed fashion by looking at real-world music and audio examples. Furthermore, common evaluation measures such as precision, recall, and F-measure are introduced including a discussion of their benefits and limitations within concrete MIR scenarios.

In our final example, we consider a task that is often referred to as harmonic–percussive separation (HPS), where the goal is to decompose a given audio signal into two parts: one consisting of the harmonic and another one of the percussive events of the original signal [4]. Since this task is very instructive from an educational point of view, it was included in the FMP notebooks. First, the task is suited to reflect on acoustic qualities of sound sources, see

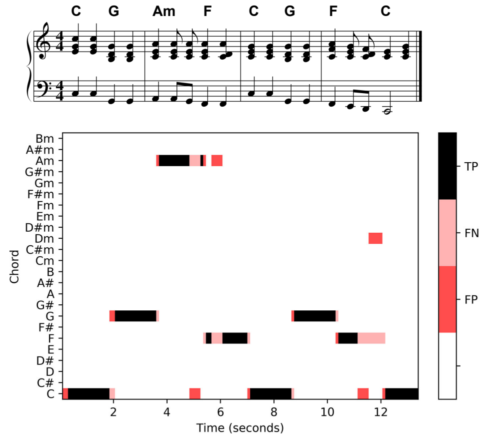


**Figure 6**. Chord recognition result for the first measures of The Beatles' song "Let It Be."

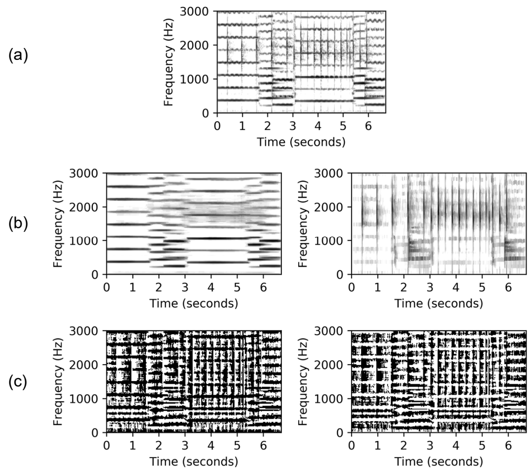


**Figure 7**. Spectrogram (a) of a recording consisting of violin sounds (harmonic components) superimposed with castanet clicks (percussive components). Furthermore, the figure shows filtered spectrograms (b) and derived binary masks (c).

Figure 7a. Second, the HPS approach discussed involves fundamental (vertical and horizontal) filtering techniques applied to a spectrogram representation, see Figure 7b. Third, it involves spectral masking techniques related to Wiener filtering, see Figure 7c. Finally, one requires signal reconstruction techniques by inverting a modified STFT. The FMP notebooks deal with these central topics using the HPS scenario as motivation and illustration.

## 5. CONCLUSIONS

The FMP notebooks put together a package for studying central MIR tasks providing detailed explanations, mathematical details, code examples, instructive sound examples, and illustrative figures within a unifying framework. While going hand in hand with existing toolboxes such as `librosa` [12], the FMP notebooks complement existing MIR resources with the aim to bridge the gap between theory and practice and by providing educational material useful for teaching and learning fundamentals of music processing.

**Acknowledgments:** We thank Stefan Balke, Sebastian Rosenzweig, Christof Weiß, and all other contributors for helping us with programming and testing. This work was supported by the German Research Foundation (MU 2686/10-1, DFG MU 2686/11-1, MU 2686/12-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institut für Integrierte Schaltungen IIS.

## 6. REFERENCES

[1] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. madmom: A new Python audio and music signal processing library. In *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*, pages 1174–1178, Amsterdam, The Netherlands, 2016.

[2] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, José R. Zapata, and Xavier Serra. Essentia: An audio analysis library for music information retrieval. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 493–498, Curitiba, Brazil, 2013.

[3] Jonathan Driedger and Meinard Müller. TSM Toolbox: MATLAB implementations of time-scale modification algorithms. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 249–256, Erlangen, Germany, 2014.

[4] Derry FitzGerald. Harmonic/percussive separation using median filtering. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 246–253, Graz, Austria, September 2010.

[5] Peter Grosche and Meinard Müller. Tempogram Toolbox: MATLAB tempo and pulse analysis of music recordings. In *Late-Breaking and Demo Session of the Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Miami, Florida, USA, October 2011.

[6] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter development team. Jupyter notebooks—a publishing format for reproducible computational workflows. In *Proceedings of the International Conference on Electronic Publishing*, pages 87–90, Göttingen, Germany, 2016.

[7] Olivier Lartillot and Petri Toiviainen. MIR in MATLAB (II): A toolbox for musical feature extraction from audio. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 127–130, Vienna, Austria, 2007.

[8] Alexander Lerch. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. John Wiley & Sons, 2012.

[9] Matthias Mauch and Sebastian Ewert. The audio degradation toolbox and its application to robustness evaluation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 83–88, Curitiba, Brazil, 2013.

[10] Daniel McEnnis, Cory McKay, Ichiro Fujinaga, and Philippe Depalle. jAudio: A feature extraction library. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 600–603, London, UK, 2005.

[11] Brian McFee, Eric J. Humphrey, and Juan Pablo Bello. A software framework for musical data augmentation. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, pages 248–254, Málaga, Spain, 2015.

[12] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. Librosa: Audio and music signal analysis in Python. In *Proceedings the Python Science Conference*, pages 18–25, Austin, Texas, USA, 2015.

[13] Meinard Müller. *Fundamentals of Music Processing*. Springer Verlag, 2015.

[14] Meinard Müller and Sebastian Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 215–220, Miami, Florida, USA, 2011.

[15] Meinard Müller, Nanzhu Jiang, and Harald Grohganz. SM Toolbox: MATLAB implementations for computing and enhancing similarity matrices. In *Proceedings of the Audio Engineering Society (AES) Conference on Semantic Audio*, London, UK, 2014.

[16] Meinard Müller, Bryan Pardo, Gautham J. Mysore, and Vesa Välimäki. Recent advances in music signal processing. *IEEE Signal Processing Magazine*, 36(1):17–19, 2019.

[17] Oriol Nieto and Juan Pablo Bello. Systematic exploration of computational music structure research. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 547–553, New York City, USA, 2016.

[18] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. mir_eval: A transparent implementation of common MIR metrics. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 367–372, Taipei, Taiwan, 2014.

[19] Christian Schörkhuber and Anssi P. Klapuri. Constant-Q transform toolbox for music processing. In *Proceedings of the Sound and Music Computing Conference (SMC)*, Barcelona, Spain, 2010.

[20] George Tzanetakis. Music analysis, retrieval and synthesis of audio signals MARSYAS. In *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*, pages 931–932, Vancouver, British Columbia, Canada, 2009.

[21] Avery Wang. An industrial strength audio search algorithm. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 7–13, Baltimore, Maryland, USA, 2003.

[22] Anna Xambó, Alexander Lerch, and Jason Freeman. Learning to code through MIR. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017.