

libfmp: A Python Package for Fundamentals of Music Processing

Meinard Müller¹ and Frank Zalkow¹

¹ International Audio Laboratories Erlangen

DOI: [10.21105/joss.03326](https://doi.org/10.21105/joss.03326)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Arfon Smith](#) ↗

Reviewers:

- [@brunaw](#)
- [@expectatronum](#)

Submitted: 27 April 2021

Published: 20 July 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

The revolution in music distribution, storage, and consumption has fueled tremendous interest in developing techniques and tools for organizing, structuring, retrieving, navigating, and presenting music-related data. As a result, the academic field of music information retrieval (MIR) has matured over the last 20 years into an independent research area related to many different disciplines, including engineering, computer science, mathematics, and musicology. In this contribution, we introduce the Python package libfmp, which provides implementations of well-established model-based algorithms for various MIR tasks (with a focus on the audio domain), including beat tracking, onset detection, chord recognition, music synchronization, version identification, music segmentation, novelty detection, and audio decomposition. Such traditional approaches not only yield valuable baselines for modern data-driven strategies (e.g., using deep learning) but are also instructive from an educational viewpoint deepening the understanding of the MIR task and music data at hand. Our libfmp package is inspired and closely follows conventions as introduced by librosa, which is a widely used Python library containing standardized and flexible reference implementations of many common methods in audio and music processing ([McFee et al., 2015](#)). While the two packages overlap concerning basic feature extraction and MIR algorithms, libfmp contains several reference implementations of advanced music processing pipelines not yet covered by librosa (or other open-source software). Whereas the librosa package is intended to facilitate the high-level composition of basic methods into complex pipelines, a major emphasis of libfmp is on the educational side, promoting the understanding of MIR concepts by closely following the textbook on Fundamentals of Music Processing (FMP) ([Müller, 2015](#)). In this way, we hope that libfmp constitutes a valuable complement to existing open-source toolboxes such as librosa while fostering education and research in MIR.

Introduction

Thanks to the rich and challenging domain of music, the MIR field has many things to offer to signal processing and other research disciplines ([Müller et al., 2019](#)). In particular, many MIR tasks can motivate application scenarios to introduce, explain, and study techniques for audio processing, time-series analysis, and information retrieval. Furthermore, the increasing availability of suitably designed software packages and freely accessible web-based frameworks have made education in MIR, signal processing, and general computer science more interactive. Since its beginnings, the MIR community has contributed several excellent toolboxes that provide modular source code for processing and analyzing music signals. Prominent examples are *essentia* ([Bogdanov et al., 2013](#)), *madmom* ([Böck et al., 2016](#)), *Marsyas* ([Tzanetakis, 2009](#)), or the *MIRtoolbox* ([Lartillot & Toivainen, 2007](#)). These toolboxes are mainly designed for research-oriented access to audio processing, yielding code for audio feature extraction and

various MIR applications. Providing modular and readable code in combination with advanced MIR pipelines, the *librosa* package (McFee et al., 2015) strives for a low barrier to entry MIR research, thus representing a prominent example of building a bridge between education and research.

Motivated by educational considerations, the FMP notebooks, which are a comprehensive collection of educational material for teaching and learning fundamentals of music processing (FMP), were recently introduced (Müller & Zalkow, 2019). Closely following the textbook by Müller (2015), the FMP notebooks comprise detailed textbook-like explanations of central techniques and algorithms combined with Python code examples that illustrate how to implement the methods. All components, including the introductions of MIR scenarios, illustrations, sound examples, technical concepts, mathematical details, and code examples, are integrated into a unified framework based on Jupyter notebooks. Thus, the FMP notebooks provide an interactive framework for students to learn about and experiment with signal processing and MIR algorithms. However, when students transition from being learners to becoming researchers, they may outgrow the FMP notebooks and begin developing their own DSP methods and programs (Müller et al., 2021). The *libfmp* package is designed to aid students with this transition by providing proper software support for building and experimenting with complex MIR pipelines. Whereas the FMP notebooks offer an introduction to fundamental concepts in MIR step by step, the *libfmp* package bundles the presented core concepts in the form of well-documented and easy-to-use Python functions. While integrating and building on the *librosa* package, *libfmp* offers alternative implementations of basic concepts in signal processing (comparing and discussing differences in the FMP notebooks). As a main contribution, *libfmp* provides various reference implementations of previously published MIR methods, not yet covered by *librosa* and other publicly available Python software.

Statement of Need

Going beyond and complementing existing Python packages, the *libfmp* package contains (previously unpublished) reference implementations of MIR algorithms from the literature as well as new Python implementations of previously published MATLAB toolboxes. For example, *libfmp* includes the core functionality of the MATLAB Tempogram Toolbox (Grosche & Müller, 2011a, 2011b)¹ for computing various mid-level tempogram representations (Grosche et al., 2010) and for extracting beat and local pulse information (Grosche & Müller, 2011a). Furthermore, *libfmp* extends the MATLAB SM Toolbox (Müller et al., 2014)² for computing and enhancing similarity matrices with applications to music structure analysis. In particular, it contains a Python reference implementation for audio thumbnailing as introduced by Müller et al. (2013) and novel functionality for generating synthetic self-similarity matrices from reference annotations. Other examples of previously unpublished reference implementations from the literature include the computation of tempo curves (Müller et al., 2009), the fragment-level retrieval of different music performances (Müller et al., 2005), and the separation of audio signals into harmonic, percussive, and residual components (Driedger et al., 2014). Also, *libfmp* contains an implementation of a novel baseline algorithm for tuning estimation. Finally, as shown in the FMP notebooks, the functions of *libfmp* can be used to conduct systematic series of experiments as described in the literature for tasks such as musically informed audio decomposition (Ewert & Müller, 2012) and automated chord recognition using different chroma feature types (Jiang et al., 2011).

¹<https://www.audiolabs-erlangen.de/resources/MIR/tempogramtoolbox>

²<https://www.audiolabs-erlangen.de/resources/MIR/SMtoolbox>

Design Choices

When designing libfmp, we had different objectives in mind. First, we tried to keep a close connection to the textbook by Müller (2015) and the FMP notebooks (Müller & Zalkow, 2019), thus establishing a close relationship between theory and practice. To this end, we carefully adopted naming conventions and the code structure to match the textbook's mathematical notions and algorithmic concepts. Furthermore, we split the package libfmp into subpackages (called B, C1, C2, ..., C8) corresponding to the textbook's chapters. Furthermore, the docstring of each libfmp function specifies the FMP notebook where the function is introduced (typically in a step-by-step fashion interleaved with additional explanations). Second, we followed many of the design principles suggested by librosa (McFee et al., 2015) to keep the entry barrier for students and researchers who may not be programming experts low. In particular, the programming style is kept explicit and straightforward with a flat, functional hierarchy. Third, based on expert knowledge, we specified a meaningful variable preset within each function, which allows users to experiment with the code immediately.

The code of libfmp is hosted in a publicly available GitHub repository.³ We also provide an API documentation for the libfmp functions,⁴ which complements the educational step-by-step explanations given in the FMP notebooks.⁵ Finally, we included the libfmp package into the Python package index PyPi, such that libfmp can be installed with the standard Python package manager pip.⁶

Acknowledgements

The libfmp package builds on results, material, and insights that have been obtained in close collaboration with different people. We would like to express our gratitude to former and current students, collaborators, and colleagues who have influenced and supported us in creating this package, including Vlora Arifi-Müller, Stefan Balke, Michael Krause, Brian McFee, Sebastian Rosenzweig, Fabian-Robert Stöter, Steve Tjoa, Angel Villar-Corrales, Christof Weiß, and Tim Zunner. We also thank the German Research Foundation (DFG) for various research grants that allowed us for conducting fundamental research in music processing (in particular, DFG-MU 2686/11-1, DFG-MU 2686/12-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS.

References

- Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J. R., & Serra, X. (2013). Essentia: An audio analysis library for music information retrieval. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 493–498. <https://doi.org/10.5281/zenodo.1415016>
- Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F., & Widmer, G. (2016). Madmom: A new Python audio and music signal processing library. *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*, 1174–1178. <https://doi.org/10.1145/2964284.2973795>

³<https://github.com/meinardmueller/libfmp>

⁴<https://meinardmueller.github.io/libfmp>

⁵<https://www.audiolabs-erlangen.de/FMP>

⁶<https://pypi.org/project/libfmp>

- Driedger, J., Müller, M., & Disch, S. (2014). Extending harmonic–percussive separation of audio signals. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 611–616. <https://doi.org/10.5281/zenodo.1415226>
- Ewert, S., & Müller, M. (2012). Using score-informed constraints for NMF-based source separation. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 129–132. <https://doi.org/10.1109/ICASSP.2012.6287834>
- Grosche, P., & Müller, M. (2011a). Extracting predominant local pulse information from music recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6), 1688–1701. <https://doi.org/10.1109/TASL.2010.2096216>
- Grosche, P., & Müller, M. (2011b). Tempogram Toolbox: MATLAB tempo and pulse analysis of music recordings. *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*.
- Grosche, P., Müller, M., & Kurth, F. (2010). Cyclic tempogram – a mid-level tempo representation for music signals. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 5522–5525. <https://doi.org/10.1109/ICASSP.2010.5495219>
- Jiang, N., Grosche, P., Konz, V., & Müller, M. (2011). Analyzing chroma feature types for automated chord recognition. *Proceedings of the AES Conference on Semantic Audio*.
- Lartillot, O., & Toiviainen, P. (2007). MIR in MATLAB (II): A toolbox for musical feature extraction from audio. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 127–130. <https://doi.org/10.5281/zenodo.1417145>
- McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., & Nieto, O. (2015). Librosa: Audio and music signal analysis in Python. *Proceedings the Python Science Conference*, 18–25. <https://doi.org/10.25080/Majora-7b98e3ed-003>
- Müller, M. (2015). *Fundamentals of music processing* [Monograph]. Springer Verlag.
- Müller, M., Jiang, N., & Grohganz, H. (2014). SM Toolbox: MATLAB implementations for computing and enhancing similarity matrices. *Proceedings of the Audio Engineering Society (AES) Conference on Semantic Audio*.
- Müller, M., Jiang, N., & Grosche, P. (2013). A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(3), 531–543. <https://doi.org/10.1109/TASL.2012.2227732>
- Müller, M., Konz, V., Scharfstein, A., Ewert, S., & Clausen, M. (2009). Towards automated extraction of tempo parameters from expressive music recordings. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 69–74. <https://doi.org/10.5281/zenodo.1416024>
- Müller, M., Kurth, F., & Clausen, M. (2005). Audio matching via chroma-based statistical features. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 288–295. <https://doi.org/10.5281/zenodo.1416800>
- Müller, M., McFee, B., & Kinnaird, K. (2021). Interactive learning of signal processing through music. *IEEE Signal Processing Magazine*, Accepted for publication.
- Müller, M., Pardo, B., Mysore, G. J., & Välimäki, V. (2019). Recent advances in music signal processing. *IEEE Signal Processing Magazine*, 36(1), 17–19. <https://doi.org/10.1109/MSP.2018.2876190>
- Müller, M., & Zalkow, F. (2019). FMP Notebooks: Educational material for teaching and learning fundamentals of music processing. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 573–580. <https://doi.org/10.5281/zenodo.3527872>

Tzanetakis, G. (2009). Music analysis, retrieval and synthesis of audio signals MARSYAS. *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*, 931–932.
<https://doi.org/10.1145/1631272.1631459>