
madmom Documentation

Release 0.17.dev0

madmom development team

Jan 06, 2022

Contents

1	Introduction	1
2	Installation	3
3	Usage	7
4	Tutorials	9
5	Development	11
6	Citation	13
7	madmom.audio	15
8	madmom.features	17
9	madmom.io	19
10	madmom.ml	21
11	madmom.utils	23
12	madmom.processors	25
13	madmom.evaluation	27
14	Indices and tables	29
15	Acknowledgements	31

CHAPTER 1

Introduction

Madmom is an audio signal processing library written in Python with a strong focus on music information retrieval (MIR) tasks. The project is on [GitHub](#).

It's main features / design goals are:

- ease of use,
- rapid prototyping of signal processing workflows,
- most things are modeled as numpy arrays (enhanced by additional methods and attributes),
- simple conversion of a workflow to a running program by the use of processors,
- no dependencies on other software packages (not even for machine learning stuff),
- inclusion of reference implementations for several state-of-the-art algorithms.

Madmom is a work in progress, thus input is always welcome.

The available documentation is limited for now, but *you can help to improve it*.

Please do not try to install from the .zip files provided by GitHub. Rather install either *from package* (if you just want to use it) or *from source* (if you plan to use it for development). Whichever variant you choose, please make sure that all *prerequisites* are installed.

2.1 Prerequisites

To install the `madmom` package, you must have either Python 2.7 or Python 3.3 or newer and the following packages installed:

To install the `madmom` package, you must have either Python 2.7 or Python 3.5 or newer and the following packages installed:

- `numpy`
- `scipy`
- `cython`
- `mido`

In order to test your installation, process live audio input, or have improved FFT performance, additionally install these packages:

- `pytest`
- `pyaudio`
- `pyfftw`

If you need support for audio files other than `.wav` with a sample rate of 44.1kHz and 16 bit depth, you need `ffmpeg` (`avconv` on Ubuntu Linux has some decoding bugs, so we advise not to use it!).

Please refer to the `requirements.txt` file for the minimum required versions and make sure that these modules are up to date, otherwise it can result in unexpected errors or false computations!

2.2 Install from package

The instructions given here should be used if you just want to install the package, e.g. to run the bundled programs or use some functionality for your own project. If you intend to change anything within the *madmom* package, please follow the steps in [the next section](#).

The easiest way to install the package is via `pip` from the [PyPI \(Python Package Index\)](#):

```
pip install madmom
```

This includes the latest code and trained models and will install all dependencies automatically.

You might need higher privileges (use `su` or `sudo`) to install the package, model files and scripts globally. Alternatively you can install the package locally (i.e. only for you) by adding the `--user` argument:

```
pip install --user madmom
```

This will also install the executable programs to a common place (e.g. `/usr/local/bin`), which should be in your `$PATH` already. If you installed the package locally, the programs will be copied to a folder which might not be included in your `$PATH` (e.g. `~/Library/Python/2.7/bin` on Mac OS X or `~/.local/bin` on Ubuntu Linux, `pip` will tell you). Thus the programs need to be called explicitly or you can add their install path to your `$PATH` environment variable:

```
export PATH='path/to/scripts':$PATH
```

2.3 Install from source

If you plan to use the package as a developer, clone the Git repository:

```
git clone --recursive https://github.com/CPJKU/madmom.git
```

Since the pre-trained model/data files are not included in this repository but rather added as a Git submodule, you either have to clone the repo recursively. This is equivalent to these steps:

```
git clone https://github.com/CPJKU/madmom.git
cd madmom
git submodule update --init --remote
```

Then you can simply install the package in development mode:

```
python setup.py develop --user
```

To run the included tests:

```
python setup.py pytest
```

2.4 Upgrade of existing installations

To upgrade the package, please use the same mechanism (`pip` vs. `source`) as you did for installation. If you want to change from package to source, please uninstall the package first.

2.4.1 Upgrade a package

Simply upgrade the package via pip:

```
pip install --upgrade madmom [--user]
```

If some of the provided programs or models changed (please refer to the [CHANGELOG](#)) you should first uninstall the package and then reinstall:

```
pip uninstall madmom  
pip install madmom [--user]
```

2.4.2 Upgrade from source

Simply pull the latest sources:

```
git pull
```

To update the models contained in the submodule:

```
git submodule update
```

If any of the `.pyx` or `.pxd` files changed, you have to recompile the modules with Cython:

```
python setup.py build_ext --inplace
```


3.1 Executable programs

The package includes executable programs in the `/bin` folder. These are standalone reference implementations of the algorithms contained in the package. If you just want to try/use these programs, please follow the [instruction to install from a package](#).

All scripts can be run in different modes: in `single` file mode to process a single audio file and write the output to STDOUT or the given output file:

```
DBNBeatTracker single [-o OUTFILE] INFILE
```

If multiple audio files should be processed, the scripts can also be run in `batch` mode to write the outputs to files with the given suffix:

```
DBNBeatTracker batch [-o OUTPUT_DIR] [-s OUTPUT_SUFFIX] FILES
```

If no output directory is given, the program writes the output files to same location as the audio files.

Some programs can also be run in `online` mode, i.e. operate on live audio signals. This requires `pyaudio` to be installed:

```
DBNBeatTracker online [-o OUTFILE] [INFILE]
```

The `pickle` mode can be used to store the used parameters to be able to exactly reproduce experiments.

Please note that the program itself as well as the modes have help messages:

```
DBNBeatTracker -h
DBNBeatTracker single -h
DBNBeatTracker batch -h
DBNBeatTracker online -h
```

(continues on next page)

(continued from previous page)

```
DBNBeatTracker pickle -h
```

will give different help messages.

3.2 Library usage

To use the library, *installing it from source* is the preferred way. Installation from package works as well, but you're limited to the functionality provided and can't extend the library.

The basic usage is:

```
import madmom
import numpy as np
```

To learn more about how to use the library please follow the *tutorials*.

CHAPTER 4

Tutorials

This page gives instructions on how to use the package. They are bundled as a loose collection of jupyter (IPython) notebooks.

You can view them online:

https://github.com/CPJKU/madmom_tutorials

As an open-source project by researchers for researchers, we highly welcome any contribution!

5.1 What to contribute

5.1.1 Give feedback

To send us general feedback, questions or ideas for improvement, please post on [our mailing list](#).

5.1.2 Report bugs

Please report any bugs at the [issue tracker on GitHub](#). If you are reporting a bug, please include:

- your version of madmom,
- steps to reproduce the bug, ideally reduced to as few commands as possible,
- the results you obtain, and the results you expected instead.

If you are unsure whether the experienced behaviour is intended or a bug, please just ask on [our mailing list](#) first.

5.1.3 Fix bugs

Look for anything tagged with “bug” on the [issue tracker on GitHub](#) and fix it.

5.1.4 Features

Please do not hesitate to propose any ideas at the [issue tracker on GitHub](#). Think about posting them on [our mailing list](#) first, so we can discuss it and/or guide you through the implementation.

Alternatively, you can look for anything tagged with “feature request” or “enhancement” on the [issue tracker](#) on GitHub.

5.1.5 Write documentation

Whenever you find something not explained well, misleading or just wrong, please update it! The *Edit on GitHub* link on the top right of every documentation page and the *[source]* link for every documented entity in the API reference will help you to quickly locate the origin of any text.

5.2 How to contribute

5.2.1 Edit on GitHub

As a very easy way of just fixing issues in the documentation, use the *Edit on GitHub* link on the top right of a documentation page or the *[source]* link of an entity in the API reference to open the corresponding source file in GitHub, then click the *Edit this file* link to edit the file in your browser and send us a Pull Request.

For any more substantial changes, please follow the steps below.

5.2.2 Fork the project

First, fork the project on [GitHub](#).

Then, follow the [general installation instructions](#) and, more specifically, the [installation from source](#). Please note that you should clone from your fork instead.

5.2.3 Documentation

The documentation is generated with [Sphinx](#). To build it locally, run the following commands:

```
cd docs
make html
```

Afterwards, open `docs/_build/html/index.html` to view the documentation as it would appear on [readthedocs](#). If you changed a lot and seem to get misleading error messages or warnings, run `make clean html` to force Sphinx to recreate all files from scratch.

When writing docstrings, follow existing documentation as much as possible to ensure consistency throughout the library. For additional information on the syntax and conventions used, please refer to the following documents:

- [reStructuredText Primer](#)
- [Sphinx reST markup constructs](#)
- [A Guide to NumPy/SciPy Documentation](#)

If you use madmom in your work, please consider citing it:

```
@inproceedings{madmom,
  Title = {{madmom: a new Python Audio and Music Signal Processing Library}},
  Author = {B{\o}ck, Sebastian and Korzeniowski, Filip and Schl{\u}ter, Jan and
↪Krebs, Florian and Widmer, Gerhard},
  Booktitle = {Proceedings of the 24th ACM International Conference on
  Multimedia},
  Month = {10},
  Year = {2016},
  Pages = {1174--1178},
  Address = {Amsterdam, The Netherlands},
  Doi = {10.1145/2964284.2973795}
}
```


7.1 Submodules

7.1.1 madmom.audio.signal

7.1.2 madmom.audio.filters

7.1.3 madmom.audio.comb_filters

7.1.4 madmom.audio.stft

7.1.5 madmom.audio.spectrogram

7.1.6 madmom.audio.cepstrogram

7.1.7 madmom.audio.chroma

7.1.8 madmom.audio.hpss

8.1 Submodules

8.1.1 madmom.features.beats

8.1.2 madmom.features.beats_crf

8.1.3 madmom.features.beats_hmm

8.1.4 madmom.features.chords

8.1.5 madmom.features.downbeats

8.1.6 madmom.features.key

8.1.7 madmom.features.notes

8.1.8 madmom.features.notes_hmm

8.1.9 madmom.features.onsets

8.1.10 madmom.features.tempo

9.1 Submodules

9.1.1 madmom.io.audio

9.1.2 madmom.io.midi

10.1 Submodules

10.1.1 `madmom.ml.crf`

10.1.2 `madmom.ml.gmm`

10.1.3 `madmom.ml.hmm`

10.1.4 `madmom.ml.nn`

`madmom.ml.nn.layers`

`madmom.ml.nn.activations`

CHAPTER 11

madmom.utils

11.1 Submodules

11.1.1 madmom.utils.midi

11.1.2 madmom.utils.stats

CHAPTER 12

madmom.processors

13.1 Submodules

13.1.1 madmom.evaluation.alignment

13.1.2 madmom.evaluation.beats

13.1.3 madmom.evaluation.chords

13.1.4 madmom.evaluation.key

13.1.5 madmom.evaluation.notes

13.1.6 madmom.evaluation.onsets

13.1.7 madmom.evaluation.tempo

CHAPTER 14

Indices and tables

- `genindex`
- `modindex`
- `search`

CHAPTER 15

Acknowledgements

Supported by the European Commission through the [GiantSteps project](#) (FP7 grant agreement no. 610591) and the [Phenicx project](#) (FP7 grant agreement no. 601166) as well as the [Austrian Science Fund \(FWF\)](#) project Z159.