

## Audio Features Workshop 2023

October 21<sup>st</sup> 2023

**Due date:** Monday November 6<sup>st</sup> 2023, 23.59h.

Submit your answers in a single zip using Brightspace.

### Introduction

For Audio Analysis there exist several interesting Python modules/libraries, such as PyAudioAnalysis, Yaafe (somewhat older), OpenSmile, and LibROSA (see for many other examples <https://wiki.python.org/moin/PythonInMusic> ).

During this workshop we explore some of the functionality of the LibROSA Library, by studying several of the example codes that are available on the [LibROSA web-site](#). It is demonstrated how the library can be used for beat tracking and voice separation. There are many more methods implemented in LibROSA to extract low- and high-level features that can subsequently be used in various MIR applications.

### Preparations (on a Unix Machine):

1. Download the *librosa\_gallery.zip* (see note) from the API2023 web site and save and unzip it in your home directory. This will create a new subdirectory *librosa\_gallery* with several example scripts and audio files. Note that, further information on LibROSA can be found on: <http://librosa.github.io/librosa/> .

Note:

- Use the command from a terminal:  
wget [https://liacs.leidenuniv.nl/~bakkerem2/api/librosa\\_gallery.zip](https://liacs.leidenuniv.nl/~bakkerem2/api/librosa_gallery.zip)

2. Make your virtual environment:
  - go to your home directory and issue the commands:  
virtualenv api --python=python3  
source ./api/bin/activate  
pip install update pip  
pip install librosa  
pip install matplotlib==3.2  
pip install pyqt5==5.15.7

Notes:

- Python3.7 or higher required. The latest version of matplotlib may cause problems for which you need to change some of the scripts.
- **pip install matplotlib** installs the latest version of matplotlib. You may have to handle deprecated parameters and function calls.
- To stop the virtual environment issue the command: **deactivate**
- [ffmpeg](#) can be installed to extend the number of formats Librosa's input function *audioread* can handle.

3. Execute your first LibROSA sample script:

- Create a file `beat_tracking.py` in the directory `librosa_gallery` with code:

```
# Beat tracking example
from __future__ import print_function
import librosa

# 1. Get the file path to the included audio example
filename = librosa.example('vibeace', hq=True)
# 2. Load the audio as a waveform `y`
# Store the sampling rate as `sr`
y, sr = librosa.load(filename)
# 3. Run the default beat tracker
tempo, beat_frames = librosa.beat.beat_track(y=y, sr=sr)
print('Estimated tempo: {:.2f} beats per minute'.format(tempo))

# 4. Convert the frame indices of beat events into timestamps
beat_times = librosa.frames_to_time(beat_frames, sr=sr)
```

To execute the script, issue the command:

```
python beat_tracking.py
```

NB Successfully installing Librosa and executing Librosa python code is part of this workshop. If after a reasonable effort, you did not succeed in completing the above preparations, please contact me ( [erwin@liacs.nl](mailto:erwin@liacs.nl) ) to resolve any issues you encountered.

### **Assignment 1:**

Listen to: `$HOME/librosa_gallery/examples/audio/Kevin_MacLeod_-_Vibe_Ace.mp3`  
Determine the tempo of this track in beats per minute. Report (in a pdf) both the calculated estimated beats per minute and the tempo of the track you determined by listening to it. Do the same for either *track01*, or *track02*, or *track03* in the *examples/audio* directory.

Note: Other recordings can be found on: <https://librosa.org/doc/latest/recordings.html>

## Assignment 2

Go to the directory `$HOME/librosa_gallery/examples`

Note that local to this directory an audio directory exists in which you can find several mp3 audio files that are used by the different example scripts we will use.

Execute the script for harmonic-percussive source separation: `plot_hprss.py`. See [https://librosa.org/doc/latest/auto\\_examples/plot\\_hprss.html#sphx-glr-auto-examples-plot-hprss-py](https://librosa.org/doc/latest/auto_examples/plot_hprss.html#sphx-glr-auto-examples-plot-hprss-py) for further details on this script.

Use the script for the audio track *Kevin\_MacLeod\_-\_Vibe\_Ace.mp3*. Report Figure 1 for this track and share your observations when comparing it with the results from Assignment 1. Do the same for either *track01*, or *track02*, or *track03*.

Note:

- You may also download and use the examples, Python-scripts or Jupyter Notebooks, from: <https://librosa.org/doc/latest/advanced.html>
- You may want to update your `plot_hprss.py` code so that loading the audio clip does not give any warnings.

## Assignment 3

Study and execute the script `plot_vocal_separation.py`. It uses the following audio track: *Cheese\_N\_Pot-C\_-\_16\_-\_The\_Raps\_Well\_Clean\_Album\_Version.mp3*. Adapt the script such that you produce an audio track (in the wav-, or ogg-format) with the vocals only. Apply your script also to *track04*.

Note:

- You may want to update your `plot_vocal_separation.py` code so that loading the audio clip does not give any warnings.

**NB** Submit your answers in a single .zip file using Brightspace on Monday November 6<sup>th</sup> 2023 23.59h.

## References and Links:

- pyAudioAnalysis: <https://github.com/tyiannak/pyAudioAnalysis>
- LibROSA: <http://librosa.github.io/librosa/>
- Yaafe: <http://yaafe.sourceforge.net/>
- OpenSmile: <https://www.audeering.com/opensmile/>
- Python in Music: <https://wiki.python.org/moin/PythonInMusic>
- Audacity (open source audio software) <https://www.audacityteam.org/>