BPM & Time-Signature Detection Toolkit

We made an audio-analysis toolkit for estimating **tempo (BPM)** and **time signature** from music recordings. It is a Python project designed as a command-line application, with MP3 input support.

The toolkit consists of a BPM/beat-tracking module and a time signature detector: the first locates the beats in the audio and estimates the tempo, while the second analyzes the beat-synchronous audio, using as input the bpm estimated by the first module, to infer the time signature (e.g., 4/4, 3/4, 6/8). Both modules support two different modes:

BPM

- A **normal** mode that uses dynamic-programming beat tracker.
- A **light-weight** mode that uses a lower sample rate and larger hop size to estimate tempo quickly.

Time Signature

- A **standard** mode, which assumes the whole song has a single time signature.
- A **varying** mode, which allows the time signature to change over time.

The implementation of the tempo tracker module is based on an onset envelope computed using spectral-flux techniques. Other possible approaches to compute the onset detection, including wavelet-based methods, were also investigated, but they did not provide any improvements. The light-weight mode, compared to the normal one, speeds up the computation and cuts some of the memory usage, proving to be very effective for longer audio files with consinstent tempo and meter.

It is interesting to point out that the toolkit can offer useful insights on the rhythmic structure of a song even when it returns an incorrect (according to the standard reading of the song) tempo or meter: this is due to the fact that there is no precise definition of what a "beat" is, and time signatures can be interpreted in different ways depending on the musical context.