ECS7006 Music Informatics 2020: Coursework 1

Deadline: Coursework 1 must be submitted on QMPlus by **10:00am Mon 9 March**. Late submissions will be penalised according to the EECS policy specified in the relevant student handbook.

There are two coursework assignments for ECS7006, each of which is worth 15% of the total mark for the module. For coursework 1, your task is to implement and test a beat tracking system for ballroom dance music. You may take a published paper and attempt to reimplement it, such as one of the approaches mentioned in lectures, or you may develop your own ideas, inspired and informed by the research literature. You must submit a ZIP file containing the python code for your solution (documented sufficiently so that it is easy for the marker to run the code), and a single PDF file (maximum 5 pages) containing a report describing and evaluating your beat tracking system.

To aid in automatic testing, your code must be callable as follows:

```
beats, downbeats = beatTracker(inputFile)
```

which is given the path name of an audio file (`inputFile`), and returns a vector of beat times (`beats`) in seconds and (optionally, for bonus marks) a vector of downbeat times (`downbeats`) in seconds. The beat times should correspond to the tactus or primary metrical level. The downbeat times should correspond to the first beat of each bar. Rather than aiming for a system that works for all styles of music, you are asked to focus on ballroom dance music.

To aid in developing and testing your code, you can download the 698 audio clips of the Ballroom dataset, which are available from:

`http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html`

Note that the link to the dataset is broken. It should be:

`http://mtg.upf.edu/ismir2004/contest/tempoContest/data1.tar.gz`

Note also that this is around 1.7 GB of audio data. You can use a subset if this is too much data to handle with your method. Ground truth annotations of beats and downbeats for this dataset are provided by Florian Krebs here:

`https://github.com/CPJKU/BallroomAnnotations`

Finally, a library for evaluation is available here:

`https://craffel.github.io/mir_eval/`

Please note the following points:

- All software is to be written in Python 3, using the standard libraries and versions that we are using in labs and tutorials (i.e. we should be able to run it easily).
- Make sure you follow the specifications in this document.
- You should write a report (max 5 pages, PDF format only) explaining your solution (assumptions, theory, design and high-level implementation, but not including the code), how you evaluated it, and pointing out its strengths, weaknesses, and any known bugs.
- Submit the code, report, and any relevant data in a ZIP file (only).
- The code and data must be sufficiently well documented for the marker to understand and evaluate it.
- Marks will be awarded as follows: report including algorithm and evaluation (60%), code quality including documentation (20%), results (20%)

If you need any clarifications regarding the practical work, please ask in lectures or via email.