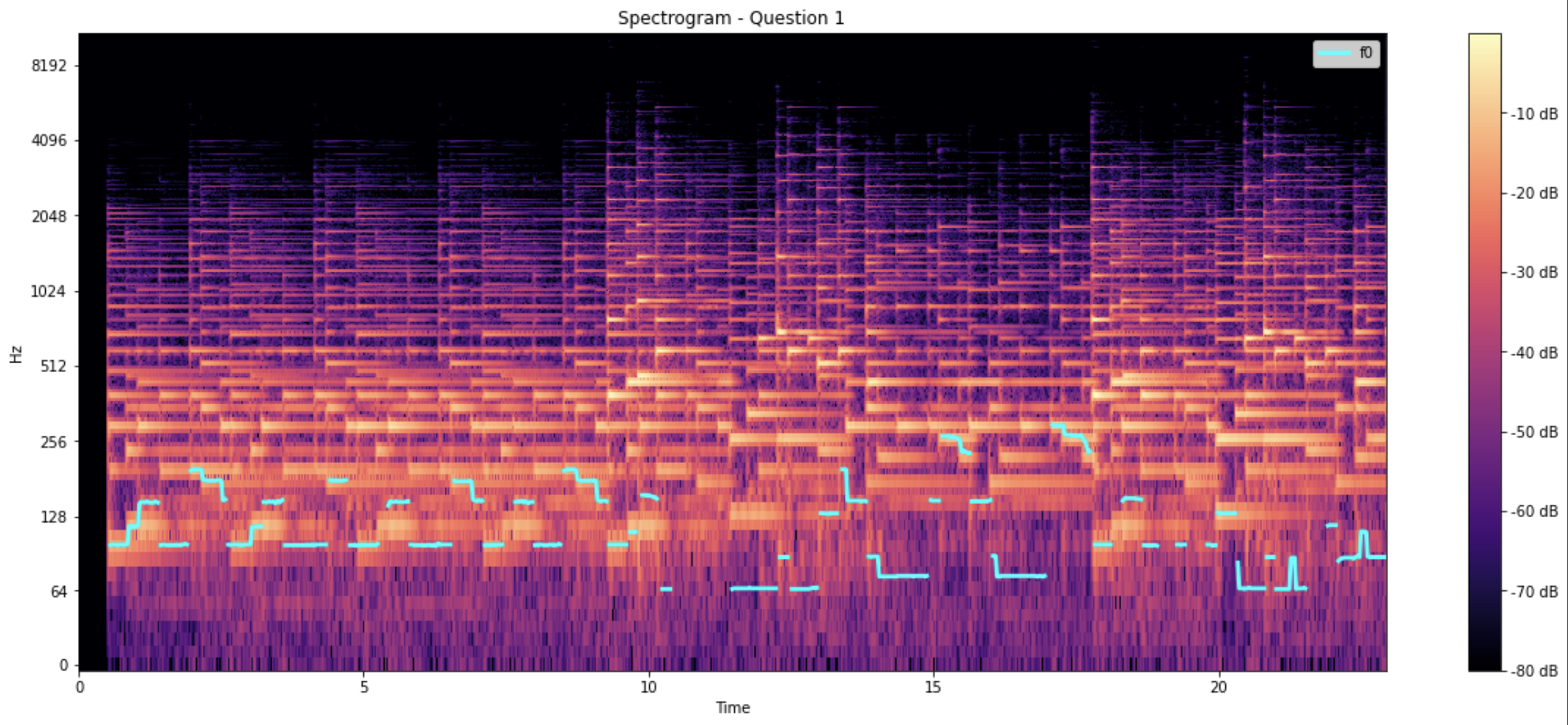
MIR - HW #1  
Audio Signal Processing & Musical Key Detection

Question 1.

一張含有 圖表 的圖片

自動產生的描述



This song[[1]](#footnote-1) is fetched from the theme song of LaLa Land, and it’s 36 seconds long. I find a piano cover, so it should be easier to read off the spectrogram. In this recording, we can clearly see the f0 in light blue line and is playing off the left hand part on the music sheet. However, by listening to the song, we can notice that G3 (196.00 Hz) is detected as G2 (98.00 Hz). As right hand joins around 11s, lower frequency is recognized (around C2).

By simply reading the strong note (highlighted in yellow), we can see the melody of both left and right hands. For example, in bar 6 (around 14s), the note is at E5 (659.25 Hz) and F5 (698.46 Hz).

1. Given and , so we know that . Therefore, the IF of x2 will be .
2. Here is how I produce the audios

一張含有 文字 的圖片

自動產生的描述

For x1, the sound is increasing throughout the first 6.5 seconds, and decreasing afterward. As for x2, the sound is like oscillating, increasing and decreasing, and is more observable in the end.

1. Here are the two spectrograms for x1 and x2. The first spectrogram does look like how the audio sound to me, but it didn’t match the idea of keep increasing like a bird’s chirp. The second spectrogram look like how it sound to me. The frequency oscillates throughout that 10 seconds.

一張含有 圖表 的圖片

自動產生的描述一張含有 文字, 裝置, 螢幕擷取畫面 的圖片

自動產生的描述

Question 2.

1. For Global Key detection, I used K-S detection with late fusion. And here is the result I got after the prediction.

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

|  |  |  |  |
| --- | --- | --- | --- |
|  | stft | cqt | cens |
| RA scores | 0.625 | 0.5833 | 0.5417 |
| WA scores | 0.6875 | 0.6958 | 0.6458 |

It looks like stft have the best result in general, it has the average of 0.656. The CQT and cens method are interesting, it is likely to detect a chord as something else (relative/perfect-fifth/parallel), and this is why they have a large difference between RA scores and WA scores.

1. For local key detection, I also used the same approach. But for the frames extraction, I use the method as described in the questions with 15 seconds before the detected time and 15 seconds after it, which is 30 seconds in total.

一張含有 文字 的圖片

自動產生的描述

I found that length of y is multiple of sampling rate and the music length. So I cut the y to a smaller duration (30 seconds) tmp\_y.

And for the RA and WA score, I decided to check my estimations one by one with the ground truth file. For example, if the estimation at 10th second is within that (start ~ end) range in the csv file, then I check the estimation correspondingly.

一張含有 文字 的圖片

自動產生的描述

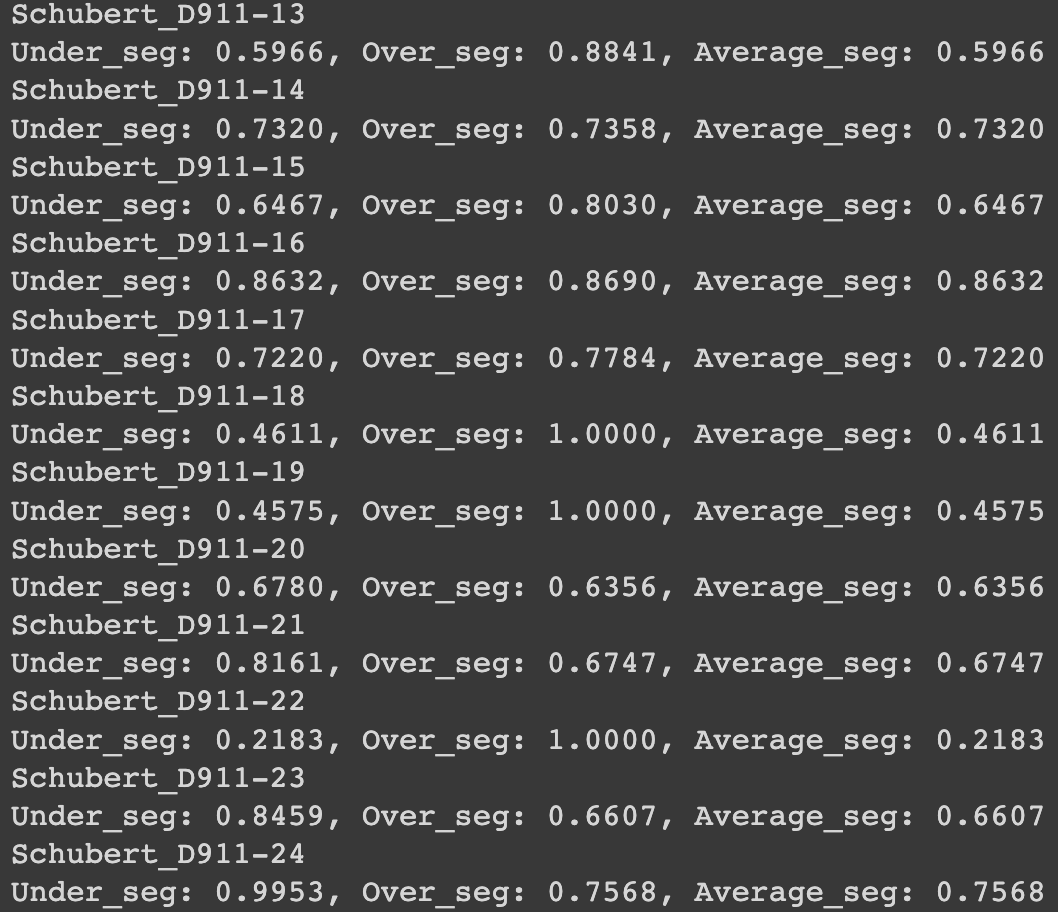
By doing so, I can get the index of the correct frame on the csv file.

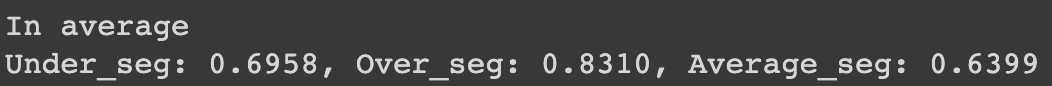
|  |  |  |  |
| --- | --- | --- | --- |
|  | stft | cqt | cens |
| RA scores | 0.4263 | 0.4940 | 0.4741 |
| WA scores | 0.5191 | 0.6080 | 0.5880 |

It is interesting that stft performs the best in global key detection while cqt and cens perform better in local key detection. I think they are predicting more accurately on the music with shorter duration.

1. I assume that the small segments with the same chord should be in a same segment, so I combine them into a big one and extract the start and end time.

一張含有 文字 的圖片

自動產生的描述 



It looks like the segmentations I made were not precise, however, over-segmentation still has a satisfying performance overall.

1. <https://www.youtube.com/watch?v=oAXXSFKfjLs&ab_channel=TheTheorist> [↑](#footnote-ref-1)