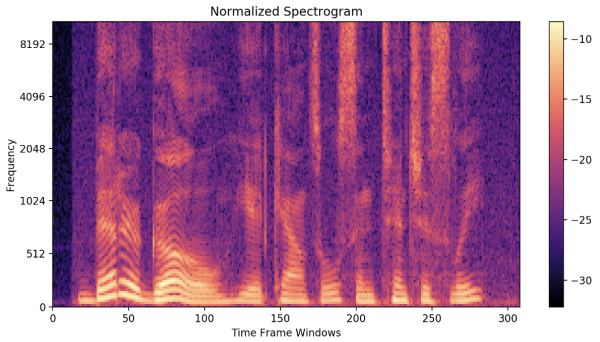
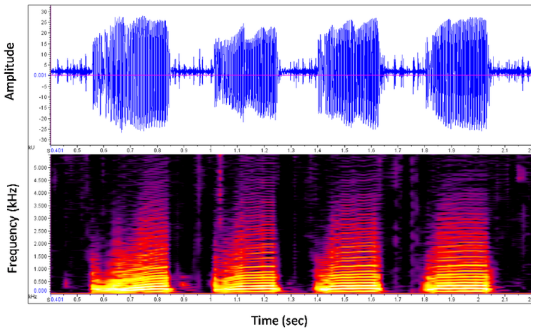
Qualitative Eclectic Art - Project Proposal

Quantitative Engineering Analysis 2

Daniel Arnott and Samuel Cabrera Valencia



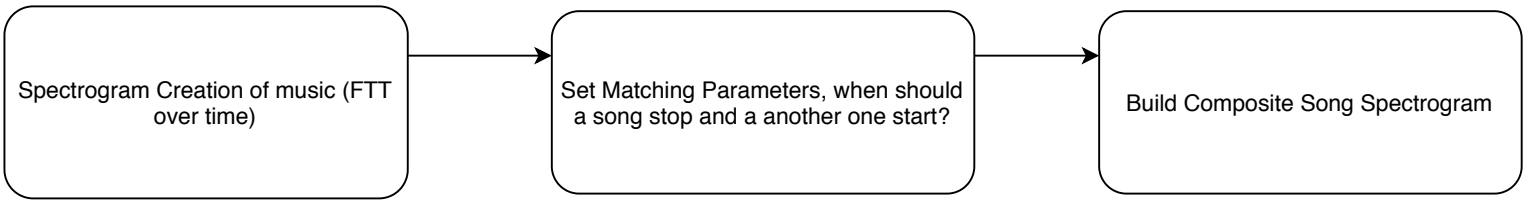


Governing Question: Can matching time frame slices of a song's spectrogram be used to sync up music?

In this project we want to explore the possibility of using spectrogram analysis and different signal processing techniques to see if we can create a program that can automatically find the best fit song to mash up with a user-inputted sing given the frequencies at a defined moment in the user-inputted song. This is very similar to the technology that the song finding app "Shazam" uses except for some key differences in implementation (2). Shazam procedurally generates a spectrogram based off the audio input it is getting from the phone's microphone and finds the song that most fits with this generated spectrogram through a large library of spectrograms that the app has saved. In our case, we will likely have a smaller library for mixing songs, and we, as users, will decide at what point in the inputted song that we want to select a new song to be "mashed in." This will be more similar to the "AutoMasherUpper" detailed in the research paper cited below by a research team at the National Institute of Advanced Industrial Science and Technology (AIST), Japan (1).

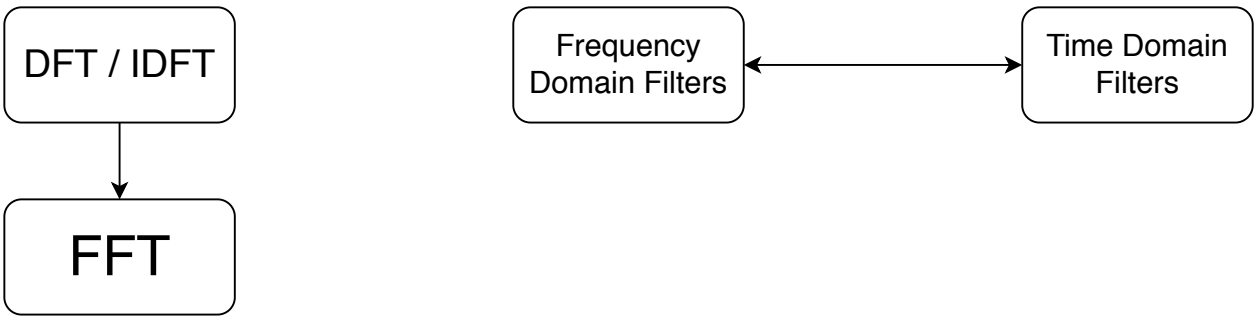
Our value add for this proposal is mainly rooted in the entertainment value that it will give users when they are making their own music. While it can be said that this tool would also add value by assisting users who might be less comfortable in music composition, the final result will more than likely be a primitive music mash-up tool (due to time spent / scoping of the project) and hence might not add the most value in this manner.

General Idea:



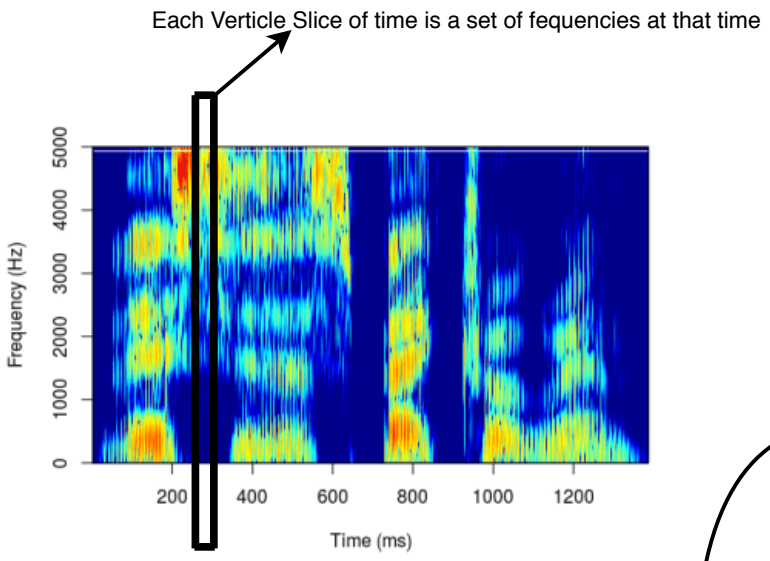
Connection to Course Material

In Module 3 of QEA 2 we have learned of a number of signal processing techniques that will help us significantly in this project as this is mainly a signal processing project.



The initial step of our algorithm is taking spectrograms of the music data that we are taking in. Spectrograms are essentially multiple Fourier Transforms over time (in this case the song length). These plots show how prevalent certain frequencies are in a given time step by adding a 3rd degree of visualization - a heat map.

Since we're generating spectrograms (which are frequency visualizations over time, in this case over the time of our songs), both frequency domain filters and time domain filters that we developed in Assignment 5 will be extremely helpful if there are certain frequencies / times that are causing inaccuracies in our signal processing.



How To Find the Distance

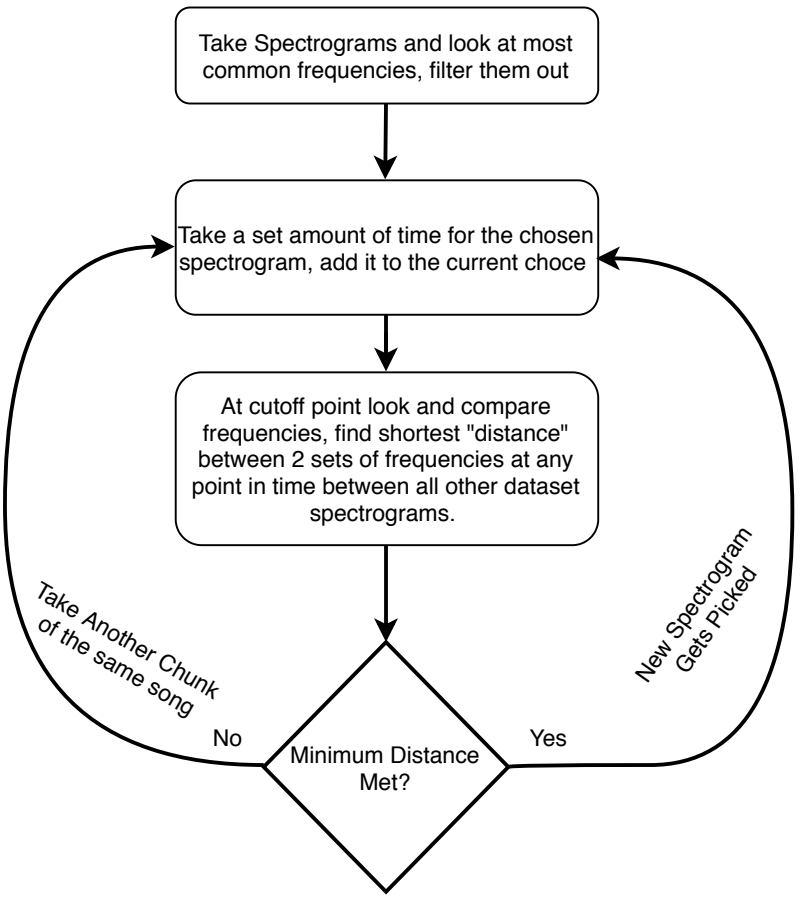
between two songs?

For a generalized formula, we can simply take the difference in value for an i-point DFT at each point I. For each difference, square it, add all of these together and take the square root. We can experimentally determine what a minimum "threshold" value is for this distance and use it as a cutoff for deciding wether to change the song or keep playing it.

How Will We measure Success?

If we can produce a final product that can satisfactorily splice songs in coherent ways, where they subjectively sound nice, we will consider this a success

Algorithm Idea



Sam Learning Goals:

-Write cleaner, functional Matlab implementation

-Get a better grasp of when to use specific filtration techniques

-Better Understand the relationship between frequency and types of music

Daniel Learning Goals

- Improve my workflow for projects in an online setting w/ a partner that I can't interact with in person

- Using MATLAB code from previous classwork and structuring it to fit the needs of the project.

- Placing myself in more situations where I am more unsure of the material to correct my misunderstandings with it

References

(1) Davies, Matthew & Hamel, Philippe & Yoshii, Kazuyoshi & Goto, Masataka. (2014). AutoMashUpper: Automatic Creation of Multi-Song Music Mashups. Audio, Speech, and Language Processing, IEEE/ACM Transactions on. 22. 1726-1737. 10.1109/TASLP.2014.2347135.

(2) Cooper, Trey. “How Shazam Works.” *Medium*, Medium, 29 Jan. 2018, medium.com/@treycoopermusic/how-shazam-works-d97135fb4582.

(3) carykh. “Computer Tries to Replicate My Voice!” *YouTube*, YouTube, 15 Aug. 2017, www.youtube.com/watch?v=jSsMqjMcRAg.

(4) carykh. “AI Evolves to Compose 3 Hours of Jazz!” *YouTube*, YouTube, 4 July 2017, www.youtube.com/watch?v=nA3YOFUCn4U.