

Report from Brainhack

Missing Title

URL is missing

1 Introduction

Resting-state fMRI (rsfMRI) data generates interesting time courses with unpredictable hills and valleys. This data in some degree resembles the notes of a music scale. Taking advantage of these similarities and using only the rsfMRI data as input, we use basic rules of music theory to transform the data into pleasant music. Our project is implemented in Python using the [midiutil library](#).

2 Approach

Data: We used open rsfMRI from the ABIDE dataset [?] preprocessed by the Preprocessed Connectomes Project [?]. We randomly chose 10 subjects preprocessed using C-PAC pipeline [?] with 4 different strategies. For reducing the data dimensionality, we chose the CC200 atlas [?].

Processing: fMRI time courses were analysed to extract pitch, tempo, and volume - 3 important attributes for generating music. For pitch, we mapped the time course amplitudes to MIDI values in the range of 36 to 84, corresponding to piano keys in within a pentatonic scale. The key of the scale is determined by the global mean ROI value (calculated across all time-points and ROIs) using the equation: $(\text{global signal} \% 49) + 36$. The lowest tone that can be played in a certain key is calculated from $(\text{key} \% 12) + 36$. The set of tones that can be played are then determined from the lowest tone using a scale. For example, the minor-pentatonic scale's set of tones is calculated by adding [0, 3, 5, 7, 10] to its lowest tone, then skipping to the next octave, and then repeating the process until the value 84 is reached. An fMRI time course is mapped to these possible tones by scaling its amplitude to the range between the smallest and largest tones in the set. If a time point maps to a tone that is not in the set, it is shifted to the closest allowable tone. An example of allowed set of tones is shown in Figure 1.

For tempo, we use the first temporal derivative for calculating the length of notes, assuming we have 4 lengths (whole, half, quarter and eighth note). In the time course, if the modulus distance between time point t and $t + 1$ is big, we interpret it as a fast note (eighth). However, if the distance between t and $t + 1$ is close to zero, we assume it is a slow note (whole). Using this approach, we map all the other notes in between.

We use a naive approach for calculating volume in a way that tackles a problem we have with fast notes: their sound is cut off due to their short duration. A simple way to solve this is to decrease the volume of fast notes. Thus, the faster the note, the lower the volume. While a whole note has volume 100 ([0,100]), an eighth note has volume 50.

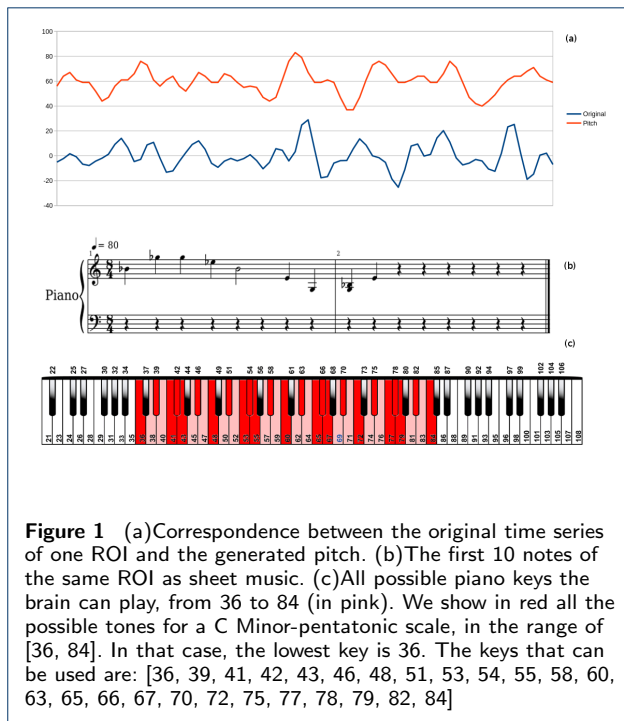
Finally, we choose the brain regions that will play. Users complain when two similar brain regions play together. Apparently, the brain produces the same music twice. However, when the regions are distinct, the music is pleasant. Thus, we use FastICA [?] for choosing brain regions with maximally uncorrelated time courses.

3 Results

A framework for generating music from fMRI data, based on music theory, was developed and implemented as a Python tool yielding several audio files. When listening to the results, we noticed that music across subjects is different. However, music generated by the same subject (4 preprocessing strategies) sounds similar. Our results sound different from music obtained in a similar study using EEG and fMRI data [?].

4 Conclusions

In this experiment, we established a way of generating music with open fMRI data following some basic music theory principles. This resulted in a somewhat naïve but pleasant music for humans. The results show an



interesting possibility for providing feedback of fMRI activity for neurofeedback experiments.

Availability of Supporting Data

More information about this project can be found at: *URL is missing*

Competing interests

Competing interests statement is missing.

Author's contributions

Author's contributions statement is missing.

Acknowledgements

Acknowledgements section is missing.