Finding Notes Within Boomwhackers Recordings

Anıl Özen

Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente

Abstract

This work finds notes within audio recordings of boomwhackers play. The project can find combinations of up to 4 notes within input signals. This is done by comparing the input signal with the recordings of the notes within its library through cross correlation check of both signal's fourier transforms. The system yields results with high accuracy on the signals of combinations of up to 4 notes.

Keywords— Boomwhackers, note finding, fft analysis, notes correlation

Contents

1 Introduction

In the Netherlands, primary school teachers, often without thorough musical education themselves, are given the task of teaching music to children. As teachers lack the formal training on music they also lack the self-efficacy to teach what they know [1]. This project aims to assist teachers by identifying notes from recordings within the classroom.

In this project I will focus on the percussion instruments, boomwhackers[1] in particular. Boomwhackers are the color coded hollow tubes with varying lengths with each having unique sound properties. Boomwhackers are getting more popular within the classroom as they are easy to use and have low price. Despite these advantages, there are not many researches around this instruments on processing signals. Also, there are no repositories open to public for researchers to develop and validate systems. Boomwhackers signal processing is still an emerging field. The target is to gather audio from microphone and extract higher meaning from raw data. In this case, the extraction is the notes contained within the signal.

To find notes within boomwhackers recordings, I propose two research questions with two sub questions:

- How can we find notes within a recording of boomwhackers
  - How can we classify a single note within boomwhacker recordings
  - How can we classify up to 4 concurrent notes
- How can this system be robust, so it can work in different environments

The proposed method searches for the notes within the given recording, by comparing the notes it has within its database. Input signals are taken to frequency domain through fast fourier transform (FFT) and compared with each note's FFT product by cross correlation check (CCC). If the signal has enough similarity it is considered to contain the note. It can find singular notes within the recordings without any problem. With multiple notes (3 and 4), it can find the notes if they are not neighbours with each other (e.g. Fa is a neighbour of Mi and Sol).

[1] `www.boomwhackers.com`

2 State of Art

2.1 Beat Extraction From Percussion

One way to extract the beats is through building neural networks. Sun et. al. used a long short-term memory neural network approach to extract beats from a recording[2]. Such system consists of three layers; forgetting information, update status, and decision output. Forgetting gate is being used to eliminate (forget) unwanted information over time. Update status has two parts, the incoming input and the current state of the network. Both are used to update the network. Each cells initially performs a filtering on the current cell state. Current cell state is processed by a sigmoid layer and its output multiplied by a tanh function that was applied on a previous cell. The layer involving these operations is called the decision layer. The overall system can be seen in Figure 1. To obtain the music beats, this system requires music database. Each file obtained from the songs provided by Echo Nest, has the feature analysis and metadata of the songs, not the audio itself. While this approach has high accuracy (0.946) and can work online, it relies on training through a database. This can prove troublesome to find the songs played by boomwhackers as they are not mainstream instruments in the music industry.

While previous work quantify the beats among 5 power levels to distinguish the beats, another approach works in frequency domain on harmonics[3]. The signal is converted to frequency from 100 to 900 Hz (the target instrument in this project is Mridanga, an ancient percussion instrument used in Carnatic music). In boomwhackers this frequency range would be 256 to 512Hz[2] as seen in table 1. This operation creates a filter of energy in harmonics. The 100 to 900 Hz range is grouped by adding the base, first and the second harmonic frequencies. The filter bank by applying harmonic grouping cepstral coefficients creates two overtones as seen in formula 1. Transforming the frequency this way creates distinct beans for different strokes as seen in 4. The weights and biases are found out from training an artificial neural network with 3 layers. There is no mention on whether this system can work in real time or not.

equation 1

$$B(x) = \sum_{n=1}^{n=3} Feq(x * n) + Bias(n) \tag{1}$$

Another project do the beat tracking in a predictive manner with memoization to decrease the computational cost, as its one metric is the project to be able to run on a Raspberry Pi[4]. This system can run in real-time with an average continuity score (AMLt) of 0.67. The system starts with 4 seconds of audio as input to initialize, and subsequently predicts and updates the estimation every 1 second. How the system works can be see in Figure 2. The system's output can be seen in Figure 3. As the results show, the accuracy of this system is less than a neural network. Also this system relies on a prediction where the beats do not vary much whereas a novice player's percussion can be highly irregular due to errors. Yet, this project is open source and can be tested without implementing straight from the start.

[2]`https://asc-mag-media.s3.amazonaws.com/datasheet/P7-7400_DS.pdf`

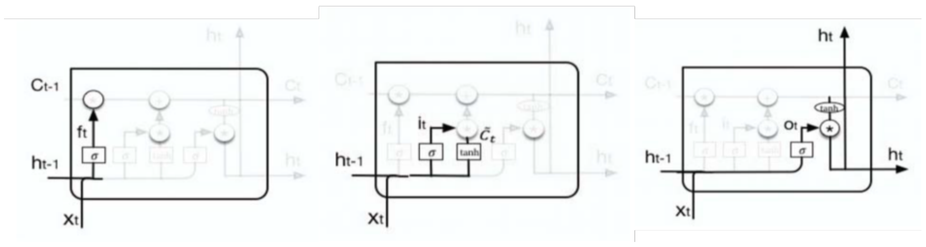


Figure 1: Overview of ltsm layers, on the left forgetting information, at the middle update status, on the right decision output

C 256 Hz	E 320 Hz	G 384 Hz
D 288 Hz	F 341.3 Hz	A 426.7 Hz
B 480 Hz	C 512 Hz	

Table 1: boomwhacer note frequencies

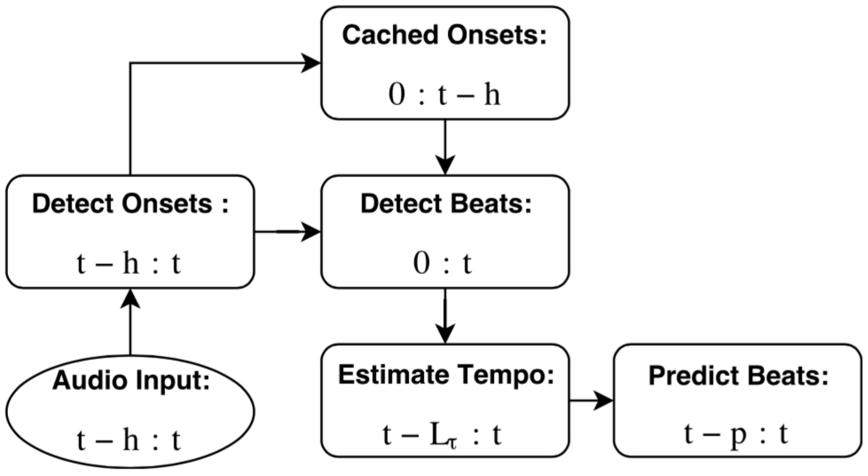


Figure 2: The flowchart of real time beat tracker RTBT

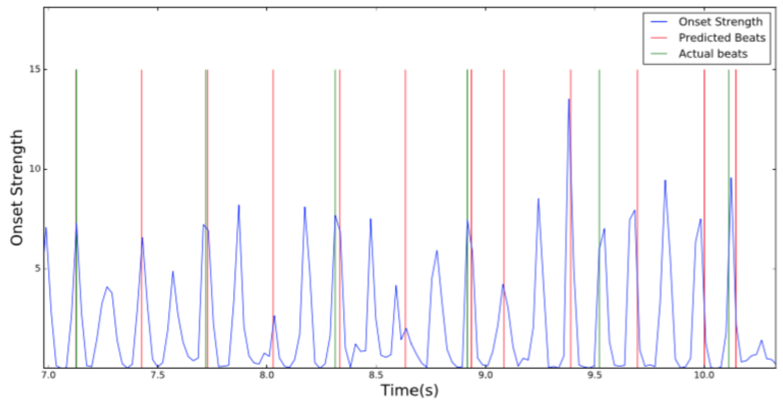


Figure 3: The output of RTBT

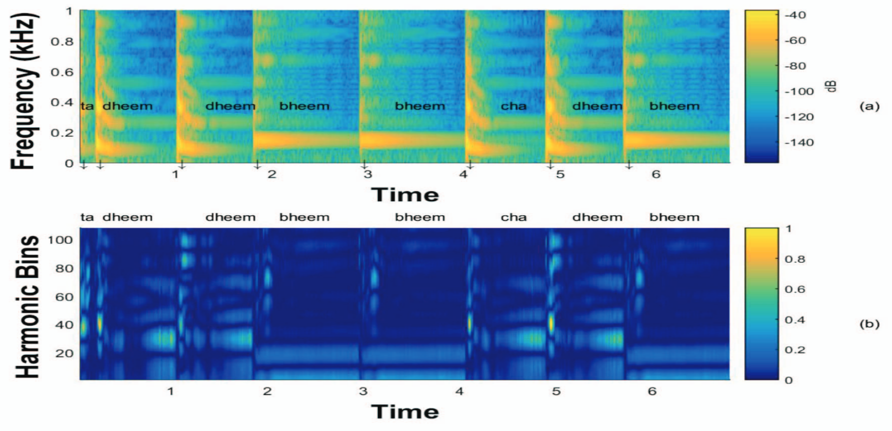


Figure 4: a) Strokes before transformation. (b) Stokes after transformation using Harmonic grouping.

3 Requirements

The main stakeholder Benno Spieker proposed multiple projects that I could pick one from. One of the problems was to track boomwhackers within the classroom when played by children. In Pabo environment, the target group who would play the instruments is the people who do not know how to play it, the novices including adults.

This project consists of multiple requirements. However, not all requirements are of the same importance. To prioritize the requirements, MoSCoW method is used [5]. With respect to the hierarchy of needs (must have at top, and won't have bottom) they are as follows:

3.1 Must have requirements

I focused on a single feature within the project, note recognition of boomwhacker recordings. Since the goal was focused around this problem, I wanted the solution to work well with high accuracy and be responsive to different environments I can simulate. Mr. Spieker gave me a playlist of different recordings with up to 3 different notes of boomwhackers being played concurrently. **So, the project should be able to annotate the input signals that includes up to 3 notes.**

Another requirement arose from the unique situation in which the project was being developed. As Covid-19 pandemic arrived to Netherlands, using the facilities within the university ceased to be possible and ordering hardware could be troublesome. To reduce the risks with the project, we decided to use common hardware we already have for the data acquisition as well as processing. So, any idea that would use external electronics such as accelerometer was discarded. **This project should use only available hardware without a need for ordering new parts.**

3.2 Should have requirements

As mentioned in Section 1, initial problem stemmed from the lack of budgets (employing class teachers instead of music teachers for music classes) so, the solution had to in a reasonable price. **There was no hard price limit, but it had to be reasonable.**

A classroom environment is a highly dynamic environment with many variables to consider. The amount of students within the classroom at a given time, the classroom layout, noise, windows being open, how students play the instruments all change the data within the recording. It is a good idea to be mindful about the dynamism within the problem and create a robust solution if possible. An over optimized solution can work in one scenario, and can fail in all the other scenarios. Such solution would be useless to the end user, if over optimization is rigid and integral to the product. The Covid-19 pandemic interfered with gathering data about the end user's work environment (the consevatorium). **That is why, the simulation of different environments had to be substitution within the limits of my home.**

3.3 Could have requirements

There are different possible scenarios how this project can be useful for the end users (the classroom teachers). An ideal product would pass through different developers so the interaction between the product and end-user is easy. Since classroom teachers would not know how to code, any tweaking within the program should not be through the code. This is especially important as the project required calibration from the user to be able to work with different environments. **Hence, using the product should not be a hassle for end user if possible.**

3.4 Won't have requirements

We can simulate the dynamism of the classroom through few controlled parameters. However, classrooms are highly dynamic and noisy environments and there is a big disparity between a classroom and a room in the house. A field test is necessary to see the performance and shortcomings of any project. Despite the necessity, it is not possible to do so, due to the Covid-19 pandemics. **The real life tests will not be conducted at this point.**

The system might benefit greatly from working online. If it were to work online, it can provide real time feedback on which notes are being played. However, it is out of scope at this point.

4 Ideation

At the start of the project we had a lot of freedom to explore possibilities on what we can develop. The base requirements was to create a system that would class teachers become better in teaching music. The other requirements, both internal and external, arrived later by talking to the stakeholder, as well as through introspection on how to build the system. Such iterative process is similar to the one explained in the manual created by our study [6].

We narrowed down the problem towards tracking and processing data from boomwhackers early. How the data would be gathered and how it would be processed was still an open question, allowing a wide design space.

4.1 How to gather data

One way to gather data was to utilize special hardware already in the market. There are products in the shelves such as SENSTROKE [3] that has built in sensors to track the quality of drum play. Such sensor is attached to the drumsticks and the play is recorded through sensors and communicates to its own app via bluetooth. While this sensor works on actual drumming experience, there are other products that works by *airdrumming*. One such product is PocketDrum [4]. Both devices can be seen in Figure 6. This device also communicates the data to its own app. Regardless of the scenario, these kinds of devices pose multiple problems. They are essentially blackboxes giving no control to user to tinker its innerworks, so there are no API's the users can use to create their own apps to fit into their unique needs. Also, the data is only accessible through the user interface of their own unique apps. It is not possible to access and process the data. Finally, they are not cheap, a single SENSTROKE sensor is €69 and $138 for a pair of PocketDrum sensor. Such price makes the project hard to be scalable.

One of the alternatives to 3rd party devices is to buy the sensors such as accelerometers and drive them through a micro controller such as Arduino Nano[5]. Even though the price is still not that cheap in the official store, there are many alternatives much cheaper from Chinese suppliers.

Opensource or propriety, I opted out of using such hardware with the concerns over limited both availability and shipment options due to Covid-19 pandemic. With such reality, I considered other more conventional options, chiefly using microphones to gather data. With microphones, the domain of data changes from mechanical side (orientation, speed, acceleration) to auditory side. In this domain it might be harder to infer how the tubes travel in the air and hit the target (handling of the instrument), it becomes easier to observe the outcome of the motion (the note that is being played). When it comes to music, auditory data has massive advantages. There are plenty of data on the internet open to public (not the case for boomwhackers but a general situation), libraries stemming from music information retrieval (commonly called MIR within its community), and research.

Microphones are easy to come by with hardware such smartphones and laptops. Third item I considered was raspberry pi with USB dongle for microphone. Smartphones have a unique advantage over other hardware. They are ubiquitous and is

[3]`https://www.senstroke.com/`
[4]`https://aeroband.net/pages/pocketdrum`
[5]`https://store.arduino.cc/usa/arduino-nano`

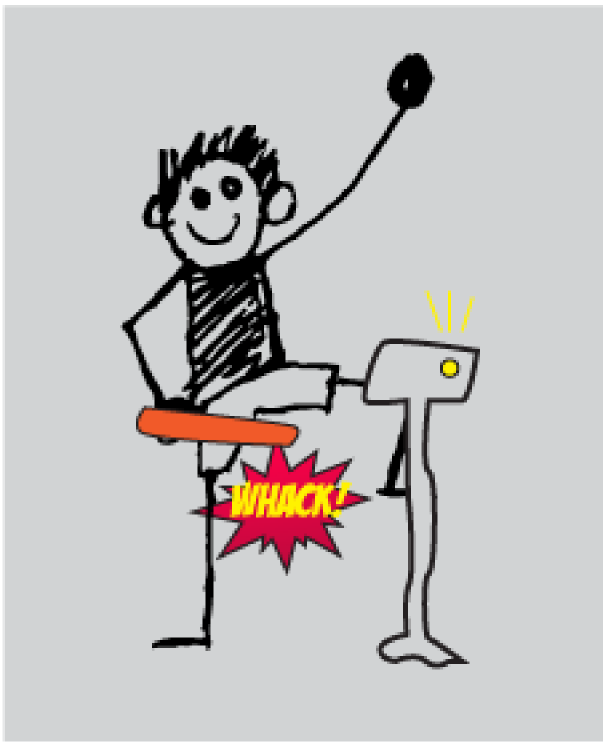


Figure 5: When the child whacks the boomwhacker smartphone blinks its flash in the back.

acquired by a large percentage of the population (92% of the children have smartphone in Netherlands[6]). Smartphones also has camera flash in the back. I considered the possibility of creating a system that tracks when the player creates a beat with boomwhacker and the smartphone (which is doing the tracking) flash its camera flash in the back which would be visible to teacher. The idea illustration can be seen in Figure 5

4.2 Processing of the data

There are many possibilities within the audio signal to process. We could process beat timings, tempo, play quality, the note that is being played. In the end we went for note annotation from the audio. To achieve this, I decided to code on Python as it has many libraries, easy to work on data, and has built-in plotter (handy for debugging). One big advantage of processing audio signals on boomwhackers is that it does not have many harmonics making the procession easier.

[6] `https://www.statista.com/statistics/946021/share-of-children-having-a-smartphone-mobile-phone-in-the-netherlands-by-age/`

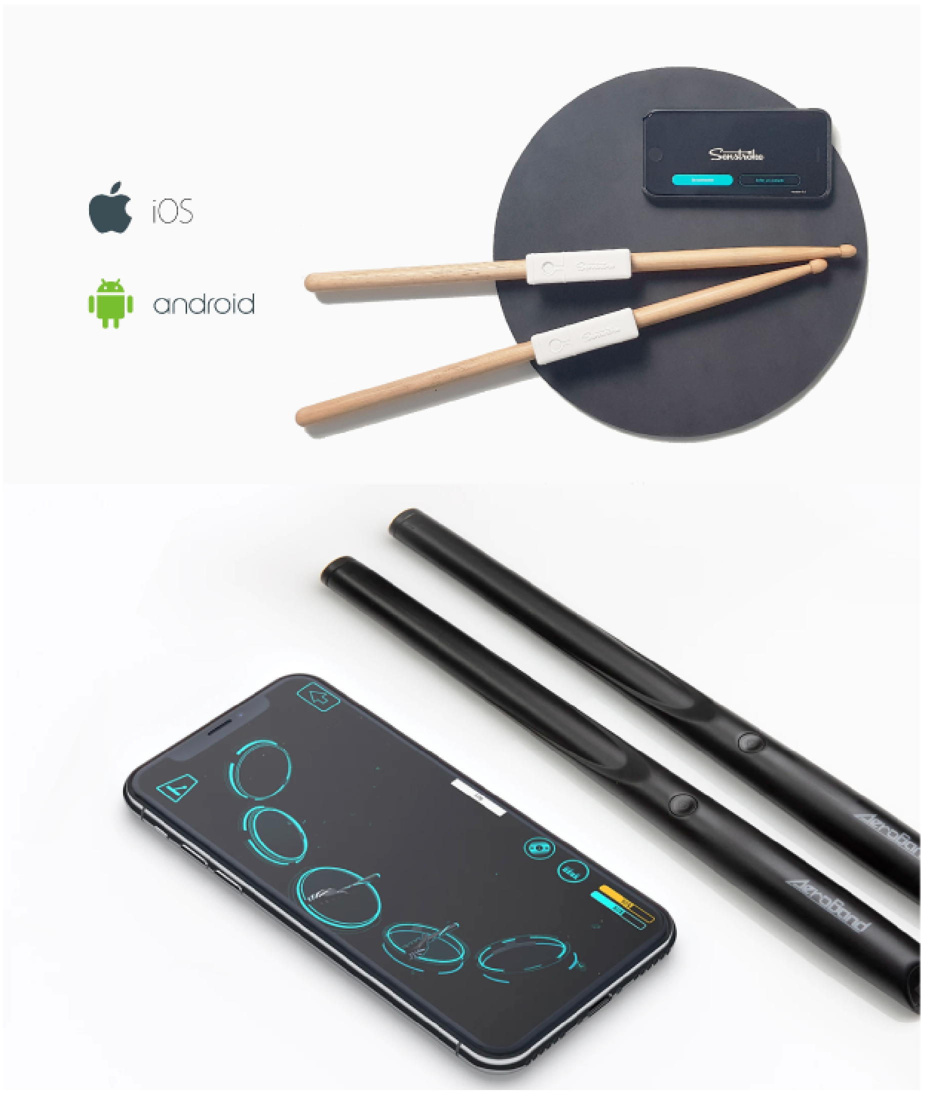


Figure 6: SENSTROKE on top, PocketDrum on bottom

5 Realisation

As mentioned before boomwhackers have really distinct patterns in terms of harmonics. Each tube has multiple harmonics and between the harmonics most of the signal is negligible. Boomwhackers have base harmonics only in the [200 600] Hz range as can be inferred from Figure 9. For each tube, the base harmonic is distinct and visible on frequency domain, but the higher harmonics are not always visible and are more dependent on how the tube is played. To test the harmonics 5 samples were taken for each note with reasonable play without emphasis on trying to play differently and their frequency analysis depicted in Figure 11. As seen in the signal even without deliberation the higher harmonics vary significantly. In some notes higher harmonics can disappear completely as seen in Figure 12.

Since higher harmonics are not always present, the focus shifted solely towards base harmonics of each boomwhacker. Base harmonics and their rise and fall stays within [200 600] Hz range further decreasing the analysed spectrum.

5.1 Peak Finding

Each note has a mountain like shape within the frequency domain around their base harmonics. With peak finding algorithms it is possible to find each peak. Python already has a peak detection library function (*scipy.signal.find_peaks*) with multiple parameters to limit the definition on peaks. With fine tuning the parameters, it is possible to find harmonics through the peak locations in the spectrum. Such a run can be seen in Figure 8

Since the peak heights vary massively from note to note, finding a threshold that encapsulates all the notes proved troublesome. This is an even bigger problem when the algorithm tries to distinguish the notes C (Do low and Do high). The first of harmonic of Do low peaks around 512 Hz, same as Do high. Just looking at signal peaks, the algorithm gets confused over the existence or non existence of Do low when Do high is present.

5.2 Correlation Check

Iterating and troubleshooting peak detection proved troublesome. Each note's peak heights vary both between the other notes and within the same note being played slightly differently. Despite high variance on peaks, the shapes stay similar. That is why a correlation check is performed on the test signals to test if base harmonics shapes exist within the signal. Correlation check compares two signals within set range and provides a result on [-1 1] with -1 negative correlation, 0 no correlation, and 1 complete correlation. Python has a library function under numpy to check the correlation (*numpy.corrcoef*)

Comparing signals through cross correlation check exploits the similarities within the note samples and differences between the notes. For this method to work, the base harmonics of each note need to be consistently similar to each other. Also, the correlation between different signals should stay low.

5.2.1 Narrowing the signal down to the bandwidth

In this project the notes are classified through their base harmonics. The samples outside the base harmonics bandwidth are not necessary to consider when doing the

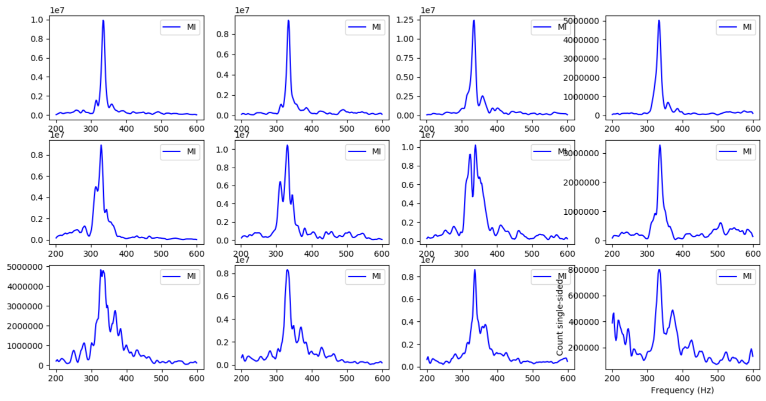


Figure 7: All the variations of note Mi played with its fourier transform in [200-600] Hz range. X axis is the type of play (cross legged, sitting, hitting on hand, hitting on tube by hand). Y axis is the type of the room the recording is taken (small room, mid size room, living room)

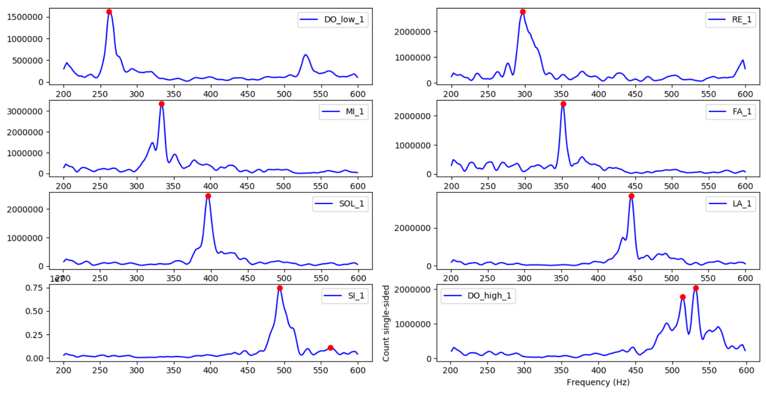


Figure 8: All the notes within the [200-600] Hz range with their peak values using a threshold of 1000000

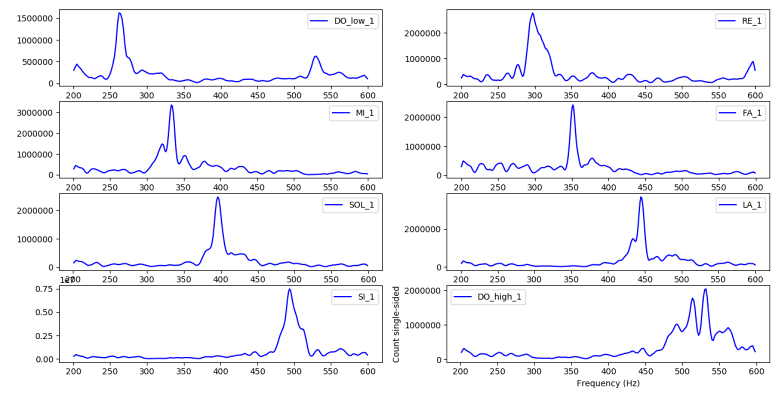


Figure 9: All the notes within the [200-600] Hz range

comparison with this approach. Each note's base harmonics frequency along with their bandwidth is depicted in Table 2. The bandwidths for each note is found through testing. The testing method will be mentioned in this chapter further ahead.

Leveling the signal: If both signals have similar shapes but their FFT values are different by a factor (e.g. signal A peaks on 1000 units and signal B peaks on 2000 units) cross correlation check produces low value, meaning the signals are different. So if we compare two of the same notes but are on different level they are also different according to the cross correlation check. To overcome such problem, one signal is taken to other signal's level using the peak value as a reference. The check is performed afterwards. The flow of operation can be seen in Figure 10.

5.2.2 Matching signal length for correlation check:

Correlation check requires the compared signals to be the same length. The notes vary on how many samples they have on the recording. When the signals combined together on Audacity, the combined signal has much more samples than singular notes.

To overcome the disparity, the first approach was to add zeros to the shorter signal until the lengths are the same. This simplistic method works fast and works good enough for comparing singular notes. However, when the difference is high adding zeros change the shape too much and correlation drops down for similar but highly different length signals.

The second approach was to expand the shorter signal to the length of longer signal by interpolation. The frequency side in interpolated linearly meanwhile the FFT side on y axis is interpolated in cubic. This method increased the performance on annotation of both singular and multiple notes.

To test the performance of two different methods using the methods multiple test are applied. The first test involves whether the method considers each recording played in the same session with same notes as similar enough. The second test is comparing different notes with each other to see if the method confuses the notes with each other. Finally, the third test is to check if the method can find all the notes within the triple

note name	base harmonics frequency	base harmonics frequency bandwidth
Do low (C)	256	[240-280]
Re (D)	288	[260-340]
Mi (E)	320	[320-380]
Fa (F)	341.3	[340-430]
Sol (G)	384	[380-460]
La (A)	426.7	[400-500]
Si (B)	480	[450-550]
Do high (C)	512	[450-550]

Table 2: the list of notes, their harmonics, and the bandwidth that encapsulates their shape with the rise and fall around the base harmonics

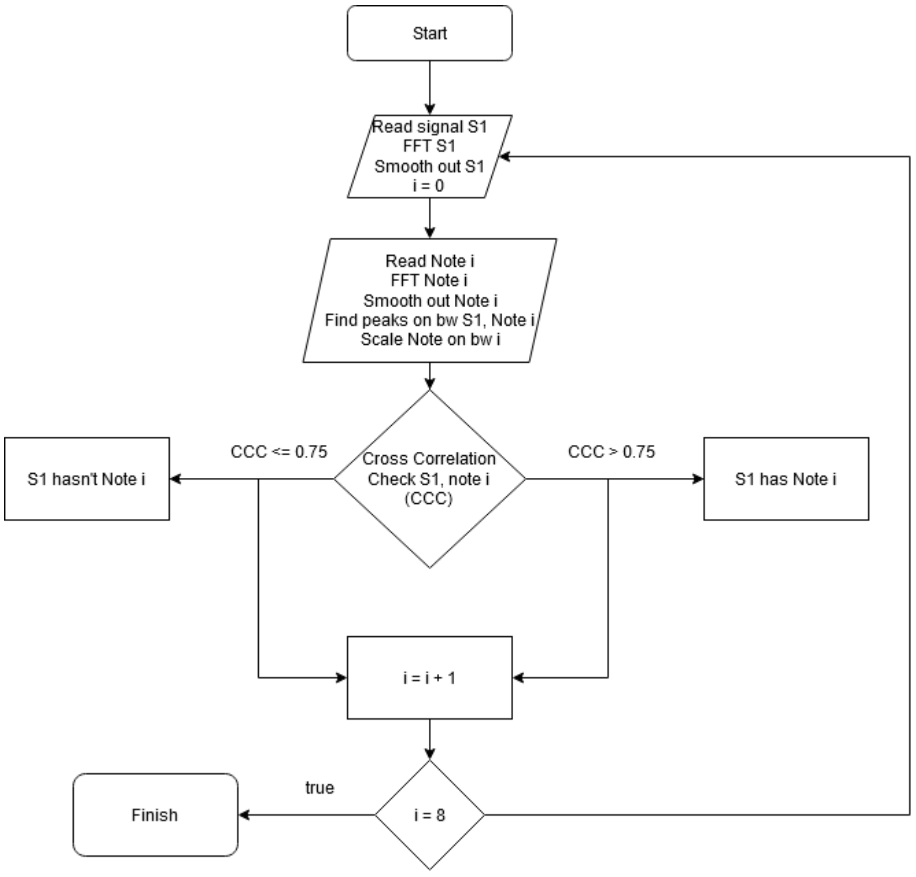


Figure 10: Flowchart of cross correlation check (bw stands for bandwidth of each harmonic.

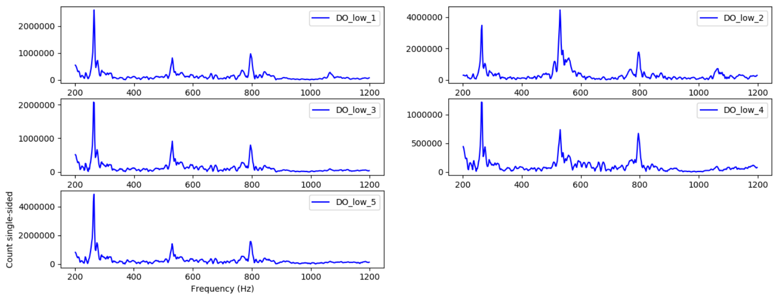


Figure 11: Frequency analysis of C (Do low) played 5 times

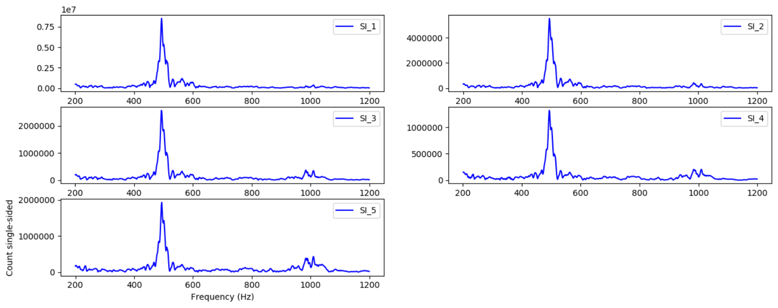


Figure 12: Frequency analysis of B (Si) played 5 times

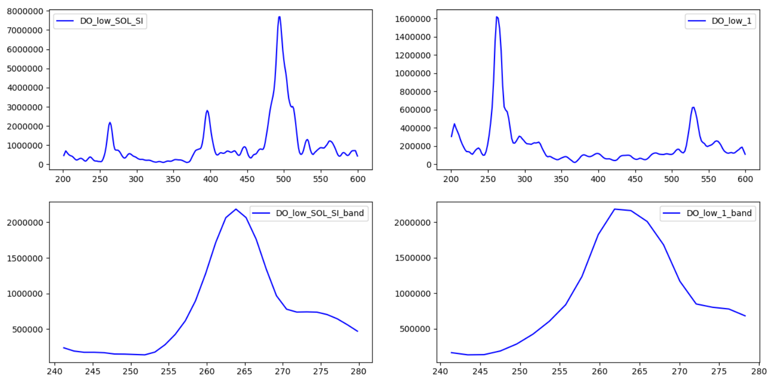


Figure 13: Comparing the combination of Do low, Sol, and Si (S1) with Do low (S2) to see that note exists within the combination. Top two graphs are the original signals within [200 600] Hz. Bottom graphs are the results of the process of signals, and those two signals are compared. The result for this run is 0.96 correlation. S1 has S2 within.

Note 1 \Note 2	Do low 1	Do low 2	Do low 3	Do low 4	Do low 5
Do low 1		0.97	0.98	0.97	0.97
Do low 2			1.00	0.99	1.00
Do low 3				1.00	1.00
Do low 4					0.99
Do low 5					

Table 3: Comparing 5 samples of Do low on frequency range [220 280] Hz with correlation check and zero padding

notes (triple notes are synthesized taking 3 note recordings and merging them together in Audacity).

Same note to same note comparison All 8 notes are compared with the notes in the same group (Do low with Do low, Re with Re, Mi with Mi and so on). To illustrate how the comparison is made the Table 3 and 4 is displayed. Among all the comparison minimum correlation is 0.16 and average correlation 0.87 for zero padding. The results for interpolation is minimum correlation 0.98 and average correlation 0.99. While both approach work well on average, zero padding can yield poor results when the notes have high sample count difference

Different notes to different notes comparison In this test each note recording is compared with all the other notes. Ideally the maximum correlation within this comparison should be low along with the average correlation. Zero padding method has maximum correlation of 0.68 and average correlation -0.20. With interpolation method maximum correlation is 0.65 and average correlation is -0.20

Finding the notes within the given input recording For this test three

Note 1 \Note 2	Do low 1	Do low 2	Do low 3	Do low 4	Do low 5
Do low 1		1.00	1.00	1.00	1.00
Do low 2			1.00	1.00	1.00
Do low 3				1.00	1.00
Do low 4					0.99
Do low 5					

Table 4: Comparing 5 samples of Do low (C) on frequency range [220 280] Hz with correlation check and interpolation

Triple Note Input	Found Notes
Do low, Fa, La	Fa, La
Do low, Mi, Sol	Mi, Sol
Do low, Sol, Si	Sol, Si
Re, Sol, Si	Re, Sol, Si

Table 5: Results on finding which notes exist within given input signal with triple notes. Using the zero padding to match sample lengths

different notes are merged together in Audacity and both methods checks which notes are in the signal. Results can be seen in Table 5 and 6. Zero padding struggles with finding the note Do low as its recording has much less samples than the one in the input signal. The complete tests are displayed at Section 6

5.2.3 Final Proposed Method

In the end, we present a project that is able to annotate boomwhackers notes by comparing the input recording (S_{input}) with the notes that were recorded to create a database of notes (S_n e.g. $S_1 = Dolow$, $S_2 = Re$ ... $S_8 = Dohigh$). In order to achieve this, the algorithm, as a start, does a fast fourier transform (FFT by *scipy.fftpack.fftfreq*) and takes the S_{input} to the frequency domain from time domain. The result of this step is depicted at Figure 14. The product of FFT is smoothed using convolution based smoothing (*numpy.convolve*[7]). From there, the transform's frequencies outside the base harmonics range ([200-600] Hz) are discarded. The product of these operations can be seen in Figure 15. Then, the remaining chunk of the transform is compared with each note. The comparison is done between the narrowed down FFT of S_{input} and FFT of S_n and is done around base harmonics of S_n (for

Triple Note Input	Found Notes
Do low, Fa, La	Do low, Fa, La
Do low, Mi, Sol	Do low, Mi, Sol
Do low, Sol, Si	Do low, Sol, Si
Re, Sol, Si	Re, Sol, Si

Table 6: Results on finding which notes exist within given input signal with triple notes. Using the interpolation to match sample lengths

Do low check, the range is [240-280] Hz so both the signal and Do low within the library is compared on this range, same is applied with other notes on their respective base harmonics range). Two signals which has the same shapes but on different scales results in low correlation.

The purpose of the comparison is to find out which note is played, not how the note is played. FFT analysis of the signals that was shown before with Figure 11 and Figure 12 as examples showed that the samples keep their shapes but not their scale along the y axis (Fourier Transform Constants). That is why we match the scales of the signals before comparing them. This is done by finding local maxima of each signal within the target range and scaling up all the samples of the signal with lower maxima. To clarify, let us call the signal with higher maxima on the target range S_{high} and the other one, with lower maxima, S_{low}. S_{low} is scaled up by a factor of $max(S_{high}/max(S_{low})$. Scaling is not enough for cross correlation check (CCC is done by *numpy.corrcoef* [8]), it also requires both signal to have same amount of samples, and it compares the signal on their shape similarity with a result of a range [-1.0-1.0] with -1.0 being reverse of the signal, 0 no correlation, and 1.0 maximum likeness. Between the signals, the one which has less samples within the target range is expanded to match the sample count of others by cubic interpolation (linear interpolation along x axis by *numpy.linspace* [9] and cubic interpolation along y axis by *scipy.interpolate.interp1d* [10]). Using cubic interpolation in this manner maintains the shape of the signal, it simply adds more in between points within the given range. In the end, both signals are ready for CCC. The steps are shown in Figure 16.

Now, CCC produces a high value (The algorithm consider 0.75 and up as a high value, [0.75-1.0]) if the signals are similar. This means, if S_{signal} has Do high in recording along with other notes, and is compared with Do low from the library (S_1), the correlation is high. Same operation is done with all 8 notes to figure out all the possible notes within the S_{signal}.

5.3 Calibration

To see how much the recording vary due to how the instrument is played and the room it is being played an experiment is conducted. The first variable is the room type, living room, a small room, and a mid size room. The second variable is the play type with, hitting the tubes on knee when cross-legged, while sitting, hitting the tubes on hand, and hitting on the tubes by hand (there are more variables to consider, but these were convenient to simulate during the Covid-19 lock down period). Using these 2 variables 12 recordings were taken and their fourier transform around the base harmonics is depicted in Figure 7. As can be seen, both the peaks and shape of the signal vary plenty. CCC on these signals yield the graph on Figure 17. In this set the minimum value is 0.19 with average 0.81. We set the threshold as 0.75 when two signals are considered the same. 21 comparison out of 79 in the set are below the value, meaning they are considered different. So, 0.265 percent of the comparisons are considered different while comparing the signals with different contexts (room and/or play type).

All the experiments were conducted in the same computer (Thinkpad p51). Here are the specs: CPU 6th Generation Intel® Core^{TM} i7-6820HQ with vPro^{TM} (2.70GHz, up to 3.60GHz with Turbo Boost, 4 Cores, 8MB Cache), RAM 16 GB DDR4 2133 MHz, hard disk 256 GB PCIe SSD OPAL2.0. Using this hardware, it takes 0.58 second to find all the notes within the given signal.

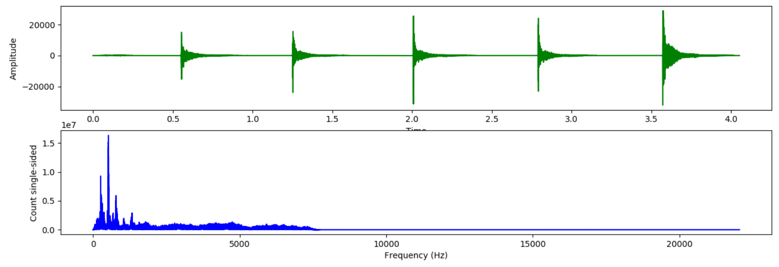


Figure 14: On top, there are recordings of 5 Do low notes and are displayed on time domain (time versus amplitude of audio). The bottom is the FFT of these 5 notes, transforming the signal to frequency domain. This is a demonstration, as each notes are exported separately (Do low 1, Do low 2 etc.)

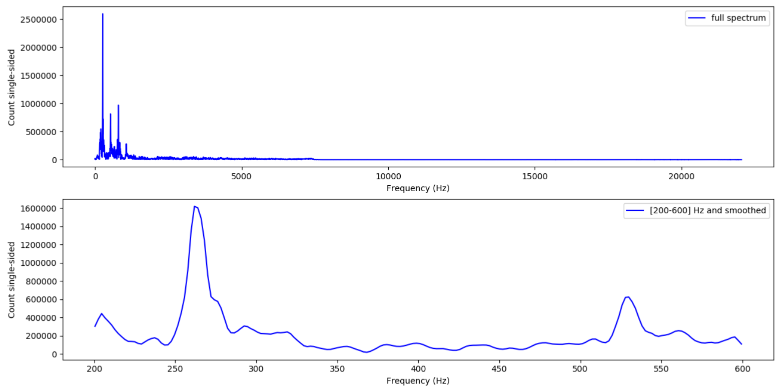


Figure 15: Top signal (Do low 1) is narrowed down to [200-600] Hz range and smoothed out, resulting in the bottom image

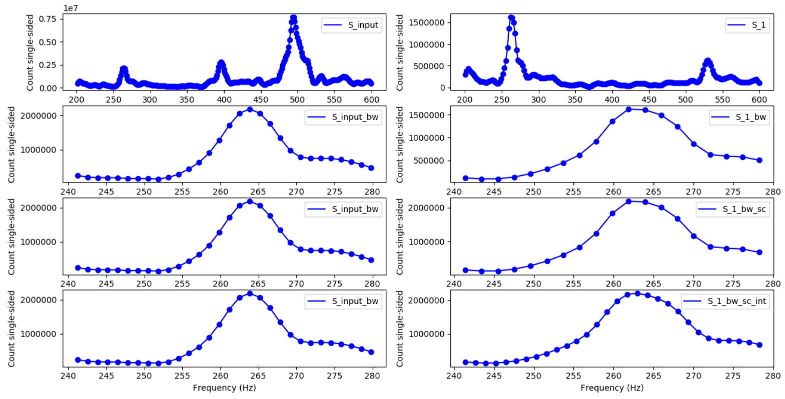


Figure 16: In each graph the circles are the actual data points and lines are put to emphasize the shape of the signal. First column is always the input signal (the combination of Do low, Sol, and Si in this case. Second column is the variations of Do low 1. First row is narrowed down signals on bandwidth. The second row is the further narrowing down around base harmonics range ([240-280] Hz for Do low). Third row is the scaling of the signals to match the levels on y axis. The last row is the result of cubic interpolation to match both signals on data size.

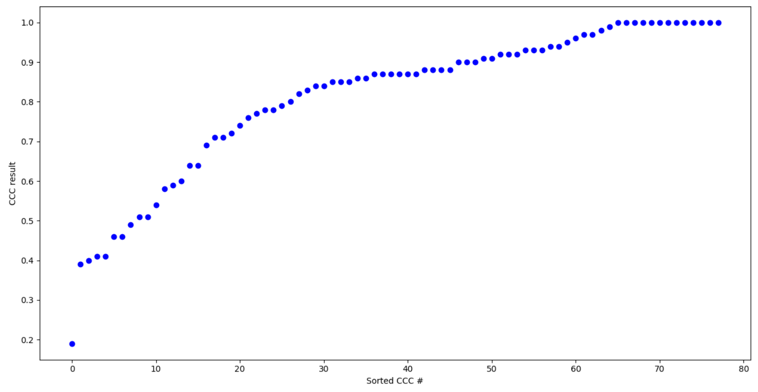


Figure 17: Sorted plot of comparison of variation pairings

6 Conclusion

This project went through iterations with each iteration being tested on how good it performs. In the end, the proposed method requires notes to be recorded to a sound database for comparison. The input signals (consisting of multiple notes) along with note recordings within the database, goes through multiple phases (FFT, smoothing ,narrowing down the bandwidth, scaling up or down, sample length match by cubic interpolation) and are compared by CCC (cross correlation check).

Besides people speaking in the classroom there can be other sources of noise such as other instruments being played at the same time, background audio being played for the practice session, or ambient noise within the background. It is important to accommodate for all the scenarios so the note annotation works in different environments. As it stands, it was not possible to test how the reverberations within the classroom change from context to context (time of day, amount of people, layout, windows open etc.).

As mentioned in Section 5 the shapes of notes when recorded can vary from room to room as well as from play type to play type. For this project, simulation of variation on room types and play types were rather limited. If we are to ignore this shortcoming, with proper calibration the algorithm has high performance for annotating the notes. For calibration, each note has to be recorded separately and extracted to the right folder with the agreed file naming conventions. This does not require high technical skills such as programming.

The most essential experiment, finding notes within a given boomwhackers recording takes 0.58 seconds in the setup I used.

6.1 Results

The proposed method went through three different tests.

Same note to same note comparison: This test was done to see if any note that was recorded within the same environment is a substitute for checking other signals. It also checks if these notes are considered the same from algorithm's standpoint. Doing this test by pairing same notes with same notes (e.g. Re_1 with Re_2 and many other pairings for all the notes) the average value is 0.99 with lowest value being 0.98. This means all the different instance (within the same context) of the same notes are considered the same. The full results are put in Appendix A

Different note to different note comparison: The purpose of this test is to find if algorithm gets confused over two notes. In these comparisons any value above 0.75 is a confusion as values above them considered to be same for the notes. It should not be the case for two different notes. If note A is compared with note B, note B is not compared with note A. That is why the list keeps getting shorter and shorter for the list. The average CCC result is -0.19 with the maximum result being 0.65. This means on average the notes are distinctly different. Also, none of the notes are confused over each other. At Appendix A the full results of CCC on comparing two different notes can be found.

Finding notes in multiple concurrent signals: This test is done to find which notes the input signal contains. The input signal is compared with each note within the database by CCC and any results above 0.75 correlation means the signal contains that note. The test results for common chords can be found in Table 7.
To see where the algorithm can fail, unusual combinations were also added. In the first set of unusual set all three notes are consecutive neighbours. The results are

chord name	file name	found notes
A minor (Am). A - C - E	LA_DO_low_MI	LA, DO low, MI
B diminished (Bdim). B - D - F	SI_RE_FA	SI, RE, FA
C major (C). C - E - G	DO_low_MI_SOL	DO low, MI, SOL
C major (C7) C - E - G - B	DO_low_MI_SOL_SI	DO low, MI, SOL, SI
D minor (Dm). D - F - A	RE_FA_LA	RE, FA, LA
D minor (Dm7) D - F - A - C	RE_FA_LA_DO_low	DO low, RE, FA, LA
E minor (Em). E - G - B	MI_SOL_SI	MI, SOL, SI
F major (F). F - A - C	FA_LA_DO_low	FA, LA, DO low
G major (G). G - B - D	SOL_SI_RE	SOL, SI, RE
G major (G7) G - B - D - F	SOL_SI_RE_FA	RE, SOL, SI

Table 7: The results of finding notes within input signal for common chords

input notes	file name	found notes
C - D - E	DO_low_RE_MI	Do_low, MI
D - E - F	RE_MI_FA	MI, FA
E - F - G	MI_FA_SOL	MI, SOL
F - G - A	FA_SOL_LA	LA
G - A - B	SOL_LA_SI	SI
A - B - Cc	LA_SI_DO_high	SI

Table 8: The results of finding notes within the combination of three neighbouring notes

depicted in Table 8. As can be seen in the table the algorithm has faulty outputs on all instances. In the second unusual set, there are two neighbouring notes and a single non-neighbouring note within the input signal. The results of that test is depicted in Table 9. Only 1 out of 6 output is correct. In both cases the outputs are unreliable.

In conclusion the algorithm can find trip and even quadruple notes combinations reliably if there are no neighbouring notes within the signal. On the other hand, If there are neighbouring notes within the input signal, the algorithm fails

Reflection on research questions

– How can we find notes within a recording of boomwhackers

input notes	file name	found notes
C - D - F	DO_low_RE_FA	DO low, RE, FA
D - E - G	RE_MI_SOL	MI, SOL, SI
E - F - A	MI_FA_LA	MI, LA
F - G - B	FA_SOL_SI	SOL, SI
G - A - Cc	SOL_LA_DO_high	LA, DO high
F - A - B	FA_LA_SI	FA, SI

Table 9: The results of finding notes within the combination of two neighbouring notes and one distant note

  - How can we classify a single note within boomwhacker recordings
  - How can we classify up to 4 concurrent notes

  With this project it is possible to find out which boomwhacker notes are present in a given signal. This is done by checking similarities with the signal, as shape of the signals kept consistent with each note if the environment is fixed. There are database notes within the system and those notes as signals are compared with the input signal. The system can find single notes without any problems. It can find notes within triple and quadruple notes recordings (tested using typical chords as inputs) played concurrently. When neighbouring notes are included in the signal, the system starts to fail.

- How can this system be robust, so it can work in different environments

  The classrooms are highly dynamic environments that can affect the sound recording of the notes. I simulated the environments I can simulate from my home to see how the signals might vary. The signal shapes vary from environment to environment. Tests within the project showed that the shapes of the signals stay similar so long as the context remains the same (how notes are played, which room they are being played). Hence, by updating the database notes (recording notes in the new environment and putting into database) and comparing those notes with the signal ensure the note annotation. This process was called calibration throughout the project. It does not require coding skills to achieve it.

7 Discussion and Future Work

7.1 Data

On gathering data, there were assumptions and limited possibilities. I only tested four kinds of play (while cross legged, sitting, hitting on hand, and hitting on tube) and just three rooms (small, mid and large size). It is possible there might be much more variations, especially when it comes to children as they do not have firm control over motor skills and motions can be erratic, as well as classrooms layout changing from day to day, week to week, even by the minute as children are forming and reforming groups and walking around. It is essential to test the system in real world and see how it performs. Maybe, the classroom is too dynamic and requires too frequent calibration. If that is the case, this system might be not feasible. To fix this issue, one can increase the robustness of the system without calibration by default or change the base approach entirely. A switch to a data driven approach like machine learning may prove to be more feasible. Whatever is the case, too frequent calibration issue would need to rework on the project. On the other hand, this project can be useful on creating a repository of boomwhackers with right annotations. Such repositories are required for supervised learning algorithms. In music information retrieval (called MIR commonly) domain repositories open to public to test and train algorithms exist. Such was not the case for boomwhackers.

7.2 Running online

The project would be more useful if it worked online and provide output in real time. However, this is an assumption and needs confirmation by the main stakeholder. If it is indeed the case (increased usability due to online processing), there would be a trade-off between accuracy and usability. This would be a new requirement and focus point, how to reconcile accuracy with usability and which one to prioritise.

7.3 Calibration

Current product requires recording boomwhacker notes, and exporting them as .wav files to right directories (with set names) as well as notes being recorded with right names convention. This process does not require coding skills, but can still take time and reading guidelines. It is possible, the teachers can get confused by what to do, and also simply lack the time for such extra steps to use the product. It is unfeasible (an open question, but common sense says unfeasible) to dedicate someone who would take care of the calibration. Although, making calibration easier is out of scope at this point, it still makes sense to document the requirements on how to calibrate as well as the shortcomings, and notify the people (future developers or end users) the shortcomings (requiring manual labor) of the product.

7.4 Noise

At the current stage the recordings of notes are taken without anybody speaking or any loud background noise. This is an unlikely scenario, not a good substitute for a classroom setting. Human speech, on the fundamental frequency, has an overlap with the boomwhacker notes. For the children's speech the fundamental frequency is 262 Hz for boys and 281 Hz for girls [11]. For adults it is on the lower side, 120 Hz for men and 210 for women [12]. The fundamental frequency of children's speech has an overlap in the bandwidth of Do low and Re. The first harmonic has an overlap with Si and Do high. For adults there is also an overlap on the fundamental frequency as well as first harmonic. If the system is to be tested within Pabo, both the teach and student would be adults. However, if the system is meant to be used in classroom the students would be toddlers, whereas the teacher an adult. I think whatever the future for this project is, it has to address noise at some point (high accuracy of annotation in a noisy environment).

7.5 Target hardware

The test system was a Thinkpad P51. It is a workstation from 2018 and currently, at 2020, is still sold for more than 1000 euros. With this setup finding the notes within a given recording takes 0.58 second. The run time is not a problem if the purpose is to find the notes after recordings are taken. However, it can become a problem if the project is taken towards online processing route. With online processing, the run time matters much more compared to former approach. With such future for the project, the future developers would need to discuss what hardware is available in the field (e.g. classroom) the project would run on.

References

[1] B. Spieker, *This thing called "handelingsverlegenheid": Teachers' lack of confidence in teaching music in Dutch primary schools: a problem that could be overcome by applying supportive technology?* Rat für Kulturelle Bildung e.V., 2019, pp. 30–35.

[2] Y. Sun, C. Jin, W. Zhao, and N. Wang, "Design of real-time rhythm tracking system based on neural network," 11 2018, pp. 356–360.

[3] S. G. Vishnu and S. G. Koolagudi, "An approach for mridanga stroke transcription in carnatic music using hgcc," in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON).* IEEE, 2019, pp. 2392–2397.

[4] I. Al-Hussaini, A. I. Humayun, S. Alam, S. I. Foysal, A. Al Masud, A. Mahmud, R. I. Chowdhury, N. Ibtehaz, S. U. Zaman, R. Hyder *et al.*, "Predictive real-time beat tracking from music for embedded application," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR).* IEEE, 2018, pp. 297–300.

[5] D. Clegg, "MoSCoW Method - wikipedia page," https://en.wikipedia.org/wiki/MoSCoW_method, [Online; accessed 11 August 2020].

[6] A. Mader and W. Eggink, "A design process for creative technology," in *DS 78: Proceedings of the 16th International conference on Engineering and Product Design Education (E&PDE14), Design Education and Human Technology Relations, University of Twente, The Netherlands, 04-05.09. 2014*, 2014.

[7] Community, "numpy.convolve - reference guide," https://numpy.org/doc/stable/reference/generated/numpy.convolve.html, [Online; accessed 11 August 2020].

[8] ——, "numpy.corrcoef - reference guide," https://numpy.org/doc/stable/reference/generated/numpy.corrcoef.html, [Online; accessed 11 August 2020].

[9] ——, “numpy.linspace - reference guide,” https://numpy.org/doc/stable/reference/generated/numpy.linspace.html, [Online; accessed 11 August 2020].

[10] ——, “scipy.interpolate.interp1d - reference guide,” https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html, [Online; accessed 11 August 2020].

[11] D. N. Sorenson, “A fundamental frequency investigation of children ages 6–10 years old,” *Journal of Communication Disorders*, vol. 22, no. 2, pp. 115–123, 1989.

[12] H. Traunmüller and A. Eriksson, “The frequency range of the voice fundamental in the speech of male and female adults,” *Unpublished manuscript*, 1995.

[13] Community, “scipy.fftpack.fftfreq - reference guide,” https://docs.scipy.org/doc/scipy/reference/generated/scipy.fftpack.fftfreq.html, [Online; accessed 11 August 2020].

Appendices

A Results

A.1 Same note to same not comparison

A.2 Notes to different notes comparison

Note 1 \Note 2	Do low 1	Do low 2	Do low 3	Do low 4	Do low 5
Do low 1		1.00	1.00	1.00	1.00
Do low 2			1.00	1.00	1.00
Do low 3				1.00	1.00
Do low 4					0.99
Do low 5					

Table 10: Comparing 5 samples of Do low on frequency range [220 280] Hz with correlation check and interpolation

Note 1 \Note 2	RE 1	RE 2	RE 3	RE 4	RE 5
RE 1		1.00	0.99	1.00	0.99
RE 2			0.99	1.00	0.99
RE 3				1.00	0.99
RE 4					0.99
RE 5					

Table 11: Comparing 5 samples of RE on frequency range [260 340] Hz with correlation check and interpolation

Note 1 \Note 2	MI 1	MI 2	MI 3	MI 4	MI 5
MI 1		1.00	1.00	0.99	0.99
MI 2			0.99	0.99	0.98
MI 3				1.00	0.99
MI 4					0.99
MI 5					

Table 12: Comparing 5 samples of MI on frequency range [320 380] Hz with correlation check and interpolation

Note 1 \Note 2	FA 1	FA 2	FA 3	FA 4	FA 5
FA 1		1.00	0.98	0.98	0.99
FA 2			0.98	0.98	0.98
FA 3				0.99	0.98
FA 4					0.99
FA 5					

Table 13: Comparing 5 samples of FA on frequency range [340 430] Hz with correlation check and interpolation

Note 1 \Note 2	SOL 1	SOL 2	SOL 3	SOL 4	SOL 5
SOL 1		1.00	1.00	0.99	1.00
SOL 2			1.00	1.00	1.00
SOL 3				1.00	1.00
SOL 4					0.99
SOL 5					

Table 14: Comparing 5 samples of SOL on frequency range [380 460] Hz with correlation check and interpolation

Note 1 \Note 2	LA 1	LA 2	LA 3	LA 4	LA 5
LA 1		0.99	0.99	0.99	1.00
LA 2			1.00	1.00	0.99
LA 3				1.00	0.99
LA 4					0.98
LA 5					

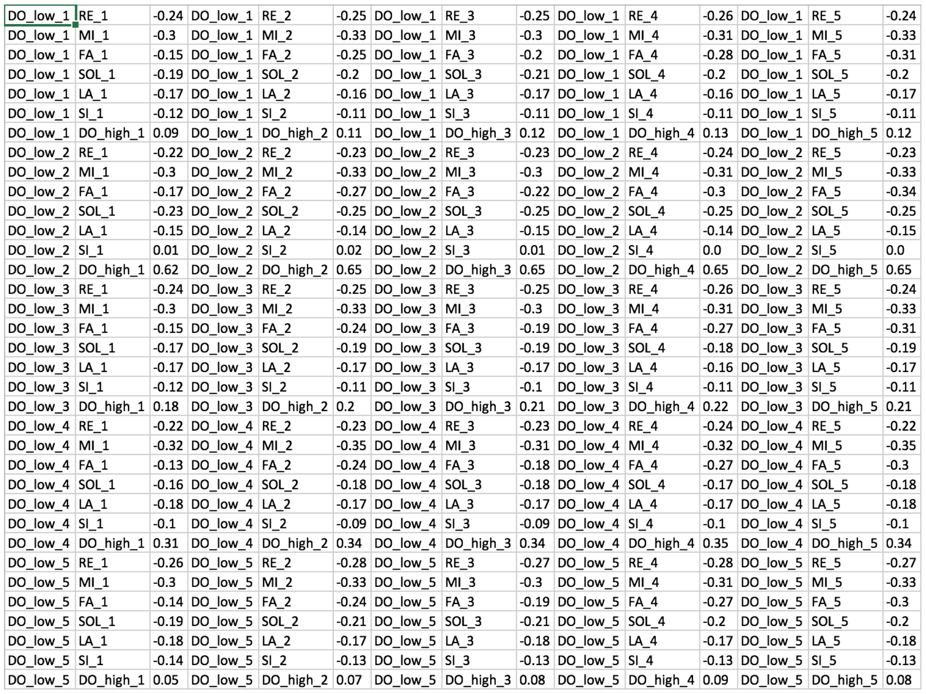
Table 15: Comparing 5 samples of LA on frequency range [400 500] Hz with correlation check and interpolation

Note 1 \Note 2	SI 1	SI 2	SI 3	SI 4	SI 5
SI 1		1.00	1.00	0.99	0.99
SI 2			1.00	0.99	1.00
SI 3				1.00	1.00
SI 4					1.00
SI 5					

Table 16: Comparing 5 samples of SI on frequency range [450 550] Hz with correlation check and interpolation

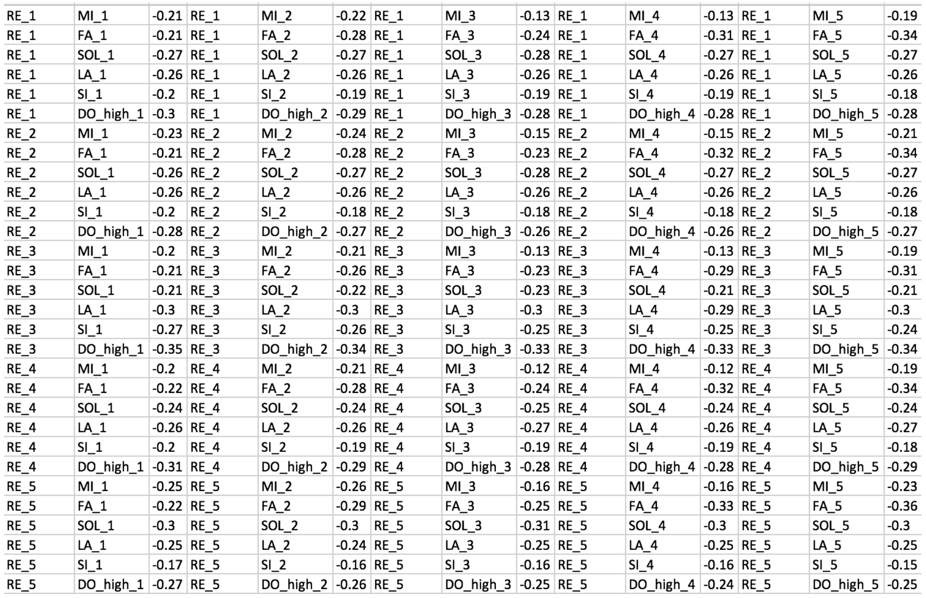
Note 1 \Note 2	DO high 1	DO high 2	DO high 3	DO high 4	DO high 5
DO high 1		0.99	0.99	1.00	0.99
DO high 2			0.99	0.99	0.99
DO high 3				1.00	1.00
DO high 4					0.99
DO high 5					

Table 17: Comparing 5 samples of DO high on frequency range [450 550] Hz with correlation check and interpolation



DO_low_1	RE_1	-0.24	DO_low_1	RE_2	-0.25	DO_low_1	RE_3	-0.25	DO_low_1	RE_4	-0.26	DO_low_1	RE_5	-0.24
DO_low_1	MI_1	-0.3	DO_low_1	MI_2	-0.33	DO_low_1	MI_3	-0.3	DO_low_1	MI_4	-0.31	DO_low_1	MI_5	-0.33
DO_low_1	FA_1	-0.15	DO_low_1	FA_2	-0.25	DO_low_1	FA_3	-0.2	DO_low_1	FA_4	-0.28	DO_low_1	FA_5	-0.31
DO_low_1	SOL_1	-0.19	DO_low_1	SOL_2	-0.2	DO_low_1	SOL_3	-0.21	DO_low_1	SOL_4	-0.2	DO_low_1	SOL_5	-0.2
DO_low_1	LA_1	-0.17	DO_low_1	LA_2	-0.16	DO_low_1	LA_3	-0.17	DO_low_1	LA_4	-0.16	DO_low_1	LA_5	-0.17
DO_low_1	SI_1	-0.12	DO_low_1	SI_2	-0.11	DO_low_1	SI_3	-0.11	DO_low_1	SI_4	-0.11	DO_low_1	SI_5	-0.11
DO_low_1	DO_high_1	0.09	DO_low_1	DO_high_2	0.11	DO_low_1	DO_high_3	0.12	DO_low_1	DO_high_4	0.13	DO_low_1	DO_high_5	0.12
DO_low_2	RE_1	-0.22	DO_low_2	RE_2	-0.23	DO_low_2	RE_3	-0.23	DO_low_2	RE_4	-0.24	DO_low_2	RE_5	-0.23
DO_low_2	MI_1	-0.3	DO_low_2	MI_2	-0.33	DO_low_2	MI_3	-0.3	DO_low_2	MI_4	-0.31	DO_low_2	MI_5	-0.33
DO_low_2	FA_1	-0.17	DO_low_2	FA_2	-0.27	DO_low_2	FA_3	-0.22	DO_low_2	FA_4	-0.3	DO_low_2	FA_5	-0.34
DO_low_2	SOL_1	-0.23	DO_low_2	SOL_2	-0.25	DO_low_2	SOL_3	-0.25	DO_low_2	SOL_4	-0.25	DO_low_2	SOL_5	-0.25
DO_low_2	LA_1	-0.15	DO_low_2	LA_2	-0.14	DO_low_2	LA_3	-0.15	DO_low_2	LA_4	-0.14	DO_low_2	LA_5	-0.15
DO_low_2	SI_1	0.01	DO_low_2	SI_2	0.02	DO_low_2	SI_3	0.01	DO_low_2	SI_4	0.0	DO_low_2	SI_5	0.0
DO_low_2	DO_high_1	0.62	DO_low_2	DO_high_2	0.65	DO_low_2	DO_high_3	0.65	DO_low_2	DO_high_4	0.65	DO_low_2	DO_high_5	0.65
DO_low_3	RE_1	-0.24	DO_low_3	RE_2	-0.25	DO_low_3	RE_3	-0.25	DO_low_3	RE_4	-0.26	DO_low_3	RE_5	-0.24
DO_low_3	MI_1	-0.3	DO_low_3	MI_2	-0.33	DO_low_3	MI_3	-0.3	DO_low_3	MI_4	-0.31	DO_low_3	MI_5	-0.33
DO_low_3	FA_1	-0.15	DO_low_3	FA_2	-0.24	DO_low_3	FA_3	-0.19	DO_low_3	FA_4	-0.27	DO_low_3	FA_5	-0.31
DO_low_3	SOL_1	-0.17	DO_low_3	SOL_2	-0.19	DO_low_3	SOL_3	-0.19	DO_low_3	SOL_4	-0.18	DO_low_3	SOL_5	-0.19
DO_low_3	LA_1	-0.17	DO_low_3	LA_2	-0.17	DO_low_3	LA_3	-0.17	DO_low_3	LA_4	-0.16	DO_low_3	LA_5	-0.17
DO_low_3	SI_1	-0.12	DO_low_3	SI_2	-0.11	DO_low_3	SI_3	-0.1	DO_low_3	SI_4	-0.11	DO_low_3	SI_5	-0.11
DO_low_3	DO_high_1	0.18	DO_low_3	DO_high_2	0.2	DO_low_3	DO_high_3	0.21	DO_low_3	DO_high_4	0.22	DO_low_3	DO_high_5	0.21
DO_low_4	RE_1	-0.22	DO_low_4	RE_2	-0.23	DO_low_4	RE_3	-0.23	DO_low_4	RE_4	-0.24	DO_low_4	RE_5	-0.22
DO_low_4	MI_1	-0.32	DO_low_4	MI_2	-0.35	DO_low_4	MI_3	-0.31	DO_low_4	MI_4	-0.32	DO_low_4	MI_5	-0.35
DO_low_4	FA_1	-0.13	DO_low_4	FA_2	-0.24	DO_low_4	FA_3	-0.18	DO_low_4	FA_4	-0.27	DO_low_4	FA_5	-0.3
DO_low_4	SOL_1	-0.16	DO_low_4	SOL_2	-0.18	DO_low_4	SOL_3	-0.18	DO_low_4	SOL_4	-0.17	DO_low_4	SOL_5	-0.18
DO_low_4	LA_1	-0.18	DO_low_4	LA_2	-0.17	DO_low_4	LA_3	-0.17	DO_low_4	LA_4	-0.17	DO_low_4	LA_5	-0.18
DO_low_4	SI_1	-0.1	DO_low_4	SI_2	-0.09	DO_low_4	SI_3	-0.09	DO_low_4	SI_4	-0.1	DO_low_4	SI_5	-0.1
DO_low_4	DO_high_1	0.31	DO_low_4	DO_high_2	0.34	DO_low_4	DO_high_3	0.34	DO_low_4	DO_high_4	0.35	DO_low_4	DO_high_5	0.34
DO_low_5	RE_1	-0.26	DO_low_5	RE_2	-0.28	DO_low_5	RE_3	-0.27	DO_low_5	RE_4	-0.28	DO_low_5	RE_5	-0.27
DO_low_5	MI_1	-0.3	DO_low_5	MI_2	-0.33	DO_low_5	MI_3	-0.3	DO_low_5	MI_4	-0.31	DO_low_5	MI_5	-0.33
DO_low_5	FA_1	-0.14	DO_low_5	FA_2	-0.24	DO_low_5	FA_3	-0.19	DO_low_5	FA_4	-0.27	DO_low_5	FA_5	-0.3
DO_low_5	SOL_1	-0.19	DO_low_5	SOL_2	-0.21	DO_low_5	SOL_3	-0.21	DO_low_5	SOL_4	-0.2	DO_low_5	SOL_5	-0.2
DO_low_5	LA_1	-0.18	DO_low_5	LA_2	-0.17	DO_low_5	LA_3	-0.18	DO_low_5	LA_4	-0.17	DO_low_5	LA_5	-0.18
DO_low_5	SI_1	-0.14	DO_low_5	SI_2	-0.13	DO_low_5	SI_3	-0.13	DO_low_5	SI_4	-0.13	DO_low_5	SI_5	-0.13
DO_low_5	DO_high_1	0.05	DO_low_5	DO_high_2	0.07	DO_low_5	DO_high_3	0.08	DO_low_5	DO_high_4	0.09	DO_low_5	DO_high_5	0.08

Figure 18: CCC results for Do low with all the other notes



RE_1	MI_1	-0.21	RE_1	MI_2	-0.22	RE_1	MI_3	-0.13	RE_1	MI_4	-0.13	RE_1	MI_5	-0.19
RE_1	FA_1	-0.21	RE_1	FA_2	-0.28	RE_1	FA_3	-0.24	RE_1	FA_4	-0.31	RE_1	FA_5	-0.34
RE_1	SOL_1	-0.27	RE_1	SOL_2	-0.27	RE_1	SOL_3	-0.28	RE_1	SOL_4	-0.27	RE_1	SOL_5	-0.27
RE_1	LA_1	-0.26	RE_1	LA_2	-0.26	RE_1	LA_3	-0.26	RE_1	LA_4	-0.26	RE_1	LA_5	-0.26
RE_1	SI_1	-0.2	RE_1	SI_2	-0.19	RE_1	SI_3	-0.19	RE_1	SI_4	-0.19	RE_1	SI_5	-0.18
RE_1	DO_high_1	-0.3	RE_1	DO_high_2	-0.29	RE_1	DO_high_3	-0.28	RE_1	DO_high_4	-0.28	RE_1	DO_high_5	-0.28
RE_2	MI_1	-0.23	RE_2	MI_2	-0.24	RE_2	MI_3	-0.15	RE_2	MI_4	-0.15	RE_2	MI_5	-0.21
RE_2	FA_1	-0.21	RE_2	FA_2	-0.28	RE_2	FA_3	-0.23	RE_2	FA_4	-0.32	RE_2	FA_5	-0.34
RE_2	SOL_1	-0.26	RE_2	SOL_2	-0.27	RE_2	SOL_3	-0.28	RE_2	SOL_4	-0.27	RE_2	SOL_5	-0.27
RE_2	LA_1	-0.26	RE_2	LA_2	-0.26	RE_2	LA_3	-0.26	RE_2	LA_4	-0.26	RE_2	LA_5	-0.26
RE_2	SI_1	-0.2	RE_2	SI_2	-0.18	RE_2	SI_3	-0.18	RE_2	SI_4	-0.18	RE_2	SI_5	-0.18
RE_2	DO_high_1	-0.28	RE_2	DO_high_2	-0.27	RE_2	DO_high_3	-0.26	RE_2	DO_high_4	-0.26	RE_2	DO_high_5	-0.27
RE_3	MI_1	-0.2	RE_3	MI_2	-0.21	RE_3	MI_3	-0.13	RE_3	MI_4	-0.13	RE_3	MI_5	-0.19
RE_3	FA_1	-0.21	RE_3	FA_2	-0.26	RE_3	FA_3	-0.23	RE_3	FA_4	-0.29	RE_3	FA_5	-0.31
RE_3	SOL_1	-0.21	RE_3	SOL_2	-0.22	RE_3	SOL_3	-0.23	RE_3	SOL_4	-0.21	RE_3	SOL_5	-0.21
RE_3	LA_1	-0.3	RE_3	LA_2	-0.3	RE_3	LA_3	-0.3	RE_3	LA_4	-0.29	RE_3	LA_5	-0.3
RE_3	SI_1	-0.27	RE_3	SI_2	-0.26	RE_3	SI_3	-0.25	RE_3	SI_4	-0.25	RE_3	SI_5	-0.24
RE_3	DO_high_1	-0.35	RE_3	DO_high_2	-0.34	RE_3	DO_high_3	-0.33	RE_3	DO_high_4	-0.33	RE_3	DO_high_5	-0.34
RE_4	MI_1	-0.2	RE_4	MI_2	-0.21	RE_4	MI_3	-0.12	RE_4	MI_4	-0.12	RE_4	MI_5	-0.19
RE_4	FA_1	-0.22	RE_4	FA_2	-0.28	RE_4	FA_3	-0.24	RE_4	FA_4	-0.32	RE_4	FA_5	-0.34
RE_4	SOL_1	-0.24	RE_4	SOL_2	-0.24	RE_4	SOL_3	-0.25	RE_4	SOL_4	-0.24	RE_4	SOL_5	-0.24
RE_4	LA_1	-0.26	RE_4	LA_2	-0.26	RE_4	LA_3	-0.27	RE_4	LA_4	-0.26	RE_4	LA_5	-0.27
RE_4	SI_1	-0.2	RE_4	SI_2	-0.19	RE_4	SI_3	-0.19	RE_4	SI_4	-0.19	RE_4	SI_5	-0.18
RE_4	DO_high_1	-0.31	RE_4	DO_high_2	-0.29	RE_4	DO_high_3	-0.28	RE_4	DO_high_4	-0.28	RE_4	DO_high_5	-0.29
RE_5	MI_1	-0.25	RE_5	MI_2	-0.26	RE_5	MI_3	-0.16	RE_5	MI_4	-0.16	RE_5	MI_5	-0.23
RE_5	FA_1	-0.22	RE_5	FA_2	-0.29	RE_5	FA_3	-0.25	RE_5	FA_4	-0.33	RE_5	FA_5	-0.36
RE_5	SOL_1	-0.3	RE_5	SOL_2	-0.3	RE_5	SOL_3	-0.31	RE_5	SOL_4	-0.3	RE_5	SOL_5	-0.3
RE_5	LA_1	-0.25	RE_5	LA_2	-0.24	RE_5	LA_3	-0.25	RE_5	LA_4	-0.25	RE_5	LA_5	-0.25
RE_5	SI_1	-0.17	RE_5	SI_2	-0.16	RE_5	SI_3	-0.16	RE_5	SI_4	-0.16	RE_5	SI_5	-0.15
RE_5	DO_high_1	-0.27	RE_5	DO_high_2	-0.26	RE_5	DO_high_3	-0.25	RE_5	DO_high_4	-0.24	RE_5	DO_high_5	-0.25

Figure 19: CCC results for Re with all the other notes

MI_1	FA_1	0.02	MI_1	FA_2	-0.03	MI_1	FA_3	0.02	MI_1	FA_4	-0.07	MI_1	FA_5	-0.09
MI_1	SOL_1	-0.23	MI_1	SOL_2	-0.21	MI_1	SOL_3	-0.23	MI_1	SOL_4	-0.2	MI_1	SOL_5	-0.2
MI_1	LA_1	-0.32	MI_1	LA_2	-0.32	MI_1	LA_3	-0.32	MI_1	LA_4	-0.32	MI_1	LA_5	-0.31
MI_1	SI_1	-0.24	MI_1	SI_2	-0.24	MI_1	SI_3	-0.24	MI_1	SI_4	-0.23	MI_1	SI_5	-0.23
MI_1	DO_high_1	-0.39	MI_1	DO_high_2	-0.38	MI_1	DO_high_3	-0.39	MI_1	DO_high_4	-0.38	MI_1	DO_high_5	-0.39
MI_2	FA_1	0.02	MI_2	FA_2	-0.03	MI_2	FA_3	0.02	MI_2	FA_4	-0.08	MI_2	FA_5	-0.09
MI_2	SOL_1	-0.23	MI_2	SOL_2	-0.21	MI_2	SOL_3	-0.23	MI_2	SOL_4	-0.21	MI_2	SOL_5	-0.2
MI_2	LA_1	-0.31	MI_2	LA_2	-0.31	MI_2	LA_3	-0.31	MI_2	LA_4	-0.31	MI_2	LA_5	-0.3
MI_2	SI_1	-0.21	MI_2	SI_2	-0.21	MI_2	SI_3	-0.21	MI_2	SI_4	-0.21	MI_2	SI_5	-0.21
MI_2	DO_high_1	-0.36	MI_2	DO_high_2	-0.35	MI_2	DO_high_3	-0.36	MI_2	DO_high_4	-0.35	MI_2	DO_high_5	-0.36
MI_3	FA_1	0.03	MI_3	FA_2	-0.03	MI_3	FA_3	0.02	MI_3	FA_4	-0.08	MI_3	FA_5	-0.09
MI_3	SOL_1	-0.29	MI_3	SOL_2	-0.27	MI_3	SOL_3	-0.29	MI_3	SOL_4	-0.27	MI_3	SOL_5	-0.26
MI_3	LA_1	-0.29	MI_3	LA_2	-0.29	MI_3	LA_3	-0.29	MI_3	LA_4	-0.29	MI_3	LA_5	-0.28
MI_3	SI_1	-0.14	MI_3	SI_2	-0.14	MI_3	SI_3	-0.14	MI_3	SI_4	-0.14	MI_3	SI_5	-0.14
MI_3	DO_high_1	-0.25	MI_3	DO_high_2	-0.25	MI_3	DO_high_3	-0.25	MI_3	DO_high_4	-0.24	MI_3	DO_high_5	-0.25
MI_4	FA_1	0.03	MI_4	FA_2	-0.03	MI_4	FA_3	0.02	MI_4	FA_4	-0.08	MI_4	FA_5	-0.09
MI_4	SOL_1	-0.28	MI_4	SOL_2	-0.26	MI_4	SOL_3	-0.28	MI_4	SOL_4	-0.25	MI_4	SOL_5	-0.25
MI_4	LA_1	-0.3	MI_4	LA_2	-0.3	MI_4	LA_3	-0.3	MI_4	LA_4	-0.3	MI_4	LA_5	-0.29
MI_4	SI_1	-0.17	MI_4	SI_2	-0.17	MI_4	SI_3	-0.17	MI_4	SI_4	-0.17	MI_4	SI_5	-0.17
MI_4	DO_high_1	-0.29	MI_4	DO_high_2	-0.28	MI_4	DO_high_3	-0.29	MI_4	DO_high_4	-0.28	MI_4	DO_high_5	-0.29
MI_5	FA_1	0.0	MI_5	FA_2	-0.06	MI_5	FA_3	-0.01	MI_5	FA_4	-0.11	MI_5	FA_5	-0.12
MI_5	SOL_1	-0.25	MI_5	SOL_2	-0.23	MI_5	SOL_3	-0.25	MI_5	SOL_4	-0.22	MI_5	SOL_5	-0.22
MI_5	LA_1	-0.3	MI_5	LA_2	-0.31	MI_5	LA_3	-0.3	MI_5	LA_4	-0.31	MI_5	LA_5	-0.3
MI_5	SI_1	-0.2	MI_5	SI_2	-0.2	MI_5	SI_3	-0.2	MI_5	SI_4	-0.2	MI_5	SI_5	-0.2
MI_5	DO_high_1	-0.35	MI_5	DO_high_2	-0.34	MI_5	DO_high_3	-0.35	MI_5	DO_high_4	-0.34	MI_5	DO_high_5	-0.35

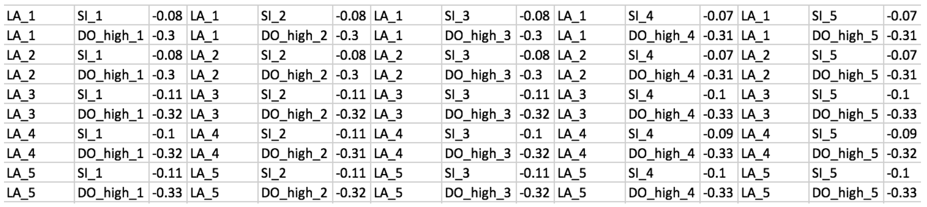
Figure 20: CCC results for Mi with all the other notes

FA_1	SOL_1	-0.12	FA_1	SOL_2	-0.11	FA_1	SOL_3	-0.14	FA_1	SOL_4	-0.11	FA_1	SOL_5	-0.1
FA_1	LA_1	-0.34	FA_1	LA_2	-0.35	FA_1	LA_3	-0.35	FA_1	LA_4	-0.35	FA_1	LA_5	-0.33
FA_1	SI_1	-0.21	FA_1	SI_2	-0.21	FA_1	SI_3	-0.21	FA_1	SI_4	-0.2	FA_1	SI_5	-0.2
FA_1	DO_high_1	-0.33	FA_1	DO_high_2	-0.32	FA_1	DO_high_3	-0.33	FA_1	DO_high_4	-0.33	FA_1	DO_high_5	-0.33
FA_2	SOL_1	-0.13	FA_2	SOL_2	-0.12	FA_2	SOL_3	-0.15	FA_2	SOL_4	-0.12	FA_2	SOL_5	-0.11
FA_2	LA_1	-0.31	FA_2	LA_2	-0.31	FA_2	LA_3	-0.31	FA_2	LA_4	-0.31	FA_2	LA_5	-0.3
FA_2	SI_1	-0.09	FA_2	SI_2	-0.09	FA_2	SI_3	-0.09	FA_2	SI_4	-0.08	FA_2	SI_5	-0.08
FA_2	DO_high_1	-0.28	FA_2	DO_high_2	-0.27	FA_2	DO_high_3	-0.27	FA_2	DO_high_4	-0.27	FA_2	DO_high_5	-0.27
FA_3	SOL_1	-0.1	FA_3	SOL_2	-0.09	FA_3	SOL_3	-0.12	FA_3	SOL_4	-0.09	FA_3	SOL_5	-0.08
FA_3	LA_1	-0.32	FA_3	LA_2	-0.32	FA_3	LA_3	-0.33	FA_3	LA_4	-0.33	FA_3	LA_5	-0.31
FA_3	SI_1	-0.1	FA_3	SI_2	-0.1	FA_3	SI_3	-0.1	FA_3	SI_4	-0.09	FA_3	SI_5	-0.09
FA_3	DO_high_1	-0.26	FA_3	DO_high_2	-0.26	FA_3	DO_high_3	-0.26	FA_3	DO_high_4	-0.26	FA_3	DO_high_5	-0.26
FA_4	SOL_1	-0.12	FA_4	SOL_2	-0.11	FA_4	SOL_3	-0.14	FA_4	SOL_4	-0.11	FA_4	SOL_5	-0.1
FA_4	LA_1	-0.3	FA_4	LA_2	-0.3	FA_4	LA_3	-0.31	FA_4	LA_4	-0.3	FA_4	LA_5	-0.29
FA_4	SI_1	-0.06	FA_4	SI_2	-0.05	FA_4	SI_3	-0.05	FA_4	SI_4	-0.04	FA_4	SI_5	-0.04
FA_4	DO_high_1	-0.26	FA_4	DO_high_2	-0.26	FA_4	DO_high_3	-0.26	FA_4	DO_high_4	-0.26	FA_4	DO_high_5	-0.26
FA_5	SOL_1	-0.13	FA_5	SOL_2	-0.11	FA_5	SOL_3	-0.15	FA_5	SOL_4	-0.11	FA_5	SOL_5	-0.1
FA_5	LA_1	-0.34	FA_5	LA_2	-0.34	FA_5	LA_3	-0.34	FA_5	LA_4	-0.34	FA_5	LA_5	-0.33
FA_5	SI_1	-0.12	FA_5	SI_2	-0.12	FA_5	SI_3	-0.12	FA_5	SI_4	-0.1	FA_5	SI_5	-0.11
FA_5	DO_high_1	-0.32	FA_5	DO_high_2	-0.31	FA_5	DO_high_3	-0.32	FA_5	DO_high_4	-0.32	FA_5	DO_high_5	-0.32

Figure 21: CCC results for Fa with all the other notes

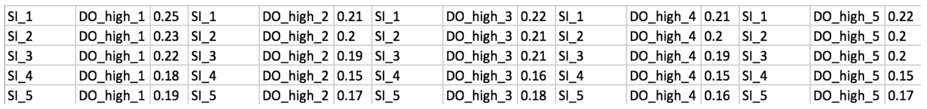
SOL_1	LA_1	-0.27	SOL_1	LA_2	-0.28	SOL_1	LA_3	-0.27	SOL_1	LA_4	-0.27	SOL_1	LA_5	-0.25
SOL_1	SI_1	-0.27	SOL_1	SI_2	-0.27	SOL_1	SI_3	-0.26	SOL_1	SI_4	-0.26	SOL_1	SI_5	-0.26
SOL_1	DO_high_1	-0.43	SOL_1	DO_high_2	-0.41	SOL_1	DO_high_3	-0.42	SOL_1	DO_high_4	-0.42	SOL_1	DO_high_5	-0.42
SOL_2	LA_1	-0.27	SOL_2	LA_2	-0.28	SOL_2	LA_3	-0.27	SOL_2	LA_4	-0.28	SOL_2	LA_5	-0.25
SOL_2	SI_1	-0.25	SOL_2	SI_2	-0.26	SOL_2	SI_3	-0.25	SOL_2	SI_4	-0.25	SOL_2	SI_5	-0.25
SOL_2	DO_high_1	-0.42	SOL_2	DO_high_2	-0.4	SOL_2	DO_high_3	-0.42	SOL_2	DO_high_4	-0.41	SOL_2	DO_high_5	-0.41
SOL_3	LA_1	-0.26	SOL_3	LA_2	-0.27	SOL_3	LA_3	-0.26	SOL_3	LA_4	-0.27	SOL_3	LA_5	-0.24
SOL_3	SI_1	-0.26	SOL_3	SI_2	-0.26	SOL_3	SI_3	-0.26	SOL_3	SI_4	-0.25	SOL_3	SI_5	-0.26
SOL_3	DO_high_1	-0.42	SOL_3	DO_high_2	-0.41	SOL_3	DO_high_3	-0.42	SOL_3	DO_high_4	-0.42	SOL_3	DO_high_5	-0.42
SOL_4	LA_1	-0.29	SOL_4	LA_2	-0.3	SOL_4	LA_3	-0.28	SOL_4	LA_4	-0.29	SOL_4	LA_5	-0.27
SOL_4	SI_1	-0.25	SOL_4	SI_2	-0.26	SOL_4	SI_3	-0.25	SOL_4	SI_4	-0.25	SOL_4	SI_5	-0.25
SOL_4	DO_high_1	-0.42	SOL_4	DO_high_2	-0.4	SOL_4	DO_high_3	-0.41	SOL_4	DO_high_4	-0.41	SOL_4	DO_high_5	-0.41
SOL_5	LA_1	-0.28	SOL_5	LA_2	-0.29	SOL_5	LA_3	-0.27	SOL_5	LA_4	-0.28	SOL_5	LA_5	-0.26
SOL_5	SI_1	-0.26	SOL_5	SI_2	-0.26	SOL_5	SI_3	-0.26	SOL_5	SI_4	-0.25	SOL_5	SI_5	-0.26
SOL_5	DO_high_1	-0.42	SOL_5	DO_high_2	-0.41	SOL_5	DO_high_3	-0.42	SOL_5	DO_high_4	-0.41	SOL_5	DO_high_5	-0.41

Figure 22: CCC results for Sol with all the other notes



LA_1	SI_1	-0.08	LA_1	SI_2	-0.08	LA_1	SI_3	-0.08	LA_1	SI_4	-0.07	LA_1	SI_5	-0.07
LA_1	DO_high_1	-0.3	LA_1	DO_high_2	-0.3	LA_1	DO_high_3	-0.3	LA_1	DO_high_4	-0.31	LA_1	DO_high_5	-0.31
LA_2	SI_1	-0.08	LA_2	SI_2	-0.08	LA_2	SI_3	-0.08	LA_2	SI_4	-0.07	LA_2	SI_5	-0.07
LA_2	DO_high_1	-0.3	LA_2	DO_high_2	-0.3	LA_2	DO_high_3	-0.3	LA_2	DO_high_4	-0.31	LA_2	DO_high_5	-0.31
LA_3	SI_1	-0.11	LA_3	SI_2	-0.11	LA_3	SI_3	-0.11	LA_3	SI_4	-0.1	LA_3	SI_5	-0.1
LA_3	DO_high_1	-0.32	LA_3	DO_high_2	-0.32	LA_3	DO_high_3	-0.32	LA_3	DO_high_4	-0.33	LA_3	DO_high_5	-0.33
LA_4	SI_1	-0.1	LA_4	SI_2	-0.11	LA_4	SI_3	-0.1	LA_4	SI_4	-0.09	LA_4	SI_5	-0.09
LA_4	DO_high_1	-0.32	LA_4	DO_high_2	-0.31	LA_4	DO_high_3	-0.32	LA_4	DO_high_4	-0.33	LA_4	DO_high_5	-0.32
LA_5	SI_1	-0.11	LA_5	SI_2	-0.11	LA_5	SI_3	-0.11	LA_5	SI_4	-0.1	LA_5	SI_5	-0.1
LA_5	DO_high_1	-0.33	LA_5	DO_high_2	-0.32	LA_5	DO_high_3	-0.32	LA_5	DO_high_4	-0.33	LA_5	DO_high_5	-0.33

Figure 23: CCC results for La with all the other notes



SI_1	DO_high_1	0.25	SI_1	DO_high_2	0.21	SI_1	DO_high_3	0.22	SI_1	DO_high_4	0.21	SI_1	DO_high_5	0.22
SI_2	DO_high_1	0.23	SI_2	DO_high_2	0.2	SI_2	DO_high_3	0.21	SI_2	DO_high_4	0.2	SI_2	DO_high_5	0.2
SI_3	DO_high_1	0.22	SI_3	DO_high_2	0.19	SI_3	DO_high_3	0.21	SI_3	DO_high_4	0.19	SI_3	DO_high_5	0.2
SI_4	DO_high_1	0.18	SI_4	DO_high_2	0.15	SI_4	DO_high_3	0.16	SI_4	DO_high_4	0.15	SI_4	DO_high_5	0.15
SI_5	DO_high_1	0.19	SI_5	DO_high_2	0.17	SI_5	DO_high_3	0.18	SI_5	DO_high_4	0.16	SI_5	DO_high_5	0.17

Figure 24: CCC results for Si with all the other notes