2227216m

MUSIC CURATION & ANALYTICS 2018 REPORT

INTRODUCTION

Music scholars have more digital tools available than ever to conduct research and analysis on the history, styles, and sounds of music in the shape of digital archives and libraries, with their extensive digitized, recorded and encoded music collections. Large music datasets are not only found in academic institutions, they are also found in the commercial sphere, with companies such as Spotify utilizing a huge dataset, which is essential for their music recommendation features (Palmer 2013). An example of a large scholarly music dataset is the Million Song Dataset, which aims to provide a research resource for evaluating musical data with the intention of helping researchers who work in the Music Information & Retrieval (MIR) field (Bertin-Mahieux et al 2011).

The Music Encoding Initiative (MEI) is an open-source project with a group of researchers including computer scientists, musicians, librarians, musicologists, and librarians who work together to formulate methods to digitally represent music (Roland et al 2014). The initiative has developed an eXtensible Markup language schema, also called MEI, which is a standards-based language, which uses XML. MEI has the capability to allow collaborative editing, and the ability to integrate with "online publication systems for the digital consumption of musical texts" (Sanchez-Jara et al. 2017).

This paper gives a report on my University of Glasgow Music Curation and Analytics project and will include a discussion of the different standards and methodologies that were used to curate, notate, encode, and analyse different types of musical data. My choice was to curate a music dataset that curates and analyses a selection of music from the soundtracks of the following three John Carpenter movies:

- Halloween (1978)
- The Fog (1980)
- Escape from New York (1981)

Included in the project is the following:

- html pages containing musical notation and metadata encoded in MEI language; audio examples of the musical notation; and links to examples of some of the visual analyses conducted during the project
- A dataset that contains different audio analyses, musical notation files, midi files, and audio files
- A written reflection on the project

The dataset was uploaded from my local repository onto GitHub, which is an online software configuration management system, which lets developers share and collaborate projects (Arora et al 2016).

The html home page can be found at: https://allan666gla.github.io/MCA/index.html

The GitHub repository can be found at: https://github.com/ALLAN666GLA/MCA

PART ONE: DIGITAL MUSIC NOTATION.

For the project, I decided to transcribe selected music from three John Carpenter Movies, which were produced around the same period – 1978 to 1981. The motivation for the theme of the project was from a personal admiration for the music, and because there was no similar collection available online, with the exception of musical transcriptions for sale, and the metadata already contained in databases such as IMDB. John Carpenter is not only famous for directing successful movies in the horror / fantasy genres, he is also famous for writing, composing and performing the music scores to many of his movies.

There are several software applications that make digital music transcription possible such as AnthemScore and Scorecloud. In this project MuseScore was the software chosen to use as it can export transcriptions to several different file types such as WAV, MIDI, MusicXML and MEI (https://musescore.org/en). To digitally transcribe from existing musical notation in MuseScore, perhaps in the form of printed sheet music, the process is arguably much easier than transcribing by just listening to the music, because all of the musical information is there to observe. The lack of availability of musical transcriptions of John Carpenter's music gave the project a creative feel so a lot of the techniques I used were based on my previous knowledge of music production software Ableton Live (https://www.ableton.com/en/).

I transcribed two pieces of music from each of the three movies:

- The Fog: Main Theme / Aboard the Seagrass Part 3
- Halloween: Main Theme / The Bogeyman
- Escape from New York: Main Theme / Across the Road

The first piece I transcribed was 'Aboard the Seagrass' from The Fog. The process of trying to transcribe the music in MuseScore was difficult at first because the music has no fixed tempo, so any attempt to transcribe the piece resulted in it playing out of time. In order to produce a transcription as close to the original as possible, I decided to use Ableton Live to analyse the audio wave of the original music and match the timing of the notes to the musical notes I entered with midi. In Ableton this matching of timings is possible as you can enter midi without a grid layout. Figure 1 shows the top layer as the original audio wave, and the second layer as the midi input.

The next step after creating a midi file of the music was to load it onto MuseScore, which translated the midi notation to a musical notation. When listening back to the music on MuseScore, the pitch of the notes and the timing of the notes sounded very similar to the original, which is what I tried to achieve by using Ableton. The downside is that a lot of new information appeared on the transcription, such as the variety of bar and note lengths. This arguably distorts the transcription as musicians may find the transcription confusing, disorientating and inaccurate. Figure 2 shows

the transcription of ‘Aboard the Seagrass’, with all the musical information derived from the midi.

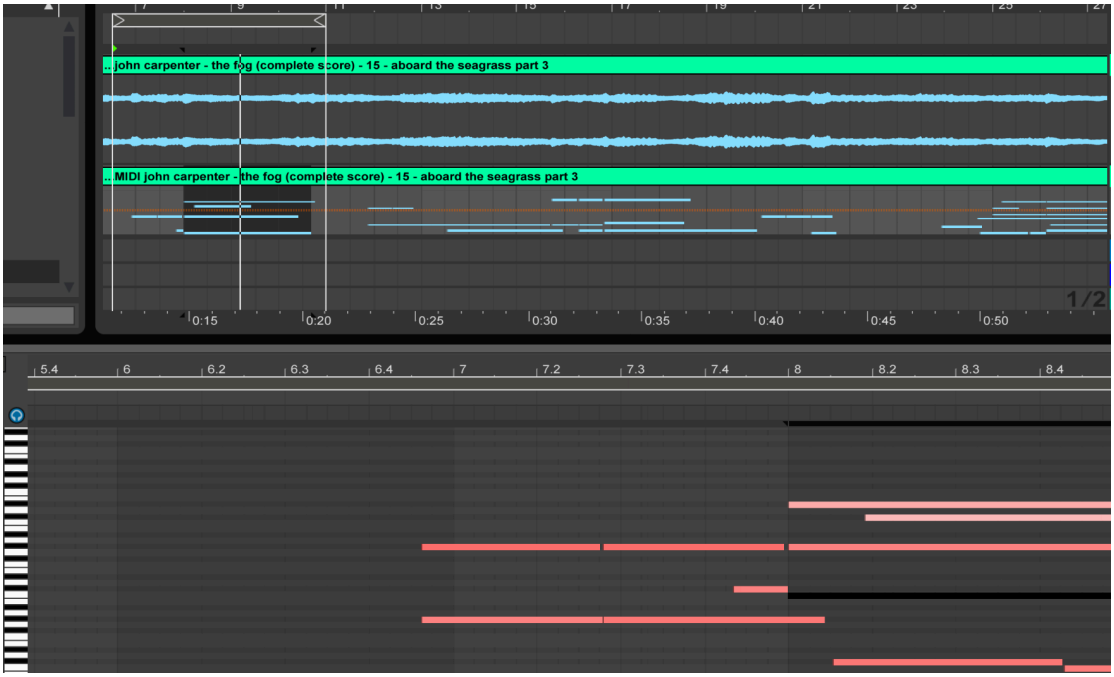

Figure 1: Audio analysis of ‘Aboard the Seagrass’ using Ableton Live

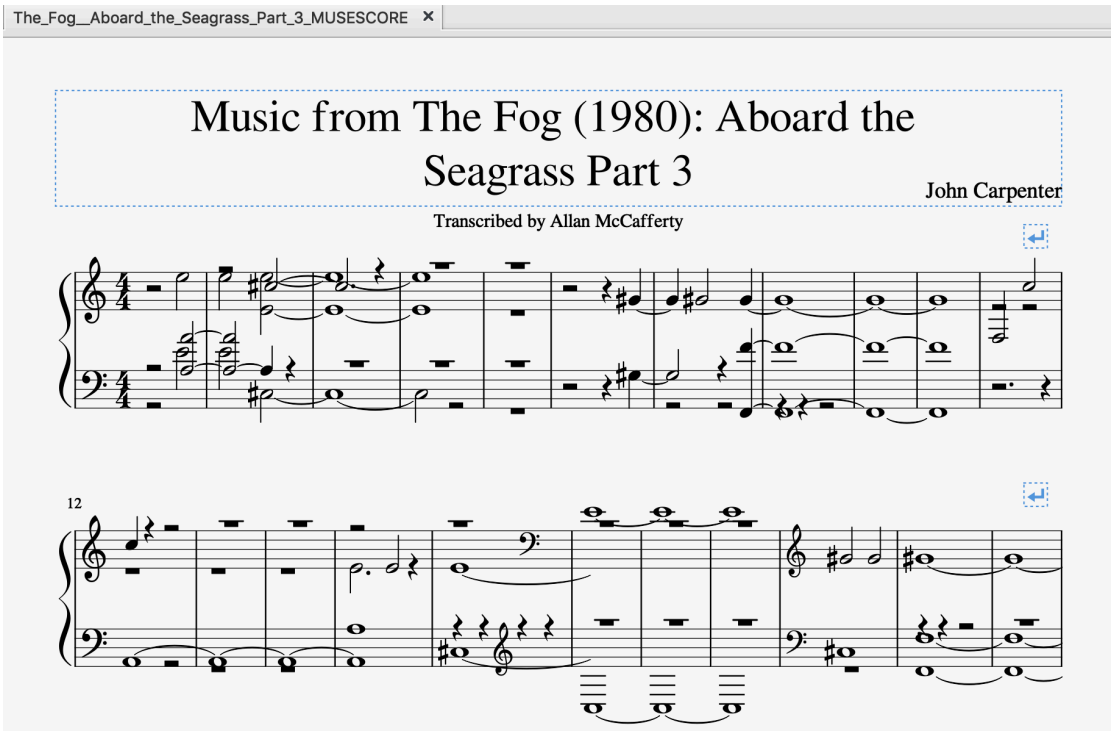

Figure 2: Transcription of ‘Aboard the Seagrass’ in MuseScore

PART TWO: MUSIC ENCODING / MEI

Before work could begin on encoding the music the MuseScore files had to be converted into the MusicXML format, which is a machine-readable Markup language used for sharing musical sheet music (Geertinger 2013). The MusicXML was then converted into an MEI file via the Verovio.org site (http://www.verovio.org/mei-viewer.xhtml). The MEI file can be viewed on the Verovio MEI viewer, which shows a webpage with musical notation. It can also be edited to add, delete or change musical data, and add metadata with a text-editor. For the project I used Atom (https://atom.io), and later in the project the MEI files are encoded into the html pages.

When comparing the MEI file with musical notation, I observed that there are several differences in terminology. For example, a 'bar' is called a layer in MEI, 'dur' is the duration of a note equated in numbers; 'oct' is an octave of note in numbers; and 'pname' is the name of the pitch on the chromatic scale. To test the how the coding reflects the notation. I edited some of the musical information such as the octaves, the pitch names, and the duration. I uploaded back into MuseScore and the music was different. Figure 3 shows the change in code from the original, to the new version. During this experiment I learned that while it is relatively simple to edit the music information, the full MEI file was very large which would make it difficult and time-consuming to code an entire MEI file.

ORIGINAL:

```
<layer xml:id="layer-0000001776135570" n="1">
                    <note xml:id="note-0000000039560439" dur="4"
oct="5" pname="g" stem.dir="down" />
                    <note xml:id="note-0000001732793335" dur="4"
oct="5" pname="c" stem.dir="down" />
                    <note xml:id="note-0000002114011406" dur="4"
oct="5" pname="g" stem.dir="down" />
                    <note xml:id="note-0000001408836129" dur="4"
oct="5" pname="c" stem.dir="down" />
                  </layer>
```

NEW VERSION:

```
<!-- change of code, notes changed from original in experiment --></-->
                  <layer xml:id="layer-0000001776135570" n="1">
                    <note xml:id="note-0000000039560439" dur="4"
oct="3" pname="g" stem.dir="down" />
                    <note xml:id="note-0000001732793335" dur="4"
oct="4" pname="c" stem.dir="down" />
                    <note xml:id="note-0000002114011406" dur="2"
oct="6" pname="a" stem.dir="down" />
                    <note xml:id="note-0000001408836129" dur="4"
oct="2" pname="c" stem.dir="down" />
                  </layer>
```

Figure 3: Change in MEI coding lab experiment

PART THREE: BASIC ANALYSIS

The first method of music analysis I conducted was performed in jSymbolic (http://jmir.sourceforge.net/jSymbolic.html), which is a software application for extracting data features that can be used in MIR (Mckay and Fujinaga 2006). Figure 4

is a screenshot, which shows some of the features in jSymbolic. The process of creating a jSymbolic analysis was as follows:

- Upload MIDI file to jSymbolic
- Decide the features to extract
- Select the 'Convert from ACE XML to CSV to give a CSV File for analysis in Microsoft Excel
- Extract features

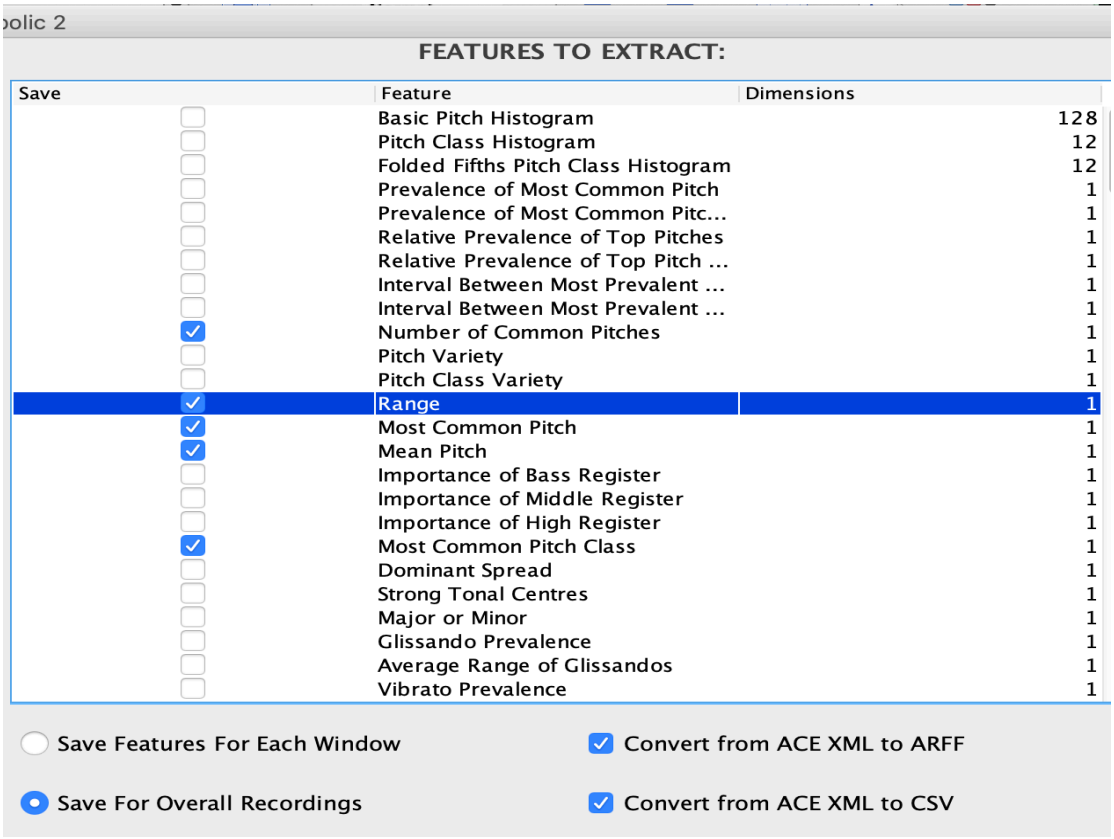

Figure 4: Example of Features in jSymbolic

Once the CSV File was extracted, I opened with Excel to observe a chart of the musical features (Figure 5), and then I converted the file into a 3D chart to give a better visualisation of the extracted features from jSymbolic (Figure 6). There are limitations to this analysis process due to the music being extracted from a MIDI file, which leaves out information such as dynamics, and sound information such as tone and timbre. There is also a bias with this analysis process as it is solely for a melodic analysis (Coronel 2013).

B	C	D	E	F	G	H	I	J	K	L	M
Number_of_	Number_of_	Number_of_	Number_of_	Range	Mean_Pitch	Mean_Pitch_	Most_Comm	Most_Comm	Interval_Bet	Pitch_Variab	Melodic_Inte
37	12	3	1	62	60	5.189	52	4	7	11.85	0.01952

Figure 5: Extracted features in Microsoft Excel

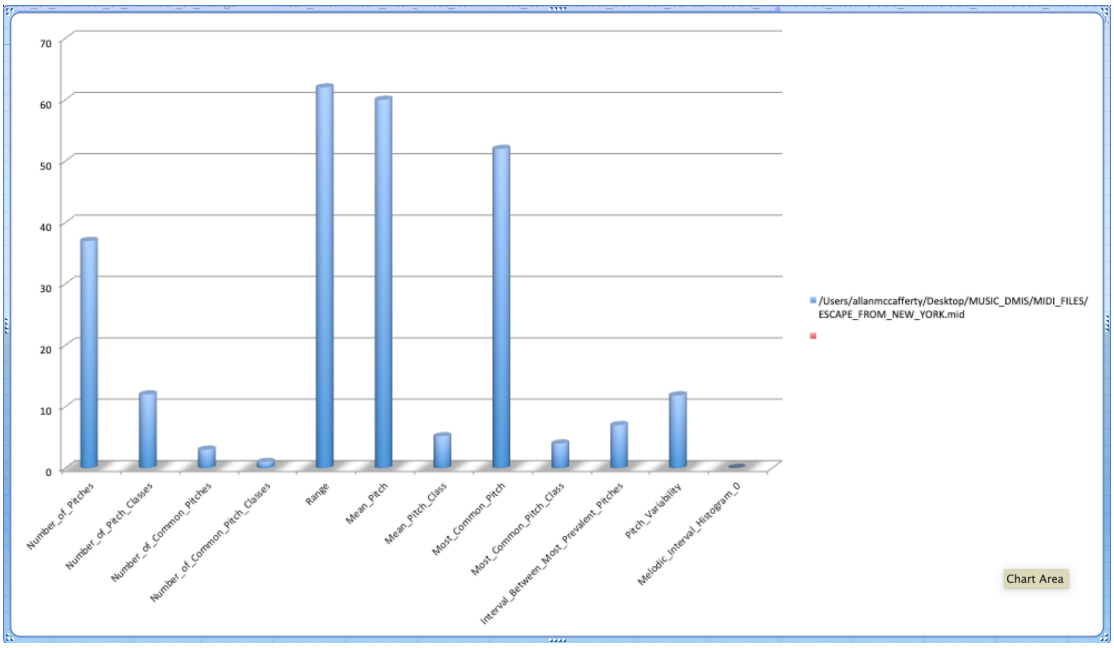

Figure 6: 3D chart of extracted features for analysis

The second method of music analysis I conducted was using Music21 software, which is a modular list of programs using Python coding language to extract and present data related to musical information (Tymoczko 2013). The Music21 analyses were conducted from the uploading the MEI file in Python, rather than the MIDI file.

I generated a histogram of the number of pitches (Figure 7), and a piano roll of notes quarter length of pitch (Figure 8).

```
In [7]: p.plot('histogram', 'octave', xHideUnused=False, yAxisLabel='Number of Pitches')
```

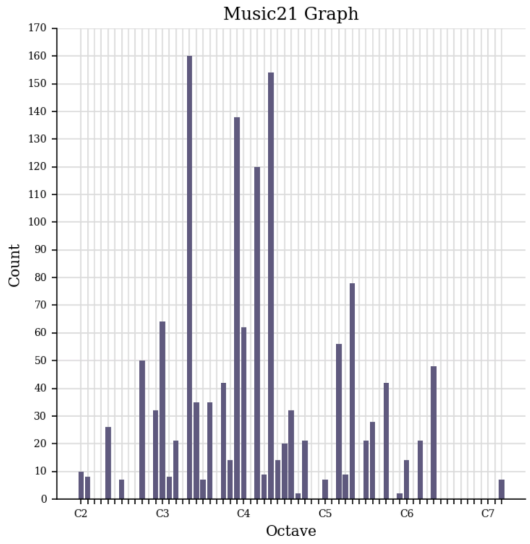

Figure 7: Histogram of number of pitches in Music21

```
In [6]: voice = p.parts[0]
        voice.measures(1,99).plot()
```

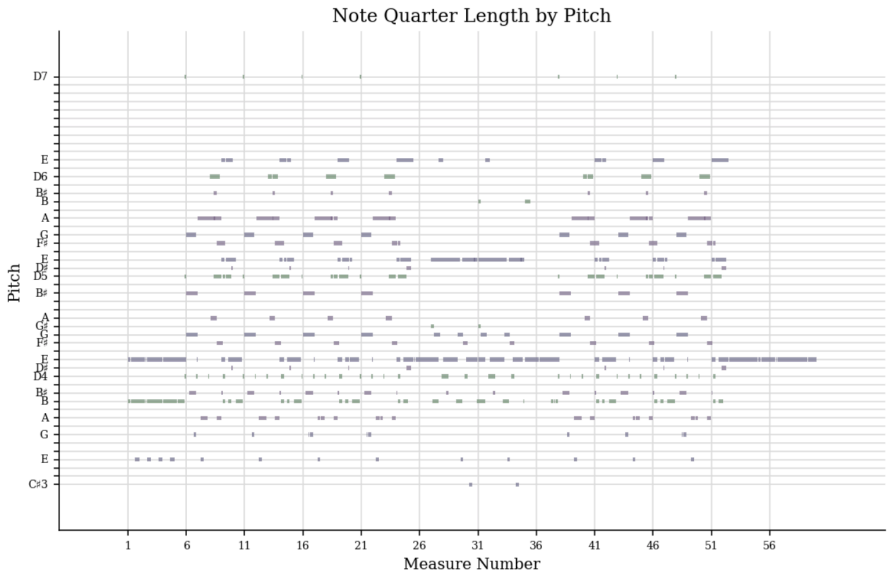

Figure 8: Piano roll of quarter notes by pitch in Music21

Figures 6-8 are analyses of the 'Main Theme' of Escape from New York and from my analysis I observed several things about the music:

- Every note in the chromatic scale is used, but two are much less frequent: C# is not used until the middle eight of the music. Perhaps this was used to increase the drama of the music
- The chord progression of the middle eight is Em, C, Am & C#. What I observed is that the note E is the common note in each of these chords.
- E is the dominant and most used note in the piece.

PART FOUR: DESCRIBING DIGITAL CONTENT WITH METADATA

This part of the project was to encode metadata into the MEI files, which would display on my html pages. The most important metadata I decided to use were the name of the piece and from which film/year; the name of the composer; the name of the encoder. This was possible by entering the metadata details into the MEI file (Figure 9), and, with the use of an xml:id attribute. For example, 'title1' is the name of the first part of the metadata and this is also shown in the html file next to 'textContent' (Figure 10), it was possible to link the metadata between the MEI file and the html file. Figure 11 shows how it the metadata is displayed on the halloween.html page.

```
<meiHead>
    <fileDesc>
        <titleStmt>
          <title xml:id="title1">Halloween: Main Theme (1978)</title>
          <title type="subtitle">
            <identifier>Encoded Music of John Carpenter</identifier>
          </title>
          <composer xml:id="title2">Composed by John Carpenter</composer>
          <respStmt>
              <resp xml:id="title3">Encoded by</resp>
              <resp xml:id="title4">Allan McCafferty</resp>
          </respStmt>
        </titleStmt>
        <seriesStmt>
          <title>AN ANALYSIS OF SELECTED MUSIC FROM JOHN CARPENTER MOVIES</title>
          <editor>Allan McCafferty</editor>
          <identifier>MCA-2018</identifier>
          <contents>
            <p>Halloween: Main Theme, Halloween: The Bogeyman, Escape from New York: Main Theme,
            Escape from New York: Across the Road, The Fog: Main Theme, The Fog: Aboard the Seagrass</p>
          </contents>
        </seriesStmt>
        <pubStmt></pubStmt>
    </fileDesc>
```

Figure 9: Halloween MEI file Metadata

```
/* Deal with the response */
xhr.onreadystatechange = function () {
  var DONE = 4
  var OK = 200
  if (xhr.readyState === DONE) {
    if (xhr.status === OK) {
      var parser = new DOMParser()
      var meiDoc = parser.parseFromString(xhr.responseText,"text/xml")
      document.querySelector("#meta").innerHTML += "<p>" + meiDoc.querySelector("[*|id='title1']").textContent + "</p>"
      document.querySelector("#meta").innerHTML += "<p>" + meiDoc.querySelector("[*|id='title2']").textContent + "</p>"
      document.querySelector("#meta").innerHTML += "<p>" + meiDoc.querySelector("[*|id='title3']").textContent + "</p>"
      document.querySelector("#meta").innerHTML += "<p>" + meiDoc.querySelector("[*|id='title4']").textContent + "</p>"
      document.querySelector("#meta").innerHTML += "<p>" + meiDoc.querySelector("[*|id='title5']").textContent + "</p>"
```

Figure 10: Halloween.html xml:id attributes

PART FIVE: MUSIC AS SOUND

For this part of the project, I worked with three tracks from different musical genres to compare and analyse the sound waves. They were downloaded from the Free Music Archive. Figure 12 is a table with relevant metadata about the music files.

ARTIST	TRACK	LENGTH	GENRE	BITRATE	CREATIVE COMMONS LICENCE ATTRIBUTION	FILE/ AUDIO FORMAT
KieLoKaz	Rainy Nights	05:55	SOUL-RnB, Soundtrack, Modern Jazz	320000	Non Commercial- No Derivatives	Wav
Ultracat	Space Love Attack	04:11	Electronic, House	306949	Non Commercial 3.0 International Licence	Wav
Yan Terrian	Rose Baba	20:30	Pop, Ambient, Instrumental	320000	Share Alike Licence	Wav

Figure 12: Metadata table of selected tracks

Links to tracks:

- 'Rainy Nights' <http://freemusicarchive.org/music/KieLoKaz/Walker_Traffic/Rainy_Nights_KieLoKaz_ID_159>
- 'Space Love Attack' <http://freemusicarchive.org/search/?quicksearch=space+love+attack>
- 'Rose Baba' <http://freemusicarchive.org/search/?sort=track_date_published&d=1&quicksearch=rose+baba>

Figure 13 is an audio analysis and comparison of KieLoKaz's "Rainy Nights" made using the Sonic Visualiser software (https://www.sonicvisualiser.org). Sonic Visualiser is an "application designed for musicologists, archivists, signal-processing researchers, and others who need to view and analyse music audio files" (Computer Music Journal writers 2008). The top layer is the raw WAV file and the second layer is a Spectogram. Figure 14 is a similar Sonic Visualiser analysis of Ultracat's "Space Love Attack", and figure 15 is a similar visualisation of Yan Terrian's "Rose Baba".

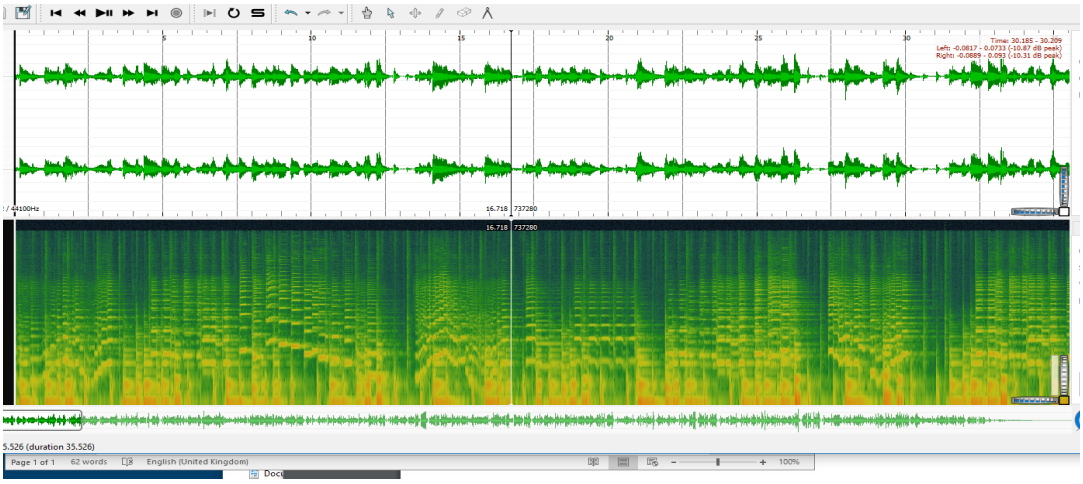

Figure 13: 'Rainy Nights' WAV & Spectogram

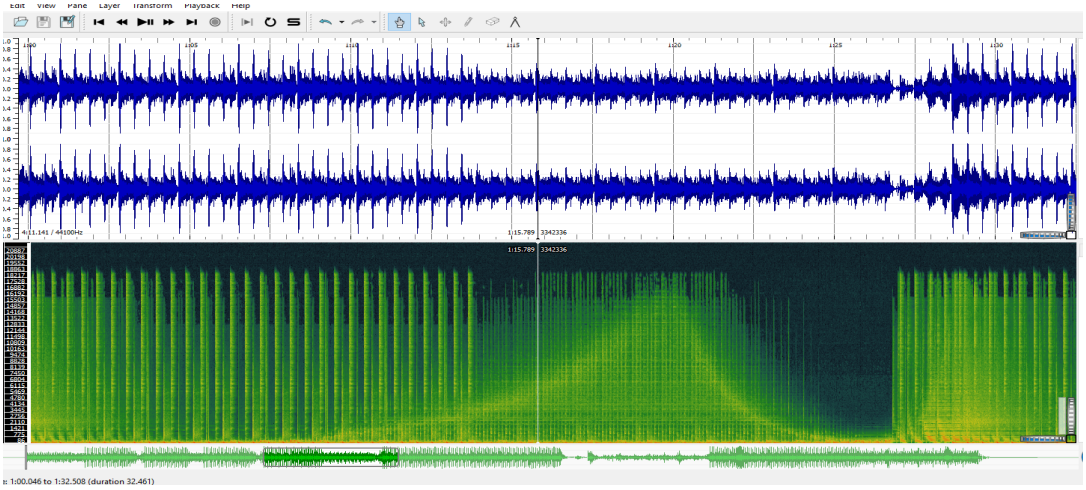

Figure 14: 'Space Love Attack' WAV & Spectogram

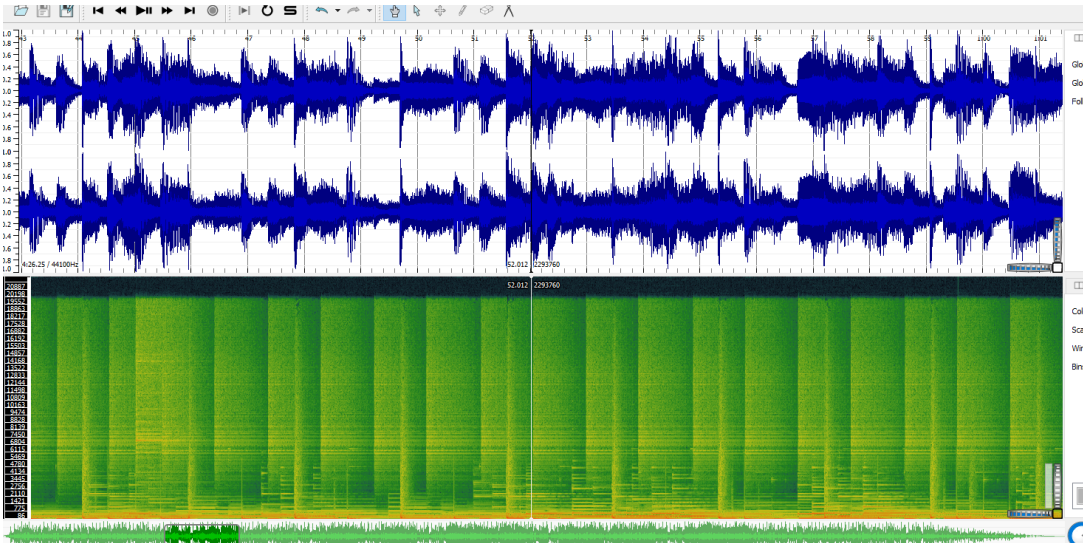

Figure 15: ‘Rose Baba’ WAV & Spectogram

One main advantage of the time=frequency analysis, as shown in the bottom layer of each image, is that it is possible to view and analyse more musical information than a waveform-based analysis. Some of the information that can be analysed in the spectrogram includes lengths of notes, melodies, notes, rhythms, chords, and timbre. Interestingly, in “Space love Attack”, in which the genre is Electronic House, the Spectrogram shows a rather clear image of the moment when the build-up is taking place. The build-up, or sometimes known as a break is when the beats have temporarily been taken out the song to create a powerful impact when they come back which is very common in Dance music. In this case is it is highlighted with a frequency sweep, which ascends to a peak, then descends just before the music and beats start back. This was a quick analysis of the track, in which the genre of the music was perhaps easier to distinguish without even listening to the music

PART SIX: SIMILARITY AND TRANSCRIPTION

This part of the project was to perform a similarity analysis of a selection of 10 tracks. The data was extracted from the mean Chroma features using Music21 in Python language. Figure 16 is comparison of the 10 tracks in the collection (tracks 7,8, and 9 were replaced by three of my chosen tracks). The 10 plots in the image show the notes charted from the chromatic scale of each track.
Tracks 0, 1, 2, and 3 are from the classical genre; tracks 4, 5, and 6 are from the rock genre; track 7 is from the jazz genre; track 8 is from the pop genre; and track 9 is from the ambient pop genre.

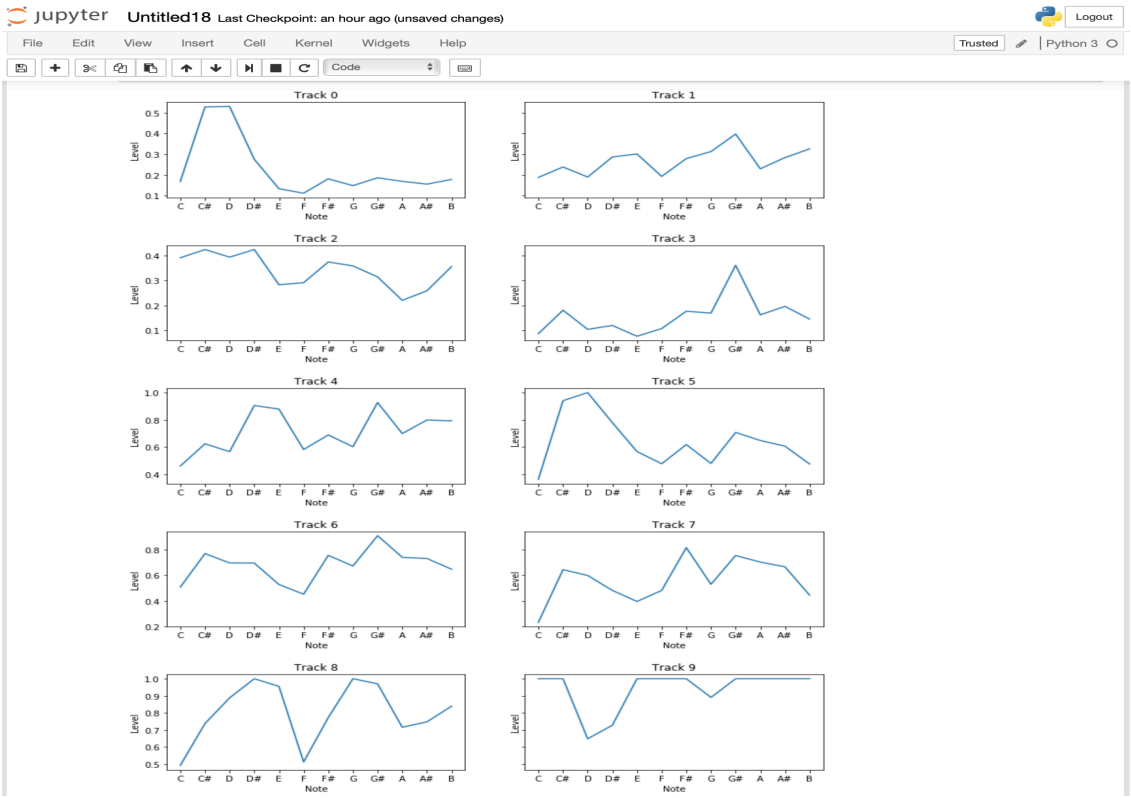

Figure 16. Mean representation chart

This is a comparison of the 10 tracks using the Chroma-features matrix in Music21.

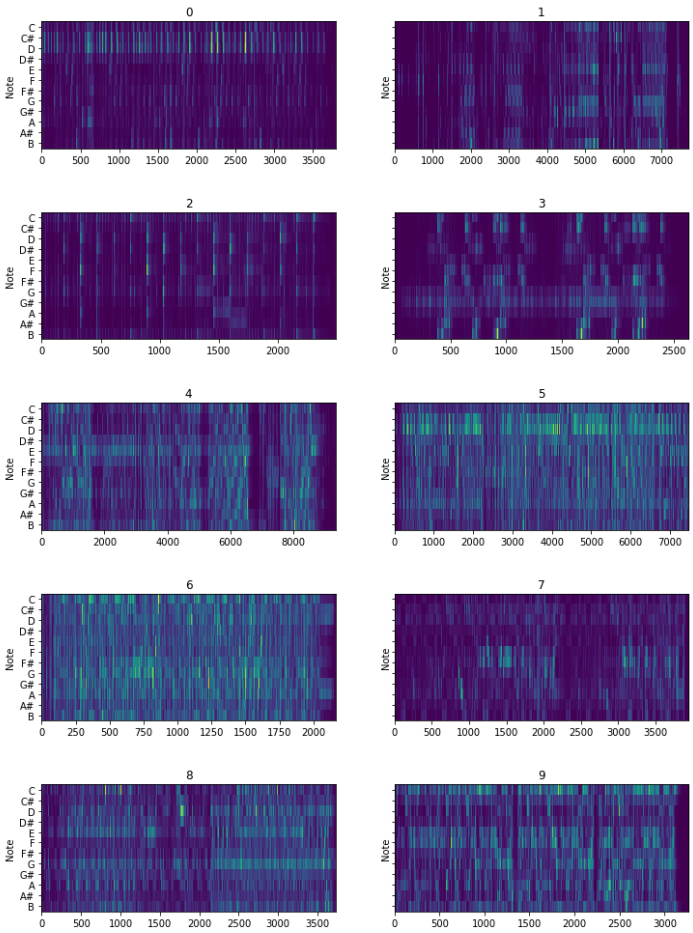

Figure 17. Chromogram comparison

Figure 17 is a comparison of the 10 tracks, which shows 10 Chromograms made using Python. From observing and comparing the 10 plots, it was possible to notice that the different tracks have different lengths, different dynamics, and different amounts of music information. Tracks 4 to 9 (with the exception of track 7) appear quite similar in the amount of notes and information. One could guess they were from similar genres, when in actual fact they are considered to be different genres. Tracks 4, 5 and 6 are from the rock genre and are very similar to one another. Track 8 is from the pop genre, and track 9 is from the ambient-pop genre, and both are very similar to one another. Track 7 is from the jazz genre and appears very similar to tracks 0-3, which are from the classical genre. It is possible that because of the acoustic instrumentation, the spaces between the notes of the music, and the gentler musical dynamics of the jazz track, that it appear more similar to the classical tracks than the rock and pop tracks.

The next part of the project was to compare original musical notation from a MuseScore file with a MuseScore file converted into a Wav file, then changed into a midi file via the Sonic Visualizer software, then loaded back into MuseScore as a new musical transcription. The idea is to see what information, if any, is lost following the conversion processes.

Figure 18 shows the original MuseScore transcription with the three different instrumentation parts – The Electric Bass, the Grand Piano, and the Contrabass. It

shows that the piece is in 4/4; the BPM is 120; the clef is a bass clef; has 5 bars; the notes are all either quarter notes or crotchets; and there are several rests.

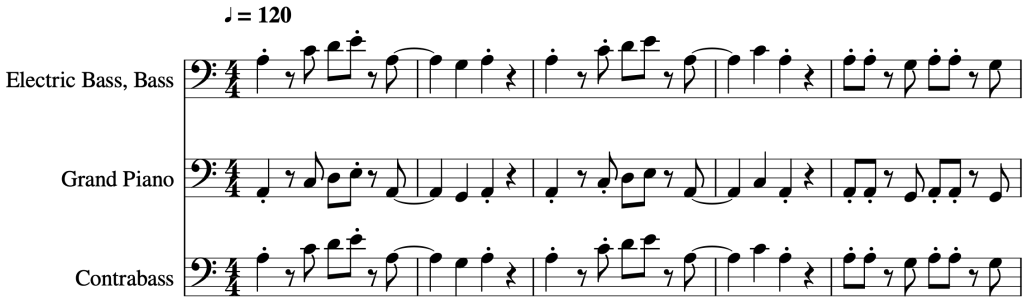

Figure 18. Original MuseScore Transcription

There are several differences in the second transcription (Figure 19). Firstly it is obvious that the instrumentation has changed the transcription to contain one treble clef and one bass clef instead of the 3 bass clefs from the original. The other differences are that some of the notes have changed from the original's crotchets and quarter notes, into minims, and that augmentation dots have appeared. When playing back the second piece it appears to be playing chords rather than the singular notes of the original.

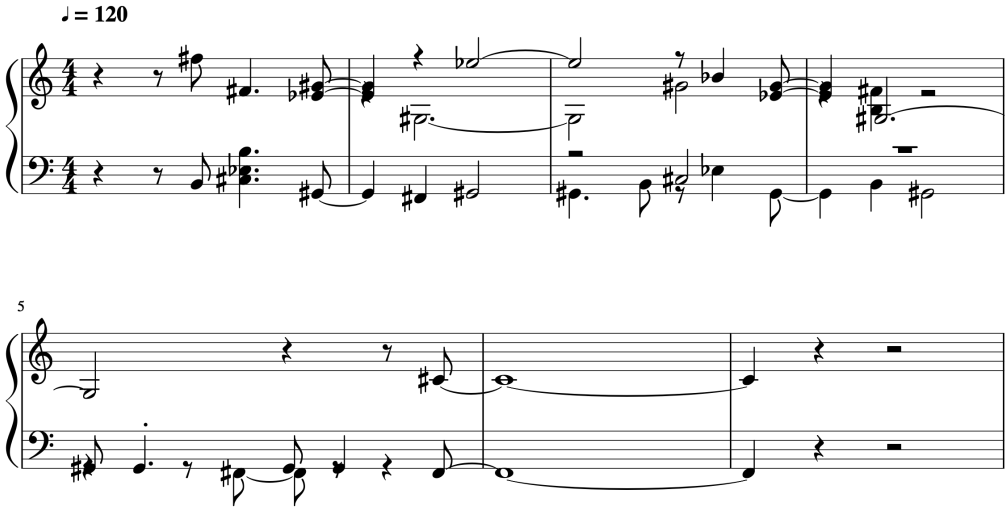

Figure 19. Second transcription

CONCLUSION

On reflection, I have learned that in curating a large music dataset, a standards-based approach for encoding musical data such as MEI or MusicXML is valuable as it is easy to store, retrieve, share, and edit the information to keep continuity and similarity in the metadata. The limitations I encountered in Musescore when transcribing a piece of music without a fixed tempo made the software feel too advanced for a beginner, or a even data scientist who is trying to preserve music digitally. I found limitations in my analysis when using jSymbolic, as it was purely a melodic analysis and did not analyse the actual audio.
Integrating the Sonic Visualiser, Music21, and jSymbolic applications used during this project into one consolidated platform would be a huge improvement for the analysis of music.

BIBLIOGRAPHY

Andrei D. Coronel, "Building an Initial Fitness Function Based on an Identified Melodic Feature Set for Classical and Non-Classical Melody Classification", *International Conference on Information science and Applications (ICISA),* (June 2013), [online], <https://ieeexplore-ieee-org.ezproxy.lib.gla.ac.uk/document/6579433> [Accessed December 6th 2018]

Axel Teich Geertinger, "Turning Music Catalogues into Archives of Musical Scores – Or Vice Versa: Music Archives and Catalogues Based on MEI XML", *Fontes Artis Musicae*, (January 2014), 61, 1, [online] https://www-jstor-org.ezproxy.lib.gla.ac.uk/stable/24330408?pq-origsite=summon&seq=1#page_thumbnails_tab_contents [Accessed December 6th 2018]

Chris Mckay, and Ichiro Fujinaga, "jSymbolic: A Feature Extractor for MIDI Files", *Proceedings of the International Conference on Music Information Retrieval*, (2006), p. 302-305, [online], <http://jmir.sourceforge.net/publications/ICMC_2006_jSymbolic.pdf> [Accessed Dec 2nd 2018]

Computer Music Journal Writers, "Products of Interest", Computer Music Journal, 32, 1, (Spring 2008), p. 111, [online], <http://muse.jhu.edu.ezproxy.lib.gla.ac.uk/article/233906> [Accessed December 7th 2018]

Dmitri Tymoczko, "Review of Michael Cuthbert, Music21: A Toolkit for Computer-aided Musicology", *MTO: A Journal of the Society for Music Theory*, 19, 3, (August 2013), [online], <http://mtosmt.org/issues/mto.13.19.3/mto.13.19.3.tymoczko.html> [Accessed December 6th 2018]

Jason Palmer, "Analytics at Spotify", *Spotify Labs*, (May 13th 2013), [online], https://labs.spotify.com/2013/05/13/analytics-at-spotify/ [Accessed December 2nd 2018]

Javier Merchan-Sanchez-Jara, Jose Antonio Cordon Garcia, and Raquel Gomez Diaz, "Towards a Hypermedia Model for Digital Scholarly Edition of Musical Texts Based on MEI (Music Encoding Initiative) Standard; Integration of Hidden Traditions Within Social Editing Paradigm", *Association for Computing Machinery*, (October 2017), p. 18-20, [online], http://delivery.acm.org.ezproxy.lib.gla.ac.uk/10.1145/3150000/3145446/a99-merchan-sanchez-jara.pdf?ip=130.209.6.61&id=3145446&acc=ACTIVE%20SERVICE&key=C2D842D97AC95F7A%2EC612DDE1DA0E84ED%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1544097850_35b0668f9d104ab5b7073ba964c9f43d [Accessed December 4th 2018]

Perry Roland, Andrew Hankinson, and Laurent Pugin, "Early Music and the Music Encoding Initiative", Early Music, 42, 4, (November 1st 2014), p. 605-611, [online], https://academic-oup-com.ezproxy.lib.gla.ac.uk/em/article/42/4/605/2928467 [Accessed December 6th 2018]

Rita Arora, Sanjay Goel, and Ravi Kant Mittal, "Supporting Collaborative Software Development over GitHub". *Software: Practice and Experience*, 47, 10, (December 16th 2016), [online], <https://onlinelibrary-wiley-com.ezproxy.lib.gla.ac.uk/doi/full/10.1002/spe.2468> [Accessed December 6th 2018]

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere., "The Million Song Dataset". *In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, (2011), [online], https://labrosa.ee.columbia.edu/millionsong/ [Accessed December 2nd 2018]