

SoundMap: A Visualization Tool to Explore Multi-Attribute Sound Data

Elizabeth Reid
reid42@cs.ubc.ca

Mifta Sintaha
msintaha@ece.ubc.ca

Nichole Boufford
ncbouf@cs.ubc.ca

I. INTRODUCTION

NOTE: Red text indicates revisions since project proposal.

In the field of signal processing and audio analysis, there are many standard techniques for visualizing a single sound file. For example, simple waveplots can be used to see the original amplitude of a sound signal with respect to time, or spectrograms can be used to see the breakdown of individual frequencies within a signal. Other techniques, such as chromagrams or Tonnetz networks, can be used to visualize pitch relationships. However, these visualization techniques are not always scalable. For example, spectrograms encode frequency using vertical position, time using horizontal position, and amplitude with colour—while this is a useful method of exploring sound frequency features in a single file, trying to compare even ten spectrograms on a single screen becomes challenging. Users must either reduce the size of the spectrograms, making it difficult to differentiate features, or must scroll between them and thus cannot directly compare all the files at once. It is also difficult to visualize the relationships between audio and non-audio attributes using canonical techniques, let alone compare how multiple combinations of these attributes relate to each other.

Additionally, many of the canonical audio visualization techniques are not accessible to novice users. It is difficult to both read and create these visualizations directly without at least a minimal understanding of audio analysis and signal processing, and pitfalls such as choosing the wrong sample rate or window function can distort results in unexpected ways. However, both expert and non-expert users may still wish to explore audio data visually. For example, one might want to identify relationships between vocalizations and animal behaviour, explore differences in speech among people with voice pathologies, investigate similarities between music genres or artists, or study recordings of different instruments for pedagogical reasons. In all cases, users would benefit from a system that can handle a large number of sound files, is easy to use and understand, provides flexibility to show relationships between audio attributes and non-audio attributes, and allows both broad and granular exploration of the sound files in their datasets.

We propose the visualization tool *SoundMap* to satisfy these requirements, which allows users to explore many multi-attribute sound files simultaneously. Taking inspiration from previous work in faceted browsing [6, 13], *SoundMap* al-

lows users to visualize both individual audio and non-audio attributes as well as combinations of attributes. To facilitate the needs of both novice users and experts, **beeswarm charts** and bar charts are used to visualize broad relationships among many sound files, and interactions allow users to “zoom in” to a single sound if they want to see a more detailed spectrogram. To illustrate how *SoundMap* can be used for different types of sound datasets, we evaluate the tool on two different example datasets: CatMeows [7], and the Free Music Archive (FMA) [4].

II. PERSONAL EXPERTISE

Elizabeth has the most background experience working with audio data—she is currently taking the CPSC 554X Machine Learning and Signal Processing class, and completed a bachelor’s thesis on mathematically modelling the singing voice, for which she wrote a program implementing an existing model of voice vibrato that required both analyzing and synthesizing sound data. She also has a bachelor’s degree in music, and therefore has strong knowledge of music theory and structure. She is a master’s student in the Department of Computer Science, and has worked in a number of programming languages in the past, including some minimal work in JavaScript.

Mifta is a master’s student in the Department of Electrical & Computer Engineering and her research is in the area of Software Engineering. She has over 4 years of experience working in the industry as a Full Stack Software Engineer and has previously worked with ReactJS and D3 library.

Nichole is a master’s student in Computer Science. She has software development experience including 1 year of experience as a Full Stack Software Developer using ReactJS.

III. RELATED WORK

A. Visualizing Sound Files

To visualize a single sound file, common techniques such as waveforms or waveplots, spectrograms, and chromagrams are typically used. Sometimes more specialized visualizations are also adopted, particularly for music data, such as Tonnetz grids or isochords as shown in [2], which allows users to visualize more complex tonal relationships within music. Similarity matrices can also be used to analyze a single sound file, which can be particularly useful for longer files [3]. To compare a small number of sound files together, the chosen visualization is usually layered horizontally or vertically: for example, the free to use *Sonic Lineup* [14] application is one such example,

and many other audio programs like *Audacity* [1] also display sound file visualizations like waveplots vertically.

In cases where one must visualize a large number of sound files, similarity measures are usually employed. The resulting similarity mapping is typically arranged into a network [3, 9], or displayed in a spatial field [3]—however, any underlying information about how the similarity between items was calculated is usually hidden from users. For example, the content based graph visualization described by Muelder, Provan, and Ma [9], and the *Islands of Music* system by Pampalk et al. [11] are two examples of using similarity to visualize a large number of music files. Though the first uses graph based visualizations and the second a topographical visualization technique, both systems cluster similar songs together and display metadata information such as album art or song genre. However, it is not clear from either visualization how specific audio features contribute towards similarity clustering, or even what audio features are present in the data at all. Therefore, while similarity approaches can be useful to organize large datasets, they do not easily facilitate data exploration tasks.

B. Faceted Browsing

Lee et al. [6] introduces *FacetLens*, which is an interactive visualization tool for exploring relationships and trends to make sense out of faceted data. It uses linear facets which allows users to identify trends and compare them simultaneously. Our audio datasets also contain many faceted metadata attributes in addition to sound attributes, requiring a tool that can effectively represent the facets along with the sound attributes in a meaningful way and help identify the relationships between non-audio and audio attributes. We also plan to add multiple views to compare trends simultaneously, similar to *FacetLens*. However, *FacetLens* is only suitable for general datasets which do not provide any audio specific views, like a spectrogram. It also does not have any summary view to show the aggregated value of the quantitative attributes grouped by the categories.

C. Waveform Visualization

Similar to sounds, haptic data can also be visualized as waveforms. Seifi, Zhang, and MacLean [13] show the visualization of multiple vibrations. This work targets novice users, providing an easy to understand interface. Additionally, they provide the ability to view many marks at a time and allow filtering by categorical attributes, revealing that faceted browsing techniques can be effectively applied to sound-like datasets. One of our goals is to create an abstraction from waveforms to sound attributes to enable viewing of many sounds files at once. This is similar to some of the views that *VibViz* provides for viewing a collection of vibrations, such as using coloured dots on a 2D axis.

IV. DATA AND TASK ABSTRACTION

A. Domain

SoundMap is a tool to support audio analysis, a domain which traditionally uses specialized visualization techniques

such as spectrograms, waveplots, or chromagrams to visualize audio attributes. In audio analysis, sound attributes of interest may vary depending on context. For example, fundamental frequency (i.e., the main pitch we recognize in a sound, like a musical note) is a common attribute of interest for short audio samples like a single note played on an instrument, a single spoken word or vowel sound, or a single animal vocalization. However, fundamental frequency is a much less useful attribute for audio files that contain many overlapping sounds (there may be multiple fundamental frequencies that are difficult to distinguish from one another) or are longer than a few seconds (fundamental frequency may change significantly over the course of the entire audio file). Music is one such example, as songs usually contain multiple instruments and/or vocals, and generally last several minutes. Common attributes relevant to music analysis include chroma features representing the strength of the 12 semitone pitch classes throughout a piece of music, mel-frequency cepstral coefficients (MFCC's), which are often used in speech recognition [5] or music information retrieval (MIR) [15] tasks, as well as other spectral features such as spectral bandwidth, spectral rolloff, and zero-crossing rate.

Additionally, sound files may contain several metadata features. These may include contextual information, such as the situation in which the sound was recorded, technology used to capture the recording, or information about what or who was recorded. As an example, an audio file for a song may have relevant metadata for the song's artist, album, language, or genre. As metadata features vary between audio datasets, a single dataset may have several metadata features with just a few levels each, a few metadata features with several levels, or any other such combination.

People interested in exploring an audio file dataset may be interested in audio attributes, metadata, or both. Additionally, despite many traditional methods being tailored towards expert users, novices may also wish to visualize relationships in audio data. Thus, *SoundMap* is intended to support both novice and expert users who wish to explore any audio dataset that contains both metadata and sound attributes.

B. Data

As discussed in the domain description, audio data may have a variety of sound and external attributes of interest. To ensure that *SoundMap* can effectively visualize different kinds of audio datasets, we selected two example datasets with significantly different qualities to evaluate the tool.

1) *MeowAnalysis*: The first dataset of interest was derived from the CatMeows dataset compiled by Ludovico et al. [7]. The original dataset consists of 440 short .wav files, each containing a single cat vocalization. Metadata information, such as the recording scenario stimulus, cat ID, owner ID, cat breed, and sex, is encoded in the file name. Guided by previous work on cat vocalization studies [12, 16], we derived five relevant audio attributes from the original dataset. The Python audio processing library *Librosa* [8] was used to estimate mean, maximum, and minimum fundamental frequency for

TABLE I
CLASSIFICATION OF ATTRIBUTES IN THE MEOWANALYSIS DATASET.

| Attribute | Description | Type | Items/Range |
|------------------|---|--------------|-------------------------|
| Cat ID | Unique ID corresponding to each cat | Categorical | 21 |
| Owner ID | Unique ID corresponding to each cat's owner | Categorical | 12 |
| Stimulus | Situation in which the meow was recorded. Situations included brushing, isolation in an unfamiliar environment, and waiting for food. | Categorical | 3 |
| Breed | Cat breed, either Maine Coon or European Shorthair | Categorical | 2 |
| Sex | Specifies whether the cat is male or female, and has been spayed or neutered | Categorical | 4 |
| Mean Fundamental | Mean of the estimated fundamental frequency in Hz (pitch) for the entire meow. | Quantitative | 124.589 – 1122.891 |
| Max Fundamental | Max value of the estimated fundamental frequency in Hz (pitch) for the meow. | Quantitative | 456.570 – 2205.000 |
| Min Fundamental | Min value of the estimated fundamental frequency in Hz (pitch) for the meow. | Quantitative | 21.554 – 26.957 |
| Peak Frequency | Estimated value for peak frequency in Hz (loudest frequency that occurred during the meow). Note this may occur at an overtone higher than the fundamental frequency. | Quantitative | 450.000 – 6960.000 |
| Duration | Total meow time in seconds, not including any silences before, after, or during the meow. | Quantitative | 0.006 – 1.847 |
| | | | Total: 440 meows |

each vocalization. Analysis of the fast Fourier transform was also used to estimate peak frequency, and signal strength was used to differentiate between noise and meows in order to calculate duration. The resulting audio attributes were combined with metadata attributes into our final dataset, which we designate as the MeowAnalysis dataset. A detailed description of this dataset is shown in Table I.

2) *SongAnalysis*: Our second dataset of interest was derived from the Free Music Archive (FMA) dataset created by Defferrard et al. [4], which was originally created for MIR and machine learning tasks. This dataset already consisted of several pre-processed audio attributes for over 100,000 tracks. While we intend for *SoundMap* to support a large number of sound files, we are also limited by metadata feature levels (for example, the raw dataset also contains over 16,000 artists). Therefore, we randomly selected a subset of the original dataset with an upper limit of 15 artists. This resulted in a more manageable dataset of 71 total songs, which we call the *SongAnalysis* dataset. Additionally, several of the original audio features were removed, as they contained very little variance and would therefore not be useful to visualize. We also retained only the most relevant metadata features, such as artist, album, and genre. A detailed classification of this dataset is shown in Table II.

Notably, the MeowAnalysis dataset contains very short audio samples, and the resulting audio attributes are all relatively simple. However, this dataset contains a much larger total

number of elements. On the other hand, the *SoundAnalysis* dataset contains fewer total elements, but the length of the audio files resulted in numerous audio attributes that are closely linked, such as the 12 chromagram semitone attributes, or the seven Tonnetz attributes. Recognizing that meaningful audio attributes are context dependent, and to keep the scope of our project manageable, we assume that users of *SoundMap* wish to visualize audio data that has been pre-processed as a .csv file consisting of metadata and individual audio attributes, and treat each audio attribute independently. In order to display spectrograms, we also require an additional **two columns** in each dataset containing an array representation of the audio file data **and the sample rate of the audio file, respectively**. **As these array representations can potentially consist of hundreds of thousands of samples for just a few minutes of audio data, we limit our arrays to at most 15 seconds of sound samples. While this limitation is not ideal, the alternative would require users to upload each audio file to our app for processing, which could take over 10 minutes in the case of the *SongAnalysis* dataset. We also note that in the case of longer files, these shortened spectrograms serve as a kind of “visual preview” of song details, similar to song preview features used by companies such as Spotify.**

C. Tasks

At a high-level, *SoundMap* is intended to support the exploration of multi-attribute sound data. Specifically, users

TABLE II
CLASSIFICATION OF ATTRIBUTES IN THE SONGANALYSIS DATASET.

| Attribute | Description | Type | Items/Range |
|-------------------------|---|--------------|------------------------|
| Album | Album the song belongs to. Some songs are single tracks, in which case the album is described as N/A. | Categorical | 22 |
| Artist | Song artist. | Categorical | 15 |
| Genre | Genre of the song. Note that some songs have not been assigned a genre, in which case the genre is described as unknown. | Categorical | 6 |
| Mean STFT Chroma | Chromagram values calculated over the short time Fourier transform for the entire song. Note that there are 12 separate chromagram attributes, one for each semitone note name (C, C#, D, D#, E, F, F#, G, G#, A, A#, B). The range for each attribute is approximately the same. | Quantitative | 0.1 – 0.6 |
| Mean Tonnetz | Projection of chromagram values onto a 6-dimensional basis, representing harmonic relationships (perfect fifth, minor third, and major third). There are therefore 6 separate tonnetz attributes, one for each basis element. The range for each attribute is approximately the same. | Quantitative | -0.03 – 0.04 |
| Mean Spectral Bandwidth | Difference between the highest and lowest frequency in the spectrum. | Quantitative | 467 – 2700 |
| Mean Spectral Contrast | Contrast in energy between the top quantile (peak energy) to that of the bottom quantile (valley energy). This is calculated over 7 frequency bands (from low to high frequency). Each of the seven attributes has approximately the same range. | Quantitative | 15.2 – 32.1 |
| Mean Spectral Rolloff | The center frequency for a spectrogram bin where at least 85% of the energy of the spectrum is contained in this frequency bin and the bins below. | Quantitative | 534 – 4460 |
| Mean Zero Crossing Rate | The rate at which a signal (e.g., as a waveform) crosses the line $y = 0$. | Quantitative | 0.0168 – 0.124 |
| | | | Total: 71 songs |

can explore how metadata attributes and audio attributes are related, as well as view individual items in detail. We organize specific user tasks into the abstract task categories *analyze*, *search*, and *query*, as described in [10].

1) *Analyze*: These tasks are centered around data consumption and discovery.

- Audio feature analysis: The user can visualize all sound files with respect to a single audio attribute. For example, they can see the mean frequency distribution of all files in their dataset, and can determine the average frequency for most of the files.
- Audio and metadata analysis: The user can visualize all sound files with respect to a single audio attribute and a single metadata attribute. For example, they can see whether breed has any effect on the distribution of mean frequency in the MeowAnalysis dataset.
- Detailed individual analysis: By selecting a single sound within the larger overview, the user can see more detailed audio information for the sound file.

2) *Search*: These tasks are centered around data exploration.

- Browse an area: With a known location in mind with respect to either metadata or audio data, the user can investigate sounds within that area, opening detailed views if desired. For example, a user looks for pop songs with high spectral bandwidth.
- Outliers: The user looks for unexpected outliers within their dataset. For example, a user checks for any sound files that have an abnormally high or low fundamental frequency.

3) *Query*: These tasks are centered around comparison and filtering, particularly between multiple metadata and audio attributes.

- Visualize multiple audio attributes: Users can compare multiple audio attributes at the same time, and track the location of a single sound or group of sound files across all attribute displays. For example, a user can compare a meow’s frequency and duration together.

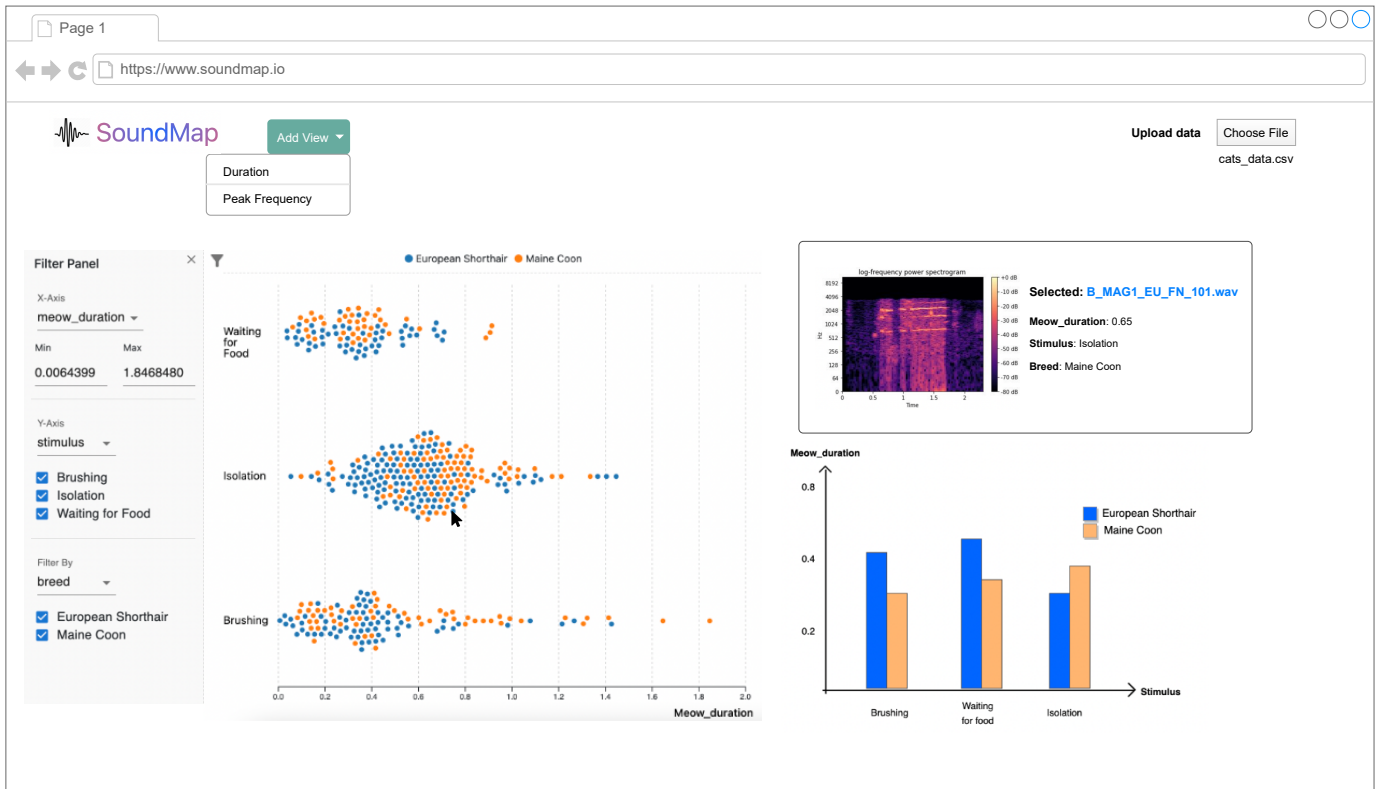


Fig. 1. Landing page of *SoundMap* containing the Overview, Detailed view and Summary

- Visualize multiple metadata attributes: Users can compare multiple metadata attributes at the same time, with respect to one or more audio attributes. For example, a user arranges songs by genre, and additionally highlights songs that belong to a specific album, all with respect to duration.
- Filter: Users may filter both metadata and audio attributes. For example, a user displays only one type of cat breed, and displays only sounds that have a frequency between 500 Hz and 600 Hz.

V. PROPOSED SOLUTION

A. Overview

To encode the relationship between categorical and quantitative attributes at a time, we plan to use a **beeswarm chart** chart where points are used as marks to encode the individual sound files. **Beeswarm charts are useful for spreading out dot points to avoid overlap, especially when one of the axes shows categorical attributes.** The horizontal axis will represent the scalar attribute of interest and each sound file will be represented by a mark on the plot. The vertical axis will be used to group data points by categorical attributes. For example, if there were three categories in the selected categorical attribute, the marks would be separated into three distinct positions on the vertical axis. Each of the axes in the **beeswarm chart** will be configurable through dropdown menus. The **beeswarm chart** values in the y-axis can be filtered out

using checkboxes shown in **Fig. 1**. The x-axis is configured through an input field by increasing or decreasing the range of the min and max values. The values can be further filtered by a second dropdown, which color encodes the data points based on the categorical attribute chosen. **To address scalability issues,** at most 6 options can be selected such that 6 different colors can be shown at a time. Each change in the dropdowns and checkboxes will reload the graph with an animation effect to indicate change. **The page layout will be responsive to the data in order to accommodate various attributes and its levels within the view.**

B. Detailed View

A detailed view of the sound file will be shown **in a separate view beside the beeswarm chart.** This **detailed view** will be shown when a single point is clicked, which will then display a spectrogram of the sound as well as other details like the categorical attributes and value it corresponds to in the graph.

C. Summary View

A direct summary of all the filters applied to the **beeswarm chart** overview can be visualized in a grouped bar chart. The x-axis will have the categorical attribute used in the **beeswarm chart** y-axis, and the y-axis of the bar chart will show the quantitative attribute of the **beeswarm chart** x-axis. The bars will be grouped by the categorical filter applied on the **beeswarm chart** view and color encoded in the same way.

D. Multiple Views

Additionally, multiple views can be shown at the same time by adding views of each of the quantitative attributes in a grid view. This can be helpful in comparing all the data and observing the relationships between multiple audio and non-audio attributes at a glance. We plan to add a toolbar to control the types of graph shown at a time. For instance, after a view with another quantitative attribute is added, it will duplicate the current view and be displayed below with the chosen quantitative attribute. To compare a particular type of chart easily, the user can select from a toolbar the type of chart to view in a juxtaposed manner.

VI. IMPLEMENTATION APPROACH

We plan to develop the visualization system as a web application using the React framework and d3 visualization library. We also plan to use Material-UI for adding the UI elements like dropdowns, input boxes, checkboxes, buttons etc. Building it as a web application adds the ability to upload multiple datasets, thereby making the tool more dynamic.

VII. USAGE SCENARIO

Imagine you are a research veterinarian with an audio dataset of cats having certain attributes (breed, gender etc.) in various stimuli, and you want to visualize the relationship between each of the attributes to identify patterns. To view the effect of various attributes on the meowing of cats [7], the user of the tool is taken to the landing page of *SoundMap*. *SoundMap* takes in a csv file as an input which must contain rows of data corresponding to categorical, quantitative, and audio attributes. The landing page will consist of a sample csv file. In order to compute the spectrograms, audio should be encoded in the csv as time series arrays of floating point numbers, which correspond to the first 15 seconds of each audio file in the dataset. The user can upload their dataset as a csv file and will be shown a modal window to select the categorical, quantitative and sound attributes retrieved from the file. After choosing the grouping, the page will load an overview of the first quantitative and categorical attribute in the x and y axis respectively. The x and y axis will have a dropdown beside them to give the user the ability to change the attribute when needed. The graph will be a beeswarm chart of individual sound files encoded with a single color. The y-axis dropdown will show the options of the selected categorical attribute and all of the options will be checked by default. The user can choose to deselect some options to show only the selection categories in the y-axis. The x-axis will have scalar values of quantitative attribute shown by a range from the minimum to maximum value found for that attribute in the csv. The user can also tune the minimum and maximum value of the x-axis to narrow down the sound files further. For example, the user may choose to only observe the sound files having a meow duration of 0.75 to 1.25 seconds.

There will be a filter dropdown on the left of the beeswarm chart for the user to choose another categorical attribute they wish to filter the view with. This attribute will also have the

TABLE III
ESTIMATED WORK TIMELINE

| Milestone | Deadline | Est. Hours | Owner |
|-------------------------------|----------|------------|-----------|
| Proposal | Oct. 21 | 13 | All |
| Learn D3 | Oct. 22 | 10 | All |
| Create Dashboard Scaffolding | Oct. 24 | 2 | Mifta |
| Preprocess Dataset | Oct. 29 | 30 | Elizabeth |
| Add Attribute Selection Modal | Oct. 31 | 10 | Mifta |
| Implement Overview | Nov. 5 | 20 | Mifta |
| Add Filters | Nov. 6 | 10 | Mifta |
| Implement Detailed View | Nov. 22 | 20 | Elizabeth |
| Implement Multi View | Nov. 29 | 10 | Elizabeth |
| Implement Summary View | Nov. 20 | 10 | Nichole |
| Implement Comparison Toolbar | Nov. 22 | 10 | Nichole |
| Update | Nov. 16 | 10 | All |
| Peer Reviews | Nov. 17 | 10 | All |
| Post-update Meeting | Nov. 24 | 2 | All |
| Address Feedbacks | Dec. 5 | 20 | TBD |
| Final Presentation | Dec. 15 | 20 | All |
| Final Paper | Dec. 17 | 25 | All |

options selected by default and each of the options will be encoded by individual colors in the beeswarm chart. The user may choose to analyze individual sound files by clicking on the dots in the beeswarm chart to analyze the spectrogram. This information can help in understanding the frequencies of a signal as it varies with time. Additionally, the user can add multiple overviews of each quantitative attribute for comparison by clicking the "Add View" button and choosing another quantitative attribute. By selecting one or more points in the original view, the same points in the new quantitative view will be highlighted as well. For example, the researcher may open both the mean frequency and duration view, select a group of meows with high mean frequency, and see that the duration of these high-frequency meows range from 0.5 to 0.75 seconds.

The user can also see a grouped bar chart of the attributes and filters chosen in the beeswarm chart. This will show the average of the quantitative attribute in the corresponding beeswarm chart grouped by the two categorical attributes. The summary chart will show information based on the filters applied to the overview table.

VIII. MILESTONES

We are aiming to spend around 232 hours on this project. Table III shows an estimated timeline for the milestones of the project. Status of each milestone is indicated with text colors where green indicates *Completed* and orange indicates *In Progress*.

REFERENCES

- [1] Audacity. URL: <https://www.audacityteam.org>.
- [2] Tony Bergstrom, Karrie Karahalios, and John C. Hart. "Isochords: visualizing structure in music". In: *Proc. Graphics Interface (GI)*. May 2007, pp. 297–304. DOI: 10.1145/1268517.1268565.
- [3] Matthew Cooper et al. "Visualization in Audio-Based Music Information Retrieval". In: *Computer Music Journal* 30.2 (2006), pp. 42–62. URL: <https://www.jstor.org/stable/3682003>.

- [4] Michaël Defferrard et al. “FMA: A Dataset For Music Analysis”. In: *arXivPreprint* (2017), pp. 1–8. DOI: arxiv:1612.01840.
- [5] Chadawan Ittichaichareon, Sloveby Suksri, and Thaweesak Yingthawornsuk. “Speech Recognition using MFCC”. In: *Proc. International Conf. Computer Graphics (ICGSM)*. July 2012, pp. 135–138. DOI: 10.13140/RG.2.1.2598.3208.
- [6] Bongshin Lee et al. “FacetLens: Exposing Trends and Relationships to Support Sensemaking within Faceted Datasets”. In: *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*. 2009, pp. 1293–1302.
- [7] Luca Andrea Ludovico et al. *CatMeows: A Publicly-Available Dataset of Cat Vocalizations*. Aug. 2020. DOI: 10.5281/zenodo.4008297.
- [8] Brian McFee et al. *librosa/librosa: 0.8.1rc2*. May 2021. DOI: 10.5281/zenodo.4792298.
- [9] Chris Muelder, Thomas Provan, and Kwan-Liu Ma. “Content Based Graph Visualization of Audio Data for Music Library Navigation”. In: *Proc. IEEE International Symp. Multimedia*. Dec. 2010, pp. 129–136. DOI: 10.1109/ISM.2010.27.
- [10] Tamara Munzner. *Visualization Analysis and Design*. New York: A K Peters/CRC Press, Oct. 2014. DOI: 10.1201/b17511.
- [11] Elias Pampalk, Andreas Rauber, and Dieter Merkl. “Content-based organization and visualization of music archives”. In: *Proc. 10th ACM International Conf. Multimedia (MULTIMEDIA)*. Dec. 2002, pp. 570–579. DOI: 10.1145/641007.641121.
- [12] Susanne Schötz and Joost van de Weijer. “A Study of Human Perception of Intonation in Domestic Cat Meows”. In: *Proc. 7th International Conf. Speech Prosody (SpeechProsody)*. May 2014, pp. 874–878. DOI: 10.21437/SpeechProsody.2014-163.
- [13] Hasti Seifi, Kailun Zhang, and Karon E. MacLean. “VibViz: Organizing, visualizing and navigating vibration libraries”. In: *Proc. IEEE World Haptics Conf. (WHC)*. 2015, pp. 254–259.
- [14] *Sonic Lineup*. URL: <https://www.sonicvisualiser.org/sonic-lineup/>.
- [15] Bokyoung Sung et al. “Effective Digital Music Retrieval System through Content-based Features”. In: *World Academy of Science, Engineering, and Technology* 50 (2009), pp. 721–726.
- [16] Seong C. Yeon et al. “Differences between vocalization evoked by social stimuli in feral cats and house cats”. In: *Behavioural Processes* 87.2 (June 2011), pp. 183–189. DOI: 10.1016/j.beproc.2011.03.003.