

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

Elizabeth Reid, Mifta Sintaha and Nichole Boufford

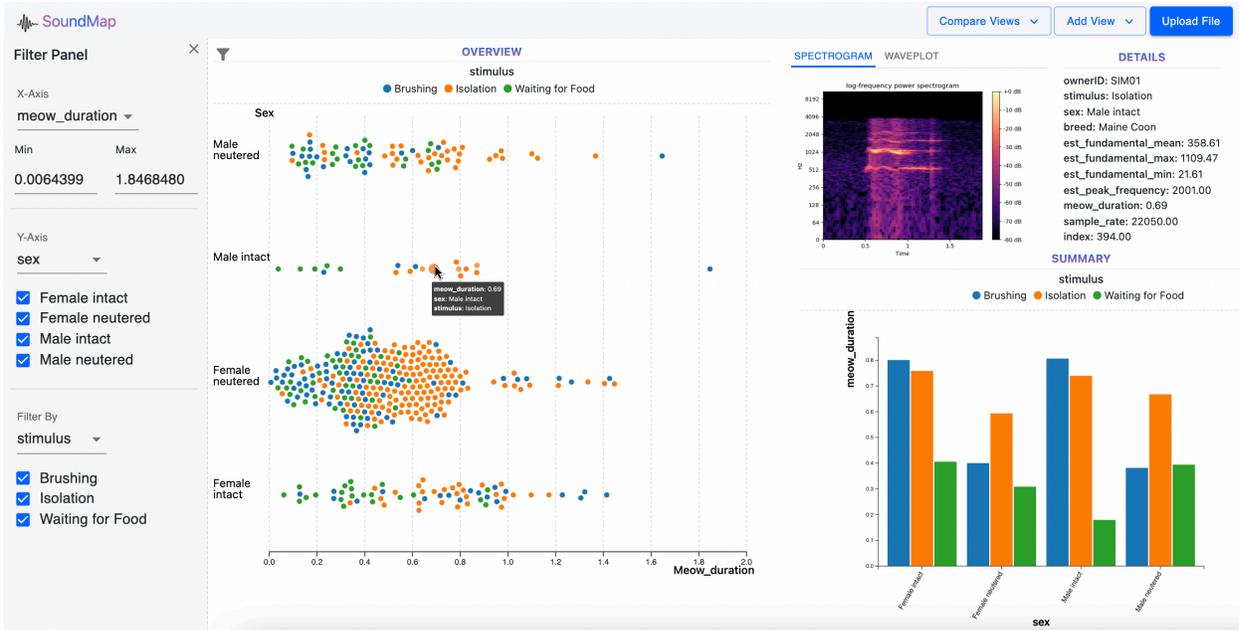


Fig. 1. The main view of *SoundMap* showing the Overview, Detailed View of a selected sound file, and the Summary view using an audio dataset of cat meows. The filter panel on the left is used to change the x-axis, y-axis and grouping attribute of the Overview, which consequently re-renders the summary view with the updated information. The toolbar at the top allows the users to add multiple views, compare, and reset views.

**Abstract**—Current sound visualization methods, including spectrograms, chromagrams, and waveplots, provide detailed information about individual sound files but do not scale for visualizing multi-attribute sound datasets. We introduce *SoundMap*, an interactive visualization tool for analyzing sound datasets. *SoundMap* allows analysts with any level of experience to view many files with audio and non-audio attributes at once. Users can filter on specific features to enhance browsing and exploration. The tool provides an overview of all files, aggregate summary views, and traditional spectrograms to enable users to compare attributes across a collection of sound files.

**Index Terms**—Audio visualization, multi-attribute, multi-faceted

## 1 INTRODUCTION

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

- Elizabeth Reid is with Department of Computer Science, The University of British Columbia. E-mail: reid42@cs.ubc.ca.
- Mifta Sintaha is with Department of Electrical and Computer Engineering Science, The University of British Columbia. E-mail: msintaha@ece.ubc.ca.
- Nichole Boufford is with Department of Computer Science, The University of British Columbia. E-mail: ncbouf@cs.ubc.ca.

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 [9, 17], *SoundMap* allows 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 [2] 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 or waveplot. To illustrate how *SoundMap* can be used for different types of sound datasets, we evaluate the tool on two different example datasets: CatMeows [10], and the Free Music Archive (FMA) [6].

## 2 RELATED WORK

### 2.1 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 [3], 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 [5]. 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* [18] 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 [5, 12], or displayed in a spatial field [5]—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 [12], and the *Islands of Music* system by Pampalk et al. [14] 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.

### 2.2 Faceted Browsing

*FacetLens* [9] is an interactive visualization tool for exploring relationships and trends in order 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 added 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 or waveplot. It also does not have any summary view to show the aggregated value of the quantitative attributes grouped by the categories.

### 2.3 Waveform Visualization

Similar to sounds, haptic data can also be visualized as waveforms. Seifi, Zhang, and MacLean [17] 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.

## 3 DATA AND TASK ABSTRACTION

### 3.1 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, as there may be multiple fundamental frequencies that are difficult to distinguish from one another. Files that are longer than a few seconds are also not suited to fundamental frequency attributes, since 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 [8] or music information retrieval (MIR) [19] 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 features 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.

### 3.2 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.

#### 3.2.1 MeowAnalysis

The first dataset of interest was derived from the CatMeows dataset compiled by Ludovico et al. [10]. 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 [16, 20], we derived five relevant audio attributes from the original dataset. The Python audio processing library *Librosa* [11] was used to estimate mean, maximum, and minimum fundamental frequency for 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 1.

Table 1. 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
			<b>Total: 440 meows</b>

### 3.2.2 SongAnalysis

Our second dataset of interest was derived from the Free Music Archive (FMA) dataset created by Defferrard et al. [6], 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 selected a subset of the original dataset based on music genre, choosing a single artist from each of the seven genre types. Artists were chosen who had at least five songs of the given genre listed in the FMA dataset. This manual selection step resulted in a more manageable dataset of 98 total songs, which we call the SongAnalysis dataset. Additionally, we removed several of the original audio features from the SongAnalysis dataset, 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 2.

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 applications such as Spotify<sup>1</sup>.

### 3.3 Tasks

At a high-level, *SoundMap* is intended to support the exploration of multi-attribute sound data. Specifically, users can explore how metadata attributes and audio attributes are related, as well as view individual items in detail. To that end, *SoundMap* can be used to answer questions such as:

- How do audio attributes like fundamental frequency or sound duration vary across different metadata categories?
- What is the behaviour of frequency and/or amplitude in a particular sound file?
- What is the average duration or fundamental frequency of a given sound file dataset?

We further organize specific user tasks into the abstract task categories *analyze*, *search*, and *query*, as described by Munzner [13].

#### 3.3.1 Analyze

These tasks are centered around data consumption and discovery.

- *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.

<sup>1</sup><https://www.spotify.com>

Table 2. Classification of attributes in the SongAnalysis dataset.

Attribute	Description	Type	Items/Range
Album	Album the song belongs to. For single tracks the album is described as N/A.	Categorical	10
Artist	Song artist.	Categorical	7
Genre	Genre of the song. If a song has not been assigned a genre, the genre is described in this dataset as unknown.	Categorical	7
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	461 – 2020
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	13.0 – 27.0
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	449 – 3370
Mean Zero Crossing Rate	The rate at which a signal as a waveform crosses the line $y = 0$ .	Quantitative	0.0172 – 0.0859
			<b>Total: 98 songs</b>

### 3.3.2 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.

### 3.3.3 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.
- *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.

## 4 SOLUTION

We propose *SoundMap*—a vis tool that allows users to analyze multi-attribute audio data effectively with ease. The main features of *SoundMap* are described below:

### 4.1 Attribute Selection

As a dynamic tool, we incorporate a way to allow users to upload their own dataset as a csv file in *SoundMap*. After uploading the csv file, we display a modal that allows users to select the categorical, quantitative and sound attributes that they want to visualize. This helps users specify only the particular attributes that they might be interested in visualizing, and also reduces clutter. Using a modal is also helpful to accurately identify attribute types, instead of determining it dynamically. Fig. 2 shows the attribute selection modal of the MeowAnalysis dataset.

### 4.2 Overview

To encode the relationship between categorical and quantitative attributes, we use a beeswarm chart where points are used as marks to encode the individual sound files. Beeswarm charts are useful for displaying many data points, which are binned to each category level and spread out to avoid occlusion. We use the `d3.forceSimulation` to jitter the dots by simulating the collision of the points for half the

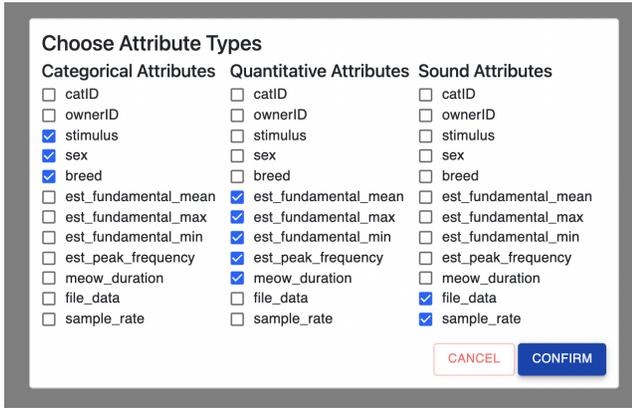


Fig. 2. Attribute Selection Modal shown after MeowAnalysis.csv file is uploaded.

dataset length. The horizontal axis represents the scalar attribute of interest. The vertical axis is used for binning each sound file based on the categorical attributes. On the top left of the overview chart, there is a filter icon to toggle the filter panel into the view. The filter panel, as shown to the left of the Overview in Fig. 1, is used to configure the axes in the overview and summary chart through dropdown menus. The beeswarm chart values in the y-axis can be filtered by selecting checkboxes. By default, all the values will be selected. The x-axis range can be configured through the input fields shown below the dropdown menu by adjusting the minimum and maximum values. The data can be further grouped by another category using the “Filter By” dropdown, which color encodes the data points based on the categorical attribute chosen. The legends pertaining to this “Filter By” attribute are then shown at the top of the chart. Deselecting the category values in the grouping attribute removes the data from the view. We use a category color scheme in D3 called `schemeCategory10` [4] which has a range of 10 different colors for encoding this filter attribute. The reason we chose this color scheme is because of its high distinguishability factor in each bin. To address scalability issues, we repurpose the colors for categories consisting of more than 10 levels. Each change in the dropdowns and checkboxes reloads the graph with an animation effect to indicate change. The page layout is also responsive to the data in order to accommodate various attributes and category levels within the viewport. For instance, if there are fewer datapoints, the radius of the dots increase and the height of the chart is adjusted to fit within the viewport, while also avoiding overlap of the dots with the axis.

### 4.3 Detailed View

A detailed view of the sound file is rendered to the top right of the beeswarm chart when a single point is clicked in the overview chart as shown in Fig. 1. To indicate that a point is selected in the overview chart for detailed analysis, the radius of the selected sound file is doubled for differentiation. The detailed view shows a spectrogram and a waveplot of the sound file as well as other detailed data as text, such as the categorical attributes and quantitative values for the sound file. The spectrogram helps to understand the breakdown of individual frequencies within a sound signal over time, whereas the waveplot helps to understand how the amplitude of a sound file varies over time. Using tabs, a user can switch between the spectrogram and waveplot to analyze the frequency breakdown or the amplitude trend over time for that sound file. Fig. 3 shows the detailed view of a song file from the SongAnalysis data. The left pane shows the audio plots and the right pane shows all the metadata of that song file. The right pane is scrollable to allow visibility of all metadata attributes within the panel.

### 4.4 Summary View

A direct summary of all the filters applied to the beeswarm overview chart is visualized in a grouped bar chart. The length of the bar is encoded with the average quantitative value of all sound files belonging

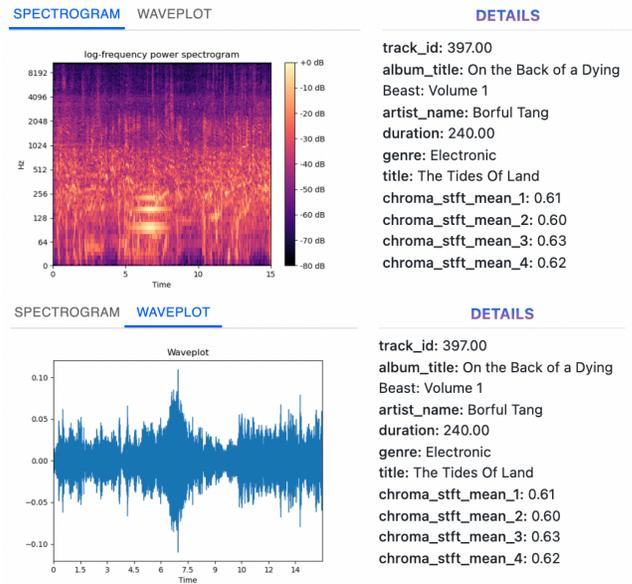


Fig. 3. Detailed view of the song “The Tides Of Land” which belongs to the Electronic genre. The figure demonstrates the tab switching between spectrogram and waveplot.

to a specific category indicated by the x-axis. The axes positions in the summary chart are swapped compared to their original positions in the overview chart, i.e., the x-axis shows the categorical attribute and the y-axis shows the quantitative attribute. Hovering over the bars displays a tooltip showing the exact value of the attributes in the axes. The bars are grouped using the same grouping attribute used in the overview chart and color encoded in the same way as shown in Fig. 1.

### 4.5 Multiple Views

A single “view” consists of an overview, summary view, filter panel, and details panel. We added support for multiple views by allowing users to add new views with a different quantitative attribute. In case a user wishes to keep the current view intact and just analyze the data with different sets of attributes, they can do so by adding a new view from the top toolbar. This reduces cognitive load on the user when they want to compare trends or relationships with different set of attributes on the same dataset without losing information about the previous data they were analyzing. By clicking on the “Add View” dropdown button, the user is shown options to choose a quantitative attribute—once selected, it duplicates the current view, scrolls to the bottom, and displays the new view just below the first view with the chosen quantitative attribute. The user can configure the axes in this new view using the filter panel of that view. Multiple views are useful for comparing all the data and observing the relationships between multiple audio and non-audio attributes at a glance. Fig. 4 shows two views stacked together where the bottom view is the newly added view.

As an additional step, the user can click the “Compare Views” dropdown button, which appears only when there is more than one view, and choose the chart type to compare. As there are three types of panels, namely Overview, Summary and Detailed View, the user can select one of these panels to compare at a time. Fig. 4 shows the two dropdowns in the toolbar along with their options. The comparison mode displays the charts in a juxtaposed manner by taking the snapshot of all the views. The snapshots prevent the user from mutating the axes or groupings by hiding the filter panel. However, when comparing between Overviews, users can select a point to see the location of the same sound file in each view as shown in Fig. 9. In the comparison mode, users can hover over the dots or bars in Overview or Summary view via tooltips, which show information about the values in each axis. They can also remove certain views by clicking on the cross icon. To go back to the original view, the user can click on the “Reset” button in the toolbar and reset

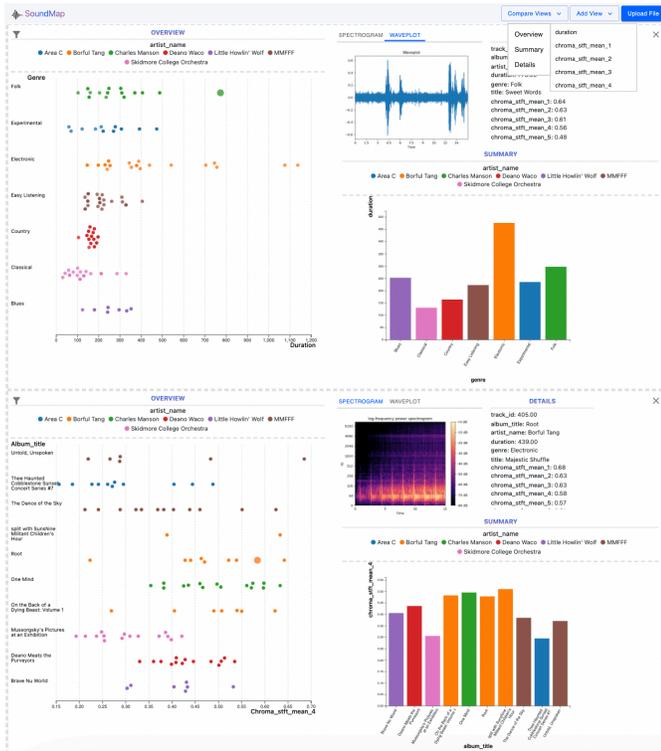


Fig. 4. Multiple views of SongAnalysis dataset. Selecting a quantitative attribute from the “Add View” dropdown creates a new view just below the current one. The toolbar shows the options of the two dropdowns—Compare Views and Add View.

the comparison mode.

## 5 IMPLEMENTATION

We used Python scripts for data wrangling and the Librosa [11] library for deriving audio features such as file data and sample rate from raw audio files. We developed the visualization system as a web application using the React<sup>2</sup> framework and D3.js<sup>3</sup> visualization library. We also used Material-UI<sup>4</sup> for adding the UI elements like dropdowns, input boxes, checkboxes, and buttons. The application consists of a server backend integrated with the Python framework, Flask<sup>5</sup>. We used REST API to communicate with the server which generates the corresponding spectrogram and waveplot whenever the user clicks on a dot in the beeswarm chart. We also used the Python library Librosa [11] and Matplotlib<sup>6</sup> to generate these charts and send the resulting plot as a base64 encoded image to the client. Building *SoundMap* as a web application gives users the ability to upload their own datasets, thereby making the tool more dynamic. Our implementation is publicly available in github<sup>7</sup> for open source usage.

## 6 RESULTS

### 6.1 Usage Scenario: MeowAnalysis

Consider a research veterinarian with an audio dataset of cats having certain attributes (breed, gender, etc.) in various stimuli, who wants to visualize the relationship between each of the attributes to identify patterns. To view the effect of various attributes on cat vocalizations [10], the user of the tool is taken to the landing page of *SoundMap*. The

<sup>2</sup><https://reactjs.org>

<sup>3</sup><https://d3js.org>

<sup>4</sup><https://mui.com>

<sup>5</sup><https://flask.palletsprojects.com>

<sup>6</sup><https://matplotlib.org>

<sup>7</sup><https://github.com/msintaha/SoundMap>

tool takes in a csv file as an input which must contain rows of data corresponding to categorical, quantitative, and audio attributes. In order to compute the spectrograms and waveplots, audio should be encoded in the csv as time series arrays of floating point numbers, which correspond to at most the first 15 seconds of each audio file in the dataset. Keeping this in mind, our user uploads their dataset as a csv file and is shown a modal window to select the categorical, quantitative and sound attributes retrieved from the file as shown in Fig. 2. After choosing the attribute groupings, the page will load an overview of the first quantitative and categorical attribute in the x and y axis respectively. The user can also reconfigure the axes using the filter panel, which appears by clicking on the *filter icon* on the top left of the overview chart.

As our user is interested in exploring the relationships between meta-data attributes and cat meow features, they first consider the Overview in *SoundMap* to get a sense for some initial relationships in the data. The Overview displays a beeswarm chart of individual sound files color encoded with the grouping attribute. The y-axis dropdown shows the options of the selected categorical attribute, and all of the options will be checked by default. An initial secondary filter is also automatically selected, again with all options selected by default. In this case, the y-axis is ordered by cat sex and additionally filtered by stimulus, with respect to the audio feature “estimated mean fundamental frequency” (see Fig. 6). The user notes that there are some outliers present in the “female neutered” category, and that the “male intact” category has very few cats. Wanting to eliminate this data from consideration, the user decides to deselect some options to show only the “female neutered” and “male neutered” categories in the y-axis. To eliminate the outliers, the user also decides to tune the minimum and maximum value of the x-axis in the filter panel. For example, the user may choose to only observe the sound files having a mean fundamental frequency of 100 to 400 Hz. Fig. 7 shows the result of applying both filters to the Overview.

After exploring the relationship between sex and mean frequency, the user decides to use the filter panel to select new y-axis, x-axis, and filter attributes. In this case, they select “stimulus” for the y-axis, “meow duration” for the x-axis, and “breed” for the filter. After observing in the Overview that the isolation stimulus seems to elicit longer meows, they consult the Summary view to confirm their assumptions. The Summary view allows the user to see a grouped bar chart of the attributes and filters chosen in the beeswarm chart, and displays the average of the quantitative attribute along the beeswarm’s x-axis grouped by the three categorical attributes on the beeswarm’s y-axis. The summary chart also shows information based on the filters applied to the Overview via color encoding. In this case, the user confirms that the isolation stimulus has a higher average meow duration, and additionally, they note that Maine Coon cats generally have longer meows than the European Shorthair breed (see Fig. 8).

The user may also choose to analyze individual sound files by clicking on the dots in the beeswarm chart to analyze the spectrogram or waveplot. This information can help in understanding the frequencies and amplitudes of the cat meows as it varies with time.

Finally, the user can add multiple overviews of each quantitative attribute for comparison by clicking the “Add View” button and choosing another quantitative attribute. If the user is interested in comparing the beeswarm plots in particular, they may then select “Compare Views” and choose “Overview” from the drop-down menu. Once both Overviews are open, the user can track the location of an individual point in both views by clicking on it. For example, they select a point with high mean frequency, and see that it has a relatively short duration of approximately 0.4 seconds (see Fig. 9).

### 6.2 Usage Scenario: SongAnalysis

In another usage scenario, imagine a musician that draws inspiration from a variety of albums and artists. The user has the sound files from some of their favourite artists and has exported them to a csv with categorical, quantitative and sound attributes. On their own, the musician can create and analyze each song spectrogram and waveplot individually, but they are interested in learning if there are any general trends or similarities in their music dataset. Again, the user will upload their csv file to *SoundMap* and will select their attributes of interest,

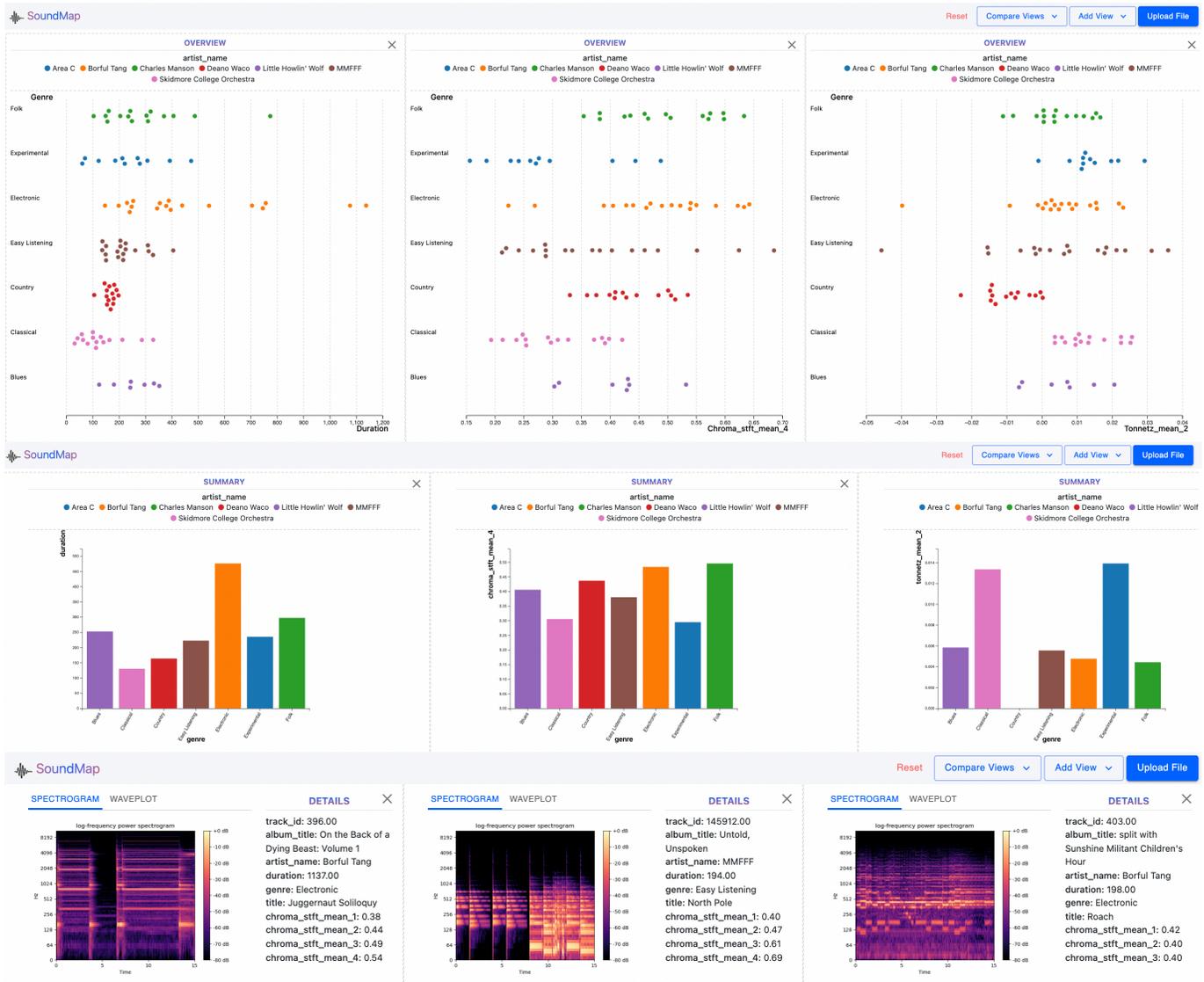


Fig. 5. Comparison mode of Overview, Summary and Detailed View juxtaposed with three different quantitative attributes. The detailed view shows three randomly selected points from each of the overview charts. The toolbar shows the *Reset* button which appears only when the comparison view is active.

such as artist name, duration, tonnetz mean, and spectral rolloff mean. The user sees a beeswarm chart with their first selected quantitative values, tonnetz mean, on the x-axis and the first selected categorical attribute, artist name, on the y-axis. The grouped bar chart shows the averages of the same quantitative values grouped by the categorical attributes. This time, the user wants to take advantage of the multi-view feature to retain the current view for later use. They select their first attributes of interest and notice some trends in the beeswarm chart and grouped bar chart. Keeping these charts in view, they select the “Add View” button and select “spectral rolloff mean” for the second quantitative attribute. The user changes the categorical attribute in this new view to genre and analyzes the relationship between genre and artist. The user confirms that each artist in the data produces songs belonging to only one type of genre as indicated by the single groupings in the bar chart in Fig. 4. The user switches back to artist, then compares side-by-side the views for each quantitative attribute. To get a closer look at the different clusters in both beeswarm charts, the user clicks “Compare Views” and selects Overview. The user notices that an artist they know and enjoy, “Borful Tang”, has clusters very similar to an artist they are less familiar with, “Area C” and makes a note to listen to their album (see Fig. 10).

## 7 DISCUSSION

### 7.1 Strengths

We believe that *SoundMap* effectively allows both novice and expert users to visualize and understand relationships between a large number of sound files with many attributes. The overall dashboard is simple, and interactions with the interface are intuitive. For novices in particular, the use of the beeswarm plot and bar chart provide a familiar way to explore sound data, while the detailed view spectrogram and waveplot allow experts the granularity that they are accustomed to from traditional audio analysis.

Another major strength of *SoundMap* is that it is generalizable to other signal processing tasks, and could be used to visualize sensor data or vibrations. It is also particularly useful for machine learning tasks, as it allows users to easily explore how different features are related to one another or identify outliers. In general, the tool could save users time as they will not have to create numerous individual visualizations manually—simply loading in the dataset will allow them to effectively explore relationships in their data.

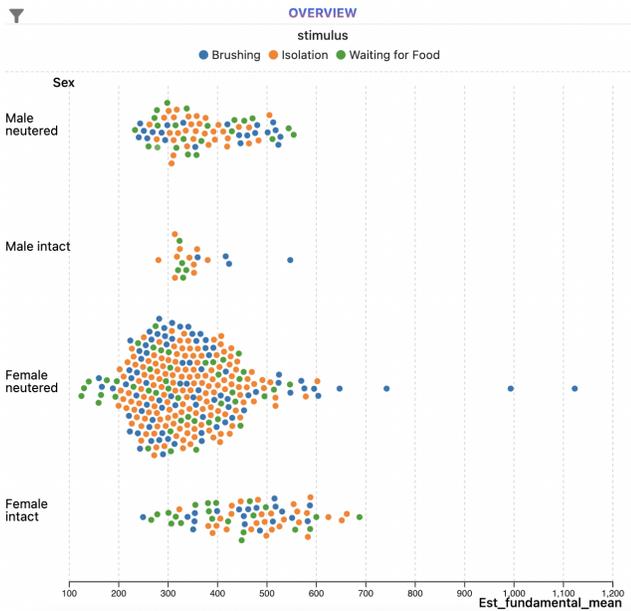


Fig. 6. Overview upon loading the MeowAnalysis dataset to *SoundMap*. Outliers are clearly visible in the “female neutered” category.

## 7.2 Limitations

We encountered limitations due to time constraints, technical issues, and the nature of the data itself. For example, we did not have enough time to implement interactive elements in the detailed view, or add the ability to play sound files directly within the app. *SoundMap* also requires users to upload preprocessed datasets, which assumes that users either already have access to datasets with extracted sound features or are able to extract these features themselves. While an expert user might be able to create an appropriate dataset easily, novice users would be better supported if they were able to upload individual sound files of interest directly to our tool. However, adding support for sound file uploads would have added an additional layer of complexity to the project that we did not have time to address, as we would need to consider factors such as long upload times, integration of metadata features, and extracting only relevant audio features for any given set of sound files.

We also encountered technical issues with JavaScript when trying to upload files larger than approximately 450 MB, as any file over this size limit could not be uploaded to *SoundMap*. This size limit forced us to limit the total number of songs in our SongAnalysis dataset, and we also reduced the file data attribute to the first 15 seconds of each song in order to appropriately reduce the dataset’s size for uploading. Finally, we also had to consider the cardinality of categorical attributes when creating the SongAnalysis dataset, since categorical attributes with high cardinality resulted in very thin bars on our bar chart.

## 7.3 Challenges and Lessons Learned

We encountered several challenges while working on the project, many of which arose due to a lack of previous experience working with D3. For example, we encountered a particularly confusing bug that resulted from an interaction between the filter panel and the D3 svg placement which would occasionally make individual points in the beeswarm plot difficult to click. For those of us who were less experienced with React, it also took much longer than expected to get familiar with standard React idioms, and debugging in general was quite challenging.

We also faced significant challenges setting up the detailed view. In particular, we had initially tried to process the file data and generate a spectrogram entirely from the client side of our application, but this turned out to be a pitfall. While we did find a useful library to generate the relevant audio data [15], the results needed significant

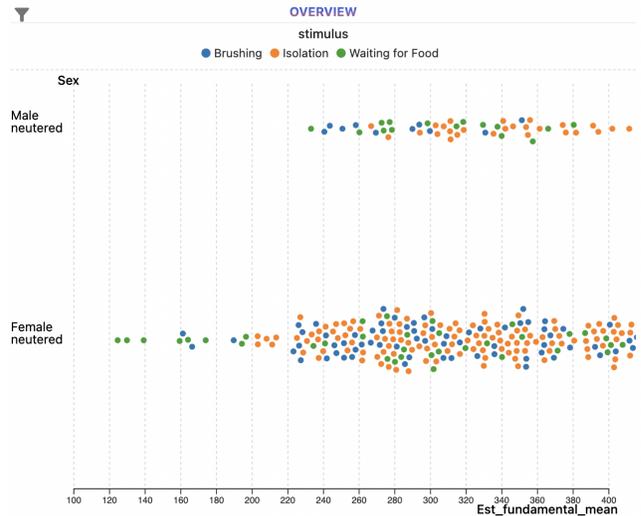


Fig. 7. Overview after filtering by sex and fundamental frequency range.

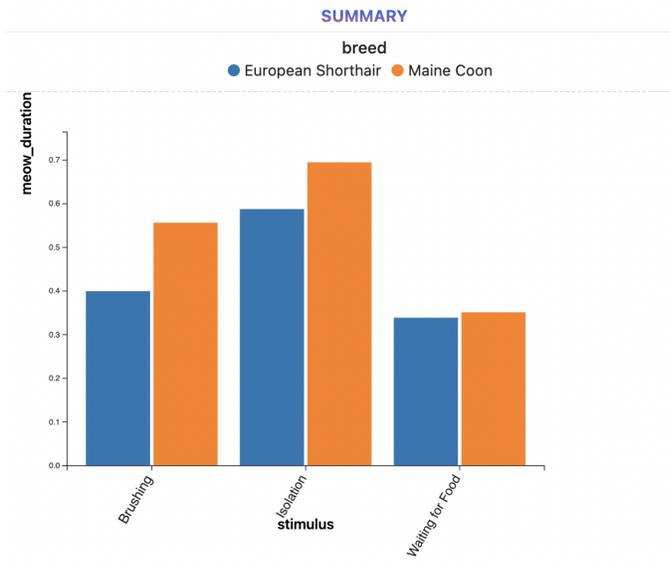


Fig. 8. Summary view for average meow duration with respect to stimulus, filtered by breed.

further processing in order to aggregate data into bins, since the raw data had too many points to render the heatmap properly. The main reason we persisted with this initial setup was because D3 could provide useful interactive elements like zooming or panning. However, we eventually realized that it would take too much time to get the frontend spectrogram generation working properly, and shifted our focus to generate spectrograms and waveplots from the backend instead.

Making the view responsive to the dataset was another challenge. We had to make sure that the summary charts and overview charts are visible within the viewport as much as possible and the axes of each of the charts are aligned properly when the detailed view is visible. The MeowAnalysis and SongAnalysis data are quite different, especially in the aspect of sparseness, with the former being more dense compared to the latter. This difference in sparseness required some fine-tuning in the chart width and height, so that the data points in the charts are not too densely or loosely packed, while also maintaining visibility within the viewport.

Another pitfall was that we originally chose a subset of songs randomly for our SongAnalysis dataset, which resulted in categorical

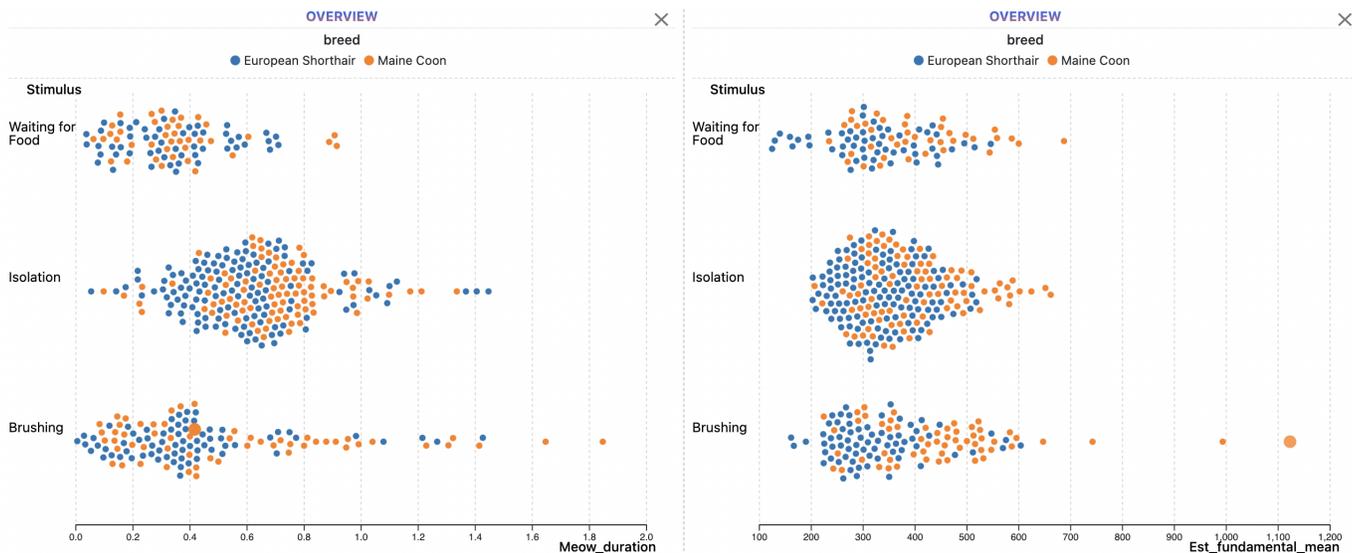


Fig. 9. Comparing two Overviews directly for the MeowAnalysis dataset. The same selected point is emphasized by an increased radius in both charts.

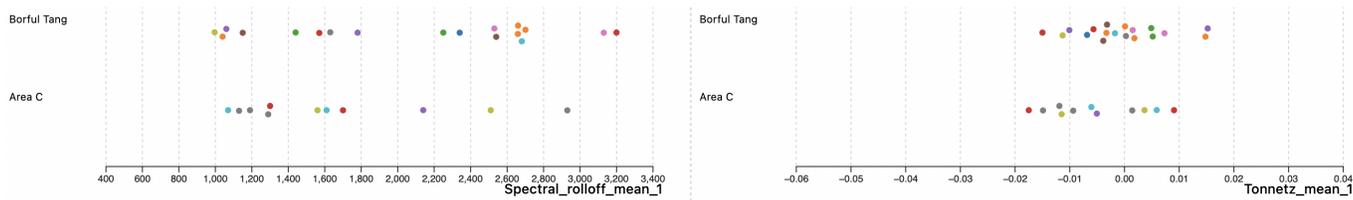


Fig. 10. Compare view of artists “Borful Tang” and “Area C”. The figure shows similar dot clusters for both spectral rolloff mean and tonnetz mean.

features with a high cardinality and few individual songs per category level. This initial version of the dataset was not particularly interesting to visualize, and we decided to collect a new set of songs more strategically for the final version of the dataset. Since downloading and processing the songs for this dataset took a few hours each time, we inadvertently created additional work for ourselves that could have been avoided if we had taken more time at the beginning of the project to consider what we wanted our SongAnalysis dataset to look like, and the potential consequences of curating the dataset randomly.

Finally, since some members of our team worked on Mac and some on Windows, we encountered several problems when setting up Node.js<sup>8</sup> for the first time on the Windows machine specifically. We had to spend a considerable amount of time dealing with configuration issues, which caused a delay in some of our initial milestone deadlines. Additionally, starting up the app locally on the Windows machine sometimes took upwards of ten minutes, a problem which we were never able to solve, but likely had to do with the aforementioned configuration issues.

Therefore, our main takeaways from the project are as follows:

- Learning new libraries and frameworks will take longer than expected.
- If something is not working, do not wait to ask others for help.
- Try to establish early on if a particular library or method is the right choice for the problem.
- When selecting a subset of data from a larger dataset, it is important to think critically about the features of the smaller dataset rather than trying to extract a subset randomly.

<sup>8</sup><https://nodejs.org>

- Allow more time for initial setup in the project planning stage, especially when people are working on different operating systems.

## 7.4 Future Work

Future work should include implementing the features we did not have time to complete at this stage of the project, such as the detailed view interactivity, a preview of the sound file with playback, additional forms of aggregation with bar charts besides average values and dynamic uploading, and data processing of sound files directly within the tool. We also plan to add the ability to export views containing charts of interest in a PDF format which can be useful for researchers or analysts for writing reports. Furthermore, it would be worthwhile to explore potential solutions for high cardinality categorical data. For example, grouping similar items together as a single bar in the bar chart and enabling pan and zoom interactions might help address readability concerns in the summary view [7]. We can also use horizontal bar charts instead of vertical ones to address space issues. Finally, it would be useful to conduct user studies with both domain experts and novice users to access whether or not *SoundMap* adequately fulfills user needs while remaining accessible to novices.

## 8 MILESTONES

We had initially aimed to spend around 232 hours on this project. Table 3 shows a detailed outline of the milestone descriptions as well as the estimated and actual time it took to complete each tasks. In total, we spent approximately 233 hours to complete the project.

## 9 CONCLUSION

This work presented a way to visualize multi-attribute sound data that is not only easy to use and analyze from an end-user perspective, but also provides flexibility to browse many attributes simultaneously. By targeting both expert and novice users, this tool can help researchers and

analysts identify and extrapolate complex, multi-faceted relationships and outliers efficiently. We believe this work can lay the foundation to support more intuitive and interactive multi-attribute audio visualisations that require limited user training and do not overwhelm the user with complicated charts.

## ACKNOWLEDGMENTS

We would like to thank Tamara Munzner for her feedback on earlier plans and iterations of our system as well as our peers for sharing their critiques on the project proposal.

## REFERENCES

- [1] Audacity. <https://www.audacityteam.org>.
- [2] Beeswarm. <https://flowingdata.com/charttype/beeswarm/>.
- [3] T. Bergstrom, K. Karahalios, and J. C. Hart. Isochords: visualizing structure in music. In *Proc. Graphics Interface (GI)*, pp. 297–304, May 2007. doi: 10.1145/1268517.1268565
- [4] Color schemes. <https://observablehq.com/@d3/color-schemes#Category10>.
- [5] M. Cooper, J. Foote, E. Pampalk, and G. Tzanetakis. Visualization in Audio-Based Music Information Retrieval. *Computer Music Journal*, 30(2):42–62, 2006.
- [6] M. Defferrard, K. Benzi, P. Vanderghenst, and X. Bresson. FMA: A Dataset For Music Analysis. *arXivPreprint*, pp. 1–8, 2017. doi: arxiv:1612.01840
- [7] Is your data too big to visualize? [https://www.sas.com/en\\_ca/insights/articles/big-data/is-your-data-too-big-to-visualize1.html](https://www.sas.com/en_ca/insights/articles/big-data/is-your-data-too-big-to-visualize1.html).
- [8] C. Ittichaichareon, S. Suksri, and T. Yingthawornsuk. Speech Recognition using MFCC. In *Proc. International Conf. Computer Graphics (ICGSM)*, pp. 135–138, July 2012. doi: 10.13140/RG.2.1.2598.3208
- [9] B. Lee, G. Smith, D. Tan, M. Czerwinski, and G. Robertson. Facetlens: Exposing trends and relationships to support sensemaking within faceted datasets. In *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pp. 1293–1302, 2009.
- [10] L. A. Ludovico, S. Ntalampiras, G. Presti, S. Cannas, M. Battini, and S. Mattiello. CatMeows: A Publicly-Available Dataset of Cat Vocalizations, Aug. 2020. doi: 10.5281/zenodo.4008297
- [11] B. McFee, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, D. Ellis, J. Mason, E. Battenberg, S. Seyfarth, R. Yamamoto, viktorandreevichmorozov, K. Choi, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, D. Hereñú, F. Stöter, P. Friesch, A. Weiss, M. Vollrath, T. Kim, and Thassilo. librosa/librosa: 0.8.1rc2, May 2021. doi: 10.5281/zenodo.4792298
- [12] C. Muelder, T. Provan, and K. Ma. Content Based Graph Visualization of Audio Data for Music Library Navigation. In *Proc. IEEE International Symp. Multimedia*, pp. 129–136, Dec. 2010. doi: 10.1109/ISM.2010.27
- [13] T. Munzner. *Visualization Analysis and Design*. A K Peters/CRC Press, New York, Oct. 2014. doi: 10.1201/b17511
- [14] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proc. 10th ACM International Conf. Multimedia (MULTIMEDIA)*, pp. 570–579, Dec. 2002. doi: 10.1145/641007.641121
- [15] H. Rawlinson, N. Segal, and J. Fiala. Meyda: an audio feature extraction library for the web audio api. In *Proc. 1st Web Audio Conf. (WAC)*, 2015.
- [16] S. Schötz and J. van de Weijer. A Study of Human Perception of Intonation in Domestic Cat Meows. In *Proc. 7th International Conf. Speech Prosody (SpeechProsody)*, pp. 874–878, May 2014. doi: 10.21437/SpeechProsody.2014-163
- [17] H. Seifi, K. Zhang, and K. E. MacLean. Vibviz: Organizing, visualizing and navigating vibration libraries. In *Proc. IEEE World Haptics Conf. (WHC)*, pp. 254–259, 2015.
- [18] Sonic Lineup. <https://www.sonicvisualiser.org/sonic-lineup/>.
- [19] B. Sung, K. Koo, J. Kim, M. Jung, J. Kwon, and I. Ko. Effective Digital Music Retrieval System through Content-based Features. *World Academy of Science, Engineering, and Technology*, 50:721–726, 2009.
- [20] S. C. Yeon, Y. K. Kim, S. J. Park, S. S. Lee, S. Y. Lee, E. H. Suh, K. A. Houpt, H. H. Chang, H. C. Lee, B. G. Yang, and H. J. Lee. Differences between vocalization evoked by social stimuli in feral cats and house cats. *Behavioural Processes*, 87(2):183–189, June 2011. doi: 10.1016/j.beproc.2011.03.003

Table 3. Project Milestones. "All" means all members contributed equally.

Milestone	Description	Deadline	Est. Hours	Actual Hours	Completed Date	Owner
Proposal	Brainstorm ideas and write the proposal.	Oct. 21	13	12	Oct. 21	All
Learn d3	Watch tutorials, browse examples and read documentation.	Oct. 22	10	12	Nov. 10	All
Create Dashboard Scaffolding	Use a React boilerplate and add Material-UI and d3 libraries. Also create basic components.	Oct. 24	2	1	Oct. 23	Mifta
Preprocess Dataset	Generate audio attributes for the Meow-Analysis dataset, collect and download song files for the SongAnalysis dataset, clean up missing or corrupted data from within the SongAnalysis features	Oct. 29	30	28	Nov. 15	Elizabeth
Add Attribute Selection Modal	Add modal and buttons from Material-UI, parse the uploaded csv using d3.	Oct. 31	10	8	Oct. 31	Mifta
Implement Overview	Find working examples and observable notebooks of beeswarm charts and implement in the tool. Plug in the real data and adjust the width and height. Add tooltips.	Nov. 5	20	25	Nov. 10	Mifta
Add Filters	Add a collapsible panel with checkboxes, dropdowns and input fields. Plug in the attributes from modal selection. Re-render chart on any change.	Nov. 6	10	12	Nov 15	Mifta
Implement Detailed View	Set up backend generation of spectrograms, and link to front end. Add circle radius adjustment upon point selection, and text details display.	Nov. 22	20	26	Oct. 21	Elizabeth
Implement Summary View	Create a basic barchart and then plugin the data from csv. Add tooltips, adjust height, wrap label text and rotate to avoid overlap	Nov. 20	10	17	Nov. 23	Nichole
Update	Read through the feedbacks and update the proposal write-up. Also create an updated mockup with the progress	Nov. 16	10	8	Nov. 16	All
Peer Reviews	Read project update of the assigned team. Write strengths and critiques. Discuss in class	Nov. 17	10	4	Nov. 17	All
Post-update Meeting	Demo the current progress. Discuss limitations such as high cardinality of categories and how to overcome this.	Nov. 24	2	1	Nov. 24	All
Implement Multi View	Maintain an array of quantitative, categorical and grouping attribute. Add a dropdown for selecting quantitative attributes called <i>Add View</i> . Selecting an option from this dropdown appends a new view to the array and renders the array of views.	Nov. 29	10	6	Dec. 1	Nichole
Implement Comparison View	Add a dropdown for selecting chart type called, <i>Compare Views</i> . Selecting the chart type should hide other charts and filter panel. It should also disable click events but retain mouseover events for tooltip display.	Nov. 22	10	6	Dec. 3	Mifta and Elizabeth
Address Feedbacks	Fix bugs and make the view responsive based on the data. Tune the height of charts, bars and circle radius to find the optimum sizes of each charts.	Dec. 5	20	25	Dec. 13	All
Final Presentation	Outline the main slide titles and work on respective assigned slides. Prepare demo video.	Dec. 15	20	15	Dec. 15	All
Final Report	Clean up code in the repository. Create teaser image and other screenshots for the report. Write the report.	Dec. 17	25	27	Dec. 17	All