Searching for Auditory Responses in the Visual Cortex

Kenneth Li, Rui Qin, Sadie Richardson, Aziza Salako, Elliot Smith

Data Science Projects | Final Report

November 14, 2018

Introduction and Project Background

Our brain constantly integrates auditory and visual information as it processes stimuli to form one coherent view of the world. There are projections directly linking auditory and visual cortex in rodents and primates, integrating these modalities at early stages of sensory processing. In one study, groups of rats were trained to simultaneously perform judgment tasks and temporal order judgement tasks, concluding that rats represent an effective model for studying audiovisual temporal synchrony at both the neuronal and perceptual level (Schormans et al., 2016). In fact, using multisite probes, single-units across multiple cortical layers can be sampled to identify the influence visual stimuli have on neural firing in the auditory cortex of awake mice (Morrill et al., 2018). These methods of sampling have concluded that the deepest cortical layers appear to be an important locus for crossmodal integration in the auditory cortex. Dr. Jacob Reimer, an Assistant Professor of Neuroscience at the Baylor College of Medicine has conducted extensive data collection of the mouse visual cortex to affirm this.

Dr. Reimer's team has collected large-scale recordings of single cell activity in the primary visual cortex of awake-behaving mice (each recording is more than 6000 cells recorded simultaneously), along with ambient noise recorded via an ultrasonic microphone and two-photon imaging methods for multi-neuronal recording with molecular techniques for circuit tracing and manipulation. As the mice were being recorded, they were also running on a treadmill; data of the treadmill's velocity as well as measurements of the mouse's pupil reactivity and sizes were also recorded. This dataset is unique because it captures the recordings of activity of 6,333 neurons in a mouse's primary visual cortex, which tell us how active individual neurons in the brain were at moments in time. Prior research has proven that lateral inhibition shapes frequency tuning in primary auditory cortex via an unconventional mechanism: non-preferred tones suppress both excitatory and inhibitory synaptic inputs onto layer 2/3 cells ("network suppression"). However, the data was calculated via *in vivo* whole-cell recording and two-photon Ca2+ imaging, and was taken only while data was collected while anesthetized mice were listening to multiple frequencies - at which no point was there visual stimuli (Kato et al., 2017).

It should be noted that mice are good subjects for this audiovisual research due to their impressive hearing range. While humans are able to hear between 64 and 23,000 Hertz (Reynolds et al., 2010), mice have a much broader range and can hear frequencies of about 1,000 to 100,000 Hertz - if the sounds are of sufficient quality and unobstructed. An increased

hearing range provides more audio-visual neural interactions that can be analyzed. Additionally, using mice is beneficial, as they allow for cross-species and cross-cortical area comparison in order to identify similarities and differences between the species-specific algorithms that are developed for the neocortex.

Dr. Reimer's goal is to use computational and theoretical methods (Tolias et al., 2018) of data analysis on the neural network to create models for cortical circuit function based on in vivo function of awake, behaving animals. Traditionally research proving an audio-visual interconnectedness used magnetic imaging in order to identify where signals were converging. However Dr. Reimer's research is unique as it is one of the only data set in the world that has data on a micro level, where in its recorded the activity of 5000 single cells in the primary visual cortex of the mouse. In the past, electrophysiological experiments that used imaging were technically and financially limited, and not entire replicable (Petro et al., 2017). As a Data Science Projects team, our overarching goal for the semester is to process the ambient noise recordings, detect acoustic features like mouse calls, human voices, and other noises, and identify neurons in the visual cortex that respond to these auditory cues.

Data Description

The data for our project consists of two simultaneous 90-minute recordings: one of mouse neural data and one of auditory data. These two recordings are stored in different files. Mouse neural data is stored in an h5 file, while the environmental audio data is stored as a flac file.

The data recordings are from a single experiment, in which there were two mice (a male and a female) each positioned on a treadmill, with their brains being recorded by a scanning electron microscope that recorded activity level of neurons in the visual cortex. Two different tones were played during the 90-minute experiment for short bursts of time, and the two tones never overlapped each other.

One of our data files contains information that tells us about the mouse's activities and surroundings, with recordings from three main sources: the mouse's visual cortex neurons, the mouse's pupil, and the treadmill the mouse was standing on. The data includes recordings of activity of 6,333 neurons in a mouse's primary visual cortex, which tell us how active individual neurons in the brain were at moments in time. The non-audio data file also has data for the location of each cell in a 3-dimensional section of the brain. Data pertaining to the mouse's pupil

includes its radius and the direction it is facing at points in time. The treadmill velocity is also recorded in this file at different points in time. Each of the recordings in this file were taken at a different sampling frequency. The pupil was recorded at 20 Hz, the treadmill velocity was recorded at 200 Hz and the neurons were recorded at 6.3 Hz.

The flac file contains auditory data from the mouse and its surroundings. This auditory data was recorded using a super-wide-band microphone at 200k Hz and should include noises from pieces of machinery as well as the tones played and any vocalizations the mouse makes.

The neural and auditory data were simultaneously recorded and can be temporally aligned with each other by using the auditory data to determine the time when the microscope scanner (that was recording neural data) turned on. This scanner makes a distinct noise that we will be able to find in the audio recording and use to align the two data sets.

Data Methodology

We used a variety of data science techniques to organize and clean the non-audio data and extract features from the audio data. The most important methods we used to prepare our data for analysis were interpolation and normalization, and we used spectral analysis to extract audio events.

Interpolation

Since all of the recordings from the mouse data were taken at distinct frequencies, it was imperative to use interpolation to align the recordings to a uniform time scale in order to conduct instantaneous comparison. Our methodology for normalizing the time scale so that all variables have data for the same time points is as follows. Since the neurons were recorded at the lowest frequency (6.3 Hz), we interpolated points from the pupil and treadmill recordings to match that time scale using linear interpolation. We defined our axis, the time scale of the neurons, and then, to interpolate our pupil and treadmill values at those time ticks, drew a straight line between the two surrounding points of our treadmill and pupil data and inferred what the pupil or treadmill value would have been at that specific time on the cell data time scale. We chose to interpolate to the frequency of the neuron recordings because it is the lowest time interval we can achieve without loss of data. Normalizing the time scale allows us to make feature by feature comparisons between key variables in the non-audio data file.

We also used the same method of interpolation once we combined the non-audio data and the audio data. The audio data was recorded at a much higher frequency than the cell trace, so we used the cell trace time scale to interpolate audio values to the times the cell traces were recorded.

Cell Trace Normalization

Since each neuron's cell trace is recorded on a different scale, it was important for us to normalize the cell trace data so we have values we can compare across neurons. We chose to normalize each neuron trace by the median value rather than the mean, since the median is robust to outliers. To normalize, we centered the values according to the median of the neuron's cell trace and then divided each data point by the median absolute standard deviation for that cell.

Spectral Analysis

Our preliminary spectral analysis of the audio file was done in three steps. First, we plotted the raw data trace as samples vs amplitude of the sample in order to visualize the audio data and get a rough idea of times when auditory events are happening. Next, we plotted a power spectrum, a plot of frequency vs power, for short samples of time. We conducted Fast Fourier Transformation (FFT) on the data in this step with the Python package "scipy". The spectrogram calculation is done using a multi-tapered method, i.e. by combining multiple overlapping slices of the raw data, each slice transformed into a spectrum with FFT. We can plot the spectrogram using common commands in Python packages, and we also plotted spectrums of some certain regions of interest to validate the scale of the audio data and the range of frequencies present in the recording, for example, the resonant scanning frequency of the machine used to record the neural data has a frequency of 12 kHz.

To extract features from the audio data, we implemented bandpass filters that eliminated all frequencies from the recording except for a small range around the known frequency of a tone played during the experiment. We picked an order for the filter and generated the filter coefficients for a bandpass filter, giving the Butterworth filter function the filter order, and the cutoff frequencies. From this filtered data, we were able to find the onset and offset time of each tone (Scipy Cookbook, 2012).

Data Splitting

We prepared our data for creating predictive models by splitting it into three parts: a validation

set (⅙), training set (⅔), and testing set (⅙).

Since we are working with time series data with precise audio events, we decided splitting the data completely randomly would not be wise because it would break up the audio event and its response period. Instead, we decided to chunk the data in a logical and ordered manner based on the audio events and *then* randomly split the chunks of data into training, validation and test data sets.

Our methodology for doing this is as follows. First, we identified times at which there was an audio event using the filtering method described above on our raw audio trace and the frequency of each tone. At these times, we used a 1 to denote that an audio event was happening, and outside of these times we denoted the lack of an audio event by a 0. Now, for both frequencies at which our audio events occur, we have a vector of 1's and 0's denoting whether or not the tones at each frequency were active during the experiment and at what times.

We joined these frequency vectors with our non-audio data and interpolated the times down to the cell trace times as previously described. At this point, we have a data frame that directly relates the time of an audio event with other pertinent data about the mouse, notably the neuron activity and pupil radius.

Now that we have a complete matrix with all the desired audio and neural data we may begin the actual data splitting process. We randomly selected our cuts based on non-event status, so that we would not break up any audio events. To do this, we first pulled out all the times at which there was not an audio event. Then, we found the 29 time points that would split the data into 30 equal bins. We added the audio events back into the data and split it into 30 bins according to the selected non-audio times. From these 30 bins, we randomly selected 20 to create our training set and 10 to create our test set. Finally, we have a training set (⅔), and test set (⅓) that we can later split into a validation set (⅙) and testing set (⅙).

Project Goals

The overall objectives for our project are to (1) complete spectral analysis of the audio recordings in order to identify auditory events at particular frequencies, (2) convert auditory events into analyzable features, such as binary arrays indicating presence and absence of an event, and (3) examine neuron activity in relation to these acoustic events, as well as pupil and

treadmill data, allowing us to identify visual cortex cells that are responsive to different types of stimuli, especially auditory stimuli.

For the first phase of our project, our objectives included completing preprocessing of the data in the non-audio file, including using linear interpolation to time-align the recordings of the neurons, pupil and treadmill that were taken at different frequencies, and completing spectral analysis to understand our audio data. Our goals for preliminary analysis of the audio data were to plot the spectrogram of the recording and identify auditory events by observing the spikes in the spectra and features in the spectrogram.

For the second phase of the project, our primary goal was to completely merge our audio and non-audio data sets and properly align them so the data would be completely ready for statistical analysis. Most of the tasks required to complete this goal involved feature detection from the audio file, which we broke into the following steps:

1. Identify the frequency of the two tones by examining the spectrogram.
2. Filter the audio trace at each of the two tone frequencies.
3. Use the filtered trace to identify the onset and offset times of the two tones.
4. Create binary variables to represent whether each tone is on or off over time.

Once we created two binary variables, we aimed to merge these on/off indicator variables with the non-audio data and time-align them so we could build statistical models to compare neural and audio data.

Another goal for the second phase of the project, that we completed in tandem with the feature detection and data organization steps, was to explore methods for finding relationships between neural data and other variables. We used the variables in the non-audio file to conduct this preliminary analysis. Using the pupil and treadmill data, we aimed to investigate methods for finding correlation and building predictive models that we would be able to apply to audio data later in the course of the project.

The ultimate goal for this project was to determine whether visual cortex neurons respond to auditory stimuli, and if so get an understanding of how. We used the merged data to build statistical models that compare neural activity to audio events. Through our analysis of the audio and neural data, we hoped to determine whether an audio event merely alerts the mouse and causes it to pay more attention or whether neurons in the visual cortex actually perceive some specific information about the auditory signals. We will address this overarching question with a

series of hypotheses that, when examined holistically, will provide insight into how much information the visual cortex processes about auditory events. The hypotheses are as follows:

1. Visual cortex neurons will show increased activity in response to the onset of a tone
2. Visual cortex neurons will show increased activity in response to the offset of a tone
3. Visual cortex neurons will show increased activity in response to the presence of any tone
4. Visual cortex neurons will show increased activity in response to specific tones (some neurons will be more tuned to tone A, while others will be more tuned to tone B)
5. Some visual cortex neurons might distinguish between tones A and B

In order to evaluate these hypotheses and achieve our ultimate goal of assessing how visual cortex neurons respond to auditory stimuli we will engineer audio features that correspond to each of the conditions in the hypotheses above. Then we will construct univariate and multivariate models to determine relationships between each of the different audio features and neural activity.

Results

Audio Analysis

Our analysis of the auditory data was primarily focused on conducting spectral analysis to identify important audio events. We first plotted the spectrogram, a visual representation of our ambient sound recording that displays a multi-tapered spectra over time, for the whole recording. We will validate this spectrogram by observing audio frequencies we know should exist and then use it to visualize the audio data before extracting features.

To validate the audio processing and spectrogram, we expect to find a 12 kHz peak throughout most of the spectrogram due to the audio emission from the microscope that is recording the neural data. Another validation measure will be whether the time length of this 12kHz peak matches the neural recording length in the non-audio data. The following plots follow the steps described in our methods for exploratory visualization of our audio data.

Figure 1, below, is a plot of the raw audio data with the x axis as the time in seconds and the amplitude of the signal on the y-axis. The Python packages “librosa” and “soundfile” were used to extract the recording data (samples and sample rate) and we plotted the raw sound using “matplotlib”, also in Python. This is a relatively rudimentary plot because we can only see areas

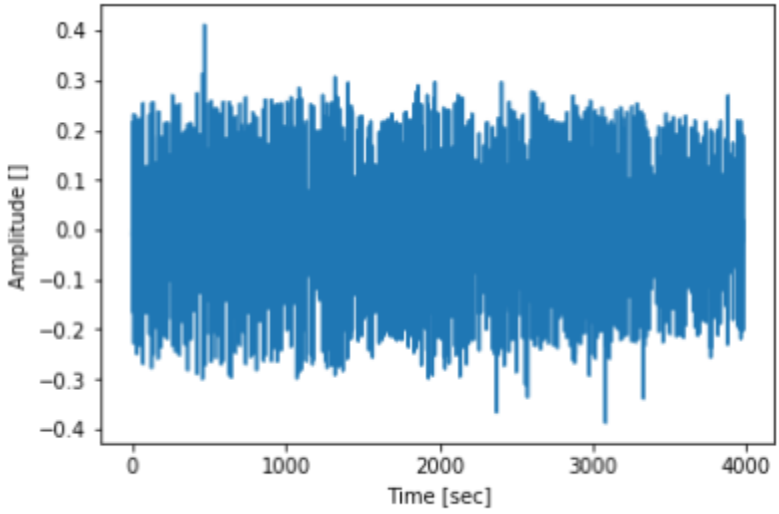

Figure 1: Raw trace of the audio data

of high intensity of noise, but we cannot discern the frequencies of noise during these events. We will further investigate these spikes by making spectrums and spectrograms.

The second step of our spectral analysis is to plot the spectrum for small time windows. The spectrum is a visual representation of frequencies with their powers over some time span. To create the spectrum, fast fourier transform was used to split sound waves from the raw trace into different frequencies with powers.

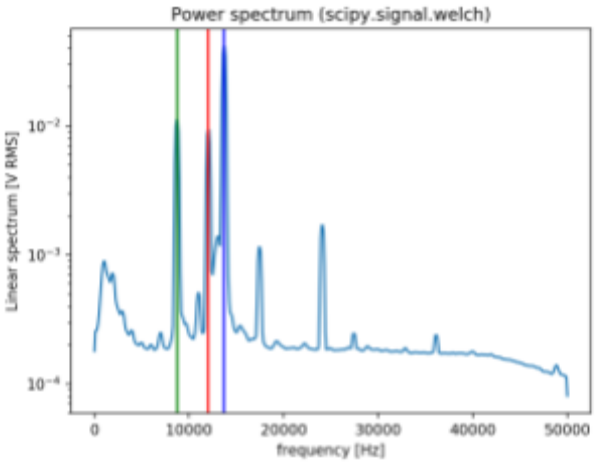

Figure 2: Power Spectrum for the First Minute of Recording

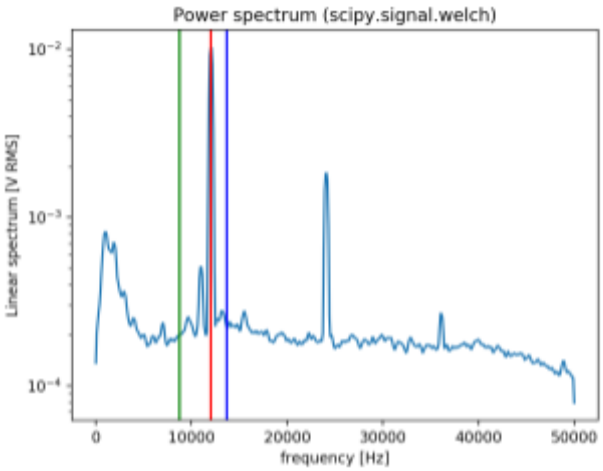

Figure 3: Power Spectrum from a Period of No Activity

Figure 2 and Figure 3 are contrasting spectra from different parts of the recording: one from the first minute of the recording in which both tones were played and one from a period of the recording during which no tones were played. We expect to see a peak at around 12 kHz in both recordings because the machine that emits that noise was running. We do see the peak (marked by the red line) which validates both plots. In Figure 2, we observe two other tones with high peaks (marked by the green and blue lines). These peaks exist at the frequencies of the tones that were played during the recording, and we will explore these frequencies later through plotting the spectrogram and creating filters for event detection. In Figure 3, we observe only the 12 kHz

peak and do not see peaks at the frequencies of the two tones. This is to be expected since the plot is from a period of the recording during which no tones were played.

After validating the ambient recording data with spectra, we plotted spectrograms using the Python packages “matplotlib” and “scipy”. The spectrogram is created by combining spectra from a 8000 sample window and a step size of 1000 samples. In the following spectrogram plots, the x-axis is the time of the recording, the frequencies are shown on the y-axis and the degree of amplitudes (power) is displayed by the color of the pixel in the plot.

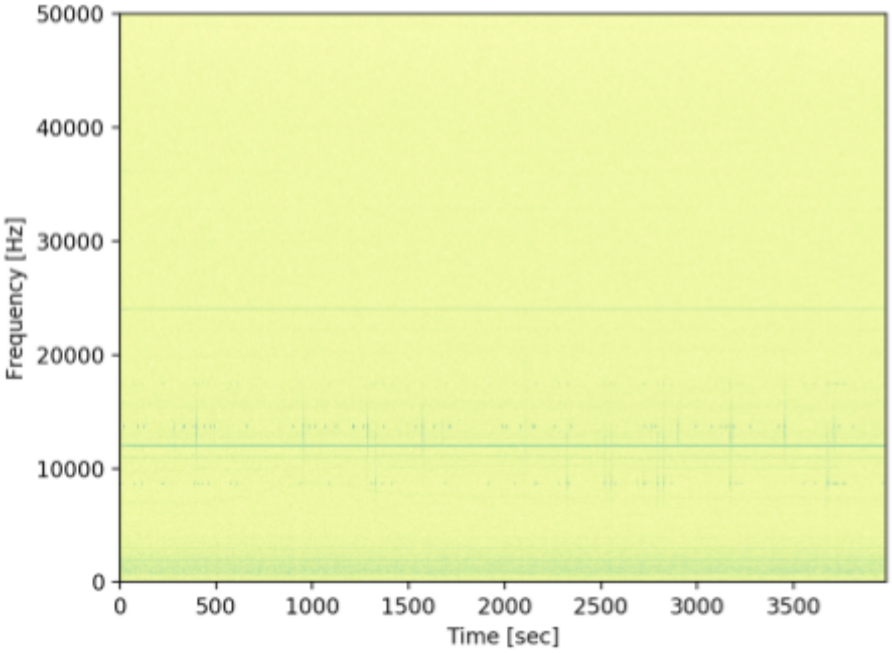

Figure 4: Spectrogram of the Entire Recording

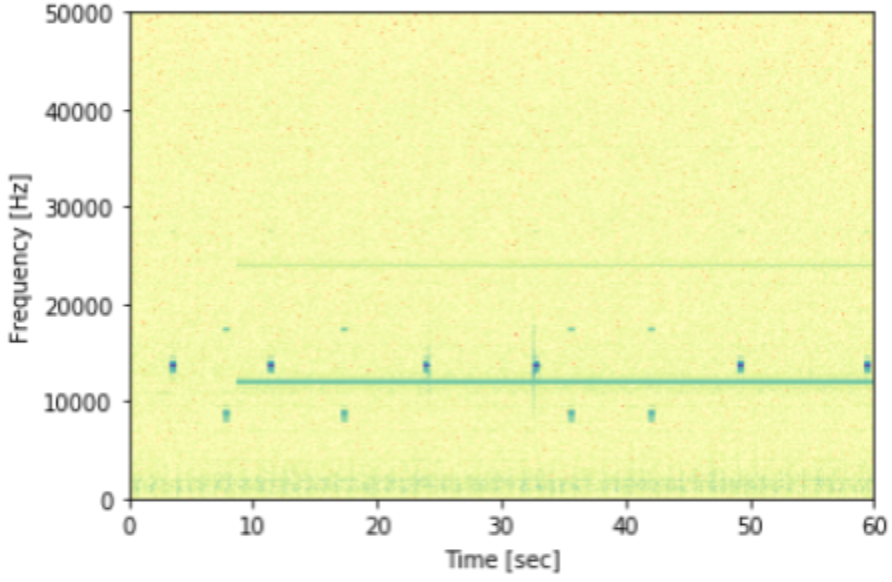

Figure 5: First minute of spectrogram

We could not clearly observe any auditory events in Figure 4, the spectrogram for the whole recording, since the entire recording is large and events become condensed when plotted, but we do see a band at 12 kHz, which represents the sound from the microscope scanner and once again validates our plot.

We narrowed the window of the spectrogram in order to better observe both the 12 kHz peak and the

tones that were presented during the recording. In Figure 5, we clearly see both tones, represented in the spectrogram as blue dots. One tone is a higher frequency than the 12 kHz band and the other is lower than the 12 kHz band. This spectrogram provides a good visual representation of the audio stimuli presented throughout the recording and will guide our feature detection of the tones at each of these frequencies.

Another use for our spectrogram plots is to identify the part of the audio file in which the neural data is being recorded. We will use the length of the 12 kHz peak to once again validate the data, and we will use the onset and offset of the 12kHz peak to determine the part of the audio data for which we have neural recordings. By inspection of the beginning and end of the spectrogram, we calculated the total time of the 12 kHz peak to be 6363.6 s, starting 9 s into the audio recording and ending 3 s before the end. This is within an acceptable margin of the non-audio recording length of 6363.4 s, meaning the time we observe the 12 kHz peak is almost exactly same as the time we know the scanner was on and emitting this frequency. We also know to align the beginning of our neural recording with the 9 s mark of the audio data.

Audio Feature Detection

To conduct feature detection for the audio events, we followed a variety of steps to find which frequencies were active during which periods of time. Using the spectrograms in Figures 4 and 5, we are able to visualize the frequencies and intensities of audio events over time. In the plots, we see the expected band at 12 kHz and two distinct rows of darker points, showing time periods at which the corresponding frequency (on the y axis) had higher intensity.

We needed to know the frequencies of both of the tones in order to determine the times at which the tones were played during the recording. We found the tones by looking at the spectrogram and spectrums we constructed of the audio data to find the peak frequencies in the recording, listed in the table below.

Table 1: Final Peak Frequencies

Final Peak Frequencies
(8712.5, 8775.0)
(12037.5, 12050.0)

(13600.0, 13800.0)

From this table, we see that there is a peak frequency at 12 kHz, which is to be expected given that we know the microscope machinery emits this tone. The other two peak frequencies are of the tones played during the experiment. We found that one tone had a frequency of between 8712.5 and 8775.0 Hz, while the other tone had a frequency of between 13600.0 and 13800.0 Hz. We will use these frequencies to filter our data and find the times the tones were played.

In order to conduct analysis on the relationship of neural data to audio events, we need to find the times at which the two different tones observed in the spectrogram are on and off. To identify these times, we implemented a bandpass filter on the raw audio data for the two frequencies, identified above, that were played during the experiment. We used the "SciPy" package in Python to actually implement our Butterworth bandpass filter for each tone.

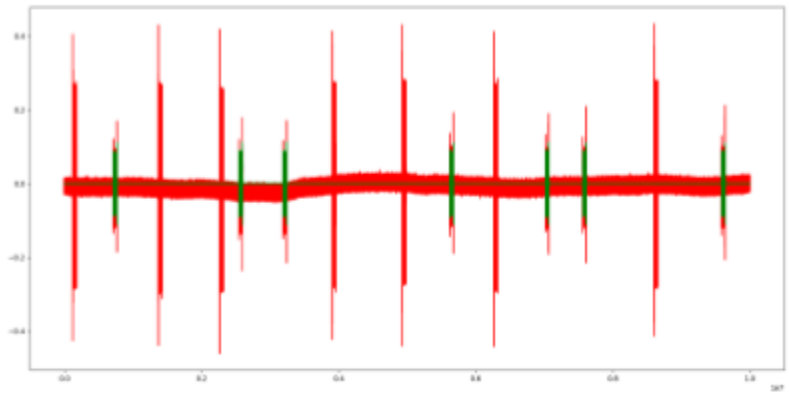

Figure 6: Audio filtered for 8600-8800 Hz range

Figure 6, left, shows the raw trace of the audio signal in red compared to the filtered audio for the frequency of the lower tone in green. From this plot, it is clear to see that the bandpass filter helped us identify the times at which the tone was either on or off.

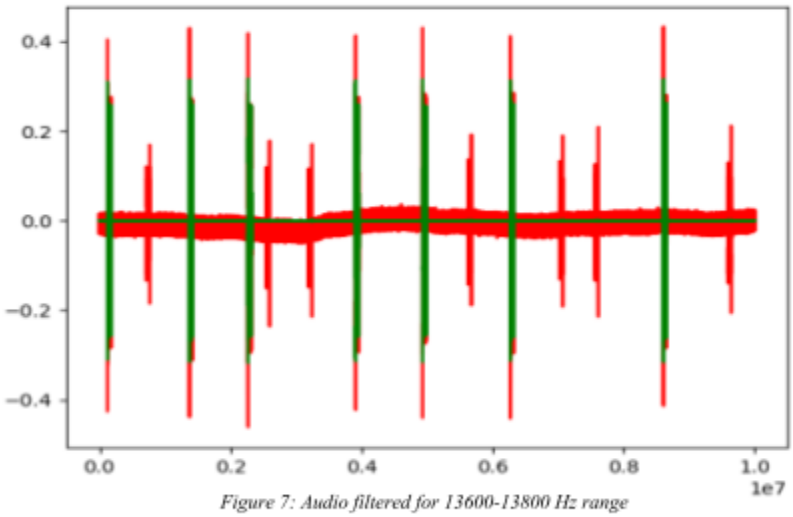

Figure 7: Audio filtered for 13600-13800 Hz range

Figure 7, left, shows the output of the bandpass filter for frequencies in the 13600-13800 Hz range. Once again, we see the raw data in red and the filtered data in green, which shows us that the bandpass filter successfully only let the desired frequencies through.

Once we filtered our data for each of the two known tones, we were able to find the onset and offset times for both of the tones. We used these onset and offset times to create two binary indicator variables that, for the whole length of the recording, will be 0 if the tone is off and 1 if the tone is on.

Using these variables, we will engineer audio features for each of the hypotheses, described in the project goals section, that we seek to evaluate. The features will be binary vectors (arrays of 0s and 1s) that use 1 to indicate the outcome of interest. Notably, we will engineer a variable for the presence of any tone by combining the times at which tone A and tone B are present into a single indicator variable that will be 1 when either tone is expressed and 0 when there is no tone playing. We will also engineer a variable to represent the onset of a tone that will be 1 when either tone is starting (first 5 ms) and 0 when a tone is playing or there is no tone. We will create a similar variable for the offset of the tones as well.

Neural Tuning: Pupil and Treadmill Data

Before we analyzed the neural data in conjunction with the audio data, we wanted to do some preliminary neural tuning to explore relationships between the neural data and the other variables recorded in the non-audio data set, such as pupil radius and treadmill velocity. In our analysis, we aimed to note correlations and trends between variables that would either be important to note as confounders or could help shape our future analysis. This preliminary analysis will help us to better understand the way neuron responses vary with regard to visual stimuli and movement of the mouse. By exploring the neural data in this way, we hope to not only familiarize ourselves with the way neural data looks but also discover some useful data science methods that we can later use with our audio data.

First, we looked at the relationship between cell trace, a measure of the neuron's activity, and treadmill velocity in order to see which cells in the mouse's visual cortex are most related to physical movement of the mouse. The first method we used to assess the relationship between cell trace and treadmill velocity was Pearson's correlation coefficient assessed contemporaneously (without lag).

Figure 8 shows the Pearson correlation coefficient for each neuron's cell trace and treadmill velocity. Most correlation values are between -0.05 and 0.05, with a few neurons exhibiting higher correlation values of over .15. In this plot, we can see that not many neurons are strongly related to treadmill velocity. This plot is also interesting because it gives us an understanding of

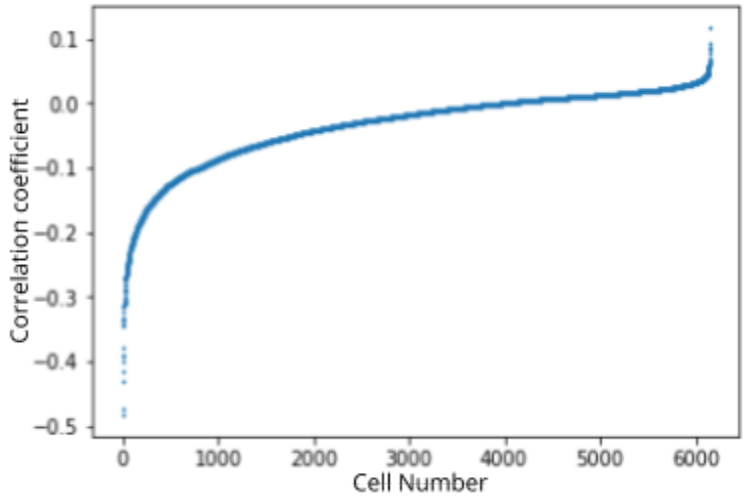

Figure 8: Contemporaneous Pearson's correlation of cell trace and treadmill velocity for each neuron

the way correlation between neural activity and a stimulus is different than correlation in other fields. In neuroscience, a correlation of 0.2 is considered fairly high since neurons respond to many stimuli, which creates a lot of noise in neuron correlation. This will be an important takeaway as we move into audio analysis.

Next, we did a similar analysis to look at the relationship between cell trace and pupil radius in order to see how the neurons that were recorded in this experiment responded to the different visual stimuli that the pupil was responding to. We wanted to conduct this analysis since pupil radius has an expected relationship with audio events (the pupil is thought to dilate when a tone is played, as a consequence of the mouse's surprise). We created a plot to detail the correlation between cell trace and the pupil radius for each of the neuron cells in the visual cortex that were recorded. Our goal for this analysis was to interpret how each of the neural cells was correlated with the pupil radius of the mouse; this will help us understand if certain cells had more neural activity as the pupil radius dilated and contracted and were perhaps more tuned to visual stimuli.

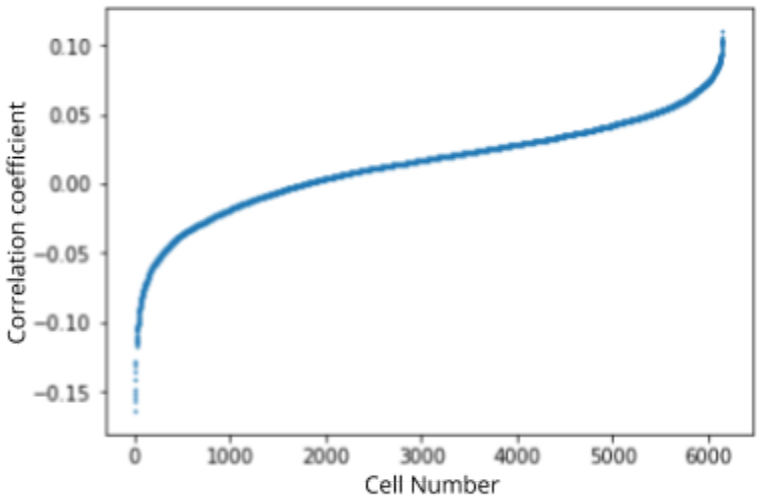

Figure 9: Contemporaneous Pearson's correlation of cell trace and pupil radius for each neuron

Once again we can glean a few important pieces of information from the plot of correlation. We see that the Pearson's coefficient values are fairly low for all cells; they are concentrated between -0.1 and 0.1. Once again, the fact that all the correlations are low can be attributed partly to the type of data we are observing and the fact that correlations in neural data are lower than those in other

fields due to noise in brain activity.

Neural Tuning: Audio Data

Now that we have combined the neural data and auditory indicator variables into a single data set that is properly time-aligned through interpolation, we will be able to investigate our main research questions. We engineered audio features with specific hypothesis, or question we wanted to answer, in mind, so now we will conduct tests between the cell activity and each of the audio features to assess our hypotheses. The relationship between features and their hypotheses is outlined in Table 2, below.

Table 2: Audio features and the hypotheses they will help to evaluate

Hypothesis	**Audio Feature**
1. Visual cortex neurons will show increased activity in response to the presence of any tone	Tone presence (binary variable: 1 for times at which either tone is playing, 0 otherwise)
2. Visual cortex neurons can distinguish between tones A and B	Tone activity (categorical variable: 0 when no tone is playing, -1 when Tone A is playing, 1 when Tone B is playing)
3. Visual cortex neurons will show increased activity in response to the onset of a tone	Tone onset (binary variable: 1 for first 5 ms following onset of a tone, 0 otherwise)
4. Visual cortex neurons will show increased activity in response to the offset of a tone	Tone offset (binary variable: 1 for the first 5 ms following offset a tone, 0 otherwise)
5. Visual cortex neurons will show increased activity in response to specific tones (some neurons will be more tuned to tone A, while others will be more tuned to tone B)	Tone A (binary variable: 1 for times at which the 13700 Hz tone is playing, 0 otherwise) and Tone B (binary variable: 1 for times at which the 8750 Hz tone is playing, 0 otherwise)

We started with computing the correlation coefficient between each of the audio features and the cell trace for each neuron, as we did for the pupil and treadmill data. As an example of the output from these correlations is shown in Figures 11 and 12.

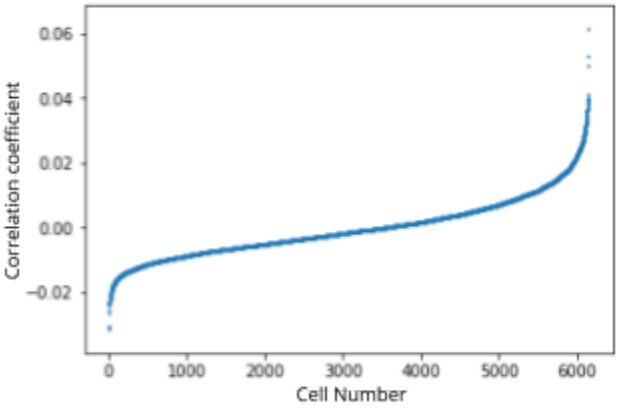

Figure 11: Contemporaneous Pearson's correlation of cell trace and tone A (13700 Hz) for each neuron

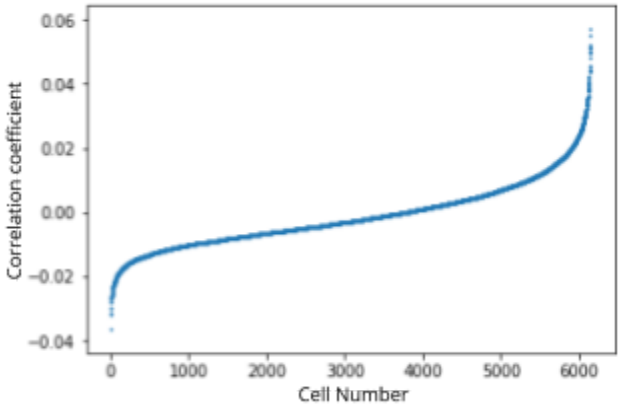

Figure 12: Contemporaneous Pearson's correlation of cell trace and tone B (8750 Hz) for each neuron

The curves of correlation coefficients for cell trace and each of the two tones look similar to the relationship between cell activity and the non-audio variables we explored, which is reassuring that we have calculated these variables correctly, but really not very interesting. Correlation coefficient is not the best way to evaluate the relationship between a cell's activity and a tone, so we quickly moved into modeling for more sophisticated analysis.

For our analysis, we chose to focus on the two most pertinent hypotheses above, hypothesis 1 and hypothesis 2, because they would help us first evaluate if visual cortex neurons respond to sound in general, and then assess if they respond differently to specific tunes. We first used univariate linear regression to find the relationship between each cell and each desired audio feature. In these univariate models, audio events were the predictor variable and neural activity was the output variable. We incorporated varying degrees of lag in the audio data, shifting the audio data between 2.5 seconds ahead and 2.5 seconds behind the neural data, including each .5 second interval in between. Using the 11 p-values generated by these regressions, we picked the lowest one for each cell to find the best lag for each cell individually. We did this because neurons are thought to respond to stimuli at various time delays depending on where they are in the sequence of neurons and other factors, possibly related to location and function, so they likely do not all fire in response to a stimulus at the same time.

Once we found the best lag for each cell and the corresponding p-value, we used those p-values to select the neurons with the strongest relationship to each of the audio features we wished to evaluate. For our modeling, we used a p-value cut off of 0.000001 and selected only the cells with p-values less than this threshold to complete our analysis. The table below illustrates an

example of the cells, lags and p-values we were evaluating (note that this is just for reference, and is not a comprehensive list of all the neurons used to create our models).

Table 3: Cells with lowest p-value when regressed on tone A and tone B

Tone A (13700 Hz)		Tone B (8750 Hz)	
Cell Number	**p-value**	**Cell Number**	**p-value**
4411	4.65E-15	5088	2.00E-13
578	1.34E-16	2832	4.63E-13
3646	6.85E-15	2325	6.23E-13
3022	1.49E-14	5630	1.52E-12
1294	1.93E-14	1710	2.00E-12
3400	2.70E-14	2607	5.24E-12
4930	3.69E-14	2295	5.69E-12
4929	4.01E-14	2611	7.60E-12
4311	1.20E-13	4323	8.29E-12
2855	2.17E-13	2529	1.57E-11
5495	3.86E-13	3563	2.15E-11
5595	1.43E-12	4389	2.85E-11
1345	2.92E-12	4627	5.55E-11
4195	4.61E-12	47	6.59E-11
3673	7.22E-12	2834	7.06E-11
2924	8.36E-12	3358	1.25E-10
4446	4.85E-11	2531	1.72E-10
3762	5.02E-11	1133	2.86E-10
3559	7.09E-11	1307	2.95E-10
1217	8.97E-11	1055	4.71E-10

This step of univariate analysis, also called marginal regression, was useful because it allows us to select different subsets of a neurons that are most related to each different audio feature so that our multivariate regression is more directed and hopefully more successful.

Modeling

Through marginal regression, we used a p-value cut off of 0.000001 to select approximately 52 neurons for each of our two main hypotheses (hypothesis 1: neurons can distinguish between any tone playing or no tone playing and hypothesis 2: neurons can distinguish between no tone playing, tone "A" playing or tone "B" playing) that we we will use in our modeling techniques. We believe that since these neurons are the most significantly related to their corresponding audio feature, they will be strong predictors to help us evaluate our hypotheses. The goal of our modeling phase was the following: given a particular level of neuron activity in the visual cortex (isolating the neurons by significance), can our model(s) accurately predict whether or not a tone was playing at that time and can it accurately discriminate between tones playing. To answer this, we tried three separate modeling approaches: Logistic Regression, Random Forest and Decision Tree.

First, we will look at whether these models can predict whether either of the tones were playing (using the "any tone" feature) based on the neural activity.

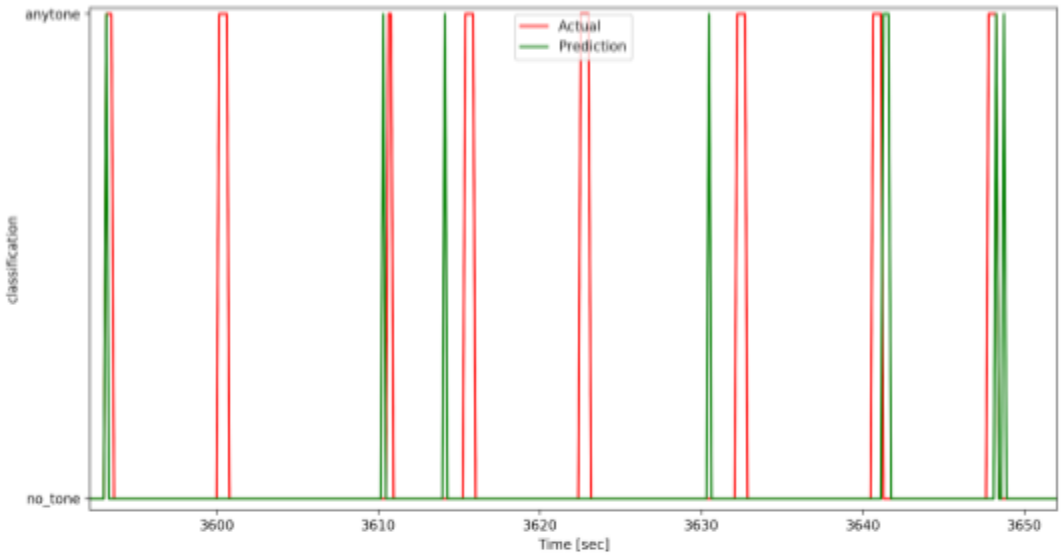

Figure 13: Plot representing Logistic Regression predicted events and actual events for any tone vs no tone

Our best results came from our Logistic Regression model where we achieved a 65.8% accuracy. Accuracy here is defined as the number of correctly predicted audio events divided by the total number of audio events. Figure 13, above, shows the actual events compared to events predicted

by the Random Forest using our selected neurons. We observe that the Random Forest model does not predict all of the actual events, but it does get a majority of them. Our second best result came from the Decision Tree where we achieved an accuracy of 63% and our worst result was the Random Forest where we achieved an accuracy of 61.9%. Our result of 65.8% is relatively strong given the context of what we are trying to predict, especially the noisiness of neuron response in the brain. This model has a greater than 50% chance of correctly predicting an event at any moment in time over the course of the experiment. Since neurons can roughly predict when any tone is on versus when not tone is on, we interpret these results to mean that visual cortex neurons are attentive to audio.

Now that we have used neuron activity to predict whether a tone any tone is playing, will attempt to train a model that can both determine whether a tone is playing or not and discriminate between two unique tones.

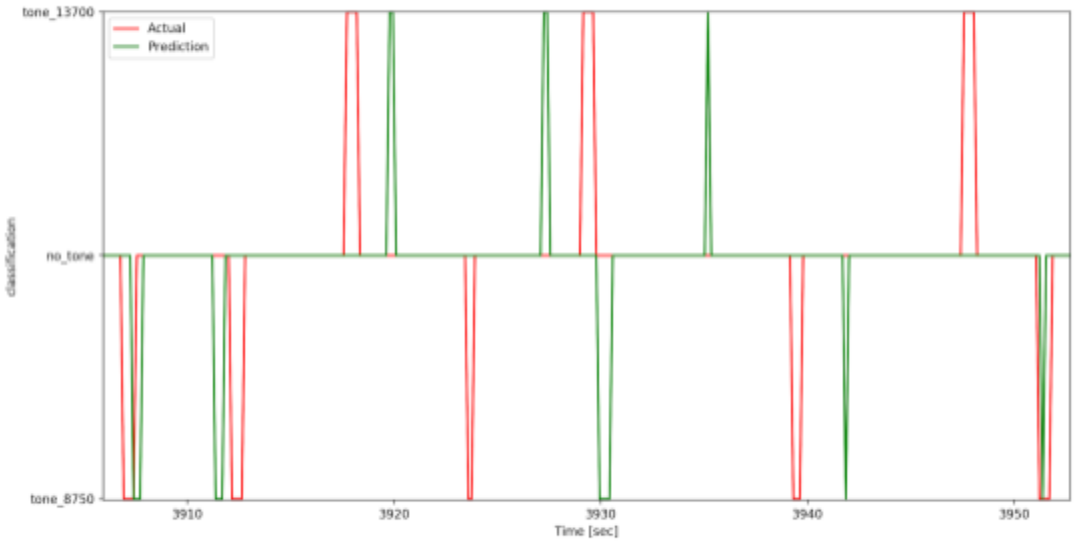

Figure 14: Plot representing Logistic Regression predicted events and actual events for tone A vs tone B vs no tone

Again, our best result came from our Logistic Regression model where we achieved an accuracy of 37.7% (accuracy is once again defined as the number of correctly predicted events divided by total number of audio events). Our second best result was the Random Forest with an accuracy of 35.5%, followed by the Decision Tree with an accuracy of 34.1%. Our results, not surprisingly, are much lower in this case, since our model has to not only predict whether a tone is playing but also discriminate between two unique tones.

We interpret this accuracy to mean that visual cortex neurons do not have much capability to distinguish between different tones. In our interpretation of these results, it is also important to consider the domain and experiment in which we are working: we are trying to determine whether or not the visual cortex (not the auditory cortex!) has any ability to determine whether sound is being played and discriminate between distinct sounds. Given this information, while our results may not be particularly strong, they do convey some evidence that the visual cortex has a role to play in the brain's ability to understand and process auditory events.

Conclusion

Through our analysis, we were able to successfully process the audio data and build audio features based on the times at which each of the tones were played. We used correlation coefficients to do some preliminary exploratory data analysis of the relationships between neural activity and other variables before turning to univariate and multivariate predictive modeling to evaluate the relationships between audio events and neural activity. We used a two-step approach to first select the neurons most related to the audio events through marginal univariate regression and then predict audio events based on the activity of these selected neurons using different types of models.

We found that visual cortex neurons are attentive to audio, since our models were able to predict when any tone was playing fairly well using only the neural data. Our models for distinguishing between tones were less accurate, which leads us to believe that these neurons do not fully process the characteristics of audio events. Based on these results, we conclude that visual cortex neurons have some audio processing capabilities, but not full audio processing.

Now that we have explored how visual cortex neurons respond to outside tones, an interesting secondary analysis would be to investigate how they respond to noises created by the mice themselves. We could use similar techniques to the ones used in spectral analysis we conducted identify mouse tones, based on sex, and determine if there is any causal relationship between mice hearing sounds of the opposite sex while they were on the treadmill, and if this changed their immediate neural activity. This is an achievable goal, as there is evidence to prove that a female mouse vocalizes during interactions with a male, and that the acoustic features of female and male vocalizations differ during specific behavioral contexts (Warren et al., 2018).

Appendix

COMPUTING PACKAGES USED

Librosa

A python package for music and audio analysis that we used for loading audio input, computing mel spectrograms, locating beat events, and saving beat tracker outputs to CSV files.

Source: McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. "librosa: Audio and music signal analysis in python." In Proceedings of the 14th python in science conference, pp. 18-25. 2015.

MatPlotLib

A Python 2D plotting library that we used for generating plots, histograms, power spectra, bar charts, error charts, and scatter plots.

Source: Matplotlib: A 2D Graphics Environment" by J. D. Hunter In Computing in Science & Engineering, Vol. 9, No. 3. (2007), pp. 90-95

Scipy

A free and open-source Python library used for scientific computing and technical computing, specifically in optimization and interpolation.

Source: Travis E. Oliphant. Python for Scientific Computing, Computing in Science Engineering, 9, 10-20 (2007),DOI:10.1109/MCSE.2007.58 (publisher link)

K. Jarrod Millman and Michael Aivazis. Python for Scientists and Engineers, Computing in Science & Engineering, 13, 9-12 (2011), DOI:10.1109/MCSE.2011.36 (publisher link

Butterworth

This cookbook recipe demonstrates the use of scipy.signal.butter to create a bandpass Butterworth filter. scipy.signal.freqz is used to compute the frequency response, and scipy.signal.lfilter is used to apply the filter to a signal.

Source: Weckesser, Warren. https://scipy-cookbook.readthedocs.io/items/ButterworthBandpass.html. Web. 24 Sept. 2018.

LITERATURE REFERENCED

Kato, HK. et al. “Network-Level Control of Frequency Tuning in Auditory Cortex.” Neurons. July 19 (2017): 037. Web. 31 Oct 2018.

L. S. Petro, A. T. Paton, L. Muckli Phil. Trans. R. Soc. B 2017 372 20160104; DOI:10.1098/rstb.2016.0104. Published 2 January 2017

Morrill RJ et al. “ Visual Information Present in Infragranular Layers of Mouse Auditory Cortex.” Ed. Kevin Paterson. PLoS ONE 8.9 (2013): e74986. PMC. Web. 24 Sept. 2018.

Pnevmatikakis, EA et al. “Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data.” Neurons. Jan 20 (2016): 037. Web. 24 Sept. 2018.

“Research.” Andreas Tolias Lab, 1 July 2011, toliaslab.org/research/.

Scheepers, Christoph et al. “Listening to Limericks: A Pupillometry Investigation of Perceivers’ Expectancy.” Ed. Kevin Paterson. PLoS ONE 8.9 (2013): e74986. PMC. Web. 24 Sept. 2018.

Schormans, Ashley L. et al. “Audiovisual Temporal Processing and Synchrony Perception in the Rat.” Frontiers in Behavioral Neuroscience 10 (2016): 246. PMC. Web. 24 Sept. 2018.