UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

ANTONIO HENRIQUE DE OLIVEIRA FONSECA

Detection and classification of ultrasonic vocalizations from neonatal mice using machine learning

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Microeletronics

Advisor: Prof. Dr. Sergio Bampi
Coadvisor: Prof. Dr. Marcelo O. Dietrich

Porto Alegre
July 2019

CIP — CATALOGING-IN-PUBLICATION

Fonseca, Antonio Henrique de Oliveira

Detection and classification of ultrasonic vocalizations from neonatal mice using machine learning / Antonio Henrique de Oliveira Fonseca. – Porto Alegre: PGMICRO da UFRGS, 2019.

88 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Microeletrônica, Porto Alegre, BR–RS, 2019. Advisor: Sergio Bampi; Coadvisor: Marcelo O. Dietrich.

1. Vocal Behavior. 2. Open-source Software. 3. Convolutional Neural Network. 4. Diffusion Maps. 5. Manifold Alignment. I. Bampi, Sergio. II. Dietrich, Marcelo O.. III. Título.

“If I have seen farther than others,
it is because I stood on the shoulders of giants.”
— Sir Isaac Newton

AKNOWLEDGEMENTS

The present work would not be possible without the help of several people along the way.

First, I am be eternally grateful to Dr. Marcelo Dietrich and Dr. Sergio Bampi for the excellent mentorship and patience with which they guided me throughout the past years. Thank you for all the constructive comments and correction.

I would like to thank my labmates for all the help and feedback I have received. Your help with the experiments and our hours of brainstorming were definitely essential to the success of this work.

In the same way, I am thankful for the support my family has given to me since always. I got to where I am today because you never gave up on me despite all the difficulties.

And to my girlfriend, Kendall, and friends: you have been my family away from home. Thank you for being so understanding during the sleepless nights and weekends at home studying. Thank you for all the help reviewing and sending feedback on my work, and most of all, thank you for being part of my life.

ABSTRACT

The study of animal behavior has fascinated scientists for hundreds of years. An important source of behavior information can go unnoticed to heedless ears, which is the emission of ultrasonic vocalization (USV) by certain species of mammals, such as mice. With the goal of having an accurate and flexible system to detect and classify USVs emitted by mice, here we describe the development of VocalMat, a software tool to analyze USVs in audio files. VocalMat uses image-processing and differential geometry approaches to detect USVs in spectrograms, eliminating the need of user-defined parameter tuning. Moreover, VocalMat classification module uses computational vision and machine learning methods to classify USVs into distinct categories. In a data set of >4,000 USVs emitted by infant mice, VocalMat detected more than $> 98\%$ of the USVs and accurately classified $\approx 85\%$ of USVs when considering the most likely label and $\approx 95\%$ when considering the two most likely labels. We used Diffusion Maps and Manifold Alignment to analyze the probability distribution of USV classification among different groups, which provided a robust method to quantify and qualify the vocal repertoire of mice in different experimental conditions. Thus, VocalMat allows accurate and highly quantitative analysis of USVs, opening the opportunity for detailed analysis of this behavior.

Keywords: Vocal Behavior. Open-source Software. Convolutional Neural Network. Diffusion Maps. Manifold Alignment.

Detecção e classificação de vocalizações ultra-sônicas de camundongos neonatais usando aprendizado de máquina

RESUMO

O estudo do comportamento animal fascina cientistas há centenas de anos. Uma fonte importante de informações sobre o comportamento pode passar despercebida a ouvidos descuidados, que é a emissão de vocalização ultrassônica (USV) por certas espécies de mamíferos, tais como camundongos. Com o objetivo de ter um sistema preciso e flexível para detectar e classificar USVs emitidos por camundongos, aqui descrevemos o desenvolvimento do VocalMat, uma ferramenta de software para analisar USVs em arquivos de áudio. O VocalMat usa abordagens de processamento de imagem e geometria diferencial para detectar USVs em espectrogramas, eliminando a necessidade de ajuste de parâmetro definido pelo usuário. Além disso, o módulo de classificação VocalMat usa visão computacional e métodos de aprendizado de máquina para classificar USVs em categorias distintas. Em um conjunto de dados de >4.000 USVs emitidos por filhotes de camundongos, VocalMat detectou mais de 98% dos USVs e classificou com precisão $\approx 85\%$ de USVs ao considerar categoria mais provável e $\approx 95\%$ ao considerar as duas categorias mais prováveis. Utilizamos Diffusion Maps e o Manifold Alignment para analisar a distribuição de probabilidade da classificação de USV entre diferentes grupos, o que forneceu um método robusto para quantificar e qualificar o repertório vocal em diferentes condições experimentais. Assim, o VocalMat permite uma análise precisa e altamente quantitativa das USVs, abrindo a oportunidade para uma análise detalhada deste comportamento.

Palavras-chave: Comportamento Vocal, Open-source Software, Convolutional Neural Network, Diffusion Maps, Manifold Alignment.

LIST OF ABBREVIATIONS AND ACRONYMS

CDF	Cumulative Distribution Function
CNN	Convolutional Neural Network
FM	Frequency modulated
LMF	Local median filtering
MSE	Mean squared error
SNR	Signal-to-noise ratio
STFT	short-time Fourier Transform
USV	Ultrasonic vocalization

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Vertebrates use vocal communication to transmit information about the state of the caller and influence the state of the listener. This information can be relevant for the identification of individuals or groups (HOFFMANN; MUSOLF; PENN, 2012); group status (e.g.: dominance, submissive, fear or aggression) (NYBY; DIZINNO; WHITNEY, 1976); next likely behavior (e.g.: approach, flee, play or mount) (NEUNUEBEL et al., 2015); environment conditions (e.g.: presence of predators, location of food) (SLOBODCHIKOFF et al., 2012); and facilitation of mother–offspring interactions (D'AMATO et al., 2005).

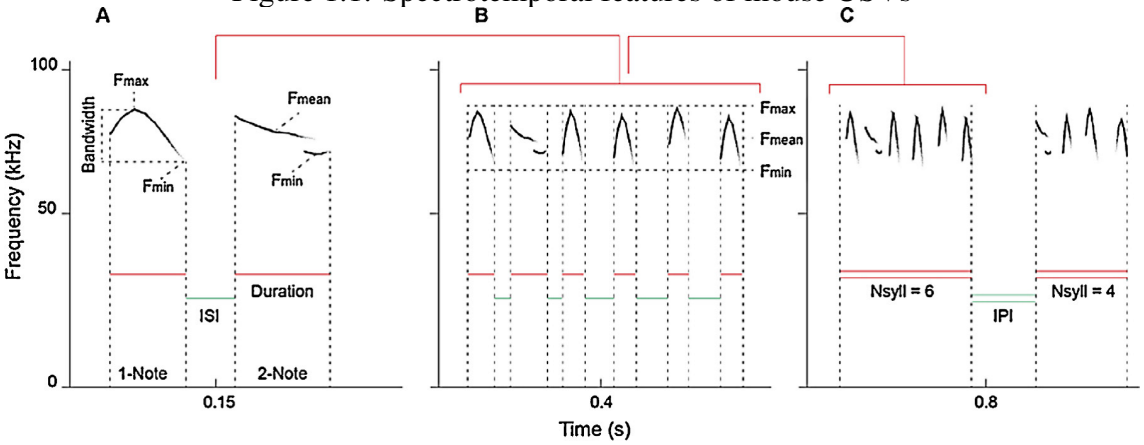
Ultrasonic vocalizations (USVs) emitted by mice occur in the frequency range between 30 – 110 kHz, above the human hearing range (≈2 – 20kHz). These USVs are organized in *phrases* or *bouts* composed by sequences of *syllables*. The syllables are defined as continuous units of vocal sound not interrupted by a period of silence. The syllables are composed of one or more notes and are separated by salient pauses and occur as part of sequences (ARRIAGA; ZHOU; JARVIS, 2012) (Figure 1.1). These transitions across syllables do not occur randomly (HOLY; GUO, 2005), and the changes in syllables sequences, prevalence and acoustic structure match current behavior (CHABOUT et al., 2015), genetic strain (SEGBROECK et al., 2017; SCATTONI; RICCERI; CRAWLEY, 2011), and developmental stage (GRIMSLEY; MONAGHAN; WENSTRUP, 2011). USVs are most commonly emitted by mouse pups (SCATTONI et al., 2008) and are modulated during development (GRIMSLEY; MONAGHAN; WENSTRUP, 2011). In the adult mouse, USVs are emitted in both positive and negative contexts (ARRIAGA; JARVIS, 2013). Thus, understanding the complex structure of USVs emitted by mice is key to advancing vocal and social communication research in mammals.

1.1 M.Sc. Thesis Organization

In this dissertation, we first review the literature in the analysis of USVs emitted by mice and highlight some key deficiencies (Chapter 1). Next, we present VocalMat, a software tool able to accurately detect and classify USVs in audio files. VocalMat makes use of image-processing and differential geometry techniques to detect USVs in spectrograms (Chapter 2.3 and 3.1), eliminating the need of parameter tuning. Moreover, VocalMat uses computational vision techniques and machine learning methods to classify

USVs (Chapter 2.6 and 3.2). The output of VocalMat is then visualized through non-linear dimensionality reduction (Chapter 2.7.1). We then present an example of application of VocalMat by quantifying differences in vocal behavior of mice under different experimental conditions (Section 3.3). At the end, we summarize our findings ((Section 4) and set the next steps to this project ((Section 4).

Figure 1.1: Spectrotemporal features of mouse USVs



Source: (HECKMAN et al., 2016)

Figure 1.1: **(A)** Spectrotemporal features of a vocalization in a typical spectrogram featuring indication of syllable duration, and various properties regarding the frequency **(F)**, i.e. F_{max}, F_{min}, F_{mean}, F_{start}, F_{end}, and bandwidth. The red continuous lines indicate the duration of individual syllables. Green continuous lines indicate the duration of the inter-syllable intervals (ISI). The black dashed vertical lines indicate the start and ending of a syllable/phrase. **(B)** Graphical representation of a phrase, with red continuous lines indicating the duration of individual syllables. **(C)** Representation of a song with the red double continuous lines indicating the duration of each phrase, while the green double continuous line indicates the inter-phrase interval (IPI).

1.2 Earlier methods used for the analysis of mouse USVs

An initial attempt to classify USVs from mice was made by Holy and Guo (HOLY; GUO, 2005) based on observations of Sewell and colleagues on mating behavior, in which relatively sudden, large changes in frequency were reported (SEWELL, 1972). In their approach, each syllable was described by extracting the dominant frequency (or pitch) as a function of time. For each syllable, the frequency at an instant t was compared to its neighboring frequencies in bins of approximately 1 ms. In an experiment with 15,543 syllables, four distinct clusters of frequency changes were identified. The first two clusters were labelled as (1) *downward* (from high to low frequency) and (2) *upward* (from low to high frequency) jumps, both described as 'low jumps' due to their frequency range (35-50 kHz). The third cluster was labelled as (3) *high jumps*, represented by frequency jumps

from 70-90 kHz to 55-70 kHz. The fourth cluster was represented by the gradual shift in frequency occurring at most time points, and was characterized based on the fitting to sine waves by scaling and shifting both time and frequency axes for maximal alignment. This fourth type of USVs were labelled as *sinusoidal sweeps*. The authors concluded that discrete categories of syllables exist. Even though the classification system was based on the frequency jumps, the authors acknowledged that other USV features could also be important for a more robust classification system.

In addition to the frequency jumps characterized by Holy and Guo (HOLY; GUO, 2005), a subsequent work (PORTFORS, 2007) described sub-types of simple syllables consisting of single harmonic whistles, labelling them as u-shaped modulated frequencies, frequency modulated up-sweeps and down-sweeps, constant frequencies and hump-shaped modulated frequencies. No specific tool was developed in order to quantitatively describe the syllable categories and counting was manually performed by the investigators.

Soon after, syllable categories were extended to 10 distinct types, based on internal frequency changes, lengths and shapes (SCATTONI et al., 2008). It included categories such as complex, harmonics, two-syllable, composite, short, frequency steps, flat and chevron. The number of syllable categories was again extended to incorporate reverse chevron, low frequency harmonic syllables and noisy syllables (GRIMSLEY; MONAGHAN; WENSTRUP, 2011). However, no automatic method was developed to classify the vocal repertoire, which demanded substantial inputs from the investigator and manual inspection of the USVs.

Arriaga and colleagues developed a syllable identifier based on a modified version of Holy and Guo's tool (ARRIAGA; ZHOU; JARVIS, 2012). Syllables with duration longer than 10 ms were identified and classified according to the presence or absence of instantaneous 'frequency jumps' separating notes within a syllable. The morphologically simplest note type that did not contain any frequency jumps was classified as Type A. The next most complex contained two notes separated by a single upward or downward frequency jump (Types B and C, respectively). More complex syllables were identified by the series of upward and downward frequency jumps occurring as the fundamental frequency varies between notes of higher and lower frequency (Types D–K). Much rarer syllable types (about 1%) were grouped together. The changes in acoustic features across animals and/or experiments were inferred through the analysis of spectral features (e.g.: mean frequency, frequency modulation, spectral purity and standard devi-

ation of frequency distribution) calculated from the spectrograms of each syllable type. In total, this tool classified 8 common and 3 rare syllable categories ordered in increasing complexity based on the number and direction (downward or upward) of instantaneous frequency jumps. Of note, no validation for their classification method was presented.

Chabout and colleagues developed a new version of Arriaga's customization, named Mouse Song Analyzer v1.3 (CHABOUT et al., 2015). The main modifications were (1) the decrease of the minimum duration threshold for USV detection to 3 ms (in contrast to 10 ms adopted by Arriaga (ARRIAGA; ZHOU; JARVIS, 2012)) and (2) the minimum separation time between two syllables to 10 ms (no minimum separation interval was found for Arriaga's customization). The authors classified the USVs according to their frequency changes into four categories: upward, downward, multiple pitch jumps, and no jumps; similarly to Holy and Guo's (HOLY; GUO, 2005). Using this method, up to 16% of total detected USV candidates were not classifiable. These unclassified USV candidates had diverse sources, such as syllables overlapping with mechanical noise and non-vocal noise made by the animals (e.g., scratching, walking, chewing and aggressive behavior). Comparisons between automated and manual methods on example sonograms from all syllable categories found about 95% overlap. In their analysis, sub-types of USVs without frequency jumps were not evaluated.

Also in 2015, a new method was developed to detect USVs in audio files (NEUNUEBEL et al., 2015). This tool was not developed to classify USVs, but simply to detect it in an array of microphones with the goal of identifying the mouse vocalizing in a social group. The method consisted of a multi-taper spectral (MT) analysis and was named Ax (Acoustic Segmenter). After removing signals below 30kHz, overlapping segments in time were Fourier transformed using multiple discrete prolate spheroidal sequences as windowing functions. An F-test was used to infer whether each time-frequency point was significantly above noise based on these independent estimates of intensity. The data were combined in a single spectrogram whose pixel size corresponded to the time resolution of the shortest segment and frequency resolution of the longest. An interesting aspect of Ax is the use of image processing libraries to detect USVs with intensity above background noise. Additionally, the use of a P-value based statistical test to detect pixels of higher intensity than the background allowed for correction for a dynamic (rather than static) background noise. In the original publication, we did not find any reference to the accuracy of the method in identifying USVs in audio files. Also, the method has some contingencies that make USVs of short duration, but relevant changes in frequency, rarely

detectable.

1.3 Latest methods developed for the analysis of mouse USVs

1.3.1 VoICE: Vocal Inventory Clustering Engine

An unsupervised approach for grouping vocal elements into categories was developed in 2015 and named VoICE (BURKETT et al., 2015). VoICE uses audio files trimmed with only one USV per file as input. In order to divide the original audio file in multiple files containing only one USV, Sound Analysis Pro (SAP) - a software tool originally developed to segment birdsong vocalizations - was used for this pre-processing step (TCHERNICHOVSKI et al., 2000). Syllables were scored in pairwise fashion to determine acoustic similarities. An average frequency for every 0.9 ms of an USV was calculated, leading to poor performance when dealing with harmonic components. The comparison between USVs was done by calculating Pearson correlation of raw frequencies, such that USVs with similar contour, frequency range, and temporal overlap would have correlation scores closer to 1. This high degree of dimensionality provided greater specificity in grouping similar USVs, as compared to clustering methods based on a finite number of acoustic features. The spectral co-similarity relationships between syllables were next subjected to hierarchical clustering, to generate a dendrogram, which was then trimmed into clusters using an automated tree-pruning algorithm (LANGFELDER; HORVATH, 2007). The high degree of call-to-call variability caused VoICE to detect multiple clusters of USVs, indicating that the number of call types was highly sensitive to the amount of variability allowed within each cluster. Key advantages of VoICE over other clustering methods included that the number of clusters (in this case, syllable or call types) was not dictated by the experimenter, providing for unbiased calculation of vocal repertoire.

1.3.2 MUPET: Mouse Ultrasonic Profile ExTraction

Another recent work on syllable classification was published by Van Segbroeck and colleagues (SEGBROECK et al., 2017). In their work, MUPET (Mouse Ultrasonic Profile ExTraction) addresses the challenge of USV detection using an optimization of signal-to-noise ratio (SNR) to maximize syllable detection. Such optimization includes

the calculation of the power of the spectral energy for the signal greater than a noise floor threshold. Such thresholding method restricts the possibility of dynamic removal of background noise. In addition to this threshold, the user also sets the minimum and maximum syllable duration; minimum, total and peak syllable energy, and the minimum inter-syllable interval that is needed to separate rapidly successive notes into distinct syllables. Next, MUPET calculates a spectrogram by using a short-time Fourier Transform (STFT) with analysis window of 2 ms that is shifted every 1.6 ms and then normalized to unit energy to prevent the decomposition process from being dominated by high-energy syllables. Gammatone filters are applied in order to reduce the dimensionality of the data maintaining their salience. In this process, the extracted syllable shapes are centered along the time and frequency axes and subsequently vectorized before stacking into a data matrix. The algorithm iteratively clusters the syllables by spectral shape using K-means, consequently the frequency range of the syllable is not a parameter for clustering. As consequence of the clustering method adopted, MUPET also depends on the user to define the number of different syllable types that are present. MUPET provides four measures of model strength (Bayesian information criterion, average likelihood of the centroid, overall repertoire score and goodness of fit for the elements in the cluster) for each repertoire size to aid the user in selecting an appropriate number of categories, but the number of clusters is ultimately defined by the user.

1.3.3 DeepSqueak: deep-learning based method for detection and analysis of USVs

The most recent work on the field of USV analysis is DeepSqueak (COFFEY; MARX; NEUMAIER, 2019). This is the first software tool to apply deep learning methods to analyze USVs from mice (and rats). DeepSqueak uses regional convolutional neural networks (Faster-RCNN) (REN et al., 2017) for USV detection and classification. By using such approach, the authors claim to increase detection rate, reduce false positives and analysis time, besides classifying calls and perform syntax analysis automatically.

Regarding USV detection, DeepSqueak uses four trained networks: (1) mouse USVs; (2) short rat USVs; (3) long 22kHz rat USVs; and (4) general purpose network. These networks were trained with manually labelled USVs and their results show high recall, according to the authors. To work around the mechanical and electrical noise detected by their networks as USV candidates, the authors included a post-hoc de-noising network, in which the user can train a network to identify different kinds of noise. Each

USV candidate detected has its contour calculated by dividing the geometric mean of the power spectrum by the arithmetic mean and subtracting from 1, which results on the removal of the non-tonal features of the USV, such as broad spectrum noise and harmonic components, which are typically features of mice USVs (SCATTONI et al., 2008; GRIMSLEY; MONAGHAN; WENSTRUP, 2011). These contours are then used for USV classification.

DeepSqueak offers two methods for USV clustering. The first applies K-means clustering based on three weighted inputs: shape, frequency, and duration calculated in 10 points along the USV. The number of clusters might be set by the user or estimated by the Elbow method (ALDENDERFER; BLASHFIELD, 1984). The second method is based on time-warping and adaptive resonance theory neural network. For both clustering methods, the clusters may be manually labelled or removed according to the users discretion. DeepSqueak also includes a supervised neural network classification method, which classifies USVs into five categories (split, inverted U, short rise, wave and step). For this classification task, a Convolutional Neural Network was trained, although the authors suggest that their unsupervised method to be more efficient (COFFEY; MARX; NEUMAIER, 2019).

DeepSqueak was shown to outperform MUPET's precision and recall rates in detecting USVs under different levels of Gaussian white noise (COFFEY; MARX; NEUMAIER, 2019). However, the criterion or definition of an USV used for manual validation of datasets was not clear, making it difficult to comparatively test the performance of DeepSqueak.

1.3.4 Summary of published methods for USV detection and classification

In conclusion, major advances in USV detection and classification have been made in the past years. Regarding USV detection, the majority of the tools available still depend substantially on user inputs, which compromises the usability of these tools to compare results across laboratories and to analyze large data sets of audio files. Along those lines, DeepSqueak seems to be the most autonomous USV detection method developed thus far. In terms of USV classification, the lack of consensus regarding USV types and categories and the lack of understanding of the biological function of different USVs, preclude a better definition of a most valid method (supervised or unsupervised) to classify USVs.

During my dissertation, we aimed at developing a robust and automated method

to detect mouse USVs in audio files. Additionally, my goal was to classify USVs automatically, providing a distribution of probabilities for each detected USV as a function of known USV types. The software tool that we developed was named VocalMat and makes use of image-processing and differential geometry approaches to detect USVs in spectrograms, eliminating the need of parameter tuning. Moreover, VocalMat uses computational vision techniques and machine learning methods to classify USVs. VocalMat shows very high sensitivity in detecting USVs, outperforming previous tools. Additionally, the probabilistic classification method allows the analysis of mouse vocal repertoire by nonlinear dimensionality reduction tools as exemplified by the application of Diffusion Maps and Manifold Alignment to an experimental data set. Thus, VocalMat is an alternative to other methods to detect and classify mouse USVs in an automated and flexible manner that can easily be used by any experimentalist. Because VocalMat is an open-source software, it can also be easily customized to different experimental needs.

2 VOCALMAT: METHODS FOR DATA ACQUISITION, PROCESSING, AND CLASSIFICATION

2.1 Animals

All mice used to record the emission of USV were 5-15 days old from both genders. Dams used were 2–6 months old and were bred in our laboratory. The following mouse lines purchased from The Jackson Laboratories were used: C57Bl6/J, NZO/HlLtJ, 129S1/SvImJ, NOD/ShiLtJ, and PWK/PhJ. All mice were kept in temperature- and humidity-controlled rooms, in a 12/12 hr light/dark cycle, with lights on from 7:00 AM to 7:00 PM. Food and water were provided *ad libitum*. All procedures were approved by the IACUC at Yale University School of Medicine.

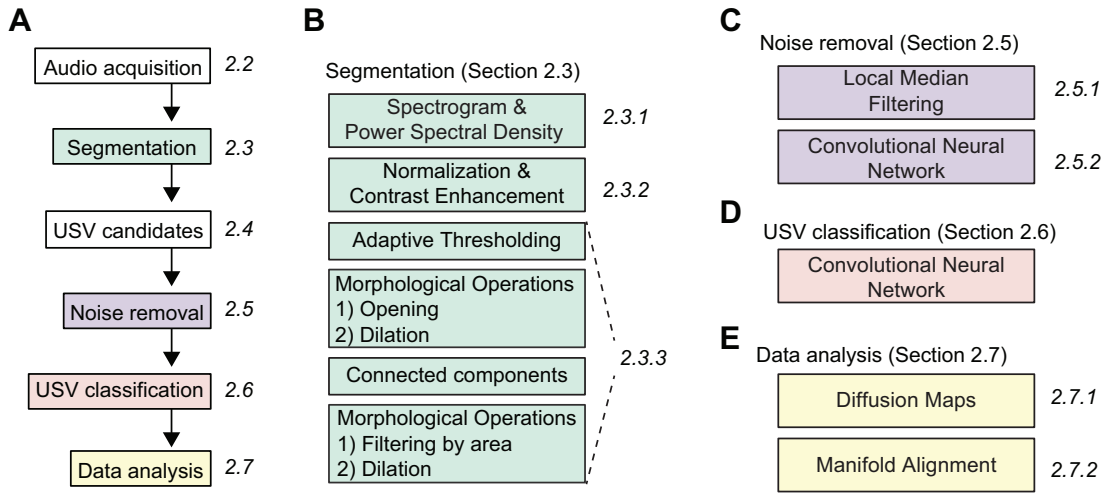
2.2 Audio Acquisition

Mice were placed inside a box (40 x 40 x 40 cm) covered by anechoic material (2" Wedge Acoustic Foam, Auralex) in order to attenuate external noise. Four boxes were recorded simultaneously containing one mouse in each. Audio files were recorded using the recorder module UltraSoundGate 416H and a condenser ultrasound microphone CM16/CMPA (Avisoft Bioacoustics, Berlin, Germany) placed 15 cm above the animal unless otherwise stated. The experiments were recorded with a sampling rate of 250 kHz. The recording system had a flat response for sounds within frequencies between 20 kHz and 140 kHz, preventing distortions for the frequency of interest. The recordings were made by using Avisoft RECORDER 4.2 (version 4.2.16; Avisoft Bioacoustics) in a Laptop with a processor Intel i5 2.4 GHz and 4 GB of RAM. Using these settings, ten minutes of audio recording generated files of approximately 200 MB.

2.3 Segmentation of USV candidates

VocalMat was developed to make use of the newest libraries of Matlab 2018b. The general workflow is summarized in Figure 2.1.

Figure 2.1: Diagram representing the major steps used by VocalMat in the analysis of audio files



Source: The author

Figure 2.1: **(A)** Workflow of the main steps used by VocalMat, from audio acquisition to data analysis. **(B)** Steps used by VocalMat within the segmentation process. **(C)** The two steps used for noise removal. **(D)** USV classification is performed by a Convolutional Neural Network. **(E)** Two steps used for the analysis of USV repertoire, taking into account the probabilistic distribution of USV classes provided by the Convolutional Neural Network. Numbers in italic next to boxes indicate the respective methods section where the processes are described.

2.3.1 Spectrogram and power spectral density calculation

USVs were segmented on the audio files by analysis of their spectrograms. Aiming the configuration that would grant us the best resolution for the spectrograms, the spectrograms were calculated through a short-time Fourier transformation (STFT) using the following parameters: 1024 sampling points to calculate the discrete Fourier transform (NFFT = 1024), Hamming window with length 256 and half-overlapping with adjacent windows. The mathematical expression that gives us the spectral power is shown below:

$$STFT\{x[n]\}(m,\omega) = X(n,\omega) = \sum_{n=-\infty}^{\infty} = x[n]w[n-m]e^{-j\omega n} \tag{2.1}$$

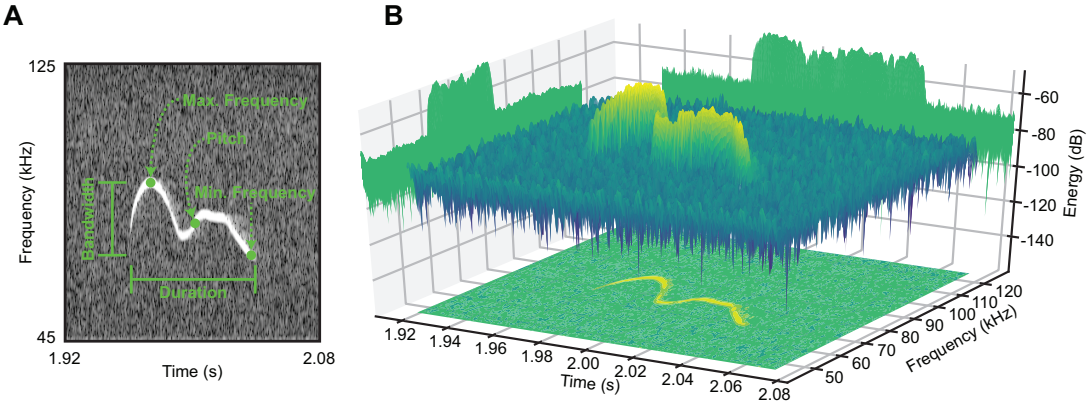
where $x[n]$ is the original signal and the window function $w[n]$, which is nonzero only for a short period of time. By this process, the original signal is divided into chunks that overlap with their neighbours in order to reduce artifacts at the boundary. Each chunk is Fourier transformed, and the complex result is added to a matrix, which records magnitude (m) and phase (ω) for each point in time and frequency. The resulting phase (ω) in a short-Fourier transform is continuous, but since computers compute the STFT using Fast Fourier transform, both variables are discrete and quantified.

The spectral power is then given by

$$P(m,\omega) = 10\log\left|\sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega n}\right|^2 \tag{2.2}$$

The resulting spectrogram consists on 3D matrices, in which each element corresponds to a time stamp, frequency and intensity (power spectrum represented as decibels) in space (Figure 2.2). The spectrogram is then analyzed in terms of its time and frequency plane, in which the intensity is represented by the brightness in a gray-scale image (Figure 2.2). We used a high pass filter (45 kHz) to eliminate sources of noise in the audible range and to reduce the amount of data stored (GRIMSLEY; MONAGHAN; WENSTRUP, 2011).

Figure 2.2: 3D and 2D representations of an USV



Source: The author

Figure 2.2: **(A)** Single USV segmented in a gray scale image with its spectral features indicated. **(B)** Illustrative example of the 3D properties of an USV in a spectrogram.

2.3.2 Normalization and contrast enhancement

Since USVs present higher intensity than the background and to avoid setting a fixed threshold for USV segmentation, we used contrast adjustment to highlight putative USV candidates and to reduce the variability across audio files. Contrast adjustment was obtained using the following equation for a corrected image:

$$J = \left(\frac{\frac{|10log(P)|}{max(10log(P))} - L_{in}}{H_{in} - L_{in}}\right)^{\gamma} \tag{2.3}$$

where H_{in} and L_{in} are the highest and the lowest intensity values of the adjusted image, respectively, and P is the power spectrum for each time-frequency point (pixel of the image). The parameter gamma shapes the curve describing the relationship between the values in the original image and the adjusted image, in such way that if gamma is less than 1, the mapping is weighted toward higher (brighter) output values. If gamma is greater than 1, the mapping is weighted towards lower (darker) output values. This re-scaling of the intensity values in the original gray-scale image to new values in the adjusted image will be such that 1% of data is saturated at low and high intensities of the original image. The gamma used for our application was $\gamma = 1$.

2.3.3 Adaptive thresholding and morphological operations

Due to non-stationary background noise and dynamic changes on the intensity of USVs within and between audio files, we use adaptive thresholding methods to binarize the spectograms. The threshold is computed for each pixel using the local mean intensity around the neighborhood of the pixel (BRADLEY; ROTH, 2007). This method preserves hard contrast lines and ignores soft gradient changes. The integral image consists of a matrix $I(x, y)$ that stores the sum of all pixel intensities $f(x, y)$ to the left and above the pixel (x, y). The computation is given by the following equation:

$$I(x, y) = f(x, y) + I(x - 1, y) + I(x, y - 1) - I(x - 1, y - 1) \tag{2.4}$$

Therefore, the sum of the pixels values for any rectangle defined by a lower right corner (x_2, y_2) and upper left corner (x_1, y_1) is given as:

$$\sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} f(x, y) = I(x_2, y_2) - I(x_2, y_1 - 1) + I(x_1 - 1, y_2) - I(x_1 - 1, y_1 - 1) \tag{2.5}$$

Then, the method computes the average of an $s \times s$ window of pixels centered around each pixel. The average is calculated considering neighbouring pixels on all sides for each pixel. If the value of the current pixel is t percent less than this average, then it

is set to black, otherwise it is set to white, as shown in the following equation:

$$C(x,y) = \frac{1}{(y_2 - y_1)(x_2 - x_1)} . \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} f(x,y) \tag{2.6}$$

where $C(x,y)$ represents the average around the pixel (x,y).

The binarized image is then constructed such as that pixels (x,y) with intensity t percent lower than $C(x,y)$ are set to black (BRADLEY; ROTH, 2007):

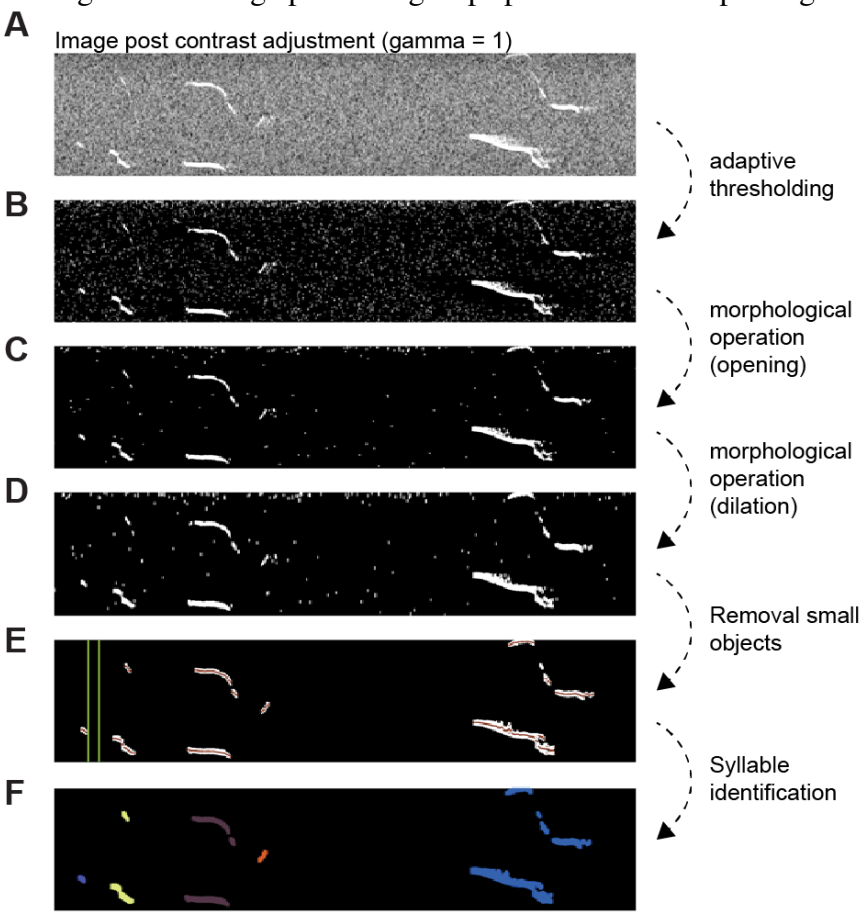
$$B(x,y) = \begin{cases} 0, & \text{if } f(x,y) \leq (1-t)C(x,y) \\ 1, & \text{otherwise} \end{cases} \tag{2.7}$$

where t represents the sensitivity factor and it was empirically chosen as $t = 0.2$ for our application. The segments are then subjected to a sequence of morphological operations: (i) opening (erosion followed by a dilation) with a rectangle 4 x 2 pixels as kernel; (ii) dilation with a line of length l = 4 and $\angle$ 90° relative to the horizontal axis as kernel; (iii) filtering out blobs (i.e., dense set of white pixels) with $<$ 60 pixels (correspondent to approximately 2 ms syllable); and (iv) dilation with a line of length l = 4 and $\angle$ 0°, making blobs proportional to their original shape (Figure 2.3A-E).

2.4 Listing USV candidates

The output of the segmentation process returns a list of blobs that are candidates to be an USV. However, not necessarily all blobs correspond to a single USV. This list of detected blobs can contain noise (i.e., blobs that are not part of any USV) and blobs that belong to the same USV. Therefore, the next step of the algorithm aggregates detected blobs that belong to the same USV. A minimum 10 ms interval between two successive syllables is assumed (CHABOUT et al., 2015). In order to reduce the amount of data while keeping relevant information for each USV, the features extracted from detected blobs are represented by a mean frequency every 0.5 ms. Means are calculated for all the individual blobs, including the ones overlapping in time (harmonic components).

Figure 2.3: Image processing steps performed on a spectrogram



Source: The author

Figure: 2.3: **(A)** Segment of a spectrogram post contrast adjustment ($\gamma = 1$). **(B)** Output image post binarization using adaptive thresholding. **(C)** Resulting image from the opening operation with rectangle 4x2. **(D)** Result from the dilation with line l=4 and $\angle$ 90°. **(E)** Removing too small objects ($\leq$ 60 pixels) and mean of cloud points detected for each object (white blob) being shown in red and green lines shows an interval of 10 ms. **(F)** Result after separating syllables based on the criterion of maximum interval between two tones in a syllable. The different colors differentiate the syllables from each other.

2.4.1 Detection of harmonics

Harmonic components are also referred as nonlinear components or composite (SCATTONI; RICCERI; CRAWLEY, 2011; SCATTONI et al., 2008). Here, we did not consider harmonic components as a different syllable, but rather as an extra feature of a syllable (GRIMSLEY; MONAGHAN; WENSTRUP, 2011). Therefore, each USV may or may not present a harmonic component. A harmonic component was considered as a continuous blob (with no jumps in frequency) overlapping in time with the main component of the USV (similar to (GRIMSLEY; MONAGHAN; WENSTRUP, 2011)).

2.5 Eliminating noise

The process of detecting USV candidates through a series of image processing steps might introduce noise. Additionally, other sources of external noise can influence the rate of detected USV candidates. We used two methods to eliminate noise from the pool of USV candidates: Local Median Filtering and Convolutional Neural Network.

2.5.1 Local Median Filtering

Noise due to the segmentation process is in the form of pixels or aggregate of pixels that are not associated to an event in the recording (a real USV or external noise) and are part of the pool of USV candidates. In order to determine if an USV candidate is relevant for further analysis, we perform a test - Local Median Filtering - to compare the intensity of the pixels in the blob attributed to USV candidate k (from now on referred as X_k) to the intensity of the pixels in a window that contains the blob (referred as W_k).

The bounding box that defines this window is a rectangle with its four vertices defined as a function of the frequencies (F_k) for USV candidate k and its time stamps (T_k). Thus, the bounding box is defined as follows:

$$W_k = \begin{cases} (max(F_k) + 2.5)kHz, \\ (min(F_k) - 2.5)kHz, \\ (max(T_k) + 0.1)s, \\ (min(T_k) - 0.1)s \end{cases} \tag{2.8}$$

As seen in Equation 2.8, a 200 ms interval is analyzed around the USV candidate. Such a wide interval may present more than one USV in W_k. However, the amount of pixels in X_k represents only 2.43 $\pm$ 0.10 % (mean $\pm$ SEM; median = 1.27, 95% CI [2.22 , 2.63]; n = 59,781 images analyzed) of the total number of pixels contained in the window W_k. Given this proportion between the number of pixels in X_k and W_k, the median of the intensity distribution of the whole window W_k (referred as $\widehat{W_k}$) tends to converge to the median intensity of the background.

We used the ratio $\widehat{X_k}/\widehat{W_k}$ to exclude USV candidates that correspond to segmentation noise. We first calculated the cumulative distribution function (CDF) for each set of USV candidates (now referred as Υ). To find the inflection point in Υ, a second order

polynomial fit for every set of 3 consecutive points was used to obtain local parametric equations ($\Upsilon(t) = (x(t), y(t))$) describing the segments of Υ. Since the calculation of the inflection point is done numerically, the number of points chosen for this calculation should be such that we can have as many points of curvature as possible while preserving information of local curvature. Then, after a screening for the best number of points, Υ was down-sampled to 35 equally spaced points and the inflection point was calculated. Using the local parametric equations, we calculated the tangent and normal vectors on each of the 35 points. Using these vectors, we estimated the changing rate of the tangent towards the normal at each point, which is the curvature κ (O'NEILL, 2006) and can be calculated as follows:

$$\kappa = \frac{\det\left(\Upsilon', \Upsilon''\right)}{\|\Upsilon'\|^3} \tag{2.9}$$

or by using the parametric equations:

$$\kappa = \frac{x'y'' - x''y'}{(x^2 + y^2)^{3/2}} \tag{2.10}$$

The inflection point is then determined as the point with maximum curvature and adopted as threshold τ for all the USV candidates in the audio file. This threshold is calculated individually for each audio file since it can vary according to the microphone gain and the distance of the microphone from the sound source (see below). In audio files with very low number of USVs, τ was not detected. In these cases, a default threshold $\tau = 0.92$ was adopted as a conservative threshold, since no audio file presented inflection point as high as 0.92 in our *training* set.

Candidates satisfying Equation 2.11 are kept for the following steps of analysis:

$$\left\{X_k \in \chi | \widehat{X_k} \leq \tau \widehat{W_k}\right\} \tag{2.11}$$

where χ represents the set of USV candidates that survived the Local Median Filtering. Of note, the intensity of each pixel is calculated in decibels, which is given in negative units due to the low power spectrum.

2.5.2 Influence of the microphone gain on the threshold τ

In order to estimate the energy or intensity of a vocalization candidate k, we convert the power spectrum P obtained by a short-time Fourier transformation (STFT) of the audio recording (see Section 2.3) to decibels. This conversion consists on evaluating the power as ten times the base-10 logarithm of the measured power.

We compare the intensity of a vocalization candidate $\widehat{X_k}$ to its background $\widehat{W_k}$ by evaluating the ratio of the two quantities, which could also be expressed in its logarithmic form:

$$\frac{\widehat{X_k}}{\widehat{W_k}} = \frac{10\log_{10}(\widehat{P_x})}{10\log_{10}(\widehat{P_w})} \tag{2.12}$$

where $\widehat{P_x}$ is the mean power spectrum for points detected as part of a vocalization candidate and $\widehat{P_w}$ is the mean for the background surrounding the vocalization candidate k.

Assuming the gain as a constant α modulating the power spectrum, we will have the following expression:

$$\frac{\widehat{X_k}}{\widehat{W_k}} = \frac{\log_{10}(\alpha\widehat{P_x})}{\log_{10}(\alpha\widehat{P_w})} = \frac{\log_{10}(\alpha) + \log_{10}(\widehat{P_x})}{\log_{10}(\alpha) + \log_{10}(\widehat{P_w})} \tag{2.13}$$

where a gain $\alpha = 1$ would represent maximum gain and $\alpha = 0$ represents a microphone with no gain at all. We can appreciate that as the gain approaches zero, the ratio $\widehat{X_k}/\widehat{W_k}$ tends to 1.

$$\frac{\widehat{X_k}}{\widehat{W_k}} = \lim_{\alpha\to 0} \frac{\log_{10}(\alpha) + \log_{10}(\widehat{P_x})}{\log_{10}(\alpha) + \log_{10}(\widehat{P_w})} = 1 \tag{2.14}$$

This shows a progressive shift of the ratio $\widehat{X_k}/\widehat{W_k}$ towards the unit as we reduce the gain, resulting in a shift of the cumulative distribution function of the ratio $\widehat{X_k}/\widehat{W_k}$ and, consequently, the threshold τ towards 1.

One important implication from this deduction to our work is that τ must necessarily move to values closer to 1 as the gain is reduced. Therefore, any τ lower than the correspondent τ for maximum gain ($\tau = 0.9$) can be treated as a miscalculation and the default $\tau = 0.92$ should be assumed.

2.5.3 Convolutional Neural Network

We use Convolutional Neural Networks (CNN) to eliminate external noise from the pool of USV candidates and later to separate those USVs in distinct classes (see below). We imported the layers from AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) and customized in order to handle spectrograms. Briefly, the last three layers of the pre-trained network were replaced in order to handle a classification for two classes (*USV* and *noise*) or twelve classes (eleven *USV types* + *noise*) (see more details below). The network with new layers was then retrained to learn the classification task for our data set. The main components of the network are:

Convolutional layer: performs feature extraction by computing the output of neurons that are connected to small regions in the input volume (usually referred as receptive field). The receptive field connects to each neuron with weights obtained via back-propagation training. The weights are equal within a feature map (also referred as activation map or kernel). However, different feature maps within the same convolutional layer have different weights, allowing multiple features to be extracted at each location (LECUN et al., 1998).

Rectified linear unit (ReLU): applies an element-wise activation function, $f(x) = max(0, x)$. This function is composed of two linear segments, thresholding any negative values to 0 and preserving any positive value. This non-linearity is faster to compute than other activation functions and eliminates the need for contrast-normalization or any other data pre-processing to avoid saturation (NAIR; HINTON, 2010; JARRETT et al., 2009; HINTON et al., 2012).

Max pooling: performs a down sampling operation along the spatial dimensions (width, height) by propagating the maximum value within a receptive field to the next layer. The output is invariant to shifts within the field (HUANG et al., 2007).

Fully-connected layer: interprets the features extracted throughout the convolutional and pooling layers. Each neuron in this layer is connected to all the nodes in the previous volume (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Dropout: temporarily drops a random chosen unit and all its connections from the network during training, forming a different network with the remaining neurons. It reduces the dependence on particular units and chances of over-fitting by the network (HINTON et al., 2012).

Cross channel normalization: this local response normalization implements a form of lateral inhibition that creates a competition among the neurons for the most prominent activity peak. This normalization is performed over a neighborhood and boosts neurons with larger activation than their neighbors. The output of this process implements what could be understood as brightness normalization. (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

The last layer of the stack is Fully connected Softmax layer, which is a classifier based on cross-entropy loss. The Softmax classifier gets its name from the Softmax function, which is used by the Classification Output layer to normalize the output from the fully connected layer into a probability distribution consisting of k probabilities that sum to one. The Classification Output layer is the responsible for assigning labels. For multi-class classification problems, the software assigns each input to one of the k mutually exclusive classes. The loss (error) function comes from information theory, and for this case is the cross entropy function for a 1-of-k coding scheme:

$$E(\theta) = -\sum_{i=1}^{n}\sum_{j=1}^{k} t_{ij} \ln\left(y_j(x_i,\theta)\right) \tag{2.15}$$

where θ is the parameter vector, t_{ij} is the indicator that the i^{th} sample belongs to the j^{th} class, and $y_j(x_i,\theta)$ is the output for sample i. The output $y_j(x_i,\theta)$ can be interpreted as the probability that the network associates i^{th} input with class j, that is, $P(t_j = 1|x_i)$. In other words, we are minimizing the cross-entropy between the estimated class probabilities $y_j(x_i,\theta)$ and the "true" distribution t_{ij}, which in this interpretation is the distribution where all probability mass is on the correct class. The output unit activation function is the softmax function:

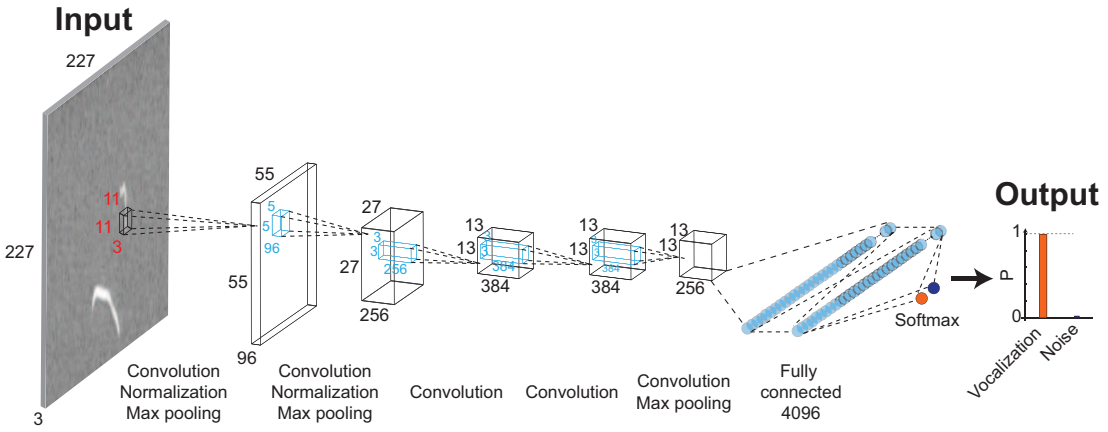
$$y_j(x,\theta) = \frac{\exp\left(a_j(x)\right)}{\sum_{i=1}^{k}\exp\left(a_i(x)\right)} \tag{2.16}$$

where $0 \leq y_j \leq 1$, $\sum_{i=1}^{k} y_i = 1$ and a_j is the activation of the j^{th} unit (class) in the last layer.

The outputs of the segmentation process with detected USV candidates were centralized in windows of 220 ms. These windows were twice the maximum duration of USVs observed in mice (GRIMSLEY; MONAGHAN; WENSTRUP, 2011) and were framed in individual 227 x 227 pixels images. Each image was then manually labeled by an experienced experimenter as noise (including acoustic or segmentation noise) or

real USV. This labeled dataset was used to train the CNN to classify the USV candidates in noise or USV (Figure 2.4).

Figure 2.4: Diagram of the Convolutional Neural Network used to remove noise from the pool of USV candidates



Source: The author

Figure 2.4: Representation of the AlexNet architecture post-learning transfer. In this process, the pre-trained network was used as starting point and the last three layers were replaced by new fully connected layers with size two (corresponding to classes *USV* and *noise*, as illustrated on the figure). The network with new layers was then retrained to learn the classification task for our images.

Our data set consisted of 12,120 images, in which 2,024 were manually labeled as noise. This dataset corresponds to mice of different strains (C57Bl6/J, NZO/HlLtJ, 129S1/SvImJ, NOD/ShiLtJ, and PWK/PhJ) and ages (5, 10, and 15 days of age) for both genders. To validate the training performance, the data set was split into two disjoint sets; *training* set (90%) and a *validation* set (10%).

The CNN was trained using stochastic gradient descent with a minimum batch size of $M = 128$ images and the number of iterations per epoch was given as $\lfloor \frac{N}{M} \rfloor$, where N is the size of the training samples. The maximum number of epochs was set to 100. Through a screening process for the set of hyper-parameters that would maximize the average performance of the network, the chosen learning rate was $\alpha = 10^{-4}$, momentum of 0.9 and weight decay $\lambda = 10^{-4}$. The training and validation data were shuffled at every epoch during training. The training was set to stop when the classification accuracy on the validation set did not improve for 3 validations in a row. When running in a GeForce GTX 980 TI, the training time was approximate 30 ± 10 min. The accuracy reached in the training process was 96.74% after 10 minutes of training and reached the stopping criteria with accuracy 99.01% after 40 minutes. We did not use unsupervised pre-training.

It is worth to note that other statistical methods were also tested for this USV/noise

classification task, such as Hierarchical clustering (Section 5.2.1) and Random Forest (Section 5.2.2). These methods are no longer used in the main structure of VocalMat due to their high dependence on the result of the segmentation steps, which is often not optimal.

2.5.4 Testing detection performance

In order to evaluate the performance of VocalMat, neonatal mice were recorded during 10 minutes upon social isolation in different conditions (Table 2.5.4). The spectrograms were manually inspected for the occurrence of USVs. The starting time for the detected USVs was recorded. USVs automatically detected by VocalMat with a start time matching manual annotation (± 5 ms of tolerance) were considered correctly detected. USVs manually detected with no correspondent USV given by VocalMat were considered *false negative*. The false negatives originated from missed USVs or USVs that the software labeled as noise. Finally, USVs registered by VocalMat without a correspondent in the manual annotation were considered *false positive* (see Table 2.3). In order to compare VocalMat to the other tools available, the same metrics were applied to the output of Ax (NEUNUEBEL et al., 2015), MUPET (SEGBROECK et al., 2017) and DeepSqueak (COFFEY; MARX; NEUMAIER, 2019).

2.6 Classification of USVs

We used a similar CNN as described for noise elimination as a method for USV classification, which results in a probability distribution across USV classes rather than a single label.

2.6.1 CNN for USV classification

In order to set a reference for the classification CNN, we adapted the definition of USV types given by Scattoni (SCATTONI et al., 2008) and Grimsley (GRIMSLEY; MONAGHAN; WENSTRUP, 2011), except the types that do not correspond to ultrasound range. The USV types used are illustrated in Figure 2.5 and described here:

Complex: 1-note syllables with two or more directional changes in frequency > 6 kHz. A total of 350 images were used for training.

Step up: 2-notes syllables in which the second element was ≥ 6 kHz higher from the preceding element and there was no more than 10 ms between steps. A total of 1,814 images were used for training.

Step down: 2-notes syllables in which the second element was ≥ 6 kHz lower from the preceding element and there was no more than 10 ms between steps. A total of 389 images were used for training.

Two steps: 3-notes syllables, in which the second element was ≥ 6 kHz or more different from the first, the third element was ≥ 6 kHz or more different from the second and there was no more than 10 ms between elements. A total of 701 images were used for training.

Multiple steps: 4-notes syllables or more, in which each element was ≥ 6 kHz or more different from the previous one and there was no more than 10 ms between elements. A total of 74 images were used for training.

Up-frequency modulation: Upwardly frequency modulated with a frequency change ≥ 6 kHz. A total of 1,191 images were used for training.

Down-frequency modulation: Downwardly frequency modulated with a frequency change ≥ 6 kHz. A total of 1,775 images were used for training.

Flat: Constant frequency syllables with modulation ≤ 5 kHz. A total of 1,135 images were used for training.

Short: Constant frequency syllables with modulation $\leq$ 5 kHz and duration $\leq$ 12 ms. A total of 1,713 images were used for training.

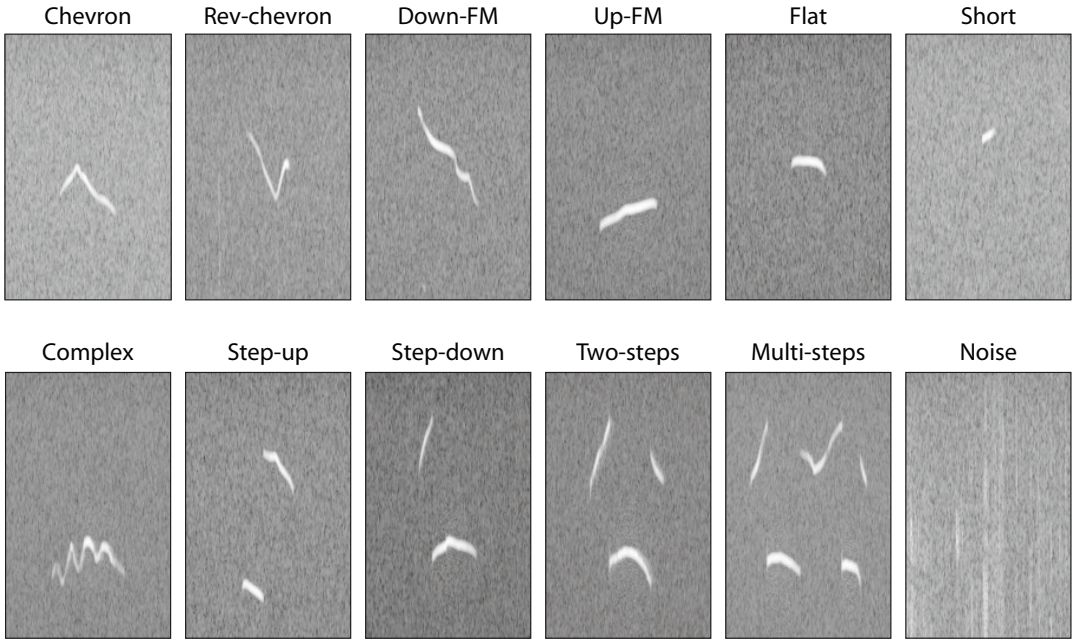
Chevron: Shaped like an inverted *U* in which the peak frequency was $\geq$ 6 kHz than the starting and ending frequencies. A total of 1,594 images were used for training.

Reverse chevron: Shaped like an *U* in which the peak frequency was $\geq$ 6 kHz than the starting and ending frequencies. A total of 136 images were used for training.

Noise: Any sort of mechanical or segmentation noise detected during the segmentation process as an USV candidate. A total of 2,083 images were used for training.

In order to purposely create some overlap between the categories, USV with segments oscillating between 5 and 6 kHz were not defined or used for training. The assumption is that the CNN should find its own transition method between two overlapping categories.

Figure 2.5: Representative images of each USV type used in the classification Convolutional Neural Network.



Source: The author

Figure 2.5: Images containing USVs candidates as output of the segmentation process were labeled in twelve categories according to the call types defined above.

The training set was built by manually classifying the 12,955 USVs recorded from mice of different strains (C57Bl6/J, NZO/HlLtJ, 129S1/SvImJ, NOD/ShiLtJ, and PWK/PhJ) and ages (5, 10, and 15 days of age) for both genders. To validate the performance of the algorithm, 10% of the samples were used. The training parameters were the

same as described above for the CNN used to eliminate noise. The final validation accuracy was 95.28% after 17 minutes of training. We did not use unsupervised pre-training.

2.6.2 Testing classification performance

In order to evaluate VocalMat's performance in classifying USVs, USVs emitted from neonatal mice (Table 2.5.4) were manually inspected. VocalMat exports spectrograms containing each detected USV candidate. These spectrograms were manually labeled by a trained experimenter based on the definitions of USV categories (see above). The experimenter only assigned the most likely label for each USV candidate. The labels given by the experimenter were then compared to the labels given by the CNN and. Cases where there was a match were considered as a *correct label*. To further evaluate the classification performance, we also considered as *correct label* up to the second most likely label given the CNN, as indicated in the text.

2.7 Data analysis

2.7.1 Diffusion maps for output visualization

The main characteristic of VocalMat is the possibility of classifying USVs as a distribution of probabilities over all the possible labels. Since we classify USV candidates in 12 categories, to have access to the distribution of probabilities, we would need to visualize the data in 12 dimensions. Here, as an example of analytical methods that can be applied to the output data from VocalMat, we used *Diffusion Maps* (COIFMAN et al., 2005) to reduce the dimensionality of the data to three dimensions. Diffusion Maps allows a remapping of the data into an Euclidean space, which ultimately results on a clustering of USVs based on the similarity of their probability distribution. The connectivity between two data points in a Euclidean manifold is defined by a Gaussian kernel function. Such kernel provides the similarity value between two data points i and j as follows:

$$W_{ij} = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{2.17}$$

where W_{ij} represents the similarity value between observations i and j. The parameter σ corresponds to the bandwidth and it is set based on the average Euclidean distance observed between observations of the same label. For our application, $\sigma = 0.5$ was set based on the distance distribution observed in our data.

The similarity matrix is then turned into a probability matrix by normalizing the rows:

$$p(j|i) = \frac{W_{ij}}{\sum_k W_{ik}} = D^{-1}W = M_{ij} \tag{2.18}$$

where $\sum_k W_{ik} = D_{ii}$ has the row sum of W along its diagonal. The matrix M gives the probability of walking from node i to any other node. In other words, the probability that the USV i is close to another USV given their probability distribution.

Once we take one step in such Euclidean space, the probabilities are updated, since the set of likely nodes for the next move are now updated. This idea of moving from node to node while updating the probabilities, gives us a "diffused map". This process of taking steps is represented by:

$$p(t, j|i) = e_i^T M^t e_j \tag{2.19}$$

which is the probability of reaching j from i after t steps in this map. For our application, we use $t = 2$.

Next, we find the coordinate functions to embed the data in a lower dimensional space. Such result is given by the eigenvectors of M. Although M is not symmetric (its rows were normalized in Equation 2.18), the eigendecomposition can still be calculated through the SVD decomposition (GOLUB; KAHAN, 1965):

$$M_s = D^{1/2}MD^{-1/2} = D^{1/2}D^{-1}WD^{-1/2} = D^{-1/2}WD^{-1/2} \tag{2.20}$$

and since $D^{-1/2}$ and W are symmetric, M_s is also symmetric and allow us to calculate its eigenvectors and eigenvalues. For the sake of notation, let's consider:

$$M_s = \Omega\Lambda\Omega^T \implies M = D^{-1/2}\Omega\Lambda\Omega^T D^{1/2} \tag{2.21}$$

The eigendecomposition of M is easily visualized if we rewrite Equation 2.21 by properly identifying the eigenvectors. So let's set $\Psi = D^{-1/2}\Omega$ (right eigenvectors of M) and $\Phi = D^{1/2}\Omega$ (left eigenvectors of M). As we can see, $\Phi^T = \Psi^{-1}$, therefore they are

mutually orthogonal and M and M_s are similar matrices. Thus,

$$M = \Psi\Lambda\Psi^{-1} = \Psi\Lambda\Psi^{T} \tag{2.22}$$

and the diffusion component shown in Equation 2.19 is incorporated as the power of the diagonal matrix composed by the eigenvalues of M:

$$M^{t} = \Psi\Lambda^{t}\Phi^{T} \tag{2.23}$$

Here we use the scaled right eigenvectors by their corresponding eigenvalues ($\Gamma = \Psi\Lambda$) as the coordinate functions. Since the first column of Γ is constant across all the observations, we use the 2nd to 4th coordinates in our work.

2.7.2 Repertoire analysis via Manifold Alignment

The result of the embedding by Diffusion Maps allows 3D visualization of the probability distribution for the USVs. The direct comparison of different 3D maps is difficult to obtain as the manifolds depend on data distribution, which contains high variability in experimental samples. To address this problem and compare the topology of different manifolds, we considered this a transfer learning problem (PAN; YANG, 2010). We used a manifold alignment method for heterogeneous domain adaptation (WANG; MAHADEVAN, 2011; TUIA; CAMPS-VALLS, 2016). Using this method, two different domains are mapped to a new latent space, where samples with the same label are matched while preserving the topology of each domain.

We used the probability distribution for the USVs for each data set to build the manifolds (WANG; MAHADEVAN, 2011). Each manifold was represented as a Laplacian matrix constructed from a graph that defines the connectivity between the samples in the manifold. The Laplacian matrix is then defined as $L = W_{ij} - D_{ii}$ (see Equation 2.17).

The final goal is to remap all the domains to a new common space such that samples with similar labels become closer in this new space while samples with different labels are pushed away while preserving the geometry of the manifolds. It leads to the necessity of three different graph Laplacians: L_s (relative to the similarity matrix and responsible for connecting the samples with same label), L_d (dissimilarity matrix and responsible for connecting the samples with different labels), and L (similarity matrix responsible for preserving the topology of each domain). Wang and Mahadevan (WANG;

MAHADEVAN, 2011) show that the embedding that minimizes the joint function defined by the similarity and dissimilarity matrices is given by the eigenvectors corresponding to the smallest non-zero eigenvalues of the following eigendecomposition:

$$Z(L + \mu L_s)Z^T V = \lambda Z L_d Z^T V \tag{2.24}$$

where Z is a block diagonal containing the data matrices $X_i \in \mathbb{R}^{d_i \times n_i}$, ($n_i$ samples and d_i dimensions for the i^{th} domain) from the two domains. Thus, $Z = diag(X_1, X_2)$. The matrix V contains the eigenvectors organized in rows for each domain, $V = [v_1, v_2]^T$. The μ is weight parameter, which goes from preserving both topology and instance matching equally ($\mu = 1$) or focus more on topology preservation ($\mu > 1$).

From Equation 2.24, We then extract $N_f = \sum_{i=1}^{D} d_i$ features, and the projection of the data to this new common space $\mathcal{F}$ will be given by

$$P_{\mathcal{F}}(X_i) = v_i^T X_i \tag{2.25}$$

In order to measure the performance of the alignment, linear discriminant analysis (LDA) (MCLACHLAN, 2004) is used to show the ability of projecting the domains in a joint space. The LDA is trained on half of the samples in order to predict the other half. The error of the alignment is given as the percentage of samples that would be misclassified when projected into the new space (overall accuracy).

Another measurement to quantify the quality of the alignment is by calculating the agreement between the projections, which is given by Cohen's Kappa coefficient (κ) (AGRESTI, 2018). In this method, the labels are treated as categorical and the coefficient compares the agreement with that expected if ratings were independent. Thus, disagreements for labels that are close are treated the same as labels that are far apart.

Cohen's coefficient is defined as:

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \tag{2.26}$$

where p_0 is the observed agreement ($p_0 = \sum_{i=1}^{k} p_{ii}$ for a confusion matrix $p = n/N$, in which n is the raw confusion matrix and N is the total number of samples, composed by the projection of the k labels), which corresponds to the accuracy; p_e is the probability of agreement by chance ($p_e = \frac{1}{N^2} \sum_{i=1}^{k} p_{i\cdot} p_{\cdot i}$, where $p_{i\cdot}$ is the number of times an entity of label i was labelled as any category and $p_{\cdot i}$ is the number of times any category was predicted as label i). Therefore, a $\kappa = 0$ represents no agreement (or total misalignment

of manifolds) and $\kappa = 1$ is a total agreement.

In this context, the overall accuracy (OA) is given by $OA = \sum_{i=1}^{k} p_{ii}/N$, where N is the total number of samples.

The asymptotic variance for κ is given as follows:

$$\hat{\sigma}^2(\hat{\kappa}) = \frac{1}{N}[\frac{\theta_1(1-\theta_1)}{(1-\theta_2)^2} + \frac{2\theta_1(1-\theta_1)(2\theta_1\theta_2-\theta_3)}{(1-\theta_2)^3} + \frac{(1-\theta_1)^2(\theta_4-4\theta_2^2)}{(1-\theta_2)^4}] \tag{2.27}$$

where

$$\theta_1 = \frac{1}{n}\sum_{i=1}^{k} n_{ii} \tag{2.28}$$

(which turns into accuracy once it is divided by N),

$$\theta_2 = \frac{1}{n^2}\sum_{i=1}^{k} n_{i.}n_{.i} \tag{2.29}$$

$$\theta_3 = \frac{1}{n^2}\sum_{i=1}^{k} n_{ii}(n_{i.} + n_{.i}) \tag{2.30}$$

$$\theta_4 = \frac{1}{n^3}\sum_{i=1}^{k}\sum_{j=1}^{k} n_{ij}(n_{j.} + n_{.i})^2 \tag{2.31}$$

From Equation 2.27 We can calculate the Z score, which can express the significance of our κ:

$$Z = \frac{\kappa}{\hat{\sigma}^2(\hat{\kappa})} \tag{2.32}$$

And the 95% confidence interval as

$$CI = [\kappa + 1.96\sqrt{\hat{\sigma}^2(\hat{\kappa})}, \kappa - 1.96\sqrt{\hat{\sigma}^2(\hat{\kappa})}] \tag{2.33}$$

A third form of error measurement is the evaluation of the projection per USV class from each domain remapped into the new space. This method is based on the fact that this new space is the one in which the cost function expressed by Equation 2.24 is minimized and, therefore, the projection from each domain into the new space has its own projection error for each class. As a consequence, the mean of the projection error from each domain to the new space for each class can be used as a quantitative measurement of misalignment of projected domains.

Table 2.1: Description of CNN architecture for vocalization identification

Layer	Operation	Number of units	Kernel Size	Stride	Padding
0	Input	-	227x227x3	-	-
1	Convolution RELU	96	11x11x3	4	-
2	Cross Channel Normalization	5 channels per element	-	-	-
3	Max Pooling	-	3x3	2	-
4	Convolution RELU	256	5x5x48	1	2
5	Cross Channel Normalization	5 channels per element	-	-	-
6	Max Pooling	-	3x3	2	-
7	Convolution RELU	384	3x3x256	1	1
8	Convolution RELU	384	3x3x192	1	1
9	Convolution RELU	256	3x3x192	1	1
10	Max Pooling	-	3x3	1	1
11	Fully connected ReLU	4096	-	-	-
12	Dropout	50%	-	-	-
13	Fully connected ReLU	4096	-	-	-
14	Dropout	50%	-	-	-
15	Fully connected Softmax	2 (USV+noise) or 12 (11 USV types + noise)	-	-	-

Source: The author

Table 2.2: Summary of experimental conditions covered in the test data set

Age	Microphone gain	Chamber	Heating
P9	Maximum	Yes	No
P9	Maximum	Yes	No
P9	Maximum	Yes	No
P10	Intermediary	No	No
P10	Intermediary	No	No
P10	Maximum	Yes	Yes
P10	Maximum	Yes	Yes

Source: The author

Summary of different conditions tested: 1) The age of the animals (given as days postnatal); 2) The gain of the microphone utilized for the recording; 3) If the experiment was performed inside a climate chamber, that could either provide an extra acoustic isolation from external noise or increase echoing.4) If there was any heating source turned on, which could possibly represent an increase in noise.

Table 2.3: Summary of possible outcomes for the detection validation

Manual	Automated	Actual meaning	Label
Detected	Detected	Success	True positive
Detected	Not detected	Missed or classified as noise	False negative
Not detected	Detected	Noise	False positive

Source: The author

Summary of manual validation: USVs automatically detected by VocalMat with a start time matching manual annotation ($\pm$5 ms of tolerance) were considered *True positive*. USVs manually detected with no correspondent USV given by VocalMat were considered *false negative*. The false negatives were associated to missed USVs or USVs that the software labeled as noise. Finally, USVs registered by VocalMat without a correspondent in the manual annotation were considered *false positive*

.

3 UTRASONIC VOCALIZATIONS DETECTION AND CNN CLASSIFICATION RESULTS

3.1 Detection of USVs

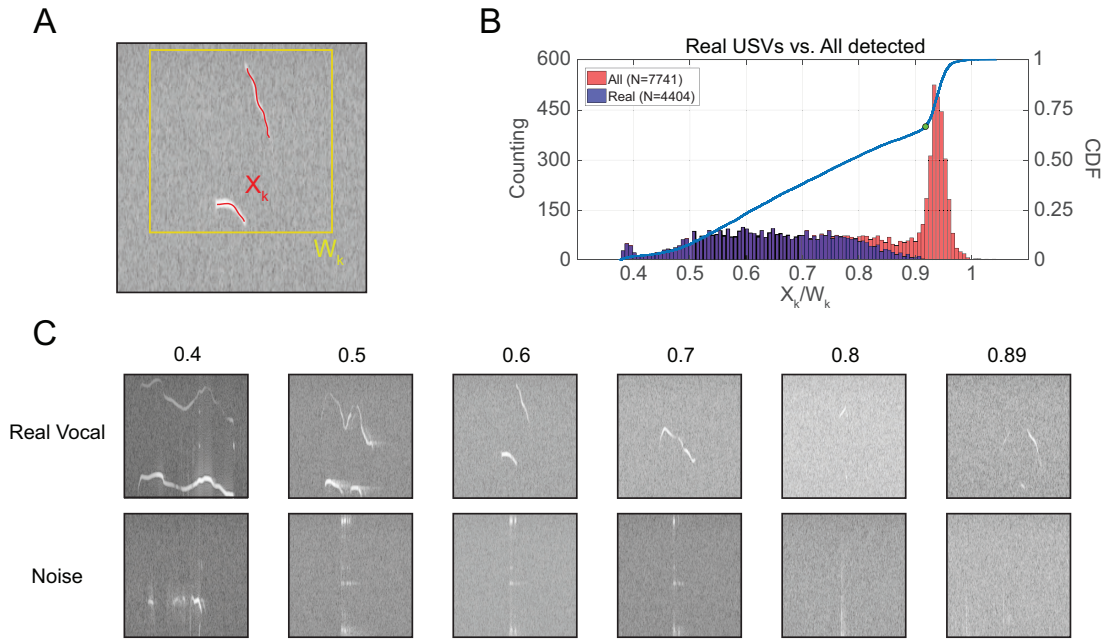
3.1.1 Detection of mouse USVs using imaging processing

Digitally recorded audio files were transformed into spectrograms and analyzed as gray scale images Figure 2.2A. Using imaging processing tools, we automatically segmented USV candidates in these gray scale images, preserving relevant features such as duration, frequency, intensity, and harmonic components (Figure 2.2A). These USV candidates (59,781 in the *training* data set, see Section 2.3) include real USVs and noise. However, visual inspection revealed that noise generated by the segmentation process dominated the pool of USV candidates. We noticed that this type of noise had very low intensity compared to real USVs. Thus, we reasoned that we could use the distribution of intensity of the USV candidates to eliminate large portion of segmented noise.

We first calculated the median intensity of the pixels in each detected USV candidate k, referred as $\widehat{X_k}$. We also calculated the median intensity of the background pixels in a bounding box surrounding the detected USV candidate, referred as $\widehat{W_k}$ (Figure 3.1A). We then calculated the ratio $\widehat{X_k}/\widehat{W_k}$ and its corresponding distribution (Figure 3.1B). In the distribution, the peak shown at high $\widehat{X_k}/\widehat{W_k}$ contained USV candidates of very low intensity. As the intensity of these USV candidates were similar to the intensity of the background, we reasoned that this peak corresponded to the noise generated by the segmentation process.

Next, we manually inspected the spectrograms and labeled USV candidates in a subset of audio files (hereafter, *test* dataset). Using VocalMat, a total of 7,741 USV candidates were detected, representing 1.76 times more USV candidates than real USVs (4,404 detected by VocalMat from 4,409 detected manually). The distribution of $\widehat{X_k}/\widehat{W_k}$ for real USVs and for noise confirmed the peak at high $\widehat{X_k}/\widehat{W_k}$ in the distribution was dominated by USV candidates that were noise (Figure 3.1B). The $\widehat{X_k}/\widehat{W_k}$ of real USVs (mean = 0.642, SEM = 1.841×10^{-3}, median = 0.640, 95% CI [0.638, 0.646]; N = 4,404) was significantly lower than the $\widehat{X_k}/\widehat{W_k}$ of noise (mean = 0.920, SEM = 9.605×10^{-4}, median = 0.935, 95% CI [0.920, 0.924]; N = 3,337; $P < 10^{-15}$, D = 0.894, Kolmogorov-Smirnov test; Figure 3.1B). These results suggest that a set value of $\widehat{X_k}/\widehat{W_k}$ (i.e., threshold) is

Figure 3.1: Applying Local Median Filtering and defining τ



Source: The author

Figure 3.1: **(A)** Example of a detected USV. The red dots indicate the points detected as part of the USV (X_k) and the rectangle in yellow indicates the neighborhood of the USV (W_k) used by the Local Median Filtering. **(B)** Distribution of the ratio $\widehat{X_k}/\widehat{W_k}$ for USV candidates manually identified (blue) and all the candidates (red). **(C)** Examples of real USVs and noise identified in different regions of the distribution $\widehat{X_k}/\widehat{W_k}$.

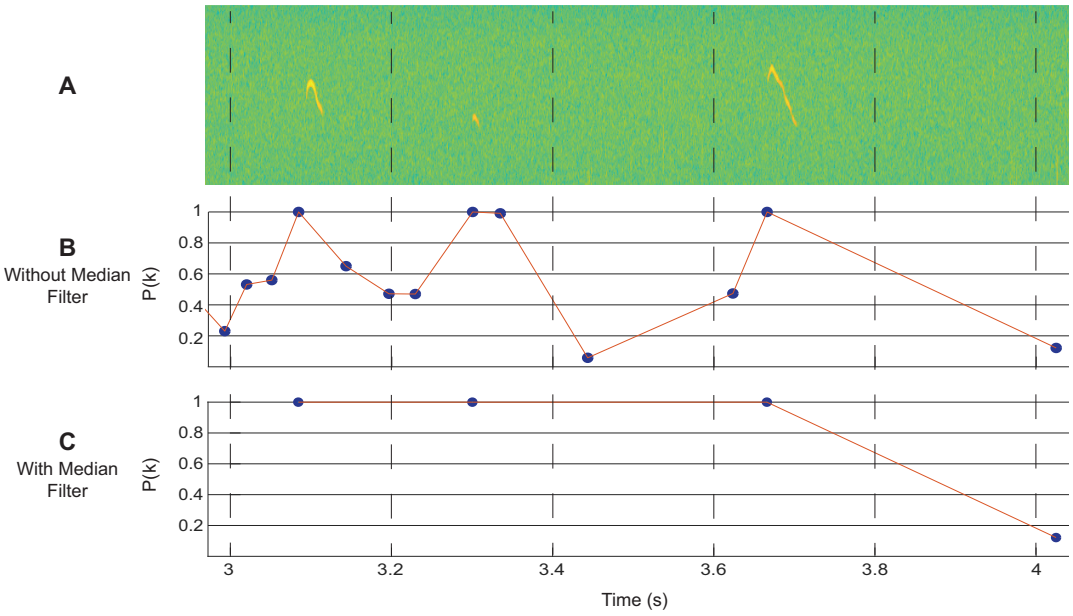
useful to eliminate a substantial amount noise from the pool of USV candidates before further analysis.

To determine a threshold value for $\widehat{X_k}/\widehat{W_k}$ (or τ) that could separate real USVs and noise in the distribution of $\widehat{X_k}/\widehat{W_k}$ without eliminating real USVs, we automatically calculated the inflection point in the cumulative distribution of $\widehat{X_k}/\widehat{W_k}$ (Figure 3.1B and Section 2.5.1). We used τ to effectively eliminate segmentation noise from the pool of USV candidates, which has proved itself useful to reduce the number of false positives reported by our tool. To illustrate its performance, Figure 3.2A shows a segment of a spectrogram with 3 visible USVs. The lack of a Local Median Filtering allows all the USV candidates to arrive to our classifier (Section 2.5.3), which assigns a probability P of being an USV greater than 0.5 for many candidates that do not correspond to real USV (Figure 3.2B). On the other hand, the removal of the segmentation noise by the Local Median Filtering alows only significant candidates to arrive to the classifier, resulting in a more reliable output (Figure 3.2C).

In the *test* data set, 5,171 out of 7,741 USV candidates survived this step. This number includes real USVs (4,397) and remaining noise of lower $\widehat{X_k}/\widehat{W_k}$. Of note, the

manual inspection of the spectrograms reveled that 7 USVs were eliminated in this step due to their high $\widehat{X_k}/\widehat{W_k}$ (mean = 0.941, SEM = 5.871×10^{-3}, median = 0.943, 95% CI [0.927, 0.956]; N = 7). The remaining noise in the pool of USV candidates was of high intensity and commonly originated from external sources. Thus, in a theoretical experimental setting with complete sound insulation and without the generation of noise by the movement of the animal, no further step should be required to identify real USVs using VocalMat. Since this is difficult in real experimental conditions, we used a second step in the noise elimination process.

Figure 3.2: Example of the effectiveness of the Local Median Filtering in reducing USV candidates



Source: The author

Figure 3.2: **(A)** Example of a segment of spectrogram with 3 USVs. **(B)** The analyze of this segment without Local Median Filtering results on an elevated number of false positives (noise detected as USV). The blue dot indicates the probability given by our classifier to having an USV starting at the point in time. **(C)** The result of the analyses of the same segment with the help of Local Median Filtering. The trivial candidates (noise segmentation) are no longer included and noise from external source is classified as having low probability of being an USV.

3.1.2 Eliminating noise using machine learning

Using convolutional neural networks (CNN), we trained VocalMat to identify noise and USVs using the *training* data set (see Section 2.3). We then validated the performance of VocalMat using the 5,171 USV candidates in the *test* data set that survived

the local median filtering step. The output of the CNN was the probability of each USV candidate been USV or noise. The highest probability defined the label of the candidate.

The rate of detected USVs labeled as such (true positives or sensitivity) was 99.09 $\pm$ 0.24% (mean $\pm$ SEM; median = 99.31; 95% CI [98.49, 99.69]). Using linear regression analysis between manually validated data and the true positives of the CNN revealed an almost-perfect linearity ($r^2 = 0.99$, 95% CI [0.95; 1.00]), $P < 10^{-4}$, and slope $\alpha = 0.97$), suggesting high accuracy of VocalMat in detecting USVs from audio files.

We further calculated other measures of performance. The rate of detected USVs labeled as noise (false negatives) was 0.64 $\pm$ 0.28% (mean $\pm$ SEM; median = 0.45; 95% CI [0; 1.33]). The rate of detected noise labeled as such (true negative rate or specificity) was 92.43 $\pm$ 1.00 % (mean $\pm$ SEM; median = 93.15; 95% CI [89.97; 94.88]). The rate of detected noise labeled as USV (false positive) was 7.57 $\pm$ 1.00% (mean $\pm$ SEM; median = 6.84; 95% CI [5.11; 10.03]), representing a total of 48 wrongly detected USVs out of the 5,171 USV candidates in the *test* data set. Finally, the rate of USVs not detected (missed rate) was 0.27 $\pm$ 0.09% (mean $\pm$ SEM; median = 0.22; 95% CI [0.04; 0.50). Overall, the measured accuracy for VocalMat was 98.30 $\pm$ 0.24% (mean $\pm$ SEM; median = 98.41; 95% CI [97.69; 98.91]) for the 7 audio files that were manually validated.

3.1.3 Performance of VocalMat compared to other tools

In order to evaluate the performance of VocalMat in detecting USVs compared to other published tools, we analyzed the same *test* data set with Ax (NEUNUEBEL et al., 2015), MUPET (SEGBROECK et al., 2017), and DeepSqueak (COFFEY; MARX; NEUMAIER, 2019).

Ax demands a series of manual inputs for their detection algorithm. We tried three different settings to get as close as possible to the number of USVs in the ground-truth (Table 3.1). For the configuration that minimized the number of missed USVs (3rd configuration), the percentage of missed USVs was 11.09 $\pm$ 1.38 % (mean $\pm$ SEM; median = 12.46, 95% CI [7.53, 14.65]). However, this configuration also generated the most number of false positives, detecting 132.5 $\pm$ 14.92 % (mean $\pm$ SEM; median = 142.0, 95% CI [94.13, 170.8]) more USVs than the ground-truth. On the other hand, the configuration that minimized the false positives (2nd configuration) showed a false positive rate of 81.22 $\pm$ 13.27 % (mean $\pm$ SEM; median = 82.12, 95% CI [47.12, 115.3]), while missing 30.85 $\pm$ 3.26 % (mean $\pm$ SEM; median = 33.00, 95% CI [22.46, 39.25]) of the USVs.

Ax does not separate the selected USV candidates in real USV or noise, therefore no false negative rate was calculated.

Table 3.1: List of parameters used for Ax

	Trial 1	Trial 2	Trial 3
FS	2.50E+05	2.50E+05	2.50E+05
NFFT	64	64	32
NW	6	6	6
K	11	11	11
PVAL	0.05	0.5	0.5
channels	-	-	-
frequency_low	4.50E+04	4.50E+04	4.50E+04
frequency_high	1.20E+05	1.20E+05	1.20E+05
convolution_size	[1300, 0.001]	[1300, 0.001]	[1300, 0.001]
minimum_object_area	18.75	18.75	18.75
merge_harmonics	1	1	1
merge_harmonics_overlap	0.9	0.9	0.9
merge_harmonics_ratio	0.1	0.1	0.1
merge_harmonics_fraction	0.9	0.9	0.9
minimum_vocalization_length	0	0	0

Source: The author

MUPET has a lower number of parameters to be set by the user. We tested eight different configurations of MUPET to measure its performance in detecting USVs in the validated *test* data set (Table 3.2). The configuration that showed better overall performance had a rate of missed USVs of 23.62 $\pm$ 5.54 % (mean $\pm$ SEM; median = 21.16, 95% CI [9.36, 37.87]), a rate of false positives of 9.26 $\pm$ 2.01 % (mean $\pm$ SEM; median = 8.20, 95% CI [4.08, 14.14]) and a rate of false negatives of 14.79 $\pm$ 3.99 % (mean $\pm$ SEM; median = 12.70, 95% CI [4.50, 25.06]). It's important to emphasize that these tests with Ax and MUPET did not explore the whole combination of parameters possible, implying that a better set of parameters could potentially optimize the detection task for our *test* data set.

Table 3.2: List of parameters used for MUPET

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8
noise-reduction	5	5	5	1	1	0.5	2	1
minimum-syllable-duration	2	2	2	2	2	2	2	2
maximum-syllable-duration	200	200	200	200	200	200	200	200
minimum-syllable-total-energy	-15	-15	-25	-25	-10	-25	-25	-35
minimum-syllable-peak-amplitude	-25	-25	-35	-35	-16	-35	-35	-45
minimum-syllable-distance	5	10	10	10	10	10	10	10

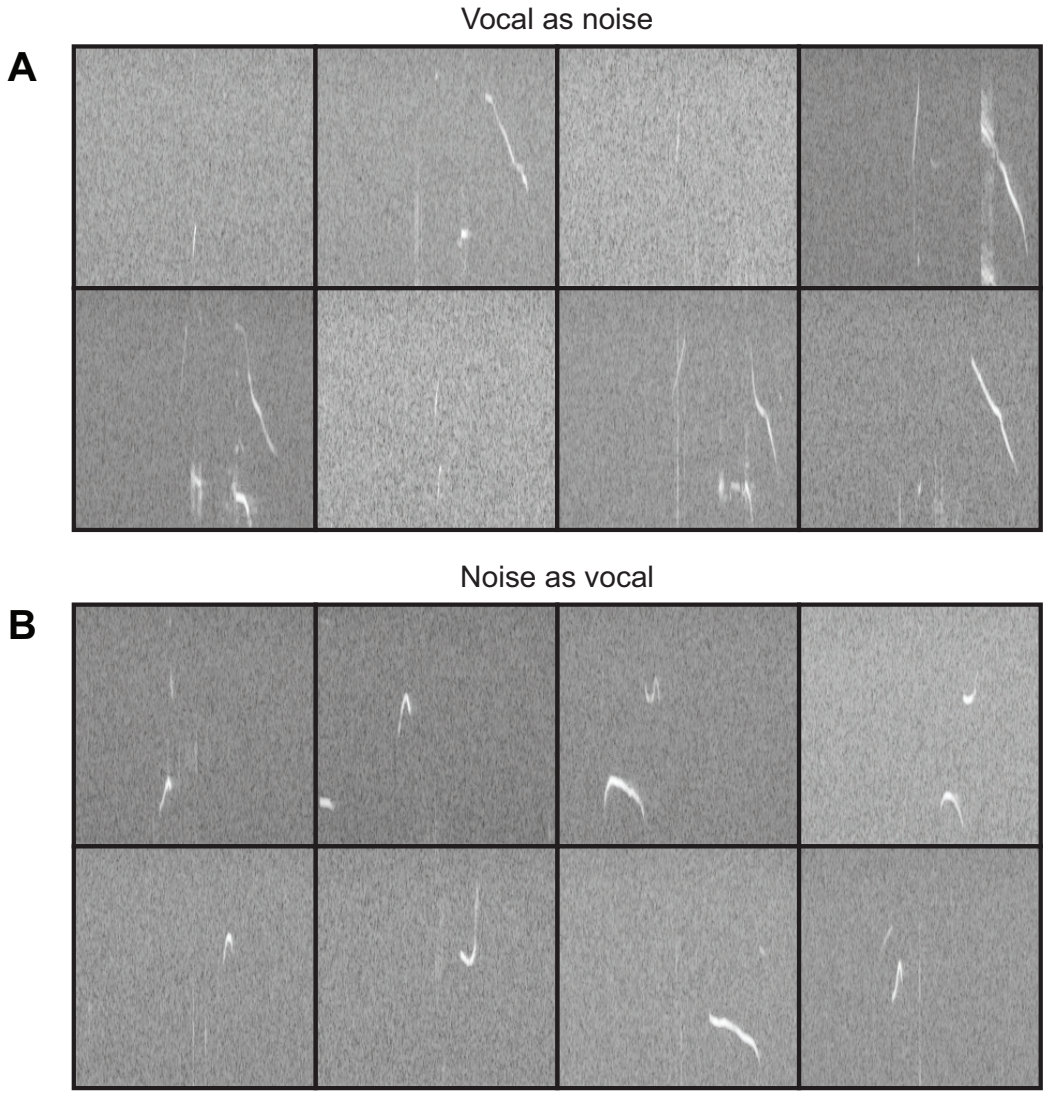
Source: The author

Differently from the previous tools, DeepSqueak does not demand manual setting of parameters for detection. We correlated USVs found by DeepSqueak with the time stamp of the USVs in the ground-truth. Because DeepSqueak is not formally trained to identify the start time of USVs with precision, we used increasing tolerance for mismatches in the starting time (± 5, ± 10, ± 15 and ± 20 ms). Using 5 ms mismatch, the rate of missed USVs by DeepSqueak was 41.11 $\pm$ 7.45 % (mean $\pm$ SEM; median = 35.72, 95% CI [21.94, 60.28]) and the rate of false positives was 21.24 $\pm$ 5.52 % (mean $\pm$ SEM; median = 17.21, 95% CI [7.02, 35.44]). With increasing tolerance (± 10, ± 15 and ± 20 ms), we observed a gradual decrease in the rate of missed USVs and in the rate of false positives. The best values obtained were a rate of missed USVs of 25.49 $\pm$ 4.04 % (mean $\pm$ SEM; median = 23.24, 95% CI [15.10, 35.88]) and a rate of false positives of 5.61 $\pm$ 2.01 % (mean $\pm$ SEM; median = 3.50, 95% CI [0.44, 10.79]). The manual inspection of the USVs detected by DeepSqueak revealed cases of more than one USV being counted as a single USV, which could lead to inflated number of missed USVs. Finally, as we did not train the network on our *test* data set, it is possible that DeepSqueak could present a much better performance than what we report here if custom-trained.

3.1.4 Characteristics of mislabeled USV candidates by VocalMat

USVs that were labeled as noise by VocalMat were most commonly overlapping in time with real noise or had spectral features in common with noise, such as fast change in frequency (Figure 3.3A). On the other hand, noise that was labeled as USV was most commonly due to segmentation noise occurring too close to a real USV or originated by an external source with spectral features different from the noise used for training the CNN (Figure 3.3B).

Figure 3.3: Example of mislabelling



Source: The author

Figure 3.3: Examples of mislabelling by VocalMat. In these examples, the most central object is the one being classified. In panel **(A)** we see examples of USVs being classified as noise, likely due to its sharp change in frequency in such short interval, resembling noise. In panel **(B)** we see examples of noise being classified as USVs, likely due to its proximity to real USVs.

For USVs wrongly labelled as noise (false negative), the probability of being noise was 0.80 $\pm$ 0.03 (mean $\pm$ SEM; median 0.81; 95% CI [0.73; 0.87]), while for noise labelled as USV (false positive), the probability of being USV was 0.79 $\pm$ 0.02 (mean $\pm$ SEM; median 0.81; 95% CI [0.75; 0.84])). These probabilities contrasted with cases in which VocalMat correctly identified USV and noise. USVs that were correctly identified had a probability of being USV of 0.99 $\pm$ 4.38 $\times$ 10^{-4} (mean $\pm$ SEM; median = 1.00; 95% CI [0.995; 0.997]). Noise that were correctly identified had a probability of been noise of

$0.96 \pm 3.00 \times 10^{-3}$ (mean ± SEM; median 0.99; 95% CI [0.957; 0.969]). These results indicate that the probability given by the CNN can be used to detect likely errors in classification. These *flagged* candidates could then be manually inspected by the investigator to correct the misclassification and train the machine.

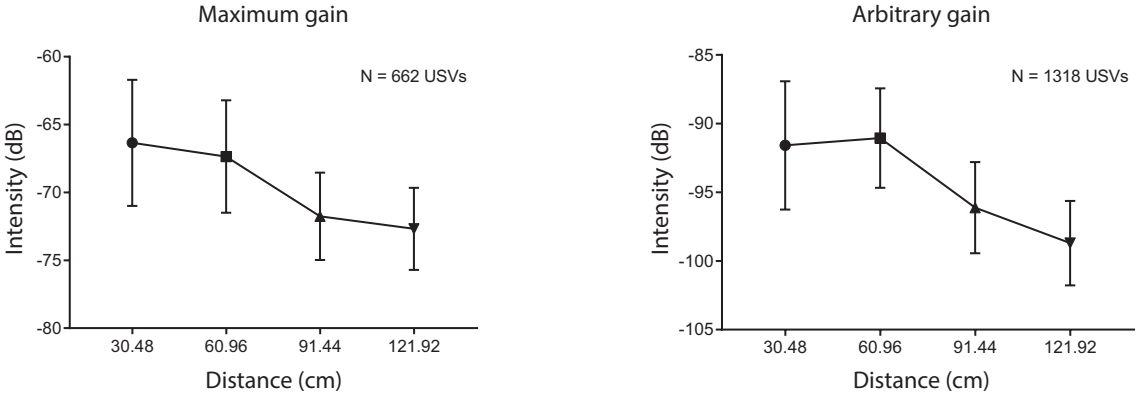
3.1.5 Detection of harmonic components

To measure the performance of VocalMat for detection of harmonic components, we compared the output of VocalMat with the *test* dataset. The rate of true positives was 93.32 ± 1.96 % (mean ± SEM; median = 92.18; 95% CI [88.54, 98.11]). The rate of USVs that were wrongly labelled as having a harmonic component (false positive) was 5.39 ± 1.18 % (mean ± SEM; median = 5.17; 95% CI [2.50, 8.27]). The rate of harmonic components that were not detected (false negative) was 6.68 ± 1.96 % (mean ± SEM; median = 7.82, 95% CI [1.89, 11.46]). All combined, the error rate in identifying harmonic components was 12.19 ± 3.44 % (mean ± SEM; median = 11.92, 95% CI [3.34, 21.03]).

3.1.6 Influence of the microphone's distance in the detection of USVs

USVs are attenuated with distance more than audible sounds in a non-linear fashion (VLADIŠAUSKAS; JAKEVIČIUS, 2004). This characteristic makes it difficult to predict the decay in USV intensity unless the decay is empirically tested in each experimental condition. Thus, we estimated the impact of the microphone's distance to the recording chamber on the performance of VocalMat to detect USVs. We performed a new set of simultaneous recordings with four microphones positioned at 30.48 cm (12"), 60.96 cm (24"), 91.44 cm (36"), and 121.92 cm (48") of distance from the recording chamber. The average loss in mean intensity was -0.09 ± 0.05 dB/cm (mean ± SD) for maximum microphone gain and -0.07 ± 0.05 dB/cm (mean ± SD) for half microphone gain (Figure 3.4).

Figure 3.4: Mean intensity as function of distance to the microphone



Source: The author

Figure 3.4: Graphical representation of the loss in intensity as function of the distance between the sound source and the microphone when set for maximum gain **(A)** and arbitrary/half gain **(B)**. Only USVs detected simultaneously by all 4 microphones were considered for these statistics.

3.2 Classification of USVs

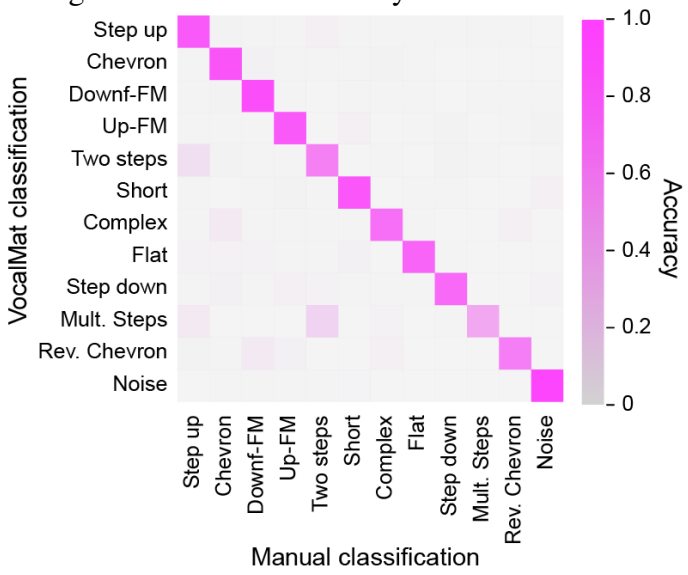
Additionally, we trained a second network to classify USVs into 11 categories plus noise, providing a probability for each class (see Section 2.6). First, we compared the most likely label assigned by the CNN to the labels assigned by the investigators (i.e., ground-truth). We considered a *correct* assignment by the CNN when both labels matched. The overall accuracy in the *test* data set was 86.05 %, with a Cohen's Kappa of $\kappa = 0.84$ (CI 95% [0.83 , 0.85], variance $\hat{\sigma}^2(\hat{\kappa}) = 2.26^{-5}$, and Z score = 176.8 (Figure 3.5 and Table 3.2). When considering the two most likely labels given by the CNN, the overall accuracy was 94.34 % (Table 3.2). Thus, this second network provides categorical classification of USVs emitted by mice with high accuracy. Moreover, because the network outputs a probability distribution for each USV category, it provides the opportunity to use this output for further data analysis (see example below).

3.3 Biological application

Upon parental separation, infant mammals vocalize to attract the caregivers (HOFER, 1994). In mice, infants emit USVs when separated from the dam which functions as a location signal (EHRET, 2005). In our group, we are interested in understanding the neuronal circuits and behavioral details underlying the emission of USVs by infant mice.

Using the tool described in this Dissertation (VocalMat), we found that a population of neurons in the mammalian hypothalamus that expresses the Agouti-related peptide

Figure 3.5: Overall accuracy for USV classification in distinct types.



Source: The author

Figure 3.5: Performance of VocalMat in classifying USVs in 11 distinct types when compared to the Ground Truth (Manual Classification). The most likely label according to VocalMat is compared to the label assigned by an experienced experimenter.

(hereafter, Agrp neurons) modulate the emission of USVs in infant mice (ZIMMER et al., 2019). More specifically, using transgenic tools to activate Agrp neurons in ten days old mice, we observed an increase of 61% in the emission of USVs (Figure 3.6A-B) We then used VocalMat to analyze the spectral features of more than 45,000 USVs recorded during these experiments. We found that activation of Agrp neurons altered the features of these USVs, as they were in average: (1) 4 ms shorter in duration, (2) 3 kHz lower in pitch, and (3) 2 kHz broader in bandwidth when compared to the Control group (Figure 3.6C-D).

In contrast, animals unable to release the transmitter GABA specifically by the Agrp neurons - an essential inhibitory transmitter for the function of these neurons - presented lower USV emission (Figure 3.7A) (ZIMMER et al., 2019). Moreover, these mice deficient in GABA release by Agrp neurons also presented changes in the spectral features of the emitted USVs, which were broadly opposing to the effects reported for Agrp neuron activation (Figure 3.7B-D).

Next, we used VocalMat to classify the USVs emitted by infant mice in the experimental conditions described above (see Section 2.6.1). Even though there was a significant increase in the emission of USVs upon activation of Agrp neurons in ten days old mice, the relative frequency of usage for most of USV types had no significant statistical difference when compared to the Control group, with the exception of Chevron-like USVs (Figure 3.6F-G) (ZIMMER et al., 2019). In contrast to Agrp neuron activation, infant mice lacking GABA release by Agrp neurons emitted relatively more Short-like

Table 3.3: Accuracy per class

Type	N	Mean ± SEM (%)	Median [95% CI] (%)
Step up	902	83.58 ± 6.50	91.56 [66.85, 100.00]
Chevron	758	85.37 ± 3.93	85.28 [75.25, 85.48]
Two steps	579	74.41 ± 4.16	70.47 [63.71, 85.11]
Down-FM	557	90.74 ± 1.23	90.83 [87.56, 93.91]
Up-FM	485	88.04 ± 2.38	87.59 [81.90, 94.17]
Short	358	88.28 ± 1.88	89.62 [83.45, 93.11]
Complex	281	76.64 ± 3.72	76.24 [67.07, 86.22]
Flat	190	84.20 ± 4.14	83.51 [73.56, 94.84]
Step down	142	84.74 ± 4.60	83.77 [72.90, 96.58]
Mult. steps	80	45.89 ± 10.70	38.10 [16.18, 75.61]
Rev. Chevron	61	65.18 ± 14.17	73.87 [28.74, 100.00]
Noise	511	96.67 ± 0.55	96.67 [95.23, 98.10]

Source: The author

USVs (Figure 3.7E-F). Thus, VocalMat allowed us to identify a large number of USVs in audio files, providing detailed analysis of the vocal behavior of infant mice in an experimentally relevant setting.

3.3.1 Analysis of the vocal repertoire

In addition to the most likely USV type, VocalMat provides the probability distribution for each USV to be classified as one of the 11 categories (plus noise). This data provides a more detailed understanding of the USV repertoire emitted by mice, as it allows the visualization and quantification of USVs in a less deterministic manner.

To achieve this goal, we verified the extent to which the repertoire of USVs emitted by infant mice after activation of Agrp neurons had any significant change in structure despite the apparent similar relative frequency of usage of the USV types (with the exception of Chevron-like USVs; see above). We used Diffusion Maps (Section 2.7.1) to project the probability distribution of the USVs into an Euclidean space.

Our experiments were composed by four experimental contexts: (1) Agrp neurons

Table 3.4: Sensitivity considering the two most likely labels

Type	N	Mean ± SEM (%)	Median [95% CI] (%)
Step up	902	91.64 ± 4.86	97.18 [79.15, 100.00]
Chevron	758	96.08 ± 1.36	97.20 [92.57, 99.58]
Two steps	579	91.43 ± 1.80	91.77 [86.79, 96.06]
Down-FM	557	97.08 ± 0.98	96.93 [94.57, 99.59]
Up-FM	485	96.25 ± 1.40	97.30 [92.66, 99.84]
Short	358	96.53 ± 1.12	96.72 [93.66, 99.41]
Complex	281	92.10 ± 2.30	91.44 [86.20, 98.00]
Flat	190	94.21 ± 3.96	97.73 [84.02, 100.00]
Step down	142	96.11 ± 1.99	97.96 [91.01, 100.00]
Mult. steps	80	83.64 ± 7.33	85.71 [63.28, 100.00]
Rev. Chevron	61	77.65 ± 15.75	91.29 [37.17, 100.00]
Noise	511	98.00 ± 0.45	97.87 [96.84, 99.17]

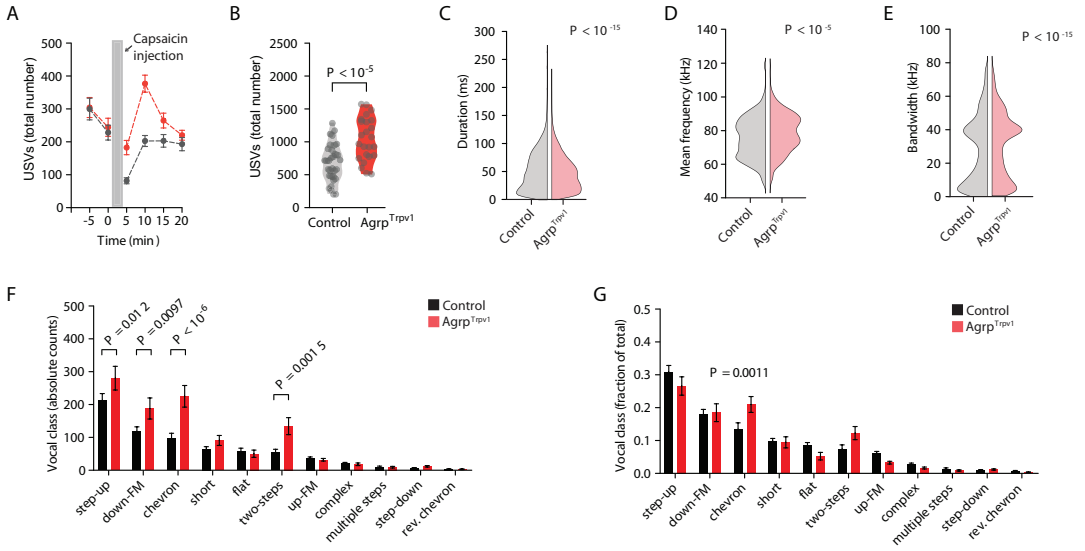
Source: The author

activation group (Test group) and (2) Control group; where each group was tested (3) before activation (1st stage) and (4) post activation (2nd stage). We compared all four conditions against each other and visually verified that the 3D structures of the USV repertoires present some degree of similarity, but were rotated or translated in space when compared to each other (Figure 3.8).

Next, to further compare the USV repertoires of the different experimental conditions, we aligned the manifolds (Section 2.7.2). Figure 3.9A shows the examples of two manifolds (left and right) in their respective original domains as result of the dimensionality reduction (Section 2.7.1). The misalignment between the manifolds is better appreciated when overlapping the two domains (Figure 3.9B). The result of the manifold alignment (Section 2.7.2) is shown in Figure 3.9C. The visual result of this alignment is shown in Figure 3.10A for each pair of experimental conditions.

At first, we estimated the similarity between the 3D structures by calculating the pairwise distance between the centroids of USV categories within each manifold (Figure 3.10B). The pairwise distance matrices provide a metric for the manifold structure, allowing a direct comparison between the different groups. Interestingly, activation of

Figure 3.6: Activation of Agrp neurons in ten-days-old mice increases USV emission.



Source: (ZIMMER et al., 2019), modified by the author

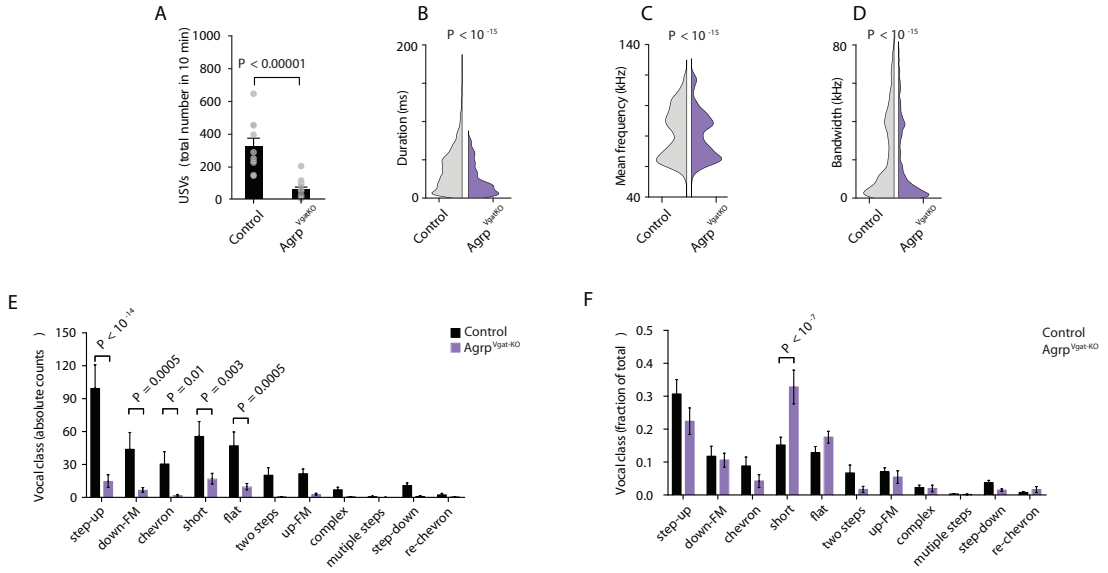
Figure 3.6: **(A)** Number of USVs per 5-min bins upon activation of Agrp neurons in ten-days-old mice. **(B)** Total number of USVs emitted during experiment. **(C-E)** Distribution of spectral features: duration **(C)**, pitch **(D)** and bandwidth **(E)**. **(F-G)** Frequency of usage in absolute count and percentage per call type. The p-values reported were corrected by multiple comparisons using Holm-Sidak method.

Agrp neurons (2nd stage of the Test group) changed the structure of the matrix with an increase in the centroid distances of USVs of 'two-steps' type, and a decrease in the centroid distances of USVs of 'short' type compared to the other groups (Figure 3.10B). We then quantified the similarity between these matrices (Figure 3.10C) and verified a high correlation between all groups, except when compared to the 2nd stage of the Test group. Similar results were obtained by using Cohen's coefficient and Overall Accuracy (Figure 3.10C). Thus, the use of the probability distribution for vocal classification and Diffusion Maps allows the identification of experimental conditions based on the altered vocal repertoire.

Interestingly, our results show a lower average pairwise correlation coefficient ρ among animals with no Agrp activation than the activated ones (Control 1st: 0.57 ± 0.24; Control 2nd: 0.57 ± 0.26, Test 1st: 0.53 ± 0.26, Test 2nd: 0.66 ± 0.22 (mean $\pm$ SD), Control 1st vs Control 2nd: $p > 0.99$, Control 1st vs Test 1st: $p = 0.07$, Control 1st vs Test 2nd: $p < 10^{-4}$, Control 2nd vs Test 1st: $p = 0.04$, Control 2nd vs Test 2nd: $p < 10^{-4}$, Test 1st vs Test 2nd: $p < 10^{-4}$, Kruskal-Wallis test corrected for multiple comparisons by Dunn's test). This result suggests that the activation of Agrp neurons reduces the variability of the vocal repertoire of infant mice.

Both the overall accuracy and Cohen's coefficient show a poor alignment between

Figure 3.7: Ten-days-old mice deficient in GABA release by Agrp neurons have impaired USV production when isolated from the nest.



Source: (ZIMMER et al., 2019), modified by the author

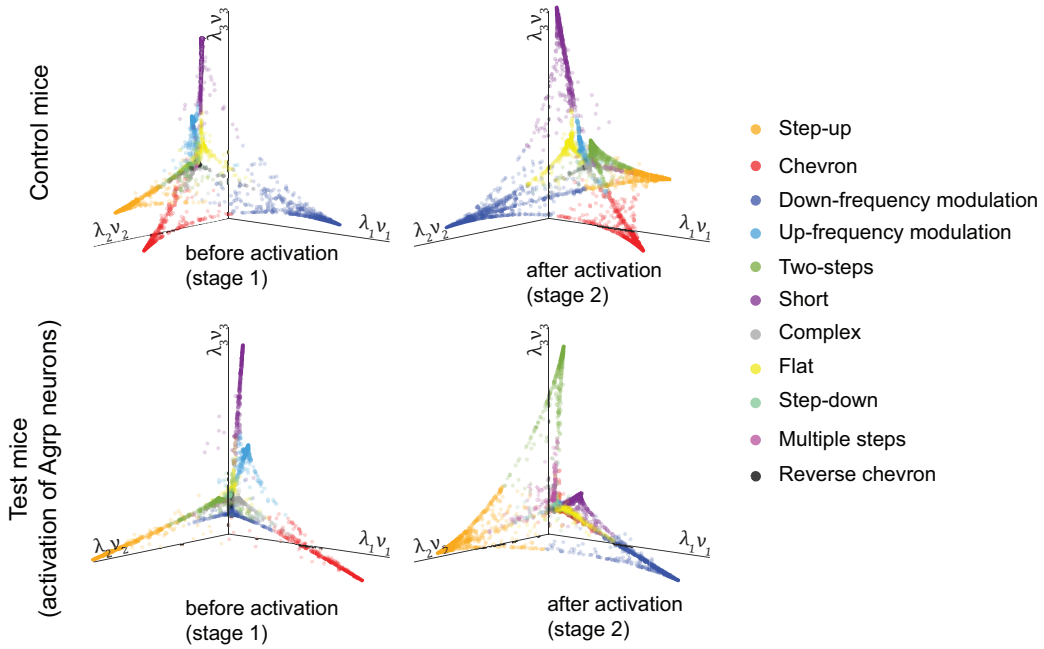
Figure 3.7: **(A)** Total number of USVs emitted during experiment. **(B-D)** Distribution of spectral features: duration **(B)**, pitch **(C)** and bandwidth **(D)**. **(E-F)** Frequency of usage in absolute count and percentage per call type. The p-values reported were corrected by multiple comparisons using Holm-Sidak method. (ZIMMER et al., 2019)

the first and second stages of the Test group (Test 1st vs Test 2nd: OA = 71.23%, κ=0.67), while presenting a higher alignment for first stage experiments (Control 1st vs Test 1st: OA = 86.13% , κ=0.84) and stages of the Control group (Control 1st vs Control 2nd: OA=90.59%, κ=0.89).

When combining the Control group in both stages and Test group in the first stage (ie, Control 1st vs Control 2nd, Control 1st vs Test 1st, and Control 2nd vs Agrp 1st), the measurements of alignment show a significantly higher similarity among these experimental contexts (κ=0.87 $\pm$ 2.86 $\times$ 10^{-2}, mean $\pm$ SD; and OA =89.01 $\pm$ 2.50, mean $\pm$ SD) than in combinations that include the second stage of the Test group (ie, Control 1st vs Test 2nd; Control 2nd vs Test 2nd; and Test 1st vs Test 2nd: κ =0.69 $\pm$ 2.73 $\times$ 10^{-2}, mean $\pm$ SD, $p = 0.0015$, 2-tailed unpaired Welch's t-test; OA =73.94 $\pm$ 2.88, mean $\pm$ SD, $p = 0.0026$, 2-tailed unpaired Welch's t-test). This result illustrates a robust structural change in the vocal repertoire upon Agrp neurons activation, which can be effectively represented by any of the alignment measurements used.

Next, we evaluated the accuracy of the projections of each domain into the new common space (aligned projections) regarding the different USV types. Since each domain has its own projection accuracy per class, we averaged the accuracy per class across

Figure 3.8: USV domains post dimensionality reduction

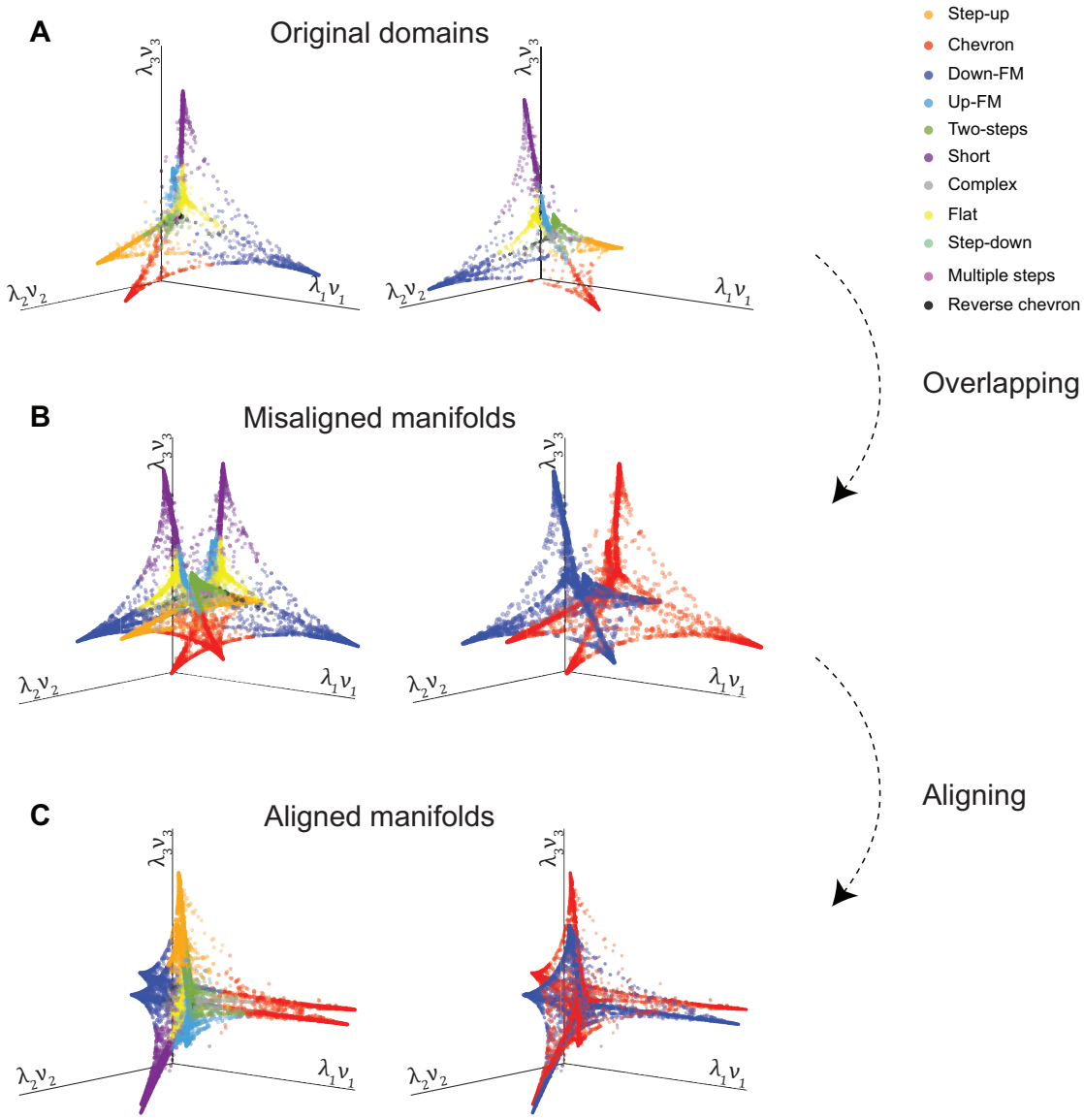


Source: The author

Figure 3.8: 3D projections for the first (before injection) and second (post injection) stages for both Control and Test (Agrp-Trpv1) mice. Agrp-Trpv1 animals allow activation of Agrp neurons specifically upon peripheral injection of a selective agonist, capsaicin. Therefore, we have two stages in the experiment, before and after capsaicin injection. Colors identify the different USV types.

domains and used this measurement as goodness of alignment between classes when projected into the new space (Figure 3.11). We verified how the combinations involving animals with Agrp neurons activated show a significant drop in alignment for Short ($p < 10^{-4}$, 2-tailed unpaired t-test), Complex ($p = 0.016$, 2-tailed unpaired t-test), Multiple steps ($p = 0.031$, 2-tailed unpaired t-test), Up-FM ($p < 10^{-4}$, 2-tailed unpaired t-test) and Two steps ($p = 0.027$, 2-tailed unpaired t-test), indicating a deviation of the regular structure of USV domain of these animals (first stage) when compared to the USV domain post Agrp activation. This result points out to an internal shift towards more complex USVs emitted by animals with Agrp neurons activated and explains the observed spectro-temporal changes in USV despite the apparent invariant frequency of USV usage. Up to date, this type of observation was not possible with the conventional USV classification methods and illustrates the richness of details that VocalMat is able to extract from animal vocal behavior.

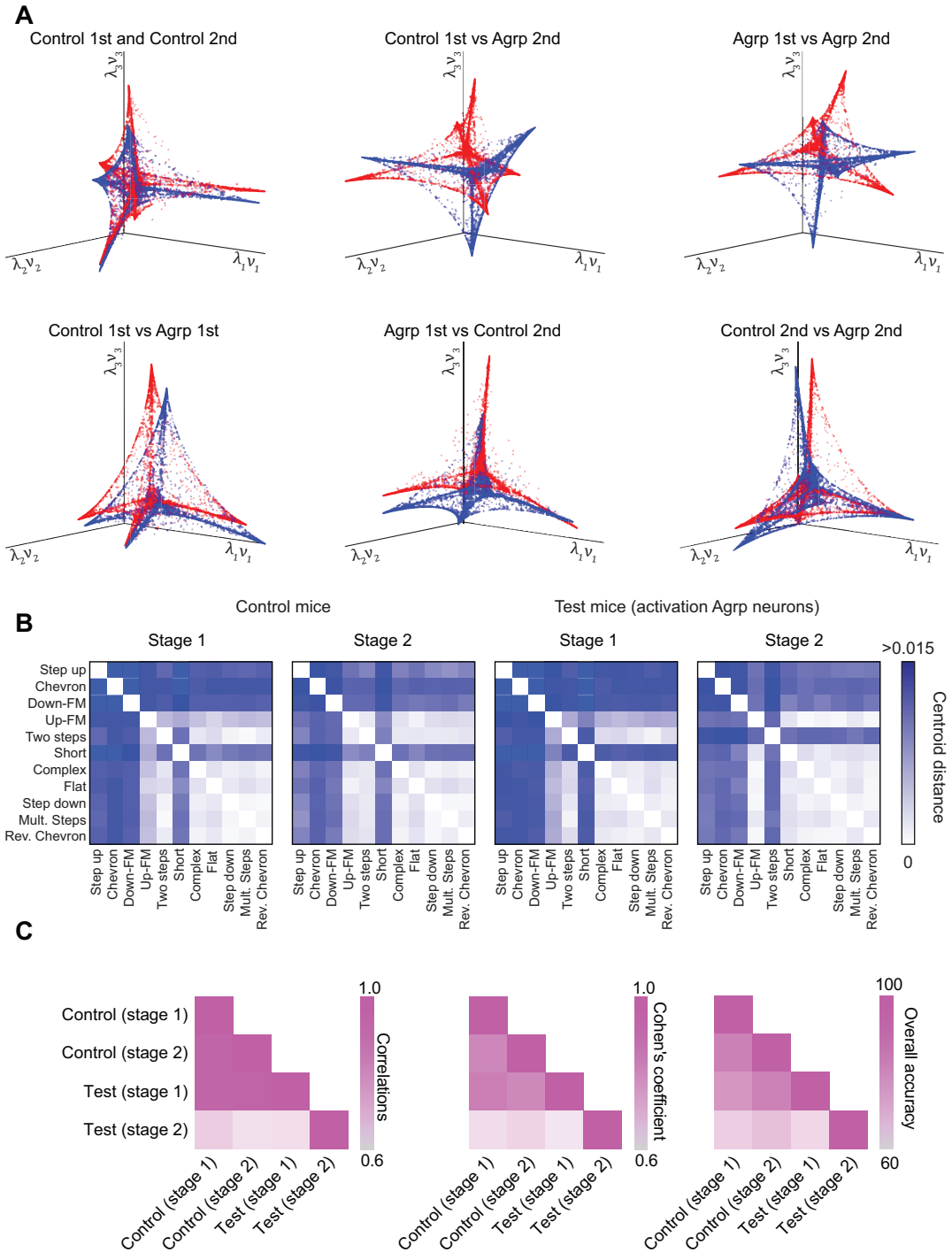
Figure 3.9: USV domains post dimensionality reduction



Source: The author

Figure 3.9. **(A)** Example of two manifolds (left and right) in their respective domain, which represents the structural properties of each vocal repertoire. The colors identify the distinct call types. **(B)** The direct overlap of the domains shows the misaligment between the manifolds according to the call types (left) and overall structure (right), difficulting a direct similarity quantification between the repertoires. **(C)** Result post manifold alignment according to call types (left) and overall structure (right) (Section 2.7.2).

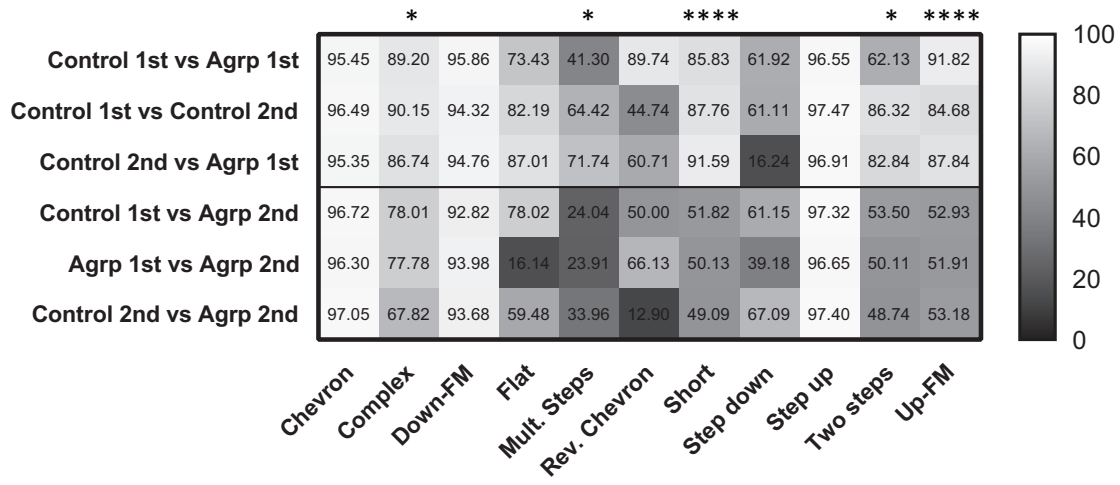
Figure 3.10: Quantification of vocal repertoire similarity between the different experimental contexts



Source: The author

Figure 3.10: **(A)** Result of the manifold alignment for each pair of experimental conditions. **(B)** Mean distance between cetroids for each pair of call types. The distance between centroids underlines the geometry of the manifold. **(C)** Measurements used to quantify the performance of the alignment operation: Pearson correlation coefficient (ρ) for each pair of matrices of distance between centroids (shown in **(B)**), Cohen's coefficient and Overall accuracy (Section 2.7.2)

Figure 3.11: Combination of projection accuracy for pair of manifolds.



Source: The author

Figure 3.11: Mean projection accuracy per call type.

Table 3.5: Measurements of quality of alignment for manifolds X_1 and X_2

		X_1 projected		**X_2 projected**		**Combined (min (X_1,X_2))**	
X_1	**X_2**	**OA**	**kappa**	**OA**	**kappa**	**OA**	**kappa**
Control 1st	Agrp 1st	86.13	0.8392	92.26	0.9101	86.13	0.8392
Control 1st	Control 2nd	90.59	0.8899	91.38	0.8990	90.59	0.8898
Control 2nd	Agrp 1st	90.30	0.8874	90.65	0.8914	90.30	0.8874
Agrp 1st	Agrp 2nd	81.62	0.7862	71.23	0.6682	71.23	0.6682
Control 2nd	Agrp 2nd	85.27	0.8222	76.97	0.7225	76.97	0.7226
Control 1st	Agrp 2nd	87.06	0.8472	73.63	0.6924	73.63	0.6924

Source: The author

4 CONCLUSIONS AND FUTURE WORK

We reported the development of VocalMat, a MATLAB-based software to automatically detect and classify mouse USVs with high sensitivity and high-throughput. VocalMat eliminates noise from the pool of USV candidates, preserves the main statistical components for the detected USVs, and detects harmonic components. Additionally, VocalMat architecture uses machine learning algorithms to further filter USV candidates in *noise* and *real USV* or in 11 different USV types adopted. VocalMat is open-source and it is already customized to run in clusters with Slurm job scheduler.

VocalMat adds to the repertoire of tools developed to study mouse USVs (SEGBROECK et al., 2017; BURKETT et al., 2015; CHABOUT et al., 2015; ARRIAGA; ZHOU; JARVIS, 2012; HOLY; GUO, 2005; COFFEY; MARX; NEUMAIER, 2019). We only found one study that reported the sensitivity to detect vocalizations (HOLY; GUO, 2005). In this manuscript, the authors reported a sensitivity of $> 95\%$ compared to $> 98\%$ achieved by VocalMat. Because these previous tools depend on several parameters defined by the user, it is difficult to efficiently compare their performance to VocalMat, but our tests show VocalMat outperforming the other tools in sensitivity and accuracy, at least in our test data set. As previously stated, our process of choosing the parameters for those tools was done such that they could show their highest accuracy and sensitivity, but it does not exclude the possibility of a more optimal configuration that could lead to better performance.

Regarding USV classification, we have shown a high performance of VocalMat in assigning a label to an USV according to its probability distribution by picking the most likely ($> 86\%$ of accuracy) and second most likely label ($\approx 95\%$ of accuracy). Importantly, by treating USV classification as a problem of probability distribution across the USV types, VocalMat provides a more flexible classification method. Such method allowed us to visualize the repertoire of USVs using 3D plotting, which provides a means to visually inspect the similarities between USV types. We were then able to compare the structure of these repertoires and quantify similarity scores across experimental conditions, taking into account the entire repertoire of USVs or a specific USV type. Our analysis illustrate complex spectro-temporal changes that occur in mouse vocal behavior that were not readily accessible using previous methods.

VocalMat uses a pattern recognition approach based on Convolutional Neural Networks, which learns directly from the training set without the need of feature extraction

via segmentation processes (SCHMIDHUBER, 2015; KRIZHEVSKY; SUTSKEVER; HINTON, 2012). This characteristic of VocalMat provides unique adaptability and the possibility to be used in different experimental settings. VocalMat was developed and trained to identify USVs emitted by mouse infants. However, since the CNN can be easily trained using different data sets, VocalMat could potentially handle vocalizations from other species as well.

We are now working on methods to further refine the detection process and reduce the number of false positives. To this end, we have been experimenting with different Deep Learning architectures for semantic segmentation. Based on the published results by DeepSqueak (COFFEY; MARX; NEUMAIER, 2019) using Faster-RCNN, which lead to low accuracy in time stamp for the USVs, we intend to use Deep Learning to refine - rather than replace - our current segmentation process . Thus, detected USV candidates would go through one more processing step, which would then keep only significant pixels for further analysis. By doing so, we expect a decrease in number of false positives, a reduction in amount of data stored for the USV classification task, and a more accurate extraction of spectral features for each USV type.

In summary, VocalMat is a new software tool to detect and classify mouse USVs with exquisite sensitivity and accuracy while keeping all the relevant spectral features, including the existence of harmonic component in the USVs. By combining this robust method of vocal analysis with other biological data (e.g., in vivo neuronal recordings, behavioral and physiological measurements), we are now able to have a more complete view of mouse vocal behavior.

5 SUPPLEMENTARY SECTIONS

In this section we will present methods that were tested or were once part of VocalMat. For each method presented below, we mention some of the limitations intrinsic to the technique and why it was later removed from the main body of the tool. As the these previous versions were no longer continued, many of the results that will be presented in this section are preliminary, which accounts for simple graphics and not shallow discussions. Nonetheless, for the sake of completeness of this work, we will briefly comment on our previous version.

5.1 Filtering noisy vocalizations by analysis of probability density functions

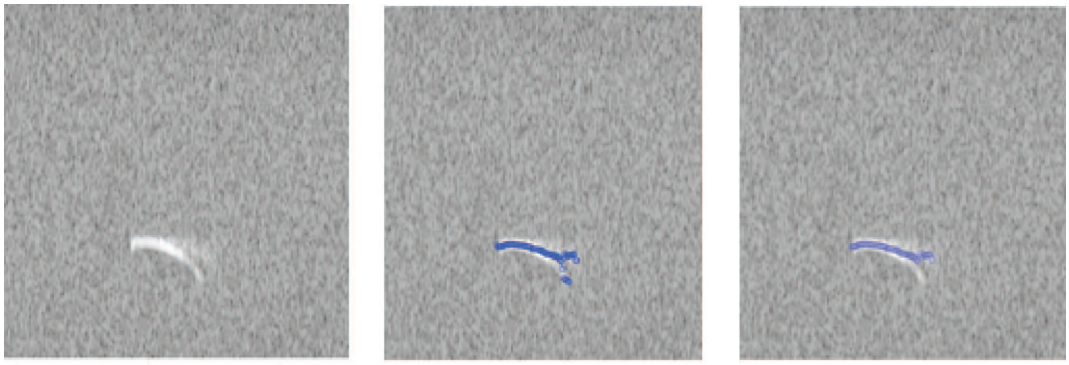
The Local Median filter described in Section 2.5.1 applied to the USV candidates had the goal to completely remove a candidate from the list based on its median energy of points detected as being part of an USV when compared to the median energy distribution of the surroundings. Here we tried a softer approach, which intended to just refine the segmentation rather than remove the candidate.

As the segmentation process might include some extra points to the vocalization due to a distortion (e.g.: echo on the walls) or even external noise, the next filter targets points that have an energy below an acceptable level of energy for a given set of points detected as being part of an USV. The acceptable deviation of energy intra-vocalization is estimated based on the pattern of energy distribution presented by the points detected.

This filter should not be applied to all the detected USVs, otherwise even for a set of points that already brings a good representation of the USV shape, we would be eliminating points, which would be counterproductive for a future classification of the USV. Thus, one of the requirements the USV candidate has to attend in order to go through this filtering is that the energy (or intensity) distribution has to be a multimodal distribution, which means that energy distribution should present at least two significant distinct peaks. This restriction comes from the fact that if the distribution for an USV is Gaussian, either it is a real USV without noise or it is pure noise (observable in our data but not shown here), in both cases, eliminating points is not going to bring any benefit.

To better illustrate the process, Figure 5.1A shows an USV candidate without any points detected. Figure 5.1B shows the same USV but with the points overlapping the center of the clouds detected during segmentation, as explained Section 2.3. The energy

Figure 5.1: Result from the filtering by probability analysis



Source: The author

Figure 5.1: Example of vocalization without the points of detection **(A)**, then showing the points detected **(B)** and the result after the noise removal method **(C)**.

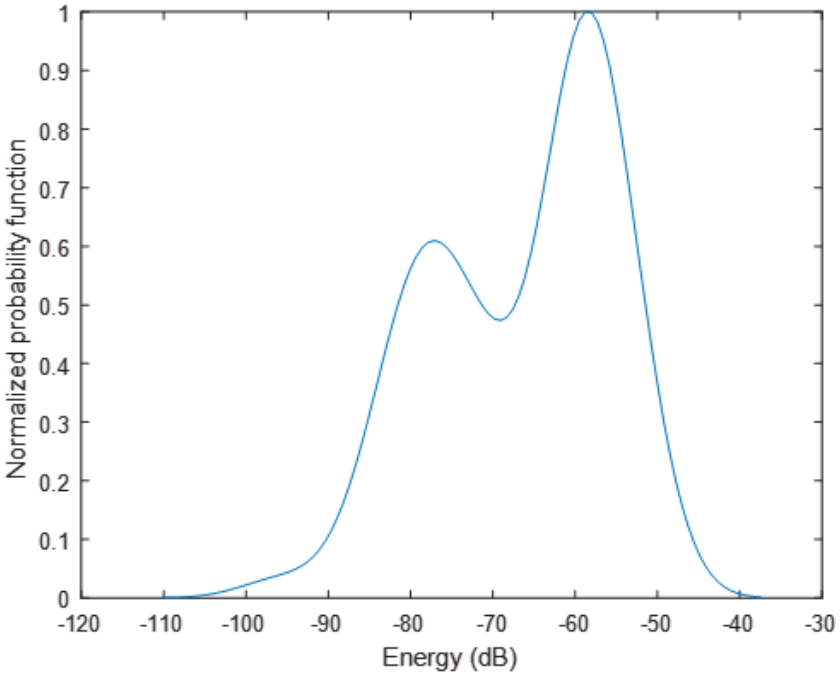
distribution of the points detected in Figure 5.1B is shown in Figure 5.2 . As we can see, the distribution has features of bimodality, which qualifies this USV candidate to go through the steps of filtering by standard deviation.

Another important factor that has to be taken in consideration is the fact that in many cases the harmonic component of an USV (when it exists), usually presents lower energy than the fundamental component of the call. So checking just for the bimodality on the distribution is not enough, but it's also necessary to check for the possibility of these points with distinct energy being part of a harmonic component.

Harmonic components in mice USV presents a known structure and typical variance across different calls. Neunuebel and colleagues (NEUNUEBEL et al., 2015) described harmonic components as overlapping signals when their lengths overlapped in time for more than 90% of the shortest signal, and the frequencies of 90% or more of the overlap were within 10% of a factor of two or three of each other. As this might be a very restrictive way of describing a harmonic, we decided to define what might not be a harmonic. Therefore, this filtering method would target only overlapping signals that overlapped for less than 30%, which is just one third of what a harmonic could be and can prevent us from eliminating potential harmonics. This length constrain can be easily expressed as a ratio between the peaks height in the energy distribution, since the height of the peaks represent the number of detected points with certain intensity.

Besides the energy distribution, the way how the energy is distributed across frequency also brings relevant information to understand which points are actually part of the USV or not. The principle is that by analyzing the way how the energy changes in the

Figure 5.2: Energy distribution of an USV



Source: The author

Figure 5.2: Normalized energy distribution for USV shown on Figure 5.1B

time window where the USV was detected, we should have a good understanding of the USV shape.

To illustrate the concept, consider again the USV represented on Figure 5.1. For example, as explained in Section 1, spectrograms are a graphical representation of how the energy is distributed across frequency for a given interval of time, and can be represented as a surface as shown in Figure 2.2A.

The distribution of energy across the frequency for the whole time window would be the correspondent to get the maximum energy for each frequency point for the time window where we have the USV, which could also be represented as the rotation of the plot in Figure 2.2A in order to see the frequency as abscissas and energy as ordinates, generating the distribution seen as a wall projection in Figure 2.2A. We then normalize this distribution, which tell us the likelihood of an USV existing in that bandwidth.

Once we have the curve of energy distribution across frequency and the curve of normalized probability density function of energy, we are able to estimate what is the probability of a point being part of an USV given its frequency and energy compared to the surroundings.

So, for a given USV, its energy distribution will be represented by the smoothed

probability density function (BOWMAN; AZZALINI, 1997) given by

$$\widehat{I}(i) = \frac{1}{n}\sum_{k=1}^{n} w(i - i_k; h) \tag{5.1}$$

where i denotes the points at which the density $I(i)$ must be estimated. The set $\{i_1, \ldots, i_n\}$ denotes the observed data and i_k is the center of the interval $[-h, h]$ that contains n elements. The parameter h (also known as smoothing parameter or bandwidth) in this case represents a subset of points in the array of points detected for an USV and plays an important role in describing the manner in which the probability associated with each observation is spread over the surrounding sample space. The parameter w is itself the probability density function with its variance controlled by the interval h and conveniently chosen represented by a normal density function. In other words, the smoothed probability density function is an average of normal density functions.

In a similar way, we can also describe the smoothed distribution of intensity across frequencies I_2 as

$$\widehat{I_2}(i) = \frac{1}{n}\sum_{k=1}^{n} w(i - i_k; f) \tag{5.2}$$

where the smoothing parameter f is given as frequency range and w is the probability density function for the energy for the time interval where the USV happened. Since both curves are normalized, for an energy x frequency point, if the product of the probabilities is close to one, it means the point has high energy when compared to the surroundings and lies on the bandwidth where the points with high energy are located on the spectrum. Otherwise, either it has low energy or relies away from the bandwidth where the high energy points are located in the spectrum, which in both cases would indicate the low probability of that point being part of an USV. The result of such process is illustrated in Figure 5.1C .

The minimal probability that a point needs to be kept alive is 0.25, which would mean that at least 0.5 of each probability function should be indicating a reasonable probability of being an USV or one of the functions showing a very high probability in order to keep the point alive to the next steps. Therefore, the decision for keeping of discarding the points will be given by

$$i = \begin{cases} 1, & \text{if } \widehat{I}.\widehat{I_2} > 0.25 \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

This method was later removed from the main body of VocalMat since the classification of the USVs by CNN was less dependent on the segmentation and more robust to noise. Also, this method proved it self efficient only to handle noise originated by the segmentation steps, and not cases of mechanical noise. The mechanical type of noise is the most common found in our recordings, which then reinforced the idea of discarding this approach.

5.2 Methods for USV/noise differentiation

Once the process of segmentation and labelling of the USVs is done, a new step is required in order to evaluate the quality of the USV detected and eliminate noise wrongly labelled as USVs. The previous versions of VocalMat used various approached to determine if a detected USV is indeed something relevant. For that, statistical measurements were taken from the set of points detected in order to estimate the level of dispersion of the points, and a machine learning approach is used to combine all the information collected to decide if an USV will be kept or not. In this section we describe some of the previous approaches used to filter out the noise detected from our USV candidates.

5.2.1 Hierarchical clustering

One visual feature that is very significant to determine if an USV detected is indeed an USV or noise is the level of organization of the points contained in that USV. A classic way of making inferences about the organization level of a set of data is by calculating the entropy, which in its most general definition, numerically represents the homogeneity of a sample.

Previous works shown by Tchernichovski (TCHERNICHOVSKI et al., 2000) and Sirotin (Sirotin, Costa and Laplagne (2015)) have used entropy in the past to detect USVs. The principle of this technique relies on the fact that entropy for a given interval of time in the recording will drop significantly, indicating the existence of a possible USV for that interval. However, as shown by Sirotin, this technique can't tell you more than a probability of existing an USV in a time interval and nothing about the actual spectral features of it, which we believe to be fundamental to the process of understanding the animal behavior associated to each syllable.

As the points believed to be part of an USV have already been detected, a logical way to estimate the organizational level of these set of points is by estimating the capacity of these points to be organized in clusters. In other words, if the points detected can be organized in few distinct clusters, it means that the points are likely related to a real USV, while a larger number of clusters indicate the disorganization level is higher and more likely it is noise. The most classic clustering methods demand as input the number of clusters to be identified in the dataset. This is not a trivial information to extract from the points since the USVs can present different shapes as shown in the literature review. A suitable method for clustering under this conditions is the hierarchical clustering.

Hierarchical clustering is a method for cluster analysis designed to build a hierarchy of clusters either in an agglomerative or divisive way. In the agglomerative approach, each observation starts in its own cluster, and pairs of clusters are merged as one moves up to the hierarchy. The clusters are then sequentially combined into larger clusters, until all elements end up being in the same cluster. At each step, the two clusters separated by the shortest distance are combined.

In our application, the shortest distance is the single linkage clustering, in which the distance between two clusters is determined by a single element pair, namely those two elements (one in each cluster) that are closest to each other. The shortest of these links that remains at any step causes the fusion of the two clusters whose elements are involved. The method is also known as nearest neighbour clustering. Mathematically, the linkage function – the distance $D(X, Y)$ between clusters X and Y – is described by the expression

$$D(X, Y) = min_{x \in X, y \in Y}(d(x, y)) \tag{5.4}$$

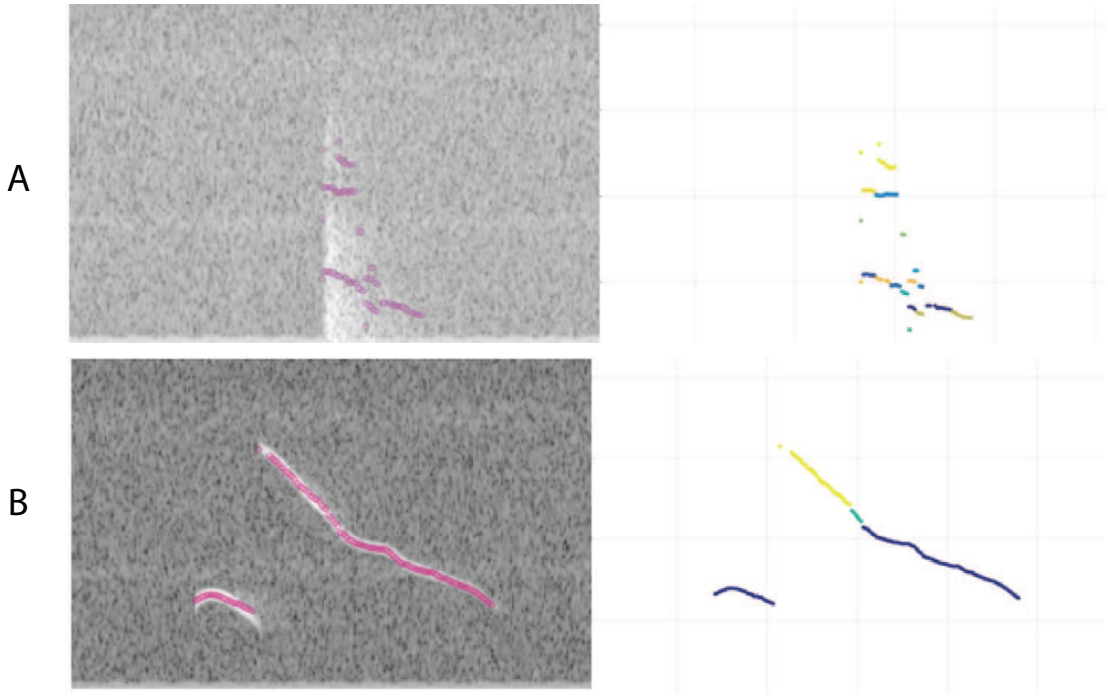
where X and Y are any two sets (clusters) and $d(x, y)$ denotes the distance between the two elements x and y.

Logically, this agglomerating process should continue until a point where the shortest distance between the clusters is already too big to be considered still part of the same cluster and that is why a cut off criterion must be defined. Since the time resolution is 0.5ms while the frequency resolution is 244Hz, and the frequency range considered for our analysis relies between 45kHz and 120kHz, the frequencies were divided by a factor of 104 in order to reduce the magnitude of the constant that defines maximum distance between clusters. By this way, the resolution in frequency is still greater than the time resolution, resulting in a higher importance to jumps in frequency than in time, which is

exactly what we want.

By clustering the detected points for each USV candidate detected, we are able to evaluate the organizational level of the points. The result will implicate in a greater number of clusters for USVs candidates that are actually noise and fewer number of clusters for real USVs. The Figure 5.3 illustrate the results given by the hierarchical clustering for sequence of points that are actually noise (Figure 5.3A) and for a real USV (Figure 5.3B).

Figure 5.3: Example of cluster formation for USV and noise



Source: The author

Figure 5.3: Illustration of points detected for a vocalization candidate that is noise **(A)** and real USV **(B)** with their respective cluster represented on the right by different colors.

This method was being used in association with Random Forest (Section 5.2.2), but it was later dropped from the main body of VocalMat since CNN (Section 2.5.3) outperformed Random Forest in identifying noise.

5.2.2 Random Forest

As has been discussed up to this point, there are several features that must be taking into consideration in order to correctly distinguish a real USV from noise. A smart way of concatenating all this information in a useful manner is by developing a model able to mix all these different features and measurements and make a prediction, besides

the possibility of updating the model to make the machine "smarter" in performing the task.

Decision-tree classifiers are attractive because of their advantages, such as straight forward training and extremely fast for classification. The traditional decision-tree classifier is built by the use of heuristics to construct the tree for optimal classification or to minimize its size. However, trees constructed with fixed training data are subject to overfitting of the data and shows low accuracy for unseen data due to the tree's low flexibility.

Ho (HO, 1995) introduced the concept of multiple decision trees constructed in a randomly selected subspace for classification and showed that it could achieve an increased flexibility of the model while still preserving the accuracy on the training data.

This concept was then extended by Breiman (BREIMAN, 1996) and his bagging predictors, where he proposes to grow each tree by random selection (without replacement) from the samples in the training set.

Breiman's procedure (BREIMAN, 2001) consists on n trees, where for a k^{th} tree out of the n, a random vector Θ_k is generated, independent of the past random vectors $\Theta_1, \ldots, \Theta_{(k-1)}$ but with the same distribution. This tree is a grown using the training set and Θ_k, resulting in a classifier $h(x, \Theta_1)$ where x is an input vector. For instance, the process of bagging the random vector Θ is correspondent to generate counts in n boxes resulting from N darts thrown at the boxes, where N is number of samples in the training set. The nature and dimensionality of Θ depends on its use in tree construction.

After a large number of trees is generated, they vote for the most popular class. So the whole process consists in building decision trees randomly and letting them decide the result for a given input. This process is called Random Forest.

To better exemplify the theory, let's think about a dataset S given by N samples (rows in S) and M features (columns in S) with a observed output C for each sample.

$$S_{n,m} = \begin{pmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,m} & C_1 \\ f_{2,1} & f_{2,2} & \cdots & f_{2,m} & C_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f_{n,1} & f_{n,2} & \cdots & f_{n,m} & C_N \end{pmatrix} \tag{5.5}$$

So, in a given training set with N observations, features $F = f_1, f_2, \ldots, f_m$ with responses $C = C_1, \ldots, C_n$, by bagging B times we would have B random subsets of the

training data. Each selection (bagging) will get randomly $\sqrt{M}$ subfeatures out of the M features sample.

$$S_b = \begin{pmatrix} f_{a,A} & f_{a,B} & \cdots & f_{a,K} & C_a \\ f_{b,A} & f_{b,B} & \cdots & f_{b,K} & C_b \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f_{k,A} & f_{k,B} & \cdots & f_{k,K} & C_k \end{pmatrix} \tag{5.6}$$

where $(a, b, \ldots, k) \in \{1, \ldots, N\}$ and $(A, B, \ldots, K) \in \{1, \ldots, \sqrt{M}\}$

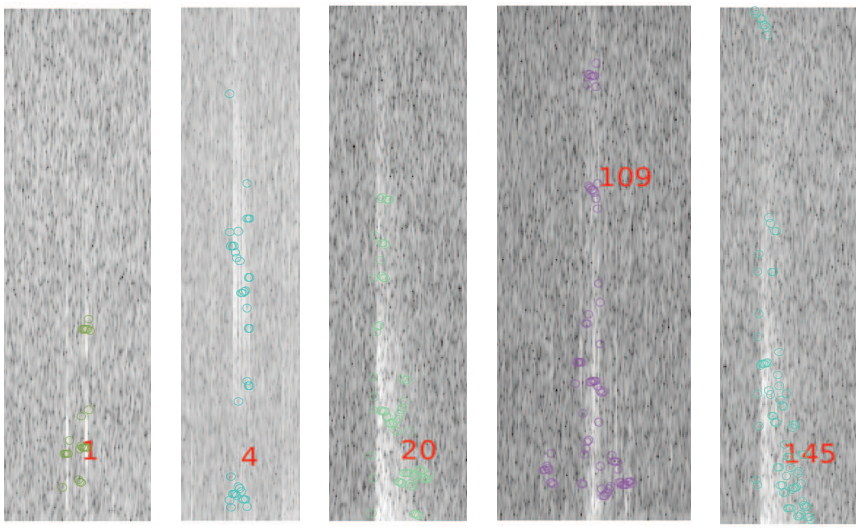
Prediction for new data f' is gotten by regression towards the mean

$$\bar{s} = \frac{1}{B} \sum_{b=1}^{B} s_b(f') \tag{5.7}$$

For our training, 8827 samples (USVs) were manually classified as real USV or noise based on the spectrograms. We analyzed 10 features to identify an USV:

1. **Peaks of intensity below 50kHz**
   As illustrated by Figure 5.4, a characteristic of burst noise is the higher intensity in low frequencies (generally below 55kHz) and a decrease of intensity as it keeps extending to higher frequencies. The distribution of intensities across frequencies for a burst noise is also illustrated in Figure 5.5. Thus, the presence of a peak for low frequencies can indicate that the set of points correspond to noise.
2. **Maximum prominence** The prominence of the peak in the intensity distribution also brings a lot of information about the USV. As we can verify by comparing Figures 5.5A and B, rather than identifying peaks in this distribution, identifying relevant peaks is more informative about the USV. Once the noise presents a relatively constant decrease in intensity as the frequency goes higher, more likely the noise won't present relevant peaks in its distribution. A relevant peak is defined as a peak at least 30% higher than its neighborhood.
3. **Maximum prominence and correlations** As much as the intensity distribution for the time interval where the USVs happens, the distribution of the points detected as part of the USV are also important to identify noise among the real USVs. The distribution of the points detected across the frequencies is a measurement already taken in consideration by itself in the hierarchical clustering method, but the overlap

Figure 5.4: Example of common burst noise



Source: The author

Figure 5.4: Illustrative example of the commonly found burst noise in a spectrogram.

of the intensity distribution around the USV and the frequency distribution for the USV, can bring extra information.

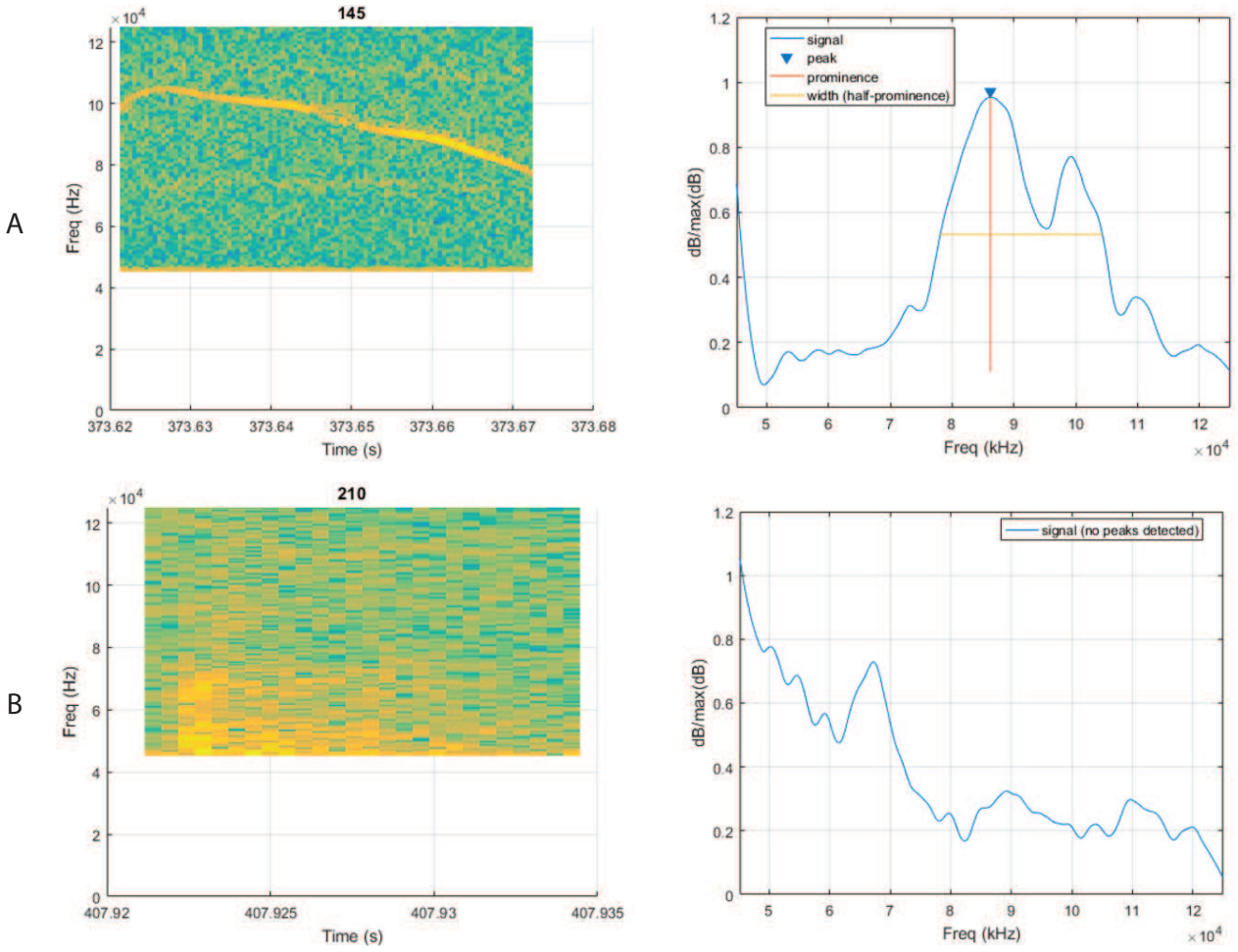
As we can see in Figure 5.5A, around 373.65s, the USV loses energy and gets back to its original intensity. However, this weakening costs the non-segmentation of points in that region and causes the valley seen in the Figure 5.6A in the frequency distribution (in blue) around 93kHz and consequently in the overlapping as well (Figure 5.6). But in general, a real USV presents a more continuous probability density distribution, which differs from the typical distribution for noise (Figure 5.5B).

The correlation coefficient between the curves of intensity distribution and frequency distribution is also a feature analyzed by the machine learning method, as well as the correlation between the intensity distribution and the result of the multiplication between the intensity and frequency distributions. In theory, the correlation between the frequency and intensity curves should be higher for a real USV (Figure 5.6A) than for noise (Figure 5.6B).

4. **Median and Mean distance between points**

   As explained previously, USV is detected point by point in time and those points with significance enough to be interpreted as part of an USV are then highlighted with a marker. The overall distance between the points of a real USV should be shorter than the distance between points in noise. So the median and distance between successive points detected for an USV should also be a good feature to

Figure 5.5: Intensity distribution for real USV and noise



Source: The author

Figure 5.5: Intensity distribution for a real USV **(A)** and noise **(B)**

describe the noise.

5. **Mean peaks/valleys**

   For the cases where more than one relevant peak is detected, the average of those points is calculated in order to give us a better idea about the overall. In a similar way, the valleys are also taken in consideration.

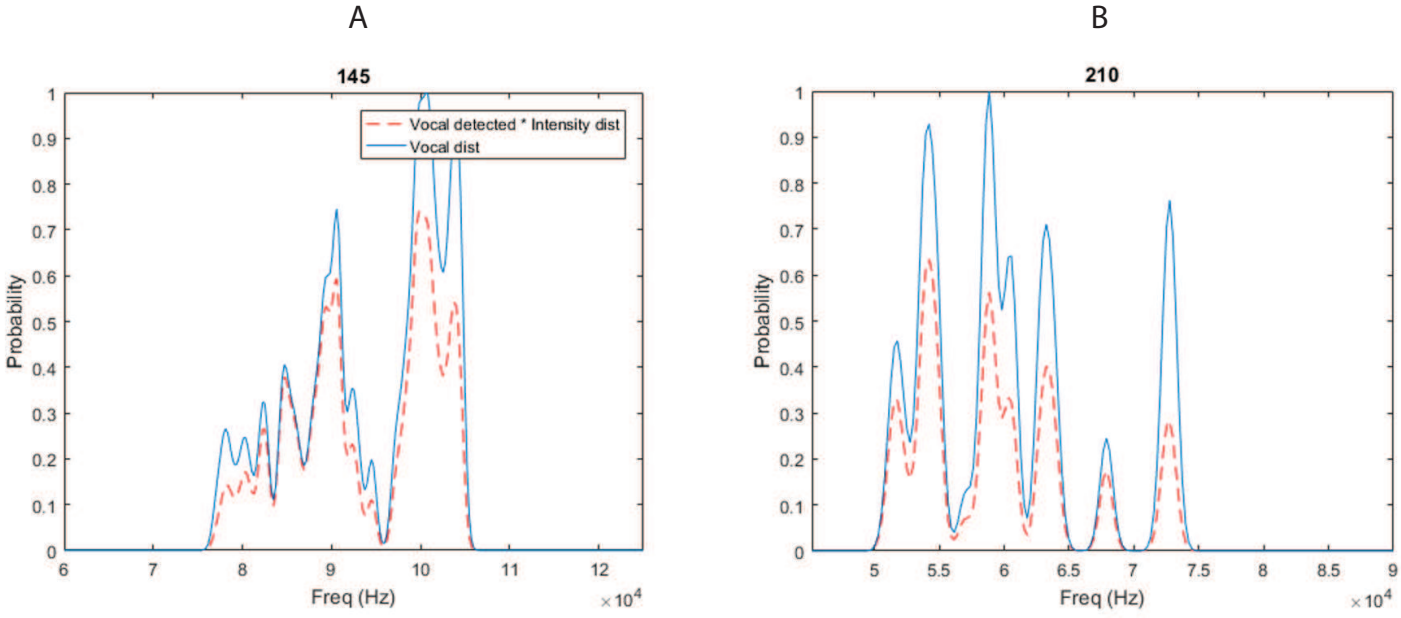
6. **Hierarchical clustering**

   As the hierarchical clustering (Section 5.2.1) already makes an analysis of the set of points in order to identify noise, the output of the clustering is also one feature used as input to Random Forest.

7. **Duration**

   Either a too short or too long USV has chances to be noise based on previous works Grimsley, Monaghan and Wenstrup (2011). For the case of too long USV candi-

Figure 5.6: Correlation between the curves of intensity distribution and frequency distribution



Source: The author

Figure 5.6: Overlapping of curves for frequency distribution of a vocalization (in blue) and the overlapping of the frequency distribution and intensity distribution (traced red line), represented for a real USV **(A)** and noise **(B)**

dates detected, it could be easily a high frequency constant noise like usually from electrical source, and examples of too short duration could be absolutely anything, but most part of the time associated with a poor segmentation.

The out-of-bagging error is conditioned to the number of trees you have in your model due to the fact that as you increase the number of trees making prediction to your input, closer you get to an accurate result.

The error can be calculated by creating the classifiers (B trees made out of S) and take inputs (f_i, C_i) from the original training set S, select all S_k which does not include (f_i, C_i). By this way you are able to compare the prediction for an input that is not part of your training set but for which you still have a ground truth. For our data, we monitored the mean squared error (MSE) for our out-of-bag observation in the training data with 200 trees. Our test shown that the MSE was already in its minimal for 60 trees.

The process of choosing the features to be used as inputs can be tricky due to the fact that we don't know how relevant all these features are to determine the output. A method to verify the relevance of a feature is by permuting the values of a feature across every observation in the data set for each feature and measure how much worse the MSE becomes after the permutation. As there is no way to estimate a p-value for relevance in

this case, we can compare each variable to a random variable, which we know there is absolutely no correlation to our samples. The result of such procedure for our data set showed that all the variables were relevant for the our predictions when compared to a random variable.

Comparing the ground-truth to the number of detected USVs, we obtained great correlation ($R^2 = 0.9969, P < 10^{-4}, \alpha = 1.023$). The rate of false negatives was $3.114 \pm 3.395\%$ and the rate of false positives was $0.361 \pm 0.182\%$. This was a great performance for USV detection, but still lower than what was obtained by using CNN (Section 3.1.2).

5.3 Methods for USV classification

In this section we describe the other methods tested for USV classification

5.3.1 Random Forest

Given the good performance of Random Forest to distinguish real USVs from several different types of noise, we decided to take this method one step further by using the same principle to also classify USVs into their different types. However, instead of using Random Forest for a binary output as seen in the application for noise identification (Section 5.2.2), in this section Random Forest will be used as categorical classifier with multiple classes, where a set of predictor variables will be associated to a class according to the mode of the classes. In addition, the chosen predictors for this new implementation should be able to express the spectral shape of the USV.

The process of growing the trees follows the same procedure described in Section 5.2.2. The predictors are estimated based on the observation of 15 and 30 points equally spaced detected within the USV segmented. A detailed explanation for each one of the predictors is given below.

1. **Time resolution**

   Distance in time between two successive points among the 15/30 points being evaluated. This is an expression of time resolution for the USVs being analyzed. Since the extracted number of points has to be the same for all the USV (either 15 and 30), it is important to know the time resolution used. In cases where the USV is too long, 15 or 30 points might not be enough to represent the shape, as some sort of

aliasing effect.

2. **Duration**
   Distance in time between the first and last point detected for a given USV. Important parameter to identify short calls.
3. **Bandwidth**
   Distance in frequency between the maximum and minimum frequency detected for a USV. Important parameter when trying to distinguish flat and short calls.
4. **Slopes**
   This parameter gives a slope between every two successive points detected, representing the frequency change as function of time. This might be an important predictor to identify up and down frequency modulation call types.
5. **Jumps**
   Gives the distance in frequency between every two successive points. Helpful predictor to identify calls with steps (step up, step down, two steps and multiple steps) and noise, since noise usually presents many jumps in frequency in a very random way.
6. **Higher jumps**
   Identify jumps in frequency greater than 8kHz from a low to high frequency. Important predictor to identify step up and two steps calls.
7. **Lower jumps**
   Quantifies number of jumps greater than 8kHz from high to low frequencies, identifying step down and two steps calls.
8. **Intensity**
   Summarizes power spectral energy of each point, which is important information to distinguish a real USV from a noise generated during segmentation.
9. **Frequency**
   Returns the frequency of each point detected within the USV. This parameter might be helpful to spot noise generated during segmentation process, since this kind of noise is usually found in very high and very low frequencies (upper and lower borders of the spectrograms).

As mentioned previously, all these predictors are estimated based on a subset of the points detected within the segmented USV. Initially 15 points are chosen such that they are equally spaced samples of the same USV, extracting the spectral shape of the

call by collecting the parameters mentioned above. In a second round, 30 points are selected, also equally spaced samples of the same call, doubling the resolution. By this way, it is expected that outliers and eventual noise (from the audio or from the segmentation process) will be easily identified by combination of the predictors. Besides that, the predictors used previously for noise in the recordings (shown in Section 5.2.2) were again used as predictors in this stage, in order to ensure that noise would be properly identified. With that, a total of 236 predictors were used as inputs for this new Random Forest implementation for USV classification.

The number of trees to be grown was set to 200, but simulations show that with 100 trees the mean square error of the classification was already the minimal. The number of USVs used for training of this new Random Forest was 7891 USVs recorded from pups in isolation.

The output of the classifier gives you a probability of a given data sample to belong to each one of the classes. Each one of the 200 trees were built by randomly selecting $\sqrt{M}$ features from the list of predictors, where M in this case is 236 predictors. As consequence, the output give us a reasonable idea about how reliable is the label given to the USVs and a better idea about the dynamic with which the animals use USVs and change spectral shapes.

Figure 5.7A shows an example of this gradual change in the spectral features of the USVs in a phrase. Random Forest gives the fitting probability of the 4 USVs shown in Figure 5.7A for each one of the call types. This output allow us to verify interesting properties of the phrases used by the animals, such as the one shown in Figure 5.7B.

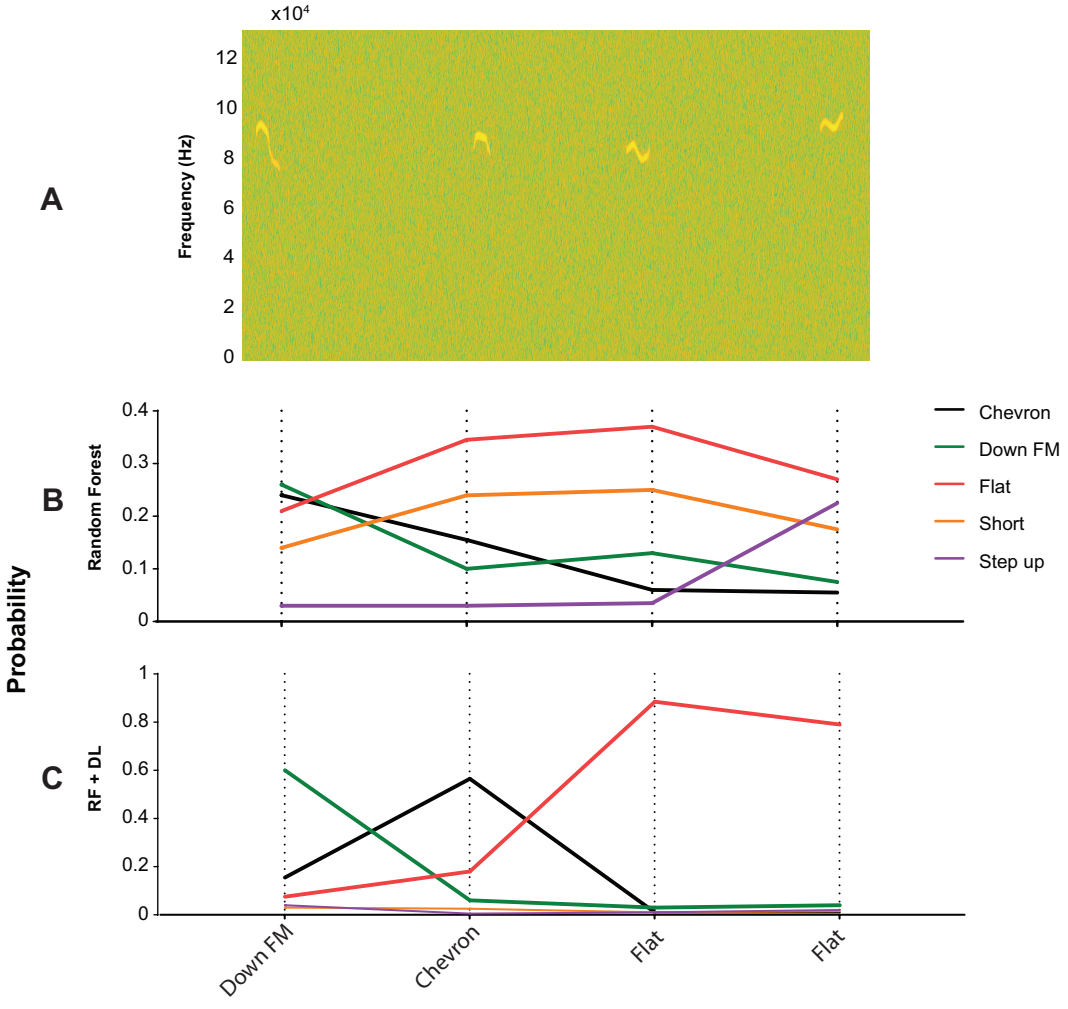
In Figure 5.7B we see that rather than moving from one call type to another completely different, the pups in isolation modify spectral shapes gradually while vocalizing, as we can see by the gradual change from a Down-FM (that resembles some properties of chevron) to a Chevron with very low oscillation in frequency (almost the same proportion to a Flat) and then a sequence of Flats. This kind of pattern is often seen in sequences of calls as reported by Chabout and colleagues (CHABOUT et al., 2015) and Castelucci (CASTELLUCCI; MCGINLEY; MCCORMICK, 2016). However, none of the current tools are able to perform such analysis given their classification methods.

Chabout (CHABOUT et al., 2015) analyzed the sequence organization of male mice calls in different contexts, where they found that calls without jumps had higher probability of starting a sequence and that those calls were repeated in loops in different contexts. Given the generalization of their “call without jumps” types, it is likely that

there are relevant information being missed by such a simplistic analysis and that in reality, those repeated calls were actually very dynamic and meaningful. Meanwhile, our approach brings a less discrete method able to quantify the dynamic of the calls and their fitting probability to the known call types.

By taking the most likely label as the right label to a call, this method granted us with 88.7% of accuracy in a test with 686 USVs candidates (583 classifiable USVs and the remaining had too much noise or could not be manually classified by the experimenter).

Figure 5.7: Comparing classification performance by Random Forest and its combination with Deep learning (CNN)



Source: The author

Figure 5.7: Same of a spectrogram **(A)** assessed by Random Forest **(B)** and a compositional model of Random Forest and Deep Learning. A notable improvement in performance was obtained, demonstrated in **(C)** by higher probabilities to specific class types. The x-axis labels shows the ground truth (given manually by experienced experimenter).

A way to evaluate how confident the Random Forest classifier was regarding the assigned label is by comparing the score given to the most likely ($P1$) call type (the one

that will give the label to the sample being analyzed) and the second most likely ($P2$). Our tests show that among the USVs correctly classified, the average highest score $P1$ was 0.69$\pm$0.11 while $P2$ was 0.12$\pm$0.05. The ratio between the two highest probabilities are in the order or 5.75 folds. On the other hand, the scores $P1$ and $P2$ for the USVs wrongly classified is showed a significant overlap between $P1$ and $P2$, resulting in a smaller ratio $P1/P2$. Comparing both distributions, it is possible to verify the potential existence of a linear separation between the two groups as function of the ratio $P1/P2$, as some sort of reliability threshold.

Analyzing the probabilities given by the Random Forest classifier (Figure 5.7b) we see how mixed are the probabilities and how it impacts the ratio prob1/prob2, which will be constantly dragged down by having such little probabilities. For example, the first USV detected in Figure 5.7A was classified by the Random Forest classifier as being a Down-FM (higher probability, $P = 0.26$), but with similar probability of being a Chevron ($P = 0.24$). With such similar probabilities, any kind of label given to this USV would be essentially a guess. Also, Random Forest depends on the accuracy of the segmentation process, which is subject to problems due to the presence of noise in the recording or abrupt change in the background noise.

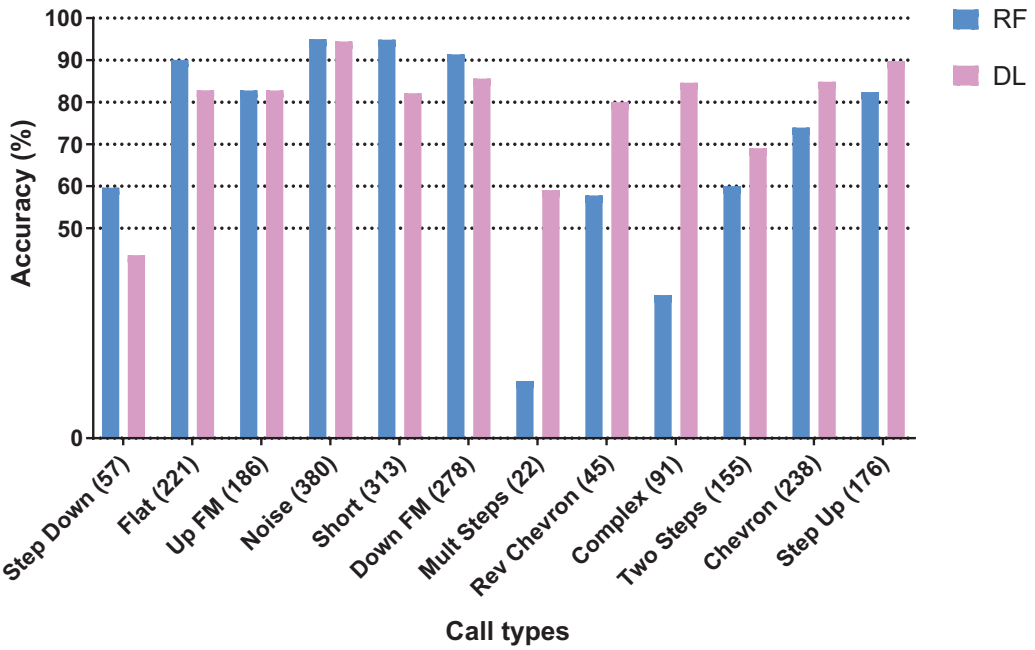
Given this dependence of Random Forest on the segmentation, we decided to test a second kind of classifier to make the output more reliable (greater ratio $P1/P2$). Then we moved to tests with a method less dependent on the segmentation: the CNNs.

5.3.2 Combining Random Forest and CNN

The implementation of the CNN has been discussed in Section 2.5.3. In a test made with 2169 USVs audio file, the performance per class is shown in Figure 5.8 for each one of the machines. The results show Random Forest performing better classification for Step down, Flat and Short USVs. Deep learning performed as well as the Random Forest for classifying Noise, Up-FM and DownFM, but showed better performance for Reverse chevron, Complex, Chevron, Two steps, Multiple steps and Step up. The number besides the name of the call in Figure 5.8 shows the number of USVs of each type used for the test.

The overall performance for Random Forest was 81.74% of accuracy while CNN had 83.45%. By considering the cases where at least one of the machine learning methods was able to classify the USV matching the manual classification, we would be able to

Figure 5.8: Comparing classification performance by Random Forest and CNN



Source: The author

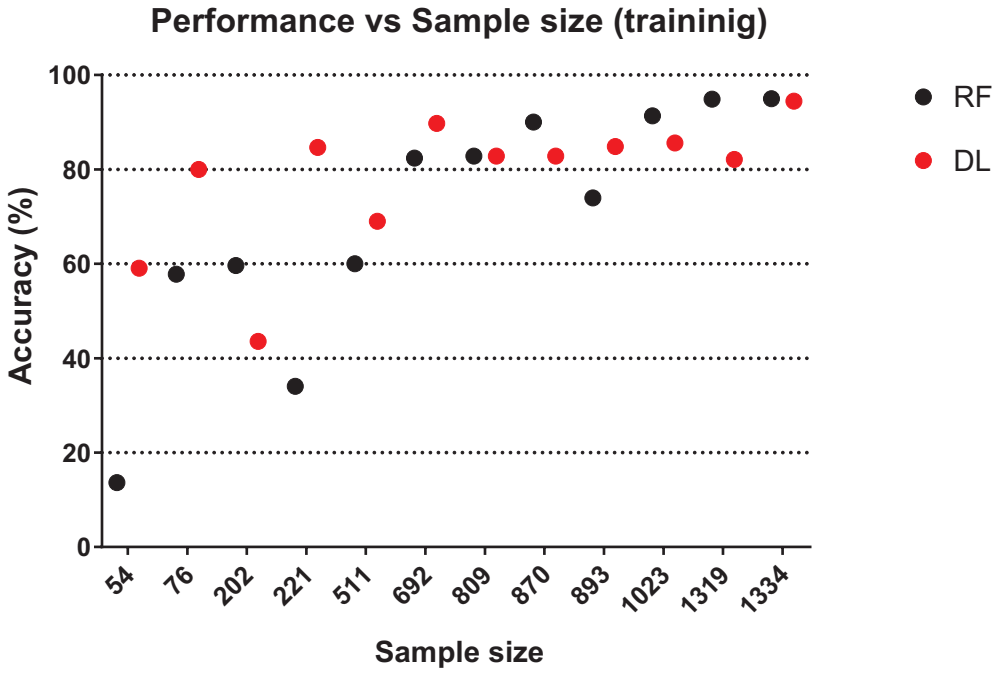
Figure 5.8: Performance per class for the two methods (Random Forest – RF, Deep Learning (CNN)– DP) when classifying the same data set (N=2169 USVs).

reach 95% of accuracy. This gave us reason to pursue a model to combine the output from the two methods.

The relative poor performance for Step down, Multiple steps and Complex can be explained by looking at the number of samples used for training the machines for each call type, as can be seen in Figure 5.9, which matches the theoretical learning curve described by Figueroa (FIGUEROA et al., 2012). The correlation between the performance of the machines and sample size used for training, which matches the performance seen in Figure 5.8, indicates that increasing the number of samples for the under represented call types might lead to an improvement on the machine's performance. However, this correlation also shows that increasing the number of samples seems to reach a saturation point, indicating the existence of an optimal number of samples for training, which in this case should be around 800 USVs per call type.

Developing a combinational model based on the probability density function given by two different sources is not trivial. A simplistic approach could be used, such as averaging the predictions for each class. However, such approach would not consider the individual performance of the classifier to handle samples from each class. As example, if we take the prediction of a multiple steps (Figure 5.8) call given by Random Forest

Figure 5.9: Performance by Random Forest and CNN as function of sample size



Source: The author

Figure 5.9: Correlation between sample size and performance of both machine learning methods testes (Random Forest – RF, Deep Learning – DP). It is possible to verify the importance of the sample sizing for a good classification performance.

and simply average with the one from Deep Learning, we would be considering that both methods have the same weight for making a decision about the final probability, which is visibly not true.

A second approach could be a weighted average of the individual probabilities, but choosing the weights also has to be done carefully and with statistical rigor in order to avoid weights taking your classifier to the wrong path. Fortunately, there are machine learning methods capable of estimating the appropriate weights for functions in order to minimize errors.

An efficient method to combine the output from our two methods is still under development, but as a first trial, a new Random Forest was created in order to combine the probability distribution from the Random Forest and Deep Learning implemented for syllable classifications. Therefore, the input for this new Random Forest is the probability distribution for the 12 different classes given by two different classifiers, totaling 24 predictors.

For a proof of concept, the samples used for testing the machines performance in the previous step (2169 USVs) were now used as training data for the new Random Forest

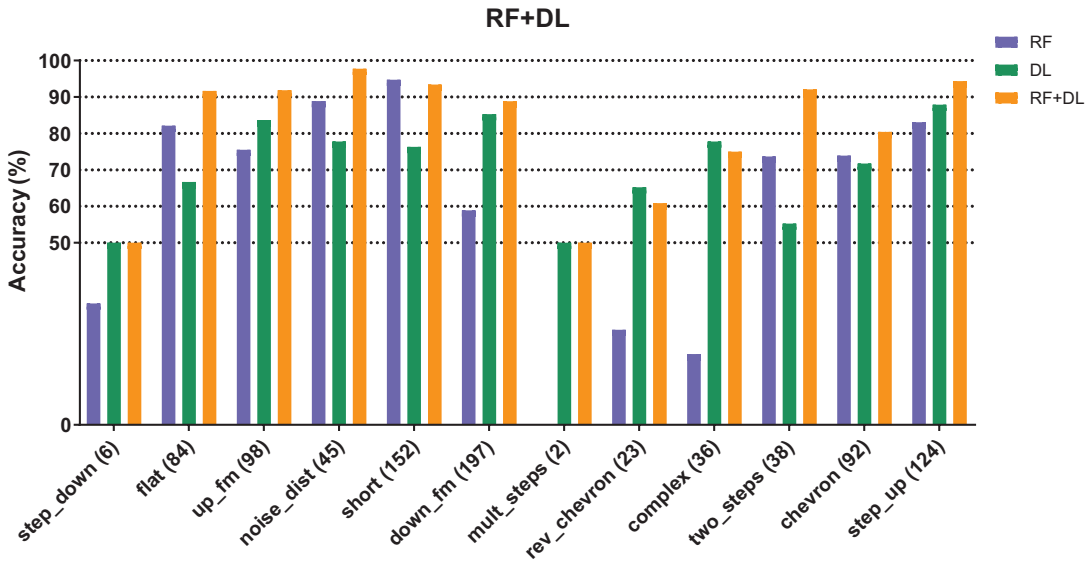
classifier. As seen before, Random Forest is able to estimate the importance of a predictor for a prediction, which can also be interpreted as the weight of the predictor and that is exactly what is necessary for our compositional data model.

A simple test using 913 new USVs recorded from pups in isolation shows how promising is this approach (Figure 5.10). Separately, Random Forest had 73.24% of accuracy in this classification process, while Deep Learning had 78.04% and the compositional model (a second Random Forest) had 89.08% of accuracy, which matches the expectation based on the observation of Figure 5.8 as its test associated.

Comparing to the performance obtained with the first classifier (just Random Forest) to performance of newest solution (Random Forest and Deep Learning), it is possible to see how the compositional method refines the original Random Forest approach, as seen in Figure 5.7C.

Despite the great performance observed for a small test data set here reported for Random Forest and its combination with CNN, later results showed that Random Forest was not efficient in extracting higher order of information from the USVs, such that the final label was still too attached to the segmentation result and too sensitive to noise. Thus, even the combination of Random Forest and CNN was not as great as the result seen for just CNN after further training, which resulted in Random Forest being removed from the main body of VocalMat.

Figure 5.10: Comparing performance by Random Forest, CNN and combination



Source: The author

Figure 5.10: Results obtained for the classification of 913 call into their respective types by the three classifiers: Random Forest – RF (73.24% of accuracy), Deep Learning (78.04%) and the compositional method RF+DL (89.08%)

REFERENCES

AGRESTI, A. **An introduction to categorical data analysis**. [S.l.]: Wiley, 2018.

ALDENDERFER, M. S.; BLASHFIELD, R. K. Cluster analysis sage publications. **Newbury Park, Cal**, 1984.

ARRIAGA, G.; JARVIS, E. D. Mouse vocal communication system: Are ultrasounds learned or innate? **Brain and Language**, Elsevier, v. 124, n. 1, p. 96–116, 2013.

ARRIAGA, G.; ZHOU, E. P.; JARVIS, E. D. Of mice, birds, and men: the mouse ultrasonic song system has some features similar to humans and song-learning birds. **PloS one**, Public Library of Science, v. 7, n. 10, p. e46610, 2012.

BOWMAN, A. W.; AZZALINI, A. **Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations**. [S.l.]: OUP Oxford, 1997.

BRADLEY, D.; ROTH, G. Adaptive thresholding using the integral image. **Journal of graphics tools**, Taylor & Francis, v. 12, n. 2, p. 13–21, 2007.

BREIMAN, L. Bagging predictors. **Machine learning**, Springer, v. 24, n. 2, p. 123–140, 1996.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BURKETT, Z. D. et al. Voice: a semi-automated pipeline for standardizing vocal analysis across models. **Scientific reports**, Nature Publishing Group, v. 5, p. 10237, 2015.

CASTELLUCCI, G. A.; MCGINLEY, M. J.; MCCORMICK, D. A. Knockout of foxp2 disrupts vocal development in mice. **Scientific reports**, Nature Publishing Group, v. 6, p. 23305, 2016.

CHABOUT, J. et al. Male mice song syntax depends on social contexts and influences female preferences. **Frontiers in behavioral neuroscience**, Frontiers, v. 9, p. 76, 2015.

COFFEY, K. R.; MARX, R. G.; NEUMAIER, J. F. Deepsqueak: a deep learning-based system for detection and analysis of ultrasonic vocalizations. **development**, v. 4, p. 21, 2019.

COIFMAN, R. R. et al. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 102, n. 21, p. 7426–7431, 2005.

D'AMATO, F. R. et al. Pups call, mothers rush: does maternal responsiveness affect the amount of ultrasonic vocalizations in mouse pups? **Behavior genetics**, Springer, v. 35, n. 1, p. 103–112, 2005.

EHRET, G. Infant rodent ultrasounds–a gate to the understanding of sound communication. **Behavior genetics**, Springer, v. 35, n. 1, p. 19–29, 2005.

FIGUEROA, R. L. et al. Predicting sample size required for classification performance. **BMC medical informatics and decision making**, BioMed Central, v. 12, n. 1, p. 8, 2012.

GOLUB, G.; KAHAN, W. Calculating the singular values and pseudo-inverse of a matrix. **Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis**, SIAM, v. 2, n. 2, p. 205–224, 1965.

GRIMSLEY, J. M.; MONAGHAN, J. J.; WENSTRUP, J. J. Development of social vocalizations in mice. **PloS one**, Public Library of Science, v. 6, n. 3, p. e17460, 2011.

HECKMAN, J. et al. Determinants of the mouse ultrasonic vocal structure and repertoire. **Neuroscience & Biobehavioral Reviews**, Elsevier, v. 65, p. 313–325, 2016.

HINTON, G. E. et al. Improving neural networks by preventing co-adaptation of feature detectors. **arXiv preprint arXiv:1207.0580**, 2012.

HO, T. K. Random decision forests. In: IEEE. **Proceedings of 3rd international conference on document analysis and recognition**. [S.l.], 1995. v. 1, p. 278–282.

HOFER, M. A. Hidden regulators in attachment, separation, and loss. **Monographs of the Society for Research in Child Development**, Univ of Chicago Press, 1994.

HOFFMANN, F.; MUSOLF, K.; PENN, D. J. Spectrographic analyses reveal signals of individuality and kinship in the ultrasonic courtship vocalizations of wild house mice. **Physiology & behavior**, Elsevier, v. 105, n. 3, p. 766–771, 2012.

HOLY, T. E.; GUO, Z. Ultrasonic songs of male mice. **PLoS biology**, Public Library of Science, v. 3, n. 12, p. e386, 2005.

HUANG, F. J. et al. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: IEEE. **Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on**. [S.l.], 2007. p. 1–8.

JARRETT, K. et al. What is the best multi-stage architecture for object recognition? In: IEEE. **Computer Vision, 2009 IEEE 12th International Conference on**. [S.l.], 2009. p. 2146–2153.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105.

LANGFELDER, P.; HORVATH, S. Eigengene networks for studying the relationships between co-expression modules. **BMC systems biology**, BioMed Central, v. 1, n. 1, p. 54, 2007.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, IEEE, v. 86, n. 11, p. 2278–2324, 1998.

MCLACHLAN, G. **Discriminant analysis and statistical pattern recognition**. [S.l.]: John Wiley & Sons, 2004.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th international conference on machine learning (ICML-10)**. [S.l.: s.n.], 2010. p. 807–814.

NEUNUEBEL, J. P. et al. Female mice ultrasonically interact with males during courtship displays. **Elife**, eLife Sciences Publications, Ltd, v. 4, 2015.

NYBY, J.; DIZINNO, G. A.; WHITNEY, G. Social status and ultrasonic vocalizations of male mice. **Behavioral biology**, Academic Press, 1976.

O'NEILL, B. Elementary differential geometry. In: ____. [S.l.]: Elsevier, 2006. p. 1–100.

PAN, S. J.; YANG, Q. A survey on transfer learning. **IEEE Transactions on knowledge and data engineering**, IEEE, v. 22, n. 10, p. 1345–1359, 2010.

PORTFORS, C. V. Types and functions of ultrasonic vocalizations in laboratory rats and mice. **Journal of the American Association for Laboratory Animal Science**, American Association for Laboratory Animal Science, v. 46, n. 1, p. 28–34, 2007.

REN, S. et al. Faster r-cnn: towards real-time object detection with region proposal networks. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, IEEE, n. 6, p. 1137–1149, 2017.

SCATTONI, M. L. et al. Unusual repertoire of vocalizations in the btbr t+ tf/j mouse model of autism. **PloS one**, Public Library of Science, v. 3, n. 8, p. e3067, 2008.

SCATTONI, M. L.; RICCERI, L.; CRAWLEY, J. N. Unusual repertoire of vocalizations in adult btbr t+ tf/j mice during three types of social encounters. **Genes, Brain and Behavior**, Wiley Online Library, v. 10, n. 1, p. 44–56, 2011.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural networks**, Elsevier, v. 61, p. 85–117, 2015.

SEGBROECK, M. V. et al. Mupet—mouse ultrasonic profile extraction: A signal processing tool for rapid and unsupervised analysis of ultrasonic vocalizations. **Neuron**, Elsevier, v. 94, n. 3, p. 465–485, 2017.

SEWELL, G. S. n. Ultrasound and mating behaviour in rodents with some observations on other behavioural situations. **Journal of Zoology**, Wiley Online Library, v. 168, n. 2, p. 149–164, 1972.

SIROTIN, Y. B.; COSTA, M. E.; LAPLAGNE, D. A. Rodent ultrasonic vocalizations are bound to active sniffing behavior. **Olfactory memory networks: from emotional learning to social behaviors**, Frontiers Media SA, 2015.

SLOBODCHIKOFF, C. et al. Size and shape information serve as labels in the alarm calls of gunnison's prairie dogs cynomys gunnisoni. **Current Zoology**, Oxford University Press Oxford, Uk, v. 58, n. 5, p. 741–748, 2012.

TCHERNICHOVSKI, O. et al. A procedure for an automated measurement of song similarity. **Animal behaviour**, Elsevier, v. 59, n. 6, p. 1167–1176, 2000.

TUIA, D.; CAMPS-VALLS, G. Kernel manifold alignment for domain adaptation. **PloS one**, Public Library of Science, v. 11, n. 2, p. e0148655, 2016.

VLADIŠAUSKAS, A.; JAKEVIČIUS, L. Absorption of ultrasonic waves in air. **Ultragarsas**, v. 50, n. 1, p. 46–49, 2004.

WANG, C.; MAHADEVAN, S. Heterogeneous domain adaptation using manifold alignment. In: **Twenty-Second International Joint Conference on Artificial Intelligence**. [S.l.: s.n.], 2011.

ZIMMER, M. R. et al. Functional ontogeny of hypothalamic agrp neurons in neonatal mouse behaviors. **Cell**, Elsevier, 2019.