DeepSqueak 1.0 was designed and tested with MATLAB 2017b. To run DeepSqueak, navigate to the main DeepSqueak folder in MATLAB, and type "DeepSqeak" into the command line. DeepSqueak will add itself to the MATLAB path after running.

Main Window



Time (s)

合 🥼 🕪

inferno

0

Selection Review >>

2

3

4

5

1. Call Statistics

• See Export to Excel

2. Extracted Contour

• Call <u>contour</u>, slope line.

3. Spectral Gradient of spectrogram

• Looks neat.

4. Tonality and Soundwave

• Tonality vs time, overlaid on sound wave.

• Yellow regions are above the <u>tonality threshold</u>. **5. Position in File**

USV Detection

Before vocalizations can be detected, an <u>audio folder</u>, neural network folder, and <u>output folder</u> must be selected.

After detecting calls, we recommend using a <u>post hoc denoising network</u> to remove false positives.

To process a single file:

- 1. <u>Select the desired audio file in the "Audio Files" drop-down menu</u>
- 2. Select an appropriate neural network in the "Neural Networks" dropdown menu
- 3. Click "Detect Calls"
- 4. Enter detection settings as described below, and click "OK"

To process files in a batch, or to process a single file with multiple networks:

- 1. Click "Multi Detect"
- 2. A list of the audio files in the current folder will appear. Select the desired audio files. Multiple files may be selected by holding the Ctrl key while clicking. Click "OK" to proceed.
- 3. After selecting the audio file(s), a box will appear, listing available neural networks. Select a maximum of two, and click "OK".
- 4. Enter detection settings as described below, and click "OK".

Detection Settings:

- Total Analysis Length
 - Length, in seconds, of the audio file to process.
 - Set to 0 to process the entire file
 - If analysis length is greater than the file duration, a warning will be displayed in the command line, and the entire file will be processed
- Analysis Chunk Length
 - Length of the audio sections to process at a time, in seconds
 - Files are processed in short chunks. The maximum length of each chunk is dependent on available GPU memory, frequency

cutoffs, and the neural network

- We found that a GPU with two gigabytes of memory performs well with four second chunks for short rat and mouse calls, and fifteen second chunks for long rat calls
- If the GPU runs out of memory, a warning message will be displayed in the command line
- Overlap
 - Amount of overlap between audio chunks, in seconds
 - Value should be about the length of a call
- Frequency Cut Off High
 - Regions of the spectrogram above this value are ignored
- Frequency Cut Off Low
 - Regions of the spectrogram below this value are ignored
- Score Threshold
 - Detected items with a score (likelihood of being a hit) below this value are automatically removed
- Power Threshold
 - Detected items with an amplitude below this value are automatically removed
- Append Date to File Name
 - If value is "1", the detection time will be appended to the end of the file name

To load audio, select the folder where the audio is stored by selecting "**File > Select Audio Folder**".

If the folder is successfully loaded, the audio files will appear in the "Audio Files" drop down menu.

DeepSqueak is capable of reading WAV (*.wav), .FLAC (*.flac), and Ultravox (*.UVD) audio files.

DeepSqueak was tested with a sampling frequency of 250 kHz; however, the spectrograms are created using fft windows of constant duration, rather than constant sample numbers, so other sample rates are accepted.

Detection files are saved in a folder specified in "File > Select Detection Folder".

To convert detection files to other formats, or export call statistics, see <u>Import & Export</u>

Detected calls are saved as a MATLAB structure with the following fields:

- Rate:
 - Audio sample rate
- Box
 - Position of the call in the audio file
 - 1x4 matrix, [Begin Time (s), Minimum Frequency (kHz), Duration (s), Frequency Range (kHz)]
- RelBox
 - Position of the call in the Audio field
 - 1x4 matrix, [Begin Time (s), Minimum Frequency (kHz), Duration (s), Frequency Range (kHz)]
- Score
 - Neural network score
- Audio
 - Audio containing the call
- Accept
 - Status of call
 - 1 is accepted, 0 is rejected
- Type
 - Call category
- Power
 - Amplitude of call

Training Detection Networks

USV Detection >>

You probably won't need to do this.

To change the network architecture, edit "TrainSqueakDetector.m" **To create an image database for training a faster-RCNN detector:**

合 🌰 🗅

- 1. Select "Tools > Network Training > Create Training Data"
- 2. In the dialog box, select all files to create images from.
- 3. Enter spectrogram settings.
 - FTT windows length, overlap, and NFFT are specified in seconds.
 - Amplitude cutoff: all values above this are set to one.
 - Bout length: calls within this distance are placed into a single image.
 - If value is not equal to zero, only a single file can be processed at a time.

To train a faster-RCNN detector:

- 1. Select "Tools > Network Training > Train Network"
- 2. In the dialog box, select all training tables from which to train from (saved in "DeepSqueak\Training\").
- 3. Decide whether or not to use a pre-trained network as the starting point.
- 4. Training will take hours. When finished a save dialog will appear.

After Calls have been detected, the user may manually or automatically classify calls and remove non-calls.



Manual Selection Review

Loading Call Files

- 1. Select a folder with detection files from "File > Select Detection Folder"
- 2. Select a file in the "Detected Call Files" drop down menu
- 3. Click "Load Calls"

Navigation

To view the next call in the file, use the left and right arrow keys, "q" and "e" keys, and/or the "Previous Call" and "Next Call" buttons.

Drag the scroll bar above the spectrogram to jump to a position in the file. After clicking on the scroll bar, the arrow keys will also move the slider (to prevent this, click somewhere else on the main window).

Redraw Boxes

To redraw the box, press "d" or click the "Redraw" button, and drag the mouse to create a box over the desired region.

Call Classification

Accept or reject calls with "a" and "r", or the "Accept Call" and "Reject Call" buttons.

Define call categories with "Tools > Call Classification > Add Custom Labels".

Use keys 1-9 to apply a category to a call.

Play Calls

Play the current call through the default sound device by pressing "p", or the "Play Call" button.

Playback rate may be changed under "Tools > Change Playback Rate".

Automatic Selection Review

Change Score Threshold:

- 1. Load a call file.
- 2. Select "Tools > Automatic Review > Change Score Threshold"
- 3. Enter a score threshold, and click "OK". All calls below this threshold will be rejected.

Change Power Threshold:

- 1. Load a call file.
- 2. Select "Tools > Automatic Review > Change Score Threshold"
- 3. Enter a score threshold, and click "OK". All calls below this threshold will be rejected.

Remove Rejected Calls

- 1. Load a call file.
- 2. Select "Tools > Automatic Review > Remove Rejected Calls"
- 3. All rejected calls will be removed from the call file.

Post Hoc Denoising / False Positive Removal

Supervised Classification

Unsupervised Classification

Keyboard Shortcuts:	
Action:	Key:
Play Call (rate can be changed in Tools > Change Playback Rate)	р
View Next Call	right arrow, e
View Previous Call	left arrow, q
Accept Call	a
Reject Call	r
Redraw Box	d
Classify Call (change categories in Tool > Call Classification > Add Custom Labels)	1-9

Copyright $\ensuremath{\mathbb{C}}$ 2018 by Russell Marx & Kevin Coffey. All Rights Reserved.

🚖 🌗 📫



Call statistics, as well as unsupervised clustering, are calculated on spectrotemporal contours.

The contour is extracted by taking the maximum intensity at each time point, where both tonality and amplitude exceed a set threshold.

The tonality and amplitude thresholds may be changed in "Tools > Change Contour Threshold".

Higher values of tonality and amplitude will result in more conservative contours.

Copyright © 2018 by Russell Marx & Kevin Coffey. All Rights Reserved.

Post Hoc Denoising

Selection Review >>

Although the call detection network will reject many non-vocal noise events, we have tuned to value sensitivity over precision.

False positives may be automatically identified and rejected using a post hoc neural network. The network must be located in

"DeepSqueak\Denoising Networks\CleaningNet.mat".

To use the post hoc denoiser:

- 1. Select "Tools > Automatic Review > Post Hoc Denoising"
- 2. In the list box, select the detection files to denoise. All noise events found will be classified as "Noise" and rejected.

To train a post hoc denoiser:

- 1. Select "Tools > Network Training > Train Post Hoc Denoiser"
- 2. In the list box, select the detection files to to use for training.
 - Calls labeled as "Noise" are used as negative training sample.
 - Accepted calls not labeled as "Noise" are used as positive training samples.
- 3. When training is finished, the new network will be automatically saved as "DeepSqueak\Denoising Networks\CleaningNet.mat".
 - Training will overwrite the older network. It is wise to create a backup of the old network.

Call Classification

In addition to <u>manually classifying</u> calls, DeepSqueak includes two automated methods.

<u>Unsupervised clustering</u> applies featured-based machine learning with *k*-means to cluster calls, by minimizing the variance between a call's features and the nearest prototype cluster.

<u>Supervised classification</u> uses a convolution neural network to classify calls based on the spectrogram.

We've found that creating clustering with unsupervised methods, and using the cleanest clusters to train a supervised classification network, resulted in fast and accurate clustering.

Clusters may be viewed and renamed with "Tools > Call Classification > View Clusters"

Unsupervised Clustering

Call Classification >>

The unsupervised clustering function uses *k*-means on perceptually relevant dimensions of the <u>extracted contour</u>, to place calls into a predefined number of clusters.

Each call is segmented into six partitions. The *k*-means algorithm operates on the slope and frequency of each partition, as well as the sinuosity of the first and second half of the call, and the call duration.

To perform unsupervised clustering using *k*-means:

- 1. Click "Tools > Call Classification > Unsupervised Clustering"
- 2. Select the detection files to cluster OR select the saved contours.
- 3. After the detection files are processed, you may save the extracted contours for faster loading.
- 4. Choose the clustering method. ARTwarp is still experimental, so *k*-means is currently recommended.
- 5. Enter the weights (relative importance) of each dimension.
- 5. When asked whether to use an existing model, click "No".
- 7. Enter the number of call categories. Our work-flow involved producing more clusters than desired, and training a <u>supervised neural network</u> on the best clusters.
- 3. Once clustering finishes, you will be prompted to save the model. This is optional.
- A new interface will appear, showing the clusters. This interface can also be found under "Tools > Call Classification > View Clusters"
 - Name the clusters by entering a name in the text box. Clusters with the same name will be merged upon saving.
 - View different clusters with the "Next" and "Back" buttons.
 - View more calls within a cluster with the "Next Page" and "Previous Page" buttons.
 - **Reject calls by clicking on them**. Calls highlighted in red will be rejected upon saving.
 - Update the call files by clicking "Save", or redo the clustering with "Redo"

Supervised Classification

Call Classification >>

Calls may be classified with a supervised neural network. This network operates on the spectrogram, rather than the contour.

合 🌰 🗅

To use a supervised classifier:

- 1. Select "Tools > Call Classification > Supervised Classification"
- 2. In the list box, select the detection files to classify, and click "OK".

To train a supervised classifier:

- 1. Select "Tools > Network Training > Train Supervised Classifier"
- 2. Select the detection files to to use for training.
 - Rejected calls and "Noise" labeled calls will be ignored.
- 3. When training is finished, the new network will be automatically saved as "DeepSqueak\Denoising Networks\CleaningNet.mat".
 - Training will overwrite the older network. It is wise to create a backup of the old network.
- 4. Specify the spectrogram frequency range, and click "OK".
- 5. Save the network when training finishes.

After classifying calls, you may visualize the syntax graphs, and save the transition probability matrix.

- 1. Click "Tools > Call Classification > Syntax Analysis"
- 2. In the dialog box, select the files you wish to use.
- 3. A dialog box will appear. Enter the maximum call separation, in seconds, to define a bout, and a threshold for excluding uncommon call categories.
- 4. After the files are loaded, a transition matrix, and syntax graph will appear.
- 5. A dialog box will appear, giving the option to save the transition matrix and syllable counts as an Excel table.

Other Tools

(11)

Merge Detection Files Seperate Long 22s

Copyright © 2018 by Russell Marx & Kevin Coffey. All Rights Reserved.

Merge Detection Files

Other Tools >>

合 🐢 📫

Sometimes, such as when creating ground truth tables, or using multiple networks, it is useful to merge detection files.

To merge files:

- 1. Select "Tools > Merge Detection Files"
- 2. Select all files to merge, with Ctrl-Click.
- 3. Select the corresponding audio file.

Copyright © 2018 by Russell Marx & Kevin Coffey. All Rights Reserved.

Separate Long 22s

Other Tools >>

Particularly for rat long 22s, the detection algorithm may place multiple calls into a single box.

To separate these calls, we apply *k*-means to the tonality, and create new boxes from unbroken regions of high tonality.

To separate calls:

- 1. Select "Tools > Separate Long 22s"
- 2. In the dialog box, select the detection file.
- 3. In the dialog box, select the correspond audio file.

Import & Export

DeepSqueak can export call annotations and statistics to Excel files for further analysis. <u>See Export to Excel.</u> DeepSqueak can also export spectrograms and audio.

Detected calls may be viewed in the context of an audio file in Raven Lite, by exporting a detection file as a Raven log.

DeepSqueak is capable of importing file annotations from Raven, Ultravox, and MUPET. It is useful for comparing analysis packages and creating ground truth tables.

Copyright © 2018 by Russell Marx & Kevin Coffey. All Rights Reserved.

Import & Export >>



To export call statistics to an Excel file:

- Select a folder containing detection files with "File > Select Detection Folder"
- 2. Click "File > Import / Export > Export to Excel Log (Call Statistics)"
- 3. In the list box, select the files to be exported, and click "OK"
- 4. Specify whether or not to include rejected calls
- 5. Select the folder to place the Excel files

The following call statistics are included in the output file:

- ID
 - Call ID
- Label
 - Call category
- Accepted
 - 1 is accepted, 0 is rejected
- Score
 - Neural network score
- Begin Time (s)
 - Calculated from contour
- End Time (s)
 - Calculated from contour
- Call Length (s)
 - End Time Begin Time
- Principle Frequency (kHz)
 - Median frequency of the contour
- Low Freq (kHz)
 - Lowest frequency of the contour
- High Freq (kHz)
 - Highest frequency of the contour
- Delta Freq (kHz)
 - High Freq Low Freq
- Frequency Standard Deviation (kHz)
 - Standard Deviation of the contour

- Slope (kHz/s)
 - Slope of the contour
- Sinuosity
 - Length of the path between the first and last points on the contour, divided by the euclidean distance between the first and last points
- Max Power
 - Average power of the contour
- Tonality
 - Values near one are more tonal; lower signal to noise ratio
 - One minus the geometric mean of the spectrogram, divided by the arithmetic mean
 - One minus spectral flatness, or tonality coefficient, or Wiener entropy