HybridMuse

▼ About

Mice use ultrasonic vocalizations (USVs) to convey a variety of socially relevant information. These vocalizations are affected by the sex, age, strain, and emotional state of the emitter and can thus be used to characterize it. Current tools used to detect and analyze murine USVs rely on user input and image processing algorithms to identify USVs, requiring ideal recording environments. More recent tools which utilize convolutional neural networks (CNN) models to identify vocalization segments perform well above the latter but do not exploit the sequential structure of audio vocalizations. On the other hand, human voice recognition models were made explicitly for audio processing; they incorporate the advantages of CNN models in recurrent models that allow them to capture the sequential nature of the audio. Here we describe the Hybrid CNN-BiLSTM Mouse USV Extractor (HybridMuse) software, an audio analysis tool that combines convolutional (CNN) and recurrent (RNN) neural networks for automatically identifying, labeling, and extracting recorded USVs. Following training on manually labeled audio files recorded in various experimental conditions, HybridMuse outperformed the most commonly used benchmark model utilizing deep-learning tools in accuracy and precision. Moreover, it does not require user input and produces reliable detection and analysis of USVs recorded under harsh experimental conditions. We suggest that HybrideMuse will enhance the analysis of murine USVs and facilitate their use in scientific research.

▼ Installation

HybridMuse model is a novel neural network based on CNN-LSTM architecture. However, some of the features that support the automatic detection, training, and browsing rely on built-in functions that were introduced in later versions of MATLAB (2020a and later).

The next Matlab toolboxes are required:

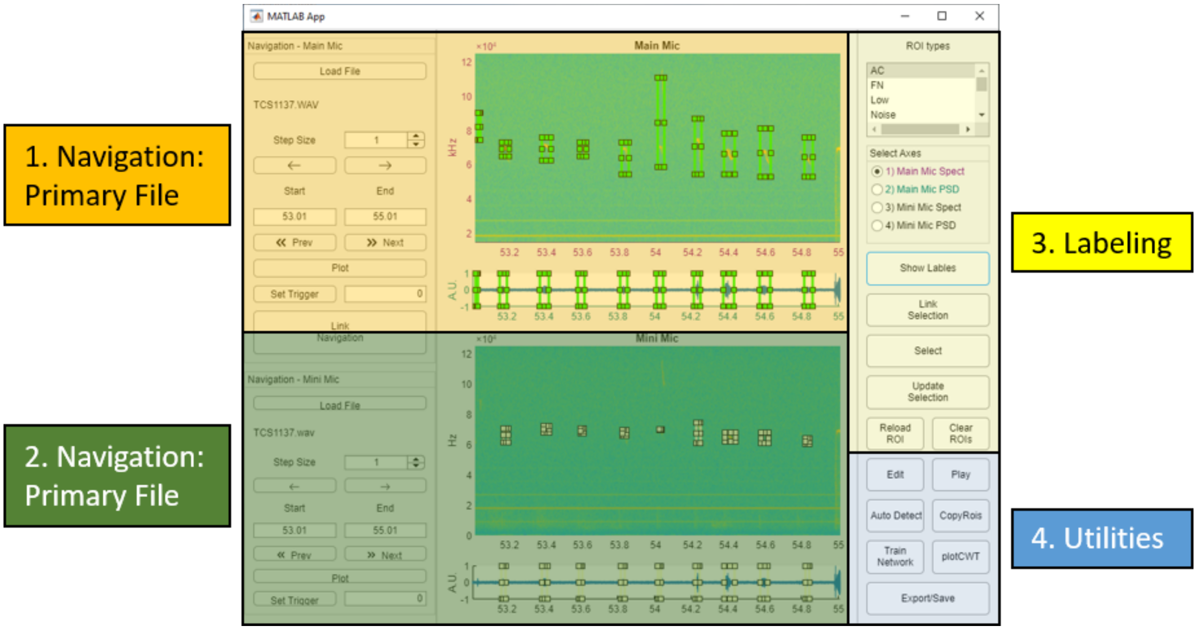
- Signal Processing Toolbox
- Deep Learning Toolbox
- Image Processing Toolbox

- Statistics and Machine Learning Toolbox
- Wavelet Toolbox
- Curve Fitting Toolbox
- Audio Toolbox
- Robust Control Toolbox
- Computer Vision Toolbox

To install HybridMuse:

1. Download HybridMuse.zip.
2. Extract the zip file.
3. Navigate to the extracted folder.
4. Type "HybridMuse" into the command line.

▼ Main Window



The Main Window can be divided into 4 panels: primary and secondary files navigation panels, labeling panel, and utility panel.

Navigation Panels:

- Load files - from disk as a .wav or .mat files or from workspace
- Browse files - slide over an audio file or jump to the next detected USV
- Load a secondary file recorded simultaneously to compare
- Set trigger times and link (synchronize) navigation between the two files
- Axes - There are four axes, two to each file. The upper axes is a spectrogram of the audio file and the lower axes is a waveform.

Labeling Panel:

- Manually label USVs or adjust automatically labeled USVs.
- Resize, remove, reload or rename USVs

Utilities Panel:

- Edit - Fine-tune the spectrogram and image parameters
- Play - Play the audio file and listen to the vocalizations
- Auto Detect - run the HybridMuse model and on single files or folders and detect USVs
- Train Network - Fine-tune the detection network by retraining it with labels data
- plotCWT - Plot a CWT to better represent data with a wider range of important frequencies
- Export - Export the audioClip objects or the figures to the workspace, or save them to disk

▼ The audioClip objects

To organize the information on each audio file, we implemented a custom class names audioClip that contains properties and methods for the audio files.

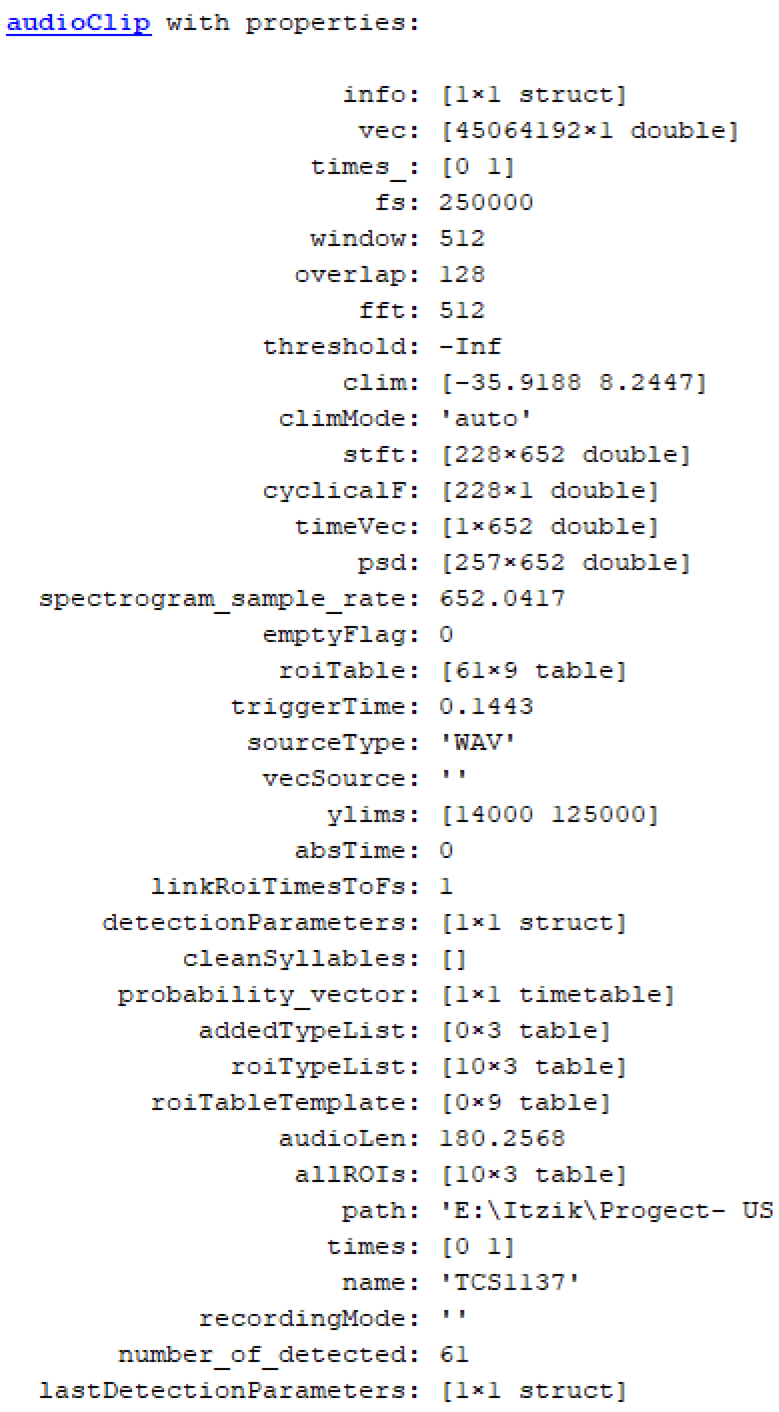
These include the audio vector and the path to the file, a table with detected USVs, customized labels, and more.

This is why audioClip is a handle class, meaning that you can pass it into functions without duplicating the variable, this is important since instances of this class may contain large audio files.

You can lunch the HybridMuse app (after installation and first activation) by loading or creating the audioClip object and calling the "explore" method.

You can also lunch the HybridMuse app by calling one or two (primary and secondary) audioClip objects as arguments:

```
HybridMuse(obj1, obj2)
```

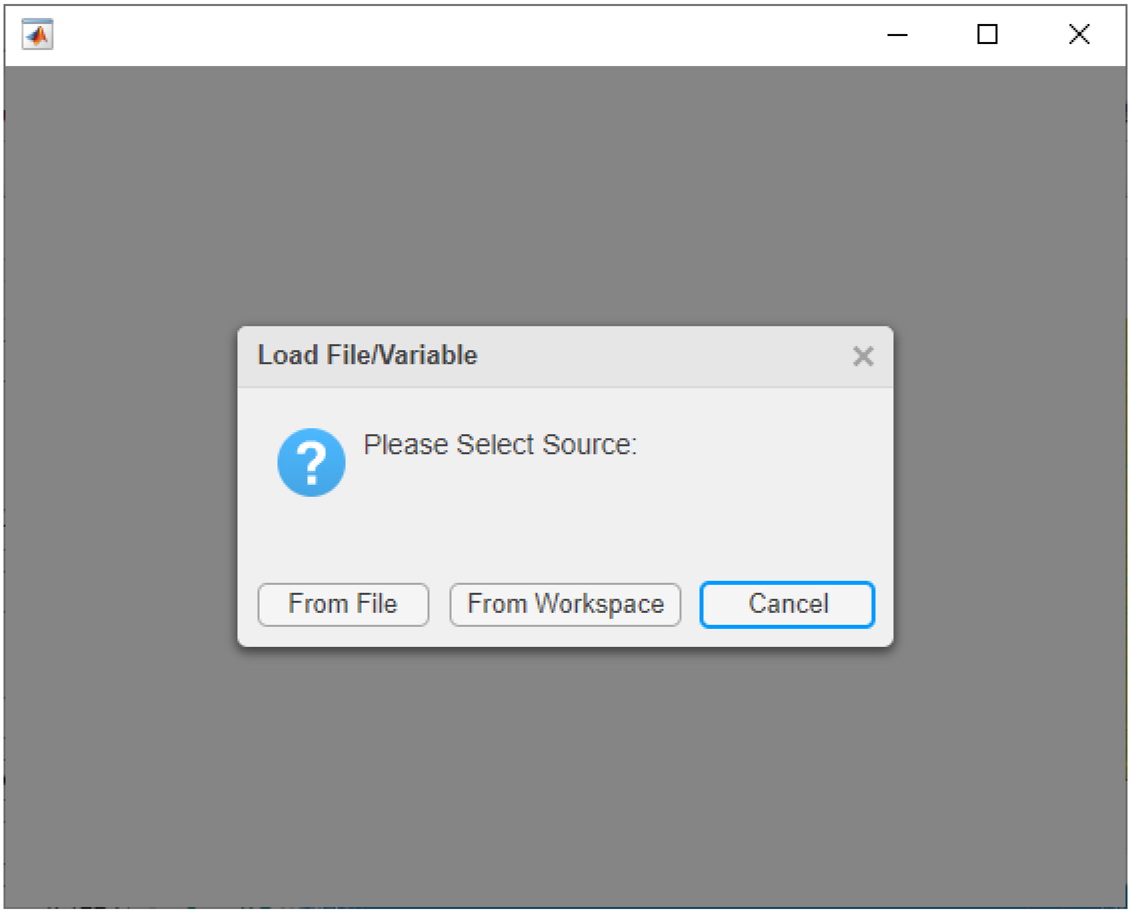


```
  audioClip with properties:

                       info: [1×1 struct]
                        vec: [45064192×1 double]
                     times_: [0 1]
                         fs: 250000
                     window: 512
                    overlap: 128
                        fft: 512
                  threshold: -Inf
                       clim: [-35.9188 8.2447]
                   climMode: 'auto'
                       stft: [228×652 double]
                  cyclicalF: [228×1 double]
                    timeVec: [1×652 double]
                        psd: [257×652 double]
    spectrogram_sample_rate: 652.0417
                  emptyFlag: 0
                   roiTable: [61×9 table]
                triggerTime: 0.1443
                 sourceType: 'WAV'
                  vecSource: ''
                      ylims: [14000 125000]
                    absTime: 0
           linkRoiTimesToFs: 1
        detectionParameters: [1×1 struct]
             cleanSyllables: []
         probability_vector: [1×1 timetable]
              addedTypeList: [0×3 table]
                roiTypeList: [10×3 table]
           roiTableTemplate: [0×9 table]
                   audioLen: 180.2568
                    allROIs: [10×3 table]
                       path: 'E:\Itzik\Progect- US
                      times: [0 1]
                       name: 'TCS1137'
              recordingMode: ''
         number_of_detected: 61
    lastDetectionParameters: [1×1 struct]
```

```
HybridMuse(obj1)
HybridMuse(obj1, obj2)
obj1.explore
```

▼ Load files



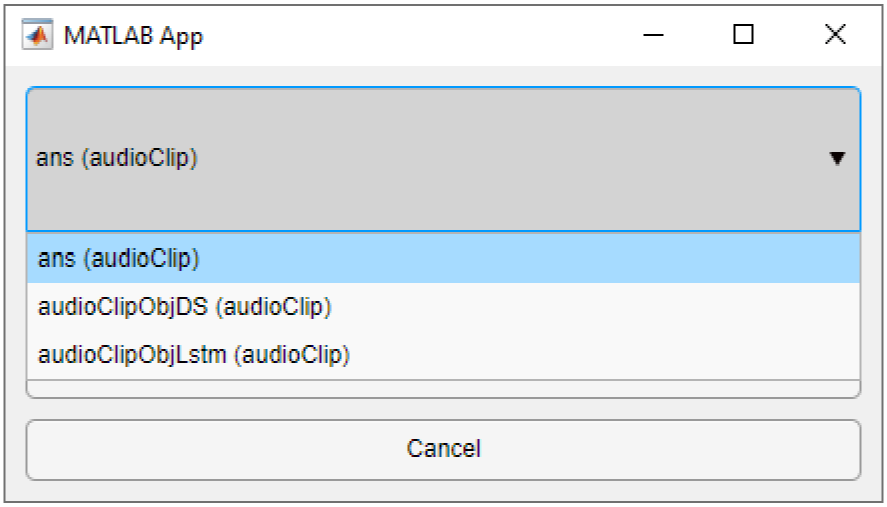
Load File/Variable

Please Select Source:

You can load audio from file or from the workspace.

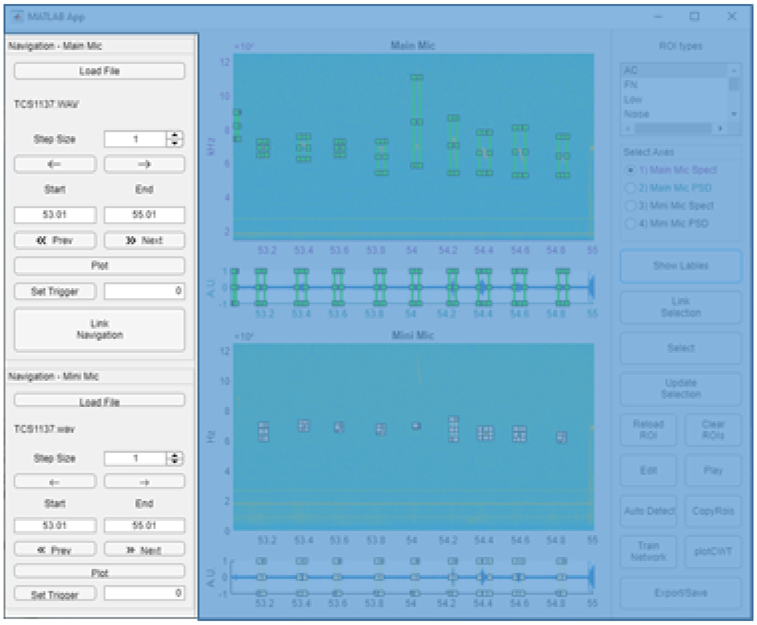
To load from file, click on "From File" and select a suitable file; the files must be .wav or .mat files. If the files are .mat files, they must contain a variable of class "double" or "audioClip".

To load a file from the workspace, click "From Workspace" button and select the variable from the list.



▼ Browse Files

▼ Navigation



Navigation - Main Mic

TCS1137.WAV

Step Size: 1

Start: 53.01 End: 55.01

Set Trigger: 0

Navigation - Mini Mic

TCS1137.wav

Step Size: 1

Start: 53.01 End: 55.01

Set Trigger: 0

After loading the files, you can browse through them using the navigation panels.

If you loaded an audioClip object, you will also see the automatically or the manually labeled USVs (if there are any).

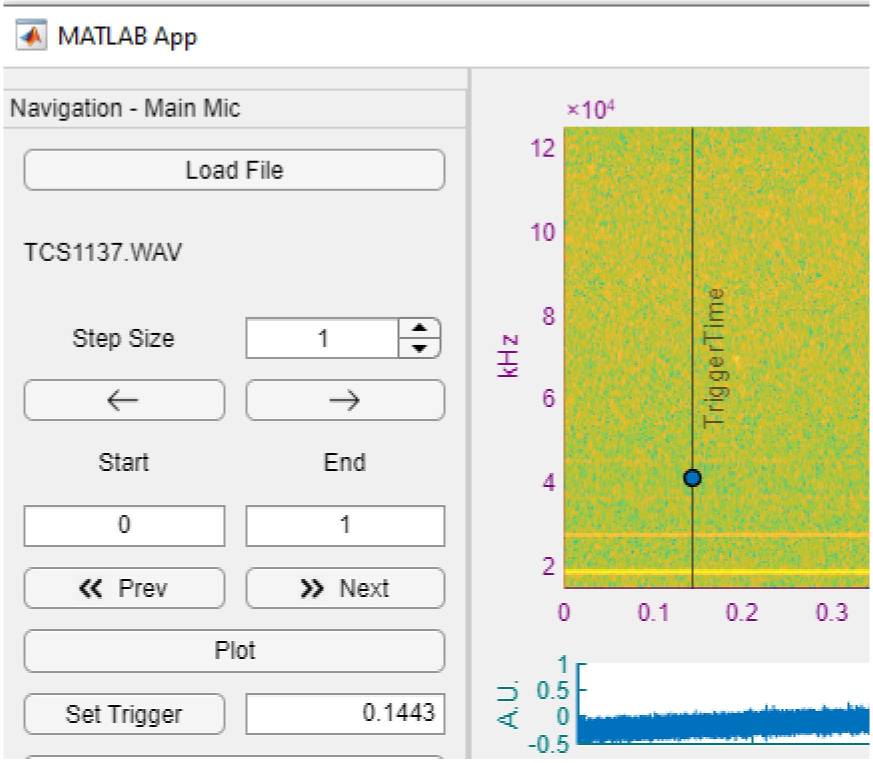
To browse the files, set the desired step size and click on the left or right arrows. You can also jump to any part of the file by editing the start and end boxes and clicking "plot" to update the values.

You can also jump to the next or previous USVs by clicking on the right or left double-arrows. Set the width of the displayed USV by right-clicking on either of the double-arrows and selecting the "change jump range" option.

▼ Primary and Secondary audio files

You can load a secondary audio file or other time-series file and synchronize the two signals.

If the recording did not start at the same time, you must either crop one of the signals before loading or set its trigger time to reflect the difference between the two signals. This can be easily done by editing the numeric text box or by dragging the "Trigger time" line.



TCS1137.WAV

Step Size 1

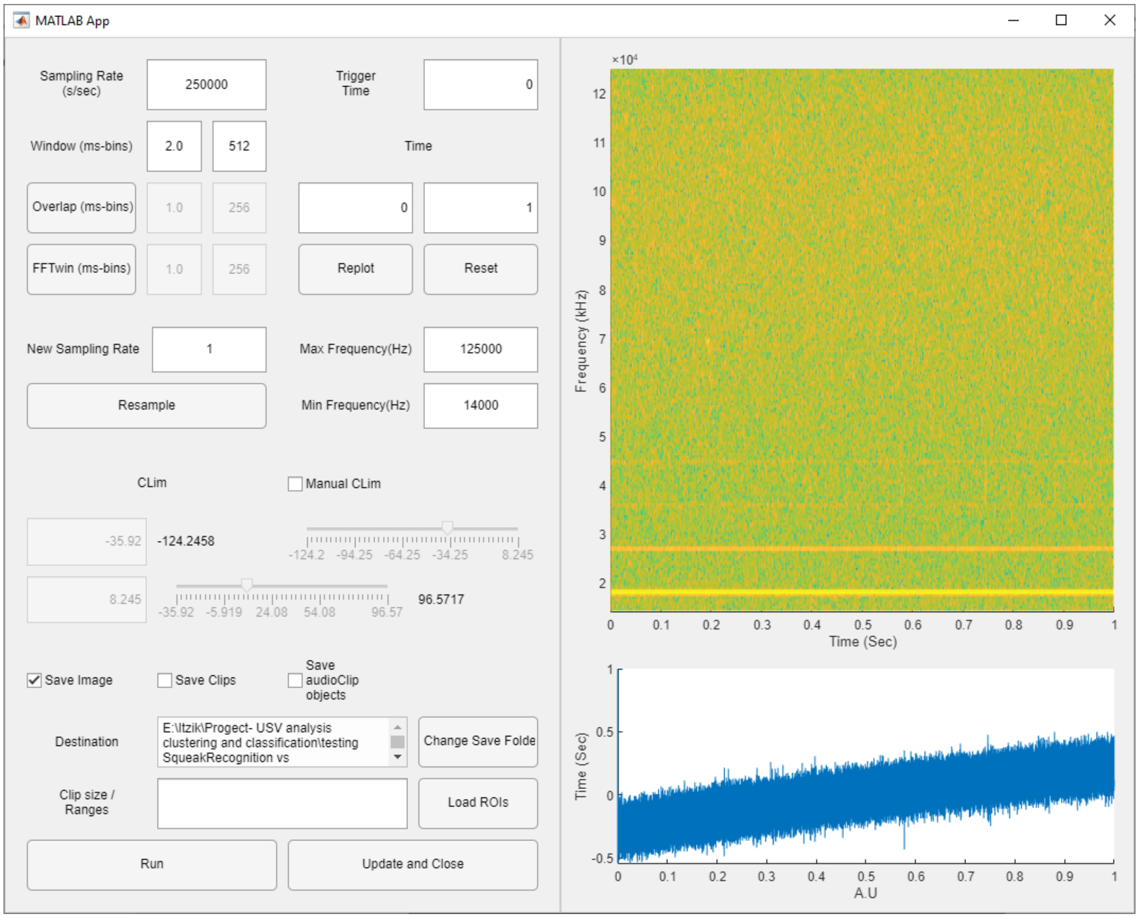
Start 0

End 1

Set Trigger 0.1443

▼ Refining the spectrogram

You can refine the spectrogram by changing the frequency boundaries (YLim) and color boundaries (CLim) of the image. This can be done by the "**Edit**" button in the "utilities panel".



MATLAB App

Sampling Rate (s/sec): 250000

Trigger Time: 0

Window (ms-bins): 2.0 512

Time

Overlap (ms-bins): 1.0 256

0 1

FFTwin (ms-bins): 1.0 256

New Sampling Rate: 1

Max Frequency(Hz): 125000

Min Frequency(Hz): 14000

CLim

□ Manual CLim

-35.92 -124.2458

8.245 96.5717

☑ Save Image □ Save Clips □ Save audioClip objects

Destination: E:\Itzik\Progect- USV analysis clustering and classification\testing SqueakRecognition vs

Clip size / Ranges

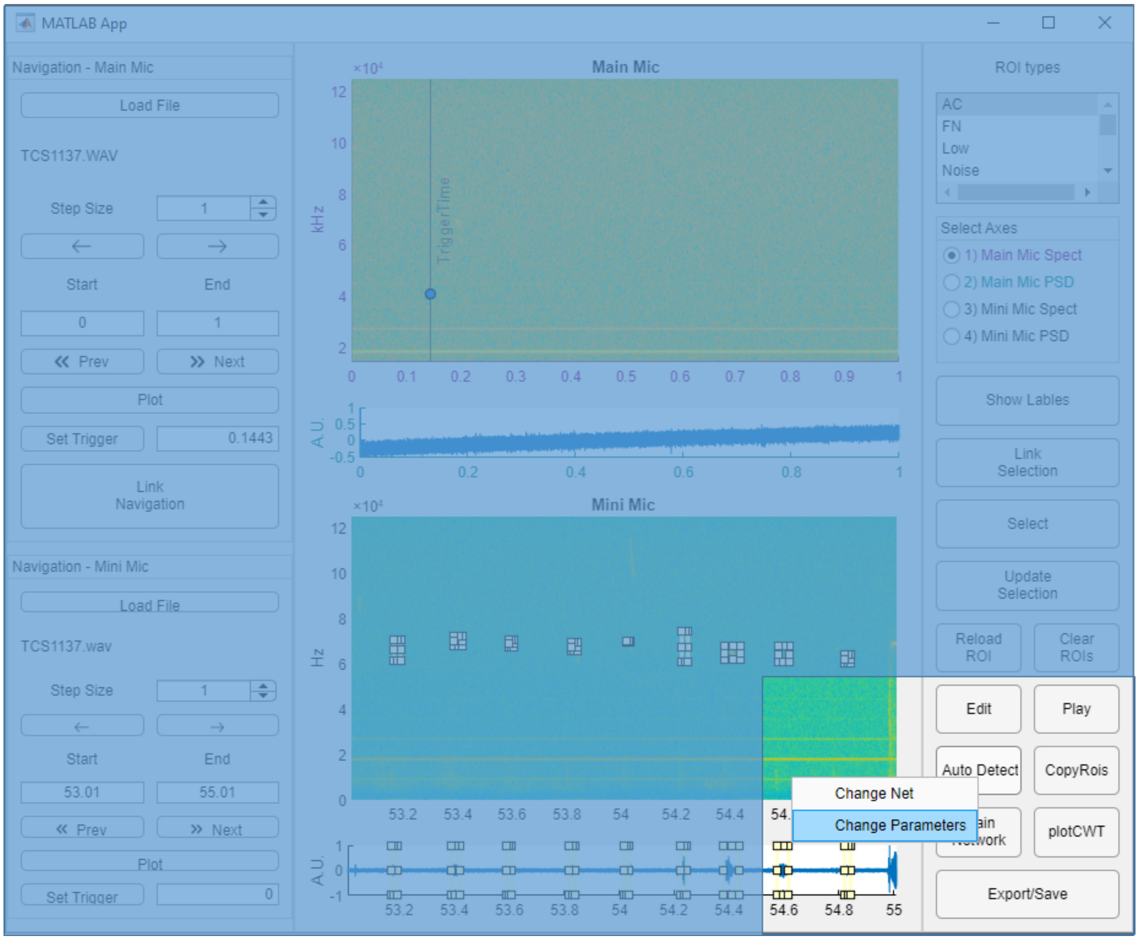
Here you can also change the spectrogram parameters such as window length, overlap, and more.

Finally, if the audio file is too large and your machine is unable to handle it, you can use the bottom part to split the file into smaller audio files. You can also create example images of the detected USVs and export them here.

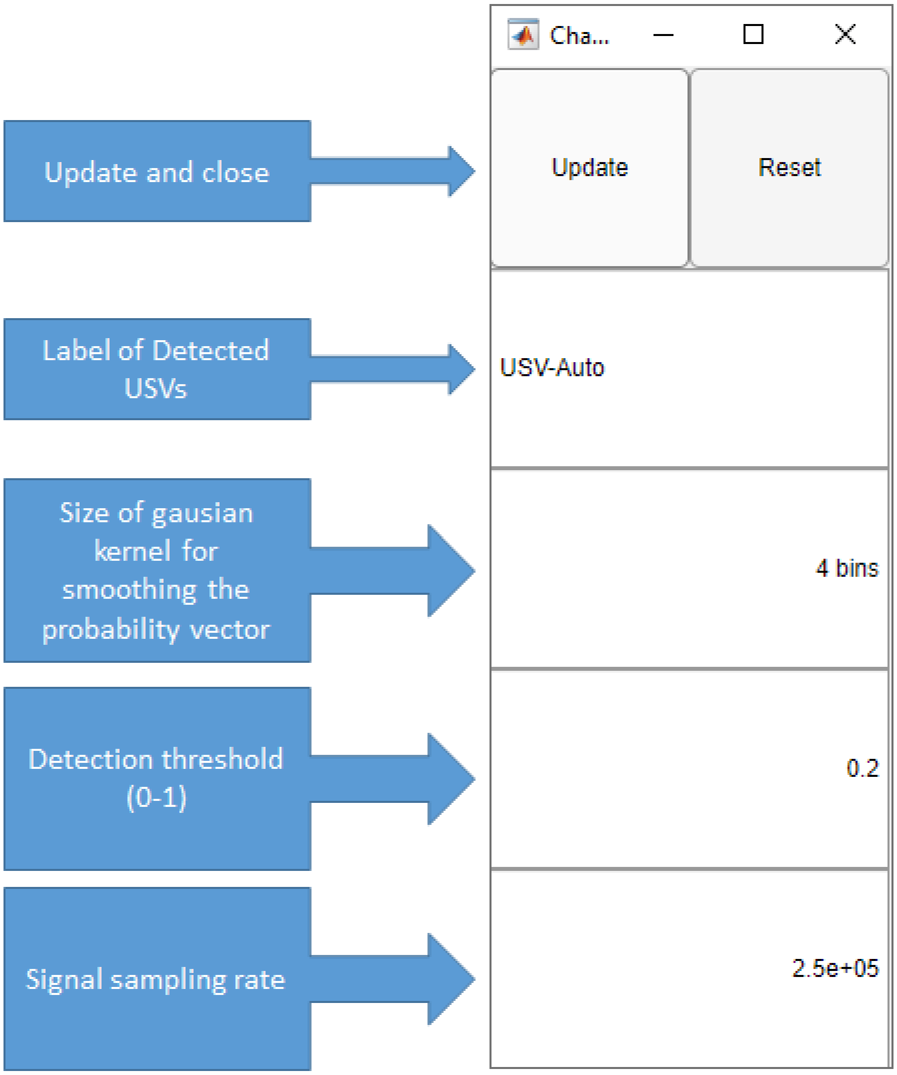
After you have done with the setting, click "Update and Close" or simply close the window before returning to the HybridMuse app.

▼ Detect USVs

▼ Setting detection parameters



You must set the detection parameters. HybridMuse was designed to minimize the dependencies on user input and adjustments. Therefore the default parameters should in work in most cases, however, you can change the detection threshold and the size of the Gaussian kernel used to smooth the prediction vector.



Label of Detected USVs
USV-Auto

Size of gausian kernel for smoothing the probability vector
4 bins

Detection threshold (0-1)
0.2

Signal sampling rate
2.5e+05

You can change the labels which the network will assign to the detected USVs. It is recommended to first add the new label and select a color and add a description to the label (See edit labels in "Label USVs" section)

Next, make sure that the sampling rate is correct. The model was tested on 2.5kHz sampling rate.

Finally, press the Update button to close the window.

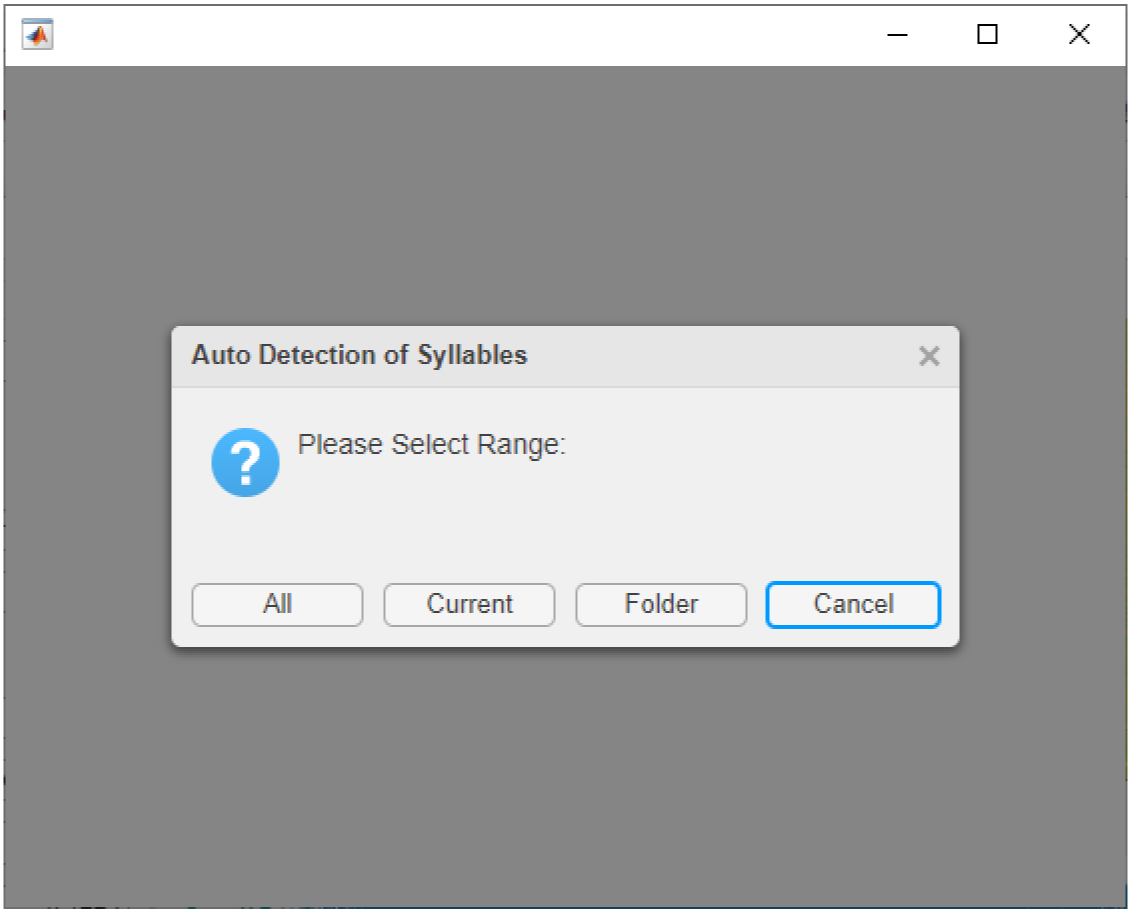
▼ Load network

The HybridMuse model is saved in the "autodetection → model" folder and it should be found by the app. However, if the app is unable to find and load this model, you will be asked to manually select the network.

If you would like to load a different (fine-tuned model), right-click on the Auto Detection button and select "Change Net".

▼ Running the detection

Click on the Auto Detection button and choose whether you would like to detect USVs in the current segment of the audio file, the entire audio file, or an entire folder containing audio files.


Auto Detection of Syllables

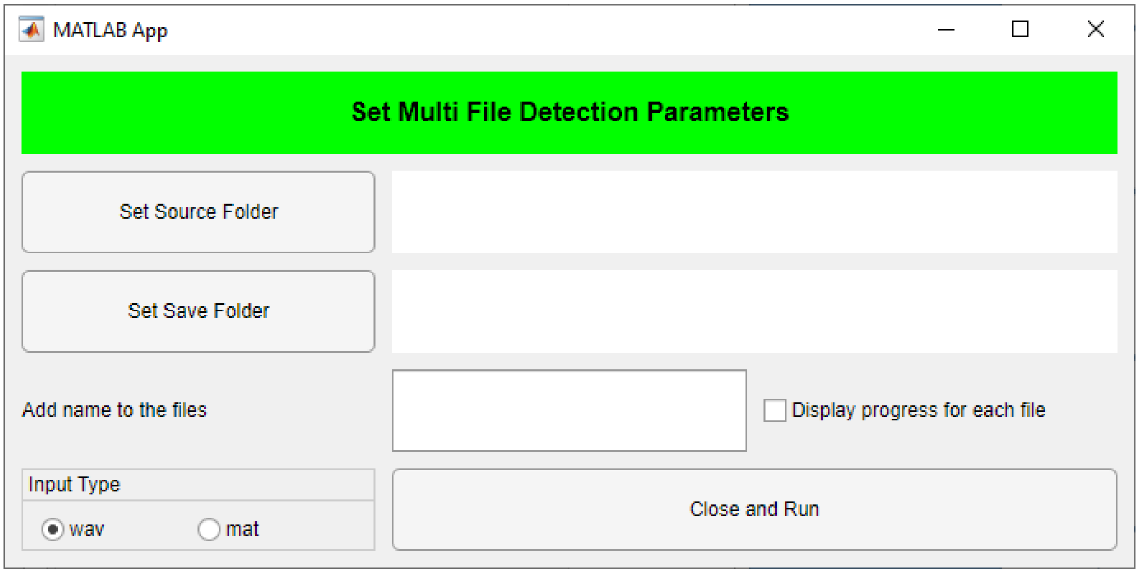
Please Select Range:

If you choose the folder option, you will get to the next window where you must select the source folder with the audio files, the destination folder where you would like to save the detection files, and the file type.

If you would like to see a progress bar for each file, check the "Display progress" box

If you want to prepend a name to the files, edit the "add name" text box.

Finally, click on Close and Run to run.


MATLAB App

Set Multi File Detection Parameters

Set Source Folder

Set Save Folder

Add name to the files

□ Display progress for each file

Input Type

● wav ○ mat

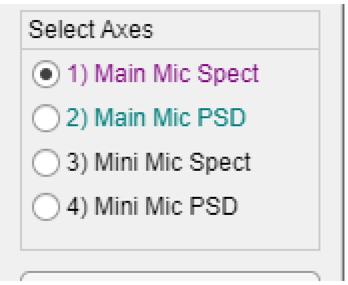
Close and Run

▼ Label USVs

▼ Select axes

After loading a file, you may add or edit labels in the spectorgram.

First, choose the desired axes


Select Axes

● 1) Main Mic Spect

○ 2) Main Mic PSD

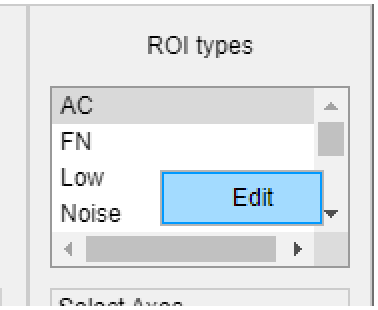
○ 3) Mini Mic Spect

○ 4) Mini Mic PSD

Notice the the "Spect" and "PSD" axes are connected since each pair represent the same audio file.

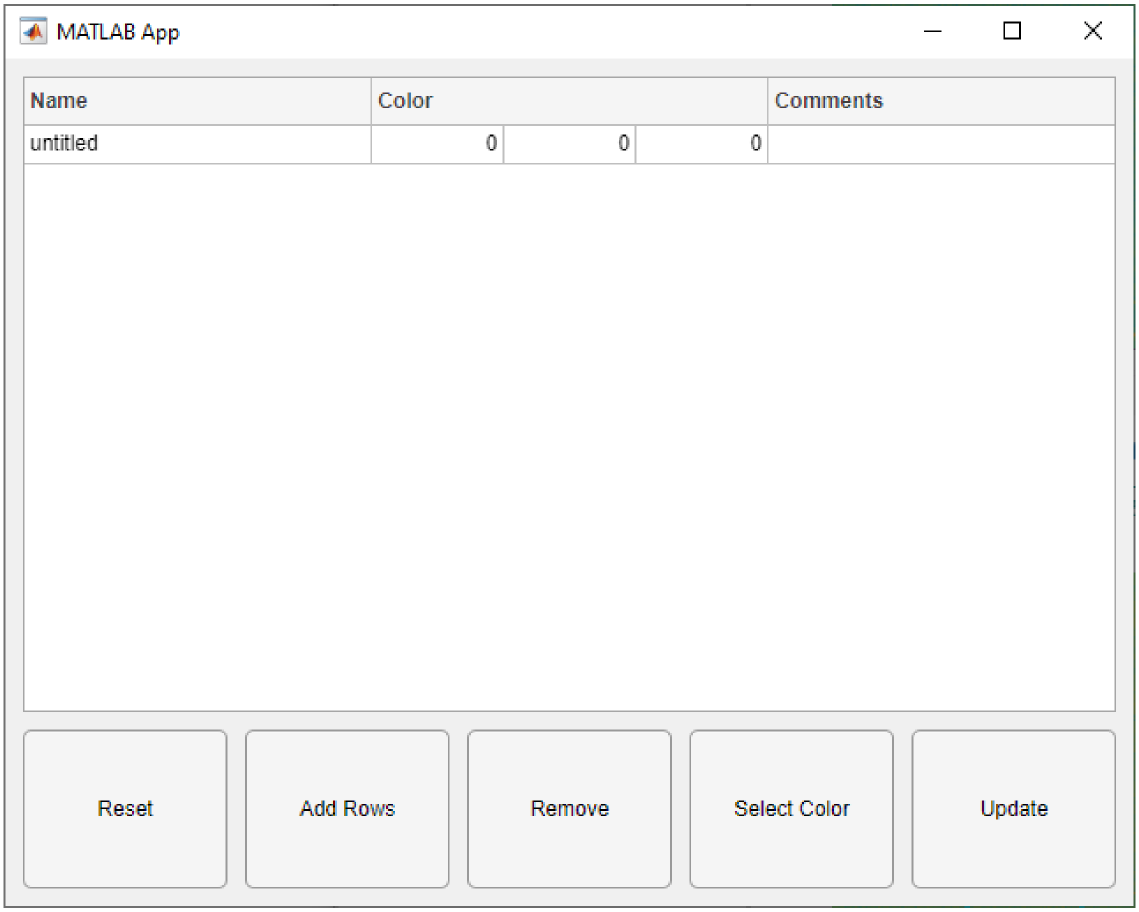
▼ Select Label

Next, choose the label from the ROI types panel.



▼ Edit label properties

You can choose the label from the default labels, or add new labels by right-clicking the list and clicking on the "Edit" button.



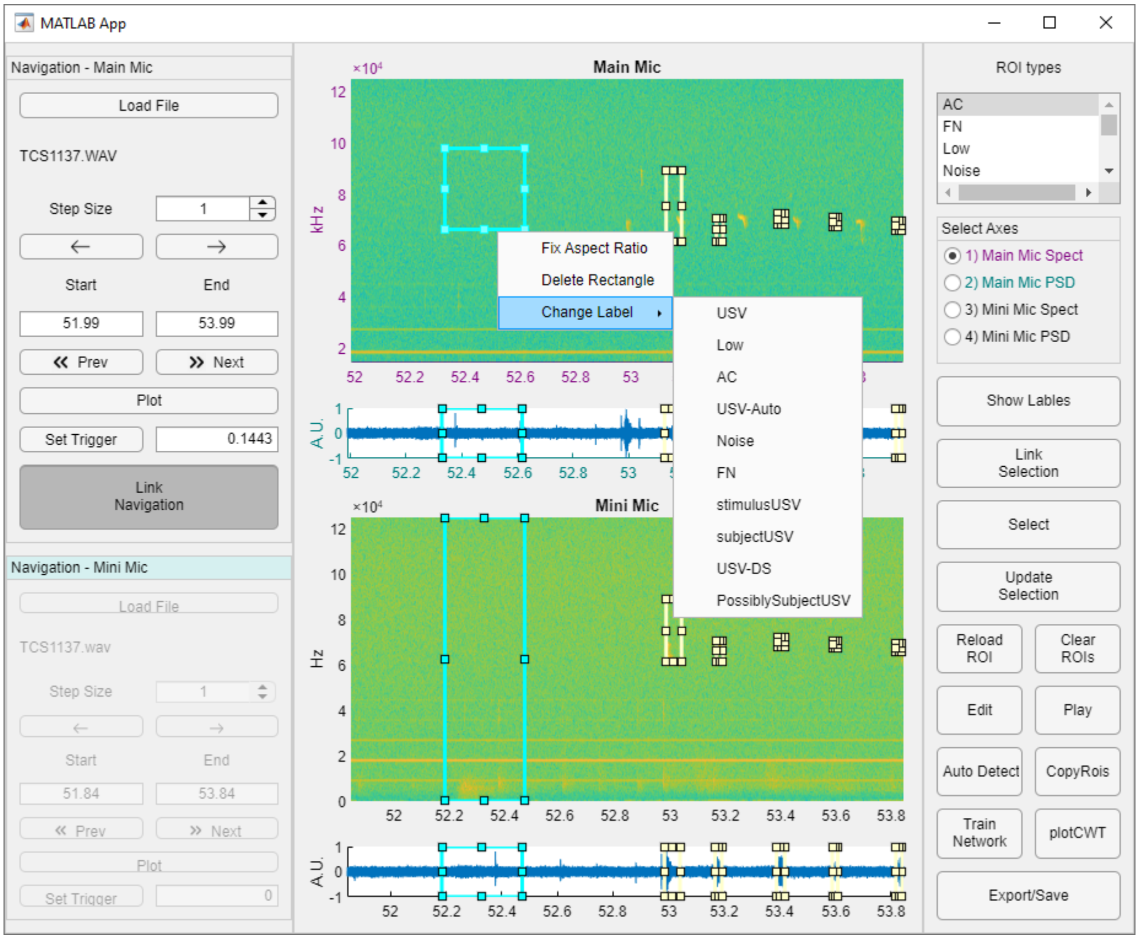
Name	Color			Comments
untitled	0	0	0	

You may not edit the default labels, however, you may add new labels or edit non-default labels.

▼ Labeling USVs

After selecting the axes and label, click on the "Select" button to start labeling, and click on the Esc button on your keyboard to exit selection mode.

- ▼ Modify labels



Step Size 1

Start 51.99

End 53.99

Set Trigger 0.1443

Select Axes: ● 1) Main Mic Spect

You can modify the labels of a USV by right-clicking on the box and choosing a new label, or resize, move or delete the box. However, after resizing or moving, you must click on the "Update selection" button to save the changes. Click on "Reload ROI" button to undo.

- ▼ Mirror labels

You can mirror the selection from one file to the other file by clicking on the CopyRoi; this will duplicate any labels from the selected file to the other.

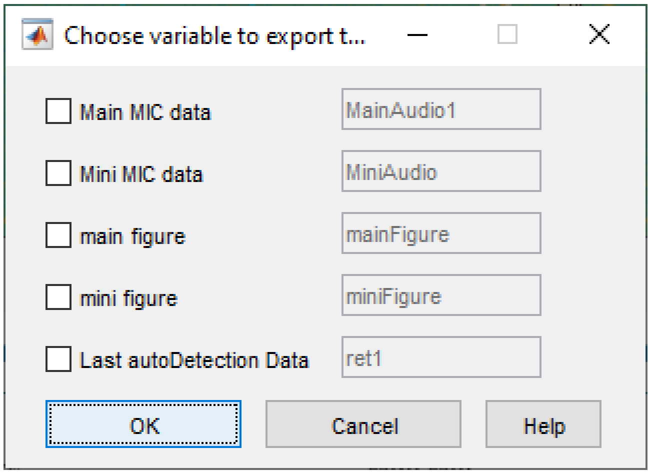
- ▼ When it becomes impossible to interact with the label boxes.

In some rare cases, interaction with the label-boxes become impossible, in that case, click on "Select" button once and exit selection mode (Esc) to refresh the plot.

▼ Export

Performing auto-detection on a folder, automatically saves the files in the selected folder and no exporting is needed. However, if you modified a file and would like to save you work, then click on the "Export Save" button.

This will prom the next window:



Choose variable to export t...

☐ Main MIC data	MainAudio1
☐ Mini MIC data	MiniAudio
☐ main figure	mainFigure
☐ mini figure	miniFigure
☐ Last autoDetection Data	ret1

In which you may choose what to export and save.

The exported or saved variable includes a summary of the file and an audioClip object in the "obj" field.

▼ Retrain network

To retrain the model, you must first generate labeled data, and then create training files.

▼ Data preparation

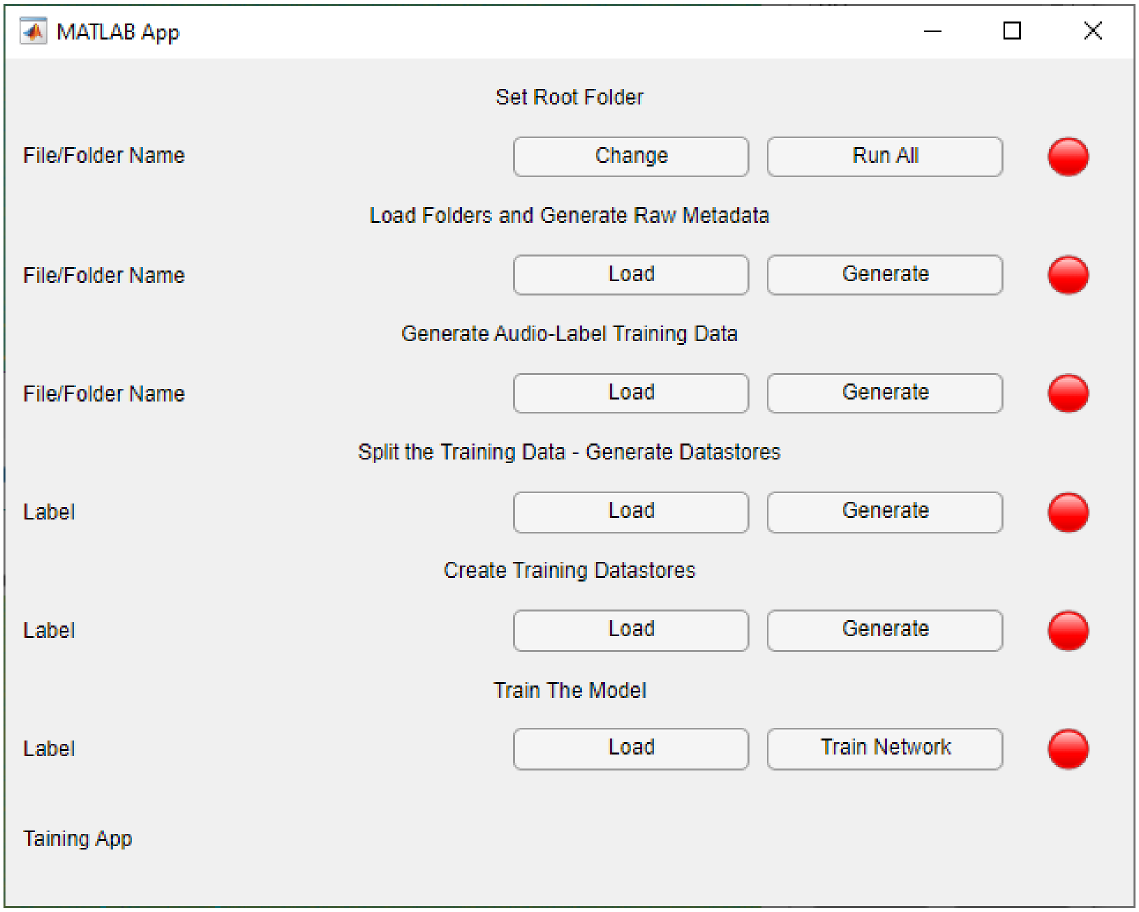
To retrain the model, prepare a folder with the next sub-folders:

- audio - containing wav files
- labels - containing csv files with matching names. the csv files must contain time_start, and time_end for the beginning and ending time stamps of each USVs.

- Noise - This is optional, in case you would like to train on the background noises
- Noise Labels - This is also optional, make sure that this folder is empty before you start

▼ Select the root folder

Click on the Train Network button to open the model training app



The default folder is the pwd. Change this by clicking on the "Change" button and selecting your prepared main folder.

▼ Run training

Click on "Run All" button to perform all next steps:

- Generate data stored and validate

- Create training data by segmenting the audio files into short 2 second files and adding the labels and save
- Split the files into audio and noise data stores and save
- Split the files into training and testing data stores and save
- Train the model and save checkpoints

Each step requires what the previous step generate, and it can either be generated (first time) or loaded (after being generated and saved).