ABRA Overview

The Auditory Brainstem Response Analyzer (ABRA) tool is an open-source web application in Streamlit allowing users to batch-upload and analyze ABR data to identify hearing thresholds and amplitudes. To run the application locally, please see the installation guide below.

The ABR waveform peaks and troughs are identified using a convolutional neural network (CNN), a supervised learning model trained with human assessments of latency and amplitude serving as ground truth. Unsupervised machine learning methods are also provided to assess the amplitude. Please find more details on the model underpinnings and purpose here: An Open-Source Deep Learning-Based GUI Toolbox For Automated Auditory Brainstem Response Analyses (ABRA).

Installation Guide

Step 1: Opening Command Terminal

We need to open the command terminal to copy the github repository to our system, install all the dependencies required by our application and run the application.

For Windows:

1. **Open the Command Prompt:**
    - Press Win + S to open the search bar, type cmd, and press Enter.
    - Alternatively, open the Start menu, scroll down to "Windows System," and click on "Command Prompt."
2. **Verify the Command Prompt is Ready:**
    - You should see a window with a black background and a command prompt that starts with your computer's username, such as C:\Users\YourName>.
    - You can now enter commands directly, like pip --version, to verify pip's installation.

For macOS:

1. **Open the Terminal:**
    - Press Cmd + Space to open Spotlight Search, type Terminal, and press Enter.
    - Alternatively, go to Applications > Utilities, then double-click on Terminal.
2. **Verify the Terminal is Ready:**
    - A Terminal window will open with a prompt, usually displaying something like your username@MacBook ~ %.
    - You can now enter commands such as python3 -m pip install --upgrade pip to install or verify pip.

For Linux:

1. **Open the Terminal (methods may vary slightly depending on the distribution):**
   - **Ubuntu:** Press Ctrl + Alt + T to open Terminal. Alternatively, you can find "Terminal" in the application menu.
   - **Fedora:** Press Super (Windows key), type Terminal, and press Enter.
   - **Other Distributions:** Most Linux distributions have a Terminal option in the applications or main menu.
2. **Verify the Terminal is Ready:**
   - The Terminal prompt will display, often in the form username@hostname:~$.
   - You can now enter commands like sudo apt update and curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py to install pip.

Step 2: Installing pip

Pip is the package installer for Python and is typically included with Python distributions. If you need to install pip, follow these steps based on your operating system.

For Windows:

1. **Download and Install Python**:
   - Go to the Python download page and download version 3.11.8.
   - Run the installer and make sure to check the box for "Add Python to PATH" before clicking "Install Now."
2. **Verify pip Installation**:
   - Open a Command Prompt (type cmd in the Start Menu and press Enter).
   - Run the following command to check if pip is installed:

     pip --version

   - If the version displays, pip is installed successfully. If not, you may need to reinstall Python and ensure "Add Python to PATH" is selected.

For macOS:

1. **Install Homebrew** (recommended if not already installed):
   - Open the Terminal and enter:

     /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

2. **Install Python 3.11.8**:
   - Once Homebrew is installed, install Python by running:

brew install python@3.11.8

3. **Verify pip Installation**:
   - Check if pip is installed by running:

     python3 -m pip install –upgrade pip

   - If the latest version is already installed, you will see a message saying, ‘requirement already satisfied.’ If it’s not, you will see the version of pip that was just installed.

<u>For Linux:</u>

1. **Install Python 3.11.8 and pip**:
   - Most Linux distributions come with Python installed, but you can install the specified version and pip (for Python 3.11.8) with the following commands:

     sudo apt update
     sudo apt install python3.11 python3.11-venv python3.11-distutils

   - Then, install pip for Python 3.11.8:

     curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
     python3.11 get-pip.py

2. **Verify pip Installation**:
   - Run the following command to confirm pip is installed:

     pip --version

Step 3: Cloning GitHub Repository

To download the project files from GitHub to your local machine, you need to clone the repository.

<u>For Windows:</u>

1. **Ensure Git is Installed:**
   - If you haven't already, download and install Git for Windows from git-scm.com.
   - During installation, you can choose the default options.
2. **Clone the Repository:**
   - In the Command Prompt, navigate to the directory where you want to clone the repository using the cd command. For example:

cd path\to\your\directory

- Run the following command to clone the repository :

  git clone https://github.com/ucsdmanorlab/abranalysis.git

For macOS:

1. **Ensure Git is Installed:**
   - Git is usually pre-installed on macOS. You can check by running the command git --version in the Terminal. If it's not installed, you can install it using Homebrew:

     brew install git

2. **Clone the Repository:**
   - In the Terminal, navigate to the directory where you want to clone the repository using the cd command. For example:

     cd /path/to/your/directory

   - Run the following command to clone the repository

     git clone https://github.com/ucsdmanorlab/abranalysis.git

For Linux:

1. **Ensure Git is Installed:**
   - Most Linux distributions come with Git installed. You can check by running the command git --version in the Terminal. If it's not installed, you can install it using:

     sudo apt install git

2. **Clone the Repository:**
   - In the Terminal, navigate to the directory where you want to clone the repository using the cd command. For example:

     cd /path/to/your/directory

   - Run the following command to clone the repository:

     git clone https://github.com/ucsdmanorlab/abranalysis.git

Alternatively, you can follow the steps listed out here to clone the repository on your system.

Step 4: Navigate to the Cloned Repository:

After cloning, move into the project directory by running:

cd abranalysis

You are now in the project folder and can follow the further instructions to install dependencies and run the application.

Step 5: Installing Project Dependencies

Once you have pip set up, you can install all necessary packages listed in **requirements.txt**, by running the following code in this directory:

pip install -r requirements.txt

Note: Ensure that your python version is 3.11.* to avoid compatibility issues with the dependencies present in requirements.txt.

If you are currently using a different version of python, we recommend creating a separate anaconda environment. See the documentation linked here.

Step 6: Running the application locally

Run the following command in your terminal in the same directory as your requirements.txt file and cloned git repository. The application will open in your selected browser window:

streamlit run ABRA_v1.0.0.py

To close the app, hit **Ctrl + C** (for both Windows and Mac) in the terminal.

Step 7: Check for application updates

ABRA is undergoing continuous improvements, so be sure to check for updates in the GitHub repository. If your local version is behind, make sure to **close the app** and navigate to the project directory to run the following command to pull the latest changes:

git pull

This will update your local copy without needing to re-clone the repository.

Minimum File Requirements

Data Overview: Before uploading any files, confirm contents are meeting the minimum requirements. ABR data should be recorded across multiple time points with the coinciding frequency and sound level/attenuation (dB). Each file should contain results per subject.

Required Units:

- **Frequency:** Hz only
- **Sound/Decibel:** ABRA may be calibrated to interpret sound as an absolute term (Level) or relative to an assigned baseline (Post-Attenuation)
    - Note the appropriate mode in the “FIle Uploading” section.
- **ABR data:** Microvolts or nanovolts only
    - Note the metric used in the “FIle Uploading” section.

Time Intervals: The ABRA tool was primarily trained on ABR data recorded over 10ms with 244 time points, but this is not a requirement. The following methods are employed to create a standard waveform for analysis across different testing methods and subjects:

For files with less than 244 time points per test, cubic interpolation is used to increase the number of data points to 244 while preserving the smoothness of the curve.

For files with more than 244 time points per test, linear interpolation is employed to decrease the number of points while preserving the shape of the curve without additional complexity.

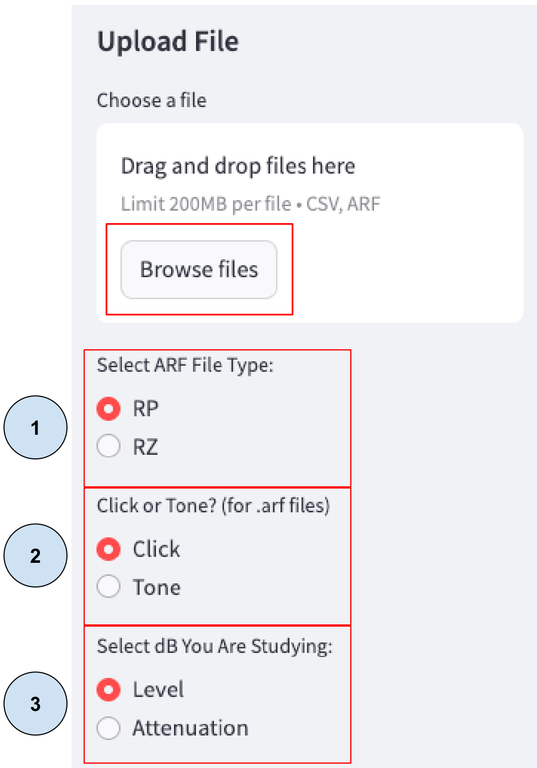
For more details on the applied interpolation method, please refer to the study linked here.

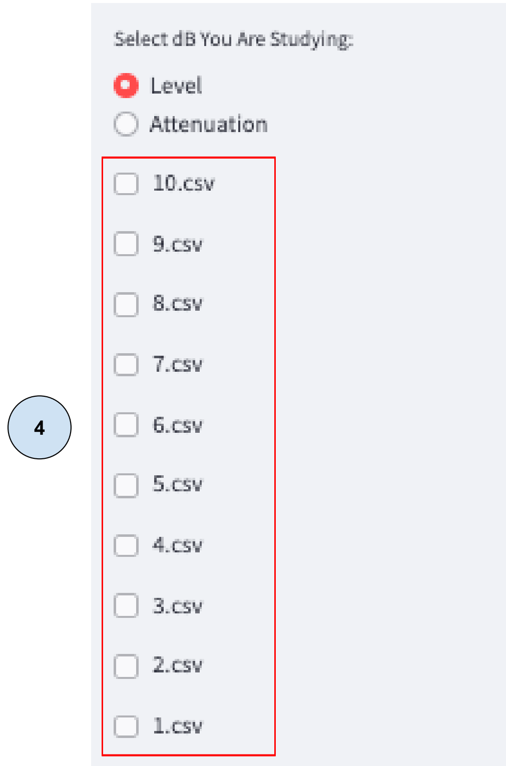
Format: Once the data is confirmed, check that the column headers match the below. You may also see a csv example below or linked here:

- **Freq(Hz):** frequency data
- **Level or Post-Atten:** decibel data
- **0,1,2,etc. :** ABR data

Freq(Hz)	Level(dB)	0	1	2	3	4	5	6	7	8	9
100	70	-0.0083304	-0.0243214	-0.0446716	-0.0395266	0.011921372	0.05572595	0.037208	0.044037387	0.04209653	-0.0050488
100	0	-0.2503932	-0.2503849	-0.2573406	-0.2684239	-0.2626362	-0.3120822	-0.3534935	-0.3582582	-0.3443138	-0.3158473
100	5	0.3649233	0.373810131	0.398919028	0.409454492	0.373636443	0.337830118	0.307999819	0.267804314	0.237223254	0.185323714
100	10	0.1807231	0.185209945	0.175645491	0.153714688	0.109396051	0.04801817	-0.0111143	-0.0226384	0.00566427	0.028257085
100	15	0.06054268	0.03824372	0.025114167	0.013596809	-0.0043657	-0.0047401	0.00791821	-0.0248831	-0.0865842	-0.1483755
100	20	0.1693397	0.212649357	0.194556441	0.154043086	0.114946604	0.09910958	0.044877258	0.003922	-0.0125721	-0.0148933
100	25	-0.0640472	-0.0656569	-0.0627761	-0.0299759	0.00777361	0.050106262	0.08796633	0.122801866	0.124778449	0.08568556
100	30	-0.0152603	-0.0130612	-0.013438	-0.039369	-0.0815572	-0.097542	-0.1277477	-0.1281518	-0.1167214	-0.1072119
100	35	-0.0099649	-0.075258	-0.0662898	-0.0409999	-0.0397074	-0.0632075	-0.0664273	-0.0228447	0.04137616	0.0266271
100	40	0.02961592	0.04215395	0.020483750	-0.0227714	-0.0492612	-0.0611469	-0.0533565	-0.0617991	-0.0619253	-0.0547162

File Uploading





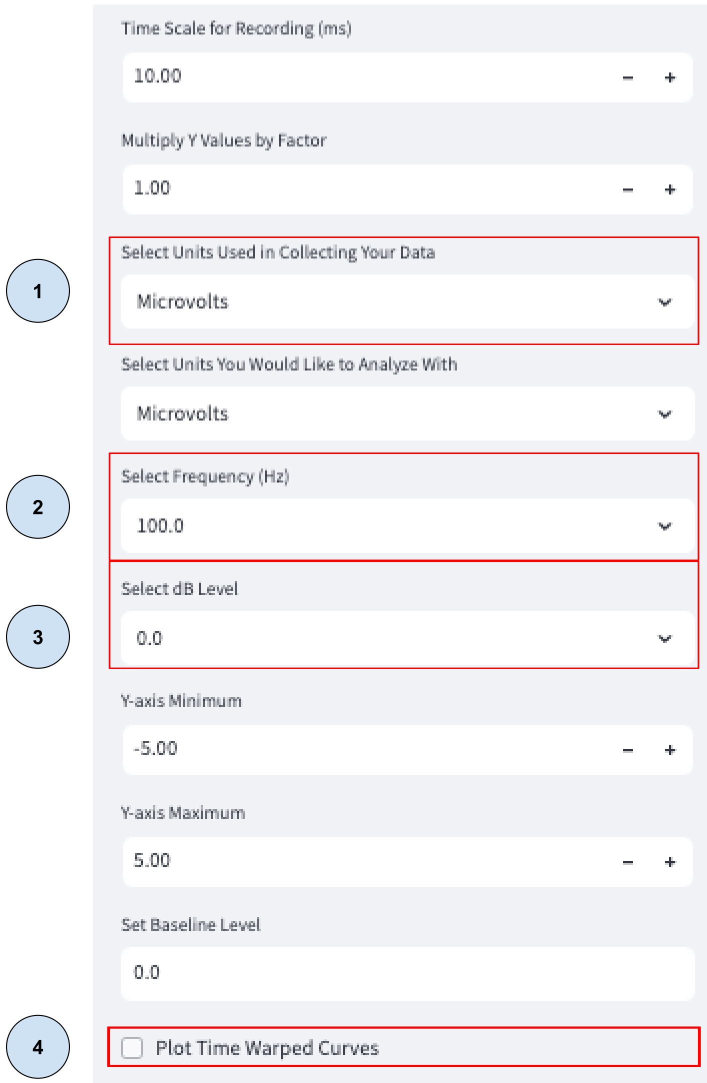
1 - Once the app is set-up on your local machine, you're ready to upload your files. You can upload one or multiple .csv or .arf files for visualization. Note the size limitations per file.

2- For Tucker Davis ARF files only: Select the file source (BioSigRP or BioSigRZ). If uploading CSV files, leave these at the default settings.

3- Select the sound format (level or attenuation) depending on your file contents. If your file organizes the decibel data as "Post-Atten" instead of "Level (dB)", change the setting to Attenuation.

4 - Once uploaded, select the file(s) you'd like to visualize before calibrating the visualizer settings. Data from multiple graphs will be displayed in separate, clearly labeled plots.

Plot Settings



You may update the plot range settings here per your preferences. Note the highlighted sections above.

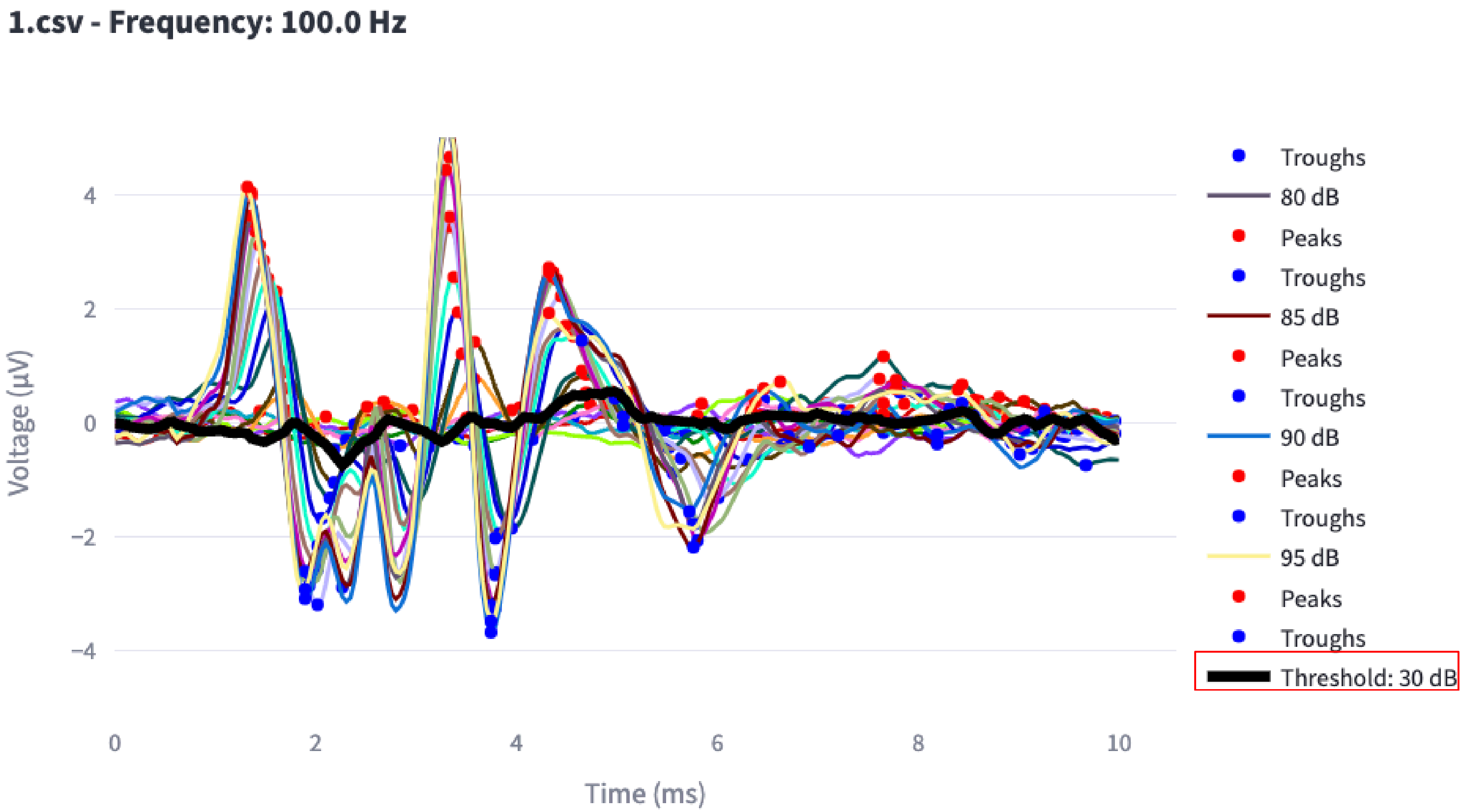
1 - Update the units to match your ABR data (microvolts or nanovolts). Note that you can change the units displayed in the option just below.
2 - Select the frequency you'd like to analyze
3 - Select the decibel you'd like to analyze - Plot Single Wave (Frequency, dB) only, other plots will display all available decibel data
4 - Select the Plot Time Warped Curves - Plot Waves at Single Frequency only

Wave Plotting Options

There are multiple options to plot ABR data with options, shown below. Note that all plots and tables have options for exporting by hovering in the top right corner.

File Name	Frequency (Hz)	dB Level	Wave I amplitude (P1-T1) (μV)	Latency to First Peak (ms)	Amplitude Ratio
1.csv	100	0	0.5421	2.7049	
1.csv	100	5	0.3697	1.9672	
1.csv	100	10	0.3498	4.7131	

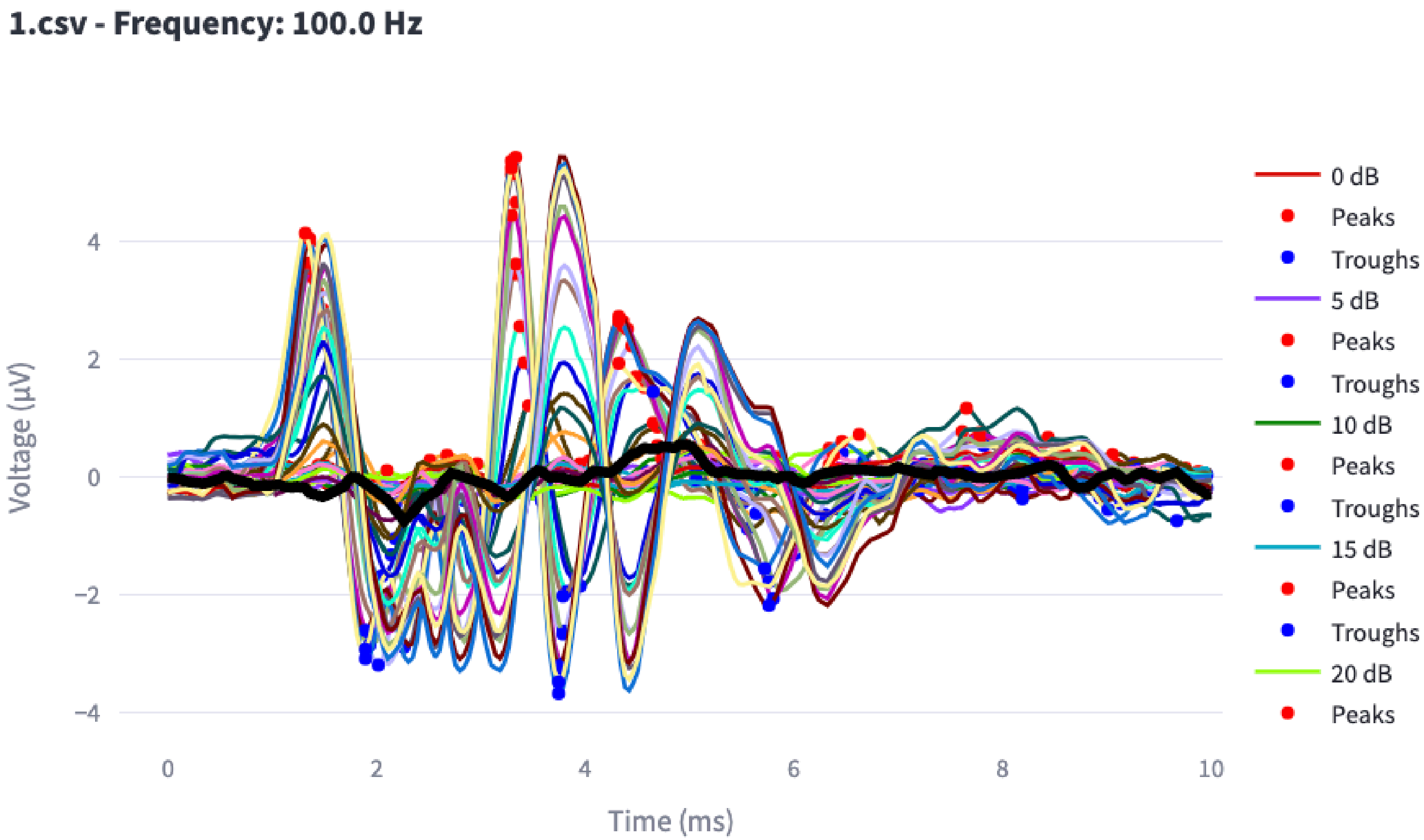
Plot Waves at Single Frequency: Displays ABR waveforms for all recorded sound levels/attenuations at the selected frequency. This also includes the model-identified Wave I Amplitude (μV), Latency to first peak (ms) and the Amplitude ratio (Peak 1/Peak 4) metrics tabulated below. The model-identified threshold value will be noted in the legend on the right (depending on the number of waveforms produced, you may need to scroll).



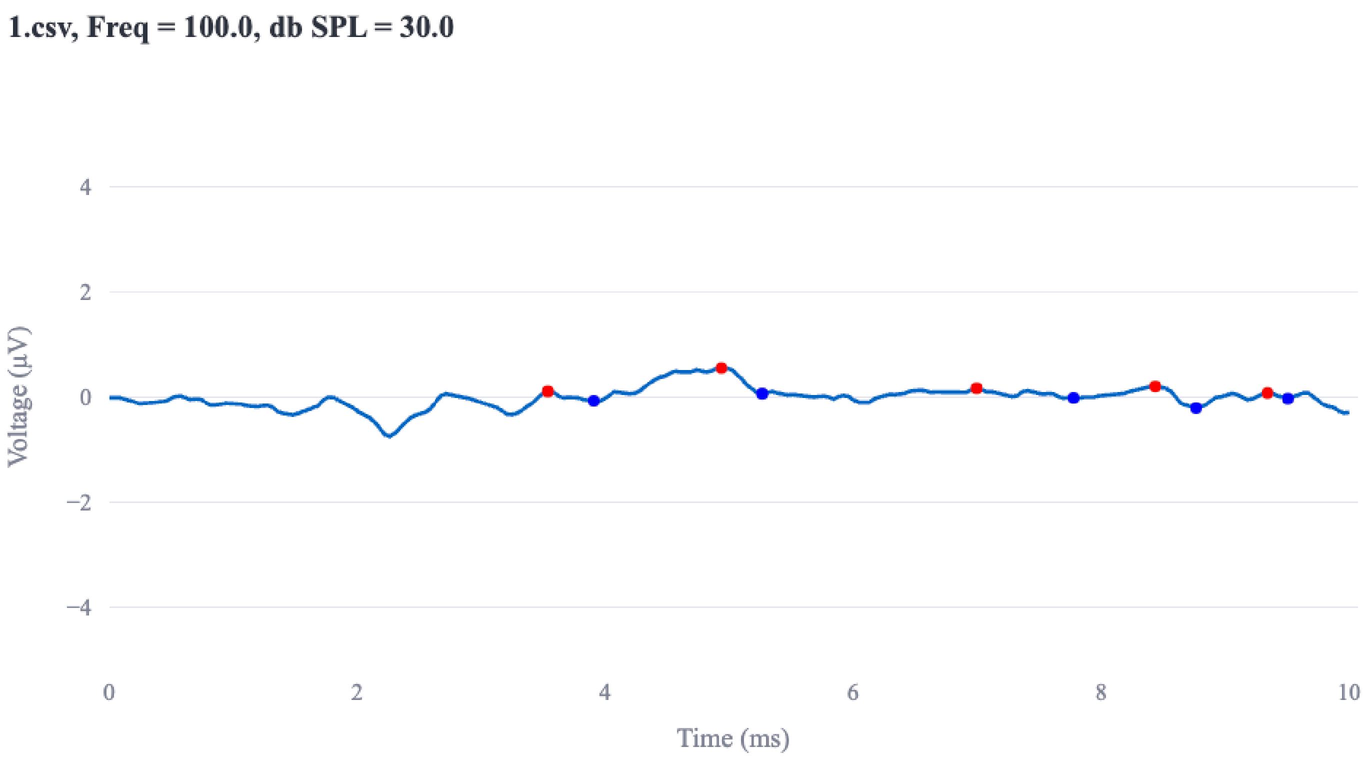
Time Warping (Plot Waves at Single Frequency only):
As seen above, waveforms tend to vary in latency, which can interfere with understanding the peaks/troughs per frequency. By selecting “Plot Time Warped Curves”, we are able to align the latency of the different waveforms using the elastic square-root slope framework and visualize the peaks/troughs grouped across the different decibel levels at the selected frequency.

Note that this option should only be selected when assessing the amplitude, not the latency.

[An Open-Source Deep Learning-Based GUI Toolbox For Automated Auditory Brainstem Response Analyses (ABRA).]

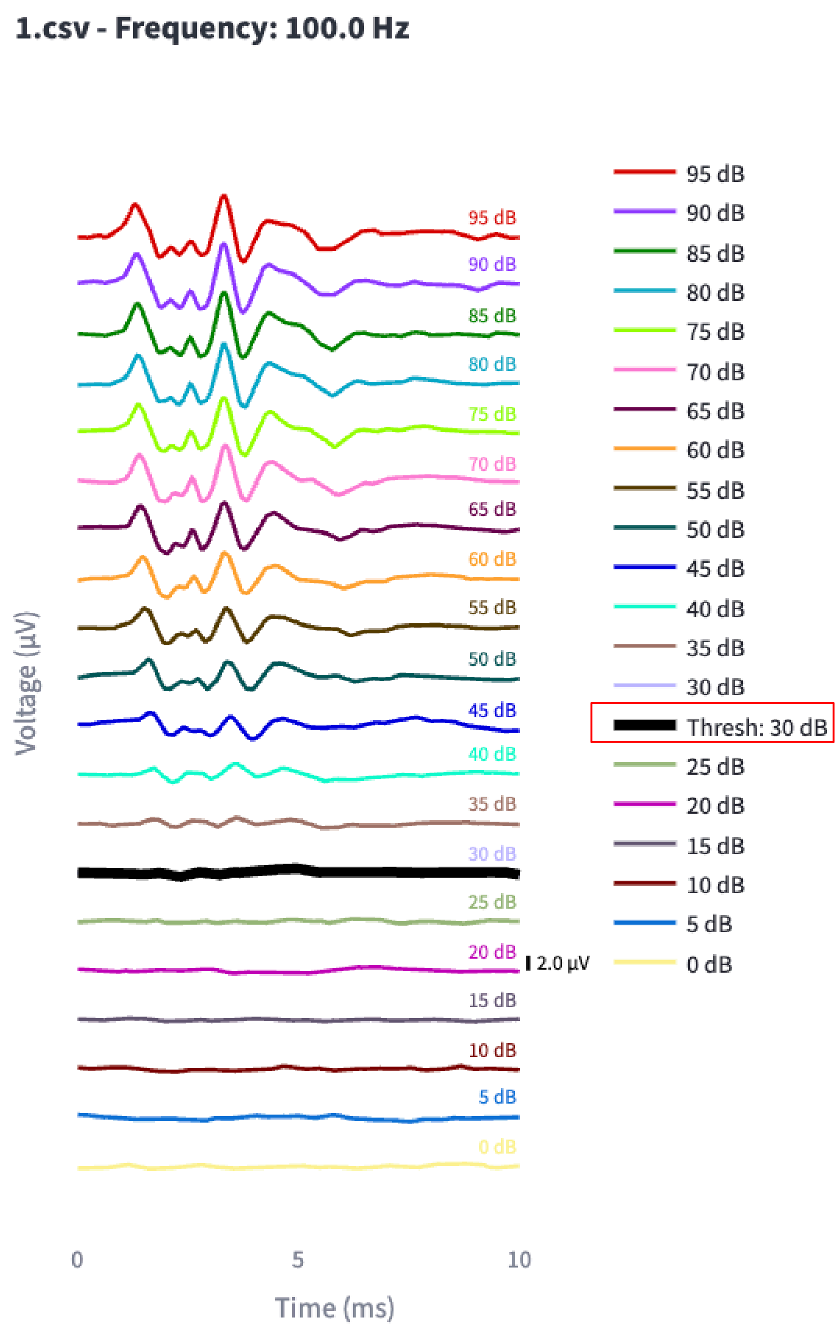


Plot Single Wave (Frequency, dB): Displays a single ABR waveform for the selected sound level/attenuation at the selected frequency. This also includes the model-identified Wave I Amplitude (μV), Latency to first peak (ms) and the Amplitude ratio (Peak 1/Peak 4) metrics tabulated at this decibel level.

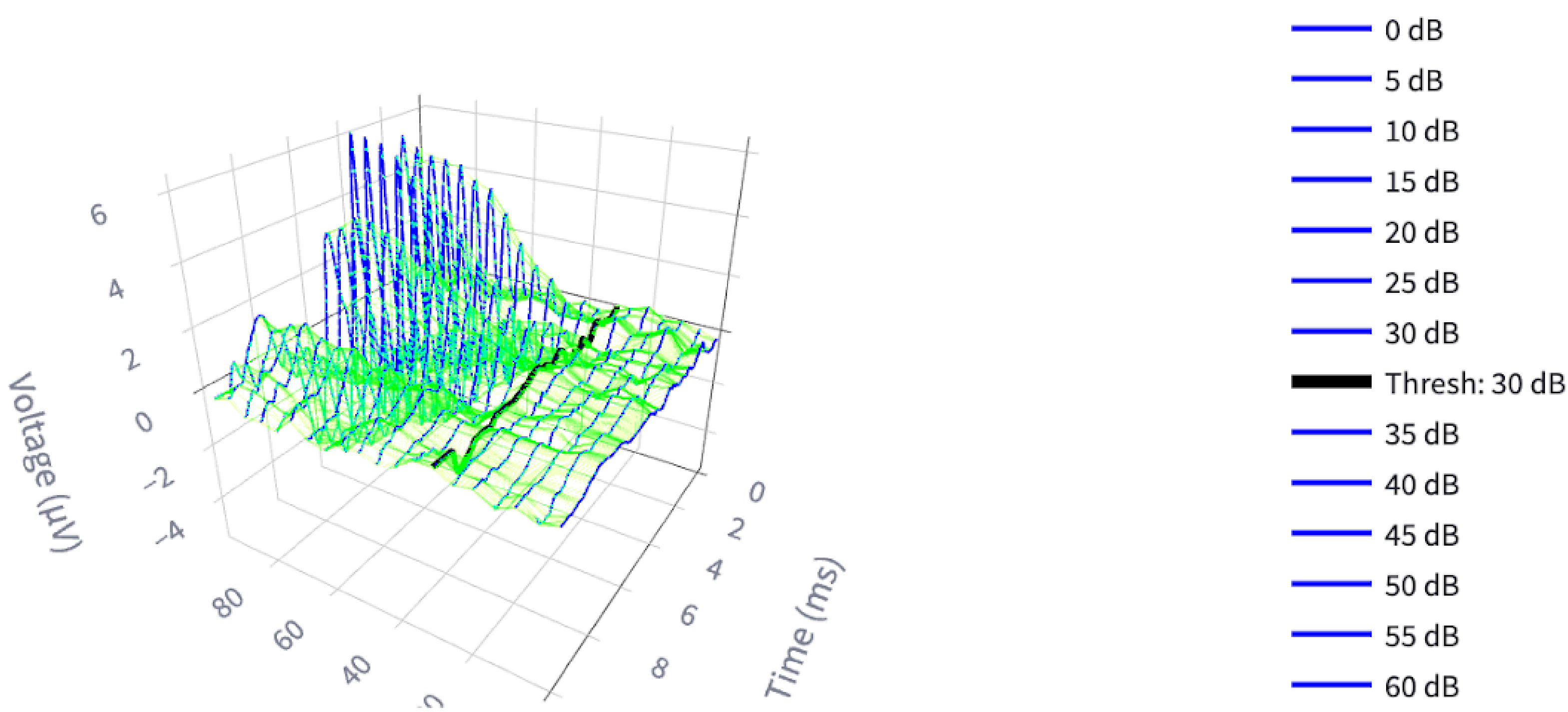


File Name	Frequency (Hz)	dB Level	Wave I amplitude (P1-T1) (μV)	Latency to First Peak (ms)	Amplitude Ratio
1.csv	100	30	0.1792	3.5246	

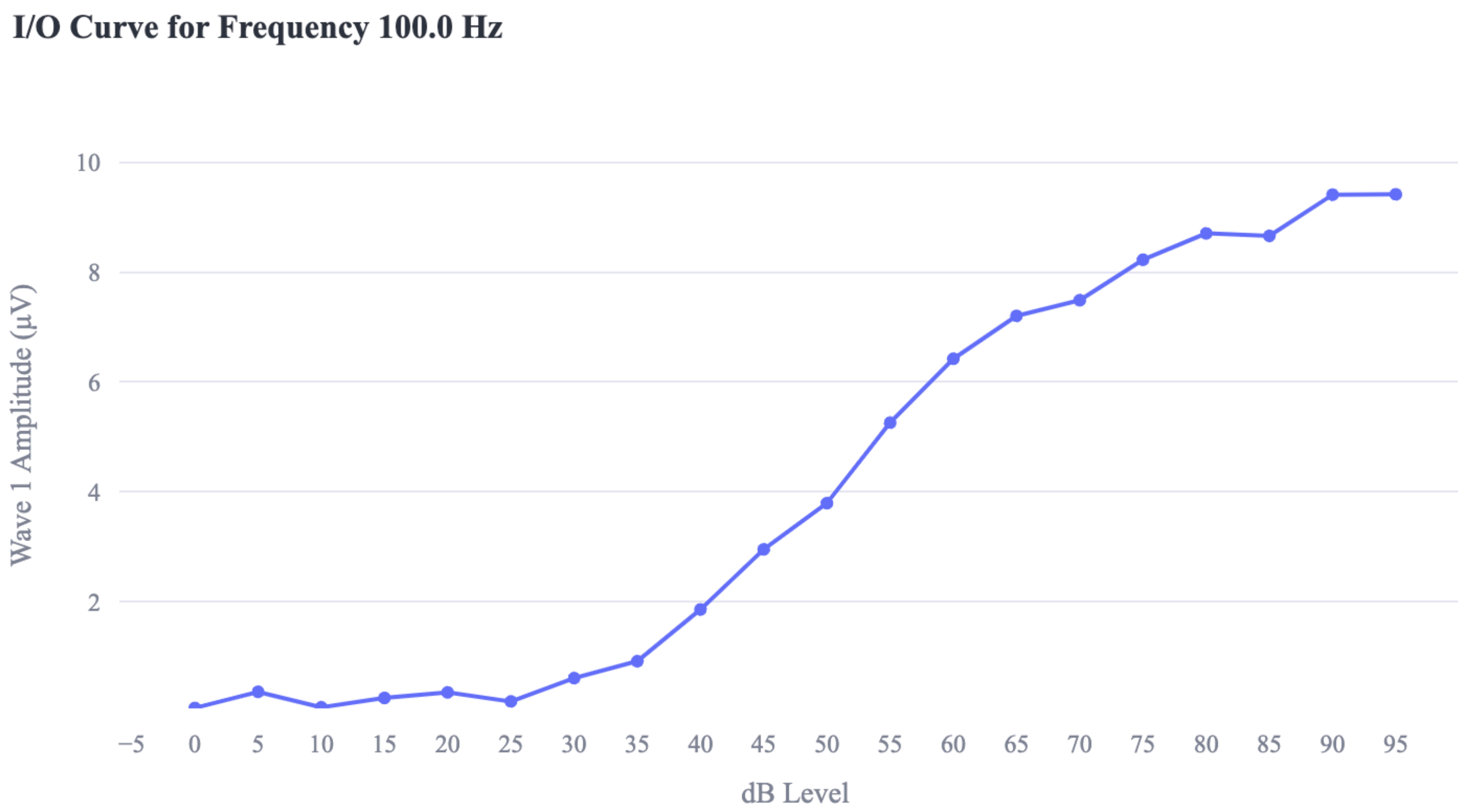
Plot Stacked Waves at Single Frequency: Displays ABR waveforms for all recorded decibel level/attenuation at the selected frequency stacked to see each waveform shape individually. The model-identified threshold value is noted in the legend on the right.



Plot 3D Surface: Displays ABR waveforms for all recorded dB level/attenuation at the selected frequency in a 3D space with an added dimension for different dB levels to see each waveform shape. The model can be dragged around for custom perspective for the user. Useful for visualizing the amplitude and structure difference among waves of different dB levels. The threshold value is noted in the legend on the right.



Plot I/O Curve: Displays graph of amplitude of first wave in an ABR waveform of different dB levels at the selected frequency. The curve reflects the relationship between input (dB level) and output (Wave 1 amplitude), offering insights into the response strength.



Filename	Frequency	Threshold	Unsupervised Threshold
3.csv	100	30	45
3.csv	3,000	80	15
3.csv	6,000	40	15
3.csv	12,000	15	55
3.csv	18,000	20	15
3.csv	24,000	35	45
3.csv	30,000	40	15
3.csv	36,000	40	50
3.csv	42,000	40	40

Return All Peak Analysis: Displays a table that shows the features like the Wave I Amplitude (μV), Latency to first peak (ms) and the Amplitude ratio (Peak 1/Peak 4) for each selected file at every frequency and dB level, along with the estimated threshold for the given file and frequency.

File Name	Frequency (Hz)	dB Level	Wave I amplitude (P1-T1) (μV)	Latency to First Peak (ms)	Amplitude Ratio
3.csv	100	0	0.0623	1.5984	
3.csv	100	5	0.3576	2.9918	
3.csv	100	10	0.0761	1.8443	
3.csv	100	15	0.2452	1.9672	
3.csv	100	20	0.3468	2.2131	
3.csv	100	25	0.1829	4.4672	
3.csv	100	30	0.6072	1.8033	
3.csv	100	35	0.915	1.8033	
3.csv	100	40	1.8571	1.7213	
3.csv	100	45	2.9519	1.6393	