VoiceLab: Software for Fully Reproducible Automated Voice Analysis

David R. Feinberg

1McMaster University, Hamilton, Ontario, Canada
feinberg@mcmaster.ca

10.21437/Interspeech.2022-113

Abstract

There’s a problem with acoustic analyses because you often need to hand adjust parameters meaning you can only process them individually or in small batches. This creates two key problems. First, it compromises the reproducibility of measurements because setting parameters by hand requires specialist knowledge and is often poorly documented. Second, it means that you can’t easily process large samples of voices, which creates bottlenecks in workflows. This issue is compounded by researchers looking to use increasingly large and diverse samples. To address these issues, VoiceLab software offers automated acoustical analysis and automatically logs analysis parameters. VoiceLab analyses are fully reproducible and require little to no knowledge about acoustical analysis from the user. Analysis parameters can also be manually adjusted by experts. VoiceLab is used primarily by researchers studying person perception, creating reproducible voice manipulations, developing voices for conversational agents, and creating feature sets for machine learning.

Index Terms: acoustic analysis, replication, acoustic phonetics

1. Introduction

Measuring voices quickly and accurately is challenging. Research shows that in particular, the quality and replicability of formant frequency measurements depends on the expertise of who is measuring the voices [1] parameters they used, but this rarely happens, especially when measurement parameters are adjusted by hand [1]. Measuring formant frequencies by hand is very time consuming, creating a research bottleneck. One common solution is to tolerate measurement error and use default software settings [2, 3]. However generated, measurement error could be in part fueling the replication crisis [4]. Since it is critical to increase sample sizes, population diversity, and replicability, automated solutions are desired.

2. Acoustic Analysis Software

2.1. Praat and VoiceSauce

The packages most comparable to VoiceLab are Praat [5], and Voice Sauce [6], as VoiceLab uses algorithms from each of these packages. Both Praat and Voice Sauce are highly regarded and well cited. While incredibly powerful, Praat, and Voice Sauce can be difficult to use, especially for beginners and people outside of speech science. This is because substantial knowledge of acoustic phonetics is needed to properly set analysis parameters. In light of this, many researchers do not modify the default settings. Furthermore, setting up Voice Sauce on non-windows platforms is challenging.

2.2. VoiceLab

VoiceLab software is an open-source (MIT License) software package based on Praat and VoiceSauce that is designed to address these issues. VoiceLab automates acoustical analyses with an easy to use interface designed for beginners. VoiceLab automatically determines acoustical analysis parameters by a series of multipass analyses and heuristics that and achieves results highly comparable to those created when using by-hand acoustical analysis [7], and can effectively replace using gender to determine analysis settings (see supplementary material for validation analysis: https://osf.io/3wr6k/files/). The analyses are not perfect, but do substantially reduce error as compared to using program defaults [8]. As such, in VoiceLab, advanced users can still specify their own analysis parameters, and still take advantage of the easy-to-use batch-processing interface. VoiceLab can handle analyzing thousands of speech files in batch runs, and outputs results and analysis parameters for each voice into.xlsx files.

VoiceLab also automates a number of voice manipulations such as amplitude normalization, reversing sounds, pitch manipulations, and formant scaling. To help create reproducible stimuli, all analysis parameters used for manipulations are also automatically saved to.xlsx files.

As VoiceLab’s algorithms are deterministic, to reproduce any analysis, manipulation, or visualization in VoiceLab, simply load the voices and select the relevant analyses, manipulations, and/or visualizations from the settings menu, and press “Start”. To reproduce VoiceLab analyses in Praat and/or VoiceSauce, the settings.xlsx file provides all of the information relevant to properly configure each software package for each analysis.

3. VoiceLab Analysis

3.1. Pitch/Fundamental Frequency

Voice pitch has a bimodal distribution among male and female adults [9] Because of this distribution, research suggests using gender initially to determine pitch ceiling and floor values to increase accuracy of fundamental frequency and/or pitch measurements [9, 10, 11]. This practice increases accuracy, but only when voices conform to researchers’ perceptions of gender identity. For example [12] show that there is considerable overlap in male and female vocal ranges, with low female voice pitch at 110Hz, and high male pitch at 185Hz. Both of these cases would apply the wrong analysis parameters if using gender as the deciding factor. There are many more cases where gender does not capture the relevant information about voice pitch such as among many post-menopausal women [13, 14, 15].

VoiceLab software uses a two pass analysis to measure pitch. First it measures pitch with settings slightly wider than Praat’s default pitch analysis settings (F0 floor = 60Hz; F0 ceiling = 600Hz). If pitch is less than 180Hz, VoiceLab uses (F0 floor = 60Hz; F0 ceiling = 250Hz); If pitch is greater than 170Hz, VoiceLab uses (F0 floor = 100Hz; F0 ceiling = 500Hz). This leads to fewer gender-age errors in measure-
ing voice pitch and is more accurate than using default settings. Advanced users can select Auto-Correlation or Cross-Correlation. See supplementary material for validation analysis: https://osf.io/3wr6k/files/.

VoiceLab uses the aforementioned algorithm for all acoustic measurement based on Praat’s pitch floor/ceiling values such as jitter, shimmer, and harmonics to noise ratio, and even setting formant-frequency analysis parameters. It is not used in Subharmonic calculations, which come from Voice Sauce.

3.2. Formant Frequencies

Praat’s recommended formant frequency settings are based on work showing that in certain vowels, voices with low fundamental frequencies can confuse the formant analysis [17]. To compensate for this, Praat recommends starting with maximum formant frequency to 5.5kHz for female voices and 5kHz for male voices [5], [17]. To get the most accurate and more reproducible measures from Praat, without an expert adjusting by hand, VoiceLab uses the Formant Path Algorithm from Praat [8], however VoiceLab set the number of formants to 5.5, rather than Praat’s default of 5, as this more closely aligned with hand-measures from Hillenbrand [7]. This option out-competed several other potential algorithms and software settings (see supplementary materials: https://osf.io/3wr6k/files/). Formant Path does a sweep of maximum formant values and finds the settings that minimize the error of a polynomial line of best fit to each of the formants individually [5][8]. If an advanced user wants to tune the parameters themselves, VoiceLab provides the interface to manually adjust these parameters.

3.3. Subharmonics

Subharmonic to harmonic ratio and Subharmonic pitch are measures from Open Sauce [6], a Python port of Voice Sauce [16]. These measurements do not use any Praat or Parselmouth code. As in [6] and [16], subharmonic raw values are padded with NaN values to 201 data points. The MatLab and Python versions of this script are nearly identical [17].

3.4. Energy

I replicated Voice Sauce’s Energy algorithm in Python using pitch measures from Praat to decouple Pitch and Energy. I replicated Voice Sauce’s Praat-based Pitch algorithm exactly (r=1). The Energy calculation based on this pitch measure correlated with Voice Sauce’s algorithm at r=0.99938368. See supplementary material for validation analysis: https://osf.io/3wr6k/files/.

3.5. Vocal-Tract length estimation

Estimating vocal-tract length and/or height from the voice is a well-researched topic. There are several different methods described in a recent meta-analysis [20]. To the best of my knowledge, these methods are not currently present in any acoustical analysis software. Therefore, VoiceLab software implements the following estimates of vocal-tract length from formant frequencies:

**Formant Frequency Dispersion** [18]

\[ Df = \frac{1}{n} \sum_{i=2}^{n} f_i - f_{i-1} \]  

**Formant Position**[19]

First, z-score the formant frequencies 1-4 separately across the sample:

\[ Z_f_i = \frac{x_i - \mu}{\sigma} \]  

Then take the mean across formants

\[ F_n = \frac{1}{n} \sum_{i=1}^{n} Z_{f_i} \]  

**Geometric Mean Formant Frequency**[20]

\[ GeometricMean = \left( \prod_{i=1}^{n} f_i \right)^{\frac{1}{n}} \]  

**F_n and VTL ΔF** [21]

\[ F_i \] is the slope of 0 intercept regression between \[ \frac{2n-1}{2} \Delta f \] and the mean of the first 4 formant frequencies. \( c \) is the speed of sound.

\[ VTL_{F_i} = \frac{1}{n} \sum_{i=1}^{n} (2n - 1) \left( \frac{c}{4f_i} \right) \]  

\[ VTL_{F_n} = \frac{1}{n} \sum_{i=1}^{n} (2n - 1) \frac{f_n}{4c} \]  

**Mean formant frequency** [22]

\[ F_n = \frac{1}{n} \sum_{i=1}^{n} f_i \]  

and Principle Components Analysis (PCA) [24].

Here, the means of the first 4 formants are first z-scored, and then run through a PCA algorithm. The exact formula and code for the PCA used in VoiceLab can be found here: https://github.com/scikit-learn/scikit-learn/blob/baf0ea25d/sklearn/decomposition/_pca.py#L116.

VoiceLab returns the predictions of the first factor of the PCA analysis. Usually there is only 1 factor, but only 1 factor is ever returned.

3.6. Noise Measures

VoiceLab software automatically calculates Harmonics to Noise Ratio, Cepstral Peak Prominance, and each of the 5 and 6 jitter and shimmer measures from Praat (respectively). VoiceLab also automatically calculates single factor PCA measures for Jitter and Shimmer, separately. Parameters for HNR, Jitter, and Shimmer are automatically calculated using the aforementioned multipass pitch algorithm. See validation analyses.

3.7. Spectral Shape

VoiceLab measures spectral shape using Praat’s measures of Centre of Gravity Standard Deviation Kurtosis Band Energy Difference Band Density Difference. VoiceLab also calculates Spectral Tilt by returning the slope of a linear regression between frequency and spectral density in power spectra of each sound [28].

\[ Z_f_i = \frac{x_i - \mu}{\sigma} \]  

\[ F_n = \frac{1}{n} \sum_{i=1}^{n} Z_{f_i} \]  

\[ GeometricMean = \left( \prod_{i=1}^{n} f_i \right)^{\frac{1}{n}} \]  

\[ F_i \] is the slope of 0 intercept regression between \[ \frac{2n-1}{2} \Delta f \] and the mean of the first 4 formant frequencies. \( c \) is the speed of sound.

\[ VTL_{F_i} = \frac{1}{n} \sum_{i=1}^{n} (2n - 1) \left( \frac{c}{4f_i} \right) \]  

\[ VTL_{F_n} = \frac{1}{n} \sum_{i=1}^{n} (2n - 1) \frac{f_n}{4c} \]  

\[ F_n = \frac{1}{n} \sum_{i=1}^{n} f_i \]  

and Principle Components Analysis (PCA) [24].

Here, the means of the first 4 formants are first z-scored, and then run through a PCA algorithm. The exact formula and code for the PCA used in VoiceLab can be found here: https://github.com/scikit-learn/scikit-learn/blob/baf0ea25d/sklearn/decomposition/_pca.py#L116.

VoiceLab returns the predictions of the first factor of the PCA analysis. Usually there is only 1 factor, but only 1 factor is ever returned.

3.6. Noise Measures

VoiceLab software automatically calculates Harmonics to Noise Ratio, Cepstral Peak Prominance, and each of the 5 and 6 jitter and shimmer measures from Praat (respectively). VoiceLab also automatically calculates single factor PCA measures for Jitter and Shimmer, separately. Parameters for HNR, Jitter, and Shimmer are automatically calculated using the aforementioned multipass pitch algorithm. See validation analyses.

3.7. Spectral Shape

VoiceLab measures spectral shape using Praat’s measures of Centre of Gravity Standard Deviation Kurtosis Band Energy Difference Band Density Difference. VoiceLab also calculates Spectral Tilt by returning the slope of a linear regression between frequency and spectral density in power spectra of each sound [28].
3.8. Speech Rate
VoiceLab measures speech rate using an updated implementation of an automated speech rate algorithm [25]. This should not be used with vowel sounds, but rather, multisyllabic speech like sentences or phrases. Note that this script, and any others noted in this paper, have been re-written in Python for VoiceLab’s batch processing engine. VoiceLab does not run Praat scripts. Everything is coded in Python.

3.9. Long Term Average Spectra (LTAS)
VoiceLab measures the following from LTAS from Praat: Mean (dB), slope (dB), local peak height (dB), standard deviation (dB), LTAS spectral tilt slope (dB/Hz), and Spectral Tilt intercept (dB).

4. Visualizations

4.1. Spectrograms
Praat creates greyscale spectrograms. VoiceLab software uses MatPlotLib [26] to automatically create high-quality, publication ready full colour spectrograms. See Fig 1. Users have the option to overlay pitch, intensity, and formant frequencies over the spectrogram as in Praat. VoiceLab allows the user to select from a wide variety of colour maps. See Figure 1.

4.2. LPC Power Spectra
VoiceLab creates power spectra of selected sounds and overlays an LPC curve on the plot. See Figure 2.

5. Voice Manipulations
VoiceLab provides several voice manipulations to help researchers automate and replicate stimuli creation. VoiceLab does the following:

- Manipulate Pitch
- Manipulate Formants
- Manipulate both Pitch and Formants together
- Resample Sounds
- Reverse Sounds
- Scale Intensity
- Trim Silences
- Trim Sound

VoiceLab uses default values similar to those first used in prior work [27] and can also be user-specified. Sound files are saved as .wav files in the folder specified by the user. VoiceLab automatically sets manipulation parameters using the same algorithm as used for measuring voice pitch and formants. This is imperative for automatic voice manipulations. Often when lowering the pitch of male voices, values within the manipulated voice pitch goes below Praat’s 75 Hz pitch floor default. For example, if lowering the pitch of a voice that is on average 80-95Hz, we need to drop the pitch floor in creating the manipulation object and measuring the result. VoiceLab does this automatically, and in opposite fashion for higher-pitched voices. As the formant manipulation is based on separating source and filter, an accurate pitch characterization is also very important. Automatic formant settings are also used here. Trimming silences is based on detecting sounds above an amplitude threshold.

6. Using VoiceLab software

6.1. The Load Voices Window
VoiceLab software was designed for ease of use. Users first load their voice files (wav, mp3, .aiff, ogg, aic, au, nist, or flac) in the “Load Voices” tab using the “Load Voices” button. Users can also remove voices using the “Remove Voices” Button. Users can press “Start” to run the default analyses (Duration, Pitch, Subharmonics, Harmonics-to-Noise Ratio, Jitter, Shimmer, Cepstral Peak Prominance (CPP), Formants, Vocal Tract Estimates, Intensity, Spectral Tilt, RMS Energy, Spectral Shape). Non-defaults include: Speech Rate (for multisyllabic sounds only), LTAS, MFCCs, Voice Manipulations (Pitch, Formants, Pitch + Formants), Trim Sounds, Resample Sounds, Reverse Sounds, Scale Intensity), and Visualizations (Spectrograms and Power Spectra) Users can also play voices and stop the playing of a sound by pressing the appropriate buttons. Help brings up a browser window with our help page: https://voice-lab.github.io/VoiceLab/. See Figure 3.

6.2. The Settings Window
Users can use the default and automated analyses and analysis parameters, or go to the “Settings” tab and select their own analyses/visualizations/manipulations. See figure 4.
6.3. The Results Window

Results automatically populate in the "Results" tab. Pressing the "Save results" button saves two files: The analysis results are recorded in \texttt{voicelab\_results.xlsx}, and the analysis parameters are recorded in \texttt{voicelab\_settings.xlsx}. The analysis parameters allow full replication of all analyses. All analyses using Parselmouth can also be fully replicated in Praat. VoiceLab does not round or truncate decimal places because we believe in providing full results and open data. The number of values in decimal places are not misleading, rather they are exactly the same one would get if running this analysis in Praat. It is recommend users round these values themselves.

See figure 5.

7. Comparison to Praat

VoiceLab uses Parselmouth [28], a set of Python bindings to Praat’s [5] source code. All analyses in VoiceLab that use Praat code are exactly the same as using Praat (given the same analysis parameters) because under the hood, VoiceLab is running Praat’s source code, not an approximation of it. Full validation analyses demonstrating this can be found in supplemental materials here: https://osf.io/3wr6k/files/.

8. Intended and Extended Usage

VoiceLab’s algorithms are best tuned for modal adult human vowel sounds. You can adjust parameters for other types of speech, speech from children, non-human vocalizations, etc. VoiceLab recommends people use the formant analysis in VoiceLab only for short vowel sounds, but VoiceLab also offers automated LTAS and MFCC measurements from Praat that might be better options for longer sounds. Measures of speech rate [25]) should only be used on multi-syllabic sounds, and won’t produce results if the sound is to short and/or monosyllabic.

9. Software architecture

VoiceLab software is written predominantly with a Python package called Parselmouth [28], which gives access to Praat’s C/C++ code in Python. Subharmonic and Energy measures come from Open Sauce, a Python port of Voice Sauce [6]. VoiceLab offers a friendly user interface where no installation is needed, works across operating systems, automatically outputs all parameters needed to replicate analyses in Praat or Voice Sauce, and provides clean data files.

10. Conclusions

VoiceLab Software provides reproducible automated acoustical analyses. This software allows non-experts to comfortably and easily measure voices. Due to the automatic nature of the software, we increase inclusivity in acoustical analysis by taking gender out of the equation. Analyses are largely based on Praat and Voice Sauce, and are all open-source, so users do not have to trust the underlying algorithms.

11. Acknowledgements

I thank Yannik Jadoul, Paul Boersma, and Yen Shue for their many helpful discussions. This research was funded by a Local Software Support Pilot Initiative Grant from CANARIE. I also can’t thank enough Oliver Cook, who was hired on that grant for writing the original GUI and data pipeline. Thanks also to Benedict Jones for comments and feedback on drafts of this manuscript.

12. References

Figure 5: The VoiceLab Output Window


