



Validation of a Python-Based Graphical User Interface for Calculation of Relative Fundamental Frequency

Mara R. Kapsner-Smith¹, Aideen Gill^{2,3}, Abigail Muzila^{1,4}, Tamara Polyakova^{1,4}, Austeja Rubeski^{1,4}, Joshua Kline^{2,3}, Cara E. Stepp^{1,4,5}, Jenny Vojtech^{1,2,3}

¹Department of Speech, Language, and Hearing Sciences, Boston University, Boston, MA 02215

²Delsys, Inc., Natick, MA 01760

³Altec, Inc., Natick, MA 01760

⁴Department of Biomedical Engineering, Boston University, Boston, MA 02215

⁵Department of Otolaryngology – Head and Neck Surgery, Boston University School of Medicine, Boston, MA 02215

Abstract

Objectives: The purpose of this study was to validate PyRFF, a semi-automated software for calculating relative fundamental frequency (RFF). PyRFF was developed in Python, a widely used open source programming language, making it freely accessible and broadly applicable for researchers and clinicians. By providing an accessible tool for RFF extraction, PyRFF has the potential to facilitate wider adoption of RFF measures in both research and clinical settings.

Study Design: Validation study

Methods: To assess the accuracy of PyRFF, RFF measures were extracted using both PyRFF and a previously validated, semi-automated algorithm implemented in MATLAB. The outputs from the two programs were then compared using statistical measures of error, including root mean square error (RMSE) and mean bias error (MBE).

Results: The results demonstrated near-identical outputs between PyRFF and the MATLAB-based algorithm. RMSE and MBE values were close to zero for all comparisons, indicating minimal discrepancy between the two methods.

Conclusions: PyRFF offers a valid, freely available software platform for extracting RFF measures, ensuring accessibility for a broad user base of researchers, clinicians, and students.

Corresponding Author: Mara Kapsner-Smith, Department of Speech, Language, and Hearing Sciences, Boston University, 111 Cummington Mall, Room 225 – 2nd floor, Boston, MA 02215, mkapsner@bu.edu.

Conflict of Interest

The authors have declared that no competing interests existed at the time of publication.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

The availability of free, user-friendly software is essential for advancing research and clinical application of RFF in voice assessment. By providing an alternative to MATLAB-based methods, PyRFF lowers financial and technical barriers, promoting further exploration of RFF as a potential tool in voice disorder assessment and treatment.

Keywords

voice disorders; relative fundamental frequency; acoustic measures; vocal hyperfunction; voice; dysphonia

Introduction

Vocal hyperfunction is characterized by muscle tension in and around the larynx during voicing, and may occur as a primary cause of dysphonia, or as a compensation for other conditions that impact the voice (e.g., vocal fold paralysis; Belafsky et al., 2002; Hillman et al., 2020). Clinical assessment of vocal hyperfunction relies heavily on subjective measures such as auditory-perceptual judgments of vocal effort, laryngeal palpation, and visual-perceptual judgments of videostroboscopy, which are prone to bias and poor reliability (Eadie et al., 2011; Sauder et al., 2020; Stepp, Heaton, et al., 2011; Yiu et al., 2014; Zraick et al., 2011). However, relative fundamental frequency (RFF) is an objective measure that has shown promise for clinical research and potentially for assessment of voice disorders including hyperfunctional voice disorders, hyperkinetic dysarthria, and adductor-type laryngeal dystonia (Eadie & Stepp, 2013; Goberman & Blomgren, 2008; Heller Murray et al., 2017; Kapsner-Smith et al., 2022; Roy et al., 2016; Stepp, 2013; Stepp et al., 2010). As a non-invasive and objective acoustic measure, RFF may provide an indirect measure of laryngeal tension (McKenna et al., 2016; Serry et al., 2021; Stepp et al., 2010). Previous work has validated a semi-automated graphical user interface (GUI) for calculating RFF (Lien et al., 2017; Vojtech et al., 2019).¹ However, because the GUI was developed in MATLAB (MathWorks, 2021), user accessibility is limited by the cost of software licenses and user comfort with running scripts in a development environment. A free software package that can be implemented independently of specific code development platforms, without requiring expensive software licenses or other secondary software would facilitate broader validation and adoption of RFF measures.

RFF is a measure of cycle-to-cycle changes in the fundamental frequency (f_0) of vocal fold vibration in the last 10 cycles before (offset) and the first 10 cycles after (onset) phonation ceases for production of a voiceless consonant. In people with typical voices, f_0 decreases slightly during offset cycles relative to steady-state phonation and increases at onset before gradually returning to steady-state levels (Stepp et al., 2010; Watson, 1998). RFF is thought to be affected by changes in vocal fold posture, laryngeal muscle tension, and transglottal pressure that occur during transitions between voiced and voiceless sounds (Ladefoged, 1967; Serry et al., 2021; Stevens, 1977; Watson, 1998). RFF has been shown to modulate with emotional state in people with typical voices, likely due to change in muscular activation (van Mersbergen & Lanza, 2019). In people with vocal hyperfunction,

¹Available at: <https://sites.bu.edu/stepplab/research/rff/>

RFF measures may be atypical due, in part, to baseline elevated laryngeal tension (Heller Murray et al., 2017). Previous work has shown that RFF values are lower in speakers with voice disorders characterized by laryngeal tension including vocal hyperfunction (Ferrán et al., 2024; Stepp et al., 2010), Parkinson's disease (Stepp, 2013), and laryngeal dystonia (Eadie & Stepp, 2013) compared to speakers without voice disorders. Measures derived from RFF, such as the values of the boundary cycles (offset cycle 10 and onset cycle 1), discriminate between speakers with and without hyperfunctional voice disorders and may have diagnostic utility (Cheema et al., 2024; Heller Murray et al., 2017; Kapsner-Smith et al., 2022). Further, there is evidence that RFF measures are sensitive to voice therapy outcomes (Roy et al., 2016; Stepp, Merchant, et al., 2011). RFF also decreases in speakers without voice disorders in adverse speaking conditions such as wearing a face mask and in association with other markers of vocal fatigue (McKenna, Kendall, et al., 2022; McKenna et al., 2023), and increases with training and implementation of vocal health strategies in the workplace (McKenna et al., 2024). Thus, RFF may provide a clinically meaningful indirect measure of laryngeal tension.

To date, clinical studies of RFF have been limited to controlled settings in a fairly small number of labs that have access to MATLAB software (McKenna, Vojtech, et al., 2022). However, to fully establish the clinical utility of RFF, independent validation studies are needed across diverse clinical environments and populations. Expanding access to RFF analysis requires moving beyond MATLAB, which is costly and less accessible, to free alternatives. Although executable GUI-based MATLAB software may be produced using MATLAB Compiler to run outside the development environment, the resulting executables require installation of the MATLAB Runtime Environment (MCR). The MCR is a large (~1.5 GB) dependency that must be installed separately for each platform and it must match the MATLAB version used for compilation. This creates two key barriers in clinical and educational settings: 1) Technical burden, as users must locate, install, and maintain a compatible runtime on their system; and 2) Feasibility due to file size and permissions. Hospital and clinic IT policies often restrict installation of large third-party runtimes, especially those that update frequently. Therefore, this study aimed to validate PyRFF, a Python-based tool for calculating RFF. Python is an open-source programming language that does not require costly software licenses. Python software can be packaged as lightweight, standalone executables that run across platforms without additional dependencies. By creating a user-friendly program that runs independently without specialized software, this study aimed to make RFF analysis more accessible to researchers and clinicians.

The development of PyRFF involved translating the freely available MATLAB-based “aRFF-AP” algorithm into Python. The algorithm was designed to extract RFF values from vowel–voiceless consonant–vowel utterances containing the fricative /f/ (i.e., VfV). To calculate RFF, the algorithm comprises subroutines to (1) extract voice f_0 and pitch strength contours for each audio file; (2) calculate probable fricative locations based on spectral energy content; (3) identify the last 10 cycles of phonation offset and the first 10 cycles of phonation onset using the f_0 and pitch strength contours; and (4) calculate RFF measures, which are saved to an Excel spreadsheet (Microsoft Corporation, 2024). Besides relying on MATLAB to run it, users had to manually manipulate parameters within a script, such as setting the minimum and maximum f_0 values and specifying the absolute directory

paths for audio input and RFF output folders. This required users to have a basic level of programming knowledge. In contrast, the updated version of PyRFF features a user-friendly GUI that prompts users to easily input values and select files or directories, eliminating the need to manually edit script parameters or type paths. This new interface streamlines the process, making it accessible to users without requiring prior knowledge of the MATLAB development environment or scripting.

Because implementing the algorithms used for semi-automated RFF calculation in Python required recreating core signal processing routines, validation against the original MATLAB-based implementation (aRFF-AP) was essential. Python and MATLAB use different signal processing libraries and computational backends, which can lead to subtle differences in outputs even when algorithms are theoretically identical. Thus, validating PyRFF using aRFF-AP as the gold standard ensured that the Python-based tool preserved the integrity of the original processing steps and produced comparable and reliable results. Using a test set of voice samples from individuals with and without voice disorders, we compared the outputs of PyRFF to those from the validated aRFF-AP tool. We hypothesized that the performance of the two programs would not exhibit clinically meaningful differences based on error comparisons between the Python and MATLAB algorithms in three areas: (1) automated f_0 and pitch strength contour calculations, (2) consistency of automated fricative identification, and (3) RFF calculation. Validation of the Python-based RFF GUI (PyRFF) will improve accessibility of the measures, facilitating further research and potential clinical applications.

Methods

Informed consent was obtained, and the study carried out in compliance with the Boston University Institutional Review Board (protocol #2625) or University of Washington Institutional Review Board (protocol #36181).

Participants

Extant speech recordings were selected from a database of study participants, comprising 100 speakers with a range of voice disorders and 100 control speakers with typical voices (see Table 1 for demographic information). The group with voice disorders required a diagnosis from a referring laryngologist. Voice diagnoses are listed in Table 2. The control group had no self-reported history of speech, language, or hearing disorders, and were age- and sex-matched within a five-year range.

For each participant, a speech-language pathologist—who was blinded to the study conditions—completed the 100-mm visual analog scale for overall severity of dysphonia from the Consensus Auditory-Perceptual Evaluation of Voice (CAPE-V) using the nonlinearly placed textual severity labels as originally published (American Speech-Language-Hearing Association, 2002). Mean overall severity of dysphonia scores were as follows: 13.0 (SD = 8.3, range = 1–39) for the control group, 19.0 (SD = 19.6, range = 0–100) for the voice disorder group.

Speech Recordings

All signals were acquired digitally, and analysis occurred offline. Participants were recorded in a sound-attenuated booth, waiting area of a clinic, or quiet room at one of three locations using Shure (Niles, IL, USA) microphones: (a) at Boston Medical Center with a dynamic headset microphone (model WH20XLR), (b) at Boston University with a condenser headset microphone (model SM35XLR), (c) at the University of Washington with a dynamic headset microphone (model WH20XLR).

Acoustic signals were sampled at 44.1 kHz with 16-bit resolution. Speech recordings consisted of three tokens for each of three vowel–voiceless consonant–vowel combinations with the voiceless consonant, /f/ (i.e., VfV tokens): /αfα/, /ifi/, and /ufu/. Participants were instructed to produce the VfVs with equal stress on both syllables. Audio recordings were segmented into three .WAV files, one for each VfV type.

Procedures

Converting the RFF Algorithm from MATLAB to Python: The MATLAB-based aRFF-AP algorithm described in Vojtech et al. (2019) was translated in Python 3.8 (Python Software Foundation, 2019). See Lien et al. (2017) and Vojtech et al. (2019) for details regarding development and refinement of the algorithm.

The structural organization of aRFF-AP was schematized to identify the individual scripts that comprise the algorithm. In total, the algorithm consisted of one main script entitled **main_RFF**, four nested scripts (**find_fo_aswipep**, **find_fric_aswipep**, **find_RFFcycles_aswipep**, **RFFoutput**), and 12 dependencies (i.e., functions upon which the main or nested scripts rely to run; Fig. 1).

Ensuring numerical equivalence across platforms required recreating core signal processing routines. MATLAB's built-in functions (e.g., *filter*, *filtfilt*, *designfilt*, and *butter*) rely on proprietary backend optimizations and often include implicit scaling, padding, or normalization steps. These operations are not always transparent, which posed challenges in replicating identical behavior in Python. In contrast, Python's *scipy* and *numpy* libraries implement signal processing with stricter requirements for input formatting and without MATLAB's automatic handling of edge artifacts or default filter orderings.

For instance, converting the low-pass and high-pass filters used to process pitch strength and energy contours required explicit replication of phase and gain behavior by careful matching of MATLAB filter design parameters and zero-phase filtering behavior using *scipy.signal.filtfilt*. We encountered minor—but meaningful—differences in edge behavior that affected pitch strength estimation until manually corrected. Similarly, frequency-domain analyses required tuning Python's FFT operations to match MATLAB's default windowing conventions and resolution handling. These platform-specific differences meant that achieving numerical equivalence required deep reconstruction of signal processing components rather than direct code translation.

In addition, the dependencies used in aRFF-AP had to be translated and, in some cases, reimplemented in Python. Pitch estimation using Auditory-SWIPE' in MATLAB

benefits from proprietary interpolation and error-reduction heuristics. In Python, we ported the open-source, MATLAB-based Auditory-SWIPE' from scratch using the published specifications by Camacho (2007), aligning sampling windows, harmonic weights, and temporal smoothing explicitly. The translation of the algorithms was a collaborative, iterative process. The fricative detection process, which relies on spectral energy ratios, was also reimplemented using `numpy.fft` and band-specific filtering. Ensuring numerical stability and matching MATLAB's output required validation against edge cases and atypical fricative durations. Initially, the authors (A.M., T.P., and A.R.) converted the code line-by-line, followed by more comprehensive efforts (led by A.G. and J.V.) to ensure functional equivalence when transitioning from MATLAB's built-in functions to Python.

Initial validation used a “bottom-up” approach where individual Python functions were tested before integrating the complete workflow. To ensure consistent testing conditions, we generated reference variables by running aRFF-AP on a single /afa/ (“aafaa”) recording in MATLAB, then used these same inputs to test the corresponding Python functions. This approach allowed systematic validation of each processing step before testing the complete algorithm.

After the 12 dependencies were converted, the five scripts involved in the overarching algorithm workflow were converted. The main *_RFF* script, which manages the overall process, allows users to define key parameters for signal processing (such as input and output file paths, f_0 bounds, expected number of VfVs, and methods for displaying results). The Python version of this script was updated with prompts to simplify user input without requiring modifications to code. From here, there are three primary routines that prepare for RFF calculation. The first routine, *find_contours_aswipep*, estimates voice f_0 and pitch strength over time for the selected file. This routine uses Auditory-SWIPE' for f_0 estimation (Camacho, 2007), based on an evaluation of f_0 estimators conducted by Vojtech et al. (Vojtech et al., 2019). The second routine, *find_fric_aswipep*, automatically locates the probable locations of the voiceless consonant /f/ in the VfV tokens within each sound file using ratios of high- and low-frequency energy. The user is then prompted to manually confirm or adjust the locations. The third, *find_RFFcycles_aswipep*, uses the fricative location(s) identified in the second routine, as well as manual corrections provided by the user via a GUI, to determine 10 voicing cycles before voice offset (i.e., before the /f/) and 10 after voice onset (i.e., after the /f/). Starting from the center of each fricative, a sliding window the length of the reciprocal of the f_0 estimate (i.e., the period) searches in both directions for peaks and troughs to identify vocal cycles in the offsets and onsets of the vowels (i.e., boundary cycles). Finally, the *RFFoutput* routine uses the cycle data to calculate RFF or reject the instance based on criteria defined in Lien et al. (2017). The f_0 of each offset cycle is calculated in semitones relative to offset cycle 1 and the f_0 of each onset cycle is calculated relative to onset cycle 10. Offset cycle 1 and onset cycle 10 are considered “reference” cycles since they are closest to the midpoint of either vowel in a VfV token; in this way, they represent the f_0 of (pseudo)steady-state phonation.

Updating the GUI Architecture: Although aRFF-AP includes a GUI for viewing and correcting fricative locations (Fig. 2A), the user was required to manually edit code to

specify analysis parameters such as f_0 bounds and file paths. PyRFF integrates these functions into a comprehensive GUI that includes:

- Integrated file and folder selection dialogs with field validation for f_0 bounds and other analysis parameters (Fig. 2B)
- Navigation controls for reviewing previous analyses and resetting automatically detected fricative locations, with improvements based on user feedback (Fig. 2C)
- Automated error logging with user-friendly messages
- One-click export of results to Excel format

These improvements eliminate the need for programming knowledge while maintaining the same analytical rigor as the original MATLAB implementation.

Evaluating the Accuracy of RFF Algorithm Translation: After translating the entire RFF codebase from MATLAB to Python—resulting in the creation of the new RFF GUI called “PyRFF”—the accuracy of the translation was evaluated using the 200-speaker dataset. Audio files were processed with both aRFF-AP and PyRFF to extract voice onset and offset RFF measures. Three main evaluations were performed: (1) extraction of f_0 and pitch strength contours via *find_contours_aswipep*, (2) automated identification of fricative locations using *find_fric_aswipep*, and (3) RFF calculations and/or token rejections (via *RFFoutput*) based on the same fricative locations as inputs to *find_RFFcycles_aswipep*.

To eliminate potential variability from user input during validation, a trained analyst (M.K.S.) manually identified and corrected fricative locations using the aRFF-AP interface. These corrected locations were then used as identical inputs for both MATLAB and Python RFF calculations, ensuring that any differences in output reflected algorithmic rather than user-input variations. The resulting RFF output for each Vfv token included semitone (ST) values for voice offset cycles 1–10 (the last 10 cycles before the voiceless consonant) relative to offset cycle 1, and onset cycles 1–10 (the first 10 cycles after the voiceless consonant) relative to onset cycle 10. Any vowels that could not be analyzed were assigned a rejection code with a specific reason for failure, and these instances were compared between the two algorithm versions.

Statistical Analysis

The f_0 and pitch strength contours generated by *find_contours_aswipep* were compared across audio signals using two error metrics: mean bias error (MBE) and root mean square error (RMSE). MBE estimated the average deviation between the contours from each algorithm, whereas RMSE measured the precision of the contours, regardless of whether the errors were positive or negative.

Automated fricative locations were compared by calculating the absolute time difference (in seconds) between the locations identified by the two algorithms. To normalize this comparison, the time difference was converted into a relative distance, expressed in terms of the average f_0 measured in the audio file. This provided a measure of how many voice cycles separated the fricative locations.

For RFF values, results from the *find_RFFcycles_aswipep* and *RFFoutput* scripts were compared when using the same manually corrected fricative locations as inputs. The comparison focused on the mean values for voice offset cycles (1–10), onset cycles (1–10), as well as for cycle 10 of voice offset and cycle 1 of voice onset. Previous research (e.g., Heller Murray et al., 2017) suggests that these particular cycles may be most sensitive to group differences in some clinical populations, which is why they were prioritized.

To assess the concurrent validity of PyRFF compared to the validated aRFF-AP algorithm, we calculated Pearson's r , RMSE, and MBE for the extracted RFF values from both programs. These statistical measures were chosen to evaluate the strength and accuracy of the relationship between the two algorithms' outputs, to confirm that PyRFF performs similarly to the original aRFF-AP algorithm.

Results

For descriptive purposes, RFF cycle averages are presented in Fig. 3 for each program. The group means and confidence intervals were nearly identical for the two programs. The type and number of RFF cycle rejections for each program are presented in Table 3. The cycle rejection results were nearly identical.

Contour Extraction and Fricative Locations

Results of comparisons of f_0 and pitch strength contours extracted automatically in the MATLAB software and PyRFF are shown in Table 4. The mean RMSE and mean MBE were near zero for both comparisons. The mean difference in automated (i.e., not manually verified or corrected) fricative locations between the two programs was <0.01 cycles ($SD = 0.01$). All fricative locations were verified manually. Of 600 audio files, 129 required manual adjustments of one or more fricative locations.

RFF Calculations

Results for comparison of RFF calculations between the MATLAB software and PyRFF are shown in Table 5. All RMSE and MBE values were close to zero, and Pearson's r was 1.00 for all comparisons, suggesting near-perfect correspondence between the output of the two programs.

To assess the robustness of RFF analysis across different voice qualities, we compared rejection rates between speaker groups as part of a *post hoc* analysis. Higher rejection rates in dysphonic voices would indicate potential limitations for clinical application, whereas similar rates would suggest robust performance across voice disorders. Results are shown in Table 6. Overall, there were approximately 7% more rejections in the voice disorder group than the control group. There were two speakers in the voice disorder group with no analyzable RFF cycles. All control speakers had analyzable RFF cycles.

Discussion

RFF calculations were extracted using a software package written in Python (Python Software Foundation, 2019) and were compared to output from a previously validated

semi-automated algorithm (Lien et al., 2017; Vojtech et al., 2019) written in MATLAB (MathWorks, 2021). Results were compared for absolute errors (MBE, RMSE) as well as consistency of RFF values (Pearson's r). The two programs performed similarly, with very small, average errors in f_0 and pitch strength contour extraction (mean RMSE = 0.09 and <0.01, respectively; mean MBE <0.01), and very small average differences in automated fricative location (<0.01 cycles). After manual adjustments of fricative locations, RMSE and MBE statistics calculated on RFF calculations indicated near zero error between the two programs, with perfect correlation. Although the two programs perform the same data extraction and calculation procedures, differences can occur across platforms (e.g., MATLAB vs. Python), for example due to differences in how functions are written, such as filters used for audio processing, or small differences in how floating point and complex numbers are handled in the language. Thus, this study was necessary to ensure that differences were minimized and non-significant between the two implementations of RFF.

One challenge in the acoustic analysis of dysphonic voices is the robustness of a given measure to noisy or aperiodic signals (Titze, 1995). As a *post hoc* analysis, the number and type of RFF cycle rejections were compared between speakers with and without voice disorders. There were 7% more RFF cycle rejections in the voice disorder group than the control group, with the majority of these rejections being due to the algorithm being unable to locate periodic voice cycles during signal pre-processing (*find_fric_aswipep*). This resulted in 2/100 speakers in the voice disorder group with no analyzable RFF cycles, compared to 0/100 in the control group. These results reflect a relatively high success rate for acoustic analysis of dysphonic voices, with only 2% of the voice disorder sample that did not have any usable cycles.

Although these findings suggest that RFF may be a fairly robust acoustic measure for these signal types, it is important to consider the perceived severity of dysphonia in the analyzed sample. The MATLAB-based algorithm was enhanced in Vojtech et al. (2019) through rule-based signal processing incorporating the measure of pitch strength (Camacho, 2008) to account for differences in dysphonia severity. However, the algorithm may still struggle with more aperiodic voices. In the voice disorder group, severity spanned the full scale (0–100), whereas in the control group, it ranged only from 1 to 39. Although this distinction is important, the mean dysphonia severity values were relatively similar between groups (19 vs. 13), suggesting that, on average, the voice disorder group did not exhibit severe dysphonia—an important consideration when interpreting these findings.

Development of a standalone Python executable for semi-automated RFF calculation has several technical advantages over MATLAB. Deployment of executable software compiled in MATLAB requires that the user install a MATLAB runtime environment (MCR), a large dependency that must be installed separately for each platform and must match the MATLAB version used for compilation. These technical barriers may be prohibitive in clinical and education settings due to IT policies and user knowledge requirements. Further, it may complicate deployment of updated versions of software compiled in newer versions of MATLAB. In contrast, PyRFF was implemented in pure Python and compiled into a small, cross-platform standalone executable. It requires no runtime environment or

administrator permissions. This makes PyRFF substantially more accessible for clinicians, students, and researchers unfamiliar with software development.

It is important to note that the scope of this paper is limited to concurrent validation of PyRFF relative to the previously validated MATLAB semi-automated RFF algorithm (Vojtech et al., 2019). This concurrent validation provides no information about the validity or interpretation of RFF measures themselves, or reliability of user input. Information about validity, interpretation, and potential clinical utility of RFF measures has been published in a number of previous studies (for a review related to voice disorders, see McKenna, Vojtech, et al., 2022). Future studies are recommended to assess usability of the enhanced interface and optimization benchmarking.

Our findings support the concurrent validity of PyRFF compared to a previously validated semi-automated GUI for RFF extraction (Lien et al., 2017; Vojtech et al., 2019). PyRFF is freely available for download at <https://sites.bu.edu/stepplab/research/rff/> and requires no experience with coding or development environments to install and use. The availability of free and user-friendly software to extract RFF is a crucial step for clinical validation and application. Further research—facilitated by the availability of this software—may lead to an affordable, accessible, and non-invasive clinical tool for assessment of vocal hyperfunction.

Conclusions

Although RFF has shown promise as an objective acoustic indicator of vocal hyperfunction, the accessibility of the measure was previously hindered by high-cost software licensing and the need for basic programming skills. This study provides evidence for the concurrent validity of PyRFF, a freely available software package for RFF extraction based on the previously validated semi-automated MATLAB algorithm, aRFF-AP. PyRFF offers a user-friendly, cost-free solution for measuring RFF to facilitate downstream clinical validation and implementation.

Acknowledgments

This work was supported by the National Institute on Deafness and Other Communication Disorders under grant R01DC015570 (Stepp) and R43DC019585 (Vojtech), as well as by the De Luca Foundation.

Abbreviations

GUI	graphical user interface
RFF	relative fundamental frequency
VfV	vowel-fricative-vowel
f_0	fundamental frequency

References

- Belafsky PC, Postma GN, Reulbach TR, Holland BW, & Koufman JA (2002). Muscle Tension Dysphonia as a Sign of Underlying Glottal Insufficiency. *Otolaryngology–Head and Neck Surgery*, 127(5), 448–451. 10.1067/mhn.2002.128894 [PubMed: 12447240]
- Camacho A (2007). SWIPE: A sawtooth waveform inspired pitch estimator for speech and music [University of Florida, Gainesville, FL, USA].
- Cheema AJ, Marks KL, Ghasemzadeh H, Van Stan JH, Hillman RE, & Mehta DD (2024). Characterizing Vocal Hyperfunction Using Ecological Momentary Assessment of Relative Fundamental Frequency. *Journal of Voice*. 10.1016/j.jvoice.2024.10.025
- Eadie TL, Sroka A, Wright DR, & Merati A (2011). Does Knowledge of Medical Diagnosis Bias Auditory-Perceptual Judgments of Dysphonia? *Journal of Voice*, 25(4), 420–429. 10.1016/j.jvoice.2009.12.009 [PubMed: 20347262]
- Eadie TL, & Stepp CE (2013). Acoustic Correlate of Vocal Effort in Spasmodic Dysphonia. *Annals of Otolaryngology, Rhinology & Laryngology*, 122(3), 169–176. 10.1177/000348941312200305 [PubMed: 23577569]
- Ferrán S, Rodríguez-Zanetti C, Garaycochea O, Terrasa D, Prieto-Matos C, del Río B, Alzuguren MP, & Fernández S (2024). Relative Fundamental Frequency: Only for Hyperfunctional Voices? A Pilot Study. *Bioengineering*, 11(5), 475. 10.3390/bioengineering11050475 [PubMed: 38790342]
- Goberman AE, & Blomgren M (2008). Fundamental frequency change during offset and onset of voicing in individuals with Parkinson disease. *Journal of Voice*, 22(2), 178–191. 10.1016/j.jvoice.2006.07.006 [PubMed: 16950600]
- Heller Murray ES, Lien YS, Van Stan JH, Mehta DD, Hillman RE, Pieter Noordzij J, & Stepp CE (2017). Relative fundamental frequency distinguishes between phonotraumatic and non-phonotraumatic vocal hyperfunction. *Journal of Speech, Language, and Hearing Research : JSLHR*, 60(6), 1507–1515. [PubMed: 28595317]
- Hillman RE, Stepp CE, Van Stan JH, Zanartu M, & Mehta DD (2020). An Updated Theoretical Framework for Vocal Hyperfunction. *American Journal of Speech-Language Pathology*, 29(4), 2254–2260. 10.1044/2020_AJSLP-20-00104 [PubMed: 33007164]
- Kapsner-Smith MR, Díaz-Cádiz ME, Vojtech JM, Buckley DP, Mehta DD, Hillman RE, Tracy LF, Noordzij JP, Eadie TL, & Stepp CE (2022). Clinical Cutoff Scores for Acoustic Indices of Vocal Hyperfunction That Combine Relative Fundamental Frequency and Cepstral Peak Prominence. *Journal of Speech, Language, and Hearing Research*, 65(4), 1349–1369.
- Ladefoged P (1967). Three areas of experimental phonetics: Stress and respiratory activity, the nature of vowel quality, units in the perception and production of speech. Oxford University Press.
- Lien YS, Heller Murray ES, Calabrese CR, Michener CM, Van Stan JH, Mehta DD, Hillman RE, Noordzij JP, & Stepp CE (2017). Validation of an algorithm for semi-automated estimation of voice relative fundamental frequency. *Annals of Otolaryngology, Rhinology and Laryngology*, 126(10), 712–716. 10.1177/0003489417728088 [PubMed: 28849664]
- MathWorks. (2021). MATLAB 2021 [Computer software].
- McKenna VS, Gustin RL, Howell RJ, Patel TH, Emery MB, Kendall CL, & Kelliher NJ (2024). Developing Educational Health Modules to Improve Vocal Wellness in Mask-Wearing Occupational Voice Users. *Journal of Voice*, 38(4), 966.e9–966.e18. 10.1016/j.jvoice.2021.11.006
- McKenna VS, Heller Murray ES, Lien Y-AS, & Stepp CE (2016). The Relationship Between Relative Fundamental Frequency and a Kinematic Estimate of Laryngeal Stiffness in Healthy Adults. *Journal of Speech, Language, and Hearing Research : JSLHR*, 59(6), 1283–1294. 10.1044/2016_JSLHR-S-15-0406 [PubMed: 27936279]
- McKenna VS, Kendall CL, Patel TH, Howell RJ, & Gustin RL (2022). Impact of Face Masks on Speech Acoustics and Vocal Effort in Healthcare Professionals. *The Laryngoscope*, 132(2), 391–397. 10.1002/lary.29763 [PubMed: 34287933]
- McKenna VS, Patel TH, Kendall CL, Howell RJ, & Gustin RL (2023). Voice Acoustics and Vocal Effort in Mask-Wearing Healthcare Professionals: A Comparison Pre- and Post-Workday. *Journal of Voice*, 37(5), 802.e15–802.e23. 10.1016/j.jvoice.2021.04.016

- McKenna VS, Vojtech JM, Previtera M, Kendall CL, & Carraro KE (2022). A Scoping Literature Review of Relative Fundamental Frequency (RFF) in Individuals with and without Voice Disorders. *Applied Sciences*, 12(16), Article 16. 10.3390/app12168121
- Microsoft Corporation. (2024). Excel (Version Microsoft 365) [Computer software]. Microsoft Corporation.
- Python Software Foundation. (2019). Python Language (Version 3.8) [Computer software].
- Roy N, Fetrow RA, Merrill RM, & Dromey C (2016). Exploring the clinical utility of relative fundamental frequency as an objective measure of vocal hyperfunction. *Journal of Speech, Language, and Hearing Research : JSLHR*, 59(5), 1002–1017. 10.1044/2016_jslhr-s-15-0354 [PubMed: 27768175]
- Sauder C, Nevdahl M, Kapsner-Smith M, Merati A, & Eadie T (2020). Does the accuracy of case history affect interpretation of videolaryngostroboscopic exams? *The Laryngoscope*, 130(3), 718–725. [PubMed: 31124157]
- Serry MA, Stepp CE, & Peterson SD (2021). Physics of phonation offset: Towards understanding relative fundamental frequency observations. *The Journal of the Acoustical Society of America*, 149(5), 3654–3664. [PubMed: 34241131]
- Stepp CE (2013). Relative fundamental frequency during vocal onset and offset in older speakers with and without Parkinson's disease. *The Journal of the Acoustical Society of America*, 133(3), 1637–1643. [PubMed: 23464033]
- Stepp CE, Heaton JT, Braden MN, Jette ME, Stadelman-Cohen TK, & Hillman RE (2011). Comparison of neck tension palpation rating systems with surface electromyographic and acoustic measures in vocal hyperfunction. *Journal of Voice : Official Journal of the Voice Foundation*, 25(1), 67–75. 10.1016/j.jvoice.2009.08.001 [PubMed: 20347260]
- Stepp CE, Hillman RE, & Heaton JT (2010). The impact of vocal hyperfunction on relative fundamental frequency during voicing offset and onset. *Journal of Speech, Language, and Hearing Research*, 53(5), 1220–1226.
- Stepp CE, Merchant GR, Heaton JT, & Hillman RE (2011). Effects of Voice Therapy on Relative Fundamental Frequency during Voicing Offset and Onset in Patients with Vocal Hyperfunction. *Journal of Speech, Language, and Hearing Research*, 54(5), 1260–1266. 10.1044/1092-4388(2011/10-0274)
- Stevens KN (1977). Physics of laryngeal behavior and larynx modes. *Phonetica*, 34(4), 264–279. 10.1159/000259885 [PubMed: 594161]
- Titze IR (1995, February). Workshop on Acoustic Voice Analysis: Summary Statement. National Center for Voice and Speech, Denver, CO, USA.
- van Mersbergen M, & Lanza E (2019). Modulation of Relative Fundamental Frequency During Transient Emotional States. *Journal of Voice*, 33(6), 894–899. 10.1016/j.jvoice.2018.07.020 [PubMed: 30172671]
- Vojtech JM, Segina RK, Buckley DP, Kolin KR, Tardif MC, Noordzij JP, & Stepp CE (2019). Refining algorithmic estimation of relative fundamental frequency: Accounting for sample characteristics and fundamental frequency estimation method. *Journal of the Acoustical Society of America*, 146(5), 3184. 10.1121/1.5131025 [PubMed: 31795681]
- Watson BC (1998). Fundamental frequency during phonetically governed devoicing in normal young and aged speakers. *Journal of the Acoustical Society of America*, 103(6), 3642–3647. [PubMed: 9637045]
- Yiu EM, Lau VC, Ma EP, Chan KM, & Barrett E (2014). Reliability of laryngostroboscopic evaluation on lesion size and glottal configuration: A revisit. *Laryngoscope*, 124(7), 1638–1644. 10.1002/lary.24521 [PubMed: 24222186]
- Zraick R, Kempster GB, Connor NP, Thibeault S, Klaben BK, Bursac Z, Thrush CR, & Glaze LE (2011). Establishing validity of the Consensus Auditory-Perceptual Evaluation of Voice (CAPE-V). *American Journal of Speech-Language Pathology*, 20(1), 14–22. 10.1044/1058-0360(2010/09-0105) [PubMed: 20739631]

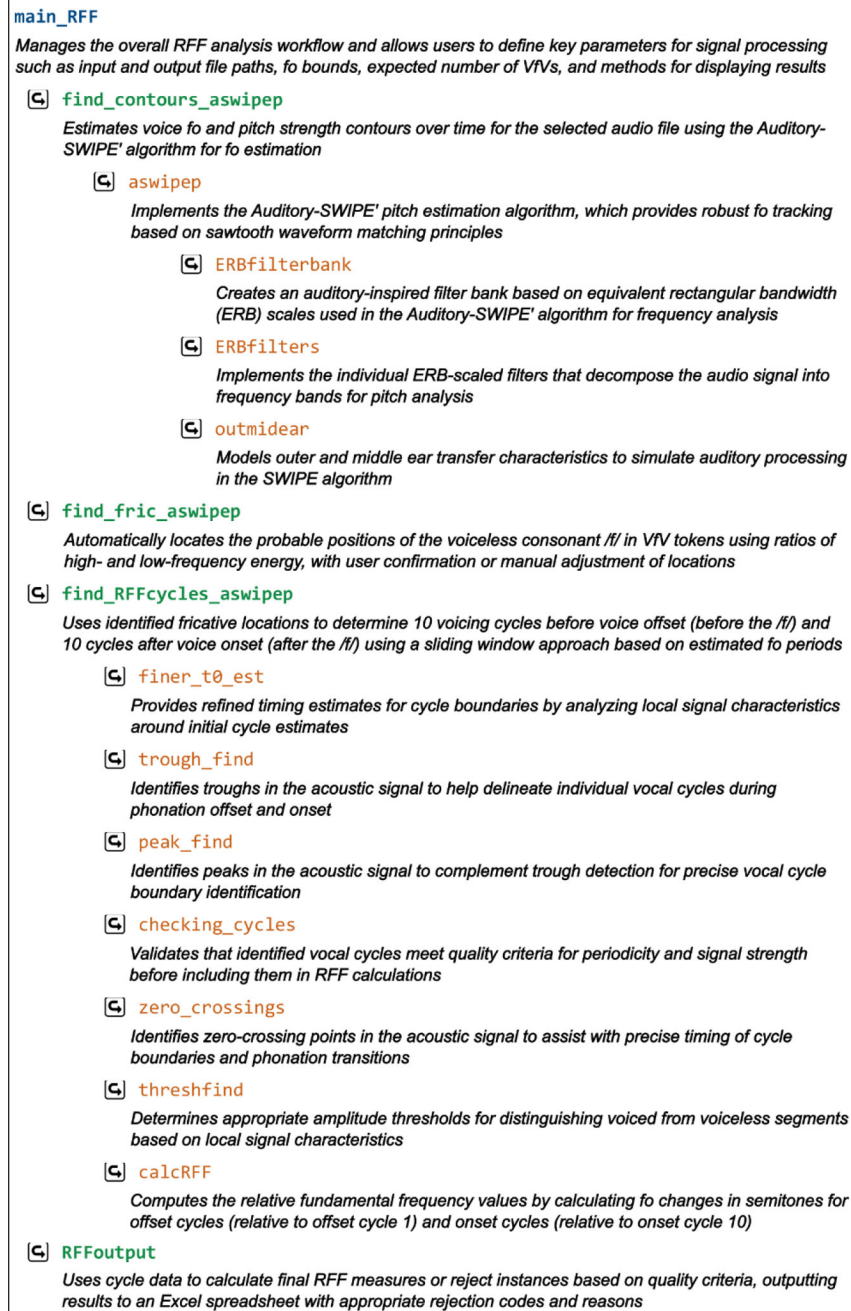
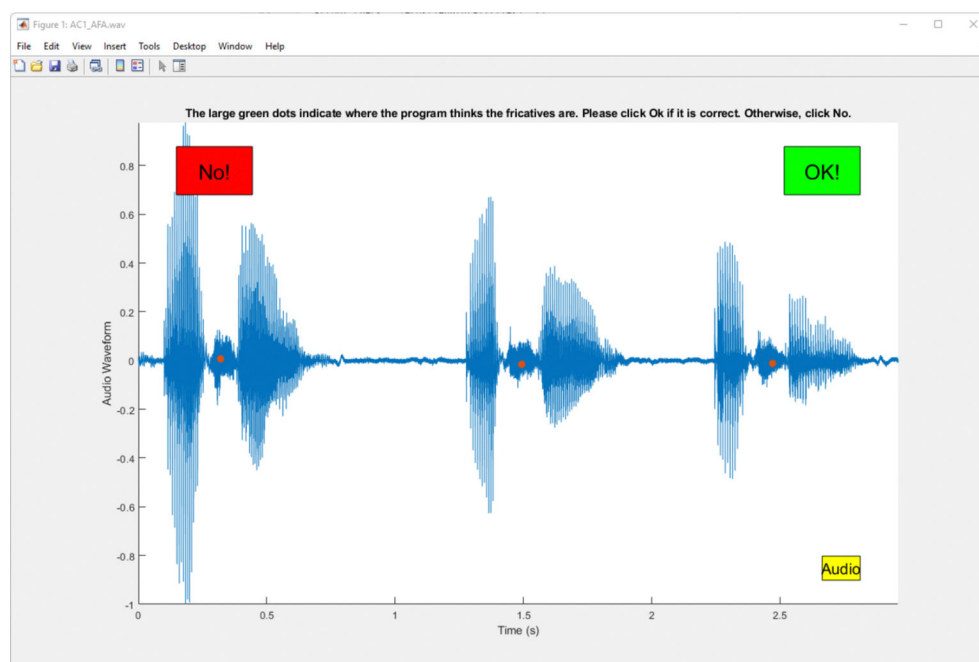


Figure 1.
Structural organization of the aRFF-AP algorithm.



PyRFF: Set Analysis Parameters and Audio Directory

Select Audio Directory:

C:\PyRFF

Minimum pitch (default 50 Hz):

Maximum pitch (default 400 Hz):

VCVs per sample (default 3):

☒ Combine VCVs in Output?

☐ Overwrite Existing?

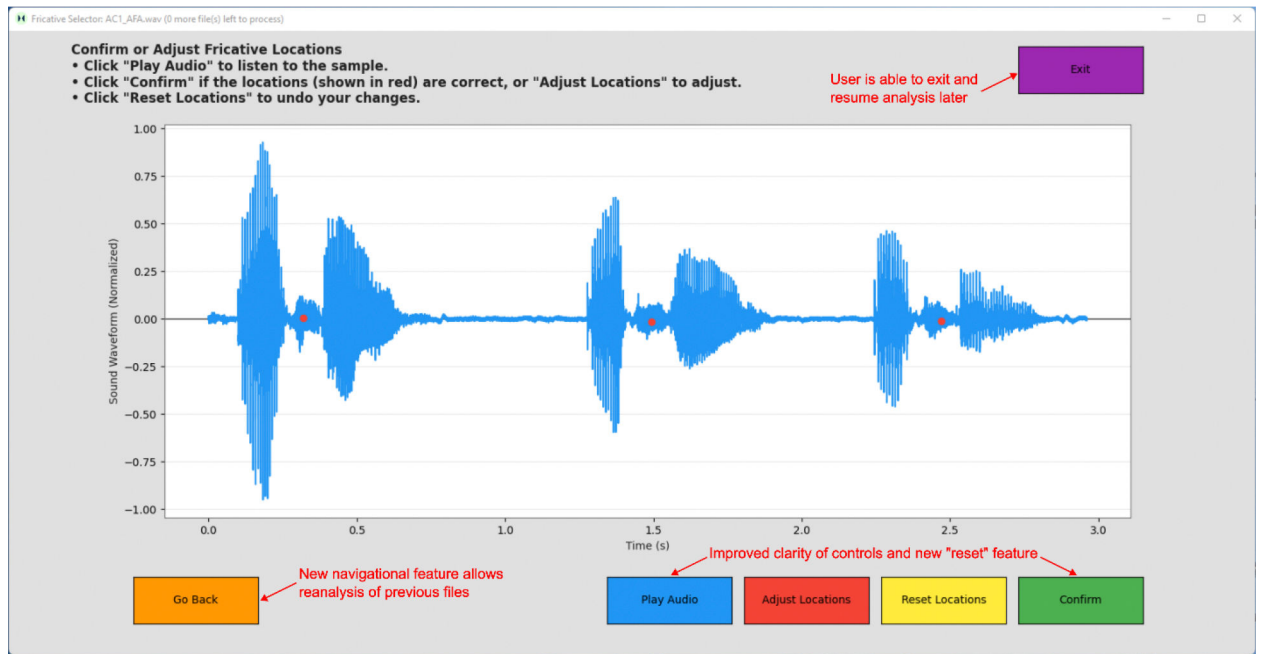


Figure 2.

a) Original aRFF-AP fricative selection GUI in MATLAB-based software. b) Conversion of hard coded inputs to PyRFF GUI, including file locations and analysis parameters. c) Improved PyRFF fricative selection GUI.

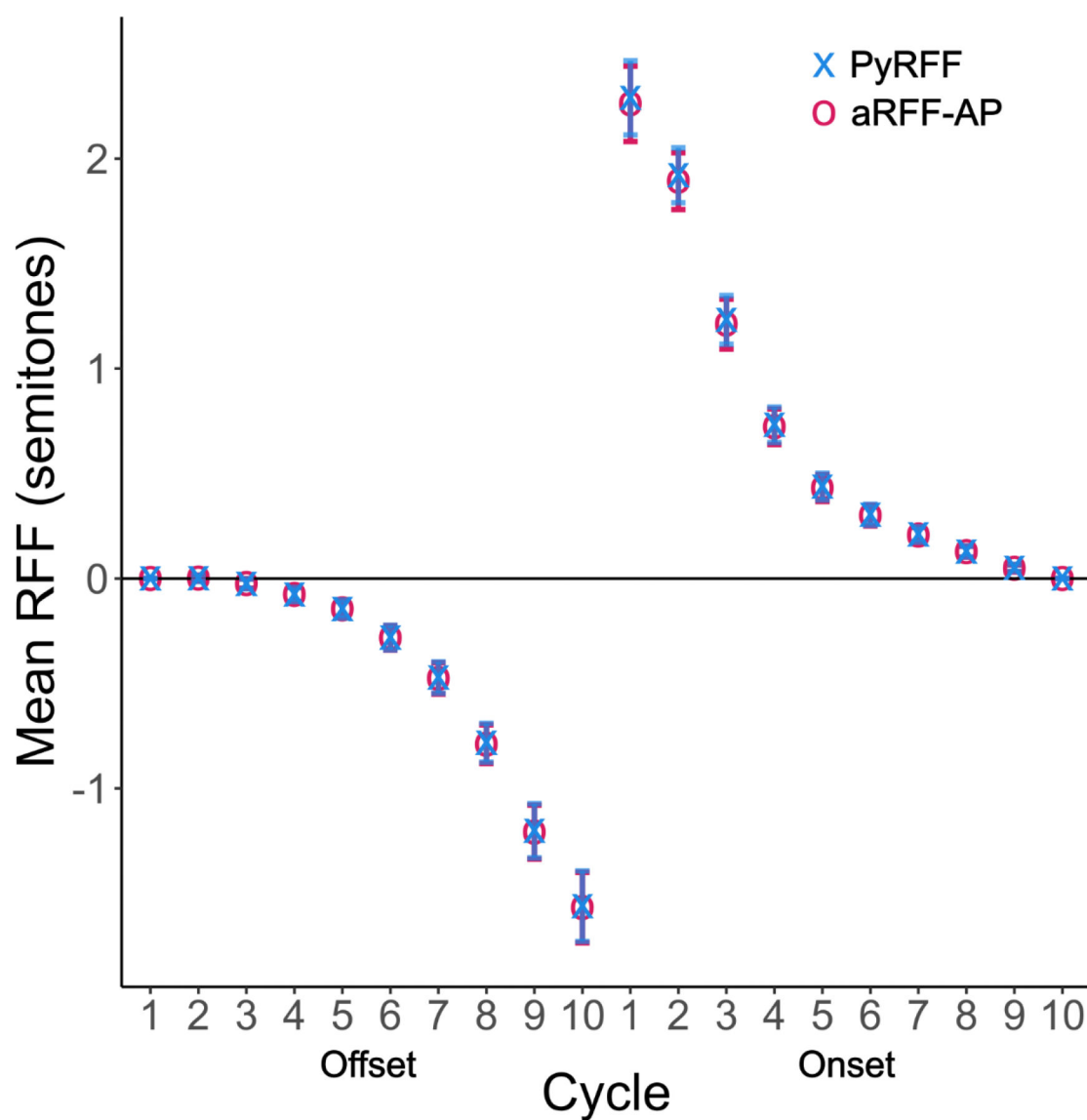


Figure 3.
Mean RFF cycle values and 95% confidence intervals from aRFF-AP and PyRFF output.

Table 1.

Demographics and perceptual ratings of voice severity for speakers included in the sample.

Group	Age	Sex	Overall Severity of Dysphonia (0–100)
Control	<i>M</i> = 54 years	45 male	<i>M</i> = 13
	Range 20 – 85	55 female	Range 1–39
	<i>SD</i> = 19.0		<i>SD</i> = 8.3
Voice Disorder	<i>M</i> = 54 years	45 male	<i>M</i> = 19
	Range 18 – 84	55 female	Range 0–100
	<i>SD</i> = 19.1		<i>SD</i> = 19.6

Table 2.

Diagnosis information for voice disorder group.

Diagnosis	<i>n</i>
Parkinson's Disease	44
Primary Muscle Tension Dysphonia	35
Vocal Fold Nodules	7
Vocal Fold Edema	3
Vocal Fold Scarring	3
Upper Respiratory Infection	2
Vocal Fold Polyp	1
Vocal Fold Granuloma	1
Vocal Fold Paralysis	1
Gastroesophageal Reflux Disease	1

Table 3.

RFF cycle rejection totals for aRFF-AP and PyRFF.

Rejection Reason	Offset Cycles		Onset Cycles	
	aRFF-AP	PyRFF	aRFF-AP	PyRFF
Too few periodic cycles	195	195	406	407
Fails to reach steady state or unstable	0	0	0	0
Glottalized	19	19	69	69
Aperiodic or irregular	0	0	0	0
Sharp transition	657	657	355	354

Table 4.

Mean errors in contours extracted automatically in the Matlab and Python programs, averaged across speakers.

Contour	Mean RMSE	Mean MBE
f_0 (Hz)	0.09 ($SD = 0.43$)	7.17E-4 ($SD = 0.01$)
Pitch strength (a.u.)	-1.79E-6 ($SD = 2.22E-6$)	9.34E-6 ($SD = 1.62E-5$)

Table 5.

Comparisons of RFF output from semi-automated MATLAB and Python programs with manual corrections made in MATLAB and imported into Python.

RFF Variable	<i>r</i>	RMSE	MBE
Overall Mean	1.00 ($p < .001$)	1.40E-9	9.24E-11
Mean Offset	1.00 ($p < .001$)	1.32E-9	-8.53E-12
Mean Onset	1.00 ($p < .001$)	1.36E-9	4.20E-11
Offset 10	1.00 ($p < .001$)	1.53E-9	1.23E-10
Onset 1	1.00 ($p < .001$)	1.39E-9	-1.28E-10

Table 6.

RFF cycle rejection totals by speaker group.

Rejection Reason	Control	Voice Disorders
Too few periodic cycles	263	339
Fails to reach steady state or unstable	0	0
Glottalized	46	42
Aperiodic or irregular	0	0
Sharp transition	513	498
Total rejections	822	879