Automated RFF Estimation (v.1.0)

For questions/comments, email Jenny Vojtech at jmvo@bu.edu

The MATLAB-based automated RFF algorithm was developed for high-quality acoustic recordings of the exact stimuli described in Lien (2015) and Lien et al. (2017). Below are the instructions that provide recommendations for acceptable changes to the algorithm; any additional changes beyond these recommendations (e.g., different stimuli combinations, lower quality sound recordings) may alter your output. Exercise caution when interpreting results after applying additional changes to this algorithm.'

MATLAB Toolbox Requirements:

- Econometrics Toolbox
- Curve Fitting Toolbox
- Signal Processing Toolbox
- Statistics Toolbox

Automated RFF estimation instructions:

1. Save each of the audio files to perform automated RFF estimation on within the same folder. Ensure that each of the audio file names are distinguishable, such as **subject_parameter_stimuli**.wav (e.g., "**C01_healthy_afa**.wav")

   Note. Do not use "+" or "–" in the filenames.

2. Open **main_RFF.m**

3. Change lines 10–12 to indicate the minimum possible pitch (`min_pitch_set`), maximum possible pitch (`max_pitch_set`), and the maximum number of VCV instances in a recording (`maxpkn`). For example, a stimuli of /ufu ufu ufu/ would have a `maxpkn = 3`.

```
% default settings
min_pitch_set = 50; %default min pitch is 50 Hz
max_pitch_set = 400; %default max pitch is 400 Hz
maxpkn = 3; % initially assume that there will be 3 VCV instances
```

4. **Input:** Change line 19 (variable "`folname`") to the path of your wav files; remember to include a backslash ("\") at the end of the path.

5. **Output:** Create a performance folder in the location where the RFF variables and figures should be saved. Change line 13 (variable "`perform_fol`") to indicate the location; remember to include a backslash ("\") at the end of the path.

   Note. All automated RFF analyses will be saved in a single Excel document "aRFF.xlsx" ("aRFF" stands for "automated RFF") with separate sheets for each audio file analyzed.

6. If you made separate recording files for /afa/ /ifi/ /ufu/ (e.g., "C01_healthy_afa.wav", "C01_healthy_ifi.wav", and "C01_healthy_ufu.wav") and want to combine them into one file, set `stimulicomb = 1`, otherwise set `stimulicomb = 0` in line 14.

   Note. When `stimulicomb = 0`*, the analysis of each audio file will be saved in a separate sheet within the "aRFF.xlsx" Excel file. By setting* `stimulicomb = 1`*, the analysis of each audio file will be saved in the same sheet of "aRFF.xlsx" in a side-by-side format.*

7. After altering these variables click **Run** in the MATLAB editor to run the automated RFF estimation algorithm.

8. A GUI will pop up for each recording, prompting to decide if the algorithm located the fricatives correctly.

a. Click "**OK!**" if location of the fricative marker is within the span of the fricative.

b. If the location of the marker is not in the fricative, or not every fricative has been identified, click "**NO**" to manually identify the fricatives.

i. Click on the waveform at each fricative location (one click per fricative; e.g., three fricatives should be three separate mouse clicks)

ii. After marking the fricatives, press **Enter** on the keyboard; the newly marked locations of each fricative should appear

iii. Press "**OK!**" to move to the next recording

9. After you have identified all the fricative locations across all of your recording files, the algorithm will work on locating the vocal cycles. More figures will quickly appear and disappear on your screen during this process.

10. Wait until the algorithm finishes running, then open the "aRFF.xlsx" Excel file to see the results of the automated RFF estimation algorithm.

11. Check the output waveforms by looking at the figures stored in the performance folder for each recording. If it looks like the algorithm is taking the wrong cycle, delete the corresponding RFF in the Excel file.

Additional Notes:

If you mess-up on identifying the fricatives but already pressed the "**OK!**" button to move onto the next audio file, do the following:

1. Press **Ctrl+C** while the MATLAB command window is active to terminate the program

2. In the MATLAB workspace, find "`filei`" (`filei` counts which file the program is on; e.g., if `filei = 37`, then you may have messed up identifying the fricatives on the 36th file).

a. Change Line 24 by inputting the instance to restart at (`filei = x:length`, where **x** = the filename you want to re-identify the fricative locations for)

b. Hit "Save" in the MATLAB editor

3. In the MATLAB workspace, double-click "`allfiles`" and find the row that corresponds with number of the file you messed up on in the structure; this will tell you the name of the folder that corresponds to the problematic audio file.

4. Navigate to the performance folder and delete that instance (in this example, if row 36 is "resonance_ufu" then delete the "resonance_ufu" sub-folder located within the performance folder)

5. Return to the **main_RFF.m** script in the MATLAB editor and click **Run**; the script should pick back up on the item that was messed up (in this example, "resonance_ufu")

References:

- Lien, Y.S. "Optimization and automation of relative fundamental frequency for objective assessment of vocal hyperfunction," *Diss.* Boston University, 2015.
- Lien Y.S., Heller Murray E.S., Calabrese C., Michener C.M., Van Stan J., Mehta D.D., Hillman R.E., Noordzij J.P., Stepp C.E. "Validation of an algorithm for semi-automated estimation of relative fundamental frequency," *Annals of Otology, Rhinology & Laryngology*, Vol. 126(10), pp. 712-716, 2017.