Using RSAseconds for any data that in the form of a single IBI series

# Before Starting:

1. Clean your data. RSAseconds does not automatically identify missed beats, outliers, or other sources of noise that are typically removed via visual inspection or use of a preprocessing program.
2. Open Matlab.
3. Unzip the “RSAActiheart.zip” folder. Copy the subfolder “program” in a directory of your choice.
4. Make sure that Matlab looks in this folder by setting the path:
   * You can set the path by typing: addpath(‘location’) in the Matlab command window. Here, location is the directory that the program is in. Remember to put the ‘’. Example: addpath('X:\Desktop\\_RSA\Distribution')
   * Or you can go to “File-> set path” in the upper left hand corner of the Matlab command window.
   * If you do not have administrative privileges, you cannot save the path settings and have to set the path each time you wish to use RSAseconds.

# Using RSAseconds3 Function (version 3)

Function: RSAseconds3(where, low, high)

The arguments within the function are:

**where:** the location of your IBI files. These files must be cleaned data in excel. Each file should contain only the IBI series (as a column vector) for one individual. So, each individual has their own file. Make sure nothing else is in this folder except these individual files.

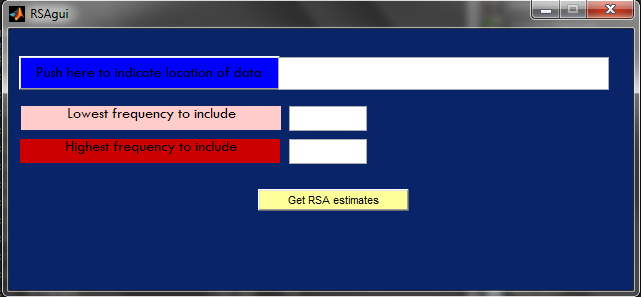
**low:** the lower frequency to be included in the RSA range. For adults this is typically .12.

**high:** the upper-bound frequency. For adults this is typically .40.

You can use this function on its own or the GUI (described below).

# Using RSAseconds GUI

1. After following the above steps, type RSAgui into the Matlab command window. You should see this:



1. Click “Push here to indicate location of data” to do just that. Indicate the directory **where** discussed above.
2. Enter the lower and upper bounds for the frequency range of interest. In human adults, it is usually around .12-.40 (Hz), for instance.
3. Click “Get RSA estimates”. This will generate new files in a subfolder within the **where** directory called “RSA” where the RSA estimates across time are provided (see below for details).
4. Once completed, “All done!” will be displayed in the Matlab command window.

# Output

## Excel containing the time-varying RSA estimates

In the **where** directory that you indicated in RSAgui you will find a folder “RSA” that contains output files that correspond to each of the participants. They will be excel files with the same name as the original file but with “\_RSA” added to the end.

Open one up. If you go to the “RSAseconds” worksheet, you’ll see two columns that are like this:

|  |  |
| --- | --- |
| 16 | 5.181235 |
| 17 | 5.398286 |
| 18 | 5.434318 |
| 19 | 5.169993 |
| 20 | 5.328098 |
| … | … |
| 189 | 5.143205 |

Where “…” indicates the series continues but is not picture here.

The first column is the second in real time that the RSA estimate corresponds onto. It starts at 16 seconds rather than 1 because the windows are 32 seconds long, hence there is no window for numbers 1 through 15 because there were not enough time points. The next estimate comes from seconds 2 thru 33, and 17 is approximately the midpoint. This continues until the end.

The second column is the RSA estimate corresponding to the segment number.

The “IBIseries” worksheet has the IBI series concatenated across segments.

## File containing problem cases

In the **where** directory you will also find a file called, “problems.mat”. This file has a structure, “problems”. If you open it, you will see that there is a variable called “problems.length”. This variable indicates the files that were too short for this analysis. These files are moved to the “problems” folder and analysis is continued on other files.

# Technical Details

The function RSAseconds (which is called upon from RSAgui) first concatenates the IBI series. MindWare usually exports them in epochs. Concatenation simply vertically appends each segment’s series. Prior to appending a given epoch, the IBI series for that epoch is searched to identify outliers that are at least 3 standard deviations from the mean. Should one or more outliers exist, values are imputed by taking the difference of the real length of the epoch in seconds (as indicated by MindWare) and the sum of the IBIs that are not outliers. This gives us the length of time that the outlier(s) should be. The average of this value is imputed. Next, the last and first values of subsequent series across epochs are added since together they give the time from the last R peak in one segment to the first one in the following segment.

RSAseconds then interpolates this IBI series so that estimates are obtained at 4 Hz (i.e., every 250 msecs) using a cubic spline (de Boor, C., *A Practical Guide to Splines*, Springer-Verlag, 1978). This makes the series equidistant, enabling time series analysis as opposed to point-process analysis. Prior to further analysis this series is mean-centered.

Prior to frequency analysis, Peak Matched Multiple Windows (PM MW) tapering windows are created using a matlab script and method introduced in Hansson & Jönsson (2006; “Estimation of HRV spectrogram using multiple window methods focusing on the high frequency power”. Medical Engineering & Physics, 28, 749-761).

Finally, short-time Fourier transform (STFT) is conducted on the time series with 32 second windows and 31 second overlap. Similar to traditional RSA analysis, the absolute values are then squared to obtain power estimates, of which the natural log is taken. The windows created in accordance with Hansson & Jönsson (2006) are used instead of the typical Hanning because they were found to reap better results when using 32-second segments.