RSAseconds User’s Manual

Contents

[Before Starting: 1](#_Toc404519056)

[Using RSAseconds Function (version 3) 1](#_Toc404519057)

[Using RSAseconds GUI 2](#_Toc404519058)

[Output 2](#_Toc404519059)

[Excel containing the RSA estimates 2](#_Toc404519060)

[File containing problem cases 3](#_Toc404519061)

[Technical Details 3](#_Toc404519062)

# Before Starting:

1. Clean your data. RSAseconds is not intended to be a method for processing IBI series. Most data-collection platforms have software for this, and to ensure fidelity with your other projects, it makes sense to use the same method for preprocessing. Thus this toolbox will not identify missed beats or other sources of noise that are typically removed via visual inspection or use of a preprocessing program. (However, RSAseconds does impute the mean for values that are >3 standard deviations from the mean, and makes notes of these individuals.)
2. Open Matlab.
3. Unzip the “RSAseconds.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\Skowrun\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 RSAseconds Function (version 8)

Function: RSAseconds8(where, low, high)

The arguments within the function are:

**where:** the location of your MindWare output files. These files must be cleaned data in excel with the worksheets “HRV Stats” and “IBI Series”. Each individual has their own file. Make sure nothing else is in this folder except these individual files.

*Example (for Mac users):* where = (‘/Users/~/Dropbox/IBI data/’)

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

*Example:* low = 0.12

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

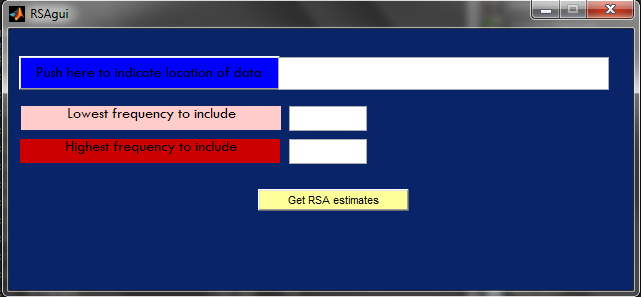
*Example:* high = 0.40

After setting these values, you would simply type *RSAseconds8(where, low, high)* into the console.

You can use this function on its own using these commands. If you would prefer, you can use the GUI instead (described below).

# Using RSAseconds GUI

1. After following the steps in the “Before starting” section, 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 contains lists of files that had issues (explained below). Files that have: (a) large amounts of missing data; or (b) missing worksheets; or (c) too short an IBI series are moved to the “Problems” folder with no analysis conducted.

The first, “problems.outliers”, contains the names of files that had outliers in the IBI series that were greater than 3 times the standard deviation. Given the high kurtosis often seen in the data, 3 times the standard deviation (Std) seems more prudent than the typical 2 times the Std, which results in most subjects having outliers. The researcher may want to examine these files to ensure that the data have been cleaned appropriately. Another cause may be the rare case that an R peak occurred right in between two segments. When concatenating segments, the program adds the last value of one segment to the first value of the next segment since usually the R peak following the last one in the one segment occurs in the next one. If this does not occur, and the R peak is exactly in between the segments, very large values will result when concatenating them. In any case, analysis was still conducted on these files. This list is simply here to alert the user of these potentially problematic files.

The second variable, “problems.skips”, indicates which participants had segments that were skipped. The program will compute RSA estimates, but the results may not be trustworthy because of the gap. One solution may be to create 2 files – one with the segments that are before the skipped one, and another with the segments after the skipped one. Then run “RSAgui” on these. Make sure the data are in a tab (i.e. worksheet) called “IBI series”.

Next we have, “problems.sheet”, which indicates that the individual is missing one of the two worksheets expected in MindWare output. Specifically, both the “HRV Stats” and “IBI Series” sheets are required for this toolbox. These files are moved into the “Problems” folder, and no analysis is completed. *Note: if your data were not processed with MindWare, then use the RSAseconds toolbox originally designed for Actiheart output. That toolbox can be used when the IBI series is in one vector.*

Finally, “problems.length”, contains a list of individual files for whom the length of the IBI series was too short for this analysis*.* These files are moved into the “Problems” folder and no analysis is completed.

# 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.