

## Benchmarking heart rate variability toolboxes

Adriana N. Vest, PhD,<sup>a,b,\*</sup> Qiao Li, PhD,<sup>a</sup> Chengyu Liu, PhD,<sup>a</sup> Shamim Nemati, PhD,<sup>a</sup>  
Amit Shah, MD, MSCR,<sup>b,c</sup> Gari D. Clifford, PhD<sup>a,d</sup>

<sup>a</sup> Department of Biomedical Informatics, Emory University School of Medicine, Woodruff Memorial Research Bldg, Suite 4100, 101 Woodruff Circle, Atlanta, GA, United States

<sup>b</sup> Department of Epidemiology, Rollins School of Public Health at Emory University, 1518 Clifton Road NE, Room 3053, Atlanta, GA, United States

<sup>c</sup> Division of Cardiology, Department of Medicine, Emory University School of Medicine, 101 Woodruff Circle Suite, 319 Woodruff Memorial Research Building, Atlanta, GA, United States

<sup>d</sup> Department of Biomedical Engineering, Georgia Institute of Technology, 1760 Haygood Drive, HSRB Suite W200, Atlanta, GA, United States

### Abstract

**Background:** Heart rate variability (HRV) metrics hold promise as potential indicators for autonomic function, prediction of adverse cardiovascular outcomes, psychophysiological status, and general wellness. Although the investigation of HRV has been prevalent for several decades, the methods used for preprocessing, windowing, and choosing appropriate parameters lack consensus among academic and clinical investigators.

**Methods:** A comprehensive and open-source modular program is presented for calculating HRV implemented in Matlab with evidence-based algorithms and output formats. We compare our software with another widely used HRV toolbox written in C and available through [PhysioNet.org](http://PhysioNet.org).

**Results:** Our findings show substantially similar results when using high quality electrocardiograms (ECG) free from arrhythmias.

**Conclusions:** Our software shows equivalent performance alongside an established predecessor and includes validated tools for performing preprocessing, signal quality, and arrhythmia detection to help provide standardization and repeatability in the field, leading to fewer errors in the presence of noise or arrhythmias.

© 2017 Elsevier Inc. All rights reserved.

### Keywords:

Heart rate variability; Toolbox benchmarking; Peak detection; Physiological signal processing

### Introduction

Interest in heart rate variability (HRV) and cardiovascular dynamics signal processing has seen a recent resurgence due to the increased availability of devices and wearables that record physiological signals. Methods that measure cardiovascular dynamics can be used to detect changes in the autonomic nervous system [1–3] and hold promise as tools that can aid in disease tracking, wellness promotion, and risk stratification. The non-invasive nature of HRV measurement makes it particularly attractive as a long-term health tracking tool, or component of a more comprehensive health monitoring framework.

Despite its popularity in research and relatively long history, there is still much disagreement in the methods by which researchers apply HRV signal processing. This disagreement limits meaningful comparisons between studies and scientific repeatability, especially when in-house, custom, non-public software are used. Unfortunately, few HRV programs are

rigorously designed and tested with methods that are clear and open access. Additionally, of the open-source HRV programs available, many are poorly documented, no longer supported by their original authors, or have broken dependencies that require extensive troubleshooting. Regardless, no existing HRV software toolbox, to our knowledge, provides a comprehensive suite of validated tools. More specifically, such software should undergo a validation process in which the output is rigorously compared with expected values based on a standardized input; furthermore, it should be compared to another set of well-developed HRV tools for consistency.

Perhaps the most used, and trusted HRV toolbox, is that written by Mietus and Moody, available from [PhysioNet.org](http://PhysioNet.org) [4]. PhysioNet's HRV Toolbox, an open-source package that is written in C and performs general HRV statistics and spectral analysis. This toolbox has the unique feature of compatibility with the QRS detectors, data libraries, and processing and evaluation tools of Physionet's Waveform Database (WFDB) Software Library. However, installation is nontrivial and the preprocessing and other variables associated with it require some expert use significant domain knowledge.

\* Corresponding author at: Department of Biomedical Informatics, Emory University School of Medicine, Woodruff Memorial Research Bldg, Suite 4100, 101 Woodruff Circle, Atlanta, GA 30322, United States.

E-mail addresses: [adriana.vest@emory.edu](mailto:adriana.vest@emory.edu) (A.N. Vest).

To address the issues of validation, standardization, repeatability, we have developed a comprehensive and open-source HRV analysis toolbox. The toolbox has been designed to accept a wide range of cardiovascular signals and analyze those signals with a variety of classic and modern signal processing methods. The toolbox includes many features not offered in other programs, including peak and pulse detection, signal quality analysis, rhythm detection, beat classification, general HRV statistics, phase rectified signal averaging (PRSA) techniques for deceleration and acceleration capacity, pulse arrival time (PAT), and cardiopulmonary coupling methods. The toolbox is written in the Matlab programming language and does not have any dependencies on external software or libraries. (A list of minimal in-built Matlab toolboxes are provided in the [Appendix A](#). The toolbox was designed to use the minimal number of dependencies and the most basic operators to future-proof the code base as much as possible.) The toolbox can process raw waveform data (such as electrocardiograms) as well as derived RR-interval data. Although it was designed not to deal with file formats, the toolbox does natively support MAT, CSV, or WFDB-compatible annotation formats (version 10.5.24 and earlier) without relying on PhysioNet's WFDB libraries (or other libraries). If users wish to export results from the HRV Toolbox, a function is included that allows for standard WFDB compatible output annotation files or CSV output files.

Preprocessing and data cleaning is an important aspect of signal processing that often is overlooked or poorly documented in the use of HRV analysis. The Matlab HRV Toolbox described here employs several methods to prepare data for HRV estimation, including assessing signal quality and detecting arrhythmias, erroneous data, and noise. These segments of data, which must be excluded from HRV analysis, can then be systematically removed based on threshold settings selected by the user or recommended in previously validated studies. In particular, our toolbox contains one initialization (or header) file which lists all the options available, with typical default settings. In this way, a user may easily identify which settings need to be given considerable thought (all the ones listed) and provide this listing in a publication.

This publication outlines the tools contained within the Matlab HRV Toolbox and presents results from a validation of the peak detectors and metric calculations with the C Toolbox by Mietus and Moody. For a detailed overview of the signal processing issues related to HRV, we refer the reader to Clifford et al. [5].

## Methods

### Toolbox design

The HRV Toolbox described here employs an initialization file that sets up global variables that deal with thresholds, window settings, noise limits, and spectral analysis limits. Once the researcher has selected parameters for the analysis, the physiological waveforms can be uploaded into Matlab for processing. The toolbox can accept

electrocardiogram (ECG), blood pressure (ABP), and photoplethysmogram (PPG) data and has validated beat detectors for each of these signals. The available beat detectors for ECG include Matlab versions of the PhysioNet tools *sqrs* [6,7], *wqrs* [8,9], and *jqrs* [10–12]. For pulse detection, the Matlab version of *wabp* [13] is customized for both blood pressure waveforms and PPG waveforms.

To quantify the signal quality of the various waveforms, a signal quality index (SQI) is calculated on a rolling window for the duration of the waveform. The toolbox uses *bsqi* [14] for ECG, *jsqi* [13,15] for BP, and *psqi* [16] for PPG. The user can set a threshold for acceptable SQI, which is then used during the preprocessing step to determine which segments of the waveform should not be analyzed. Ventricular tachycardia/fibrillation detection is performed using a state-of-the-art method published by Li et al. [17].

Waveforms are next converted to RR-intervals by taking the consecutive differences of the beat locations in contiguous data (where segments have not been removed). If the user desires to use RR interval data instead of the raw waveforms, the RR interval time series can be loaded into the HRV Toolbox directly, although signal quality and VF detection cannot then be performed. Atrial fibrillation (AF) is detected on the RR interval time series using the method published by Oster et al. [18]. Data is preprocessed by flagging and removing or interpolating through data with a change in RR interval greater than a threshold set by the user. The interpolation method is chosen by the user but options include cubic spline and linear interpolation functions. Gaps are also flagged and removed during this step. Time domain metrics are then calculated on the time series. Again, we note that removal without replacement is recommended [19].

For frequency domain calculations, the power spectral density (PSD) of the RR interval time series can be generated using several methods. Those methods include: the Lomb Periodogram, the Welch PSD estimate, the Burg PSD estimate, and the discrete or fast Fourier transform. An option to resample the RR interval time series is provided to users since the methods other than the Lomb Periodogram assume that the time series is uniformly sampled. Re-sampling the RR interval time series involves interpolating through the signal (such as by linear or cubic spline interpolation) and re-sampling at regular intervals specified by the resampling frequency. All PSD estimates calculated by the HRV Toolbox described here can accept frequency bin size as a parameter, which improves control over the reproducibility of the resulting analysis.

After the PSD is calculated, various frequency domain HRV metrics are calculated. The sum of power in the various frequency bands is calculated as is the total power in the spectrum. These spectral metrics can be normalized to the variance of the RR interval time series, or to another measure. As stated above, many researchers normalize the sum of the power spectral density plot to variance because of the mathematical equivalency of the two. The choice of normalization is up to the user, but explicitly specified in the set-up of the analysis.

### Toolbox validation

To evaluate the performance of the toolbox against accepted methods available in the field, annotations and

metrics were compared between the HRV Toolbox described here and previously published or public tools from PhysioNet.

#### Peak detectors

Three peak detectors are provided within the Matlab HRV Toolbox described here. In order to compare their performance to previously published peak detectors, ECG waveforms from the MIT-BIH Arrhythmia Database were processed with the provided Matlab peak detectors *jQRS.m* [10–12], *wQRS.m* [8,9], and *sQRS.m* [6,7], and with *wQRS.c* [8,9], *sQRS.c* [7,8], and *gQRS.c* [8] available from the WFDB software package. The open-source *epilimited* ‘Pan and Tompkins’ method [20] was not included for this analysis because its performance on the MIT-BIH arrhythmia database, on which it was optimized without cross validation, has been widely reported. Additionally, *gQRS* and *jQRS* are similar algorithms (based on energy), adapted to long term and noisy recordings. Annotation files were generated from the peak detectors and were compared against reference annotations provided in the database using PhysioNet’s *bxb.c* function with a match threshold of 100 ms. The first lead of all the database records was analyzed (~48 records, ~30 min each). The *bxb.c* function compares each beat with the reference annotation. If the two annotations match within 100 ms, the beat is considered matching to the reference. The F1 score is reported as a measure of performance.

#### HRV metric comparison

To isolate the comparison of HRV metrics from other signal processing variables, annotation files were downloaded from the MIT Normal Sinus Rhythm Database and minimally preprocessed with PhysioNet’s HRV Toolbox. Before preprocessing, annotation files were segmented into 5 min windows with 4 min of overlap between windows. Windows with possible AF or with greater than 5% of the data missing were not analyzed. The data was then fed into both PhysioNet’s HRV Toolbox and the HRV Toolbox described here (after removing the mean in the case of spectral metrics). Mean NN interval, PNN50, RMSSD, SDNN, HF, LF, LF/HF ratio, and total power were calculated on each window over the entirety of the 24 h recording for each patient ( $n = 18$ ). The spectral metrics were calculated using the Lomb Periodogram and normalized per the method in the C implementation of the function in Numerical Recipes in C [21]. Results were compared on each patient using *mxm*, a PhysioNet function that finds the root mean squared error according to the equation

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (X_{\text{Test}} - X_{\text{Standard}})^2}{n}}$$

and normalizes to the mean value of the accepted standard, which in this case was taken to be the value given by the PhysioNet HRV Toolbox implementation.

Previous experience using various HRV toolboxes indicates that changing the frequency binning when performing spectral estimation can change the results

generated. The frequency vectors were standardized between the PhysioNet HRV Toolbox and the Matlab HRV Toolbox described here.

## Results and discussion

#### Peak detectors

When tested on the MIT Arrhythmia Database data the Matlab peak detectors performed similarly to the C versions, as shown in Table 1. The F1 scores for each of the detectors were above 90. A nonsignificant difference for the Matlab and C versions of *wQRS* are observed (due to window edge effects). A larger difference is observed in the two versions of *sQRS* (which was not written specifically for this toolbox), but since the *QRS* detector is not recommended for use in either the C or Matlab version, the differences are unimportant. We note, however, that additional analyses on more databases should (and will) be performed to fully analyze their performance, particularly on noisy data. In previous publications we have shown the superiority of *jQRS* for long term noisy recordings, with a winning entry in the PhysioNet Challenge 2014 [11]. The reader should not assume a high F1 indicates a better performing algorithm in reality, but rather that these perform well for noise free databases, and in particular on the MIT-BIH database, on which they were trained, with no independent out of sample testing. As such they are highly over-trained to these databases and likely to be underperforming on other databases. The fact that the higher scoring algorithms trigger on slopes, rather than energy explains their noise sensitivity. It is this fact that is exploited to enable signal quality metrics [14].

#### HRV metrics

The Matlab HRV Toolbox described here generated results which were within 0.16% normalized RMSE of PhysioNet’s HRV Toolbox (Table 2) on all metrics tested on the MIT Normal Sinus Rhythm Database. The metrics with the highest error were PNN50 and RMSSD. When a closer inspection was performed on the way these metrics were calculated, it was discovered that the PhysioNet’s HRV Toolbox removed additional data points on the edge of the windows compared to the method by the Matlab HRV Toolbox. This accounted for the minor differences. The

Table 1  
Performance of peak detectors.

Peak detector	Recommended application	F1	St dev
<i>wQRS.c</i>	Low noise scenarios or as a comparator to detect noise	99.00	1.89
<i>wQRS.m</i>	Low noise scenarios or as a comparator to detect noise	99.04	1.84
<i>sQRS.c</i>	Low noise scenarios or as a comparator to detect noise	98.19	4.22
<i>sQRS.m</i>	Low noise scenarios or as a comparator to detect noise	96.33	6.38
<i>jQRS.m</i>	Long term moderate noise recordings, such as in ICU or Holter	93.02	12.27
<i>gQRS.c</i>	Moderate noise ICU or Holter recordings	95.72	14.84



Table 2  
Comparison of Matlab and C HRV Toolboxes.

Metric	Normalized RMSE Matlab vs. C
Mean	0.0007
pNN50	0.1596
RMSSD	0.1029
SDNN	0.0010
VLF	0.0018
LF	0.0022
HF	0.0036
LFHF	0.0025
TTLTPWR	0.0014

remainder of the error is likely due to round off of constants that can be performed differently in Matlab and in C. However, these errors should not significantly affect any analysis.

Although these results show the Matlab HRV Toolbox can perform similarly to the C version, future analyses will evaluate the effect of the differing methods of preprocessing, frequency binning, and normalization. Care was taken to ensure consistency in this analysis; preprocessing, frequency binning, and normalization were all standardized between the two HRV Toolboxes. However, these processing steps tend to vary drastically between researchers and an analysis on how they impact results and conclusions would be instructive.

## Conclusions

This article presents the outline of an open-source standardized HRV toolbox and some of the issues surrounding its use. Comparison to standard open-source software demonstrate that it can be used as benchmarking system for HRV studies, FDA filings, and industrial applications (due to its BSD licensing). In particular we highlight the fact that small differences in preprocessing and QRS detection have large effects on reported indices. Future articles will expand on the documentation and add further tools to the code base.

## Acknowledgements

The authors wish to acknowledge the National Institutes of Health (Grant # NIH K23 HL127251) and Emory University for their financial support of this research. GC is partially supported by National Science Foundation Award 1636933 and the National Institutes of Health, the Fogarty International Center and the Eunice Kennedy Shriver National Institute of Child Health and Human Development, grant number 1R21HD084114-01. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the National Institutes of Health. The authors also wish to thank George Moody and Joe Mietus for posting the C HRV toolbox which serves as the baseline comparison point for this article.

## Appendix A

The current version (1.0) of the HRV toolbox requires the following Matlab configuration: Matlab (v 9.1), Signal Processing Toolbox (v 7.3), and Statistics and Machine Learning Toolbox (v 11.0).

## References

- [1] Malik M. Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology. Heart rate variability. Standards of measurement, physiological interpretation, and clinical use. *Circulation* 1996;93(5):1043–65.
- [2] Clifford G. Signal processing methods for heart rate variability. University of Oxford; 2002.
- [3] Pan Q, Zhou G, Wang R. Do the deceleration/acceleration capacities of heart rate reflect cardiac sympathetic or vagal activity? A model study. *Med Biol Eng Comput* 2016;54(12):1921–33.
- [4] Mietus JE, Goldberger AL. Heart rate variability analysis with the HRV Toolkit - PhysioNet. [January, 23 2014 August 2016]; Available from: <https://physionet.org/tutorials/hrv-toolkit/> Accessed on August 28, 2016.
- [5] Clifford GD, Statistics ECG, Noise Artifacts, Data Missing. In: Clifford GD, Azuaje F, McSharry P, editors. Advanced methods and tools for ECG data analysis. Artech House, Inc.; 2006.
- [6] Moody GB. WFDB programmer's guide. [2015 March 28, March 2017]; Available from: <https://physionet.org/physiotools/wpg/> Accessed on March 1, 2017.
- [7] Engelse W, Zeelenberg C. A single scan algorithm for QRS-detection and feature extraction. *Comput Cardiol* 1979;6(1979):37–42.
- [8] Moody GB. WFDB Applications Guide. [2015 May 28, August 2016]; Available from: <https://www.physionet.org/physiotools/wag/> Accessed on August 28, 2016.
- [9] Zong W, Moody G, Jiang D. A robust open-source algorithm to detect onset and duration of QRS complexes. *Comput Cardiol* 2003;30:737–40.
- [10] Behar J, et al. A comparison of single channel fetal ECG extraction methods. *Ann Biomed Eng* 2014;42(6):1340–53.
- [11] Johnson AE, et al. R-peak estimation using multimodal lead switching. *Comput Cardiol* 2014;41:281–4.
- [12] Johnson AE, et al. Multimodal heart beat detection using signal quality indices. *Physiol Meas* 2015;36(8):1665.
- [13] Sun JX. Cardiac output estimation using arterial blood pressure waveforms. *Mass Inst Technol* 2006.
- [14] Li Q, Mark RG, Clifford GD. Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a Kalman filter. *Physiol Meas* 2008;29(1):15–32.
- [15] Sun J, et al. Estimating cardiac output from arterial blood pressure waveforms: a critical evaluation using the MIMIC II database. *Comput Cardiol* 2005;32:295–8.
- [16] Li Q, Clifford GD. Signal quality and data fusion for false alarm reduction in the intensive care unit. *J Electrocardiol* 2012;45(6):596–603.
- [17] Li Q, Rajagopalan C, Clifford GD. Ventricular fibrillation and tachycardia classification using a machine learning approach. *IEEE Trans Biomed Eng* 2014;61(6):1607–13.
- [18] Oster J, Clifford GD. Impact of the presence of noise on RR interval-based atrial fibrillation detection. *J Electrocardiol* 2015;48(6):947–51.
- [19] Clifford GD, Tarassenko L. Quantifying errors in spectral estimates of HRV due to beat replacement and resampling. *IEEE Trans Biomed Eng* 2005;52(4):630–8.
- [20] Pan J, Tompkins WJ. A real-time QRS detection algorithm. *IEEE Trans Biomed Eng* 1985;3:230–6.
- [21] Press WH, et al. Ch 13: Fourier and spectral applications, in numerical recipes in C: the art of scientific computing. 2nd ed. Cambridge University Press; 1992.