ECGlib: Library for Processing Electrocardiograms

Lars Johannesen[1,2], Jose Vicente[3], Loriano Galeotti[3], David G Strauss[3]

[1] Division of Pharmacometrics, Office of Clinical Pharmacology, Office of Translational Sciences, Center for Drug Evaluation and Research, US Food and Drug Administration, USA
[2] Department of Clinical Physiology, Karolinska Institutet and Karolinska University Hospital, Stockholm, Sweden
[3] Division of Physics, Office of Science and Engineering Laboratories, Center for Devices and Radiological Health, US Food and Drug Administration, USA

Abstract

To facilitate evaluation of ECGs in the FDA ECG Warehouse, public databases and clinical trials we have developed a C++ library for processing ECGs (ECGlib). ECGlib has a modular design, and is capable of handling files stored in many different formats, e.g. ISHNE, Physionet and FDA HL7. Moreover, ECGlib provides functions to do standard ECG signal processing, such as noise removal, QRS detection, classification, median beat creation and ECG waveform delineation. The performance of the different components of ECGlib has been evaluated using publicly available databases from Physionet (MIT-BIH and QTDB). The performance of ECGlib processing methodologies is comparable to state-of-the-art methods. We have also developed a MATLAB/Octave interface for ECGlib and are working on an interface for Python, R and Julia. Lastly, ECGlib comes with a set of command line tools that utilize parallel processing to quickly enable researchers to process large databases. We believe frameworks such as the one described can be used to facilitate research of ECG signals and we are working on making the library publicly available under an open source license.

1. Introduction

Since 2005, all new non-cardiovascular drugs have had to undergo a Thorough-QT study, a dedicated clinical study to assess the potential to induce fatal arrhythmias. The US Food and Drug Administration (FDA) has required the digital ECGs recorded during the Thorough-QT study to be uploaded to the FDA ECG Warehouse, which currently houses more than five-million 10 second ECGs from more than 200 Thorough-QT studies [1].

Recently, there has been several research efforts within the FDA to analyze the ECGs collected in these studies for various research purposes. In order to process these ECGs, it was necessary to develop a framework, that allows for detection of QRS complexes, and thereby RR-series and creation and delineation of median QRS-T waveforms to allow for evaluation of new ECG biomarkers. Several other ECG processing libraries and frameworks have been developed previously [2–7], but these are either not widely available or implemented primarily in MATLAB [4–7], thus limiting the use, constrained to working with files in a particular format [2] or limited in functionality, e.g. missing waveform delineation [3].

Therefore, we sought to develop a framework that would allow working with multiple formats, implemented in C++ to allow for easy integration into either MATLAB/Octave, Julia, Python, R or any other higher level interpreter or directly into a user-interface.

2. Methods

2.1. Framework

We developed a framework for processing ECG signals stored in different formats, such as ISHNE, Physionet and HL7 annotated xml. An overview of the framework is provided in Figure 1. The core of the framework is the ECGdata class, which consists of information about the ECG recording such as subject ID, recording start, sampling rate etc., the annotations and ECG data. Various functions for opening different formats work by loading data into this common structure, which is the input to different functions of the library. Similarly, functions for exporting data use the common data structure and convert it to different formats.

After loading an ECG into the ECGdata structure, QRS complexes can be detected using the so-called U3-method proposed by Marchesi and Paoletti [8, 9], which we have previously implemented in Python/C++ [10]. Briefly, the ECG is transformed using a non-linear transform that en-

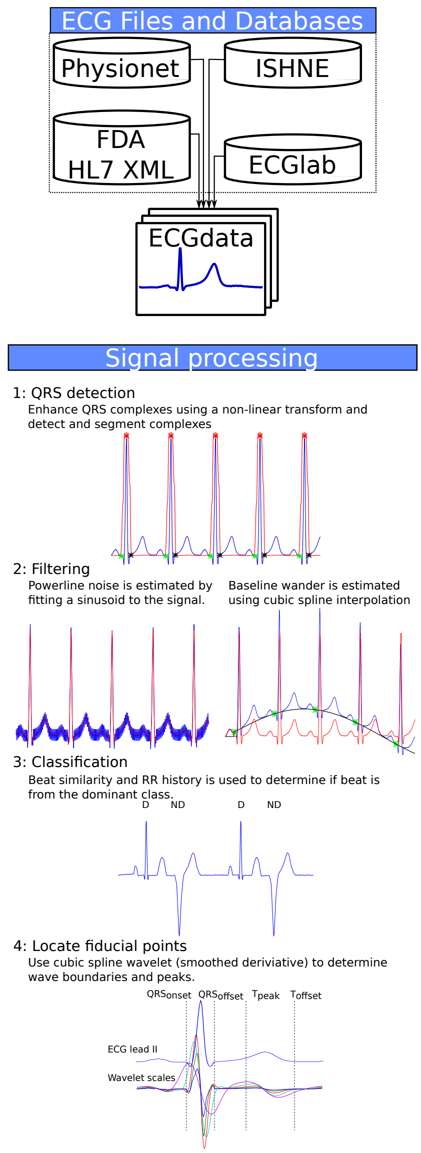

Figure 1. Overview of the different components in the ECGlib framework.

hances the QRS complex, see Figure 1, and QRS complexes are then detected and delineated. Afterwards, noise reduction methodologies that rely on QRS complex detection can be employed. Currently, there are three filtering methods implemented in ECGlib: baseline wander removal using cubic spline interpolation and powerline removal, see Figure 1, or a standard IIR filter.

The next step of the algorithm is to create median beats and detect waveform peaks and boundaries. The creation of the median beat increases the signal-to-noise (SNR) ratio, under the assumption that the noise is random and uncorrelated with the signal. If non-dominant beats, likely beats of non-sinus origin, or noise artefacts falsely triggering a QRS detection are included it would lead to a reduction of the SNR. Thus, before creating the median beat we use a crude beat classifier that classifies beats into dominant and non-dominant beats, based on waveform similarity and the regularity of the RR interval for each beat.

The last step of the processing flow is to delineate the ECG waveforms. This is performed in two steps: determination of the wave peaks and boundaries using a wavelet-based delineator [11–13] and secondly using the least-squares approach [14] to locate the offset of the T-wave more precisely. The wavelet delineation uses the cubic spline wavelet proposed by Li et al. [15], which can be interpreted as bandpass filtered first-deriviate, see Figure 1. Each of the different scales has different frequency content. The lower scales (higher frequency) are used for detecting QRS peaks and the higher scales (lower frequency) for P/T peaks. A minimum and a maximum slope then forms a peak in the ECG signal and the significance of the peaks is based on the slopes in the wavelet scales and the signal amplitude. For more details on the wavelet delineation see references [11–13].

2.2. Implementation

The framework is implemented in C++ using publicly available libraries, such as BOOST [16] and Armadillo [17].

Currently, the framework is available in MATLAB and Octave using the MEX interface and we are working on making the framework available in Python, R and Julia.

Lastly, we have also developed a set of command line tools that supports parallel execution using the message passing interface (MPI) which could be used to quickly batch process larger studies. Running the command line tools with one thread it is possible to process about 2-3 ECGs/s or a around over 9000 ECGs per hour which is comparable to other frameworks [18]. The output of the batch processing can then be reviewed using ECGlab [19].

2.3. Data

To evaluate the performance of the QRS detector, classifier and the delineator we used Physionet, which is an open database commonly used to evaluate performance of algorithms.

The QRS detector performance was evaluated using MIT-BIH [20], which consists of 48 half-hour two-lead recordings from patients with different types of arrhythmias and contains the locations of the QRS complexes, reviewed by cardiologists. The recordings in MIT-BIH are sampled at 360 Hz with a resolution of 11-bit over a 10 mV range.

Performance of the classifier was evaluated using the same database as used for the QRS detector. As the goal of the classification is to determine which beats are from the dominant class in the record, we re-classified all the beats, originally classified manually, for each record into dominant or non-dominant based on each individual record.

Finally, the performance of the delineator was evaluated using the QTDB database [21], which consists of 105 recordings, resampled at 250 Hz from various Physionet databases such as MIT-BIH, MIT-BIH ST and European S-T database. Briefly, the QTDB contains records annotated manually for P_{onset},P_{peak}, P_{offset}, QRS_{onset}, R_{peak}, QRS_{offset}, T_{onset}, T_{peak} and T_{offset}. The records contain one annotation for both leads.

2.4. Statistical analysis

For the QRS detection, the results are presented as the error rate and for the beat classification the classification accuracy, sensitivity and specificity will be computed.

For both the detection and classification the results are summarized for all records. Delineation results are computed as proposed by Martinez et al. [11]. Briefly, as there is only one annotation for both leads, annotation (from either lead) closest to the manual annotation was used for computing the difference. If there was no point within 200 ms, it was assumed to be missed detection. As missing annotations does not mean that the wave was not present, we cannot compute the false positives, but only false negatives [22]. Therefore, we are only reporting sensitivity. Lastly, the results are presented as the average of the inter-record mean and standard deviation.

All statistical analysis was done using R 2.15.0 (The R Project for Statistical Computing, Austria, Vienna).

3. Results

3.1. QRS detector

A false error rate of 0.58% was obtained, which is comparable to other state-of-the-art QRS detectors which have an error rate on the same data in the range of 0.5 to 0.9% [8, 11, 23].

3.2. Classification

A classification accuracy of 89.5% was obtained (91.5% sensitivity and 74.1% specificity) for classifying beats into dominant or non-dominant.

3.3. Delineation

The performance of the delineation is listed in table 3.3, and is comparable to the MATLAB version of our algorithm [13] and other ECG delineation algorithms [11, 12], for the P-wave and QRS-complex.

Table 1. Delineation results

Fiducial point	Mean±SD	Sens
P_{onset}	-2.16±14.0	96.2%
P_{peak}	-1.1±10.4	95.8%
P_{offset}	8.4±15.1	97.3%
QRS_{onset}	-3.6±12.9	99.5%
QRS_{offset}	1.56±12.0	99.9%
T_{peak}	-1.8±13.9	85.7%
T_{offset}	-13.5±19.1	85.6%

4. Discussion

Analysis of ECGs requires several preprocessing steps to clean up the ECGs, detect QRS complexes and delineate ECG waveforms. Several algorithms for each step of the ECG processing flow have been proposed, but to date none of them are freely available in the form of a library, that can be used across platforms and programming libraries. Combining all the processing methodologies into a common library that can work with any ECG file format has the advantage to make ECG research easier. It also has the advantage of providing a common framework for ECG signal processing, which can help facilitate sharing of ECG analysis algorithms.

Therefore, we sought to implement standard ECG processing methodologies in C++ in a modular fashion that allows researchers to process ECGs in various formats, or extend the library to support other formats easily. Moreover, because C++ was selected as the language, the implemented methodologies can be made available to interpretation languages commonly used in signal processing such as MATLAB/Octave or Python. In fact, we have already implemented a full MEX interface and are working on making the library available in Python, Julia and R.

The currently implemented methodologies have been tested using publicly available databases from Physionet such as MIT-BIH (detection, classification) and QTDB (delineation) and the performance are comparable to other state-of-the-art algorithms. However, there is room for improvement for the T-wave delineation performance. The beat classification, is however, rather crude as the goal of the algorithm is not to classify beats into types of beat but just by similarity, thus providing reliable median beats.

The developed framework is currently being used for analysis of data from Thorough-QT studies and other clinical studies [24] as well as FDA sponsored clinical trials, such as the one described in [25]. We are currently developing a graphical user interface for evaluation and adjudication of measurements [19]. The library, in its current stage, is focused on analyzing short signals, but we are planning on extending the library to be able to process long-term recordings such as 24-hour Holter recordings.

We believe that frameworks such as the one described in this paper can be used to facilitate ECG research and we are working to make our framework open-source.

Acknowledgements

This project was supported in part by FDAs Critical Path Initiative, Office of Womens Health, and appointments to the Research Participation Programs at the Oak Ridge Institute for Science and Education through an interagency agreement between the Department of Energy and FDA.

Disclaimer

The opinions presented here are those of the authors, and not necessarily those of the US Food and Drug Administration. No official support or endorsement by FDA is intended nor should be inferred. The mention of commercial products, their sources, or their use in connection with material reported herein is not to be construed as either an actual or implied endorsement of such products by FDA.

References

[1] Stockbridge, N et al. Dealing with Global Safety Issues - Was the Response to QT-Liability of Non-Cardiac Drugs Well Coordinated? Drug Saf 2013;36:167–82.

[2] Goldberger, AL et al. PhysioBank, PhysioToolkit, and PhysioNet Components of a New Research Resource for Complex Physiologic Signals. Circ 2000;101:e212–220.

[3] Hamilton, PS. Open Source ECG Analysis Software Documentation, 2002.

[4] Man, SC et al. Beats: An Interactive Research Oriented ECG Analysis System. Comput Cardiol 2010;37:1007–10.

[5] Draisma, HHM et al. LEADS: An Interactive Research Oriented ECG/VCG Analysis System. Comput Cardiol 2005; 32:515–18.

[6] Bolea, J et al. Biosigbrowser, biosignal processing interface. In Information Technology and Applications in Biomedicine, 2009. ITAB 2009. 9th International Conference on. IEEE, 2009; 1–4.

[7] Vidaurre, C and Sander, TH and Schloegl, A. BioSig: The Free and Open Source Software Library for Biomedical Signal Processing. Comput Intelligence Neuro 2011; 2011.

[8] Paoletti, M and Marchesi, C. Discovering dangerous patterns in long-term recordings using a fast QRS detection algorithm and explorative data analysis. Comput Methods Programs Biomed 2006;82(1):20–30.

[9] Marchesi, C and Paoletti, M. ECG Processing Algorithms for Portable Monitoring Units. Internet J Med Tech 2004; 1(2).

[10] Johannesen, L and Galeotti, L. Automatic ECG quality scoring methodology: mimicking human annotators. Physiol Meas 2012;33(9):1479–89.

[11] Martinez, JP et al. A wavelet-based ECG delineator: evaluation on standard databases. IEEE Trans Biomed Eng 2004; 51(4):570–81.

[12] Boichat Nea. Wavelet-based ECG delineation on a wearable embedded sensor platform. Wearable and Implantanble Body Sensor Networks 2009;256–61.

[13] Johannesen, L et al. A Wavelet-Based Algorithm for Delineation and Classification of Wave Patterns in Continuous Holter ECG Recordings. Comput Cardiol 2010;37:979–82.

[14] Xue, Q and Reddy, S. New algorithms for QT dispersion analysis. Comput Cardiol 1996;293–96.

[15] Li, C and Zheng, C and Tai, C. Detection of ECG characteristic points using wavelet transforms. IEEE Trans Biomed Eng 1995;42(1):21–8.

[16] BOOST. C++ Libraries. URL `http://www.boost.org`.

[17] Anderson, C. Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments. URL `http://arma.sf.net`.

[18] Sarapa, N et al. Comparison of QTinno, a fully automated electrocardiographic analysis program, to semiautomated electrocardiographic analysis methods in a drug safety study in healthy subjects. J Electrocardiol 2009; 42:358–66.

[19] Vicente, J et al. ECGlab: User Friendly ECG/VCG Analysis Tool for Research Environments. Comput Cardiol 2013; 40:In press.

[20] Moody, GB and Mark, RG. The impact of the MIT-BIH Arrhythmia Database. IEEE Eng in Med and Biol 2001; 20(3):45–50.

[21] Laguna, P et al. A Database for Evaluation of Algorithms for Measurement of QT and Other Waveform Intervals in the ECG. Comput Cardiol 1997;24:673–6.

[22] Vila, JA et al. A new approach for TU complex characterization. IEEE Trans Biomed Eng 2000;47(6):764–72.

[23] Afonso, VX et al. ECG beat detection using filter banks. IEEE Trans Biomed Eng 1999;46(2):192–202.

[24] Benton, RE et al. Greater quinidine-induced QTc interval prolongation in women. Clin Pharmacol and Thera 2000; 67(4):413–8.

[25] FDA Seeks Alternatives To QT Interval For Assessing Risk Of Torsade de Pointes, 2013.

Address for correspondence:

Lars Johannesen
10903 New Hampshire Avenue
Building 62 RM2127E
Silver Spring, Maryland, 20910
USA
lars.johannesen@fda.hhs.gov