Open Source ECG Analysis

P Hamilton

E.P. Limited, Somerville, USA

Abstract

Each year companies and researchers expend significant resources developing basic beat detection and classification software. In an effort to reduce this duplication of effort we are developing and making available open source ECG analysis software.

Our open source QRS detectors have sensitivities and positive predictivities that are close to 99.8% on the MIT/BIH and AHA arrhythmia databases. Our beat classifier has a sensitivity of 93.91% and a positive predictivity of 96.48% on the MIT/BIH arrhythmia database and a sensitivity of 93.2% and a positive predictivity of 97.83% on the AHA arrhythmia database.

Since we have posted our source code, over 350 users have downloaded our ECG analysis software. Downloads have been nearly equally divided between students, researchers, and commercial developers.

1. Introduction

Computer analysis of ECG signals is common today. Devices on the market that analyze ECGs, such as patient monitors, stress test systems, and Holter analysis systems, do a good job of detecting beats and classifying arrhythmias. However, new companies are constantly emerging and applying new technologies in an effort to make smaller and cheaper ECG analysis systems. Other companies may require ECG analysis to improve the performance of other medical diagnostic systems. Each new company must implement their own ECG analysis software, duplicating the efforts of every other company. Similarly, researchers who use ECG analysis for work in areas such as Heart Rate Variability (HRV) need to develop beat detection and classification software. Thirty years of research on computer analysis of ECG signals has produced a great many methods for detecting and classifying beats, but there is still a significant effort required to translate theory into implementation.

In an effort to reduce this industry and research wide duplication of effort, E.P. Limited has been developing and has released open source software for ECG analysis. We have developed C functions that implement the most basic ECG analysis operations, detection and classification of individual beats. By building on this open source software we new companies will be able to bring reliable systems to market more quickly and researchers will be able to spend more time exploring new diagnostic techniques rather than implementing beat detectors.

In this paper, we present an overview of the algorithms used in our software and the performance of these functions on standard ECG databases. We refer the reader to the open source code (available at www.eplimited.com) for the details of our algorithms.

2. QRS detector

Our QRS detectors are based on the QRS detector originally developed by Pan and Tompkins [1] in assembly language for implementation on a Z80 microprocessor and later improved and ported to C by Hamilton and Tompkins [2]. This QRS detector uses a single ECG channel and was originally designed to operate at 200 Hz. The advantages of this QRS detection algorithm are that it is efficient and easily modified for different sample rates.

In our present software release, we have included the original QRS detector and two variations on that detector. One variation represents a more efficient general implementation that conforms to ANSI/AAMI EC13 [3]. We developed the second variation for implementation on a PIC16F877 processor.

2.1. QRS detection preprocessing

Figure 1 shows the operations that make up our QRS detector algorithm. The QRS detector filters the ECG signal to create a local estimate of the power in the in the QRS bandwidth. The filters in our QRS detector are described in [2]. Because all the filters are based on moving window averages, adapting them to different sample rates only requires changing the number of samples in the moving averages.

Our preprocessor implementation differs from those

described in [1] and [2] in its use of a rectified rather than a squared signal and an 80 ms averaging window rather than a 150 ms averaging window. We have found that using rectification rather than squaring reduces the gain sensitivity of the algorithm, and recent work [4] indicates a narrower window improves the performance of the beat detector.

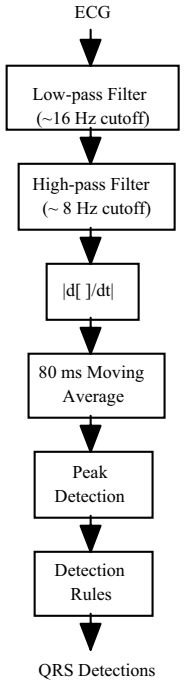

Figure 1. QRS detection operations.

2.2. QRS detection rules

The following are the basic detection rules for the beat detector:

1. Ignore all peaks that precede or follow larger peaks by less than 200 ms.
2. If a peak occurs, check to see whether the ECG signal contained both positive and negative slopes. If not, the peak represents a baseline shift.
3. If the peak occurred within 360 ms of a previous detection and had a maximum slope less than half the maximum slope of the previous detection assume it is a T-wave.
4. If the peak is larger than the detection threshold call it a QRS complex, otherwise call it noise.
5. If an interval equal to 1.5 times the average R-to-R interval has elapsed since the most recent detection, within that interval there was a peak that was larger than half the detection threshold, and the peak followed the preceding detection by at least 360 ms, classify that peak as a QRS complex.

The detection threshold used in 4 and 5 above is calculated using estimates of the QRS peak and noise peak heights. Every time a peak is classified as a QRS complex, it is added to a buffer containing the eight most recent QRS peaks. Every time a peak occurs that is not classified as a QRS complex, it is added to a buffer containing the eight most recent non-QRS peaks (noise peaks). The detection threshold is set between the mean or median of the noise peak and QRS peak buffers according to the formula:

$$\text{Detection\_Threshold} = \text{Average\_Noise\_Peak} + TH*(\text{Average\_QRS\_Peak} - \text{Average\_Noise\_Peak})$$

where *TH* is the threshold coefficient (generally between 0.3125 and 0.475). Similarly, the R-to-R interval estimate used in 5 is calculated as the median or mean of the last eight R-to-R intervals.

2.3. QRS detector variations

After the release of our initial QRS detector, we developed a QRS detector that was suitable for implementation in a PIC16F877 microprocessor. Because of the limited RAM in the PIC16F877, we eliminated rules 3 and 2 (both rules required a buffer of past slopes). We also simplified the implementation by using means rather than medians to calculate the average QRS peak, average noise peak, and average R-to-R intervals used for setting thresholds.

Surprisingly, these simplifications actually improved the performance of the QRS detector, so we incorporated them into a more portable version of the PIC16F877 QRS detector. We then modified this platform independent, simplified QRS detector so that it would pass the ANSI/AAMI EC13 standard. We included a hard minimum on the detection threshold to avoid detection of QRS complexes with amplitudes less than 150 uV. Also for EC13, we reduced the blanking interval around each peak from 200 to 195 ms to avoid missed detections at heart rates near 300 beats-per-minute.

Table 1. Performance statistics for QRS detectors.

	MIT/BIH		AHA	
Detector	Sens.	+Pred.	Sens.	+Pred
Original	99.74%	99.81%	99.47	99.73
PIC16F877	99.80%	99.80%	99.44	99.59
Simplified/EC13	99.77%	99.79%	99.51	99.68

We tested all three variations on the MIT/BIH arrhythmia database[5] and the AHA arrhythmia database

[6]. Table 1 lists the sensitivities and positive predictivities for the three different QRS detectors. All three beat detectors perform comparably. The PIC16F877 implementation performs slightly worse on the AHA database than the other two, because the fiducial marks on record 7209 are annotated on the early side of the QRS complex and our algorithm produced detections on the late side of the QRS complex, resulting in both false positive and false negative detections for 107 beats.

3. Beat classification

When a beat is detected, our algorithm classifies it as either normal or ventricular. The R-to-R interval and a one-second buffer containing the sampled beat are input to the classification algorithm. Figure 2 illustrates the inputs, operations, and features used in our beat classification algorithm.

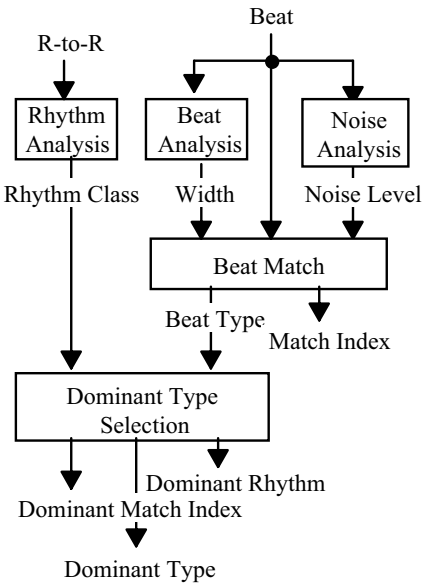

Figure 2. Inputs, operations, and features used for beat classification

3.1. Width estimation

We initially estimate onset, end, and width of a beat by searching to either side of the detection point for 25 ms isoelectric regions having signal ranges of less than 100 μV. We then search to either side of the detection point for slopes that are significantly smaller than the slopes near the detection point. If the two estimates are close, we use the isoelectric based width estimate, otherwise, we use the slope based width estimate.

3.2. R-to-R interval classification

As part of our algorithm, we characterize R-to-R intervals as Normal-Normal (NN), Normal-Ventricular (NV), Ventricular-Normal (VN), Ventricular-Ventricular, or Unknown-Unknown according to the following rules:

- Two intervals match if they are each within 12.5% of their mean;
- An interval is classified as NN if it matches the previous NN interval;
- An interval is classified as NV if it is less than 75% of the most recent NN interval;
- An interval is classified as VV if it matches the previous NV or VV interval; and
- An interval is classified as VN if it is longer than the previous NV interval.

3.3. Noise detection

Similar to the low frequency noise measurement proposed by Moody *et al*. [7], our algorithm characterizes low frequency noise by the signal range in an interval between the end of the previous T-wave and the beginning of the following P-wave. Our algorithm characterizes high frequency noise by the signal range of the bandpass filtered beat outside of the QRS complex.

3.4. Beat matching

We compare beats to previously detected beats based on their average point-by-point difference in a 300 ms window surrounding the detection point. We use two measures of similarity. In the first similarity measure, we scale the beats so that their amplitudes match. In the second similarity measure, we do not scale the beats. In both cases, we normalize the measure by the amplitude of the beats.

Beat matching is influenced by the noise level, the R-to-R interval classification, and beat width. Stricter limits are used to match beats if they are premature. The algorithm is less likely to start a new beat type if the signal is noisy or the rhythm is regular.

3.5. Dominant normal beat selection

The dominant normal beat type is selected as the most frequently occurring normal beat type in the previous 180 beats. For the purpose of determining the dominant normal beat type, a beat is considered to be normal if it is part of a regular rhythm, there have been at least two consecutive beats of that type, and the beat width is less than 130 ms. If none of the last 180 beats meets these criteria, the most frequently occurring beat type is selected as the dominant beat type.

The dominant rhythm is classified as regular if fewer than 60 R-to-R intervals in the last 180 R-to-R intervals have been classified as QQ.

3.6. Beat classification

After a beat has been matched to a previous beat type, the algorithm classifies the beat according to the features described above. In fact, the algorithm performs three classifications. First, the algorithm attempts to classify the type that the beat matches according to the rules:

- Classify the type as normal if no dominant type has been identified yet and the last five beats were the same type.
- Classify the type as normal if the last two were the same type and the beat width is no more than 20 ms wider than the dominant beat type.
- Classify the type as ventricular if it is premature, part of a bigeminal rhythm, and is wider than 100 ms.
- Classify the type as normal if it is part of a bigeminal rhythm and is not premature.

If the beat's type has been classified, the beat takes on that classification. If the type has not been classified, the algorithm uses a more local classification based on the previous eight beats of that type. The previous eight beats of each type are classified based on how premature they were, the beat width, and their similarity to the dominant beat. If three of the last four beats or six of the last eight beats of the beat's type have been classified as ventricular, the beat is classified as ventricular, otherwise no local classification is made.

If no local classification is made the individual beat is classified based on twenty rules that take into account:

- Beat width;
- R-to-R interval classification;
- Noise level;
- Similarity to the dominant beat type;
- Width, relative to the width of the dominant beat, and
- Dominant rhythm

3.7. Results

We tested our beat classifier on the MIT/BIH and AHA databases. Our classifier detected ventricular beats on the MIT/BIH with a sensitivity of 93.91% and a positive predictivity of 96.48%. On the AHA database our classifier detected ventricular beats with a sensitivity of 93.26% and a positive predictivity of 97.83%.

4. Discussion

In this paper we have briefly described our open source beat detector and beat classifier. Both algorithms include details which, due to limited space, we have not described, but we believe that this is probably true of many published ECG analysis algorithms. Our algorithms not include any significant new concepts or innovations. What is new about our work is that we have made the source code available, so others will have complete access to the details of the algorithms. In addition to the source code, more detailed documentation is also available at www.eplimited.com.

It is our hope that others whose research involves beat detection and analysis or who are developing medical devises that perform beat detection and analysis will build directly on our work and make their modifications and improvements available as open source code. Since we posted our source code, at least 350 different users have downloaded our programs. Downloads have been roughly evenly distributed between students, academic researchers, and commercial product developers. Though there have been questions concerning the implementation, no one has yet contributed any improvements, additions, or useful modifications to the software.

Acknowledgements

This work was supported by grant R43 HL65813 from the National, Heart, Lung, and Blood Institute of the NIH.

References

[1] Pan J, Tompkins WJ. A real-time QRS detection algorithm, IEEE Trans. Biomed. Eng., 1985;32: 230-6.
[2] Hamilton PS, Tompkins WJ. Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database. IEEE Trans. Biomed Eng. 1986;33:1157-65
[3] ANSI/AAMI EC13:1992. American National Standard: Cardiac Monitors, Heart Rate Meters, and Alarms. Available at www.aami.org.
[4] Urrusti JL, Tompkins WJ. Performance evaluation of an ECG QRS complex detection algorithm. Proc. Annual International Conference of the IEEE Engineering inMedicine and Biology Society. 1993: 800–1.
[5] The MIT-BIH Arrhythmia Database CD-ROM. Available from the Harvard-MIT Division of Health Sciences and Technology, 1992.
[6] AHA Database for Evaluation of Ventricular Arrhythmia Detectors CD-ROM. Available from www.healthcare.ecri.org, 1997.
[7] Moody G, Mark R. Development and evaluation of a 2-lead ECG analysis program. Computers in Cardiology. 1982; 39-44.

Patrick Hamilton.
E.P. Limited
35 Medford St.
Somerville, MA 02143
pat@eplimited.com