Matlab Implementation of Pan Tompkins ECG QRS Detector.

Hooman Sedghamiz

March, 2014

Abstract

This open source code is a complete MATLAB implementation of Pan Tompkins ECG QRS detector. This work implements all of the main steps proposed in the original algorithm. A preliminary comparison of the MIT-BIH indicated similar sensitivity and accuracy between this implementation and the original proposed work.

1 Background

Pan and Tompkins proposed an accurate real-time ECG beat detector in 1985 that has been successfully used in many commercial devices [1]. The algorithm was originally implemented on a microprocessor and later in C programming language [2]. This work implements the same algorithm in MATLAB environment with the hope that it can be more easily accessible to researchers in the field of biomedical engineering.

2 Method

2.1 Preprocessing:

Steps:

1. Bandpass filter(5-15 Hz).
2. Derivating filter to high light the QRS complex.
3. Signal is squared.
4. Signal is averaged (MWI) to remove high frequency noise (0.150 seconds length).
5. Depending on the sampling frequency of your signal the filtering, options above are changed to best match the characteristics of the ECG signal

2.2 Decision Rule

At this point in the algorithm, previous steps have generated a roughly pulse-shaped waveform at the output of the signal. The decision as to whether a pulse corresponds to a QRS complex (as opposed to a high-sloped T-wave or a noise artefact) is performed with an adaptive thresholding operation and other decision rules outlined below;

- **FIDUCIAL MARK** - The waveform is first processed to produce a set of weighted unit samples at the location of the MWI maxima. This is done in order to localize the QRS complex to a single instant of time.

- **THRESHOLDING** - When analyzing the amplitude of the MWI output, the algorithm uses two threshold values (THR_{SIG} and THR_{NOISE}, appropriately initialized during a brief 2 second training phase) that continuously adapt to changing ECG signal quality. The first pass through $y[n]$ uses these thresholds to classify the each nonzero sample ($CURRENTPEAK$) as either *signal* or noise: If $CURRENTPEAK > THR_SIG$, that location is identified as a QRS complex candidate and the signal level (SIG_{LEV}) is updated: $SIG_{LEV} = 0.125 \times CURRENTPEAK + 0.875 \times SIG_{LEV}$ If $THR_{NOISE} < CURRENTPEAK < THR_{SIG}$, then that location is identified as a noise peak and the noise level ($NOISE_{LEV}$) is updated: $NOISE_{LEV} = 0.125 \times CURRENTPEAK + 0.875 \times NOISE_{LEV}$. Based on new estimates of the signal and noise levels (SIG_{LEV} and $NOISE_{LEV}$, respectively) at that point in the ECG, the thresholds are adjusted as follows: $THR_{SIG} = NOISE_{LEV} + 0.25 \times (SIG_{LEV} - NOISE_{LEV})$; $THR_{NOISE} = 0.5 \times THR_{SIG}$;These adjustments lower the threshold gradually in signal segments that are deemed to be of poorer quality.

- **SEARCHBACK FOR MISSED QRS COMPLEXES** - In the thresholding step above, if $CURRENTPEAK < THR_SIG$, the peak is deemed not to have resulted from a QRS complex. If however, an unreasonably long period has expired without an abovethreshold peak, the algorithm will assume a QRS has been missed and perform a searchback. This limits the number of false negatives. The minimum time used to trigger a searchback is 1.66 times the current R peak to R peak time period (called the RR interval). This value has a physiological origin - the time value between adjacent heartbeats cannot change more quickly than this. The missed QRS complex is assumed to occur at the location of the highest peak in the interval that lies between THR_SIG and THR_NOISE. In this algorithm, two average RR intervals are stored,the first RR interval is calculated as an average of the last eight QRS locations in order to adapt to changing heart rate and the second RR interval mean is the mean of the most regular RR intervals . The threshold is lowered if the heart rate is not regular to improve detection.

- **ELIMINATION OF MULTIPLE DETECTIONS WITHIN REFRACTORY PERIOD** - It is impossible for a legitimate QRS complex to occur if it lies within 200ms after a previously detected one. This constraint is a physiological one due to the refractory period during which ventricular depolarization cannot occur despite a stimulus [1]. As QRS complex candidates are generated, the algorithm eliminates such physically impossible events, thereby reducing false positives.

- **T WAVE DISCRIMINATION** - Finally, if a QRS candidate occurs after the 200ms refractory period but within 360ms of the previous QRS, the algorithm determines whether this is a genuine QRS complex of the next heartbeat or an abnormally prominent T wave. This decision is based on the mean slope of the waveform at that position. A slope of less than one half that of the previous QRS complex is consistent with the slower changing behaviour of a T wave otherwise, it becomes a QRS detection. Extra concepts: beside the points mentioned in the paper, this code also checks if the occured peak which is less than 360 msec latency has also a latency less than $0,5 * mean_R R$ if yes this is counted as noise!

3 Evaluation of several MITDB database

Table 1: Detailed Performance of the beat detector on 10 sample records of MITDB*

Record (No.)	**Total (No. Beats)**	**FP (Beats)**	**FN (Beats)**	**Failed (Beats)**	**Failed (%)**
100	2274	0	0	0	0
101	1874	0	6	6	0.32
102	2187	0	0	0	0
104	2230	1	2	3	0.13
105	2572	48	32	80	3.11
108	1824	61	71	132	7.24
200	2601	1	3	4	0.15
202	2146	0	6	6	0.279
219	2312	0	0	0	0
222	2634	131	2	133	5.04

*Sample recordings were chosen randomly.

References

[1] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.

[2] P. S. Hamilton and W. J. Tompkins, "Quantitative Investigation of QRS Detection Rules Using the MIT/BIH Arrhythmia Database," *IEEE Transactions on Biomedical Engineering*, vol. BME-33, no. 12, pp. 1157–1165, 1986.