Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 9 (2013) 1159 – 1165

Procedia Technology

CENTERIS 2013 - Conference on ENTERprise Information Systems / ProjMAN 2013 - International Conference on Project MANagement / HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies

A comparison of three QRS detection algorithms over a public database

Raúl Alonso Álvarez[a], Arturo J. Méndez Penín[a], X. Antón Vila Sobrino[a]*

[a]*Computer Science Department, Universidad de Vigo, Ourense 32004, Spain*

Abstract

We have compared three of the best QRS detection algorithms, regarding their results, to check the performance and to elucidate which get better accuracy. In the literature these algorithms were published in a theoretical way, without offering their code, so it is difficult to check its real behaviour over different collections of ECG records. This work brings the community our source code of each algorithm and results of its validation over a public database. In addition, this software was developed as a framework in order to permit the inclusion of new QRS detection algorithms and also its testing over different databases.

Keywords: ECG; QRS; Phasor

1. Introduction

An electrocardiogram, also called ECG or EKG, reflects the electrical activity of the heart. Every heart contraction produces an electrical impulse that is caught by electrodes placed on the skin. The heartbeat

* Corresponding author.
E-mail address: anton@uvigo.es

doi:10.1016/j.protcy.2013.12.129

produces a series of waves whose morphology varies over time. These waves are caused by voltage variations of the cardiac cells [1].

Electrocardiogram analysis provides important and relevant information about heart state. Physicians all over the world are using it to diagnose cardiac diseases, which are one of the main causes of mortality in our society. Nowadays ECG information can be easily digitized and processed by a computer. Thus, using the power of computers, we can detect heart diseases or anomalies that otherwise could only be detected by experts physicians.

In ECG processing it is very important to accurately detect heartbeats, because it is the base for further analysis and can also be used to get information about heart rate. The energy of heartbeats is mainly located in the QRS complex, so an accurate QRS detector is the most important part of ECG analysis.

QRS detection is difficult, because the beat morphology varies along the time, and different sources of noise can be present. Most QRS detection algorithms have two differentiated stages: preprocessing and decision [2]. In preprocessing stage different techniques are applied to the signal, such as linear and non-linear filtering or smoothing to attenuate P and T waves as well as noise. In decision stage the most important task is the determination of thresholds and in some cases the use of techniques to discriminate T waves. Some algorithms include another decision stage to reduce false positives.

Software QRS detectors are still an important topic in research. Nowadays there are published many algorithms for detecting heartbeats, but most of them do not offer the source code and have not been validated on the same databases. Usually these algorithms are explained in a theoretical way, while others only include a few guidelines for real implementation. It becomes necessary, therefore, a tool that allows users to implement their own algorithms and compare its performance against different databases.

To evaluate the performance of each algorithm there are available databases, which contain a large variety of ECGs, as well as signals that are rarely observed but clinically important. Some of these databases are: MIT-BIH [3], QT Database (http://www.physionet.org/physiobank/database/qtdb/doc/node3.html) or AHA (http://www.heart.org/HEARTORG/). In this work we used the MIT-BIH Arrhythmia Database.

This work was developed using the free software programming language R [4], very powerful using matrix operations and simple to plot the results, which makes it very adequate for signal processing. We linked R with C language, this way computationally intensive tasks are calculated in C, in order to reduce processing time.

2. State of the Art

During the last 30 years there have been proposed lots of algorithms for QRS detection. There are many approaches, from artificial neural networks or genetic algorithms to wavelet transforms, filter banks, heuristic methods or machine learning methods [5].

Table 1. Performance of some QRS detection algorithms. For each algorithm we show the best results provided by their authors.

Algorithm	Database	Sensitivity	Pos. Predictivity
N. Arzeno *2008* [6]	MIT-BIH	99.68%	99.63%
J. Martinez *2004* [7]	MIT-BIH, QT, ST-T, CSE	99.66%	99.56%
B. Abibullaev *2011* [8]	MIT-BIH	97.2%	98.52%
J. Pan *1985* [9]	MIT-BIH	99.3%	-
P. Hamilton *1986* [10]	MIT-BIH	99.69%	99.77%
A. Martinez *2010* [11]	MIT-BIH, QT, ST-T, TWA	99.81%	99.89%

Most algorithms are developed in research groups, they own the source code and they only share with the scientific community the behavior of their algorithms with a few guidelines and the results of its validation over a database. Table 1 shows the performance of some of the most cited QRS detection algorithm.

As can be seen, sensitivity and positive predictivity oscillate around 99.3 and 99.8, except the case of Abibullaev.

3. Methods

The implementation described in this paper is part of a package that we are currently developing in our research group. We are creating a software tool that will permit users to read ECG signals (in different formats), analyze them using different algorithms, visualize its features and export some results.

Currently our package already includes functions to read different ECG formats (like the one used in physionet databases), three algorithms for QRS detection, some functions to display ECG signal with annotations over it and the possibility to exporting heart beat positions.

This software (still under construction) was developed using the programming language R [4] (free software). We have chosen this programming language because it is widely known in the scientific community and it is easy for non programming experts. Among its advantages are: it is open source, it is based on matrix operations (very adequate for signal processing) and it provide simple plotting functions. Nevertheless R is extremely slow in computationally intensive tasks, for this reason we linked R with C language, this way computationally intensive tasks (such as loops) are calculated in C, which really reduces processing time. This software can be downloaded from *http://recg.milegroup.net* as a compressed R package.

In this section we introduce a description of the algorithms and briefly explain some of the used techniques. We decided to implement two algorithms based on digital filters, Pan & Tompkins algorithm [9] and Hamilton & Tompkins algorithm [10] and a new algorithm based on the phasor transform [11].

3.1. Pan & Tompkins algorithm

This algorithm is based on the slope, amplitude and width of the signal. It is divided in two different stages: preprocessing and decision. In the preprocessing stage the signal is prepared for later detection, removing the noise, smoothing the signal and amplifying the QRS slope and width. Later in the decision stage, thresholds are applied to the signal in order to remove noise peaks and consider only signal peaks.

The first step is preprocessing, where the signal is passed through a block of filters to reduce noise and influence of the T wave. Next, the derivative is applied, providing complex slope information. Then the signal is squared point by point, intensifying the slope and reducing false positives. Finally, a moving window integrator is applied, including information about slope and width of the signal.

In the decision stage two sets of thresholds are applied to both, derivative signal and the moving window integration signal (henceforth mwi). By using thresholds in both signals, the reliability of detection is improved.

Thresholds float over the noise, adjusting them to the signal changing conditions automatically. We establish the fiducial mark in R peaks, hence we first detect all local maxima peaks (Fig. 1-a) and then applying thresholds we only consider those peaks exceeding the thresholds. To detect a local maxima we use a moving window, when the middle point is the maximum value of the window, is considered a local maxima.

To classify the two first R peaks correctly, we split the signal into segments (Fig. 1-b), in which at least one heartbeat is present, and we look for the maximum and minimum peak in each segment. And then we calculate the median of these maximum (minimum) peaks, and set a threshold at 35% of this median (we determined this percentage experimentally). We consider the first peaks exceeding the threshold as the two first R peaks (Fig. 1-c) and we classify all peaks that not exceed this threshold until the second R peak, as noise. Then we apply the equations according to author's indications [9].

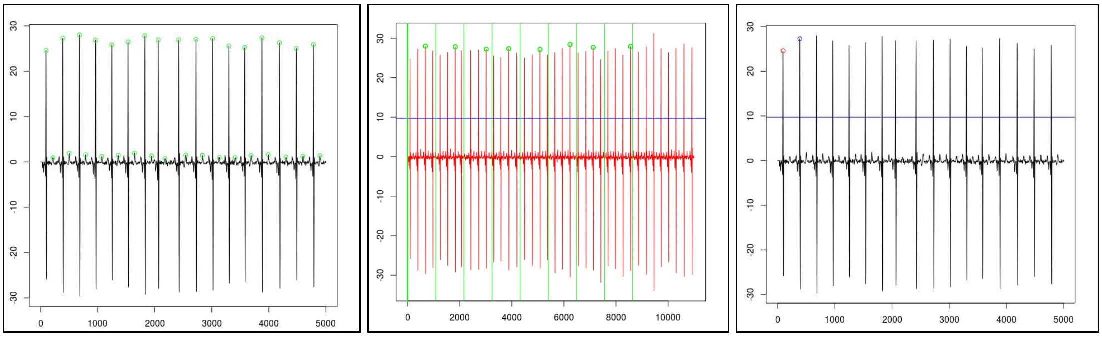

Fig. 1 (a) Local maxima peaks (b) Splitting the signal in segments (c) Detecting the two first R peaks

A peak is considered as valid when it matches in the derivative and mwi signals, because we look for peaks in both signals. Because it is very unlikely to detect a peak in the same position in both signals, we establish an interval of 50 samples to consider a peak match.

3.2. Hamilton & Tompkins algorithm

Hamilton & Tompkins algorithm [10] is very similar to the previous one, sharing the same preprocessing stage, however is completely different in the decision stage. They focused on optimizing decision rules, they tested the performance of three estimators (mean, median and an iterative peak level) to place the adaptive threshold.

We used the median as estimator, which is the one they reported as the best, both for detection accuracy and for fewer false positives.

To avoid the problem of multiple detection due to ripples, we have used a different technique (easier than the author's algorithm). In our case, a peak is only considered when it is the higher value in a window interval, this way multiple detection is avoided.

This algorithm and the previous, use a technique to detect lost peaks, consisting in a search back when a peak is not detected in a certain time, and a refractory blanking to ignore those peaks closer to an already detected peak.

3.3. Phasor transform algorithm

The detection algorithm based on the Phasor transform [11] (henceforth PT) is recent, year 2010, and it is characterized by its robustness, low computational cost and mathematical simplicity. This algorithm converts each ECG sample into a complex number (called phasor), preserving its information, regarding root mean square and phase values. Thus, considering instantaneous phase variation in consecutive samples of the phasor transformed ECG, the slight wave variations in the original signal, are maximized. This way the authors apply thresholds to detect the QRS complex.

In the preprocessing stage the signal is passed through a forward/backward high-pass filter to remove the baseline wander. Moreover, although it is not mentioned in the article, it is convenient to normalize the signal before applying the PT. We tried to normalize the signal using a moving window, but finally we discarded this option because it added much processing time and the results improvement was poor. However we achieve better results using an adaptive threshold.

Due to the experience we had from the previous algorithms, we decided to calculate the adaptive threshold based in the median of the latest detected peaks, placing the threshold 0.001 below the median of the last peaks.

We also used the search back and a technique to remove false positives, but instead of using only the last RR interval, we decided to improve it using the median of the past 8 RR intervals as in Hamilton & Tompkins [10] algorithm. After applying these techniques we reached a little better detection rates.

4. Results and Discussion

In this section we present a comparison of the results obtained with the three implemented algorithms. The presented values are the results we got after multiple testing with different values of each paramenter in each algorithm.

We can get better results in sensitivity or predictivity changing the parameters, but improving a term causes worsen the other. For example reducing the window width to search the local maxima peaks, the algorithm would detect more peaks, which would increase the sensitivity, however the predictivity would be worse. Hence, results in tables are the best for each algorithm in terms of sensitivity and positive predictivity, but they can vary changing algorithm parameters.

In Table 2 are presented the average results obtained after validated with each algorithm the entire database. These results are closer to those reported by their authors, but in case of phasor algorithm our results are a bit far. Authors of phasor transform work, do not fully explain the necessary preprocessing operations before applying the transform [11]. This may be the reason why we were not able to achieve better results.

Table 2. Mean results for each algorithm over all database records

Algorithm	Sensitivity	Pos. Predictivity	RMS RR Error
Pan & Tompkins	99.79% ± 0.34	99.84% ± 0.42	95.88 ms
Hamilton & Tompkins	99.54% ± 0.69	99.42% ± 1.19	107.03 ms
Phasor	98.41% ± 4.95	86.75% ± 17.30	354.84 ms

As can be seen perform of Pan & T. and Hamilton & T. is good, with values between 99.42 and 99.84. Phasor transform has worse performance, specially the positive predictivity (86.75).

To compare the performance among algorithms it is important to check the significant differences between them, presented in Table 3. From this table we can distinguish between significant differences in sensitivity and positive predictivity.

Table 3. Significant differences in the comparison among algorithms

Compared algorithms	Sensitivity P value	Pos. Predictivity P value
Pan & T. vs Hamilton & T.	0.0301	0.1231
Pan & T. vs Phasor	0.0616	3.73e-06
Hamilton & T. vs Phasor	0.1266	6.85e-06

In terms of sensitivity we can conclude that Pan & T. algorithm achieved the best result, because it gets the best average results (Table 1) and also the difference is significant compared with Hamilton & T.algorithm.

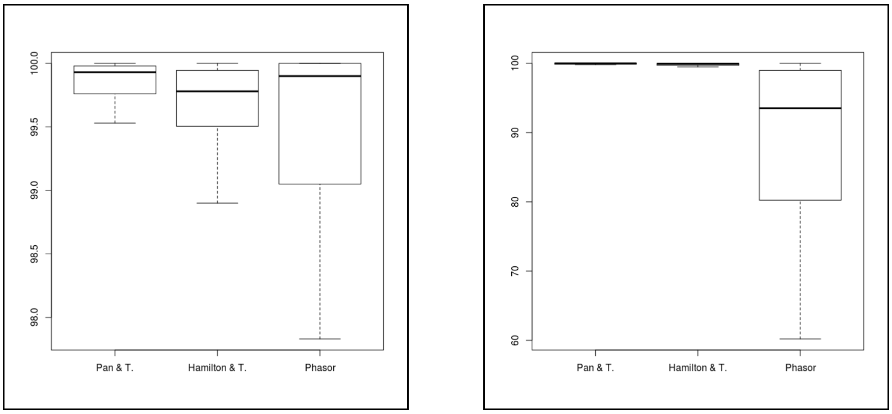

Fig. 2 (a) Sensitivity comparison among algorithms (b) Predictivity comparison among algorithms

The comparison between Hamilton & T. and Phasor algorithms shows that the first gets better average results, however the differences are not significant.

In terms of predictivity, the difference between Pan & T. and Hamilton & T. is not significant, however both algorithms get better results than Phasor algorithm, whose predictivity is very low and the differences are significant.

Another comparison is shown in Fig. 2, where the distribution of the results is shown in a boxplot. The dark horizontal line represents the median values.

In Fig 2-a we can appreciate that the best results are achieved by Pan & T. algorithm, furthermore it is noteworthy that Hamilton & T. algorithm gets better results in average than Phasor one, however Phasor median values are better than Hamilton & T. ones. This is caused because there are some records in the database where Phasor algorithm gets very bad results, while Hamilton & T. is more consistent.

Regarding the predictivity values (Fig 2-b), results show that the worst values are provided by the Phasor algorithm.

5. Conclusions and future work

In this research work we have compared the performance of three of the best QRS detection algorithms, getting the results presented in the previous section.

As a result, besides the implementation of the three algorithms, we have begun the development of an ECG analysis software tool. This tool will permit to read ECGs records in different formats, to develop an ECG analysis, visualization of different plots and the automated validation of each algorithm over any database. At this moment, this tool has already incorporated: three QRS detection algorithms, different plotting functions and scripts for automatic validation.

We have created our software as an open source tool to share it with the scientific community. This way anyone can use it as a basis for its own work. In addition, we provide the implementation of some beat detection algorithms, one of them with a good accuracy that greatly facilitates the detection or subsequent delineation of all ECG waves.

Our software is also suitable for those who want to start in electrocardiography, offering the opportunity to test some functionality, and access the source code and documentation, where each developed algorithm is explained.

A future improvement will be: to complete the package, to accept more ECG formats, and to validate these algorithms over different databases.

References

[1] L. Sörnmo and P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. Elsevier Academic Press, June 2005

[2] O. Pahlm and L. Sörnmo, "Software QRS detection in ambulatory monitoring – a review", *Medical and Biological Engineering and Computing*, vol. 22, no. 4, pp. 289-297, 1984.

[3] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database", *Engineering in Medicine and Biology Magazine*, IEEE, vol. 20, no. 3, pp. 289-297, 1984.

[4] R Core Team, R: *A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900041-07-0.

[5] B.-U. Kohler, C. Henning, and R. Olgmeister, "The principles of software QRS detection", *Engineering in Medicine and Biology Magazine*, IEEE, vol. 21, no. 1, pp. 42-57, 2002.

[6] N. Arzeno, Z.-D. Deng, and C.-S. Poon, "Analysis of first-derivative based QRS detection algorithms", *Biomedical Engineering*, IEEE Transactions on, vol. 55, no. 2, pp. 478-484, 2008.

[7] J. Martinez, R. Almeida, S. Olmos, A. Rocha, and P. Laguna, "A wavelet based ECG delineator: evaluation on standard databases", *Biomedical Engineering*, IEEE Transactions on, vol. 51, no. 4, pp. 570-581, 2004.

[8] B. Abibullaev and H. Seo, "A new QRS detection method using wavelets and artificial neural networks", *Journal of Medical Systems,* vol. 35, no. 4, pp. 683-691, 2011.

[9] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm", *Biomedical Engineering*, *IEEE Transactions on,* vol. BME-32, 1985.

[10] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH Arrhythmia database", *Biomedical Engineering, IEEE Transactions on,* vol. BME-33, no. 12, pp. 1157-1165, 1986.

[11] A. Martínez, R. Alcaraz, and J. J. Rieta, "Application of the phasor transform for automatic delineation of single-lead ECG fiducial points", *Physiological Measurement*, vol. 31, no. 11, p. 1467, 2010.