Analysis of electrocardiographic (ECG) signals

Biljana Vitanova
OBSS 2024/25 , FRI, UL
bv7063@student.uni-lj.si

***Abstract*—This project explores QRS complex detection using multichannel signals, including ECG and BP, from the PhysioNet 2014 Challenge datasets. Two datasets were used: 'set-p' and 'training'. Preprocessing steps included downsampling, trimmed moving average filtering, and range filtering to enhance detection. The algorithm performed well on the 'set-p' dataset, matching results from the original paper, but showed weaker performance on the 'training' dataset.**

I. Introduction

Accurate detection of QRS complexes is essential for analyzing heart rate and diagnosing conditions such as arrhythmias and myocardial infarctions. However, relying solely on ECG signals can sometimes lead to errors due to noise, signal artifacts, or physiological factors that obscure QRS detection. To improve robustness and accuracy, multivariate channel combinations—such as blood pressure (BP), central venous pressure (CVP), pulmonary artery pressure (PA), and arterial pressure (ART)—can be used to complement ECG signals. These additional channels provide alternative perspectives on cardiac activity, helping to mitigate issues like signal dropout or distortion in any single channel.

This project aims to implement one of the algorithms introduced by Marcus Vollmer for multichannel QRS detection [1]. The approach incorporates multiple stages to ensure reliable and accurate detection of QRS complexes in multivariate datasets.

II. Methodology

A. Preprocessing

Each signal undergoes a series of preprocessing steps. First, the signal is **downsampled**. Next, a **trimmed moving average filter** is applied to smooth the signal by removing sudden spikes or drop-offs. This filter operates on a sliding window to the left of each sample, discarding a certain percentage of the highest and lowest values in the window before calculating the mean of the remaining samples.

The signal is then **standardized** to ensure consistency across different channels by transforming it to have a mean of 0 and a standard deviation of 1. Finally, a **range filter** is applied using a sliding window to compute the difference between the local maxima and minima within the window.

B. Beat Extraction

The range signal is first **smoothed** to reduce noise. Within a sliding window, **local maxima** and **minima** are calculated to identify significant changes in the signal. A beat is detected when the range signal exceeds an **adaptive threshold** based on these maxima and minima, and it remains **consistent** over a short **constancy window** to ensure it's not noise. A new beat is only accepted if a lower signal value exists between consecutive detections. In this stage, **noisy segments** are identified where the difference between local minima and maxima is small. These segments are later used during the merging stage to exclude unreliable detections.

C. Relevant channels

To determine whether a channel is suitable for beat detection, the **relative RR intervals** (the ratio of two consecutive RR intervals) are calculated. For a healthy heart, these ratios typically fall within the range of [0.8, 1.2] for at least 80% of the beats, indicating regular heart rate variability. Channels that satisfy this criterion are then used for multivariate combination.

D. Delay correction

Different signals are measured at slightly different times due to the varying time it takes for the signal to reach each device. For the channels excluding ECG, a **dynamic delay correction** is performed by aligning detected beat positions with the nearest reference beats from the ECG. This correction is further refined using a **median filter** applied iteratively over **60-second** intervals.

E. Multivariate combination

Beat positions from all channels are **sorted**, and a heartbeat is considered safe if it appeared across all **non-noisy channels** and the time differences between consecutive detections were within a defined limit. The final heartbeat position was taken as the **average of these detections**.

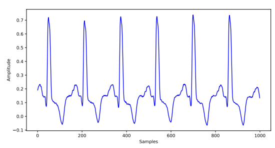
III. Experiments

For this project, two datasets from the PhysioNet 2014 Challenge [2] were used. The first dataset, set-p, contains 100 signals sampled at 250 Hz. The second dataset, the training dataset, also contains 100 signals but is sampled at 360 Hz. Both datasets include signals from various channels, such as ECG, BP, EOG, ART, and PAP.

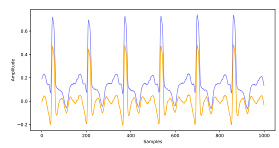
For the preprocessing step, I downsampled the signals to **200 Hz**. In the paper [1], the recommended sampling rate is 80 Hz; however, I obtained better results using 200 Hz.

When applying trimming, I used a **0.2-second** window for the ECG setting and a **1.0-second** window for the BP setting, with an **alpha** value of **0.25**. For the range filter, I used a **0.25-second** window for ECG and a **0.4-second** window for BP.

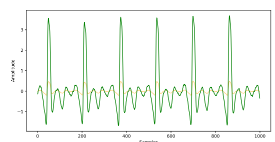
In all cases, I considered the ECG channel to be healthy, although this was not always the case.



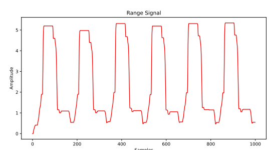
(a) Raw signal



(b) TMA filtered signal



(c) Standardized signal



(d) Range signal

Fig. 1: Preprocessing step

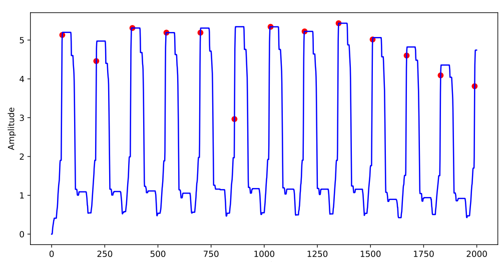


Fig. 2: Beat Extraction Step

IV. Results and Discussion

A. Results

TABLE I: Performance metrics on the **set-p dataset**

	Sensitivity [%]	Positive Predictivity[%]
Average	99.85	99.33
Gross	98.92	99.33

TABLE II: Performance metrics on the **training dataset**

	Sensitivity [%]	Positive Predictivity[%]
Average	56.99	60.07
Gross	56.30	62.52

B. Discussion

On the 'set-p' dataset, the results were as good as those reported in the original paper. However, when evaluated on the 'training' dataset, the performance was worse. In my observation, many true detections were missed, likely because the final step of the algorithm—verifying candidate detections—was not implemented. Including this step I believe would improve detection accuracy, as it ensures the detection does not rely solely on all channels.

V. Conclusion

The QRS detection algorithm performed well on the 'set-p' dataset, achieving results comparable to the original study. However, its weaker performance on the 'training' dataset highlights areas for improvement. Implementing the final candidate verification step I believe is crucial to address missed detections and improve accuracy.

References

[1] M. Vollmer, "Robust detection of heart beats using dynamic thresholds and moving windows," in *Computing in Cardiology 2014*, 2014, pp. 569–572.

[2] G. Moody, B. Moody, and I. Silva, "Robust detection of heart beats in multimodal data: The physionet/computing in cardiology challenge 2014," in *Computing in Cardiology 2014*, 2014, pp. 549–552.