Henry Butlin, 2024-05-13

electronRx Cardio Challenge Narrative

To get an idea of which processing techniques to apply, I created plots of the raw PPG data. The main features that seemed to be hindering peak-finding were signal noise and baseline drift, and that the relative amplitudes of each peak varied quite a lot. My first approach was a bandpass filter, as it would both de-noise the signal and effect a baseline correction. To do this, I modified some C code I had written previously for coursework on Fourier Transforms. The generated plots were visually 'cleaner', but the extracted cardiac metrics were quite far off the expected values. This was probably due to the missing data, and the implementation of an unsophisticated filter. Although I had an idea of how this could be fixed, I guessed that fixing these problems would take longer than the scope of the challenge, so I changed my approach.

To de-noise the data I used a Savitzky-Golay filter, and subtracted a fitted polynomial from the raw data as a baseline correction. By eye the generated plots seemed cleaner, and extracted metrics were closer to those expected. As a sanity check, I wrote a peak-finding script that found local maxima in a moving window of data to act on the raw data. The metrics this approach gave were different from the filtered ones, but similarly close to the expected ones. I concluded from this that the signal pre-processing I applied does not have a clear positive effect on the extracted values, although it does markedly improve the visual appearance of the data. A quick fix I used to deal with the missing data was to modify the measured heart rate (number of peaks) relative to the fraction of missing data. In the future I would attempt to fill the gaps with (e.g.) maximum entropy interpolation, or apply a deep learning approach like recurrent neural networks.

Both methods used to process the signal lacked sophistication, and gave results of the correct order but in some cases far off the expected ones. To fix this, I would go with a more polished method like the automatic multiscale-based peak detection (AMPD) algorithm from an article I found[1], which gives blood volume pulses as an example of its possible applications.

A rudimentary approach to distinguish between atrial fibrillation and sinus rhythm might be to make use of the rMSSD and SDNN metrics. Since AF is characterised by irregular heartbeats, larger deviations in the successive differences between heartbeats would imply a higher likelihood of being in AF. A more reliable and sophisticated method could be a deep learning model, like the one used in a study[2] from 2020. The study seems highly relevant to this task, and found that a deep convolutional-recurrent neural network could accurately predict if a subject was in AF given raw PPG data. To visualise the results of this project I made plots of the data, marked detected peaks, and printed the extracted cardiac metrics to the console.

[1] Scholkmann, F., Boss, J., & Wolf, M. (2012). An Efficient Algorithm for Automatic Peak Detection in Noisy Periodic and Quasi-Periodic Signals. Algorithms, 5(4), 588–603. https://doi.org/10.3390/a5040588
[2] Aschbacher, K *et al.* (2020). Atrial fibrillation detection from raw photoplethysmography waveforms: A deep learning application. Heart Rhythm O2, 1(1), 3–9. doi:10.1016/j.hroo.2020.02.002. [PMCID: PMC8183963]