

# Codeanalyse und Limitationen

Da es mein eigener Code ist, habe ich auf die Hervorhebung von „positiven“ Sachen verzichtet, und mich ausschließlich auf die Limitationen fokussiert die ich dann selber im Nachhinein beim Testen entdeckt habe.

## Peak Detection

Die größte Limitation des Codes, in der Funktionalität, liegt auf jedenfall bei der r Peak Detection. Es gibt keine Berücksichtigung von Signal Veränderungen (wie es bspw in den sample Files sichtbar ist), und keine tatsächliche adaptive Schwellenwertanpassung. Das macht den Algorithmus sehr unrobust, vor allem bei größeren Analysefenstern.

Auch in kleineren Fenstern kann es hier schon zu Problemen kommen, bspw. wenn die Patient:innen sich bewegen und daher die Baseline driftet.

Die Limitation des Algorithmus ist besonders sichtbar beim SDNN (wie auch im Paper beschrieben wird), aber auch beim NN50, wo größere Fenster auch mehr Abweichungen produzieren. Das lässt sich möglicherweise auf eine fehlerhafte und ungenau werdende Peak Detection schieben.

Das Removal von Outlier hilft ein bisschen, aber trotzdem nicht besonders viel, wenn sich innerhalb des Fensters die Baseline verschiebt. Wenn man höhere Clipping nimmt (aber auch bei höheren Intervallen), wird das Ergebnis ungenauer (es gibt mehr verpasst Peaks) und es kommt zu „unphysiologischen HR“. Im Test liegen die „Spitzen“ bei 235 bpm: das kommt durch die Abtastrate und die refractory period (und die daraus folgende minimumDistance [in samples] zur Peak Detection). Das ist kein Zufall und ist ein weiterer Hinweis auf eine wenig robuste Peak Detection.

Besser wäre es eben den Schwellenwert adaptiver zu gestalten, oder anstatt von Spitzen zu suchen, einen Algorithmus zu implementieren der den Anstieg kontrolliert (r Peaks haben einen rasanten Anstieg, der unter der Baseline beginnt, und dann einen rasanten Abfall bzw. einpendeln).

Interessant ist, dass Peaks gefunden werden, bei einem statischen Signal. Hier wäre zu erwarten gewesen, dass es verworfen wird, oder anderweitig erkannt.

## Hauptanalyse

analyze( file gibt None zurück und printet auf die Konsole. Für ein Tool oder Code was man dann zur Weiterverarbeitung nutzen möchte, ist das nicht ideal. Beim erstellen von Test Cases muss man stdout gesondert capturen, was in dem Fall okay ist, aber in einer richtigen Nutzung problematisch ist, und das Tool ungeeignet macht. Die einzelnen Methoden kann man zwar separat aufrufen, aber da kritischer Code in analyze( file ist, kommt man nicht drumrum eben diese Methode nutzen zu müssen.

## HRV Metriken

Hier würde sich auch eine Artefakt Korrektur, bzw zu mindestens ein Hinweis, anbieten, bei unphysiologischen NN Intervallen. Auch das SDANN wird nicht angezeigt wenn das Signal grade so zu kurz ist (länger als 5 Minuten, aber kürzer als 10). Für diesen Fall hätte man vielleicht irgendwas machen können, wie bspw überlappende Fenster (aber da bin ich mir unsicher ob das überhaupt zielführend für diese Metrik ist)