university of groningen

faculty of science and engineering

Development of an interface to improve HRV analysis: combining automatic and manual processing

Guillermo Llopis

university of groningen

faculty of science and engineering

University of Groningen

Development of an interface to improve HRV analysis: combining automatic and manual processing

Master's Thesis

To fulfill the requirements for the degree of
Master of Science in Computational Cognitive Science
at University of Groningen under the supervision of
Dr. Fokie Cnossen (Artificial Intelligence, University of Groningen)
and
Liam Wietzorrek (University Medical Center Groningen)

Guillermo Llopis (s4897668)

February 16, 2024

Contents

Abstract

In response to the limitations of existing heart rate variability (HRV) analysis tools, we have developed a novel software that amalgamates advanced automatic correction capabilities with an intuitive user interface. Traditional HRV programs were often criticized for their outdated design, lack of user-friendliness, and overly time-consuming processes. Additionally, more modern programs rely heavily on automatic corrections without allowing users to verify or adjust these modifications, leading to potential inaccuracies.

Our program addresses these issues by providing a comprehensive suite of automatic correction options while maintaining user oversight, enabling practitioners to validate and, if necessary, override automated edits to the RR interval data. The software has undergone rigorous validation against manually annotated datasets, demonstrating high concordance with expert analysis. Feedback from initial user evaluations has been instrumental in identifying further enhancements, such as refining beat detection algorithms for noisy signal environments and incorporating additional features suggested by end-users. This user-focused approach to HRV analysis not only streamlines the process but also ensures a higher level of precision and customization, catering to the specific needs of researchers and clinicians in the field.

1 Introduction

We begin this report by exploring the Electrocardiogram (ECG) and Heart Rate Variability (HRV), focusing on their roles in understanding stress within the medical profession, particularly among surgeons. Initially, we cover the basics of ECG and HRV, including their recording methods, as discussed in Section 1.1. This foundation is crucial for understanding how these tools can assess the stress levels faced by surgeons and the potential of this stress to lead to medical errors.
Next, we delve into the relationship between HRV and stress, and its significance for medical professionals, as outlined in Section 1.2. This connection is vital because it demonstrates the importance of accurately measuring stress and its impact on the medical field. We expand the discussion to include the application of HRV in monitoring stress among surgeons and the implications of stress-induced errors in medical practice.
Our project aims to enhance the speed and accuracy of HRV analysis, qualities that current HRV analysis options do not always combine effectively. We focus on understanding both the technical and user-specific aspects of HRV analysis. This objective is detailed in Sections 1.3 and 1.4, where we highlight the development of a user-friendly system. By mapping the journey from basic HRV concepts to their application in stress measurement, and then to the creation of a practical analysis system in Section 1.5, we underscore the project's goal: to develop a solution that is both scientifically robust and directly applicable to its intended users. This approach ensures that the system advances the scientific understanding of HRV and meets the practical needs of users, allowing them to comprehend what HRV reflects at each moment. Ultimately, this aims to reduce the impact of stress on medical professionals and enhance patient care.

1.1 Introduction to Electrocardiogram and Heart Rate Variability

1.1.1 Electrocardiogram

The Electrocardiogram (ECG) is the signal that represents the electrical activity of the heart. The typical ECG signal is shown in Figure 1, where the different events are shown (P, Q, R, S and T). The QRS complex in an electrocardiogram (ECG) is a crucial aspect of understanding the heart's electrical activity and is deeply intertwined with the cardiac cycle. This cycle comprises two main phases: systole, where the heart muscle contracts to pump blood, and diastole, during which the heart relaxes and fills with blood. The electrical impulse that drives this cycle originates in the heart's natural pacemaker, the sinoatrial (SA) node, located in the right atrium. It travels through the atria, triggering their contraction and pushing blood into the ventricles. The impulse then reaches the atrioventricular (AV) node, where it briefly slows, allowing the ventricles to completely fill with blood, before continuing along the His-Purkinje system to the ventricles. [43]
The QRS complex represents this crucial moment of ventricular depolarization. It begins with the Q wave, often a subtle initial negative deflection following the atrial depolarization indicated by the P wave, and signifies the start of the ventricles' depolarization. This is followed by the R wave, an upward deflection representing the main mass of the ventricles depolarizing. The R wave is normally used to identify the beats, as it is the highest and most easily identifiable peak in the signal. The complex concludes with the S wave, a downward deflection indicating the final depolarization at the base of the heart. These components collectively signal the systolic phase of the cardiac cycle, where the heart pumps blood to the lungs and the rest of the body. The QRS complex's duration, amplitude, and morphology are crucial in diagnosing various cardiac conditions, such as ventricular hypertrophy, bundle branch blocks, and myocardial infarction. It's through this complex that clinicians can gain

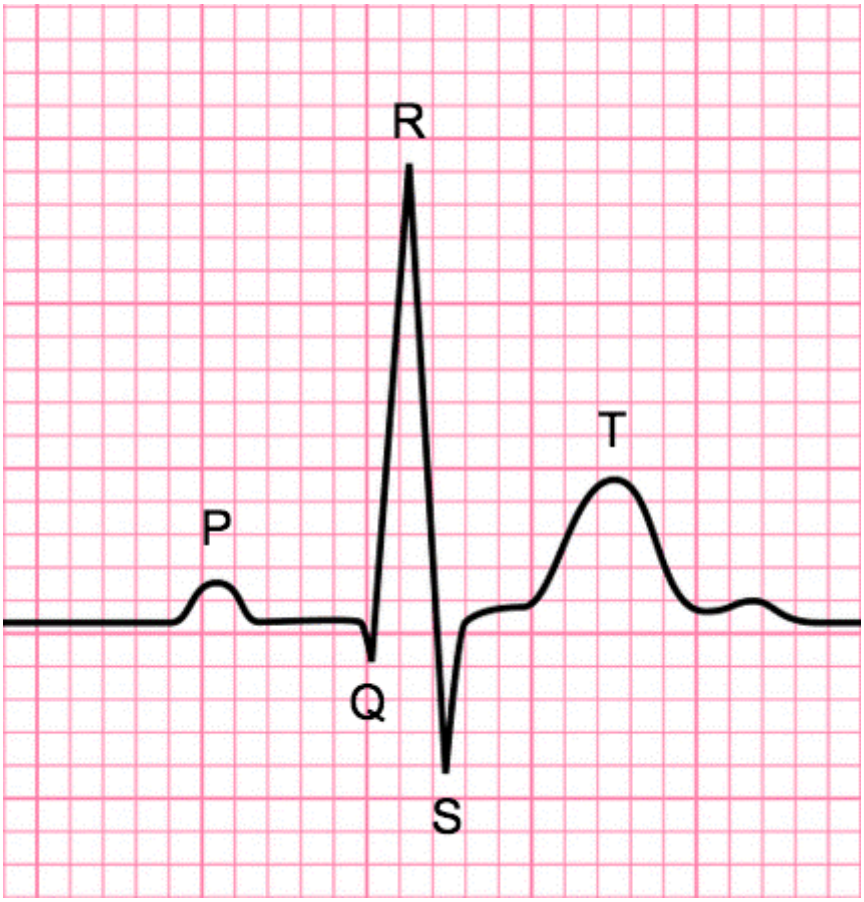


Figure 1: ECG signal [43]

vital insights into the heart's electrical functioning and overall health, making it an indispensable tool in cardiac diagnosis and treatment. The average resting heart rate for an adult is between 60 and 100 beats per minute (bpm) [16]. However, this range can vary depending on factors such as age, sex, physical fitness, and overall health. For instance, athletes or physically active individuals may have a lower resting heart rate, while those with certain medical conditions may have a higher resting heart rate [16].
And the actual heart rate at certain moment depends on the situation. Being under stress or doing a physical activity can increase the heart rate. The heart is a muscle that pumps blood through the body. During exercise or stress, the body needs more oxygen and nutrients, so the heart has to work harder to supply them. As a result, the heart rate increases. For example, Formula 1 drivers can reach 200 beats/minute during a race [13].

1.1.2 ECG recording

ECG measuring does not require a big equipment and can be done almost anywhere. There are many ways this can be done, some of the most popular examples are:

- Standard 12-lead ECG. It involves placing 10 electrodes in different parts of the body to record the ECG from different angles. This is the standard way of measuring ECG in a clinical setting, as it is also the most accurate one.

- Holter monitor. Portable device that records the ECG during many hours or days. It can be

Figure 2: Polar band chest ECG recording sensor

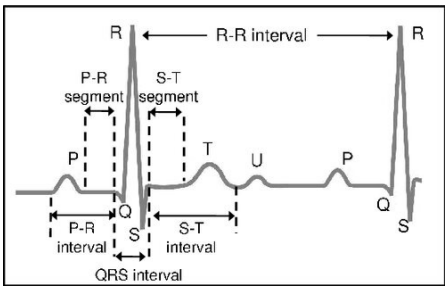


Figure 3: RR interval [35]

worn by a person who goes about their daily life.

- Event monitor. The person wears the device and this starts recording when an event is detected. This can be triggered by the patient or an abnormality detected by the sensor.

- Stress test. The ECG is recording while the patient does a physical activity, like running on a treadmill.

- Mobile devices. As mentioned before, ECG can be tracked just by a smart watch. This can be connected to an app and the user can see their HRV and possible advice.

One example of these devices is shown on Figure 2, where we can see a sensor used to record ECG. This device is called Polar band H10 and consists of a band that is worn directly on the chest, and it sends the recorded data via bluetooth to a mobile device.

1.1.3 Heart Rate Variability

As the number of heartbeats per minute is a good representation of the human physiology, it can be used to understand what is happening in the human body in a certain situation. Or to predict how a person is feeling at a particular moment. This kind of study is called Heart Rate Variability (HRV). It consists on analysing the variation of the heart rate. For this, R peaks in the ECG are detected and the distance between them, called RR interval, is measured, as shown in Figure 3. So, in the end the RR interval values are measured for the entire signal, and the HRV can be analysed.

1.2 Heart Rate Variability, stress and medical error

1.2.1 Applications of Heart Rate Variability

ECG analysis is a versatile tool in cardiology. It is used to see how the heart is working and possible problems. This can be useful to detect arrhythmias, heart attacks, or for preoperative screening, among other applications. But there are other applications of ECG that just consider the HRV, which are the ones we focus on for this project. This can be useful in a clinical background, where a cardiologist checks the signal looking for possible issues or malfunctioning. For example, HRV analysis can be used as a non-invasive tool to assess the severity of COPD (chronic obstructive pulmonary disease) and predict the risk of exacerbations [47]. But it can also be useful applied for research purposes or to recommend lifestyle improvements for the person. As an example, we find that that HRV analysis can be used to assess the effects of mindfulness meditation on the autonomic nervous system and that it can be a useful tool for evaluating the efficacy of mindfulness-based interventions [23].
One big advantage of using ECG is that it is easier to measure than other signals. Cardiac images have more detail and are the best approach when detail is needed [18]. But ECG can be used in a wider range of applications. It is even possible to use it remotely, for example with smart watches or other portable devices.
Some important clinical applications of HRV analysis are the following ones:

- Cardiovascular risk assessment. Finding patterns on the HRV can be useful to evaluate the risk of a cardiac problem. HRV markers are associated with cardiovascular clinical profile, as they are strong independent predictors of survival in heart failure [2].

  For example, [44] did a classifier for the prognosis of cardiovascular risk. They used statistical tools and neural networks with HRV features to make the classifier.

- Stress and mental health assessment. Lower HRV can be related to higher levels of anxiety and depression, like explained earlier.

  As an example, [15] compared HRV for individuals with and without depression. They also checked the influence of antidepressant medication on HRV results, finding a correlation with the change of severity in depression.

There are also non clinical HRV applications:

- Sports performance and training. It can be used by coaches or athletes to evaluate levels of recovery and rest. This can be useful to optimise the training and avoid injuries [41].

- Stress management and wellness programs. It can be useful to improve rest by copying strategies and relaxation techniques. HRV analysis is one of the most popular biofeedback techniques used for stress management [61].

- Workplace health and productivity. It can be useful to identify potential workplace stressors and improve productivity [37].

- Neuromarketing research. Seeing how people react to different designs or product can help companies decide what option they should pursue [48].

1.2.2 Heart Rate Variability and stress

An important application of HRV is stress assessment and management. To understand why HRV is useful to assess it, we need to consider that the heart rate is controlled in the body by the Autonomous Nervous System (ANS), which controls in general the involuntary actions of the body. The ANS also consists of two branches: the Sympathetic Nervous System (SNS) and Parasympathetic Nervous System (PNS). The SNS is the one that activates the higher heart rate. On times of stress, the body releases more adrenaline and cortisol to prepare for action. This increases the heart rate, blood pressure or respiratory rate, which help muscles to be more prepared for the imminent action [25].
The PNS, on the other hand, is activated during periods of rest. The body releases acetylcholine to prepare itself for periods of rest or to conserve energy. This makes the heart rate, blood pressure and respiratory rate decrease and enhances other parts of the body like the digestive system.
We can see that it is very easy to distinguish between SNS and PNS activation. In general, SNS means more activation in the circulatory system, which can be seen with higher heart rate, but also with changes in other metrics that we will see later. This means that heart rate and other HRV metrics can be good indicators of the stress that a person has at a certain moment.

1.2.3 Stress and Medical error

Medical error was seen as something rare and very unlikely few decades ago. However, during the last decades it has been recognised as an important cause of death [8]. In fact, it is already recognised as one of the leading death causes in many Western countries. A good way of avoiding this is identifying why and where it happens and taking actions to avoid it. Many variables can affect the performance of the doctors, for example social stressors from daily life or time pressure. This can lead to different errors, for instance to medication errors [49].
One cause of human error is stress. There is research that identifies stress as one of the main reasons for people working worse [54]. And the medical field is no exception. Like mentioned before, one of the applications of HRV analysis is finding stressors in the workplace, and the same thing can be done in the medical field.
Medical professional work in a stressful environment. The risk of the actions they need to take and the busy environments where they work contribute to the workplace being a big stressor. The relation between medical error and stress has also been explored in the surgical environment. The surgical performance is highly affected by the stress, which can be measured from the ECG [14].
Being able to identify stressors can be useful to avoid it in the future. This can be done with two steps [19]:

- Error Reporting Systems, that Encouraging the reporting of errors without punitive measures can help in understanding the causes of errors, including the role of stress. This leads to better strategies to prevent future errors.
- Stress Management Programs: Implementing programs to manage stress among healthcare workers, such as counseling, workload management, and training on coping mechanisms, can reduce the incidence of medical errors.

The current project is related to a study done at the University Medical Center Groningen (UMCG). This study records ECG in surgeons during a surgery or during a full week to measure their stress levels. This project consists on making a pipeline in which HRV analysis can be done. Although the context of this project is the stress in surgeons, the HRV analysis is similar for any of its applications. This pipeline does not necessarily focus on stress, but in HRV analysis in general.

1.3 Signal analysis

HRV analysis is done from specific variables that are measured from the RR intervals. But before being able to measure them, certain processing steps need to be done on the ECG signal. The first step in the analysis is the beat detection. This is normally done by detecting the R-peaks of the signal, which is the highest peak in the beat. On some signals, the R peak is easily identifiable and the beat detection can be accurate. But the recorded ECG can include interference and noise from other signals, so it is necessary to remove as much of this noise as possible. For this reason, a good first preprocessing step is performing a band-pass filter to the signal. This removes the slow stationary components in the signal as well as the small high frequencies.

Beats are detected more accurately after the band-pass filtering. However, the beat detection still never gets a perfect performance. Depending on the method, the results can be highly reliable or almost perfect, but there will always be some chance of error, as some noise components will still be present. And, in HRV analysis, just a difference in a single beat can mean that the results are very different [21]. So, it is always necessary to do some fine-tuning and double checking the beat detection.

This is done through outliers detection and correction process. An outlier is any inter beat interval that is different from the others and can affect the accuracy of the results. This can happen when the beat detection algorithm fails in the detection, which can mean having an extra beat or missing an existing beat. But an artifact can also happen when the inter beat interval is different because of natural reasons. In this case, the ECG is properly recorded and the R peak is properly annotated, but a correction needs to be done anyway. The reason is that this single beat just reflects a specific change and not the general trend of the HRV, so it can affect the results. Some of this natural artifacts are ectopic beats or extrasystoles [40].

An ectopic beat refers to a heartbeat that originates from an abnormal part of the heart, outside of the regular, rhythmic pacemaking system. Normally, the heart's rhythm is governed by the sinoatrial (SA) node, but in the case of an ectopic beat, other cells in the heart spontaneously generate an electrical impulse, causing a premature heartbeat. These ectopic beats can originate from the atria (atrial ectopic beats) or the ventricles (ventricular ectopic beats). They are often felt as a "skipping" in the heartbeat or a palpitation.

An extrasystole is a type of ectopic beat. Specifically, it refers to an extra, premature heartbeat that interrupts the regular rhythm of the heart. This premature beat is followed by a pause as the heart's electrical system resets itself. The next normal heartbeat then often feels stronger, as the heart has had more time to fill with blood during the pause. Extrasystoles can occur in healthy hearts and may not signify a serious condition, although they can be more common or problematic in various cardiac diseases.

Outliers detection and correction can be done manually or automatically. The problem with doing the outliers detection automatically is that it is not still reliable enough, so what is detected as an outlier is not always the case and the correction might be wrong. For this reason, it is recommended to do a manual correction of outliers. But the downside of doing this is that the user has to check all the signal, a process that takes a long time.

Outliers are not the only thing to correct on the ECG signal. It is also common that some segments of the signal are not very reliable, due to different noise sources. These segments are highly problematic, as they barely reflect the ECG signal. The best approach is to consider them as noise and omit them during the later analysis. Again, this can be done manually or automatically with the same advantages and disadvantages as when correcting outliers. The automatic process is faster, but the manual process is more accurate.

Once the signal has been corrected, and noise and artifacts are omitted or corrected, the metrics can be calculated. Metrics are statistical values that reflect the HRV trends of the signal. The goal is to compare HRV trends through the signal, so the signal needs to be split in samples and the results are measured for each of these samples individually. Samples are usually a few minutes long, in order to include a sufficient number of beats to measure HRV metrics. Metrics are calculated for each of these segments and a comparison between them is done with different statistical methods.
Metrics will be explained in detail later in the project. But to have an initial overview, we can mention some of the most popular methods:

- Time domain metrics. These are the metrics directly measured from the inter beat intervals.
- Frequency domain metrics. Frequencies from the inter beat interval measures are detected, split and the power density of each segment is reported.
- Nonlinear metrics. There are also non linear mechanisms on HRV.

1.4 User-centered design

Interaction design is the idea of designing a software or machine considering the user preferences [42]. A requirements analysis is done to understand the user requirements and be able to design the technology accordingly. In this first phase of the design, the designer talks to potential users who can tell how they struggle with the current design and what they would like it to be improved. Also, the designer imagines future users or scenarios and how the design should be so there are as few problems as possible.
Also, an interface should be user-friendly and easy to use. The problem with some programs used in the medical field is that they are very unfriendly to use. They can be too technical or have a poor design that makes them inconvenient. There are some factors that can be considered to improve, and design keys that can be included on the prototype so it is more convenient for the target user.
Another issue to consider is the time efficiency. In signal analysis in general and ECG in particular, it is impossible to get a perfect accuracy of automatic algorithms, as explained before for the outliers and noise correction. It is always more reliable to do the processing manually. But this takes time. The idea of this project is to mix automatic and manual correction. First, the signal will be processed automatically, detecting noise and outliers. After the automatic correction, the more problematic segments will be detected and shown to the user for confirmation.
But it is also hard to decide what automatic algorithm to use. A good one can get a high accuracy but the user will need to wait for it to finish. In this project, we also want to consider this conflict. The selected algorithms will get a good accuracy but not necessarily the best one.
Once the requirements have been specified and the goal of the project is clear, a process called iterative design starts. The iterative design process is a methodology used in various fields, including engineering, software development, and product design, characterized by repeated cycles of design, prototyping, testing, and refinement. This approach allows for continuous improvement and adaptation of a product or system based on feedback and performance in real-world scenarios. Here's a brief overview:

- Initial Design and Prototyping: The process begins with an initial design based on the project requirements and objectives. This initial design is then turned into a prototype, which can range from a simple conceptual model to a more functional representation of the final product.

- Testing and Analysis: The prototype is subjected to various tests to evaluate its performance, usability, functionality, and adherence to the intended design specifications. This testing can involve both technical assessments and user testing, where feedback from potential users is gathered.
- Feedback and Refinement: The results from the testing phase are analyzed, and feedback is incorporated into the design. This often leads to identifying areas for improvement, modifications, or sometimes even a complete redesign of certain aspects of the product or system.
- Iteration: The design is revised based on the feedback and analysis, and a new prototype is developed. This cycle of design, prototype, test, and refine is repeated – each iteration aiming to improve upon the previous version.
- Finalization: Once the design meets the necessary criteria and stakeholder satisfaction, the iterative process is concluded. The final design is then ready for production, implementation, or deployment.

The key advantages of the iterative design process include greater flexibility, the ability to adapt to changing requirements, improved end-user satisfaction, and often a more effective and efficient final product. It offers flexibility for changes based on feedback or evolving requirements.

1.5 Objectives and antecedents

1.5.1 Current ECG analysis pipeline

To understand the problems in the current ECG analysis pipeline and design new programs for the task, we take the pipeline used by researchers at the UMCG (University Medical Center Groningen) as an example. Here, students or researchers need to analyse ECG with different purposes. For that, they use a variety of software that are not very convenient because of different reasons. These programs are old, use different file formats that they need to keep switching and are not very user friendly. They also have the problem that the programs do not have much AI integrated and too much user effort is needed for the ECG processing, making them overly time-consuming for large amounts of data.

- The first of these programs is PreCAR. In this program, they can see the ECG signal and process it. They see graphs corresponding to the signal and some other variables of interest like the Inter Beat Interval (IBI). The steps a user does when using PreCAR are:
  - Doing an automatic R-peak detection by choosing a threshold and delay.
  - Checking each heartbeat to look for possible errors in the R-peak detection. It is hard to get a perfect R-peak detection with complex algorithms, but the correction this program uses is quite simple, as it just considers a threshold and delay. This leads to many errors in the detection. These errors can be a peak that is not detected or a point that is detected as a peak when it really is not. And they can happen for the signal not being too reliable, physiological effects or simply because there is always an error range for this detection. Also, looking at each beat can take a lot of time, so it might also be useful to check the IBI, since having an outlier in these values is a good indicator of a wrong R-peak detection.
  - Artifact correction: when an error is detected, the user can add a missing R-peak or delete a wrongly detected one.
  - Choosing a sample interval to get results from it.

- The next program is Carspan. For this program it is necessary to have the signal with annotated R-peaks and the steps are the following ones:

  - Choosing frequency bands.
  - Pre-processing: process time series.
  - Analysis: calculate spectral function.
  - All these previous steps lead to some results, which can be downloaded to use in later analysis.

- Later analysis. This step involves using the file got in the previous program and doing a statistical analysis. This analysis depends on the purpose of the study and can be done with programs like R or SPSS.

As mentioned, the current project where ECG is processed at the UMCG has the goal of analysing stress in medical professionals. In this case, it is done for surgeons in particular. For that, these professionals are asked to wear an ECG recorder for a full week. This will results in a long signal, that users will later have to analyse looking at each heartbeat. At the end, users will have to check around 800000 heartbeats for each participant, which is too much to do manually. It would take a long time and it is a repetitive process and a strenuous repetitive task.

The other option they have is using a more modern program called Kubios [55]. This program is more visually appealing and has the advantage of letting the user do the automatic artifact correction. It includes a free version, which is quite limited and does not even include the ECG import option. This means that the user can just see and correct the RR intervals, which potentially makes the analysis less accurate, as the user does not really know when the correction is right.

With the premium version, the ECG can be imported. This version lets the user import data from many files formats and the interface is more friendly than the previous ones. We can see the example on Figure 4. On the left, we have all the analysis options. The graphs are on the right, with some options like choosing regions or editing peaks. And on the bottom we see the metrics. Kubios also includes some more uncommon metrics, like ECG waveform analysis or stress/recovery index, which can be useful for sports. But the user freedom is still very limited. This is because the control of the graph is inconvenient, as scrolling is not straight forward and the ECG and RR graphs cannot be aligned. Also, it is not possible to annotate the automatic corrections to review them.

Figure 4: Kubios main interface

Overall, the current pipeline has two options and there is a conflict between time efficiency and user freedom when deciding which one to use. The idea of this research is to develop an interface to analyse ECG that includes the advantages of the 2 methods explained before: the former one, that gives the user freedom to analyse the signal and decide what each outlier is. And Kubios, that has a proper automatic ECG correction, so the user does not need to spend too long with it.
The interface should include an ECG visualizer, maybe also a heartbeat visualizer together with the ECG. On the left there should be options to choose the correction threshold and sliding windows. During the requirements analysis, this will be researched more in detail.

1.5.2 Goal of the project

Using automatic methods in ECG to remove artifacts and to detect R-peaks has been implemented with different approaches. But this never gets a perfect accuracy, so it is not as reliable as having a human checking the signal. For this reason, using a combination of automatic and manual correction of the signal can improve the processing without making humans spend too long working with it.
The first research question is what the requirements for this application would be. There is a real scenario in which users do not have the proper program to work and they have to struggle with old programs and file converters. We want to know what users would need to make this pipeline easier. Also, the design of the interface is another research question. There are many steps in the pipeline

and knowing how to display all the information is important, so the users can work with the software having as few problems or doubts as possible. We can summarise the goals of this research in:

- Understanding what problems users have with the current HRV analysis programs. This involves understanding what is needed to combine user freedom and time efficiency. But also to see what other requirements the users need.
- Developing a program in which all automatic analysis options are available and the user can modify them. The question here is whether it is possible to combine an automatic algorithm with human fine-tuning that corrects possible errors of the algorithm.

Overall, we want to do a user-friendly program in which the user can easily analyse HRV. This program should let the user manually edit the signal but it should also have some automatic analysis tools to make the process faster. We name it EZCardio.
The first step to build the program is to do the requirements analysis. After that, the program will be developed and evaluated. But before all this, we do a literature review to know what options we can choose for the processing steps: beat detection and outliers and noise correction. And we will also see what other software exists to analyse HRV.

2 Background Literature

In this section, we delve into the fundamental aspects of Heart Rate Variability (HRV) analysis, a cornerstone of our research. We begin by exploring the technical processes integral to HRV analysis, including beat detection, outlier and noise detection, and signal detrending, as presented in subsection 2.1. Each of these steps is critical for ensuring the accuracy and reliability of HRV measurements, serving as the foundation upon which further analysis is built. We not only outline the various methods employed in each of these processes but also detail the specific techniques we have adopted for our program, providing a rationale for their selection.
Furthermore, we examine the metrics used to interpret HRV data, covering time domain, frequency domain, and nonlinear metrics in subsection 2.2. This exploration is not merely an academic exercise but a practical guide to understanding the diverse ways HRV can be quantified and analyzed, each offering unique insights into autonomic nervous system function and stress levels. Our discussion extends to the existing ECG analysis software in subsection 2.3, highlighting the current landscape of tools available and setting the stage for our contribution to this field.
The extensive coverage of HRV analysis methods and metrics in this section is intentional. It aims to equip the reader with a thorough understanding of the theoretical underpinnings and practical considerations that inform our program's design. By providing a detailed account of the methods chosen and their application within our work, we lay a solid foundation for the subsequent sections of this thesis, where these methodologies are applied to analyze HRV data, particularly in the context of stress in the medical profession.

2.1 HRV analysis

2.1.1 Beat detection

Detecting beats on the ECG can be a complex task because of the complexity of the signal. There are many waves and frequencies, but the signal can also differ much between different people or even between different intervals of the same person.
There are mainly two steps involved on the peak detection [60]. First, it is necessary to clean the signal as much as possible, by removing noise and artifacts. Then, a suppression step is performed. Here, all the waves in the signal except for the R-peaks are removed. So, at the end, only the R-peaks remain and they are easily detected.
Although beat detection may seem straight-forward, it is hard to get an outstanding result with any of the existing algorithms. There are many algorithms that get very high metrics, with accuracy around 98 or 99%, but getting a perfect algorithm for this task seems to be impossible. Many of these algorithms have existed already for decades and, despite their simplicity, they are still the best known methods to do the beat detection. They are:

- Wavelet transform. The ECG signal is decomposed into multiple scales using wavelet functions. This decomposition helps capture details at different scales, making it easier to identify features like R-peaks. [32]

- Empirical Mode Decomposition (EMD). The ECG signal is decomposed into a set of Intrinsic Mode Functions (IMF) using the EMD algorithm. Each IMF represents a specific oscillatory component of the signal, with varying scales and frequencies. IMFs are analyzed to identify peak-like patterns that correspond to R-peaks in the ECG signal. Peak detection algorithms can be applied to each IMF to locate potential R-peaks. [38]

- Hilbert transform. The Hilbert Transform is applied to the ECG signal to create an analytic signal. This analytic signal has two components: the original signal and a phase-shifted signal, which is 90 degrees out of phase with the original. By calculating the magnitude of the analytic signal, an envelope signal is obtained. The envelope represents the magnitude or amplitude of the original ECG waveform. Peaks in the envelope signal correspond to R-peaks in the ECG signal. [31]

- Pan-Tompkins. It tries to find the peaks based on the derivative of the signal. The point where the derivative is 0 is the point in which the signal slope changes sign, so where the peak should be [39].

Of these methods, Pan-Tompkins is the most popular one, as it normally gets the best results [60]. It has the following steps:

1. Filter. The ECG signal is first filtered to remove noise and baseline wander. A bandpass filter is applied, typically within the range of 0.5 to 50 Hz, to retain the frequency components of interest.

2. Derivation. The derivative of the filtered signal is computed to emphasize the steep slopes associated with the QRS complex. This helps in detecting the sharp R-peaks.

3. Squaring function. The squared values of the differentiated signal are computed. This step further enhances the R-peaks, making them prominent.

4. Integration. A moving average is calculated over a short window (usually around 150 ms) to integrate the squared signal. This window moves through the signal, emphasizing the QRS complex, which typically has higher amplitude.

5. Adjusting thresholds. An adaptive threshold is applied to the integrated signal to determine potential QRS complex peaks. The threshold adapts based on the signal's characteristics to handle varying noise levels.

6. Decision. When a peak higher that then threshold is found, it is detected as an R-peak.

There is not a single way of adjusting the thresholds and doing the decision step. There are many implementations of the Pan-Tompkins algorithm that change how they do this. To detect a peak you can consider the amplitude, the slope, shape, distance to the previous peak and many other variables. So, there are many implementations of this algorithm that just differ on the last step. One of these implementations that gets good metrics is the one of PALMS. [51] For each detected peak, it checks the distance to the previous one and the virtual next one to see if it really is a peak.
Another modified version of Pan-Tompkins is Pan-Tompkins++ [17]. This version of the algorithm consists of the main steps that were explained and taking the peaks of the resulting signal in intervals of 231 ms. After that, a threshold is chosen and the peaks that are higher than the threshold are considered R peaks. But there are some extra decisions that improve the performance of this model. For each peak, the distance with the previous detected beat is measured to see if it is likely to have missed a beat or detected an incorrect one.

- If the distance of the current peak with the previous detected R peak is lower than 360 ms or half of the mean of the previous RR intervals. A slope test is done. If the current slope is lower than 60 % of the slope of the previous R peak, the current peak is more likely to be a T wave.

- If the distance of the current peak with the previous detected R peak is higher than 1000 ms or 166 % of the mean of the previous RR intervals. A new threshold is considered based on the current threshold and the amplitude of the previous detected R peaks. If the maximum value in the window between the previous and the current peaks is higher than this new threshold, it is classified as an R peak.

- If the distance of the current peak with the previous detected R peak is higher than 1400 ms. A new threshold is considered as 20 % of the current threshold. If the maximum value in the window between the previous and the current peaks is higher than this new threshold, it is classified as an R peak.

At the end, the thresholds are updated considering different equations depending on whether the peak was an R peak or not. With this implementation, it is less likely to miss beats that have low amplitude or come after high peaks. Also, it is less likely to misidentify T waves as R peaks. A more mathematical explanation can be found on Appendix A.
Apart from traditional methods, during the last years many deep learning methods for r-peak detection have appeared. These methods need big datasets of annotated ECG signals, which now are available. Some deep learning beat detection methods can be found online. Some examples are:

- 1D-CNN (1 dimensional Convolutional Neural Network) [62]

- SONN (Self Organised Neural Network) [12]

- RPNet (another CNN) [59]

- LSTM (Long Short Term Memory) [26]

All these examples get a higher accuracy than the traditional methods, including Pan-Tompkins. This difference is small in clean signals, as traditional methods already get more than 99 % accuracy. But in some noisy signals traditional methods can get a much lower accuracy, sometimes around 80 or 90 %, while deep learning methods still get an accuracy close to 100 %[30]. However, their time efficiency is quite worse than the one of traditional methods. While the fastest deep learning methods can take 100 ms to analyse detect the beats in 20 seconds of ECG [12], traditional methods can just do it in around 10 or 20 ms [17].
When deciding between using deep learning or traditional methods for analyzing ECG signals, it really depends on what you need it for. For example, smartwatches that check your heart rate in real-time work better with deep learning. This is because they can process the data quickly enough for immediate use, and they are pretty good at detecting heartbeats accurately. But, the purpose of our project is a bit different. We're looking at very long signals, and it is important that the analysis is done even more quickly because users are checking the work themselves. In situations where it is important to work fast and you need to easily explain how the process works, traditional methods are usually better.

2.1.2 Outliers detection

Like explained in the previous paragraphs, the beat detection never gets 100 % accuracy, so there will always be extra or missing beats in the detection. But we also mentioned on the introduction that some beats can be problematic although they are not outliers. An example is the ectopic beats, that can be defined as disturbances of the cardiac rhythm frequently related to the electrical conduction system of the heart. Ectopic beats or any other beats that are longer or shorter than the rest also need

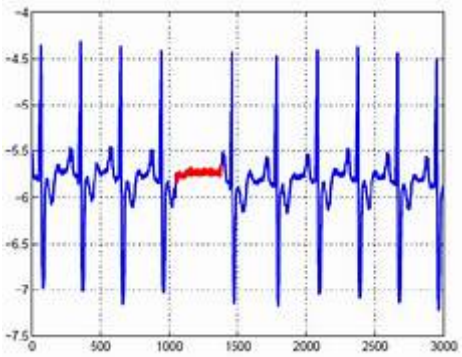


Figure 5: Example of an outlier

to be considered outliers. Despite being natural beats, they can highly alter the metrics, so they need to be corrected.
An example of an outlier can be seen on Figure 5. Whether the reason is a different natural beat or an artifact, the RR corresponding to the longer beat should be interpolated to be similar to the others and not alter the metrics. For these reasons, it is necessary to check all the signal, once the beats have been detected, to find problems and correct them.
There are different methods to detect and correct outliers. The most basic one is establishing a threshold and considering outliers all the values that are farther to the mean than the threshold. These outliers can just be corrected with interpolation. To interpolate an outlier, we change its value for the mean of the 10 surrounding inter beat intervals.
Another method proposed by Kubios [28] is to do follow a conditional logic scheme to identify if an inter beat interval is an outlier or not. Based on different thresholds and moving windows, beats can be identified as an outliers or normal. And, in case it is an outlier, it is possible to specify which kind of outlier it is based on same thresholds and also propose the proper correction. The idea of this method is explained in more detail in the Appendix B.

2.1.3 Noise detection

Apart from the outliers, it is also possible that we find noisy segments on the signal. These are parts of the signal that do not show information of the ECG signal for different reasons. Some examples of noise source are [29]:

- Muscle Artifacts: Muscle contractions, such as those caused by patient movement or shivering, can introduce noise into ECG recordings. These noise artifacts often appear as high-frequency spikes.
- Baseline Drift: Slow, low-frequency variations in the baseline voltage of the ECG signal can occur due to factors like respiration, changes in electrode-skin contact, or body movement. This baseline drift can obscure the ECG waveform.
- Power Line Interference (AC Noise): Electrical interference from power lines (usually 50 or 60 Hz, depending on the region) can be picked up by ECG electrodes, leading to periodic noise in the signal.
- Electromyographic (EMG) Interference: Electromyographic activity from nearby muscles, particularly in limb leads, can contaminate ECG recordings. This is common when the patient is tense or moving.

- Electrode Artifacts: Poor electrode-skin contact or loose electrodes can result in noise. Dry or dirty electrodes may not provide a good electrical connection.
- Respiratory Artifacts: Respiratory movements can cause changes in electrode-skin impedance, resulting in baseline wander or noise in the ECG.
- Patient Motion: Sudden movements or coughing by the patient during ECG recording can introduce noise into the signal.

These segments need to be detected like the outliers. But the correction in this case it to remove these parts from the analysis. The ECG in these intervals is unknown, so omitting the intervals is the only possible solution.
A possible approach is to check how many outliers there are in an interval. This process starts by choosing windows of certain duration. Some basic statistics are measured for each window and the standard deviation of each is measured. This can be the heart rate, for example. The deviation is compared for different intervals. If there is an interval that has a different trend to the others, it can be considered as noise.
The method proposed by Kubios [55] consists of checking how many outliers there are on each interval. If an interval has too many outliers, it is considered noisy. They check the relative time that outlier peaks occupy in the interval compared to the time occupied by non-outlier peaks. If the outliers time is long enough, there is noise.
Another possibility is to use machine learning algorithms with this purpose. Many have appeared in the last years, specially deep learning algorithms. There is no available data with annotated noise on ECG, so each article proposes a different way of getting the data. Most of these methods get the data from a Physionet database of noisy ECG [34] and use normal ECG as non-noise labeled data. This is used for example to train a CNN network [4]. An alternative is to create a new dataset [53].
However, using deep learning for this task, like for beat detection, takes much longer than non-DL methods. And it does not get a high accuracy. In fact, DL methods for noise detection tend to get lower accuracy than non-DL methods. In the best cases, these methods just get an accuracy of 80 % [63] [9]. Non-DL methods do not have a known accuracy because they depend on the threshold that the user specifies. But they always have the advantage of being faster and more explainable, which is again what we want to have in this program.

2.1.4 Signal detrending

In electrocardiogram (ECG) signal processing, stationarity refers to the property of a signal where its statistical properties, such as mean, variance, and autocorrelation, remain constant over time. Stationarity is a common assumption in many signal processing techniques, including ECG analysis, because it simplifies the analysis and allows for the application of various tools and algorithms. However, ECG signals often violate this stationarity assumption due to several factors, and detrending is a preprocessing step used to mitigate these issues.
The low frequency components of the ECG cause nonstationarities in HRV that can affect the metrics. These nonstationarities should be removed so the metrics are more reliable, as well as the comparison between different samples. There are different methods to do this, called detrending methods.
Detrending methods typically involve applying a low-pass filter or polynomial fitting to estimate and remove the baseline drift or low-frequency components from the signal. Common approaches include using moving average filters or polynomial fitting techniques like linear or cubic splines. We can see an example on Figure 6.

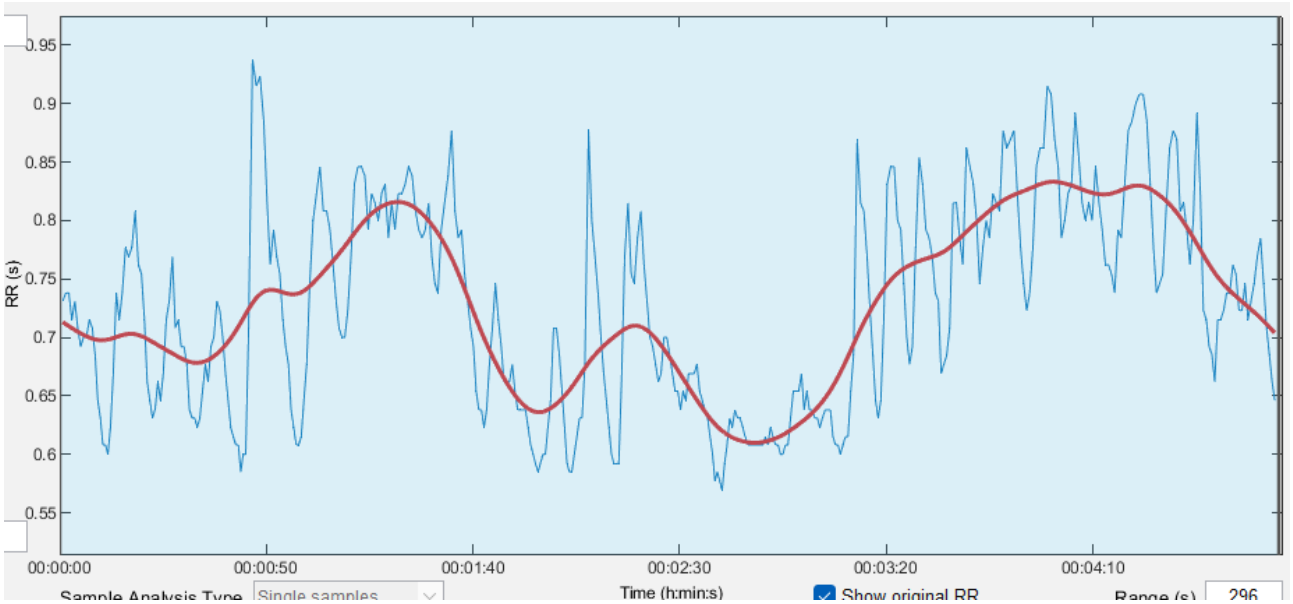


Figure 6: Example of detrending: the red line is the non-stationary component that would be substracted

The most popular method is doing a polynomial model. This is typically of first, second or third order. We get the polynomial component of the signal and substract it, so we just have at the end the stationary values.

Another option is doing a more advanced detrending method. An option is the one presented by Kubios. [56] This method is based on smoothness priors regularisation and is suitable also for more complex trends of the time series. The method works like a time-varying high-pass filter, smoothing the data according to the value of the smoothing parameter. The bigger the smoothing parameter is the lower is the cutoff frequency of the filter.

2.2 HRV metrics

HRV metrics are just different ways to measure and understand heart rate variability. There are mainly three kinds: time domain metrics, frequency domain metrics, and nonlinear metrics.

Time domain metrics are the simplest type. They look at how the time between each heartbeat changes over a period. These metrics are easy to calculate and understand, making them a good starting point for analyzing heart rate data.

Frequency domain metrics are a bit more complex. They break down the heart rate data into different components based on how fast or slow the heart beats. These metrics can tell us more about the balance between the sympathetic and parasympathetic nervous systems.

Lastly, we have nonlinear metrics. These do not fit into the time or frequency categories because they look at the heart rate data in unique ways to find patterns that are not obvious. Nonlinear metrics can be really useful for digging deeper into the heart's behavior, especially in situations where the other two types of metrics might not show the full picture.

In this section we provide a list of the metrics that were included in the program, with brief explanation of each. A more detailed explanation of the metrics, including equations and figures can be found in Appendix C.

2.2.1 Time domain metrics

The time metrics are directly measured from the inter beat interval values. They provide simple, straightforward measures of HRV and are often used for basic HRV assessments. They are [50]:

- Mean RR. The mean of the inter beat interval values.
- Heart rate. The number of beats in a minute. The Heart Rate values are measured each for 5 beats by default, but it can changed by the user in the Settings page. The metrics of minimum HR, maximum HR and the standard deviation are also reported.
- SDNN. The standard deviation of RR intervals.
- RMSSD. The root mean square of successive differences.
- NNXX and pNNXX. NNXX is the number of successive intervals differing more than XX ms. XX is by default 50, so it represents the number of successive intervals differing more than 50 ms. But it can be modified in Settings.

Time domain metrics are used to conduct basic assessments of HRV and get simple, clinically interpretable metrics, such as SDNN and RMSSD. For example, in a clinical setting, SDNN and RMSSD can help evaluate autonomic nervous system function in patients with heart conditions.
Apart from the normal time metrics, for some applications it is also useful to get geometric metrics. These are the values that can be calculated from the RR histogram. To be able to compare, a histogram (for this program we set a fixed bin width 1/128 seconds to allow comparison between different sample) is measured and metrics are derived from it. The geometric metrics are:

- HRV triangular index. [33] It is the integral of the histogram (amount of RR intervals) divided by the height of the histogram.
- TINN [10]. Baseline width of the RR histogram evaluated through triangular interpolation.
- Stress index. Indicative of the stress [5] [24].

2.2.2 Frequency domain metrics

Frequency domain metrics provide information about the distribution of HRV in different frequency bands. They are used to gain insights into the specific autonomic control mechanisms. The spectrum estimates are done for three frequency bands [50]. which are:

- High-Frequency (HF) Component: This component, usually in the range of 0.15 to 0.4 Hz, is primarily associated with parasympathetic (vagal) activity. The HF component is related to respiratory sinus arrhythmia and increases during slow, deep breathing. It reflects the body's ability to modulate heart rate in response to respiration and is an indicator of parasympathetic nervous system activity.
- Low-Frequency (LF) Component: The LF component, ranging from 0.04 to 0.15 Hz, is more complex. It is influenced by both sympathetic and parasympathetic nervous systems. The LF component is thought to reflect baroreceptor activity and is also influenced by various physiological factors, including thermoregulatory, humoral, and hormonal mechanisms.

- Very Low Frequency (VLF) Component: The VLF component, less than 0.04 Hz, is less well understood but is thought to be associated with longer-term regulatory mechanisms, such as thermoregulation, renin-angiotensin system activity, and other hormonal factors.

- LF/HF Ratio: The ratio of LF to HF components is often used as an indicator of the balance between sympathetic and parasympathetic activity. A higher LF/HF ratio suggests increased sympathetic dominance or reduced parasympathetic activity.

For each frequency range, the following parameters are used [50]:

- Peak frequency in Hz for each band.

- Absolute power for each band in ms^2 or log.

- Relative power: absolute power of each band divided by the total power.

- Normalized power: normalised values of low and high frequency bands.

The frequency metrics are measured from the Power Spectrum Density (PSD) estimate. It is a way to show how the strength (power) of the signals (heartbeats in this case) is spread across different frequencies. It also assumes equidistant sampling, so prior to getting it, a cubic spline interpolation to 4 Hz is done.
The PSD spectrum can be estimated by two different methods: the Fast Fourier Transform (FFT) and the parametric autoregressive (AR) method. The FFT breaks down a complex heart rate signal into simpler parts (frequencies) to understand the rhythm better. It is a simple method and ideal for signals that do not change much per time. The AR method, instead of breaking down the signal like FFT, models the signal as if it is being generated by a process that relies on its previous values. It is like predicting the next heartbeat based on the past ones. The AR provides more resolution in the metrics, specially for short samples, and can be more accurate in signals that change much over time [50].
Another option is to get the Lomb-Scargle periodogram. This is different from the Welch periodogram in not assuming equidistant sampling.

2.2.3 Nonlinear metrics

Nonlinear metrics are a way to analyze HRV that goes beyond the simple up and down patterns of heartbeats or their frequency. Although they tend to have a hard physiological interpretation of the metrics, their use in HRV analysis has increased. There are nonlinear mechanisms involved in heart rate regulation. They explore the complexity and underlying dynamics of the heart rate time series. They are used for advanced assessments, especially in research settings. Some methods to get them are Poincaré plot, approximate and sample entropy and detrended fluctuation analysis:

- Poincaré plot [22]. Graphical representation of the correlation between successive RR intervals. The standard deviation of the points perpendicular to the line-of-identity denoted by SD1 describes short-term variability which is mainly caused by RSA.

  SD2 is the standard deviation along the line-of-identity and describes the long-term variability.

- Detrended Fluctuation Analysis (DFA) [50]. It measures the correlation within the signal. The DFA correlations are divided into short-term and long-term fluctuations.

- Sample entropy (SampEn) [27]. SampEn is a measure of the regularity or complexity of a time series. It quantifies the likelihood that similar sequences of data points will remain similar when one or more data points are added. In HRV analysis, SampEn is often used to assess the complexity of RR interval time series (the time between successive R-peaks in an ECG signal). Lower SampEn values indicate higher regularity and predictability in heart rate fluctuations, while higher values suggest more complex and irregular patterns.
- Approximate Entropy (ApEn) [57]. ApEn is a measure of the unpredictability of fluctuations in a time series. It quantifies the likelihood that similar sequences of data points will remain similar within a defined tolerance level. Similar to SampEn, ApEn is used in HRV analysis to assess the complexity and regularity of RR interval time series. Lower ApEn values indicate higher regularity and predictability in heart rate fluctuations, while higher values suggest more complex and irregular patterns.

An application of nonlinear metrics is in advanced research or to explore the complex dynamics of HRV. For example, in stress research, nonlinear measures like ApEn and DFA can help understand the impact of chronic stress on HRV complexity.

2.3 Existing ECG analysis software

In the introduction we mentioned that we want to improve the programs used at the UMCG, which are PreCAR and Carspan or Kubios. But there are some other programs that can be used for HRV analysis. Most of them appeared as research tools from different universities. Some examples are:

- PALMS [11]. This is an open source interface coded in Python and available on GitHub. Here, the user can see an ECG signal and get additional graphs from it, like the inter beat intervals graph. It includes beat detection and signal quality detection algorithms. It is possible to edit and annotate the signal with events, changing beats or adding samples.

  A big advantage of PALMS is that is open source and the code is available online. So, the user can modify the things they want, like the algorithms are some properties of the interface. We can see the main PALMS interface on Figure 7.

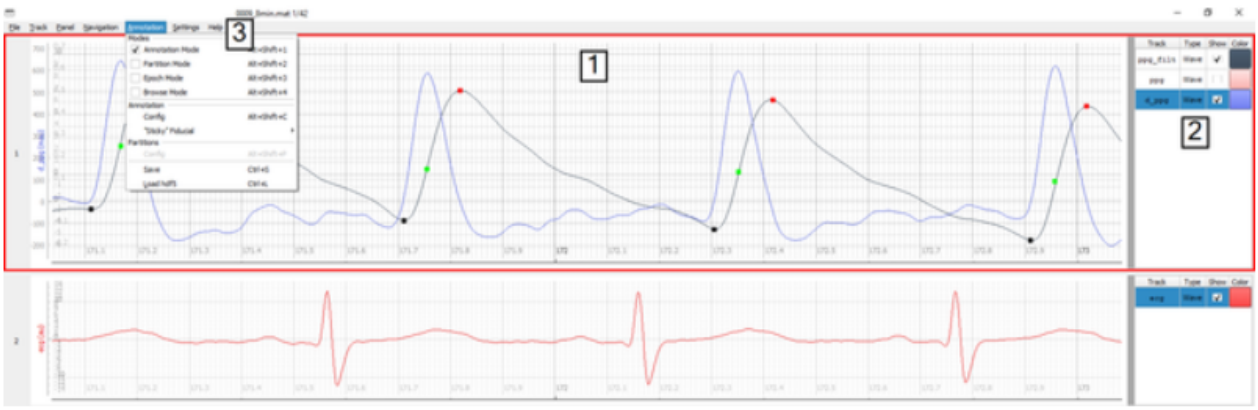


Figure 7: PALMS main interface

However, PALMS is not very convenient for end users, as it does not have an executable version. Also, the automatic analysis options are limited to the beat detection, there is not any other additional algorithm as outliers or noise correction.

- Nevrokard [36]. Expensive software that is widely used. However, it is very user-unfriendly, having complex screens in which it is not clear what each option means. Also, it just works for Windows and the processing options are quite limited. It provides different individual software, each one for different applications:

  – aHRV: advanced Heart Rate Variability analysis.

  The standard version is for humans, but there is also another version for small animals, who are supposed to have much higher heart rates. We can see it on Figure 8.

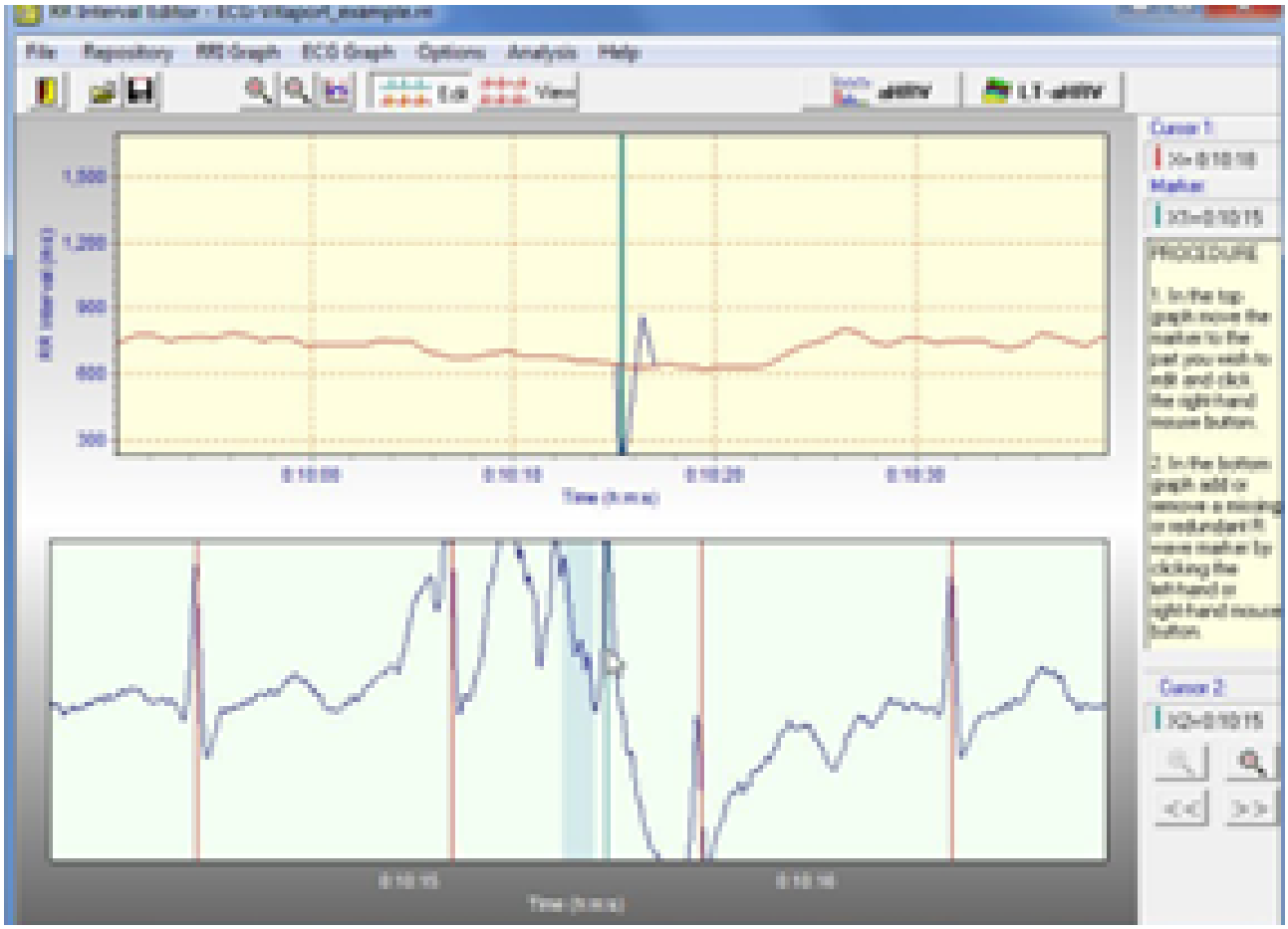


Figure 8: Nevrokard main interface

It can take ECG or RR and it is possible to manually edit beats, but you have to manually go to the beat to edit and the editing option is not very user-friendly.

It is also possible to automatically correct abnormal beats with the selected parameters and correction method (deletion or interpolation). It outputs all kind of metrics.

– LT-aHRV: Long-Term advanced Heart Rate Variability analysis. Takes up to 120 hours. Works as aHRV but with more focus on temporal analysis.

– OSAS. Comparison of HRV parameters in the time and frequency domains during periods of wakefulness and sleep, thus enabling the doctor to quantify the suspicion of obstructive sleep apnea in the patient.

– Other options:

  - BPV. Blood Pressure Variability Analysis monitors.
  - BRS. Baroreflex Sensitivity Analysis.
  - CVPA. Cardiovascular Parameters recorded by non-invasive continuous cardiac output monitor in both time and frequency domains.
  - LDDA. Laser Doppler Data Analysis for evaluation of microvascular perfusion.

- Artiifact [20]: a tool for heart rate artifact processing and heart rate variability analysis.

Figure 9: Artiifact main interface

A single application window where you see all the steps at the time, works for ECG or HR. We can see the main interface on Figure 9. You can do the preprocessing automatically, but you also have the option of manually correcting some things. The processing steps are good, but the software is old and the used techniques are very outdated. Also, the interface is not very user friendly, having too many graphs and text. The analysis options are hard to find. The program has the following functions:

  - ecgExtract: gets IBI from ECG.
  - ibiArtifactProcessing: artifact correction and detection. Based on algorithm by Bernston et al (1990).
  - hrvAnalysis: get the metrics.
  - distributionStatistics: the only metrics are normality and dispersion on IBI distribution.

- Clifford Laboratory [58]. University laboratory that has some open source code for signal analysis done in Matlab. You need to install Matlab and open the code from there. There is a

toolbox for cardiovascular signal analysis. It provides time and frequency metrics for ECG and pulsatile waveforms, but also metrics like acceleration or deceleration capacity and pulse transit time. As signal processing methods, they say they include state of the art peak detectors, signal quality processing units and beat/rhythm phenotyping. Preprocessing methods are really quite simple and automatic, although there is the option of specifying the parameters they will use.

With this toolbox, you can just import a data file that has RR in the first column and time in the second one. You can also add annotator files. Again, this program is not convenient for end users for all the reasons mentioned.

- AcqKnowledge [3]. Automated ECG analysis that identifies all the points in the ECG complex. It calculates the intervals between waves and all the details in the signal, not just the HR options. So, also P-Height, PRQ interval and similar things. It can get data from multi-lead ECG recordings. It includes a lot of options for analysis, but maybe too much. And the interface is not very friendly.

- Cardiolund [1]. Software to purely automatically analyse the ECG, which can even be done online. Features:

  - Identifies beats and groups them into beat classes with different morphologies.
  - Analyses the beat sequence in detail providing markers for a wide range of events.
  - Calculates the standard measures such as Heart Rate, RR variability and the corrected QT intervals, as well as R tag and QT interval markers for presentation in a user interface.
  - Performs robust interval measurements based on averaged beats, and categorises the signals or signal segment into one of 12 categories for efficient identification of important cases in large databases or important segments in long recordings.

  Steps in the analysis:

  - Re-sampling to adjust the signal to the internal sampling rate of the algorithm,
  - Pre-processing to remove disturbances that may disturb the beat detector.
  - Stim detection and removal to remove pacemaker spikes that may disturb the beat detector.
  - Beat detection where all individual events in the signal are found and identified in terms of morphology.
  - Rhythm analysis where all individual events are summarised into descriptors of the rhythm pattern.
  - Average beat analysis and interval measurements is where all intervals (ECG measures) are calculated.
  - Postprocessing to evaluate possible problems that have occurred during the analysis.
  - P wave analysis to identify if P waves are present which supports the following categorisation step.
  - Categorisation to summarise all different features of the signal into a decision on which category the signal belongs to.

- gHRV [46]. Imports data from WFDB, ASCII, IBI and Polar (HRM) and Suunto (SDF) monitors. 2 steps in pre-processing:

– Outliers removal. Automatic with adaptive thresholding or manual like PreCAR.

– Frequency domain analysis with a linear interpolation method.

Intervals of physiological interest within the heart beat time series may be annotated making use of tags in so-called episodes. It is possible to do statistical significance tests in time-domain, frequency-domain or non-linear indexes. They use Kolmogorov-Smirnov to compare 2 different samples. It appears on Figure 10.

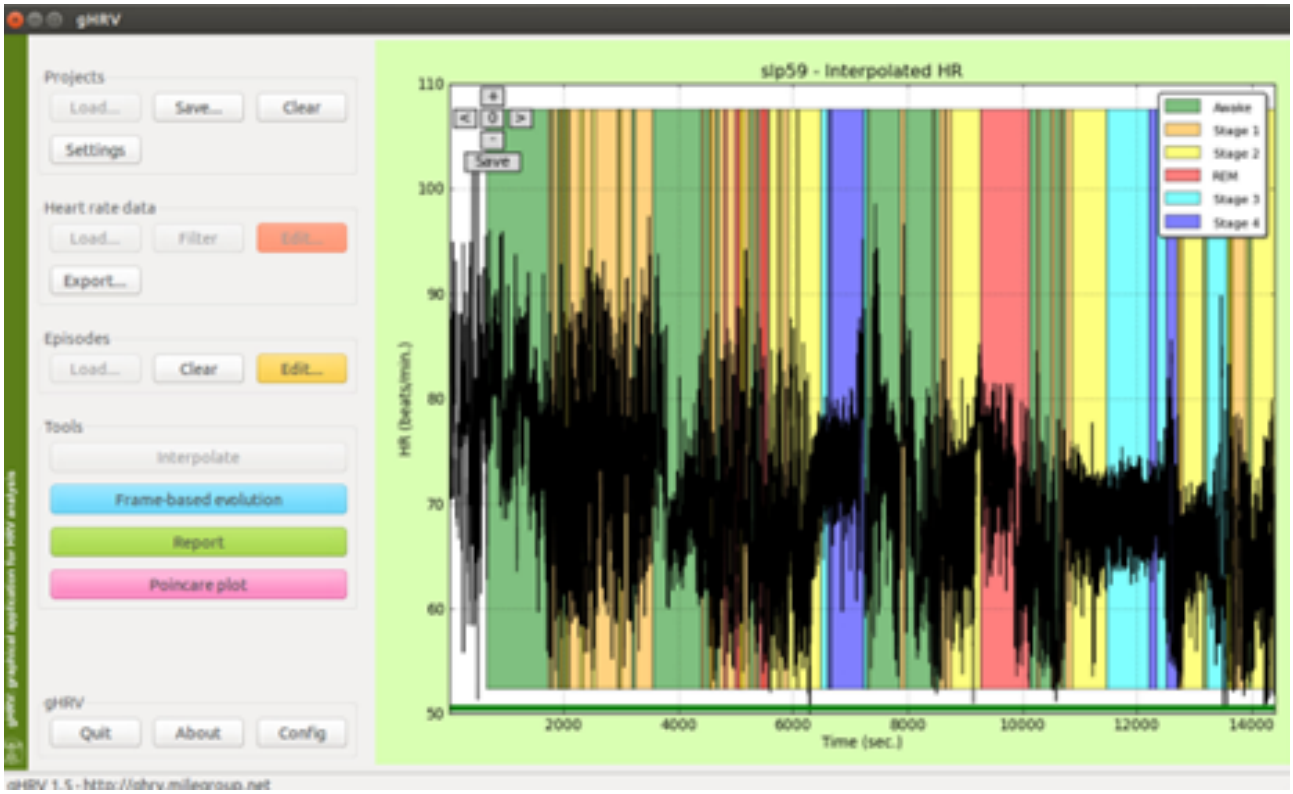


Figure 10: gHRV main interface

- hrVAS [45]. This program just includes simple processing steps and metrics, but the interface is very friendly. The processing steps are easily seen on the left, while the graphs of the metrics are on the right. We can see it on Figure 11.

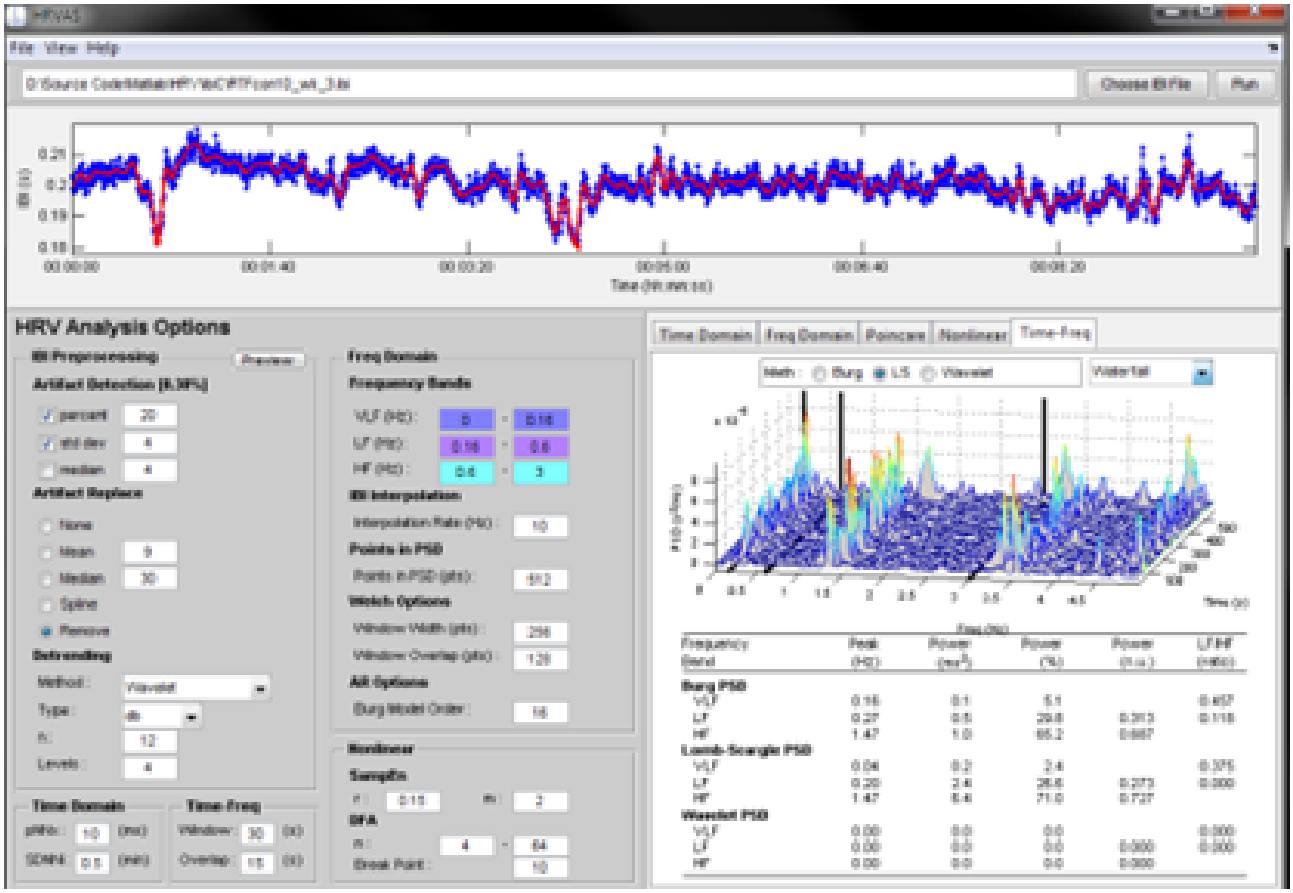


Figure 11: HRVAS main interface

- Physiozoo. Open source software with all options for a proper analysis (not manual+automatic) with a proper interface. You can import ECG or RR for various animals and you see ECG/RR in the main screen with small RR graph of all the signal. Sliding window to choose what interval to see in the main graph.

  A disadvantage is that the data import is limited, just allowing to use hea, mat and txt and their columns need to have headers to be identified. For the peak correction, all peaks are first corrected and they are marked. You have some arrows to go to the next peaks and you can do automatic correction. But you can't change a single peak and you can't see just the outliers. In other words, the manual editing options are non existing. You also can choose some settings for the peak detection and correction, but also for preprocessing to get better HR values (moving average, range, quotient or combined).

To sum up, we can find a lot of programs to analyse ECG and HRV. Although there are so many programs, they are very different. The interfaces are very different, but normally very unfriendly. They have too much information and it is hard to know where to start the analysis or even to know what the functionality of each button is.
Also, the steps used in most of the programs are too simple. Using a threshold to detect an outlier, for example, is quite useless. The signal can differ much even for the same participant, so just setting a threshold is too simple for this problem. It seems that all these programs are quite out-dated and not convenient anymore for the task.
The exception, however, is Kubios. It provides a more friendly interface, although still too busy and

inconvenient in some cases. EZCardio will get some ideas from Kubios, but it will have a different approach to do the overall analysis, trying to give more freedom to the user.

3 Requirements analysis

3.1 Objectives

Before coding EZCardio, it is necessary to understand what the users really need. We have seen that current programs have many problems and are not very convenient for the task. We know that there is a conflict between the user freedom and time efficiency in the existing software. But we need to understand why exactly this is the case and how these two advantages could be combined in a new program. For this, we do a requirements analysis, in which we ask the users about their needs or preferences when working with HRV and try to come up with the best program design for this task.
The approach we follow to do the requirements analysis includes two steps. First, we talk to users about how they do HRV analysis and what problems they find. This helps us get a better understanding of the requirements they have and where more value can be added in comparison to the other programs.
Then, we do a use case analysis. This consists of identifying all the steps that the user needs to do when analysing HRV and how each one should be included in the program. For this step, we consider the requirements from the user interviews, but also the current literature to take into account what techniques are possible to include. This step is important to ensure the program accommodates all necessary steps.
After doing the interviews and use cases analysis, we will conclude this chapter by identifying all the requirements considering its type. This works as a summary of the chapter and the guideline to follow later during the design and prototyping.

3.2 Methods

3.2.1 Interviews

We started with talking to the users. It is necessary to know what the user preferences are, since we need the interface to include everything necessary to improve the pipeline. And also that this is shown in the best way for the users to do the analysis. To analyse user preferences, some users were interviewed and asked about the current problems they have and hear their opinions on possible future solutions.
Normally, this program is used by PhD students and master or bachelor students who are doing a research project. In this case, these two kinds of users have a different view of the program. PhD students have a big project with some research questions. They plan the pipeline of the project, so they decide for example if Kubios or PreCAR should be used depending on the data and what exactly they want to do with the data. As they are aware of different projects or situations in which ECG analysis is required, they have a wide understanding of the pipeline. For master and bachelor students the situation is different. They just do a project and are not so aware of other possible applications of ECG.
With this idea on mind, the questions to ask for PhD or master/bachelor students should be different. First of all, a semi-structured interview is planned for 2 PhD students. The idea is to get a better understanding of the problems that they have. The interviews were open discussions, but some questions were also planned considering the basics of HRV analysis.
In this case, the users (PhD students) are expected to have knowledge about the ECG signal and HRV itself, not just about how to do the analysis. So, the questions also ask about advantages and disadvantages of each alternative they have at the moment and things they miss. The list of questions can be found in Appendix D, but the ideas that were asked are:

- Evaluation of the necessary data inputs for HRV analysis, including the sufficiency of ECG values and the requirement for additional information or signals.
- Determination of the most effective data display and correction features, including whether to show ECG, HR, or RR signals, the utility of a full signal overview alongside detailed segments, and preferences for manual correction interfaces.
- Exploration of potential improvements in user interaction, such as the introduction of automatic sampling for signal analysis and the identification of features needed for comprehensive manual correction of heartbeats.

On the other hand, the questions for master students should be different. It is also possible to do a semi-structured interview in which we get information we need to know but we can also hear about the user experiences and ideas we may not have thought about before. But the questions now should be more focused on the experience rather than also including technical aspects. The users may have some knowledge of the ECG signal for their current research, but not such a wide theoretical knowledge as PhD students. The full list of questions can be found in Appendix D, but the main ideas are:

- Inquiry into users' experience with specific HRV analysis software (Kubios, PreCAR, and Carspan) and their interaction with various data sources (Cortrium, Polar, etc.), to understand their familiarity and preferences.
- Exploration of the types of ECG studies conducted by the users, including the nature of the data analysis performed post-acquisition, to gauge the range of applications and analyses being undertaken.
- Feedback on real-time versus post-processing result visualization in Kubios, including a preference for preemptive noise detection, to optimize the processing workflow.
- Evaluation of PreCAR's efficiency and effectiveness, specifically the balance between automatic correction and manual confirmation, and the criteria for software preference among users.
- User interface design preferences regarding navigation through signal outliers, including the placement of navigation tools in relation to signal analysis options and graphical displays.
- Preferences on data display, specifically the simultaneous viewing of ECG and HR data, the alignment and interval synchronization of these data streams, and the utility of viewing multiple ECG channels concurrently or selecting among them.

3.2.2 Use cases

A use case is a powerful analysis and communication tool that focuses on the division of tasks between user and system, which helps to provide a clear account of each scenario and capture the interactions. In this case, a summary of the full task is done, considering the opinion of the users and the previous literature research. This section can be seen as a summary of the notes taken from the interviews. This is useful to define all the steps that should be included in the new HRV analysis pipeline, considering the user preferences and challenges of each step.

3.3 Results

The results include the outcome of the interviews and the use cases. The full list of all the ideas that were collected and the explanation of each can be found in Appendix E, here just a summary is given.

3.3.1 Outcome of interviews

In general all users said that the current programs they use for HRV analysis have some limitations. They all shared some ideas of how these could be improved. The PhD students responses focused on the following points:

- Flexibility in Data Input and Visualization:

  Users value the option to choose between ECG and HR data upon import, citing the variability in sensor reliability (e.g., Cortrium vs. Polar Band) and the need for multiple data types. Additionally, the ability to select from multiple ECG channels and visualize ECG and HR simultaneously or separately, with customizable scales, is crucial for accommodating different study needs and sensor capabilities.

- Enhanced Data Management Features:

  The necessity of including a datetime column or a time index for signal data to track gaps or deletions accurately. Users also highlighted the importance of integrating multiple files into a single dataset to simplify the analysis of long-duration recordings that may experience intermittent connection issues.

- User Interface and Interaction Improvements:

  Suggestions for a more user-friendly interface focused on easier zooming and scrolling within graphs, indicating that current solutions like Kubios could improve in this area. The ability to navigate between outliers directly within the main analysis interface was also favored.

- Manual Correction and Noise Detection:

  The identification of IBI outliers as a critical feature for indicating the need for manual signal correction, with preferences for a straightforward correction process. Additionally, the potential for pre-analysis noise detection was discussed, especially beneficial for signals with significant noise, allowing users to clean the data before detailed analysis.

- Results Processing and Output Options:

  A desire for flexible results output, including the option to download raw or normalized data in a CSV format, and the ability to see results in real-time or after processing. The idea of automatic sample selection was highlighted as a significant time-saver, allowing for specification of sample duration, repetition, and overlap parameters.

- Software Preference and Functionality:

  The choice between Kubios and PreCAR depends on the reliability of the ECG signal and the user's ability to understand and control outlier detection and correction. Users preferred Kubios for reliable ECG signals due to its outlier management but sought improvements for handling less reliable data, including better noise detection and correction functionalities.

The following ideas were collected from the interviews with master students:

- Diverse Preferences on Real-Time Results and Interface Simplicity:

  Users are divided on viewing results in real-time during processing, with some preferring a focus on signal analysis space. There's a split in preference for software interface complexity, with a desire for both simplified interfaces like PreCAR for ease of use and comprehensive ones like Kubios for detailed analysis.

- Noise Detection and Correction Process:

  There is a consensus on the importance of accurate noise detection with the option for manual verification and correction, highlighting the need for user control in identifying and managing noisy data segments.

- Efficiency and Analysis Time Concerns:

  Users uniformly find PreCAR's analysis time to be lengthy, especially for longer signals, though it's preferred for its accuracy in analyzing short signals. The option to save and continue processing later is highly valued for improving workflow efficiency.

- Visualization and Data Management Preferences:

  The ability to simultaneously view ECG and HR data is widely favored, with users emphasizing the need for flexibility in choosing which data to display. There's a universal agreement on the usefulness of graph alignment and the option to confirm outliers as reviewed, enhancing user experience in data analysis and tracking.

- Software Selection Based on Task Specificity:

  Preferences between Kubios and PreCAR vary based on the specific analysis needs and signal duration, with a general agreement on the value of each tool for different scenarios. Users express a desire for streamlined navigation and outlier management, suggesting a focus on user-friendly design and functionality enhancements.

3.3.2 Use cases outcome

We considered different actions that the user needs to perform to analyse HRV. These involve some requirements that the new program needs to have in order to let the user do these actions in the best possible way. The full explanation of the use cases can be found in Appendix F, but the summary is:

- User can open a file with RR or ECG choosing some options for the import.
- User sees the signal with possibility of zoom or change display (HR, RR).
- User selects different samples automatically.
- User selects a threshold to see IBI outliers highlighted.
- User sees the general ECG signal and chooses an interval to correct.
- User sees an interval and corrects the heartbeat.
- User wants to perform noise detection similarly to outliers detection.
- User wants to get csv of results.
- User wants to save current analysis to continue later.
- User wants to get a pdf of results in plots

3.4 List of requirements

After the interviews and use cases, we have an idea of what the requirements of the program are. We know the user preferences from the interviews, and also all the steps that are necessary to do the HRV analysis. To conclude this chapter, we can have a final list of the requirements grouped by their type. This is useful later when doing the program prototype and during the evaluation.

3.4.1 Functional requirements

All the steps that are involved in the ECG processing need to be included, with all the options that the current pipeline allows plus potential extra functionalities. But we also need to plan the program from what it needs to have to improve the current pipeline. This means considering the current problems the users have, and understanding how they can be improved. We can focus on three topics to determine the main requirements, in order to combine user freedom and accuracy during the analysis.

- Functional requirement 1. Give the user full control of the signal. This can be done with:
  - Providing useful and visible annotations.
  - Possibility to edit any annotation as wanted.
  - Easy zooming and scrolling.
  - Possibility to synchronize or desynchronize graphs.
- Functional requirement 2. Include all the automatic analysis options. This is to avoid a full manual correction and provide the user with more assistance during the processing. We can consider:
  - Noise correction, outliers correction and sample selection.
  - Possibility to choose settings of each analysis step and apply with just a click.
- Functional requirement 3. The user can immediately see each change that was done automatically, and modify it when needed.
  - Possible to annotate the changes done manually and go to each of them on the graph. Ideally, this should be done with a selector that lets the user go from one annotation to the next one.
  - Possible to manually delete or edit any of these changes. This can be done with adding r-peak, deleting r-peak, removing part of the signal or skipping outlier if it is good. This outlier is then marked as corrected. And the same approach with noise intervals, being able to add, delete or edit the length.

3.4.2 Data requirements

This means capturing the type, volatility, size/amount, persistence, accuracy and value of the required data.
The data is one ECG signal at the time. It should be as long as possible, taking hundreds or even thousands of hours. The input signal comes from an ECG reader, which is normally Polar Band or Cortrium. It can also be Hexoskin and ideally it should be possible to import from different sensors. Having these different file format options is defined as Data requirement 1.

The user should be able to choose to import ECG or just RR intervals, which is defined as Data requirement 2. It is also possible that there is more than one column for ECG. This should be selected in the beginning and the interface options change depending on this first selection.
All the possibilities that need to be included in the data import are defined as Data requirements 3. They are:

- Header lines
- Data type. ECG or HR.
- Data column. If ECG, also include option of choosing multiple columns.
- Data units. Voltage or millivoltage for ECG and seconds or milliseconds for HR.
- Column separator. Coma, semicolon, space, or a different character.
- Time index column. If the file includes data time, option to choose it. This can also be ignored.
- Time format and time units.

3.4.3 Environmental requirements

These refer to the context of use and the circumstances in which the product will operate.

- Environmental requirements 1 The interface will be part of an ECG analysis pipeline. It is necessary to output a file that is useful to continue this process. This file must include the results that students will later use in SPSS or R. That is why it is important to let the user select how they want the results to be saved. Also let the user download the results as a pdf or in more personalised ways.
- Environmental requirements 2 There should also be an option for users to check for more information when they do not understand something. An information tool button should be included that explains all the functionalities and what each result means.
- Environmental requirements 3 Considering technical environment, the product should run in any operative system ideally, but at least in Windows, which is the system in which the current pipeline works. The computational power required needs to be low, since students that use the software will not necessarily have very powerful computers. The R-peak detection uses AI and might be more complex, so it is necessary to make sure it is not too computational demanding.
- Environmental requirements 4 Also, it is important to note that this program can be used in any kind of computer. It is specially important to consider the size. It can be used in small screens or big monitors. For this reason, the interface should be able to adapt to any screen size and still look nice. For example, not showing too much empty space on a big screen or too big things on a small screen.

3.4.4 User characteristics

These refer to key attributes of the intended user's group, such as user's abilities and skills, educational background, preferences, personal circumstances, physical or mental disabilities. Also, the user might be an expert, novice, casual user or frequent user.

- User requirements 1 Self explanatory, where each step is easy to understand.
- User requirements 1 Minimize computer procedures. The user should just need one file to import in the beginning and does not need to do any file type change or complex computer task.

3.4.5 Usability goals and user experience goals

These refer to specific measures for the usability goals of the product are agreed upon early in the development process and are used to track progress as development proceeds. The goal of the new interface is to save time while keeping the proper ECG processing by letting the user decide what to do with different heartbeats. Just the minimum things to detect and annotate the heartbeat should be shown. But there should be other options that let the user have their own preferences about how to display things and what things to display.
To understand this in detail, we can refer to some of the usability points that the users asked for during the interviews:

- Usability requirements 1 Possible to hide or show parts of the screen. Specially the results preview. Some users want to see them while doing the analysis, but others prefer not to see them, as they take a big part of the screen. Ideally, the user should choose to open them or hide them until the analysis is done.
- Usability requirements 2 Choose samples at once instead of one by one. In all previous programs, the user needs to specify each sample to add, with the start and end. But they would prefer to be able to add many at the time, by specifying how many or for all the signal.
- Usability requirements 3 Save analysis and option to load it later again. In case the file is long and they do not have time to correct all the outliers at once.
- Usability requirements 4 Option to append results to existing file. Useful when the input file was too long and had to be split. Instead of having different results files that correspond to the same person, it should be possible to directly append the results to the same file.
- Some requirements regarding the import options.
  - Extra import requirements 1 That they can open multiple signals at the same time. This is useful when they record the signal in an environment in which the connection can be lost for a short interval. Instead of opening few minutes signals one by one, they would like to import it as one and consider the time lost between as noise.
  - Extra import requirements 1 Option to split the signal. For signals that are too long and impossible to analyse all at once.

4 Prototype development

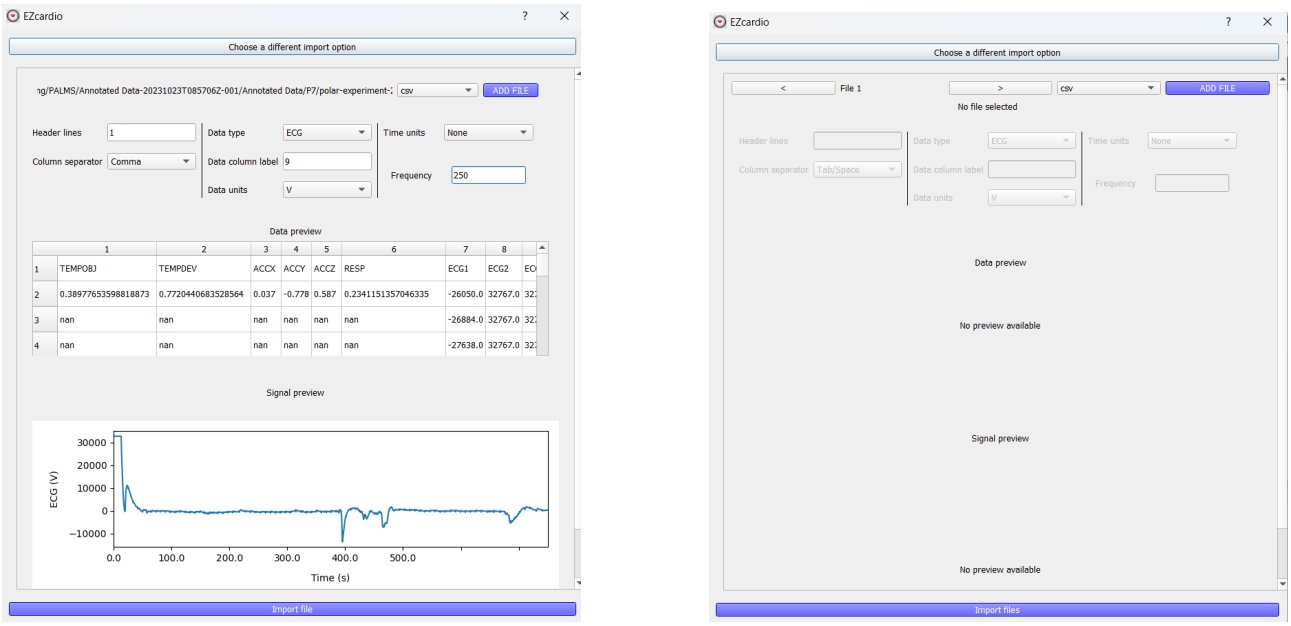
Based on the results of the requirements analysis we developed a working prototype version of EZ-Cardio. To enable using open source libraries we used Python version 3.8. This language was chosen because of being open source, the most popular one in signal processing and having different open source libraries available. This lets us use open source codes for some steps of the analysis and plotting the signal. Also, with python it is possible to make executable files for different operating systems, making it cross-platform.
The prototype is explained in chronological order considering the normal process the user would follow to do the analysis. From the moment in which they open the program to when they export the final results. In each step, the requirements that have been considered are explained. This can also be seen in Table 1, where all the requirements are listed and the section in which they were implemented is also shown.

Table 1: List of Requirements and section in which their implementation is explained

Type	**Number**	**Brief explanation**	**Section**
Functional	1	Give the user full control of the signal	4.3.2
	2	Include all the automatic analysis options	4.3.1 and 4.3.3
	3	The user can see each change just after it is done and modify it	4.5
Data	1	Different file types import	4.1
	2	Import ECG or RR	4.1
	3	Import options that can be specified	4.1
Environmental	1	Metrics export	4.2 and 4.6
	2	Option to see more information	4.2 and 4.3
	3	Multi-platform and computationally efficient	Introduction
	4	Different computer sizes	4.2
User	1	Self-explanatory	4.2
	2	Minimize computer procedures	4.2
Usability	1	Possible to hide or show parts of the screen	4.6.1
	2	Choose many samples at once	4.4
	3	Save analysis and load later	4.2
	4	Append results to existing file	4.2
Extra Import	1	Open multiple signals at the same time	4.1
	2	Split a long signal	4.1

4.1 Data import

In the prototype the first page is the import page, where users can the files they want to analyse. This page needs to meet the data requirements specified before. The user can choose certain file and where the data is on the file, which can refer to ECG or RR (Data requirement 2). We can see this on Figure 12a



(a) Data import for a single file

(b) Data import for multiple files

Figure 12: Data import screens

There are four import options in total, being the default one the option to import a single file. But the extra import requirements are also implemented. The Extra import requirement 1, which is option to include many files on the same import is included. This is useful in cast the file was split because the connection was lost. The user can choose ”Import many files”, shown in Figure 12b. All these files will be loaded and the ECG signal will be shown latter with black spaces in the intervals where the signal was lost.
The opposite option, which is Extra import requirement 2, is also allowed. It can be useful when the signal is too long and can cause problems in the program or take too long to load. In this case, the user can split it in shorter signals. And the last option is to load an existing analysis to continue with it.
As shown in the example, the screens are very similar. To start with the singe file one, we can see four different sections. On top, the user can choose the kind of file and open the directory to choose it. The supported file types were chosen considering what sensors are used at the UMCG and what the most common ECG files are. In order to meet Data requirement 1, we include:

- EDF format. It is one of the most popular file types to work with ECG. And it is also the one that Cortrium monitor uses.
- CSV format. Apart from being a possible ECG output file, it is the most popular data file in general. If someone wants to do some preprocessing on the data and then use the signal on the program, they can easily output a csv file.
- TXT format. This is one of the most popular formats for signal analysis. It is also the type used by Polar band sensors, the ones I used the most during the project.

The second row in the import file corresponds to the import options. To meet Data requirement 1, we include all the options explained in the previous section. The only thing extra is that the user can decide to choose time data or not. If no time column is specified, the user needs to input the frequency of the signal.

The third and fourth rows correspond to the first visualization of the data. The data preview is opened just after the file is selected. The user can see here all the data, which can be useful to specify the data and time columns. The signal preview just appears when the user specifies in which columns the data is.
By clicking on the import button, the data is loaded and the analysis starts with the beat detection algorithm.
The other data import pages are very similar. For the multiple files option, the only different is that we see arrows on top. So, we can change the current file we are seeing.

4.2 General interface

Once importing the data, we find the general interface page. It has 3 main components, like shown in Figure 13. On the left we see the interface icons and the analysis options. On the right we see the graphs and editing options. And on the bottom we see the results. This screen has been designed so all the components are easily recognisable, in order to meet User requirement 1.
We can also change the size of these components by clicking on the resize buttons. These are intended to give more space to the graphs component when the user wants, meeting Usability requirement 1. So, the left one makes the left options disappear, while the bottom one makes the results disappear. By clicking on the same button again, the size goes back to the original one. This also makes the program adaptable to different screen sizes, meeting Environmental requirement 4.

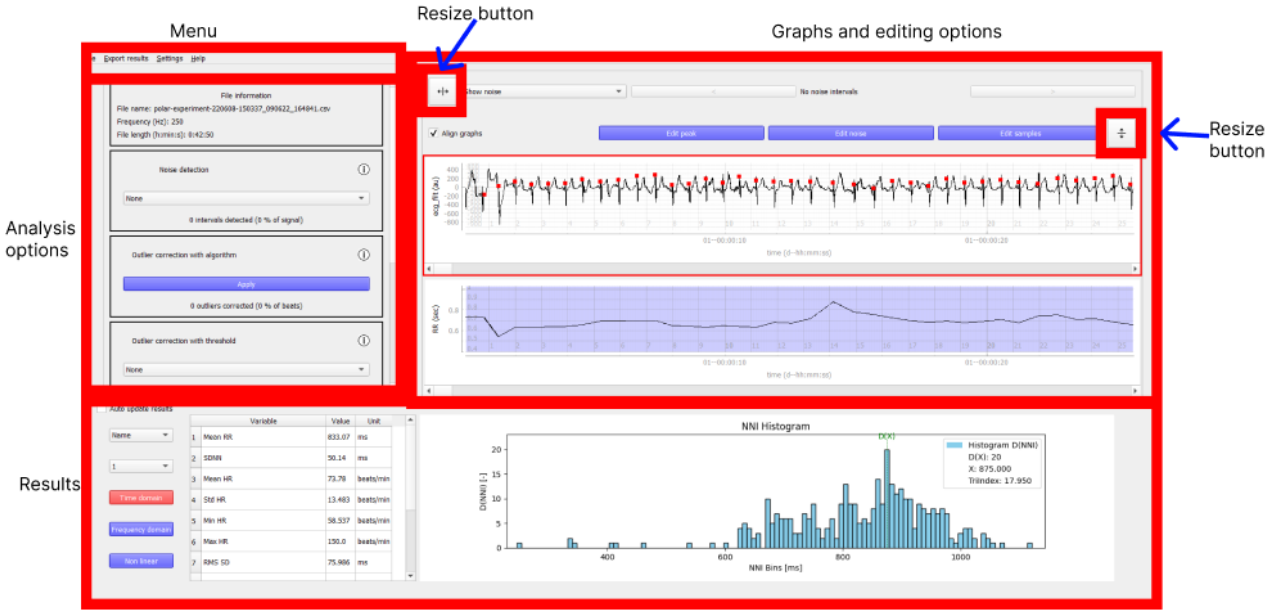


Figure 13: Full view of the interface

On top of the screen we can see a menu with all the general options the program has. From left to right, we see the following actions with different elements:

- File. It has the option ”Open new” to open the main import page. Before opening the import, it will ask the user if they want to save the current analysis. This action also has the options ”Save” and ”Save as” to save the analysis directly. This option to save the analysis and load it again later meets Usability requirement 3.

- Export results. 2 actions: exporting to a new file, which meets Environmental requirement 1 or appending on an existing file, meeting also Usability requirement 4. The directory will be opened and the user can specify the name of the new or existing file.

- Settings. Just one action to open the settings.

- Help. Two possible actions. One to open the manual, where a general explanation and instructions about the program are given. And an option to open a pop-up window where the user can see all the shortcuts. This is one of the ideas we use to meet User requirement 1.

Apart from the import screen, this main screen is the only page that the program has. This also makes the analysis easier, as the user does not have to deal with different programs or sending data from one to another. The full analysis can be done in just this screen, meeting User requirement 2.

4.3 ECG analysis

In order to meet Functional requirement 2, all the HRV analysis options need to be included. Best detection and outliers and noise correction are implemented in the program, like explained in this section

4.3.1 Beat detection

After loading the data, the signal is filtered to remove low and high frequencies. The first processing step is the r-peak detection. There are many ways of doing this, and a lot of them are available online as open source.
At the end, Pan-Tompkins was used as the default beat detection method. The variant is Pan-Tompkins++ [17], which was explained on the previous section. The reason why we use this method is the combination of accuracy and time efficiency.
There are two details that were implemented to make the algorithm work better. The first one is the interpolation. When the time column is known, the frequency can be measured and does not need to be specified by the user. However, the frequency normally has small variations over the signal, which can make any beat detection algorithm fail too many times. One option is to measure the exact frequency of each interval and apply the algorithm for each of them. But this will still have to consider average frequencies for samples and would be very inefficient.
The solution is to do an interpolation of the data points. The function interp1d from scipy allows specifying a time column and a desired frequency. The data points will be changed to match the desired frequency. This will be the frequency of the first 10000 data points. This way, we have an algorithm that works better in an efficient way.
The second detail that was implemented is the up-sampling to 2000 Hz. The main advantage of this step is more precision in the detected beat. Pan-Tompkins can have small variations in where exactly the peak is detected, which can cause small differences in the inter beat intervals. An example can be seen on Figure 14, in which we apply Pan-Tompkins without up-sampling. We can see that the detected peaks are not always in the exact point where they should be. By up-sampling, this is much less likely to happen. In this case, we do not upsample all the signal, since this would be time inefficient. What we do is to detected the beats and get an interval around each of the beats. Each of these intervals is individually upsampled, and the R peak of each interval is the highest point in each interval.

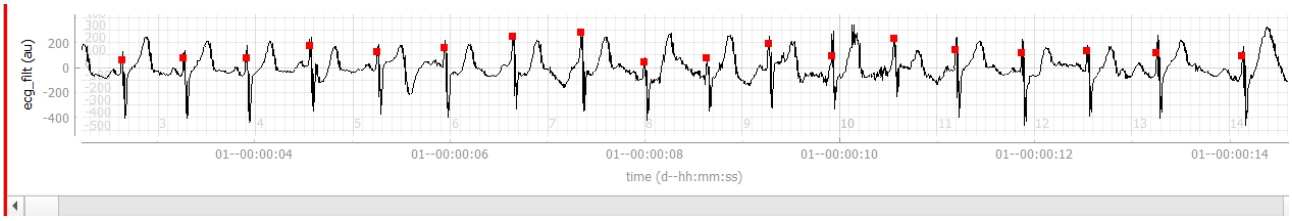


Figure 14: Example of the precision of the beat detection algorithm

4.3.2 Signal visualization

The visualization of the signal is on the top right of the screen. It shows two graphs, one with the ECG signal and the other with the inter beat intervals. Each graph has a range slider to change the current position and can be scrolled in or out with double finger panning or with the mouse wheel. Both graphs also have a grid and axis. The vertical axis shows the ECG voltage or inter beat interval time. The bottom axis is the same for both signals. It shows inside the graph the local time, which is the seconds since the first moment of the signal. But the main axis shows the global time. In order not to show the full datetime, it just shows the day with the time. The graphs are shown on Figure 15

Figure 15: Signals visualization with the graphs and editing options

The ECG graph is the one on top. The detected beats from the implemented Pan Tompkins variant are shown here in red color and the noise in black. The editing options related to peak or noise must be done in this graph. The RR graph is the one on the bottom. It is updated with each change and shows the time of each beat since the last beat. This graph also shows the samples and they can be edited here.
The checkbox to align graphs tells if the two graphs will always show the same interval or not. When clicked, this will happen. And each interval change in one of the graphs (zoom or displacement) will also happen on the other graph. If it is unchecked, the opposite will happen. A change in one of the graphs will just change it, but not the other. This is useful for meeting Functional requirement 1.
To avoid confusion with having different intervals when the checkbox is unchecked, the range sliders are always shown. The sliders always refer to the full signal, so seeing their position and length

represents where the current interval is. But a highlighted region also appears on the RR graph when the current interval is inside the ECG current interval, like shown on Figure 16. It can be seen that the graphs offer a lot of control for the user, meeting Functional requirement 1.

Figure 16: Example of unaligned graphs

4.3.3 Noise and outliers detection

The analysis options are on the top left of the screen, shown on Figure 17. These are the steps the user should follow to correct the signal: noise detection, outliers detection and correction and sample selection. The order of these steps can differ depending on user preferences, but the given one is recommended.

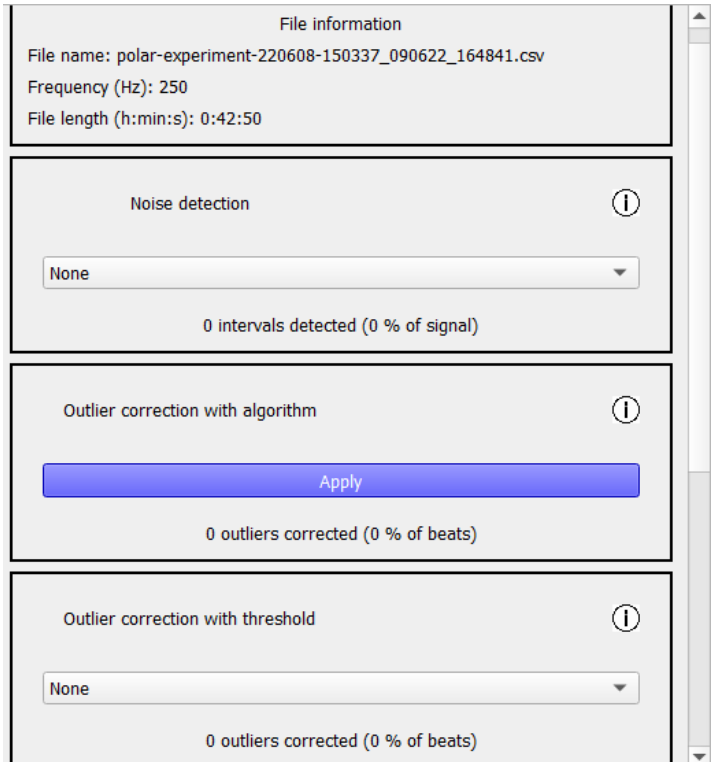


Figure 17: Analysis options

On top, we see some basic information of the signal. This includes the file path, which also shows the name, the frequency and the length.
Next, we have the options of the analysis. First is the noise detection. The basic option to detect noise just looks for parts of the signal with null or very low amplitude. This is done before the beat detection, so the beat detection is not so affected by these noisy intervals.
We can also specify the noise detection level to localise noise intervals with more precision. The noise detection algorithm implemented here looks for all the outliers and checks if there are intervals with too many outliers. So, there are three things that the noise detection considers: the threshold used to get the outliers, how close an outlier has to be to the closest outlier on an interval to be considered from that interval, and the ratio of the interval with outliers to be considered noise. The different levels refer to the values of these two things. The differences in numbers are: [28]

- Very low: the outliers threshold is 0.45 seconds, the maximum distance between outliers is 15 beats and the minimum noise ratio is 0.4.
- Low: the outliers threshold is 0.35 seconds, the maximum distance between outliers is 25 beats and the minimum noise ratio is 0.35.
- Medium: the outliers threshold is 0.25 seconds, the maximum distance between outliers is 35 beats and the minimum noise ratio is 0.3.
- High: the outliers threshold is 0.15 seconds, the maximum distance between outliers is 45 beats and the minimum noise ratio is 0.25.
- Custom: the values can be specified by the user on the settings screen.

.

When marking the noise, the user can see how many intervals were considered noise and what percentage of the signal they represent. The intervals appear as black rectangles on the ECG and RR graphs and they can be edited or deleted. The signal will ignore these noisy segments in the following steps, unless they are removed.

The outlier detection works similarly to the noise detection. Noise detection works for segments where there are many outliers, but single outliers should also be detected and confirmed by the user. In EZCardio we include two different methods for outlier correction. The first one is an automatic correction that, considering different threshold and conditions, is able to identify all the outliers and correct them according to the problem. This method is the one used by Kubios and was already explained in the background section.

The other method is detecting outliers based on a single threshold. The user can choose one of the threshold options. For each RR value, a window of the 10 surrounding beats is considered. If the current RR value is more different to the mean of the window by the threshold, it is an outlier. The correction in this case is always changing this value by the mean of the window. For the threshold detection method, we have the following options [55]:

- Very Low: Threshold of 0.45 seconds between inter beat intervals.
- Low: Threshold of 0.35 seconds between inter beat intervals.
- Medium: Threshold of 0.25 seconds between inter beat intervals.
- Strong: Threshold of 0.15 seconds between inter beat intervals.
- Very strong: Threshold of 0.05 seconds between inter beat intervals.

The algorithm is supposed to give better results, but the threshold method is easier to understand, so we include both and give the user the freedom to choose. It is also possible to use one method after the other. This is useful in case the user wants to use the algorithm but they are afraid it may miss some outliers. With the threshold detection is harder that remaining outliers are missed. In the final evaluation of the program we will also check if this is the case.

Like in the noise detection option, we can also see here a text that says how many outliers there are and what percentage of the total number of beats these are. This could be seen in Figure 17.

When setting any of the three analysis options, we will see that the selector on the top right of the screen becomes active. The idea of this selector is to show the intervals where noise or outliers happen, so the user can quickly go to the interval, check the correction and edit it manually if needed. The options to to the manual editing are explained on the next subsection.

On Figure 18 we can see an example of the selector working. In this case, the algorithm method is being used. All the detected outliers are added to the signal and we can check each of them with the selector. The outliers appear as vertical red lines in the signal. In case we are also using noise correction or outliers correction with threshold, we can change the dropdown button left to the selector to see the corresponding annotated intervals.

Figure 18: Example of outliers detected

4.4 Sample selection

The third block on the analysis options is the sample selection. Samples are normally chosen to get metrics for each sample and compare them. Some users prefer to choose them before the rest of the analysis, while others prefer to do it after it. Both options are possible with this interface. This is shown on Figure 19.

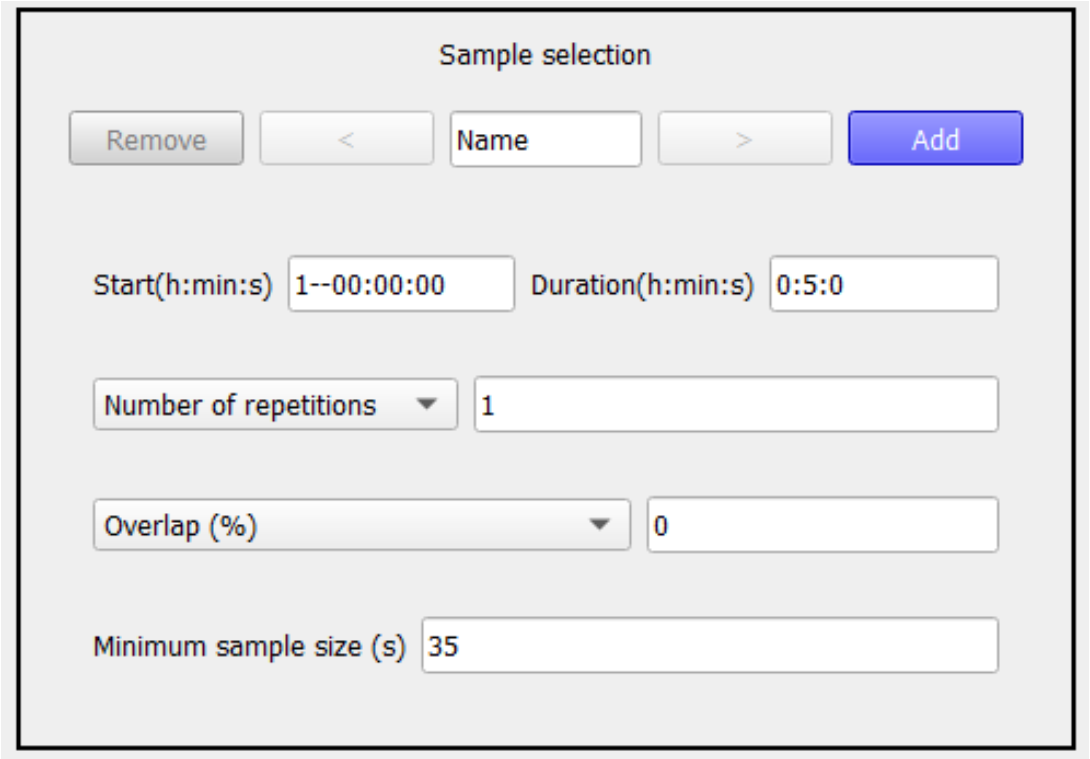


Figure 19: Sample selection options

The first row of this sample selector represents what sample it is. It is possible to have samples and subsamples. We can choose one sample of 5 minutes segments for all the signal, one sample with 3 minutes segments just until certain point and one sample with just one 5 minutes subsample in the middle of the signal. To achieve this, we have another selector in which we specify the options for the current sample and how many subsamples it will have. The option of adding many subsamples at the same time makes the program meet Usability requirement 2.
For the current sample, we can choose the name and the options. The options are the following ones:

- Starting point of the first subsample. Default is the start of the signal.
- Duration of each subsample. Default is 5 minutes.
- Until when the sample will take. This can be how many subsamples there will be or until when subsamples will be added. Default is untill the end of the signal.
- Overlap or space between samples. Default is no overlap.
- Minimum sample size. Samples can have noisy segments as long as the non-noisy segment is higher than the desired threshold. Default is 100 seconds.

When all the options are specified, we can click on the ”ADD” button and the sample will be added. All the subsamples will appear on the graph. To add another sample with different subsamples, we can press on the right arrow and so the same thing: choose sample options and add it again. The subsamples that will be shown on the RR graph are the ones of the current selected sample on the left. In case the current sample has not been added yet, no subsample will be shown on the RR graph.

4.5 ECG editing

The ECG and RR graphs have three possible options. In order to make changes on the signal, the user needs to click on the corresponding editing option. The idea is that the user will use one of the automatic options and then check the results, changing if it necessary, meeting Functional requirement 3. I would also be possible to do changes in the annotation without using any of the automatic options (except for beat detection, which is always done when loading the signal).

4.5.1 Peak editing

There are four possibilities on the peak editing option:

- Add peak. Left clicking on certain point will add a peak. This will be shown in red on the ECG graph and the RR graph will be updated.
- Delete peak. Right clicking on certain red point on the ECG graph will delete it. The RR graph will also be updated.
- Add interpolation. With CTRL left click on certain red point, an interpolation will be added. This will appear on the RR graph.
- Delete interpolation. When CTRL right click on a red point that has an interpolation, it will be deleted an not interpolated anymore when updating the RR graph.

4.5.2 Noise

It is possible to modify segments also directly from the graph. These segments will also appear on the selector when marking segments. So, if a noise segment is directly added on the graph, it will appear when marking noise. There are four possibilities on the noise editing option:

- Adding noise segment. With CTRL left click, a segment appears with center on the point where it was clicked. It has a default length of 5 seconds, but it can be modified later by the user.
- Deleting noise segment. With CTRL right click on an existing noise segment, it disappears.
- Moving noise. With SHIFT left click on certain noise segment area and dragging it, the noise moves to a new point.
- Increasing or decreasing noise size. With SHIFT left click on one of the limits of the noise segment, we can just move the start/end point.

4.5.3 Sample

Sample selection can be done for subsamples, as explained before. When a sample has been created, the user can modify each of the subsamples manually. The possibilities for sample editing are exactly the same ones as for noise editing:

- Adding sample. With CTRL left click, a segment appears with center on the point where it was clicked. It has a default length of 5 seconds, but it can be modified later by the user.
- Deleting sample. With CTRL right click on an existing sample, it disappears.
- Moving sample. With SHIFT left click on certain sample area and dragging it, the sample moves to a new point.
- Increasing or decreasing sample size. With SHIFT left click on one of the limits of the sample, we can just move the start/end point.

4.6 Metrics

On this subsection we explain what HRV metrics, also defined as HRV results, we can see with EZCardio. One Figure 13 we could see the results section on the bottom. There, we can choose the sample and subsample from which we want to get the results, as they are gotten for individual segments. Then, we click on the button of the type of results we want to see. These can be time, frequency or nonlinear results.
We also have the option to update the results. There is a checkbox to specify that the results update will happen automatically. So, with each change on the signal, the current selected results will be computed and shown again. In case this option is disabled, the results can be updated by clicking on the desired option. All this options are shown on Figure 20.

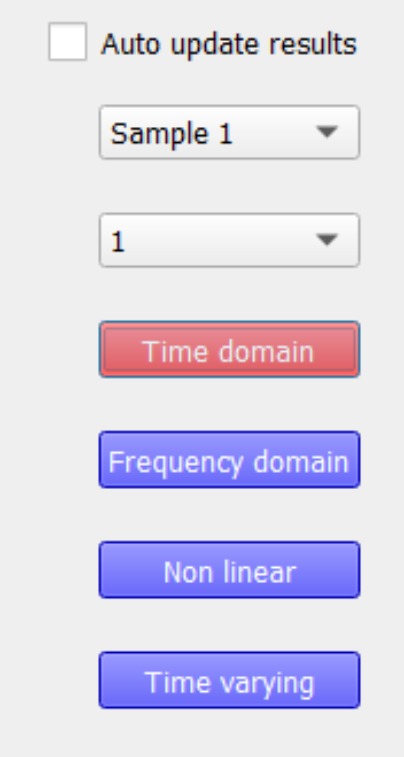


Figure 20: Results buttons

To avoid stationarities that can alter the results, the program automatically does detrending. This is getting the repeated trend over the signal and removing it from the main signal. This way, the statistical results are more reliable.

4.6.1 Time analysis

On the time results option, three different kind of results are reported (Figure 21). There is a table on the left with the numerical results, that are time and geometric results. Time results are the ones directly computed from the detrended values of the inter beat interval values. The exception is mean RR and HR values, that are directly computed from the RR values (with no detrending). They are:

- Mean RR. They are computed from the non-detrended RR values.
- SDNN. Standar deviation of the RR values.
- HR values: mean, maximum, minimum and standar deviation. They are computed from the the average every certain number of beats and the non-detrended RR values.
- RMSSD
- NNXX
- pNNXX

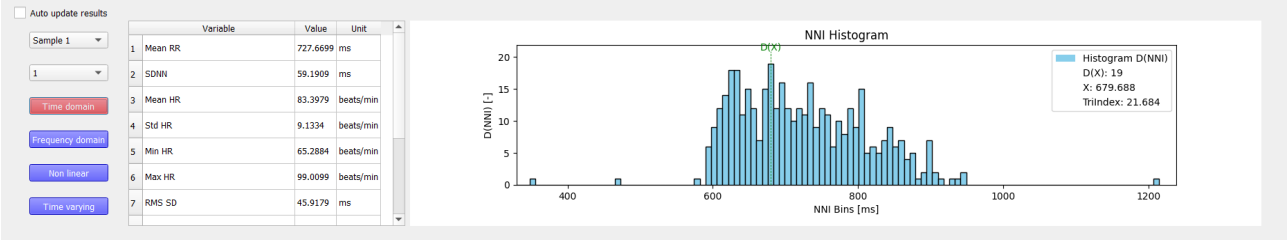


Figure 21: Time results

The geometric results are computed from the histogram distribution of inter beat intervals. The histogram has segments of 50 ms. They are:

- HRV triangular index
- TINN
- Stress index

On the right part of the time results, the histogram is also reported. It has segments of 50 ms.

4.6.2 Frequency analysis: welch, autoregressive, lomb

The frequency results are shown in the same way independently of the method used. On the screen, we show the results based on the Welch or the Autoregressive method. In the settings we can specify that we want to use the Lomb-Scargle method instead of the Welch one. They are shown on Figure 22.

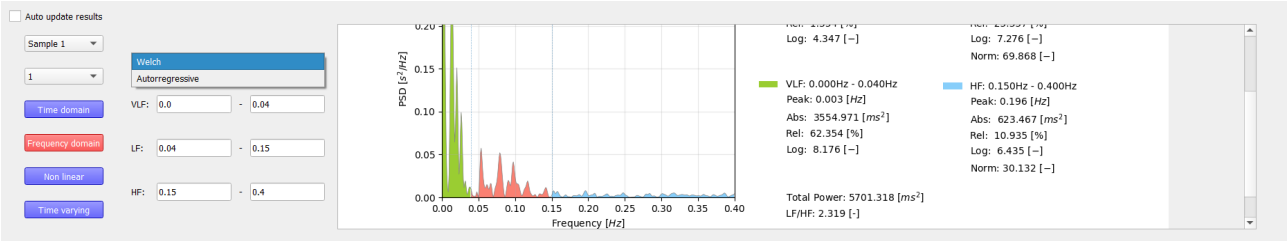


Figure 22: Frequency results

We can see a graph with the frequencies distribution. These frequencies can be modified by the user. By default, they are:

- Very low frequency: from 0 to 0.04 Hz.
- Low frequency: from 0.04 to 0.15 Hz.
- High frequency: from 0.15 to 0.4 Hz.

The results that are reported for each frequencies interval are:

- Peak value: point with highest frequency. In Hz.
- Power: total density. It is reported in ms2, log, percentage, nu and a ratio between high and low frequency.

4.6.3 Nonlinear analysis

User can also select non linear metrics, like shown in Figure 23, where we can see two different graphs, one corresponding to the Poincare plot and the other to the Detrended Fluctuation Analysis. These graphs include the following metrics:

- Poincare SD1
- Poincare SD2
- Ratio SD2/SD1
- Approximate entropy
- Sample entropy
- Detrended Fluctuation Analysis: alpha 1
- Detrended Fluctuation Anlysis: alpha 2

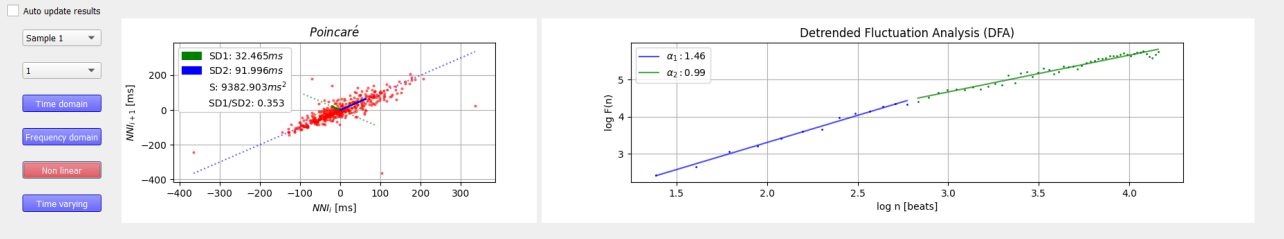


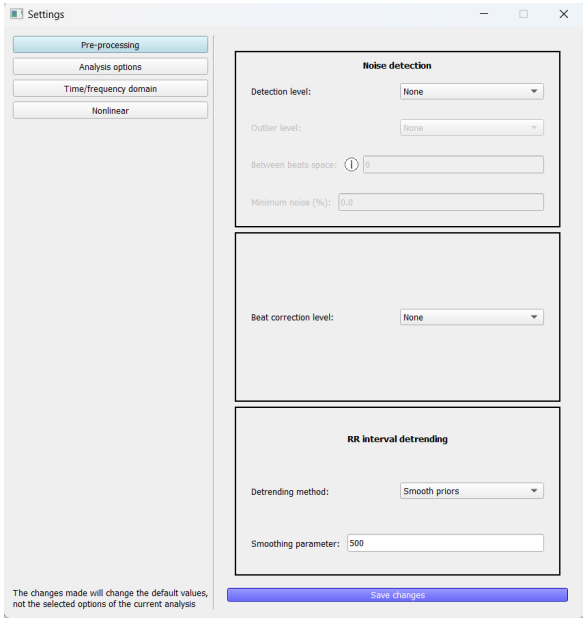
Figure 23: Nonlinear results

4.7 Settings

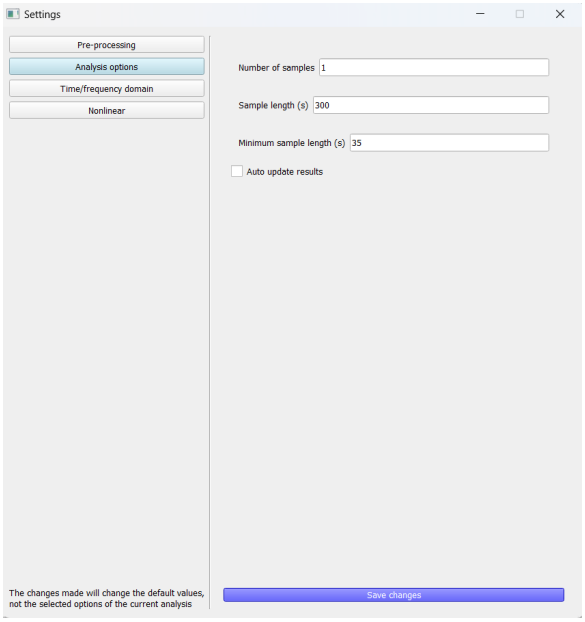
The settings correspond to the details of the program that can be modified by the user. Some of them have the goal of changing the default values of some variables. So, the next time the interface is loaded, these variables will start with the new values. Other settings have the goal of adding more detail in the analysis. Not all the options are displayed on the main interface, so some things can just be changed in the settings screen. There are four different settings sections:

- Processing options (Figure 24a). These ones include three sections:
  - Noise detection. There are three options on the noise detection algorithm. This algorithm first detects outliers and then checks for intervals where there are too many. So, first of all we need to specify what outlier detection method we will use. It can be based on the threshold or the algorithm. Then, we can specify how to decide if an interval is too noisy. We do that by specifying how close two outliers must be to be in the same noise interval. And how much of the signal has to be noisy to consider the interval noisy.

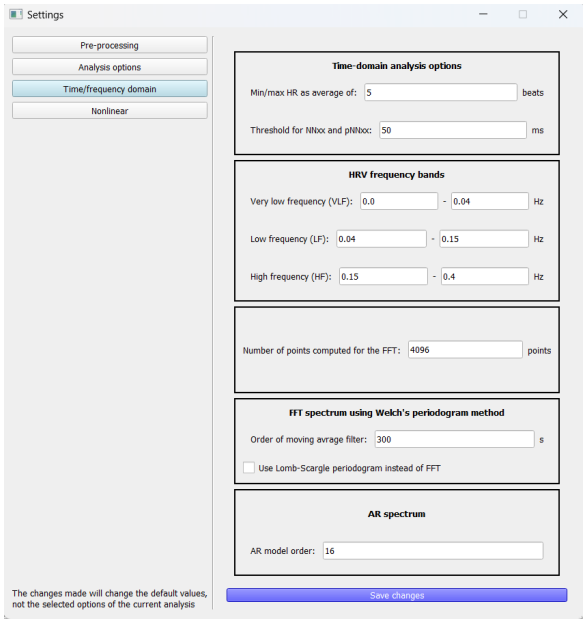
    The first line in this section lets us do these things automatically. We can choose Very low, low, medium or high and the values will be automatically set. If we choose Custom, we must specify these values.



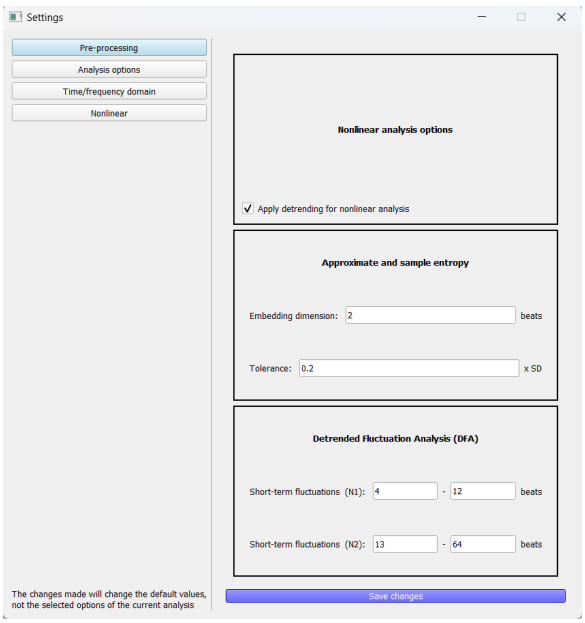
(a) Preprocessing settings



(b) Analysis options



(c) Time/frequency settings



(d) Nonlinear settings

Figure 24: Settings

  - Beat detection. This just has the option of the outlier detection method that will be used. It is the same that appears in the analysis options and it will change the default value.
  - Detrending. There are two possibilities: a polynomial approximation, that can be from first to third order, and the smoothn priors method. In case we choose the later, we also need to specify the smoothin parameter.

- Analysis options(Figure 24b). In this part we can specify the sample selection default values and whether the results will be automatically updated or not. The sample options are the number of subsamples, the length of each subsample and the minimum sample length.

- Time/frequency results (Figure 24c). In this screen there are five sections:

  - We can first specify the time results options. These are the number of beats that will be used to get the average of Heart Rate results. And the threshold to compute nnXX and pnnXX in ms.
  - The frequency limits. We can change the minimum and maximum frequencies for each frequency interval in Hz.
  - The number of points computed for the Welch periodogram.
  - Settings of the Welch method. The order of the moving average filter in seconds. And if we want to get the Lomb-Scargle method results instead of the Welch ones.
  - The Autoregressive mthod model order.

- Non linear results (Figure 24d). There are three different sections:

  - Checkbox to specify if we want to do detrending before getting the nonlinear results.
  - Approximate and sample entropy. Embedding dimension in number of beats and tolerance, relative to the standard deviation.
  - Detrended Fluctuation Analysis options. To specify the minimum and maximum number of beats of the short-term fluctuations and the long-term ones.

5 Technical evaluation

To do the evaluation of the program, we need to see if it meets the user requirements, but also if the analysis it does is correct. We start with this technical evaluation. There are many steps and calculations it does, so we also need to make sure the results are good. In this section, we compare EZCardio with other existing software and see if the RR intervals and metrics that it gives are correct. But apart from accuracy, the other advantage that EZCardio is expected to have is time efficiency. In 5.3 we also see how the prototype is currently performing on this aspect.

5.1 Visual validation

To check how the program is doing the analysis, we checked the results of each step to see if they do what they are intended to do. We can summarise the steps in beat detection, noise detection and outliers detection.

For the visual detection of the beats, we can check some intervals from different files. In Figure 25 we can see that the beat is accurate, although not always exactly on the R wave. This should not be a problem when getting the results, but increasing the window to upsample around the detected beat would be useful to increase the detail.

To compare the RR intervals with the real ones, we take Polar band files. This sensor outputs directly the RR intervals and the ECG. Polar band is popular for having a high accuracy in finding R peaks. Although the ECG recording quality can highly depend on the conditions, the accuracy of the peak detection remains high [52], so it was a good option to check the accuracy of EZCardio's beat detection method.

We use the ECG signal with EZCardio and get the RR intervals measured just after the beat detection. We plot these values with the Polar band RR intervals and find quite a high correlation, like we see on Figure 26 and 25.

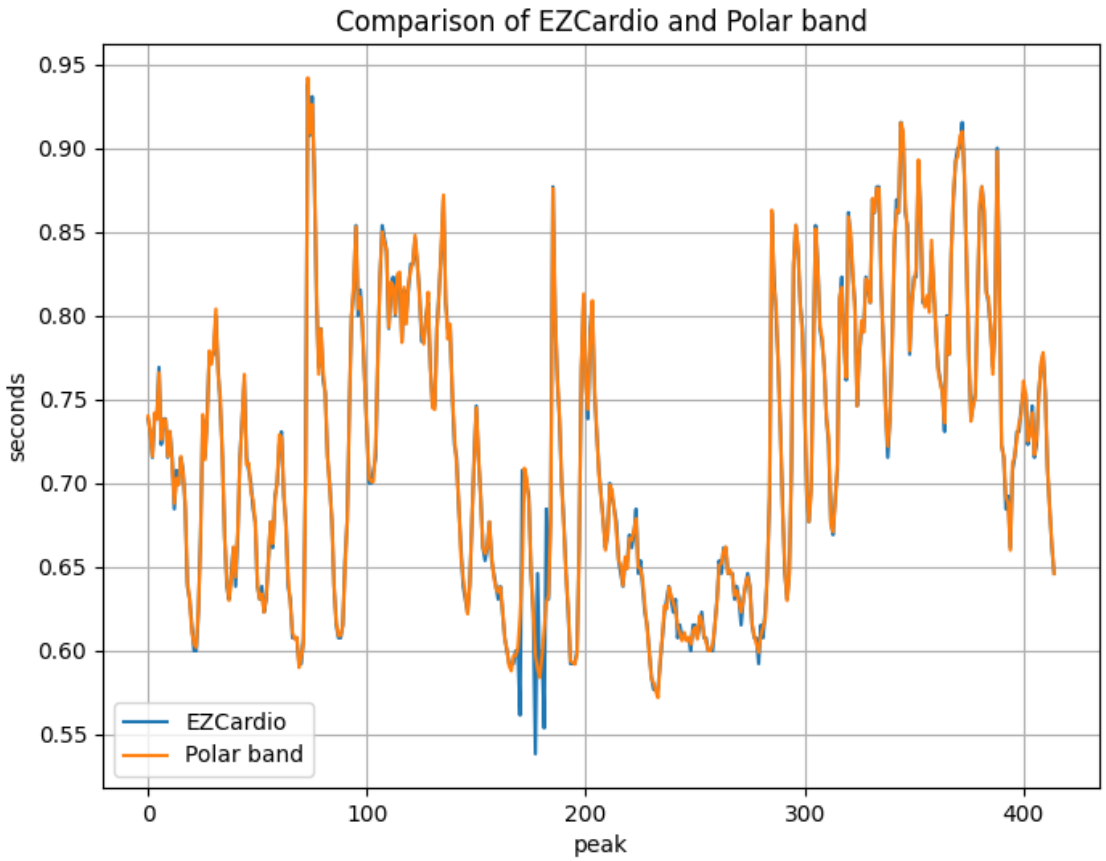


Figure 25: Comparison of the RR intervals from the beats detected by EZCardio and the original ones from Polar band

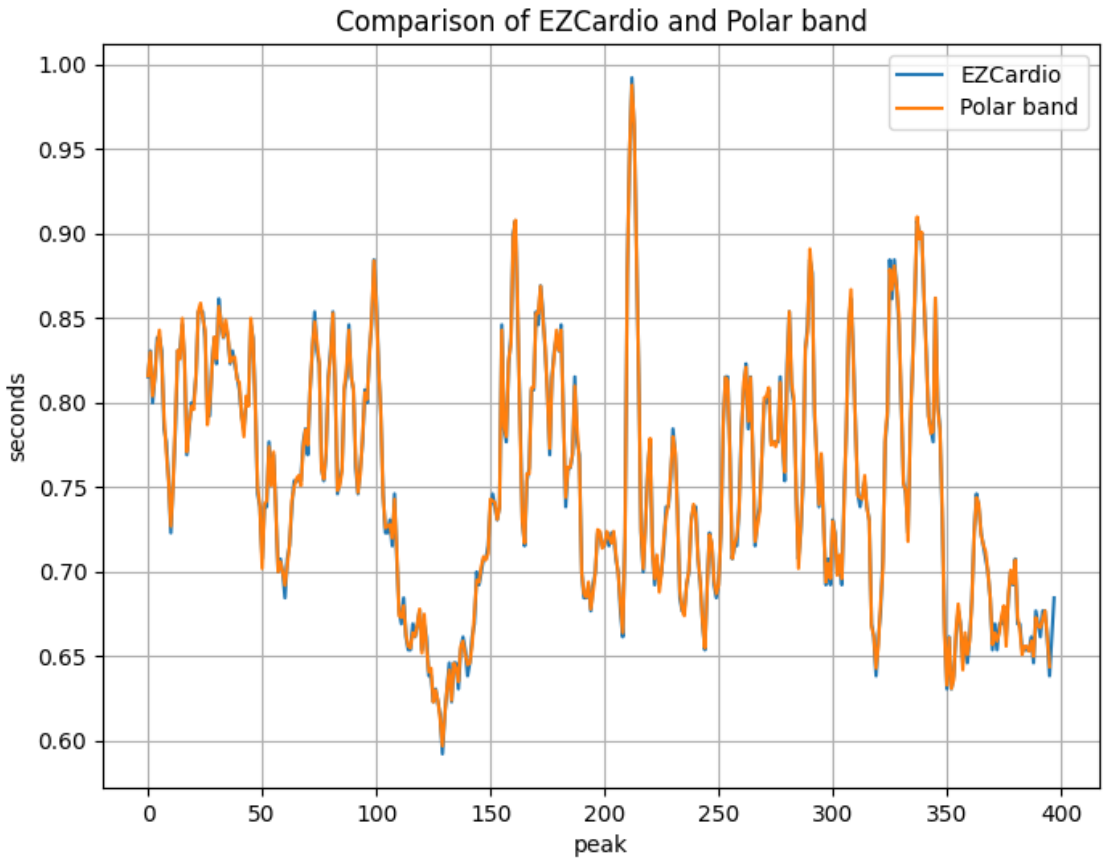


Figure 26: Comparison of the RR intervals from the beats detected by EZCardio and the original ones from Polar band

From these two intervals (EZCardio RR intervals and Polar band RR intervals), there is just two peaks with high differences, where it seems that the beat detector made a mistake. If we go to the part of the signal where these differences happen, we find that the beat detector added two extra peaks. These differences were expected, as they represent outliers in the analysis. We just need to make sure that the outlier correction (that we see on the next subsection) is working as expected, so these intervals where we see differences are corrected. If the outliers correction works, the EZCardio and Polar band intervals will be the same.

To check what the program detects as noise, we can check Figure 27. This is the end of one of the Polar band files. We can see something typical in ECG recording, which is a lot of noise in the last part of the signal, corresponding to the ECG electrodes being removed before the recording is disconnected. On the left part, the signal was just properly recording ECG. But we can see around the middle of the fragment that the signal starts having high peaks or being 0 for many seconds. This does not represent ECG, just noise, so it should be detected by the algorithm.

We first tried using just the ”Very low” noise correction and the program identifies this interval as noise, as we can see with the blue part in Figure 27. This means that, even when choosing the less restrictive noise detection option, the most evident noise intervals are already detected. If we choose a more restrictive correction ”High”, which is not shown in the figure, we find more intervals detected as noise, although some of them are not really noise. High correction might be useful for noisier signals,

but this one is quite clean. A recommendation should be done to the user in the documentation that a lower noise detection is more convenient when the signal is clean, while high correction can be used in more problematic signals.

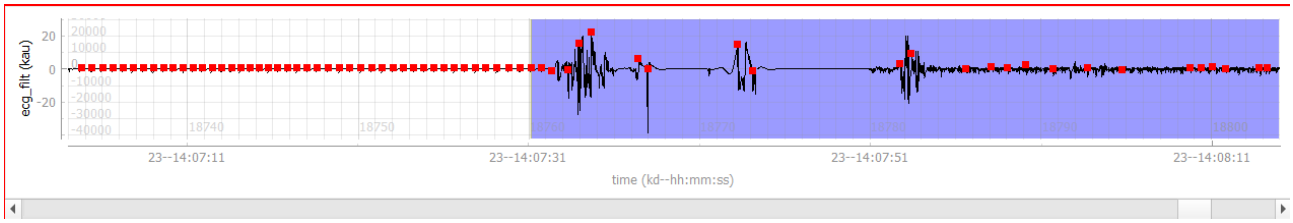


Figure 27: Example of the last interval of the signal detected as noise

Beat and noise detection seem to work as expected, but we still had to check the last processing step, which is outliers correction. In this case, we do not only do detection, as with noise, but also the correction. We see some example of outliers in Figures 28 and 29. In the graphs on top of each figure we see the ECG signal with the outliers corrected with a vertical red line and a red circle representing the new peak annotation. In the graphs on the bottom of each figure we see the current RR intervals (after outliers correction) represented by the dark line, while the original RR intervals are represented by the light line. In the first example, we can see that the correction is not done exactly on the beat, so we would need to manually annotate it. But in general, we find that the correction makes quite a lot of sense.

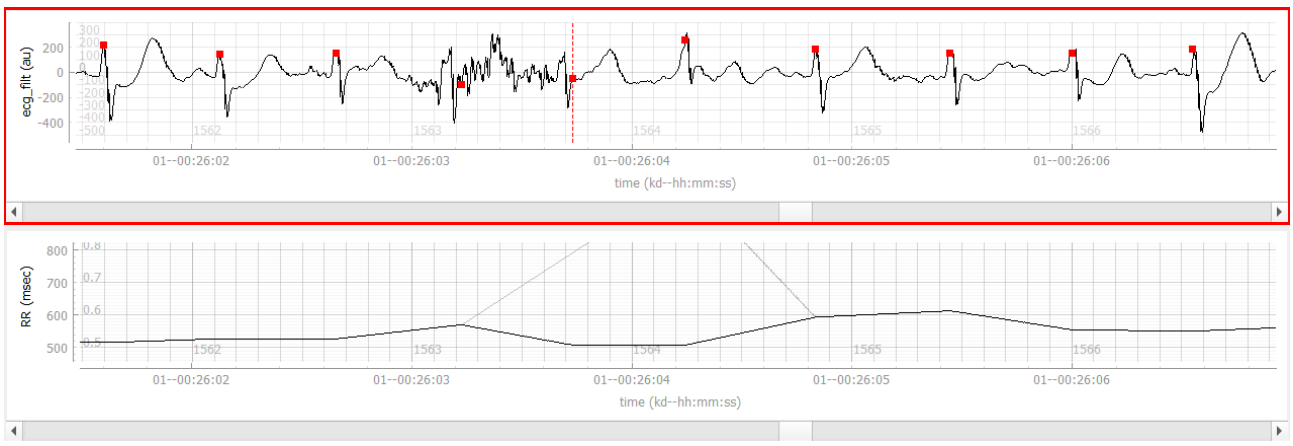


Figure 28: Example of an outlier corrected by EZCardio (adding a missing beat)

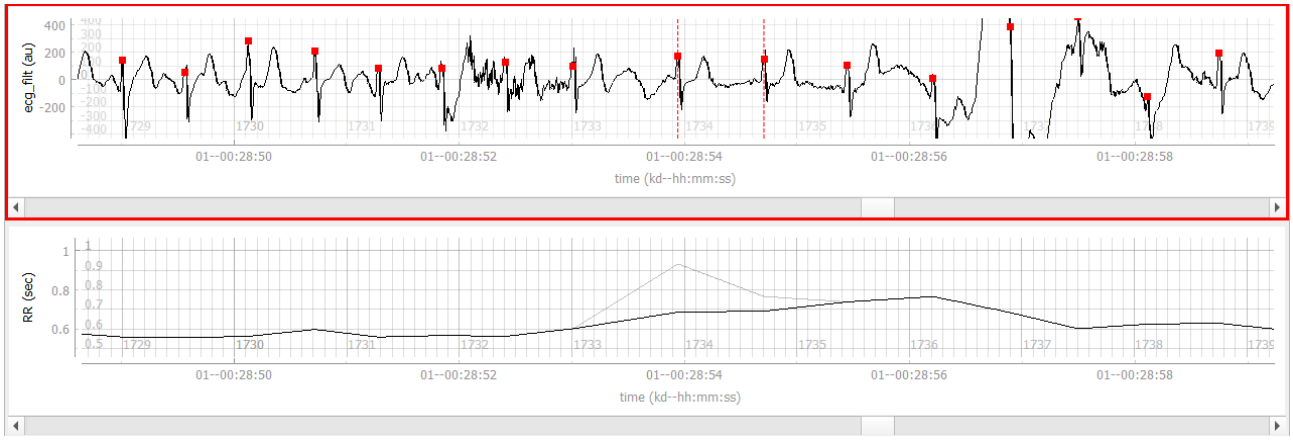


Figure 29: Example of an outlier corrected by EZCardio (interpolation)

5.2 Correction accuracy

To get a better understanding on whether the program is doing a good analysis, we need to compare the RR intervals we would get using EZCardio with another source. we conducted a detailed analysis comparing its performance to that of Kubios, a well-known software in the field. First of all, we looked at how each program analyzes RR intervals. Our comparison involved using five ECG signals obtained with a Cortrium ECG sensor, which are the signals used in the stress analysis research that this project belongs to. These signals were manually annotated for accuracy using PreCar software, serving as our 'ground truth' or reference standard.
PreCar software allows for initial beat detection with user adjustments for accuracy. This manual annotation process establishes a reliable baseline against which we can measure the performance of automatic analysis tools like EZCardio and Kubios.
The Comparison Process has two steps:

- Kubios Automatic Correction: Kubios provides an automatic outlier correction feature. It calculates the RR intervals using its built-in algorithm without further user intervention. This method's results were then compared to our ground truth to determine accuracy.

- EZCardio with User Confirmation: EZCardio also offers an automatic outlier correction. However, it adds a crucial step – after the automatic correction, users can review and adjust the corrections if necessary. This process aims to enhance accuracy by combining automated analysis with human oversight.

Given the possibility of discrepancies in beat detection (such as differing numbers of beats due to noise identification or missed/extra beats), a straightforward comparison was not feasible. Instead, we aligned the RR interval arrays from each software with the ground truth and identified matches within a 10 ms threshold—a more stringent criterion than the commonly used 100 ms, aiming for higher accuracy in our comparison.
The results, detailed in the table below, show the percentage of RR intervals from Kubios and EZCardio that matched our manually annotated ground truth (PreCAR annotated file). A higher percentage indicates greater accuracy. Our findings reveal that EZCardio, with its user confirmation step, consistently achieved closer alignment with the manual annotations compared to the fully automatic process of Kubios. This was particularly evident in the second and third ECG signals, where EZCardio matched the ground truth perfectly.

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5
Kubios (automatic)	99.28	99.28	99.87	99.22	99.64
EZCardio (automatic + manual confirmation)	99.86	100	100	99.75	99.93

Table 2: Comparison of the similarity (in percentage) of Kubios and EZCardio methods compared with the PreCAR manual annotated file, considered as the ground truth

This table underscores EZCardio's advantage in allowing users to refine automatic corrections manually, leading to results that more closely match the manually annotated ground truth compared to Kubios's fully automated approach.

5.3 Correction efficiency

We have seen that manually confirming the automatic corrections (EZCardio method) gives better accuracy than not doing it (Kubios method). But it still does not always get the same accuracy as the full manual correction. To compare EZCardio's correction with the manual one, we can start by seeing which one does the analysis faster, given by the processing time.
The processing time is always taken for the full analysis process, from when the file is loaded on the screen to when all the outliers have been corrected. Using PreCAR, the number of outliers to correct is higher, but using EZCardio the number is much lower, as the user just needs to do a confirmation of the algorithm corrections, not a full manual correction.
On Table 3 we can compare the efficiency of the full manual correction (which is the same PreCAR corrected file as before) and the algorithm + manual confirmation correction. EZCardio is always between 5 and 10 times faster, which is a very big improvement. In an experiment were many ECG signals need to be analysed, this can save a lot of time.

		Signal 1	Signal 2	Signal 3	Signal 4	Signal 5
Processing time (mm:ss)	EZCardio	00:51	01:33	02:01	01:07	00:39
	PreCAR	09:11	09:26	09:57	09:49	10:01
Number of corrections	EZCardio	6	19	13	8	2
	PreCAR	10	19	13	15	5

Table 3: Comparison of efficiency doing a full manual processing and an automatic+manual processing. Dark green means perfect result, bright green means acceptable result

Regarding the number of corrections, we count how many corrections were done with each program. EZCardio counts the number of corrections that the algorithm did plus the number of corrections done manually afterwards. With PreCAR, the user has corrected all the outliers that there were in the signal, so this can be seen as the ground truth, the number that EZCardio should also have.
On Table 3 we can see again that the correction for the second and third signals is exactly the same for the two options. On the other signals most of the corrections are also the same, but there are few differences between the corrections. When looking at where these corrections happen, we find a common pattern: low signal quality intervals. We can see an example on Figure 30. In these intervals, there are many outliers together and the algorithm does not detect them as outliers but as normal beats, as shown in Figure 30. The reason is that the variance of the RR values in a low quality interval is

high, so a different RR value can be considered a normal one. There was one of these intervals with non-detected outliers in each of the 3 signals.

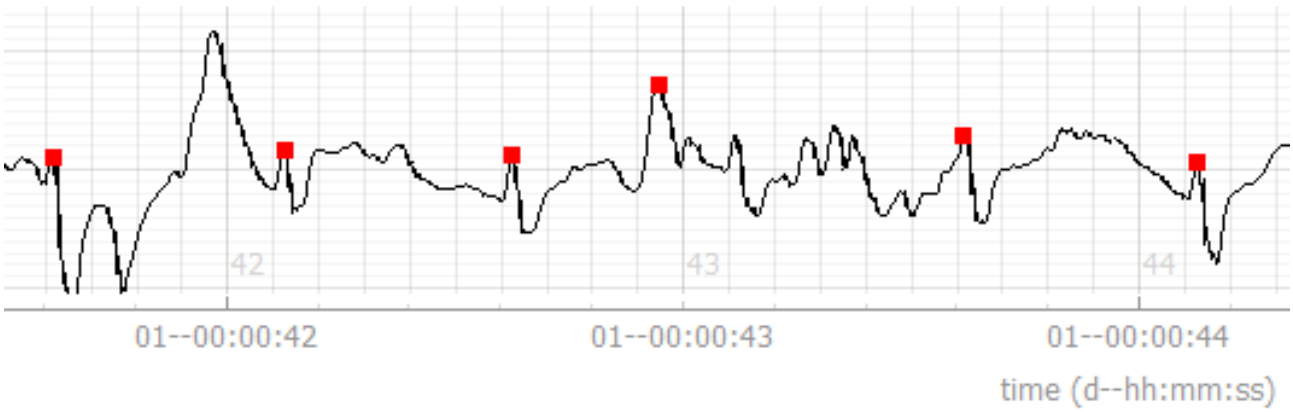


Figure 30: Example of non detected outliers in low quality region

5.4 Metrics comparison

Apart from the RR intervals, it is also necessary to test that the HRV metrics we are getting are correct. The methods used to get the metrics are gotten from the software pyHRV [7] with small modifications. This is an open source python library with many HRV metrics and graphs. These metrics have already been evaluated in detail [7], so it is not necessary to do a full evaluation again. The goal of testing the metrics here is to see if they improve after doing the correction.

To compare the results, we used the same 5 Cortrium files again, which were the ones available in the stress analysis project, and compared the manually annotated values with each of EZCardio corrections. To have a more reliable comparison, we get the metrics of the manually annotated file from Kubios, as it is the gold standard HRV analysis software. The process is then to manually annotate all the RR intervals using PreCAR and to import these values into Kubios, where the metrics can be exported. This gives the ground truth. On the other hand, the metrics for EZCardio are measured from EZCardio itself. What researchers care about the results is not the exact values but the trend through the samples, in other words, the relative difference compared to the previous sample. Apart from the Kubios and EZCardio corrections, we also get the metrics of the original RR intervals (after beat detection and with no outliers correction). These way we compare three different possibilities. The first one with just doing beat detection, the second one doing beat detection and outliers automatic correction (what Kubios could do) and the third one doing beat detection, outliers correcion and manual outliers confirmation (what EZCardio could do). And the ground truth to compare is the full manual processing, using PreCAR and Kubios.

For each of the 5 files, we get samples of 5 minutes and measure the value of different metrics in each sample. From these values, and starting from the second sample, we get the proportions relative to the first sample, considered as the baseline. For example, for a file that just has 3 samples, if for certain metric the value of the first sample is 5, the second sample is 10 and the third sample is 7.5, the proportions that we would compare are 1.5 (second sample / first sample) and 2 (third sample / first sample). We do this for every Cortrium file and HRV metric, having a list of values for each of these cases.

We repeat the process for each of the 3 possibilities we explained before (no outliers correction, kubios and EZCardio), getting 3 lists of values for each file and metric. We want to see how each is similar

to the ground truth (full manual processing with PreCAR and Kubios), so we measure the correlation of each of the 3 options with the ground truth. A very high correlation (higher than 0.8) will mean that the results are almost identical. A correlation lower than 0.6 means that there is something that should be improved.
We choose five metrics that can summarise the analysis for being the most popular ones or the most descriptive for each domain. We compare the correlation for the mean RR intervals (Table 4), the RMSS (Table 5), frequency ratio with Welch method (Table 6) and with Fast Fourier Transform method (Table 7) and the proportion of SD1 and SD1 (Table 8). Again, we compare the 3 possibilities of the analysis to see when we get the best results.
We can see that all the values except for the Mean RR are highly affected by outliers. Although we say before that the RR values are very similar, the results may not show any correlation sometimes. In some signals they do, but in other they do not. So, it is unreliable to do a full analysis just using automatic correction, as this can work for some signals but not for others.
But with the manual confirmation that we can do with EZCardio, we always find a very high correlation with the full manual processing. This means that, although there were few differences in the RR intervals values, they were not enough to affect the results.

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5
No outliers correction	0.928	0.819	0.971	0.84	0.997
Kubios automatic	0.927	0.819	0.971	0.838	0.997
EZCardio	0.931	0.872	0.97	0.914	0.997

Table 4: Correlation of mean RR in each possibility with the ground truth (full manual processing with PreCAR and Kubios). Dark green means high correlation, bright green means significant correlation and red means no correlation

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5
No outliers correction	0.947	0.796	0.427	0.683	0.915
Kubios automatic	0.941	0.844	0.428	0.697	0.931
EZCardio	0.964	0.822	0.96	0.999	0.994

Table 5: Correlation of RMSSD in each possibility with the ground truth (full manual processing with PreCAR and Kubios). Dark green means high correlation, bright green means significant correlation and red means no correlation

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5
No outliers correction	0.858	0.91	0.835	0.264	0.921
Kubios automatic	0.855	0.98	0.11	0.067	0.897
EZCardio	0.941	0.994	0.981	0.906	0.948

Table 6: Correlation of Autoregressive frequencies ratio in each possibility with the ground truth (full manual processing with PreCAR and Kubios). Dark green means high correlation, bright green means significant correlation and red means no correlation

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5
No outliers correction	0.604	0.856	0.364	0.219	0.454
Kubios automatic	0.607	0.655	0.364	0.168	0.412
EZCardio	0.966	0.997	0.978	0.668	0.948

Table 7: Correlation of FFT frequencies ratio in each possibility with the ground truth (full manual processing with PreCAR and Kubios). Dark green means high correlation, bright green means significant correlation and red means no correlation

	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5
No outliers correction	0.604	0.856	0.364	0.219	0.454
Kubios automatic	0.607	0.655	0.364	0.168	0.412
EZCardio	0.845	0.968	0.985	0.959	0.997

Table 8: Correlation of SD2/SD1 in each possibility with the ground truth (full manual processing with PreCAR and Kubios). Dark green means high correlation, bright green means significant correlation and red means no correlation

6 User evaluation

6.1 Introduction

The other evaluation we do is to talk to users again and ask them what they think about the program, to see if it meets their requirements and they would prefer it instead of the other programs. Considering users feedback, we do an additional usability analysis to see what parts of the program should be improved.
In the beginning of the project, a requirements analysis was done to see what users wanted in order to perform the HRV analysis. From these needs, the specifications and goals of the program were defined.

6.2 Methods

6.2.1 User interviews

We did another round of interviews with HRV researchers to see what they think about the program. 5 interviews were done with different researchers that include PhD students and different kinds of professors. All have experience and current projects with HRV with different applications, including psychology research, sports science and neurorehabilitation. They are researchers from the University Medical Center of Groningen and the German Sports Univerity of Cologne.
The evaluation was done by sending the program to the researchers and meeting while they used the program for the first time, to hear their first impressions. They were told to simulate doing the HRV analysis saying any thought that they had and some questions were also asked. Some people preferred not to install the program, so just a demo explanation while using the program was done. And they could give any feedback.
In this case, the interviews are done with people that use HRV in different fields, which is also useful to see how interdisciplinary the program can be. Ideally, the program should be useful for any kind of HRV application, as the analysis is exactly the same. But there might be specific requirements of one field that have not been included.
In this user study no specific questions were planned, as the approach was to let the user work with the program and hear they feedback while doing each step of the analysis. We plan a unstructured interview with the idea was to get feedback from mainly three different aspects.
First, we wanted to check the explainability. We wanted to see what parts of the program were harder to understand. While the user was doing an analysis, we could see where they got stuck or if they made mistakes because they did not understand what something in the program was expected to do.
Second, we wanted to check if the requirements from the previous user study were met. This means getting the list of requirements that we had before and consider it as a checklist, checking that each requirements was met.
And third, we also asked users for more general or personal feedback. The advantage of having 5 people that use HRV in different disciplines is that they work with different settings and preferences. Hearing the feedback from 5 different points of view could help us seeing how the program can be improved.
Apart of user feedback, we employed heuristic principles to refine our interface design systematically. Heuristics, in the context of human-computer interaction, are pre-established guidelines that serve as a benchmark for user-friendly design. By integrating these principles with the user feedback, we aim to elevate the usability of our software to align with established standards and best practices. For this

purpose, the most common heuristic principles are identified. An analysis of the program is done considering each of them and possible improvements are planned.

6.2.2 Heuristics analysis

Heuristic evaluation stands as a cornerstone in the domain of usability analysis, offering a systematic approach for identifying interface design flaws that impede user interaction. This analysis employs a set of established usability principles, or heuristics, as a benchmark to evaluate the user interface of EZCardio, with the aim of uncovering usability problems that detract from an intuitive user experience. Given the critical role of user satisfaction and efficiency in the success of digital solutions, this evaluation seeks to pinpoint actionable insights that can guide the refinement of EZCardio's design. The points we focus on to do the heuristics evaluation are:

- User is always informed about everything that happens in the program.
- User control and freedom.
- Consistency and standards.
- Minimize user working memory.
- Minimalist design.
- Useful error messages and documentation.
-

6.3 Results

The general feedback from the interviews was quite positive. All users though the program was more convenient than any of the existing HRV analysis programs and they would use it for their research. We report the user feedback considering the same requirements list that was used in the first user study, explaining if each requirement was met and further changes that users recommended. Below a summary of the user evaluation is provided, the full evaluation is shown in Appendix G.

6.3.1 Functional requirements

- Give the user full control of the signal
  - Providing useful and visible annotations. The peaks and outliers were easily identified. Just the possibility to improve the visualization of the original RR (the one in lighter black) was mentioned. This is the same color as the grid, so it can be hard to see. Changing the color or even not showing the grid should be considered to make this line more visible.
  - Possibility to edit any annotation as wanted. Nothing was missing for the users. But the shortcuts were not so intuitive in the beginning, as sometimes CTRL or SHIFT buttons are involved. To make it more intuitive, just left and right click should include all the options. The shortcuts and manual editing options were preferred by the users instead of not having the buttons. Although they need to click on the button and then do the correction, this was preferred instead of being able to do everything at the same time. The reason is that they would need to use many different shortcuts that included editing noise, peaks and samples together. Per separate, it is just few simple shortcuts.

  - Easy zooming and scrolling and Possibility to synchronize and desynchronize graphs. It was highly valued that the user has much control over what to visualize in the graph. The options of synchronizing or desynchronizing the graphs got good feedback, as well as zooming in and out so easily.

- Include all the automatic analysis options

  - Noise correction, outliers correction and sample selection. The program includes all the options that the user would need to do the analysis.

    But it was sometimes hard to understand the analysis options. Users started focusing more on the graphs and the manual correction before using the options on the left of the screen. This may just be that they were trying to get familiarised with the different options before starting the analysis, but it would be useful to make sure the automatic analysis is used before the manual correction, as this is the purpose of the program.
  - Possibility to choose settings of each analysis step and apply with just a click. The function of each analysis step can be easily recognised with the icons. It is easy to use or not to use with just a click and more details on how each step will be used can be selected on the settings.

- The user can immediately see each change that was done automatically, and modify it when needed.

  - Possible to annotate the changes done manually and go to each of them in the graph. The idea of having the selector to mark the outliers or noise and instantly seeing them was very liked by all users.

    However, some users did not consider the selectors when opening the signal and started visually scanning all the signal. This means that the distribution of the components in the interface is not optimal, as this can be missed, also because no other program has something similar to the selector, so users are not used to use it for the analysis.
  - Possible to manually delete or edit any of these changes. This can be done with the same editing shortcuts explained before.

6.3.2 Data requirements

The import page was liked in general, although more negative feedback was given here. Some people found the options to choose the data confusing, as it was not clear what each of them should represent on the data. One comment that was mentioned is that it would be better to have some kind of order for the options, so the user sees first the most important options.

6.3.3 Environmental requirements

- File output. The option to get the metrics was fine in general, as it just outputs everything that can be useful. But it was also mentioned that it would be better to have an option of exporting just one metric that the user can choose.

- Information buttons. They appear next to each option in the program and it is also possible to open the documentation from the program.

- Technical environment. Mac and Linux versions of the program are created. Many users have Mac computers, so also having a version for it was convenient.

  And the program can be used in any kind of computer. The size is around 120 Mb, which is something any computer can handle. And algorithms in general are quite simple, not using any complex computation.

6.3.4 User characteristics

- Self explanatory. When opening the main screen, users find it easy to identify the different options. The general structure is similar to other programs, so it was not hard to see what each component represented. When looking for certain options or trying to do something, they could quickly find it on the screen.

- Minimize computer procedures. For the analysis they just need a file to import in the beginning and after doing some easy steps with the program, they will get a new file that they can use for further analysis. To even minimize more the computer procedures, more file types should be allowed in the import.

6.3.5 Usability goals

- Hide or show parts of the screen. Resize buttons were easily recognised by the users. They could hide or show the results and the analysis options.

- Choose many samples at once. The sample selection was also hard to understand. The most common way of selecting samples is by manually adding them, one by one, so the option to add many at the same time was not understood in the beginning. Adding different sample blocks with different names is also new, and users though that blocks referred to individual samples. This should be made more clear. Furthermore, people also find useful to see how many outliers there were at each single sample, and get a warning if the number is big.

- Save analysis to import later. One of the options in the menu lets the user save the analysis.

- More import options. The other options can be easily identified in the import page.

- Append results to existing file. This can also be done from the menu and is easily understood.

6.3.6 Heuristics analysis results

- User is always informed about everything that happens in the program.

  This principle should be improved when selecting a sample. As mentioned before, the sample selection was confusing, specially when users tried to add a sample manually. They want to see the sample added, but this just happens when the current sample block has already been created.

- User control and freedom.

  It is also important not to make assumptions about what option the user would prefer and try to give as many options to choose as possible. For this program it was assumed that using Pan Tompkins for example was the best beat detection method, as well as the outliers based noise

detection to identify noise. Although these methods have many advantages over alternative ones, users may also want to have the option to choose a different method.

Another important point is that the selector currently just has the option to go to the previous or next outlier or noise interval. There should be the option to go to a particular one with for example a dropdown button.

- Consistency and standards. It may not be so consistent to have a different selection option for the two outliers correction. If the user wants to use both corrections and then manually check the outliers, they will just be able to see a part of the total outliers at once. It is better to see all the outliers to have a better understanding of what the full correction is doing.

- Minimize user working memory.

  The shortcuts could be improved. With the current version, this can be seen by keeping the mouse in the corresponding editing button. The shortcuts of the selected option could remain on the screen, so the user does not need to remember them.

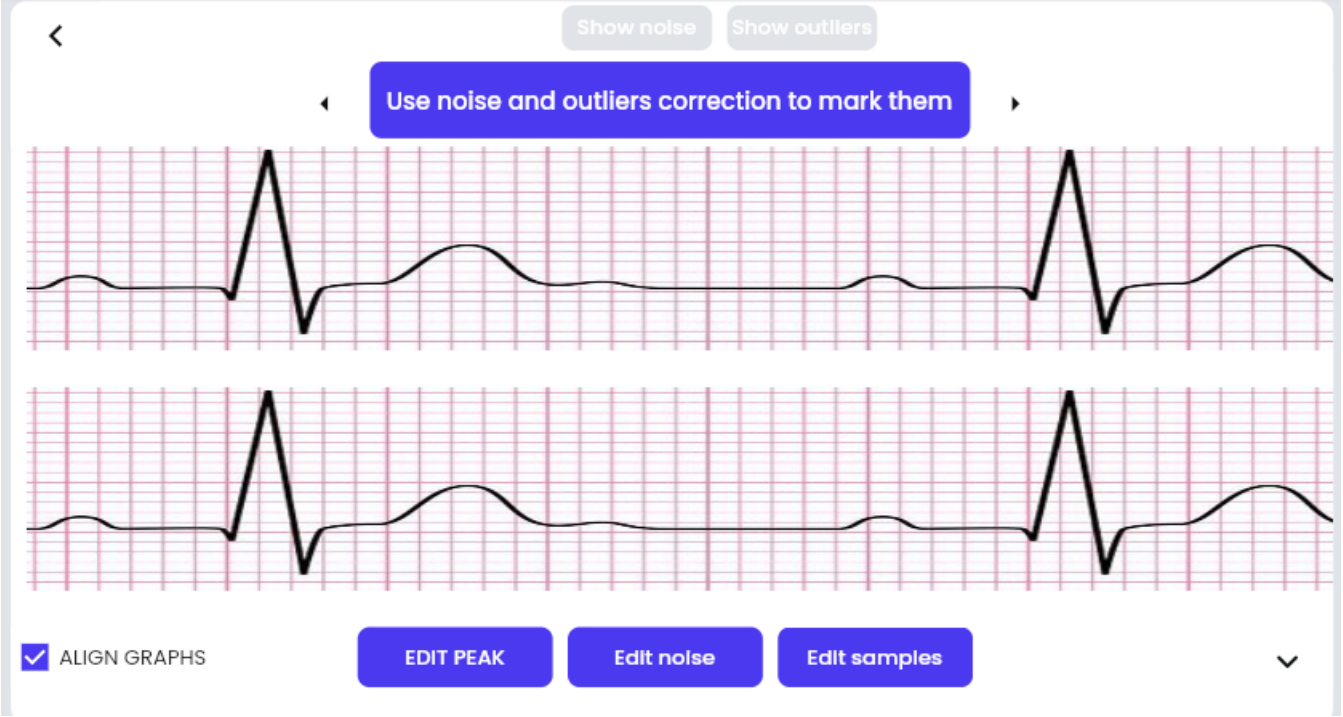


Figure 31: Improved design for the graphs

7 Discussion

In this last section we conclude the report by explaining what the next steps to improve the program would be. The first subsection summarises the changes that should be done considering the usability feedback in the previous section. The second subsection explains the next steps in developing the program from a more general point of view, also referring to technical improvements. The last subsection gives a conclusion of the report, explaining if and how the initial goals of the project were met.

7.1 Future usability improvements

The program in general meets the requirements. But there are still many changes that would be useful to improve the program, specially related to explainability. Most of the changes that users would like to have are just design issues or small options to improve the convenience of the program. So, in general, it would be possible to do a full HRV analysis using EZCardio.
From the feedback that the users gave, we find that some changes should be done in the next version of the program. The full list and explanations can be found in Appendix H, but a summary is:

- Selector more centered and with color.

  Centralize and color-highlight the selector for immediate visibility upon program startup, emphasizing its importance for noise and outlier correction. Simplify the interface by resizing adjacent buttons and arrows, and adopt blue for attention and red for activation to streamline user navigation. Suggesting a design overhaul to prioritize the selector and simplify annotation options.

  We can see a possible new design for this part in Figure 31.

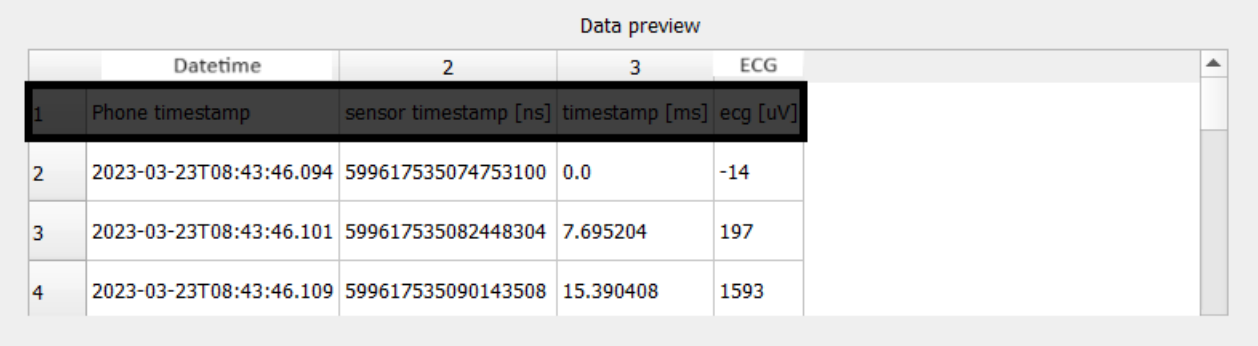


Figure 32: Improved design for import page in which ”Header lines” is 1, ”Data column” is 4 and ”Time column” is 1

- In the import page, improve the understanding of each import option.

  Reorganize the import options for intuitive navigation and implement real-time data previews highlighting selected options. This change aims to facilitate understanding and selection of header lines, data columns, and time columns, mirroring practices from other programs like Kubios. For example, in Figure 32 we see the data preview when we have chosen 1 for ”Header lines”, 4 for ”Data column” and 1 for ”Time column”.

- Changes related to samples

  Disable the ”Edit sample” button until a sample block is added to prevent premature editing attempts, and refine terminology for clarity (e.g., ”segments” for ”samples”). Introduce more visible markers for sample start and end points on graphs, and consider displaying key sample metrics (outliers, stationarity) in an accessible manner, with alerts for threshold breaches to aid in identifying problematic samples without cumbersome navigation.

- Option to just export one result

  Provide a more granular export feature allowing users to select specific results for export through a simplified interface, with a default ”Export all” option and the ability to individually select results for inclusion in the output.

- Option to mark or correct outliers

  Offer users the choice to mark without automatically correcting outliers and noise, to accommodate preferences for manual correction or verification, especially useful in dual correction scenarios or when precision is paramount.

- Adding time varying results

  Implement a feature for previewing how certain variables evolve over time within the program, offering a preliminary analysis before exporting data for further examination, enhancing the user’s ability to assess trends and anomalies directly within the software environment.

7.2 Future work

As mentioned, there are still some important changes that should be implemented to improve EZCardio. The first thing to consider is that EZCardio’s correction still does not get perfect accuracy in some signals, and this should be the first next goal. As explained with Figure 30, the problem is that

some outliers are missed when a fragment of the signal has lower quality. For the signals tested in this research and probably most of the used ECG signals, the current method can work. But for more noisy signals this can be problematic, as the signal will have lower quality and more outliers will be missed.

A possibility to improve this would be to find an additional way to detect the outliers. This could be done using for example a more restrictive threshold correction after using the algorithm algorithm. Adding an additional outliers detection step, for example one just based on a restrictive threshold, would be able to identify the outliers that were missed by the algorithm. To avoid modifying too much of the signal, this could be included only as a detection step, without correcting the outliers. So, the user could mark these outliers in the selector and check each of them. With this option it is much less likely that outliers are missed, but it should be tested to see if there would still be missing outliers and how the time efficiency would change, as now the user would need to check more beats.

Another option to detect the outliers would be to measure the signal quality for specific regions. This can be done in different ways. A common approach is to get the PSD of each interval in the signal to check if the frequencies that reflect ECG waveforms are represented [6]. Another possible step would be marking high peaks, as the beat detection precision ussually decreases after high peaks. This should be evaluated again specially for the time efficiency, as adding these options would mean adding an additional step in the analysis.

Also, we can include an improved beat detection method for very noisy signals. There are some Deep Learning models that show a very high accuracy on noisy signals. Although this would take time, it would be better for the user when the signal is very noisy and Pan Tompkins would get a low accuracy. A problem of including this model is that the size of the executable program would be much higher, so ideally this should happen in a server. Ideally, a future version of EZCardio would connect to a server that has all the algorithms, so the program the user needs would be very light (even a website) and the analysis would be done much faster.

In case we do not implement the Deep Leaning option, we still need to worry about the noisy intervals and the inverse signals. Beat detection methods are very sensitive to noisy segments. As mentioned earlier, just a short noisy interval can make the beat detection unreliable for the following seconds. It is possible to consider that interval as noisy, but the beat detection is done before this noise annotation. So, a further improvement would be to let the user redo the beat detection when there is noise (or for any other reason). This could be done by right-clicking on the graph and having a menu with an options saying ”Re-use beat detection from here”. Or with another option in the menu on top in which the user can indicate the start and end of the interval where they want to re-detect beats.

Furthermore, there should be an option to invert the signal. An inverted signal always makes the beat detection fail and algorithms that detect if ECG is inverted are highly unreliable, while for a human that is very easy to indicate. An additional option on the top menu should give the option of inverting the signal, after which the beat detection would be done again for the full signal.

The next important changes would be the ones related to usability. We already explained what users think about the program and the modifications that they would like to have. Although the feedback was very positive and the proposed changes are secondary, implemented the design for the graphs shown in Figure 31 would improve the usability. Some other changes are also convenient, but not the priority for the next version of EZCardio.

Lastly, ECG is related to many different fields and it allows more analysis possibilities than the ones mentioned on this research. There are different kinds of analysis that can be done with ECG and EZCardio, with small changes, could also support. There can be a detection of cardiac pathologies or abnormal beats. And by including respiratory data, it would also be possible to measure exercise results, useful to optimize training and measure rest periods.

7.3 Conclusion

EZCardio has been tested for its accuracy and usability and results have been highly successful for both. It is a user-friendly program that lets users do an accurate HRV analysis by having more control and knowledge of what they are doing. But also in a more time efficient way than having to visualise the full signal. The hypothetical improvements that were implemented were highly appreciated by the users, who recognised the program is better than anything else they have tried for HRV analysis.
Moreover, the results evaluation confirms that an analysis done with EZCardio would be accurate enough, as the correlation of the results in all the signals is very high. And the results would be almost identical to results gotten from manually annotated files and Kubios, which currently is software world leader and most trusted tool for HRV analysis.
The main improvement of this software is the time efficiency. Users want to do the analysis as fast as possible, but deciding to do so means losing reliability of the analysis with the current programs, which was also seen in this research. With EZCardio, the analysis can be done much faster while keeping the analysis accuracy, something no other HRV analysis software had ever achieved.
All these improvements were not just validated by the results comparison but also by user feedback. The general feedback was very positive and all the users asked if they could use the software for their research, as they think it is a unique tool that improves any other similar program. Lack of time is a problem that many researchers face and they recognise it can lead to human error, as outliers can also be missed by the human eye.
In case the user would like to do the full manual analysis with no automatic correction or marking outliers, EZCardio also has big advantages. The usability of the program, giving more freedom for the user with things like expanding parts of the screen and hiding others or zooming and scrolling in the signal was highly appreciated.
On the other hand, there are still further improvements that should be done in the program. Although it could already be used for HRV studies with high reliability, there are some improvements that would be convenient before this happens.

Bibliography

[1] Mary Adeniji, James Brimicombe, Martin R Cowie, Andrew Dymond, Hannah Clair Lindén, Gregory YH Lip, Jonathan Mant, Madhumitha Pandiaraja, Kate Williams, Peter H Charlton, et al. Prioritising electrocardiograms for manual review to improve the efficiency of atrial fibrillation screening. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 3239–3242. IEEE, 2022.

[2] Tricia Adjei, Wilhelm von Rosenberg, Takashi Nakamura, Theerasak Chanwimalueang, and Danilo P Mandic. The classa framework: Hrv based assessment of sns and pns dynamics without lf-hf controversies. *Frontiers in physiology*, 10:505, 2019.

[3] Anthony Almond. Acqknowledge (software). *The International Encyclopedia of Communication Research Methods*, pages 1–2, 2017.

[4] Sardar Ansari, Jonathan Gryak, and Kayvan Najarian. Noise detection in electrocardiography signal for robust heart rate variability analysis: A deep learning approach. *PubMed*, Jul 2018.

[5] RM Baevsky and AP Berseneva. Methodical recommendations use kardivar system for determination of the stress level and estimation of the body adaptability standards of measurements and physiological interpretation, 2008.

[6] Syed Khairul Bashar, Eric Ding, Allan J Walkey, David D McManus, and Ki H Chon. Noise detection in electrocardiogram signals for intensive care unit patients. *IEEE Access*, 7:88357–88368, 2019.

[7] Pedro Miguel Caridade Gomes. *Development of an open-source Python toolbox for heart rate variability (HRV)*. PhD thesis, Hochschule für angewandte Wissenschaften Hamburg, 2019.

[8] Niki Carver, Vikas Gupta, and John E Hipskind. Medical error. In *StatPearls [Internet]*. StatPearls Publishing, 2022.

[9] Radhika Dua, Jiyoung Lee, Joon-myoung Kwon, and Edward Choi. Automatic detection of noisy electrocardiogram signals without explicit noise labels. *arXiv preprint arXiv:2208.08853*, 2022.

[10] Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology. Heart rate variability: standards of measurement, physiological interpretation, and clinical use. *Circulation*, 93(5):1043–1065, 1996.

[11] Andrejs Fedjajevs, Willemijn Groenendaal, Carlos Agell, and Evelien Hermeling. Platform for analysis and labeling of medical time series. *Sensors*, 20(24):7302, 2020.

[12] Moncef Gabbouj, Serkan Kiranyaz, Junaid Malik, Muhammad Uzair Zahid, Turker Ince, Muhammad EH Chowdhury, Amith Khandakar, and Anas Tahir. Robust peak detection for holter ecgs by self-organized operational neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[13] Dan Gordon. Intense training – why f1 is one of the most physically and mentally demanding sports on the planet, May 2021.

[14] PD Grantcharov, T Boillat, S Elkabany, Katarzyna Wac, and H Rivas. Acute mental stress and surgical performance. *BJS open*, 3(1):119–125, 2019.

[15] Ralf Hartmann, Frank M Schmidt, Christian Sander, and Ulrich Hegerl. Heart rate variability as indicator of clinical state in depression. *Frontiers in psychiatry*, 9:735, 2019.

[16] Heart.org. Heart rate values, Mar 2021.

[17] Md Niaz Imtiaz and Naimul Khan. Pan-tompkins++: A robust approach to detect r-peaks in ecg signals. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2905–2912. IEEE, 2022.

[18] Ami E Iskandrian and Ernest V Garcia. *Nuclear cardiac imaging: principles and applications*. Oxford University Press, 2008.

[19] Ben-Tzion Karsh, Kamisha Hamilton Escoto, John W Beasley, and Richard J Holden. Toward a theoretical approach to medical error reporting system research and design. *Applied ergonomics*, 37(3):283–295, 2006.

[20] Tobias Kaufmann, Stefan Sütterlin, Stefan M Schulz, and Claus Vögele. Artiifact: a tool for heart rate artifact processing and heart rate variability analysis. *Behavior research methods*, 43:1161–1170, 2011.

[21] Kathi J Kemper, Craig Hamilton, and Mike Atkinson. Heart rate variability: impact of differences in outlier identification and management strategies on common measures in three clinical populations. *Pediatric research*, 62(3):337–342, 2007.

[22] Ahsan Habib Khandoker, Chandan Karmakar, Michael Brennan, Marimuthu Palaniswami, and Andreas Voss. *Poincaré plot methods for heart rate variability analysis*. Springer, 2013.

[23] Ulrich Kirk and Johanne L Axelsen. Heart rate variability is enhanced during mindfulness practice: A randomized controlled trial involving a 10-day online-based mindfulness intervention. *PloS one*, 15(12):e0243488, 2020.

[24] Kubios. Analysis methods - kubios 2023, Jan 2023.

[25] Kubios. Hrv and relation with ans and pns, Aug 2023.

[26] Juho Laitala, Mingzhe Jiang, Elise Syrjälä, Emad Kasaeyan Naeini, Antti Airola, Amir M Rahmani, Nikil D Dutt, and Pasi Liljeberg. Robust ecg r-peak detection using lstm. In *Proceedings of the 35th annual ACM symposium on applied computing*, pages 1104–1111, 2020.

[27] Duan Liang, Shan Wu, Lan Tang, Kaicheng Feng, and Guanzheng Liu. Short-term hrv analysis using nonparametric sample entropy for obstructive sleep apnea. *Entropy*, 23(3):267, 2021.

[28] Jukka A Lipponen and Mika P Tarvainen. A robust algorithm for heart rate variability time series artefact correction using novel beat classification. *Journal of medical engineering & technology*, 43(3):173–181, 2019.

[29] AC Vinzio Maggio, María Paula Bonomini, Eric Laciar Leber, and Pedro David Arini. Quantification of ventricular repolarization dispersion using digital processing of the surface ecg. *Advances in Electrocardiograms-Methods and Analysis*, pages 181–206, 2012.

[30] Anita Miftahul Maghfiroh, Syevana Dita Musvika, Levana Forra Wakidi, Lamidi Lamidi, Sumber Sumber, Muhmmad Ridha Mak'ruf, Andjar Pudji, and Dyah Titisari. State-of-the-art method to detect r-peak on electrocardiogram signal: a review. In *Proceedings of the 1st International Conference on Electronics, Biomedical Engineering, and Health Informatics: ICEBEHI 2020, 8-9 October, Surabaya, Indonesia*, pages 321–329. Springer, 2021.

[31] Nikunj Hasmukhbhai Makwana, Nikunj Makwana, Nishant Mishra, and Balwalli Sagar. Hilbert transform based adaptive ecg r-peak detection technique. *International Journal of Electrical and Computer Engineering*, 2(5):639, 2012.

[32] Mostefa Merah, TA Abdelmalik, and BH Larbi. R-peaks detection based on stationary wavelet transform. *Computer methods and programs in biomedicine*, 121(3):149–160, 2015.

[33] Rudolf Metelka. Heart rate variability-current diagnosis of the cardiac autonomic neuropathy. a review. *Biomedical Papers of the Medical Faculty of Palacky University in Olomouc*, 158(3), 2014.

[34] G Moody, W Muldrow, and R Mark. The mit-bih noise stress test database. *Computers in cardiology*, pages 381–384, 1984.

[35] Sidharth Nabar, Ayan Banerjee, Sandeep KS Gupta, and Radha Poovendran. Generative model-driven resource-efficient monitoring of ecg.

[36] Nevrokard.eu. Heart rate variability, baroreflex sensitivity blood pressure variability software designers 2023, 2023.

[37] Krzysztof Nowak, Anna Olga Kuzminska, and Katarzyna Kinga Kowalczyk. The effect of overflow at workplace on employees productivity and well being. *Economic and Social Development: Book of Proceedings*, pages 322–331, 2018.

[38] Saurabh Pal and Madhuchhanda Mitra. Empirical mode decomposition based ecg enhancement and qrs detection. *Computers in biology and medicine*, 42(1):83–92, 2012.

[39] Jiapu Pan and Willis J Tompkins. A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering*, (3):230–236, 1985.

[40] Mirja A Peltola. Role of editing of r–r intervals in the analysis of heart rate variability. *Frontiers in physiology*, 3:148, 2012.

[41] Daniel J Plews, Paul B Laursen, Jamie Stanley, Andrew E Kilding, and Martin Buchheit. Training adaptation and heart rate variability in elite endurance athletes: opening the door to effective monitoring. *Sports medicine*, 43:773–781, 2013.

[42] Jennifer Preece, Helen Sharp, and Yvonne Rogers. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2015.

[43] Medical Exam Prep. The basics of ecg interpretation (part 3 – waves, segments intervals), Mar 2016.

[44] Juan F Ramirez-Villegas, Eric Lam-Espinosa, David F Ramirez-Moreno, Paulo C Calvo-Echeverry, and Wilfredo Agredo-Rodriguez. Heart rate variability dynamics for the prognosis of cardiovascular risk. *PloS one*, 6(2):e17060, 2011.

[45] John T Ramshur Jr. Design, evaluation, and application of heart rate variability analysis software (hrvas). 2010.

[46] L Rodríguez-Liñares, María J Lado, XA Vila, Arturo J Méndez, and Pedro Cuesta. ghrv: Heart rate variability analysis made easy. *Computer methods and programs in biomedicine*, 116(1):26–38, 2014.

[47] Adriano L Roque, Vitor E Valenti, Thais Massetti, Talita Dias Da Silva, Carlos Bandeira de Mello Monteiro, Fernando R Oliveira, Álvaro Dantas de Almeida Junior, Sheylla Nadjane Batista Lacerda, Gustavo Carreiro Pinasco, Viviane Gabriela Nascimento, et al. Chronic obstructive pulmonary disease and heart rate variability: a literature update. *International archives of medicine*, 7(1):1–8, 2014.

[48] Jana Rybanská, L'udmila Nagyová, and Ingrida Košičiarová. The use of hrv analysis in the marketing research. *The agri-food value chain: challenges for natural resources management and society*, pages 1050–1057, 2016.

[49] Abdul Salam, David M Segal, Munir Ahmad Abu-Helalah, Mary Lou Gutierrez, Imran Joosub, Wasim Ahmed, Rubina Bibi, Elizabeth Clarke, and Ali Ahmed Al Qarni. The impact of work-related stress on medication errors in eastern region saudi arabia. *International Journal for Quality in Health Care*, 31(1):30–35, 2019.

[50] Fred Shaffer and Jay P Ginsberg. An overview of heart rate variability metrics and norms. *Frontiers in public health*, page 258, 2017.

[51] Janko Slavic. Python tools. *GitHub*, 2013.

[52] Sarah E Stahl, Hyun-Sung An, Danae M Dinkel, John M Noble, and Jung-Min Lee. How accurate are the wrist-based heart rate monitors during walking and running activities? are they accurate enough? *BMJ Open Sport—Exercise Medicine*, 2(1), 2016.

[53] Jennifer Stanford, Online School, Conner Galloway, Inc Alivecor, and Alexander Valys. *Deep Convolutional Neural Networks for Noise Detection in ECGs*. 2018.

[54] Nahoko Takada, Tipporn Laohakangvalvit, and Midori Sugaya. Human error prediction using heart rate variability and electroencephalography. *Sensors*, 22(23):9194, 2022.

[55] Mika P Tarvainen, Juha-Pekka Niskanen, Jukka A Lipponen, Perttu O Ranta-Aho, and Pasi A Karjalainen. Kubios hrv–heart rate variability analysis software. *Computer methods and programs in biomedicine*, 113(1):210–220, 2014.

[56] Mika P Tarvainen, Perttu O Ranta-Aho, and Pasi A Karjalainen. An advanced detrending method with application to hrv analysis. *IEEE transactions on biomedical engineering*, 49(2):172–175, 2002.

[57] Radhagayathri K Udhayakumar, Chandan Karmakar, and Marimuthu Palaniswami. Approximate entropy profile: a novel approach to comprehend irregularity of short-term hrv signal. *Nonlinear Dynamics*, 88:823–837, 2017.

[58] Adriana N Vest, Giulia Da Poian, Qiao Li, Chengyu Liu, Shamim Nemati, Amit J Shah, and Gari D Clifford. An open source benchmarked toolbox for cardiovascular waveform and interval analysis. *Physiological measurement*, 39(10):105004, 2018.

[59] Sricharan Vijayarangan, R Vignesh, Balamurali Murugesan, SP Preejith, Jayaraj Joseph, and Mohansankar Sivaprakasam. Rpnet: A deep learning approach for robust r peak detection in noisy ecg. In *2020 42nd annual international conference of the IEEE engineering in medicine & biology society (EMBC)*, pages 345–348. IEEE, 2020.

[60] A Yadav and N Grover. A review of r peak detection techniques of electrocardiogram (ecg). *Journal of Engineering and Technology (JET)*, 8(2):115–134, 2017.

[61] Bin Yu, Mathias Funk, Jun Hu, Qi Wang, and Loe Feijs. Biofeedback for everyday stress management: A systematic review. *Frontiers in ICT*, 5:23, 2018.

[62] Muhammad Uzair Zahid, Serkan Kiranyaz, Turker Ince, Ozer Can Devecioglu, Muhammad EH Chowdhury, Amith Khandakar, Anas Tahir, and Moncef Gabbouj. Robust r-peak detection in low-quality holter ecgs using 1d convolutional neural network. *IEEE Transactions on Biomedical Engineering*, 69(1):119–128, 2021.

[63] Zhongyao Zhao, Chengyu Liu, Yaowei Li, Yixuan Li, Jingyu Wang, Bor-Shyh Lin, and Jianqing Li. Noise rejection for wearable ecgs using modified frequency slice wavelet transform and convolutional neural networks. *IEEE Access*, 7:34060–34067, 2019.

8 Appendix

8.1 Appendix A. Pan-Tompkins++

All the steps involved in the Pan-Tompkins++ beat detection are explained on Algorithm 1.

Algorithm 1 Pan-Tompkins++

```
1: Preprocessing Phase
2: Apply a bandpass filter with a passband of 5–18 Hz to remove the noise in ECG
3: Differentiate the signal
4: Perform point by point squaring
5: Smooth the signal using a 60ms wide flattop window function
6: Apply moving window integration with a 150ms wide window
7: Decision Phase
8: Select all the peaks that are at least 231ms apart
9: Initialize the thresholds
10: for each Peak do
11:     if Peak >Threshold1 then
12:         Classify it as R-peak
13:         Adjust SPK and NPK using Rule-1
14:         if number of detected beats >8 then
15:             Calculate the Mean RR-interval of the eight most recent beats
16:             Calculate current RR-interval
17:             if RR-interval <360ms OR RR-interval <0.5 * Mean RR-interval then
18:                 Classify it as T-wave or QRS complex based on slope
19:                 Adjust SPK and NPK using Rule-1, if QRS complex found
20:             else if RR-interval >1s OR RR-interval >1.66 * Mean RR-interval then
21:                 Threshold3 = 0.5 * Threshold2 + 0.5 * MEANSB
22:                 if Peak >Threshold3 then
23:                     Classify it as R-peak
24:                     Adjust SPK and NPK using Rule-2
25:                 else if RR-interval >1.4s then
26:                     if Peak >0.2 * Threshold2 then
27:                         Classify it as R-peak
28:                         Adjust SPK and NPK using Rule-2
29:                     end if
30:                 end if
31:             end if
32:         end if
33:     end if
34: end for
35: Threshold1 = NPK + 0.25 * (SPK - NPK)
36: Threshold2 = 0.4 * Threshold1
```

SPK and NPK are adjusted with Rule 1, that is

$$SPK = 0.125 * PEAK + 0.875 * SPK$$

$$NPK = 0.125 * PEAK + 0.875 * NPK$$

, or by Rule 2, which is

$$SPK = 0.75 * PEAK + 0.25 * SPK$$

$$NPK = 0.75 * PEAK + 0.25 * NPK$$

, where PEAK is the value of the current detected PEAK.
And the thresholds are initialized with the equations

$$Threshold1 = MAXF/3$$

, where MAXF is the maximum amplitude in the first 2s interval of the signal,

$$Threshold2 = 0.5 * MEANF$$

, where MEANF is the average amplitude in the first 2s interval of the signal,

$$Threshold3 = 0.5 * Threshold2 + 0.5 * MEANSB$$

, where MEANSB is a window with the preceding 3 QRS complexes and the following 3 peaks

8.2 Appendix B. Kubios outlier correction algorithm

The steps of the decision algorithm can be found on Figure 33.

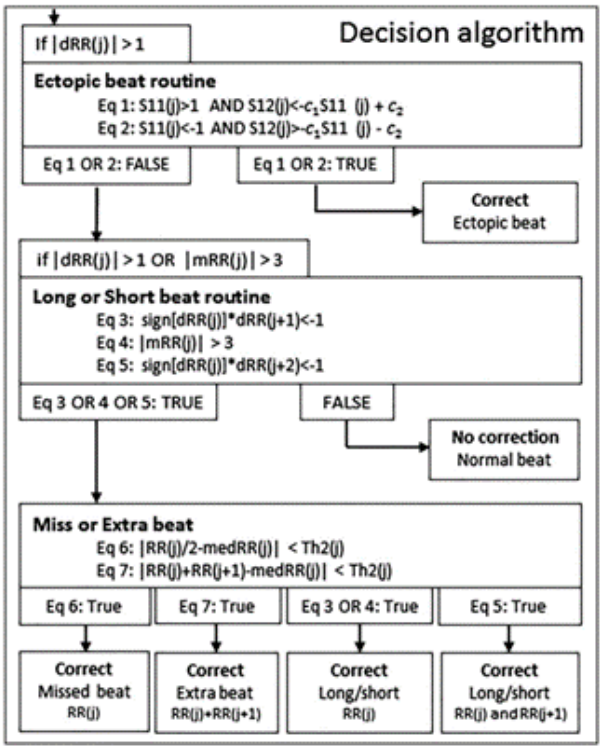


Figure 33: Process of detecting outliers with Kubios method

These steps include some variables. Each can be measured in the following way:

1. dRR is measured and considered a normalised value that says if a peak is abnormal or not.

$$dRRs(j) = RR(j) - RR(j-1), j = 2,...,N$$

$$Th1(j) = \alpha * QD[|dRRs(j-45,...,j+45|], j = 1,...N$$

$$dRR(j) = \frac{dRRs(j)}{Th1(j)}, j = 1,...,N$$

2. An ectopic beat appears in a negative-positive-negative segment (NPN) or positive-negative-positive (PNP).

$$S_1 1(j) = dRR(j), j = 1,...,N$$

$$S_1 2(j) = \begin{cases} \text{max[dRR(j-1), dRR(j+1)],} & \text{if } dRR(j) > 0 \\ \text{min[dRR(j-1), dRR(j+1)],} & \text{if} dRR(j) < 0 \end{cases}$$

$$Ectopicbeat = \begin{cases} S_1 1(j) > 1 AND S_1 2(j) < -c_1 S_1 1(j) - c_2 \\ OR \\ S_1 1(j) < -1 AND S_1 2(j) > -c_1 S_1 1(j) + c_2 \end{cases}$$

3. mRR is calculated from the difference between individual RR intervals and an 11-beat median RR interval. This is done for extra beats and detector errors where every second beat is missing, since this cannot be detected from dRR.

$$mRRs(j) = RR(j) - median[RR(j-5,...,j+5)], j = 1,...,N$$

$$mRRs(j) = \begin{cases} \text{2 mRRs(j),} & \text{if} mRRs(j) < 0, j = 1,...,N \\ \text{mRRs(j),} & \text{if} mRRs(j) \geq 0, j = 1,...,N \end{cases}$$

$$Th2(j) = \alpha * QD[|mRRs(j-45,...,j+45|], j = 1,...N$$

$$mRR(j) = \frac{mRRs(j)}{Th2(j)}, j = 1,...,N$$

4. Long beats (or missing beats) form positive-negative (PN), short beats form negative-positive (NP) and extra beats form NPP or NNP.

$$S_2 1 = dRR(j), j = 1,...,N$$

$$S_2 2 = \begin{cases} \text{min[dRR(j+1), dRR(j+2)],} & \text{if } dRR(j) \geq 0 \\ \text{max[dRR(j+1), dRR(j+2)],} & \text{if} dRR(j) < 0 \end{cases}$$

$$Longorshort = \begin{cases} S_2 1 > 1 AND S_2 2 < -1 \\ OR \\ S_2 1 > 1 AND S_2 2 < -1 \\ OR \\ |mRR| > 3 \end{cases}$$

5. medRR is the median beat interval for 11 beats (5 before and 5 after current beat).

6. Correction of erroneous beats is done depending on the kind of beat:

    - Extra beats: removing the extra R-wave detection corresponding to the detected short RR interval and RR interval series is then recalculated.
    - Missed beats: adding a new R-wave occurrence time so that it divides the detected long RR interval into two equal halves and RR interval series is then recalculated.
    - Long or short beats: interpolating new values to the RR time series. Interpolation means adding the value that makes more sense to the distribution. For example, if there's 0.8 and 0.9 and one value to change in the middle, it would be 0.85.
    - Ectopic beats: replacing corrupted RR times by interpolating RR values.

8.3 Appendix C: HRV metrics equations and figures

8.3.1 Appendix C.1: Time domain metrics

- Mean RR. The equation is

$$\overline{RR} = \frac{1}{\mathrm{N}} * \sum_{n=1}^{\mathrm{N}} RR_n$$

  where N is the total number of inter beat intervals in the sample and

$$RR_n$$

  is the value of the inter beat interval number n.

- Heart rate. It is given by the equation

$$\overline{HR} = \frac{60}{\overline{RR}}$$

- SDNN. The equation is

$$SDNN = \sqrt{\frac{1}{N-1} \sum_{n=1}^{\mathrm{N}} (RR_n - \overline{RR})^2}$$

- RMSSD. The equation is

$$RMSSD = \sqrt{\frac{1}{N-1} \sum_{n=1}^{\mathrm{N-1}} (RR_(n+1) - RR_n)^2}$$

- NNXX and pNNXX. The corresponding relative amount, is given by

$$pNNXX = \frac{NNXX}{N-1} * 100\%$$

Geometric metrics:

- HRV triangular index. [33] We can see it on Figure 34.

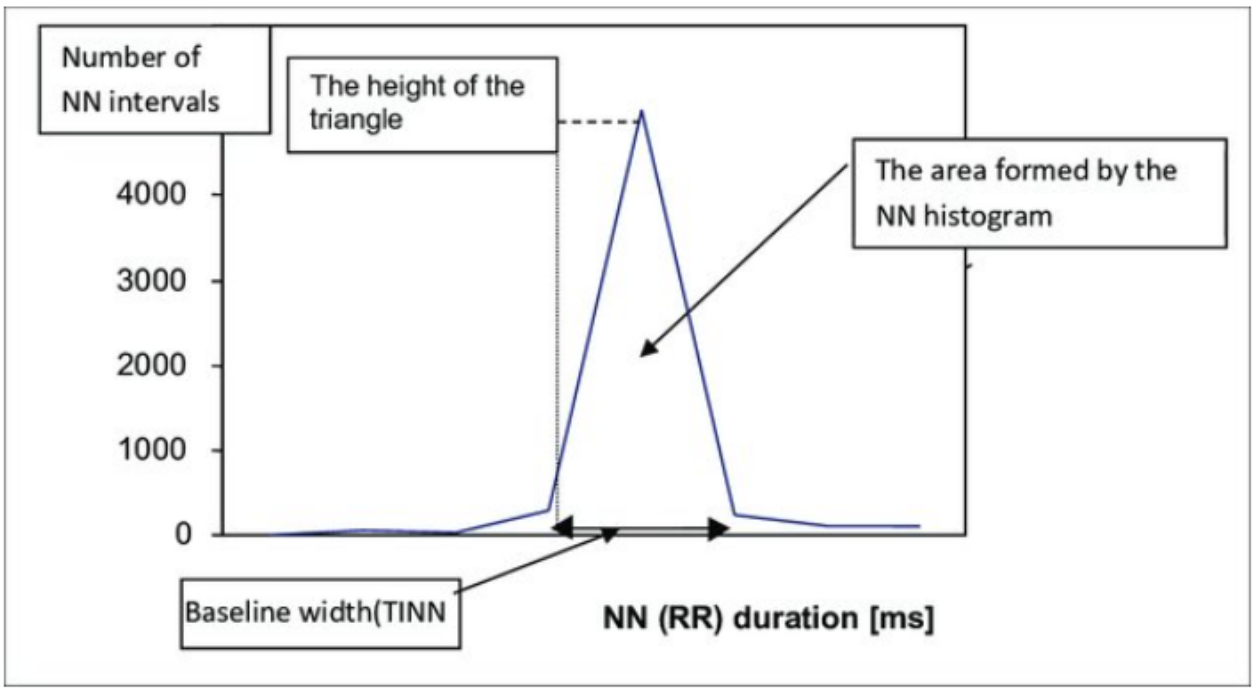


Figure 34: HRV triangular index

- TINN [10] We can see the components of the equation on Figure 35. the sample density distribution D is constructed, which assigns the number of equally long NN intervals to each value of their lengths. The most frequent NN interval length X is established, that is, Y=D(X) is the maximum of the sample density distribution D. The HRV triangular index is the value obtained by dividing the area integral of D by the maximum Y. When the distribution D with a discrete scale is constructed on the horizontal axis, the value is obtained according to the formula HRV index=(total number of all NN intervals)/Y. For the computation of the TINN measure, the values N and M are established on the time axis and a multilinear function q constructed such that q(t)=0 for $t \leq N$ and $t \geq M$ and q(X)=Y, and such that the integral of

$$\int_0^{\inf} (D(t) - q(t))^2 \, dt$$

is the minimum among all selections of all values N and M.

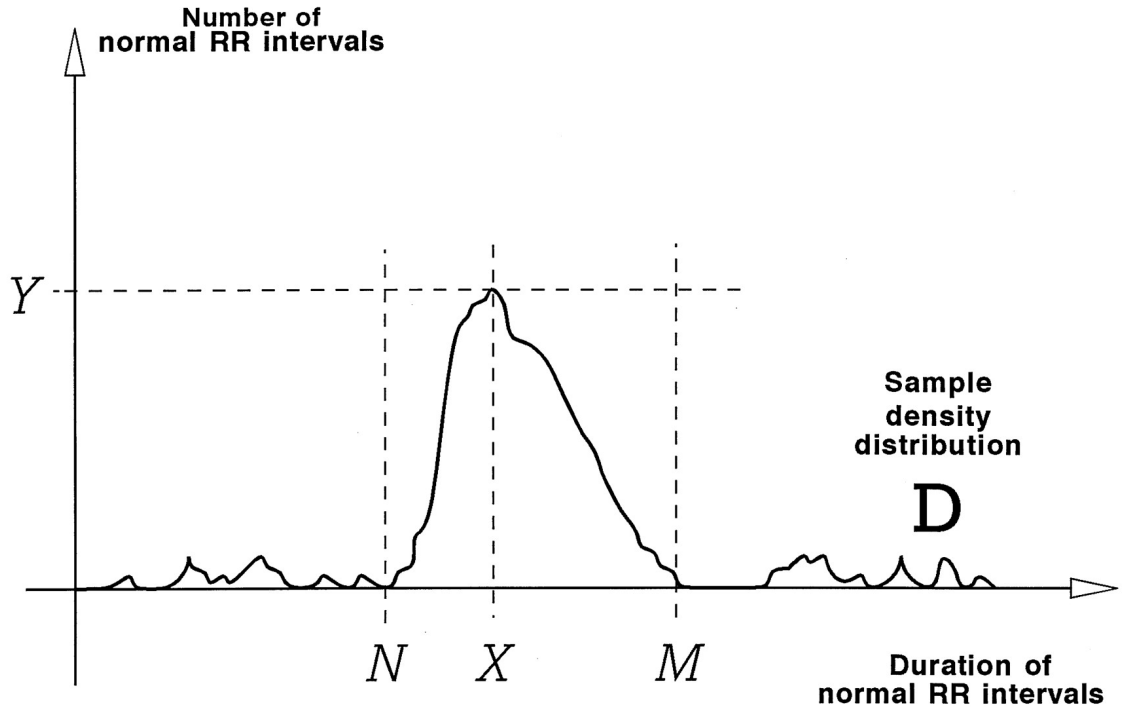


Figure 35: TINN measurement

- Stress index. From Baevsky [5], on Figure 36 [24]. The equation is

$$SI = \frac{AMox100\%}{2MoxMxDMn}$$

The components of the equation are:

  - AMo: mode amplitude in percent. The height of the normalised RR interval histogram with bin width 50 ms.

  - Mo: mode, most frequent RR interval. In this case, it is just the median of the RR intervals.

  - MxDMn: variation scope reflecting degree of RR interval variability. The difference between longest and shortest RR interval values.

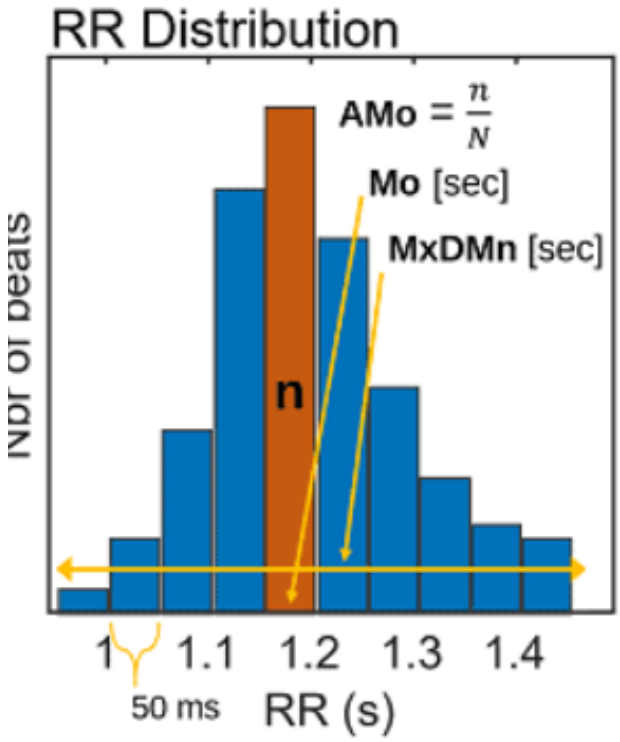


Figure 36: Stress index measurement

Before measuring SI, the low frequency trend is removed by detrending. And once the SI is measured, the square root is taken to transform the SI values to a normal distribution.

8.3.2 Appendix C.2: Nonlinear metrics

- Poincaré plot [22]. SD1 is given by

$$SD1^2 = \frac{1}{2}SDSD^2$$

where SDSD is

$$SDSD = \sqrt{E[\triangle RR_n^2] - \overline{\triangle RR_n}^2}$$

, which is equal to RMSSD for stationary time series.

SD2 is given by

$$SD2^2 = 2SDNN^2 - \frac{1}{2}SDSD^2$$

- Detrended Fluctuation Analysis (DFA) [50]. The first step to get it is integrating the RR interval time series

$$y(k) = \sum_{j=1}^{k}(RR_j - \overline{RR}), k = 1, ..., N$$

RR is the average RR interval.

Then, y_n(k) is measured. It is the integrated series divided into segments of equal length n and after fitting a least squares line into the data. Next, y(k) is detrended by substracting the local

trend within each segment and calculating the root-mean-square fluctuation by doing

$$F(n) = \sqrt{\frac{1}{N}\sum_{k=1}^{\mathrm{N}}(y(k) - y_n)^2}$$

The computation is repeated over different segment lengths to yield the index F(n) as a function of segment length n. Typically F(n) increases with segment length. A linear relationship on a double og graph indicates presence of fractal scaling and the fluctuations can be characterized by scaling exponent alpha. Different values of alpha indicate the following:

- alpha=1.5: Brown noise.
- $1 \leq$ alpha ≤ 1.5: Different kinds of noise.
- alpha=1.5: Brown noise.
- alpha=1: Noise.
- $0.5 \leq$ alpha ≤ 1: Large values are likely to be followed by large value and vice versa.
- alpha=0.5: White noise.
- $0 \leq$ alpha ≤ 0.5: Large value is likely to be followed by small value and vice versa.

The DFA correlations are divided into short-term and long-term fluctuations. The short-term fluctuations are characterized by the slope alpha1 with range $4 \leq$ n ≤ 12, while the slope alpha2 is obtained from the range $13 \leq$ n ≤ 64 and characterizes long-term fluctuations.

- Sample entropy (SampEn) [27]. SampEn is a measure of the regularity or complexity of a time series. It quantifies the likelihood that similar sequences of data points will remain similar when one or more data points are added. In HRV analysis, SampEn is often used to assess the complexity of RR interval time series (the time between successive R-peaks in an ECG signal). Lower SampEn values indicate higher regularity and predictability in heart rate fluctuations, while higher values suggest more complex and irregular patterns.

- Approximate Entropy (ApEn) [57]. ApEn is a measure of the unpredictability of fluctuations in a time series. It quantifies the likelihood that similar sequences of data points will remain similar within a defined tolerance level. Similar to SampEn, ApEn is used in HRV analysis to assess the complexity and regularity of RR interval time series. Lower ApEn values indicate higher regularity and predictability in heart rate fluctuations, while higher values suggest more complex and irregular patterns.

8.4 Appendix D: Requirements analysis questions

Questions planned for PhD students:

- To import data, is it enough just choosing one column as the ECG values? Or is there any other information that can be necessary or useful for HRV analysis?

- In PreCAR, the user can see different graphs and choose between them, including three different ECG signals. Is it necessary to include all these graphs and the 3 signals to do the analysis? Or just one of them?

- Is it better to show ECG, HR or RR? (Considering that the manual correction will be done on the ECG signal).
- If ECG is displayed, just an interval of the signal will be displayed. Is it also useful to see a smaller view of the full signal?
- To do the manual correction, the program will decide what heartbeats are more problematic and show them to the user. Are IBI outliers a good indicator of these problematic beats?
- For the manual correction, would you like to see the problematic heartbeat on a pop-up window, or the general interface is fine?
- To do the manual correction, the user should be able to add a non-detected R-peak, remove a bad-detected R-peak or delete part of the signal. Is there something else needed?
- All the programs let the user choose samples manually, but none of them lets them have automatic sampling (choosing duration of intervals and creating equal samples all over the signal. Would this be useful?

Questions planned for master students:

- Have you used Kubios?
- Have you used Precar and Carspan?
- Have you worked with Cortrium data? And Polar? And other?
- What kind of ECG study did you do? What later analysis did you do with the data?
- If they used Kubios
  - Kubios shows the results of the processing while you are doing it. Do you like to have the results at the same time you do the processing? Or just after you finish it?
  - Would you like an option to detect noise before the R-peak detection?
- If they use PreCAR
  - How long does it take to analyse a signal using PreCAR?
  - If you combine automatic correction with manual confirmation, do you think HR outliers are a good indicator to ask for user confirmation?
  - Why would you prefer to choose PreCAR or Kubios?
- In a new program all the outliers are marked. You see the first outlier and the part of the signal where it appears. You have an arrow to see the part of the signal where the next outlier is. Do you want to have the arrow with the signal analysis options? With the signal graph? Both?
- Would you like to see ECG and HR at the same time?
- If you see ECG and HR at the same time, should they be aligned? Or have different intervals?
- Would you like to have the option of seeing different ECG channels at the same time? Or just one of them but you can choose with?
- Would you like to confirm that you have checked an outlier although you did not change anything? (So, you can see later how many you have left to correct).

8.5 Appendix E: User input during requirements analysis

The following ideas were gotten from the interviews with one of the PhD students:

- Possibility of choosing ECG or HR when importing data, just the option of ECG is not enough. The reason is that there are different sensors that are used to analyse ECG and they do not have the same characteristics. These sensors are Cortrium and Polar Band. With Polar band sensors HR is very reliable, so in these cases some people would prefer the HR correction directly like in Kubios.
- When importing data, always let the user choose a datatime column. If this does not exist, it is necessary to create a time index column. This is useful in cases when a part of the signal has to be deleted later, so it is clear that there is a gap in the values.
- There are 3 ECG channels because some sensors have 3 channels, so you can always get the most reliable signal out of 3. If ECG is selected (not HR), let the use choose more than 1 column in case they use more sensors. Include the 3 signals but just show one of them with the option of choosing or seeing more at the same time. There might be signals in which for a certain period one ECG is better but later it changes.
- It might be useful to plot ECG or HR at the same time, but not necessary. Deciding what to plot depends on the user and it is better to allow as much flexibility as possible. One person may just have ECG, other just HR and other both at the same time.
- It is very important that the graphs to see ECG and/or HR have an easy way of zooming. Using zoom in Kubios is not very convenient and this would be a useful improvement. Also, scrolling to the left or right in the signal should be easier.
- An IBI outlier is a good indicator to see when to manually check the signal. This correction should just be adding a missing R-peak or deleting an existing one. The way these outliers can be shown to the users can be in the same general interface where the ECG is plotted, with a button to go left or right to see the previous or next outlier.
- Noise detection in the signal might be useful, but not fully necessary if users can check IBI outliers and correct them manually. Like explained before, a time column should be included in case an interval in the signal is removed.
- The results should be downloaded on a csv file. This should include the option of seeing raw or normalised results (results of each sample divided by results of the full signal) or the values for the full signal directly. The user should be able to choose what they want to get.
- Option to aggregate different files. Sometimes the signal needs to record for many hours but it losses connection every some time. Now, the way of analysing this is by analysing each file individually, but it would be better to be able to integrate them as a single file. This avoids too much user effort but also improve the noise detection and outliers detection.

The following ideas with the interview with the other one:

- Option to choose between the 3 ECG channels, but then just plot 1 signal.
- Seeing ECG and HR at the same time would be nice. Also having different scales would be useful but making sure it is clear what time interval each graph has at every moment.

- An IBI outlier is enough indicator for needing a correction if signal is clean. Otherwise, there might be a lot of outliers for one kind of noise and other noises that should be considered outliers are not.
- If the ECG signal is reliable, users prefer to use Kubios instead of PreCAR. The problem with Kubios is that, if ECG is not so reliable, they do not know what the outliers that Kubios is removing are. But if the signal is clean, they can agree that the outliers will be problems and can remove them. ECG is more useful; HR is used just if the signal is clean.
- If the signal is more problematic or has more noise, a noise detection step before the analysis would be very useful. With this, users can delete intervals of the signal and have a clean ECG. When detecting R-peaks and checking for outliers, they know that outliers are really problems that they have to correct.
- Having an automatic sample selection is useful. Normally, samples are very similar but they have to be chosen individually (in PreCAR and Kubios). Saying the duration of the sample, how many repetitions it should have, time between repetitions and overlap between them may save same time for the user in the analysis.
- Seeing results at the same time as the user is doing the pre-processing like Kubios does is unnecessary.

And the input from master students was:

- About seeing results while doing the processing like in Kubios, there where different opinions. Some users said that it made no sense and it would be better to have a bigger part of the screen to see the signal, so results should just be shown after the processing. But others said that they preferred to see the results so they knew the effect of the changes they were doing.
- Users liked the option of detecting noise if it is accurate. But they would always want to check this noise to see if it is really noise and then decide what to do. In case there is noise, they could delete this interval.
- The time that analysing a signal in PreCAR varies a lot for each user, but they all agree that it is too long.
- All users agreed that HR outliers are a good indicator to check what parts of the signal should be corrected.
- Users would use PreCAR in short signals. They all agree that the analysis is more accurate than with Kubios, but it takes too long.
- Having the option of saving the processing progress to continue with it later was also mentioned among different students.
- About the design, there were different opinions. Some students find Kubios has too much unnecessary information and it is hard to start with it, so they would prefer something more simple like PreCAR. But others appreciate the user-friendly interface of Kubios.
- For the question on where to see the arrows to go to the next outlier, students preferred to have this just on one place, as duplicate can be confusing. It was preferred in the analysis place and just say the outlier title on top of the graph.

- ECG and HR at the same time was the best option for many students, although some of them said just ECG would be enough. They all agreed that it was unnecessary to have the option of seeing different ECG channels at the same time. For this, probably the best option is to give the freedom of choosing what to display.

- About aligning the graphs, all students said that they should be aligned. When asked if the option of having different intervals would also be useful, they were not very convinced, but they agreed it may be nice to have it as long as the option to align the graphs is still there. And, if they have a different interval, it should be clear so there is not confusion.

- Checking when and outlier is correct to see later that it is checked was considered useful by all the students.

8.6 Appendix F: Full list of use cases with explanations

- Opening the file The user will need the ECG signal or the HR (also option for RR in case). The options to include on this step are describe in the data functionalities.

- Seeing the signal with possibility of zoom or change display (HR, RR)

  Zooming should be easy just with the fingers. This is something that other programs make more complex. For example, in Kubios you need to choose the zoom tool and then select an interval. Moving in the signal should also be easily done by scrolling.

  To change what signal to show, a small place can be saved like in PreCAR in which the user can check what signals they want to see.

  Button on top right to make signal screen full-screen. When this, button to reduce it again.

- Choosing samples

  Avoid the necessity of choosing every single sample individually. Option of saying beginning and end, space, between samples, repetitions and overlap percentage.

- User selects a threshold to see IBI outliers highlighted

  The threshold is a percentage and the IBI values that are lower or higher than the threshold get displayed.

  One way of displaying them is to see the number of outliers just under that threshold choosing option and saying how many have been corrected. Also, include an option in the main plot to go directly to the next outlier (and says current selected outlier: outlier n out of m, with arrows to go before or after). If clicking on the n/m text, you see a pop-up list with all the outliers and a tick if they have been corrected. You can go to one of them by clicking on it.

- User sees the general ECG signal and chooses an interval to correct

  Sliding window to know what part of the full ECG is being displayed at the moment. Under the main plot, you see 0 on the left and the signal duration on the right. You see 2 circles that delimit your interval window, and you can move this circle. If moving the line between the circles, you move the full interval (so, duration is fixed). There are also squares to specify the start and end of the current window.

- User sees an interval and corrects the heartbeat

  The heart beat can be corrected on the same window or a pop-up. It makes sense that is on the same window. User can add r-peak, delete, or confirm, so outlier is done.

- Noise detection

  There is a button to check for noise in all the signal. Before r-peak detection, the user can see if removing a part of the signal.

  Intervals working in the same way as HR outliers, so the user can go to the next one and manually edit it. Option to increase ending or starting point by moving the window axe.

- User wants to get csv of results

  Possibility of choosing what results to get. Raw or normalised, what variables and how to show them on the file.

- User wants to save current analysis to continue later

  Possibility of doing this, but considering what can be saved in the initial csv file and if the import should change.

- User wants to get a pdf of results in plots

8.7 Appendix G: Full user evaluation results

8.7.1 Appendix G.1: Functional requirements

- Give the user full control of the signal
  - Providing useful and visible annotations. Annotations are shown for detected or corrected peaks with a red circle, outliers with a vertical line, the previous RR (before any correction) with a lighter black than the current RR, noise blocks in black and sample blocks in blue. All these annotations could be easily identified by the user. Just the possibility to improve the visualization of the previous RR was mentioned. This is the same color as the grid, so it can be hard to see. Changing the color or even not showing the grid should be considered to make this line more visible.
  - Possibility to edit any annotation as wanted. This can be done with the edit buttons and shortcuts. There are options to edit peaks (add, delete or interpolate), edit noise (add, delete or edit) and edit samples (add, delete or edit). Nothing was missing for the users. But the shortcuts were not so intuitive in the beginning, as sometimes CTRL or SHIFT buttons are involved. To make it more intuitive, just left and right click should include all the options. Depending on if editing peaks, noise or samples, a different action should be performed.

    The shortcuts and manual editing options were preferred by the users instead of not having the buttons. Although they need to click on the button and then do the correction, this was preferred instead of being able to do everything at the same time. The reason is that they would need to use many different shortcuts that included editing noise, peaks and samples together. Per separate, it is just few simple shortcuts. And they also felt they would make less errors by having the buttons. But the shortcuts should be improved to be more intuitive. Creating a sample or noise should just be done with one click, starting

the sample on the mouse down and finishing on mouse up. Also, with the current version of the program, it is just possible to edit samples once the current block is added, but this was misunderstood by users. If they have the sample editing option on the graph, it is confusing to try to add a sample and not see anything appearing on the graph.

- Easy zooming and scrolling and Possibility to synchronize and desynchronize graphs. It was highly valued that the user has much control over what to visualize in the graph. The options of synchronizing or desynchronizing the graphs got good feedback, as well as zooming in and out so easily. There were also some comments of possible improvements. The contrast between the current and the original RR corrections should be more visible. And the separation of different samples in the same group should also be more visual.

- Include all the automatic analysis options

  - Noise correction, outliers correction and sample selection. The program includes all the options that the user would need to do the analysis. And the accuracy in general is good enough to help the users making the analysis faster while not missing important corrections.

    But it was sometimes hard to understand the analysis options. Users started focusing more on the graphs and the manual correction before using the options on the left of the screen. This may just be that they were trying to get familiarised with the different options before starting the analysis, but it would be useful to make sure the automatic analysis is used before the manual correction, as this is the purpose of the program.

  - Possibility to choose settings of each analysis step and apply with just a click. The function of each analysis step can be easily recognised with the icons. It is easy to use or not to use with just a click and more details on how each step will be used can be selected on the settings.

- The user can immediately see each change that was done automatically, and modify it when needed.

  - Possible to annotate the changes done manually and go to each of them in the graph. The idea of having the selector to mark the outliers or noise and instantly seeing them was very liked by all users. When asked if they would trust it and if they would save time with it, both answers were positive. Some users said that they sometimes do not do a full check of the ECG signal to correct outliers and they rely on the automatic correction, due to lack of time. But with this, the manual correction would be very fast, so they would always do it. About the possibility of just checking the outliers and might miss necessary corrections, they said that this can happen anyway with a manual correction. Users can be scrolling the ECG signal and accidentally skip one of the needed corrections. But with this program, it is also easier to go through the signal than with any other HRV program, thanks to do option of easily zooming in or out. So, although they wanted to manually see the full signal, this would be easier to do with EZCardio.

    However, some users did not consider the selectors when opening the signal and started visually scanning all the signal. This means that the distribution of the components in the interface is not optimal, as this can be missed, also because no other program has something similar to the selector, so users are not used to use it for the analysis. After few beats corrected, they started using the selector, so it is not a big issue. But it would be

better to improve the distribution to make sure users easily see it and understand what it means.

- Possible to manually delete or edit any of these changes. This can be done with the same editing shortcuts explained before.

8.7.2 Appendix G.2: Data requirements

It is possible to import ECG or RR files in different formats: delimiter-based files and edf. Some options can be specified to choose what to import from the file and feedback is provided as the user fills the options, showing a signal preview.
The import page was liked in general, although more negative feedback was given here. Some people found the options to choose the data confusing, as it was not clear what each of them should represent on the data. One comment that was mentioned is that it would be better to have some kind of order for the options, so the user sees first the most important options. For example, the first input should be if importing ECG or RR. And the option to import respiratory data was also missed by some people.

8.7.3 Appendix G.3: Environmental requirements

- File output. The option to get the results was fine in general, as it just outputs everything that can be useful. But it was also mentioned that it would be better to have an option of exporting just one results. Normally, researchers just want to use one or few results, and they need to process the results file that has everything in order to get just what they want. Being able to choose a single or just some of the results would make this step easier. And, although not so important, some people would also like to export the figures that can be seen on the results preview.

- Information buttons. They appear next to each option in the program and it is also possible to open the documentation from the program.

- Technical environment. Mac and Linux versions of the program are created. Many users have Mac computers, so also having a version for it was convenient.

  And the program can be used in any kind of computer. The size is around 120 Mb, which is something any computer can handle. And algorithms in general are quite simple, not using any complex computation.

8.7.4 Appendix G.4: User characteristics

- Self explanatory. When opening the main screen, users find it easy to identify the different options. The general structure is similar to other programs, so it was not hard to see what each component represented. When looking for certain options or trying to do something, they could quickly find it on the screen.

- Minimize computer procedures. For the analysis they just need a file to import in the beginning and after doing some easy steps with the program, they will get a new file that they can use for further analysis. To even minimize more the computer procedures, more file types should be allowed in the import.

8.7.5 Appendix G.5: Usability goals

- Hide or show parts of the screen. Resize buttons were easily recognised by the users. They could hide or show the results and the analysis options.

- Choose many samples at once. The sample selection was also hard to understand. The most common way of selecting samples is by manually adding them, one by one, so the option to add many at the same time was not understood in the beginning. Adding different sample blocks with different names is also new, and users though that blocks referred to individual samples. This should be made more clear. After explaining how the sample selection worked, it was easy to understand and use, and users said it was very useful to add blocks instead of adding each sample individually.

  There were also some comments about this component. First, some people though that using the term ”sample” can be confusing, as this can also refer to the data points of the signal. It would be better to say ”segment”. And ”minimum sample size” was misinterpreted as the number of participants in the experiment, so saying instead something like ”minimum duration” would be more convenient.

  Furthermore, people also find useful to see how many outliers there were at each single sample, and get a warning if the number is big. There was another comment saying that it would be useful to see some stationary values, as the RR intervals should not be stationary in order to get reliable results. EZCardio currently solves this by letting the user detrend the RR values, but it would also be possible to show some values about stationarity in case some people find this useful.

- Save analysis to import later. One of the options in the menu lets the user save the analysis.

- More import options. The other options can be easily identified in the import page.

- Append results to existing file. This can also be done from the menu and is easily understood.

8.7.6 Appendix G.6: Heuristics evaluation

- User is always informed about everything that happens in the program.

  This was taken into account when adding the loading boxes. Some actions take time to complete, so the user needs to know that the program is busy computing and they will see the result of the action when the loading box is closed.

  This principle should be improved when selecting a sample. As mentioned before, the sample selection was confusing, specially when users tried to add a sample manually. They want to see the sample added, but this just happens when the current sample block has already been created.

- User control and freedom.

  The program was build taking this principle as the most important one. The user can easily control the analysis by applying automatic corrections and easily modifying them when necessary.

  To allow more user control, it was also possible to do two outlier correction at the same time. The user can choose to use the algorithm and then a threshold correction if they want to see all possible outliers.

However, it is also important not to make assumptions about what option the user would prefer and try to give as many options to choose as possible. For this program it was assumed that using Pan Tompkins was the best beat detection method, as well as the outliers based noise detection to identify noise. Although these methods have many advantages over alternative ones, users may also want to have the option to choose a different method. Different possibilities could be given so the user can choose.

Another important point is that the selector currently just has the option to go to the previous or next outlier or noise interval. There should be the option to go to a particular one with for example a dropdown button. Some kind of annotation could also be added to identify if that annotation has already been seen or to let the user identify when they want to check it later.

- Consistency and standards.

  The consistency makes sure that similar actions lead to similar purposes. For example, in the program we can control noise and sample intervals in the same way, with the same shortcuts and same options. We also see the same selector to mark outliers or noise intervals and check each one of them.

  It may not be so consistent to have a different selection option for the two outliers correction. If the user wants to use both corrections and then manually check the outliers, they will just be able to see a part of the total outliers at once. It is better to see all the outliers to have a better understanding of what the full correction is doing.

- Minimize user working memory.

  This is done by always showing in the screen the information that the user wants to know. In this program there is not much that the user needs to memorise. But the shortcuts could be improved. With the current version, this can be seen by keeping the mouse in the corresponding editing button. The shortcuts of the selected option could remain on the screen, so the user does not need to remember them. They can be shown on the bottom left of the screen, where some details are shown sometimes. Although this might not be necessary if the shortcuts are simplified, and just left or right click are intuitively used to add/delete.

- Minimalist design.

  This was highly taken into account when designing the program. Other HRV programs show too much information on the screen at the same time, some of which is never used by most people. A frequent comment during the requirements analysis was that they saw too many buttons in the programs they used and did not really know what some of the buttons were meant to do.

  In EzCardio, just the necessary options are shown. It is possible to set certain details about the analysis in the settings, but in the main screen just the necessary options are shown.

- Useful error messages and documentation.

  Many errors that can happen when using the program are identified and the user sees a notification when they happen. Also, it is possible to see the documentation to see more information about the analysis. The documentation is intended to explain detail about the analysis, so the user can understand what exactly is happening. Ideally, the user would also read the documentation before starting to use the program, so also some tips and the steps to follow are recommended. But often this does not happen and the users starts using the program directly. Anyway, the program itself is also intended to be self explanatory and easy to understand even without having read the documentation.

8.8 Appendix H: Full explanation of future usability improvements

- Selector more centered and with color.

  The selector is one of the most important usability advantages of EZCardio and seeing it from the first moment you open the program should be necessary. As participants missed it in the beginning, it is necessary to highlight it somehow. The first change would be to show it more centered on the graph, as now it has two big buttons on the left and it appears on the right side. The buttons still need to be used, but they can be made smaller. Also, the left and right arrows that represent the selector should be smaller so it is clear they represent the option to go to the previous or next outlier or noise.

  Blue color should also be included to get user attention when opening the program. In the beginning, it can say ”Use noise and outliers correction to mark them” so the next thing the user sees is the analysis options on the left. This way, the user first focuses on the selector, then on the left options and later on the signal. Seeing these 3 things in this order is the best way to understand what the program is about and how you should proceed with the analysis.

  Additionally, just two annotation options should be shown: one for outliers and one for noise (instead of two for outliers and one for noise). This can be done with a more convenient design than the dropdown button, for example with two buttons on top of the selector. This would be disabled when no automatic analysis has been done. Once enabled, they would follow the blue for available but inactive and red for activated, to be consistent with the rest of the interface design.

  We can see a possible new design for this part in Figure 31. Here we show the mentioned changes. Also, the edit buttons row is moved to the bottom, so the selector is the only thing on top and the first component the users will look at. This way the resize buttons or also moved to the corners, so it is easier to understand them. And the icon is modified to make clear their action will be to expand.

  To highlight the selector even more the first time the users sees the program, we can set the default noise and outliers correction to ”Medium”. The user can set the default correction, but the first time they use the program, this will be set to ”Medium”, so they can already see the outliers and noise marked.

- In the import page, improve the understanding of each import option.

  The first change could be to change the order in which the options are shown. But what would really help here is to show what each option represents in the data preview as you change it. This is an option that thee programs like Kubios have. When setting the header lines, we can see should show the starting row in the data preview highlighted. And when changing the data column line, we can also highlight this column on a different color. And the same for the time column. For example, in Figure 32 we see the data preview when we have chosen 1 for ”Header lines”, 4 for ”Data column” and 1 for ”Time column”.

- Changes related to samples

  To avoid users wanting to manually add samples before a sample block exists, we can disable the ”Edit sample” button when the current sample on the left options has not been added. If the users tries to click on this button, they see a message saying that they can just manually edit samples when they have added a block.

Then, the recommended usability changes should also be implemented. This are saying segments instead of samples, duration instead of size and showing more visible in the graphs the start and end of each sample.

For the outliers and stationarity values of each sample, they cannot be added on the left analysis options, as there we just see blocks, not individual samples. However, on the results preview we are checking each sample individually. We could add another results button that shows general variables, like in this case ratio of outliers, stationarity and some others. But if there are just two values, it is unnecessary to add a new full section, so they should be added somewhere else. They could be displayed on another table before the time results one.

Like we do not expect the user to go to each sample to check these values, we also need to show a warning when any of these values reach a threshold. This should be done just after the sample block is created. And the warning message should remain, so the user does not need to memorise which are the problematic samples and they can just see it on screen. This can just appear as an extra row in the sample options.

The last problem was that sample blocks were misinterpreted as single samples. To avoid this, we can show on top ”Add sample block”, instead of just samples.

- Option to just export one result

  When choosing one of the export options, a pop up window opens. There we can see a checkbox that says ”Export all” that is active by default. There are also some lines that we can click to add results one by one. The output file shows three rows and multiple columns. The first row is the sample name, the second is the start and end time, and the third is the value.

- Option to mark or correct outliers

  To give the user more control on the analysis, we can also enable an option to just mark outliers. On each outliers option, there is a checkbox that says ”Also correct outliers”. Then the block names should also change to ”Outlier detection” instead of ”correction”. This can be preferred by some users specially when doing double correction, with the algorithm and the threshold.

  The same change could also be implemented for noise intervals. However, noise intervals are not deleted, they are just marked and the part of the signal omitted unless the interval is deleted. So, the difference between marking and correcting noise would just be user confirmation, which is an additional task for the user.

- Adding time varying results

  Time varying results shows how certain variable changes along the signal. This can be seen on the results file, by comparing the values for different samples. But having a preview of this change in the program can also be useful. So, users can have an initial view of the tendency of that results before exporting the file and doing the next steps.

8.9 Appendix I: Reported bugs in the program

The program was evaluated in Windows and Mac devices and it works for both. However, the original implementation was done in Windows and some issues are seen on Mac.

- Opening a new file in Mac

After having opened one file in EZCardio, we have a menu with many options on top. We can click on “Menu -¿ File -¿ Open new file” and we should see the import page again. This works in Windows, but on Mac it just closes the program with a crash report.

- Loading box not visible in Mac

  Some processes in EZCardio take time, like the beat detection or outliers correction. So, while these are happening, it is convenient to show a loading box that notifies the user the process is still going on. However, this box does not appear on Mac devices.

- Need to adapt formats for Mac

  Although not a bug, it would be necessary for usability. Some figures or text do not have the proper size.

Some other secondary issues are seen in the program, problems that happen on any operating system.

- Grid on second screen disappears

  When opening the program on a second screen, like a monitor, the grids disappears of the ECG and RR graphs disappear. Although the axes and annotations still work the same way and the grid is unnecessary, so this is not a big problem. Even not using the grid at all could be considered.

- Scrolling with keypad

  Scrolling with the keypad (using double fingers) makes the graph zoom, while is should move the current window to the right or the left.

- Slider not completed when full signal

  When zooming out to see the full signal, the slider does not show a full range, but it misses the first part.

- Not fully aligned segments between ECG and RR graphs

  There can be not very important issues similar to the last one. For example, noise takes a bit to adapt to the exact interval where it is when zooming out. Or adding noise at the end of the ECG graph shows it a bit earlier on the RR graph. The values are still good, it is just a displaying problem, so it is also a secondary bug.