sciendo
Polish Journal of Medical Physics and Engineering
The Journal of Polish Society of Medical Physics
ISSN 1898-0309, doi: 10.2478/pjmpe-2025-0002

March 2025
Vol 31, Issue 1

Technical Note

CoRRection – an open source software tool for RR intervals processing

Magdalena MIKIELEWICZ [1,ACD,*], Jakub Sławomir GĄSIOR [2,ABCEF], Marcel MŁYŃCZAK [1,AEF]

[1]*Faculty of Mechatronics, Institute of Metrology and Biomedical Engineering, Warsaw University of Technology, Warsaw, Poland*
[2]*Department of Pediatric Cardiology and General Pediatrics, Medical University of Warsaw, Warsaw, Poland*
[*]*Corresponding author: Magdalena Mikielewicz; mikielewiczmagdalena@gmail.com*

(received 4 October 2024; revised 25 November 2024; accepted 5 December 2024)

Abstract

Introduction: An analysis of Heart Rate Variability (HRV) is widely used in clinical and research. To properly calculate HRV parameters, the RR intervals series must be properly preprocessed. There are already automated tools for correcting these artifacts; however, they are not fully transparent and fully customizable. To address these limitations, we introduce CoRRection, a semi-automatic tool for RR interval correction, which integrates both automatic and manual approaches providing greater control over which intervals are corrected.
Material and Methods: The application offers detection and correction methods. Additionally, it allows to manually remove an artifact before applying a correction method. It also allows to clean artifacts in shorter segments which enables applying different detection methods in one example, e.g., gathered during intense exercise. The application provides a test to assess signal stationarity (Augmented Dickey-Fuller test). After signal processing the report is created, containing the number of probes deleted and modified.
Results: The proposed CoRRection application allows both technical and non-technical users to preprocess RR signals with a better control over the process by enabling the use of semi-automatic approach. A few examples of studies which required Correction's unique approach of semi-automatic correction were presented. The need for different methods and not only considering different examinations, but also different segments within one recording was emphasized.
Conclusions: We have presented a new application designed to preprocess RR intervals signal with a few popular detection and correction methods. This approach enables a good compromise between a precise manual approach and a less time-consuming automatic approach. The semi-automatic method of artifact correction empowers users to explore multiple identification and correction methods (and to choose the one best fitted to the data or measurement conditions), and offers a better understanding of the examination preprocessing tools. This may also result in a better trust in the automatic approach among the users.

Keywords: RR intervals; biomedical engineering; artifact correction; biosignals.

Introduction

Since 1996, when the Task Force document on heart rate variability (HRV) was published,[1], interest has grown not only in the clinical application of HRV quality of data prior to the signal processing performance.[2] One key concern is the presence of ectopic or aberrant RR intervals, which can directly affect the analysis. Available applications (e.g., Kubios[3], easieRR[4] or ARTiiFACT[5]) offer automatic detection and correction methods for these intervals. The automatic editing of RR intervals is convenient and can perform adequately, especially in time series with low variability. However, many experts agree that a careful selection of the editing method is crucial.[4] The lack of transparency in how algorithms are implemented can lead to reluctance in using automated tools[4] and may hinder the replication of study methodologies.[6]. Additionally, applying the same methods to signals collected at rest may yield slightly different accuracy compared to their application to signals gathered during high-intensity exercise.[7] The age of participants is another key factor to consider when choosing the appropriate artifacts correction method.[8] Leaving artifacts undetected in the RR interval time series may introduce bias into the interpretation of HRV values. Therefore, selecting the right correction method is essential, as it can significantly influence the accuracy of HRV calculation.[7]

This paper aims to present CoRRection, an application designed to enable semi-automatic correction of artifacts in RR interval series. The semi-automatic approach combines established automatic methods for artifact detection, with

Authors' contribution: A – Research concept and design, B – Collection and/or assembly of data, C – Data analysis and interpretation, D – Writing the article, E – Critical revision of the article, F – Final approval of the article.

adjustable settings, and enhances them with the ability to manually address artifacts. Users can correct excessive artifacts flagged by the automatic system or add artifacts overlooked by it. This approach improves the artifact correction processes, offering the efficiency of automation while ensuring greater control over which intervals are modified.

Material and Methods

Artifacts detection methods

Giles and Draper (2018)[7] proposed six types of artifacts that can occur in the RR interval signal, as described in **Table 1**. However, since the analysis relies solely on the registered series of RR intervals, it is not possible to define an algorithm capable of automatically identifying artifacts of types T4–T6, which require electrocardiography (ECG) signal.[7] Consequently, it was decided only to define the methods of detection of artifacts of type T1, T2, and T3.

For T1 artifacts, testing the application on data collected under various conditions (rest, physical training, post-exercise recovery) revealed that the initially suggested threshold of 20 ms was inadequate for identifying artifacts and had to be rejected. This change is motivated by the rapid heart rate change that can occur during physical activity. To address this, an interactive field was introduced, allowing users to adjust thresholds based on the physiology of each individual and the conditions of the study. For simplicity, it was assumed that an artifact cannot belong to multiple categories simultaneously, with T1 artifact type being assigned the highest priority.

Additionally, for T2 and T3 artifacts, the condition of avoiding disturbance to two adjacent points was not enforced. Application testing revealed that artifacts rarely occur as single samples and are more commonly found in sequences of few consecutive samples. Therefore, the condition was almost never satisfied. The final list of artifact types distinguished for the this work is presented in **Table 2**. While the threshold values for identifying artifacts should ideally be customized for each participant to account for individual physiology, a default value has been proposed for general use.

In addition to detecting T1–T3 artifacts, the application enables users to detect artifacts using few popular algorithms:

– Algorithm proposed by Tarvainen and Lipponen for the Kubios application.[9] It allows for the detection of ectopic, missing or extra beats, and intervals that are too short or too long. Algorithm relies on several equations applied to each data point, which are discussed furtherer in Technical Appendix. In the application, artifacts identified using this method are labeled as "Tarvainen," after the first author.
– Algorithms proposed by Piskorski and Guzik, which are based on filtering Poincaré plots.[10] Algorithms offer several filtering options for artifacts identification: annotation, square and quotient filtering. The annotation filter was excluded from the application, as it requires annotated recording. Quotient filter is considered to be more aggressive than a square filter. The formulas behind those algorithms are described in Technical Appendix.

Moreover, the application enables users to mark artifacts that do not meet the criteria for classification as T1-T3, Tarvainen, Quotient or Square. These artifacts are categorized under the label "Other."

Table 1. Table presenting artifacts defined in the literature (as per D. A. Giles, 2018[7]).

Artifact type	Description
T1	A difference greater than 20 ms at a single interval.
T2	A long interval, followed by a short interval with the two points on either side were unaffected.
T3	Short interval, followed by a long interval with the two points on either side were unaffected.
T4	Missed interval on the HRM (Heart Rate Monitor) that may be equal to two or three ECG RR intervals.
T5	Extra or short RR intervals from the HRM that may be equal to one on the ECG.
T6-a	RR interval(s) omitted by the HRM, undetectable.
T6-b	RR interval(s) omitted by the HRM, detectable

Table 2. Artifacts defined in the correction application.

Artifact type	Description
T1	A discrepancy greater than 200 ms (by default – editable) at a single interval.
T2	A long interval, followed by an interval shorter at least 400 ms (by default – editable).
T3	Short interval, followed by an interval longer at least 400 ms (by default – editable).
Tarvainen	Ectopic/missed/extra beat or interval too short or too long detected with a use of Tarvainen’s algorithm.
Quotient	Interval detected with a use of quotient filtering.
Square	Interval detected with a use of square filtering.
Other	An artifact not meeting any defined conditions , not easily characterized.

To illustrate the use cases for the CoRRection application, tests were conducted on existing methods. In **Figure 1**, it can be observed that while the Tarvainen method primarily identifies real artifacts, it occasionally misclassifies correct probes as artifacts when surrounded by a few erroneous samples. **Figure 2** shows an example where the quotient detection method was applied. This method also has a tendency to overmark probes as artifacts. The square filtering method detects artifacts when a probe differs too much or too little from the next one. However, as shown in **Figure 3**, this method should be used cautiously. During an examination where a patient performed high-intensity exercise, most of their RR intervals were incorrectly marked as artifacts due to insufficient variation in the signal.

Artifacts correction methods

Several methods can be used to handle the correction of RR artifacts:[5,7,11,12]

- Removal: incorrect RR intervals are removed from the time series. This operation can significantly impact the calculated HRV parameters due to the shortening of the signal. Short-term studies and their frequency-domain parameters are most sensitive to this method.[7]
- Interpolation: replaces data marked as artifacts with new, interpolated values, preserving the overall length of the study.[13]
  - Zero-order (mean): entails substituting the artifact value with the mean value calculated in the surrounding RR interval set. For longer segments of the study containing artifacts, this method assigns the same value for the entire segment, resulting in a flat shape of the signal in that time interval, introducing false trends.
  - First-order (linear): a straight line is determined in irregular intervals to obtain new values. Similar to zero-order, inclined shapes may appear on longer segments of artifacts, introducing false trend.
  - Cubic interpolation: utilizes 4 reference points to calculate a polynomial. This type of interpolation does not result in flat signal segments. The use of this method may have a negative impact on the result due to introducing false signal correlations, as the series of heartbeats is a complex non-regular signal.
- Moving average: a specific type of average calculated in consecutive windows of a specified length moving along the signal. The final result is the average of the arithmetic means calculated for successive windows.
- Spline, cubic spline interpolation: values are estimated based on multiple reference points by fitting a third-degree polynomial. Similar to cubic interpolation, spline interpolation may also introduce false correlations into the signal
- No correction (means, that an artifact was not detected by any used method).

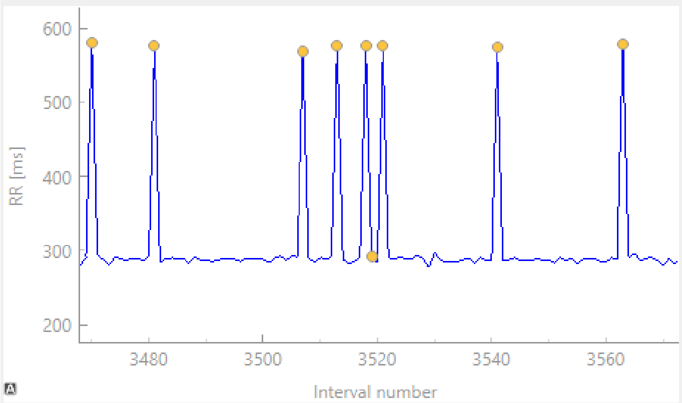

Figure 1. Artifacts detected using Tarvainen identification algorithm.

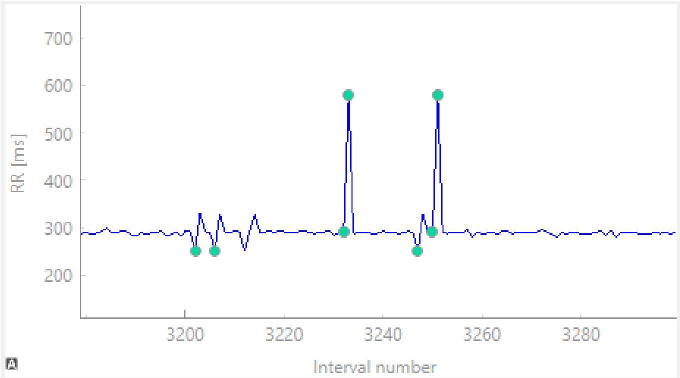

Figure 2. Artifacts detected using Quotient filtering method.

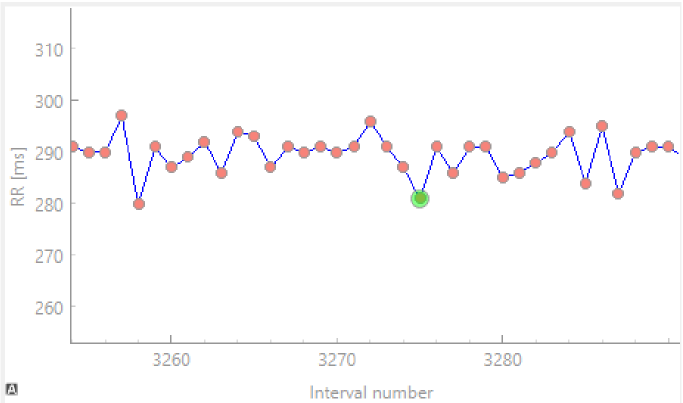

Figure 3. Artifacts detected using Square filtering method.

The application offers four artifact correction options from the list above, plus one additional method:

- Linear interpolation,
- Cubic spline interpolation,
- Deletion,
- Moving average algorithm with a window length of 7,
- PreMean.

The cubic interpolation method was not utilized as it is very similar to cubic spline interpolation but less commonly used. Additionally, zero-degree interpolation was excluded, but a new technique also based on the mean value has been proposed.

The previously mentioned correction methods mostly rely on finding replacement values using both preceding and succeeding values, with the exception of artifact removal. The newly proposed correction method, termed “pre mean,” utilizes the average of the intervals preceding an artifact. This approach is based on the idea that subsequent heartbeats should not influence the value of a given interval, whereas preceding intervals might. The “pre mean” approach allows users to specify the number of preceding intervals considered, ranging from 2 to 10, based on user’s preference.

Stationarity assessment

The goal of preprocessing RR intervals is to prepare them for HRV analysis across time, frequency, and non-linear domains. For spectral analysis, the Fourier transform is used due to its computational efficiency. However it is sensitive to unevenly spaced RR interval and requires the signal to remain stationary (i.e. statistical quantities do not change over time).[14] To overcome the challenge of unequal distances between signal samples, artificial interpolation is applied.[15]

The prepared signal can be subjected to e.g. spectral analysis only if stationarity is confirmed.[14] Stationarity can be assessed using the Augmented Dickey-Fuller (ADF) unit root test.[16] The null hypothesis of the ADF test posits the presence of a "unit root," which indicates non-stationarity, while the alternative hypothesis suggests its absence. If the test's p-value is less than the accepted significance level of 0.05, the null hypothesis is rejected, indicating that the series is stationary.[17] The results of this test are available within the application.

Application and its Graphical User Interface (GUI)

The application is designed to be accessible to both technical and non-technical users. To achieve this, a user-friendly Graphical User Interface (GUI) has been developed. The user can load a file, select detection method, make manual changes to mark or unmark artifact and correct chosen set of intervals with appropriate method. Once the corrections are complete, the cleaned signal and a summary of the actions performed can be saved. Application’s GUI is divided into segments, as illustrated in **Figure 4**:

– Data can be loaded into the application from .txt files, Microsoft Excel worksheets (.xls, .xlsx), or comma-separated files (.csv), with a comma used as the delimiter, using the “Load file (.txt, .csv, .xlsx or .xls)" button (section 1 of **Figure 4**). Example data that can be loaded to an application, as a .txt file can be found in **Table 3**.
– Visualization (section 2 in **Figure 4**) comprises two plots. On the left panel, the graph of RR interval series is displayed, allowing users to mark or unmark detected artifacts with a mouse click. The Poincaré plot is shown on the right panel.
– Artifact identification (section 3 in **Figure 4**) can be performed using one of the automatic methods (T1-T3, or the Tarvainen, Quotient and Square filtering auto-detection buttons). For T1-T3 artifacts, users can adjust the interval differencing time via the corresponding textbox. Description of those methods can be found under *“Artifact detection methods”*.

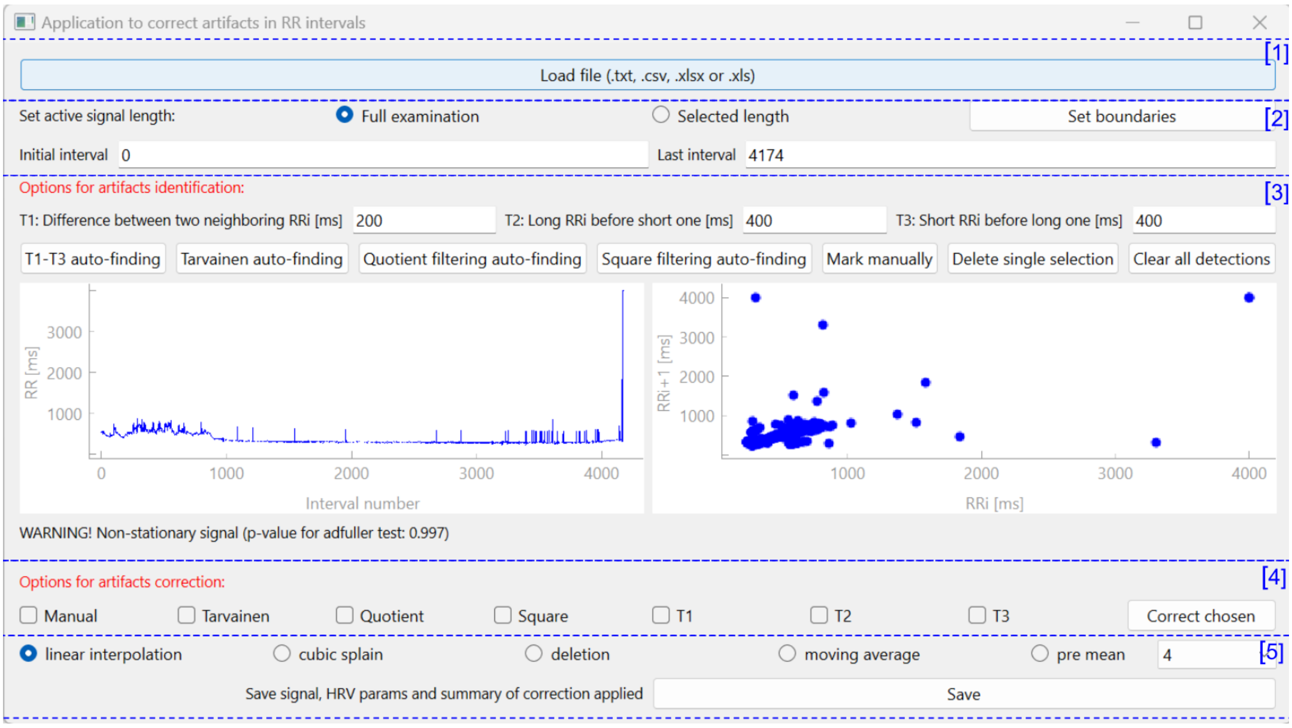

Figure 4. GUI of the CoRRection application with sections marked.

– To correct identified artifacts, the user must first configure the appropriate settings (section 4 of **Figure 4**). The user begins by selecting which group(s) of artifacts to correct—this can be one or multiple groups at a time. The default option is to apply artifact detection and correction methods to the full length of signal. Taking into consideration that RR signal may vary significantly in time series an option to apply algorithms to shorter segments was introduced. To correct signal in specific time period user may define signal of interest by option "Set active signal length". It allows for defining the first and last probe to which algorithms will be applied. User can change boundaries of active signal multiple times by clicking on "Selected length" button. This option may be especially useful for correcting RR intervals which include warm up, exercise and recovery phases, especially while applying threshold dependent detection methods like T1-T3. This option allows also for saving shorter RR series e.g., if the signal contains movement artifacts at the beginning or end of the signal. Next, the user chooses a correction method. The default one is "linear interpolation". For the "pre mean" option, the user can also adjust the number of samples used to calculate a new interval value. The description of correction methods can be found under *"Artifacts correction methods"*.
– Once signal preprocessing is complete, the results can be saved in a file with the same extension as the input file (section 5 of **Figure 4**). In addition to the cleaned signal, a summary of the actions performed on the signal is generated in a separate .txt file. The content of this summary file is displayed in **Table 4**.

Technical information

The application was developed in Python, an open-source programming language. The graphical interface was created using PyQt6, a wrapper for the Qt library. Most of the algorithms used for artifact correction were implemented using scipy, numpy and pandas modules.

The source code of the application, along with an .exe file enabling to run the application without the need of code compilation can be found on GitHub:
https://github.com/MMikielew/coRRection

Results

An example of an examination processed using the CoRRection application is shown in **Figures 5-7**. **Figure 5** displays the raw examination data as it was loaded into the application. **Figure 6** illustrates the artifacts detected after applying detection methods T1-T3, with parameters set at 200 ms for each artifact type and using the Tarvainen method, as well as the manual marking of a few samples that were missed by these methods.

The signal was cleaned using linear interpolation for automatically detected artifacts, while manually marked artifacts, mostly identified as movement-related rather than misread heartbeats, were removed as they occurred towards the end of the examination. The effect of applied correction methods is shown in **Figure 7**. Summary of corrections applied to this examination is shown in **Table 4**.

Table 3. Example of 10 first rows of a .txt file used as an input for the application.

544
551
580
574
539
556
575
552
559
540

Table 4. Summary of corrections applied to a study, containing 84 Tarvainen artifacts corrected using linear interpolation and 15 samples removed from the signal.

Number of removed artifacts: 15
Number of corrected artifacts: 84
Count for linear interpolation: 84
Count for cubic splain: 0
Count for moving average: 0
Count for pre mean: 0
Count for Tarvainen: 84
WARNING! Non-stationary signal (p-value for adfuller test: 0.29)

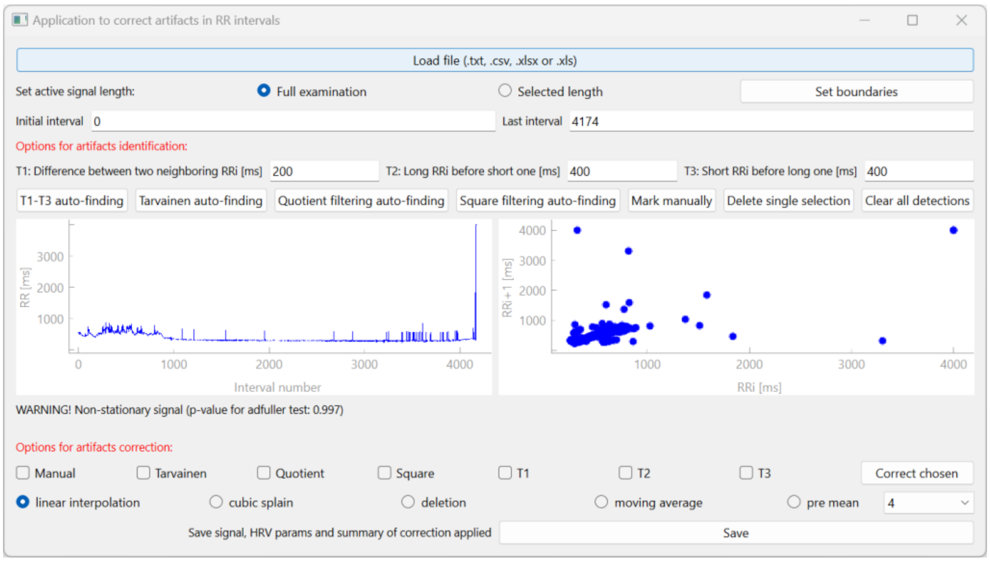

Figure 5. A sample raw examination loaded into the CoRRection.

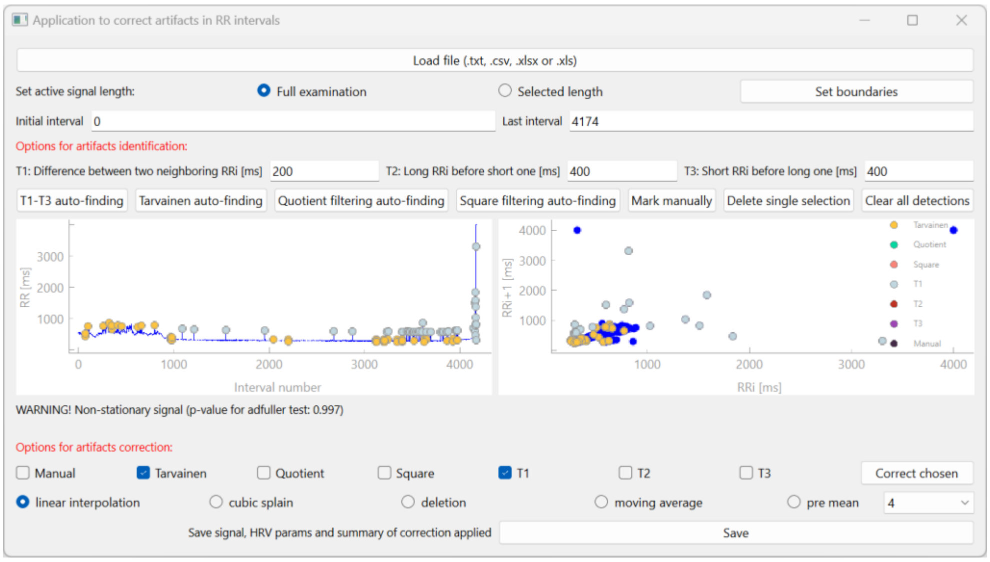

Figure 6. A sample raw examination with automatically detected artifacts with a use of T1-T3 and Tarvainen methods.

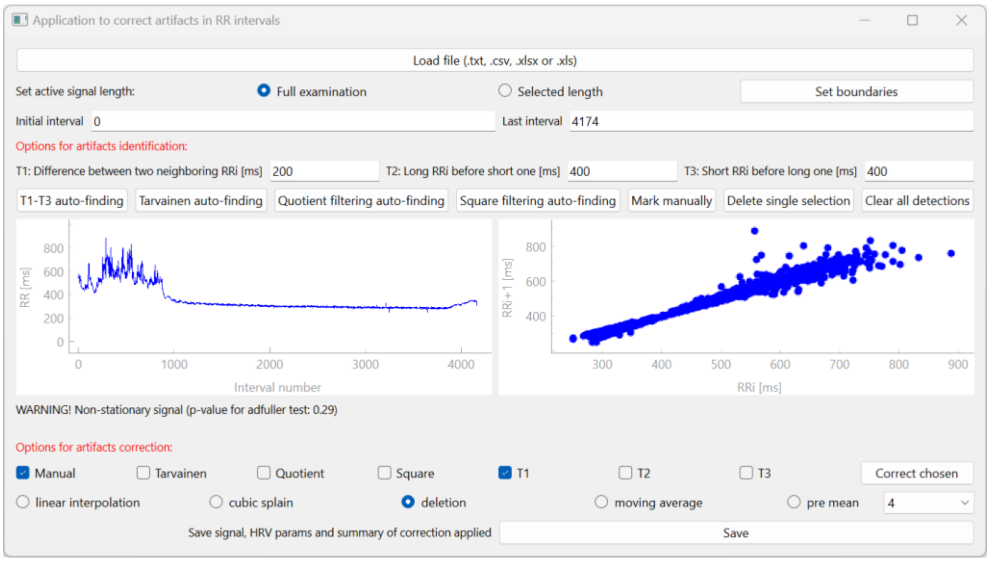

Figure 7. A sample examination preprocessed with a use of CoRRection.

The concept of manually adjusting correction methods was also highlighted for certain examinations. For example, in **Figures 8-10**, the identification of T1–T3 artifacts demonstrates the potential for overcorrection in some studies. **Figure 8** shows a raw examination containing data from the warm-up, exercise, and cooldown phases. **Figure 9** illustrates the outcome of automatic RR artifact detection for T1 types with a threshold set at 200. While the artifacts identified in the middle segment are indeed valid, the same threshold configuration incorrectly flags physiologically normal samples as artifacts requiring correction. To effectively utilize the T1 detection method, the active segment definition was applied, designating the second part of the study as the active segment (indicated by red vertical lines on the plot). **Figure 10** displays the results of artifact correction applied exclusively to the data recorded after the warm-up phase.

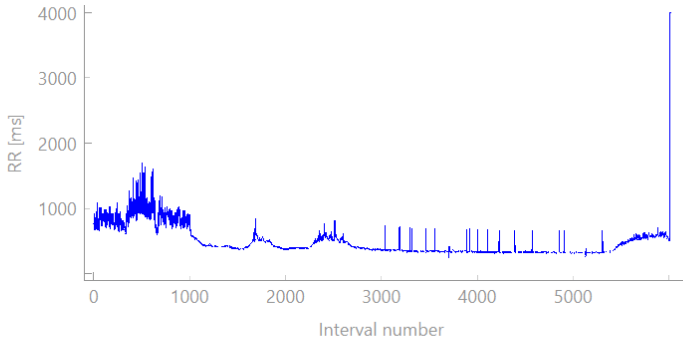

Figure 8. A sample examination containing warmup and exercise phases.

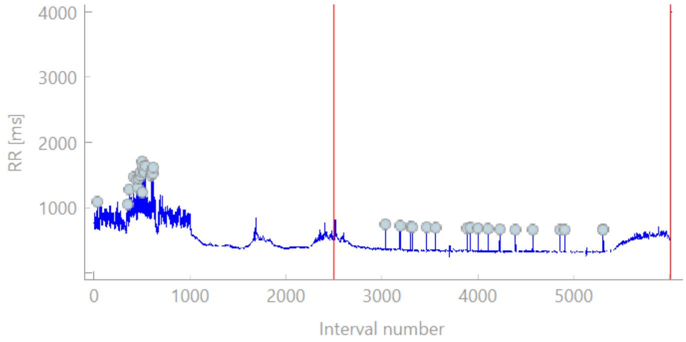

Figure 9. A sample examination containing warmup and exercise phases with artifacts detected as T1 type with threshold set to 200 ms marked.

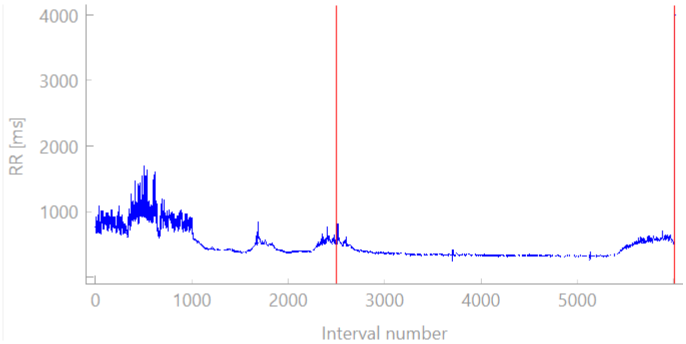

Figure 10. A sample examination after correction of T1 artifacts type within a specified active segment (marked with red vertical lines).

Discussion

There are already a few applications that offer artifact handling, such as:

- PyBios[12]: A freeware tool for cardiovascular signal analysis with a focus on non-linear HRV analysis. It allows users to manually adjust artifact detection thresholds but does not support the correction of individual detections.
- EasieRR[4]: An open-source application that enables manual correction of detections. It does not support automatic artifact detection and requires ECG signals for analysis.
- ARTiiFACT[5]: An open-source software that uses only Berntson's algorithm[18] for artifact detection. It does not support manual detection, and artifacts are either deleted or replaced using linear or cubic spline interpolation.

and many other applications enabling automatic correction (PhysioScripts[19], HRVFrame[20], Kubios HRV[3], Ghrv[21], HRVanalysis[22], PhysioNet HRV toolkit[23], PhysioNet Cardiovascular Signal Toolbox[24], PhysioZoo[25], pyHRV[26], HRVTool[27], RR-APET[28], HRAExplorer (*https://hraexplorer.com/*).

The CoRRection application was developed in response to lack of few functionalities that seemed crucial for the preprocessing of non-stationary RR signal gathered in non-static conditions. Such conditions present different characteristics, making it challenging to select a single identification method that would be effective for each scenario. Addressing this need, the idea of applying identification algorithms to shorter signal segments was introduced. Each segment and each participant could be treated individually, based on unique characteristics of the current phase of the examination.

An application with various artifact identification and correction methods, no matter how advanced, is never able to recover lost data ideally as the physiological signals are very unpredictable. Additionally, the more noise is in the signal, the less variability will be present in the signal after applying the interpolation techniques. Currently, no artifact correction method was found to have similar to a professional accuracy in correcting the data, and medical personnel still consider manual correction to be more reliable.[13]

Both identification and correction methods give users the opportunity to test multiple options, which can lead to a better understanding of correction mechanisms and gain better trust in automatic corrections. This may help popularize the use of more automatic solutions and reduce the time spent on preprocessing of examination.

The next possible step to improve the application functionalities is to introduce a correction proposal tool. The idea is to propose dividing identified artifacts into groups, which will be assigned appropriate correction method. For example, if numerous artifacts are detected at the beginning and the end of signal, they may be caused by motion occurring when the measuring device is put on or taken off. These intervals are unrelated to one's physiology so should be deleted from the signal. However, this method should not be applied in other cases. Different heuristic approaches could also be applied to enhance the correction process.

Conclusions

We have introduced a new open-source application designed to assist researchers and clinicians in efficiently preparing RR intervals for HRV analysis. The semi-automatic approach not only accelerates the preprocessing workflow but also ensures transparency regarding modifications made to the signal.

References

1. 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. 1996;93(5):1043-1065. https://doi.org/10.1161/01.CIR.93.5.1043
2. Billman GE. Heart Rate Variability ? A Historical Perspective. Front Physiol. 2011;2. https://doi.org/10.3389/fphys.2011.00086
3. Tarvainen MP, Niskanen JP, Lipponen JA, Ranta-aho PO, Karjalainen PA. Kubios HRV – Heart rate variability analysis software. Comput Methods Programs Biomed. 2014;113(1):210-220. https://doi.org/10.1016/j.cmpb.2013.07.024
4. Rasmussen JH, Rosenberger K, Langbein J. EasieRR: An open-source software for noninvasive heart rate variability assessment. Codling E, ed. Methods Ecol Evol. 2020;11(6):773-782. https://doi.org/10.1111/2041-210X.13393
5. Kaufmann T, Sütterlin S, Schulz SM, Vögele C. ARTiiFACT: a tool for heart rate artifact processing and heart rate variability analysis. Behav Res Methods. 2011;43(4):1161-1170. https://doi.org/10.3758/s13428-011-0107-7
6. Klein J. Improving the reproducibility of findings by updating research methodology. Qual Quant. 2022;56(3):1597-1609. https://doi.org/10.1007/s11135-021-01196-6
7. Giles DA, Draper N. Heart Rate Variability During Exercise: A Comparison of Artefact Correction Methods. J Strength Cond Res. 2018;32(3):726-735. https://doi.org/10.1519/JSC.0000000000001800
8. Martinez P, Grinand M, Cheggour S, Taieb J, Gourjon G. How to properly evaluate cardiac vagal tone in oncology studies: a state-of-the-art review. J Natl Cancer Cent. 2024;4(1):36-46. https://doi.org/10.1016/j.jncc.2024.02.002
9. Lipponen JA, Tarvainen MP. A robust algorithm for heart rate variability time series artefact correction using novel beat classification. J Med Eng Technol. 2019;43(3):173-181. https://doi.org/10.1080/03091902.2019.1640306
10. Jarosław Piskorski, Przemysław Guzik. Filtering Poincaré plots. https://doi.org/10.12921/CMST.2005.11.01.39-48
11. Rincon Soler AI, Silva LEV, Fazan R, Murta LO. The impact of artifact correction methods of RR series on heart rate variability parameters. J Appl Physiol. 2018;124(3):646-652. https://doi.org/10.1152/japplphysiol.00927.2016
12. Silva LEV, Fazan R, Marin-Neto JA. PyBioS: A freeware computer software for analysis of cardiovascular signals. Comput Methods Programs Biomed. 2020;197:105718. https://doi.org/10.1016/j.cmpb.2020.105718
13. Peltola MA. Role of editing of R–R intervals in the analysis of heart rate variability. Front Physiol. 2012;3. https://doi.org/10.3389/fphys.2012.00148
14. Magagnin V, Bassani T, Bari V, et al. Non-stationarities significantly distort short-term spectral, symbolic and entropy heart rate variability indices. Physiol Meas. 2011;32(11):1775-1786. https://doi.org/10.1088/0967-3334/32/11/S05
15. Li K, Rüdiger H, Ziemssen T. Spectral Analysis of Heart Rate Variability: Time Window Matters. Front Neurol. 2019;10:545. https://doi.org/10.3389/fneur.2019.00545
16. Josef Perktold, Skipper Seabold, Kevin Sheppard, et al. statsmodels/statsmodels: Release 0.14.2. Published online April 17, 2024. https://doi.org/10.5281/ZENODO.593847
17. Prabhakaran S. Augmented Dickey-Fuller (ADF) Test - Must Read Guide - ML+. Machine Learning Plus. November 2, 2019. Accessed September 15, 2024. https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/
18. Berntson GG, Quigley KS, Jang JF, Boysen ST. An Approach to Artifact Identification: Application to Heart Period Data. Psychophysiology. 1990;27(5):586-598. https://doi.org/10.1111/j.1469-8986.1990.tb01982.x
19. Christie IC, Gianaros PJ. PhysioScripts: An extensible, open source platform for the processing of physiological data. Behav Res Methods. 2013;45(1):125-131. https://doi.org/10.3758/s13428-012-0233-x

20. Jovic A, Bogunovic N. HRVFrame: Java-Based Framework for Feature Extraction from Cardiac Rhythm. In: Peleg M, Lavrač N, Combi C, eds. Artificial Intelligence in Medicine. Vol 6747. Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2011:96-100. https://doi.org/10.1007/978-3-642-22218-4_13
21. Rodríguez-Liñares L, Lado MJ, Vila XA, Méndez AJ, Cuesta P. gHRV: Heart rate variability analysis made easy. Comput Methods Programs Biomed. 2014;116(1):26-38. https://doi.org/10.1016/j.cmpb.2014.04.007
22. Pichot V, Roche F, Celle S, Barthélémy JC, Chouchou F. HRVanalysis: A Free Software for Analyzing Cardiac Autonomic Activity. Front Physiol. 2016;7. https://doi.org/10.3389/fphys.2016.00557
23. Heart Rate Variability Analysis with the HRV Toolkit. Accessed September 15, 2024. https://archive.physionet.org/tutorials/hrv-toolkit/
24. Vest AN, Da Poian G, Li Q, et al. An open source benchmarked toolbox for cardiovascular waveform and interval analysis. Physiol Meas. 2018;39(10):105004. https://doi.org/10.1088/1361-6579/aae021
25. Behar JA, Rosenberg AA, Weiser-Bitoun I, et al. PhysioZoo: A Novel Open Access Platform for Heart Rate Variability Analysis of Mammalian Electrocardiographic Data. Front Physiol. 2018;9:1390. https://doi.org/10.3389/fphys.2018.01390
26. Highlights — pyHRV - OpenSource Python Toolbox for Heart Rate Variability 0.4 documentation. Accessed September 15, 2024. https://pyhrv.readthedocs.io/en/latest/
27. Vollmer M. HRVTool - an Open-Source Matlab Toolbox for Analyzing Heart Rate Variability. In: ; 2019. https://doi.org/10.22489/CinC.2019.032
28. McConnell M, Schwerin B, So S, Richards B. RR-APET - Heart rate variability analysis software. Comput Methods Programs Biomed. 2020;185:105127. https://doi.org/10.1016/j.cmpb.2019.105127

Technical Appendix

Tarvainen's algorithm for artifacts identification

Tarvainen's algorithm relies on several formulas. The variables used are listed in **Table TA1**, and the decision equations are presented in **Table TA2**.

Algorithms by Piskorski and Guzik

Square filter removes probes that fall outside of the "square" boundaries, defined by conditions xn/n+1 > 300, xn/n+1 < 2000. Quotient filter identifies data points as artifacts if any of following conditions are met:

$$\frac{x_n}{x_{n+1}} \geq 1.2 \text{ or } \frac{x_n}{x_{n+1}} \leq 0.8, \qquad \frac{x_{n+1}}{x_n} \geq 1.2 \text{ or } \frac{x_{n+1}}{x_n} \leq 0.8.$$

Source code structure

All codes used in the application can be found in GitHub repository https://github.com/MMikielew/coRRection. Implementation of algorithms responsible for artifact identification and correction can be found under 'artifacts' module in GitHub repository: *https://github.com/MMikielew/coRRection/blob/main/app/artifacts.py*

To ensure smooth installation, users should note that the gevent module, which relies on Cython, may present challenges during compilation; using a precompiled version is recommended for environments where building from source is not feasible.

Table TA1. Decision equations for Tarvainen's algorithm.

Number	Equation	Decision
1	S11(j) >1 and S12(j) < -c1S11(j) + c2	Ectopic beat
2	S11(j) < -1 and S12(j) > -c1S11(j) - c2	Ectopic beat
3	|dRR(j)| > 1 or |mRR(j)| > 3	Check for 4-6 conditions
4	sign[dRR(j)]*dRR(j+1) < -1	Check for 7-8 conditions, if 7 and 8 not fulfilled: Long/short RR(j)
5	|mRR(j)| > 3	Check for 7-8 conditions, if 7 and 8 not fulfilled: Long/short RR(j)
6	sign[dRR(j)]*dRR(j+2) < -1	Check for 7-8 conditions, if 7 and 8 not fulfilled: Long/short RR(j) and RR(j+1)
7	|RR(j)/2 - medRR(j)| < Th2	Missed beat RR(j)
8	|RR(j)+RR(j+1)-medRR(j)| < Th2(j)	Extra beat RR(j)+RR(j+1)

Table TA2. Variables used in decision equations in Tarvainen's algorithm.

Variable	Value
α	5.2
dRRs(j)	RR(j) – RR(j-1), j = 2 … N
Th1(j)	α QD [|dRRs(j-45 … j+45)|], j = 1… N where QD is quartile deviation
dRR(j)	dRRs(j)/Th1(j), j = 1 … N
mRRs(j)	RR(j) – median[RR(j-5… j+5)], j = 1 … N
Th2(j)	α QD [|mRRs(j-45 … j+45)|], j = 1… N
mRR(j)	mRRs(j)/Th2(j), j = 1… N
S11(j)	dRR(j), j = 1… N
S12(j)	max[dRR(j-1), dRR(j+1)], if dRR(j) > 0 min[dRR(j-1), dRR(j+1)], if dRR(j) < 0