

Article

Not peer-reviewed version

GUI-Based ECG Viewer for R-Wave Correction and Fast Updating of R-R Interval Time Series

[Yutaka Yoshida](#)^{*} and Kiyoko Yokoyama

Posted Date: 2 September 2025

doi: 10.20944/preprints202509.0164.v1

Keywords: ECG viewer; GUI; R-R interval; signal correction; bio signal processing



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

GUI-Based ECG Viewer for R-Wave Correction and Fast Updating of R-R Interval Time Series

Yutaka Yoshida ^{1,*} and Kiyoko Yokoyama ²

¹ Graduate School of Design&Architecture, Nagoya City University, 2-1-10, Kita Chikusa, Chikusa-ku, Nagoya 464-0083, Japan

² Graduate School of Date Science, Nagoya City University, 1, Yamanohata, Mizuho-cho, Mizuho-ku, Nagoya 467-8501, Japan

* Correspondence: yoshida@sda.nagoya-cu.ac.jp

Abstract

Electrocardiogram (ECG) is widely used to assess cardiac function, autonomic activity, and exercise responses. Accurate detection of R-wave peaks and precise calculation of R-R intervals (RRI) are essential for heart rate variability (HRV) analysis, which is applied to stress assessment, sleep stage evaluation, and detection of subtle cardiac changes. Automated R-peak detection algorithms, ranging from the classical Pan–Tompkins method to recent deep learning approaches, enable rapid processing of large datasets but remain susceptible to misdetections caused by noise, baseline fluctuations, or prominent T-waves. Conventional correction methods often rely on adjusting filters or thresholds, but these approaches risk introducing new errors in other regions, highlighting a trade-off problem and emphasizing the need for a user-friendly manual correction function. To overcome these limitations, we proposed a graphical user interface (GUI)-based ECG viewer implemented in Fortran to ensure high computational efficiency. The system enables interactive insertion or deletion of R-wave peaks, followed by immediate recalculation of RRIs and automatic updates of associated analyses, including power spectral density, histogram, Lorenz, and polar plots. Validation with synthetic ECG signals at four sampling frequencies (125, 250, 500, and 1000 Hz) and three time scales (2, 5, and 10 s) demonstrated correction errors below 0.7% and stable update times within 20–30 ms. Beyond R-wave correction, the viewer allows manual extraction of arbitrary waveform landmarks, such as P- or T-wave peaks, facilitating diverse applications such as QT/PR interval measurement, gold-standard dataset creation, and interactive teaching. Overall, the proposed platform provides a reliable, fast, and versatile tool for ECG editing, with strong potential for research, clinical arrhythmia correction, and educational use.

Keywords: ECG viewer; GUI; R-R interval; signal correction; bio signal processing

Electrocardiogram (ECG) is widely used as an essential measurement tool across various medical and physiological fields, including the diagnosis of cardiovascular diseases, assessment of autonomic nervous system activity, and exercise physiology research [1–5]. Since ECG captures the electrical activity of the heart over time, analysis of the rhythm and morphology of the waveform enables the evaluation of both normal and abnormal cardiac events, as well as subtle changes in cardiac function. In particular, the R-R interval (RRI), defined as the interval between consecutive R-peaks, serves as a fundamental metric in heart rate variability (HRV) analysis and is broadly applied in studies assessing autonomic function, monitoring stress load, analyzing sleep stages, and evaluating exercise response [6–8]. Such applications of ECG highlight its central role in bio signal processing, where cardiac dynamics are analyzed alongside other physiological signals to assess systemic conditions. Furthermore, RRI time series data, when combined with techniques such as power spectral analysis, time series analysis, and nonlinear analysis, have been reported to detect subtle changes in cardiac activity that cannot be captured by mean heart rate alone [9–14]. Therefore,

accurate detection of R-peaks and precise calculation of RRI are critically important in ECG analysis research.

Automated R-peak detection algorithms, originating from the Pan–Tompkins method proposed in 1985 [15] and subsequent improvements, have been widely utilized to support large-scale data analysis in both research and clinical settings. These algorithms combine multiple processing steps—including noise removal via bandpass filtering, R-peak enhancement using differentiation and squaring, peak shaping through moving integration, and peak extraction via thresholding—to achieve rapid and automated R-peak detection [16,17]. More recently, deep learning-based methods for R-peak detection have also been proposed [18–20].

However, baseline variations caused by experimental conditions or subject movement, cases where T-waves exceed R-waves [21], and the presence of noise can result in inaccurate R-peak identification. Such errors reduce the precision of RRI calculation and may introduce inaccuracies in subsequent analyses, such as HRV or stress evaluation. Manual correction of misdetections is burdensome even in short recordings, and the workload increases dramatically in long-term recordings such as 24-hour Holter ECG. Therefore, an efficient and user-friendly function that allows analysts to quickly and intuitively correct R-wave positions is essential for both short- and long-duration data. Also, traditional approaches typically correct misdetections by re-adjusting filters or thresholds, but these methods carry the risk of introducing new errors outside the corrected regions, underscoring the need for a manual yet efficient correction function that avoids such trade-offs.

Previously developed software tools include Kubios HRV, widely used for HRV analysis [22]; RR-APET, a Python-based graphical user interface (GUI) tool allowing R-peak editing and batch processing [23]; ECGdeli, a MATLAB-based ECG visualization tool [24]; as well as CEPs [25] and PhysioZoo [26]. While these tools offer various metrics and functionalities for HRV and ECG signal analysis, their role in the broader domain of bio signal processing remains limited, particularly in terms of GUI-based operations that enable intuitive correction of R-peak misdetections and immediate updates of RRI time series.

To address these challenges, this study proposes an ECG viewer equipped with GUI-based correction functionality. The system allows simple manual correction of failed R-peak detection or T-wave misidentification and instant updating of the RRI time series. By complementing the limitations of automated detection algorithms, the proposed system aims to improve both the efficiency and accuracy of ECG analysis.

2. Design

2.1. Programming Language and Drawing Library

The computational core of the system was implemented in Fortran (FTN95 compiler, Silverfrost Ltd., Manchester, UK), and the DISLIN graphics library was used for drawing. Fortran was chosen because of its high numerical computation performance and compatibility with existing code assets. DISLIN provides versatile two- and three-dimensional graphics functions as well as bitmap output, making it suitable for waveform visualization and interactive drawing.

For the development environment, the Fortran editor on Windows (Plato IDE, version 4.3.0, Silverfrost Ltd., Manchester, UK) was used to describe both numerical processing and GUI logic. DISLIN (version 10.0, Dislin Software, Göttingen, Germany) was employed for graphics, and the linkage to DISLIN was performed through dynamic linking. In Plato IDE, diss32.lib was added as the link library, enabling calls to DISLIN functions. At runtime, disdll.dll, diss32.dll, and the FTN95 runtime library salflib.dll were placed in the same directory, allowing graphics functions to be accessed via dynamic linking. The specifications of the PC and display used in this study are summarized in Table 1.

Table 1. Environmental specifications.

System Specifications	
Operating System	Windows 10, 64-bit
Processor	Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz 3.00 GHz
Installed RAM	16.0 GB
Display Specifications	
Graphics Card	Intel(R) UHD Graphics 630 (128 MB)
Panel Type	27-inch wide TFT LED, non-glare, ADS panel
Maximum Resolution	1920×1080
Pixel Pitch (H × V)	0.3114 mm × 0.3114 mm
Display Area (H × V)	597.888 mm × 336.312 mm
Maximum Colors	16.77 million colors
Response Time	14 ms [GTG] (6.1 ms [GTG] with overdrive enabled)

2.2. Viewer Design and Operation

The viewer was designed using the DISLIN functions CALL CURVE and CALL BARS, as illustrated in Figure 1. First, the CALL METAFL(‘XWIN’) function was used to define a graphical output window, with the window size set to approximately 465 mm (H) × 310 mm (V). The screen layout was organized as follows:

- (a)Top panel: ECG waveform (cyan) and peak flags (magenta)
- (b)Middle panel: RRI time series (cyan)
- (c)Bottom panel: From left to right, the power spectral density function of the RRI time series (yellow), histogram (orange), Lorenz plot (green), and polar plot (peach, circular waveform)

The four analyses were mathematically defined as follows:

Power spectral density (PSD): estimated using the fast Fourier transform (FFT) of the RRI time series $r(n)$:

$$P(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} r(n) e^{-2\pi f n / N} \right|^2 \tag{1}$$

Histogram: distribution of RRI values $r(n)$, represented as frequency counts within bins:

$$h(k) = \sum_{n=0}^{N-1} \delta(r(n) \in bin_k) \tag{2}$$

Lorenz plot: scatter plot of successive RRIs: (r_n, r_{n+1})

Polar plot: circular mapping of RRIs:

$$(r_n, \theta_n), \quad \theta_n = 2\pi(n/N) \tag{3}$$

In addition, a toolbar was placed at the top of the screen. The toolbar contained the following buttons: Insert, Delete, Resolution: up, Resolution: down, Reset, and Quit, which could be operated using the left mouse button. Right-clicking was disabled.

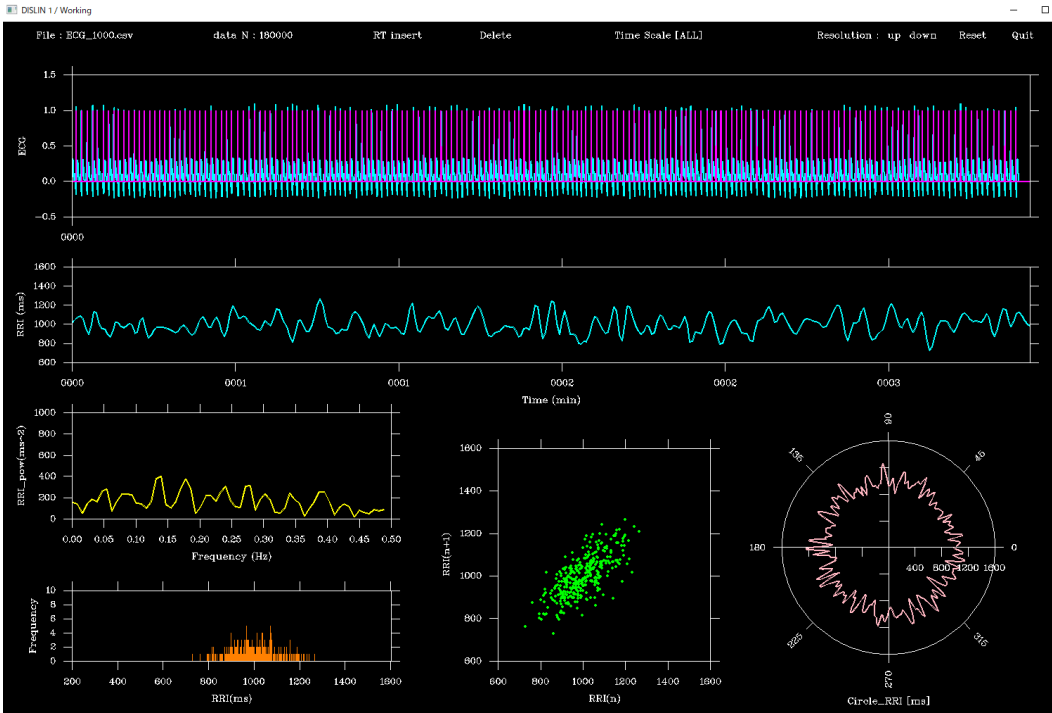


Figure 1. Graphical ECG viewer. ECG sampling frequency:1000Hz, data length:3min, data number:180000.

2.2.1. Time-Scale Adjustment

Clicking the Resolution: up button shortened the time scale of the ECG display, e.g., 60 sec → 30 sec → 10 sec, down to a minimum of 1 sec. Conversely, the Resolution: down button extended the scale, e.g., 120 sec → 180 sec → 300 sec, up to a maximum of approximately 1 hour at a sampling frequency of 1000 Hz.

These operations affected only the time scale of the ECG graph, without influencing other plots. Clicking outside the toolbar advanced the time axis by one time-scale unit. Similarly, pressing the numeric keypad “6” or the right arrow key advanced the display by half a time scale unit, while the Backspace key moved back one unit and the numeric keypad “4” or left arrow key moved back by half a unit. Clicking Reset restored the full-length ECG display.

2.2.2. Insert Function

Clicking the Insert button activated insertion mode, indicated by the button turning yellow. In this mode, clicking within the ECG graph frame or pressing the Enter key displayed a vertical line (flag), and the corresponding time was recorded. Clicks or key presses outside the frame were ignored. This function allowed the user to add flags at arbitrary time points simply by positioning the mouse cursor and clicking. Clicking Insert again terminated the mode.

2.2.3. Delete Function

The Delete function removed a flag and its associated time, using an operation similar to the Insert function. By clicking or pressing Enter just before the target flag, the flag and its recorded time were deleted. The deletion interval was fixed at 0.2 sec, independent of sampling frequency. This feature was useful for removing misdetected peaks caused by P or T waves.

2.3. Operational Flowchart and Algorithm

The processing flow of the system is illustrated in Figure 2. After acquisition, ECG data were converted into a three-column two-dimensional array, *array(3,), storing measurement time, ECG values, and peak information as follows:

- (a) array(1, i): Measurement time (in seconds or sampling index). For example, at a sampling frequency of 1000 Hz, values such as 0.001 sec, 0.002 sec, 0.003 sec, ... were stored.
- (b) array(2, i): Sampled ECG waveform (raw data).
- (c) array(3, i): Flag indicating whether an R-wave peak was detected (1 for peak, 0 otherwise).

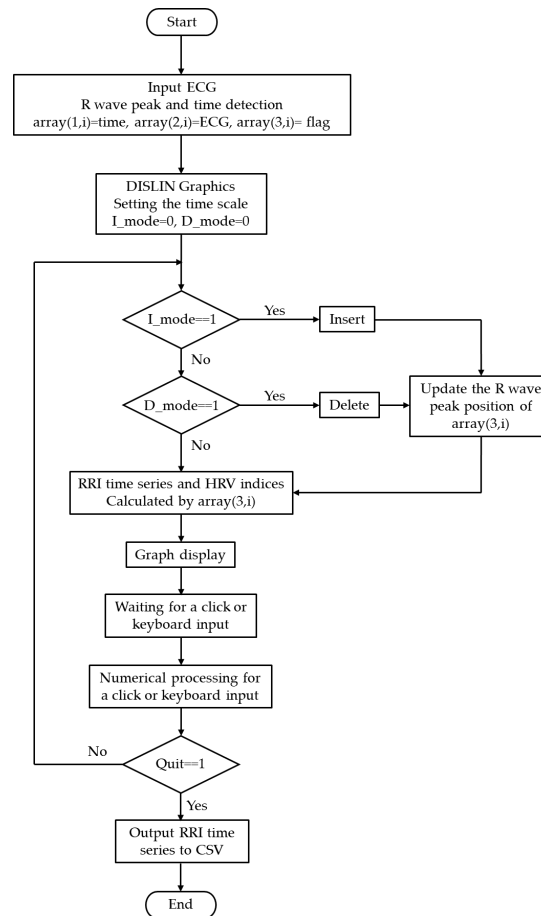


Figure 2. System flowchart.

The RRI is calculated by extracting the measurement times corresponding to array(3, i) = 1 and computing the time differences between consecutive peaks. With this structure, when the user interactively inserts or deletes an R-wave peak, only the corresponding flag value in array(3, i) is updated, while the raw ECG data remain unchanged. After editing, the RRI is recalculated from the updated array(3, i), interpolated using a spline function, and other graphs (RRI time series, power spectrum, Lorenz plot, histogram, and polar plot) are updated rapidly. This non-destructive raw-data approach with flag updates preserves data integrity and enables lightweight, fast updates. Thus, error correction and temporal adjustments can be performed intuitively and safely.

Figure 3 compared the ECG before and after editing for a 5 sec segment around 2 minutes. In insert mode, the P and T wave times for this segment were extracted, whereas in delete mode, three R waves at the same time were removed. Compared with the unedited version, the RRI time series and other graphs changed accordingly.

After displaying the graphs, the system waited for user input, responding interactively to mouse clicks and keyboard commands. To convert mouse-click positions on the screen into time information, the following equation (Equation (4)) was defined:

$$T_{pos} = x1 + \frac{\text{Time scale} \times (NX - X_{left})}{X_{right} - X_{left}} \quad (4)$$

Here, x1 is the starting time of the currently displayed ECG (sec), Time scale is the display length (sec), NX is the mouse-click X coordinate (in pixels), Xleft is the X coordinate of the left ECG axis, and Xright is the X coordinate of the right ECG axis. This transformation ensured that any mouse click on

the screen could be mapped precisely to the corresponding time in the ECG time series, allowing accurate addition or deletion of R-wave peaks.

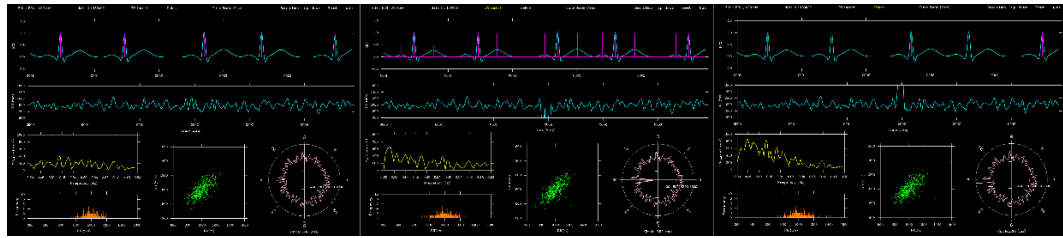


Figure 3. Comparison before and after insertion/deletion of ECG flags via the GUI. Left: before editing, Center: insert mode, Right: delete mode. Editing the flags alters the RRI time series, which propagates to the other graphs.

2.4. ECG Simulation Data for Accuracy Evaluation

Because R-wave correction in the GUI was performed by converting screen coordinates to time, errors could arise depending on the accuracy of this transformation. Such errors were expected to depend on the ECG sampling frequency and the time scale used for display. To evaluate the accuracy of the system, a synthetic ECG dynamic model with known parameters was employed, and the RRI calculation error introduced by GUI-based operation was quantified. The synthetic ECG model was constructed by approximating the P, Q, R, S, and T waves using Gaussian functions. The equation for the model is shown in Equation (5) [27]. For each wave k , peak amplitude (A_k), peak time offset (t_k), and standard deviation (σ_k) were specified. Each parameter was varied randomly within $\pm 10\%$ of its reference value, reproducing beat-to-beat morphological variability. Reference values are listed in Table 2.

$$ECG(t) = \sum_{k \in \{P, Q, R, S, T\}} A_k \cdot \exp \left\{ -\frac{(t-t_k)^2}{2\sigma_k^2} \right\} \quad (5)$$

Table 2. Parameters of a synthetic ECG dynamic model.

Wave	t_k [sec]	A_k	σ_k [sec]
P	0	0.15	0.025
Q	0.15	-0.1	0.012
R	0.2	1	0.015
S	0.25	-0.25	0.015
T	0.5	0.35	0.05

Simulation data were generated at sampling frequencies of 125 Hz, 250 Hz, 500 Hz, and 1000 Hz, with cycle lengths set to approximately 1 sec, yielding 180 sec of synthetic ECG signals. RRI values were calculated from each signal and defined as the gold standard. Then, across 12 conditions (four sampling frequencies \times three time scales: 2 sec, 5 sec, and 10 sec), the author manually clicked on R-wave peaks using the mouse cursor, and relative errors between the GUI-derived RRI and the gold standard were computed using Equation (6). In this validation, the 180-s ECG contained approximately 180 R-wave peaks. All peaks were manually clicked to detect RR intervals, and 180 relative errors were calculated for each condition. The mean and standard deviation of these errors were then obtained.

$$\text{Relative error} = \frac{|GUI_{RRI} - simulation_{RRI}|}{simulation_{RRI}} \times 100 \quad (\%) \quad (6)$$

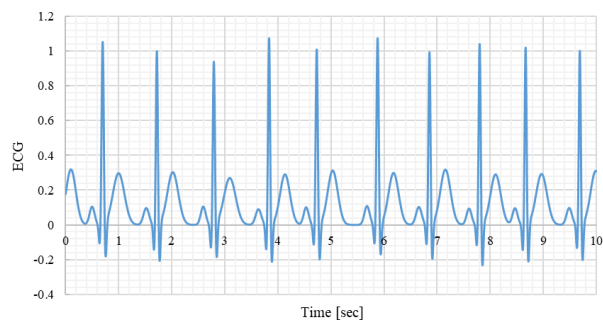


Figure 4. Example of a synthetic ECG signal. Sampling frequency 1000Hz.

Additionally, to evaluate system responsiveness, the processing time between using Insert/Delete and updating the graphical display (see Figure 2 flowchart) was measured using the Fortran CALL CPU_TIME function. For each condition (four sampling frequencies × two modes), 30 clicks were performed, and the mean response time was calculated.

3. Results

Table 3 summarizes the mean and standard deviation of simulated RRI and GUI-derived RRI for each sampling frequency. The results showed that across all sampling frequencies, the mean and standard deviation of GUI-derived RRI were nearly identical to those of the simulated RRI, regardless of the time scale. Table 4 shows the relative error between GUI-derived and simulated RRI for different sampling frequencies and time scales. The smallest relative error was $0.146 \pm 0.264\%$ at 1000 Hz with a time scale of 2 sec, while the largest error was $0.694 \pm 0.601\%$ at 125 Hz with a time scale of 10 sec. The smallest standard deviation was $\pm 0.132\%$ at 1000 Hz with a time scale of 10 sec. Table 5 presents the mean and standard deviation of operation response times for Insert and Delete modes across different sampling frequencies. The mean response time for Insert ranged from 25 to 29 msec, while that for Delete ranged from 24 to 30 msec. The standard deviation of response times ranged from 8 to 17 msec for Insert and from 8 to 14 msec for Delete.

Table 3. Mean and standard deviations of the simulated RRI and GUI_RRI for each sampling frequency.

Sampling frequency [Hz]	Simulation RRI [ms]	GUI_RRI [ms]		
		Time scale [sec]		
		2	5	10
125	992±103	991±102	992±103	991±102
250	1007±97	1006±97	1006±98	1006±98
500	996±94	995±94	995±94	995±94
1000	992±100	992±100	992±100	992±100

Table 4. Relative error of the GUI_RRI for the simulation RRI.

Sampling frequency [Hz]	Time scale [sec]		
	2	5	10
125	0.146±0.264	0.362±0.434	0.694±0.601
250	0.168±0.186	0.372±0.288	0.674±0.492
500	0.188±0.142	0.372±0.271	0.649±0.491

1000	0.175±0.132	0.325±0.224	0.646±0.481
------	-------------	-------------	-------------

Relative errors were expressed as mean ± standard deviation (%).

Table 5. Operation response time in insert and delete for each sampling time.

Sampling frequency [Hz]	Insert response time [ms]			Delete response time [ms]		
	Time scale [sec]			Time scale [sec]		
	2	5	10	2	5	10
125	28±10	27±17	27±10	28±10	28±12	27±12
250	25±8	26±16	27±13	26±10	26±10	26±9
500	28±9	29±14	28±8	28±9	28±14	30±11
1000	28±15	27±16	27±17	24±8	30±8	29±8

4. Discussion

The GUI-based ECG viewer developed in this study is characterized by its ability to intuitively and immediately correct R-wave misdetections, distinguishing it from conventional software. For example, Kubios HRV covers a wide range of indices required for heart rate variability analysis and is widely used in both clinical and research settings; however, its R-peak correction functionality is limited [22]. RR-APET, a Python-based tool, allows GUI-driven R-peak editing and batch processing, but it is not fully optimized for real-time RRI updates [23]. ECGdeli is a powerful MATLAB-based tool for ECG visualization and fiducial point detection [24], and tools such as CEPS [25] and PhysioZoo [26] provide specialized functions for complexity analysis and mammalian HRV analysis, respectively. However, all of these tools primarily focus on analysis capabilities, and their GUI interfaces offer limited functionality for interactive and immediate correction of misdetections. More recently, attempts have been made to benchmark HRV analysis toolboxes [28,29], and LightWAVE [30] allows waveform and annotation editing in a browser environment, yet the functionality required in educational or short-term experimental settings—namely, simple correction of misdetections with simultaneous immediate updates of the RRI series and analysis results—is still insufficient. Our system is unique in this regard, as it allows the correction of R-wave misdetections through the GUI while instantly updating the RRI time series, making it particularly useful for short-term experiments and educational settings.

From an engineering perspective, the system achieves both high accuracy and immediate responsiveness. The relative error of RRI calculated after GUI-based correction was found to be less than 0.7%, demonstrating that manual corrections can be incorporated without compromising reliability. As shown in Table 4, relative error is minimally affected by the sampling frequency but is sensitive to the Time scale setting. This is because a longer Time scale compresses the waveform per beat, making it difficult to accurately click the R-peak with a mouse, whereas a shorter Time scale enlarges the waveform, facilitating visual peak identification and reducing relative error. Therefore, appropriate Time scale settings are important for achieving high correction accuracy via GUI operation. Notably, at 125 Hz with a 2 s time scale, the relative error was smallest on average (0.146%), but the standard deviation was relatively large (±0.264%). This suggests that with coarse sampling, even a slight deviation in mouse clicking leads to substantial variability, since each sample point corresponds to a relatively large time step (8 ms). Conversely, at 1000 Hz with a 2 s time scale, the relative error showed the smallest standard deviation (±0.132%). This is because the temporal resolution is high (1 ms per sample), so even if the mouse click deviates by one sample, the resulting error remains small. Furthermore, as shown in Table 5, the time required to update the RRI after correction is approximately 20–30 ms and remains stable regardless of the sampling frequency or Time scale. This update time includes not only RRI recalculation but also updates of four graphs

based on the RRI time series: power spectral density, Lorenz plot, histogram, and polar plot. Despite the relatively high computational load of these analyses, the total processing time remains within 20–30 ms, highlighting the system's fast responsiveness, which offers significant advantages for research and clinical applications by providing immediate feedback.

The implementation of the system in Fortran is also an important strength in terms of processing speed. Fortran is a compiled language specialized for numerical computation, providing faster execution compared to interpreted languages such as Python or MATLAB. It is widely used in fields such as fluid dynamics, high-speed mathematical model development, and aerospace engineering [31–33], and its long-standing track record in high-performance computing (HPC) and numerical analysis is well recognized in academia. For example, its compiled array-based optimization structures and explicit memory access directives can achieve several- to hundreds-fold speed improvements compared to Python [34]. Moreover, Fortran supports High Performance Fortran (HPF) for HPC, enabling efficient parallelization using FORALL constructs and PURE procedures [35]. In this system, the GUI and computational modules are closely integrated, and although the entire waveform is recalculated each time, high responsiveness is still achieved. Thus, the Fortran implementation represents a rational design choice emphasizing both performance and responsiveness.

Overall, the results indicate that the system combines low error, low latency, and high efficiency, making it applicable not only for short-term experiments in research settings but also for arrhythmia correction in clinical practice and waveform comprehension in educational contexts. It should also be noted that the present study focused mainly on the development and validation of the R-wave correction algorithm. Broader aspects of usability, such as interface layout, toolbar design, and user experience, were not considered. Future work should therefore include systematic evaluation of user-friendliness and improvements in GUI design to enhance accessibility and practical utility in both clinical and educational environments.

5. Conclusions

In this study, we developed a GUI-based ECG viewer with an embedded algorithm for intuitive R-wave correction. The system enabled immediate updates of the RRI time series and related analyses, with correction errors below 0.7% and update times of 20–30 ms. These results demonstrate reliable real-time feedback suitable for both research and educational applications.

Beyond R-wave correction, a notable strength of the system is its ability to manually extract arbitrary waveform landmarks, such as P-wave onset or T-wave peak. This flexibility extends its applicability to various purposes, including precise measurement of QT and PR intervals, creation of gold-standard datasets for algorithm validation, and educational use where students can interactively annotate different ECG components.

Therefore, the proposed system not only enhances efficiency and accuracy in bio signal processing but also holds promise as a versatile tool bridging research, clinical, and educational applications.

Supplementary Materials: The following supporting information can be downloaded at the website of this paper posted on Preprints.org, Video S1: Demonstrating the ECG viewer; File S2: synthetic ECG signal used for verification. Name: S1; Type: Video (.mp4); Description: Demonstration of GUI-based ECG viewer operation (90 sec). Name S2; Type: File(.csv); Description: Synthetic ECG signal used for verification (125, 250, 500, 1000Hz). The time, ECG amplitude, and R wave peak flag are recorded.

Author Contributions: Conceptualization, Y.Y.; methodology, Y.Y.; software, Y.Y.; validation, Y.Y., K.Y.; formal analysis, Y.Y.; investigation, Y.Y.; resources, Y.Y.; data curation, Y.Y.; writing—original draft preparation, Y.Y.; writing—review and editing, Y.Y.; visualization, Y.Y.; supervision, K.Y.; project administration, K.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article and Supplementary Materials. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare that they have no competing interests.

Abbreviations

The following abbreviations are used in this manuscript:

ECG	Electrocardiogram
GUI	Graphical User Interface
HPC	High Performance Computing
HPF	High Performance Fortran
HRV	Heart Rate Variability
RRI	R–R Interval

References

1. Ahmad, M.; Ahmed, A.; Hashim, H.; Farsi, M.; Mahmoud, N. Enhancing Heart Disease Diagnosis Using ECG Signal Reconstruction and Deep Transfer Learning Classification with Optional SVM Integration. *Diagnostics* 2025, 15, 1501.
2. Nechita, L.C.; Nechita, A.; Voipan, A.E.; Voipan, D.; Debita, M.; Fulga, A.; Fulga, I.; Musat, C.L. AI-Enhanced ECG Applications in Cardiology: Comprehensive Insights from the Current Literature with a Focus on COVID-19 and Multiple Cardiovascular Conditions. *Diagnostics* 2024, 14, 1839.
3. Grégoire, J.M.; Gilon, C.; Carlier, S.; Bersini, H. Autonomic Nervous System Assessment Using Heart Rate Variability. *Acta Cardiol.* 2023, 78, 648–662.
4. Brazdzionyte, R.; Velicka, D.; Motiejunaite, K.; Poderiene, K.; Kairiukstiene, Z. Advanced ECG Analysis to Evaluate Multimodal Exercise Effects on Cardiovascular Health. *Medicina* 2025, 61, 473.
5. Beltrame, J.F.; La, S.; Marathe, J. Clinical Utility of the Humble Exercise ECG Stress Test. *J. Am. Coll. Cardiol.* 2024, 83, 300–302.
6. Peabody, J.E.; Ryznar, R.; Ziesmann, M.T.; Gillman, L. A Systematic Review of Heart Rate Variability as a Measure of Stress in Medical Professionals. *Cureus* 2023, 15, e34345.
7. Eddie, D.; Bentley, K.H.; Bernard, R.; Mischoulon, D.; Winkelman, J.W. Aggregating Heart Rate Variability Indices Across Sleep Stage Epochs Ignores Significant Variance Through the Night. *Sleep Med.* 2022, 90, 262–266.
8. Hebisz, R.G.; Hebisz, P.; Zatoń, M.W. Heart Rate Variability After Sprint Interval Training in Cyclists and Implications for Assessing Physical Fatigue. *J. Strength Cond. Res.* 2022, 36, 558–564.
9. Moura-Tonello, S.C.G.; Carvalho, V.O.; Godoy, M.F.; Porta, A.; Leal, Â.M.O.; Bocchi, E.A.; Catai, A.M. Evaluation of Cardiac Autonomic Modulation Using Symbolic Dynamics After Cardiac Transplantation. *Braz. J. Cardiovasc. Surg.* 2019, 34, 572–580.
10. Santos, M.C.S.; Silveira, L.C.L.; Moura-Tonello, S.C.G.; Porta, A.; Catai, A.M.; Souza, G.D.S. Heart Rate Variability in Multibacillar Leprosy: Linear and Nonlinear Analysis. *PLoS One* 2017, 12, e0180677.
11. Georgieva-Tsaneva, G.; Gospodinova, E. Heart Rate Variability Analysis of Healthy Individuals and Patients with Ischemia and Arrhythmia. *Diagnostics* 2023, 13, 2549.
12. Hargens, T.A.; Chambers, S.; Luden, N.D.; Womack, C.J. Reliability of the Heart Rate Variability Threshold During Treadmill Exercise. *Clin. Physiol. Funct. Imaging* 2022, 42, 292–299.
13. Kumar, S.M.; Vaishali, K.; Maiya, G.A.; Shivashankar, K.N.; Shashikiran, U. Analysis of Time-Domain Indices and Frequency-Domain Measures of Heart Rate Variability Derived from ECG Waveform and Pulse-Wave-Related HRV Among Overweight Individuals: An Observational Study. *F1000Res.* 2023, 12, 1229.

14. Miyagi, T.; Yamazato, M.; Nakamura, T.; Tokashiki, T.; Namihira, Y.; Kokuba, K.; Ishihara, S.; Sakima, H.; Ohya, Y. Power Spectral Analysis of Heart Rate Variability as a Screening Tool for Detecting Sympathetic and Parasympathetic Nervous Dysfunctions in Parkinson's Disease. *BMC Neurol.* 2022, 22, 339.
15. Pan, J.; Tompkins, W.J. A Real-Time QRS Detection Algorithm. *IEEE Trans. Biomed. Eng.* 1985, 32, 230–236.
16. Neri, L.; Oberdier, M.T.; Augello, A.; Suzuki, M.; Tumarkin, E.; Jaipalli, S.; Geminiani, G.A.; Halperin, H.R.; Borghi, C. Algorithm for Mobile Platform-Based Real-Time QRS Detection. *Sensors* 2023, 23, 1625.
17. Agostinelli, A.; Marcantoni, I.; Moretti, E.; Sbröllini, A.; Fioretti, S.; Di Nardo, F.; Burattini, L. Noninvasive Fetal Electrocardiography Part I: Pan-Tompkins' Algorithm Adaptation to Fetal R-Peak Identification. *Open Biomed. Eng. J.* 2017, 11, 17–24.
18. Yun, D.; Lee, H.C.; Jung, C.W.; Kwon, S.; Lee, S.R.; Kim, K.; Kim, Y.S.; Han, S.S. Robust R-Peak Detection in an Electrocardiogram with Stationary Wavelet Transformation and Separable Convolution. *Sci. Rep.* 2022, 12, 19638.
19. Chen, H.; Maharatna, K. An Automatic R and T Peak Detection Method Based on the Combination of Hierarchical Clustering and Discrete Wavelet Transform. *IEEE J. Biomed. Health Inform.* 2020, 24, 2825–2832.
20. Rizwan, A.; Priyanga, P.; Abualsauod, E.H.; Zafrullah, S.N.; Serbaya, S.H.; Halifa, A. A Machine Learning Approach for the Detection of QRS Complexes in Electrocardiogram Using Discrete Wavelet Transform Algorithm. *Comput. Intell. Neurosci.* 2022, 2022, 9023478.
21. Kenny, B.J.; Brown, K.N. ECG T Wave. *StatPearls*, 2022. Available online: <https://www.ncbi.nlm.nih.gov/books/NBK538264/> (accessed on 20 August 2025).
22. Tarvainen, M.P.; Niskanen, J.P.; Lipponen, J.A.; Ranta-Aho, P.O.; Karjalainen, P.A. Kubios HRV: Heart Rate Variability Analysis Software. *Comput. Methods Programs Biomed.* 2014, 113, 210–220.
23. McConnell, M.; Schwerin, B.; So, S.; Richards, B. RR-APET: Heart Rate Variability Analysis Software. *Comput. Methods Programs Biomed.* 2020, 185, 105127.
24. Pilia, N.; Nagel, C.; Lenis, G.; Becker, S.; Dössel, O.; Loewe, A. ECGdeli: An Open Source ECG Delineation Toolbox for MATLAB. *SoftwareX* 2021, 13, 100639.
25. Mayor, D.; Panday, D.; Kandel, H.K.; Steffert, T.; Banks, D. CEPS: An Open Access MATLAB Graphical User Interface for the Analysis of Complexity and Entropy in Physiological Signals. *Entropy* 2021, 23, 321.
26. Behar, J.A.; Rosenberg, A.A.; Weiser-Bitoun, I.; Shemla, O.; Alexandrovich, A.; Konyukhov, E.; Yaniv, Y. PhysioZoo: A Novel Open Access Platform for Heart Rate Variability Analysis of Mammalian Electrocardiographic Data. *Front. Physiol.* 2018, 9, 1390.
27. McSharry, P.E.; Clifford, G.D.; Tarassenko, L.; Smith, L.A. A Dynamical Model for Generating Synthetic Electrocardiogram Signals. *IEEE Trans. Biomed. Eng.* 2003, 50, 289–294.
28. Vest, A.N.; Li, Q.; Liu, C.; Nemati, S.; Shah, A.; Clifford, G.D. Benchmarking Heart Rate Variability Toolboxes. *J. Electrocardiol.* 2017, 50, 744–747.
29. Vest, A.N.; Da Poian, G.; Li, Q.; Liu, C.; Nemati, S.; Shah, A.J.; Clifford, G.D. An Open Source Benchmarked Toolbox for Cardiovascular Waveform and Interval Analysis. *Physiol. Meas.* 2018, 39, 105004.
30. Moody, G.B. LightWAVE: Waveform and Annotation Viewing and Editing in a Web Browser. *Comput. Cardiol.* 2013, 40, 17–20.
31. Kulikov, I.; Chernykh, I.; Karavaev, D.; Prigarin, V.; Sapetina, A.; Ulyanichev, I.; Zavyalov, O. A New Parallel Code Based on a Simple Piecewise Parabolic Method for Numerical Modeling of Colliding Flows in Relativistic Hydrodynamics. *Mathematics* 2022, 10, 1865.
32. Kim, B.; Yoon, K.S.; Kim, H.-J. GPU-Accelerated Laplace Equation Model Development Based on CUDA Fortran. *Water* 2021, 13, 3435.
33. Kan, Y.; Liu, X.; Liu, H. An Image and State Information-Based PINN with Attention Mechanisms for the Rapid Prediction of Aircraft Aerodynamic Characteristics. *Aerospace* 2025, 12, 434.

34. Aruoba, B.; Fernández-Villaverde, J. A Comparison of Programming Languages in Macroeconomics. *J. Econ. Dyn. Control* 2015, 58, 265–273.
35. Markus, A.; Mullin, L. Implementing Mathematics of Arrays in Modern Fortran: Efficiency and Efficacy. *Software* 2024, 3, 534–548.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.