# pyPSG: Integrated Python Toolbox for Multimodal Cardiovascular Signal Analysis and Biomarker Extraction

Szabolcs M. Péter[1], Bálint Kristóf[1], Kristóf Müller[1] and Márton Á. Goda[1]

[1]Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Budapest, Hungary

## Abstract

*Most diagnostic methods for cardiovascular diseases (CVDs) are time- and expertise-intensive. Machine learning offers a promising solution by enabling automated analysis of physiological signals. Effective feature engineering from physiological signals is essential for developing reliable machine learning models. This paper presents pyPSG, an open-source Python toolbox that integrates electrocardiogram (ECG), photoplethysmography (PPG), oxygen saturation ($SpO_2$), and heart rate variability (HRV) analysis within a unified framework. The toolbox provides standardized functions for signal loading, preprocessing, fiducial point detection, biomarker extraction, and visualization by integrating existing open-source toolboxes, while providing a framework that can be extended to support additional physiological signals. To evaluate the toolbox, biomarkers extracted using pyPSG were used to classify recordings from the 2015 PhysioNet/Computing in Cardiology Challenge dataset using Logistic Regression, Random Forest, and XGBoost classifiers. Furthermore, the proposed pyPSG workflow was compared with multiple PhysioNet Challenge 2015 solutions. The algorithm developed using the new pyPSG toolbox did not outperform existing approaches, but achieved comparable performance. pyPSG is available on: https://pypsg-github.readthedocs.io/*

## 1. Introduction

Cardiovascular diseases (CVDs) are responsible for approximately 17.9 million deaths annually [1]. With the help of machine learning (ML) methods, easily measurable biological signals such as PPG can be utilized to complement or even replace traditional, more complex diagnostic procedures. To support the preprocessing of such ML approaches, a Python-based toolbox has been developed, capable of handling PPG, electrocardiographic (ECG), oxygen desaturation ($SpO_2$), heart rate variability (HRV), and beat rate variability (BRV) signals. It is important to note that individual toolboxes for each signal type already exist, as well as the PhysioZoo platform [2], which integrates several of these tools. The proposed new toolbox integrates them into a unified environment and extends their functionality.

### 1.1. Cardiovascular signals

Combining multiple physiological signals improves the robustness of cardiovascular analysis. A unified platform also enables multimodal signals to be visualized and analyzed in a standardized format. Under challenging clinical conditions, individual signals are often degraded by noise, motion artifacts, or sensor disconnections. In such cases, missing or unreliable information can be compensated for using complementary signal modalities. Furthermore, different physiological signals reflect different physiological processes, making multimodal analysis particularly valuable for complex tasks such as arrhythmia classification [3]. Additionally, multimodal analysis enables the extraction of metrics derived from multiple signal modalities, such as Pulse Transit Time (PTT). PTT represents the temporal delay between the electrical activation of the heart detected by ECG and the arrival of the corresponding pulse wave at a peripheral measurement site [3].

### 1.2. Multimodal physiological toolboxes

Over the past decades, various open-source toolboxes have been developed for physiological signal processing. These tools support a wide range of signal modalities and provide various functionalities, including preprocessing, fiducial point detection, heart rate variability analysis, and biomarker extraction. However, their capabilities differ regarding supported signal types, extracted biomarkers, and processing pipelines. Consequently, multimodal cardiovascular analysis often requires combining several independent software packages, resulting increased implementation complexity. Multimodal physiological signal toolboxes and their key capabilities relevant to pyPSG are summarized in Table 1.

| | pyPSG | NeuroKit2 | pyPhysio | BioSPPy |
|---|---|---|---|---|
| **ECG** | | | | |
| Prefiltering | ✓ | ✓ | ✓ | ✓ |
| Fiducial det. | ✓ | ✓ | ✓ | ✓ |
| Signal quality | ∼ | ✓ | ✓ | ✓ |
| Biomarker eng. | ✓ | ∼ | ✗ | ∼ |
| **PPG** | | | | |
| Prefiltering | ✓ | ✓ | ✓ | ✓ |
| Fiducial det. | ✓ | ✓ | ✓ | ✓ |
| Signal quality | ∼ | ✗ | ✓ | ✗ |
| Biomarker eng. | ✓ | ∼ | ✗ | ∼ |
| **HRV** | | | | |
| Biomarker eng. | ✓ | ✓ | ✓ | ✓ |
| **$SpO_2$** | | | | |
| Processing | ✓ | ✗ | ✗ | ✗ |

Table 1: Capabilities of Python-based multimodal physiological signal toolboxes relevant to pyPSG. ✓: available; ✗: not available; ∼: partial/limited support.

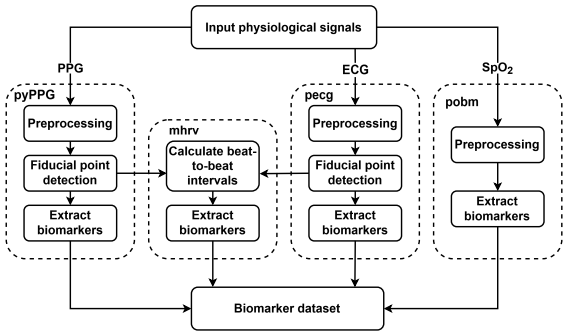


Figure 1: Overview of the pyPSG processing workflow. ECG, PPG, and $SpO_2$ signals are processed using integrated toolboxes.

## 2. Materials and Methods

### 2.1. pyPSG overview

The pyPSG toolbox was developed to provide a unified framework for cardiovascular signal analysis. Instead of using separate software packages for different physiological signals, pyPSG integrates them into a single Python package. The toolbox currently supports electrocardiography (ECG), photoplethysmography (PPG), oxygen saturation ($SpO_2$), and heart rate variability (HRV). The package has a modular architecture consisting of two main subpackages: *IO* and *Biomarkers*. The IO package contains functions for reading, visualizing and exporting physiological signals, while the Biomarkers package implements signal-specific preprocessing and biomarker extraction methods. Additional modules provide utility functions and example scripts demonstrating the usage of the toolbox. The toolbox currently supports EDF files as the primary input format. It provides standardized functions for signal loading, visualization, and biomarker extraction. It can also be easily integrated with other Python-based signal processing toolboxes. The project is open-source, available on PyPI, and supported by online documentation containing installation instructions, tutorials and usage examples.

### 2.2. Processing pipelines

As illustrated in Fig. 1, each supported signal is processed using its own dedicated pipeline while maintaining a common interface. This allows different physiological signals to be analyzed within the same workflow.

The PPG processing pipeline is based on the *pyPPG* toolbox [4]. First, the raw PPG signal is preprocessed by applying bandpass and moving average filters and calculating its first, second, and third derivatives. Next, pulse wave segmentation is performed, where systolic peaks are detected by an enhanced version of the Aboy beat detector [4]. Based on the sytolic peaks, pulse onset and offset are identified to determine individual pulse waves. The next step is the fiducial point detection, during which dicrotic notches and dicrotic peaks are identified in the PPG signal, and predefined fiducial points in its derivatives. All fiducial points are postprocessed for correction. The final step is the biomarker engineering, during which 74 standerdized PPG biomarkers are calculated in four categories: biomarkers derived from the PPG signal, signal ratios, PPG derivatives, and derivative ratios.

The ECG signal is processed using the *PhysioZoo* ECG Python toolbox (*pecg*) [5]. Firstly, the ECG signal is preprocessed by applying bandpass and optional notch filters. Next, fiducial points are detected, R-peaks by epltd algorithm, then the P, QRS, T waveforms and their boundaries. In the next step biomarkers are engineered from the detected fiducial points. A total of 22 ECG biomarkers are extracted, including 14 intervals duration-based biomarkers and 8 wave characteristic biomarkers. Finally, median, minimum, maximum, Q1 and Q3 statistics of the biomarkers are computed.

The $SpO_2$ processing is performed using the *pobm* toolbox [6]. First, the raw $SpO_2$ signal is preprocessed using a delta filter and a block of data filter. The delta filter removes abrupt changes exceeding a predefined percentage threshold, and the block of data filter removes error values below 50% saturation. Following preprocessing, biomarkers describing general statistics, complexity, periodicity, desaturation events, and hypoxic burden are computed.

The HRV processing pipeline is based on the *mhrv* toolbox [7]. The *mhrv* toolbox was originally implemented in MATLAB. To provide a unified framework within pyPSG, the relevant functiolalities were implemented in Python. First, beat-to-beat intervals are derived from the detected ECG R-peaks or PPG systolic peaks. The resulting interval series is then divided into consecutive analysis windows.

Within each window, HRV and BRV biomarkers are computed in the time-domain, frequency-domain, nonlinear, and fragmentation categories. Finally, summary statistics, including the mean, median, standard deviation, $25^{th}$ and $75^{th}$ percentiles, interquartile range, skewness, and kurtosis, are calculated for each biomarker.

## 2.3. Dataset

To evaluate the proposed toolbox, a publicly available training dataset of the 2015 PhysioNet/Computing in Cardiology Challenge [8] was used. The 2015 PhysioNet Challenge focuses on the reduction of false life-threatening arrhythmia alarms in intensive care units (ICUs). The dataset contains multimodal physiological recordings, including two ECG leads and one or more pulsatile waveforms (PPG and/or arterial blood pressure), collected from bedside patient monitors. In each of the recordings, the alarm appears at 5:00 minutes from the start. According to the ANSI/AAMI EC13 standard, the start of the alarm event can precede the alarm by a maximum of 10 seconds, so it occurs between 4:50 and 5:00. In total, the database contains 1250 recordings, divided into 750 training and 500 test recordings. Since the test set is not publicly available, only the training dataset was used in this study.

## 2.4. ML using pyPSG biomarkers

Figure 2 summarizes the machine learning pipeline used for classification of the 2015 PhysioNet Challenge dataset. The recordings were divided into consecutive 30-second windows, and biomarker extraction was performed independently for each window. ECG and HRV biomarkers were extracted from the ECG signals, while PPG and BRV biomarkers were obtained from the PPG signal. Windows with insufficient signal quality were excluded from processing to avoid unreliable feature extraction. The extracted biomarkers were stored in a feature matrix, where each row corresponded to a signal window and each column represented a physiological biomarker.

The feature matrix was divided into training and test sets using GroupShuffleSplit with an 80:20 ratio to prevent data leakage. Logistic Regression (LR), Random Forest (RF), and XGBoost (XGB) classifiers were trained and evaluated using Recursive Feature Elimination (RFE) for feature selection. To reduce the influence of random train-test splits, the complete training and evaluation procedure was repeated 25 times using different random seeds, and the median performance metrics were reported. Model performance was evaluated using the area under the receiver operating characteristic curve (AUC), accuracy, precision, sensitivity, specificity, and F1-score metrics.

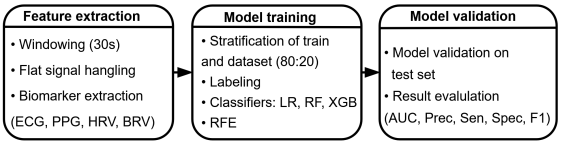


Figure 2: Machine learning workflow used for biomarker-based classification

| Model | AUC | Prec | Sens | Spec | F1 |
|---|---|---|---|---|---|
| LR | 0.57 | 0.52 | 0.64 | 0.49 | 0.67 |
| RF | 0.85 | 0.75 | 0.81 | 0.74 | 0.79 |
| XGB | **0.86** | **0.77** | **0.83** | **0.76** | **0.80** |

Table 2: Median classification performance of LR, RF, and XGB classifiers. Parameters: 25 train-test split iterations, 80:20 train:test ratio, top 20 biomarkers.

## 2.5. Framework for evaluation

The proposed workflow was compared with previously published methods developed for the 2015 PhysioNet Challenge. For the comparison, the first [9]- and second [10]-ranked solutions of the real-time event were selected based on their ranking, and fifth [11]-ranked solution was selected due to its similarity to the proposed pyPSG-based approach. To ensure a fair comparison, a unified framework has been developed. Since the official test set is not publicly available, the comparison was performed using the publicly available training dataset. The original MATLAB implementations provided by the Challenge organizers were reproduced with only minor modifications related to data access and file handling, with the modified implementations available in the project repository. To enable a direct comparison, the reproduced methods were evaluated using the same train-test splits and performance metrics as the proposed pyPSG-based workflow, with the exception of AUC, since the original solutions didn't provide class probabilities. Since the first- and second-ranked Challenge solutions were originally designed to analyze only a short signal segment preceding the alarm, an additional comparison was performed using the full available signal duration.

# 3. Results

## 3.1. Cross validation

The median classification performance of the evaluated machine learning models is presented in Table 2. Among the tested classifiers, XGB achieved the best overall performance with an AUC of 0.86 and an F1-score of 0.80. Consequently, the XGB model was selected for all subsequent comparisons with previous Challenge methods and existing physiological signal analysis toolboxes.

## 3.2. Comparison with previous results

The performance of the proposed workflow was compared with three reproduced top-ranked solutions from the 2015 PhysioNet Challenge. As shown in Table 3, the reproduced Challenge methods achieved higher sensitivity and F1-score than the proposed pyPSG-based workflow. The first-ranked solution obtained the highest precision, specificity, and F1-score, while the fifth-ranked solution achieved perfect sensitivity. Although the proposed workflow did not outperform the reproduced methods, it achieved competitive performance using a generalizable biomarker-based machine learning pipeline.

| Rank | Prec. | Sens. | Spec. | F1 |
|---|---|---|---|---|
| Rank 1 | **0.806** | 0.938 | **0.866** | **0.866** |
| Rank 2 | 0.785 | 0.961 | 0.833 | 0.849 |
| Rank 5 | 0.760 | **1.000** | 0.802 | 0.860 |
| pyPSG | 0.770 | 0.830 | 0.760 | 0.800 |

Table 3: Median performance of reproduced Challenge solutions and the pyPSG-based algorithm. Parameters: 25 iterations, 80:20 train:test split, top 20 biomarkers.

The results of the additional comparison with the first- and second-ranked Challenge solutions on the full 5-minute signal duration can be seen in Table 4. The overall performance of all methods decreased when the full 5-minute signal duration was used. However, the performance degradation of the proposed pyPSG-based workflow was considerably smaller than that of the reproduced Challenge methods.

| Rank | Prec. | Sens. | Spec. | F1 |
|---|---|---|---|---|
| Rank 1 | 0.572 | **0.932** | 0.550 | 0.709 |
| Rank 2 | **0.825** | 0.514 | **0.930** | 0.633 |
| pyPSG | 0.680 | 0.710 | 0.780 | **0.760** |

Table 4: Performance comparison using the full 5-minute signal duration. Performance is reported using precision, sensitivity, specificity, and F1-score.

# 4. Conclusion and Discussion

Future work will focus on the further development and validation of pyPSG. This includes robust validation of the toolbox on larger-scale PSG datasets. In addition, the toolbox will be extended to support phonocardiography (PCG) as the next physiological signal. The project is open-source, with the toolbox available on GitHub and while the experimental and comparison code is available in a separate repository.

# References


[1] Kumar R, Garg S, Kaur R, Johar M, Singh S, Menon SV, Kumar P, Hadi AM, Hasson SA, Lozanović J. A comprehensive review of machine learning for heart disease prediction: challenges, trends, ethical considerations, and future directions. Frontiers in Artificial Intelligence 2025; 8:1583459.

[2] Behar JA, Levy J, Zvuloni E, Gendelman S, Rosenberg A, Biton S, Derman R, Sobel JA, Alexandrovich A, Charlton PH, et al. Physiozoo: the open digital physiological biomarkers resource. In 2023 Computing in Cardiology (CinC), volume 50. IEEE, 2023; 1–4.

[3] Tejedor J, García CA, Márquez DG, Raya R, Otero A. Multiple physiological signals fusion techniques for improving heartbeat detection: A review. Sensors 2019;19(21):4708.

[4] Goda M Charlton PH, Behar JA. pyppg: a python toolbox for comprehensive photoplethysmography signal analysis. Physiological Measurement 2024;45(4):045001. URL https://doi.org/10.1088/1361-6579/ad33a2.

[5] Gendelman S, Biton S, Derman R, Zvuloni E, Levy J, Lugassy S, Alexandrovich A, Behar JA. Physiozoo ecg: Digital electrocardiography biomarkers to assess cardiac conduction. In 2021 Computing in Cardiology (CinC), volume 48. IEEE, 2021; 1–4.

[6] Levy J, Álvarez D, Rosenberg AA, Alexandrovich A, Del Campo F, Behar JA. Digital oximetry biomarkers for assessing respiratory function: standards of measurement, physiological interpretation, and clinical use. NPJ digital medicine 2021;4(1):1.

[7] Behar JA, Rosenberg AA, 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. Frontiers in physiology 2018;9:1390.

[8] Clifford GD, Silva I, Moody B, Li Q, Kella D, Shahin A, Kooistra T, Perry D, Mark RG. The physionet/computing in cardiology challenge 2015: reducing false arrhythmia alarms in the icu. In 2015 Computing in Cardiology Conference (CinC). IEEE, 2015; 273–276.

[9] Plesinger F, Klimes P, Halamek J, Jurak P. False alarms in intensive care unit monitors: detection of life-threatening arrhythmias using elementary algebra, descriptive statistics and fuzzy logic. In 2015 Computing in Cardiology Conference (CinC). IEEE, 2015; 281–284.

[10] Kalidas V, Tamil LS. Enhancing accuracy of arrhythmia classification by combining logical and machine learning techniques. In 2015 Computing in Cardiology Conference (CinC). IEEE, 2015; 733–736.

[11] Antink CH, Leonhardt S. Reducing false arrhythmia alarms using robust interval estimation and machine learning. In 2015 Computing in Cardiology Conference (CinC). IEEE, 2015; 285–288.



Address for correspondence:

Szabolcs M. Péter
peter.szabolcs.matyas@hallgato.ppke.hu
AIMS-lab, PPKE ITK, Budapest, Hungary