pyPPG Documentation

Release 1.0.0

Author

Sep 23, 2023

CONTENTS:

CHAPTER

ONE

PYPPG TOOLBOX DOCUMENTATION

A toolbox for finger photoplethysmogram (PPG) analysis, including beat detection, fiducial point detection, and comprehensive assessment of standard biomarkers.

If you use the pyPPG resource, please cite:

Goda MA, Charlton PH, and Behar JA, **'pyPPG: A Python toolbox for comprehensive photoplethysmography signal analysis'***, [Under review]*

1.1 Introduction

pyPPG is a standardised toolbox to analyze long-term finger PPG recordings in real-time. The toolbox extracts state-of-the-art PPG biomarkers (*i.e.* pulse wave features) from PPG signals. The algorithms implemented in the *pyPPG* toolbox have been validated on freely available PPG databases. Consequently, *pyPPG* offers robust and comprehensive assessment of clinically relevant biomarkers from continuous PPG signals.

1.2 Description

The following steps are implemented in the **pyPPG** toolbox:

1. **Loading a raw PPG signal**: The toolbox can accept PPG signals in various file formats such as *.mat*, *.txt*, *.csv*, or *.edf*. These files should contain raw PPG data along with the corresponding sampling rate.
    - *.mat*: Data should be stored in two variables within the file: (i) 'Fs' representing the sampling frequency, and (ii) 'Data', a vector containing the raw PPG signal.
    - *.txt*: The raw PPG signal should be stored in tabular form (single tab or space-delimited), and you need to provide the sampling frequency as an input parameter to the script using 'fs'.
    - *.csv*: This format stores raw PPG signal data with comma separation. Similar to .txt, the sampling frequency must be provided as an input parameter to the script using 'fs'.
    - *.edf*: The European Data Format is supported, and it applies 'Pleth' channel by default. However, if using a different channel name, then the user needs to define it themselves.
2. **Preprocessing**: The raw signal is filtered to remove unwanted noise and artifacts. Subsequently, the signal is resampled to 75 Hz.
3. **Pulse wave segmentation**: The toolbox employs a peak detector to identify the systolic peaks. It uses an improved version of a beat detection algorithm originally proposed in (Aboy et al. 2005). Based on the peak locations, the toolbox also detects the pulse onsets and offsets, which indicate the start and end of the PPG pulse waves.
4. **Fiducial points identification**: For each pulse wave, the toolbox detects a set of fiducial points.

5. **Biomarker engineering**: Based on the fiducial points, a set of 74 PPG digital biomarkers (*i.e.* pulse wave features) are calculated.

The *pyPPG* toolbox also provides an optional PPG signal quality index based on the Matlab implementation of the work by (Li et al. 2015).

The toolbox identifies individual pulse waves in a PPG signal by identifying **systolic peaks (sp)**, and then identifying the **pulse onset (on)** and **offset (off)** on either side of each systolic peak which indicate the start and end of the pulse wave, respectively.

1.3 Installation

Available on pip, with the command: **pip install pyPPG**

pip project: https://pypi.org/project/pyPPG/

For more details see the pyPPG example code and pyPPG YouTube video

1.4 Requirements

1.4.1 Python Requirements:

Python == 3.10

scipy == 1.9.1

numpy == 1.23.2

dotmap == 1.3.30

pandas == 1.5.0

wfdb == 4.0.0

mne == 1.5.0

All the python requirements are installed when the toolbox is installed, so there is no need for any additional commands.

1.5 Documentation:

https://pyppg.readthedocs.io/en/latest/

1.6 The main components:

1. **Software**
   - An open-source algorithmic **pyPPG** toolbox, which loads a PPG signal, preprocesses it, segments individual pulse waves, identifies fiducial points, and calculates a set of biomarkers. This can be used within your own data analysis code using the **pyPPG** API.
2. **Databases**
   - The **pyPPG** signal analysis is based on the following datasets:
     - BIDMC Dataset
     - MESA Dataset
   - Further PPG datasets:
     - Collection of Peter Charlton
     - Collection of Physionet

All PPG measures can be further adapted for the analysis for efficient heart rate measurement as well as health assessment with clinically relevant biomarkers.

1.7 Acknowledgments

This work was supported by the Estate of Zofia (Sophie) Fridman and funding from the Israel Innovation Authority, and the British Heart Foundation (grant FS/20/20/34626).

1.7.1 pyPPG

pyPPG package

pyPPG.example

pyPPG.example.**ppg_example**(*data_path=''*, *fs=0*, *start_sig=0*, *end_sig=-1*, *fiducials=Empty DataFrame Columns: [] Index: []*, *process_type='both'*, *channel='Pleth'*, *filtering=True*, *fL=0.5000001*, *fH=12*, *order=4*, *sm_wins={'apg': 10, 'jpg': 10, 'ppg': 50, 'vpg': 10}*, *correction=Empty DataFrame Columns: [] Index: []*, *savingfolder='temp_dir'*, *savefig=True*, *show_fig=True*, *savingformat='csv'*, *print_flag=True*, *use_tk=False*, *check_ppg_len=True*)

This is an example code for PPG analysis. The main parts:

1) Loading a raw PPG signal: various file formats such as .mat, .csv, .txt, or .edf.
2) Get Fiducial points: extract the fiducial points of PPG, PPG', PPG" and PPG'" signals
3) Plot Fiducial Points
4) **Get Biomarkers: extract 74 PPG biomarkers in four categories:**
   - PPG signal
   - Signal ratios
   - PPG derivatives

- Derivatives ratios

5) Get Statistics: summary of the 74 PPG biomarkers
6) SQI calculation: calculates the PPG Signal Quality Index
7) Save data: save the extracted Fiducial points, Biomarkers, and Statistics into .csv file

Parameters

- **data_path** (*str*) – path of the PPG signal
- **fs** (*int*) – sampling_frequency
- **start_sig** (*int*) – beginning the of signal in sample
- **end_sig** (*int*) – end of the signal in sample
- **fiducials** (*pyPPG.Fiducials DataFrame*) – DataFrame of the fiducial points
- **process_type** (*str*) – the type of the process, which can be "fiducials", "biomarkers", or "both"
- **channel** (*channel of the .edf file*) – channel of the .edf file
- **filtering** (*bool*) – a bool for filtering
- **fL** (*float*) – Lower cutoff frequency (Hz)
- **fH** (*float*) – Upper cutoff frequency (Hz)
- **order** (*int*) – Filter order
- **sm_wins** (*dict*) – dictionary of smoothing windows in millisecond: - ppg: windows for PPG signal - vpg: windows for PPG' signal - apg: windows for PPG" signal - jpg: windows for PPG'" signal
- **correction** (*DataFrame*) – DataFrame where the key is the name of the fiducial points and the value is bool
- **savingfolder** (*str*) – location of the saved data
- **savefig** (*bool*) – a bool for current figure saving
- **show_fig** (*bool*) – a bool for show figure
- **savingformat** (*str*) – file format of the saved date, the provided file formats .mat and .csv
- **print_flag** (*bool*) – a bool for print message
- **use_tk** (*bool*) – a bool for using tkinter interface
- **check_ppg** (*bool*) – a bool for checking ppg length and sampling frequency

Returns

- fiducial points: DataFrame where the key is the name of the fiducial pints and the value is the list of fiducial points
- s: object of PPG signal

Example:

```
from pyPPG.example import ppg_example

# run example code
ppg_example(savedata=True, savefig=True)
```

pyPPG

class pyPPG.**PPG**(*s={}*, *check_ppg_len=True*)

Bases: `object`

This is class for the input PPG parameters.

Parameters

- **s** (*DotMap*) – dictionary of the PPG signal:
  - s.start_sig: beginning of the signal in sample
  - s.end_sig: end of the signal in sample
  - s.v: a vector of PPG values
  - s.fs: the sampling frequency of the PPG in Hz
  - s.name: name of the record
  - s.v: 1-d array, a vector of raw PPG values
  - s.fs: the sampling frequency of the PPG in Hz
  - s.ppg: 1-d array, a vector of the filtered PPG values
  - s.vpg: 1-d array, a vector of the filtered PPG' values
  - s.apg: 1-d array, a vector of the filtered PPG" values
  - s.jpg: 1-d array, a vector of the filtered PPG'" values
  - s.filtering: a bool for filtering
  - s.correction: DataFrame where the key is the name of the fiducial points and the value is bool
- **check_ppg_len** (*bool*) – a bool for checking ppg length and sampling frequency

get_s()

This function retrieves the dictionary of the PPG signal.

Returns

s: dictionary of the PPG signal

class pyPPG.**Fiducials**(*fp: DataFrame*)

Bases: `object`

This is class for the PPG fiducial points.

Parameters

fiducials (*DataFrame*) – DataFrame where the key is the name of the fiducial points and the value is the list of fiducial points PPG Fiducials Points

- PPG signal (fp.on, fp.sp, fp.dn, fp.dp): List of pulse onset, systolic peak, dicrotic notch, diastolic peak
- 1st derivative (fp.u, fp.v, fp.w): List of points of 1st maximum and minimum in 1st derivitive between the onset to onset intervals
- 2nd derivative (fp.a, fp.b, fp.c, fp.d, fp.e, fp.f): List of maximum and minimum points in 2nd derivitive between the onset to onset intervals
- 3rd derivative (fp.p1, fp.p2): List of points of 1st maximum and minimum in 3rd derivitive between the onset to onset intervals

get_fp()

This function retrieves the struct of the fiducial points.

Returns
fp: DataFrame of the fiducial points

get_row(*row_index: int*)

This function retrieves the specified row from the DataFrame of fiducial points.

Parameters
row_index (*int*) – the index corresponding to the row in the fiducial points DataFrame

Returns
the corresponding row in the fiducial points DataFrame

class pyPPG.**Biomarkers**(*bm_defs={}*, *bm_vals={}*, *bm_stats={}*)

Bases: object

This is class for the PPG biomarkers.

This class constitutes a comprehensive dictionary encompassing biomarker definitions, values, and statistics. Each dictionary is organized into the subsequent subdirectories:

- ppg_sig: description for the PPG signal
- sig_ratios: description for the Signal ratios
- ppg_derivs: description for the PPG derivatives
- derivs_ratios: description for the Derivatives ratios

Parameters

- **bm_defs** (*dict*) – dictionary with name, definition and unit of biomarkers in different categories:
- **bm_vals** (*dict*) – dictionary with values of biomarkers in different categories:
- **bm_stats** (*dict*) – data frame with summary of PPG features

get_bm()

This function retrieves the dictionary of the biomarkers.

Returns
bm: dictionary of the biomarkers

pyPPG.preproc

class pyPPG.preproc.**Preprocess**(*fL=0.5000001*, *fH=12*, *order=4*, *sm_wins={'apg': 10, 'jpg': 10, 'ppg': 50, 'vpg': 10}*)

Bases: object

The purpose of the Preprocess class is to filter and calculate the PPG, PPG', PPG", and PPG'" signals.

Parameters

- **fL** (*float*) – Lower cutoff frequency (Hz)
- **fH** (*float*) – Upper cutoff frequency (Hz)
- **order** (*int*) – Filter order

- **sm_wins** (*dict*) – dictionary of smoothing windows in millisecond: - ppg: windows for PPG signal - vpg: windows for PPG' signal - apg: windows for PPG" signal - jpg: windows for PPG'" signal

get_signals(*s: DotMap*)

This function calculates the preprocessed PPG, PPG', PPG", and PPG'" signals.

Parameters
s (*DotMap*) – a struct of PPG signal: - s.v: a vector of PPG values - s.fs: the sampling frequency of the PPG in Hz - s.filtering: a bool for filtering

Returns
ppg, vpg, apg, jpg: preprocessed PPG, PPG', PPG", and PPG'"

pyPPG.fiducials

class pyPPG.fiducials.**FpCollection**(*s:* PPG)

The purpose of the FiducialPoints class is to calculate the fiducial points.

Parameters
s (*pyPPG.PPG object*) – object of PPG signal

get_fiducials(*s:* PPG)

This function calculates the PPG Fiducial Points.

- Original signal: List of systolic peak, pulse onset, dicrotic notch, and diastolic peak
- 1st derivative: List of points of 1st maximum and minimum in 1st derivitive between the onset to onset intervals (u,v)
- 2nd derivative: List of maximum and minimum points in 2nd derivitive between the onset to onset intervals (a, b, c, d, e)

Parameters
s (*pyPPG.PPG object*) – object of PPG signal

Returns
fiducial points: DataFrame where the key is the name of the fiducial pints and the value is the list of fiducial points

pyPPG.biomarkers

class pyPPG.biomarkers.**BmCollection**(*s:* PPG, *fp:* Fiducials)

Bases: object

The purpose of the Biomarkers class is to calculate the PPG biomarkers.

Parameters

- **s** (*pyPPG.PPG object*) – object of PPG signal
- **fp** (*pyPPG.Fiducials object*) – object of fiducial points

get_biomarkers(*get_stat=True*)

This function retrieves the list of biomarkers, computes their values, and calculates associated statistics.

Parameters
get_stat (*bool*) – a bool for calculating the statistics of biomarkers

Returns

- bm_defs: dictionary of biomarkers with name, definition and unit
- bm_vals: dictionary of biomarkers with values
- bm_stats: dictionary of biomarkers with statistics

pyPPG.ppg_sqi

pyPPG.ppg_sqi.**get_ppgSQI**(*ppg: list*, *fs: int*, *annotation: list*)

PPG Signal Quality Index based on beat template correlation.

Parameters

- **ppg** (*int*) – a vector of PPG values
- **fs** (*int*) – Samples frequency
- **annotation** (*list*) – PPG annotation time(samples)

Returns

psqi: PPG Signal Quality Index

Reference:

- Original Matlab implementation: Qiao Li, November 10, 2014.
- Python implementation: Márton Áron Goda, PhD, November 11, 2022.

pyPPG.ppg_sqi.**use_template**(*wave*, *annotation: list*, *fs: int*)

PPG waveform template creation. Written by Qiao Li, February 21, 2011

Parameters

- **wave** – a vector of PPG wave
- **fs** (*int*) – Samples frequency
- **annotation** – PPG annotation time(sample)

Returns

- template: PPG waveform template based on normal - length beats
- valid: 1 for valid template, 0 for invalid template

pyPPG.datahandling

pyPPG.datahandling.**load_data**(*data_path=''*, *fs=nan*, *start_sig=0*, *end_sig=-1*, *channel='Pleth'*, *use_tk=True*)

Load raw PPG data.

Parameters

- **data_path** (*str*) – path of the file containing the PPG signal
- **start_sig** (*int*) – the first sample of the signal to be analysed
- **fs** (*int*) – the sampling frequency of the PPG in Hz
- **end_sig** (*int*) – the last sample of the signal to be analysed

- **channel** (*channel of the .edf file*) – channel of the .edf file
- **use_tk** (*bool*) – a bool for using tkinter interface

Returns

s: dictionary of the PPG signal:

- s.start_sig: the first sample of the signal to be analysed
- s.end_sig: the last sample of the signal to be analysed
- s.v: a vector of PPG values
- s.fs: the sampling frequency of the PPG in Hz
- s.name: name of the record
- s.v: 1-d array, a vector of PPG values
- s.fs: the sampling frequency of the PPG in Hz
- s.ppg: 1-d array, a vector of the filtered PPG values
- s.vpg: 1-d array, a vector of the filtered PPG' values
- s.apg: 1-d array, a vector of the filtered PPG" values
- s.jpg: 1-d array, a vector of the filtered PPG'" values
- s.filtering: a bool for filtering
- s.correct: a bool for correcting fiducial points

pyPPG.datahandling.**plot_fiducials**(*s:* PPG, *fp:* Fiducials, *savefig=True*, *savingfolder='temp_dir'*, *show_fig=True*, *print_flag=True*, *use_tk=False*, *new_fig=True*, *marker=[]*, *title='Detection'*, *legend_loc='upper right'*, *legend_fontsize=20*, *marker_size=60*, *facecolor=False*, *subtext={}*, *canvas=nan*)

Plot fiducial points of the filtered PPG signal.

Parameters

- **s** (*pyPPG.PPG object*) – object of PPG signal
- **fp** (*pyPPG.Fiducials object*) – object of fiducial points
- **savefig** (*bool*) – a bool for save figure
- **savingfolder** – location of the saved figure
- **show_fig** (*bool*) – a bool for show figure
- **print_flag** (*bool*) – a bool for print message
- **use_tk** (*bool*) – a bool for using tkinter interface
- **new_fig** (*bool*) – a bool for creating new figure
- **marker** (*list*) – list of fiducial points markers
- **title** (*str*) – title of the legend
- **legend_loc** (*str*) – location of the legend
- **legend_fontsize** (*int*) – fontsize of the legends
- **marker_size** (*int*) – size of markers
- **facecolor** (*bool*) – a bool for facecolor of markers

- **subtext** (*dict*) – dictionary for subplots text
- **canvas** (*FigureCanvas*) – canvas of the figure

pyPPG.datahandling.**save_data**(*s: PPG*, *fp: Fiducials*, *bm: Biomarkers*, *savingformat: str*, *savingfolder: str*, *print_flag=True*)

Save the results of the filtered PPG analysis.

Parameters

- **s** (*DotMap*) – a struct of PPG signal
- **fp** (*pyPPG.Fiducial object*) – a struct of fiducial points
- **bm** (*pyPPG.Biomarkers object*) – a dictionary of biomarkers
- **savingformat** (*str*) – file format of the saved date, the provided file formats .mat and .csv
- **savingfolder** (*str*) – location of the saved data
- **print_flag** (*bool*) – a bool for print message

Subpackages

pyPPG.ppg_bm package

pyPPG.ppg_bm.bm_extraction

class pyPPG.ppg_bm.bm_extraction.**BmExctator**(*data: DotMap*, *peak_value: float*, *peak_time: float*, *next_peak_value: float*, *next_peak_time: float*, *onsets_values: array*, *onsets_times: array*, *sample_rate: int*, *list_biomarkers: list*, *fiducials: DataFrame*)

Class that extracts the PPG biomarkers.

Parameters

- **data** (*DotMap*) – struct of PPG,PPG', PPG'', PPG''':
  - data.ppg: segment of PPG timeseries to analyse and extract biomarkers as a np array
  - data.vpg: segment of PPG'
  - data.apg: segment of PPG''
  - data.jpg: segment of PPG'''
- **peak_value** (*float*) – PPG peak value
- **peak_time** (*float*) – the time corresponding to the peak detected
- **next_peak_value** (*float*) – PPG next peak value
- **next_peak_time** (*float*) – the time corresponding to the peak detected
- **onsets_values** (*Numpy array*) – array of PPG two onsets values surrounding the peak
- **onsets_times** (*Numpy array*) – array of the two times corresponding to each onset detected
- **sample_rate** (*int*) – segment data sample rate
- **list_biomarkers** (*list*) – list of biomarkers
- **fiducials** (*DataFrame*) – location of fiducial points of the given pulse wave

pyPPG.ppg_bm.bm_extraction.**get_biomarkers**(*s:* PPG, *fp:* Fiducials, *biomarkers_lst*)

The function calculates the biomedical biomarkers of PPG signal.

Parameters

- **s** (*pyPPG.PPG object*) – object of PPG signal
- **fp** (*pyPPG.Fiducials object*) – object of fiducial points

Returns

- df: data frame with onsets, offset and peaks
- df_biomarkers: data frame with PPG signal biomarkers

pyPPG.ppg_bm.ppg_sig

pyPPG.ppg_bm.ppg_sig.**get_ppg_sig**(*s:* PPG, *fp:* Fiducials)

This function returns the biomarkers of PPG signal.

Parameters

- **s** (*pyPPG.PPG object*) – object of PPG signal
- **fp** (*pyPPG.Fiducials object*) – object of fiducial points

Returns

- biomarkers: dictionary of biomarkers of PPG signal
- biomarkers_lst: list a biomarkers with name, definition and unit

pyPPG.ppg_bm.sig_ratios

pyPPG.ppg_bm.sig_ratios.**get_sig_ratios**(*s:* PPG, *fp:* Fiducials)

This function returns the biomarkers of Signal ratios.

Parameters

- **s** (*pyPPG.PPG object*) – object of PPG signal
- **fp** (*pyPPG.Fiducials object*) – object of fiducial points

Returns

- biomarkers: dictionary of biomarkers of Signal ratios
- biomarkers_lst: list a biomarkers with name, definition and unit

pyPPG.ppg_bm.ppg_derivs

pyPPG.ppg_bm.ppg_derivs.**get_ppg_derivs**(*s:* PPG, *fp:* Fiducials)

This function returns the biomarkers of PPG derivatives.

Parameters

- **s** (*pyPPG.PPG object*) – object of PPG signal
- **fp** (*pyPPG.Fiducials object*) – object of fiducial points

Returns

- biomarkers: dictionary of biomarkers of PPG derivatives
- biomarkers_lst: list a biomarkers with name, definition and unit

pyPPG.ppg_bm.derivs_ratios

pyPPG.ppg_bm.derivs_ratios.**get_derivs_ratios**(*s:* PPG, *fp:* Fiducials)

This function returns the biomarkers of Derivatives ratios.

Parameters

- **s** (*pyPPG.PPG object*) – object of PPG signal
- **fp** (*pyPPG.Fiducials object*) – object of fiducial points

Returns

- df_biomarkers: dictionary of biomarkers of Derivatives ratios
- biomarkers_lst: list a biomarkers with name, definition and unit

pyPPG.ppg_bm.statistics

pyPPG.ppg_bm.statistics.**get_statistics**(*peaks: Series*, *onsets: Series*, *ppg_biomarkers: dict*)

The function compares the different biomedical features of PPG signal.

Parameters

- **peaks** (*Series*) – 1-d array, peaks of the signal
- **onsets** (*Series*) – 1-d array, onsets of the signal
- **ppg_biomarkers** (*dict*) – dictionary of the PPG biomarkers

Returns

df_windows: data frame with summary of PPG features

1.7.2 pyPPG example code

In this tutorial you will learn how to use **pyPPG** to engineer morphological PPG biomarkers and export their values.

Introduction

This tutorial provides step-by-step instructions for installing pyPPG and running the example code.

Step 1: Install Python 3.10

Download and install Python 3.10 on your computer or server by visiting the official Python website: Python 3.10.

Step 2: Download the Sample PPG Data

You can use the sample PPG data by downloading it from the following link: Sample PPG data.

Step 3: Create and Activate a Virtual Environment

Create a virtual environment named “ppgenv” specifically for Python 3.10 using the py launcher:

```
py -3.10 -m venv ppgenv
```

Activate the virtual environment:

```
ppgenv\Scripts\activate
```

Step 4: Install pyPPG

While the virtual environment is active, install pyPPG using pip:

```
ppgenv\Scripts\python.exe -m pip install pyPPG
```

WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead, or please review carefully the list of package requirements.

Step 5: Run the Example Code

Open the Python interpreter:

```
python
```

Run the example code, load the example files (*.mat*, *.txt*, *.csv*, or *.edf* formats) and check the results:

```
from pyPPG.example import ppg_example
ppg_example(savedata=True, savefig=True)
```

The resulting figures and outcomes are stored within the *temp_dir* folder, which is automatically generated within the project directory.

Step 6: Exit the Python Interpreter and Deactivate the Virtual Environment

To exit the Python interpreter, type:

```
exit()
```

Deactivate the virtual environment:

```
deactivate
```

You have successfully installed pyPPG, executed the example code, and explored the results. Feel free to customize and use pyPPG for your projects.

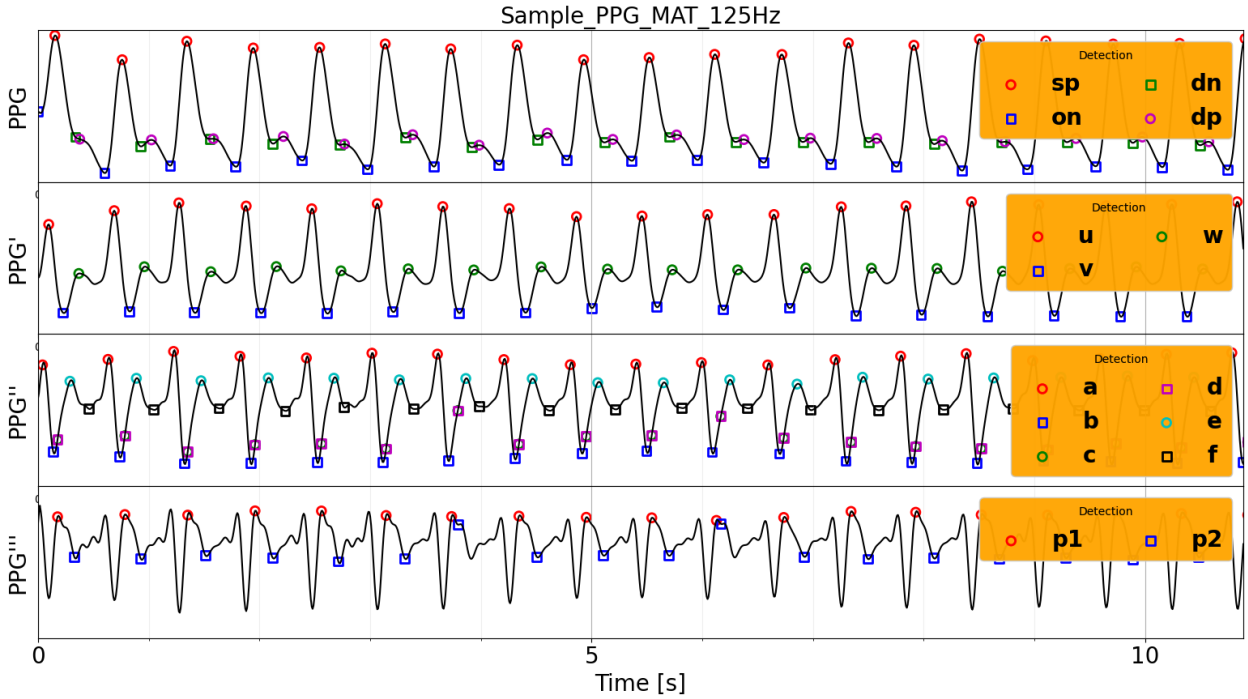
Example pyPPG code

The provided example code consists of seven modules that effectively showcase the capabilities of the pyPPG toolbox.

1. **Raw PPG Signal Loading**: This module facilitates the loading of raw PPG signals from various file formats, including .mat, .csv, .txt, or .edf.
2. **Fiducial Point Extraction**: This module focuses on extracting fiducial points from PPG signals, encompassing PPG, PPG', PPG", and PPG"'.
3. **Fiducial Points Plotting**: Here, the extracted fiducial points are visually represented through plotting.
4. **Biomarker Extraction**: This module offers the extraction of 74 distinct PPG biomarkers, categorized into:
    I. PPG signal characteristics
    II. Signal ratios
    III. PPG derivatives

IV. Derivative ratios

5. **Biomarker Statistics**: A concise summary of the 74 PPG biomarkers is provided within this module.
6. **SQI calculation**: This module calculates the PPG Signal Quality Index based on beat template correlation.
7. **Save data**: This module allows for the saving of extracted Fiducial points, Biomarkers, and Statistics into a .csv file.



1.7.3 Comprehensive PPG Analysis

In this tutorial we will learn how to extract biomarkers from a photoplethysmogram (PPG) signal.

Our objectives are to:

- Detect the standard fiducial points on PPG pulse waves
- Calculate pulse wave biomarkers from the fiducial points
- Saving data in different data format

You can use the sample PPG data by downloading it from the following link: Sample PPG data.

Setup

Import Python packages:

- Install the pyPPG toolbox for PPG analysis

```
pip install pyPPG==1.0.41
```

- Import required components from pyPPG

```
from pyPPG import PPG, Fiducials, Biomarkers
from pyPPG.datahandling import load_data, plot_fiducials, save_data
import pyPPG.preproc as PP
import pyPPG.fiducials as FP
import pyPPG.biomarkers as BM
import pyPPG.ppg_sqi as SQI
```

- Import other packages

```
import numpy as np
import sys
import json
import pandas as pd
```

Setup input parameters:

The following input parameters are inputs to the pyPPG.example module (see the documentation for further details).

```
data_path = "Sample_PPG_MAT_125Hz.mat" # the path of the file containing the PPG signal␣
↪to be analysed
start_sig = 0 # the first sample of the signal to be analysed
end_sig = -1 # the last sample of the signal to be analysed (here a value of '-1'␣
↪indicates the last sample)
savingfolder = 'temp_dir'
savingformat = 'csv'
```

Loading a raw PPG signal:

```
# Load the raw PPG signal
signal = load_data(data_path=data_path, start_sig=start_sig, end_sig=end_sig, use_
↪tk=False)
signal.v = signal.v [0:20*signal.fs] # 20 second long signal to be analysed
```

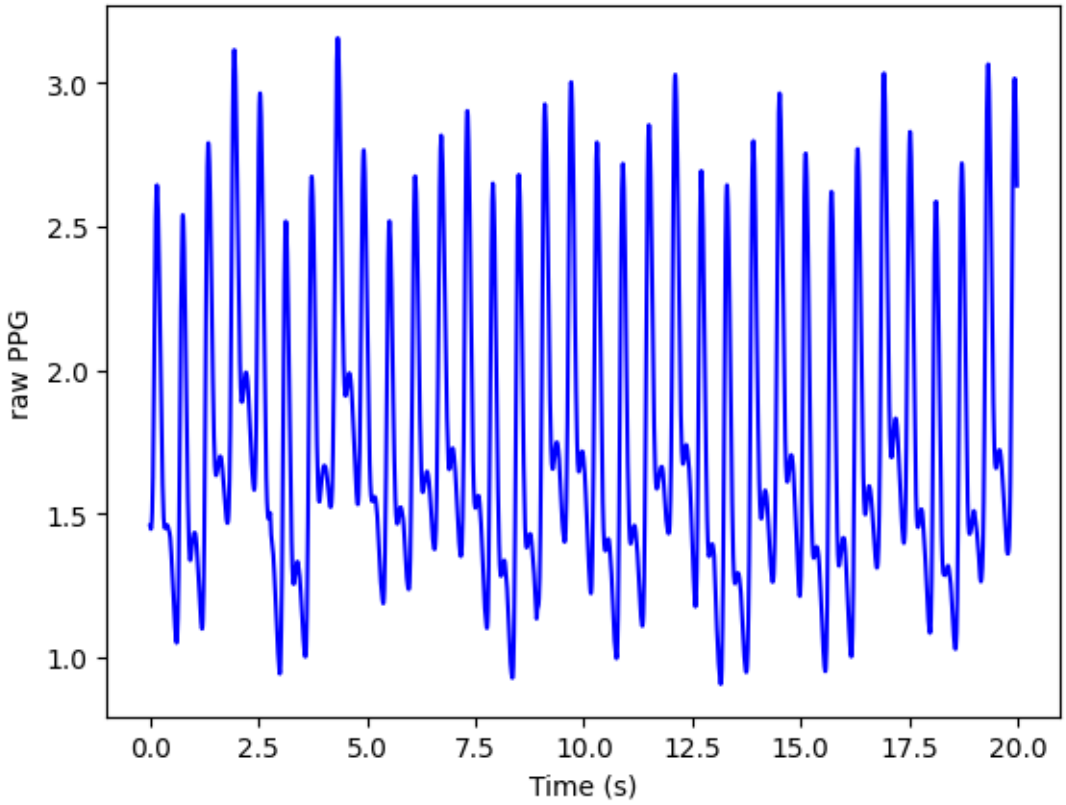
Plot the raw PPG signal:

```
# import plotting package
from matplotlib import pyplot as plt

# setup figure
fig, ax = plt.subplots()

# create time vector
t = np.arange(0, len(signal.v))/signal.fs

# plot raw PPG signal
ax.plot(t, signal.v, color = 'blue')
ax.set(xlabel = 'Time (s)', ylabel = 'raw PPG')
```

(continues on next page)

(continued from previous page)

```
# show plot
plt.show()
```



PPG signal processing

Prepare the PPG data:

Filter the PPG signal and obtain first, second and third derivatives (vpg, apg, and jpg respectively).

```
signal.filtering = True # whether or not to filter the PPG signal
signal.fL=0.5000001 # Lower cutoff frequency (Hz)
signal.fH=12 # Upper cutoff frequency (Hz)
signal.order=4 # Filter order
signal.sm_wins={'ppg':50,'vpg':10,'apg':10,'jpg':10} # smoothing windows in millisecond␣
↪for the PPG, PPG', PPG'', and PPG'''

prep = PP.Preprocess(fL=signal.fL, fH=signal.fH, order=signal.order, sm_wins=signal.sm_
↪wins)
signal.ppg, signal.vpg, signal.apg, signal.jpg = prep.get_signals(s=signal)
```

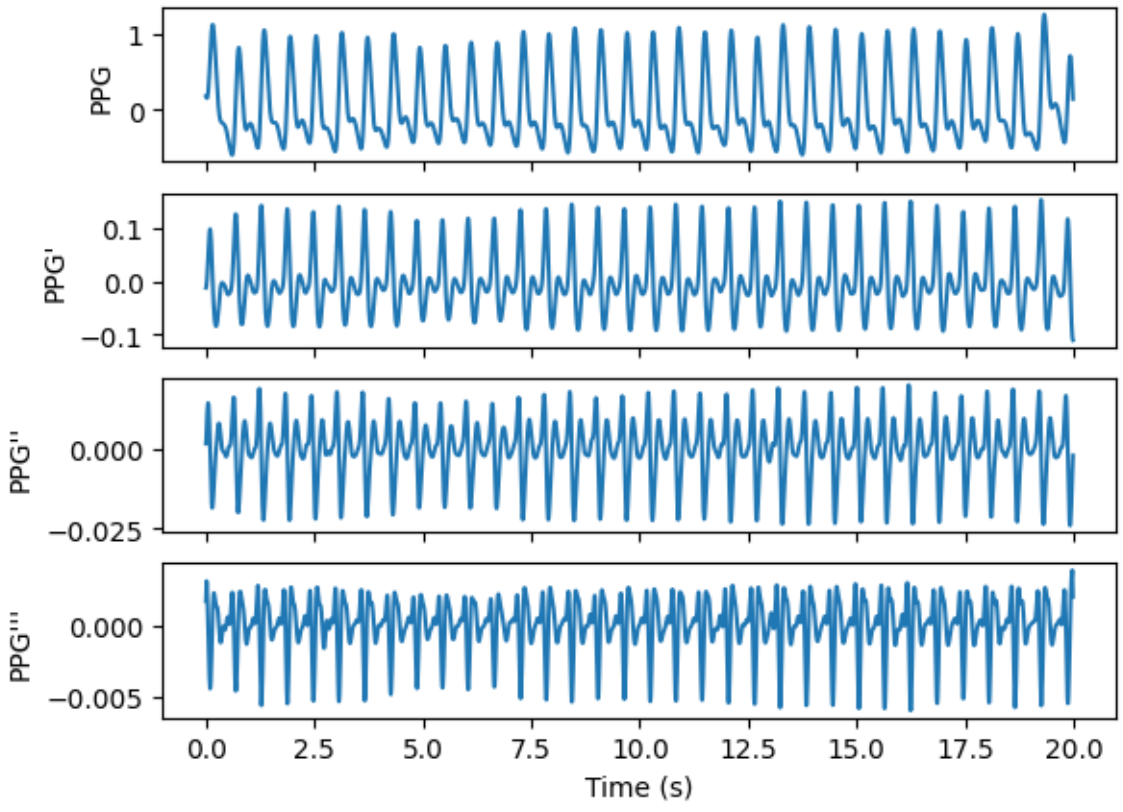
Plot the derived signals

```
# setup figure
fig, (ax1,ax2,ax3,ax4) = plt.subplots(4, 1, sharex = True, sharey = False)
```

(continues on next page)

(continued from previous page)

```
# create time vector
t = np.arange(0, len(signal.ppg))/signal.fs

# plot filtered PPG signal
ax1.plot(t, signal.ppg)
ax1.set(xlabel = '', ylabel = 'PPG')

# plot first derivative
ax2.plot(t, signal.vpg)
ax2.set(xlabel = '', ylabel = 'PPG\'')

# plot second derivative
ax3.plot(t, signal.apg)
ax3.set(xlabel = '', ylabel = 'PPG\'\'')

# plot third derivative
ax4.plot(t, signal.jpg)
ax4.set(xlabel = 'Time (s)', ylabel = 'PPG\'\'\'')

# show plot
plt.show()
```



Store the derived signals in a class

```
# Initialise the correction for fiducial points
corr_on = ['on', 'dn', 'dp', 'v', 'w', 'f']
correction=pd.DataFrame()
correction.loc[0, corr_on] = True
signal.correction=correction

# Create a PPG class
s = PPG(signal)
```

Identify fiducial points:

Initialise the fiducials package

```
fpex = FP.FpCollection(s=s)
```

Extract fiducial points

```
fiducials = fpex.get_fiducials(s=s)
```

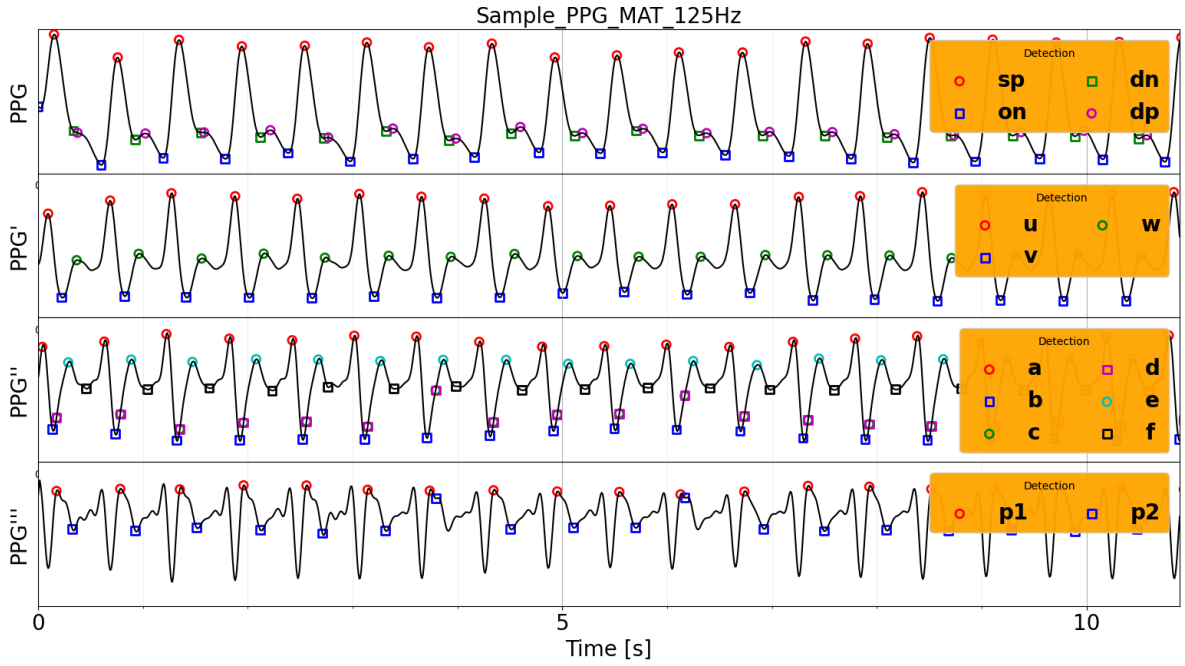
Display the results

```
print("Fiducial points:\n",fiducials + s.start_sig) # here the starting sample is added␣
↪so that the results are relative to the start of the original signal (rather than the␣
↪start of the analysed segment)
```

Plot fiducial points:

```
# Create a fiducials class
fp = Fiducials(fp=fiducials)

# Plot fiducial points
plot_fiducials(s, fp, savingfolder, legend_fontsize=12)
```

PPG fiducial points



Calculate PPG SQI:

```
# Get PPG SQI
ppgSQI = round(np.mean(SQI.get_ppgSQI(ppg=s.ppg, fs=s.fs, annotation=fp.sp)) * 100, 2)
print('Mean PPG SQI: ', ppgSQI, '%')
```

Calculate PPG biomarkers:

```
# Init the biomarkers package
bmex = BM.BmCollection(s=s, fp=fp)

# Extract biomarkers
bm_defs, bm_vals, bm_stats = bmex.get_biomarkers()
tmp_keys=bm_stats.keys()
print('Statistics of the biomarkers:')
for i in tmp_keys: print(i,'\n',bm_stats[i])

# Create a biomarkers class
bm = Biomarkers(bm_defs=bm_defs, bm_vals=bm_vals, bm_stats=bm_stats)
```

Save PPG data:

```
# Save PPG struct, fiducial points, biomarkers
fp_new = Fiducials(fp.get_fp() + s.start_sig) # here the starting sample is added so␣
↪that the results are relative to the start of the original signal (rather than the␣
↪start of the analysed segment)
save_data(s=s, fp=fp_new, bm=bm, savingformat=savingformat, savingfolder=savingfolder)
```

Extracted fiducial points

Index of pulse	on	sp	dn	dp	off	u	v	w	a	b	c	d	e	f	p1	p2
1	0	19	42	47	75	12	28	46	5	17	22	22	36	57	22	41
2	75	95	116	128	149	86	103	120	79	92	98	98	111	130	98	116
3	149	168	194	198	223	159	176	195	153	165	169	169	184	204	169	189
4	223	243	265	277	298	235	251	269	228	240	245	245	260	279	245	265
5	298	318	341	346	372	309	326	342	303	315	320	320	334	345	320	339
6	372	392	415	423	447	383	400	418	377	389	393	393	408	424	393	414
7	447	466	490	498	520	457	475	492	451	463	474	474	483	498	467	474
8	520	541	564	575	597	532	550	568	526	538	543	543	558	577	543	563
9	597	616	640	649	670	608	625	643	601	614	619	619	632	652	619	638
10	670	690	713	721	744	682	698	716	675	687	693	693	706	727	693	712
11	744	764	788	797	819	756	773	790	749	761	771	771	781	801	766	771
12	819	840	864	872	895	831	848	867	824	837	842	842	857	872	842	865
13	895	915	938	947	969	907	923	941	900	912	918	918	931	949	918	937
14	969	989	1012	1021	1043	980	997	1015	974	986	991	991	1005	1022	991	1011
15	1043	1063	1088	1092	1117	1054	1072	1089	1048	1060	1065	1065	1079	1101	1065	1085
16	1117	1138	1162	1172	1194	1130	1147	1165	1123	1135	1140	1140	1155	1175	1140	1160
17	1194	1214	1236	1247	1269	1206	1222	1240	1199	1211	1217	1217	1230	1249	1217	1236
18	1269	1289	1312	1322	1343	1280	1297	1315	1274	1286	1291	1291	1305	1324	1291	1310
19	1343	1363	1387	1394	1418	1354	1371	1389	1348	1360	1366	1366	1379	1399	1366	1384
20	1418	1438	1463	1473	1494	1430	1447	1466	1423	1435	1441	1441	1454	1476	1441	1459
21	1494	1514	1537	1547	1571	1506	1522	1540	1500	1512	1517	1517	1530	1546	1517	1537
22	1571	1590	1612	1621	1643	1581	1598	1615	1575	1587	1591	1591	1606	1623	1591	1611
23	1643	1664	1688	1693	1718	1655	1672	1689	1648	1660	1665	1665	1680	1701	1665	1685
24	1718	1739	1763	1772	1795	1731	1748	1766	1724	1736	1741	1741	1756	1775	1741	1762
25	1795	1815	1837	1849	1872	1807	1823	1841	1800	1812	1817	1817	1831	1851	1817	1837
26	1872	1890	1913	1923	1945	1882	1899	1916	1875	1887	1898	1898	1907	1927	1892	1898
27	1945	1965	1987	1998	2021	1956	1973	1990	1950	1961	1972	1972	1981	2001	1966	1972
28	2021	2040	2063	2071	2095	2031	2049	2066	2025	2036	2048	2048	2057	2071	2041	2048
29	2095	2116	2138	2149	2172	2107	2124	2142	2101	2113	2118	2118	2132	2149	2118	2138

Extracted biomarkers

Index of pulse	Tu/Tpi	Tv/Tpi	Tw/Tpi	Ta/Tpi	Tb/Tpi	Tc/Tpi	Td/Tpi	Te/Tpi	Tf/Tpi	(Tu-Ta)/Tpi	(Tv-Tb)/Tpi	Au/Asp	Av/Au	Aw/Au	Ab/Aa	Ac/Aa	Ad/Aa	Ae/Aa	Af/Aa	Ap2/Ap1	(Ac-Ab)/Aa	(Ad-Ab)/Aa	AGI	AGImod	AGIinf	AI	RIp1	RIp2	SC	IPAD
1	0.16	0.37	0.61	0.07	0.22	0.29	0.29	0.47	0.75	0.09	0.15	0.09	-0.85	-0.03	-1.30	-0.99	-0.99	0.57	-0.16	-0.13	0.31	0.31	0.11	0.67	-1.86	-1.07	4.70	4.70	-0.06	5.30
2	0.15	0.38	0.62	0.05	0.23	0.32	0.32	0.49	0.75	0.09	0.15	0.15	-0.64	0.09	-1.25	-0.77	-0.77	0.56	-0.16	-0.37	0.49	0.49	-0.28	0.28	-1.82	-1.22	-0.36	-0.36	-0.02	-3.37
3	0.13	0.36	0.61	0.05	0.21	0.27	0.27	0.47	0.73	0.08	0.15	0.14	-0.58	0.00	-1.19	-0.96	-0.96	0.43	-0.11	-0.14	0.23	0.23	0.31	0.74	-1.61	-1.13	-0.34	-0.34	-0.02	-1.58
4	0.16	0.37	0.61	0.07	0.23	0.29	0.29	0.49	0.75	0.09	0.15	0.14	-0.62	0.08	-1.31	-0.91	-0.91	0.53	-0.18	-0.25	0.40	0.40	-0.02	0.51	-1.84	-1.20	-0.35	-0.35	-0.03	-7.82
5	0.15	0.38	0.59	0.07	0.23	0.30	0.30	0.49	0.64	0.08	0.15	0.13	-0.65	0.02	-1.33	-0.92	-0.92	0.55	-0.10	-0.26	0.41	0.41	-0.04	0.51	-1.88	-1.20	0.15	0.15	-0.02	-0.83
6	0.15	0.38	0.62	0.07	0.23	0.28	0.28	0.49	0.70	0.08	0.15	0.14	-0.58	0.05	-1.20	-0.94	-0.94	0.46	-0.13	-0.16	0.27	0.27	0.21	0.67	-1.66	-1.14	-0.56	-0.56	-0.02	-3.46
7	0.13	0.37	0.60	0.05	0.21	0.36	0.36	0.48	0.68	0.08	0.16	0.14	-0.63	0.03	-1.20	-0.17	-0.17	0.50	-0.08	0.57	1.03	1.03	-1.36	-0.86	-1.69	-0.43	0.08	0.08	-0.02	-4.34
8	0.16	0.40	0.64	0.08	0.24	0.31	0.31	0.51	0.76	0.08	0.16	0.13	-0.64	0.08	-1.29	-0.97	-0.97	0.56	-0.19	-0.19	0.32	0.32	0.09	0.64	-1.85	-1.15	-0.41	-0.41	-0.02	3.23
9	0.15	0.38	0.62	0.05	0.23	0.30	0.30	0.47	0.74	0.10	0.15	0.14	-0.65	0.05	-1.34	-0.89	-0.89	0.53	-0.18	-0.28	0.45	0.45	-0.08	0.44	-1.87	-1.20	0.03	0.03	-0.02	-1.35
10	0.16	0.38	0.62	0.07	0.23	0.31	0.31	0.49	0.77	0.09	0.15	0.14	-0.62	0.04	-1.26	-0.85	-0.85	0.51	-0.14	-0.19	0.41	0.41	-0.08	0.44	-1.78	-1.11	-0.31	-0.31	-0.02	-2.98
11	0.16	0.38	0.61	0.07	0.22	0.36	0.36	0.49	0.75	0.09	0.16	0.13	-0.65	0.04	-1.23	-0.34	-0.34	0.57	-0.14	0.70	0.89	0.89	-1.12	-0.55	-1.80	-0.29	-0.04	-0.04	-0.02	-0.25
12	0.16	0.39	0.64	0.07	0.24	0.31	0.31	0.51	0.71	0.09	0.14	0.13	-0.63	0.07	-1.36	-0.94	-0.94	0.49	-0.19	-0.25	0.41	0.41	0.04	0.53	-1.85	-1.19	-0.03	-0.03	-0.02	-1.19
13	0.16	0.38	0.62	0.07	0.23	0.31	0.31	0.49	0.73	0.09	0.15	0.13	-0.67	0.06	-1.36	-0.92	-0.92	0.59	-0.18	-0.23	0.44	0.44	-0.11	0.48	-1.95	-1.14	0.09	0.09	-0.03	-0.78
14	0.15	0.38	0.62	0.07	0.23	0.30	0.30	0.49	0.72	0.08	0.15	0.14	-0.64	0.05	-1.29	-0.94	-0.94	0.51	-0.15	-0.25	0.34	0.34	0.09	0.60	-1.79	-1.20	0.13	0.13	-0.02	-0.67
15	0.15	0.39	0.61	0.07	0.23	0.29	0.29	0.48	0.77	0.08	0.16	0.13	-0.63	0.01	-1.23	-0.94	-0.94	0.50	-0.12	-0.20	0.29	0.29	0.15	0.65	-1.74	-1.17	-0.28	-0.28	-0.02	-9.05
16	0.17	0.39	0.63	0.08	0.24	0.30	0.30	0.50	0.76	0.09	0.16	0.13	-0.65	0.05	-1.38	-1.12	-1.12	0.60	-0.18	-0.20	0.26	0.26	0.26	0.86	-1.98	-1.18	-0.24	-0.24	-0.03	0.91
17	0.16	0.37	0.61	0.07	0.23	0.31	0.31	0.48	0.73	0.09	0.15	0.13	-0.67	0.08	-1.33	-0.92	-0.92	0.60	-0.17	-0.24	0.41	0.41	-0.09	0.51	-1.93	-1.15	-0.04	-0.04	-0.03	-1.15
18	0.15	0.38	0.62	0.07	0.23	0.30	0.30	0.49	0.74	0.08	0.15	0.14	-0.66	0.04	-1.25	-0.93	-0.93	0.52	-0.17	-0.25	0.32	0.32	0.08	0.61	-1.78	-1.20	0.12	0.12	-0.02	-0.70
19	0.15	0.37	0.61	0.07	0.23	0.31	0.31	0.48	0.75	0.08	0.15	0.13	-0.63	0.03	-1.20	-0.79	-0.79	0.52	-0.13	-0.18	0.41	0.41	-0.13	0.39	-1.72	-1.11	-0.23	-0.23	-0.02	-0.99
20	0.16	0.38	0.63	0.07	0.22	0.30	0.30	0.47	0.76	0.09	0.16	0.14	-0.65	0.03	-1.24	-0.86	-0.86	0.54	-0.15	-0.24	0.38	0.38	-0.05	0.49	-1.77	-1.17	-0.44	-0.44	-0.03	0.08
21	0.16	0.37	0.61	0.08	0.24	0.30	0.30	0.47	0.68	0.08	0.13	0.13	-0.65	0.07	-1.33	-0.86	-0.86	0.52	-0.14	-0.20	0.47	0.47	-0.12	0.40	-1.85	-1.12	0.11	0.11	-0.03	-0.30
22	0.14	0.36	0.59	0.05	0.22	0.27	0.27	0.47	0.70	0.08	0.15	0.15	-0.62	0.06	-1.21	-0.95	-0.95	0.49	-0.22	-0.23	0.26	0.26	0.20	0.69	-1.70	-1.21	-0.07	-0.07	-0.02	-1.49
23	0.16	0.39	0.61	0.07	0.23	0.29	0.29	0.49	0.77	0.09	0.16	0.13	-0.62	0.01	-1.24	-0.98	-0.98	0.52	-0.12	-0.18	0.26	0.26	0.21	0.72	-1.76	-1.16	-0.10	-0.10	-0.03	-0.66
24	0.17	0.39	0.63	0.08	0.24	0.30	0.30	0.50	0.75	0.09	0.16	0.14	-0.62	0.04	-1.32	-0.98	-0.98	0.51	-0.16	-0.22	0.34	0.34	0.13	0.64	-1.82	-1.19	-0.28	-0.28	-0.03	0.30
25	0.16	0.37	0.61	0.07	0.23	0.29	0.29	0.48	0.75	0.09	0.14	0.14	-0.64	0.08	-1.33	-0.97	-0.97	0.55	-0.17	-0.22	0.36	0.36	0.06	0.61	-1.88	-1.17	-0.28	-0.28	-0.03	1.09
26	0.13	0.36	0.59	0.04	0.20	0.35	0.35	0.47	0.73	0.10	0.16	0.14	-0.62	0.05	-1.21	-0.19	-0.19	0.52	-0.16	0.60	1.02	1.02	-1.35	-0.83	-1.73	-0.39	-0.04	-0.04	-0.03	0.25
27	0.15	0.37	0.60	0.07	0.21	0.36	0.36	0.48	0.75	0.08	0.16	0.14	-0.59	0.05	-1.15	-0.20	-0.20	0.50	-0.17	0.64	0.95	0.95	-1.25	-0.75	-1.66	-0.35	0.03	0.03	-0.02	-0.18
28	0.13	0.37	0.59	0.05	0.20	0.36	0.36	0.47	0.66	0.08	0.18	0.14	-0.60	0.04	-1.11	-0.12	-0.12	0.47	-0.11	0.56	0.99	0.99	-1.34	-0.87	-1.59	-0.43	0.12	0.12	-0.02	1.71
29	0.16	0.39	0.63	0.08	0.24	0.31	0.31	0.49	0.72	0.08	0.14	0.14	-0.64	0.10	-1.30	-0.89	-0.89	0.56	-0.18	-0.22	0.40	0.40	-0.07	0.49	-1.86	-1.17	-0.47	-0.47	-0.03	0.12
30	0.15	0.37	0.60	0.05	0.21	0.36	0.36	0.48	0.69	0.09	0.16	0.14	-0.66	0.09	-1.28	-0.20	-0.20	0.61	-0.17	0.59	1.08	1.08	-1.49	-0.88	-1.89	-0.40	0.00	0.00	-0.03	0.17

Biomarkers statistics

Statistics	Tu/Tpi	Tv/Tpi	Tw/Tpi	Ta/Tpi	Tb/Tpi	Tc/Tpi	Td/Tpi	Te/Tpi	Tf/Tpi	(Tu-Ta)/Tpi	(Tv-Tb)/Tpi	Au/Asp	Av/Au	Aw/Au	Ab/Aa	Ac/Aa	Ad/Aa	Ae/Aa	Af/Aa	Ap2/Ap1	(Ac-Ab)/Aa	(Ad-Ab)/Aa	AGI	AGImod	AGIinf	AI	RIp1	RIp2	SC	IPAD
mean	0.15	0.38	0.61	0.06	0.22	0.32	0.32	0.48	0.71	0.09	0.15	0.14	-0.65	0.06	-1.26	-0.64	-0.64	0.56	-0.15	0.22	0.62	0.62	-0.54	0.02	-1.82	-0.72	-0.05	-0.05	-0.03	-0.11
median	0.15	0.38	0.61	0.07	0.23	0.33	0.33	0.48	0.72	0.09	0.15	0.14	-0.65	0.05	-1.25	-0.67	-0.67	0.55	-0.15	-0.16	0.55	0.55	-0.45	0.15	-1.80	-0.81	0.01	0.01	-0.03	0.03
std	0.01	0.01	0.01	0.01	0.01	0.03	0.03	0.01	0.04	0.01	0.01	0.01	0.04	0.03	0.08	0.31	0.31	0.06	0.04	0.49	0.27	0.27	0.57	0.57	0.13	0.45	0.46	0.46	0.00	1.30
percentile_25	0.14	0.37	0.60	0.05	0.22	0.30	0.30	0.47	0.69	0.08	0.14	0.13	-0.67	0.04	-1.32	-0.93	-0.93	0.52	-0.17	-0.23	0.38	0.38	-1.07	-0.50	-1.92	-1.17	-0.21	-0.21	-0.03	-0.25
percentile_75	0.16	0.39	0.61	0.07	0.23	0.35	0.35	0.49	0.74	0.09	0.16	0.14	-0.62	0.07	-1.20	-0.35	-0.35	0.60	-0.12	0.72	0.86	0.86	-0.02	0.55	-1.73	-0.28	0.11	0.11	-0.02	0.37
iqr	0.02	0.01	0.02	0.01	0.01	0.05	0.05	0.01	0.05	0.01	0.02	0.01	0.04	0.04	0.11	0.58	0.58	0.08	0.05	0.96	0.49	0.49	1.04	1.04	0.19	0.89	0.31	0.31	0.00	0.61
skew	0.51	0.17	-0.20	0.74	0.34	0.01	0.01	0.37	-0.57	-0.26	0.02	-2.61	-0.88	0.29	-0.33	0.17	0.17	0.64	-0.46	0.10	0.33	0.33	-0.22	-0.26	-0.33	0.09	5.70	5.70	-5.54	-2.94
kurtosis	0.33	0.34	0.08	1.29	0.28	-0.80	-0.80	0.06	-0.50	-1.64	-0.44	21.57	3.31	2.59	-0.16	-1.53	-1.53	0.61	1.61	-1.87	-1.18	-1.18	-1.33	-1.41	-0.22	-1.87	61.33	61.33	53.61	19.38
mad	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.01	0.03	0.01	0.01	0.00	0.03	0.02	0.07	0.29	0.29	0.05	0.03	0.49	0.24	0.24	0.52	0.53	0.10	0.44	0.24	0.24	0.00	0.66

Biomarkers definitions

	A	B	C	D
1	**No. biomarkers**	**name**	**definition**	**unit**
2	1	Tu/Tpi	Ratio of the u-point time vs. the pulse interval	[%]
3	2	Tv/Tpi	Ratio of the v-point time vs. the pulse interval	[%]
4	3	Tw/Tpi	Ratio of the w-point time vs. the pulse interval	[%]
5	4	Ta/Tpi	Ratio of the a-point time vs. the pulse interval	[%]
6	5	Tb/Tpi	Ratio of the b-point time vs. the pulse interval	[%]
7	6	Tc/Tpi	Ratio of the c-point time vs. the pulse interval	[%]
8	7	Td/Tpi	Ratio of the d-point time vs. the pulse interval	[%]
9	8	Te/Tpi	Ratio of the e-point time vs. the pulse interval	[%]
10	9	Tf/Tpi	Ratio of the f-point time vs. the pulse interval	[%]
11	10	(Tu-Ta)/Tpi	Ratio of the difference between the u-point time and a-point time vs. the pulse interval	[%]
12	11	(Tv-Tb)/Tpi	Ratio of the difference between the v-point time and b-point time vs. the pulse interval	[%]
13	12	Au/Asp	Ratio of the u-point amplitude vs. the systolic peak amplitude	[%]
14	13	Av/Au	Ratio of the v-point amplitude vs. the u-point amplitude	[%]
15	14	Aw/Au	Ratio of the w-point amplitude vs. the u-point amplitude	[%]
16	15	Ab/Aa	Ratio of the b-point amplitude vs. the a-point amplitude	[%]
17	16	Ac/Aa	Ratio of the c-point amplitude vs. the a-point amplitude	[%]
18	17	Ad/Aa	Ratio of the d-point amplitude vs. the a-point amplitude	[%]
19	18	Ae/Aa	Ratio of the e-point amplitude vs. the a-point amplitude	[%]
20	19	Af/Aa	Ratio of the f-point amplitude vs. the a-point amplitude	[%]
21	20	Ap2/Ap1	Ratio of the p2-point amplitude vs. the p1-point amplitude	[%]
22	21	(Ac-Ab)/Aa	Ratio of the difference between the b-point amplitude and c-point amplitude vs. the a-point amplitude	[%]
23	22	(Ad-Ab)/Aa	Ratio of the difference between the b-point amplitude and d-point amplitude vs. the a-point amplitude	[%]
24	23	AGI	Aging Index, (Ab-Ac-Ad-Ae)/Aa	[%]
25	24	AGImod	Modified aging index, (Ab-Ac-Ad)/Aa	[%]
26	25	AGIinf	Informal aging index, (Ab-Ae)/Aa	[%]
27	26	AI	Augmentation index, (PPG(Tp2)-PPG(Tp1))/Asp	[%]
28	27	RIp1	Reflection index of p1, Adp/(PPG(Tp1)-PPG(Tpi(0)))	[%]
29	28	RIp2	Reflection index of p2, Adp/(PPG(p2)-PPG(Tpi(0)))	[%]
30	29	SC	Spring constant, PPG''(Tsp)/((Asp-Au)/Asp)	[nu]
31	30	IPAD	Inflection point area plus normalised d-point amplitude, AUCdia/AUCsys+Ad/Aa	[nu]

1.7.4 PhysioZoo PPG analysis

In this tutorial you will learn how to use **PhysioZoo PPG** to calculate morphological PPG biomarkers (i.e. pulse wave features) and export their values.

If you use the pyPPG resource, please cite:

Goda MA, Charlton PH, and Behar JA, '**pyPPG: A Python toolbox for comprehensive photoplethysmography signal analysis**', [Under review]

Introduction

The PPG signal is an optical measurement of the arterial pulse wave [1], *i.e.*, the wave generated when blood is ejected from the heart, temporarily increasing arterial pressure and causing vessel expansion and contraction [2], the PPG signal is influenced by a range of physiological systems, such as: the heart, including heart rate, heart rhythm, and the nature of ejection [3]; the blood vessels, including vessel stiffness, diameter, and blood pressure; the microvasculature, including peripheral compliance and resistance [3]; the autonomic nervous system which influences heart rate variability [4]; and the respiratory system, which impacts the pulse wave through changes in intrathoracic pressure [5]. Thus, there is potential to extract much physiological information from the PPG signal.

Studying the morphological characteristics of the PPG may provide information on cardiovascular health. **PhysioZoo PPG** provides a framework and tools for extracting morphological biomarkers from the PPG signal.

Performing PPG morphological analysis

Start by entering the PPG interface by clicking on the 'Pulse' menu on the top left, then load some PPG example by clicking File -> Open data file -> ppg_example.txt. The program will automaticly present the PPG file you imported.

To perform the analysis, please follow the instructions:

1. Prefiltering the signal: On the left panel, select the "Configuration" tab. On the bottom of the tab, you will find a section labeled: **Fiducials filtering parameters**.
2. Definition of the window for anlysis: On the right panel, define the W.S. (start of the window) and the W.L. (length of the window) you want to analyze. You can analyze all of your signal or part of it. Note that if you analyze a long window,it may take some time.
3. Click the **Find Fiducials** button. The fiducial poits will be detected and highlighted while the biomarkers will be automatically engineered and displayed on the lower pannels.

Congrats! You have made your first morphological analysis with **PhysioZoo PPG**! The biomarkers are divided into two different categories: Duration and Amplitudes, the statistical measurments of the biomarkers will be presented in a table, in the bottom panel.

Note: For PPG anlysis 9 statistical mesurment computed over the selected window (defined by W.S. W.L.) will be presented for each biomarker namely: signal duration; average (AVG); median (MED); standard deviation (STD); lower and upper quartiles (Q1, Q3); inter-quartile range (IQR); Skewness (SKW, indicating a lack of symmetry in the distribution; Kurtosis (KUR, indicating the pointedness of a peak in the distribution curve); and the average difference between the mean and each data value (MAD)

Exporting fiducial points

You can export the fiducial points. Go to File -> Save fiducial points. The excel file contains the computed fiducial points for each lead.

Exporting morphological biomarkers

You can export the morphological biomarkers. Go to File -> Save fiducial biomarkers. The excel file contains the engineered PPG biomarkers.

References:

[1] Charlton, Peter H., et al. "Modeling arterial pulse waves in healthy aging: a database for in silico evaluation of hemodynamics and pulse wave indexes." American Journal of Physiology-Heart and Circulatory Physiology 317.5 (2019): H1062-H1085.

[2] Alastruey, Jordi, et al. "Arterial pulse wave modeling and analysis for vascular-age studies: a review from VascAgeNet." American Journal of Physiology-Heart and Circulatory Physiology 325.1 (2023): H1-H29.

[3] Charlton, Peter H., et al. "Wearable photoplethysmography for cardiovascular monitoring." Proceedings of the IEEE 110.3 (2022): 355-381.

[4] Gil, Eduardo, et al. "Photoplethysmography pulse rate variability as a surrogate measurement of heart rate variability during non-stationary conditions." Physiological measurement 31.9 (2010): 1271.

[5] Charlton, Peter H., et al. "Extraction of respiratory signals from the electrocardiogram and photoplethysmogram: technical and physiological determinants." Physiological measurement 38.5 (2017): 669.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

INDEX

U