pecg Documentation

Author

Jan 10, 2023

POBM API REFERENCE:

Digital electrocardiography biomarkers to assess cardiac conduction.

Based on the paper S. Gendelman et al., “PhysioZoo ECG: Digital electrocardiography biomarkers to assess cardiac conduction,” 2021 Computing in Cardiology (CinC), 2021, pp. 1-4, doi: 10.23919/CinC53138.2021.9662857.

CHAPTER

ONE

INTRODUCTION

The electrocardiogram (ECG) is a standard tool used in medical practice for identifying cardiac pathologies. Because the necessary expertise to interpret this tracing is not readily available in all medical institutions or at all in some large areas of developing countries, there is a need to create a data-driven approach that can automatically capture the information contained in this physiological time series. The primary objective of this package is to identify and implement clinically important digital ECG biomarkers for the purpose of creating a reference toolbox and software for ECG morphological analysis.

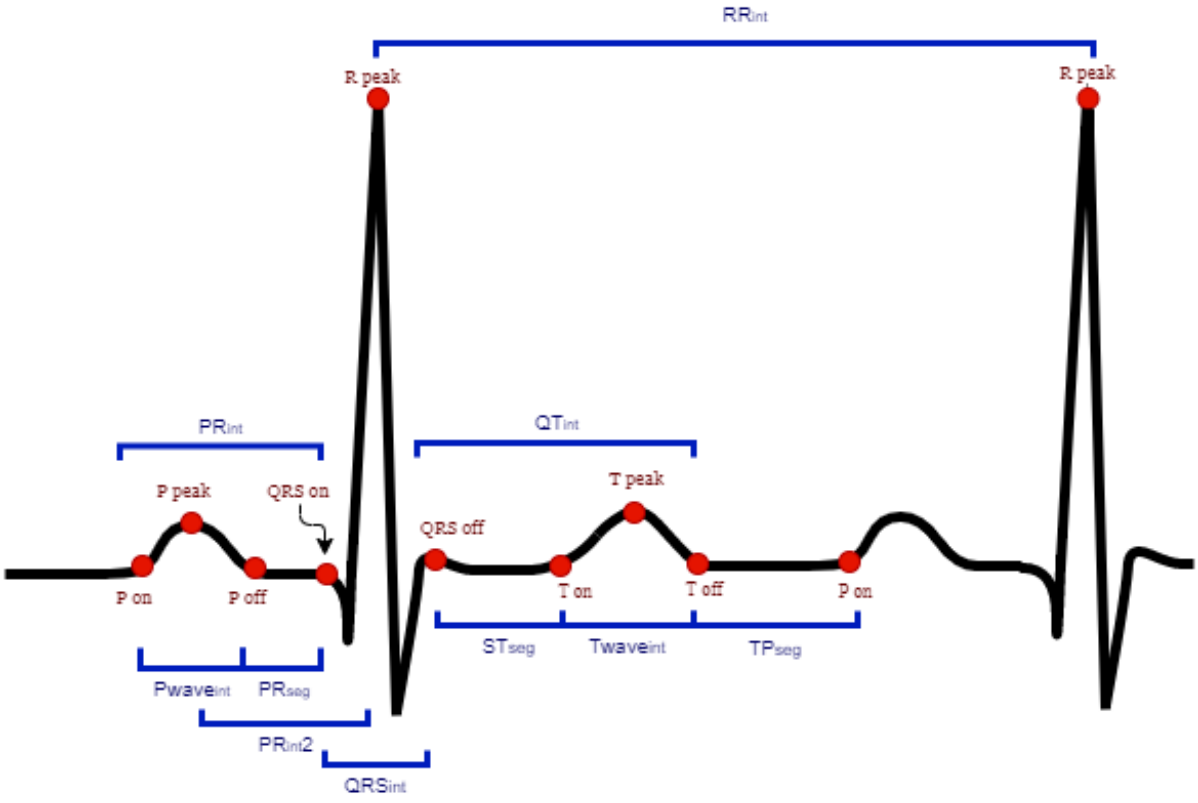
CHAPTER

TWO

DESCRIPTION

Few steps are required to extract the morphological ECG biomarkers, thos steps are impelemented in the PEBM toolbox:

1. ECG Signal Preprocessing - Before computing the ECG morphological biomarkers, prefiltering of the raw ECG time series is performed to remove the baseline wander as well as remove high frequency noise. Specifically, the toolbox include a zero phase second-order infinite impulse response bandpass filter with the passband of 0.67Hz - 100Hz to remove baseline wander and high frequency noise. Also, the toolbox include an optional Notch filter that can be set to 50 or 60Hz to remove the power-line interference.
2. ECG Fiducial Points Detection - The toolbox include the epltd R-peaks algorithem, and the the well-known wavedet algorithm for ECG fiducial points detection.
3. Engineering of ECG Biomarkers - Using the fiducial points ECG biomarkers are engineered for individual ECG cycles. When a biomarker cannot be engineered because some fiducial points could not be detected by wavedet then the feature was marked as a NaN. For an ECG channel a total of 14 features are extracted from intervals duration and 8 from waves characteristics to describe the ECG morphology.



4. Summary Statistics - For a specified time window the five summary statistics (median, min, max, Ql and Q3) are computed for all ECG biomarkers.

CHAPTER

THREE

INSTALLATION

Available on pip, with the command:

pip install pecg

CHAPTER

FOUR

REQUIREMENTS

Python >= 3.6

numpy == 1.20.2

mne == 0.23.0

scipy == 1.7.0

wfdb == 3.4.0

All the python requirements exept wfdb are installed when the toolbox is installed. To install wfbd run: pip install wfdb

CHAPTER

FIVE

SYSTEM REQUIREMENTS

For linux- to run the wavdet fiucial-points detector matlab runtime (MCR) 2021a is requierd.

For windows- to run the wavdet fiucial-points detector matlab runtime (MCR) 2020a is requierd.

If you wish to use the epltd peak detector additional wfdb toolbox is requierd.

If you don't want or can't install this - It's Ok! you can use another peak detectors from the package.

CHAPTER

SIX

DOCUMENTATION

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

6.1 pecg package

6.1.1 pecg.ecg package

6.1.1.1 pecg.ecg.FiducialPoints

class pecg.ecg.FiducialPoints.**FiducialPoints**(*signal: array*, *fs: int*)

Bases: object

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

Parameters

- **signal** – the ECG signal as a ndarray, with shape (L, N) when L is the number of channels or leads and N is the number of samples.
- **fs** – The sampling frequency of the signal.[Hz]

```
from pecg.ecg import FiducialPoints as Fp
fp = Fp.FiducialPoints(f_ecg_rec, fs)
```

wavedet(*matlab_pat: str*, *peaks: array = array([], dtype=float64)*)

The wavedat function uses the matlab algorithm wavedet, compiled for python. The algorithm is described in the following paper:[1]. The function is calculating the fiducial points of the ECG recording using wavelet transform.

Parameters

- **matlab_pat** – path to matlab runtime 2021a directory
- **peaks** – Optional input- Annotation of the reference peak detector (Indices of the peaks), as an ndarray of shape (L,N), when L is the number of channels or leads and N is the number of peaks. If peaks are not given, the peaks are calculated with the jqrs detector.

Returns

fiducials: Nested dictionary of leads - For every lead there is a dictionary that includes indexes for for each one of nine fiducials points.

[1] Martinze at el (2004), A wavelet-based ECG delineator: evaluation on standard databases. IEEE Transactions on Biomedical Engineering, 51(4), 570-581.

```
matlab_pat = '/usr/local/MATLAB/R2021a'
peaks = fp.jqrs()
fiducials = fp.wavedet(matlab_pat, peaks)
```

epltd()

This function calculates the indexes of the R-peaks with epltd peak detector algorithm. This algorithm were introduced by[2].

Returns

indexes of the R-peaks in the ECG signal, as an ndarray of shape (L,N), when L is the number of channels or leads and N is the number of peaks.

```
peaks = fp.epltd()
```

xqrs()

This function wraps the XQRS function of the WFDB package.

Returns

indexes of the R-peaks in the ECG signal, as an ndarray of shape (L,N), when L is the number of channels or leads and N is the number of peaks.

```
peaks = fp.xqrs()
```

jqrs(*thr: float = 0.8*, *rp: float = 0.25*)

The function is an Implementation of an energy based qrs detector[3]. The algorithm is an adaptation of the popular Pan & Tompkins algorithm[Page 16, 2]. The function assumes the input ecg is already pre-filtered i.e. bandpass filtered and that the power-line interference was removed. Of note, NaN should be represented by the value -32768 in the ecg (WFDB standard).

Parameters

- **thr** – threshold, default value is 0.8.
- **rp** – refractory period (sec), default value is 0.25.

Returns

indexes of the R-peaks in the ECG signal, as an ndarray of shape (L,N), when L is the number of channels or leads and N is the number of peaks.

```
peaks = fp.jqrs()
```

6.1.1.2 pecg.ecg.Biomarkers

class pecg.ecg.Biomarkers.**Biomarkers**(*signal: array*, *fs: int*, *fiducials: dict*)

Bases: object

The purpose of the Biomarkers class is to calculate the biomarkers, we divided the morphological biomarkers into two main groups: intervals and waves. :param signal: The ECG signal as a ndarray. :param fs: The sampling frequency of the signal [Hz]. :param fiducials: Nested dictionary of leads - For every lead there is a dictionary that includes indexes for for each one of nine fiducials points. this nested dictionary can be calculated using the FiducialPoints module.

[2] Pan, Jiapu, and Willis J. Tompkins. "A real-time QRS detection algorithm." IEEE Trans. Biomed. Eng 32.3 (1985): 230-236.

[3] Behar, Joachim, Alistair Johnson, Gari D. Clifford, and Julien Oster. "A comparison of single channel fetal ECG extraction methods." Annals of biomedical engineering 42, no. 6 (2014): 1340-1353.

```
from pecg.ecg import Biomarkers as Obm
obm = Obm.Biomarkers(f_ecg_rec, fs, fiducials)
ints, stat_i = obm.intervals()
waves, stat_w = obm.waves()
```

intervals()

Returns

- intervals_b: Dictionary that includes all the row data, for the **Interval duration and segments** biomarkers.
- intervals_statistics: Dictionary that includes the mean, median, min, max, iqr and std, for every **Interval duration and segments** biomarker.

Table 1: **Interval duration and segments**:

Biomarker	Description
P-waveint	Time interval between P-on and P-off.
PRint	Time interval between the P-on to the QRS-on.
PRseg	Time interval between the P-off to the QRS-on.
PRint2	Time interval between P-peak and R-peak as defined by Mao et al.
QRSint	Time interval between the QRS-on to the QRS-off.
QTint	Time interval between the QRS-on to the T-off.
QTcBint	Corrected QT interval (QTc) using Bazett's formula.
QTcFriint	QTc using the Fridericia formula.
QTcFraint	QTc using the Framingham formula.
QTcHint	QTc using the Hodges formula.
T-waveint	Time interval between T-on and T-off.
TPseg	Time interval between T-off and P-on.
RRint	Time interval between sequential R-peaks.
Rdep	Time interval betweem Q-on and R-peak.

waves()

Returns

- waves_b: Dictionary that includes all the row data, for every **Waves characteristic** biomarker.
- waves_statistics: Dictionary that includes the mean, median, min, max, iqr and std, for every **Waves characteristic** biomarker.

Table 2: **Waves characteristics**:

Biomarker	Description
P-wave	Amplitude difference between P-peak and P-off.
T-wave	Amplitude difference between T-peak on and T-off.
R-wave:	R-peak amplitude.
P-waveArea	P-wave interval area defined as integral from the P-on to the P-off.
T-waveArea	T-wave interval area defined as integral from the T-on to the T-off.
QRSArea	QRS interval area defined as integral from the QRS-on to the QRS-off.
STseg	Amplitude difference between QRS-off and T-on.
J-point	Amplitude in 40ms after QRS-off as defined by Hollander et al.

6.1.1.3 Module contents

6.1.2 pecg.Preprocessing

class pecg.Preprocessing.**Preprocessing**(*signal: array*, *fs: int*)

Bases: object

The Preprocessing class provides some routines for pre-filtering the ECG signal as well as estimating the signal quality. :param signal: the ECG signal as a ndarray, with shape (L, N) when L is the number of channels or leads and N i the number of samples. :param fs: The sampling frequency of the signal [Hz].

```
import pecg
from pecg import Preprocessing as Pre
pre = Pre.Preprocessing(signal, fs)
```

notch(*n_freq: int*)

The notch function applies a notch filter in order to remove the power line artefacts. :param n_freq: The expected center frequency of the power line interference. Typically, 50Hz (e.g. Europe) or 60Hz (e.g. US) :return: The filtered ECG signal, with shape (L, N) when L is the number of channels or leads and N is the number of samples.

```
filtered_ecg_rec = pre.notch()
```

bpfilt()

The bpfilt function applies a bandpass filter between [0.67, 100] Hz, this function uses a zero-phase Butterworth filter with 75 coefficients. :return: The filtered ECG signal, with shape (L, N) when L is the number of channels or leads and N is the number of samples.

```
filtered_ecg_rec = pre.bpfilt()
```

bsqi(*peaks: array = array([], dtype=float64)*, *test_peaks: array = array([], dtype=float64)*)

bSQI is an automated algorithm to detect poor-quality electrocardiograms. This function is based on the following paper:[1]. The implementation itself is based on:[2].

Parameters

- **peaks** – Optional input- Annotation of the reference peak detector (Indices of the peaks), as an ndarray of shape (L,N), when L is the number of channels or leads and N is the number of peaks. If peaks are not given, the peaks are calculated with jqrs detector.
- **test_peaks** – Optional input - Annotation of the anther reference peak detector (Indices of the peaks), as an ndarray of shape (L,N), when N is the number of peaks. If test peaks are not given, the test peaks are calculated with xqrs detector.

Returns

The 'bsqi' score, a flout between 0 and 1.

```
bsqi_score = pre.bsqi()
if bsqi_score < 0.8:
    print('It's a bad quality ECG recording!')
```

[1] Li, Qiao, Roger G. Mark, and Gari D. Clifford. "Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a Kalman filter." Physiological measurement 29.1 (2007): 15.

[2] Behar, J., Oster, J., Li, Q., & Clifford, G. D. (2013). ECG signal quality during arrhythmia and its application to false alarm reduction. IEEE transactions on biomedical engineering, 60(6), 1660-1666.

6.1.3 Module contents

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

INDEX