

# Locate P, Q, S and T waves in ECG

[Print to PDF](#)

## Contents

- Find the R peaks
- Locate other waves (P, Q, S, T) and their onset and offset

This example can be referenced by [citing the package](#).

This example shows how to use NeuroKit to delineate the ECG peaks in Python using NeuroKit. This means detecting and locating all components of the QRS complex, including **P-peaks** and **T-peaks**, as well their **onsets** and **offsets** from an ECG signal.

```
# Load NeuroKit and other useful packages
import neurokit2 as nk
import numpy as np
import pandas as pd
```

In this example, we will use a short segment of ECG signal with sampling rate of 3000 Hz.

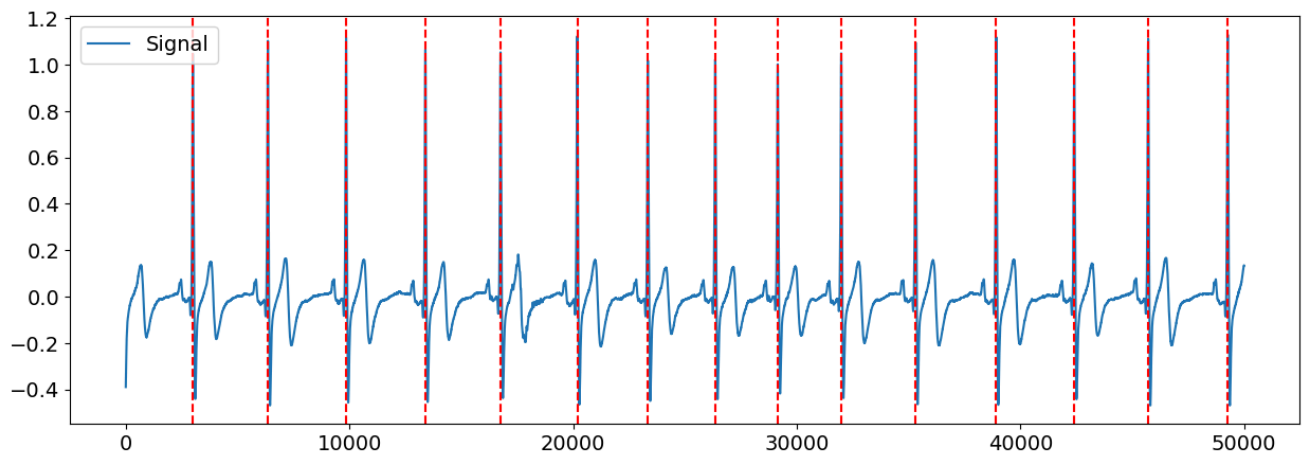
## Find the R peaks

```
# Retrieve ECG data from data folder
ecg_signal = nk.data(dataset=f"ecg_{SAMPLING_RATE}hz")
# Extract R-peaks locations
_, rpeaks = nk.ecg_peaks(ecg_signal, sampling_rate=SAMPLING_RATE)
```

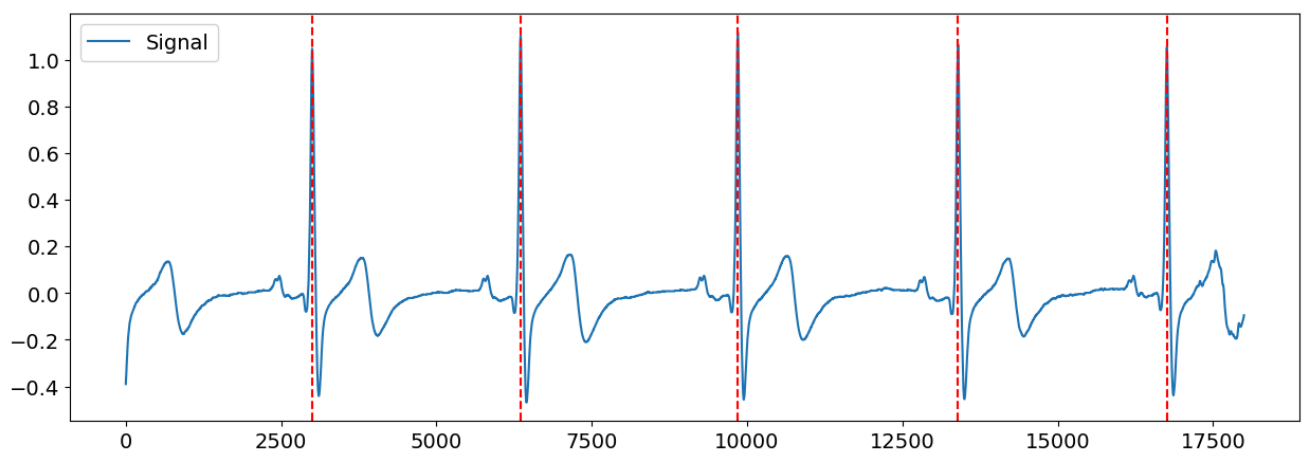
The [ecg\\_peaks\(\)](#) function will return a dictionary contains the samples at which R-peaks are located.

Let's visualize the R-peaks location in the signal to make sure it got detected correctly.

```
# Visualize R-peaks in ECG signal
plot = nk.events_plot(rpeaks['ECG_R_Peaks'], ecg_signal)
```



```
# Zooming into the first 5 R-peaks
plot = nk.events_plot(rpeaks['ECG_R_Peaks'][:5], ecg_signal[:6 * SAMPLING_RATE])
```



Visually, the R-peaks seem to have been correctly identified. You can also explore searching for R-peaks using different methods provided by NeuroKit [ecg\\_peaks\(\)](#).

## Locate other waves (P, Q, S, T) and their onset and offset

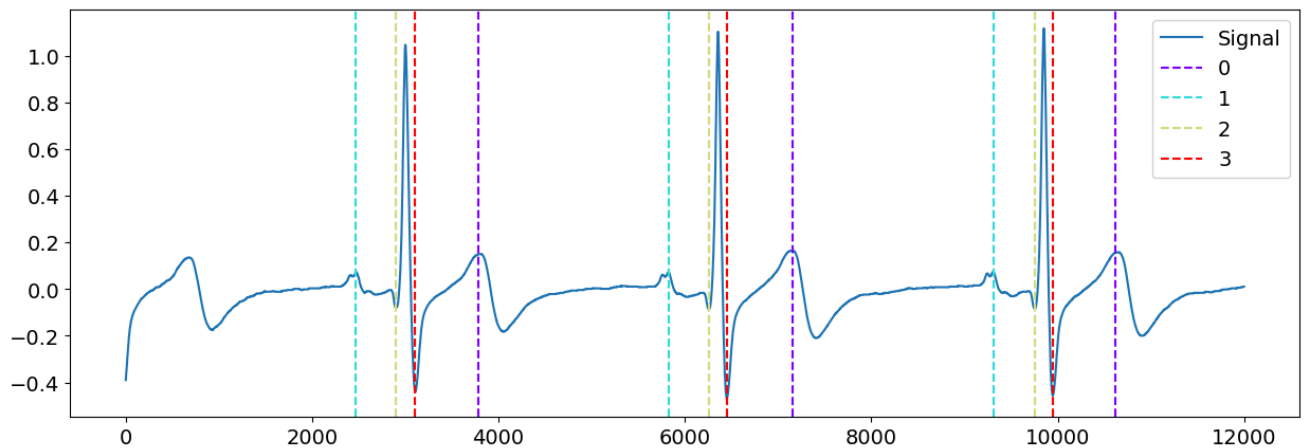
In [ecg\\_delineate\(\)](#), NeuroKit implements different methods to segment the QRS complexes. There are the derivative method and the other methods that make use of Wavelet to delineate the complexes.

### Peak method

First, let's take a look at the 'peak' method and its output.

```
# Delineate the ECG signal
_, waves_peak = nk.ecg_delineate(ecg_signal, rpeaks, sampling_rate=SAMPLING_RATE, m
```

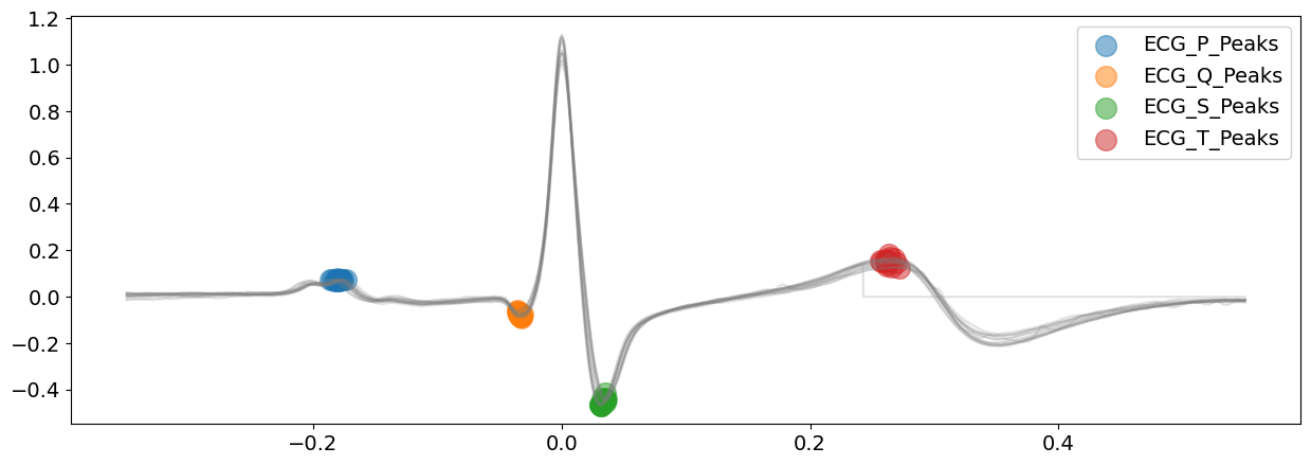
```
# Zooming into the first 3 R-peaks, with focus on T-peaks, P-peaks, Q-peaks and S-pe
plot = nk.events_plot([waves_peak['ECG_T_Peaks'][:3],
                        waves_peak['ECG_P_Peaks'][:3],
                        waves_peak['ECG_Q_Peaks'][:3],
                        waves_peak['ECG_S_Peaks'][:3]], ecg_signal[:4 * SAMPLING_RATE])
```



Visually, the 'peak' method seems to have correctly identified the P-peaks, Q-peaks, S-peaks and T-peaks for this signal, at least, for the first few complexes. Well done, *peak*!

However, it can be quite tiring to be zooming in to each complex and inspect them one by one. To have a better overview of all complexes at once, you can make use of the `show` argument in `ecg_delineate()` as below.

```
# Delineate the ECG signal and visualizing all peaks of ECG complexes
_, waves_peak = nk.ecg_delineate(ecg_signal,
                                  rpeaks,
                                  sampling_rate=SAMPLING_RATE,
                                  method="peak",
                                  show=True,
                                  show_type='peaks')
```

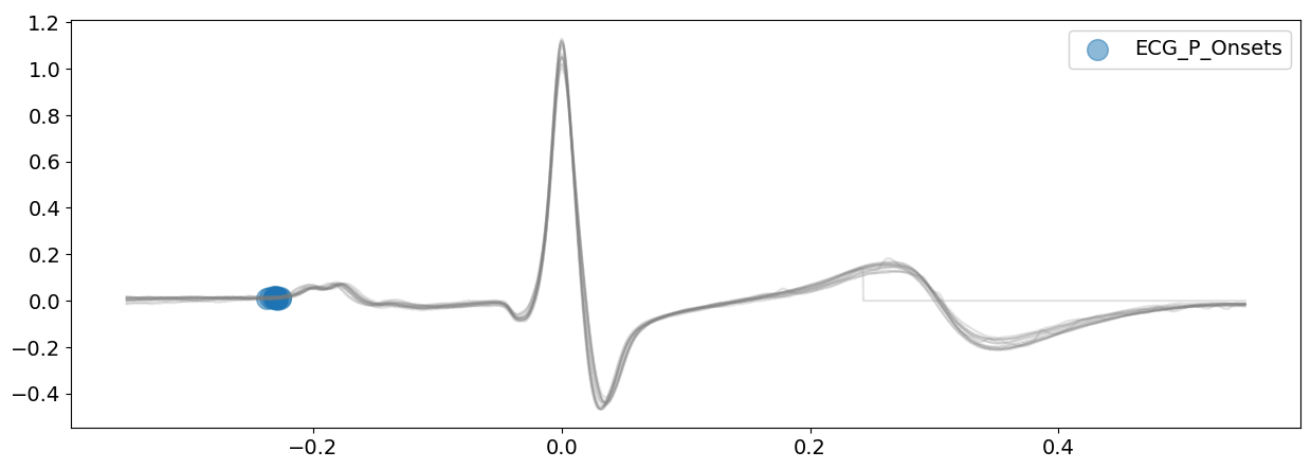


The 'peak' method is doing a glamorous job with identifying all the ECG peaks for this piece of ECG signal.

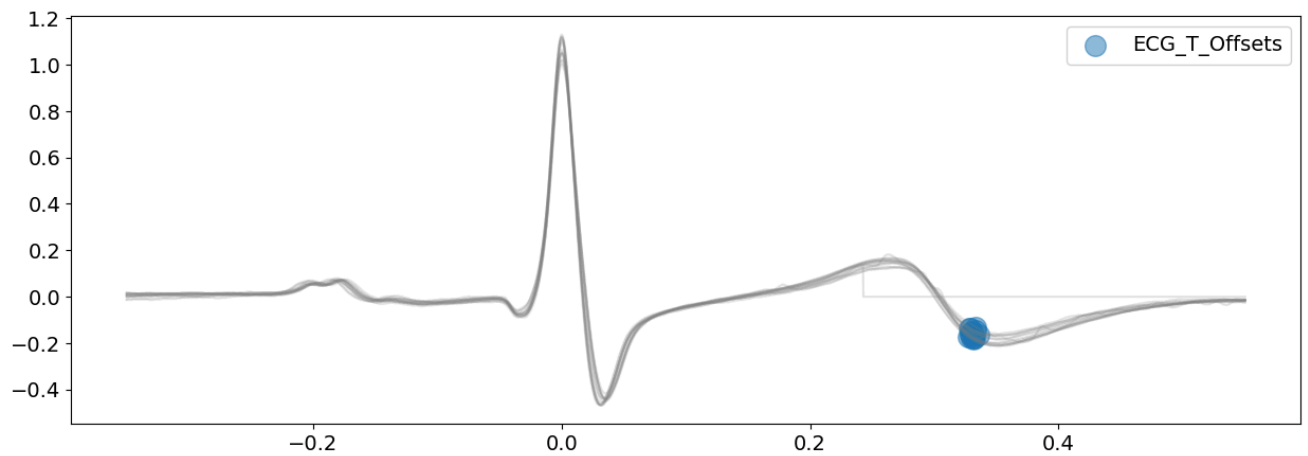
On top of the above peaks, the peak method also identify the wave boundaries, namely the onset of P-peaks and offset of T-peaks. You can vary the argument `show_type` to specify the information you would like plot.

Let's visualize them below:

```
# Delineate the ECG signal and visualizing all P-peaks boundaries
signal_peak, waves_peak = nk.ecg_delineate(ecg_signal,
                                           rpeaks,
                                           sampling_rate=SAMPLING_RATE,
                                           method="peak",
                                           show=True,
                                           show_type='bounds_P')
```



```
# Delineate the ECG signal and visualizing all T-peaks boundaries
signal_peaj, waves_peak = nk.ecg_delineate(ecg_signal,
                                           rpeaks,
                                           sampling_rate=SAMPLING_RATE,
                                           method="peak",
                                           show=True,
                                           show_type='bounds_T')
```

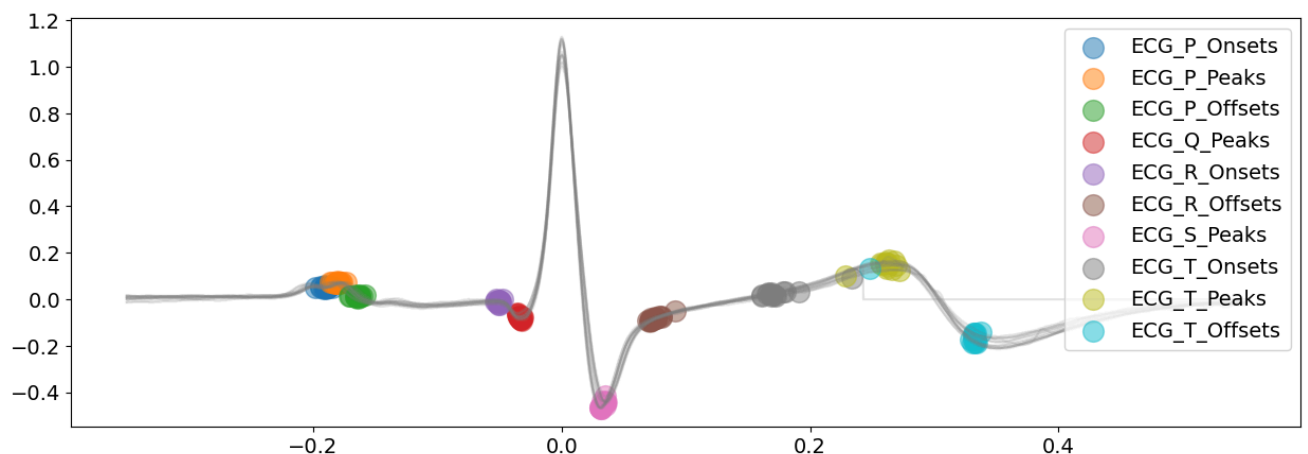


Both the onsets of P-peaks and the offsets of T-peaks appears to have been correctly identified here. This information will be used to delineate cardiac phases in [ecg\\_phase\(\)](#).

Let's next take a look at the continuous wavelet method.

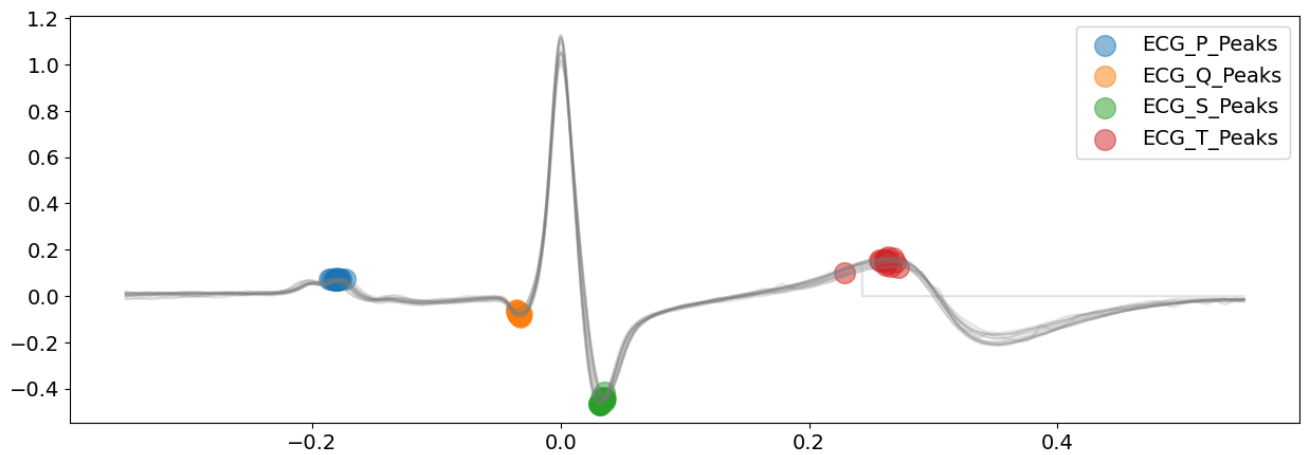
## Continuous Wavelet Method (CWT)

```
# Delineate the ECG signal
signal_cwt, waves_cwt = nk.ecg_delineate(ecg_signal,
                                         rpeaks,
                                         sampling_rate=SAMPLING_RATE,
                                         method="cwt",
                                         show=True,
                                         show_type='all')
```

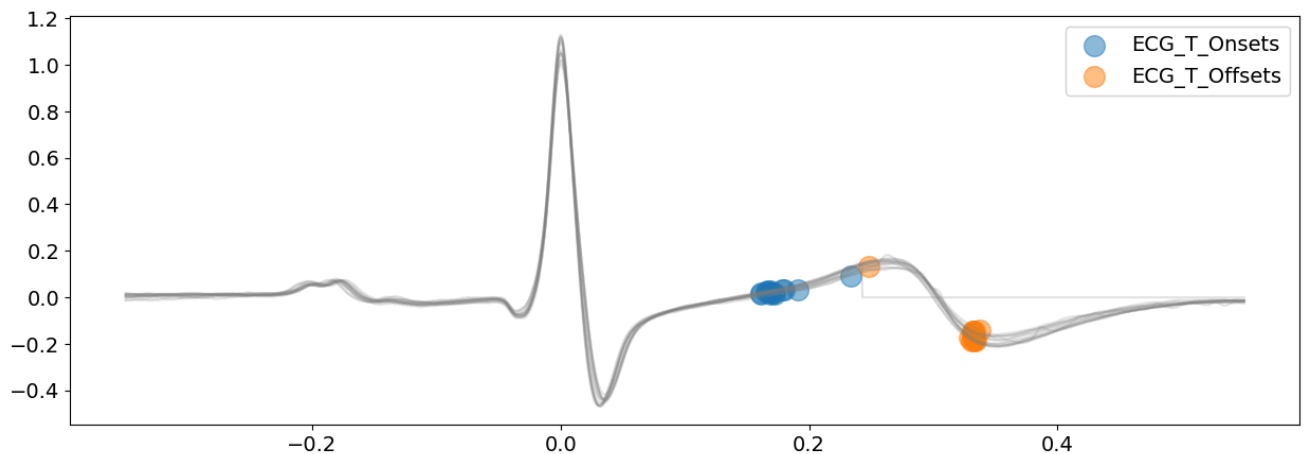


By specifying 'all' in the `show_type` argument, you can plot all delineated information output by the cwt method. However, it could be hard to evaluate the accuracy of the delineated information with everything plotted together. Let's tease them apart!

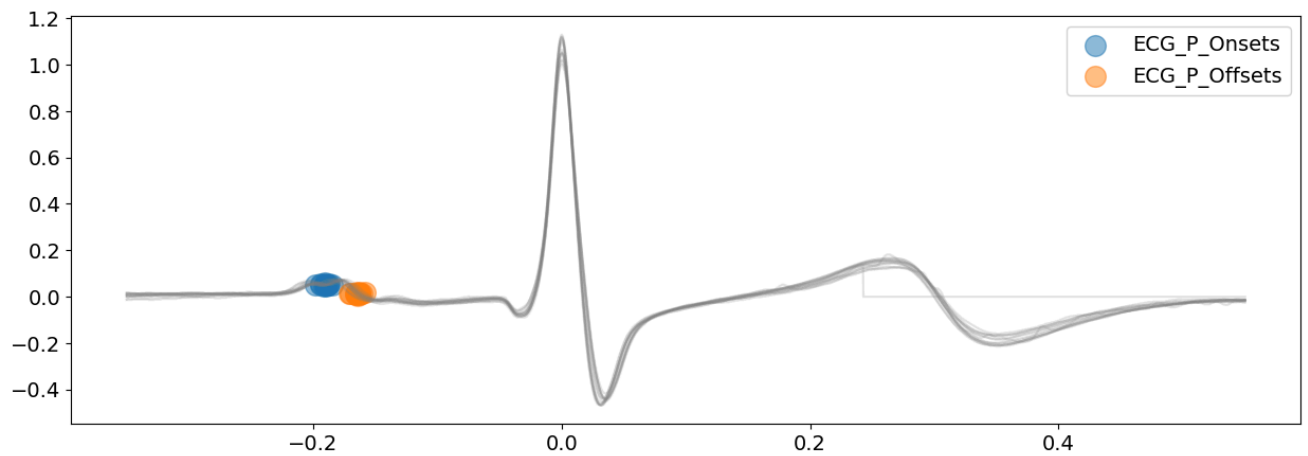
```
# Visualize P-peaks and T-peaks
signal_cwt, waves_cwt = nk.ecg_delineate(ecg_signal,
                                         rpeaks,
                                         sampling_rate=SAMPLING_RATE,
                                         method="cwt",
                                         show=True,
                                         show_type='peaks')
```



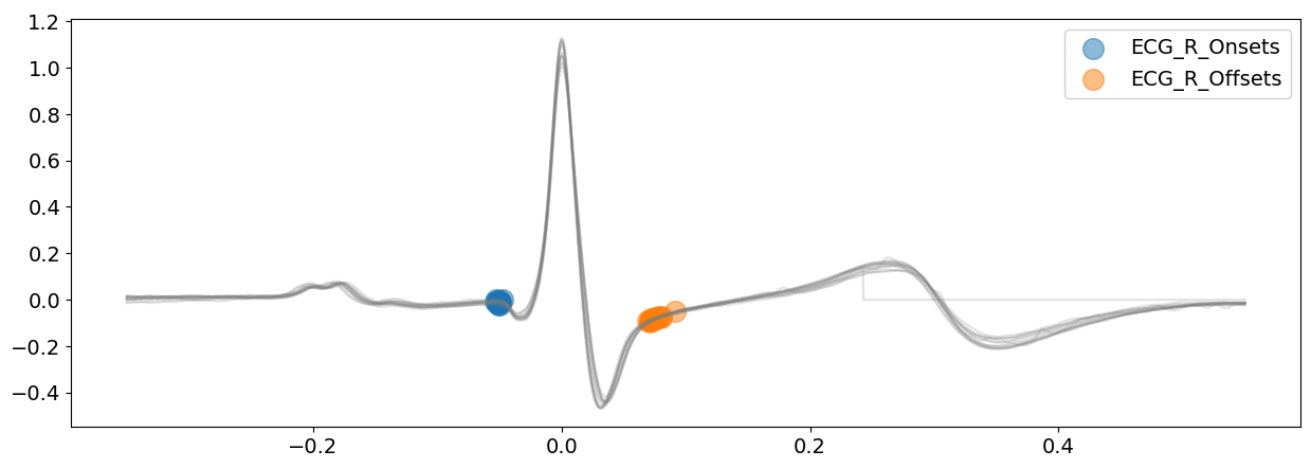
```
# Visualize T-waves boundaries
signal_cwt, waves_cwt = nk.ecg_delineate(ecg_signal,
                                         rpeaks,
                                         sampling_rate=SAMPLING_RATE,
                                         method="cwt",
                                         show=True,
                                         show_type='bounds_T')
```



```
# Visualize P-waves boundaries
signal_cwt, waves_cwt = nk.ecg_delineate(ecg_signal,
                                         rpeaks,
                                         sampling_rate=SAMPLING_RATE,
                                         method="cwt",
                                         show=True,
                                         show_type='bounds_P')
```



```
# Visualize R-waves boundaries
signal_cwt, waves_cwt = nk.ecg_delineate(ecg_signal,
                                         rpeaks,
                                         sampling_rate=SAMPLING_RATE,
                                         method="cwt",
                                         show=True,
                                         show_type='bounds_R')
```



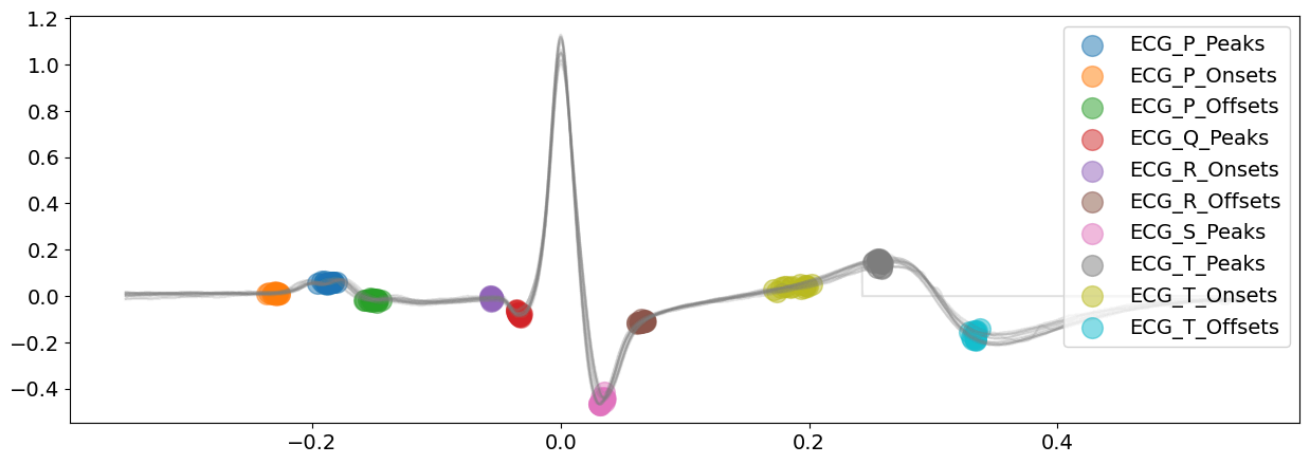
*Unlike the peak method, the continuous wavelet method does not identify the Q-peaks and S-peaks. However, it provides more information regarding the boundaries of the waves*

Visually, except a few exception, CWT method is doing a great job. However, the P-waves boundaries are not very clearly identified here.

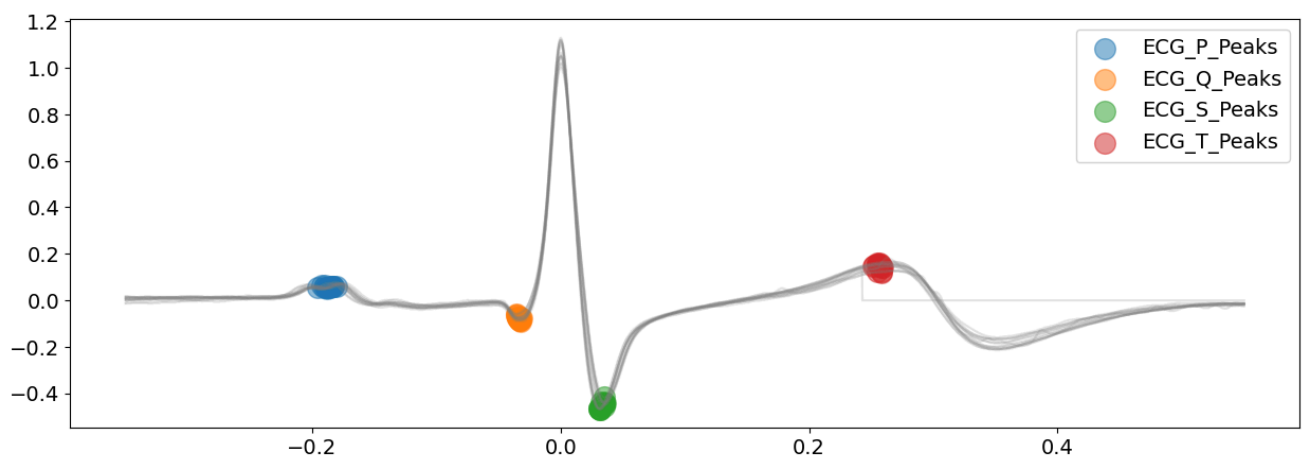
Last but not least, we will look at the third method in NeuroKit [ecg\\_delineate\(\)](#) function: the discrete wavelet method.

## Discrete Wavelet Method (DWT) - default method

```
# Delineate the ECG signal
signal_dwt, waves_dwt = nk.ecg_delineate(ecg_signal,
                                         rpeaks,
                                         sampling_rate=SAMPLING_RATE,
                                         method="dwt",
                                         show=True,
                                         show_type='all')
```

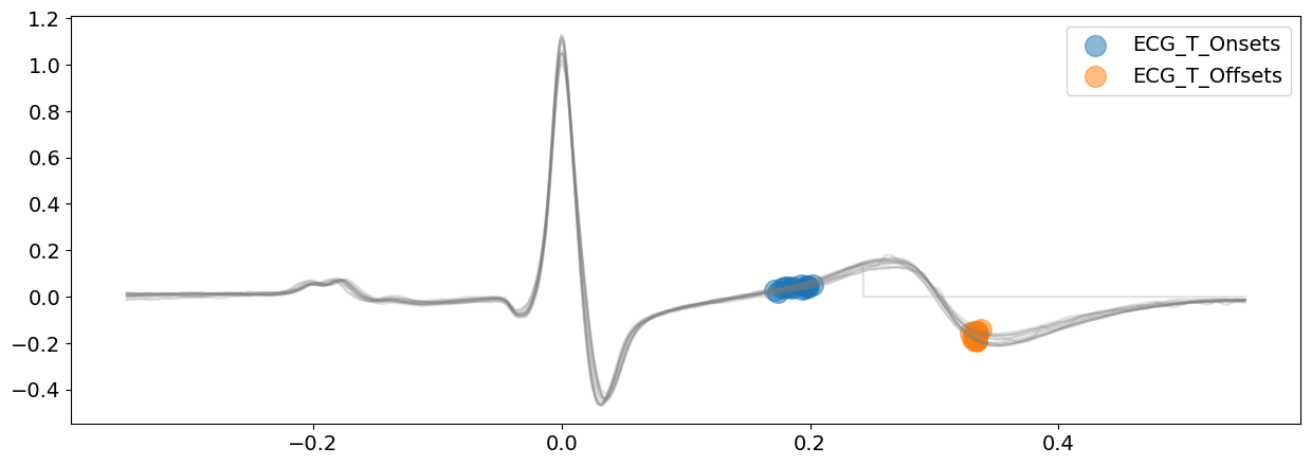


```
# Visualize P-peaks and T-peaks
signal_dwt, waves_dwt = nk.ecg_delineate(ecg_signal,
                                         rpeaks,
                                         sampling_rate=SAMPLING_RATE,
                                         method="dwt",
                                         show=True,
                                         show_type='peaks')
```

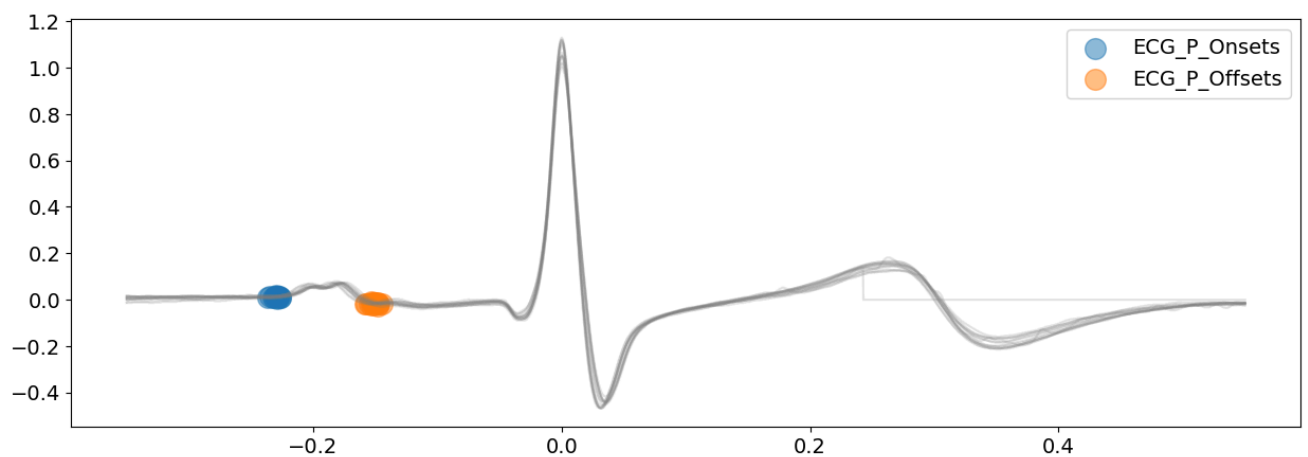


```
# visualize T-wave boundaries
signal_dwt, waves_dwt = nk.ecg_delineate(ecg_signal,
                                         rpeaks,
                                         sampling_rate=SAMPLING_RATE,
                                         method="dwt",
                                         show=True,
                                         show_type='bounds_T')
```

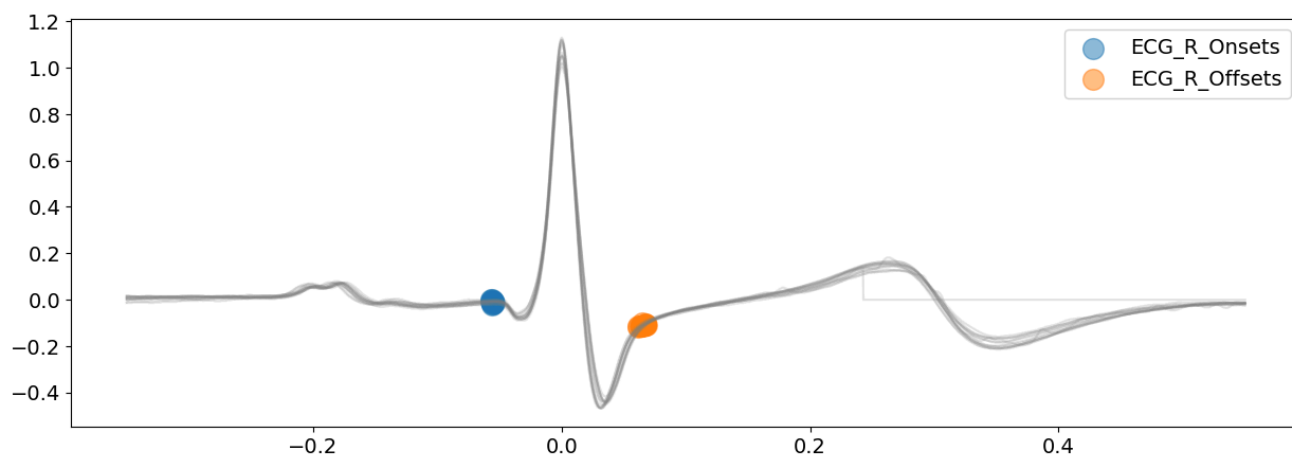




```
# Visualize P-wave boundaries
signal_dwt, waves_dwt = nk.ecg_delineate(ecg_signal,
                                         rpeaks,
                                         sampling_rate=SAMPLING_RATE,
                                         method="dwt",
                                         show=True,
                                         show_type='bounds_P')
```



```
# Visualize R-wave boundaries
signal_dwt, waves_dwt = nk.ecg_delineate(ecg_signal,
                                         rpeaks,
                                         sampling_rate=SAMPLING_RATE,
                                         method="dwt",
                                         show=True,
                                         show_type='bounds_R')
```



Visually, from the plots above, the delineated outputs of DWT appear to be more accurate than CWT, especially for the P-peaks and P-wave boundaries.

Overall, for this signal, the peak and DWT methods seem to be superior to the CWT.