

# Package: rpeaks (via r-universe)

January 25, 2025

**Type** Package

**Title** Fast detection of R peaks in ecg data

**Version** 1.5

**Date** 2020-04-29

**Author** Erik-Jan van Kesteren

**Maintainer** Erik-Jan van Kesteren <e.vankesteren1@uu.nl>

**Description** Package for fast detection of R peaks using a simplified pan-tompkins algorithm. Uses fast C++ code to speed up computations for long ecg recordings.

**License** GPL-3

**Imports** Rcpp, signal

**LinkingTo** Rcpp, RcppArmadillo

**Encoding** UTF-8

**RoxygenNote** 7.3.0

**Roxygen** list(markdown = TRUE)

**Repository** <https://vankesteren.r-universe.dev>

**RemoteUrl** <https://github.com/vankesteren/rpeaks>

**RemoteRef** HEAD

**RemoteSha** 7a4ed343b59b30320270eddac36c9ce6a8d2fe35

## Contents

rpeaks_pan_tompkins . . . . .	2
<b>Index</b>	<b>4</b>

rpeaks\_pan\_tompkins     *Detect R peaks in ecg data using the Pan-Tompkins algorithm*

---

### Description

Fast implementation of the Pan-Tompkins algorithm. The default parameters are taken from the original pan and tompkins paper.

### Usage

```
rpeaks_pan_tompkins(  
    ecg,  
    sample_rate,  
    integration_window = 0.15,  
    refractory = 0.2,  
    band_low = 5,  
    band_high = 15  
)
```

### Arguments

ecg	raw ecg data vector
sample_rate	sampling rate in Hz of the ecg
integration_window	size of the integration window in seconds
refractory	refractory period in seconds (minimum time between peaks)
band_low	lower bound of the band-pass filter in Hz
band_high	upper bound of the band-pass filter in Hz

### Details

This algorithm uses a butterworth filter of order 1 for the band-pass step, and a 3rd-order length-5 Savitzky-Golay smoothing filter to compute the derivative of the band-passed signal. Peak detection on the preprocessed signal works in a simplified way: we take the first value above the lower bound ( $3 * \text{the mean signal value}$ ) which is higher than its neighbours, and not within the refractory period after the previous R peak.

### References

Pan, J., & Tompkins, W. J. (1985). A real-time QRS detection algorithm. *IEEE transactions on biomedical engineering*, (3), 230-236.

**Examples**

```
ecg_url <- "https://physionet.org/files/ecgiddb/1.0.0/Person_01/rec_2.dat?download"
ecg_dat <- readBin(ecg_url, integer(), 500*30)
ecg_sec <- (0:(length(ecg_dat) - 1)) / 500 # rel. time in seconds
r_peaks <- rpeaks_pan_tompkins(ecg = ecg_dat, sample_rate = 500)
plot(x = ecg_sec, y = ecg_dat, type = "l", xlab = "time (seconds)", ylab = "ecg")
abline(v = r_peaks, col = "blue", lty = 3)
```

# Index

rpeaks\_pan\_tompkins, [2](#)