

Ecg_Analyzer
To be discussed
To be discussed

Filter_Params
To be arranged
Filter coeffs calculation

<<Enumeration>> Filter_Type
moving_average butterworth non-adaptive adaptive

Ecg_Baseline_Module
signal_raw: %variable length table: vector? signal_filtered: %variable length table: vector? signal_baseline: %variable length table: vector? sampling_frequency: double time_vec: %variable length table: vector? filter_type: enum filter_params: class
load_signal(signal_raw, sampling_frequency): void filter_set_properties(filter_type, filter_params): void filter_noise(): void filter_baseline(): void get_signal_raw(): %variable length table: vector? get_signal_filtered(): %variable length table: vector? get_signal_baseline(): %variable length table: vector? get_time_vec(): %variable length table: vector? filter_moving_average(): private void filter_butterworth(): private void filter_non_adaptive(): private void filter_adaptive(): private void

- Data types? Easily signal iterable
- Setting filter params?
- Filter types: non-adaptive and adaptive?
- Convolution function needed in many classes
- get\_signal\_filtered: if signal is filtered, otherwise dialog box etc.

<<Enumeration>> R_Detection_Method
pan_tompkins hilbert_transform

R_Peaks_Module
signal_filtered: %variable length table: vector? sampling_frequency: double time_vec: %variable length table: vector? r_detection_method: enum r_peaks: %variable length table: vector?
find_r_peaks(): void get_r_peaks(): %variable length table: vector? pan_tompkins(): private void filter_bandpass(): private void filter_lowpass(): private void filter_highpass(): private void differentiate(): private void square(): private void integrate(): private void hilbert_transform(): private void

- GUI: filtered or raw signal?
- Functions needed for Hilbert transform?

Waves
r_peaks: %variable length table: vector? qrs_onset: %variable length table: vector? qrs_end: %variable length table: vector? t_end: %variable length table: vector? p_onset: %variable length table: vector? p_end: %variable length table: vector?
Setting points Getting points

Waves_Module
signal_filtered: %variable length table: vector? sampling_frequency: double time_vec: %variable length table: vector? waves: class
find_waves(): void get_waves(): class

- Methods needed for waves calculation?
- Getting waves?

Time_Params
time_param1: double time_paramn: double
Setting params Getting params

HRV1
r_peaks: %variable length table: vector? time_vec: %variable length table: vector? cum_time_vec: %variable length table: vector? uniform_time_vec: %variable length table: vector? frequency_vec: %variable length table: vector? periodogram: %variable length table: vector? time_params: class frequency_params: class
calc_cum_time_vec(): void resample(): void calc_freq_vec(): void calc_periodogram(): void calc_time_params(): void calc_freq_params(): void %some helpful methods get_periodogram(): %variable length table: vector? get_freq_vec(): %variable length table: vector? get_time_params(): class get_freq_params(): class

Frequency_Params
ULF: double VLF: double LF: double HF: double freq_ULF: %variable length table: vector? freq_VLF: %variable length table: vector? freq_LF: %variable length table: vector? freq_HF: %variable length table: vector?
Setting params Getting params

Histogram
bins: %variable length table: vector? values: %variable length table: vector?
Setting Getting

HRV2
r_peaks: %variable length table: vector? time_vec: %variable length table: vector? cum_time_vec: %variable length table: vector? histogram: class tinn: %variable length table: vector? triangular_index: %variable length table: vector? poincare: class
calc_cum_time_vec(): void calc_histogram(): void calc_tinn(): void calc_triangular_index(): void calc_poincare(): void calc_SD1(): private void calc_SD2(): private void get_hist(): class get_tinn(): %variable length table: vector? get_triag_index(): %variable length table: vector? get_poincare(): class

- No idea what is needed - maybe every parameter can be a class

Poincare
???????? SD1: double SD2: double
Setting Getting