



AKADEMIA GÓRNICZO-HUTNICZA

im. St. Staszica w Krakowie
WEAIB, Katedra Informatyki
Laboratorium Biocybernetyki

Przedmiot: Przetwarzanie sygnałów w systemach diagnostyki medycznej

Nazwa projektu: PSwSDM2

Wykonali:

Konrad Strojny, Konrad Zaworski, Jerzy Głowacki, Piotr Klimiec, Dariusz Kucharski, Michał Chrzanowski, Michał Mach, Paweł Tokarz, Łukasz Dziedzic

Konsultant:

mgr inż. Tomasz Pięciak

1. [Instalacja środowiska](#)

2. [Architektura aplikacji](#)

3. [Moduł I/O - ECGInput](#)

- Opis modułu
- Opis API
- GUI

4. [Algorytmy](#)

1. [ECG BASELINE](#)

- Opis modułu
- Input/Output
- GUI
- Benchmarks

2. [R Peaks](#)

- Opis modułu
- Input/Output
- GUI
- Benchmarks

3. [HRV DFA](#)

- Opis modułu
- Input/Output
- GUI
- Benchmarks

4. [WAVES](#)

- Opis modułu
- Input/Output
- GUI
- Benchmarks

5. [HRV1](#)

- Opis modułu
- Input/Output
- GUI
- Benchmarks

1. Instalacja środowiska

W celu przygotowania środowiska do uruchomienia aplikacji należy zainstalować język Julia w wersji 0.3 i python-matplotlib:

```
sudo apt-get install julia python-matplotlib
```

oraz zaimportować następujące biblioteki w Julii:

- `Pkg.add("Gtk")`
- `Pkg.add("PyCall")`
- `Pkg.add("PyPlot")`
- `Pkg.add("DSP")`
- `Pkg.add("Dierckx")`
- `Pkg.add("ApproxFun")`
- `Pkg.add("Wavelets")`

2. Architektura aplikacji

Główna część aplikacji znajduje się w pliku `run.jl`, który należy uruchomić w terminalu Julii za pomocą polecenia `include("run.jl")`.

Plik `run.jl` ładuje szablon GUI wygenerowany za pomocą środowiska Glade, moduły znajdujące się w folderze `modules` oraz pliki obsługi sygnałów poszczególnych modułów (event handlers)

Nazwa modułu	Plik modułu
Moduł I/O: ECGInput	modules/ECGInput.jl
ECG_BASELINE	modules/Baseline.jl
WAVES	modules/Waves.jl
HRV1	modules/HRV.jl
HRV_DFA	modules/HRV_DFA.jl
R_PEAKS	modules/R_peaks.jl

Nazwa modułu	Event handler dla modułu
ECG_BASELINE	modules/Baseline_sig.jl
R_PEAKS	modules/R_peaks_sig.jl
HRV1	modules/HRV_sig.jl
HRV_DFA	modules/HRV_DFA_sig.jl
WAVES	modules/Waves_sig.jl

Szablon GUI znajduje się w pliku gui.glade. Całość zaimplementowana została z wykorzystaniem biblioteki [Gtk.jl](#).

3. Moduł I/O - ECGInput

Opis: Moduł do ładowania/zapisywania sygnałów EKG z bazy PhysioBank oraz z/do plików CSV. Do poprawnej pracy wymagana jest biblioteka WFDB i wchodzące w jej skład programy rdsamp i wfdbdesc. Więcej o nich można przeczytać pod adresem <http://www.physionet.org/physiotools/wag/intro.htm>

ECGInput zwraca obiekt Signal posiadający pięć własności:

- data - wektor próbek o wartościach typu Float32
- meta - słownik metadanych zawierających częstotliwość próbkowania, gain, itp.
- anno - słownik adnotacji o formacie {nr_próbki => symbol}
- time - czas trwania próbki
- record - nazwa rekordu z bazy danych PhysioBank

API: Dostępne są metody:

- loadsignal(record, signal=0, time="e") - łąduje sygnał z rekordu bazy PhysioBank o nazwie record (np. "aami-ec13/aami3a") i numerze signal oraz o długości time w sekundach (lub "e" dla pełnej długości sygnału). Więcej o formatach atrybutów pod <http://physionet.org/physiotools/wag/intro.htm#record>
- opensignal(filename) - otwiera pliki filename_{data,meta,anno}.csv i zwraca obiekt Signal w nich przechowywany
- savesignal(filename, signal) - zapisuje signal do trzech plików filename_{data,meta,anno}.csv
- loadRpeaks(signal) - łąduje punkty R z bazy PhysioBank i zapisuje w słowniku adnotacji
- getgain(signal) - zwraca wzmocnienie sygnału w adu/mV
- getres(signal) - zwraca rozdzielczość próbkowania w bitach (max. wartość próbki to 2^res)
- getfreq(signal) - zwraca częstotliwość próbkowania sygnału w Hz

- getPonset(signal) - zwraca posortowany wektor próbek oznaczonych jako Ponset
- getPend(signal) - zwraca posortowany wektor nrów próbek oznaczonych jako Pend
- getQRSonset(signal) - zwraca posortowany wektor nrów próbek oznaczonych jako QRSonset
- getQRSend(signal) - zwraca posortowany wektor nrów próbek oznaczonych jako QRSend
- getR(signal) - zwraca posortowany wektor nrów próbek oznaczonych jako R
- getRRIntervals(signal) - zwraca wektor interwałów pomiędzy punktami R w sekundach
- getTend(signal, w) - zwraca posortowany wektor nrów próbek oznaczonych jako Tend
- setR(signal, w) - zapisuje wektor w punktów R do słownika adnotacji
- setPonset(signal, w) - zapisuje wektor w punktów Ponset do słownika adnotacji
- setPend(signal, w) - zapisuje wektor w punktów Pend do słownika adnotacji
- setQRSonset(signal, w) - zapisuje wektor punktów QRSonset do słownika adnotacji
- setQRSend(signal, w) - zapisuje wektor punktów QRSend do słownika adnotacji
- setTend(signal, w) - zapisuje wektor punktów Tend do słownika adnotacji

Adnotacje powinny mieć jedną z wartości: "Ponset", "Pend", "QRSonset", "QRSend", "Tend", "R". Do słownika można łatwo dodawać elementy używając składni: dict[key] = value. Należy pamiętać, że słownik przechowuje nieposortowane klucze, aby je posortować należy wykonać: sort(collect(dict)).

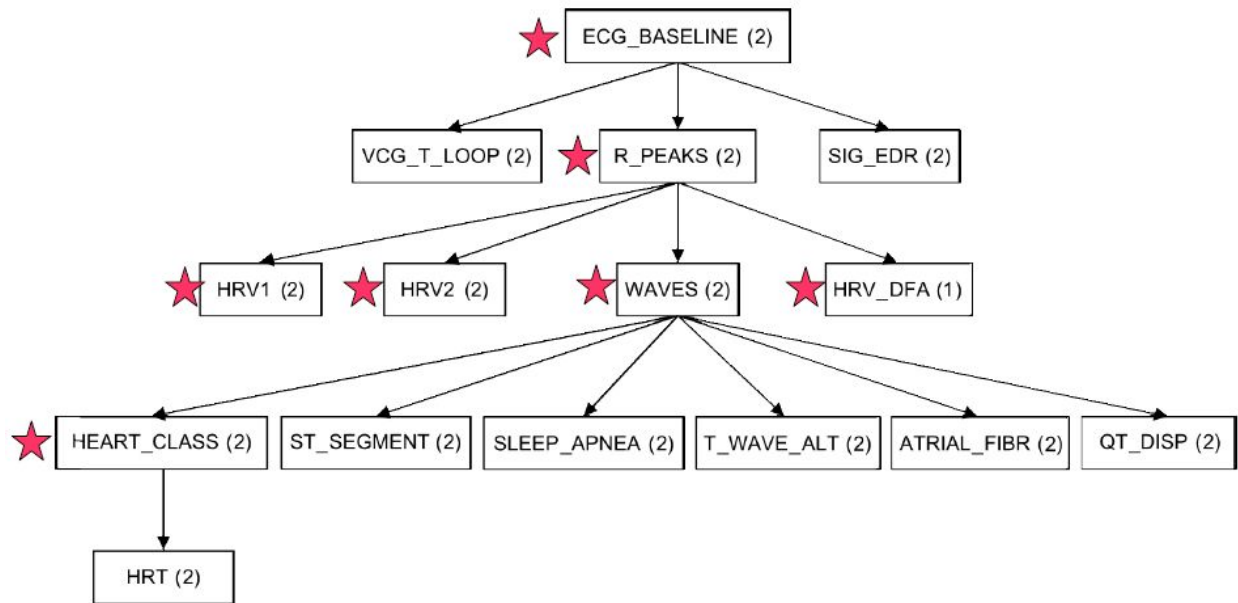
Przykładowe metadane:

Starting time: not specified
 Length: 0:59.835 (43081 sample intervals)
 Sampling frequency: 720 Hz
 File: aami3a.dat
 Description: ECG
 Gain: 130 adu/mV
 Initial value: 2072
 Storage format: 16
 I/O: can be unbuffered
 ADC resolution: 12 bits
 ADC zero: 2048
 Baseline: 2048
 Checksum: -11941

GUI:

The screenshot shows a window titled "Parametry ładowania sygnału z PhysioBank". It has two text input fields. The first field is labeled "Podaj rekord bazy danych PhysioBank:" and contains the text "mitdb/100". The second field is labeled "Podaj ilość sekund lub "e" dla pełnego rekordu:" and contains the number "5". Below these fields is a button labeled "Załaduj".

4. Algorytmy



4.1 ECG_BASELINE

Opis: Moduł programu filtrujący sygnał EKG z zakłóceń sieciowych/mięśniowych oraz usuwający z sygnału falowanie linii izoelektrycznej przy wykorzystaniu różnych metod.

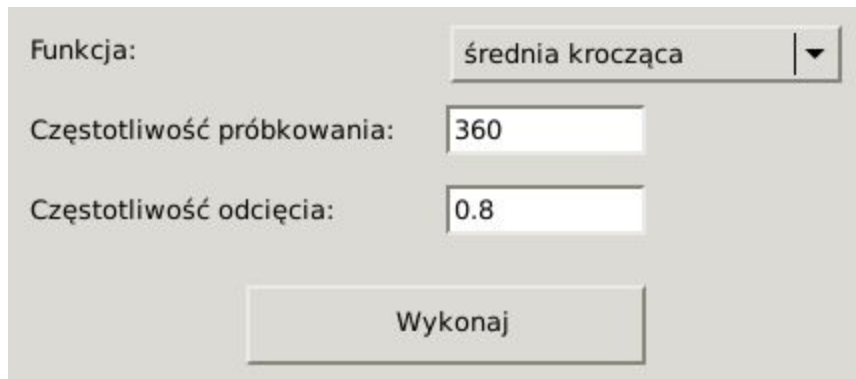
Użytkownik ma do wyboru jeden z czterech algorytmów filtrujących:

- filtr Butterwortha
- średnią kroczącą
- metody nieadaptacyjnej: filtr Savitzky-Golay'a
- metody adaptacyjnej: LMS

Input: Na wejściu modułu znajduje się tablica z próbkami, częstotliwość odcięcia (f_c , domyślnie 0.8) oraz częstotliwość próbkowania (f_s) która pobierana jest z informacji zawartych w badanym sygnale

Output: przefiltrowany sygnał, który służy jako wejście dla modułu R_PEAKS

GUI:

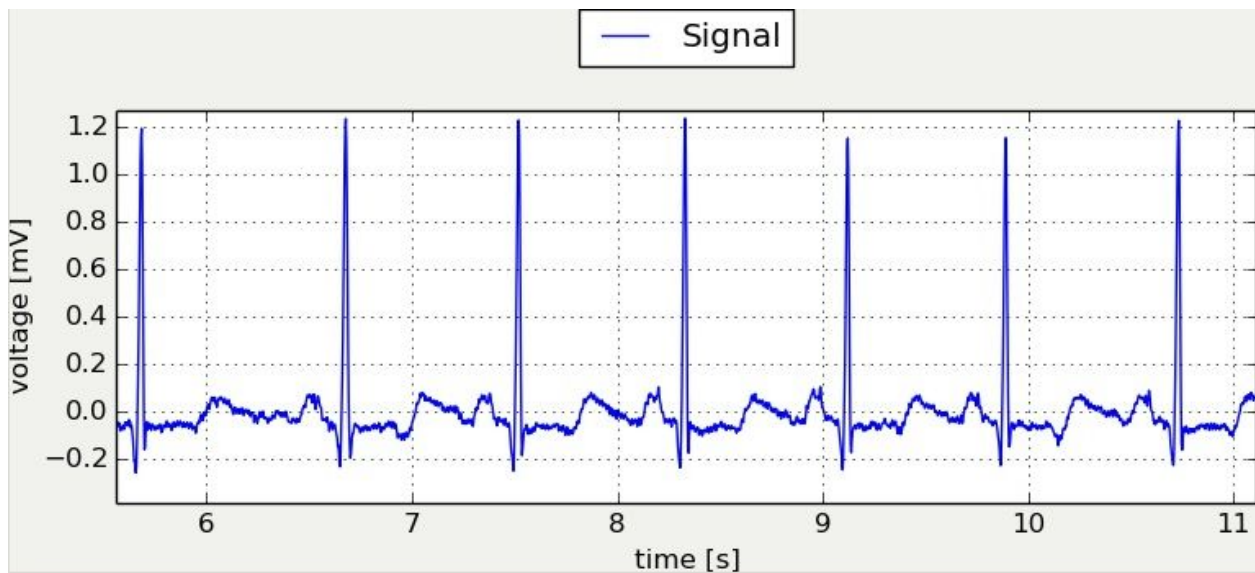


The GUI window has a light gray background. It contains three labels on the left: 'Funkcja:', 'Częstotliwość próbkowania:', and 'Częstotliwość odcięcia:'. To the right of 'Funkcja:' is a dropdown menu with 'średnia krocząca' selected and a downward arrow. To the right of 'Częstotliwość próbkowania:' is a text box containing '360'. To the right of 'Częstotliwość odcięcia:' is a text box containing '0.8'. At the bottom center is a button labeled 'Wykonaj'.

Z poziomu GUI użytkownik ma możliwość ustawienia:

- algorytmu filtrującego (Funkcja)
- częstotliwości próbkowania (domyślnie ustawiona na podstawie metadanych sygnału)
- częstotliwości odcięcia

Przycisk Wyznacz rozpoczyna wykonanie algorytmu.



Benchmarki:

Porównanie czasów wykonania algorytmów do wyznaczania linii izoelektrycznej zaimplementowanych w środowisku Matlab oraz Julia dla 10 tys. próbek.

Algorytm\Środowisko	Matlab	Julia
Moving average	0,0051 s	0,0005 s
Savitzky Golay	1,3790 s	0,5080 s
Least Mean Square	0,0019 s	0,0003 s
Butterworth	0,2738 s	0,0004 s

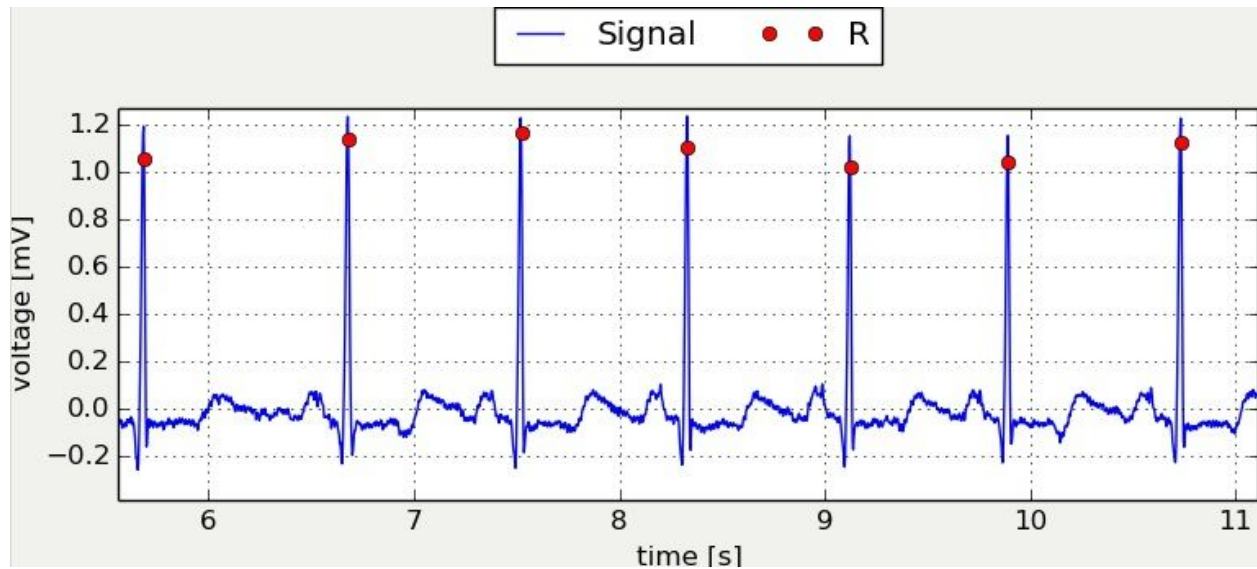
4.2 R_Peaks

Opis: moduł programu wyznaczający numery próbek załamków R oraz zaznaczający na wykresie wykryte załamki R.

Input: Na wejściu znajduje się przefiltrowany sygnał z modułu ECG_BASELINE oraz częstotliwość próbkowania sygnału. Częstotliwość próbkowania ma taką samą wartość jak w przypadku modułu ECG_BASELINE.

Output: Na wyjściu modułu zwraca tablicę z numerami próbek w których wykryto załamki R oraz wyświetla je na wykresie.

GUI:



Z poziomu GUI użytkownik ma możliwość:

- wykonania algorytmu (przycisk Wykonaj)
- wczytania informacji o załamkach R z pliku (przycisk Pobierz)

Benchmarki:

Brak

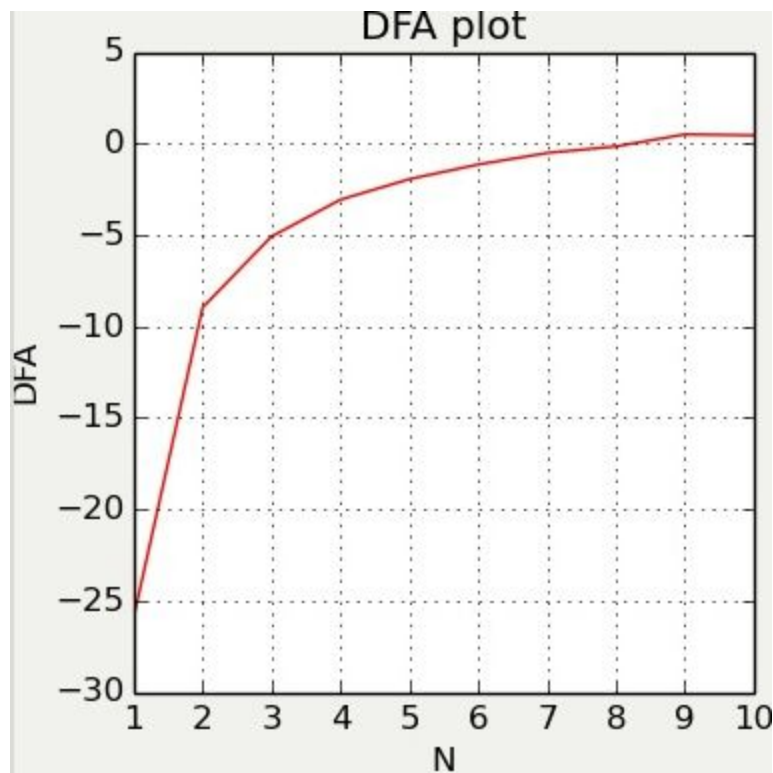
4.3. HRV_DFA

Opis: moduł programu wyznaczający, wyświetlający parametry analizy DFA w postaci tabel oraz rysujący wykresy analizy DFA (ang. Detrended Fluctuation Analysis)

Input: Na wejściu znajduje się sygnał z modułu R_Peaks.

Output: Na wyjściu modułu zwraca zwraca punkty beztrendowej fluktuacji 'a'

GUI:



Z poziomu GUI użytkownik ma możliwość:

- wykonania algorytmu (przycisk Wyznacz)

Benchmarki:

Porównanie czasowe algorytmów w Julii i Matlabie wykazało, że algorytm napisany w Julii działa znacznie wolniej niż w Matlabie. Może to być spowodowane tym, że w Julii potrzebne było zaimplementować niektóre rozwiązania, natomiast w Matlabie były gotowe funkcje.

Poniższa tabela przedstawia czasowe porównanie obu algorytmów dla testowego sygnału załamek R, sygnał został pobrany z bazy physionet.org, o długości 1000 próbek. Pomiar został wykonany pięciokrotnie, aby nie uwzględniać chwilowych obciążeń komputera, które mogły wpłynąć na czas wykonywania algorytmu.

l.p.	Julia	Matlab
1.	4,323638222s	1,070783s
2.	4,207298428s	1,030516s
3.	4,261801937s	1,01636
4.	4,266220516s	1,013983s
5.	4,976416275s	1,018457s
Średnia:	4,407075076s	1,0300198

Jak pokazuje tabela algorytm w Julii dla sygnału o długości 1000 próbek wykonuje się w czasie około 4,4s natomiast algorytm w Matlabie w czasie około 1,03s. Ponad czterokrotna różnica wykonania jest spowodowana najprawdopodobniej przez bardziej zoptymalizowane funkcje dostępne w Matlabie, których brak w języku Julia.

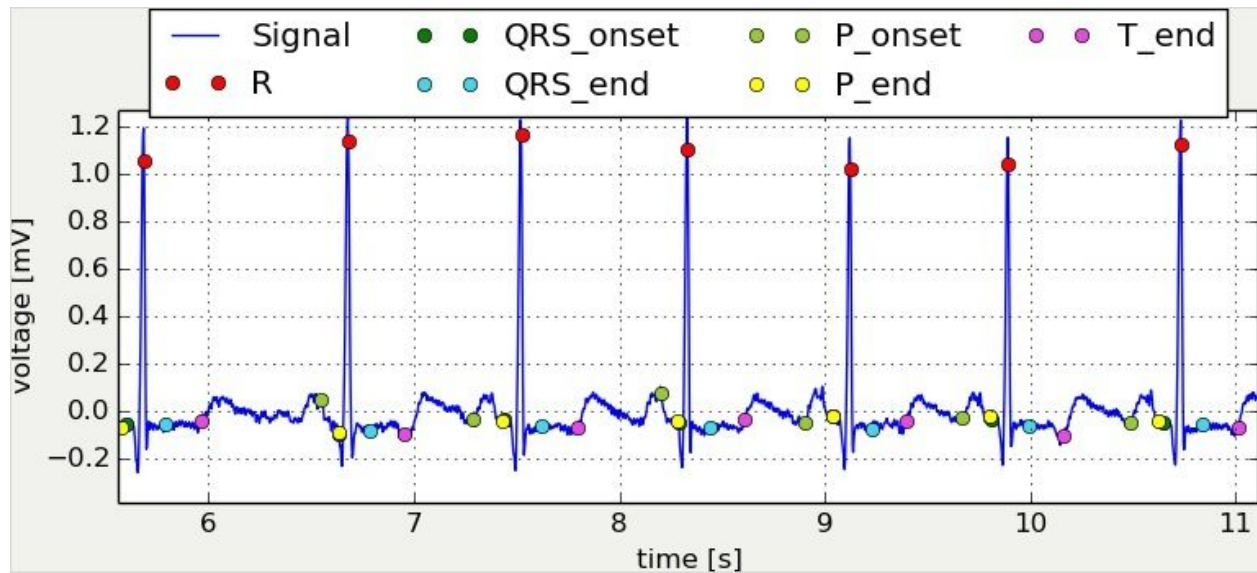
4.4. WAVES

Opis: moduł programu wyznaczający numery próbek punktów charakterystycznych: QRS-onset, QRS-end, T-end, P-onset, P-end oraz zaznaczający je na wykresie sygnału EKG.

Input: przefiltrowany ciąg próbek z modułu ECG_BASELINE oraz numery próbek załamków R z modułu R_PEAKS.

Output: Na wyjściu modułu zwraca punkty QRS-onset, QRS-end, T-end, P-onset, P-end oraz zaznaczający je na wykresie sygnału EKG

GUI:



Z poziomu GUI użytkownik ma możliwość ustawienia:

- Wielkości elementru strukturalnego
- Kroku w pochodnej

Benchmarki:

W poniższym porównaniu zestawione zostało wykonanie algorytmu detekcji w module Waves w Julii i Matlabie. Test został przeprowadzony 10 razy, wykorzystując 650000 próbek z PhysioBank, sygnał mitdb/100.

Próba	Julia	Matlab
1	3.824667554	6.980867
2	3.821696108	5.665527
3	3.851774652	5.343699
4	5.647954037	5.250748
5	3.874709417	5.315213
6	3.80537158	5.308110
7	5.397467824	5.495624

8	4.187124723	8.086126
9	3.820603037	6.855494
10	4.077118516	5.623505
Średnia	4.2308487448	5.9924913

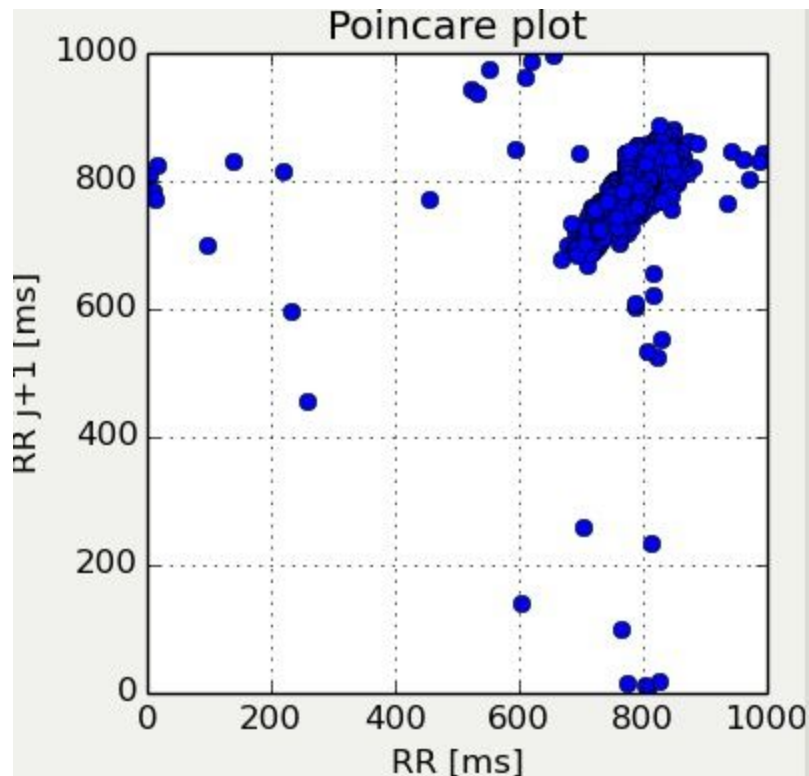
Otrzymane rezultaty pokazują, że algorytm wykonany w Julii wykonał się szybciej niż w Matlabie. Warto dodać iż w Matlabie został wykorzystany zoptymalizowany skrypt na wyszukiwanie minimum i maksimum lokalnego. W Julii algorytm został zaimplementowany “od zera” i z różnych względów może nie być najbardziej optymalny.

4.5. HRV1

Opis: moduł programu wyznaczający, wyświetlający parametry analizy czasowej iczęstotliwościowej, a także rysujący postać częstotliwościową tachogramu wraz z naniesionymi zakresami parametrów HF, LF, VLF, ULF

Input: ciąg próbek załamków R z modułu R_PEAKS

GUI:



Z poziomu GUI użytkownik ma możliwość ustawienia:

- wykonania algorytmu (przycisk Wyznacz)

Benchmarks:

Benchmarki wykonano dla nsr001.txt , chf206.txt - pliki zawierające wektor HRV wyobreniony z plików *.dat

a) dla chf206.dat size = 135471 Array{Float64}

	Julia	Matlab
freq	105,2 s	150.38,2 s
time	2.04 s	1.05 s
poincare	0.1 s	0.01 s

b) dla chf206.dat size = 30000 Array{Float64}

	Julia	Matlab
freq	7.45 s	7.45 s
time	0.53 s	0.53 s
poincare	0.02 s	0.02 s

c) dla nsr001.dat size = 106460 Array{Float64}

	Julia	Matlab
freq	130.6s	124.81s
time	0.83 s	1.08 s
poincare	0.05 s	0.01 s

d) dla nsr001.dat size = 30000 Array{Float64}

	Julia	Matlab
freq	12.08 s	13.30 s
time	0.35 s	0.26 s
poincare	0.02 s	0.002 s

Pliki testowe znajdują się w folderze: Michał Mach Pliki