Перегляд файлів екг

Для початку необхідно зайти на сайт:

<https://physionet.org/lightwave/>

Обрати

Зображення, що містить текст, електроніка, знімок екрана, програмне забезпечення

Автоматично згенерований опис

1. Щоб запустити оптимізований метод, необхідно, щоб скомпільована бібліотека на мові С була тієї ж розрядності, що і інтерпретатор Python.
2. В методі пошуку інтервальних значень, в останньому числі відкидається дробова частина (хоча воно практично на 1 більше). Для цього можна в списку останнє число замінити на обчислене значення k\_stop. Заміна виконана! (підписано Warning!)

pn\_dir*str, optional*

Option used to stream data from Physionet. The Physionet database directory from which to find the required record files. eg. For record ‘100’ in ‘<http://physionet.org/content/mitdb>’ pn\_dir=’mitdb’.

pn\_dir*str, optional*

Option used to stream data from Physionet. The Physionet database directory from which to find the required record files. eg. For record ‘100’ in ‘<http://physionet.org/content/mitdb>’ pn\_dir=’mitdb’.

id:0400 Length: nan

Протестувати і перевірити програму для випадку, коли часова послідовність складається з пари точок:

if (cut\_method == TypeOfECGCut.full and length > minutes\_points\_from\_ECG\_start):  
  
 record = open\_record(row[0], minutes\_points\_from\_ECG\_start, None)

Тест 1

Випадок 1

910401 точка

record = open\_record('0005', 910395, None)

Тобто беремо лише 5 точок

Initial length of first ECG: 6

[1.326891204607913, 1.4775357655953387]

Випадок 2

D:\Projects\ECGHiguchi\HiguchiFractalDimension\hfd.py:111: RankWarning: Polyfit may be poorly conditioned

return (-np.polyfit(np.log2(k),np.log2(L),deg=1)[0]);

Initial length of first ECG: 5

[1.8997022603180436, 2.349484991214511]

Випадок 3

Hi, Higuchi!

910401

Initial length of first ECG: 3

None

ln = find\_length\_of\_record('0005')  
print(ln)  
record = open\_record('0005', 910400, None)

D:\Projects\ECGHiguchi\venv\Scripts\python.exe D:/Projects/ECGHiguchi/main.py

Hi, Higuchi!

910401

Initial length of first ECG: 1

D:\Projects\ECGHiguchi\HiguchiFractalDimension\hfd.py:145: RuntimeWarning: divide by zero encountered in log2

k = np.logspace(start=np.log2(2),stop=np.log2(k\_stop), endpoint=True, base=2,num=num\_val,dtype=int)

D:\Projects\ECGHiguchi\venv\lib\site-packages\numpy\core\function\_base.py:158: RuntimeWarning: invalid value encountered in multiply

y \*= step

D:\Projects\ECGHiguchi\venv\lib\site-packages\numpy\core\function\_base.py:299: RuntimeWarning: invalid value encountered in cast

return \_nx.power(base, y).astype(dtype, copy=False)

Випадок 4

ln = find\_length\_of\_record('0005')  
print(ln)  
record = open\_record('0005', 910401, None)  
hfd = calculate\_higuchi(record[0],record[1])  
print(hfd)

D:\Projects\ECGHiguchi\venv\Scripts\python.exe D:/Projects/ECGHiguchi/main.py

Hi, Higuchi!

Traceback (most recent call last):

File "D:\Projects\ECGHiguchi\main.py", line 751, in <module>

hfd = calculate\_higuchi(record[0],record[1])

TypeError: 'float' object is not subscriptable

910401

Process finished with exit code 1

Тест 0

ln = find\_length\_of\_record('0005')  
print(ln)  
record = open\_record('0005', 910395, None)  
  
if (record != 'Nan'):  
 print(record)  
  
hfd = calculate\_higuchi(record[0],record[1])  
#print(hfd)

Результат

Initial length of first ECG: 6

[[0.0008047427056502928, 0.0013645637182765832, -0.0020993287973485896, 0.0008047427056502928, 0.013470693116320117, 0.2563980237828411], [0.05278415066742763, 0.045296816758746014, 0.027417311927018764, 0.01587563330844939, 0.006073750120942967, 0.09777028967062669]]

fff

[2 3]

[0.08582099859577647 0.05011175592189505]

fff

[2 3]

[0.04938320011217456 0.02712677069266467]

Тест 1

ln = find\_length\_of\_record('0005')  
print(ln)  
record = open\_record('0005', 910396, None)

ln = find\_length\_of\_record('0005')  
print(ln)  
record = open\_record('0005', 910397, None)

Результат 1

fff

[0.10168623862157232]

fff

[0.03871433181309491]

D:\Projects\ECGHiguchi\HiguchiFractalDimension\hfd.py:113: RankWarning: Polyfit may be poorly conditioned

return (-np.polyfit(np.log2(k),np.log2(L),deg=1)[0]);

D:\Projects\ECGHiguchi\HiguchiFractalDimension\hfd.py:113: RankWarning: Polyfit may be poorly conditioned

return (-np.polyfit(np.log2(k),np.log2(L),deg=1)[0]);

Лише одне значення на графіку (має бути мінімум 2)

5 и 4 точки – Polyfit may be poorly conditioned

3 и 2 точки – показывает non

1 и 0 точек

File "D:\Projects\ECGHiguchi\main.py", line 755, in <module>

hfd = calculate\_higuchi(record[0],record[1])

File "D:\Projects\ECGHiguchi\main.py", line 118, in calculate\_higuchi

HFD\_1 = HiguchiFractalDimension.hfd(np.array(ECG\_1), opt=True, num\_k=num\_k\_value, k\_max=k\_max\_value)

File "D:\Projects\ECGHiguchi\HiguchiFractalDimension\hfd.py", line 130, in hfd

k, L = curve\_length(X,\*\*kwargs)

File "D:\Projects\ECGHiguchi\HiguchiFractalDimension\hfd.py", line 53, in curve\_length

libhfd.curve\_length(k\_arr,k\_arr.size,X,N,Lk)

OSError: exception: access violation reading 0x000002770995D210

6 точек – можно использовать для випадку 2 точок polyfit

Не включалися ЕКГ зі Step. В подальшому можна зробити, щоб включалися (обрізані).

В даній роботі не включалися ЕКГ, які не мають прив’язки до вікового діапазону (рисунок 1), не мають індексу, а також ЕКГ, які мали розриви між значеннями часового ряду (рисунок 2).

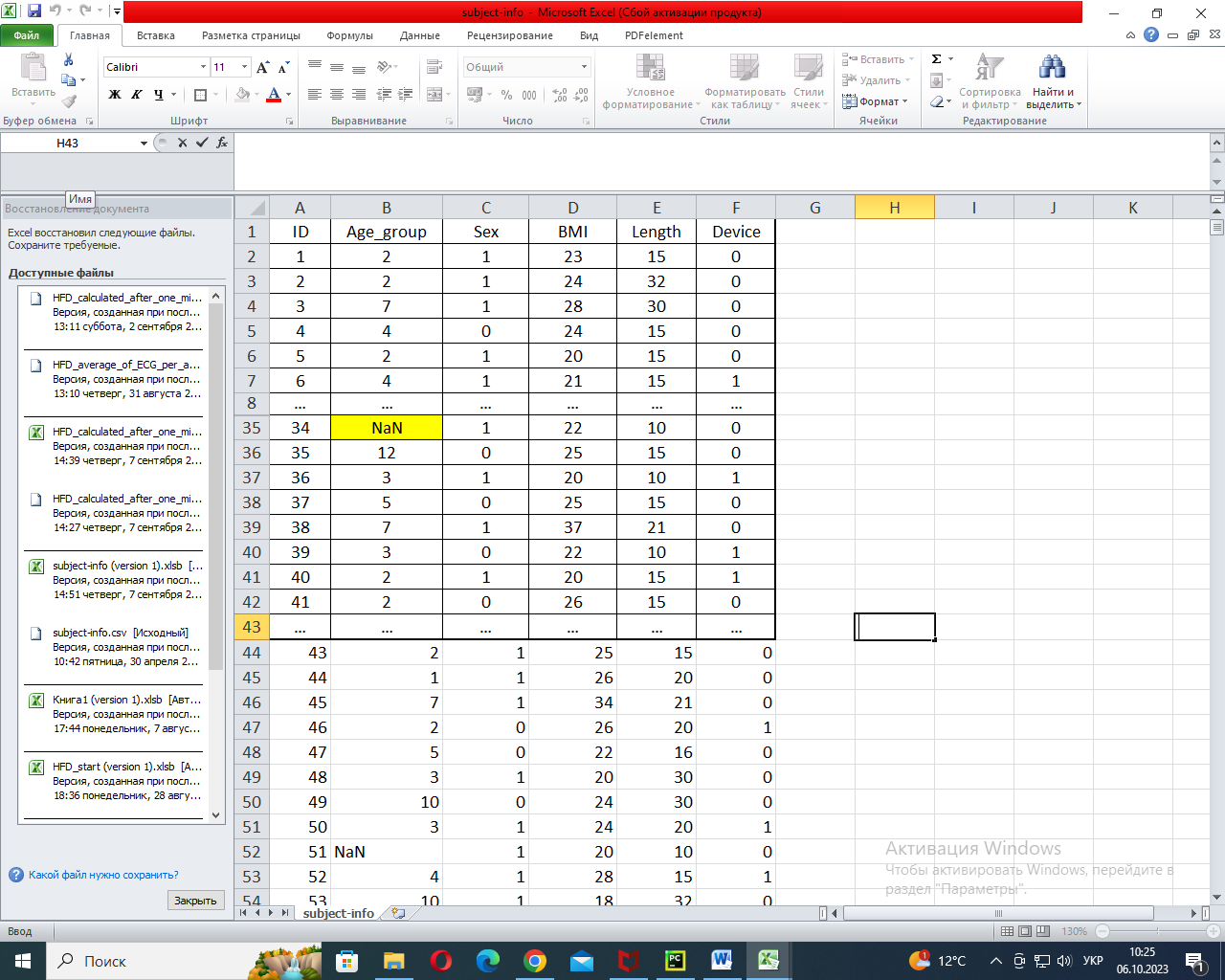


Рисунок 1 – Відсутне значення вікової групи

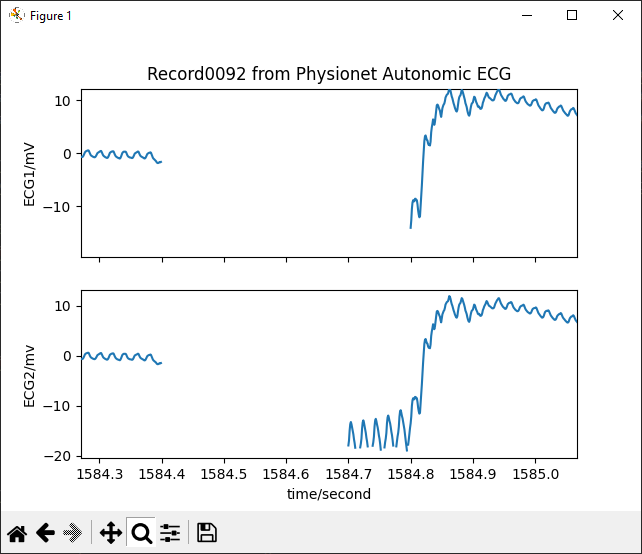


Рисунок 2 – Розриви між точками на ЕКГ

Серед тих записів, що залишилися, середня довжина 1068775.4689589303 секунди або приблизно 17,8 хвилин.

Записи, що не пройшли перевірку:

12 – step, розрив (на двох ЕКГ),

24 – step, розрив (на другій ЕКГ)

25 - step, розрив (на другій ЕКГ)

31 - step, розрив (на першій ЕКГ)

34 - немає вікової категорії

42 - step, розрив (на другій ЕКГ)

43 чи 44 – немає вікової категорії

51 – нема віку, self-organized network

52 – тиск замість другої ЕКГ

54 – степ? Нема віку

69 – артефакт

73 – степ (на другому тиску), тиск замість другої ЕКГ.

74 – шуми

75 – норм, замість другої ЕКГ тиск

76 – норм

77 - завади

92 – степ (багато завад)

98

102

110

121

143

167

172

214

216

229

232

280

293

300

311

332

335

343

346

359

389

399

400

405

421

443

448

459

478

498

500

523

569

609

612

634

656

671

685

698

718

720

751

761

773

777

779

850

869

879

891

893

902

913

917

956

963

966

987

988

1001

1011

1051

1084

1091

1119

Ідея поки що викинути всі сигнали, які мають скачки, та в яких немає віку.

Фильтрация ЭКГ сигнала является важным шагом в обработке данных для улучшения точности анализа и интерпретации. Фильтрация помогает удалить шум и артефакты, которые могут мешать правильному распознаванию значимых сигналов. Вот некоторые причины, по которым фильтрация может быть необходима:

1. **Удаление сетевого шума** (частота 50/60 Гц): Этот шум возникает из-за электропитания и может быть удалён с помощью полосового фильтра.
2. **Удаление базовой линии** (Baseline Wander): Долгосрочные колебания в сигнале, которые могут быть вызваны дыханием или движением пациента. Это можно устранить с помощью высокопроходного фильтра.
3. **Удаление высокочастотного шума**: Шумы, вызванные мышечными артефактами или другими высокочастотными источниками, можно уменьшить с помощью низкопроходного фильтра.

Пример кода на Python с использованием библиотеки SciPy для фильтрации ЭКГ-сигнала:

python

Копіювати код

import numpy as np

import scipy.signal as signal

import matplotlib.pyplot as plt

# Пример сигнала

fs = 500 # Частота дискретизации, Гц

t = np.linspace(0, 10, fs\*10) # Время

ecg\_signal = np.sin(2\*np.pi\*1\*t) + 0.5\*np.sin(2\*np.pi\*50\*t) # Пример ЭКГ с шумом

# Полосовой фильтр для удаления низкочастотного и высокочастотного шума

lowcut = 0.5

highcut = 50.0

nyquist = 0.5 \* fs

low = lowcut / nyquist

high = highcut / nyquist

b, a = signal.butter(1, [low, high], btype='band')

filtered\_ecg\_signal = signal.filtfilt(b, a, ecg\_signal)

# Визуализация до и после фильтрации

plt.figure(figsize=(10, 6))

plt.subplot(2, 1, 1)

plt.title('До фильтрации')

plt.plot(t, ecg\_signal)

plt.xlabel('Время (с)')

plt.ylabel('Амплитуда')

plt.subplot(2, 1, 2)

plt.title('После фильтрации')

plt.plot(t, filtered\_ecg\_signal)

plt.xlabel('Время (с)')

plt.ylabel('Амплитуда')

plt.tight\_layout()

plt.show()

Этот код создает синтетический ЭКГ-сигнал с шумом и применяет полосовой фильтр для его очистки.

Если у вас есть реальные данные и вам нужна помощь с их обработкой, пожалуйста, предоставьте больше информации о вашем сигнале и типе шума, чтобы можно было предложить более конкретные решения.