

Health Informatics Practical #2

Instructions:

This group assignment is expected to **require 5-8 hours to complete**. Please plan accordingly.

Due Date: 3pm EST, Oct. 17th (Thursday).

Submission Files: the signed HONOR CODE page (see HONOR CODE page), and one zipped program folder file (i.e., containing all code files)

Submission Protocol:

Upload the zipped file to your home directory on knn.bme.gatech.edu, and then send an email to 8813grading@lists.gatech.edu to inform us that you have submitted your assignment. For your email subject line, please use: **BMED8813 last name 1, last name 2, group #, PRACTICAL #2.**

Submission File Protocol:

- (1) Your submission will be one zipped folder file. The folder should contain all of the necessary files (i.e., the main program file, possible additional function files, and necessary test data files) to run your code and show the results.
- (2) The folder should be named: **< last name 1>_<last name 2>_..._<group #>**
- (3) In your folder, code files should be named as: **<last name 1>_<last name 2>_..._<group #>_<code name>.m** (or .pl)
- (4) Please write well-commented code, and include any explanations you think will help the instructor understand your program. If your program does not work completely, comments might help you get partial credit. However, you will get **NO CREDIT** if your program generates errors, generates warnings, or outputs “junk”.

If you have problems, please contact the instructors at 8813grading@lists.gatech.edu with the proper email subject line as instructed in the syllabus.

HONOR CODE

The conditions of this assignment are subject to the Georgia Institute of Technology Academic Honor Code.

I pledge that the work in this assignment, including all written code, represents the original work of BMED8813 BHI Group ____ [write group #]. I have NOT communicated with anyone (other than my group members) about the contents of this assignment, nor participated in or observed any conduct prohibited by the Honor Code.

Student Signature _____

Student Signature _____

Student Signature _____

NOTE: Please print out this page, sign the page, scan the signed page, and submit with your programming code.

If you have a problem finding a scanner to scan in the page, please type in the above HONOR CODE statement in your email message body and sign with your name.

Any assignment without the HONOR CODE PAGE will be void to ZERO grade by default.

Prerequisite Tutorials and Information:

MIT-BIH Arrhythmia Database:

The health informatics practical will require students to be familiar with the MIT-BIH Arrhythmia database. The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects. The recordings were digitized at 360 samples per second per channel. The dataset also includes independent annotations from two or more cardiologists for each patient record in a computer-readable form (approximately 110,000 annotations in all).

The MIT-BIH Arrhythmia Database can be directly accessed at the following website:
<http://www.physionet.org/physiobank/database/mitdb/>.

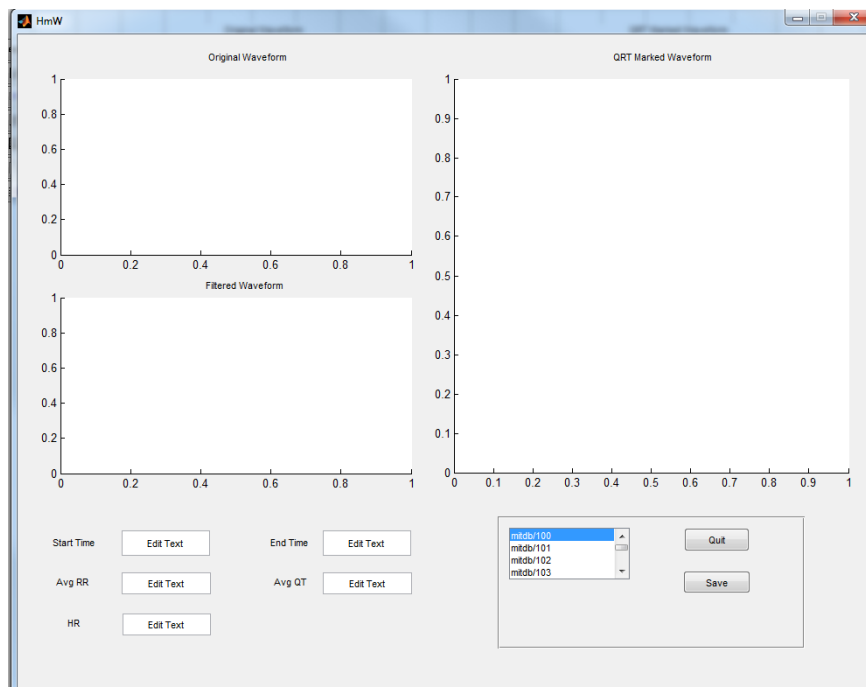
Students should use the PhysioNet MATLAB Toolbox to access the data directly from the website:

http://physionet.org/physiotools/matlab/wfdb-swig-matlab/new_version.shtml

Familiarization with different ECG waveforms:

<http://www.ijsrp.org/research-paper-1212/ijsrp-p1276.pdf>

1. In the first part of this assignment, modify the previous GUI to include additional axes for showing the Q, R, and T wave after detection. In addition, add a textbox for displaying the average QT interval as shown below.



2. In the second part of this assignment, extract the Q wave from the ECG signals mitdb/100-mitdb/109 using the following steps:

- a. In the first step, use the code given on the last page to filter the ECG using wavelets and then extract RR intervals. Describe the steps of this algorithm or draw a flow diagram that shows the steps. The purpose of this step is for students to understand the algorithm in the given code.
- b. Extract the Q wave from the ECG using the R wave positions obtained in the step above. Use the hints given below or the algorithm outlined in:

“Engelse WA, Zeelenberg C. A Single Scan Algorithm for QRS-Detection and Feature extraction. Computers in Cardiology 6:37-42 (1979)”

Hints:

- The Q wave is found between 30-150ms before the R peak position.
- The Q wave is the minimum position and is characterized by a sudden localized downward change in amplitude.
- The algorithm would involve (1) searching for the minimum point between 30 to 150 ms before the R wave (this would indicate the position of the Q peak), and (2) finding the start and end of the Q wave.
- To find the start of the Q wave, perform a local search about 55ms before the minimum point for finding the start of the Q wave. If the amplitude of the signal changes from a positive value to a negative value in consecutive samples, it denotes the Q wave beginning. If not, then the maximum value in the 55 ms range prior to the minimum point denotes the Q wave beginning. If more than 1 such value is found, then the point further from the R wave denotes the beginning.
- The end of Q wave is found using a similar search 55 ms after the minimum position. If the amplitude changes from negative to positive, then it denotes the end of Q wave. In the event no such point is found, then the end of the Q wave corresponds to the maximum value in the range. If more than one such point is found, then the point closest to the R wave is chosen.

3. In the third part of this assignment, perform the extraction of the T wave from the ECG signals mitdb/100-mitdb/109.

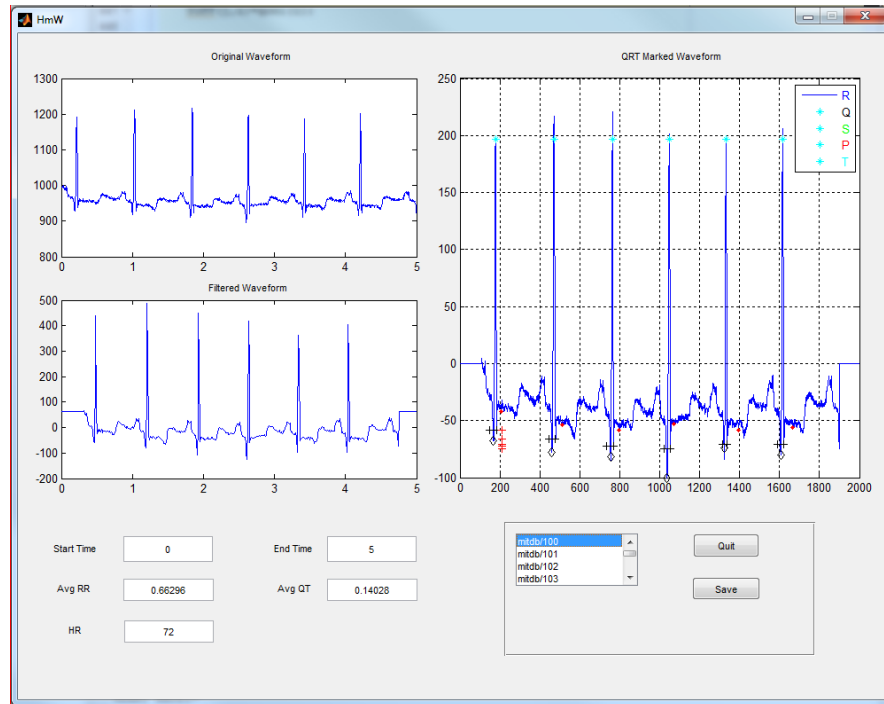
Use the same R position values extracted in the first step. Use the hints given below or implement the algorithm outlined in

“Zhang Q, Manriquez AI, Medigue C, Papelier Y, Sorine M. An Algorithm for Robust and Efficient Location of T-wave Ends in Electrocardiogram. Systemes biologiques. Publication interne n1744 | Octobre 2005”

Hint:

- The T wave is found between 70-250 ms after the R peak position.
- On lead II, the T wave is the maximum amplitude in the range and is present due to the repolarization of ventricles.
- The Algorithm would involve (1) searching for the maximum point between 70 to 300 ms after the R wave (this would indicate the position of the T peak) and (2) finding the start and end of the T wave.
- To find the start of the T wave, perform a local search about 55 ms before the minimum point. If the amplitude of the signal changes from a negative value to a positive value in consecutive samples, it denotes the T wave beginning. If not, then the maximum value in the 55 ms range prior to the minimum point denotes the T wave beginning. If more than 1 such value is found, then the point closest to the R wave denotes the beginning.
- The end of T wave is found using a similar search 55 ms after the maximum position. If the amplitude changes from positive to negative, then it denotes the end of T wave. In the event no such point is found, then the end of the T wave corresponds to the maximum value in the range. If more than one such point is found, then the point closest to the T wave is chosen.

4. Following the extraction of these points, plot the peak locations and the start and end of each peak on the GUI. Also provide the average QT interval given by the distance between the peak locations. In addition to this, plot a running waveform in the GUI.



RR Interval Code

```
%% Get Wave forms
annotator = 'atr';
Q = rdann(record,annotator);
sfreq = Q(1).sampleNumber./Q(1).timeInSeconds;
for k = str2double(t1):1:str2double(t2)
time=time+k;
RAW = rdsamp(record, 'maxt', num2str(time));
t=1:1:(size(RAW,1));
t=t/sfreq;
M=(size(RAW,1));
plot(t',RAW(:,2));
%% Setting Values
val = RAW;
v1=val(:,2)-val(1,2);
z=zeros(100,1);
A=v1;A=A';zc=A(1);
A=[z;A';z];
%% Filtering Raw Data
s = A;ls = length(s);
[c,l ]= wavedec(s,8,'db4');
ca2=appcoef(c,l,'db4',2);
base_corrected=ca2;
y=base_corrected-mean(base_corrected);
axes(handles.axes2);
K=M/length(y);
t=1:1:length(y);
t=t*K/sfreq;
plot(t(1:length(y))',y);
%% Detect R Peak
y1=y;
m1=max(y1)-max(y1)*.60;
P=find(y1>=m1);
% it will give two two points ... remove one point each
P1=P;P2=[];
last=P1(1);
P2=[P2 last];
for i=2:1:length(P1)
    if(P1(i)>(last+10))
        last=P1(i);
        P2=[P2 last];
    end
end
P3=P2*4;
Rpos=[];
for i=1:1:length(P3)
    range= [P3(i)-20:P3(i)+20];
    m=max(A(range));
    l=find(A(range)==m);
    pos=range(l);
    Rpos=[Rpos pos];
end
Ramp=A(Rpos);
R_peaks = length(Rpos)
```