General notes regarding the first section of assignment

In case any of you missed instructions for the first section of assignments here are some of the important details regarding the files that you have to submit:

Detecting heart beats (using ECG or ECG and pulsatile signals, i.e. for assignments 1.a-1.e or 1.f-1.g):

In a compressed archive (.zip) submit the following to the web classroom:

- implementation of chosen algorithm (source files and used shell scripts);
- annotation files in wfdb compatible text format generated by the chosen algorithm (having file extensions .det) (run the algorithm on at least a few records, but running the algorithm on whole databases is favored, i.e. using all of the records from either MIT-BIH DB or LTST DB);
- write a short report in a text (.txt) or document (.pdf) file with results, short explanation of the algorithm, conclusions and discussion (Abstract, Introduction, Methods, Results, and Discussion).

The detection algorithm must produce text files that are compatible with the wrann application from the WFDB Application Toolkit - this makes it possible to convert into wfdb binary format used for checking the detector performance using bxb and sumstats. Note that if your detector is implemented in C, instead of creating the latter text files (.det) you can opt to write the resulting annotations directly into wfdb binary format (using the putann function) (in this case the resulting wfdb binary files can be used directly with bxb w/o prior conversion with wrann).

The resulting .det text files must have the following columns (space separated):

Column 1: Absolute time of the annotation - fixed as 00:00:00.00;

Column 2: Fiducial point of the detected heart beat in samples (the work of your detector);

Column 3: Fixed as N (i.e. all heart beats are detected as normal);

Columns 4, 5, and 6: Fixed as 0 (no additional parameters are required).

Here is an example of the first few lines of a sample.det text file:

```
0:00:00.00 208 N 0 0 0
0:00:00.00 448 N 0 0 0
0:00:00.00 680 N 0 0 0
0:00:00.00 912 N 0 0 0
```

The .det files can then be converted into wfdb binary format (for evaluation with bxb) using the following command (the example shown below is for record s20011, the detector produced s20011.det):

```
wrann -r s20011 -a qrs <s20011.det # Convert into wfdb binary file s20011.qrs
bxb -r s20011 -a atr qrs  # Compare .qrs against the reference .atr annotation
```

(later you can calculate combined statistics using sumstats)

Additional notes

Please submit all annotation files (.det for detection, or if implemented in C, then you can submit wfdb binary files instead) produced by your implementation together with all the source files, any shell scripts you might have used, and your report in a single compressed .zip archive file. Upload the latter .zip file using the web-classroom. The files included in the archive enable us to perform an independent check whether the resulting .det files are really the product of your implementation, and also to verify the final aggregate statistics that you will have included in your .txt or .pdf report.

Here are two shell scripts which contains wfdb functions that we have already seen. You are free to use and modify them to your needs. You don't have to use these scripts, you can also write your own "helper" programs or scripts (e.g. a script that converts all available records into Matlab files using wfdb2mat, and a script for converting all produced .det files into wfdb binary format, running bxb, etc.).

run_detector.sh (for evaluation of a detector implemented in C):

```
#C program and evaluation
#####
rm eval1.txt
rm eval2.txt
FILES=/path/to/directory/with/records/*.dat

gcc -o myDetector -O myDetector.c -lm -lwfdb #compile

for f in $FILES
do
  f=$(basename $f)
  f=${f%.*}

  echo  $f
     ./myDetector -r $f #run algorithm

  #evaluate using reference annotations atr and your annotations det
  #which is already in WFDB binary format (or if your implementation
  #produced .det text files, then you first need to convert it into
  #wfdb binary annotations such using wrann - see script below)
  bxb -r $f -a atr det -l eval1.txt eval2.txt
done

sumstats eval1.txt eval2.txt > results.txt #final statistics
#Now you can copy average Se and +P from results.txt

#Columns in results.txt that are of your interest are following:
# (note that the detector does not distinguish between different
# types of heart-beats, it just needs to detect them - so it should
# detect all N, V, F, etc. heart-beats as N, hence, FP=Nn+Vn+Fn,
# for heartbeat types see:
# https://archive.physionet.org/physiobank/annotations.shtml)
# - (Nn+Vn+Fn) = true positive (N, V, or F detected as N)
# - (No+Vo+Fo) = false negative (failed to detect N, V, or F)
# -  On = false positive (heartbeat was detected where there is none)
# Now you can calculate Se, +P using formulas from lectures.
```

run_detector.sh (for evaluation of a detector implemented in Matlab/Octave):

```
#Matlab program and evaluation
#####
rm eval1.txt
rm eval2.txt
FILES=/path/to/directory/with/records/*.dat

for f in $FILES
do
  f=$(basename $f)
  f=${f%.*}

  echo  $f
  wfdb2mat -r $f #convert to Matlab format
done

#Run algorithm in Matlab. Output should be annotations in text files
#with WFDB annotator structure. See Matlab frame on the web classroom.

for f in $FILES
do
  f=$(basename $f)
  f=${f%.*}

  echo  $f

  #convert text annotator to WFDB format
  wrann -r $f -a det < $f".asc"
  #evaluate using reference annotations atr and your .det files
  bxb -r $f -a atr det -l eval1.txt eval2.txt

done

sumstats eval1.txt eval2.txt > results.txt #final statistics
#Now you can copy average Se and +P from results.txt

#Columns in results.txt that are of your interest are following:
# (note that the detector does not distinguish between different
# types of heart-beats, it just needs to detect them - so it should
# detect all N, V, F, etc. heart-beats as N, hence, FP=Nn+Vn+Fn,
# for heartbeat types see:
# https://archive.physionet.org/physiobank/annotations.shtml)
# - (Nn+Vn+Fn) = true positive (N, V, or F detected as N)
# - (No+Vo+Fo) = false negative (failed to detect N, V, or F)
# -  On = false positive (heartbeat was detected where there is none)
# Now you can calculate Se, +P using formulas from lectures.
```

In addition, note that some Octave functions tend to be slow, therefore, students who implemented the heart beat detection using Octave use records from MIT-BIH Arrhythmia database (there are less records, and they are shorter) to evaluate their chosen algorithm (instead of using the whole LTST DB).