IEEE Access
The journal for rapid open access publishing

Received November 4, 2015, accepted December 13, 2015, date of publication December 17, 2015,
date of current version December 28, 2015.

*Digital Object Identifier 10.1109/ACCESS.2015.2509426*

# An Open Source ECG Clock Generator for Visualization of Long-Term Cardiac Monitoring Data

**ALEX PAGE[1], (Student Member, IEEE), TOLGA SOYATA[1], (Member, IEEE), JEAN-PHILIPPE COUDERC[2], (Member, IEEE), AND MEHMET AKTAS[3]**
[1]Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627, USA
[2]Heart Research Follow-Up Program, University of Rochester Medical Center, Rochester, NY 14642, USA
[3]School of Medicine and Dentistry, University of Rochester Medical Center, Rochester, NY 14642, USA

Corresponding author: A. Page (alex.page@rochester.edu)

This work was supported in part by the National Science Foundation under Grant CNS-1239423, in part by the National Heart, Lung, and Blood Institute under Grant U24HL096556, and in part by Nvidia Corporation.

**ABSTRACT** The collection of long-term health data is accelerating with the advent of portable/wearable medical devices including electrocardiograms (ECGs). This large corpus of data presents great opportunities to improve the quality of cardiac care. However, analyzing the data from these sensors is a challenge; the relevant information from ∼120 000 heart beats per patient per day must be condensed into a human-readable form. Our goal is to facilitate the analysis of these unwieldy data sets. We have developed an open source tool for creating easy-to-interpret plots of cardiac information over long periods. We call these plots ECG clocks. The utility of our ECG clock library is demonstrated through multiple examples drawn from a database of 24-h Holter recordings. In these case studies, we focus on the visualization of heart rate and QT dynamics. The ECG clock concept is shown to be relevant for both physicians and researchers, for identifying healthy and abnormal values and patterns in ECG recordings. In this paper, we describe how to use the ECG clock library to analyze 24-h ECG recordings, and how to extend the source code for your own purposes. The tool is applicable to a wide range of cardiac monitoring tasks, such as heart rate variability or ST elevation. This library, which we have made freely available, can help provide new insights into circadian patterns of cardiac function in individuals and groups.

**INDEX TERMS** Electrocardiogram, heart rate variability, long QT syndrome, open-source software, visualization.

## I. INTRODUCTION

Holter monitors are portable ECG recorders used for long-term patient monitoring. They are incredibly valuable tools, as many ‘‘interesting’’ cardiac events will not occur during brief recordings in a clinic; the patient must instead be monitored throughout their normal daily activities. After recording a patient for some time — usually 24 hours — the data is extracted and analyzed. The physician is presented with a summary of events that occurred during the recording — min, max, and average value for a handful of important parameters like heart rate, for example. If a dangerous value is detected at some point in the recording, or the patient indicates that they experienced a symptom at a certain time of day, the doctor might investigate further by manually checking the ECG tracing at the time of the event. This process is somewhat tedious, particularly since doctors have many patients to monitor. Additionally, the recording summaries may drastically over-simplify the results, preventing the physician from uncovering important information; dangerous events or patterns can be washed out by the averaging process. There is a clinical need for a system that presents 24-hour ECG data in a simple form, without over-summarizing it.

While performing statistical analysis on ECG recordings in the Telemetric and Holter ECG Warehouse (THEW) [1], we realized that there was a great opportunity to study various cardiac features by hour, day of week, season, etc. Time and date can be extracted from Holter recordings, yet such analyses are rarely presented. With a feature like heart rate (HR), for example, we can provide much more than simply the mean or standard deviation in a given population; we can compute these statistics for *every minute of the day*, yielding more accurate reference values for clinical use. Futher, by using the demographic information in the THEW database, we are also able to separate results by age, gender, the presence of beta blockers, and other factors. Our analysis of heart rate in 200 healthy subjects revealed interesting features in the daily cardiac cycle, such as the difference and transition of HR between night and day, and maxima and minima around meals [2].

After our initial investigation of heart rate patterns in THEW recordings, we shifted our focus to another cardiac interval: QT. Prolongation of the QT interval can greatly increase the risk of ventricular arrhythmias, so it is an important marker for cardiologists to monitor. During the course of our research, we developed the ''QT Clock'' as a method of visualizing QTc — the ''corrected'' QT interval — over 24-hour data sets [3]. The QT Clock is a polar plot representing a 24-hour clock. The radius represents the value of the feature (QTc), and the angle is time of day. An example plot is given in Fig. 1.

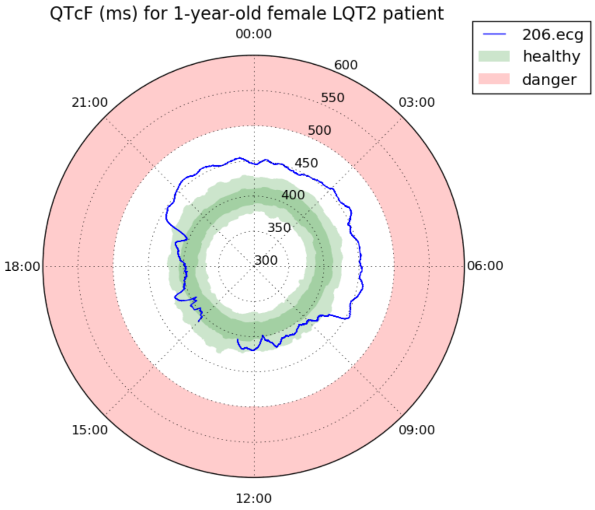


**FIGURE 1.** Example QT Clock. The recording is of a 1-year-old LQT2 female from the THEW E-HOL-03-0480-013 database. We see that she stays in the same QTc range as her healthy (i.e. no LQTS genotype) peers during the day, but has slightly prolonged QTc at night.

There are two particularly important features to note in Fig. 1: (1) the blue line, representing the value of a single cardiac feature (QTc) for a specific patient, and (2) the green area, representing the range of normal values for that feature based on analysis of recordings from healthy subjects. This presentation has many uses in clinical and research areas, as we will demonstrate in the later sections. Further, this visualization technique will continue to become more relevant; mobile ECG sensors such as [4] and [5] are becoming ubiquitous, which increases the availability of data sets for research, and will overwhelm physicians if the data cannot be condensed.

Anticipating the rapid growth of research in the area of long-term cardiac monitoring, we have expanded the *QT Clock* concept into the *ECG Clock*. That is, we apply the same QT plotting concept to other ECG parameters. The ECG Clock library we are releasing can be adapted to monitor virtually any cardiac feature over 24 hours. In this paper, we demonstrate the use of this open source tool through several case studies, focusing mainly on QTc and heart rate visualization. The remaining sections are broken down as follows: In Section II, we explain the significance of a few basic cardiac features, and how we compute those features (and their statistics) from our Holter recordings. In Section III, we describe the requirements, structure, and features of the ECG Clock library, and how to use it. In Section IV, we provide motivation for the use of ECG clocks through several case studies. Finally, we conclude in Section V with a brief discussion and pointers to future work.

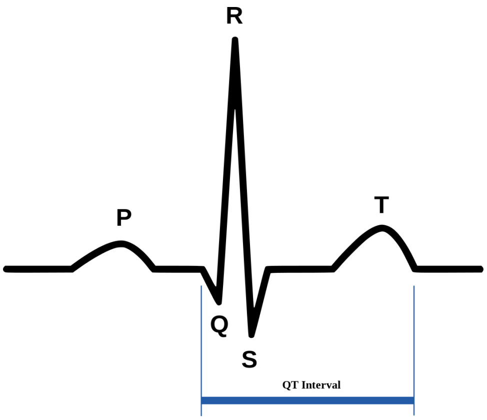


**FIGURE 2.** Typical ECG waveform for one cardiac cycle, with key ''waves'' annotated. Prolongation of the QT interval relative to the total cardiac cycle can indicate a high risk for adverse events. (Image source: SinusRhythmLabels.png by Anthony Atkielski).

## II. BACKGROUND

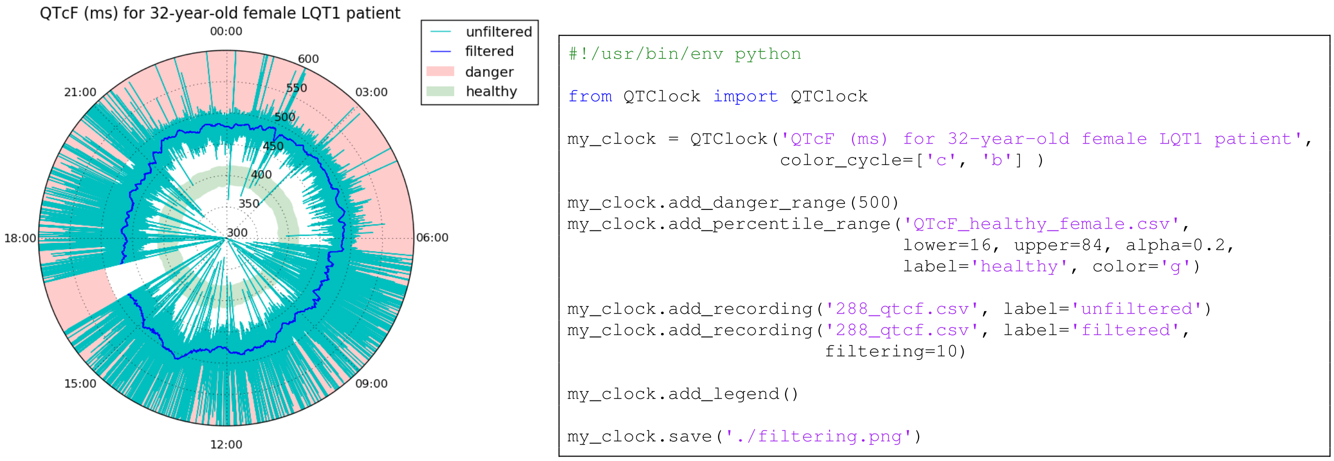
### A. CORRECTED QT INTERVAL (QTC)

The QT interval is the time from QRS onset to the end of the T wave in an ECG signal, and is depicted in Fig. 2. QTc is the ''corrected'' QT interval (based on heart rate), and is usually computed with one of the following two equations:

$$QTcB = \frac{QT}{\sqrt{RR/s}} \tag{1}$$

$$QTcF = \frac{QT}{\sqrt[3]{RR/s}} \tag{2}$$

where the ''B'' and ''F'' indicate that these are the Bazett [6] and Fridericia [7] corrections, and the division by one second is to preserve the units of QT. The normal range of QTcB is roughly 356–449 ms in men, and 369–460 ms in women [8]. QTc may be prolonged by drugs or genetic factors, and dangerous prolongation is known as the Long QT Syndrome (LQTS). When such prolongation occurs, subjects are prone to potentially-fatal arrhythmias like torsades de pointes (TdP) [9]. LQTS is responsible for an estimated 3000–4000 sudden deaths in children and young adults in the US each year [10]. At least thirteen genes have been identified that contribute to Long QT; LQT1 and LQT2 are the most common [11].



```
#!/usr/bin/env python

from QTClock import QTClock

my_clock = QTClock('QTcF (ms) for 32-year-old female LQT1 patient',
                   color_cycle=['c', 'b'] )

my_clock.add_danger_range(500)
my_clock.add_percentile_range('QTcF_healthy_female.csv',
                              lower=16, upper=84, alpha=0.2,
                              label='healthy', color='g')

my_clock.add_recording('288_qtcf.csv', label='unfiltered')
my_clock.add_recording('288_qtcf.csv', label='filtered',
                       filtering=10)

my_clock.add_legend()

my_clock.save('./filtering.png')
```

(a) (b)

**FIGURE 3. Filtering noisy annotations. This recording is from a 32-year-old female LQT1 patient in the THEW E-HOL-03-0480-013 database. Over 80,000 QTc values were annotated. Directly plotting these values produces the cyan line. Applying a median filter with a 10 minute window (i.e., passing `filtering=10` to the `add_recording()` function) produces the blue line. (a) Output. (b) Code.**

In at-risk patients, QT (and QTc) are typically computed from 10-second ECG snapshots during clinic hours. Or, if a Holter monitor is used, the physician is usually presented with very basic QT/QTc statistics such as min/max/mean for the 24 hour recording. As we mentioned in Section I, it is difficult or impossible to identify risky times of day for a given patient using only these checkup techniques. LQT2 patients, for example, are known to have more severe QTc prolongation at night, whereas LQT1 patients are more likely to experience symptoms during exercise [12]. The "concealment" of LQTS in LQT2 patients during clinic hours poses one diagnostic challenge, and the varying penetrance of various LQTS genes poses another. To overcome these obstacles, we will characterize QTc across different populations/genotypes, and also plot QTc over 24-hour recordings. This presentation will allow identification of dangerous times or events, and will also be useful in drug trials and induction protocols.

### B. THE QT CLOCK

A single 24-hour Holter recording consists of roughly 120,000 QTc data points, and the inevitable noise due to sensor or annotation issues will make any plot of this data unreadable. We usually remove this noise with a median filter, as we demonstrate in Fig. 3. The tradeoff of the filtering process is that short duration events will be removed, but this is not a concern when we are only interested in longer duration trends. If you are concerned about brief events, and your annotations are accurate enough, you may choose to plot the unfiltered beat-to-beat data.

When plotting long-term monitoring data on conventional Cartesian axes, we often found the time axis to be cumbersome. We could, for example, always start the axis at midnight, which resulted in a discontinuity when reading the plot. Or, we could align the axis to the start of the recording, but this slowed down interpretation of the plot because interesting times like "night" or "morning" were in different locations for every plot. We finally settled on the 24-hour clock in polar coordinates, which removes the discontinuities and standardizes the locations of different times of day.

In addition to the QTc data of a particular patient/recording, we may also highlight regions on the plot where QTc would be considered "normal" or "abnormal". We typically present the healthy range in green, and dangerous range in red, as in Fig. 1. We explain how these regions are generated in Section II-D.

Applications of the QT Clock to a database of LQTS Holters was the topic of [3]. The concept and components of the QT Clock form the basis for the more general ECG Clock library, which is the topic of this paper.

### C. DATA PREPARATION

In order to plot features like HR and QTc, we must first compute them. Holter recordings are generally not annotated with beat-to-beat interval values; they simply provide the raw ADC data (i.e. amplitude vs. time) for each ECG lead. We have access to hundreds of such recordings in the THEW database, and will extract the relevant values from them for our examples. To accomplish this, we use an award-winning [13] open source library developed by Yuriy Chesnokov [14] to annotate recordings from two primary THEW databases: Healthy (E-HOL-03-0202-003), and Genotyped LQTS (E-HOL-03-0480-013). The annotation library provides — among other things — the locations of Q onset, R, and T offset for every beat on each lead. We merge the annotations from all leads into a single list, keeping the median RR at each heart beat and the worst possible (i.e. longest) QT. QTc and HR are then computed from this consolidated data set. Noise/outliers are removed during plotting, as demonstrated in Section III-D. This process is described in more detail in [2] and [15]. We will not be looking at any other features (such as PR or ST) in this paper, but the method of extracting them would be similar.

### D. STATISTICS FOR DECISION SUPPORT

As we saw in Fig. 1, it is helpful to compare a single patient to a larger population, e.g. to check where the patient's QTc falls relative to healthy subjects. We would therefore like the ECG Clock Library to be capable of loading reference ranges from disk, and adding them to the plot — i.e., we want to store the values used to generate the green (i.e. healthy) range in the figure.

Typical ranges for QTc (and other ECG parameters) have been thoroughly investigated [8], but reference ranges that take precise time of day into account do not exist. Because of the time-dependence of LQTS that we have already discussed, we have developed our own reference ranges from the THEW databases mentioned in Section II-C. The ECG Clock Library includes these ranges for many populations (separated by gender, LQT genotype, etc.), in CSV files with the following columns:

| time of day | value (0th percentile) | value (1st percentile) | ... | value (100th percentile) |
|---|---|---|---|---|

.

Currently, we have only computed these spreadsheets for HR and QTc. QTc values were pre-filtered with a sliding 10-minute median, but HR was not filtered because it is much easier to annotate and therefore less noisy.

Use of these "pre-computed normal range" files is described in Section III-D. You may use our results, or provide your own values. Our results were generated using the annotation algorithm introduced in II-C ([14]). You should be warned that while this algorithm performed well in a relevant contest, it is unproven in a clinical setting. Further, the ranges were computed from our relatively small database, which becomes even smaller when broken down by gender and genotype.

## III. THE ECG CLOCK LIBRARY

The ECG Clock library is written with the primary purpose of generating plots of ECG interval values on a 24-hour axis. There is a wide range of applications for such plots, some of which will be demonstrated in Section IV. Users may modify the code if desired; it is released under the permissive MIT License, and is available at https://bitbucket.org/atpage/ecgclock/. Extending the functionality of the library to cover new cases usually only requires minor changes.

### A. SYSTEM REQUIREMENTS

To generate plots, we use Python and the well-known `matplotlib` library [16] — in fact, much of the ECG Clock code is simply wrappers around `matplotlib`. Working in Python also allows us to parse input files very easily; the `dateutil` module is quite helpful in this process. The final module that is required (outside of the standard library) is `numpy`. The code has been tested in Python 2.7 and 3.4.

### B. INPUT FILE SPECIFICATIONS

ECG annotations should be provided as CSV files. Ideally, the file will contain one {time, value} pair per row, but you may also choose to specify the column numbers to plot from a larger file. Time may be stored in relatively arbitrary strings, such as "11:16" or "2015-07-15T11:16:00.535". Values (e.g. QTc) should be stored in milliseconds, but the library will attempt to convert them from seconds if necessary. If this automatic conversion should not be applied to your data, you may choose to create a new subclass for your application, or comment/remove the `sec_to_msec()` calls. Here is the head of an example CSV file, where we have added some comments about irregularities in the data:

```
time,QTcB
07:33:06.683,0.408
07:33:07.578,0.404
07:33:08.467,0.407
07:33:09.350,0.413
07:33:10.239,0.412
07:33:11.139,0.409
07:33:12.033,0.404
07:33:15.539,0.409 # missing beats
07:33:16.361,0.415
07:33:17.178,0.411
07:33:18.011,0.453 # noise
07:33:21.117,0.425 # missing beats
07:33:21.889,0.408
07:33:22.733,0.483 # noise
07:33:23.444,0.409
```

Although an unlimited number of recordings can be added to the same plot — to view a patient's response to different prescriptions, for example — we find that the plots tend to get cluttered with more than 3–4 recordings. Incorporation of additional information (e.g. from more sensors, which are not necessarily cardiac) on the same axes is an ongoing research challenge [17]; we expect that plots of heart rate and QTc together, or QTc and TpTe, for example, will make it easier to gauge the interaction between related features.

### C. STRUCTURE AND USAGE

The ECG Clock library consists of one main class, `ECGClock`, that provides most standard functions such as the ability to add a recording or an annotation to a polar axis, save the plot to a file, etc. For our analysis of QTc, we created a subclass called `QTClock` that adds features such as highlighting standard "dangerous" ranges specific to QTc. (If, for example, you intend to look at ST segment amplitude, perhaps you would subclass `ECGClock` as `STClock` and add functions that highlight areas of ST depression or elevation in different colors, as well as changing the default axes ranges in `__init__`.) Finally, there is an `ECGFigure` class which basically holds many `ECGClock` subplots. You may think of `ECGFigure` and `ECGClock` as behaving like the `Figure` and `Axes` objects in `matplotlib`, though they are not actually extensions of those classes.

There are currently two ways to interface with the library:

1) Importing it in another Python script. Using this method, the end user will likely need less than 10 lines

of code to generate each of their plots. See `clock_example.py` in the Git repository for a demonstration, or the code in the later Figures.

2) For simple QT clocks, you may directly run `QTClock.py` from the command line. This method does not provide access to all features, but can be used to generate basic QTc plots in a single line. `QTClock.py -h` explains the syntax; an example would be:

```
./QTClock.py −i qtcf_ann.csv  −f 10
              −o output.png
```

to plot the QTc values from `qtcf_ann.csv` on a standard axis, with a 10 minute filter width, and save as `output.png`.

### D. BASIC FEATURES

*Output Options:* Plots may be saved to disk using the `save()` function, or displayed in an interactive window using `show()`. File output is ideal for batch processing, whereas the interactive view is useful for measuring QTc at "interesting" times and/or adjusting the plot range before saving.

*Ranges:* The red and green background highlights in Fig. 1 provide decision support for the clinician. The utility of the static red range is self-explanatory, and for QT clocks, it can be added and customized using the `add_danger_range()` function. The green area displayed in the figure is more dynamic; we show the interquartile range (IQR) of QTc in healthy patients (using the stored ranges from Section II-D) in dark green, and a wider percentile range (5–95%) of QTc in those patients in lighter green. This provides a reference of what is normal/healthy. You may also elect to show a static range — 350 ms to 450 ms, for example — rather than varying the values throughout the day. This is possible with the `add_healthy_range()` function. Or, instead of comparing a patient to the healthy group, you may wish to compare them to their peers with the same genotype. To do this, you may select, for example, "LQT2_female.csv" rather than "healthy_female.csv" when calling `add_percentile_range()`. Note that due to the relatively small number of recordings in the THEW databases, extreme percentile values (such as 99%) are likely to reveal noise rather than accurate values at some times of day. For this reason, `add_percentile_range()` accepts a `smoothing` argument. With our data and annotation algorithm, roughly the 20–80 percentile range seems to provide a good reference.

*Filtering:* Beat-to-beat annotations are often very noisy, requiring filtering to view. You do not need to pre-filter such data before opening it in the ECG Clock library; you may simply specify a median filter width as an argument to `add_recording()`. Five to ten minutes is typically a good width, but very noisy data may require an even wider window. If course, if your data is fairly clean — which is possible through more advanced annotation algorithms and/or having more leads to choose from — you may choose to reduce the filter width or disable it entirely. This will be necessary if you are interested in shorter duration events, e.g. certain changes in ST segment levels. Other basic filter types (like mean or max) are simple to implement, as a `general_filter()` function is provided. This function does not use a filter window size specified in *number of samples*, but in *time*. This is important because beat-to-beat samples will not be uniformly spaced. Fig. 3 shows the result of applying a 10-minute median filter to a set of QTc annotations containing lots of outliers.

*Subplots:* To display multiple plots in the same window or file, you can create an `ECGFigure` object and specify the number of rows and columns. Then, when creating an ECG Clock, you specify the "parent figure" and the clock's location on that figure. Single-plot figures also work this way, as a subplot inside a parent figure, but the parent figure is created automatically when you instantiate a clock without specifying a parent.

*Annotations:* An `add_annotation()` function is available to add text labels and arrows to a plot. You must specify the location of the arrow head (time, radius) and tail (x, y location in the figure). A typical use for this function might be to identify the location of a maximum value.

## IV. CASE STUDIES

In this section, we will demonstrate the utility of several plots from the perspectives of both clinicians and researchers.

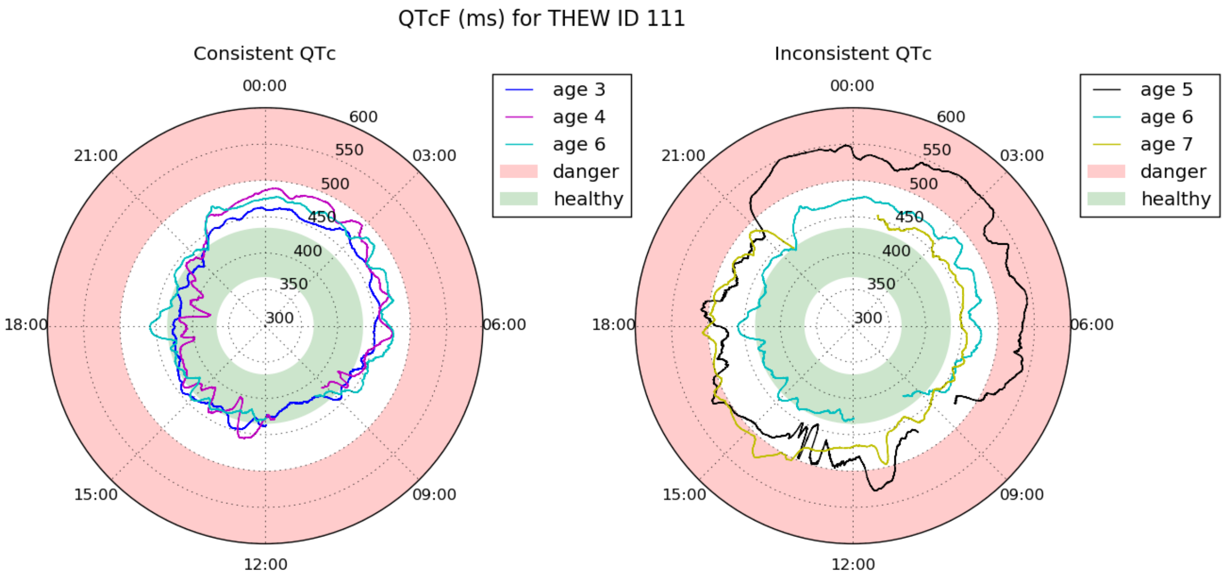
### A. QTC VS. AGE

In Fig. 4, we show the same patient's QTc recorded at five different ages. On the left, we see that QTc is very stable when comparing ages 3, 4, and 6. At ages 5 and 7, though, his QTc is prolonged; this is shown on the right, along with the age 6 plot for reference. This patient's QTc indicated relatively low risk in the recordings on the left, yet it indicates high risk during the two "anomalous" recordings [18]. Unfortunately, we do not have information about prescriptions or other possible causes for the prolongation at ages 5 and 7, but his physician would immediately investigate the cause when presented with the plot.

Finally, note the distinct "LQT2-like" shape of the plots, where QTc increases at night (similar to the patient in Fig. 1). This asymmetry could aid in diagnosis, preempting genetic testing in some cases.

### B. DRUG TRIAL

Another typical use of plotting multiple Holters on the same axes is shown in Fig. 5, where we compare a patient's baseline QTc and heart rate to his QTc and heart rate on an antiarrhythmic drug. The drug was administered in the morning, and we can see its effect on QTc increase into the afternoon, drop off until roughly midnight, and then re-emerge during sleep. The effect on heart rate is more immediate and consistent throughout the day. This presentation may encourage the drug manufacturer to perform longer-duration monitoring, and/or help to characterize the drug's typical impact.



(a)

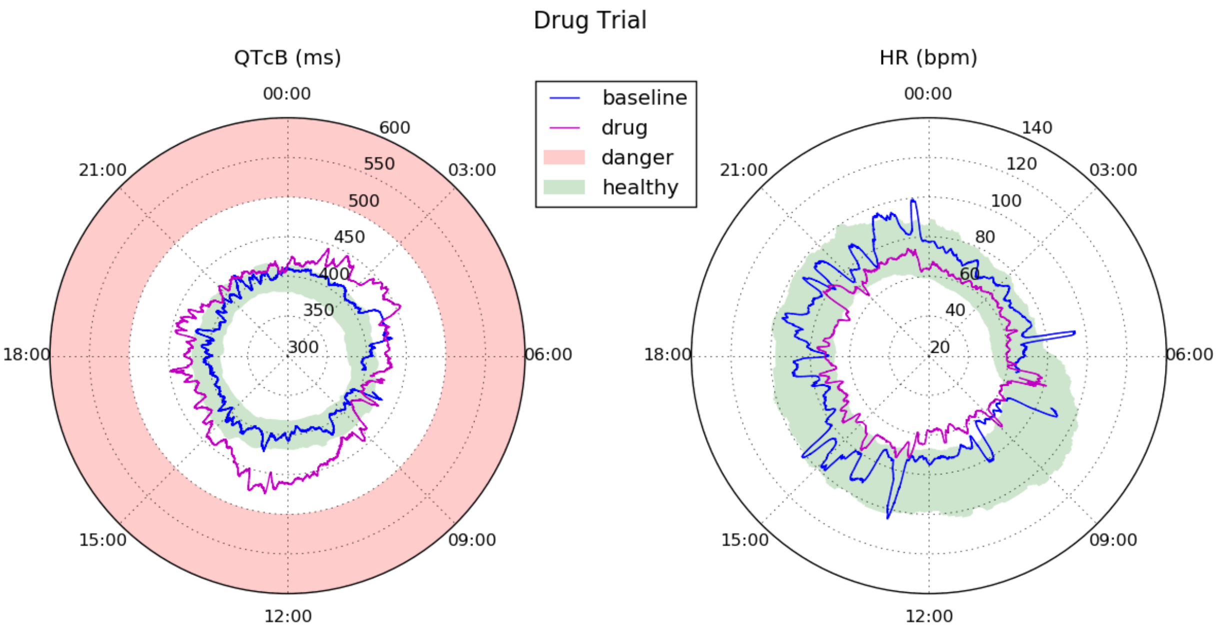
```
#!/usr/bin/env python

from ECGFigure import ECGFigure
from QTClock import QTClock

my_fig = ECGFigure(title='QTcF (ms) for THEW ID 111', ncols=2)

left_clock = QTClock('Consistent QTc', parent_figure=my_fig, subplot=1)
right_clock = QTClock('Inconsistent QTc', parent_figure=my_fig, subplot=2)

left_clock.add_default_ranges()
right_clock.add_default_ranges()

left_clock.add_recording('109_qtcf.csv', label='age 3', filtering=20, color='b')
left_clock.add_recording( '10_qtcf.csv', label='age 4', filtering=20, color='m')
left_clock.add_recording(  '8_qtcf.csv', label='age 6', filtering=20, color='c')

right_clock.add_recording( '7_qtcf.csv', label='age 5', filtering=20, color='k')
right_clock.add_recording( '8_qtcf.csv', label='age 6', filtering=20, color='c')
right_clock.add_recording( '9_qtcf.csv', label='age 7', filtering=20, color='y')

left_clock.add_legend()
right_clock.add_legend()

my_fig.save('./aging.png')
```

(b)

**FIGURE 4. QTcF evolution with age. This male LQT2 patient (on beta blockers) was recorded annually between ages 3 and 7. For the first two years, his QTc is relatively healthy during the day, and somewhat prolonged at night. In the third recording, though, his QTc has increased by over 50 ms throughout the day compared to the previous year. It then returns to the previous baseline range at age 6, then prolongs again (during the day) at age 7. These deviations from the baseline would alert the doctor to potential drug interactions or hormonal changes. Note that we have chosen to show static ranges for "healthy" and "dangerous" QTc values, using the** `add_default_ranges()` **function. (a) Output. (b) Code.**

For the heart rate plot, we use the generic `ECGClock` class and simply modify the default axis range to 20–140. The green range for heart rate should perhaps be viewed as "normal" rather than "healthy"; it represents the typical range of values for the healthy population, but values outside of that range are not necessarily dangerous.

Some patients may have drastically different sleep schedules than the average person, e.g. due to working third shift. In the figure, we notice that this patient's heart rate pattern appears to be misaligned by a couple of hours compared to the "normal" range. In cases like this, it may be desirable to rotate the "expected" range to match the patient's schedule.



(a)

```
#!/usr/bin/env python

from ECGFigure import ECGFigure
from ECGClock import ECGClock
from QTClock import QTClock

my_fig = ECGFigure(nrows=1, ncols=2, title='Drug Trial')

qt_clock = QTClock(title='QTcB (ms)', parent_figure=my_fig, subplot=1)
qt_clock.add_recording('baseline_qtcb.csv', filtering=10, label='baseline')
qt_clock.add_recording('sotalol_qtcb.csv', filtering=10, label='drug')
qt_clock.add_percentile_range('QTcB_healthy_male.csv', label='healthy', color='g',
                              lower=16, upper=84)
qt_clock.add_danger_range()
qt_clock.add_legend()

hr_clock = ECGClock('HR (bpm)', parent_figure=my_fig, subplot=2,
                    min_rad=20, max_rad=140)
hr_clock.add_recording('baseline_hr.csv', filtering=10, label='baseline')
hr_clock.add_recording('sotalol_hr.csv', filtering=10, label='drug')
hr_clock.add_percentile_range('HR_healthy_male.csv', label='normal', color='g',
                              lower=16, upper=84)

my_fig.save('./sotalol.png')
```

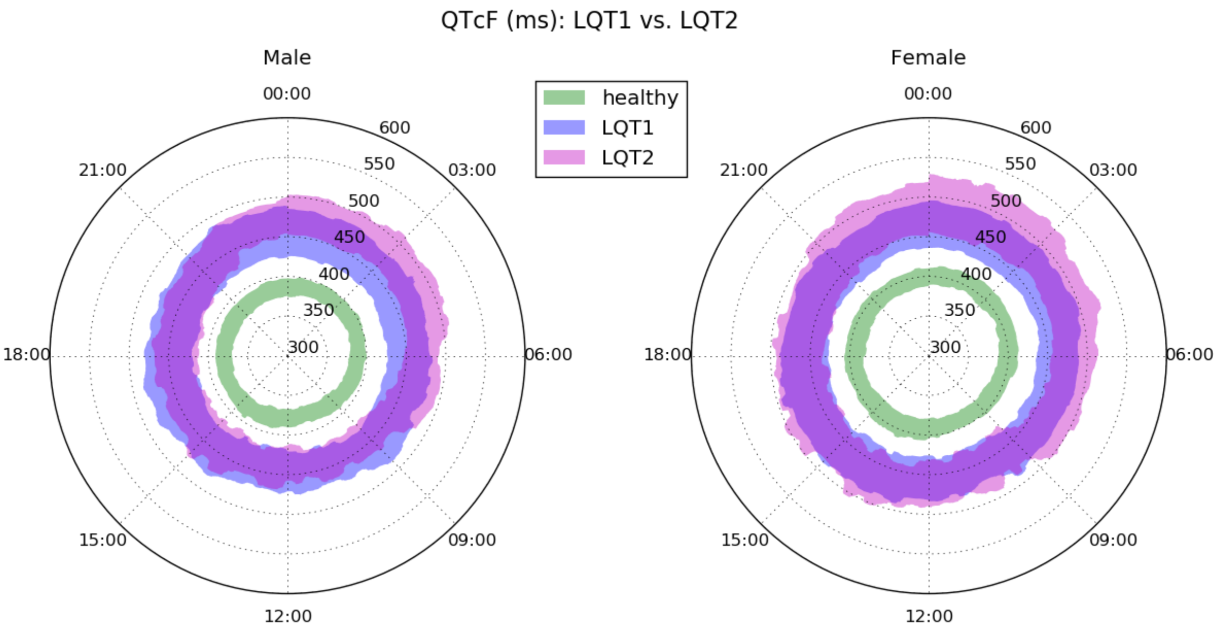
(b)

**FIGURE 5. Effects of sotalol (a beta blocker/antiarrhythmic drug) on a healthy subject. Green ranges are defined by the inner 68 percentile in healthy male patients, i.e., approximately equivalent to plus/minus one standard deviation. The patient's heart rate is lower and less variable on the drug, and their QTc is much higher — both known effects of sotalol. A cardiologist would be able to use similar plots for their patients to determine if prescriptions were working as expected, and also to monitor medication adherence. (a) Output. (b) Code.**

We will be adding an `offset` parameter to the library to allow this. (This will also be useful to adjust annotation data containing incorrect timing information.)

### C. COMPARING GENOTYPES

So far we have looked mostly at individual Holter recordings. The ability to compare *groups* of recordings is also very useful, though. For example, in a drug trial, we may want to compare a large group of baseline Holters to a group of ''on drug'' Holters. An example where we compare two groups of patients is given in Fig. 6. In this figure, we look at the inter-quartile range (IQR) of QTc in patients with LQT1 and LQT2, with healthy patients also shown for reference. We can clearly see that QTc prolongation only increases



(a)

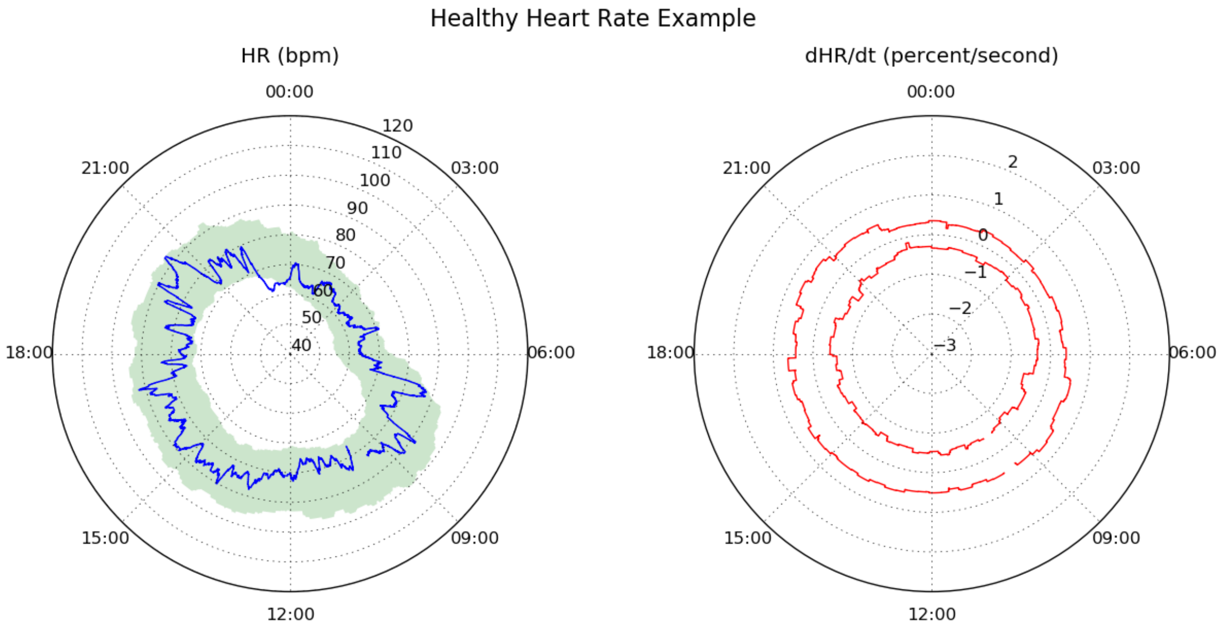
```
#!/usr/bin/env python

from ECGFigure import ECGFigure
from QTClock import QTClock

my_fig = ECGFigure(title='QTcF (ms): LQT1 vs. LQT2', ncols=2)

my_clock = QTClock('Male', parent_figure=my_fig, subplot=1)
my_clock.add_percentile_range('QTcF_healthy_male.csv', color='g', alpha=0.4,
                              label='healthy')
my_clock.add_percentile_range('QTcF_LQT1_male.csv', color='b', alpha=0.4,
                              label='LQT1')
my_clock.add_percentile_range('QTcF_LQT2_male.csv', color='m', alpha=0.4,
                              label='LQT2')

my_clock.add_legend()

my_clock = QTClock('Female', parent_figure=my_fig, subplot=2)
my_clock.add_percentile_range('QTcF_healthy_female.csv', color='g', alpha=0.4,
                              label='healthy')
my_clock.add_percentile_range('QTcF_LQT1_female.csv', color='b', alpha=0.4,
                              label='LQT1')
my_clock.add_percentile_range('QTcF_LQT2_female.csv', color='m', alpha=0.4,
                              label='LQT2')

my_clock.save('./lqt1_vs_2_IQR.png')
```

(b)

**FIGURE 6. Comparing two groups. In this case, we are curious about the difference in QTc prolongation between LQT1 and LQT2 subjects. We plot the IQR of QTc for males and females in each category, as well as for their healthy counterparts. The result clearly shows that LQT2 patients have higher prolongation at night, despite being almost indistinguishable from LQT1 patients during the day. We also note that most LQT1 patients do not enter the dangerous QTc range (>500 ms), whereas many LQT2 patients — particularly females — do at night. (a) Output. (b) Code.**

at night in the LQT2 group; in LQT1, it is quite stable. Additionally, most LQT1 patients never enter the "danger" area, whereas LQT2 patients have a relatively high chance of reaching dangerous prolongation levels during sleep. This type of plot tells a much more complete story than a list of basic statistics from the recordings.

To generate the data for this plot, we annotated the QT and RR intervals in all healthy, LQT1, and LQT2



(a)

```
#!/usr/bin/env python

from ECGFigure import ECGFigure
from ECGClock import ECGClock
from HRDerivClock import HRDerivClock

my_fig = ECGFigure(ncols=2, title='Healthy Heart Rate Example')

hr_clock = ECGClock('HR (bpm)', parent_figure=my_fig, subplot=1, min_rad=40,
                    max_rad=120)
hr_clock.add_recording('9004_hr.csv', filtering=10)
hr_clock.add_percentile_range('HR_healthy_male.csv', label='normal', color='g')

dhr_clock = HRDerivClock('dHR/dt (percent/second)', parent_figure=my_fig,
                         subplot=2, min_rad=-3, max_rad=3, tick_spacing=1.0)
dhr_clock.add_recording('9004_hr.csv', percent=True)

my_fig.save('hr_example.png')
```

(b)

**FIGURE 7. Heart rate and time derivative of heart rate for a healthy subject. The plot on the left provides the doctor with a comprehensive picture of the patient's heart rate and HRV. The green region indicates the interquartile range for heart rate in healthy subjects. The red lines on the right represent the upper and lower bounds for rate of change, dHR/dt, normalized to *percentage* change rather than *bpm* change. This shows how fast the patient's heart rate is able to change, and is an example of how the ECG Clock library can be easily extended to display unconventional features. (a) Output. (b) Code.**

Holters from the THEW database, used this information to compute QTc at every heart beat, and finally computed the percentiles for all beats in each 1-minute window. The resulting values are stored in the CSV files described in Section II-D.

### *D. HEART RATE DIPPING*

Elevated heart rate during sleep — i.e., failure of the heart rate to "dip" to a low enough level — has been associated with cardiovascular disease and an increased risk of all-cause mortality [19], [20]. Similarly, low heart rate variability (HRV) is an indicator of risk for cardiac events [21]. A heart rate plot in the same style as Fig. 1 will provide insight into these two critical pieces of information.

In Fig. 7, we observe the heart rate of a typical healthy patient. On the left, we see that the patient's heart rate drops from ∼80 bpm during the day to ∼65 bpm at night. From this plot, we also get an idea of heart rate variability (HRV); the heart rate appears to fluctuate across a range of ∼10 bpm throughout the day.

In addition to the *range* of HR, we may also be interested in its *rate of change*. The `HRDerivClock` class was created as an example of how the ECG Clock can be extended to view other features. In this class, we redefine the default axis range, and apply a `derivative()` function to the heart rate data as it is loaded. We can then plot either the derivative at each data point, or its upper and lower bounds within a sliding window. On the right side of Fig. 7, we show the upper and lower bounds. These bounds tend to stay at around ±0.5 percent/second, meaning that a change from 80 to 65 bpm (about 20%) would take at least 40 seconds. We further note that changes take place more slowly at night. A very narrow range on this plot may indicate that the patient has trouble adapting their heart rate to different situations.

## V. CONCLUSION

We have demonstrated that QT clocks could be very useful for diagnosis and monitoring of the Long QT Syndrome. They are also instructive in research involving both the congenital and drug-induced forms of this disease. Likewise, we have seen that HR clocks can be used for monitoring HRV and drug response. We expect that researchers will want to apply (and extend) these concepts to many other cardiac features, which is why we have decided to make our source code freely available.

The ECG clock in its current form is not very quantitative; we are simply plotting the values of ECG features throughout the day, and allowing the doctor to decide on a course of action. The highlighted regions can aid in this decision, but they don't currently provide numerical values/statistics like "percentage of heart beats outside of normal range", etc. Further, the annotation data could be used as input to a more advanced decision support system, providing the doctor with not only the picture, but recommendations. We will be pursuing this course of research next.

Another feature we intend to add to the library is the ability to read a set of individual annotation files (e.g., HR values for a group of 50 patients) and produce a percentile range CSV file compatible with our `add_percentile_range()` function.

ECG tracings are typically always presented at the same scale — 10 mm/mV, and 25 mm/s — allowing clinicians and researchers to develop their intuition about "normal" vs. "abnormal" tracings. Axes ranges for ECG clocks should be standardized for the same reason. For QTc, we have found that 300–600 ms is usually a good plot range, but in extreme cases it must be extended up to 700 ms. As ECG clocks become more prevalent, this type of standardization will allow clinicians to visually identify abnormalities in the same way they currently analyze ECGs, X-rays, ultrasounds, etc.

Finally, in order to make the library more accessible, we intend to develop a web interface so that non-programmers can simply upload data or annotations and generate clocks. Additionally, example IPython notebooks will simplify the process for programmers who prefer that environment.

## REFERENCES

[1] J.-P. Couderc, "The telemetric and Holter ECG warehouse initiative (THEW): A data repository for the design, implementation and validation of ECG-related technologies," in *Proc. 32nd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug./Sep. 2010, pp. 6252–6255. [Online]. Available: http://www.thew-project.org/

[2] A. Page, M. Hassanalieragh, T. Soyata, M. K. Aktas, B. Kantarci, and S. Andreescu, "Conceptualizing a real-time remote cardiac health monitoring system," in *Enabling Real-Time Mobile Cloud Computing Through Emerging Technologies*. Hershey, PA, USA: IGI Global, 2015.

[3] A. Page, M. K. Aktas, T. Soyata, W. Zareba, and J.-P. Couderc, "'QT clock' to improve detection of QT prolongation in long QT syndrome patients," *Heart Rhythm*, Aug. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1547527115011273

[4] *AliveCor Mobile ECG*. [Online]. Available: http://www.alivecor.com/, accessed Aug. 2015.

[5] *HeartCheck Handheld ECG Device*. [Online]. Available: http://www.theheartcheck.com/, accessed Aug. 2015.

[6] H. C. Bazett, "An analysis of time relations of the electrocardiogram," *Heart*, vol. 7, pp. 353–370, Aug. 1920.

[7] L. S. Fridericia, "Die systolendauer im Elektrokardiogramm bei normalen menschen und bei herzkranken," *Acta Med. Scandin.*, vol. 53, no. 1, pp. 469–486, 1920.

[8] J. W. Mason, D. J. Ramseth, D. O. Chanter, T. E. Moon, D. B. Goodman, and B. Mendzelevski, "Electrocardiographic reference ranges derived from 79,743 ambulatory subjects," *J. Electrocardiol.*, vol. 40, no. 3, pp. 228–234.e8, 2007.

[9] P. Coumel, J.-F. Leclercq, and V. Lucet, "Possible mechanisms of the arrhythmias in the long QT syndrome," *Eur. Heart J.*, vol. 6, no. suppl D, pp. 115–129, 1985.

[10] G. M. Vincent, "The molecular genetics of the long QT syndrome: Genes causing fainting and sudden death," *Annu. Rev. Med.*, vol. 49, no. 1, pp. 263–274, 1998.

[11] P. L. Hedley *et al.*, "The genetic basis of long QT and short QT syndromes: A mutation update," *Human Mutation*, vol. 30, no. 11, pp. 1486–1511, 2009. [Online]. Available: http://dx.doi.org/10.1002/humu.21106

[12] P. J. Schwartz *et al.*, "Genotype-phenotype correlation in the long-QT syndrome: Gene-specific triggers for life-threatening arrhythmias," *Circulation*, vol. 103, no. 1, pp. 89–95, 2001.

[13] G. B. Moody, H. Koch, and U. Steinhoff, "The PhysioNet/computers in cardiology challenge 2006: QT interval measurement," in *Proc. Comput. Cardiol.*, Sep. 2006, pp. 313–316.

[14] Y. C. Chesnokov, D. Nerukh, and R. C. Glen, "Individually adaptable automatic QT detector," in *Proc. Comput. Cardiol.*, Sep. 2006, pp. 337–340.

[15] A. Page, T. Soyata, J.-P. Couderc, M. Aktas, B. Kantarci, and S. Andreescu, "Visualization of health monitoring data acquired from distributed sensors for multiple patients," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015.

[16] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, May/Jun. 2007.

[17] M. Hassanalieragh *et al.*, "Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges," in *Proc. IEEE Int. Conf. Services Comput.*, New York, NY, USA, Jun./Jul. 2015, pp. 285–292.

[18] S. G. Priori *et al.*, "Risk stratification in the long-QT syndrome," *New England J. Med.*, vol. 348, no. 19, pp. 1866–1874, 2003.

[19] I. Z. Ben-Dov, J. D. Kark, D. Ben-Ishay, J. Mekler, L. Ben-Arie, and M. Bursztyn, "Blunted heart rate dip during sleep and all-cause mortality," *Arch. Internal Med.*, vol. 167, no. 19, pp. 2116–2121, 2007. [Online]. Available: http://dx.doi.org/10.1001/archinte.167.19.2116

[20] K. Eguchi *et al.*, "Nocturnal non-dipping of heart rate predicts cardiovascular events in hypertensive patients," *J. Hypertension*, vol. 27, no. 11, pp. 2265–2270, Nov. 2009.

[21] H. Tsuji *et al.*, "Impact of reduced heart rate variability on risk for cardiac events: The Framingham heart study," *Circulation*, vol. 94, no. 11, pp. 2850–2855, 1996. [Online]. Available: http://circ.ahajournals.org/content/94/11/2850.abstract

**ALEX PAGE** grew up in Maine. He received the bachelor's degrees in computer engineering, physics, and applied mathematics from the University of Rhode Island, in 2011. He entered the U.S. Navy straight out of high school. After six years of service, he moved to RI. In 2012, he was accepted to the University of Rochester's Electrical Engineering Ph.D. program, where he is currently with the Dr. Soyata's Research Group. His research is currently focused on computer systems for medical data processing.

**TOLGA SOYATA** received the B.S. degree in electrical and communications engineering from Istanbul Technical University, in 1988, the M.S. degree in electrical and computer engineering from Johns Hopkins University, in 1992, and the Ph.D. degree in electrical and computer engineering from the University of Rochester, in 1999. He joined the ECE Department, University of Rochester, in 2008, where he is currently an Assistant Professor-Research. He manages the CUDA Research Center and CUDA Teaching Center programs with the University of Rochester, and Xilinx University Program and MOSIS Educational Program with the ECE Department. He teaches courses in VLSI ASIC design, GPU parallel programming, and FPGA-based advanced digital design. His current research interests include D-health, energy aware computation, and GPU-based high-performance computing.

**JEAN-PHILIPPE COUDERC** received the Ph.D. (Hons.) degree from the French National Institute of Applied Sciences, Lyon, France, in 1997, and the M.B.A. degree from the Simon School of Business, Rochester, NY. He is an Associate Professor of Medicine with the Cardiology Department of Strong Memorial Hospital, University of Rochester (UR) Medical Center, NY, and a Research Associate Professor of Electrical and Computer Engineering with UR. He is the Director of the Center for Quantitative Electrocardiography and Cardiac Safety (THEW-project.org), and the Co-Director of the Heart Research Follow-Up Program. He is a Principal Investigator and co-Investigator in several federal-funded research grants involving the development of ECG technologies, including drug-safety markers. In addition, he is the Chief Technology Officer of iCardiac Technology Inc., a Rochester-based spinoff.

**MEHMET AKTAS** grew up in Rochester, NY. He received the B.A. degree in biology from the University of Rochester, and the master's degree in business administration from the Simon School, University of Rochester. He completed his medical school education with SUNY Upstate Medical University in 2002. He completed Internal Medicine Residency Training with Cleveland Clinic and then proceeded to the University of Rochester Medical Center (URMC), where he completed advanced fellowships in cardiovascular diseases and cardiac pacing and electrophysiology. He is an Associate Professor of Medicine with URMC. He is board certified in internal medicine, cardiovascular diseases, and cardiac pacing and electrophysiology. His clinical work involves the treatment of patients with a variety of complex heart rhythm disorders. His research is focused on improved risk stratification of patients with heart rhythm disorders and development of systems to enable early detection of arrhythmias.