Robust Detection of Heart Beats in Multimodal Data: The PhysioNet/Computing in Cardiology Challenge 2014

George Moody, Benjamin Moody, Ikaro Silva

Massachusetts Institute of Technology, Cambridge, USA

Abstract

The 15th annual PhysioNet/CinC Challenge aims to encourage the exploration of robust methods for locating heart beats in continuous long-term data from bedside monitors and similar devices that record not only ECG but usually other physiologic signals as well, including pulsatile signals that directly reflect cardiac activity, and other signals that may have few or no observable markers of heart beats. Our goal is to accelerate development of open-source research tools that can reliably, efficiently, and automatically analyze data such as that contained in the MIMIC II Waveform Database, making use of all relevant information.

Data for this Challenge are 10-minute (or occasionally shorter) excerpts ("records") of longer multi-parameter recordings of human adults, including patients with a wide range of problems as well as healthy volunteers. Each record contains four to eight signals; the first is an ECG signal in each case, but the others are a variety of simultaneously recorded physiologic signals that may be useful for robust beat detection. We prepared and posted 100 training records, and retained 300 hidden test records for evaluation of Challenge entries. A total of 1,332 entries from 60 teams were processed during the challenge period.

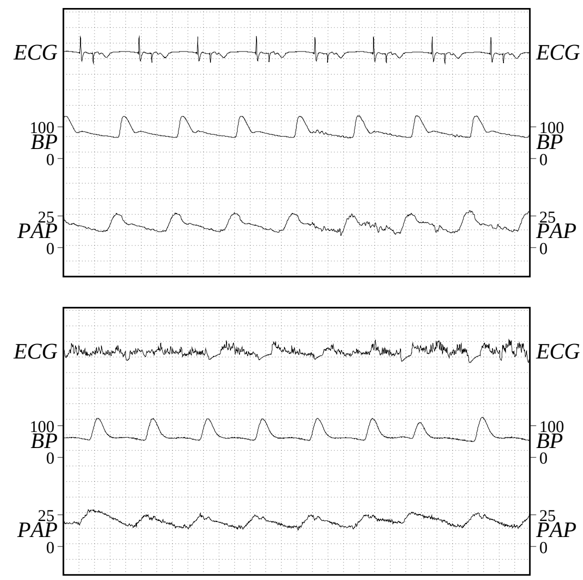

Figure 1. Two examples of records for which beat detection can be improved by utilizing information beyond the ECG channels. In the first case, the ECG contains pacemaker artifacts; in the second, the ECG is too noisy for QRS complexes to be identifiable. In both cases, clean arterial blood pressure (BP) and pulmonary arterial pressure (PAP) signals are available.

1. Introduction

Robust continuous detection of heart beats from bedside monitors plays a critical role in patient monitoring. Most existing beat detectors are QRS detectors, operating only on ECG, even though other sources of pulsatile activity, such as arterial blood pressure, are frequently being measured at the same time (for example, in an ICU.) While the ECG is available in many clinical and research settings, its quality may vary considerably over time, and on occasion the ECG signal may be missing entirely. An excellent QRS detector is thus limited by availability and quality of the incoming ECG signal. It is not clear to what extent the examination of other physiological signals such as blood pressure, electroencephalography (EEG), and respiration can help improve the detection of beats associated with heart activity. For instance, in most subjects, the observed relationships between respiration and heart rate can be used to model heart rate, and together with nearby context derived from ECG or other cardiac signals, these models could predict beat locations from respiratory signals.

In the PhysioNet/CinC 2014 Challenge, participants were given the task of writing an algorithm to examine an arbitrary multi-channel recording (provided to the participant's program in a standard format), and produce a series of annotations indicating the likely locations of heartbeats in the recording. The Challenge was divided into three phases, as shown in Table 1, and each individual or team was allowed to submit up to five entries for each phase. The major difference between the three phases, from the participants' point of view, was the number of records in the test set: 100 records were included in Phase I, then 100 new records were added for each subsequent phase, for a total of 300 records in Phase III. At the end of each phase, the five best-scoring entries were announced, and their code was published on PhysioNet, in an attempt to stimulate collaboration among the competitors. Entries that contained proprietary or copyrighted software were disqualified, and the corresponding files were removed from the published versions. Two sample entries were provided to competitors: one using a C API, and one using an Octave API [1]. Both entries used the WFDB Software Package application[2], 'gqrs', which processed only a single ECG signal in order to estimate the beat locations.

Table 1. Phases of the PhysioNet/CinC 2014 Challenge.

Phase	Period	N records
Phase I	January 7 - April 7	100
Phase II	April 16 - June 22	200
Phase III	June 23 - August 15	300

2. Challenge Data

The Challenge consisted of four data sets: a public training data set, and one hidden test set for each of the three phases. The data sets contained signals at most 10 minutes in length (or occasionally shorter). The signals were multi-parameter recordings of human adults, including patients with a wide range of problems as well as healthy volunteers. Each signal record contained four to eight signals, the first of which was always an ECG signal. The remaining signals could be any of a variety of simultaneously recorded physiologic signals that might be useful for robust beat detection. The signals were digitized at rates between 120 and 1000 Hz; in any given record, however, all signals were sampled at the same, fixed frequency. Table 3 shows the general statistics of the four data sets used for the challenge. The signal acronyms are: blood pressure (BP), arterial line (ART), pulmonary arterial pressure (PAP), and respiration (Resp).

Table 2. Data set signal type distribution.

Data set	N	BP	ART	PAP	Resp	EEG
Training	100	100	0	0	100	100
Phase I	100	14	75	70	73	14
Phase II	200	23	137	126	182	22
Phase III	300	37	194	177	163	35

The data sets for phases I, II, and III were kept hidden from all the participants. Performance of the challenge entries on these hidden test sets determined their rankings and thus the winners of the Challenge. The test sets were not available for study by participants, in order to avoid the possibility that entries could be optimized for high performance on the test data, thereby giving results that would be less predictive of performance on unknown data.

The training set was significantly different from the test sets. It was intended mainly to give the challenge organizers a way to verify that submitted entries were working as their authors intended, as well as to give participants an opportunity to see some of the problems that their entries could face in the challenge. Suggestions on how participants could extend their data set with additional PhysioNet data was posted on the Challenge website. The performance of challenge entries on the training set did not contribute in any way to their scores and ranks in the Challenge.

3. Scoring Criteria

In order to score each entry, we compared the annotations produced by the participants' code with a set of reference annotations that reflected the consensus of several expert annotators. The comparison was performed using the beat-by-beat algorithm defined by the ANSI/AAMI EC38 and EC57 standards, as implemented by the 'bxb' and 'sumstats' tools from the WFDB Software Package [2].

Each entry's output was evaluated on four performance statistics. The *gross sensitivity* (percentage of actual QRS complexes that the entry detected as such) and *gross positive predictivity* (percentage of the entry's annotations that corresponded to actual QRS complexes) were computed across the entire database, with every event having equal weight. We also computed the *average sensitivity* and *average positive predictivity* by assigning equal weight to each individual record. The overall score for the entry was the average of these four values.

Each entry was allowed to take a maximum of 40 seconds to evaluate any given record, and an average of at most 36 seconds per record. If the program took too long to complete, it was stopped at that point and scored based on the annotations it had already written.

3.1. Scoring Software

The software infrastructure required to run the Challenge is shown in Figure 2. Participants were asked to submit their entries in the form of a 'zip' or 'tar' archive that included everything needed to compile and run their program on a GNU/Linux system, together with the complete set of annotations that they expected their program to produce for the records in the public training set. This format allowed us to test and score entries completely automatically, and provide feedback to the participants in a matter

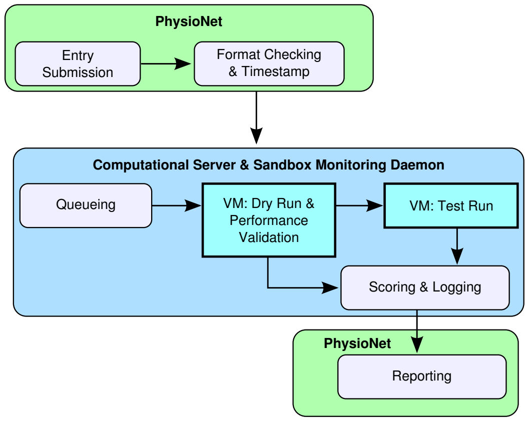

Figure 2. A schematic diagram of the software infrastructure used for evaluating Challenge entries.

of hours.

Each time an entry was uploaded to the PhysioNet server, it was then transferred to a virtual "sandbox" system. An identical copy of the sandbox was created for each entry. The scoring system would then unpack the archive, run its 'setup' script to compile it, and run its 'next' script to analyze each of the records in the training set. If the program could not be compiled, or did not produce the same annotations as the submitter expected, evaluation stopped at this point, and the error messages were sent back to the submitter.

Once an entry was successfully compiled and verified to be working correctly, the scoring system then proceeded to compute the annotations on the test set. The annotation files were collected, scored by 'bxb' and 'sumstats' as described above, and the final scores sent back to the submitter. If any errors occurred during this portion of the evaluation, they would be ignored; we did not allow the program to report back any information about the test set, apart from the final, aggregate scores.

In addition, the submitter could choose to designate an entry as a "dry run" by including a file named 'DRYRUN' in the archive; in this case, the entry would be tested on the training set, but not on the test set, and would not count against the user's limit of 5 entries per Challenge phase.

The sandbox consisted of a dual-core, 2.6 GHz AMD64 CPU running Debian GNU/Linux 7, with 2 GB of memory and 1 GB of virtual disk space for the program to use. In addition to the standard Debian packages, the sandbox included a variety of compilers, libraries, and utilties, including the WFDB Software Package (version 10.5.22), GNU Octave (version 3.6.2), and OpenJDK (version 7u55-2.4.7). This system was hosted using KVM on a computational server with an 8-core Opteron CPU and 32 GB of RAM; we allowed the server to run at most two entries simultaneously, in order to ensure that each entry would receive its fair share of memory and processor time.

We also provided a "Live DVD" image, containing all of the same software as the sandbox system, which competitors could download and run on their own machines in order to test their entries before submitting them.

4. Results

A total of 1,332 entries from 60 teams were processed during the challenge period, yielding a total of 317 scored entries. The median response time, from the moment the user submitted an entry to PhysioNet, to the moment their scores were reported back to PhysioNet, was 28 minutes in Phase I, 64 minutes in Phase II, and 100 minutes in Phase III. Thus, most entries had a response time of around 3 minutes per record throughout all three stages of the challenge. Table 3 shows the top five participants for each phase along with their average scores [3–11]. A scatter plot of the scores for all phases is shown in Figure 3.

Table 3. Official rankings. The scores for the sample entry (gqrs) are shown for comparison.

Phase I		Phase II		Phase III	
93.2	Vollmer	86.2	De Cooman	87.9	Johnson
89.2	Pangerc	86.0	Vollmer	86.7	Soo-Kng
88.9	Johannesen	85.9	Pangerc	86.6	De Cooman
88.9	Ding	85.0	Plešinger	86.4	Gierałtowski
88.7	Soo-Kng	84.6	Johnson	86.2	Vollmer
89.8	*gqrs*	85.7	*gqrs*	84.5	*gqrs*

5. Discussion

A record number of competitors and entries were scored throughout this year's Challenge. User feedback was critical during the initial phases and in identifying issues with the new scoring environment. An important concern raised was that the training data set was too easy or not representative of the entire test sets (see Table 3). The challenge FAQ was updated to suggest ways on how to augment the training set (such as using the MGH/MF Database available on PhysioNet [12]), which helped some of the competitors. Most of the top entries used pulsatile information, which led to a small advantage of up to 5% over the sample C entry that only used a single ECG signal.

Challenge participants used a variety of programming languages and libraries to implement their entries. The majority of entries were written in Octave, using the WFDB Toolbox [13], but several were written in C, C++, or Java. A few entries used Octave for their main program logic, but also used C or C++ libraries to speed up certain functions. A summary of the programming languages used is shown in Table 4.

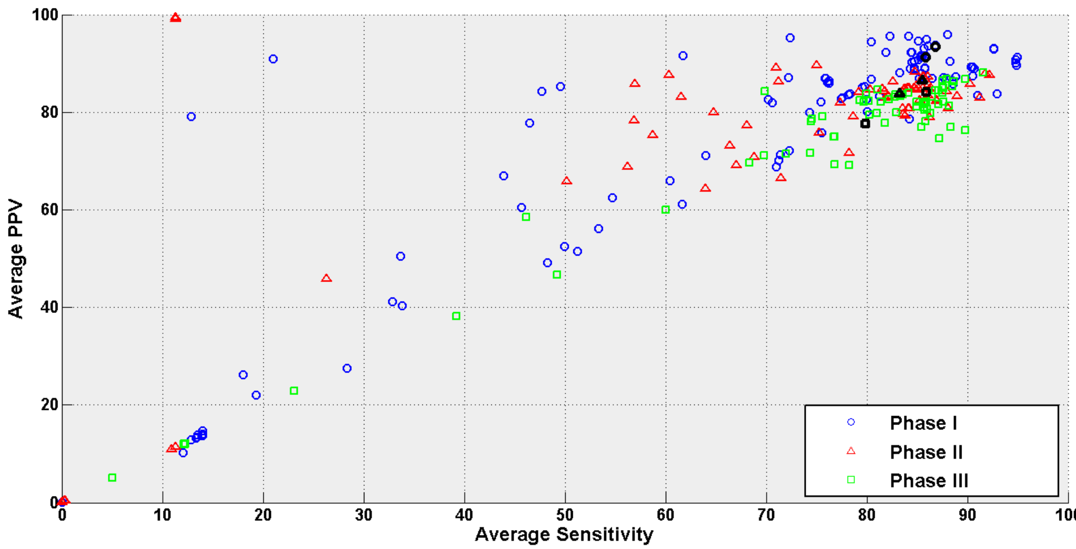

Figure 3. Performance for the 317 entries scored throughout the Challenge. The sample entries in C and Octave are shown in black.

Table 4. Programming environments used by successful entries.

Language	Entries	Teams
Octave, WFDB Toolbox	191	28
Octave, WFDB Toolbox, C/C++	23	4
Octave, C++	24	2
C	34	7
C++	8	1
Java	15	1

Acknowledgments

This work was supported by NIH/NIGMS grant R01-GM104987.

References

[1] Octave community. GNU Octave 3.6.2, 2014. URL `www.gnu.org/software/octave/`.

[2] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation 2000;101(23):E215–E220.

[3] Vollmer M. Robust detection of heart beats using dynamic thresholds and moving windows, 2014.

[4] Pangerc U, Jager F. Robust detection of heart beats in multimodal data using integer multiplier digital filters and morphological algorithms, 2014.

[5] De Cooman T, Goovaerts G, Varon C, Widjaja D, Van Huffel S. Heart beat detection in multimodal data using signal recognition and beat location estimation, 2014.

[6] Johnson AEW, Behar J, Oster J, Clifford G. R peak estimation using multimodal lead switching, 2014.

[7] Teo SK, Yang B, Huang D, Monterola C, Su Y. Robust detection of heart beats using ECG, BP and EEG signals, 2014.

[8] Gierałtowski JJ, Ciuchciński K, Grzegorczyk I, Kośna K, Soliński M, Podziemski P. Heart rate variability discovery: Algorithm for detection of heart rate from noisy, multimodal recordings, 2014.

[9] Johannesen L, Vicente J, Scully C, Galeotti L, Strauss D. Robust algorithm to locate heart beats from multiple physiological waveforms, 2014.

[10] Plešinger F, Jurčo J, Halamek J, Jurak P. Robust multichannel QRS detection, 2014.

[11] Ding Q, Bai Y, Erol Y, Salas-Boni R, Zhang X, Hu X. Multimodal information fusion for robust heart beat detection, 2014.

[12] Welch J, Ford P, Teplick R, Rubsamen R. The Massachusetts General Hospital–Marquette Foundation hemodynamic and electrocardiographic database – comprehensive collection of critical care waveforms. Clinical Monitoring 1991;7(1):96–97.

[13] Silva I, Moody G. An open-source toolbox for analysing and processing PhysioNet databases in MATLAB and Octave. Journal of Open Research Software 2014;In Press.

Address for correspondence:

Ikaro Silva
MIT Room E25-505A, Cambridge, MA 02139 USA.
ikaro@mit.edu