2015

Heartbeat location assistance for electrocardiograms

Sarah Bass
Eastern Washington University

Heartbeat Location Assistance for Electrocardiograms

A Thesis

Presented To

Eastern Washington University

Cheney, Washington

In Partial Fulfillment of the Requirements

for the Degree

Master of Science

in Computer Science

By

Sarah Bass

Spring 2015

THESIS OF SARAH BASS APPROVED BY

_____ DATE_____

DAN LI, GRADUATE STUDY COMMITTEE

_____ DATE_____

PAUL SCHIMPF, GRADUATE STUDY COMMITTEE

MASTER'S THESIS

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Eastern Washington University, I agree that the JFK Library shall make copies freely available for inspection. I further agree that copying of this project in whole or in part is allowable only for scholarly purposes. It is understood, however, that any copying or publication of this thesis for commercial purposes, or for financial gain, shall not be allowed without my written permission.

Signature_____

Date_____

Abstract

The electrocardiogram (ECG) is the main source of heartbeat analysis throughout the medical community, due to the distinctive appearance of the QRS complex at the time of each beat. There are other signals that also exhibit distinctive patterns at the time of each heartbeat; however, the ECG is still the most prevalently used. Analyzing the ECG alone can be problematic because ECG data can be noisy. The noise often makes it appear as if the heartbeat is missing or that multiple beats occurred in quick succession. This research will analyze the association between a variety of signals and the ECG, and use those associations to predict heartbeat location with greater accuracy than just analyzing the ECG alone. After analyzing all 10 minutes of an ECG signal along with other signals such as Blood Pressure (BP) and Stroke Volume (SV), the goals of this research are: (1) Preprocess the data so that signals show clear shapes and noise is minimized; (2) Generate templates from the provided signals that correspond to a clear QRS complex in the ECG to provide information for what needs to occur in other signals in order for a beat to be annotated; (3) Compare these templates to test records and annotate beats where the templates are approximately matched. By achieving the stated goals this research will aid the medical community in determining the optimal type of template to use, and the number and type of signals required to locate beats with accuracy.

Table of Contents

List of Figures

1 INTRODUCTION

The Electrocardiogram (ECG) signal is the primary resource for locating heartbeats and determining if beats are normal and healthy, or abnormal. Abnormal beats can further be classified to indicate specific heart conditions. However, ECG signals may be unreadable or unavailable, and miss important information due to electrical noise from nearby machines or patient activity [1]. In an inpatient setting, measurement of the ECG is often only one of many signals being continuously collected from patients. Other signals such as blood pressure (BP) are highly correlated to heart rate, and thus can be used to aid in the mining of heartbeat location [2].

The goal of the PhysioNet 2014 Challenge was to process several signals together from an individual patient and use their combined information to more accurately identify where heartbeats occur, even when the ECG is noisy and unreadable [3]. This is a time series data mining problem. Humans looking at time series can detect overarching shapes and repetitive patterns (here heartbeats) while ignoring inconsequential bumps or deviances [4], but getting computers to achieve these tasks is quite challenging. In order to complete the goal of identifying heartbeats using multiple signals, the following must be accomplished:

- Determine which signals contain information which can be used to assist in locating heartbeats.

- Reduce the dimensionality of each time series so that important information can quickly be found.
- Determine from the training set what shapes indicate heartbeats from pertinent signals.
- Utilize a similarity measure to determine at which points the test set signals are close enough to the expected shapes to be marked as possible beat locations.
- Combine results from multiple signals to locate heartbeats as thoroughly and as accurately as possible.

This research investigates the ability of signals including blood pressure, respiration, stroke volume, electroencephalogram, electromyogram, and electrooculogram to detect heartbeats in tandem with the ECG, producing time stamps of beat occurrence with minimal error, within a reasonable time frame.

2 BACKGROUND

The training data provided by PhysioNet includes 100 records digitized at 250 samples per second with four or more of the following signals: ECG, Blood Pressure (BP), Respiration, Electromyogram (EMG), Electrooculogram (EOG), Electroencephalogram (EEG), and Stroke Volume (SV). All records contain an ECG recording, some of which include noisy sections. Each record is 10 minutes in length from patients with normal and abnormal heart rhythms [3]. The test set, also provided by PhysioNet, includes 100 records digitized at varying frequencies. This set includes the same types of signals as the training set as well as some additional signals such as Pulmonary Arterial Pressure (PAP) and Central Venous Pressure (CVP), which are closely related to BP. Below is a description of each of these signals and their expected shapes.

2.1 Signals

ECG - Electrocardiogram: measures electrical activity from the heart through electrodes placed on the limbs and/or chest [5]. Within an ECG signal, the most prominent shape is called a **QRS complex**. Each of the most common waves within an ECG signal has a name from a letter of the alphabet, as shown in Figure 2-1. The QRS complex is a visualization of the contraction of the left and right ventricles of the heart. The Q wave is a dip in the signal just before the high R wave, and then followed by another dip – the S wave. QRS complexes usually last around 100 ms, but can last up to 250 ms. The **RR Interval** is also a commonly looked at measure in the ECG, as it measures the amount of time between consecutive R waves. R waves usually have the

highest amplitude within the signal, so this interval can give a clear indication of average heart rate.

With all of these signals, shape varies greatly between people. Sometimes there are not large R waves, but instead very deep Q waves. Sometimes the T wave is larger than the S wave, and as mentioned, the length of the whole QRS complex can vary greatly. The cause of these variations is usually arrhythmia [5].

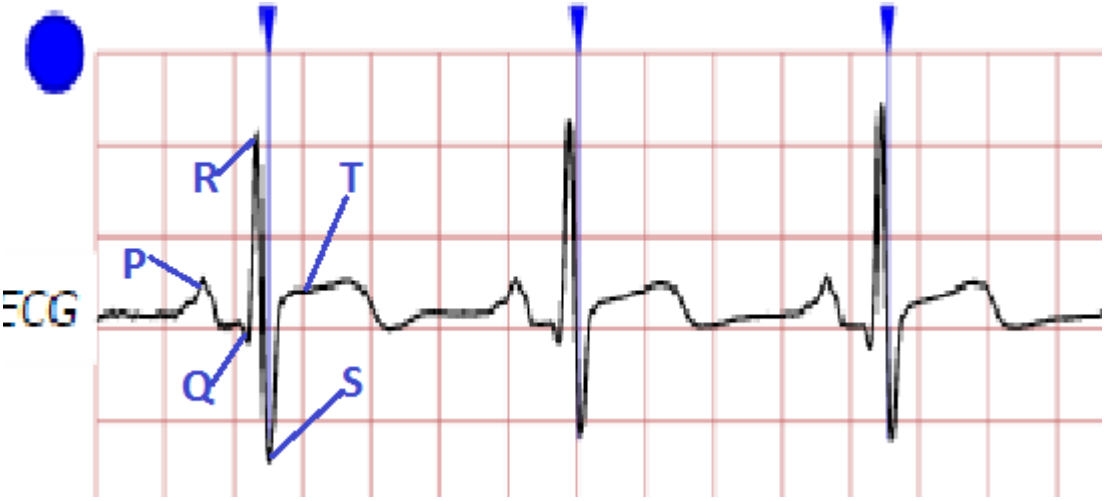


Figure 2-1: ECG signal with labeled waves

BP - Blood Pressure: measures the pressure exerted on artery walls by moving blood within those arteries [6]. This pressure varies with the contraction and release of the heart, hence the shape shown in Figure 2-2.

There are two additional types of blood pressure measured in the test set. **Central Venous Pressure (CVP)** and **Pulmonary Arterial Pressure (PAP)** measure pressure at different parts of the heart [6]. They are measured through catheters

inserted near the heart. Figure 2-2 shows BP, PAP and CVP from one person and the variation in shape between the different signals.

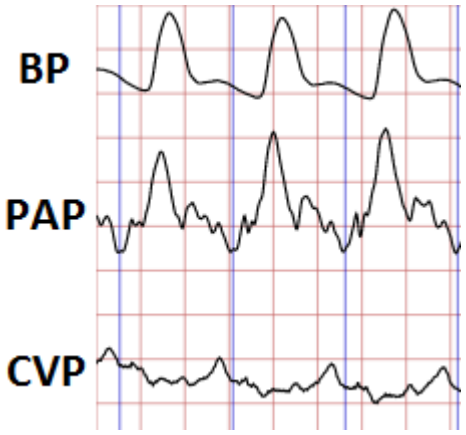


Figure 2-2: Blood Pressure, Pulmonary Arterial Pressure and Central Venous Pressure from a single patient, demonstrating the variation in shape of the signals, though all three measure pressure around the heart.

EOG – Electrooculogram: measures the electrical charge of the eye by placing electrodes on either side of the eye [7]. The charge between the front and back of the eye is opposite, so EOGs are good at measuring eye movement. Because blood flows throughout the body, there are small impulses that occur with each heartbeat, and these impulses show up in the EOG, thus indicating heartbeats.

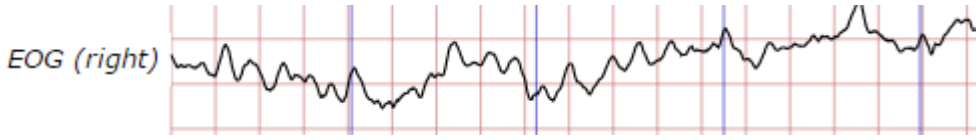


Figure 2-3: EOG signal from a patient's right eye. The blue lines indicate where each heartbeat occurs, and at each of these locations is a clear peak in the EOG.

SV – Stroke Volume: measures the amount of blood leaving the heart [8]. Figure 2-4 shows a clear arch shape with every heartbeat.

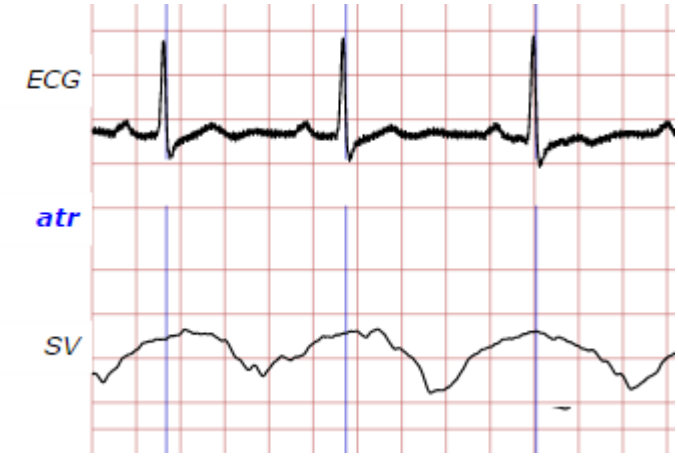


Figure 2-4: ECG and Stroke Volume correlation

Other signals contained in the training and test sets, but which were not utilized in this research due to lack of clear correlation with heartbeats, included Respiration, Electromyogram (EMG), and Electroencephalogram (EEG).

2.2 Irregularities

As mentioned briefly in the description of the ECG signal, shapes of signals vary greatly between patients. Some studies have shown that the complexity of signals is reduced with aging and disease [9]. Because patient data such as age and disease was not provided with this data set, they were not factored in to help determine expected shape. In addition to inconsistency of shape between patients, there was some variance within an individual patient's signals. The amplitude of a signal varies depending on respiration [11]. As shown in Figure 2-5, the shape can also vary suddenly within a patient's signal if an abnormal heartbeat occurs.

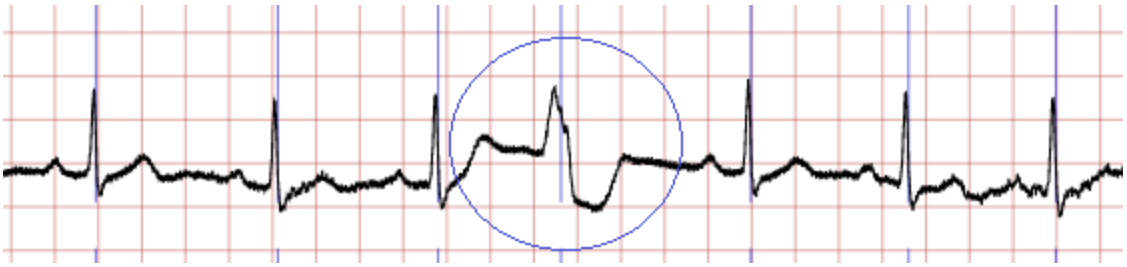


Figure 2-5: A circled abnormal heartbeat, surrounded by QRS complexes that have very consistent shapes.

Another reason for irregularity in shape even within an individual patient's signals is noise. Noise can be caused by movement of electrodes, perhaps disconnecting from the skin [10]. It can also be caused by power line noise and muscle contractions [11].

The most information leeching irregularity within all signals is the flat line. When something interferes completely with the signal being read, a horizontal line appears. There are quite a few cases when this happens in one signal and not another, so many times beats can still be found from alternative signals, but this makes for many special cases.

3 RELATED WORK

There are many ways to approach the problem of detecting heartbeats or heartbeat artifacts in various time series. Signal processing is very popular, especially using the ECG. This section will discuss the sample solution provided by PhysioNet, which uses a QRS detector (utilizing signal processing) to locate beats in an ECG signal. It will also give a brief introduction to sequential pattern mining as well as the idea of template matching, both of which were utilized in this research.

3.1 Sample Solution

Many researchers use one or several types of filters to transform a signal into a shape that is more easily readable by a computer. The sample solution provided by PhysioNet uses a low pass filter to smooth out tiny waves within the signal so that the important waves of the QRS are more clear [12]. This low pass filter is at its core averaging the last five samples and replacing the original sample value with this average value [13]. Figure 3-1 shows an example of the difference in signal shape after applying a low pass filter.

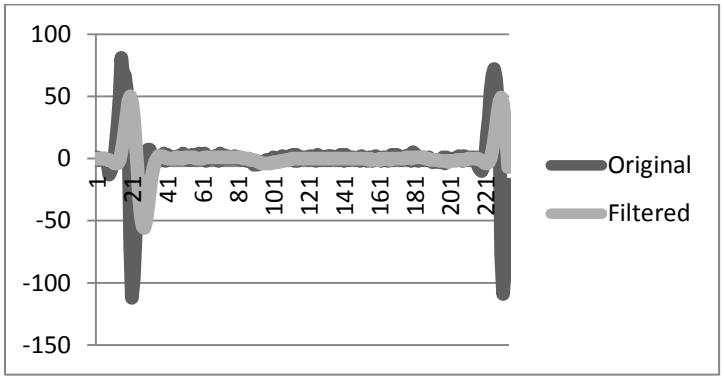


Figure 3-1: The original signal in black and the smoothed signal in grey after applying a low pass filter.

The sample solution also uses a curve length transform which takes the integral of the low pass filtered curve, creating very clear pulses throughout the QRS complexes, and stifling the P and T waves. The program then maintains an adjustable threshold for the average value of the curve length transformed signal. Following this, the complex is annotated using a series of small adjustments. This program has very high accuracy in detecting beats from the ECG as it still works well even when the shape of the beat changes. It was used in this research as a baseline off which to build.

3.2 Sequential Pattern Mining

Sequential pattern mining is a subfield of frequent pattern mining, which is the process of identifying patterns that occur frequently in a data set [14]. Sequential patterns bring in the involvement of time, which implies order. As in frequent pattern mining, a support threshold must be defined, which decides what is considered frequent. If a pattern occurs twice in thirty seconds it may be considered frequent in some cases, but if it occurs twice in 30 days, probably not.

Time	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Value	1	2	3	2	5	7	9	5	1	3	5	3	5

Table 3-1: Time Series Example

Another definition needed in sequential pattern mining is the gap constraint. Gap constraints are a limit on how far away elements can be from each other and still be considered an occurrence of a pattern. Take the time series in Table 3-1 for example. Assume the minimum support is 3. If the gap constraint were 5, then there would be three occurrences of (1,5) – at times [1,5], [9,11], and [9,13]. Thus (1,5) would be considered frequent. However, if the gap constraint were 2, then there would only

be one occurrence of (1,5) at [9,11], meaning the pattern would not be considered frequent. Gap constraints are a way to prune the number of frequent patterns generated.

One example of a sequential pattern mining algorithm applied to time series is called Multiple Width Approximate Sequential Patterns (MWASP) [15]. MWASP utilizes the idea of approximate pattern mining, which does not require an exact match in order to count as an occurrence. This idea is appealing for medical mining due to the variance of wave shapes between people as well as within an individual person's signals, along with the disturbances of noise. MWASP uses variable widths, a variation on gap constraints, to allow for small differences between pattern occurrences.

There was one submission to the PhysioNet competition which used a similar algorithm, called ConSGapMiner, to mine for sequential patterns within the signals provided [16]. They limited their patterns by length - they wanted longer patterns - as well as finding patterns with a larger number of distinct symbols. Their results were not impressive, at about 65%, but the idea was very interesting.

3.3 Template Matching

Template matching is very similar to the process that humans use to find matching patterns in data [17]. Template matching involves comparing new data with a stored version of what is being sought. If key features line up, then it is an occurrence of a match.

Alignment of the template with the signal is very important in template matching. One way this can be achieved is through a simple sliding window, a first

come first serve type approach. The window is the size of the template, and starting at the beginning of the test signal each point is lined up with the template and compared. Then the template is moved over one sample number and the process begins again. Table 3-2 shows an example of this process.

Time	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Value	1	2	3	2	5	**7**	**9**	**5**	1	3	5	3	5
Template	7	9	5										
		7	9	5									
			7	9	5								
				7	9	5							
					7	9	5						
						7	**9**	**5**					

Table 3-2: Sliding window template example – the window size here is 3, with a template of [7, 9, 5]. The template is aligned once its window starts at time 6.

Another approach involves finding a succession of important events that occur in the template, such as a steep rise followed by a steep drop [17]. Then look for the same events in the test signal, and when found, align those with the same events in the template. Then compare the signal and template sample to sample. This approach requires preprocessing of the test signal to locate those important events, which adds processing time.

The other important aspect of template matching is finding a similarity measure. This determines how closely the test signal corresponds with the template. The simplest measure is Manhattan distance – subtract template from test signal point by point, and if they are closely correlated, the result will be close to zero. Template matching has been used for classifying whole time series as well as for detecting QRS complexes in ECG signals [4].

4 METHODS USED

The methods used in this thesis are discussed below. The overall idea was to identify frequently occurring patterns in the training set under the assumption that repetitive beats would have similar values, and then search for those patterns in the test set. Initially, a sequential pattern mining algorithm was used to mine these frequent patterns. In order to apply sequential pattern mining algorithms to time series data, the numerical data had to be discretized into a smaller number of possible values – this is where Symbolic Aggregate approXimation (SAX) was used [18]. Then the CM-SPADE algorithm was then applied to find the most frequent sequential patterns [19].

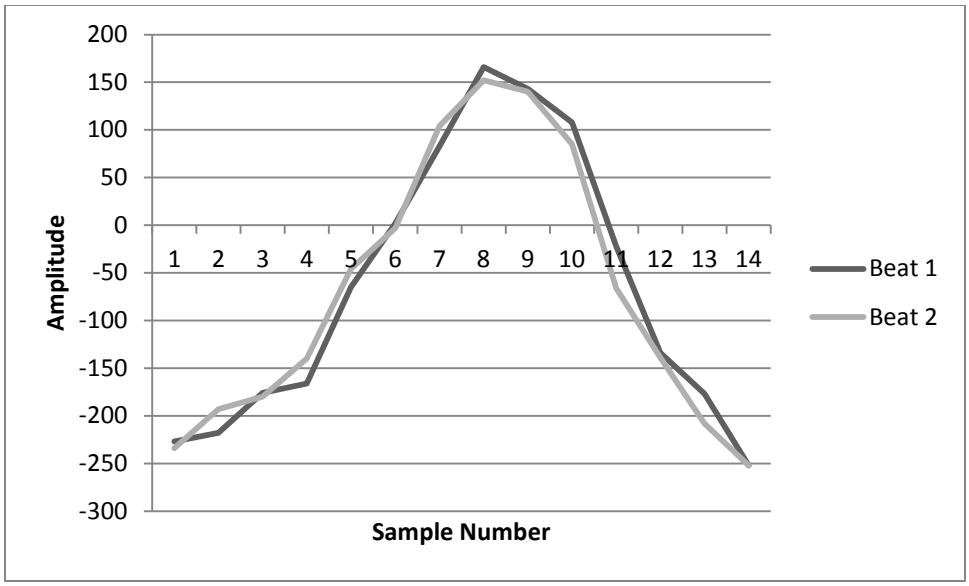
Following experimentation with frequent sequential patterns, the solution evolved into a template matching solution. By dividing the training set signals into individual heartbeats using the provided beat annotation times, a template library was created for each correlated signal type – ECG, BP, Stroke Volume, and EOG. These templates were used to verify whether or not beats proposed by the sample QRS detector provided by PhysioNet [3] were indeed legitimate beats or not. The test set record was divided into beats by the QRS detector, and then those beats were compared with the template libraries to see if they were close enough to any of the templates to be considered legitimate.

Because of the slowness of the initial template solution, k-means clustering was then used to reduce the number of templates to a more manageable number. The templates' makeup was then further refined from a simple derivative of the original

signal into a statistical representation of that portion of the signal – reducing the size of the templates from around 200 numbers to just two numbers. This not only improved timing but also improved accuracy. A similarity measure was needed in all cases to match frequent sequential patterns and templates to the test set. Sections are as follows: (1) Data Preprocessing, (2) Sequential Pattern Mining, (3) Templates.

4.1 Data Preprocessing

The main challenge in locating heartbeats is that values vary from person to person as well as from heartbeat to heartbeat within an individual person's signals. Figure 4-1 shows two R waves (the highest wave from the QRS complex) from the first two beats of a training record. Note the slight difference in shape as well as the vast number of different values found for the amplitude. Values range from -252 up to 166, which make for a possible 418 different values. These are just the values from a portion of two beats, 1/800th of the entire signal. Mining frequent sequential patterns from such a large number of possible values would not return many meaningful results, and if any were returned, they would be extremely short in length.



Beat 1	-227	-218	-176	-166	-65	2	83	166	143	108	-22	-134	-177
Beat 2	-234	-193	-180	-140	-46	-3	104	152	140	85	-66	-139	-208

Figure 4-1: Two R waves from the first two beats of the ECG Signal from training record 100 along with a table of the actual values for the amplitude of each sample.

Thus the first step when mining patterns is to make sure the alphabet size (or number of possible values) is manageable. Symbolic Aggregate approXimation (SAX) is an algorithm that solves this problem. SAX is capable of both reducing dimensionality as well as discretizing the data [18]. However, for the purposes of this paper the dimensionality reduction feature was not used.

SAX takes as input the signal to be processed as well as the desired size of the output signal and the desired size of the output alphabet. An overview of the algorithm is as follows:

1) Normalize the signal to have a mean of zero and standard deviation of 1
2) Reduce dimensionality using Piecewise Aggregate Approximation (PAA)
3) Discretize data using Gaussian distribution for specified alphabet size

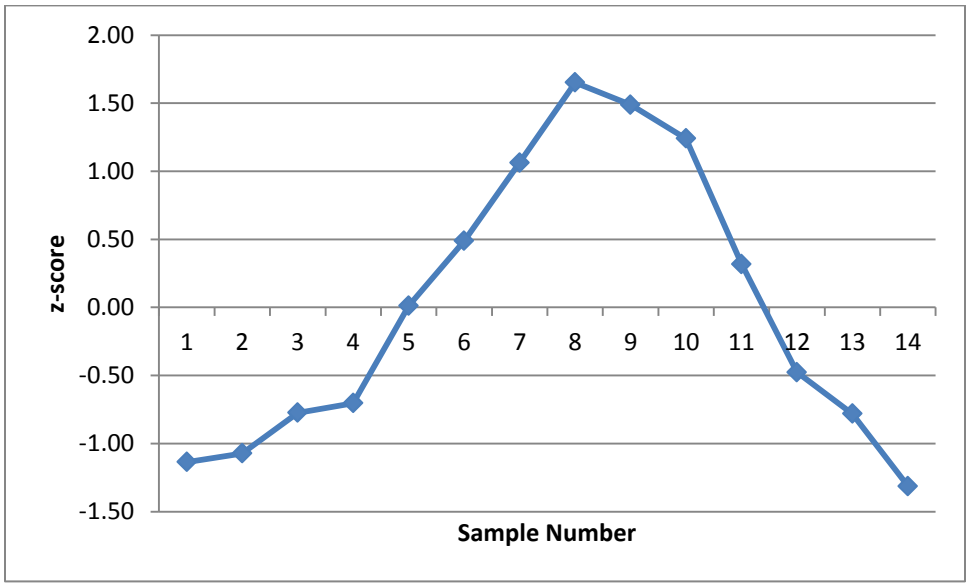
The first step taken by SAX is to normalize the signal around zero. This can be achieved through z-score normalization. Equation 4-1 shows how to calculate the normalized value from the initial value, the mean of the signal, as well as the standard deviation of the signal.

$$z = \frac{x - \mu}{\sigma}$$

Equation 4-1

μ = mean
σ = standard deviation

The wave from Figure 4-1 Beat 1 is shown normalized in Figure 4-2. The mean of the beat is approximately -66.8. The standard deviation is approximately 141.



-1.14	-1.07	-0.77	-0.70	0.01	0.49	1.06	1.65	1.49	1.24	0.32	-0.48	-0.78	-1.31

Figure 4-2: Z score normalized values for Beat 1 from Figure 4-1

In this case the second step of SAX is skipped because we keep our original signal size, so PAA is not discussed here.

Normalized time series exhibit a Gaussian distribution. Thus for step 3, SAX can use evenly distributed areas under the Gaussian curve to form discretized bins of equal size. Below is a table of values which break the curve into various equal sized areas.

BreakPts\Alpha Size	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
b1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
b2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
b3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
b4			0.84	0.43	0.18	0	-0.14	-0.25
b5				0.97	0.57	0.32	0.14	0
b6					1.07	0.67	0.43	0.25
b7						1.15	0.76	0.52
b8							1.22	0.84
b9								1.28

Figure 4-3: Breakpoint values for Gaussian distribution at various alphabet sizes (3-10)

For example, if the desired output alphabet size were three, any scores below -0.43 would be 0, between -0.43 and 0.43 would be 1, and above 0.43 would be 2. Figure 4-4 shows Beat 1 transformed into this three letter alphabet, which would be the final output from the SAX algorithm.

Beat 1	1	1	1	1	2	3	3	3	3	3	2	1	1	1

Figure 4-4: Final output for Beat 1 from SAX algorithm

Other preprocessing involved resampling test signals which were not originally sampled at the 250 samples/sec frequency. A method provided by PhysioNet accomplished this with minimal time overhead, though some noise was added into the resampled signal [3]. There were also signals provided which were completely flat through the duration of the ten minutes, and these had to be eliminated. There were also short periods within valid signals which flat lined. These were found by looking at

the average slope of the part of the signal being examined, and if it was zero, that signal's information was not utilized.

4.2 Sequential Pattern Mining

Now that the sequence has been discretized into a fairly small alphabet size, a sequential pattern mining algorithm can be applied to locate frequently occurring sequential patterns. Because actual beats in an ECG signal are several hundred samples in length, below is a simplified example of a short signal reduced to a 3 letter alphabet.

Example Signal	1	2	3	2	1	2	3	1	1	1	2	3	1	1

Figure 4-5: Signal with a three letter alphabet with a length of 14

Visually it is apparent that there exist patterns that occur several times such as 1 2 3, but how to find these in an automated way? There is an algorithm called CM-SPADE which takes as input such a signal and finds frequent sequential patterns [19]. A frequent sequential pattern is a subsequence of the initial signal that appears more than a certain number of times [14]. The user can define a threshold to say how many occurrences are needed before a pattern is considered frequent. This threshold is called the minimum support. The difference between a frequent pattern and a frequent sequential pattern is that time is also a factor. Items occurring in the pattern must occur at a time later than the items found before it.

CM-SPADE is a variant of SPADE, another sequential pattern mining algorithm [19]. SPADE uses a vertical representation of the data [20]. A vertical data representation uses the item as a hash key, and stores the times at which that item

occurs. For the example signal in Figure 4-5, the database would look as shown in Figure 4-6.

1

Seq ID	Time Stamp
1	0
1	4
1	7
1	8
1	9
1	12
1	13

2

Seq ID	Time Stamp
1	1
1	3
1	5
1	11

3

Seq ID	Time Stamp
1	2
1	6
1	10

Figure 4-6: Example signal from Figure 4-5 in vertical format. The underlined numbers above each table are the items found in the signal, and the tables contain the times at which said items occur. These tables are also called id lists. Because we have only one sequence in our example database, there is only one sequence id.

The first step for Sequential PAttern Discovery using Equivalence classes (SPADE) is to discover frequent single items within the sequence. Suppose the minimum support threshold is 3, then all three items in our example signal are frequent. Thus $F_1 = \{1,2,3\}$. SPADE finds this by simply counting the number of occurrences in the table. The next step is finding two-item sequences. The options from our dataset include 1->2, 2->3, 1->3, 2->1, 3->1, 3->2, as well as 1->1, 2->2, 3->3 – where *a*->*b* means that *a* comes before *b* in the signal. In order to find whether these two-item sequences are frequent, we can do a temporal join from the single item id lists.

A temporal join involves counting the support of associations that occur after a certain time. Take 1->2 for example. First look at the 1 id list - there is an occurrence at time 0. Then look at the 2 list and look for occurrences after time 0 - there are 4. The

second occurrence of 1 is at time 4, and there are two occurrences of 2 after time 4. Continuing on with this logic, the total support of 1->2 is 9. To optimize this process, 2->1 and 1->2 are calculated at the same time so that minimal scanning of id lists is needed.

Instead of reading from the id lists every time a new pattern is examined, CM-SPADE creates a new data structure called a Co-occurrence Map (CMAP) to store support information in a space-efficient and easily accessible form [19]. CMAP is a hash table structure that associates an item with a list of items that succeed that item a certain number of times (minimum support).

As shown in the SPADE algorithm's temporal joins, when looking for an extension of a pattern to determine if it is frequent or not, the most important thing is whether the extension occurs after that initial pattern in enough instances throughout the sequence [20]. CMAP stores the items that occur frequently after each item in the alphabet, thus making it much faster to see if a new pattern may be frequent. This is formally defined as "succeeding" an item *k*. An item *m* succeeds *k* if in the full sequence there exists an occurrence of *m* after *k*. Thus CMAP(*k*) would contain (*m*, *n*, *p*) if *m*, *n*, and *p* succeeded *k* in the original sequence more than the minimum support. So if we know that *k* is frequent and we want to see if *k*->*m* is frequent, we can first check and see if m is in CMAP(*k*). If it is not, then *k*->*m* can be pruned without scanning the id lists. This approach is similar to dynamic programming, storing values that are used more than once so as to not have to calculate them multiple times.

In order to insert CMAP pruning into SPADE (making it CM-SPADE), during the enumeration step when joining two patterns that have a common prefix, e.g. *PA* and *PB*, before performing the expensive join, check to see if *B* is in the CMAP(*A*), and if not then *PAB* may be pruned. Also CMAP(*B*) needs to be checked for *A*, and if it is present, then perform the join. If not, the entire thing can be pruned, saving a lot of time.

Once a frequent sequential pattern was found from the training set beats, that pattern was then input into a program that compared the test set's records with the pattern using a sliding window, as demonstrated in Table 3-2. When there was a match, a beat was annotated.

4.3 Templates

Template matching involves comparing test data to a standard (a template) to see whether or not it is similar [17]. Two types of templates were experimented with in this research, full beat templates and statistical templates.

4.3.1 Full Beat Templates

In this project, templates were taken first from the blood pressure signal at the time of each beat in the training set. Blood pressure was chosen as a starting point due to its clear correlation with heartbeats as well as being seemingly less susceptible to noise. Blood pressure signals come in a variety of shapes, as shown in Figure 4-7.

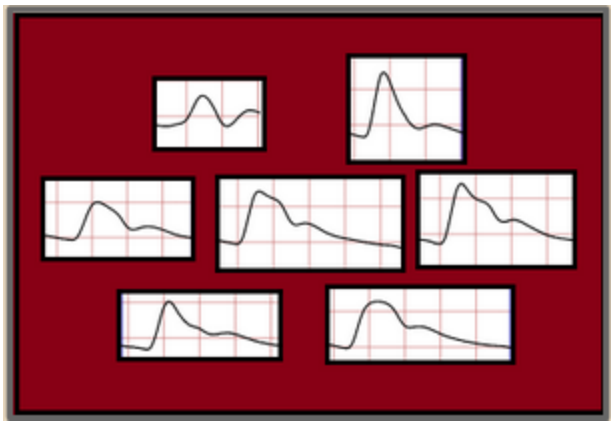


Figure 4-7: Examples of various shapes found in blood pressure signals

The initial templates directly took the values in the signal's derivative from the time between two annotated beat locations in the training set. Figure 4-8 shows one such template from the BP library of templates built from the training set.

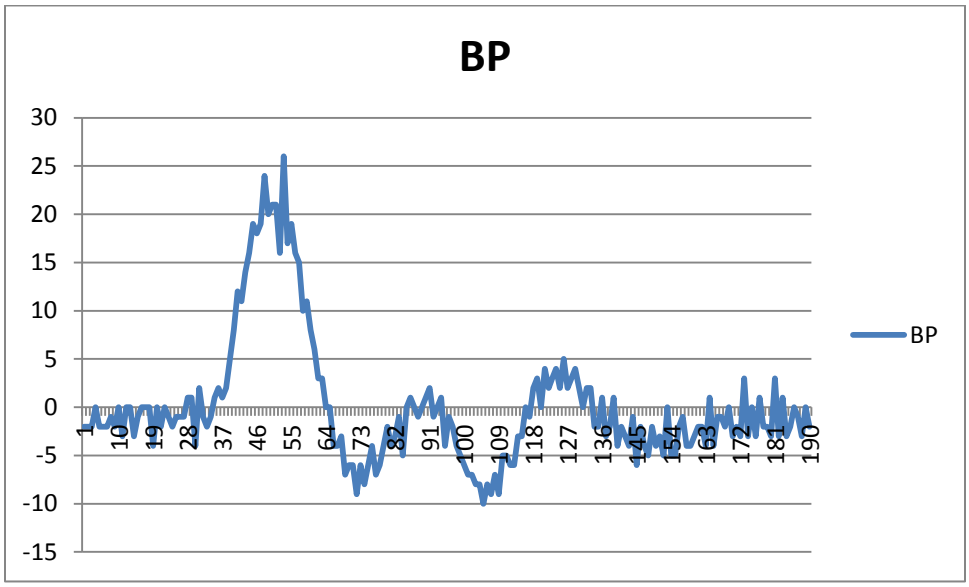


Figure 4-8: The first beat in a training record's BP signal- this template was made from the first beat of the annotation set to the second. The two beats were 190 samples apart.

Once a library of templates was created, the following succession of actions was completed:

1) Use the QRS detector from PhysioNet on the ECG signal to find beats in a test record.

2) Compare the templates in the library with each section of the blood pressure signal that corresponds with the location of the proposed beats from the QRS detector.

3) If any of the templates are within a certain distance of the proposed beat's blood pressure segment, then annotate that as a beat, otherwise skip that proposed beat.

Figures 4-9 and 4-10 below exemplify the challenge of aligning a template with an actual beat. The example compares a template to a blood pressure signal at a time where that record's ECG signal is very noisy and the QRS detector has proposed three beats in short succession. The goal is to weed out the two that are not actual beats by calculating the distance between the template and the signal using the proposed beats as starting locations for the comparisons. The template shape being compared matches the width and height of the BP signal fairly well, but placing the template at the second line clearly is much closer to the actual BP signal than placing it at the first or third line. Visually, this is easy to see. Automating this process requires choosing a distance measure.

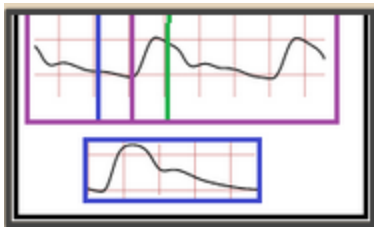


Figure 4-9: The template does not match the signal at the first line.

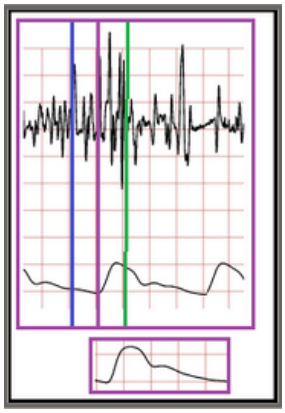


Figure 4-10: The template matches very closely to the signal at the second line, thus the second line is a good candidate for the heartbeat location.

For this project, initially Manhattan distance was chosen for its low computational cost. This is a simple distance measure that takes the absolute value of the difference between each dimension of the data. In this case, the difference was taken between the template and the test signal at each sample point within the length of the proposed beat.

$$d(R,T) = \sum_{i=1}^{n} |r_i - t_i| \qquad \textbf{Equation 4-2}$$

R is a proposed beat in a test record, *T* is a template, and *n* is the length of the longer of *R* and *T*.

Originally the idea was to compare the parts of the template and record that overlapped, but it was necessary to include very short templates in order to account for beats at the end of the record that were cut off. These short templates were being compared throughout and returning very small distances because of the relatively few

samples contained in the template, and thus noisy beats were still being marked as beats. One attempt at eliminating this problem was to only use templates under 20 samples in length if the beat encountered was the last proposed beat found by the QRS detector for that record. This resulted in fewer actual beats being found.

Cosine similarity was then applied because of its ability to normalize results based on length [14]. The formula is as follows:

$$sim(x,y) = x \cdot y / ||x|| * ||y|| \qquad \textbf{Equation 4-3}$$

In order to apply this formula to a time series, the dot product was taken between the template(x) and the test signal at the proposed beat location(y), and divided by the normalized length ||x|| and ||y||. These lengths were calculated as follows:

$$||x|| = \sqrt{(x_1^2 + x_2^2 + \ldots + x_n^2)} \qquad \textbf{Equation 4-4}$$

Here x_i is the value of each sample, and n is the length of the template. Cosine similarity returns values centered around 0.

The actual distance threshold chosen as a cutoff point for determining if a proposed beat is an actual beat or not appeared to be optimized in the range $-.000005 < x < .000005$.

The first solution that was developed as a baseline took all the templates generated from the training set without any clustering or refining of the data. This included 71720 Blood Pressure templates, 72413 ECG templates, 14869 Stroke Volume templates, and 29307 EOG templates. Experiments began with just using the blood pressure templates to verify beats proposed by the QRS detector. Later, ECG templates were used to verify QRS detector beats, and if they were not verified, those beat locations were compared with other signal templates (BP, EOG, SV) to see if they were valid. This is discussed in more detail in the Results section.

4.3.2 Clustered Templates

Clustering is a way of grouping data together without knowing specifically what the groups should look like ahead of time – this is called unsupervised learning [14]. The K-Means algorithm again requires a distance measure, and Manhattan distance was chosen as a simple, fast solution.

The *k* in K-Means represents how many clusters will be in the output. Due to the large processing time of using all full beat templates, the number of templates needed to be reduced to a more manageable size. *K* controls that desired smaller number of templates. For example, there were originally 71720 BP templates, and *k* was chosen to be 500, 100, and 20 respectively. Most records contained about 800 beats, so 500 was not much of a reduction in size. Twenty was a huge reduction, and 100 was a nice middle value. Generally, *k* is up to the user, and is another value that needed to be optimized. Also, due to clustering such a large number of templates, the

program was taking a very long time, so instead of clustering all templates at once, each training set record's beats were clustered separately.

The K-means algorithm is as follows:

1) Assign random centers for each cluster (resulting in *k* centers)

2) Find the distance between each template and the current centers and assign the template to the closest cluster.

3) Recalculate the centers of each cluster as the mean of all templates in that cluster.

4) Repeat steps 2 and 3 until no templates change clusters.

Figure 4-11 shows an example output of the K-Means clustering algorithm for a set of 11 points and a *k* of 3.

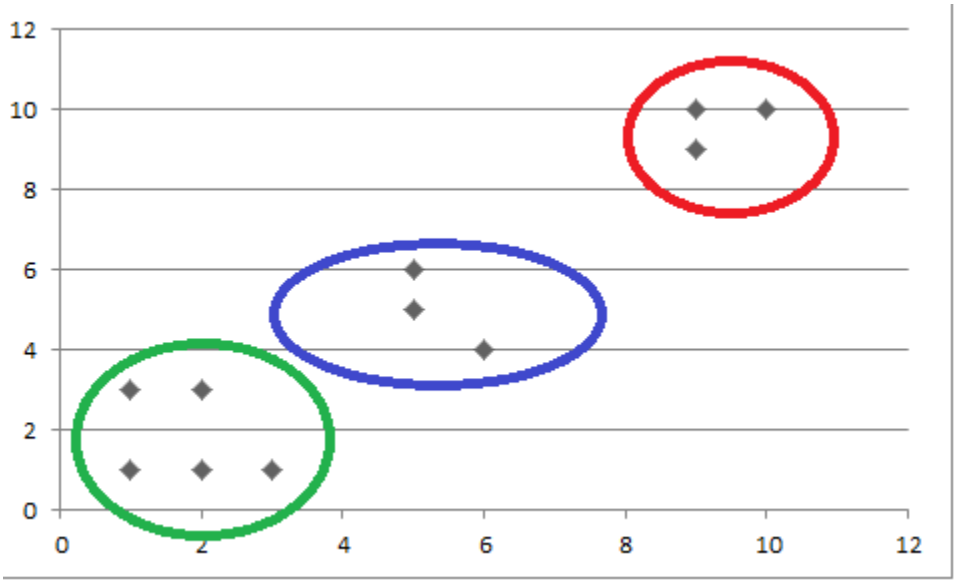
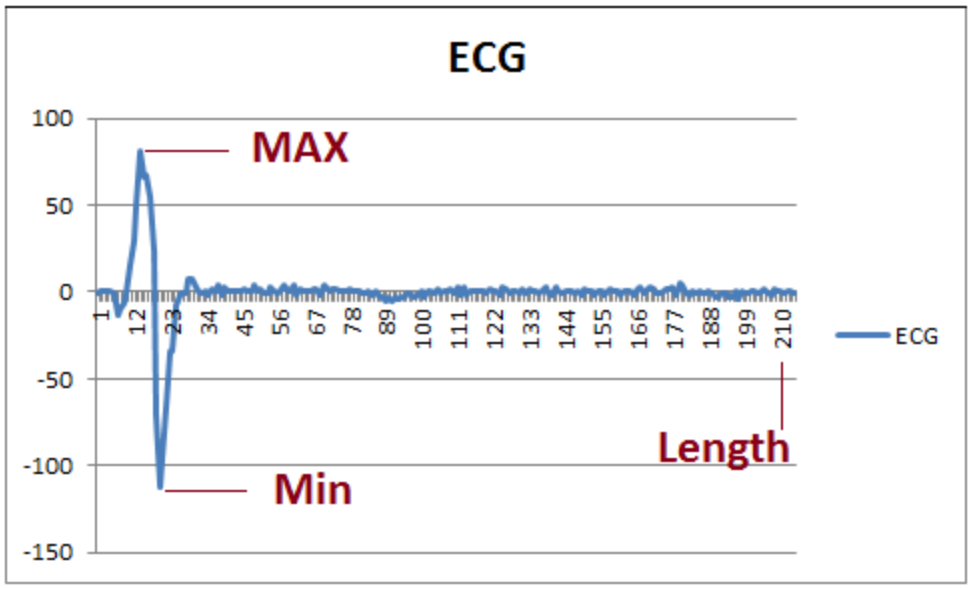


Figure 4-11: A possible K-Means clustering result, with *k*=3 for a set of two dimensional points.

Once K-means was applied to the templates with various *k* values, the library with the reduced number of templates was compared with the test set using cosine similarity. The results produced from these experiments were faster than the initial template solution. However, results were outperformed by the sample solution.

4.3.3 Statistical Templates

Since using templates of full beat values did not seem to produce optimal results, a statistical representation was substituted for the full beat representation. A statistical representation contains the important information from each beat in a succinct way, thus making processing time faster, and helping the computer to hone in on the most useful information about a beat without getting distracted by small deviances. The maximum value within the beat, the minimum value, the distance from the start of the beat to the max, the distance from the start to the min, as well as the length of the entire beat were used to represent the entire beat. These statistical values were then compared with the same statistical values calculated from the proposed beats from the test set, looking for those with a cosine similarity close to zero. Figure 4-12 shows an example of a statistical template from an ECG signal.



Max Value Time Stamp	Min Value Time Stamp	Length
12	20	213

Figure 4-12: A graph of a beat in an ECG signal, along with the values used for the statistical template. The location of the maximum and minimum are shown, as well as the length of the beat. The actual max and min values were found not to be useful.

Because each useful signal appeared to have some sort of arched shape within the beat, the maximum and minimum values were thought to provide the height of that arch, and the distance to the maximum and minimum would provide the position within the proposed beat where that arch occurred. The length of the entire beat was also important to avoid marking short proposed beats from long template values.

5 RESULTS

As mentioned in the Section 2, PhysioNet provided two data sets as part of this competition, a training set of 100 records and a test set of 100 records. PhysioNet also provided its own performance scoring system, called *bxb*, for each record, and an overall score called *sumstats*. This section will define the scoring system, show baseline results from the sample QRS detector for reference, and then go into the various results from the initial sequential pattern mining solution as well as the assortment of template solutions.

5.1 Performance Metrics

For each record, the values of importance include the total number of heartbeats, the total number of heartbeats marked by the proposed solution, the number of false positive beats – positions marked as beats by the proposed solution that were not actually heartbeats, as well as the number of false negatives – heartbeats that were missed by the proposed solution.

Record	Nn'	Vn'	Fn'	On'	Nv	Vv	Fv'	Ov'	No'	Vo'	Fo'	Q Se	Q +P	V Se	V +P	V FPR
testingset/1003	952	0	0	0	0	0	0	0	5	0	0	99.48	100.00	-	-	
testingset/1032	799	0	0	0	0	0	0	0	0	0	0	100.00	100.00	-	-	
Sum	1751	0	0	0	0	0	0	0	5	0	0					
Gross												99.72	100.00	-	-	
Average												99.74	100.00	-	-	

Total QRS complexes: 1756 Total VEBs: 0

Summary of results from 2 records

Figure 5-1: Output from *sumstats* – PhysioNet method that shows a summary of the statistics from each test record processed by the solution.

Figure 5-1 above shows the output from the *sumstats* method provided by PhysioNet. The import information is circled. Capital letters represent the actual annotated solution, and lower case letters represent the test solution that is being evaluated for accuracy. Table 5-2 shows what each symbol means.

Actual Solution Symbol	Test Solution Symbol	Meaning
N	n	Marked as a beat
O	o	Not marked as a beat

Column Title	Meaning
No	Number of beats missed by the test solution (False Negatives)
On	Number of beats added where no beats occurred by the test solution (False Positives)
Q Se	Sensitivity percentage – num of correctly marked beats/total number of correct beats * 100 = Nn/(Nn+No)*100
Q +P	Precision percentage – num of correctly marked beats/total number of beats marked * 100 = Nn/(Nn+On)*100
Total	For the competition, the rank of entries was an average of the four Q Se and Q +P values. In this case it would be: 96.85+96.45+96.46+94.05 = 383.81/4 = 95.95

Table 5-2: Shows the meanings of values appearing in Figure 5-1that are used throughout this section.

Overall, the goal of this thesis was focused on lowering the number of false positives (No) from the sample solution without too much of an increase in the number of false negatives (On). In other words, the attempt was to raise the precision without lowering sensitivity.

5.2 Sample Solution Performance

These results are for reference, since the goal was to improve upon the sample solution. As mentioned in Section 3.1, the sample solution is a QRS detector provided

by PhysioNet which locates beats solely based on information from each record's ECG signal. There is no training done based on the training set, thus these results are based solely on the test set.

The sample solution's overall score was 86.62%. The precision values (84%) were lower than the sensitivity values (88%), so the idea was to find a way to minimize the number of false positives, and thereby increase the total score.

5.3 Sequential Pattern Mining Solution Performance

This solution used SAX to put records into symbolic notation with a 20 letter alphabet, and then used the CM-SPADE algorithm from the SPMF library to mine sequential patterns in the ECG training set. One such pattern was chosen to use for testing, and when it was found a beat was marked. Figure 5-3 shows the test results from the basic solution on the training set compared with results from a similar solution published by CinC [16].

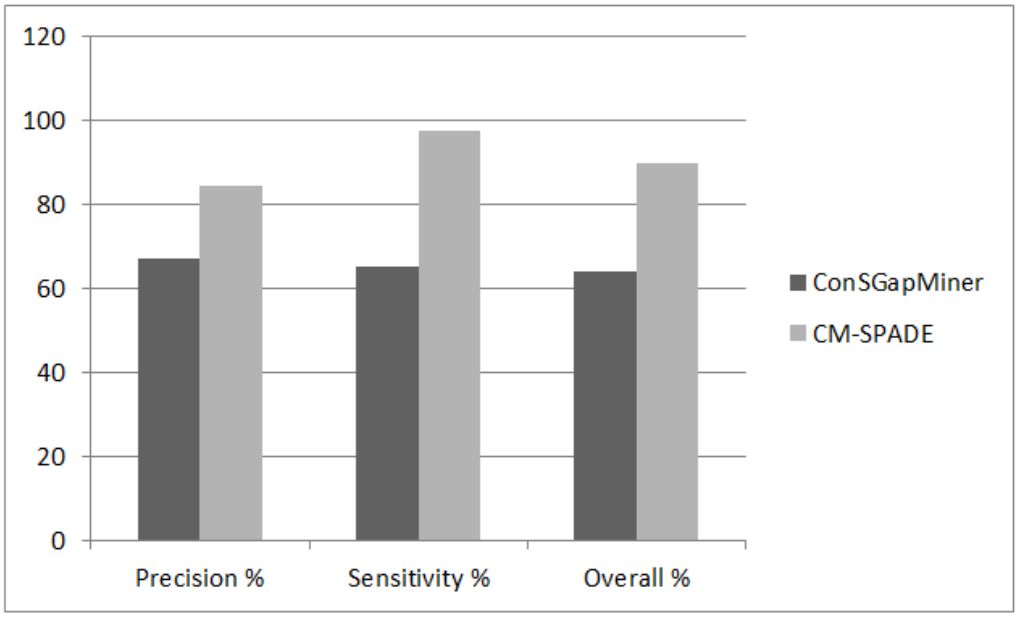


Figure 5-3: Comparison of solution published by CinC using ConSGapMiner with this research using CM-SPADE

Figure 5-3 shows that the CM-SPADE solution had higher precision, sensitivity, and total than the ConSGapMiner solution. In particular, the sensitivity was quite high, at 97%. However, the overall accuracy was significantly lower than the more competitive solutions in the competition. For example, the sample solution applied to the training set achieved a total of 99%. This research thus switched to a different approach using template matching.

5.4 Full Beat Template Solution Performance

These solutions involved templates taken directly from the signals of the training set with no reduction in the number of samples from each beat, or from the total number of beats in the training set.

The first solution attempted used actual beat locations from the training set to create templates for all beats from individual signals (ECG and Blood Pressure). Testing then used the sample solution to find proposed beat locations from ECG, and compared the blood pressure signal or the ECG signal from between proposed beats to all the respective templates. If any were within an optimized threshold of Manhattan distance, then the beat was marked. For the ECG this threshold was a distance of 900, and for blood pressure it was 1000, as these numbers were found to provide the best results. Figure 5-4 shows the resulting number of false positives and false negatives from using BP and ECG templates as compared with the sample solution.

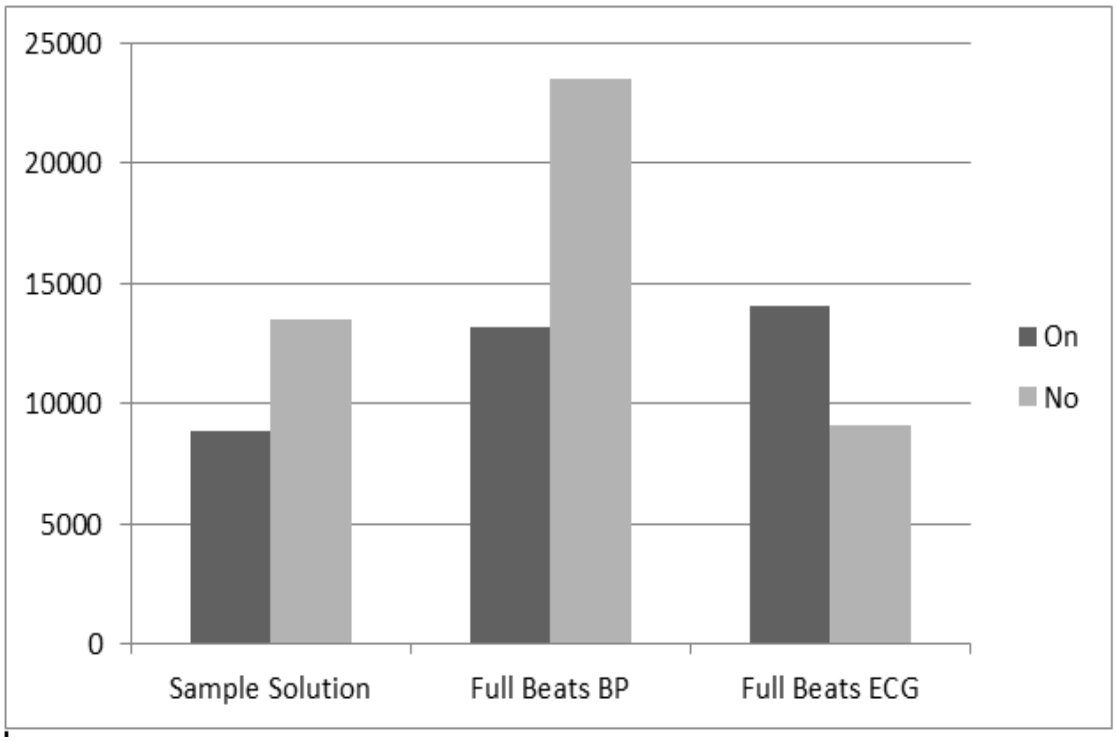


Figure 5-4: Extra beats and missing beats using Full Beat templates taken from BP signal compared with Sample Solution.

The dark columns show the number of beats that were added by the test solution that were not real beats. The light columns show the number of missing beats from the test solution annotations. Blood pressure templates presented with very poor results, decreasing both the sensitivity and the precision of the sample solution by about 10% overall. The ECG templates completed the goal of raising the precision, but did so at a steep cost to the sensitivity, and the resulting performance was approximately equal to that of the Sample Solution. Because ECG performed so much better than BP, ECG templates were used in future testing of this solution.

The next step was to try a more sophisticated distance measure. Cosine similarity takes into account the differing lengths of each template and proposed beat. The other change was to use SAX to discretize the data. An alphabet size of 20 was

used. Figure 5-5 shows results using cosine similarity and SAX on ECG templates. Using cosine similarity and SAX surprisingly decreased the accuracy of this solution, as it caused a small increase in the number of missing beats.

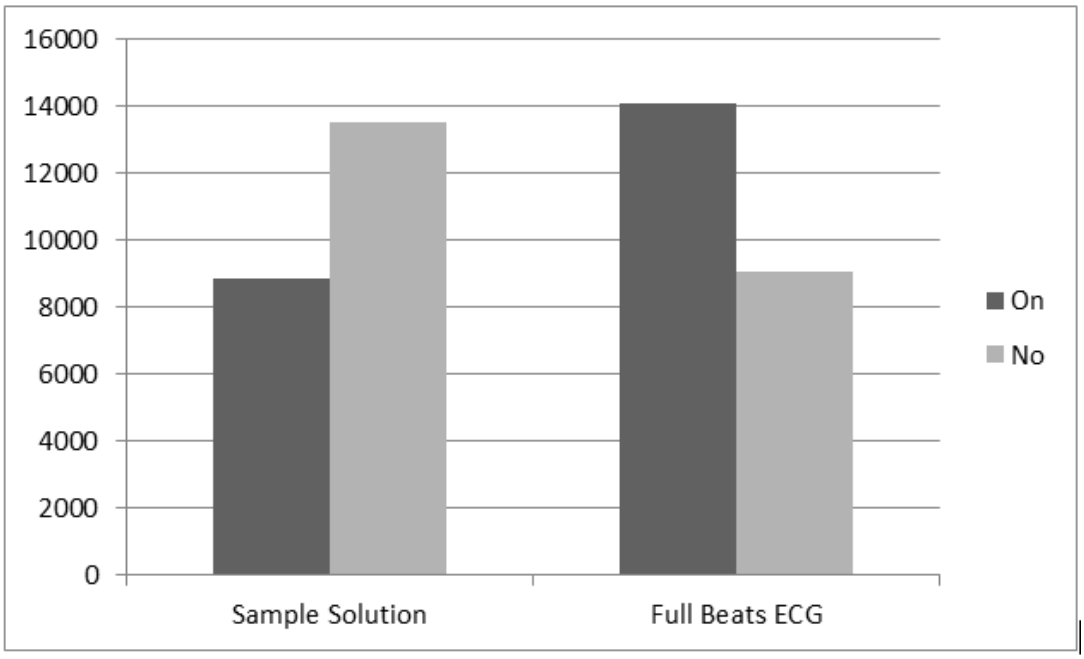


Figure 5-5: Results using templates from ECG only, comparing with cosine similarity instead of Manhattan distance, as well as preprocessing with SAX. Threshold for cosine similarity optimized around +/- .000005. SAX used an alphabet size of 20.

5.5 K-Means Template Solution Performance

The next experiment involved trying to increase performance by combining templates into clusters using the k-means algorithm. By having a smaller number of templates taken from the averages of closely related templates, the hope was that both sensitivity and precision would improve, as well as a reduction in the time it took to process each test record.

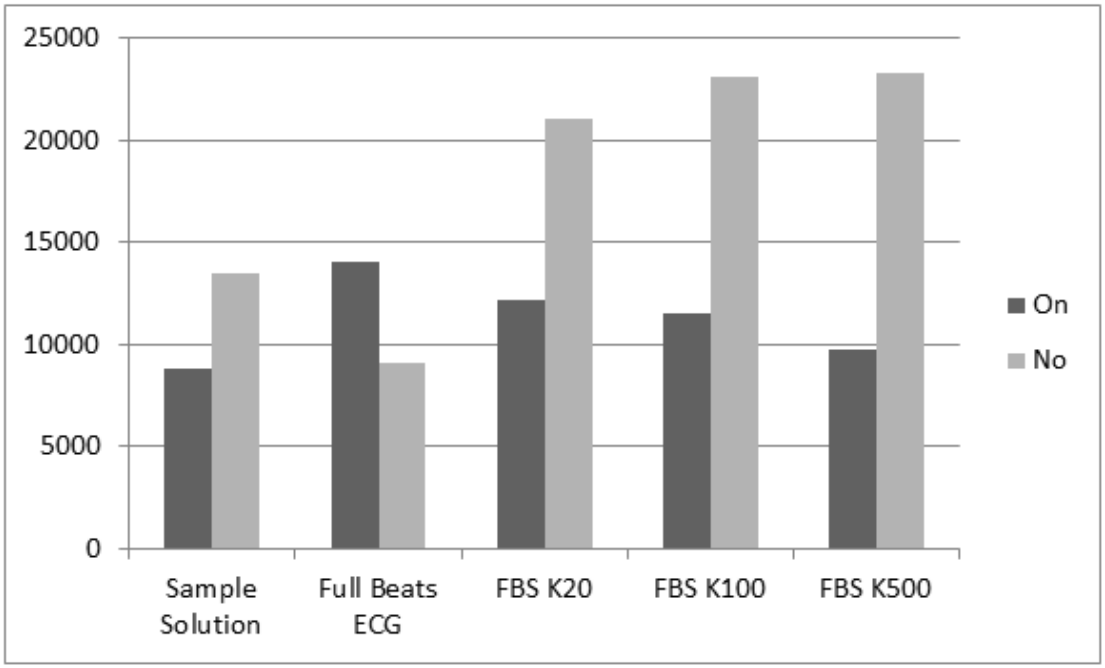


Figure 5-6: Results of clustering using *k* values of 20, 100, and 500 on the full beat templates preprocessed using SAX.

Figure 5-6 shows that as the number of clusters/templates decreases, the precision decreases. Sensitivity in all cases was prohibitively low, though interestingly there was a slight increase in sensitivity as the number of templates decreased. The results did not increase enough to compare with the sample solution though. Figure 5-7 shows the overall performance of all experiments from the full beat and k-means sections.

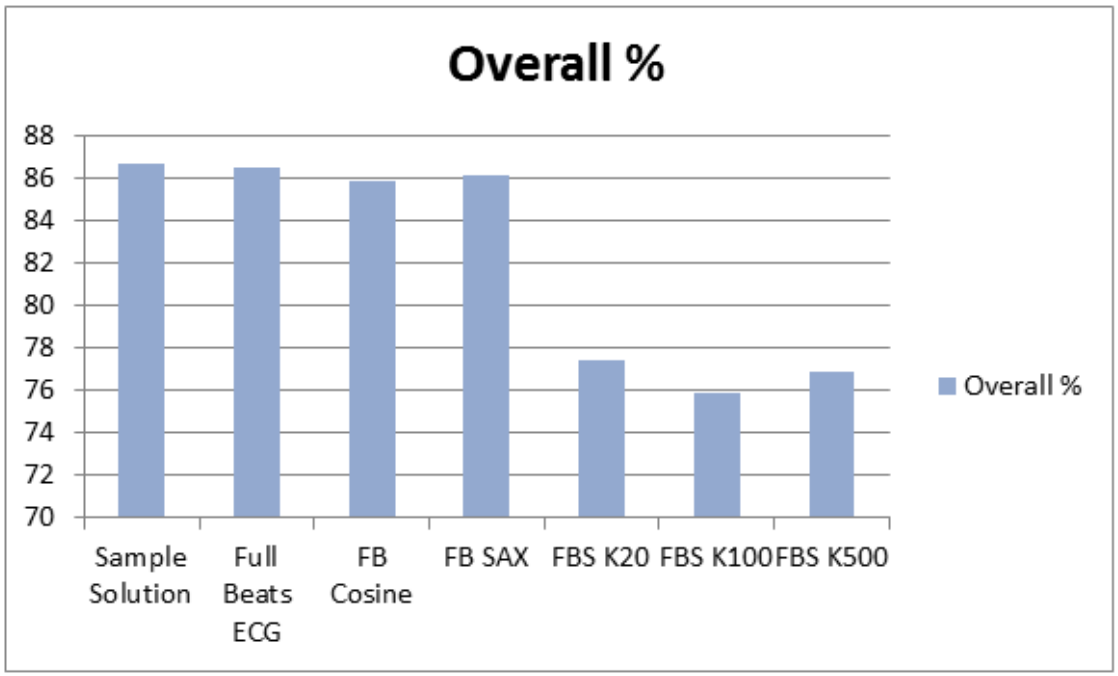


Figure 5-7: Overall performance of all solutions thus far

Notice the large drop in accuracy for the k-means solutions. However, these solutions were much faster at processing the data, as shown in Figure 5-8.

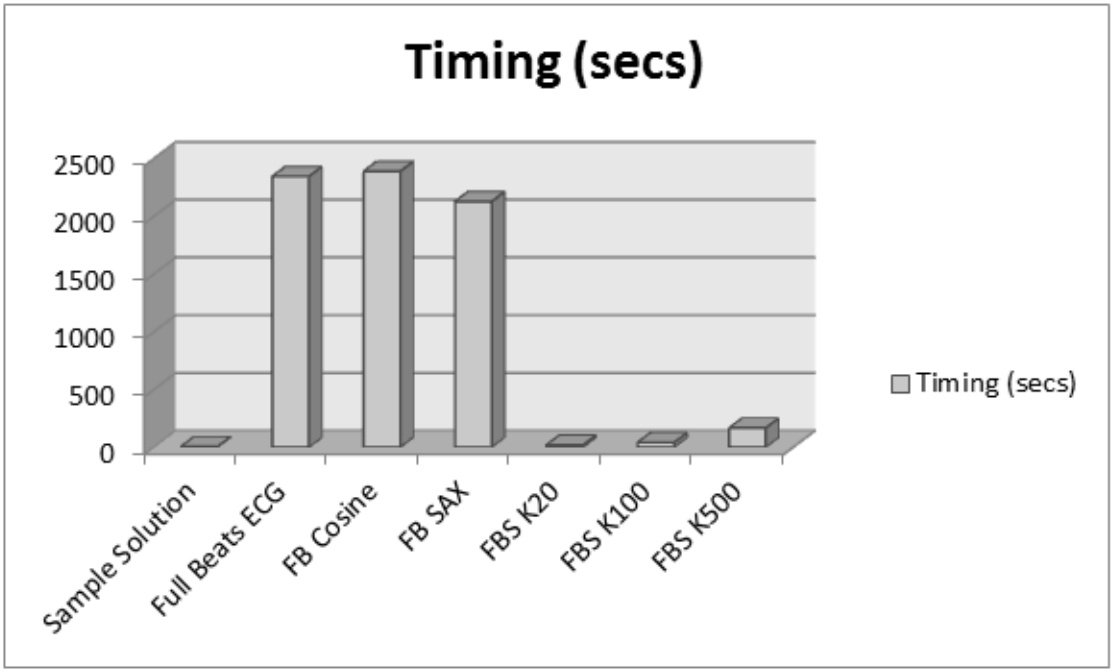


Figure 5-8: Timing of solutions in seconds

Overall this solution did not achieve the goal of increasing the precision of the sample solution without decreasing the sensitivity, thus a different experiment was attempted using statistical information about each beat as templates.

5.6 Statistical Template Solution Performance

The statistical template solution uses the same idea as full beat templates, but instead of using all the samples from the beat, a small number of specific statistical measures were stored. These measures included the following: the distance from the beginning of the beat to the maximum value in the beat, the distance from the beginning of the beat to the minimum value, the actual maximum and minimum value, as well as the total length of the beat.

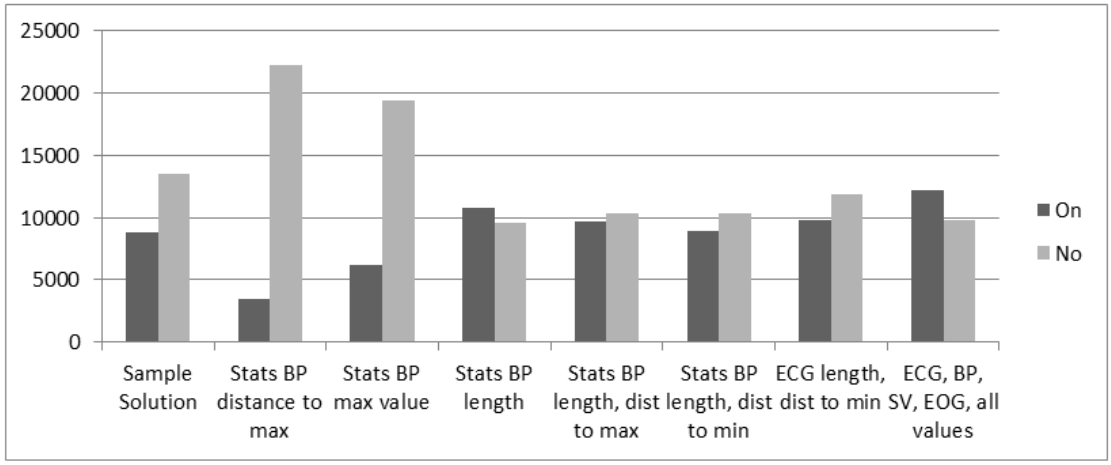


Figure 5-9: Extra and missing beat values using various statistical measures and signals

Figure 5-9 shows the number of incorrect beats marked during various experiments. Using blood pressure alone had the best results, most likely because it is not as subject to noise as ECG signals. Notice that the distance to the maximum value showed a very large decrease in the number of extra beats added, though the tradeoff

with sensitivity was also large. The precision for that solution was 94%. By using the length of the template in conjunction with the distance to the maximum value, the overall performance was slightly improved from the sample solution. The distance to the minimum value along with the length of the template showed the best results, with a decrease in missing beats of about 4000 and only a 50 extra beat gain.

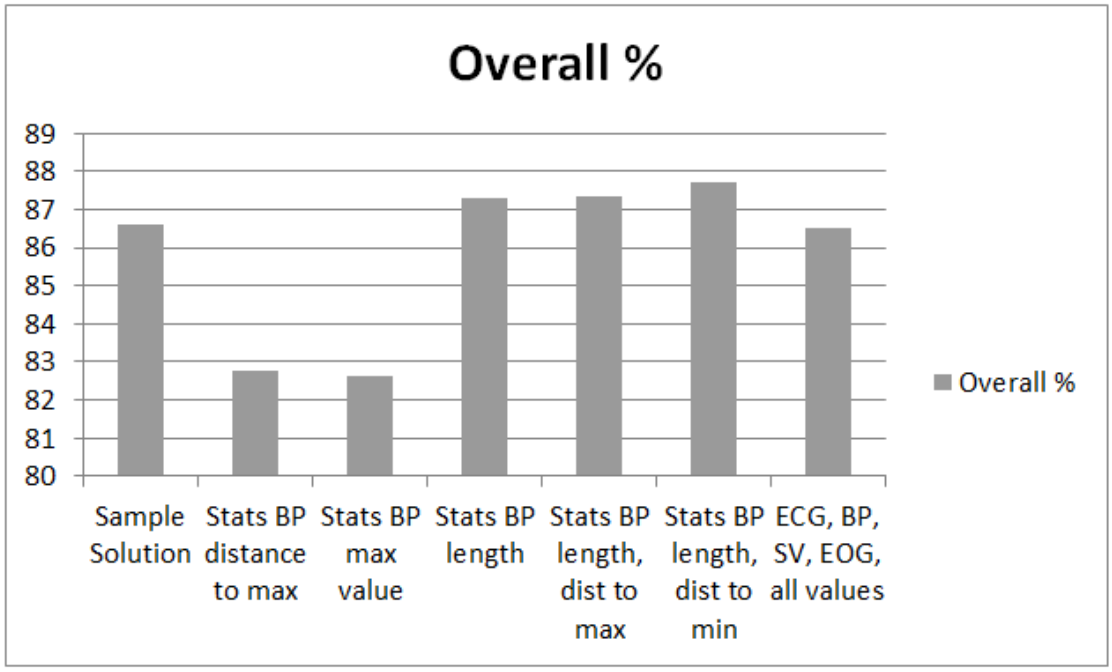


Figure 5-10: Overall performance of statistical template experiments

The overall performance of all statistical template solutions is shown in Figure 5-10. Blood pressure templates using length and distance to the minimum value of the beat show the highest performance at 87.73%. As the timings for these solutions averaged around 10 seconds, this was the best solution found out of all those presented in this thesis in terms of performance and processing time.

6 CONCLUSION & FUTURE WORK

Locating heartbeats using multiple time series data is a complex problem. Some background knowledge of the signals themselves is necessary to decide which signals are useful and in what way they may be useful. The problem presented by the PhysioNet 2014 Challenge was to use as many different signals as possible to locate heartbeats as accurately as possible.

The ECG signal is typically the signal used to locate heartbeats. At each heartbeat location, a low wave (Q), then a very high wave (R), followed by another low wave (S) makes the distinctive shape of the QRS complex that theoretically should be simple to find. However, due to variation in this shape depending on age, illness, and outside influences such as noise from electrode movement, detecting a true QRS complex becomes a challenge. The noise often features very high wave forms which can be mistaken for R waves. People with abnormal heart rhythms may not have a high R wave, but rather a very low Q wave, or even a sort of double beat for each beat, with a high P wave preceding the QRS complex, making it difficult for a computer to tell if the P or the R wave is the actual beat. These variants must either be accounted for in the QRS detector, or made up for with information from other signals that are frequently monitored in conjunction with the ECG.

Signals such as blood pressure and stroke volume closely correlate with the ECG. With a short time delay, the blood pressure signal will have a distorted arch type figure following each heartbeat. Stroke volume produces almost a perfect arch, with

the top of the arch corresponding to the heartbeat location. Electrooculograms contain small upshots at each heartbeat location. Signals such as respiration, electromyograms, and electroencephalograms do not contain any obvious heartbeat information and are thus discarded. Blood pressure in particular appeared in many training records and seems to be very commonly tracked along with the ECG and thus was the main focus of testing for this thesis.

Medical signals do not form nice smooth shapes. Instead very bumpy and noisy lines appear. Because of this, some preprocessing was needed to remove the extra noise and leave the basic shape. To do this, an algorithm called SAX was used, which takes in a time series and discretizes the data into an alphabet size chosen by the user. It was found however that the loss of information, even using a large alphabet size, was too great for this discretization to be useful.

Template matching has been used as a way to detect artifacts within time series as well as finding similarity between whole time series for many years [17]. A template is a digital copy of an artifact, or the key features of said artifact. In this thesis various types of templates were explored, including using full beats as templates as well as using just statistical information, such as the distance from the beginning of the beat to its highest point. It was found that statistical information used as templates is much faster as well as more accurate at locating heartbeats than using full beats as templates.

Attempts were made to reduce the number of full beat templates to improve accuracy and timing. The K-means algorithm was applied using a variety of *k* values to

cluster the very large number of templates created from the training set into a smaller number. This was very effective for reducing time, but the loss in accuracy proved this solution to be ineffective.

While full beat template matching can be effective at increasing the precision of a QRS detector, the results of this thesis show that the loss in sensitivity is too great in general for the solution to be considered useful. Statistical templates on the other hand are smaller in size, require very little preprocessing of data, and provide a modest increase in accuracy.

Future work that may improve the results presented here could start by reducing the dimensionality of the data through SAX before any comparison is completed. The sequential pattern mining solution could be expanded upon to develop a solution that searches for multiple patterns within the test set, as well as creating a more efficient algorithm with which to mine the training set.

The problem of locating heartbeats within multiple time series provides the researcher with opportunities to apply many different types of algorithms, and the options are basically limitless. While the results of this thesis are modest, any improvement is worthwhile, especially when it only adds negligible processing time.

BIBLIOGRAPHY

[1] A. Pachauri and M. Bhuyan, "ABP peak detection using energy analysis technique," in *Multimedia, Signal Processing and Communication Technologies (IMPACT)*, 2011, pp. 36–39.

[2] R. Palaniappan, C. Navin Gupta, C. K. Luk, and S.-M. Krishnan, "Multi-parameter detection of ectopic heart beats," in *Biomedical Circuits and Systems, 2004 IEEE International Workshop on*, 2004, pp. S2–4–1–4.

[3] (2014, Jan 29). *Robust Detection of Heart Beats in Multimodal Data: The PhysioNet/Computing in Cardiology Challenge 2014* [Online]. Available: http://www.physionet.org/challenge/2014/

[4] Philippe Esling and Carlos Agon. 2012. Time-series data mining. *ACM Comput. Surv.* 45, 1, Article 12 (December 2012), 34 pages.

[5] (2015, May 15). J. Goette, *Review of Various Biomedical Signals* [Online]. Available: https://www.microlab.ti.bfh.ch/master/biomed/biosig/public/BioSig/theDocs/1_Intro/article_A5.pdf

[6] K. Muralidhar, "Central Venous Pressure and Pulmonary Capillary Wedge Pressure Monitoring" in *Indian J. Anaesth*, 2002; 46 (4): 298-303.

[7] M. Brown, M. Marmor, Vaegan, E. Zrenner, M. Brigell, M. Bach, "ISCEV Standard for Clinical Electrooculography (EOG)" in *Doc Ophthalmol.*, 2006; 113(3): 205-12.

[8] (2015, May 15). *Stroke Volume and Cardiac Input* [Online]. Available: http://www.hsc.csu.edu.au/pdhpe/core2/focus2/focus1/4007/2-1-4/fac2_1_4_2.htm

[9] D. Kaplan, M. Furman, S. Pincus, S. Ryan, L. Lipsitz, A. Goldberger, "Aging and the Complexity of Cardiovascular Dynamics", in *Biophysical Journal*, 1991; 59(4): 945-979.

[10] A. Ghaffari, S. Atyabi, M. Mollakazemi, A. Jalali, M. Aghaamoo, D. Mafi, "A Noise Robust Method for Recognizing the Heart Beats in Multimodal Data", in *Computing in Cardiology,* 2014: 549-552

[11] G. Friesen, T. Jannett, M. Jadalla, S. Yates, S. Quint, H. Nagle, "A Comparison of the Noise Sensitivity of Nine QRS Detection Algorithms", in *IEEE Transactions on Biomedical Engineering*, 1990; 37(1): 85-98.

[12] W. Zong, G. Moody, and D. Jiang, "A robust open-source algorithm to detect onset and duration of QRS complexes," *Comput. Cardiol. 2003*, 2003.

[13] Lynn PA, "Online digital filter for biological signals: some fast designs for a small computer," *Med Biol Eng Comput.* 1977; 15:534-540

[14] J. Han, M. Kamber, J. Pei, "Data Mining Trends and Research Frontiers," in *Data Mining Concepts and Techniques*, 3rd Ed. Boston, MA: Morgan Kaufmann, 2012, ch. 13, sec. 1, pp. 585-597.

[15] K. Yip, D. Nembhard, "MWASP: Multiple-Width Approximate Sequential Patterns," in *2009 IEEE Symposium on Computational Intelligence and Data Mining.* 2009; 314-319

[16] S. Ghosh, M. Feng, H. Nguyen, J. Li, "Predicting Heart Beats using Co-occurring Constrained Sequential Patterns," in *Computing in Cardiology*, 2014: 265-268.

[17] Afonso, V. (1993). "ECG QRS Detection," in *Digital Signal Processing.* Englewood Cliffs, NJ*:* 236-264.

[18] J. Lin, E. Keogh, L. Wei, S. Lonardi, "Experiencing SAX: a Novel Symbolic Representation of Time Series," in*Data Mining and Knowledge Discovery*, 2015; 15(2): 107-144.

[19] P. Fournier-Viger, A. Gomariz, M. Campos, R. Thomas, "Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information," in *Advances in Knowledge Discovery and Data Mining*, 1st Ed. Tainan, Taiwan: Springer, 2014, pp. 40-52.

[20] M Zaki, "SPADE: An Efficient Algorithm for Mining Frequent Sequences," in *Machine Learning*, 2001; 42: 31-60.

VITA

Author: Sarah Bass

Place of Birth: San Antonio, Texas

Undergraduate Schools Attended: Indiana University at Bloomington

Degrees Awarded: Bachelor of Music in Viola Performance, 2007, Indiana University

Honors and Awards: Graduate Assistantship, Computer Science Department, 2013-2015, Eastern Washington University

Professional Experience: Associate Software Developer, FPI, Spokane, Washington, 2015 - present