–Independent Work Report Fall, 2015–

Evaluation of Bayesian Changepoint Detection of Sepsis in Hospital Patient Monitoring

Haley Beck '16

Abstract

Advances in healthcare technology have made more expansive time-series data available for modeling and monitoring health outcomes. Gaussian processes (GP), a form of Bayesian time-series monitoring, has been a popular method for modeling this data because GPs take observed values at a finite set of time points and generate robust predicted values over all unobserved time points. However, patient time-series data often undergoes drastic and sudden changes as a result of clinical events and treatment. In other contexts, research has shown that modeling these sudden changes as changepoints, that is as time points in which the model parameters of the GP alter, can improve the quality of predictions. The clinical event of sepsis, a potentially life-threatening complication of the body's infection fighting response, causes particularly dramatic changes in heart rate time-series. This paper uses sample heart rate time-series obtained from the Hospital at the University of Pennsylvania to evaluate if treating the clinical event of sepsis as a model changepoint in the heart rate GP can improve model fit. Secondly, we evaluate if detection of changepoints can lead to earlier predictions of sepsis as compared to the time of clinical diagnosis.

1. Introduction

As healthcare technology advances, the quality and availability of data that can be used for modeling and monitoring health outcomes has increased dramatically. In the noisy, dynamic, and time-sensitive environment of hospital monitoring, Bayesian time-series analysis has become popular because the approach is well suited for highly uncertain contexts [14].

This paper will focus on one particular form of Bayesian time-series monitoring, the Gaussian process (GP) which takes a finite sample of observation-value pairs

$$\{(x_1, y(x_1)), (x_2, y(x_2)), ..., (x_k, y(x_k))\}$$

and generates robust predictions of function value $y(x_*)$ over all unobserved values x_* [14]. We define a GP by its mean function and covariance kernel function, which gives the relationship between pairs of data points used for inferencing. We define the kernel by a function and a set of generative parameters, θ.

Existing research has shown that GPs can be used to model patient vitals, such as heart rate and blood pressure, as well as lab results, such as lactate levels and cell counts, during hospital admissions in order to predict when patients are at risk for adverse events or when further observations are needed (e.g. the uncertainty around the predicted white blood cell count is very large so another blood test should be performed to ensure the patient has an appropriate level).

Yet, some time series undergo drastic, sudden changes in covariance occurring at times called changepoints. In formal terms, a changepoint is defined as an abrupt variation in the generative kernel parameters of a time series [2]. Taking the changepoint into account when modeling by using different kernel parameters before and after the changepoint has been shown to improve the fit of GPs in many applications such as genomics and econometrics [2]. This paper aims to demonstrate that changepoints can be integrated into existing GPs to improve model and prediction of clinical data. More specifically, we will explore this question in terms of one possible clinical changepoint event, sepsis.

Sepsis, as defined by the the Mayo Clinic, is a "potentially life-threatening complication of an infection" in which chemicals released into the bloodstream to fight infection trigger an inflammatory cascade that can "damage multiple organ systems, causing them to fail" [10]. We choose to model this event because it has a clear clinical definition and causes drastic changes in vital signs. More specifically, this paper will model the onset of sepsis as a changepoint for patient

heart rate because increased heart rate is one of the diagnostic features of sepsis and heart rate is observed at high sampling rate because it has negligible observation cost.

We start with a short overview of the mathematical model used for Gaussian processes and changepoints (**Section 2**), before applying it to the specific problem of hospital monitoring of sepsis. In **Section 3**, we demonstrate that sepsis can, indeed, be meaningfully modeled as a change point the heart rate GP. We do this by showing that the clinical diagnosis of sepsis modeled as a changepoint improves GP fit using uncertainty ratios and Bayes Factors. **Section 4** expands on this finding by testing whether changepoint detection algorithms can detect sepsis or pre-sepsis at time points earlier than the clinical diagnosis. Early detection is a particularly motivating problem for sepsis because early treatment with antibiotics and intravenous fluids increases the chances of patient survival [10].

2. Models

2.1. Gaussian Processes

We can think of a Gaussian process (GP) in terms of a prediction problem: given a set of noisy observations of $y(x)$ at certain values of the independent variable x, what is our best estimate of $y(x_*)$ at some unobserved x_*? This becomes a regression problem of the form $y(x) = f(x) + \eta$ in which $f(x)$ is the underlying function and η is a noise process [14]. Curve fitting can be used to estimate the unknown function $f(x)$ based on observed data pairs $(x, y(x))$ allowing for extrapolation of unobserved $y(x_*)$ values. This paper focuses specifically on non-parametric GP models because they do not require any prior knowledge about the function $f(x)$, which means we do not have to know the explicit set of parameters that need to be inferred by our curve fitting.

We use a method for generating GPs that is described in detail in Roberts et al. (2010). Here, we provide a high-level description of how it works. The covariance kernel function $k(x_i, x_j)$ gives the correlation between any two observations, $y(x_i)$ and $y(x_j)$, of any sized data set at observed time points, x_i and x_j [14]. The covariance matrix K is the collection of all possible $k(x_i, x_j)$ pairs for a given time series in which the off-diagonal of the matrix defines the correlation between pairs x_i and

x_j where i is the column number and j is the row number. This means that the diagonal represents all pairs such that $i = j$ meaning it is the correlation between a given observation and itself.

Thus, noise is added to the model along the diagonal of K, as it does not affect the correlation among pairs of different observations (off-diagonal entries of the covariance matrix). If we let, σ^2 be the hyperparameter representing noise variance and I be the identity matrix, then we can rewrite a new covariance matrix, V, that accounts for noise as follows:

$$V(x,x) = K(x,x) + \sigma^2 I$$

We can write the probability of a given observation as a function drawing from a multivariate Gaussian distribution in which μ is the mean function evaluated at all locations of x [14]:

$$p(y(x)) = \mathcal{N}(\mu(x), K(x,x))$$

Now, to evaluate the GP at a new time point, x_*, we evaluate the joint distribution of the past observed data along with x_* and y_* as follows:

$$p\left(\begin{bmatrix} y \\ y_* \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mu(x) \\ \mu(x_*) \end{bmatrix}, \begin{bmatrix} V(x,x) & V(x,x_*) \\ V(x_*,x) & v(x_*,x_*) \end{bmatrix}\right)$$

in which $V(x_*,x)$ is the x_* column of the covariance matrix V and $V(x,x_*$ is its transpose) [14].

Roberts et al. (2010) shows that posterior distribution over $y*$ is Gaussian with mean and variance as follows:

$$m_* = \mu(x_*) + V(x_*,x)V(x,x)^{-1}(y - \mu(x))$$

$$\sigma_*^2 = V(x_*,x_*) - V(x_*,x)V(x,x)^{-1}V(x,x_*)$$

Expanding this to the set of unobserved time points x_* allows us to compute the posterior distribution

of $y(x_*)$. The posterior mean and variance can be defined as follows:

$$m_* = \mu(x_*) + V(x_*,x)V(x,x)^{-1}(y(x) - \mu(x))$$

$$C_* = V(x_*,x_*) - V(x_*,x)V(x,x)^{-1}V(x_*,x)^T$$

$$p(y_*) = \mathcal{N}(m_*,C_*)$$

So how do we define the kernel function, k? A valid kernel function guarantees that the resulting covariance matrix is positive semi-definite, so there are actually many valid kernel functions that could be selected. For this paper, we chose the well-tested Radial Basis Function kernel or squared exponential kernel, which handles smoothing well [14]. It follows the form:

$$k(x,x') = \sigma^2 exp\big(-\frac{(x-x')^2}{2l^2}\big)$$

for the covariance between two samples, x and x', where σ^2 is the average distance of the function away from its mean, and l is the length scale.

2.2. Changepoint Modeling

There are many different proposed algorithms for changepoint modeling. Much of the early research on changepoints proposed models that use MCMC methods, such as Stephens (1994) and Chib (1996). However, these algorithms are limited by the issue of having to diagnose convergence of the MCMC algorithms. Instead, we use an approach in this paper that avoids the MCMC problem by molding the changepoint as a point process based on Pievatolo and Green (1998) and Barry and Hartigan et al. (1992) [13] [4]. These methods can more easily be extended to our ultimate goal of online changepoint detection.

Here we provide an overview of the concept. For modeling a changepoint in one dimension, we divide the sequence of consecutive observations $x_1, x_2, ..., x_n$ into contiguous, non-overlapping

product partitions or blocks:

$$\left\{x_1,...,x_{j_1}\right\},\left\{x_{j_1+1},...,x_{j_2}\right\},...,\left\{x_{j_{b-1}+1},...,x_{j_b}\right\}$$

in which the break between partitions is called a changepoint [4]. Let $0 < \tau_1 < ... < \tau_m < n$ be the times of these changpoints. Let the j^{th} partition contain the set of sequential observations from time point $\tau_{j-1}+1$ to time point τ_j. Each partition j gets a set of parameters, which we will call σ_j. We let $\pi(\sigma_j)$ be the prior for σ_j, assuming independent priors for the parameters of each partition. Since observations are assumed to be independent and conditional on the changepoint and parameter values, then we can say that the observation at time i, y_i, in the j^{th} partition is drawn from the density $f(y_i|\sigma_j)$.

To determine priors, the Python function we use, discussed in 4, uses the product partition method described in detail in Barry and Hartigan (1992). This method obtains the prior by treating the changepoint in terms of a point process, which is defined by a probability mass function for the time between adjacent points. We can look at the point process across the interval $[1, n-1]$, and assume that the points in the corresponding point process are the locations at which the changepoints occur. With these variables defined, we can compute:

$$P(t,s) = Pr(y_{t:s}|\text{ t,s in same segment})$$

$$= \int \Pi_{i=t}^{s} f(y_i|\theta)\pi(\theta)d\theta$$

$$Q(t) = Pr(y_{t:n}|\text{ changepoint at t-1})$$

$$\text{where, } Q(1) = Pr(y_{1:n})$$

Fearnhead (2006) then uses these formulas to derive the following recursive statements to calculate the probabilities [6]. The data within a partition are assumed to be i.i.d. for some probability distribution, and the parameter values corresponding to each partition are assumed to be independent

of each other. This assumption that partitions are independent of each other allows us to use Forward-Backward recursion [15] because it ensures the Markov property.

Putting all of this together, we can compute the marginal likelihood that the data from time point j_i to j_k is produced under a single model, m, as follows [16]:

$$p(f(x)_{s:t}|m) = \int [\prod_{i=s}^{t} p(y_i|y_{1:i-1}, \theta, m)] p(\theta|m) p(m) d\theta$$

in which θ is the set of kernel hyperparameters for model m. As the current partition extends to include data generated under a different set of hyperparameters, θ', we expect the marginal likelihood to decrease. Drops in the marginal likelihood suggest that the partition has reached a changepoint and the data would be better modeled using two distinct sets of hyperparameters.

3. Aim 1: Modeling the Clinical Diagnosis of Sepsis as a Changepoint

3.1. Data and Software Details

We obtained time-series data for 4,214 patients diagnosed with severe sepsis during admission to the Hospital of the University of Pennsylvania [12]. We randomly selected 100 of these patient records for analysis, restricting this sample to well-observed patients defined as records with at least 50 heart rate observations recorded before and at least 50 after the diagnosis of severe sepsis. These time series are noisy and observations are recorded at varying intervals of time. Figure 1 illustrates the observed heart rate time series for a selected patient.

All analysis was done using Python scripts. We generated plots using the Matplotlib module and data analysis was aided by the NumPy, SciPy, and Matplotlib modules [7] [11] [8]. Scripts were modeled after examples provided by a graduate student in the group [5].

3.2. Methods and Results

First, we aimed to demonstrate that sepsis can be modeled as a changepoint in a GP and that modeling this changepoint improves the fit and predictiveness of the GP. We assumed that the time

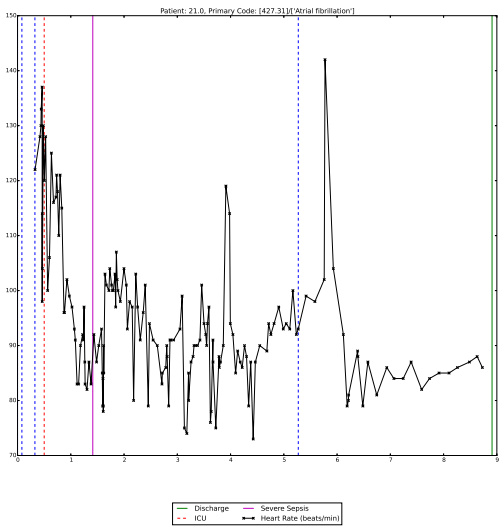

Figure 1: Example time series of observed heart rate for patient no. 21 in which the x-axis is time in hours and the y-axis is heart rate in beats per minute. Vertical lines signify clinical events define in the legend.

of clinical diagnosis of severe sepsis was in fact the true time of sepsis onset. To accomplish this first aim, we compared the fit using Bayes Factors and confidence interval size of the following models.

Let θ_b be the MLE hyperparameters for the GP modeling data before the changepoint, and let θ_a be the MLE hyperparameters for the GP modeling data after the changepoint. The null model assumes sepsis as a true changepoint meaning $\theta_b \neq \theta_a$. The alternative model assumes sepsis is not a true changepoint meaning there is no change in hyperparameters after sepsis so $\theta_b = \theta_a$.

We fit GP regression models to the data using the GPy module written in Python by the Sheffield machine learning group, which takes the observations and a kernel [3]. The x vector represented the time of vital observation, and the $f(x)$ vector, the observed heart rate value at time x. We first defined the covariance kernel as a non-informative radial basis function (RBF) or Gaussian kernel, which takes a scale length and variance hyperparameter:

```
kernel = GPy.kern.RBF(input_dim=1,
          active_dims=0, variance=1., lengthscale=10.)
gp_model = GPy.models.GPRegression(x1, y1, kernel)
```

We initially set default hyperparameters (variance = 1.0, scale length = 10.0) and allowed subsequent

calls for optimization adjusted the kernel parameters to values that maximize the likelihood of the data. The optimization function selects ten random initial values for the hyperparameters drawn from $N(0,1)$, optimizes each case, and selects as the model the best solution from the set [3]. When we report and compare hyperparameter values, we are referring to the MLE values after optimization as all models are given the same parameters initially.

Initially, we trained three optimized GP models for each patient: one fit to all observations ("full"), one fit only to observations taken before sepsis ("before"), and one fit only to observations taken after sepsis ("after"). For each model, we retrieved its corresponding optimized hyperparameter values for scale length and variance. The distribution plots 2 of the hyperparameter value for variance in the 100 test patients does not show a significant difference in the before model compared to the after model.

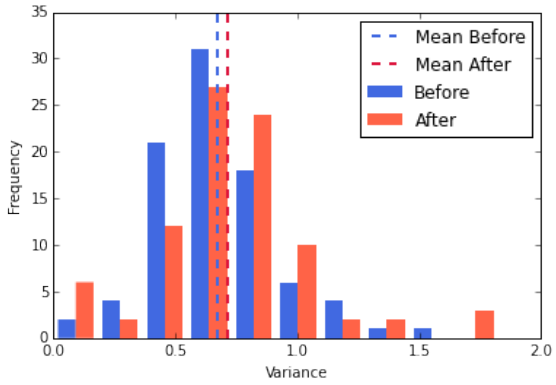

Figure 2: Histogram of variance hyperparameter values from optimized kernels fit to observations before the changepoint (sepsis event) and to observations after the changepoint for 100 patients. The mean of variance in the before models is 0.68 (std. 0.24) and of the after models is 0.72 (std. 0.33).

While there is not an apparent difference in the aggregate across patients, the hyperparameters for the before and after models were significantly different within single patients. For each patient, we took the model trained on the before observations and tested it on that same patient's observations after sepsis, and vice versa performing a sort of cross-validation. Figure 7 shows the GP plots of the cross-validation experiment for a sample patient. The x-axis is time since hospital admission in hours and the y-axis is heart rate in beats per minute. Black crosses represent observed data points

and the navy line is the GP prediction for heart rate at a given timepoint. The ligh blue shaded region represents the uncertainty around the pointwise GP predictions, which is defined as $\pm$ twice the point-wise standard deviation of the GP. The uncertainty window can also be thought of as the 95% confidence interval for the point-wise GP predictions.

Before we quantify the difference between models, we can note that simply by visual inspection there is noticeably more uncertainty around the predictions in Figure 4 and 6. Both Figure 3 and 4 are plotted using the same observation set, all the data points observed before the sepsis event. However, the GP plotted in Figure 3 was trained using only before sepsis observation whereas the GP in Figure 4 was trained using all (both before and after sepsis) observations. The considerably larger uncertainty window in 4 suggests that the GP with kernel hyperparameters trained on all observations performs worse on the data observed before sepsis than a GP with kernel hyperparameters trained on only data observed before sepsis. We can observe the same trend when we use the data observed after sepsis for testing. The uncertainty window for Figure 6, plotting how well the GP trained using all observations performs on the data observed after sepsis, is much larger than that of Figure 5, plotting the GP trained only on data observed after sepsis.

The same trend was observed for all sampled patients (see Appendix A). Visually, this supports the hypothesis that the sepsis event is a changepoint that causes an abrupt change in the generative parameters of the time series. The models trained using data from another partition, data that was generated under a different set of parameters, performed worse than models trained using only data from within a single partition. Next, we quantify this observation using two methods of comparison.

Comparing Areas of Uncertainty

First, we can quantify the difference in models by comparing the areas of uncertainty around the GP predictions. As mentioned previously, at each time point the uncertainty is $\pm$ twice the standard deviation of the GP predictive mean, which also represents the 95% confidence interval of the point prediction. We can sum over the point-wise uncertainties across the entire time series to calculate the area of uncertainty for the GP predictions on the entire series.

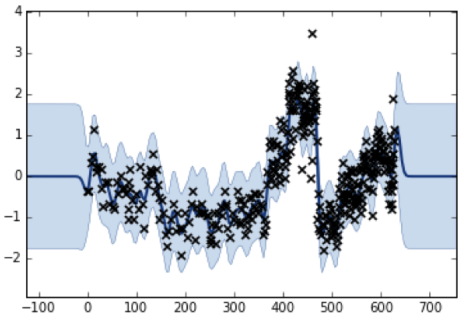

Figure 3: Train = before, Test = before

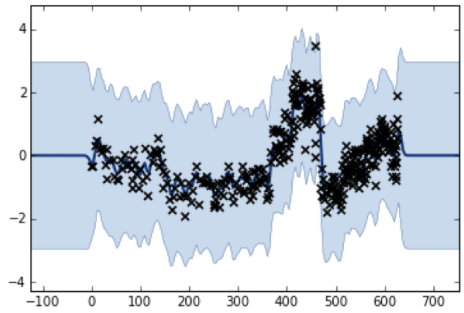

Figure 4: Train = after, Test = before

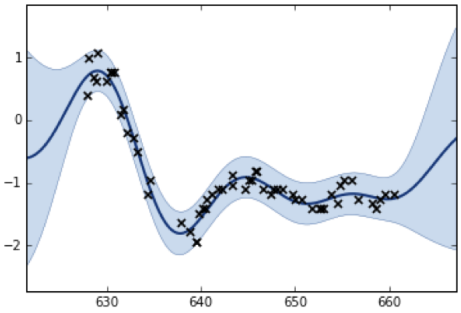

Figure 5: Train = after, Test = after

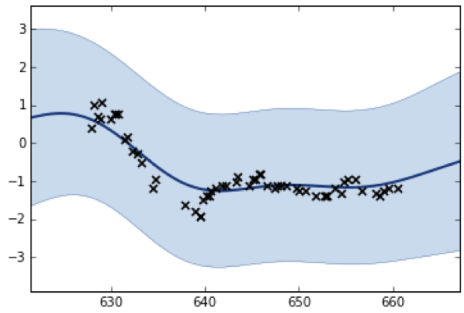

Figure 6: Train = before, Test = after

Figure 7: Cross validation plots for sample patient (patient no. 9879). (a) GP trained and tested on before sepsis observations. (b) GP trained on after sepsis observations and tested on before sepsis observations. (c) GP trained and tested on after sepsis observations. (d) GP trained on before sepsis observations and tested on after sepsis observations.

Table 1 shows the mean area of uncertainty for each GP model across all sampled patients (N=100). The models that were tested and trained within a single partition (B and D) had significantly less area of uncertainty than the models trained using all observations then tested on a single partition (A and C). We can directly compare A with B (Figure 8) and C with D (Figure 9) because each pair was tested on the same set of observations respectively. Averaged across all sampled patients, the ratio of uncertainty in A to B ($\mu = 0.51$, SD = 014) and the ratio of uncertainty in C to D ($\mu = 0.48$, SD = 0.18) is approximately one-half. Again this demonstrates that there is a meaningful difference between the two partitions, before sepsis and after sepsis, indicating the usefulness of modeling sepsis as the changepoint or break between the partitions for reducing the uncertainty associated with GPs.

Bayes Factors The Bayes Factor is the ratio of the probability of the observed data y given a null

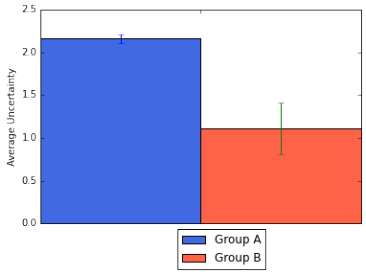

Figure 8: Comparison of mean model uncertainty for before sepsis observations under (a) no changepoint model, (b) changepoint model.

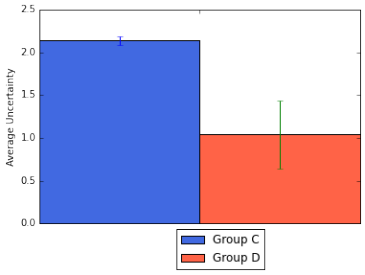

Figure 9: Comparison of mean model uncertainty for after sepsis observations under (c) no changepoint model, (d) changepoint model.

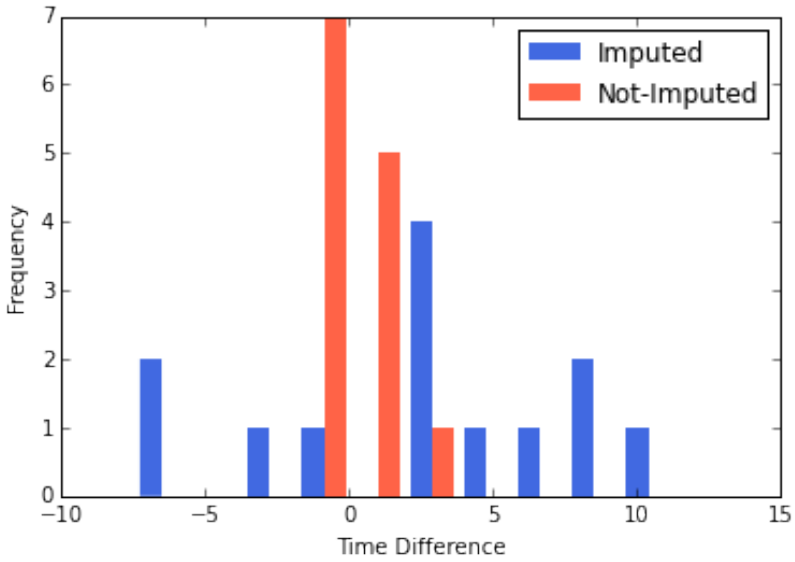

Figure 10: Difference between time of clinical diagnosis of sepsis and nearest detected changepoint time.

model, M_0, over the probability of the observed data given an alternative model, M_1:

$$B = \frac{p(y|M_0)}{p(y|M_1)}$$

In other words, it is the ratio of the marginal likelihood of the observed data in M_0 and M_1 [9]. Thus, a large Bayes factor indicates strong evidence in favor of M_0. For each patient, we trained a GP using the before sepsis observations only, using the after observations only, and using all observations to yield three sets of hyperparameters, θ_b, θ_a, and θ_{all}, respectively. For the numerator of the Bayes factor computation, we computed the probability of the before observations under

Experiment	A	B	C	D
Observations used for training	All	Before	All	After
Observations used for testing	Before	Before	After	After
Mean model uncertainty of testing	2.16	1.11	2.14	1.04
Standard deviation of mean model uncertainty	0.05	0.30	0.05	0.40

Table 1: Comparison of mean areas of uncertainty (95% CI area or ± 2 std) for all patients (N=100).

a GP given θ_b times the probability of the after observations under a GP given θ_a, to obtain the probability of the data given the null model (assuming sepsis as a changepoint, $\theta_a \neq \theta_b$). For the denominator, we computed the probability of all observations in the time series under a GP given θ_{all}, to obtain the probability of the data given the alternative model (assuming sepsis is not a changepoint, $\theta_a = \theta_b$). We computed the probability of a set of data under a certain model by exponentiating the log-likelihood of the corresponding GP using GPy.

Table 2 shows the smallest Bayes factor as well as the first quartile and mean value for the 100 sampled patients. The Bayes factor for all sampled patients showed preference for the null model that assumes sepsis as a changepoint. According to the Jeffreys' scale for interpreting Bayes factors [9], factor values larger than 10^2 or 2 on the $\log_{10}$ scale offer "decisive" evidence in favor of the null model. Thus, all except 8 of the 100 patients sampled showed "decisive" evidence in favor of the model with sepsis as a changepoint. These 8 cases still indicated that the data provides "substantial" to "very strong" evidence in favor of sepsis as a changepoint [9].

	log scale
Mean Bayes Factor	62
Minimum Bayes Factor	0.65
First Quartile (Q1) Bayes Factor	10

Table 2: Bayes Factors for 100 sample patients comparing null model ($\theta_a \neq \theta_b$) to the alternative model ($\theta_a = \theta_b$). The Bayes factor is the ratio of the probability of the observed data given the null model over the probability of the observed data given the alternative model. A large Bayes factor indicates strong evidence for the null model as compared to the alternative model.

3.3. Discussion

In this section, we treated the clinical diagnosis of sepsis as the ground truth time of sepsis onset. We demonstrated that this definition of sepsis indeed can be modeled as a meaningful

changepoint for Gaussian processes of in-hospital patient heart rate monitoring as the generative kernel hyperparameters before sepsis of a given patient are significantly different from the generative hyperparameters of the same patient after clinically diagnosed sepsis. We observed that models before the changepoint generally fit better than models after the changpeoint. We hypothesize that this is because patients are administered many drugs and fluids after the onset of sepsis and that some patients recover and are no longer septic. Thus, after the onset of sepsis there are likely additional changepoints that occur. As 1 and 2 demonstrate, integrating the sepsis changepoint into GP models reduces uncertainty across the series allowing for better predictive power for the GP model. This finding is important for two reasons.

First, while the experiments were conducted using only heart rate GP models, it is reasonable to hypothesize that sepsis also causes a changepoint in other vital signs. Improving GP fit for heart rate and other vital signs is important to all applications using these GP models for prediction, even those not primarily interested in the sepsis event.

Second, if the underlying sepsis event does in fact cause a changepoint in heart rate, then detecting changepoints can be used as a method for predicting when sepsis will occur. In this section, we purposefully ignored the inevitable error in clinically diagnosing and recording the onset of sepsis. It's unreasonable to suspect that the exact time of the clinical diagnosis of sepsis is always perfectly recorded in hospital records or that the clinical diagnosis time perfectly matches the time of the physiological onset of sepsis. There is likely a delay in both the diagnosis of sepsis, as doctors must first observe and recognize the symptoms, and in the recording of the event. In **Section 3**, we explore whether an algorithm to detect changepoints can predict that sepsis will occur earlier than the clinical diagnosis, which would enable earlier intervention potentially improving patient outcome.

4. Aim 2: Changepoint Detection of Sepsis Event

4.1. Data and Software Details

Having established that the clinical diagnosis of sepsis does, in fact, represent a statistically meaningful changepoint in the GP model, we will evaluate how well an algorithm can detect the sepsis changepoint. More specifically, we want to evaluate if an algorithm can detect the sepsis changepoint earlier than the clinical diagnosis time, which could allow for earlier clinical intervention. We tested this question on the same 100 sampled patients as discussed previously using a Python package [1] that implements the Hartigan Bayesian changepoint detection algorithm discussed in **Section 2**.

4.2. Methods and Results

We chose to initially implement changepoint detection for one-dimensional inputs because the algorithm has been optimized for speed and has been tested on a variety of datasets [1]. Since our model does not have prior knowledge, we assumed that initially it is equally likely for a changepoint to occur at any time point. Thus, we specified a uniform constant prior as the partial function for the likelihood of a changepoint given the distance to the last one.

We created an object of this function to be passed to the offline changepoint detection function. The function returns three objects: (1) an array of the $Q(t)$ that for each time point t gives the log-likelihood of the data for a partition from t to the end of the series, $[t,n]$, (2) a matrix $P[t,s]$ that gives the log-likelihood of a set of observations from time point $[t,s]$, given that there is not a changepoint between points t and s, and (3) a matrix $Pcp[i,t]$ that gives the log-likelihood that the $i-th$ changepoint occurs at time point t. The probability that a changepoint occurs at point t is found by the sum of the probabilities of $Pcp[i,t]$ for all possible i values. Probabilities are computed from the log-likelihoods using the exponentiation function $exp(i)$ from the Python module *numpy*.

```
partial(const_prior, l=(len(data)+1))
```

In general, this method using detection on one-dimension was able to strongly predict one to

many changepoints in all of the sampled patients. Empirically we observed that the algorithm runs in $O(n^2)$ where n is the number of observations in the time-series, although all sampled series ran within 10 minutes. However, a major issue with the algorithm on one-dimension assumes observations are regularly sampled as there is no dimension for time point. The clinical data, while frequently sampled was not perfectly regular. When two points are farther apart than normal, the algorithm will underestimate the prior likelihood of a changepoint occurring between them because the algorithm treats them as immediately adjacent observations.

To combat this limitation, we attempted to impute the time series so that observations would be regularly sampled, making appropriate input for the one-dimensional detector. The data was imputed using an algorithm implemented by a graduate student in the group and was smoothed by adding the pointwise standard deviation of the GP [5]. However, the variance in for the imputed data with smoothing was so small that the algorithm was extremely sensitive to any change in parameters, resulting in a high rate of false alarms. To combat this second issue, we added Gaussian noise to the time series by adding a draw from a random variable that is Gaussian with mean of zero and standard deviation equal to the measurement error, the point wise standard deviation. The noise was added using a random sampling function from the Python module numpy. However, empirically we did not observe any improvement over the smooth imputed data by adding noise.

Here we present an example case from this set of experiments for patient number 114330. The top subplot of Figure 12 shows the plot of the irregularly-sampled original observations. The blue line is the sequence of heart rate observations at each time points, connected together. The y-axis of the lower subplot of Figure 12 is the computed probability using the discussed detection algorithm that a changepoint has occurred at location x. We should note that the top and bottom subplots use the same x-axis time scale (in half hours) to allow for comparison between the two plots. In other words, a large peak in the lower subplot indicates a high probability that a changepoint has occurred at that location in the upper heart rate plot. We overlaid vertical lines on these plots to indicate the timepoints of clinically determined events such as severe sepsis and ICU admission.

We can notice that for patient number 114330, in Figure 12 the algorithm detects with high

probability a changepoint approximately 4 hours prior to the clinical diagnosis of sepsis (red line) and another changepoint with lower probability 6 hours after the clinical diagnosis. While this region of the plot visually has the most noticeable spikes of heart rate, there are other regions that show peaks and troughs yet the algorithm in this figure does not detect these false alarms with any significant probability. In Figure 11, we present the same patient except that the input observations have been imputed using GP prediction and smoothing so that the input data is regularly sampled at half-hour intervals.

To gauge how sensitive the changepoint detector is, we counted the proportion of patient models for which the algorithm detected a changepoint within x hours before the clinical diagnosis of severe sepsis. We hypothesize that features of pre-sepsis affect the kernel parameters as well and that the detection algorithm could possibly pick up this change even before the full clinical onset of sepsis. Figure 15 shows that for 70 percent of sampled septic patients, the algorithm did detect a change occurring within 12 hours of the clinical event. We discuss in the next section some possible interpretations of these results.

Additionally, we looked at the difference between the time of clinical diagnosis of sepsis and the nearest detected changepoint signal. In Figure 16, the x-axis represents the time (in hours) of the diagnosis relative to the detected point such that positive values indicate the diagnosis occurred after the detection point. The blue bars represent models imputed so observations occur regularly every half hour. The red bars indicate models that were not-imputed meaning sampling is irregular, and the time difference can only be estimated (the limitation is addressed in the next section). Overall, the detected changepoints has a slight tendency to occur before the clinical event, which supports our hypothesis that an improved algorithm can be clinically relevant by allowing for earlier detection of sepsis. Currently the algorithm works well for some models, but could be greatly improved to narrow the range. We discuss the limitations of these findings and ways to improve them through future work in the next section.

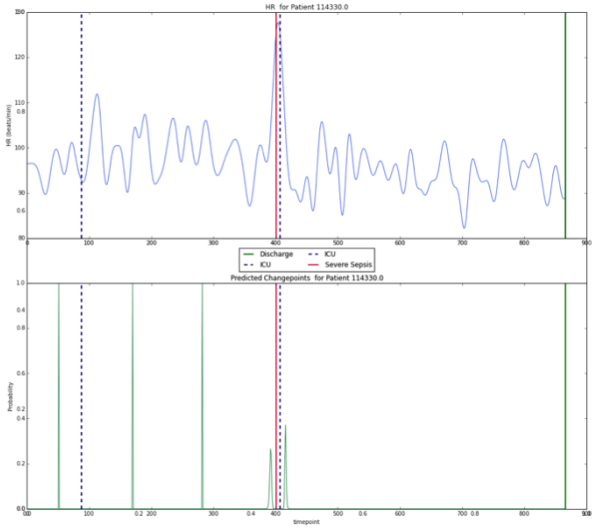

Figure 11: Imputed, Smoothed

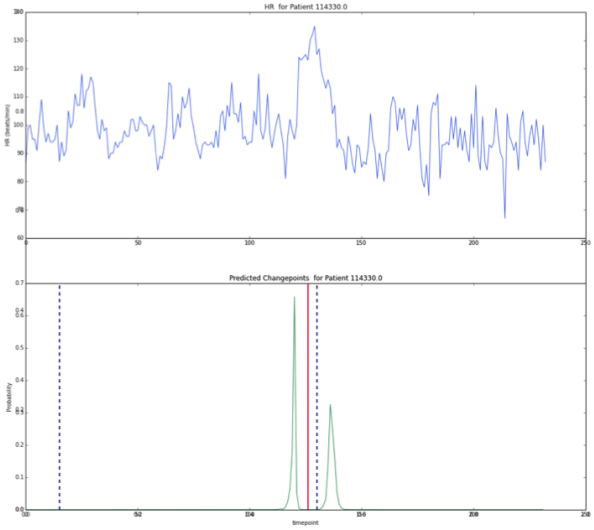

Figure 12: Non-imputed

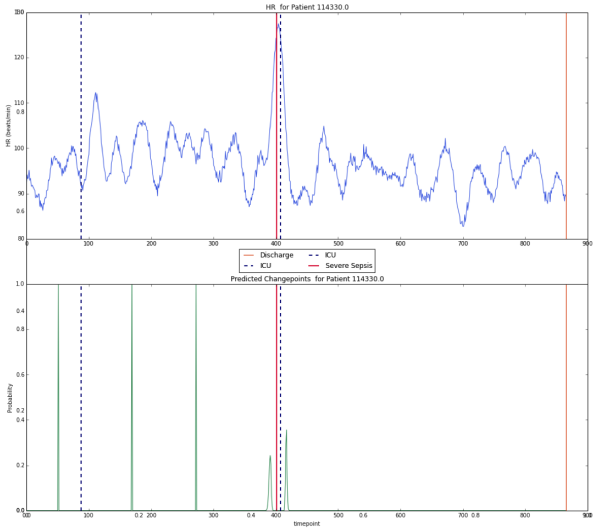

Figure 13: Imputed, Smoothed, Noise Added

Figure 14: Changepoint detection on various forms of input data from sample patient (patient no. 114330). The top subplot graphs the input values where the y-axis is heart rate in beats per minute. The bottom subplot graphs the probability that a time point has occurred at the given time point. Vertical lines indicate clinically diagnosed events in which red is severe sepsis and blue is ICU admission. The inputs of the graphs are as follows: (a) imputed data with smoothing (b) non-imputed data (c) imputed data with Gaussian noise added.

4.3. Limitations and Next Steps

There are a few inherent limitations in this analysis that are important to keep in mind and should inform the direction of future steps. Most significantly, there were no perfectly accurate observed data points. In other words, the physiological event of sepsis cannot be directly observed or verified

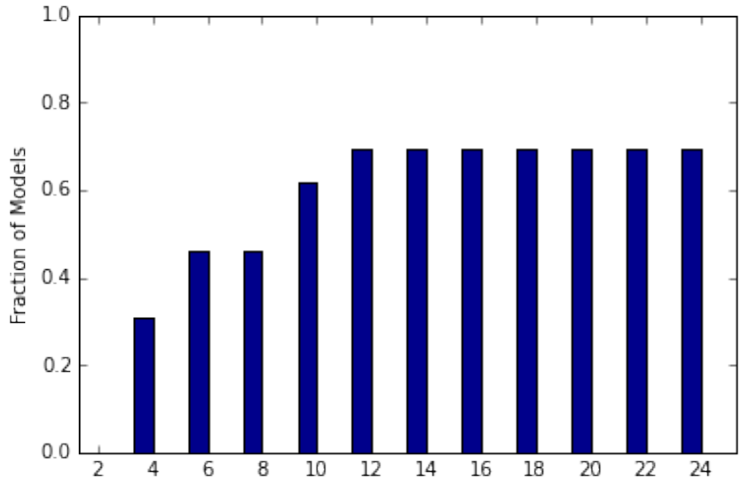

Figure 15: Proportion of models detecting a changepoint within x hours before the clinical diagnosis of severe sepsis.

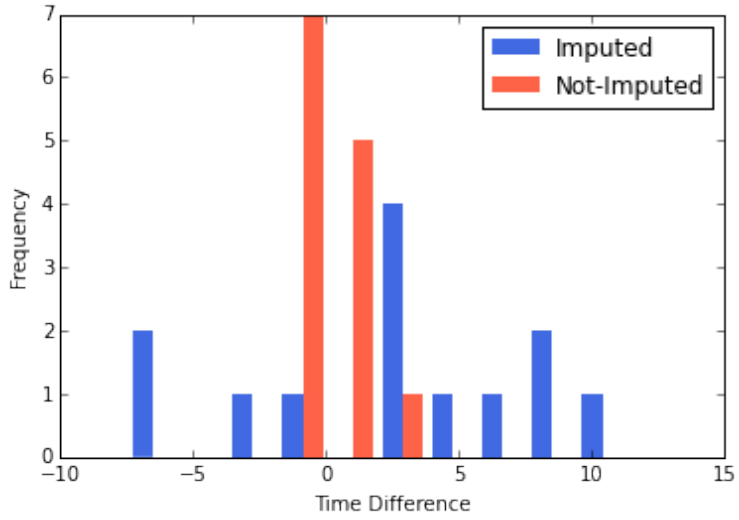

Figure 16: Time difference (hours) between the clinical diagnosis of sepsis and the nearest detected changepoint signal of $P \geq 0.2$.

so determining its time point can only be done through proxy. In **Section 3** we treated the time of clinical diagnosis of sepsis as the ground truth and assumed no input noise, but this is not realistic. Using real-world data, there was unquestionably some noise in both diagnosis and recording of the diagnosis. This makes the results of **Section 3** even stronger as even though we can't precisely determine the exact instant when sepsis occurs, the best clinical determination of this time still represents a meaningful changepoint.

The analysis in **Section 4** circumvented this potential limitation by framing the question

in terms of improvement for practical purposes: can an algorithm detect sepsis earlier than the clinical diagnosis regardless of the true time of sepsis? However, even this approach runs the risk of falsely attributing a detected changepoint to sepsis when it was actually caused by a different event; correlation does not imply causation. This complicates the detection problem as there is high risk of false alarms for other clinical events that cause similar effects on hyperparameters.

In light of these limitations, we outline future steps that should be pursued to mitigate them. To combat the issue of false alarms, there are three main approaches to be further developed. First, we can specify more about the different types of clinical events that occur. This project looked only at a single clinical event, but each of these patients undoubtably underwent an entire sequence of different events from changing beds to acquiring catheters to being administered drugs. If we can learn the characteristics of these different events and incorporate models for them into the kernel, then perhaps when a new changepoint arises in data not seen before we can more specifically determine with what clinical event it is associated. Specifically, we aim to incorporate a model for the administration of drugs as point processes that have effect on the kernel hyperparameters that decays over time.

In addition to learning about types of events across patients, we can also learn about clinical events within patients but across covariates. Patients with sepsis, as demonstrated in this paper, experience a changepoint in heart rate but also presumably in other covariates such as systolic blood pressure and temperature. An important future direction is to find a way to integrate knowledge from observations of other covariates into the prediction of a certain variable.

While this project focused on an offline algorithm, Xuan (2007) describes a method for extending the partition-product method into an online changpeoint detection program, that only relies on past observations. Our goal is to eventually implement the detector in an online context so that it can provide real-time prediction of sepsis and pre-sepsis in the clinical setting.

5. Honor Code

This work represents by own work in accordance with University regulations. /s/ Haley Beck

6. Acknowledgements

I really appreciate all the advice and guidance of Professor Engelhardt throughout this project. Also, I want to thank graduate student Li-Fang Cheng for taking the time to help me with the datasets and helping me through many obstacles.

References

[1] "Bayesian changepoint detection," https://github.com/hildensia/bayesian_changepoint_detection, accessed: 2015-04-10.

[2] R. P. Adams and D. J. MacKay, "Bayesian online changepoint detection," *arXiv preprint arXiv:0710.3742*, 2007.

[3] T. G. authors, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, 2014.

[4] D. Barry and J. A. Hartigan, "Product partition models for change point problems," *The Annals of Statistics*, pp. 260–279, 1992.

[5] L.-F. Cheng, "Upenn meddata exploration ipython notebook," personal communication.

[6] P. Fearnhead and Z. Liu, "On-line inference for multiple changepoint problems," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 589–605, 2007.

[7] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[8] E. Jones *et al.*, "SciPy: Open source scientific tools for Python," 2001, [Online; accessed 2015-04-29]. Available: http://www.scipy.org/

[9] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the american statistical association*, vol. 90, no. 430, pp. 773–795, 1995.

[10] "Diseases and conditions: Sepsis," http://www.mayoclinic.org/diseases-conditions/sepsis/basics/definition/con-20031900, Mayo Clinic, accessed: 2015-04-10.

[11] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 51 – 56.

[12] T. H. of the University of Pennslyvannia, "Anonymous patient data," unpublished.

[13] A. Pievatolo and P. J. Green, "Boundary detection through dynamic polygons," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 60, no. 3, pp. 609–626, 1998. Available: http://dx.doi.org/10.1111/1467-9868.00143

[14] S. Roberts *et al.*, "Gaussian processes for time-series modelling," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, p. 20110550, 2013.

[15] S. L. Scott, "Bayesian methods for hidden markov models," *Journal of the American Statistical Association*, vol. 97, no. 457, 2002.

[16] X. Xuan and K. Murphy, "Modeling changing dependency structure in multivariate time series," *Proceedings of the 24th international conference on Machine learning*, pp. 1055–1062, 2007.

Appendices

A.

Additional Cross Validation Graphs Cross validation plots for additional sample patients (in order, 1003, 5467, 2971, 7439, 8297, 10752, 976). For each patient there are four subplots corresponding to: (a) GP trained and tested on before sepsis observations. (b) GP trained on after sepsis observations and tested on before sepsis observations. (c) GP trained and tested on after sepsis observations. (d) GP trained on before sepsis observations and tested on after sepsis observations.

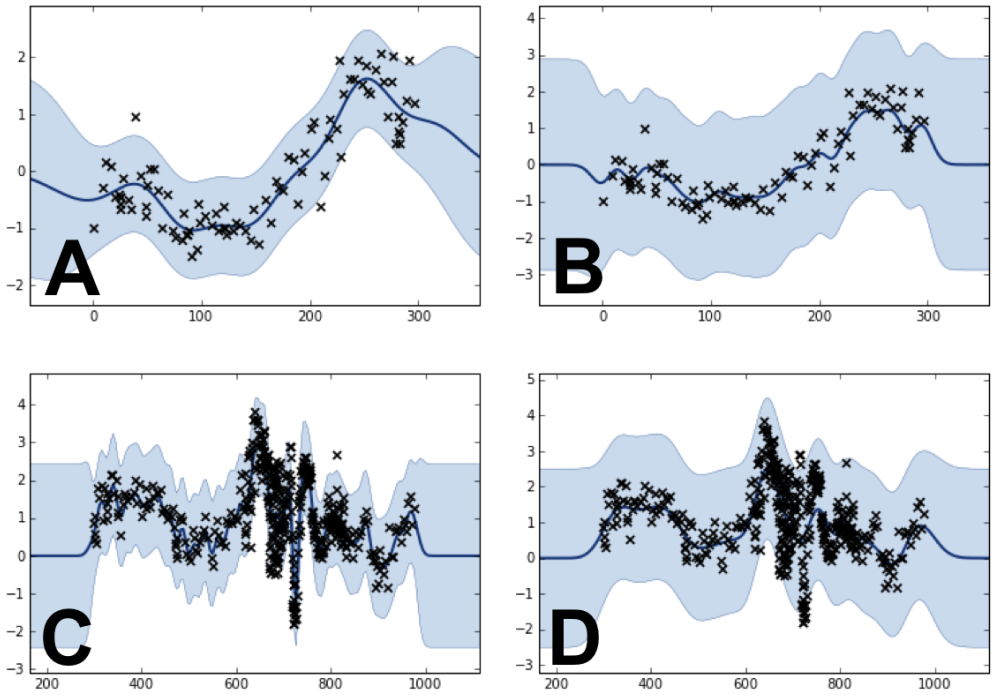

Figure 17: Patient 1003.

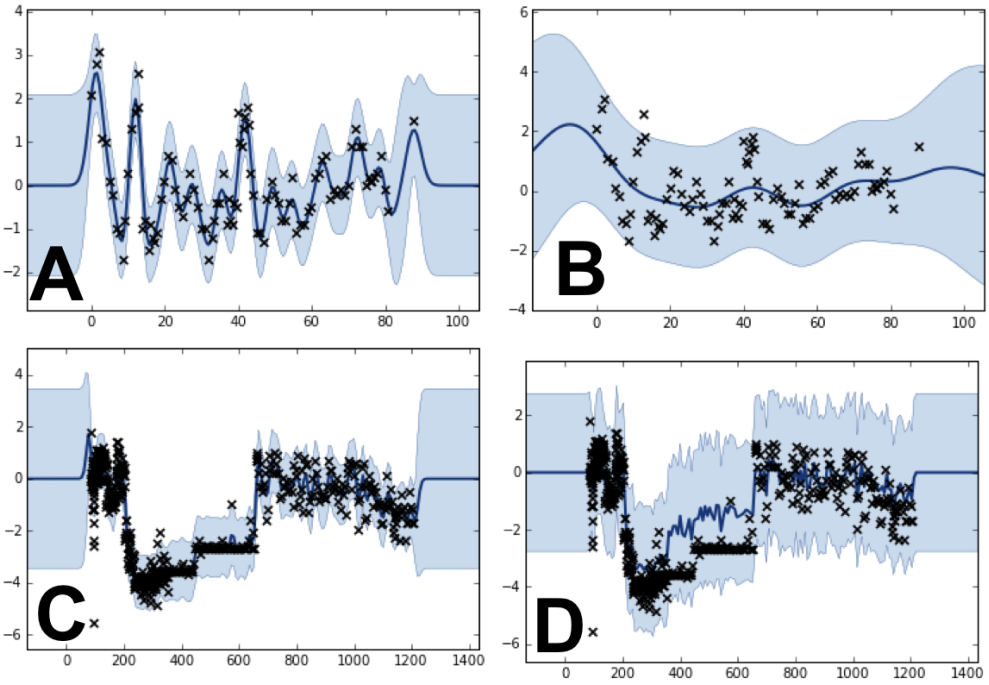

Figure 18: Patient 5467.

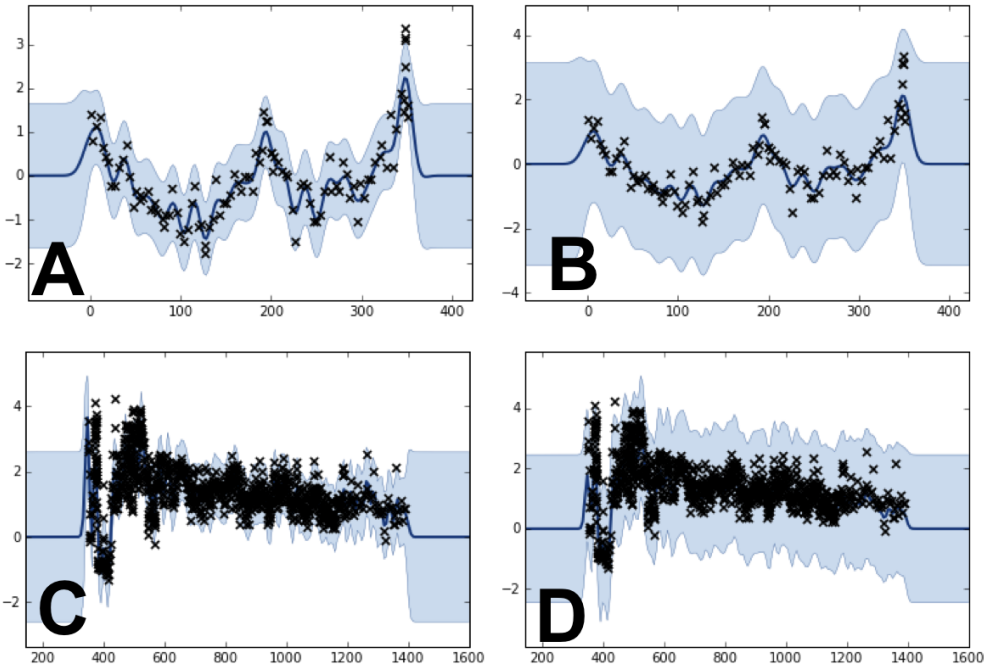

Figure 19: Patient 2971.

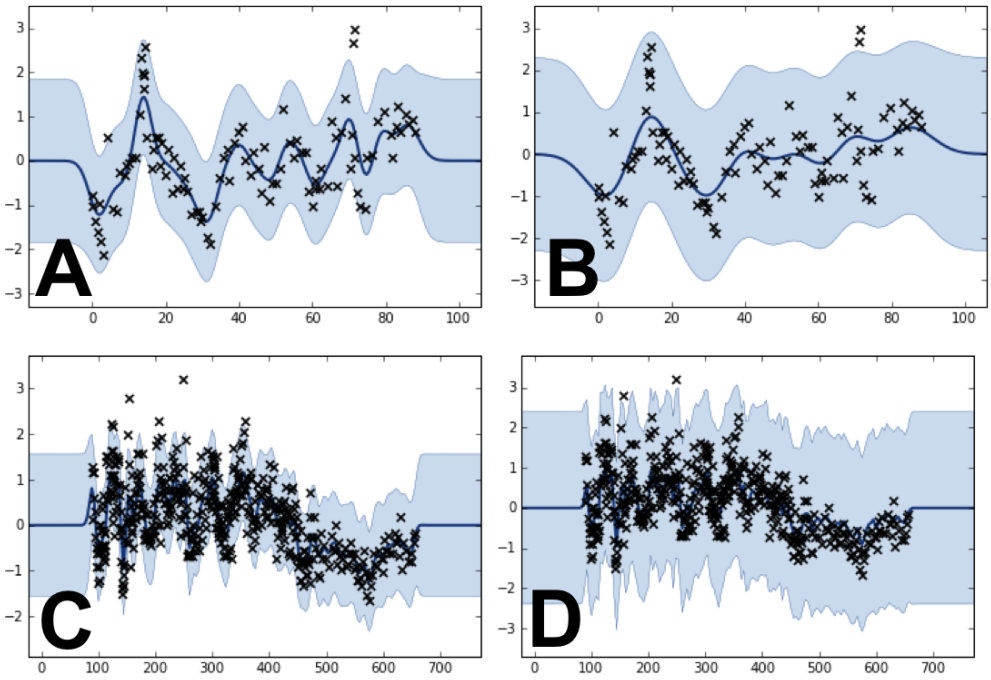

Figure 20: Patient 7439.

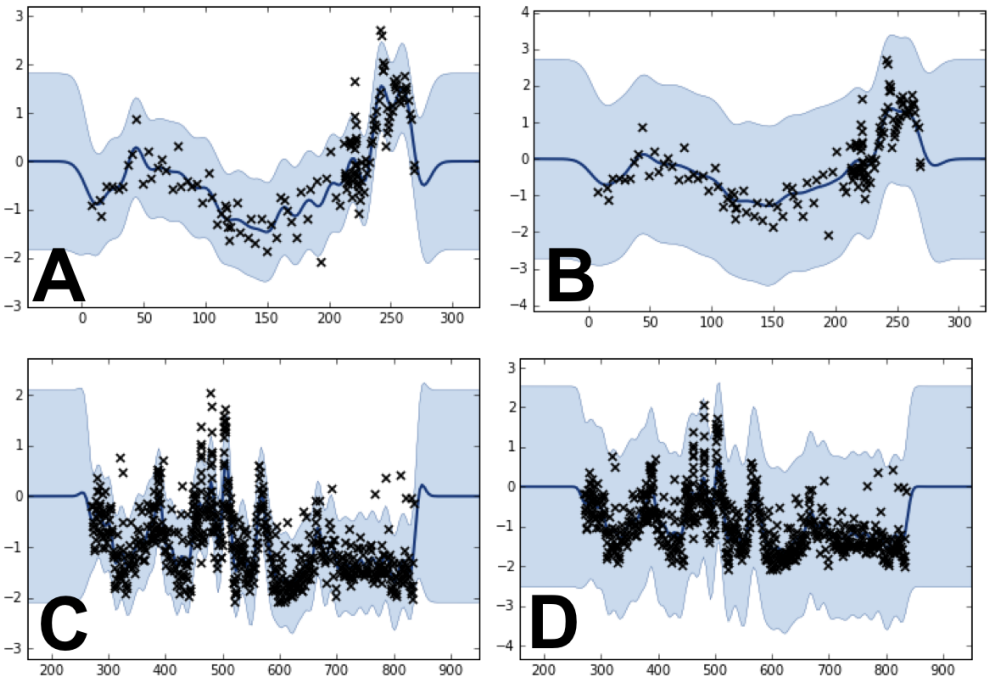

Figure 21: Patient 8297.

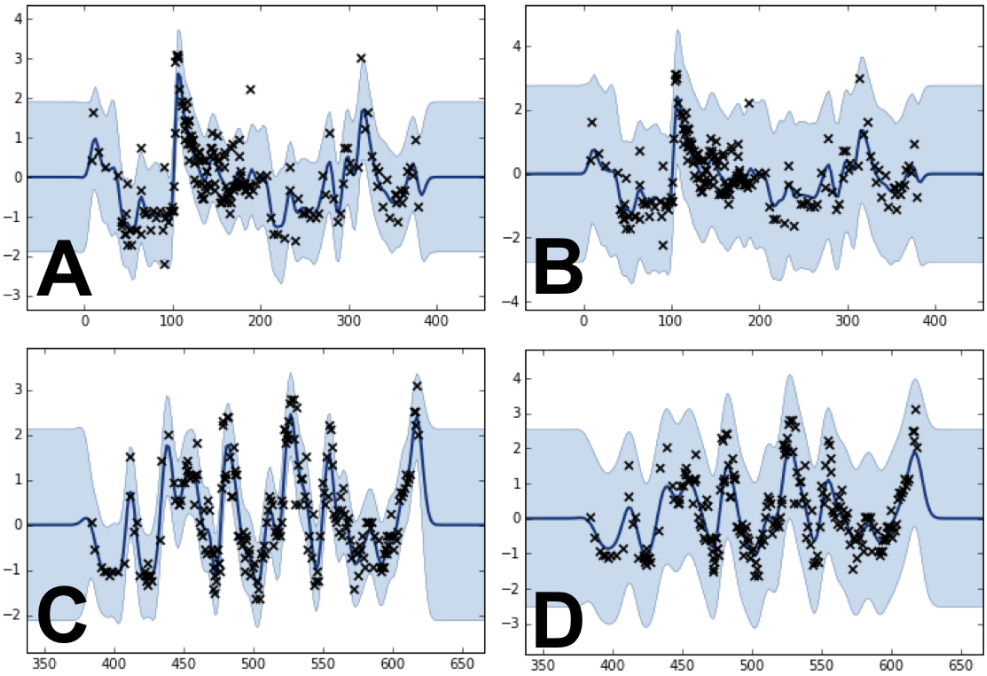

Figure 22: Patient 10752.

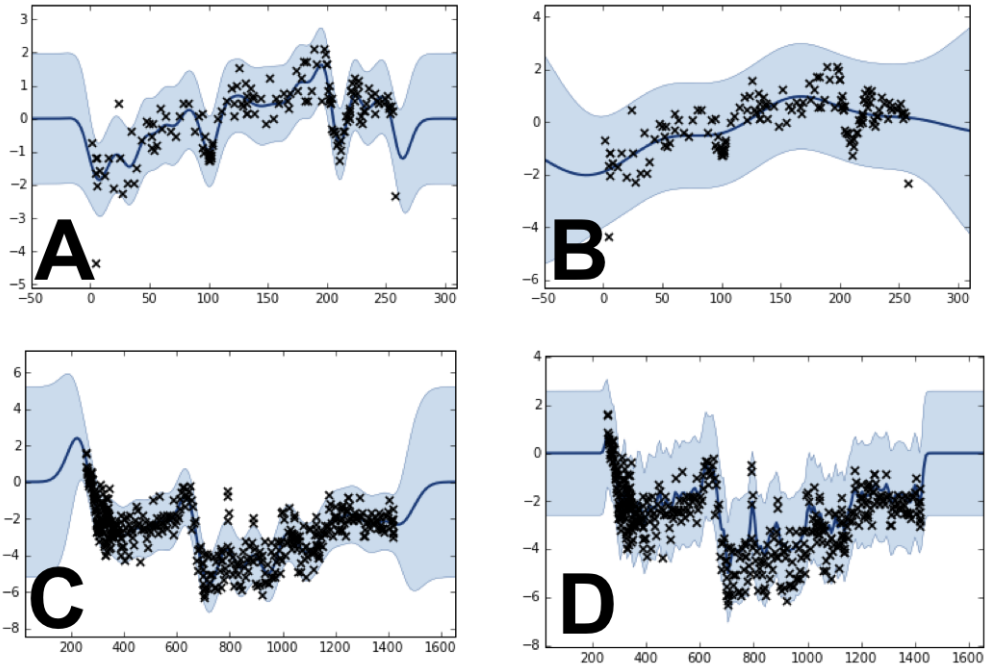

Figure 23: Patient 976.

B.

B. Additional Changepoint Detection Plots Changepoint detection plots for additional patient records. For each patient, the left subplot is the imputed time-series. The right subplot is the non-imputed time-series.

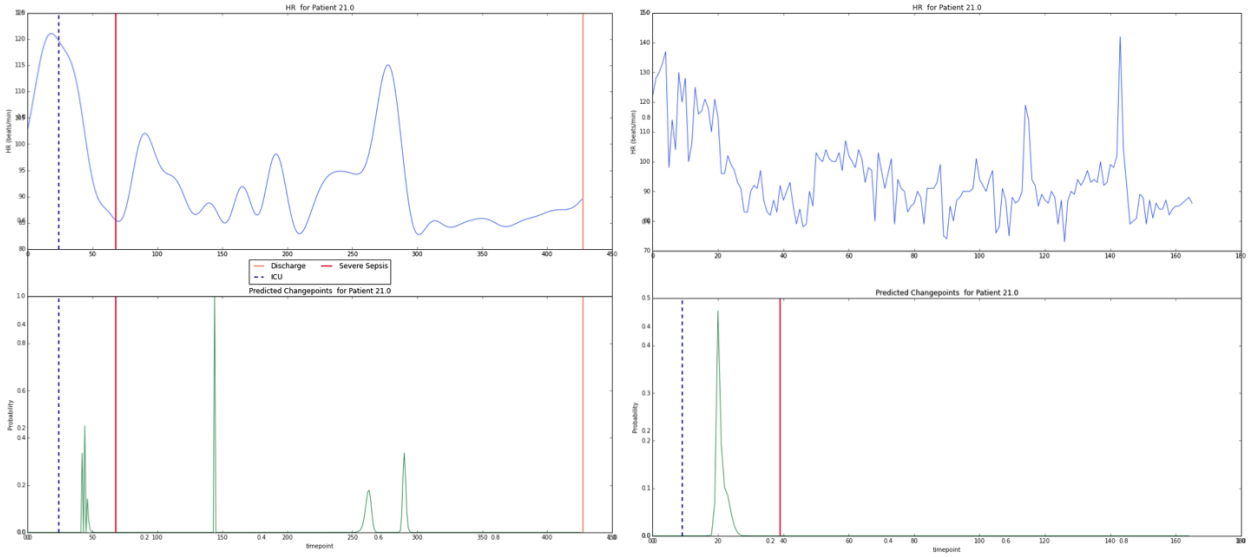

Figure 24: Patient 21.

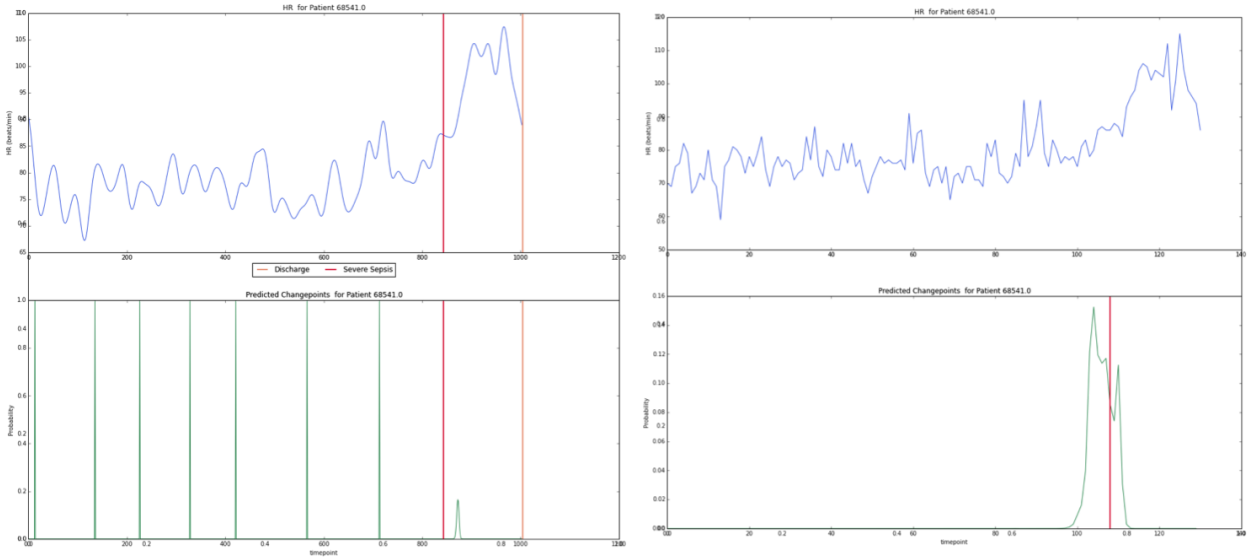

Figure 25: Patient 68541.

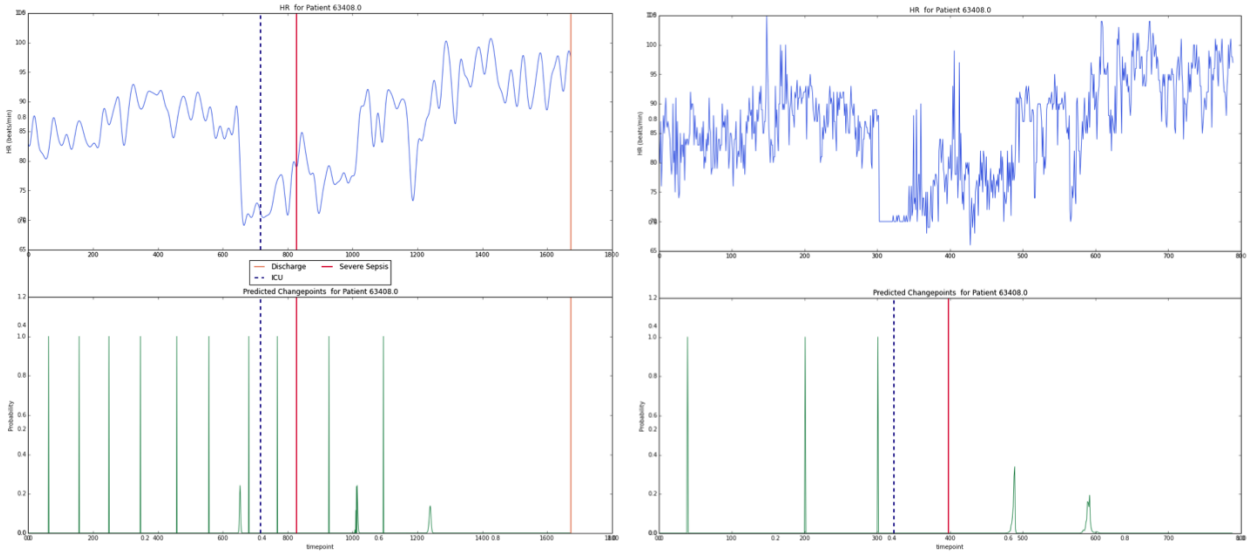

Figure 26: Patient 63408.

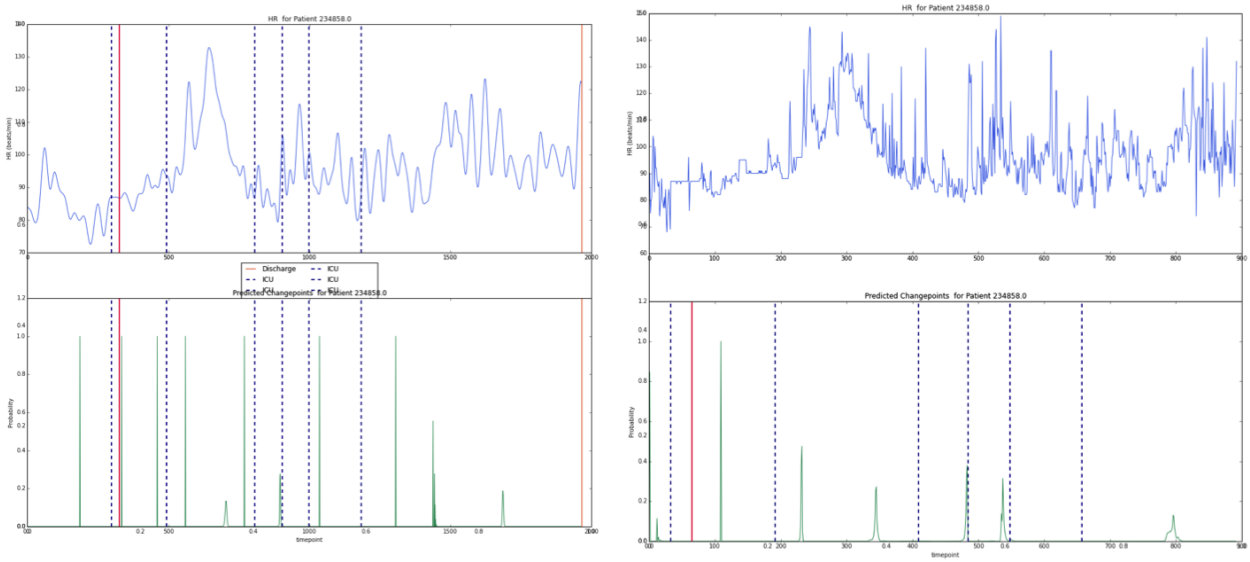

Figure 27: Patient 234858.

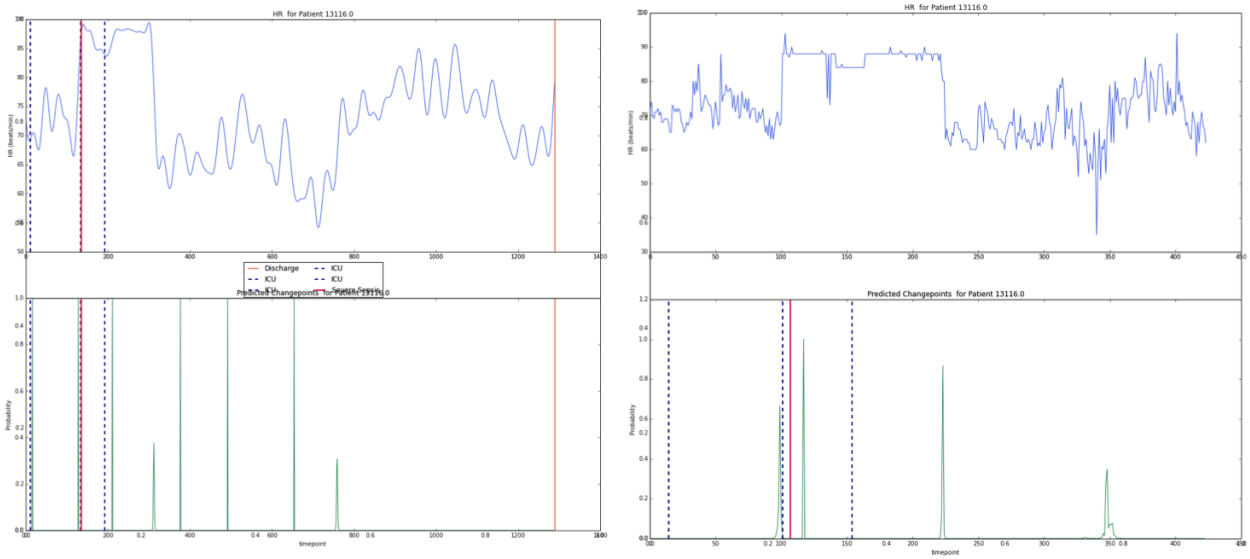

Figure 28: Patient 13116.

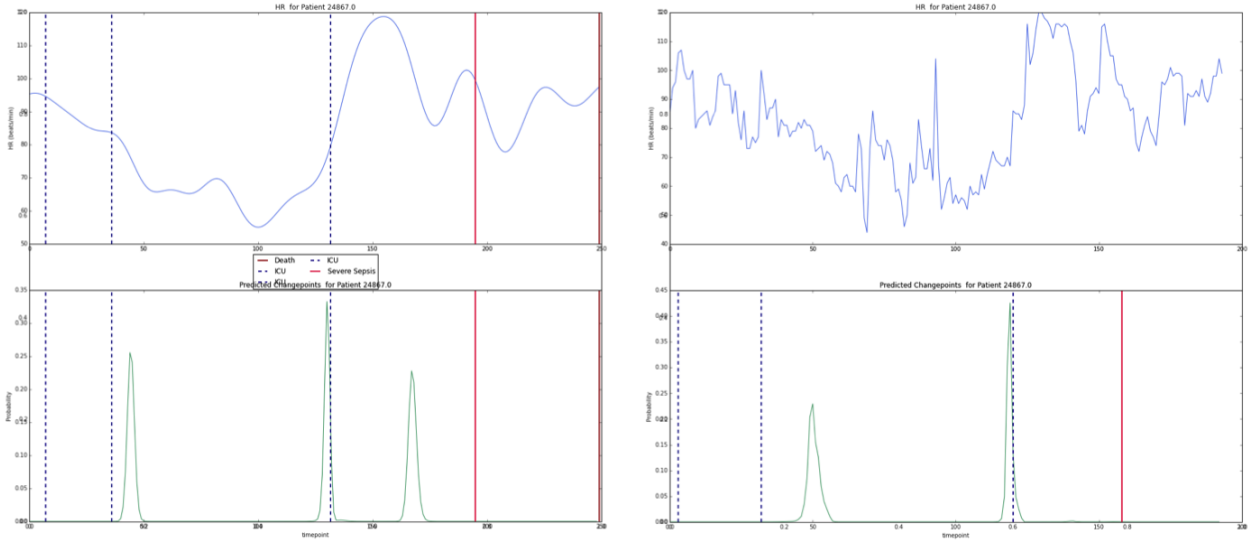

Figure 29: Patient 24867.

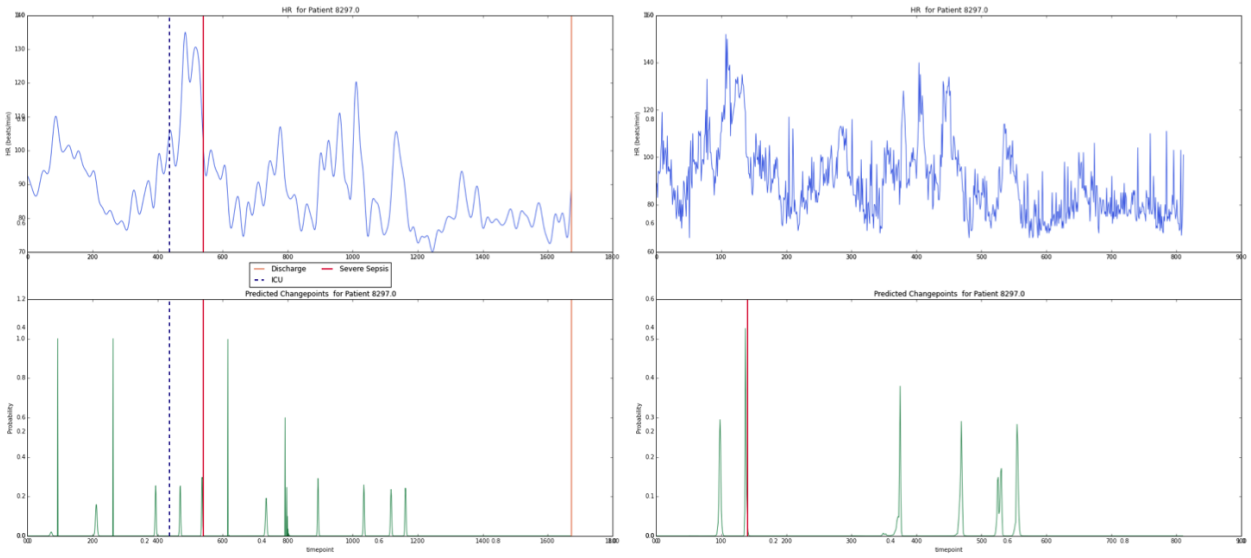

Figure 30: Patient 8297.

C.

C. Sample Python Script Example script for generating and plotting offline change point detection (i.e. Fig 11).

```
import pandas
import pylab
import csv
import numpy as np
import matplotlib.pyplot as plt
import os
import cProfile
from functools import partial
import pickle

from pandas import *

# import processed data to compare and debug
df_demogr = read_csv('/data/penn_new/lifangc/new_demogr_268000.csv')
df_bloodc = read_csv('/data/penn_new/lifangc/new_bloodc_268000.csv')
df_bedunit = read_csv('/data/penn_new/lifangc/new_bedunit_268000.csv')
df_diagnosis = read_csv('/data/penn_new/lifangc/new_diagnosis_268000.csv')
ID_ARRAY = unique(df_diagnosis.PAN)
PATIENT_NUM = ID_ARRAY.shape[0]

df_event = read_csv('/data/penn_new/lifangc/new_event_268000.csv')
df_primary = read_csv('/data/penn_new/lifangc/new_primary_268000.csv')
```

```
df_sepsis_flag = read_csv('/data/penn_new/lifangc/new_sepsis_flag_268000.csv')
df_cardiac_flag = read_csv('/data/penn_new/lifangc/new_cardiac_flag.csv')

differences = []
indices = []

patient_array = read_csv('/patient_list.csv')

# for all sample patients
for k in patient_array:
    target_patient = k
    test_id = target_patient

    # Case 1: For non-imputed data
    # load heart rate data for patient k
    for i in range(0, 1):
        test_id = target_patient
        admit_date = df_demogr.ADM_DATE[df_demogr.PAN == test_id].values[0]

        csv_file_name = '/data/penn_new/lifangc/single/patient_' + ('%06d' % np.nonzero(ID_ARRAY == test_id)) + '.csv'
        df_raw_data = read_csv(csv_file_name)
        pan = unique(df_raw_data.J00_PAN)
        assert not((len(pan) > 1)or(pan[0] != test_id))

```

```
47        feature_list = read_csv('/data/penn_new/lifangc/new_feature_list.
  csv')
48        for index in range(3, 4): # plot heart rate only
49            covariate = df_raw_data[df_raw_data.columns[index]]
50            time = df_raw_data.J01_TAKEN_DATE
51
52            # adjust time units to be in days
53            time = (time - admit_date)/(24.0*3600.0)
54
55            # array of timepoints for observations
56            x_val = time[np.isnan(covariate.values) == False].values
57            x = np.atleast_2d(x_val).T
58
59            # array of heart rates corresponding to observation times
60            y_val = covariate[np.isnan(covariate.values) == False].values
61            y = np.atleast_2d(y_val).T
62
63     # offline change point detection
64     # assume unifor prior over length of sequence
65
66     # Case 2: For imputed data
67     # load the list of PAN contained in the folder
68     pan_list = pickle.load(open('/memex/hebeck/impute/PAN_list_VAR_01.out'
  , 'r'))
69     # Load the imputed data value
70     d = pickle.load(open('/memex/hebeck/impute/PAN_' +  ('%06d' % test_id)
     + '_VAR_01.out', 'r'))
```

```
71
72      # the stored data is in a dictionary format
73      # data['time']: the resampled time, every half hour since patient
        admission (before discharge)
74      # data['imputed_value']: the predictive imputed data value
75      # data['uncertainty']: the predictive standard deviation
76      # plt.plot(data['time'], data['imputed_value'])
77
78      # Case 1: use y_val, Case 2: use d['imputed_value'] as data
79      # data = np.array(y_val, dtype =np.float64)
80      data = np.array(d['imputed_value'], dtype=np.float64)
81
82      # Offline change point detection
83      # Q = log-likelihood of data for partition from t to end of series
84      # P = matrix of P[t,s] log-likehood of set of observations from t to s
        ,
85      #        given no changepoints between t and s
86      # Pcp = matrix of log-likehood that ith changepoint occurs at t for
        Pcp[i,t]
87      # prior_func = assume constant uniform prior for likelihood of a
        changepoint given
88      #        distance from the last one
89      Q, P, Pcp = offline_changepoint_detection(data, partial(const_prior, l
        =(len(data)+1)), gaussian_obs_log_likelihood, truncate=-20)
90
91      # Plot Results
92      # vertical lines correspond to clinical events
```

```
 # note for imputed data: divide 36000 by 2 since unit of input is 0.5 hours

 # Get admission/discharge time as reference
 admit_date = df_demogr.ADM_DATE[df_demogr.PAN == target_patient].values[0]
 discharge_date = df_demogr.DISCHARGE_DATE[df_demogr.PAN == target_patient].values[0]

 # Plot outcome (death or not)
 death_flag = df_demogr.DEATH_FLAG[df_demogr.PAN == target_patient].values[0]

 # Plot bed history
 bed = df_bedunit[df_bedunit.PAN == target_patient]
 bed_num = bed.shape[0]
 if(bed_num > 0):
     bed_date = df_bedunit.IN_DATE[df_bedunit.PAN == target_patient].values
     bed_flag = df_bedunit.ICU_FLAG[df_bedunit.PAN == target_patient].values
     bed_type = df_bedunit.UNIT_MASTER_CARE_TYPE[df_bedunit.PAN == target_patient].values

 # Plot Sepsis History
 sepsis_flag = df_sepsis_flag.FINAL_SEPSIS_FLAG[df_sepsis_flag.PAN == target_patient].values
```

```
112     if(sepsis_flag == 1):
113         sepsis_date = df_sepsis_flag.FINAL_SEPSIS_DATE[df_sepsis_flag.PAN
    == target_patient].values[0]
114
115     # Plot Severe Sepsis History
116     severe_sepsis_flag = df_sepsis_flag.FINAL_SEVERE_SEPSIS_FLAG[
    df_sepsis_flag.PAN == target_patient].values[0]
117     if(severe_sepsis_flag == 1):
118         severe_sepsis_date = df_sepsis_flag.FINAL_SEVERE_SEPSIS_DATE[
    df_sepsis_flag.PAN == target_patient].values[0]
119
120     ## plot vertical lines for events
121     # Get admission/discharge time as reference
122     admit_date = df_demogr.ADM_DATE[df_demogr.PAN == target_patient].
    values[0]
123     discharge_date = df_demogr.DISCHARGE_DATE[df_demogr.PAN ==
    target_patient].values[0]
124
125     # Plot outcome (death or not)
126     death_flag = df_demogr.DEATH_FLAG[df_demogr.PAN == target_patient].
    values[0]
127
128     # Plot bed history
129     bed = df_bedunit[df_bedunit.PAN == target_patient]
130     bed_num = bed.shape[0]
131     if(bed_num > 0):
```

```
132         bed_date = df_bedunit.IN_DATE[df_bedunit.PAN == target_patient].
    values
133         bed_flag = df_bedunit.ICU_FLAG[df_bedunit.PAN == target_patient].
    values
134         bed_type = df_bedunit.UNIT_MASTER_CARE_TYPE[df_bedunit.PAN ==
    target_patient].values
135
136     # Plot Sepsis History
137     sepsis_flag = df_sepsis_flag.FINAL_SEPSIS_FLAG[df_sepsis_flag.PAN ==
    target_patient].values
138     if(sepsis_flag == 1):
139         sepsis_date = df_sepsis_flag.FINAL_SEPSIS_DATE[df_sepsis_flag.PAN
    == target_patient].values[0]
140
141     # Plot Severe Sepsis History
142     severe_sepsis_flag = df_sepsis_flag.FINAL_SEVERE_SEPSIS_FLAG[
    df_sepsis_flag.PAN == target_patient].values[0]
143     if(severe_sepsis_flag == 1):
144         severe_sepsis_date = df_sepsis_flag.FINAL_SEVERE_SEPSIS_DATE[
    df_sepsis_flag.PAN == target_patient].values[0]
145
146     x_input = np.exp(Pcp).sum(0)
147     max_x = max(x_input)
148     min_diff = 1000000000
149     dif = 0
150
151     for i in range(len(x_input)):
```

```
        if x_input[i] >= 0.1:
            dif = (severe_sepsis_date - admit_date)/(3600.0/2.0) - (i)
            if (abs(dif) < abs(min_diff)):
                min_diff=dif
    differences.append(min_diff)
    indices.append(target_patient)
print str(patient_array)
print str(differences)
print str(indices)
```