**EARLY DETECTION OF AGGRESSIVE CANCER USING LONGITUDINAL BIOMARKER MEASUREMENTS**

Maggie Wang1, Sharon S. Hori2,3,4

1Biomedical Engineering and Computer Science, Johns Hopkins University, Baltimore MD

2Radiology, Stanford University School of Medicine, Stanford CA

3Canary Center at Stanford, Stanford University School of Medicine, Stanford CA

4Molecular Imaging Program at Stanford (MIPS), Stanford CA

**INTRODUCTION**

In cancer diagnosis, timing is of the essence. As the cancer progresses, tumors cancer cells become increasingly heterogeneous and may metastasize to other portions of the body, rendering effective treatment more difficult. Tumor cells release biomarkers such as nucleic acids and proteins [1] into the blood. It is possible to detect cancer earlier by analyzing the cancer biomarker levels in a blood sample. Most clinical biomarkers are released by both tumor cells and by healthy cells. For a given one of these biomarkers, patients who are cancer-free will have some healthy baseline biomarker level in their blood. If the patient proceeds to develop cancer, this biomarker level increases from the patient’s healthy baseline at a rate proportional to tumor growth [2-3]. To accurately classify a patient’s cancer status, we must distinguish between a healthy baseline biomarker level and an abnormally high biomarker level indicative of cancer. One of the most widely used cancer biomarkers for patient screening is prostate-specific antigen (PSA), and the current clinical standard for determining cancer status using PSA is by comparing a patient’s PSA level to a threshold value of 4 ng/mL [4]. This threshold value is assumed to be the average healthy level across all patients. However, various factors other than cancer, including age, may cause a patient’s PSA healthy baseline level to increase above this threshold value, thus leading to a high false positive rate. We may encounter similar problems with other cancer biomarkers, where different patients have different healthy baseline biomarker levels. Comparing their biomarker levels to an average population healthy baseline becomes invalid, leading to inaccurate diagnoses. In addition to inter-individual variation in the definition of “healthy”, intra-individual variation [5] resulting from assay error and a patient’s own biological variation add noise to the observed biomarker measurements, making it even more challenging to differentiate between healthy and abnormal measurements.

A longitudinal inspection of biomarker levels (examining a patient’s entire blood sampling history, beginning when they are in a healthy state) can aid us in determining patient-specific healthy baseline levels and in quantifying how much variability inherently exists in the patient’s biomarker measurements. Once personal healthy baselines have been established, we can then determine whether a patient may have cancer by detecting when biomarker measurements begin to deviate from the patient’s own healthy baseline, outside of the range explainable by noise.

**MATERIALS AND METHODS**

Longitudinal biomarker measurements were simulated for *n*h=50 healthy patients and *n*c=50 cancer patients over a period of 100 days. For each patient, biomarker measurements were normalized to a specific (“personalized”) healthy baseline biomarker concentration, based on an initial *d* days of longitudinal biomarker data, using one of four methods: 1) mean subtraction, 2) z-score normalization, 3) autoregressive forecast with an expanding window, and 4) autoregressive forecast with a shifting window. Cancer status was then classified based on normalized biomarker levels using 10-fold cross validation and two supervised machine learning techniques: 1) *k*-nearest neighbors and 2) thresholding.

*Simulation of Longitudinal Biomarker Measurements* (Fig. 1.1)

The first step of simulation was to generatenoise-free biomarker measurement trajectories for *n*hpatients who remain healthy (free of cancer) and *nc* patients who develop cancer on day 500. We define the “noise-free” measurement to be the true biomarker concentration, not affected by assay or technical error.All patients are initially assumed to be in a healthy (non-cancer) state, with cancer onset assumed for *n*c patients who go on to develop cancer at *t*c = 500 day.

For healthy patients, the true baseline biomarker concentration is assumed constant, i.e.,

For each patient *i*, was chosen from a normal distribution with an adjustable variance, based on the desired level of population baseline variation in the simulation. For the subset of patients who develop cancer, the baseline biomarker concentration is

For each patient, was chosen from a normal distribution with an adjustable variance, and and were selected from a uniformly distributed range of growth and decay rates, , . The aggressive and healthy biomarker trajectories were generated for , simulating 1000 daily measurements.

The second step of simulation was to add noise to the true biomarker measurements to simulate the observed biomarker measurements. In this study, we examined 3 noise models as follows.

In the *constant error* model, each measurement in a signal received roughly the same magnitude of noise, regardless of the magnitude of the measurement itself. For each simulated patient, we created a noise array of 1000 values selected from a normal distribution, , then performed an element-wise sum of the noise array and the noise-free simulated measurements to generate noisy longitudinal measurements. The noise level was characterized by the value of .

In the *standardized fractional error* model, each measurement in a signal received noise proportional to the magnitude of the measurement. For each 1000-day simulated signal, we created an array of 1000 values selected from a normal distribution, ; this formed an array of percentages that was used to calculate proportional noise. We performed element-wise multiplication of the percentage array with the noise-free simulated signal to generate an array of fractional noise. We then added this array to the original noise-free simulated signal to generate a noisy signal. The noise level was characterized by the value of .

In the *fractional error* model, each measurement in a signal again received noise proportional to the magnitude of the measurement. For each 1000-day simulated signal, we created an array of 1000 values selected from a normal distribution, . Note that the normal distribution was not centered about 0. We multiplied each element in the newly generated array by either 1 or -1 and performed element-wise multiplication of the array with the noise-free simulated signal to generate an array of fractional noise. We then added this array to the original noise-free simulated signal to generate a noisy signal. The noise level was characterized by the value of .

The third step of simulation was to adjust how frequently measurements were made in the simulated signals. When the signals were initially generated, they each consisted of 1000 daily measurements. The sampling frequency was altered by changing the number of days between measurements; the same frequency change was applied to all simulated signals. (All signals shared the same sampling frequency). For example, if the sampling frequency was changed to 10 days, then all signals were downsampled by retaining every 10th measurement in the signal and removing all other intermediate measurements. Once downsampled, the signals had a measurement every 10 days for 1000 days, amounting to 100 measurements total.

The final step of simulation was to change how long patients were monitored, i.e. how much of their biomarker history had been observed. When the signals were initially generated, all signals consisted of daily measurements for 1000 days. The length of observation was shortened by removing the last *n* measurements of the signal, effectively truncating or slicing off a segment from the right-hand-side of the signal; the same length change was applied to all simulated signals. (All signals shared the same length of observation).

*Normalization to Patient-Specific Baselines* (Fig. 2.1)

Simulated biomarker measurements were normalized with four different methods.

(i) z-score Normalization

We found the average and standard deviation of the first 100 days of measurements; the average was the patient’s healthy baseline. Let and be the mean and standard deviation, respectively. Then for an observed measurement at time *t*, the normalized measurement was calculated as

(ii) Average Subtraction

We found the average of the first 100 days of measurements, which we called the patient’s healthy baseline. Then for an observed measurement at time *t*, the normalized measurement was calculated as

(iii) Autoregressive Forecast [6]

The concept of autoregression comes from the idea that the current time series value depends on previous values of the time series, meaning the observations are not independent and identically distributed. For example, if the temperature outside was high yesterday and the day before, we might expect it to be high again today. Mathematically, we can express a value in terms of p previous values . from the time series,

where are the coefficients explaining the dependence of on previous values is a random error term that captures random “shocks” to the system at each time t. This model is called an autoregressive model of order p, denoted AR(p), and is a form of linear regression in which the explanatory variables are the p past observations of the time series. The past observations are often called lag variables. AR(1) is a model in which each value of the time series depends on one lag variable, namely the directly preceding value. AR(2) is a model where each value is written in terms of the two directly preceding values. Before fitting the model, it is important to determine if values in the time series indeed display dependence on previous values; if so, we must then decide on an appropriate model order p. We can characterize how observations in the time series are related to one another using autocovariance. For an observation and a lag variable , the autocovariance is

Note that is the variance of the time series. The autocorrelation coefficient between an observation and a lag variable , can be calculated as

The closer is to 1, the greater the dependency between the observation and the lag. Sometimes, the dependency between the observation and the kth lag might actually be a result of a chain of dependencies, where perhaps the observation actually only depends on the first lag, but the first lag then depends on the its own first lag, and so on and so forth. To accommodate this snowball effect, we can compute partial autocorrelation coefficients, which removes the interdependencies between lags. If the time series is stationary, then the partial autocorrelation coefficients should decrease rapidly as k becomes large. We can then select a model order p, where after the first p lags, the value of the partial autocorrelation coefficient becomes negligible.

Different methods exist to fit an AR(p) model to a set of observations. The Yule-Walker equations are commonly used. Once the parameter values and have been computed, it is possible to make forecasts using the model. If the model is fit on observations that have been made up to time t, then the forecasted value for time t+1 can be estimated with the model equation and fitted parameter values and . This is known as a one-step-ahead forecast. It is possible to make an n-step ahead forecast by making repeated one-step-ahead forecasts, using previously forecasted values as lag variables. The forecasts have corresponding confidence intervals related to the errors/residuals of the model’s fit to the original observed time series.

For an *expanding window* forecast, after fitting on the first 100 days of measurements and making a one-step-ahead forecast, we refitted the AR model on the first *101* days of measurements and made another one-step-ahead forecast. We then fit the model on the first 102 days measurements, and so on. Each time a new measurement was observed, we added it to the set of previously observed measurements and refit the AR model. In this way, the entire history of measurements was incorporated into the model.

For a *shifting window* forecast, after fitting on the first 100 days of measurements and making a one-step-ahead forecast, we discarded the first measurement, refit the AR model on measurements from day 2 to 101, and made another one-step-ahead forecast. We then discarded the measurement from day 2 and fit the model to measurements from day 3 to 103, and so on, so that the window used to fit the AR model was always 100 days long.

To determine whether a newly observed value was abnormal, we compared the observed value to the model-forecasted value. We then normalized a patient to their baseline using this observed-to-forecasted distance metric. Instead of examining the observed biomarker measurements themselves, we examined the distance between the observed measurement and the forecasted value. Let and be the observed and predicted measurements at time *t*, respectively. Let be the confidence in the AR prediction. Then the normalized measurement at time *t* was defined as

To normalize an entire trajectory, we performed the above calculation at all time points.

(The autoregressive model was implemented in Python using the *statsmodels.tsa.arima\_model* module).

*Classification with k-Nearest Neighbors (k-NN)* [7]

(i) Unweighted *k*-NN

Given an input patient’s longitudinal biomarker history the algorithm finds the k- nearest neighbors to the input patient in a set of previously acquired longitudinal biomarker measurements belonging to patients with known cancer statuses. A “vote” is then cast, and the input signal is labeled with the majority label of the k-nearest neighbors. The simplest metric to quantify signal proximity is Euclidean distance, where for an unlabeled input signal and a signal in the labeled set, the distance *d* between the two is defined as

(ii) Weighted k-NN

The k – nearest neighbors are weighted or “ranked” according to their distance from the unlabeled input signal. Closer neighbors are given larger weights, and neighbors that are farther away are given smaller weights. More specifically, the weight for the *i*th neighbor out of the k- nearest neighbors is the inverse of its distance from the input signal.

Let be the weights of the nearest neighbors belonging to the aggressive class and be the weights of the nearest neighbors belonging to the non-aggressive class, where , the total number of nearest neighbors. If

the input signal is classified as aggressive with *confidence level* . Intuitively, this means that the closer the input signal is to the aggressive nearest neighbors, the more confident we are that the input signal is also aggressive. Similarly, if

the input signal is classified as non-aggressive with confidence .

If we have a set of existing patients, each with a set of longitudinal biomarker measurements, whose statuses (aggressive, non-aggressive) are known, then for a new patient with a set of longitudinal biomarker measurements, we can apply a k-nearest neighbor approach to classify the new patient’s cancer status.

For this study, the weighted *k*-NN method was used and applied in two different manners. (The nearest neighbor query was implemented in Python using the *sklearn.neighbors.KDTree* module). The first approach was to use fixed observation lengths and the second involved dynamically increasing observation lengths, mimicking a patient returning to the clinic for new blood samples. We were first interested in considering the effect of parameter values – noise, sampling frequency, and observation span – on classification accuracy (Figs. 1.2-1.9). Sets of 100 simulated patients, 50 cancerous and 50 healthy, were generated; cancer onset occurred on the very first measurement for patients with cancer. In each set, all patients had the same healthy baseline value of 1 ng/mL and some fixed combination of other parameter values; the same sampling frequency and observation span were always shared across the patient cohort. 10-fold cross validation was then used to assess classification performance. The patients were split into a training set containing 90% of the patients and a testing set containing the remaining 10% of patients. The split was made such that the ratio of healthy to cancerous patients in both the training and testing set was 50-50. Because the data were simulated, true patient cancer statuses were known for all patients. For each patient in the testing set, the *k* nearest neighbors in the training set were identified, and the test patient was classified according to the status distribution of the neighbors. The classification was made correctly if the classified label matched the test patient’s true label. The train-test split was repeated 10 times. The *accuracy* of the algorithm was calculated as the number of correctly classified test signals divided by the total number of test signals. *Sensitivity* (also known as recall) was calculated as the number of test signals classified correctly as aggressive divided by the total number of aggressive signals in the test set (. *Specificity* was calculated as the number of test signals classified correctly as non-aggressive divided by the total number of non-aggressive signals in the test set (. *Precision* was calculated as the number of test signals classified correctly as aggressive divided by the total number of test signals classified as aggressive (. The *F-score* was a weighted average of recall and precision (.

We then wanted to compare the classifier’s performance on measurements normalized using the different normalization methods. Again, sets of 100 simulated patients, 50 cancerous and 50 healthy, were generated; cancer onset occurred on day 500 for all patients with cancer (this was to allow determination of healthy baseline using measurements prior to day 500). Various combinations of baseline variation and noise level were tested. The population was simulated to have either a mean healthy baseline of 10 ng/mL and 0 ng/mL standard deviation in baseline level (Figs. 2.2-2.4) or 5 ng/mL standard deviation in baseline level (Figs. 2.5 – 2.7). Noise was applied with noise model (iii) at levels of 5% (Figs. 2.2, 2.5), 15% (Figs. 2.3, 2.6), or 30% (Figs. 2.4, 2.7). All patients were sampled every 10 days. For each simulated set bearing a different combination of baseline variation and noise level, we examined classification performance for different observation spans, i.e. we truncated the longitudinal biomarker measurements of each patient in the simulated set to the same observation length and performed 10-fold cross-validation.

In the dynamic *k*-NN approach, a set of 100 patients, 50 cancerous and 50 healthy, was generated with 10-day sampling frequency, 1000 days of observation, 15% noise, and healthy baseline of 10 ± 5 ng/mL. The set was normalized using one of the four normalization methods. The normalized set was then split into a testing and training set as detailed above. For each patient in the testing set, we began by classifying them using only their first biomarker measurement. If they were classified as healthy, we moved onto their next biomarker measurement and classified them using their first two biomarker measurements. This process was repeated until they were classified as having cancer with a high confidence level or until they reached the end of the 1000 days of observation, at which point they would be assigned a healthy status. The time of classification in days post-onset of cancer was noted. We compared the performance of this dynamic approach using an 80% confidence level threshold and 95% confidence level threshold (Fig 3.1)

*Classification with Thresholding*

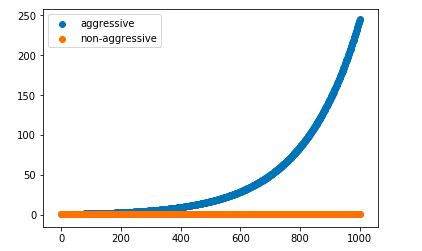
To classify with thresholding, a patient was assigned a cancerous status if their normalized biomarker level crossed a certain threshold value. As in the *k*-NN classification method, we applied thresholding in two different manners: one with fixed observation lengths and the second with dynamic incorporation of new measurements.

In the fixed length approach, a set of 100 patients, 50 cancerous and 50 healthy, was generated with a certain observation span, 10-day sampling frequency, 15% noise, and a healthy baseline of 10 ± 5 ng/mL. A second set was simulated with no variation in healthy baseline. The sets were normalized using z-score and expanding window methods. We generated a collection of threshold values to test, ranging from a very small threshold (which classified all trajectories as cancerous) to a very large threshold (which classified all trajectories as healthy). Again, the set was split into a 90-10 training-testing set. For each threshold, we classified each patient in the training set using the last observed measurement in the patient’s observation span. If the measurement was larger than the threshold, we classified the patient as cancerous; if it was smaller, we classified the patient as healthy. We then calculated the F-score of the classification using this threshold. Once all thresholds were tested, we selected the threshold that gave the largest F-score, used this “optimal” threshold to classify the patients in the testing set, and calculated the sensitivity, specificity, and F-score of the classification on the testing patients. The threshold selection and classification process was repeated for 10 training-testing splits.

We applied the dynamic approach to the same set of simulated patients used in the *k-*NN dynamic approach. The normalized set was split into a testing and training set ten times, as detailed above. A large range of threshold values was generated. For a given threshold value, we examined the first biomarker measurement for each test patient; if the measurement crossed the threshold, we classified the patient as having cancer and noted the time that the threshold was crossed (time of cancer detection). If the measurement was below the threshold, we examined the patient’s next biomarker measurement. We continued to progress through the patient’s biomarker measurements until either the threshold value was crossed or until we reached the end of the 1000 days of observation, at which point the patient would be classified as healthy. For each threshold value, we calculated the average time the threshold was crossed, and the sensitivity and specificity of classification (Fig 3.2).

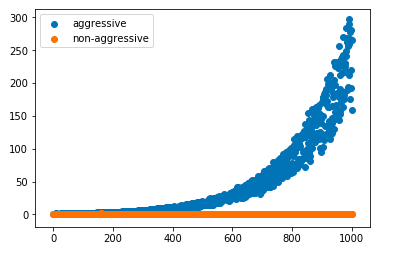
**Figure 1.1 Biomarker Measurement Simulation Summary**

The plots depict one example of an aggressive trajectory (blue) and one example of a non-aggressive trajectory (orange)

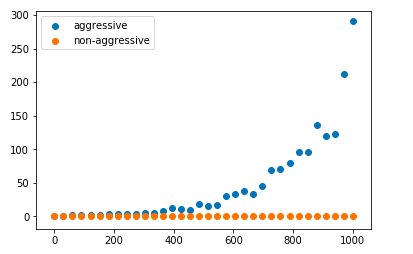
****

1. Generate a sequence of 1000 noise-free (true value)

biomarker measurements

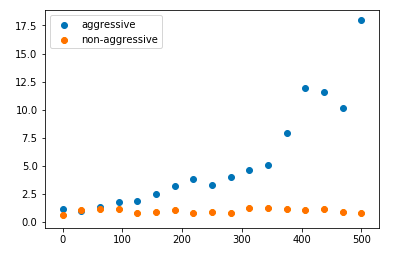


2. Add Gaussian noise to the biomarker measurements

****

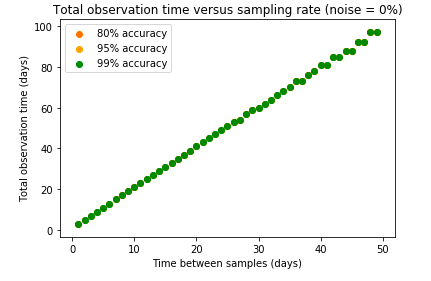
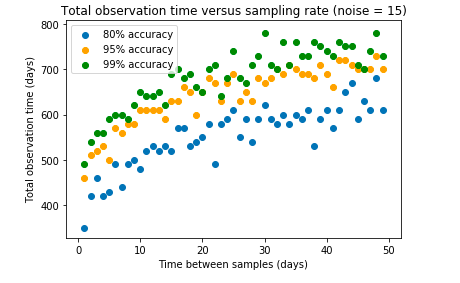
3. Change the sampling frequency by increasing the

time between samples.

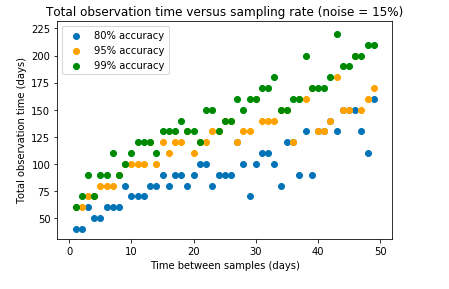
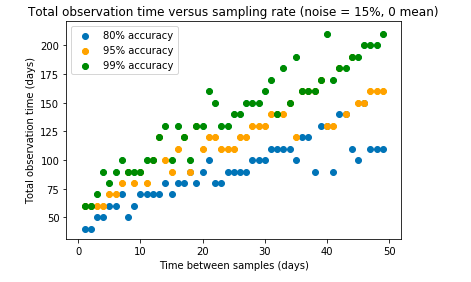
****

4. Change the length of observation (green window)

(A) (B)

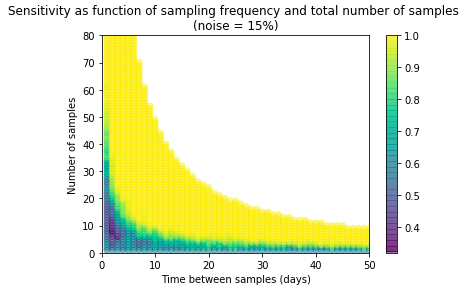
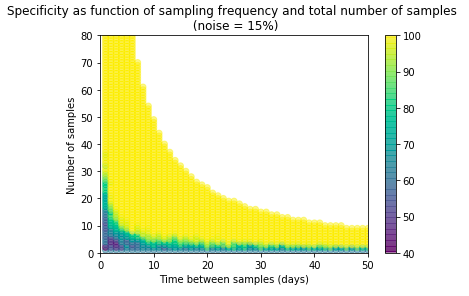


(C) (D)

****

**Figure 1.2**. **Given a sampling frequency, find the observation length needed to achieve 80%, 95%, and 99% classification accuracy.** As measurements become sparser (increasing x-axis values), the necessary observation length increases. (A) Observation length v. sampling frequency on signals with no noise. (B) Noisy signals generated with constant noise model, noise = 15. (C) Noisy signals generated with 15% noise centered around 0. (D) Noisy signals generated with 15% noise.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Time Between Observations (days)** | **Number of Observations** | **Length of Observation (days)** | **Sensitivity** | **Specificity** |
| 1 | [45.8, 1.6] | [ 44.8 1.6] | [ 0.968 0.0227684] | [ 1. 0.] |
| 2 | [27.0, 0.4] | [ 52. 0.8] | [ 0.98 0.01391402] | [ 1. 0.] |
| 3 | [20.8, 0.7] | [ 59.4 2.1] | [ 0.972 0.01897367] | [ 1. 0.] |
| 4 | [16.0, 0.6] | [ 60. 2.5] | [ 0.96 0.02529822] | [ 0.992 0.00758947] |
| 5 | [13.8, 0.6] | [ 64. 3.0] | [ 0.972 0.01897367] | [ 0.996 0.00379473] |
| 6 | [11.4, 0.2] | [ 62.4 1.3] | [ 0.964 0.02656313] | [ 1. 0.] |
| 7 | [10.8, 0.3] | [ 68.6 2.3] | [ 0.968 0.0227684] | [ 1. 0.] |
| 8 | [9.8, 0.4] | [ 70.4 3.5] | [ 0.968 0.0227684] | [ 0.988 0.00885438] |
| 9 | [9.4, 0.3] | [ 75.6 3.2] | [ 0.976 0.01770875] | [ 0.996 0.00379473] |
| 10 | [8.8, 0.3] | [ 78. 3.3] | [ 0.968 0.0227684] | [ 0.992 0.00758947] |
| 11 | [7.8, 0.4] | [ 74.8 4.8] | [ 0.968 0.0227684] | [ 0.98 0.00632456] |
| 12 | [7.8, 0.2] | [ 81.6 2.1] | [ 0.98 0.01644384] | [ 0.996 0.00379473] |
| 13 | [7.8, 0.3] | [ 88.4 4.4] | [ 0.984 0.01517893] | [ 1. 0.] |
| 14 | [7.0, 0.3] | [ 84. 4.0] | [ 0.968 0.0227684] | [ 0.976 0.00619677] |
| 15 | [6.8, 0.2] | [ 87. 2.7] | [ 0.972 0.02403331] | [ 0.992 0.00758947] |
| 16 | [6.8, 0.3] | [ 92.8 5.4] | [ 0.972 0.02150349] | [ 0.996 0.00379473] |
| 17 | [5.8, 0.3] | [ 81.6 5.7] | [ 0.972 0.01897367] | [ 0.976 0.01465093] |
| 18 | [5.8, 0.2] | [ 86.4 3.2] | [ 0.976 0.01517893] | [ 1. 0.] |
| 19 | [5.8, 0.2] | [ 91.2 3.4] | [ 0.98 0.01391402] | [ 0.992 0.00505964] |
| 20 | [5.6, 0.2] | [ 92. 4.4] | [ 0.98 0.01644384] | [ 0.996 0.00379473] |
| 30 | [4.4, 0.2] | [ 102. 6.6] | [ 0.978 0.01834121] | [ 0.996 0.00379473] |
| 40 | [3.9, 0.1] | [ 116. 5.4] | [ 0.99 0.00822192] | [ 0.998 0.00189737] |
| 50 | [3.2, 0.2] | [ 110. 8.9] | [ 0.988 0.01011929] | [ 1. 0.] |
| 60 | [3.0, 0.0] | [ 120. 0.] | [ 0.99 0.00948683] | [ 1. 0.] |
| 70 | [3.0, 0.0] | [ 140. 0.] | [ 0.998 0.00189737] | [ 1. 0.] |
| 80 | [2.8, 0.2] | [ 144. 14.3] | [ 0.992 0.00632456] | [ 0.99 0.00657951] |
| 90 | [2.7, 0.2] | [ 153. 18.4] | [ 0.992 0.00505964] | [ 0.994 0.0056921] |
| 100 | [2.2, 0.2] | [ 120. 17.9] | [ 0.98 0.01897367] | [ 1. 0.] |
| 150 | [2.0, 0.0] | [ 150. 0.] | [ 0.988 0.00885438] | [ 0.996 0.00379473] |
| 200 | [2.0, 0.0] | [ 200. 0.] | [ 0.988 0.0113842] | [ 1. 0.] |
| 250 | [2.0, 0.0] | [ 250. 0.] | [ 0.992 0.00758947] | [ 1. 0.] |
| 300 | [2.0, 0.0] | [ 300. 0.] | [ 1. 0.] | [ 1. 0.] |



(B)

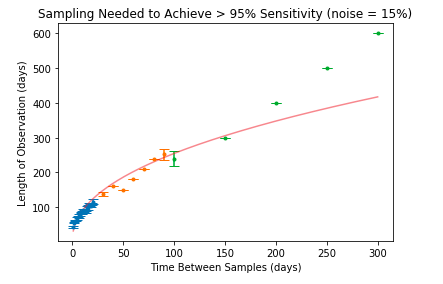
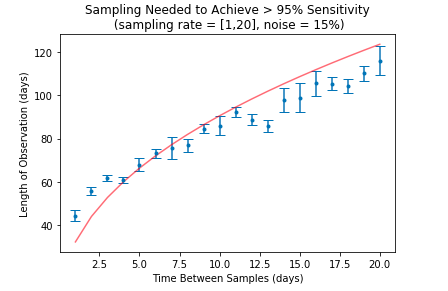
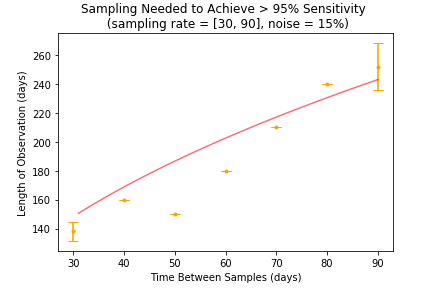
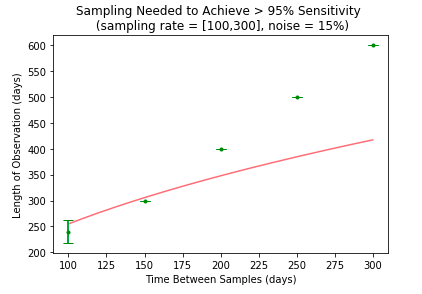
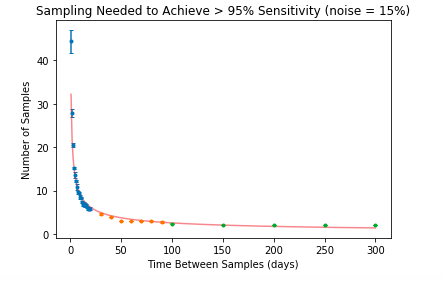
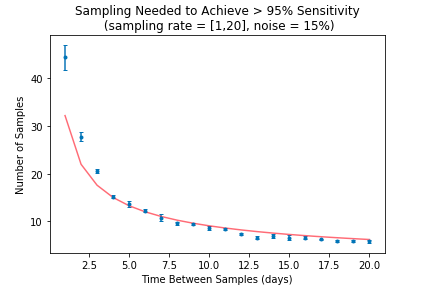
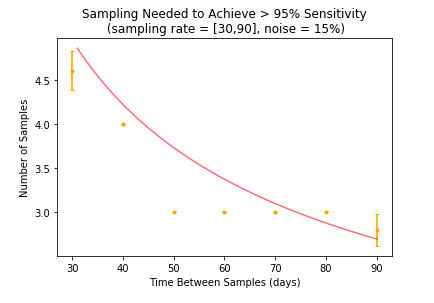
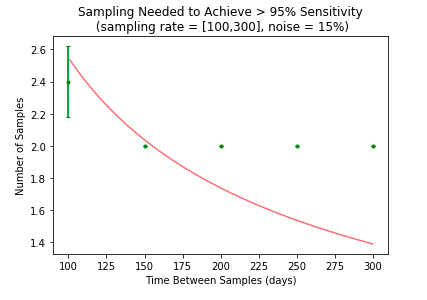
(A)

**Table 1. Sampling Needed to Achieve Greater than 95% Sensitivity (noise = 15%)**

**Figure 1.3 (previous page). Three-way Analysis of Effect of Sampling Frequency and Total Number of Samples on Algorithm Performance. (A,B)** Sensitivity and specificity at different sampling frequencies and different total sample numbers. The color of the points in the plot reflect the sensitivity/specificity value at each [sampling frequency, total number of samples] pair.

**Table 1. (previous page)** The number of samples and length of observation needed to achieve greater than 95% sensitivity at various sampling frequencies of interest, ranging from daily to every 300 days. Length of observation was calculated as the product of the number of samples and the sampling frequency. Five sets of 100 signals (50 aggressive, 50 non-aggressive) were generated, and the same sampling analysis was performed on each. The table shows the averaged results from the five iterations.

**Figure 1.4 (next page) (A1)** “Number of Observations” column versus “Time Between Observations” column from Table 1. The data were log linearized and fit using least-squares regression. The fitted curve is shown in red. **(A2,3,4)** Zoomed in views of sections along the x-axis in A1. **(B1)** “Length of Observation” column versus “Time Between Observations” column from Table 1. Since the length of observation was calculated as the product of the time between observations, the x-axis in A1, and the number of observations, the y-axis in A1, the points were fitted with the curve . **(B2,3,4)** Zoomed in views of sections along the x-axis in B1.



(B4)

(B3)

(B2)

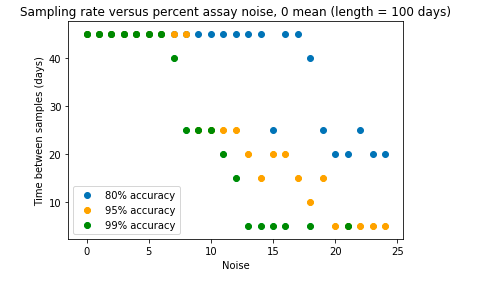
(B1)

(A4)

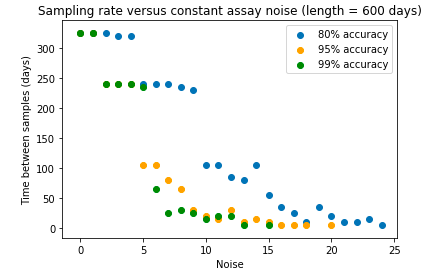
(A3)

(A2)

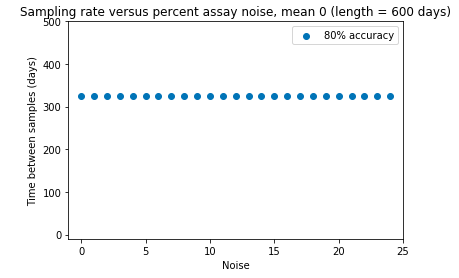
(A1)

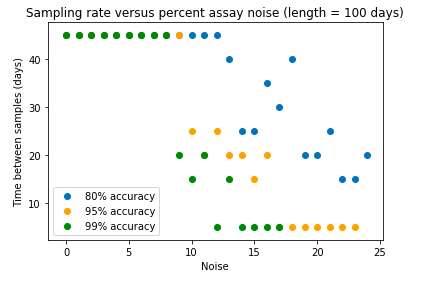
A)

**Figure 1.5. Given a level of noise in the signals, find the sampling frequency needed to achieve 80%, 95%, and 99% accuracy.** As noise increases, samples must become more frequent. (A) Noisy signals generated with constant noise model. All signals observed for 600 days. (B) Noisy signals generated with percent noise centered around 0. All signals observed for 100 days. (C) Noisy signals generated with percent noise. All signals observed for 100 days. (D) Noisy signals generated with percent noise centered around 0. All signals observed for 600 days.

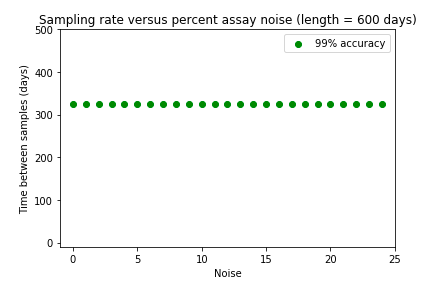


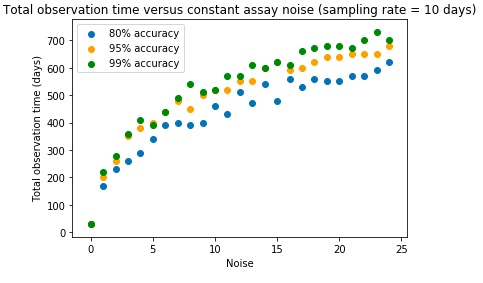
(B) (D)



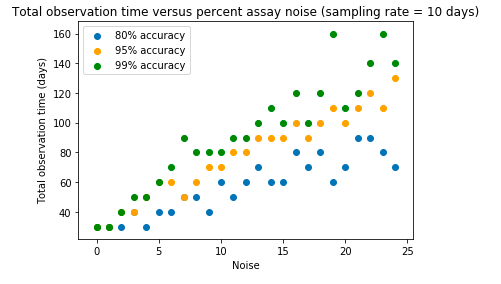
(C

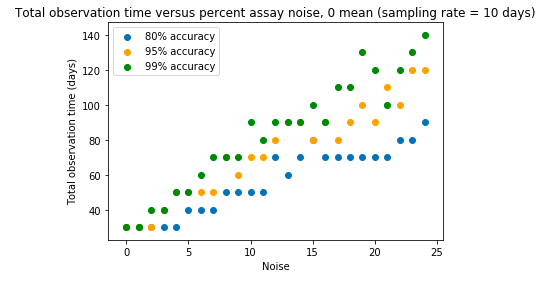


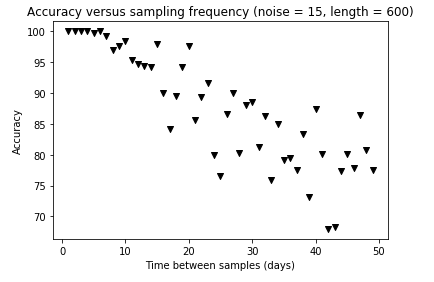
(C)

(A)

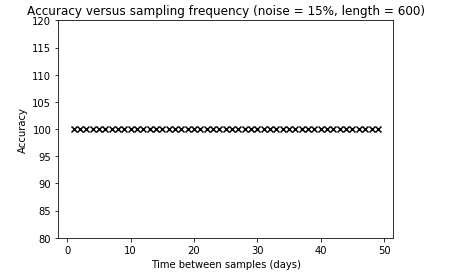
**Figure 1.6. Given a level of noise in the signals, find the observation length needed to achieve 80%, 95%, and 99% accuracy.** As noise increases, signals must become longer. All signals have measurements every 10 days. (A) Noisy signals generated with constant noise model. (B) Noisy signals generated with percent noise centered around 0. (C) Noisy signals generated with percent noise.

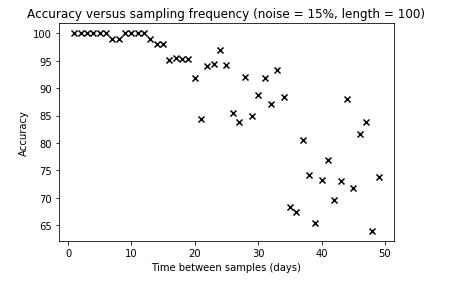
(B)

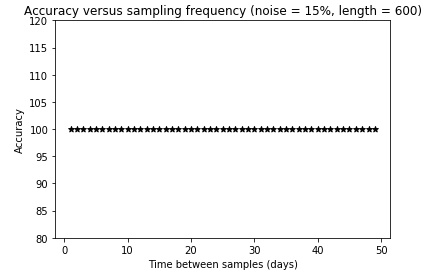
(C)

(A)

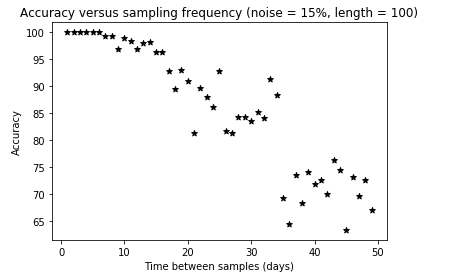
**Figure 1.7. Accuracy of classification as sampling frequency changes.** (A) Noisy signals generated with constant noise model, noise = 15. All signals have length 600 days. (B1) Noisy signals generated with 15% noise centered around 0. All signals have length 600 days. (B2). Noisy signals generated with 15% noise centered around 0. All signals have length shortened to 100 days. (C1) Noisy signals generated with 15% noise. All signals have length 600 days. (C2) Noisy signals generated with 15% noise. All signals have length shortened to 100 days.

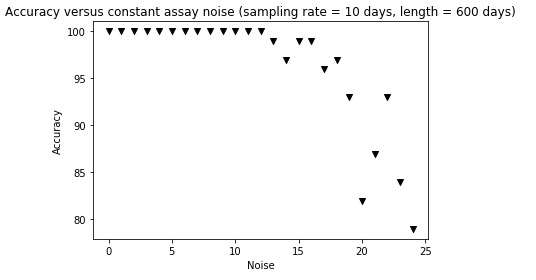
(B1) (B2)



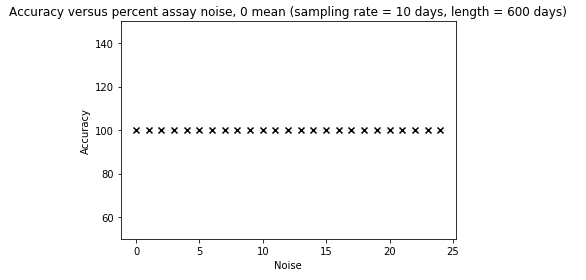
(C1)

(C2)

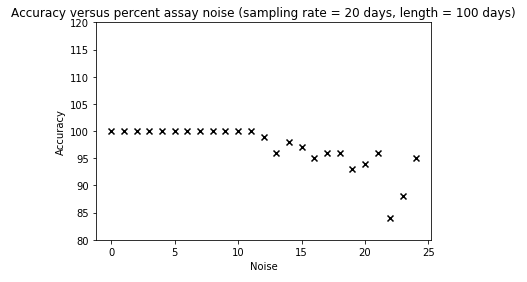


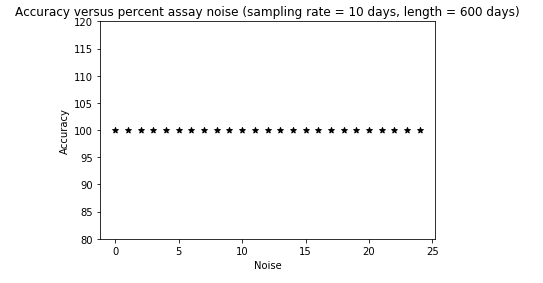
(A)

**Figure 1.8. Accuracy of classification as level of noise changes.** (A) Noisy signals generated with constant noise model All signals have length 600 days, 10 days between measurements. (B1) Noisy signals generated with percent noise centered around 0. All signals have length 600 days, 10 days between measurements. (B2). Noisy signals generated with percent noise centered around 0. All signals have length shortened to 100 days and 20 days between measurements. (C1) Noisy signals generated with percent noise. All signals have length 600 days, 10 days between measurements. (C2) Noisy signals generated with percent noise. All signals have length shortened to 100 days and 20 days between measurements

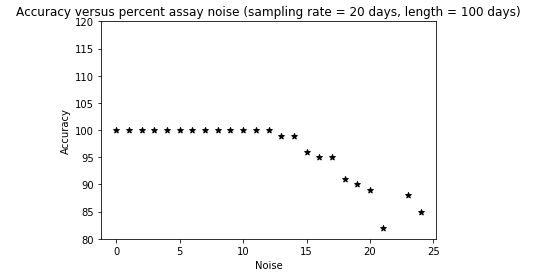
(B1)

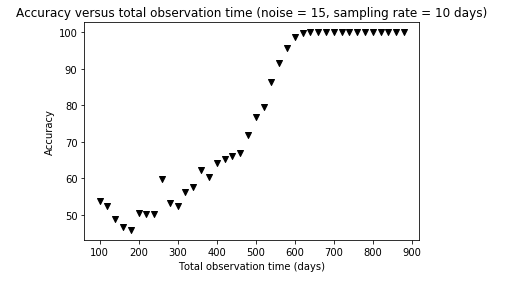
(B2)



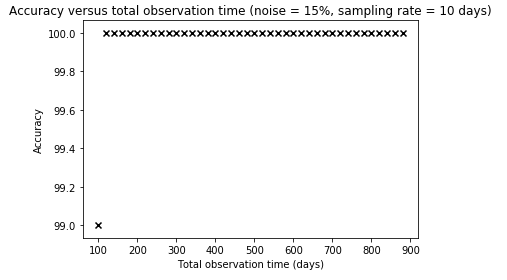
(C1)

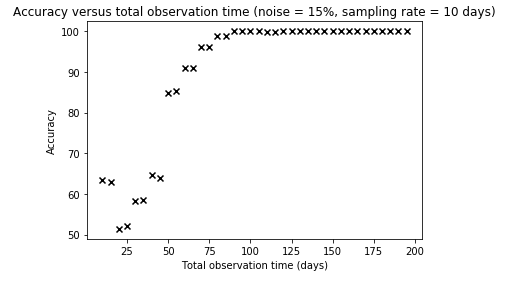
(C2)

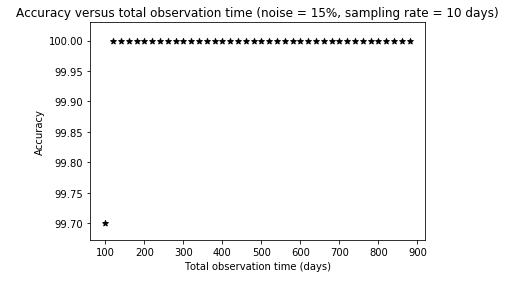


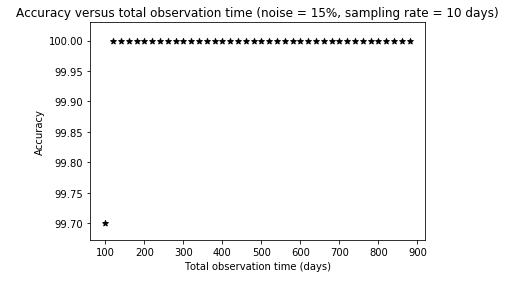
(A)

**Figure 1.9. Accuracy of classification as length of observation changes.** All signals sampled with 10 days between measurements. (A) Noisy signals generated with constant noise model, noise = 15. (B1) Noisy signals generated with 15% noise centered around 0. (B2) Noisy signals generated with 15% percent noise centered around 0. Observation length between 0 and 200 (x-axis shift) (C1) Noisy signals generated with 15% noise. (C2) Noisy signals generated with 15% noise. Observation length between 0 and 200 (x-axis shift).

(B1)

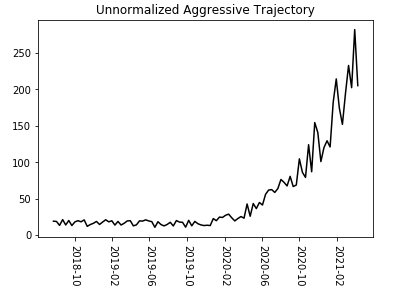
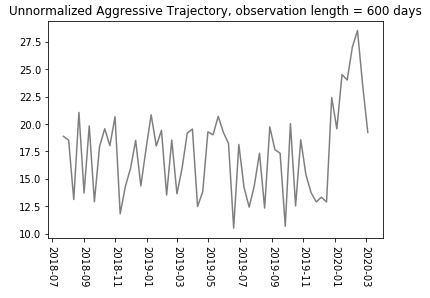
 (B2)



(C1)

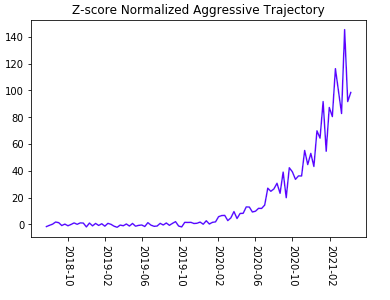
(C2)

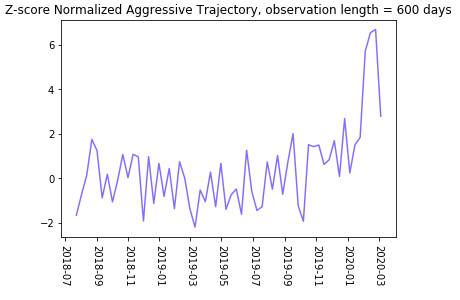
**Figure 2.1 (see next page)** **Comparison of normalization methods for a single cancerous patient’s biomarker measurements, tcaner onset = 500 days**. (A-E) Normalized biomarker measurements over the patient’s entire 1000 days of observation. (F-J) Patient’s normalized measurements of the first 600 days of observation. From top to bottom, the plots show the unnormalized measurements, z-score normalization, average subtraction, AR expanding window normalization, and AR shifting window normalization. Note the differences in y-axis scaling.



(F)

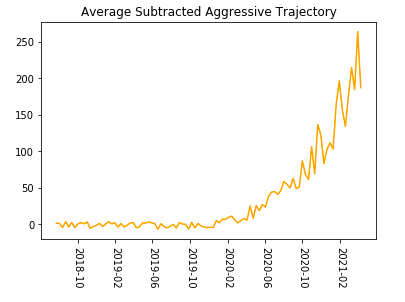
(A)

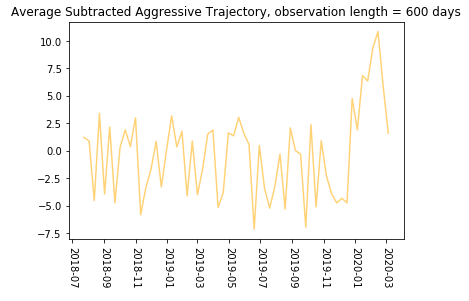


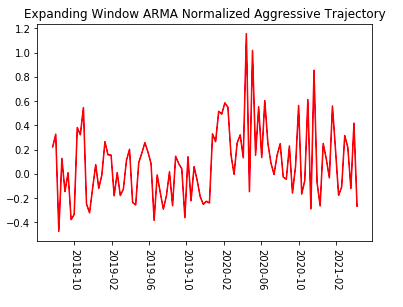
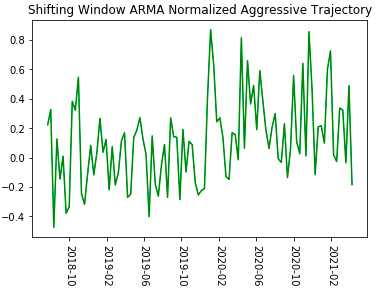
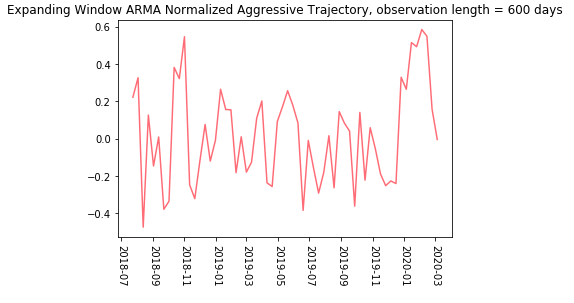
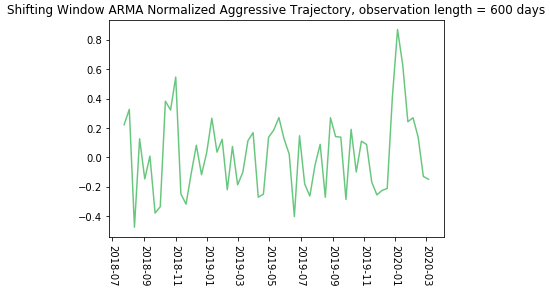


(B)

(G)







(J)

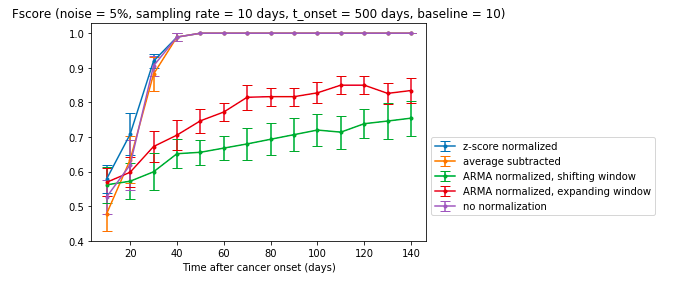
(I)

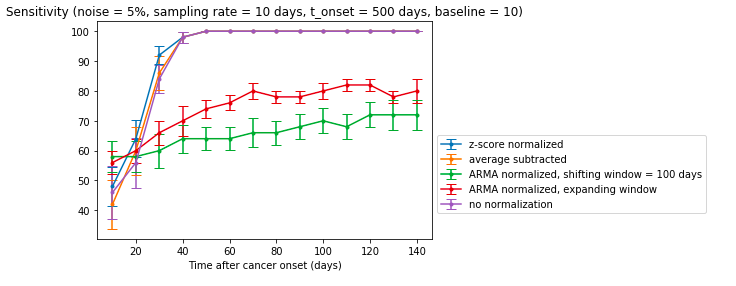
(H)

(E)

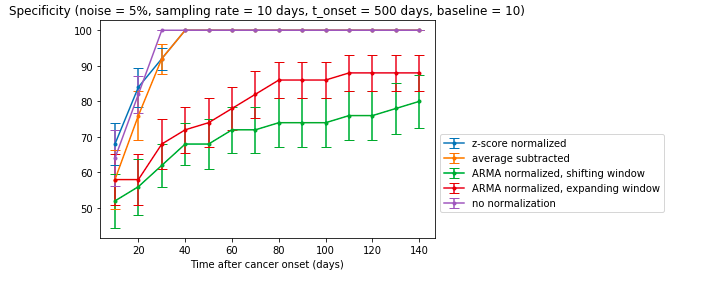
(D)

(C)

****

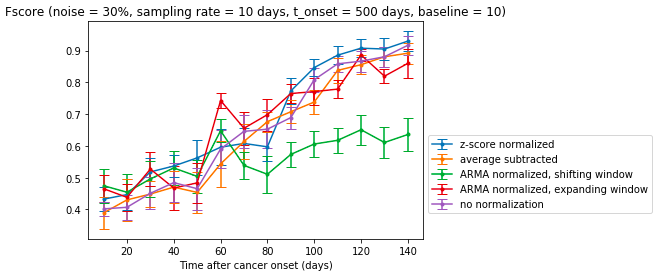
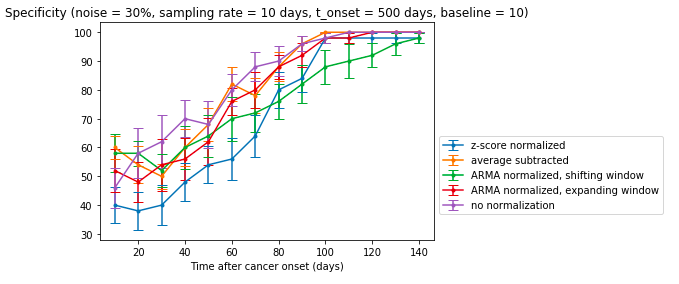
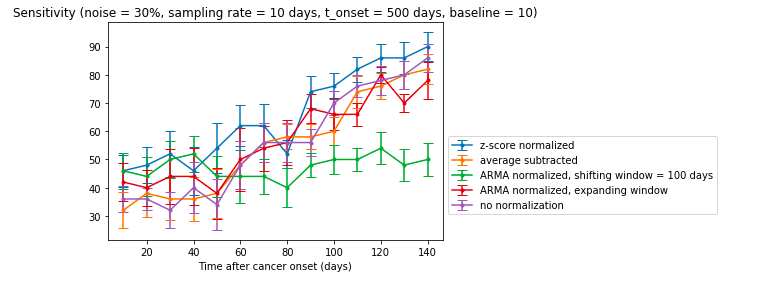
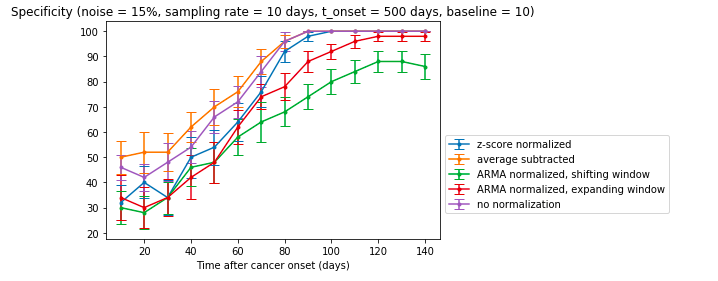
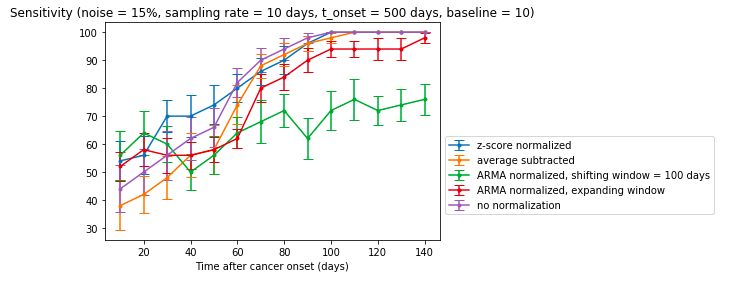
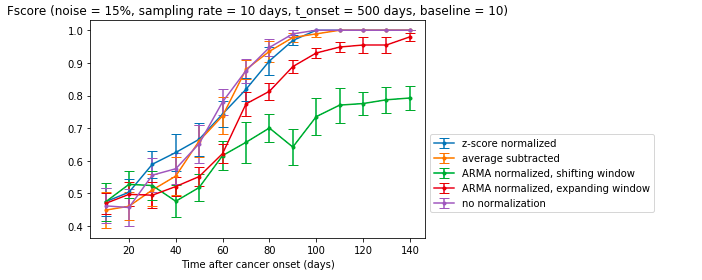
****

**Figures**

****

**Figure 2.2 F-score, Sensitivity, Specificity Comparison Using Different Normalization Methods – population healthy baseline = 10 ng/mL , noise = 5%**

The fixed length approach to *k*-NN. Observation span in days post-onset of cancer is displayed on the x-axis. The autoregressive methods significantly underperform.

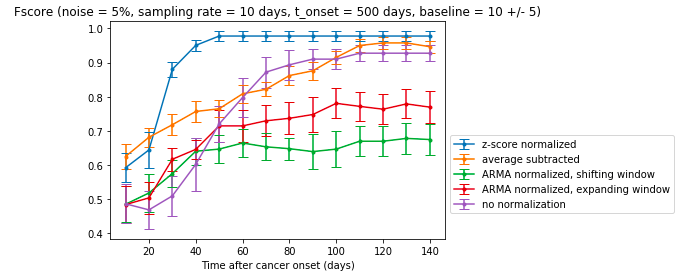


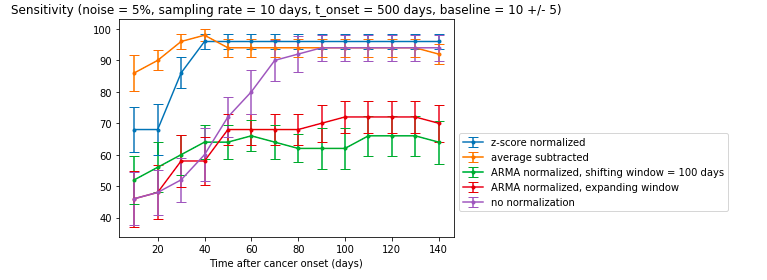
**Figure 2.3 F-score, Sensitivity, Specificity Comparison Using Different Normalization Methods – population healthy baseline = 10 ng/mL , noise = 15%**

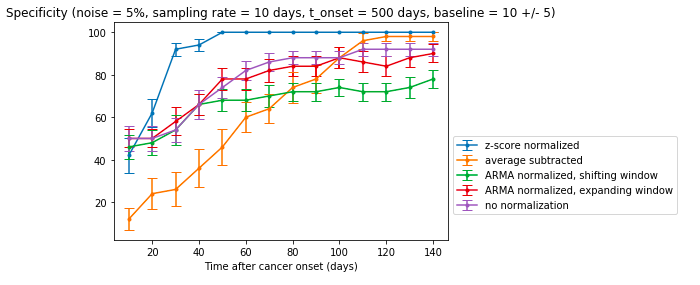
The fixed length approach to *k*-NN. Observation span in days post-onset of cancer is displayed on the x-axis. The shifting window autoregressive method underperforms.

**Figure 2.4 F-score, Sensitivity, Specificity Comparison Using Different Normalization Methods – population healthy baseline = 10 ng/mL, noise = 30%**

The fixed length approach to *k*-NN. Observation span in days post-onset of cancer is displayed on the x-axis. The shifting window autoregressive method underperforms in sensitivity.

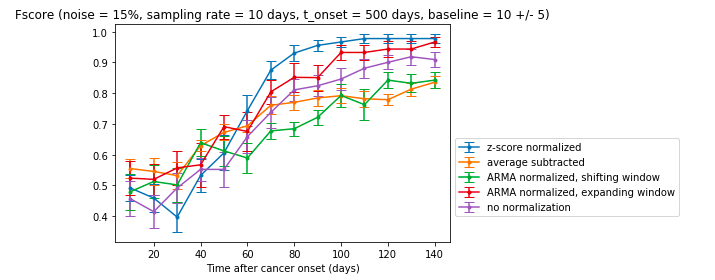


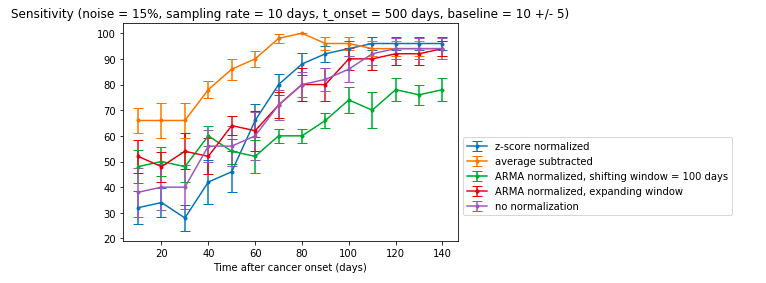


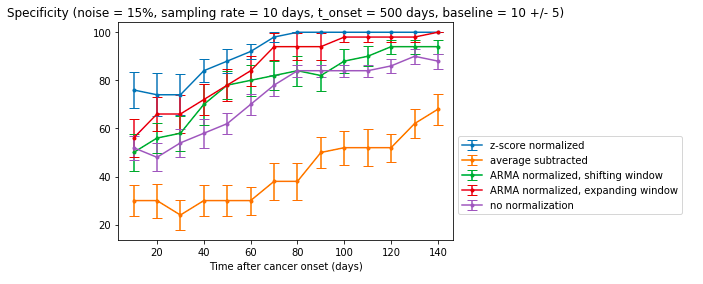


**Figure 2.5 F-score, Sensitivity, Specificity Comparison Using Different Normalization Methods – population healthy baseline = 10 ± 5 ng/mL, noise = 5%**

The fixed length approach to *k*-NN. Observation span in days post-onset of cancer is displayed on the x-axis. The autoregressive methods underperform.

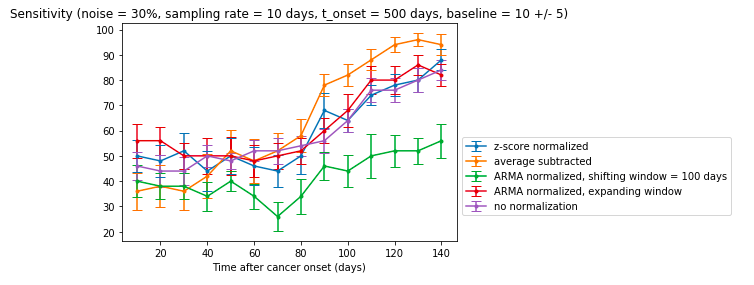
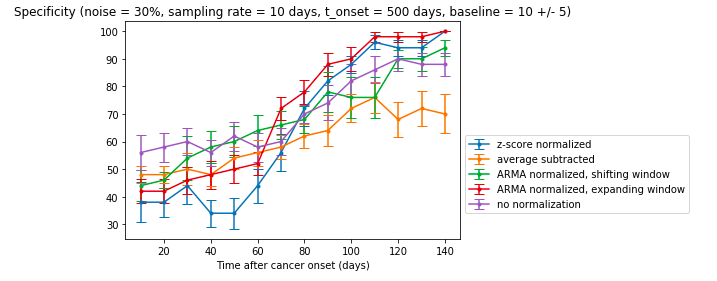
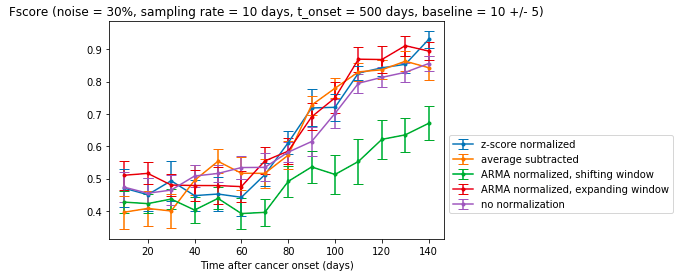






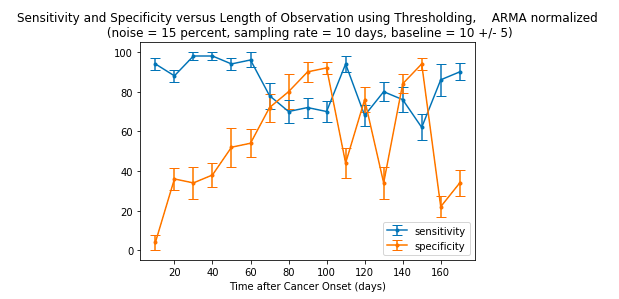
**Figure 2.6 F-score, Sensitivity, Specificity Comparison Using Different Normalization Methods – population healthy baseline = 10 ± 5 ng/mL, noise = 15%**

The fixed length approach to *k*-NN. Observation span in days post-onset of cancer is displayed on the x-axis. Average subtraction underperforms.

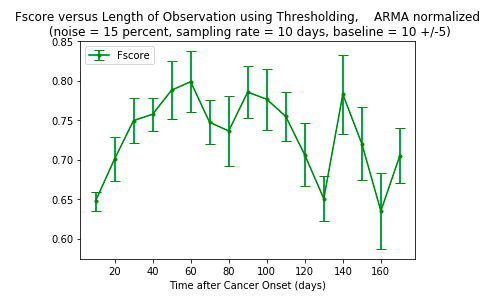


**Figure 2.7 F-score, Sensitivity, Specificity Comparison Using Different Normalization Methods – population healthy baseline = 10 ± 5 ng/mL, noise = 30%**

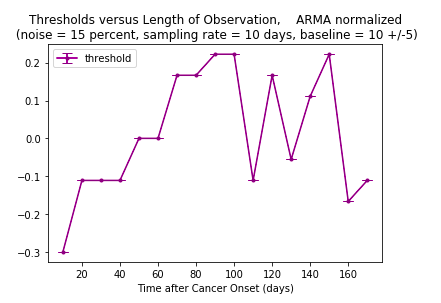
The fixed length approach to *k*-NN. Observation span in days post-onset of cancer is displayed on the x-axis. Autoregressive shifting window underperforms.



(A)

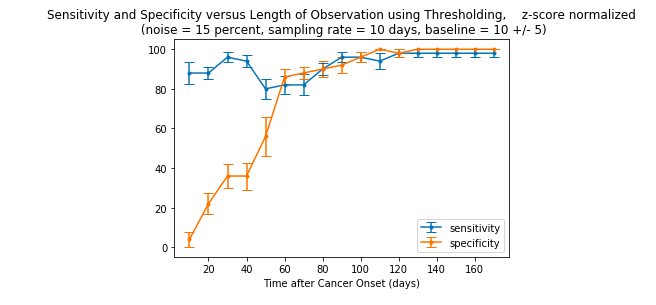


(B)



(C)

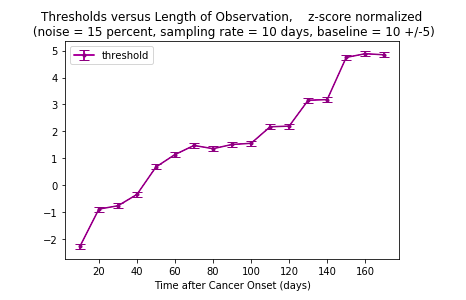
**Figure 2.8. Sensitivity, Specificity, F-scores, and Optimal Thresholds for Different Lengths of Observation on AR Expanding Window Normalized Measurements.** A) Sensitivity and specificity of classification for different observation lengths (expressed in days post-onset of cancer) B) F-score of classification for different observation lengths C) Optimal threshold value at each observation length.



(A)

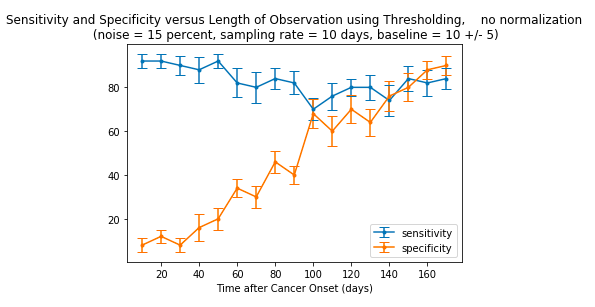


(B)

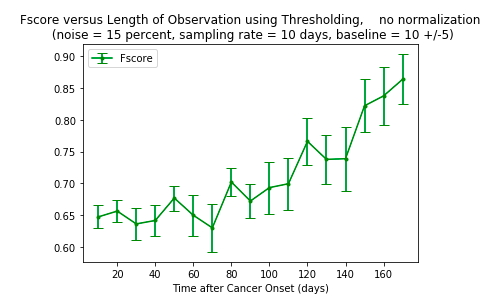


(C)

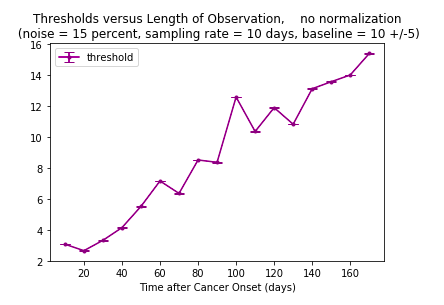
**Figure 2.9. Sensitivity, Specificity, F-scores, and Optimal Thresholds for Different Lengths of Observation on Z-score Normalized Measurements.** A) Sensitivity and specificity of classification for different observation lengths (expressed in days post-onset of cancer) B) F-score of classification for different observation lengths C) Optimal threshold value at each observation length.



(A)

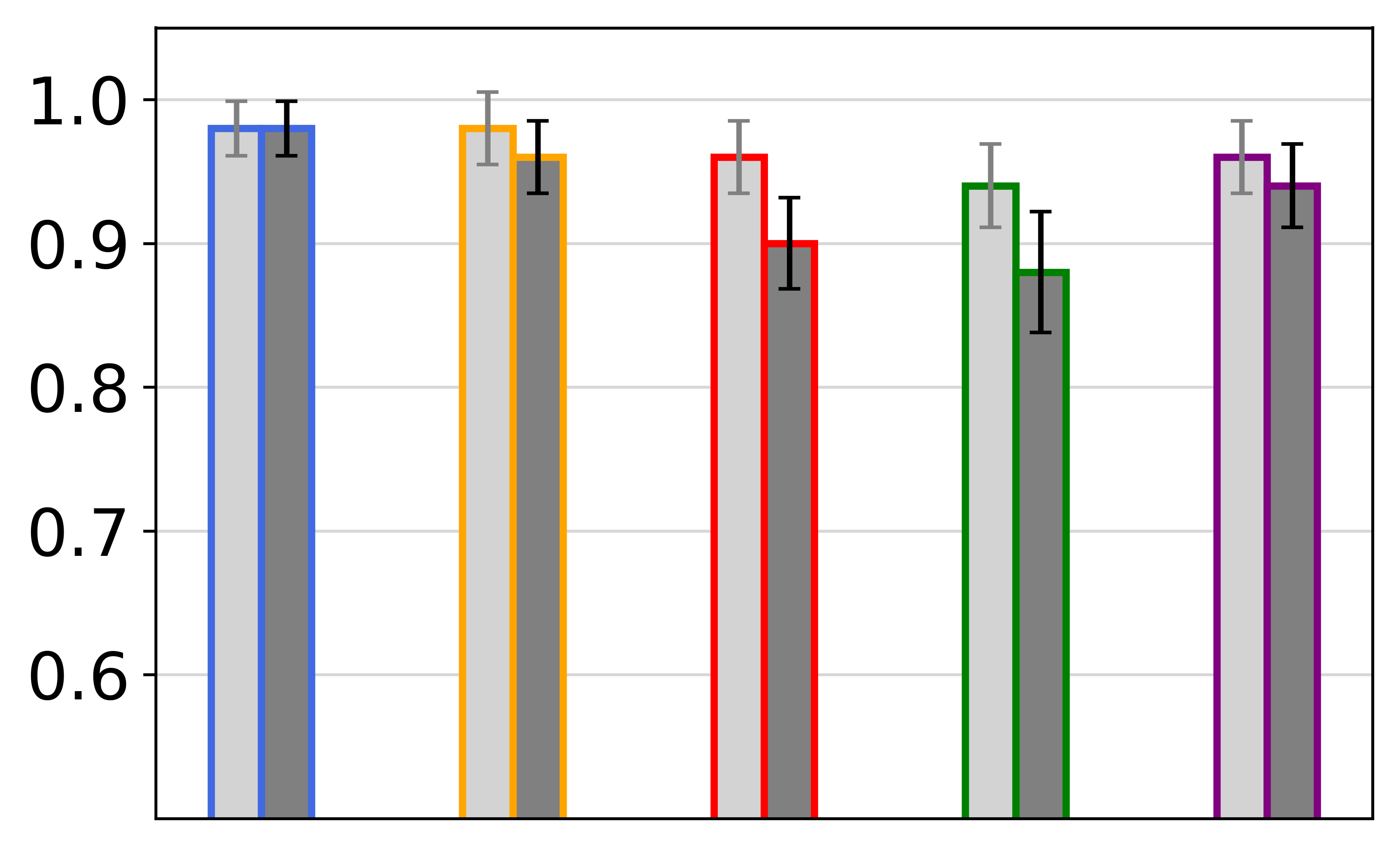
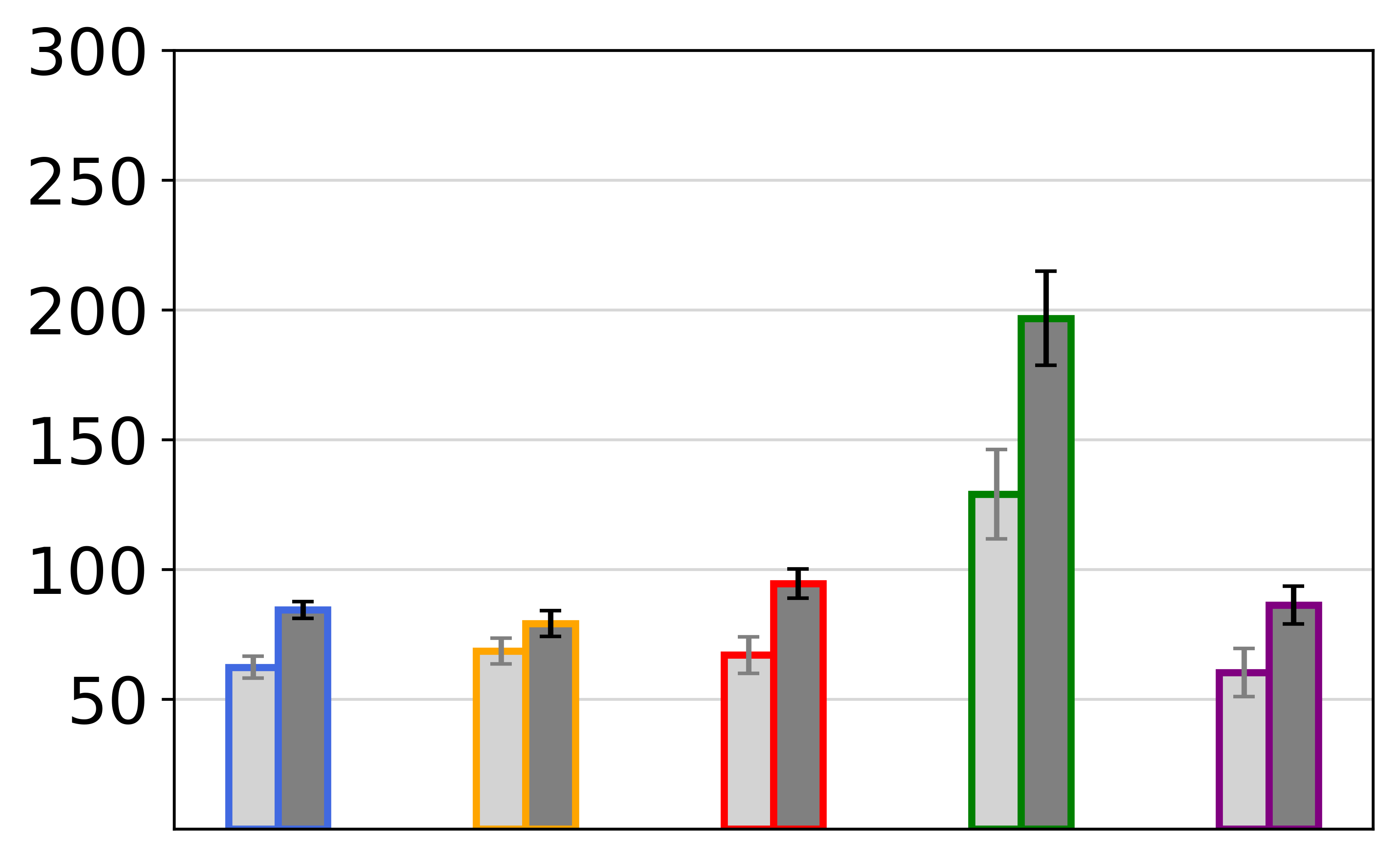
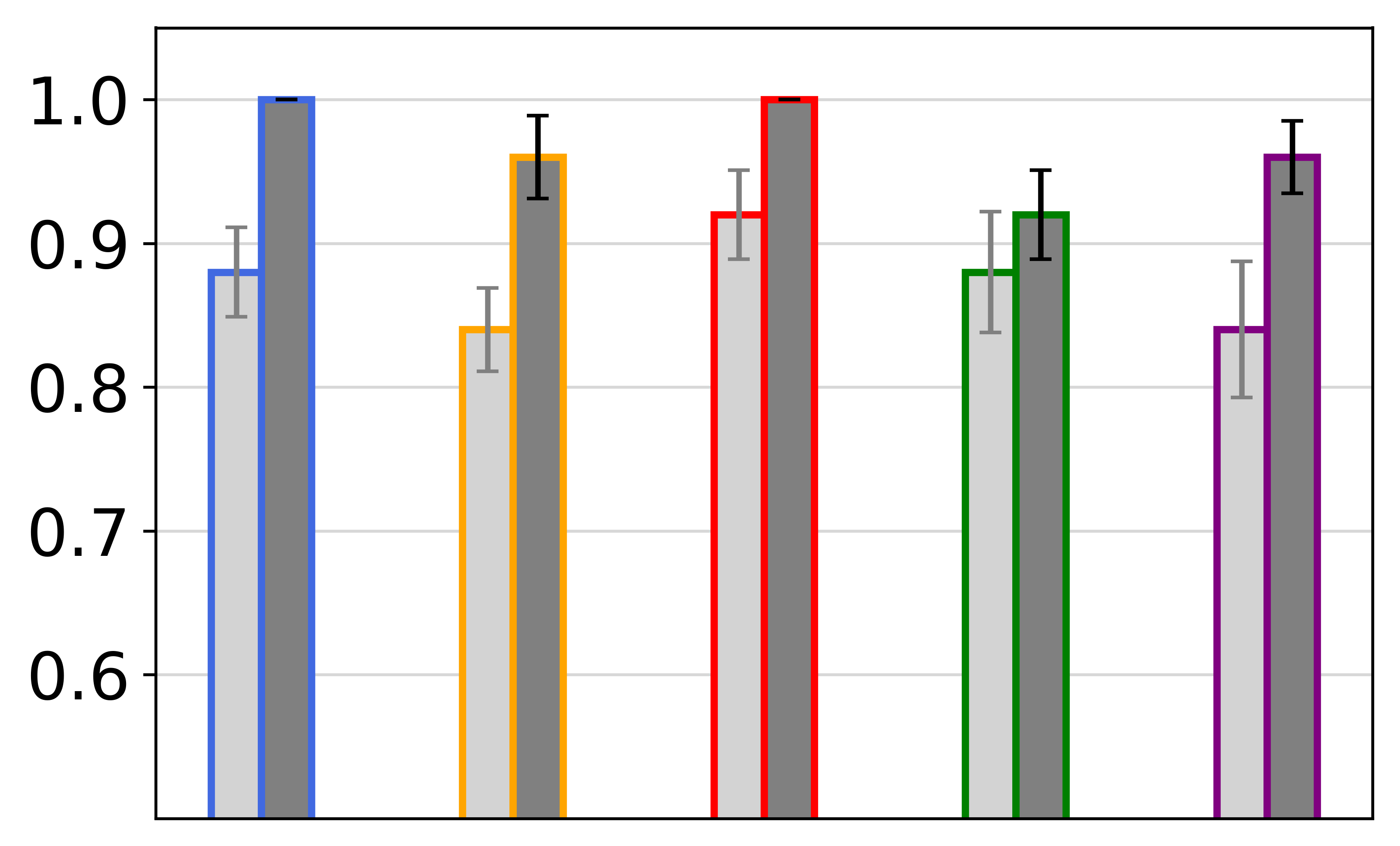


(B)



(C)

**Figure 2.10. Sensitivity, Specificity, F-scores, and Optimal Thresholds for Different Lengths of Observation on Unnormalized Measurements.** A) Sensitivity and specificity of classification for different observation lengths (expressed in days post-onset of cancer) B) F-score of classification for different observation lengths C) Optimal threshold value at each observation length.



**Figure 3.1.** ***k-*NN Dynamic Approach** (*Top*) Average time of cancer detection with classification based on either 80% (light gray) or 95% (dark gray) confidence. Autoregressive Shifting Window method required the longest observation span (125-200 days). (*Middle, Bottom*) Sensitivity and specificity of detection. Sensitivity was lowest (0.60-0.75) for Autoregressive Shifting Window method.

No norm

Shift

Expand

Subtract

Z-score

Specificity

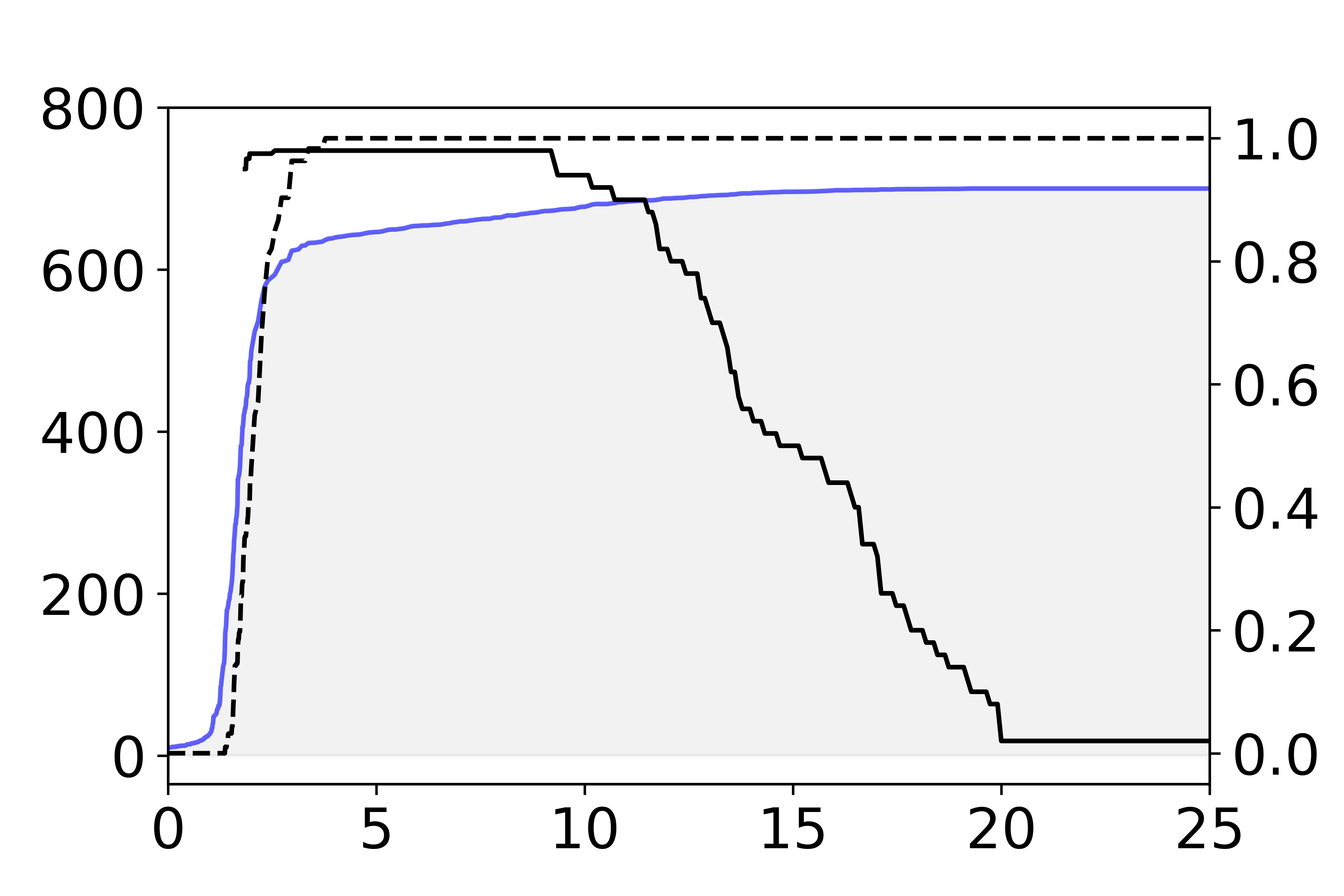
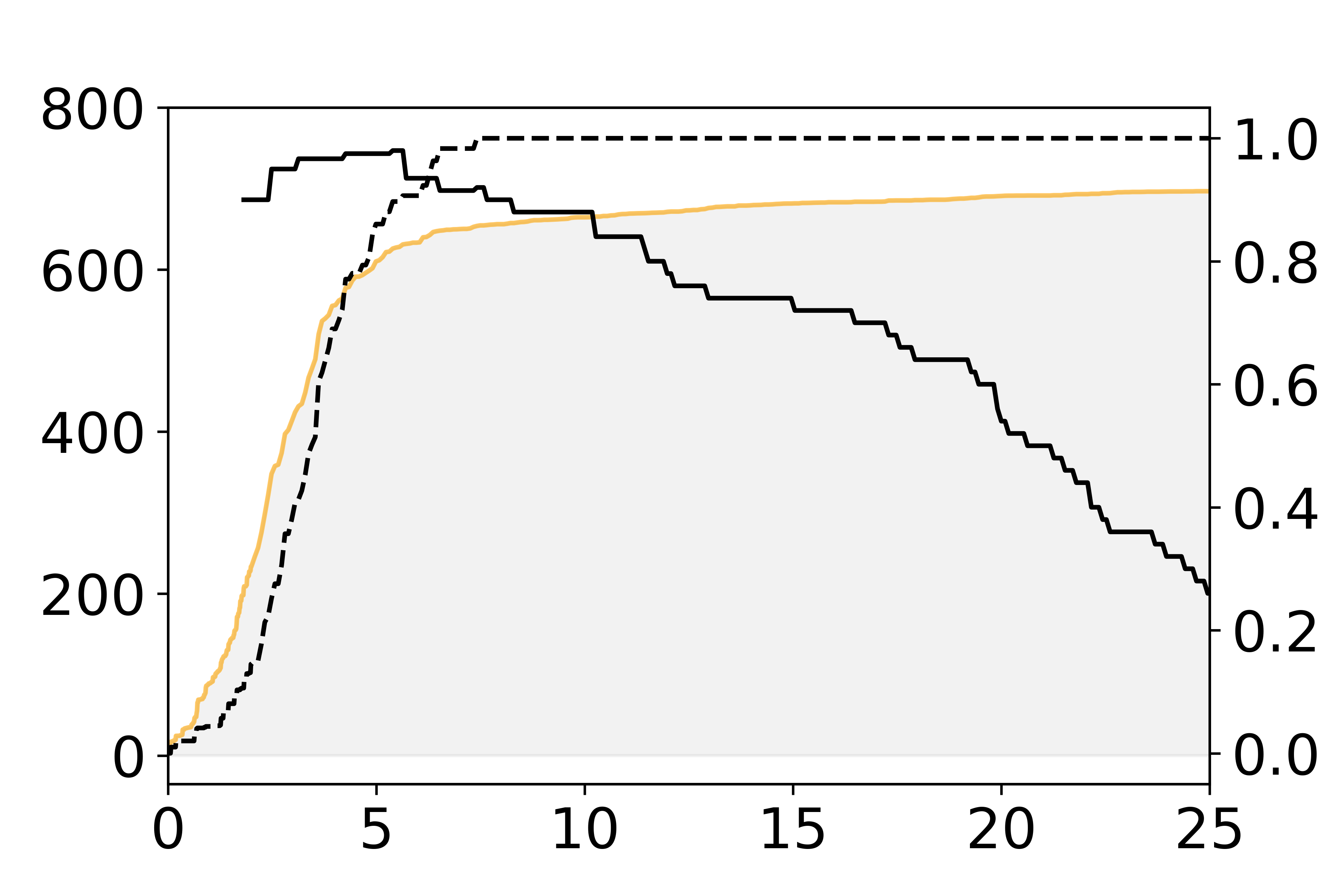
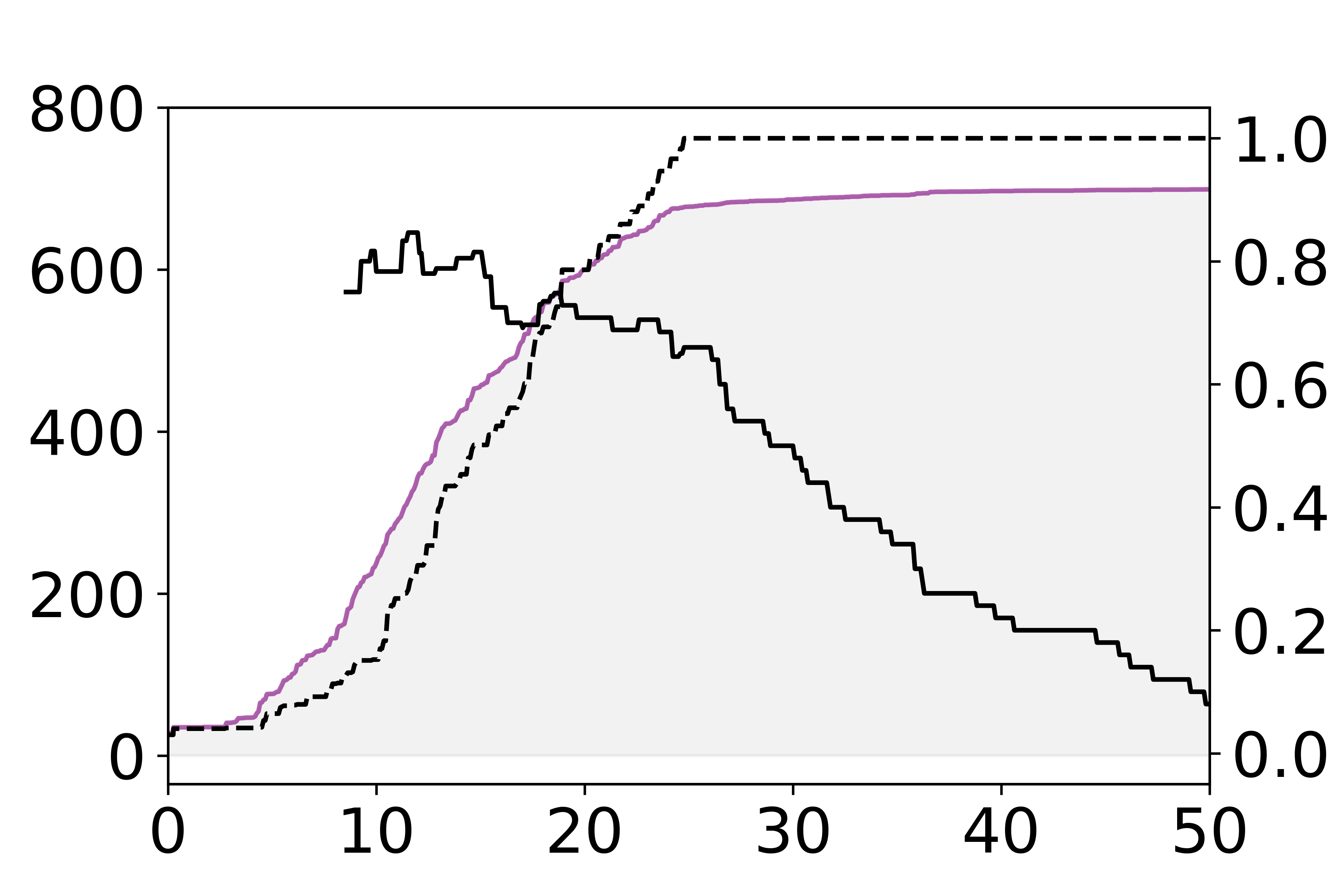
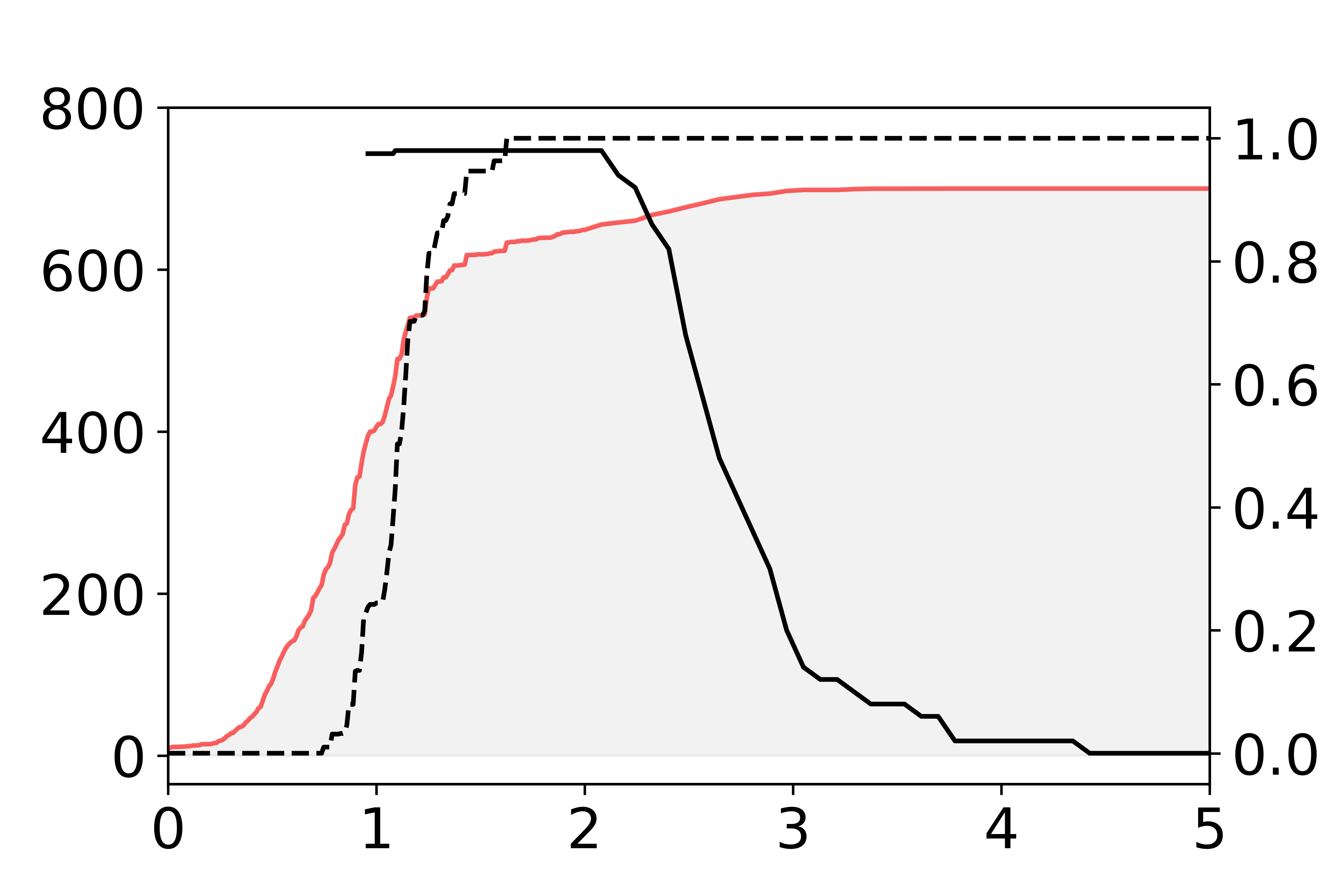
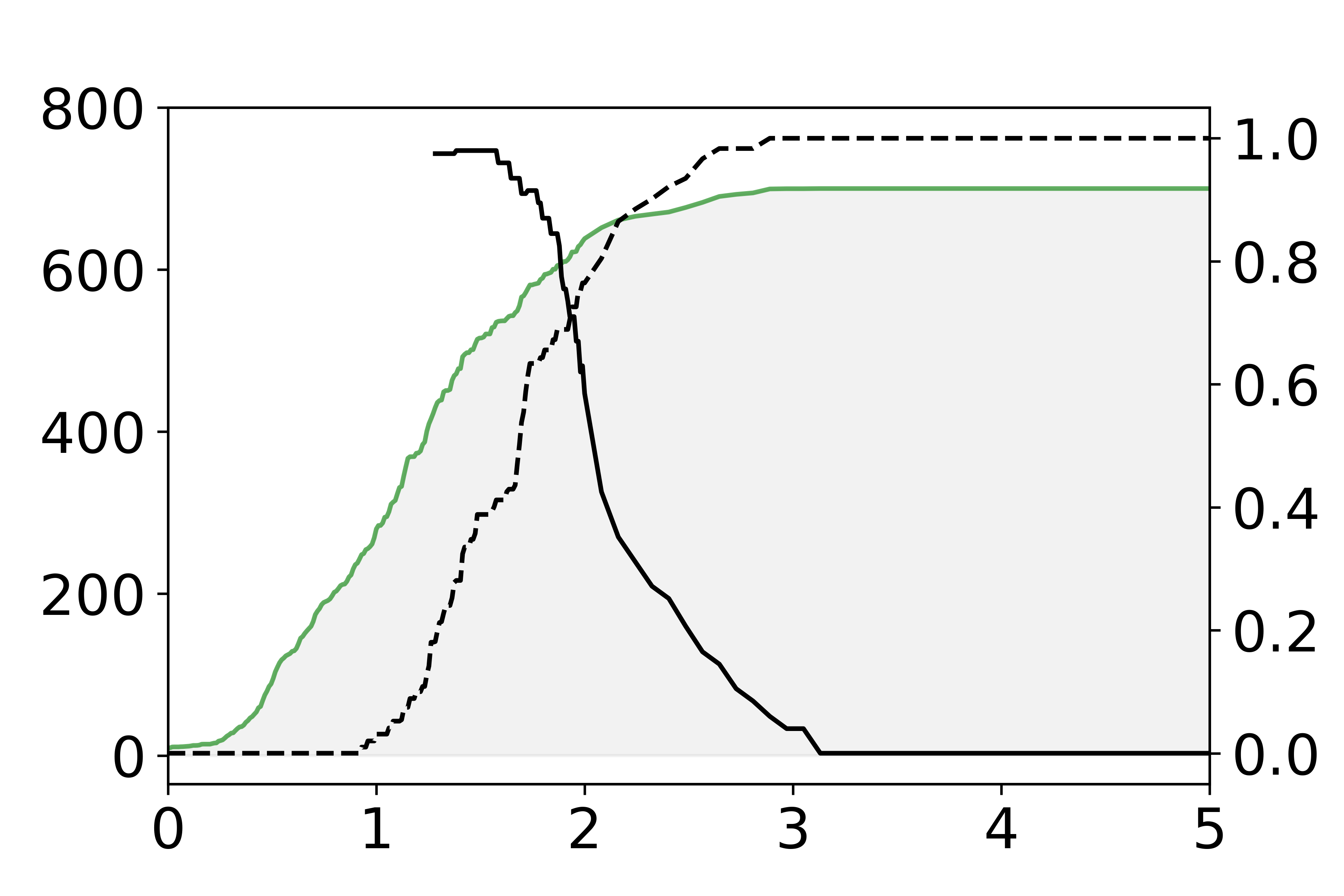
Sensitivity

Time of Detection

(days post-onset of cancer)

*Z-score*

*Average Subtraction*



Threshold (ng/mL)

Threshold (ng/mL)

Sensitivity (–) and Specificity (--)

Time of First Threshold Cross (days, )

**Figure 3.2**. **Thresholding dynamic approach.** As the threshold value increases, the average time of observation before the threshold is crossed increases. Increasing the threshold increases sensitivity

and decreases specificity. Cancer onset was assumed on day 500.

*No Normalization*

*Shifting Window*

*Expanding Window*

**RESULTS**

*k-Nearest Neighbor Classification: Fixed Length Approach*

We examined the performance of the nearest neighbor algorithm in response to changes in three main parameters: (i) the amount of noise in the measurements, (ii) the frequency of measurements, and (iii) the total amount of time over which measurements were taken. The purpose of this analysis was to examine parameters that may be adjusted in the clinic, and to determine clinically feasible noise levels and sampling frequencies that will lead to earliest detection time with highest accuracy. Keeping one of the parameter values fixed, a two-way analysis was performed on the remaining two parameters. The combinations that resulted in a classification accuracy of greater than 80%, greater than 95%, and greater than 99% were plotted, with the x-axis depicting one parameter value and the y-axis depicting the second parameter value (figs 1.2, 1.5, 1.6).

With noise fixed at 0%, only two observations were needed for the classification to achieve greater than 99% accuracy, regardless of sampling frequency (fig 1.2, A, slope = 1). With noise increased to 15%, the necessary length of observation appears to grow logarithmically as the frequency of sampling decreases. This relationship was explored in a three-way analysis of sampling frequency, total number of observations, and classification sensitivity and specificity (fig 1.3). Examining the plot visually, the border between the light yellow colored region (higher sensitivity and specificity) and the darker green colored region (lower sensitivity and specificity) seems to follow a power function with respect to sampling frequency. The x,y pairs corresponding to 95% sensitivity were log-transformed and fit using least-squares linear regression. The fit is best for higher sampling frequencies; as the time between samples increases beyond 150 days, two observations were sufficient to achieve greater than 95% sensitivity, and the points flatten to a horizontal line (fig 1.4, A4). Applying the fitted equation to the observation length, which was calculated as the product of the time between samples and the number of observations – 1, again the fit is most appropriate for higher sampling frequencies (fig 1.4, B1).

Classification accuracy was also assessed with two fixed parameters and one varying parameter. With sampling frequency and observation length fixed, increasing noise from 0 % to 10% had little effect on classification performance across all three noise models. As noise increased above 10%, accuracy decreased significantly (fig 1.8). With noise and sampling frequency fixed, increasing observation time improved classification accuracy. For all three noise models, once the observation length was long enough, classification accuracy reached its maximum 100%. This occured earliest for the 0-mean percent noise model, followed by the percent noise model, and finally the constant noise model.

Comparing across normalization methods, at low noise levels and with no variation in population baseline, using the z-score normalized, average subtracted, or unnormalized measurements produced nearly equivalent results with the nearest neighbor classification. Both the shifting and expanding window autoregressive models had lower sensitivity and specificity values (fig 2.2). At higher noise levels and no population baseline variation, the relative results of the autoregressive models improved (fig 2.3, 2.4). Upon adding variation to the healthy population baseline, at 5% noise level, the z-score normalized measurements yielded the highest f-scores. As noise level increased, the results across all methods became more similar, and none of the normalization methods clearly outperformed the others.

*k-Nearest Neighbor Classification: Dynamic Approach*

Using an 80% confidence level for classification led to early detection time and higher sensitivity, but lower specificity (fig 3.1). Using a 95% confidence threshold led to higher specificity but delayed detection time and lower sensitivity. For 80% confidence, unnormalized measurements yielded the earliest mean detection time (60 days post-onset of cancer), z-score normalization produced the highest sensitivity (0.88), and autoregressive expanding window normalization produced the highest specificity (0.92). For 95% confidence, average subtraction yielded the earliest mean detection time (79 days post-onset of cancer), and z-score normalization produced the highest sensitivity (0.98) and specificity (1.0). Overall, z-score normalization had the best performance.

*Thresholding: Fixed Length Approach*

For z-score normalized measurements, sensitivity, specificity, and the optimal threshold value increase as the observation length increases (fig 2.8). For autoregressive normalized measurements, there existed a window of observation spans for which sensitivity, specificity, and optimal threshold values peaked, after which they decreased again (fig 2.9). This can likely be explained by the model forecasts eventually “catching up” to the observed values, i.e. the model mistook the abnormally increasing biomarker measurements as an increasing healthy baseline.

*Thresholding: Dynamic Approach*

Selecting an optimal threshold requires optimizing the combination of early detection, sensitivity, and specificity. Smaller thresholds yielded earlier detection time, higher sensitivity, but lower specificity. Larger thresholds yielded higher specificity, but delayed detection time and lower sensitivity. Z-score normalization yielded the best thresholding results, as there existed a range of thresholds for which both sensitivity and specificity were at nearly 100%, and the smallest of these thresholds yielded the earliest detection time (fig 3.2, A). For all other normalization methods, the sensitivity and specificity curves intersected at a single point, indicating poorer performance and a very limited range of thresholds that could be used to optimally classify patients.

*k-NN versus Thresholding*

With the dynamic approach to *k*-NN, each normalization method had a certain time of detection (fig 3.1, top). For each normalization method, we compared the sensitivity and specificity of classification using 80% confidence *k*-NN to the sensitivity and specificity of classification using thresholding, where the threshold was chosen to give the closest possible detection time to the 80% *k-*NN approach. Across all normalization methods, *k*-NN yielded a sensitivity of 0.11 ± 0.05 higher than thresholding and a specificity of 0.14 ± 0.07 higher than thresholding. Hence, for the simulated set of patients, *k*-NN was the better performing classifier.

**DISCUSSION**

This study developed a framework for making personalized cancer diagnoses tailored to patient-specific baselines by applying supervised learning methods used for anomaly detection [8]. We demonstrated that classification accuracy is dependent on the properties of the biomarker measurements, including noise, frequency of sampling, and observation span. Using the simulated patients, if we desire a given level of accuracy and know any two of the three parameters, we are able to determine the value of the fourth parameter. For example, if we know the expected amount of assay error and want to achieve a certain classification accuracy within a certain detection time, we can calculate how often patients should visit the clinic for blood samples. This could potentially allow for a more efficient allocation of patient and clinician time and hospital resources.

By classifying patients with *k*-NN on normalized measurements, we have combined examinations of individual, patient-specific baselines with examinations of how the trends in the patient’s longitudinal biomarker levels relate to population biomarker trends. (The latter condition is absent in classification by thresholding, which only considers single biomarker measurements when assigning patient status). Baseline determination allows us to make more personalized diagnoses, and classifying through pattern recognition allows us to better harness the information stored in a patient’s longitudinal biomarker history, leading to earlier detection. Cell culture, mouse model, and patient studies are currently being planned to collect longitudinal biomarker data, and we will then be able to translate the framework developed through this study to real data.

**REFERENCES**

1. Kulasingam V and Diamandis E. Strategies for discovering novel cancer biomarkers through utilization of emerging technologies. Nature Clinical Practice Oncology. 2008. 5(10): 588 – 599.

2. Hori SS, Lutz AM, Paulmurugan R, and Gambhir SS. A Model-Based Personalized Cancer Screening Strategy for Detecting Early-Stage Tumors Using Blood-Borne Biomarkers. *Cancer Res*. 2017. 77(10): p. 2570-2584.

3. Hori SS and Gambhir SS. Mathematical Model Identifies Blood Biomarker-Based Early Cancer Detection Strategies and Limitations. *Sci Transl Med*. 2011. 3(109): p. 109ra116.

4. Lin H, McCulloch CE, Turnbull BW, Slate EH, and Clark LC. A Latent Class Mixed Model for Analyzing Biomarker Trajectories with Irregularly Scheduled Observations. Stat Med. 2000. 19(10): 1303-18.

5. Sölétormos G et.al. Biological Variation of Total Prostate-Specific Antigen: A Survey of Published Estimates and Consequences for Clinical Practice. Clinical Chemistry. 2005. 51(8): 1342-1351.

6. Adhikari R, Agrawal RK. An Introductory Study on Time Series Modeling and Forecasting. Lambert Academic Publishing. 2013.

7. Keller JM, Gray M, Givens JA. A Fuzzy *K-*Nearest Neighbor Algorithm. IEEE Transactions on Systems, Man, and Cybernetics. 1985. 15(4): 580-585.

8. Monnet A, Mittal A, Paragios N, Ramesh V. Background Modeling and Subtraction of Dynamic Scenes. Proceedings Ninth IEEE International Conference on Computer Vision. 2003. 2:1305-1312.