Open Access Medical Statistics

Dovepress
open access to scientific and medical research

Open Access Full Text Article

METHODOLOGY

Classification of biosensor time series using dynamic time warping: applications in screening cancer cells with characteristic biomarkers

This article was published in the following Dove Press journal:
Open Access Medical Statistics
8 June 2016
Number of times this article has been viewed

Shesh N Rai[1,2*]
Patrick J Trainor[2,3*]
Farhad Khosravi[4]
Goetz Kloecker[5]
Balaji Panchapakesan[4]

[1]Biostatistics Shared Facility, JG Brown Cancer Center, University of Louisville, [2]Department of Bioinformatics and Biostatistics, University of Louisville, [3]Department of Medicine, University of Louisville, KY, USA; [4]Small Systems Laboratory, Department of Mechanical Engineering, Worcester Polytechnic Institute, Worcester, MA, [5]Hematology and Oncology, James Graham Brown Cancer Center, University of Louisville, Louisville, KY, USA

*These authors contributed equally to this work.

Abstract: The development of biosensors that produce time series data will facilitate improvements in biomedical diagnostics and in personalized medicine. The time series produced by these devices often contains characteristic features arising from biochemical interactions between the sample and the sensor. To use such characteristic features for determining sample class, similarity-based classifiers can be utilized. However, the construction of such classifiers is complicated by the variability in the time domains of such series that renders the traditional distance metrics such as Euclidean distance ineffective in distinguishing between biological variance and time domain variance. The dynamic time warping (DTW) algorithm is a sequence alignment algorithm that can be used to align two or more series to facilitate quantifying similarity. In this article, we evaluated the performance of DTW distance-based similarity classifiers for classifying time series that mimics electrical signals produced by nanotube biosensors. Simulation studies demonstrated the positive performance of such classifiers in discriminating between time series containing characteristic features that are obscured by noise in the intensity and time domains. We then applied a DTW distance-based *k*-nearest neighbors classifier to distinguish the presence/absence of mesenchymal biomarker in cancer cells in buffy coats in a blinded test. Using a train–test approach, we find that the classifier had high sensitivity (90.9%) and specificity (81.8%) in differentiating between EpCAM-positive MCF7 cells spiked in buffy coats and those in plain buffy coats.

Keywords: buffy coats, cancer detection, breast cancer, epcam, MCF7, k-nn classifier, biosensors, time series, instance-based learning

Introduction

Biosensors that produce electrical signals or other time series data present a unique challenge for researchers attempting to use such sensors to classify biological samples. The analysis of such data is complicated by the undesirable variation in the time domain between samples of the same class in addition to variability in the characteristic patterns that may be shared between samples of the same class. In this article, we develop a classification methodology applicable to time series data produced by biosensors. A pseudo-distance metric known as dynamic time warping (DTW) is utilized to quantify the similarity between series in order to construct a classifier. The advantage of this methodology is that it manipulates the time domain of samples in order to better facilitate pattern detection. The use of DTW for quantifying the similarity between time series has been successfully applied in areas such as speech identification[1–3] and medical diagnostics.[4,5] After describing the methodology, we evaluate the performance of such a classifier using simulation studies. Finally, we apply the classification methodology to breast cancer cell detection using a novel single-walled carbon nanotube (CNT) biosensor device.

Correspondence: Balaji Panchapakesan
Small Systems Laboratory, Department of Mechanical Engineering, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609, USA
Email bpanchapakesan@wpi.edu

http://dx.doi.org/10.2147/OAMS.S104731

Methods

The goal of this methodology is to effectively classify time series data for which the class is unknown by utilizing a set of reference time series data for which the class is known. The construction of a similarity-based *k*-nearest neighbors (*k*-nn) classifier follows the following steps: 1) a suitable distance metric (or pseudo-metric) for quantifying the dissimilarity between two time series is defined; 2) *k*-fold cross-validation is used to select classifier's tuning parameters using training data; and 3) the classifier's performance is evaluated on an independent set of observations. We represent an individual time series as $\mathbf{y} = \{y_t : t \in T\}$, where T is a time index. We consider T to be a discrete index, which is a valid assumption given that most biosensors sample at some fixed interval. When sample annotation is explicitly shown, $y_{i,j,t}$ denotes the value of a series from the i th sample of the j th class at time $T = t$. Since the symbol k is traditionally used to represent an integer in both cross-validation and in nearest neighbors classifiers, we use k to describe the cross-validation parameter and κ to describe the classifier parameter.

Dynamic time warping

The first step we propose in developing the classifier for biosensor data is the identification of a suitable dissimilarity measure. Time series data from familiar biosensor devices such as Electrocardiographs (ECGs) as well as newer devices such as the CNT biosensors are marked by the occurrence of prototypical features that herald the class of the series. Often the time domain of such series is irrelevant and complicates automatic feature detection. As an example, two time series with similar features are shown in Figure 1A. The red series is a phase shift of the blue series with a slight vertical lift. In this situation, a traditional distance metric such as Euclidean distance would conclude that the series are very dissimilar. Yet with respect to the prototypical feature, these series are quite similar. As a result, "warping" of the time domains of one or both of the series (through the insertions of gaps) is necessary for aligning the series for proper quantification of dissimilarity.

DTW is a class of dynamic programming algorithms that have been developed in order to align time series in such a manner so that a traditional distance metric can be used to quantify dissimilarity. The resulting alignment of the two series by DTW is shown in Figure 1B.

The utilization of a distance metric on the new warped series, henceforth referred to as a DTW distance, does not satisfy the properties required to be considered as a distance metric. However, DTW distance can still be effectively employed as a dissimilarity measure for constructing classifiers as will be shown later. To calculate the DTW distance between two time series, we utilize the methods presented by Sakoe and Chiba whose original introduction of DTW methodology provides a nice resource for researchers interested in DTW.[6] Starting with two series, $\mathbf{y}_1$ of length m and $\mathbf{y}_2$ of length n, we first construct an $m \times n$ grid of nodes representing a matching of time points from both time indices as illustrated in Figure 2. A warping path,[7] $\phi(k) = \{(t, t') : k \in 1, 2, \ldots, K\}$, through the grid is then sought to minimize the DTW distance, which is defined as:

$$DTW(\mathrm{y}_1, \mathrm{y}_2) = \min_{\phi \in \Phi} \sum_{k=1}^{K} d\left[\mathrm{y}_1\left(\phi_{y_1}(k)\right), \mathrm{y}_2\left(\phi_{y_2}(k)\right)\right] w(k) \quad (1)$$

where d is a traditional distance metric such as the Euclidean metric and $w(k)$ is a positive weight function. The value of the weight function depends on the slope of the line segment joining two points in the grid. Further constraints imposed on the warping path are:

1. Boundary constraints: $\phi(1) = (1, 1)$ and $\phi(K) = (m, n)$.
2. Step size constraint: $\phi(k) - \phi(k-1) = \{(0, 1), (1, 0), (1, 1)\}$.

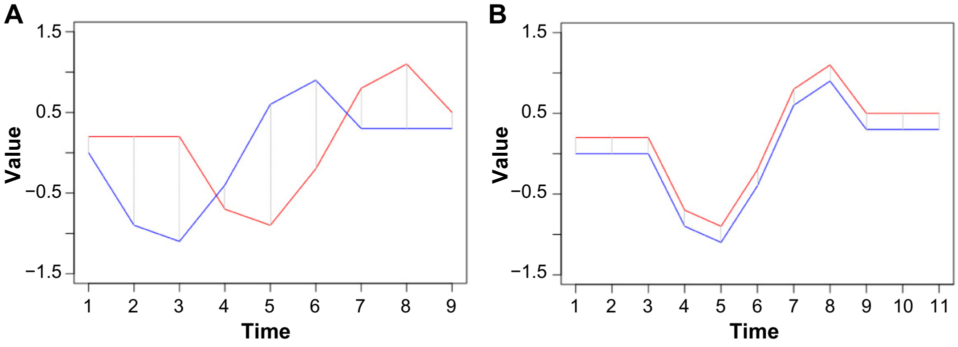

Figure 1 Dynamic time warping for time series alignment.

Notes: (**A**) Two qualitatively similar series. Red is a phase shift of blue with a slight vertical lift. Gray dashed line shows the vertical difference. (**B**) The same series after dynamic time warping alignment.

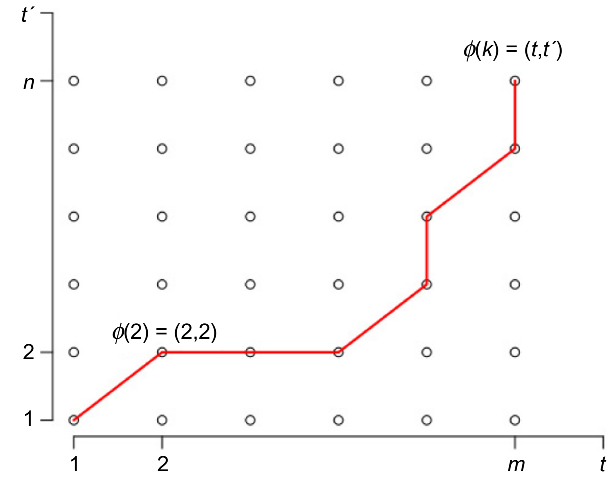

Figure 2 A grid for computing cumulative cost of alignment.

Note that condition 2 does imply that the warping function $\phi\ (k)$ is monotonically increasing. This is important as it avoids loops or other strange warping functions. While there are an infinite number of choices for step weights, in this article we have chosen to explore only two common step patterns that assume that "similar" signals are symmetric. The first choice of weights, Symmetric 1, gives an equal weight for all permissible steps, that is, $w(k) = 1\forall k$. Symmetric 2 weights on the other hand are defined by:

$$\begin{Bmatrix} 1 \text{ if } \phi(k)-\phi(k-1)\in\{(0,1),(1,0)\} \\ 2 \text{ if } \phi(k)-\phi(k-1)=(1,1) \end{Bmatrix} \tag{2}$$

A dynamic programming solution to solving Equation 1 has been previously proposed.[7] To determine the solution, two matrices $\Delta\in\mathbb{R}^{m\times n}$ and $\mathbf{\Gamma}^{(m+1)\times(n+1)}$ are computed. The first matrix $\mathbf{\Delta}$ has entries representing the pairwise distances between points in each time series, that is,

$$\delta(t,t')=d\left(y_{1,t},y_{2,t'}\right) \tag{3}$$

where d is a distance metric. The second matrix $\mathbf{\Gamma}$ represents the cumulative cost matrix that determines how the two series should be aligned. To define the second matrix $\mathbf{\Gamma}$, we begin with stipulating that $\gamma(0,0)=0$, $\gamma(t,0)=\infty$, $\forall t\in\ \{1,2,\ldots,m\}$ and $\gamma(0,t')=\infty$, $\forall\ t'\in\ \{1,2,\ldots,n\}$.

To determine the remaining entries of $\mathbf{\Gamma}$, we use one of the two following recurrence relations (the first applies to Symmetric 1 and the second to Symmetric 2).

$$\gamma_1(t,t')=\min\begin{Bmatrix}\delta(t,t')+\gamma_1(t-1,t'),\delta(t,t') \\ +\gamma_1(t-1,t'-1),\delta(t,t')+\gamma_1(t,t'-1)\end{Bmatrix} \tag{4}$$

$$\gamma_1(t,t')=\min\begin{Bmatrix}\delta(t,t')+\gamma_1(t-1,t'),\delta(t,t') \\ +\gamma_1(t-1,t'-1),\delta(t,t')+\gamma_1(t,t'-1)\end{Bmatrix} \tag{5}$$

Once $\mathbf{\Gamma}$ has been defined, the reverse of the warping path $\phi\ (k)$ can be found by back-tracing from (m,n) to $(1,1)$. This trace is governed by requiring unit step size either horizontally, vertically, or diagonally – whichever selects the minimal entry in the next step.

k-Nearest neighbors classifiers

With the definition of a pseudo-distance metric (DTW distance) for measuring the dissimilarity of two series, a k-nn classifier is a straightforward choice of classifier.[8] To build a k-nn classifier, where $k\in\mathbb{N}$, we denote series data as $\{(\mathbf{y}_i, c_i) : i\in 1,2,\ldots,N\}$, where $\mathbf{y}_i$ is a time series and $c_i = g$ ($g\in$ C denotes the known class of the series). For each observation $(\mathbf{y}_i, c_i)$, we find the neighborhood $N_\kappa\ (\mathbf{y}_i)$ of the κ series $\mathbf{y}_j$ with $i\neq j$ such that these series have minimal DTW$(\mathbf{y}_i, \mathbf{y}_j)$. We then have empirical estimates of the probability of class membership for $(\mathbf{y}_i, c_i)$:

$$\hat{P}_g(\mathbf{y}_i)=\frac{1}{\kappa}\sum_{\mathbf{y}_j\in N_\kappa(\mathbf{y}_i)}1_{c_j=g}(\mathbf{y}_j) \tag{6}$$

From this set of empirical probability estimates, we chose:

$$\hat{c}_i=\underset{g\in C}{\arg\max}\,\hat{P}_g(y_i) \tag{7}$$

In the case that there are ties between two or more classes, the ties are broken at random. $\hat{c}_i$ then represents the DTW distance-based k-nn classification of the observation $(\mathbf{y}_i, c_i)$. Of course, predicting the class of $(\mathbf{y}_i, c_i)$ given that $c_i = g$ is already known is rarely of any interest. Instead, we wish to build a library of series of known class and use this library to predict the class of series for which the class is unknown.

k-Fold cross-validation

A DTW distance-based k-nn classifier has multiple tuning parameters, such as: 1) method of normalizing or preprocessing the series; 2) the choice of weight function in the computation of DTW distances between series; and 3) the choice of k in the k-nn classifier. In order to determine the optimal values of these tuning parameters, we propose using a cross-validation approach.[8] A training set is first used for k-fold cross-validation tuning parameter selection. While the parameter space for each of the three tuning parameters is infinite, we have restricted the search to values we consider reasonable. In this article, we consider two weight functions previously discussed (Symmetric 1 and Symmetric 2). We consider all integers less than the size of the training set as candidate values of k in the k-nn classifier. Finally, we consider two normalization methods: no normalization and mean–variance normalization such that each series has mean 0 and unit variance. Normalization is one feature of this methodology that warrants further exploration. k-fold cross-validation selection of tuning parameters requires a loss function to quantify the loss of a misclassification. For simplicity, we propose using a 0–1 loss function:

$$L\left(c_i, \hat{c}_i\right) = \begin{cases} 0 & \text{if } c_i = \hat{c}_i \\ 1 \, \text{otherwise} & \end{cases} \tag{8}$$

The choice of parameters is sought that minimizes the risk function – the joint expectation of the time series and class membership:

$$R\left(c_i,\ \hat{c}_i\right) = \mathrm{E}\left(L\left(c_i,\ \hat{c}_i\right)\right) \tag{9}$$

Since no functional form of the right-hand side of Equation 9 is available, it is estimated using cross-validation. Using the notation found by Hastie et al,[8] we first randomize the training set data into K folds, which can be represented with the mapping $\kappa : \{1,2,\ \ldots,\ N\} \rightarrow \{1,2,\ \ldots,\ K\}$. The k-fold cross-validation risk estimator is then:

$$\mathrm{CV}\left(\hat{c}, \boldsymbol{\alpha}\right) = \frac{1}{N}\sum_{i=1}^{N} L\left(c_i, \hat{c}_i^{-\kappa(i)}\left(\mathbf{y}_i, \boldsymbol{\alpha}\right)\right) \tag{10}$$

where $\hat{c}_i^{-k}$ denotes the classification constructed with the k th fold removed given tuning parameters $\boldsymbol{\alpha}$. This process is illustrated in Figure 3. Practically, $CV\left(\hat{c}, \boldsymbol{\alpha}\right)$ provides an estimate of the misclassification rate on an independent test set.

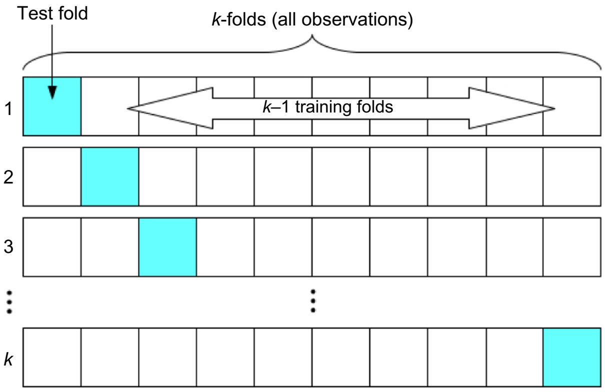

Figure 3 A schematic representation of k-fold cross-validation.

Simulation study

In order to evaluate the performance of a DTW distance k-nn classifier, we conducted a simulation study. For this study, we simulated an experiment in which a classifier that can differentiate between two classes of time series is desired, denoted by ξ and η. The simulation study sought to characterize how well such a classifier would perform in an experimental setting. Each class of time series has a different prototypical feature associated with it. To mimic "observed" series from a realistic setting, noise of random length is placed prior to and following the occurrence of the prototypical feature. Additionally, the feature itself is slightly perturbed by random noise. A set of 20 training time series with half belonging to class ξ and half belonging to class η is then used to train a k-nn classifier in which k-fold cross-validation is used to select tuning parameters. Finally, a test set of 20 time series in which the probability of allocation to either class was equal to 0.5 is then used to evaluate the performance of the classifier. The prototype series for class ξ was defined in continuous time as:

$$f\left(t\right) = \begin{cases} 0 & \text{for } t = 0 \\ \dfrac{\sin\left(t - 2\pi\right)}{t - 2\pi} & \text{for } t \in \left(0, 4\pi\right] \end{cases} \tag{11}$$

The prototype series for class ξ was defined in continuous time as:

$$f\left(t\right) = \begin{cases} 0 & \text{for } t = 0 \\ -\dfrac{\sin\left(t - 2\pi\right)}{t - 2\pi} & \text{for } t \in \left(0, 4\pi\right] \end{cases} \tag{12}$$

Both prototype series are shown in Figure 4A. Two series y_1 and y_2 with differing values of the scale parameter c perturbing the prototype feature is presented in Figure 4B.

To simulate observed series rather than prototype signals, "intro" and "outro" random walks are generated, each having random length. More specifically, two integers m_1 and m_2 are drawn with replacement from the sequence $\{1,2,\ \ldots,\ 40\}$. m_1 and m_2 random walk steps are then generated so that the steps are approximately $N(0,1)$ distributed. The intro and outro random walks are then joined to the prototype feature that has been perturbed using the following equation:

$$y_t = f\left(t\right) + y_{t-1} + \varepsilon_t,\ \ t \in \left\{\frac{k}{10}, k = 1, 2, \ldots, 126\right\} \tag{13}$$

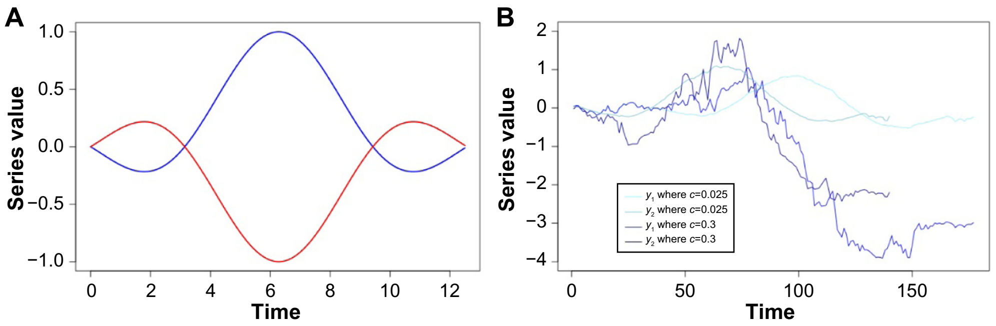

Figure 4 Generation of a prototype feature for each class.
Notes: (**A**) Prototype feature for classes ξ (blue) and η (red). (**B**) Two series y_1 and y_2 with differing values of the scale parameter c perturbing the prototype feature.

where $\varepsilon_t \sim N\left(0,\sigma_t^2\right)$, with $\sigma_t = c \times \sqrt{\left|f\left(t\right)\right|}$, $c \in \mathbb{R}$.

Simulated observed series from each class are shown in Figure 5 alongside the prototype feature for each class. This process for generating the observed series data was used to create the training set consisting of ten series of class η and ten of class ξ. Likewise, the test set was generated consisting of 20 series with the number of series belonging to each class being random with equal class membership probability. A set of training and test series was generated in this manner 10,000 times to simulate 10,000 experiments.

With each simulated experiment, fivefold cross-validation was used on the training set to determine three different tuning parameters. The first tuning parameter considered was the choice of weight function (Symmetric 1 vs Symmetric 2). The second was the choice of k for the k-nn classifier. The final consideration was whether to construct the k-nn DTW distance-based classifier on the raw series or series normalized to have mean 0 and unit variance. The cross-validation used the misclassification rate in order to select optimal tuning parameters. With each experiment, optimal tuning parameter values were found. Subsequently, an independent set of ten series was classified for quantification of error. This set of experiments was conducted for varying values of the scale parameter c. Increasing the value of c decreases the degree of qualitative similarity in the observed prototype feature between series of the same class.

Three scenarios of 10,000 simulated experiments were considered with increasing values of c. A low, medium, and high noise scenario was devised with c values of 0.15, 0.20, and 0.30, respectively. DTW distances were computed using algorithms from the *dtw* package in R.[9]

Simulation study results

The average cross-validation misclassification rate from the three scenarios of 10,000 simulated experiments is shown in Figure 6. For each scenario, the sample mean and variance of the misclassification rate are given in Table 1.

On the simulated training folds for each classifier, a 1-nearest neighbor classifier generally showed the lowest cross-validation misclassification rate. Slight dips in the cross-validation misclassification rate are observed at odd numbers, since no tie breaking occurs given an odd number of neighbors. On the simulated test sets, the classifiers performed very well for c values of 0.15 and 0.20. Classifier

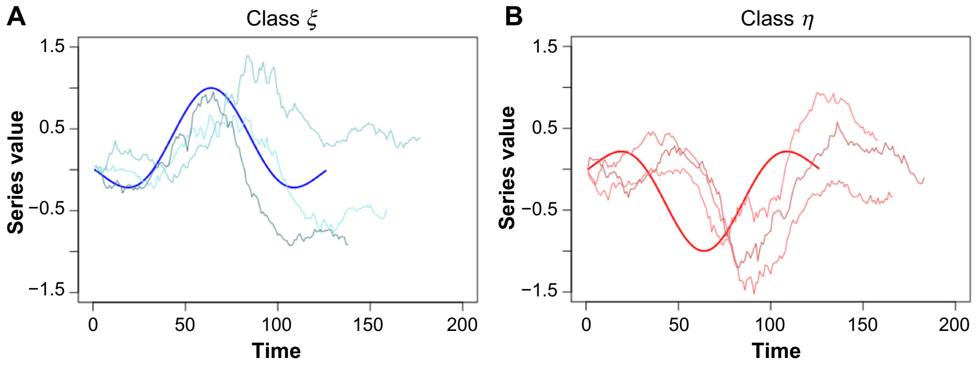

Figure 5 Simulated series from each class.
Notes: (**A**) Three series from class ξ. (**B**) Three series from class η.

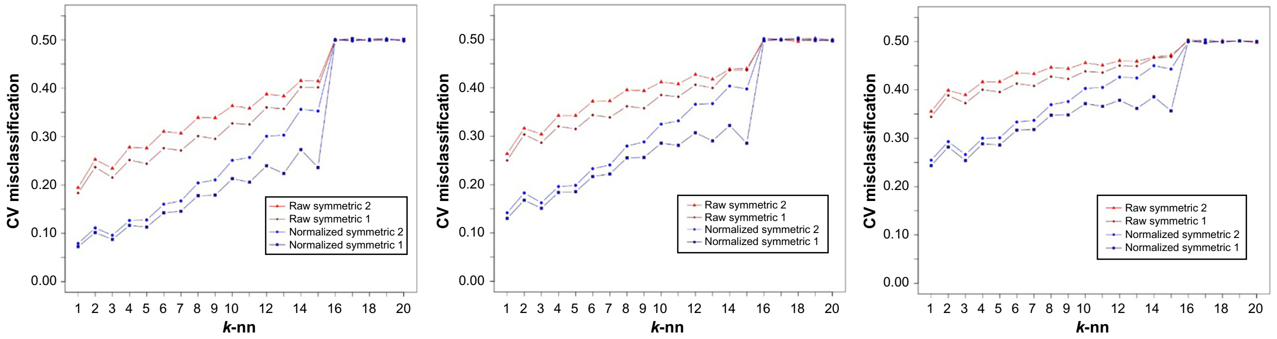

Figure 6 Training set CV misclassification rate as a function of *c*.
Notes: (**A**) *c* = 0.15. (**B**) *c* = 0.20. (**C**) *c* = 0.30.
Abbreviations: CV, cross-validation; *k*-nn, *k*-nearest neighbors.

performance diminished by the value of $c = 0.30$; however, the prototype feature in these series was fairly well perturbed by noise.

Breast cancer cell detection by EpCAM-CNT biosensors

It is becoming increasingly evident that circulating tumor cells (CTCs) in blood play a vital role in determining the spread of metastatic disease to distant sites. The detection of CTCs and their genetic makeup is therefore highly important for understanding the nature and advancement of the disease. Current technologies such as immunomagnetic methods[10] (Veridex) and CTC chips[11,12] are the primary methods to detect CTCs in blood from patients. While these technologies are impressive, they are not necessarily optimal for rapid identification in clinic. A digital device by which small drops of blood could be rapidly analyzed by microarrays for the detection of CTCs in the clinic would thus have great utility.[13,14] We hypothesized that a CNT device functionalized with an antibody would allow for the rapid detection of CTCs. These devices rely on the principle that each cancer cell possesses thousands of overexpressed particular target receptors, so that cooperative binding to cognate antibodies would yield characteristic spikes in the electrical signal due to free energy change. The reduction in free energy for specific interactions is much higher than nonspecific interactions, and one can use CNT arrays to transduce the change in free energy into electrical signal.[14] From such a CNT array, a characteristic signature indicating specific interactions (CTCs present) versus a characteristic signature indicating only nonspecific interactions (CTCs not present) could be captured. Details of the nanofabrication devices used in this article have been previously discussed by Khosravi et al.[14]

Table 1 Classifier performance during simulation study

Noise parameter (c)	Misclassification rate	
	Mean	Variance
0.15	0.074	0.069
0.20	0.131	0.114
0.30	0.230	0.177

To evaluate if devices could differentiate between MCF-7-positive samples and MCF-7-negative samples, both were tested utilizing the nanotube microarrays.[14] Human Buffy Coats were used from Biorepositories at the University of Louisville. The study was approved by the University of Louisville Institutional Review Board (IRB) #10.0428 and the IRB at Worcester Polytechnic Institute IRB #00007374. The negative samples consisted of a buffy coat sample (a layer of centrifuged blood) without the presence of breast cancer cells. The positive samples consisted of a buffy coat sample that was spiked with MCF-7 breast cancer cells. The drain current from the CNT devices was recorded continuously throughout each experiment. The resulting data consisted of time series with characteristic "spikes" occurring after the application of the sample. Two MCF-7 positive and two MCF-7 negative samples are depicted in Figure 7.

A train–test approach was then used to construct a DTW distance-based *k*-nn classifier as discussed in. The training data consisted of ten buffy coat samples and 17 spiked buffy coat samples. *k*-fold cross-validation parameter selection was conducted using tenfold cross-validation on 10,000 bootstrapped samples from the training set of 27 signals for which the class (buffy vs spiked buffy) was known. The tuning parameters selected were those that minimized the mean and variance of the misclassification rate. Once tuning parameters had been selected, 22 test signals (of class unknown to the personnel constructing the classifier) were classified using a DTW

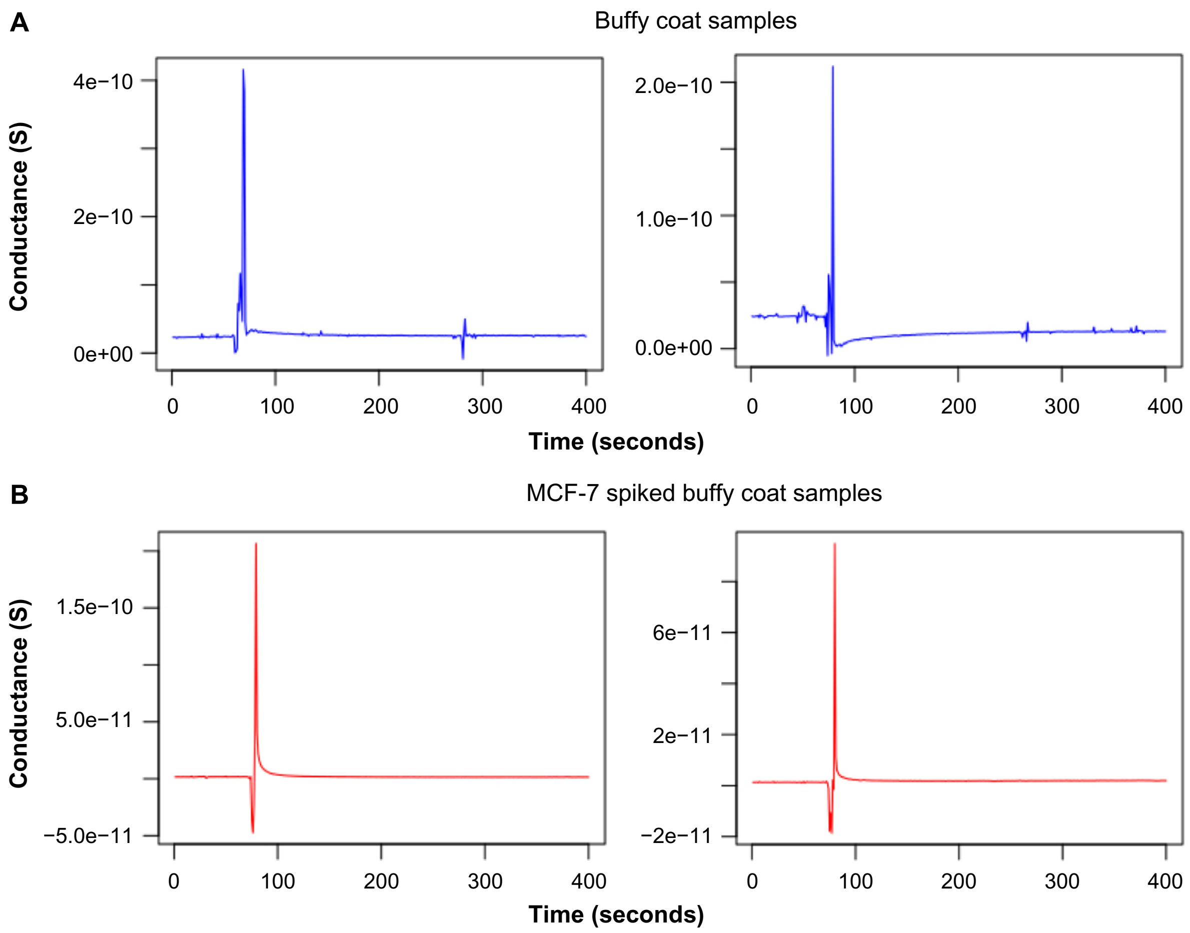

Figure 7 Plots of representative electrical signatures from samples used in the construction of a *k*-nn DTW distance-based classifier for breast cancer surface marker profiling.

Notes: (**A**) Two buffy coat samples. (**B**) Two spiked buffy coat samples.

Abbreviations: *k*-nn, *k*-nearest neighbors; DTW, dynamic time warping.

distance-based *k*-nn classifier using the training signals as reference signals. The test set misclassification rate, classifier sensitivity, and classifier specificity were then used as criteria to measure the success of the devices in discriminating between positive and negative samples. Sensitivity is defined as TP/*P*, where TP represents the number of correctly classified positive (MCF-7 spiked) samples and *P* represents the total number of positive samples. Similarly, specificity is defined as TN/*N*, where TN represents the number of correctly classified negative samples (plain buffy) and *N* represents the total number of negative samples.

Results

A heat map of the DTW signal distances and reference signals used in the *k*-fold cross-validation tuning parameter selection is shown in Figure 8. In the margins of this figure, a dendrogram of the complete-linkage agglomerative hierarchical clustering[8]

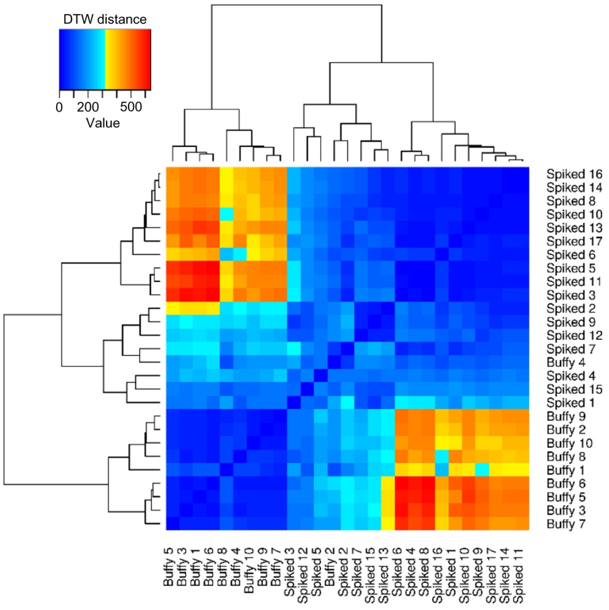

Figure 8 A heat map of training signals.

Abbreviation: DTW, dynamic time warping.

Table 2 A confusion matrix for the *k*-nn DTW-based classifier for detecting breast cancer cells in buffy coat

True class	Predicted class	
	Buffy	Spiked buffy
Buffy	9	2
Spiked buffy	1	10

Abbreviations: *k*-nn, *k*-nearest neighbors; DTW, dynamic time warping.

of the same is shown. This demonstrates that on the training data, DTW distance as a dissimilarity measure naturally partitions the sample data into two distinct clusters according to the sample class.

Tenfold cross-validation on the bootstrap samples resulted in a final DTW distance-based *k*-nn classifier utilizing a symmetric two-step pattern and three nearest neighbors. A confusion matrix for the test data is presented in Table 2. The classifier detected the MCF-7 positive samples with 90.9% sensitivity, 81.8% specificity, and a misclassification rate of 13.6%.

Discussion

In this article, we have presented a classification methodology that allows for the discrimination of biological sample types by biosensors that produce electrical signals or other time series data. Such a classifier is well suited for determining the class of series in which there exists a prototypical feature that heralds the class the series belongs to, but the time index is either irrelevant, stretched, or compressed. We have used this methodology to discriminate MCF-7 breast cancer cells present in buffy coat from buffy coat samples without cancer cells by CNT biosensor devices. By demonstrating the ability of a *k*-nn DTW distance-based classifier to predict the class of the biological sample – whether breast cancer cells are present or not – we have also demonstrated the efficacy of CNT devices for detecting spiked cancer cells in turbid media such as buffy coats. This technique could be useful for screening cancer cells from tissues, liquid biopsy samples, and CTCs in blood in a rapid manner. The handheld and portable nature of the device will be useful for clinical translation.

An important feature of this methodology is extensibility. Many alternative step sizes and alignment constraints have been proposed for the warping path used to minimize the DTW distance.[6] Additionally, there are an infinite number of choices for the distance metric used in the computation of a DTW distance. While we have chosen Euclidean for its familiarity, other distance metrics may be better suited for specific time series data. Likewise, a *k*-nn classifier was chosen for simplicity. However, the use of metric multidimensional scaling[15] to map a DTW pseudo-distance matrix constructed from biological time series to a real coordinate space, $\mathbb{R}^n$, would allow for the use of other classifiers such as linear discriminant analysis or support vector machines.

Acknowledgments

This work was partially supported by the National Cancer Institute through the awards 1R15CA156322 and 7R15CA156322, NSF CMMI 1463869, NSF DMR 1410678, and NSF ECCS 1463987 for B Panchapakesan. Dr SN Rai received support from the Wendell Cherry Chair in Clinical Trial Research and from Dr DM Miller, Director, James Graham Brown Cancer Center, University of Louisville.

Disclosure

The authors report no conflicts of interest in this work.

References

1. Berndt D, Clifford J. Using dynamic time warping to find patterns in time series. In: Fayyad UM, editor. *Advances in Knowledge Discovery and Data Mining*. Cambridge, MA: AAAI/MIT Press; 1994.
2. Fu T. A review on time series data mining. *Eng Appl Artif Intell*. 2011;24:164–181.
3. Müller M. *Information Retrieval for Music and Motion*. New York: Springer; 2007.
4. Alcaraz R, Hornero F, Rieta J. Dynamic time warping applied to estimate atrial fibrillation temporal organization from the surface electrocardiogram. *Med Eng Phys*. 2013;35:1341–1348.
5. Tormene P, Giorgino T, Quaglini S, Stefanelli M. Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. *Artif Intell Med*. 2009;45:11–34.
6. Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans Acoust*. 1978;26:43–49.
7. Rabiner L, Juang B. *Fundamentals of Speech Recognition*. Upper Saddle River, NJ: PTR Prentice-Hall, Inc.; 1993.
8. Hastie T, Tibshirani R, Friedman JH. *The Elements of Statistical Learning*. New York: Springer; 2008.
9. Giorgino T. Computing and visualizing dynamic time warping alignments in R: the DTW package. *J Stat Softw*. 2009;31:1–24.
10. Cristofanilli M, Budd GT, Ellis MJ, et al. Circulating tumor cells, disease progression, and survival in metastatic breast cancer. *N Engl J Med*. 2004;351:781–791.
11. Nagrath S, Sequist LV, Maheswaran S, et al. Isolation of rare circulating tumour cells in cancer patients by microchip technology. *Nature*. 2007;450:1235–1239.
12. Stott SL, Hsu CH, Tsukrov DI, et al. Isolation of circulating tumor cells using a microvortex-generating herringbone-chip. *Proc Natl Acad Sci U S A*. 2010;107:18392–18397.
13. Shao N, Wickstrom E, Panchapakesan B. Nanotube-antibody biosensor arrays for the detection of circulating breast cancer cells. *Nanotechnology*. 2008;19:465101.
14. Khosravi F, Trainor PJ, Rai SN, Kloecker G, Wicstrom E, Panchapakesan B. Label-free capture of breast cancer cells spiked in buffy coats using carbon nanotube antibody micro-arrays. *Nanotechnology*. 2016;27:13LT02.
15. Cox TF, Cox MAA. *Multidimensional Scaling*. London: Chapman and Hall; 2001.