Early Classification of Multivariate Time Series Using a Hybrid HMM/SVM model

Mohamed F. Ghalwash*[§], Dušan Ramljak*, Zoran Obradović*[‡]
*Center for Data Analytics and Biomedical Informatics, Temple University, Philadelphia, USA
[§]Mathematics Department, Faculty of Science, Ain Shams University, Cairo, Egypt
[‡]corresponding author: zoran.obradovic@temple.edu

***Abstract*—Early classification of time series has been receiving a lot of attention as of late, particularly in the context of gene expression. In the biomedical realm, early classification can be of tremendous help, by identifying the onset of a disease before it has time to fully take hold, or determining that a treatment has done its job and can be discontinued. In this paper we present a state-of-the-art model, which we call the Early Classification Model (ECM), that allows for early, accurate, and patient-specific classification of multivariate time series. The model is comprised of an integration of the widely-used HMM and SVM models, which, while not a new technique per se, has not been used for early classification of multivariate time series classification until now. It attained very promising results on the datasets we tested it on: in our experiments based on a published dataset of response to drug therapy in Multiple Sclerosis patients, ECM used only an average of 40% of a time series and was able to outperform some of the baseline models, which needed the full time series for classification.**

***Keywords*- early classification; early classification of multivariate time series;**

I. Introduction

A time series of several distinct (and sometimes not-so-distinct) classes can be created, by observing various patient attributes over time. Since many clinical applications tend to be very time sensitive, being able to use a shorter time interval for classification is often more favorable than having a slightly more accurate result. For example, observing only the first portion of a patient's data may allow doctors to identify a deadly condition before it has time to fully manifest, or to be able to safely stop treatment after it has cured the patient and before it can do unnecessary damage. Early classification techniques are therefore beneficial for this realm, as they offer the very favorable tradeoff of using less of a time series at the cost of some accuracy of the result. The challenge of adapting early classification for biomedical purposes is ensuring that the result is obtained early enough to make a difference, and also accurate enough for the doctor to be confident in the model's result. The goal of this paper is to introduce an early classification model that, by training on a full time series, offers classification at a very early time point during the testing phase, while staying competitive in terms of accuracy with other models that use full time series both in training and testing.

Analysis of biomedical data has been a popular topic for many years, although using this kind of data for early classification is a far less explored field. Models that are naturally able to leverage temporal information, such as HMM, have been shown to be particularly effective for biomedical data since the temporal aspect plays a vital part in describing the data. Furthermore, since HMMs are resilient to having missing values in the data, they seem like a natural fit for biomedical applications, which often contain a significant number of missing values due to the various human and technological factors involved in obtaining the data. Lin et al. [1] adapt HMMs to classify instances based on time series gene expression data, and their results reinforce the effectiveness of HMMs in a biomedical context.

Another important issue often encountered when dealing with gene expression time series data is the varied response rates problem. Due to the differences in internal chemistries of each patient, the same condition can take different amounts of time to manifest in different individuals. Thus, while the overall pattern of the expression profile for a particular condition may be very similar in each patient, it can show up as stretched or squished in time, making it much harder to detect accurately. By building a hybrid model we were able to not only provide early classification for multivariate time series, but also address this problem.

Although attaining accurate classification is the primary goal of Machine Learning (ML) task, in many applications it is important to attain decisions that are not only accurate, but can also be easily interpreted and obtained early.

This paper introduces a hybrid model specifically designed for early and accurate classification of multivariate biomedical time series data. In addition to offering good performance and outperforming several baseline methods applied to a number of datasets, we successfuly address the early classication by getting our decisions using on average only 40% of time series when testing.

The remainder of the paper is organized as follows. Section I-A discusses some of the related work pertinent to this paper, including works discussing the potency of HMMs and their applications to gene expression data. In Section II we provide an explanation of our hybrid model as well as an in-depth description of how the hybrid model is trained and used for early classification. Section III includes a description of the various datasets we used, an explanation of our experimental setup, and the results we obtained in our experiments. Finally, Section IV includes concluding remarks and possible avenues for future research.

A. Related work

Due to their flexibility and popularity in the field of statistics, HMMs have long been the model of choice for a variety of applications, including speech recognition, natural language processing, and, more recently, gene expression analysis. In his timeless tutorial on HMMs, Rabiner [2] provided the first in-depth examination of the mathematics behind these models in less abstract environments, offering insights into how these models could be used to effectively model real world systems. His analysis of the application of HMMs for speech recognition spurred researchers to adapt HMMs for this purpose, generating a vast number of papers on the topic. In his survey, Jiang [3] summarizes a number of these papers, focusing on methods that utilize discriminative training techniques to provide significant improvements in the speech recognition fields. However, in our experiments it required more computational time, without an improvement in performance.

As the aforementioned survey shows, the notion of combining generative and discriminative techniques into a single model has been around for some time. One paper that offers a very intuitive and adaptable approach to creating such a model, which was not included in the survey, was written by Huang et al [4]. Although Huang's model was designed for the task of online handwriting symbol recognition, we found that the underlying approach used to build this model could easily be adapted to the realm of early classification of gene expression time series. Despite the very different natures of the handwriting symbol and gene expression time series datasets, we were able to create an Early Classification Model (hereinafter ECM) using the same general approach. The method inherits the practical and dynamical modeling abilities from HMMs, and the robust discriminating ability from SVMs is vital for classification tasks. However, we are exploring every relevant segment length at every possible starting point, which was not done in previous work.

The application of HMMs to the field of gene expression time series classification is much more recent than their uses in speech recognition, but there has already been a significant amount of research performed in this area. Among the earliest efforts of utilizing the potency of HMMs for gene expression analysis was Schliep's [5] paper. Although focused on clustering the gene expression time series, rather than on classifying them, the authors nonetheless managed to show the utility of HMMs in the context of gene expression. This utility was attained primarily due to the fact that HMMs leverage the temporal dependencies of the data, whereas methods that ignore these dependencies lose a lot of information, as evidenced by the difference in the quality of the clusters generated by both kinds of methods.

The work of Lin et al. [1] is also noteworthy, as they were one of the forerunners of using HMMs for classification of gene expression time series. However, despite the novelty and efficiency of Lin's model, it was incapable of performing early classification of the time series. In their paper several experiments were done to perform classification at each time point using shorter time series. Their model was not patient-specific since the length of segments considered was very inflexible; that is, if the model is trained on time series of length 3 the prediction is always done at the 3^{rd} time point. Our ECM model addresses this issue directly, and is, to the best of our knowledge, the first model that is capable of patient-specific early classification of multivariate gene expression time series.

In an early classification context, the objective is to provide patient-specific classification of unknown time series as early as possible. So, instead of utilizing the whole time series, our ECM method looks into a portion (current stream) of the unknown time series and determines whether it is able to predict the label of the whole time series without looking to the rest of the time series. If ECM is able to predict at the time point which is at the end of the current stream the label is predicted. Otherwise, ECM requires more data for the unknown time series and looks at a larger segment, and does so until it is able to predict the label of the time series.

While the task of early classification is by no means new, its application to multivariate time series is an extremely unexplored field. In fact, to the best of our knowledge, the only other research in this area was performed by Ghalwash and Obradovic [6]. Their method MSD (Multivariate Shapelet Detection) adapts the use of time series segments called shapelets, which were originally developed for univariate time series, for early classification of multivariate data. Time series patterns were extracted from all dimensions of the time series that distinctly manifest the target class locally. The time series were classified by searching for the earliest closest patterns. While this shapelet-based approach was able to outperform a number of baseline models, it had several limitations: the components of the multivariate time-series shapelet needed to have the same starting positions, and the model could not handle missing values. Moreover, MSD was unable to address the varied response rate problem that is often encountered in clinical data.

II. Model Description

A. Hybrid Model for Early Classification

In this study, we propose using a hybrid model that combines a generative model (HMM) with a discriminative model (SVM) for early classification of a multivariate time series. The HMM, referred to as Λ, is used to learn the distribution of the patterns in a time series gene expression training dataset. The emission probability of the HMM is assumed to be drawn from a multivariate Gaussian distribution whose dimension equals the number of genes. The covariance matrix is assumed to be diagonal to avoid overfitting and to allow for fast computation of the model. Lin et al. [1] showed that using 2 states allows for efficient alignment of patients with different response rate, as the 2 states represent 2 phases of the response (disease in our context). The SVM takes the outputs of the HMM models as input: log likelihoods of segments in time series gene expression training dataset. The role of SVM is to determine the combination of HMM models that could be trusted when classifying the time series based on the current stream.

The reasoning behind using a hybrid model is that if a generative model encounters a short segment that is repeated often in both classes (which can be a fairly common occurrence in gene expression data), it will likely not be able to distinguish between the two classes effectively. By applying a discriminative model on top of the generative model's output, we are able to overcome this problem and improve overall performance. The details of how the hybrid model is trained and applied to the test data are provided next.

1) Data Preparation: The dataset is first divided into training and test data. The test data remains untouched until the very end, and all of the following procedures are applied to only the training data. When we mention test time series in the Training Phase, we are referring to the portion of the training data held out for testing for a particular fold of the internal cross-validation.

2) Training Phase: In general, several HMMs are trained on short time series segments and their log likelihoods on validation data are generated as features for the SVM. We opted for using an RBF kernel SVM.

STEP1: Train HMMs of specific length
Given a time series of length L from one class, we extract all disjoint segments of length l, starting from position 0, and an HMM is trained on those segments. We then shift the starting position by one time point, extract all disjoint segments of length l starting from position 1, and train another HMM on those segments. If $l <= L/2$, we generate l HMM models. We vary the length and the appropriate shifts in order to capture every possible pattern, since these patterns can have different lenghts and start at different positions. In general, if $l > L/2$, then $L - l + 1$ models are generated.

STEP2: Train HMMs for different lengths
We repeat the above procedure for different segment lengths l up to $l+k$. k is a user-defined parameter, set small enough to be able to capture all possible patterns, but $l+k$ should not exceed 50% of the time series to still be able to identify the pattern as early as possible. Hence, the maximum number of models to be generated is $(k+1)(k/2+l)$. The same technique is applied to the time series of the other class, resulting in a maximum of $N = 2(k+1)(k/2+l)$ models.

STEP3: Generate features for SVM
We apply all trained HMMs on the test data as follows: first, we take the shortest time series segment O extracted from the training data. If the length of that segment is l, we then read l time points from the test time series (O_l). Next, we ask all N HMMs to generate their log likelihoods on the test time series we are currently examining. In general, an HMM is able to generate a log likelihood on the test time series if the model is trained on segments of length shorter than or equal to the length of the test time series. If the length of the current test time series is greater than l, then the HMM uses only the last portion of the current time series, such that the length of this portion has the same length as the segments used for training the HMMs. The log likelihoods generated in this fashion, $\log P(O_l|\Lambda_j); j = \{1, \ldots, N\}$, are then considered as features for an SVM.

The next part of this step is to read one more time point from the time series, so that the current test time series length is incremented by one. All models are asked to provide their log likelihoods on this time series to be another row for the SVM. This process is repeated until we have read all of the time points of the test time series. After this stage, we will have generated (L - l + 1) instances with dimensionality N for the SVM, all of which have the class label of the test time series. Finally, we repeat this procedure for all test time series.

We repeat steps 1 - 3 five times to obtain 5-cross validations. Here we reinforce the fact that the HMMs are not used for prediction of the class of the time series, but rather as data for the SVM, which is made aware of the class labels independent of the HMMs.

After the cross validation is done, the log likelihoods generated by HMMs are ready to be used as features for the SVM. The parameters of the SVM are optimized using cross validation on the log likelihoods generated by the HMMs. Then another 5-cross validation is conducted for SVM to generate a score for each example. We use the same partitions of the 5-cross validations used for the HMM training phase to avoid any bias. When testing the SVM, the model computes a posterior probability estimate $P(y|x)$ of the class y for each example x. Namely, in a binary classification context, an SVM gives two probabilities: $P(y_1|x)$ and $P(y_2|x)$ (note that $P(y_2|x) = 1 - P(y_1|x)$). The scores for all examples are computed as the difference between the two probabilities $P(y_1|x) - P(y_2|x)$. Using the scores of all examples and their labels, we identify (using a standard ROC curve approach) the best threshold that maximizes the balanced accuracy (average between sensitivity and specificity). Finally, the SVM is trained on the training data and all HMMs are trained on the same data.

3) Testing Phase: We note here that the test data allocated in the very beginning of the procedure is never used for training either the SVM nor HMM models. For a given time series with an unknown label, we read l time points from the test time series. We then ask all HMM models to provide log likelihoods. We use their outputs as inputs to the SVM, as described above. The score of the current time series is then computed using the probabilities generated by the SVM as $P(y_1|x) - P(y_2|x)$. If the score is greater than the threshold, then the current time series is most likely from class y_1, otherwise, it is from class y_2. We then ask if the probability estimate for the selected class is high enough to be confident about the prediction. A user-defined parameter is used to measure this level of confidence. If the probability estimate is higher than the confidence level, we stop at the current time point and predict the time series. Otherwise, we read one more time point from the current time series and repeat the procedure. If we reach the end of the time series and are not able to classify the time series, we mark that test time series as "not covered time series".

4) Whole Process: The above training and testing phases are repeated 10 times using 10 different partitions (10 cross validations). The average accuracy, as well as the coverage rate (how many time series out of the test time series data were classified) and the fraction of the time series used for classification, are reported.

III. Experiments

A. Experimental setup

All of the procedures have been implemented in C++ and is publically available at `www.dabi.temple.edu/~mohamed/ECM`. We used libSVM that provides probability estimates [7]. All experiments were conducted on a PC Intel Core i7 2.8 GHz with 8GB RAM.

B. Dataset description and results

Time series gene expression experiments have been used in a variety of biomedical applications [8]. These experiments consist of building expression profiles, which are essentially functions that model the changes in the expression levels of various genes. Such changes are collected by performing multiple microarray experiments over a period of time [9]. Each experiment usually measures the expression level of multiple genes at a given time point. In this context, the gene expression time series classification problem is the process of determining the class of a previously unseen time series, based on the expression levels recorded for the time series in the training dataset. These classes can be defined in a variety of ways to help answer different kinds of questions. For example, will patient X respond well to a certain therapy or drug treatment? Is patient X going to develop sepsis in the next few hours? Is patient X recovering from a disease?

A clinical dataset, which we will refer to as MS70, was generated to study the changes in cellular functions in multiple sclerosis patients in response to drug therapy with IFNβ [10]. The dataset contains time series gene expression values for 52 patients. The patients were classified as good responders (33 patients) or bad responders (19 patients) to the drug. Blood samples were taken every 3 months in the first year and every 6 months in the second year. Some patients miss certain measurements, especially at the 7th time point. Thus, the gene expression values were measured an average 5-7 times for each subject. In order to adhere to the limitations of clinical settings (in which only a small, pre-specified number of genes is provided) and to be able to effectively compare ECM results with those attained in other studies, several datasets comprised of a fairly small number of genes were generated.

The identification of triplets of genes for a Bayes classifier of time series gene expression data of multiple sclerosis patients' response to a drug was performed in [10]. Previous research identified 12 genes in terms of triplets. Hence, we generated four datasets: Bar3A and Bar3B, which consist of one triplet of the best two triplets of genes; Bar6, which has the top two triplets; and Bar12, which has all 12 genes identified by all triplets. A discriminative HMM has been developed and applied to the MS70 dataset to reveal the genes that are associated with the good or bad responders to the therapy [1]. A total of 9 genes were found that are associated with the therapy, 7 of which are identified using the last time stamp. Hence, we constructed two datasets, called Lin9 and Lin7, consisting of 9 and 7 genes, respectively. The entire MS70 dataset, containing all 70 genes, was also used for our experiments. A mixture of hidden Markov models has been developed to identify the genes that are associated with patient response to the treatment [11]. A total of 17 relevant genes were found, so we constructed a dataset called Cos17 that is comprised of the 17 genes.

In order to have a possibility of effectively comparing our method with other methods proposed in literature we first created 8 datasets that posses the same characteristics as those presented in [10], [1], [11]. 3 of these datasets were used to determine the best confidence value to be used by our model, since it was the user-defined parameter in our method. In this set of experiments, we trained our model with a distinct value for the confidence parameter (0.4, 0.5, and 0.6) on Bar6, Bar12, and Lin9, respectively. Note that although the possible values of the confidence parameter are in the [0,1] range we noticed that accuracy drops for values of confidence parameter smaller than 0.4 and coverage suddenly drops when we choose values greater than 0.6. Both trends could be easily explained: if we allow decisions when we are not confident our model will decide early, but the accuracy will not be sufficient. On the other hand if our requests for confidence are too high we have generally higher relative accuracy scores, at the expense of coverage and earliness, and we might end up without decisions until the last time stamp. So, we omitted values below 0.4 and above 0.6, for the set of experiments dealing with gene expression classification. Finally, we opted to use

Table I
CLASSIFICATION ACCURACIES (ACC), COVERAGE (COV), AND EARLINESS (EAR) IN PREDICTION AT DIFFERENT LEVELS OF CONFIDENCE (CONF) ON VARIOUS DATA SETS.

		BAR6	BAR12	LIN9
ACC	CONF = 0.4	62	85	87
	CONF = 0.5	66	85	85
	CONF = 0.6	71	83	86
COV	CONF = 0.4	96	100	100
	CONF = 0.5	92	100	100
	CONF = 0.6	86	98	94
EAR	CONF = 0.4	44	43	43
	CONF = 0.5	49	43	44
	CONF = 0.6	51	46	50

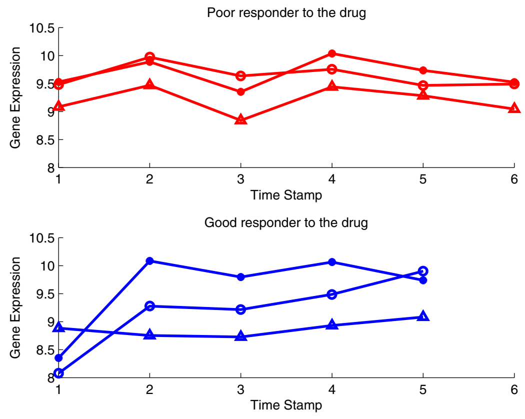

Figure 1. In the top panel, 3 genes expression time series for patient who is a poor responder to the drug are represented. In the bottom panel, 3 genes expression time series for patient who is a good responder to the drug are represented. The subjects have been correctly classified at the 3^{rd} and 4^{th} time point, respectively.

a value of 0.5 for the confidence parameter, as this value seemed to offer the middle-ground of the various tradeoffs evident in Table I.

To further explain the effectiveness of our approach we show a real case from the Bar3A dataset. Figure 1 illustrates a 3-dimensional gene expression time series (genes Caspase 2, Caspase 10, and FLIP) observed at 6 time steps. In the top panel, a patient who is a poor responder to the drug is correctly classified by our hybrid approach at the 3^{rd} time point. In the bottom panel, a patient who is good responder to the drug is correctly classified at the 4^{th} time point. Although the patterns of both patients look similar to each other such that it is hard to distinguish between them by eye, our hybrid model was able to classify them correctly and early.

Figure 2 compares different values for the user parameter confidence, which measures the confidence level the user expects from the model. Since the relative accuracy is barely affected by the value of confidence (in the specified 0.4 - 0.6 range) over the eight datasets, it is clear that the model is

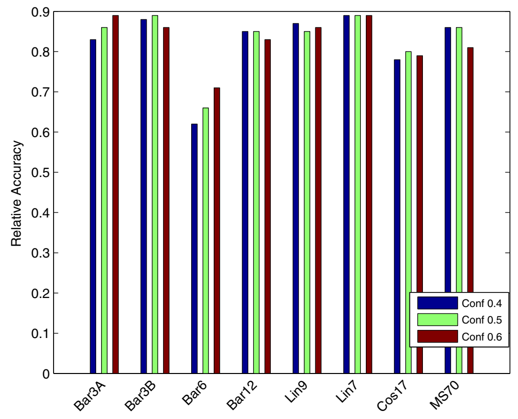

Figure 2. Measuring the sensitivity of the user parameter confidence on the relative accuracy for different datasets.

insentitive to the parameter, and the results are fairly robust.

The outcome of applying our model to the remaining 5 datasets (out of the 8 that we initially created) is shown in Table II. The relative accuracy of our model on each of the datasets is reported, along with the performance of the best models presented in literature that were applied to the same datasets. Note that the accuracy of the "best" models is based on using the entire time series, while the accuracy of our model is based on using, on average, only 3/7 of the testing time series and obtaining patient-specific decisions. This important fact emphasizes the effectiveness of our method: even though the best models on average outperform our model in the accuracy department, we are able to provide competitive accuracy scores while using a significantly smaller portion of the full time series. In two of the datasets we were actually able to outperform the best models, in spite of utilizing less than a full time-series when testing, reinforcing our model's efficiency. The performance of MSD on all of these datasets is also reported, although it performed rather poorly across the board, and could not even handle the MS70 dataset.

The accuracy of the model developed by [1] when using a shorter time series of fixed length in training and testing (Figure 3), further displays the strength of our model,

Table II
CLASSIFICATION ACCURACIES AND EARLINESS IN PREDICTION (IN PARENTHESES) FOR MSD, THE BEST MODELS IN LITERATURE, AND OUR APPROACH, ON VARIOUS DATA SETS. ~ MEANS ON AVERAGE.

DATA SET	MSD	BEST	ECM
BAR3A	72.7(~3/7)	**87.8**(7/7)	86.2(~3/7)
BAR3B	65.2(~3/7)	87.5(7/7)	**88.5**(~3/7)
LIN7	70.8(~4/7)	85.0(7/7)	**88.7**(~3/7)
COS17	66.7(~3/7)	**92.7**(7/7)	79.7(~3/7)
MS70		**81.4**(7/7)	72.0(~3/7)

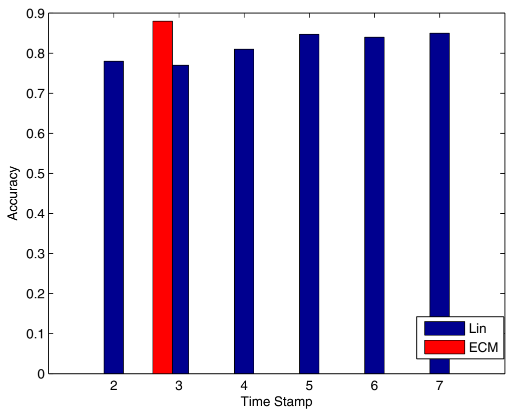

Figure 3. Comparison of the accuracy when using a shorter time series of fixed length in training and testing (Lin's model) and ECM (different segment lengths and different shifts) on Lin7 dataset. ECM accuracy at the average earliness, consisting of 3 time points, is shown. Lin's results on all time stamps were worse than ECM results.

considering that the only other method capable of early classification in a mulitvariate context was less accurate.

IV. Conclusion

In this paper we proposed a novel application of a hybrid HMM/SVM model in the realm of biomedical data. Our ECM is able to handle multivariate gene expression time series data, and is capable of performing patient-specific early classification on this kind of data with unprecedented results. On datasets gathered from real-world scenarios, our method achieved relative accuracy scores that were on average slightly lower than the best results presented in literature when models were tested on the same datasets (although on two of these datasets, our model had a higher accuracy score). However, while the models from literature attained their best results after observing the full time series, our model was able to keep up by utilizing only 40% of the whole time series, thus displaying its early classification potential. Our approach was also able to significantly outperform the only other method designed specifically for early classification of multivariate time series.

In closing, one of the assumptions of using HMMs is that there are enough examples available for the model to accomplish a sufficient amount of learning. Because medical data is often noisy and expensive to obtain, this assumption cannot always be met when real data is involved (in some real-life scenarios, the data contains only 10 or 15 examples), rendering our ECM approach useless. We intend to address this issue in the future, while still allowing for multivariate time series to be classified early.

Acknowledgment

We thank Alexey Uversky for proofreading efforts and both Alexander Yates and Alexey Uversky for valuable discussions.

This work was funded, in part, by DARPA grant [DARPA-N66001-11-1-4183] negotiated by SSC Pacific grant; and the Egyptian Ministry of Higher Education.

References

[1] T. Lin, N. Kaminski, and Z. Bar-Joseph, "Alignment and classification of time series gene expression in clinical studies," *Bioinformatics*, vol. 24, pp. i147–i155, Jul. 2008.

[2] L. Rabiner, "A tutorial on HMM and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[3] H. Jiang, "Discriminative training of HMMs for automatic speech recognition: A survey," *Computer Speech and Language*, vol. 24, pp. 589–608, Oct. 2010.

[4] B. Q. Huang, C. J. Du, Y. B. Zhang, and M.-T. Kechadi, "A hybrid HMM-SVM method for online handwriting symbol recognition," in *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, ser. ISDA '06, vol. 1. Washington, DC, USA: IEEE Computer Society, 2006, pp. 887–891.

[5] A. Schliep, A. Schönhuth, and C. Steinhoff, "Using hidden Markov models to analyze gene expression time course data," *Bioinformatics*, vol. 19, no. suppl 1, pp. i255–i263, Jul. 2003.

[6] M. Ghalwash and Z. Obradovic, "Early classification of multivariate temporal observations by extraction of interpretable shapelets," *BMC Bioinformatics*, vol. 13, Aug. 2012.

[7] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[8] T. M. Alain B Tchagang, Kevin V Bui and P. V. Benos, "Extracting biologically significant patterns from short time series gene expression data," *BMC Bioinformatics*, vol. 10, Aug. 2009.

[9] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Molecular Biology of the Cell*, vol. 9, pp. 3273–3297, Dec. 1998.

[10] S. E. Baranzini, P. Mousavi, J. Rio, S. Caillier, A. Stillman, P. Villoslada, M. M. Wyatt, M. Comabella, L. Greller, R. Somogyi, X. Montalban, and J. Oksenberg, "Transcription-based prediction of response to IFNβ using supervised computational methods," *PLoS Biology*, vol. 3(1), pp. 166–176, 2005.

[11] I. Costa, A. Schnhuth, C. Hafemeister, and A. Schliep, "Constrained mixture estimation for analysis and robust classification of clinical time series," *Bioinformatics*, vol. 25(12), pp. i6–i14, 2009.