

Mining of Multivariate Temporal Biological Data: A Framework for the Rational Design of Data- Driven Models

*R.T. Kamimura, S. Bicciato, H. Shimizu, J. Alford, and
G.N. Stephanopoulos*

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

This article was submitted to
BioKDD, 2001: Workshop on Data Mining in Bioinformatics, San
Francisco, CA, August 26-29, 2001

May 10, 2001

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

This work was performed under the auspices of the United States Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

This report has been reproduced directly from the best available copy.

Available electronically at <http://www.doc.gov/bridge>

Available for a processing fee to U.S. Department of Energy
And its contractors in paper from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-mail: reports@adonis.osti.gov

Available for the sale to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-mail: orders@ntis.fedworld.gov
Online ordering: <http://www.ntis.gov/ordering.htm>

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
<http://www.llnl.gov/tid/Library.html>

MINING OF MULTIVARIATE TEMPORAL BIOLOGICAL DATA: A FRAMEWORK FOR THE RATIONAL DESIGN OF DATA-DRIVEN MODELS

Roy T. Kamimura^{1*}, Silvio Bicchato², Hiroshi Shimizu³, Joe Alford⁴, and Greg Stephanopoulos⁵

¹*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551*

²*Department of Chemical Engineering Processes, University of Padova, Via Marzolo, 11, Padova 35131, Italy*

³*Department of Biotechnology, Faculty of Engineering, Osaka University, 2-1, Yamadaoka, Suita, Osaka 565, Japan*

⁴*Lilly Research Laboratories, Eli Lilly and Co., Indianapolis, IN 46285*

⁵*Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02319*

Keywords: data mining, neural networks, cluster analysis, SIMCA, mean hypothesis testing, multivariate time series, data-driven modeling

Abstract

A framework is presented that emphasizes the need to understand the strengths and weaknesses of the data prior to modeling. In short, given a list of constraints, the idea is to let the data sort itself along those guidelines. Once the data has been organized into some coherent faction, the user has a better understanding of what the strengths and weaknesses of the data are as the analysis proceeds. The goal is to understand the character of the data so that the user is not overwhelmed but is able to systematically organize and decompose information so as to facilitate the analysis and build an effective model. The data analyzed is that from an industrial fermentation but the framework presented is generic enough that it can be used in any application involving multivariate time series data, such as time varying microarray measurements.

Introduction

For the longest time, technology was the limiting factor in how many and the types of measurements an experimentalist was able to collect, and the data analysis focused on what data that was made available. Recent advances, such as gene chips, 2-D electrophoresis, MALDI/SELDI mass spectrometry among others, however, have reduced this bottleneck in data collection and have increased the data flowrate by orders of magnitude. Unfortunately, the resulting information overload has often overwhelmed many traditional data analysis methodologies used to dealing with either far smaller sample sizes and/or variables. In particular, the size of the newer data sets makes it difficult to perform “sanity” checks on the quality of the information coming out. There are simply either too many samples or too many variables for the scientists to consider. Approaches from the field of data mining can be used to control this tide but the data-centric focus of the analysis is more insightful than the techniques themselves.

In short, the goal of data mining is to extract knowledge from the data in the form of patterns (Fayyad, *et al.*, 1996). To achieve this, data mining typically employs a wide variety of techniques from fields such as statistics, artificial intelligence, machine learning, and pattern recognition among others. Since the emphasis is on learning what is in the data itself, data mining is more concerned with finding the patterns than employing a particular algorithm and it is this philosophy that may prove to be an invaluable asset over other data analysis methodologies. In analyzing biochemical and biotechnological systems, detailed mechanistic understanding is often limited. Thousands of measurements are collected but the relationship among them is not well defined. The end result is that researchers must make use of the measurements that are available, try to identify patterns in the existing set of measurements and attempt to correlate them to the system behavior. The drawback to such empirical methodologies is that the models are not initially based on scientific principles but derived from the data itself and so is highly dependent on the nature of the data collected. As technology makes it easier to collect large volumes of data, the likelihood that a significant fraction of it may be highly redundant or even irrelevant to the modeling objective increases. Yet, many efforts in the modeling field have often focused more on developing and comparing different data-driven models (e.g., artificial neural networks being superior to principal components analysis) rather than addressing the impact of data quality on model performance as Kell and Sonnleitner, 1995 point out. It may well be that model A outperforms model B but this may come at a price if model A is far more sensitive to spurious measurements compared to model B. The issue is to try to account for data quality as it impacts model performance

* To whom the correspondence should be addressed (royk@llnl.gov)

and to establish a sound foundation for model construction. Thus, in the spirit of Kell and Sonnleitner's paper on Good Modeling Practices (Kell and Sonnleitner, 1995), this paper presents a framework for the rational design of data-driven models and demonstrates the impact of assessing data structure on model performance. Specifically, historical records of an industrial fermentation will be used to classify the performance of new runs.

Figure 1 shows an overview of the proposed approach. The techniques listed are not meant to be an exhaustive but just a sampling of the algorithms that are available. The user can choose to use either one or several of them in combination during the course of the analysis. For those familiar with the data mining process, there are several additional steps that have not been listed but are also important such as data cleanup. Here the emphasis is more towards the model building aspect and assumes the data has already been properly transformed and cleaned.

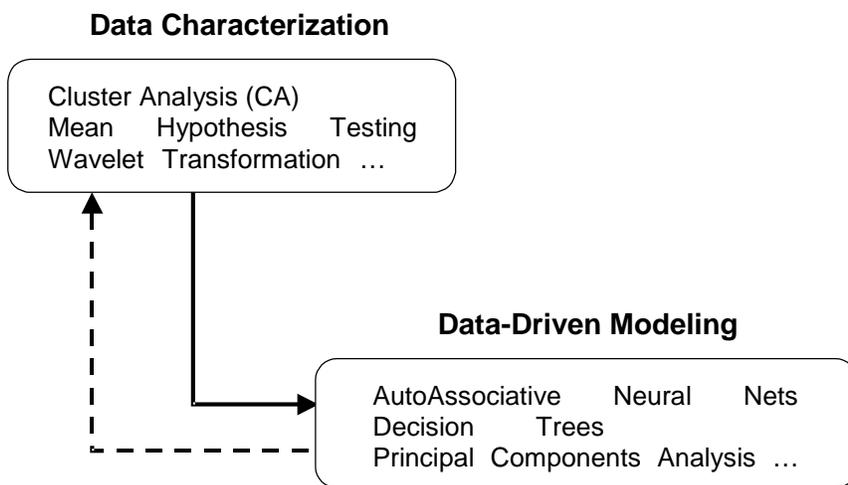


Figure 1 Overview of the framework to the rational design of data-driven models.

Methodology

The first step in this framework is characterization of the data. But prior to this, it is important that the objective of the analysis be clearly identified. The goal of data characterization is analogous to feature selection/extraction phase as its primary purpose is to reduce the data overload. Identifying the problem helps the users set criteria for what types of features/variables they consider to be relevant. In the context of this analysis, the goal is to classify samples as being either high or low quality fermentations. Hence, one criteria is to select those measurements for which the ability to discriminate between the two classes is strong. Identification of the most informative variables has relied in the past on experience or the use of variable selection techniques associated with applied statistics, such as Principal Component Analysis (PCA), regression, discriminant analysis, etc. While these techniques are effective, they suffer from the drawback that the variable selection is dependent on the quality of the other measurements. For example, in the case of multivariate regression and discriminant analysis, the presence of correlated measurements can lead to unstable model coefficients that must be removed beforehand (Dillon and Goldstein, 1984; Mardia, *et al.*, 1979). While PCA avoids this collinearity problem, it still relies on the correlational structure of all the variables for the variable selection. This aspect may be problematic if the number of non-informative (nondiscriminatory) variables is high relative to informative ones since this is equivalent to increasing the background noise of the system. Furthermore, most of these techniques do not take into account the time-dependency of the data. Variable interactions may change over time and it is not always possible to know *a priori* when these changes occur. To address some of these issues, variable selection will be done through mean hypothesis testing (MHT). Due to the lack of space, the reader is referred to Kamimura, *et al.*, 2000a for details. In brief, MHT works by identifying variables or combinations of variables that fail the null hypothesis of equivalent means. Given two populations of samples, in this case, high and low yield fermentations, this test subjects the variables to the hypothesis that their population (class) means at each point in time are equivalent. Variables that fail this test indicate that their behavior on average is statistically different in the two classes as the mean can be used as an indicator of the overall behavior of a measurement. The algorithm uses the class means and their associated covariance matrices to construct a test measure, called the Union Intersection Statistic. This value is compared to a tabulated F-value for a given significance level, normally 95%. If the test statistic exceeds the tabulated F-value, the null hypothesis is rejected

and the variable(s) under consideration are defined to be discriminating. Although similar to multivariate analysis of variance (MANOVA), the test has been adapted to handle time series data and the more general case of unequal covariance matrices is considered (Mardia, *et al.*, 1979).

Once the variables have been selected, the characterization still continues with the focus now being on the data samples themselves. It is important to consider how “well-behaved” the data are. Specifically, how homogeneous are the samples as they are labeled as belonging to a particular category. Often such assignments are made on the basis of a single measurement or an assessment made at some point in time and may not be entirely consistent. For example, a population of patients may be described as being healthy but this does not imply that all of them have very similar metabolic readings. Subclasses may exist due to factors such as race and gender. The issue of data homogeneity is often overlooked. While an expert may be able to identify an uncharacteristic sample behavior and also discern a finer classification, such decisions are often difficult to rationalize or expand to very large sets of data. In addition, when dealing with large numbers of samples containing many variables, this task can become quite daunting.

Classically, unsupervised approaches such as cluster analysis can be used to address this issue but the nature of most biotechnological (and genomic) data limit their utility - specifically, the multivariate and often temporal aspect of the data. Cluster analysis is designed to handle feature vectors where a vector contained enough information to characterize a sample. This is not plausible for bioprocess data, which is in the form of matrices with time points in the rows and measurements in the columns. In addition to the author’s knowledge, work on extending cluster analysis to time series has only considered a single variable (Shaw and King, 1992). To circumvent these obstacles, PC1 Time Series Clustering was developed (Kamimura, *et al.*, 2000b). It combines the dimensional reduction capability of principal components analysis with the grouping power of cluster analysis. The algorithm considers the multivariate nature of the data by linearly combining the most relevant variables into the first principal component, PC1. The end result is a single vector over time, which carries information on the variable interactions. It is this vector that is then subjected to a hierarchical agglomerative clustering. Samples with similar data structures are grouped into the same cluster. The number of clusters and their associated memberships reveals the extent of the heterogeneity in the data.

Once the data has been characterized using the results from mean hypothesis testing and PC1 Time Series Clustering, an autoassociative neural network (AANN) is constructed to model patterns of process behavior from historical records. Each new run is then classified by comparing the measurement profiles from the sample with those of the different class models. Model performance is determined by its ability to classify correctly lots that were not present in the training phase.

The algorithm uses a classification scheme used in conjunction with the Soft Independent Modeling of Class Analogy (SIMCA), (Wold and Sjostrom, 1977; Saner and Stephanopoulos, 1992). SIMCA works by constructing separate models for each data class and then fitting the unclassified sample to each model. If the sample resembles the members used in the training set of a class model, the resulting error in fit should be comparable to the error observed during the training phase and the sample is tentatively assigned to the class represented by that model. After all the class models examine the unclassified sample, a final classification is made by assigning the sample to the class model with the best fit AND provided all the other models reject it. If the unknown sample is accepted by more than one class, then it is considered to fall in the *both* category (if two classes are used) and in the *neither* if accepted by no group. The *both* classification denotes that the sample appears to have characteristics shared by more than one class, while the *neither* states that the sample is unlike those used in the model training sets and hence may be representative of a new class altogether. Thus, to be classified in a class, the sample must be a member of that class alone and exclusive from the others. With the use of MHT for variable selection, the likelihood of either the both or neither classification is reduced but there can always be samples that lie near the class boundaries.

Considering only two classes, the following scoring system can be used under SIMCA:

- 1 if the sample belongs to class 1 (accepted by model 1 and rejected by model 2)
- 2 if the sample belongs to class 2 (accepted by model 2 and rejected by model 1)
- 0 if belonging to both classes (accepted by models 1 and 2)
- 3 if belonging to neither class (rejected by models 1 and 2)

As mentioned previously, one of the unique features of SIMCA's classification scheme is the ability to recognize that a sample may have characteristics of several classes or none of them. There are several reasons why this is important. First, when modeling classes that are very similar to each other, this classification scheme will show many *both* classifications. This could imply that the classes, as defined, do not allow for strong discrimination or that the variables/time windows selected are not sufficiently discriminating. Second, it is important to know when the data under investigation is beyond the training set of the models. A *neither* classification simply means that

based on the training set provided, the unknown sample cannot be classified as it is sufficiently different from all the classes considered. This can be important, since a neither classification represents a situation not present in the database used in the training set and may represent a new phenomena.

Autoassociative Neural Networks

Autoassociative neural networks are a specific type of artificial neural network trained to generate an identity association in which the network outputs approximate the inputs using linear and sigmoidal transfer functions (Kramer, 1991; 1992). The architecture of these networks is such that they do not learn the identity mapping perfectly. An internal constraint in the form of a bottleneck - a layer of hidden nodes smaller in dimension than either input or output - forces the network to reproduce an n -dimensional data set at its output using only f independent variables with $f < n$. The resulting identity mapping creates a global reduction of the data dimensionality and allows the extraction of significant features by the bottleneck nodes. The concept is similar to that of principal components analysis (PCA) where, if redundancy is present, the data can be reconstructed with a dimensional set smaller than the original data.

The architecture of the autoassociative network is shown in Figure 2. The input to the network is a vector of process measurements, so the size of the input layer corresponds to the dimension of the measurement vector. Since the output layer produces a reconstructed version of the inputs it is also of the same dimension as the input. The autoassociative network shown in Figure 2 contains three hidden layers, the mapping layer which models the mapping function set, the bottleneck layer whose outputs capture the underlying correlational structure of the data, and the demapping layer which models the demapping function set. The nodes of the mapping and demapping layers must consist of nonlinear transfer function to handle any arbitrary mapping and demapping function sets selected. Nonlinear nodes, however, are not required in the bottleneck and output layers.

The training process is dynamic; the weights among the different layers of the network are allowed to evolve until the correct mapping is obtained. Network weights are adjusted by the backpropagation algorithm so that the reconstructed measurements vector at the output layer matches the input as closely as possible, in a least-squares sense, over the set of the training examples. The final internal representation developed by the training procedure retains the maximum amount of information from the original data set for a given degree of dimensional compression represented by the number of nodes in the bottleneck layer.

Mathematically, AANN can be viewed as a nonlinear generalized version of PCA. The input, mapping, and bottleneck layers together represent a nonlinear function \mathbf{G} which projects the inputs to a lower dimension space:

$$T_k = G_k(\mathbf{X}) \quad k = 1, \dots, f \quad (1)$$

where T_k is the output of the k th bottleneck node and \mathbf{X} is the input vector. The bottleneck layer, the demapping layer, and the output layer represent a second function \mathbf{H} that reproduces an approximation of the inputs from the features at the output of the bottleneck layer:

$$\tilde{\mathbf{X}}_i = H_i(\mathbf{T}) \quad i = 1, \dots, n \quad (2)$$

In summation, the data is first compressed to lower dimensionality and then reconstructed. The loss of information involved in this two-stage process is measured by the sum of the squares of the differences between the inputs and the reconstructed outputs, summed over the training set, also referred to as the sum of squared errors (SSE):

$$SSE = \sum_{p=1}^n \sum_{i=1}^m (X_i - \tilde{X}_i)_p^2 \quad (3)$$

where m is the number of the training examples used to train the network. Minimizing the SSE during network training results in maximum signal reconstruction and in minimum information loss.

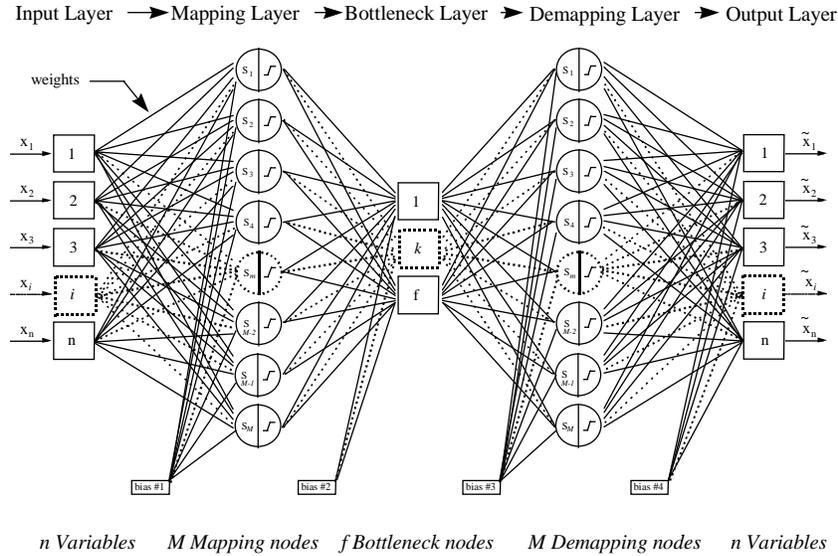


Figure 2 Architecture of AANN.

Using the SIMCA procedure, a separate network is trained for each of the two classes. The procedure for classification is as follows:

- 1) Each sample, representing a process lot with a specified time window and variables, is fitted to both models, 1 and 2.
- 2) If the error generated by the sample lies within the standard reconstruction error (the mean error observed during the training phase ± 2 standard deviations), it is tentatively assigned to that model's class. Only if the other model rejects the sample, is it finally assigned the classification. So, a class 1 rank means acceptance by eq. (4) and rejection by eq. (5), likewise for class 2:

$$SSE_{tr1} - 2 * std_1 < SSE < SSE_{tr1} + 2 * std_1 \quad (4)$$

$$SSE_{tr2} - 2 * std_2 < SSE < SSE_{tr2} + 2 * std_2 \quad (5)$$

where SSE is the sum of squared errors generated by fitting a sample to a model, SSE_{tr_i} is the mean SSE observed for the training set of model i , and std_i is the standard deviation of SSE_{tr_i} observed during the training phase of model i .

- 3) If it is observed that the sample fits both models, it is given a score of 0 - both eq. (4) and (5) are satisfied.
- 4) If it is observed that the lot does not fit either class, it is given a score of 3 - both conditions outlined in eq. (4) and (5) are rejected.

Results

The goal of this analysis was to use AANN's to classify previously unseen samples. The fermentation data analyzed is described in Table 1. On-line performance was simulated by testing the algorithm on a cross-validation data set, separate from the model training set, with the goal to determine, as early as possible, the process outcome from the measurements accumulated at a particular time point. The following aspects were investigated with respect to their effect on classification success: network design, effect of variable selection, effect of time window, and training set design. In the discussion below, lots whose numbers are between 1 and 22 (inclusive) are defined as high by the industrial source, while lots with numbers 23 to 45 (inclusive) are defined as low. It is important to note that these are the *official* classifications assigned by the industrial source prior to the analysis by the algorithms presented here. The performance of the network is based on its ability to match the *official* classification.

Table I - Conditions for industrial case study 1.

Type	industrial fermentation
Mode	fed-batch
Run length	82 time points
Number of variables	17 (only variables labeled 4-17 are used)
Number of classes	2
Lot/Class	22 for high class, 23 for low class

Network design

The network topology used in the present work is autoassociative, with two layers of non-linear hidden nodes. Both the input and output layers contain equal number of nodes to the number of variables. Bottleneck layers contained 3 nodes when considering combinations of 4 variables and 6 nodes when using all the 14 variables. The networks were trained using the conjugate gradient version of backpropagation, as presented by Leonard and Kramer (1990). In all cases, a 90% reconstruction of all variables at the output layer was chosen as convergence criterion. For simplicity, the dimensions of the mapping and demapping layers were assumed to be the same, while the number of nodes in these layers was optimized according to Kramer (1991). This yielded 8 nodes for the mapping and demapping layers when considering combinations of 4 variables and 24 nodes when using all the 14 variables. Also, sigmoidal nodes were used in the mapping and demapping layers while a linear function was applied in all other layers. Before the training phase, all the data were auto-scaled so that input and output are bounded in the range (-1,1). The networks were designed, trained, and tested using modified versions of Matlab Neural Networks Toolbox algorithms.

Effect of variable selection

Three different variable sets were examined under the same time periods, Table II. Using all 14 process measurements as a base case, the AANN system was at best able to classify only 66% of the samples correctly. For comparison, using all the variables for the entire time course, yielded a correct classification rate of 67%. In contrast, when only the most discriminating variables, uncovered by mean hypothesis testing (variables 4, 5, 9, and 11) are used, the classification rate improves to 80%. This increase suggests that the network, in attempting to reconstruct all data, sacrifices classification performance, representing an inefficient use of the data for the modeling objectives. It is interesting to also note that the neither classification (25%) is larger when all the variables are used compared to the number of neither cases (7%) when only the discriminating variable subset is considered. In the last row of Table II, the classification accuracy drops to 27% when only a nondiscriminating set of variables, labeled as 13, 14, 16, and 17, is used.

Effect of time windows

In a dynamic process, different process measurements may display varying degrees of class discriminating power at different windows of time. To study the influence of time, the process was divided into 4 time windows of 20 points each, 1-20, 20-40, 40-60, and 60-80. Mean hypothesis testing (Kamimura, *et al.*, 1999a) of the data determined that the time windows 40-60 and 60-80 are discriminating, while the 1-20 window represents a non-discriminating zone. Table II summarizes the classification results showing the influencing of time windows. It is interesting to note that most of the data classifications in the early time windows (1-20 and 20-40) are in the both category regardless of the variable subset considered. This suggests that the data from high and low classes behave so similarly to each other that they are practically indistinguishable. This observation has interesting ramifications as it may indicate that the measurement readings are not sensitive enough to allow discrimination. As the time window proceeds along the course of the fermentation (40-60, 60-80), the information becomes more discriminating and the algorithm's ability to correctly predict the process outcome increases. It should be emphasized that this improvement over time will not always occur but is data dependent, see window 60-82 for the all variable and discriminating variable subset and note how performance dropped. In Kamimura, *et al.*, 2000a, it was observed for some processes the time window for discrimination is of finite duration and disappeared during the course of the run. These observations point to the need to recognize that variable interactions may not be time-invariant and this should be considered in any effort to increase model effectiveness.

Table II - AANN classification results - Time window/Variable effect.

Variables	Time Window	Correct class	Wrong class	Neither class	Both classes
[4-17]	[1-20]	6 (13%)	4 (9%)	7 (16%)	28 (62%)
	[20-40]	13 (29%)	1 (2%)	11 (25%)	20 (44%)
	[40-60]	30 (66%)	3 (7%)	11 (25%)	1 (2%)
	[60-82]	27 (60%)	1 (2%)	17 (38%)	0 (0%)
[4 5 9 11]	[1-20]	4 (9%)	3 (7%)	3 (7%)	35 (77%)
	[20-40]	17 (38%)	6 (13%)	2 (4%)	20 (44%)
	[40-60]	36 (80%)	4 (9%)	3 (7%)	2 (4%)
	[60-82]	34 (76%)	3 (7%)	6 (13%)	2 (4%)
[13 14 16 17]	[1-20]	3 (7%)	2 (4%)	8 (18%)	32 (71%)
	[20-40]	16 (36%)	5 (11%)	5 (11%)	19 (42%)
	[40-60]	12 (27%)	1 (2%)	9 (20%)	23 (51%)
	[60-82]	12 (27%)	4 (9%)	6 (13%)	23 (51%)

Effect of training set design

As mentioned previously, the official class assignment was designated by the industrial source. The purpose of this section is to illustrate the complications that can arise when the data are not as well-behaved as one might have assumed. Using the memberships of clusters generated by PC1 Time Series Clustering (Kamimura, *et al.*, 2000b), two types of training sets, a homogeneous and a heterogeneous one, were generated. The homogeneous data sets were designed to include lots from clusters whose membership consists predominantly, if not all, of members of the same class. The heterogeneous training sets, by contrast, attempt to capture as much of the variability present in the data as possible, thus emphasizing a representative cross-section of the data from the major clusters. For the high class, this required taking members from group named 2 and group named 1 - being the two clusters where the high class lots form a significant fraction of the membership. For the low class, the selected members come from groups named 3 and 5. In Figure 3 are shown the time profiles of discriminating variables for some members of these groups (Figure 3a, high class representative lots, Figure 3b, low class typical lots, Figure 3c, lots from mixed clusters).

Table III summarizes the effect of different training sets on model performance focusing only on the 40-60 time window and using variables [4 5 9 11]. The impact of using a homogeneous or a heterogeneous design is not as strong as the variable or time window selection but is noticeable as explained by the 80% correct classification for the heterogeneous versus the 64% for the homogeneous training set. Not surprisingly, the homogeneous training set produces a larger *neither* classification rate than the heterogeneous. This can be attributed to the fact that with the homogeneous training set the net is forced to be more sensitive to small differences present in high or low lots.

Table III - AANN classification results - Training set design effect.

Data Set	Correct class	Wrong class	Neither class	Both classes
Homogeneous	29 (64%)	4 (9%)	12 (27%)	0 (0%)
Heterogeneous	36 (80%)	4 (9%)	3 (7%)	2 (4%)

Conclusions

As demonstrated with the AANN, time windows and discriminating variables have a major impact on modeling performance. Selecting the wrong measurements or focusing on nondiscriminating time windows can lead to erroneous conclusions about a model's capabilities. It is also important to recognize when to sample the data as this case shows that some variables change character over time. This observation in particular may be of relevance in the analysis of microarray data as the interactions among genes may vary over time. Increasing the sampling frequency during periods of interest will provide more insight into the system behavior.

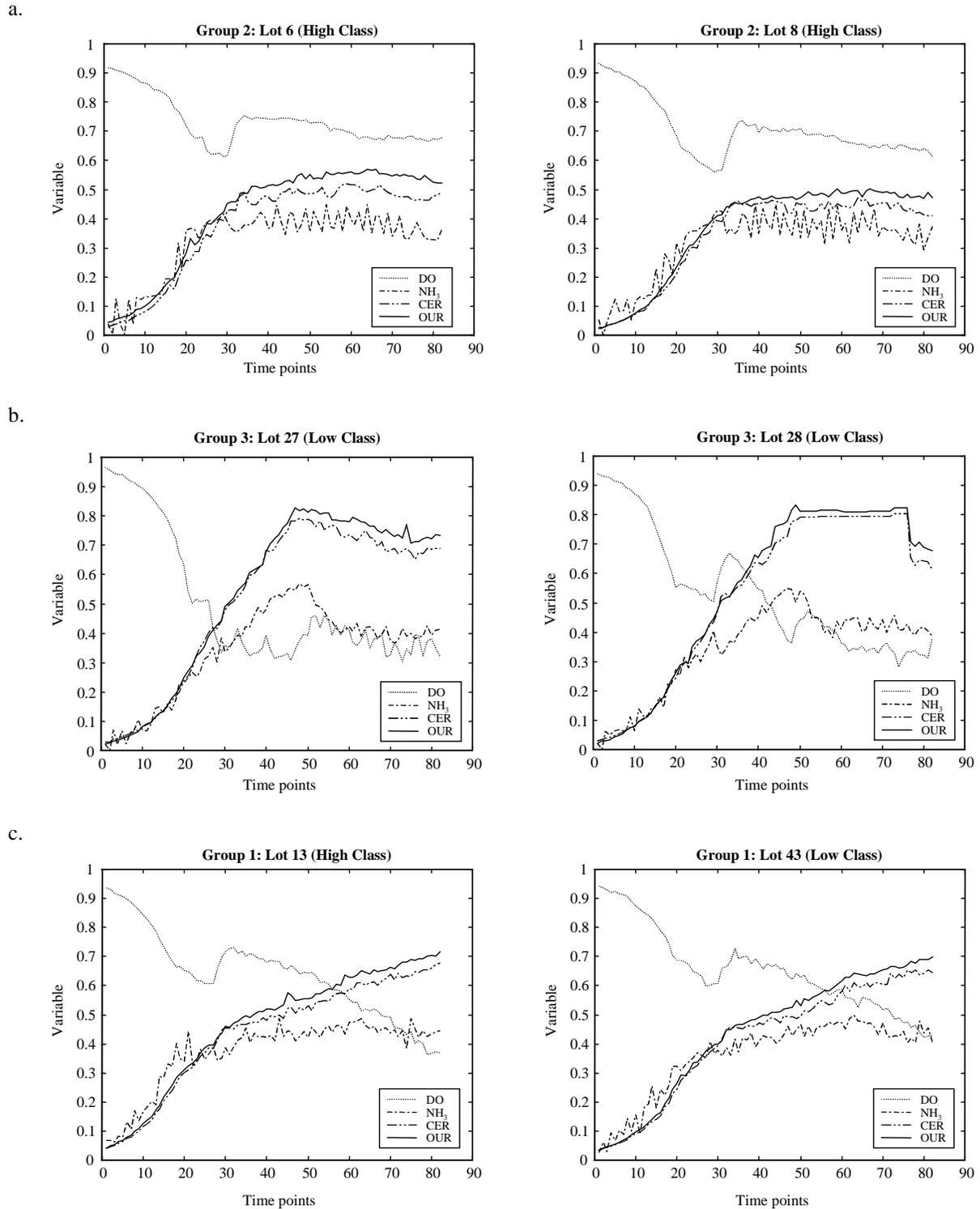


Figure 3. In all the figures only the four most discriminating variables are shown. a) Two representatives of the high class fermentation; b) two from the low; c) one from each class but belonging to a different cluster grouping. Note how both lot 13 and lot 43 resemble each other as one might expect from the clustering but both have attributes that are similar as well as different from their high and low brethren.

It is also important to recognize the type of variability that exists in the data. While methods exist to identify *outliers*, it is far more difficult to identify subclasses in the data. As with the fermentation data, it may well be that the groups the user has defined are not as homogeneous as were originally assumed. This knowledge can be used to decide if the model should be conservative (homogeneous) or robust (heterogeneous) to handle the variation observed in the database. In the former, the emphasis is on increased confidence in the model accuracy whereas the latter prefers a more robust classifier at the expense of higher misclassification. By using techniques such as mean hypothesis testing and PC1 Time Series Clustering with data-driven models such as AANN, the limit and potential value of the data that is available for analysis can be better understood. In the context of modeling, characterization of the data provides a firm foundation upon which the model can be built. These techniques have been applied with success to other systems such as vaccine manufacturing, and genomic data. With the former, it was possible to identify where in the manufacturing process was responsible for variations in product quality. In regards to the latter, the analysis identified clusters of genes from 2 different metabolic states (anaerobic and aerobic) and reduced the number of open reading frames (ORF's) that had to be considered from 6153 to 737 split over 4 distinct clusters. In particular the latter groupings were found to be highly correlated in terms of their metabolic function.

Acknowledgments

This work has been supported in part by NIH Genome Training Grant. We also acknowledge the support of the industrial participants of the MIT Consortium for Fermentation Diagnosis and Control who also provided the process data analyzed in this work.

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

References

- Davis, J., Bakshi, B. 1996. Process monitoring, Data Analysis, and Data Interpretation, pp. 1-11. In: J.F. Davis, G. Stephanopoulos, and V. Venkatasubramanian (eds.), First International Conference on Intelligent Systems in Process Engineering by AIChE Symposium Series 312, vol. 92.
- Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. 1996 *Advances in Knowledge Discovery and Data Mining*, MIT Press, Cambridge.
- Guthke, R., Ludwig, B. 1994. Generation of Rules for Expert Systems by Statistical Method of Fermentation Data Analysis. *Acta Biotechnol.* **14**, 13-26.
- Kamimura, R.T. 1997. Application of Multivariate Statistics to Fermentation Database Mining. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.
- Kamimura, R.T., Bicciato, S., Shimizu, H., Alford, J., and Stephanopoulos, G., 2000a. Mining of Biological Data I: Identifying Discriminating Features via Mean Hypothesis Testing, *Metabolic Engineering*, **2**(3), 218-227.
- Kamimura, R.T., Bicciato, S., Shimizu, H., Alford, J., and Stephanopoulos, G. 2000b. Mining of Biological Data II: Assessing Data Structure and Class Homogeneity by Cluster Analysis, *Metabolic Engineering*, **2**(3), 228-238.
- Kell, D.B., Sonnleitner, B. 1995. GMP - Good Modelling Practice: an essential component of Good Manufacturing Practice. *TIBTECH.* **13**, 481-492.
- Kramer, M.A. 1991. Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. *AIChE J.* **37**(2), 233-243.
- Kramer, M.A. 1992 Autoassociative Neural Networks. *Computers Chem. Engng.* **16**(4), 313-328.
- Leonard, J., Kramer, M.A. 1990. Improvement of the Backpropagation Algorithm for Training Neural Networks. *Computers. Chem. Engng.* **14**, 337.
- Mendenhall, W. and Sincich, T. 1992. *Statistics for Engineering and the Sciences*. 3rd edition. Dellen Publishing, San Francisco, CA.

Saner, U. and Stephanopoulos, G. 1992. Application of pattern recognition techniques to fermentation data analysis, pp. 123-128. In: Karim, M. N. and Stephanopoulos, G. (eds.), *Modeling and Control of Biotechnical Processes* by IFAC, vol. 10.

Shaw, C.T., King, G.P. 1992. Using cluster analysis to classify time series, *Physica D*. **58**, 288-298.

Wold, S., Sjostrom, M. 1997. SIMCA: a method for analyzing chemical data in terms of similarity and analogy. In: Kowalski, B. R. (ed.), *Chemometrics: Theory and Application* by ACS Symposium Series, 52.