Review

Mining bioprocess data: opportunities and challenges

Salim Charaniya[1], **Wei-Shou Hu**[1] **and George Karypis**[2]

[1] Department of Chemical Engineering and Materials Science, University of Minnesota, 421 Washington Avenue SE, Minneapolis, MN 55455-0132, USA
[2] Department of Computer Science and Engineering, University of Minnesota, 421 Washington Avenue SE, Minneapolis, MN 55455-0132, USA

Modern biotechnology production plants are equipped with sophisticated control, data logging and archiving systems. These data hold a wealth of information that might shed light on the cause of process outcome fluctuations, whether the outcome of concern is productivity or product quality. These data might also provide clues on means to further improve process outcome. Data-driven knowledge discovery approaches can potentially unveil hidden information, predict process outcome, and provide insights on implementing robust processes. Here we describe the steps involved in process data mining with an emphasis on recent advances in data mining methods pertinent to the unique characteristics of biological process data.

Introduction

In the past two decades we have witnessed a major transformation of bioprocess manufacturing. Protein-based therapeutics have overtaken natural product-based drugs as the major biologics. A majority of the protein therapeutics are produced using recombinant mammalian cells. The value of these biopharmaceuticals, most of them recombinant antibodies, exceeds US $33 billion per annum [1]. They are manufactured in modern production plants equipped with systems for automated control as well as comprehensive data collection and archiving. These archives represent an enormous opportunity for data mining in that they might unearth a wealth of information for enhancing the robustness and efficiency of manufacturing processes. However, despite the stringent process control strategies employed, variations in the final process outcome are commonly observed. With each production run valued at millions of dollars and every manufacturing plant costing hundred million dollars and upwards, there is a great potential for cost saving through mining process databases to uncover the distinguishing characteristics of a good process.

In the following we discuss the challenges associated with investigating bioprocess data and the techniques that have been previously proposed to mine process data. We describe a scheme to systematically analyze a complex bioprocess dataset, and also highlight the recent advances in data mining, which are applicable for analyzing bioprocess data.

Characteristics of bioprocess data

Any modern bioprocess plant maintains electronic records of material input (quantity, quality control records, lot number), process output (cell density, product concentration and quality, etc.) control actions (base addition, CO_2, O_2 flow rate, etc.) as well as physical parameters (agitation rates, temperature, etc.), from the frozen cell vial to the production scale bioreactors. Based on the frequency of measurements, bioprocess parameters can be categorized into different types. A vast majority of the process data are acquired on-line.

Glossary

Terminology

Class: process runs can be categorized into discrete classes (e.g. *high*, *medium*, and *low*) based on product titer, product quality, or other measures of process outcome.

Feature: a representation of the entire temporal profile, either in its entirety or abbreviated, or a small time window of a process parameter (see Figure 2 for an example) that has been treated if necessary, and is suitable for data mining.

Generalization error: the error incurred by a model in predicting the outcome of a new instance (e.g. a future process run).

Model: a set of functions that describe the relationships between the process features and the process outcome (or any other characteristic of interest).

Overfitting: a phenomenon that results when a model performs well on the training set, but has poor ability to predict the outcome of new instances.

Training and test set: training set comprises the process data from a set of runs with known outcomes, which are used to construct a model. The model is assessed by a test set, which is a set of runs (with known outcomes) that were not used for model construction.

Data pre-processing methods

Adaptive piecewise constant approximation (APCA) [12] : APCA segments a profile into unequally spaced intervals. Within each interval, the profile is abbreviated as a single value. The intervals are chosen to minimize the error due to data compression.

Discrete Fourier transform (DFT): DFT uses a linear combination of sinusoidal waves of different frequencies to represent a profile. Depending on the granularity desired, the series can be truncated after a few waves. A fast Fourier transform can also be used for efficient computation.

Discrete wavelet transform (DWT) *[48]* : DWT represents a profile as a combination of basis functions, called scaling and detail functions. Using the basis functions, the profile can be convoluted to approximate coefficients and detail coefficients. Dimensionality reduction is achieved by pruning the detail coefficients. DWT has been previously employed for representation of temporal bioprocess data [6].

Piecewise linear approximation (PLA) *[49]* : PLA compresses a complex profile into a series of linear functions. The profile is divided into short, equal-length segments and each segment is characterized by a left and/or right height and slope of linear function. PLA has been previously applied for compression of chemical process data [49].

Symbolic aggregate approximation (SAX) *[50]* : SAX is a symbolic representation of a profile. The profile is divided into equally spaced intervals and each interval is approximated by the mean value of the profile in that interval. The mean approximations for the intervals are thereafter discretized into a small number of symbols. The symbols are derived (based on the profile) such that they are equally probable.

Corresponding author: Hu, W.-S. (acre@cems.umn.edu).

0167-7799/$ – see front matter © 2008 Elsevier Ltd. All rights reserved. doi:10.1016/j.tibtech.2008.09.003 Available online xxxxxx

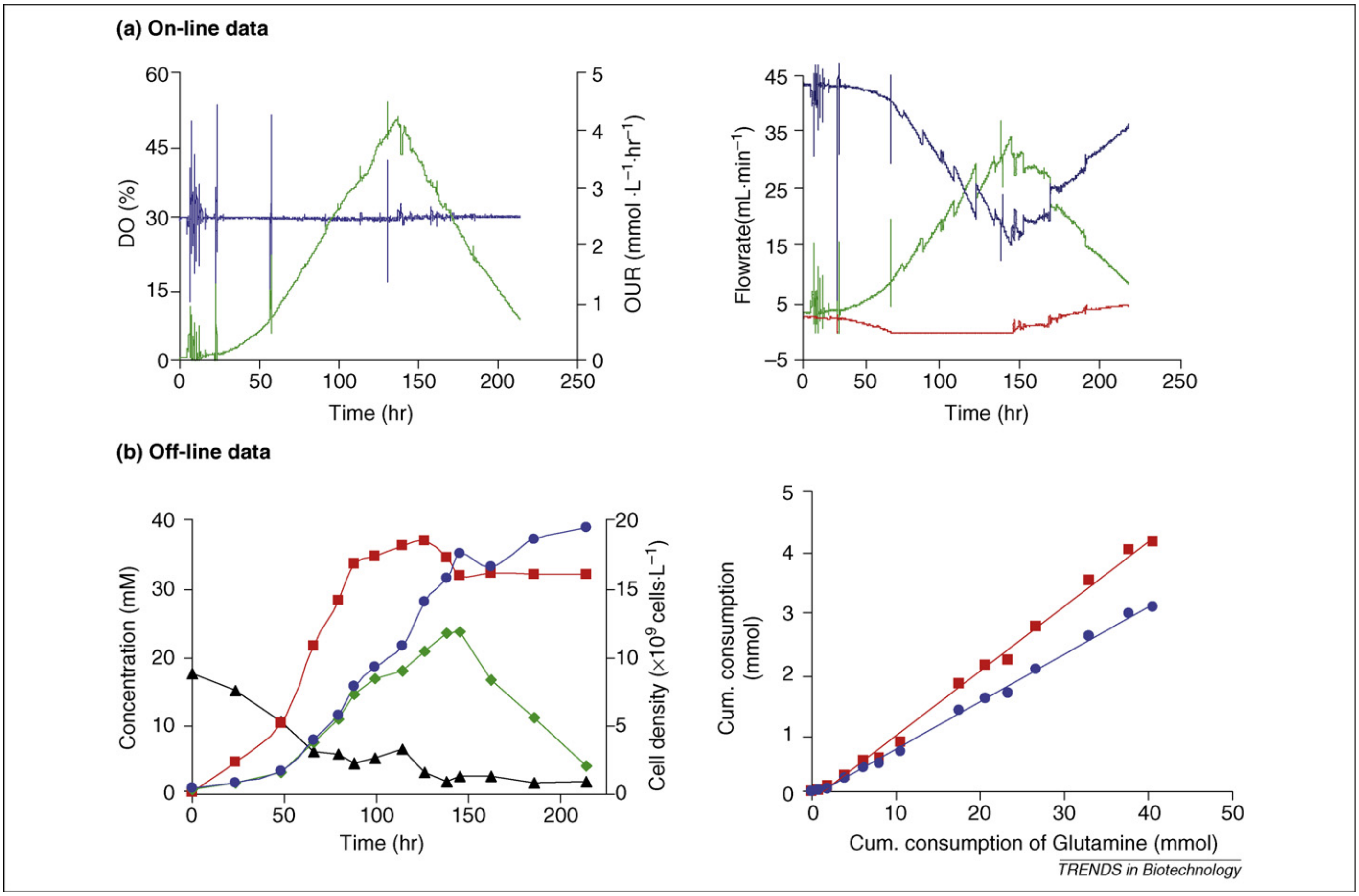

Figure 1. Example of bioprocess data. **(a)** Representative online data are shown, which are recorded every few minutes during the entire culture duration. The left panel shows the profile of typical reactor state parameters, such as percent air saturation of dissolved oxygen (shown in blue) and the oxygen uptake rate (green). The right panel shows the profile of gas flow rates as common control action parameters. Observed curves are for nitrogen (blue), oxygen (green) and carbon dioxide (red). **(b)** Typical off-line data for a process are shown. The left panel illustrates the raw data containing biochemical parameter profiles for the total cell density (blue), viable cell density (green), glucose (black), and lactic acid (red). The right panel shows the profile for parameters that have been derived from the raw data and that are physiologically relevant, such as the cumulative consumption or production of nutrients and metabolites. Shown here are the consumption of threonine (red) and phenylalanine (blue) with respect to the consumption of a key nutrient glutamine. The slope of the linear regression provides the stoichiometric ratio of threonine and phenylalanine with respect to glutamine.

However, a few key parameters, such as viable cell density and concentrations of product and some metabolite and nutrients, are measured off-line (Figure 1). While the off-line parameters are measured periodically, many on-line parameters are measured continuously with respect to the time scale of the production cycle. Additionally, the information about some process parameters may be available at a single time point only. For example, product concentration and quality index might be measured at the final time point, before or after product recovery. Bioprocess data are thus heterogeneous with respect to time scale. Process data are also heterogeneous in terms of data types. Some parameters are continuous, such as cell and product concentrations, pH, whereas others are discrete or even binary, such as the valve settings for nutrient feeding and gas sparging, which can only be in the ON or OFF state. Even quality-related parameters for either raw material or product can be discrete. For example, the glycosylation profile as a measure for the quality of a glycoprotein is often evaluated by the discrete distribution of different glycans. Due to these heterogeneities in time scales and data types, bioprocess data are significantly different from the data arising in other application areas in which data mining methods have been used (e.g. retail records). These heterogeneities should be taken into consideration when data mining methods are devised.

Knowledge discovery and bioprocesses

The aim of mining bioprocess data is to uncover knowledge hidden within the enormous amounts of data associated with different process runs that can be used to improve and enhance the robustness and efficiency of production processes. This is achieved by analyzing different types of process runs to identify novel and useful relations and patterns that associate various aspects of the production process with different measures of process outcome, such as product titer and product quality. These process outcome measures are often used to categorize process runs into different *classes*. For example, if product titer is the outcome of interest, the different runs can be classified as 'high' or 'low' producing runs. Similarly, process runs can be grouped as 'good' or 'bad' using product quality as the metric of process outcome. The notion of gaining knowledge by scrutinizing large volumes of data has been applied to a wide array of problems ranging from image

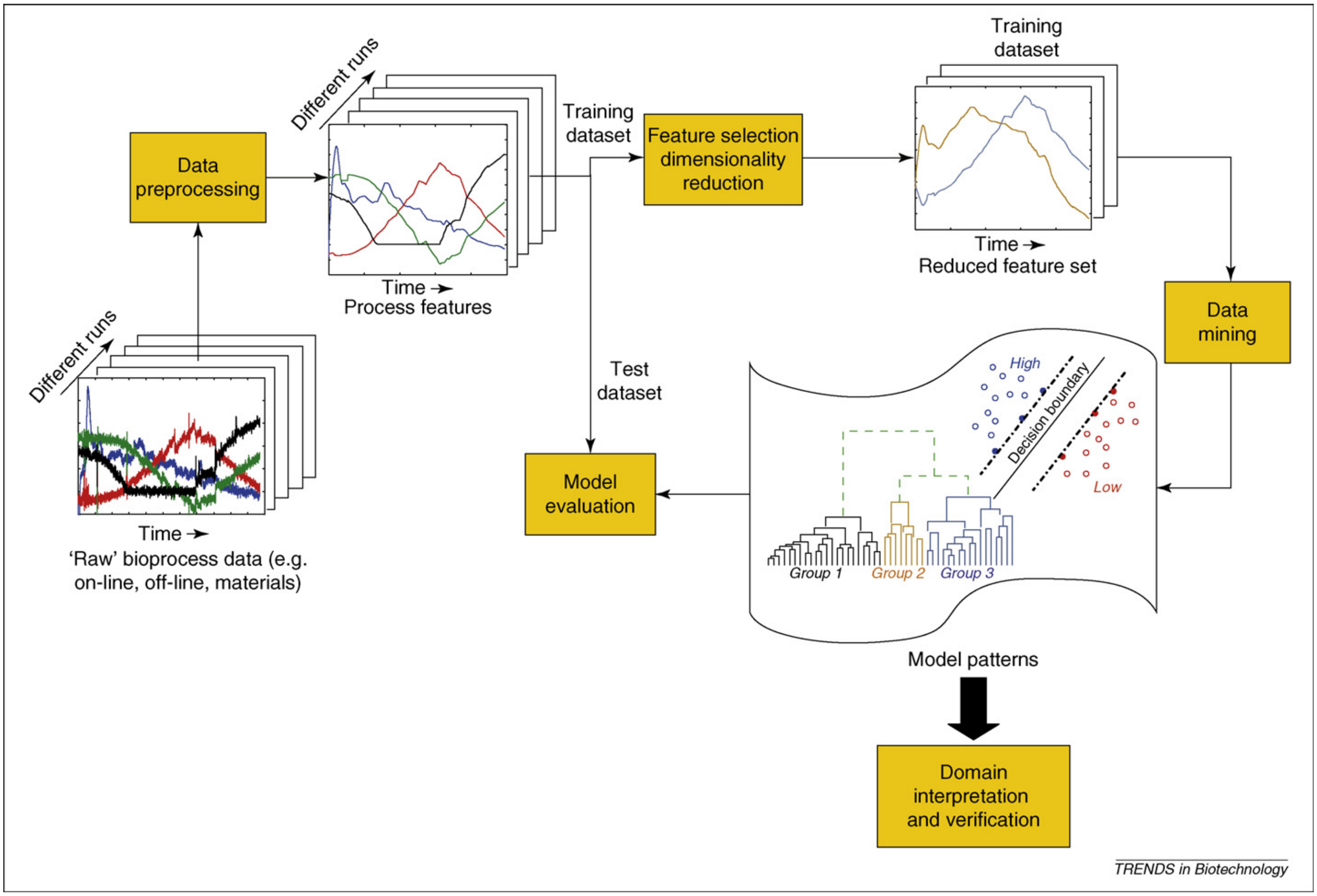

Figure 2. An approach for data-driven knowledge discovery in bioprocess databases. Process data includes off-line and on-line parameters, as well as raw material logs. Representative raw profiles from four temporal process parameters of a single run are shown. Process data from several runs are preprocessed to extract compact and smoothened features that depict the underlying process signals. The entire dataset is then split into a *training* subset, which is used for model construction, and a *test* subset, which is used for model assessment. Feature selection or dimensionality reduction is implemented on the training dataset. For example, principal component analysis (Box 1) can be used to identify two dominant patterns in the dataset shown here and thereby reducing the number of the initial features by half. Data mining methods are applied on the reduced feature set with the aim to discover model patterns, which are subsequently evaluated on the test dataset. The training and evaluation procedure can be repeated multiple times for further refinement of the model. Thereafter, the model patterns can be interpreted and verified by process experts, and the gained knowledge can be used for process enhancement.

classification in astronomy to identifying fraudulent activities in financial transactions [2].

A typical knowledge discovery process entails several iterative steps (Figure 2). These steps include: data preprocessing, feature selection and/or dimensionality reduction, data mining, and expert analysis for interpretation of the results. The data acquired in a bioprocess typically include some parameters that are not readily amenable for analysis. The data preprocessing step transforms these data into a form (called *feature*) that is suitable for the subsequent steps. This usually involves various steps including data cleaning, normalization, transformation, denoising, and missing value imputation. In the subsequent step of feature selection or dimensionality reduction, the obtained features are analyzed to obtain the set of features that is best suited for data mining. This often involves the selection of those features that correlate most with process outcome, and the combination of highly correlated features. The data mining step applies various computational methods, such as pattern recognition and machine learning to discover any significant trends within the data. These trends are useful for describing any correlations between process parameters and for developing *models* to predict the process outcome. Finally, during the expert evaluation step, the validity of the produced results is assessed by those knowledgeable of the process (domain experts) to discern the effect of the discovered correlations on cellular physiology and process outcome.

Data preprocessing

Modern production plants are electronically supervised and create process records that are well-characterized and less prone to human errors, which significantly reduce some of the preprocessing requirements that are often associated with data cleaning and missing values imputation. However, the temporal nature of the data obtained from fermentation, cell culture, and downstream processes creates some unique challenges that need to be addressed with data preprocessing methods.

In particular, on-line parameters are often recorded every few minutes for the entire culture period that can last from a couple of days to two weeks. The culture period may even extend to a few months for some continuous or perfusion-based processes. The resulting long time series need to be preprocessed to extract the features that compactly and smoothly represent the underlying process signals. In addition, preprocessing is also important to eliminate the noise that may be present in process measurements due to instrument limitations and sampling artifacts. The work of Cheung *et al.* [3,4] and Bakshi *et al.* [5,6] laid the framework for extracting useful information from temporal process parameters. Cheung *et al.* proposed a triangular representation method in which a parameter profile was segmented into different time intervals. Within each interval, the first and second order derivatives of the profile were used to represent an increasing or decreasing trend. Bakshi *et al.,* by contrast, proposed the use of wavelet decomposition to deduce temporal features. Besides these two approaches, several other approaches can be used, such as *discrete Fourier transform*, methods for piecewise approximation (such as *piecewise linear approximation*, *adaptive piecewise constant approximation*), and *symbolic aggregate approximation (SAX)*. Among these, SAX leads to a string-based representation of a parameter profile. This representation is directly amenable to several string manipulations and data mining methods that have been developed for the analysis of protein and DNA sequences, including methods for protein structure predictions [7] and discovery of *cis*-regulatory elements [8].

In addition, due to the occurrence of a lag phase or due to variations in the growth rate, the time series obtained from different runs may not be temporally aligned. As a result, identical time points might not represent similar process states. Ignoring such time scale differences and directly comparing identical time points across different runs, for example by mean hypothesis testing methods [9,10], can lead to incorrect results. This problem can be addressed by aligning the time series of different runs during the preprocessing step. A dynamic time warping strategy, originally developed for speech recognition [11], can be used to align the time profiles, or their approximate representations [12,13].

Feature selection – dimensionality reduction

The feature selection step is used to identify features which are significantly correlated to the process outcome. A large number of feature selection approaches have been developed that can be categorized into filter and wrapper approaches (Box 1). These methods are useful for constructing models to predict the process outcome (discussed in the following section). For example, Huang *et al.* [9] and Kamimura *et al.* [10] used filter approaches that were based on hypothesis testing to select relevant features. Other studies have employed wrapper approaches based on decision trees to identify the key parameters that differentiate process runs into high and low productivity classes [5,14–16]. These studies identified specific time points, or time windows, during which one or more features could discriminate between runs in different outcome classes.

Due to the temporal nature of process data, feature selection methods must take into account the sequence of events. To this end, statistical methods can be used to assess the significance of a feature, i.e. to assess its ability to distinguish the process runs from different classes. In bioinformatics applications, several hypothesis testing methods have been proposed with the aim of identifying genes that are temporally differentially expressed between two or more phenotypes [17–19]. Such methods can also be used to evaluate the relative importance of temporal process features in discriminating runs from different groups.

The temporal profiles of some features within individual runs may be correlated. For example, oxygen uptake rate and cell density are often correlated, at least in the exponential growth stage of the culture. Hence, such features provide information that is often redundant. Dimensionality reduction techniques are commonly used to obtain a set of features independent from each other using methods such as principle component analysis (PCA) or non-negative matrix factorization (NMF) (Box 1). For example, Kamimura *et al.* [20] used a PCA-based approach to approximate multiple time-dependent process features of each run as a single temporal pattern, the so-called first principal component (PC1). This reduced feature was subsequently used to cluster process runs into different groups, which corroborated with their known classes.

Data mining

Data mining approaches can be broadly categorized as either descriptive or predictive. Descriptive approaches aim to discover patterns that characterize the data, whereas predictive approaches aim to construct models (e.g. functions) to predict the outcome of a future run by learning from the observed parameters.

Box 1. Feature selection and dimensionality reduction

Feature selection [51]

Filter methods

Filter methods select relevant features independently of the data mining step. For example, features that discriminate process runs from two or more outcome-derived classes can be identified using hypothesis testing methods, such as a *t*-test (e.g. selection of genes for expression-based tumor classification [52]).

Wrapper methods

Wrappers are iterative approaches, where feature selection relies on the results of the subsequent data mining step. Thus, for example, a subset of features is selected and its suitability is evaluated from the error rate of the predictive classifier learned from that subset. Approaches in which features are progressively added (forward selection) or removed (backward elimination) can be applied for the selection of an optimal feature subset. However, these approaches are computationally expensive and potentially suboptimal for large datasets. Alternatively, change in an objective function upon addition or removal of a feature can also be used as a feature selection strategy.

Dimensionality reduction

Multivariate temporal features of each process run can be represented as a two-dimensional matrix comprising *m* parameters sampled at *n* time intervals. Principal component analysis (PCA) [53] determines the linear correlation structure of this process data matrix as a set of patterns, called principal components (PCs). The first few PCs, which highlight the most dominant correlation patterns among the process parameters, are typically used for dimensionality reduction. The profile of any temporal parameter can be regenerated as a weighted, linear combination of the PCs. Non-negative matrix factorization (NMF) [54] is another dimensionality reduction method used to identify linear correlations between process parameters.

Descriptive approaches

The descriptive approaches fall into two categories: identifying interesting patterns in the data and clustering the data into meaningful groups.

Algorithms for finding patterns in very large datasets have been one of the key success stories of data mining research. These methods aim to analyze the features of various runs to identify a pattern that is observed in a large number of runs. A pattern can correspond to specific values of a subset of features or a specific temporal profile of a particular feature (Box 2). Any pattern must occur frequently across different process runs to be considered statistically significant and interesting [21,22]. Patterns discovered from process data can provide insights into the relationship between different features, and can also be used to discover association rules. For example, specific (on a per cell basis) glucose consumption and lactate production rates of Chinese hamster ovary cells may vary under different growth conditions. However, a switch from lactate production to lactate consumption occurs only within a small window of low specific glucose consumption rate (feature 1) and low specific growth rate (feature 2). Analyzing process data from a large number of runs can reveal the values of the specific rates at which this metabolic change is likely to occur.

Box 2. Descriptive data mining methods

Pattern discovery

Various algorithms have been developed that can mine process data to discover patterns (i.e. relations) among the features of the different runs that satisfy certain constraints (properties). The type of constraints can correspond to a minimum number of runs in which a pattern is observable (minimum frequency) and/or the minimum number of features the pattern should contain (minimum length) [22]. These constraints are used to steer the data mining algorithms towards finding interesting patterns. The most efficient approaches for finding these patterns (e.g. FPgrowth [55], LPminer [22]) do so by extending them incrementally (as long as they satisfy the specified constraints) and simultaneously eliminating the portions of the dataset that do not contain the pattern under consideration.

Clustering [56]

Clustering methods can be differentiated along multiple dimensions, one of them being the top-down (partitional) or bottom-up (agglomerative) nature of the algorithm. Partitional methods initiate with all process runs (or object/record) belonging to one cluster and they are divided into designated number of clusters. *K*-means, partitioning around medoids (PAM), self-organizing maps (SOM), and graph-based clustering methods are popular examples of partitional algorithms. By contrast, agglomerative methods start with each run belonging to a separate cluster and the clusters are merged, based on the similarities of their feature profiles, until the runs have been grouped into a pre-specified number of clusters. Hierarchical agglomerative clustering is the most commonly used agglomerative method.

The task of identifying the 'natural' clusters in a dataset is nontrivial and hence the choice of a suitable clustering algorithm is not universal. The clustering algorithm should accommodate the similarity metric that is appropriate for comparing process data from different runs. Additionally, parameters such as the optimization function for partitional methods or the linkage function for merging two clusters in agglomerative methods should be carefully chosen. Most statistical packages, such as S-Plus (commercial) (http://www.insightful.com/), and R (open source) (http://www.r-project.org/), provide a range of clustering methods. Alternatively, dedicated toolkits for clustering are also available (e.g. Cluster [57], CLUTO [58]).

Clustering methods (Box 2) can be used to group different process runs into subsets (groups) of runs according to the similarity in the behavior of some features. For example, in some process runs the time profiles of cell density and metabolite concentrations are more similar to one another than in the remaining runs being studied and these can be clustered into one group. Clustering can thus provide insights into different types of runs. In addition, by using various cluster visualization tools (e.g. Spotfire [23]), these methods can also identify the features that distinguish the clusters. Clustering tools are extensively used in the analysis of large-scale gene expression datasets [24]. For example, use of hierarchical clustering to group gene expression profiles of several prostate cancer and normal prostate samples identified clinically relevant tumor subtypes that could be correlated with increased disease recurrence [25].

A critical element of clustering methods is the approach used to estimate the similarity between any two runs based on their set of temporal features. To account for the heterogeneity of the temporal features associated with each run, the similarity between two runs is often assessed in two steps. First, the similarity between the corresponding temporal features of a pair of runs is determined and second, the overall similarity between the runs is established by aggregating the individual feature-wise similarities (Figure 3). The feature-wise similarity can be computed using various approaches [26]. The most commonly used are Euclidean distance, cosine similarity, and the Pearson's correlation coefficient. Other measures that are based on information theory, such as mutual information, can also be used [27]. Mutual information estimates the general dependency between the profiles of two (or more) features, but can only be used for features that have discrete values (e.g. a SAX-represented profile). Note that these methods for assessing similarity can be applied for comparing the same feature across different runs (for pattern recognition), as well as comparing different features of the same run (for dimensionality reduction).

Predictive approaches

Predictive approaches can be used to analyze a set of process runs that exhibit different outcomes (e.g. final product concentration) to identify the relationship between process features and the outcome. The discovered relationships (called model or classifier) can be used to predict the process outcome and provide key insights into how the predicted outcome might affect other features of the run, thereby allowing for an intelligent outcome-driven refinement of the process parameters. Commonly used predictive methods (Box 3) include regression, decision trees (DT), artificial neural networks (ANN), and support vector machines (SVM). These methods have been designed for problems that arise when process runs are divided into discrete classes. Often, the process outcome (such as product titer) is a value within a certain range, rather than a discrete variable (such as high- or low-producing runs). In such cases, one can divide the outcome into several classes. Alternatively, regression-based methods can be employed to predict an outcome variable that is continuous.

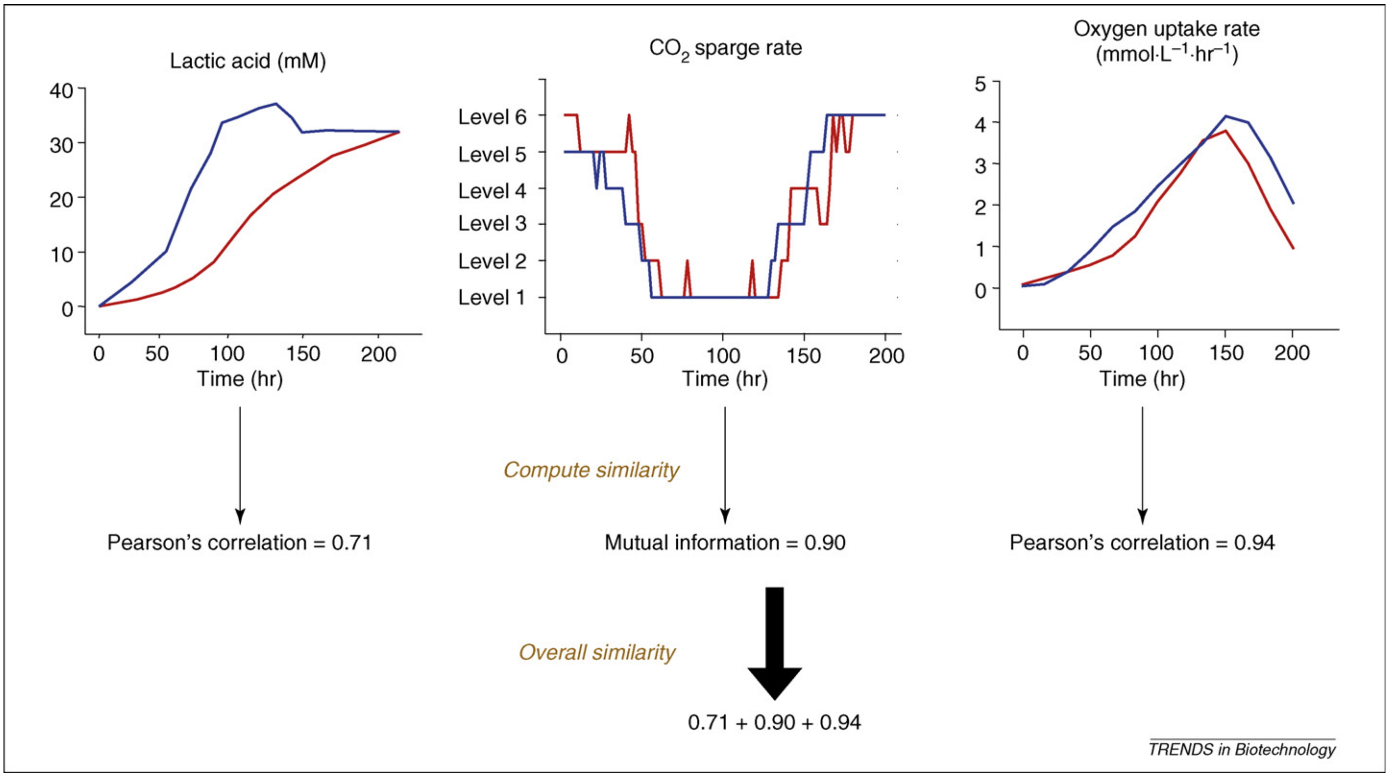

Figure 3. An approach to determine the similarity between different process runs. The profiles of different run features, i.e. lactic acid concentration, CO$_2$ sparge rate, and oxygen uptake rate (OUR), are shown for two runs (in red and blue). The obtained continuous profiles of lactic acid and OUR were compared using a Pearson's correlation [26]. The noisy and long raw profiles of CO$_2$ sparge rates were discretized into six levels using symbolic aggregate approximation (SAX) method [50]. The levels 1 through 6 represent increasing intervals of CO$_2$ sparge rates. The discrete profiles of CO$_2$ sparge rates were compared by estimating their mutual information. The overall similarity between the two runs can then be estimated as an aggregate of these similarities. Before aggregation, the similarity metrics should be normalized to ensure that they have the same range. When prior knowledge is available, the aggregation of the feature-wise similarities can be done in a weighted fashion to give greater importance to some of the features.

Predictive approaches have been extensively used to analyze bioprocess data. Several studies have employed ANNs to predict the output of a fermentation process as a nonlinear function of the process inputs [28–31]. ANN models can also be used in conjunction with optimization methods to identify the combination of process inputs that are able to maximize the desired output [15,32]. Decision trees have also been beneficial for identifying the process trends that allow one to discriminate between runs with high and low productivity [5,16]. For example, a low glucose feed rate was identified as the most discerning process feature for a high productivity run [16]. More recently, a

Box 3. Predictive data mining methods

Three of the commonly used predictive methods are summarized below. Other methods, such as *k*-nearest neighbors [59], and Bayesian networks [60] can also be employed. For simplicity, a binary scheme in which process runs are classified as 'high' or 'low' is used in these descriptions.

Artificial neural networks (ANN) [61]
ANN models attempt to imitate the signal processing events that occur in the interconnected network of neurons in the brain. An ANN consists of several nodes that are organized into two or more layers. The first layer serves as input for process features and the final layer determines the run outcome. Any intermediate layers are referred to as hidden layers. Every node of a hidden layer receives all inputs from the previous layer, performs a weighted average of the inputs and sends its output to the next layer after a threshold transformation. A sigmoidal transformation is commonly used instead of a sharp threshold function. This process is continued until the final output layer is reached. The weighting factors and threshold parameters are learnt from the training runs in an attempt to minimize the error in classifying the runs.

Decision trees (DT) [62]
DT-based classifiers classify runs recursively based on chosen thresholds for one or more features. The process feature that provides most information about the classes is used to split the runs into two or more branches. Splitting thus results in 'child' nodes that are most separated from each other in terms of the class. Thus, selecting a feature and its threshold for the split is a key exercise for DT classifiers. This division is repeated until all the runs at a particular node belong to a single class (terminal node) or one or more stopping rules are satisfied. A top-down interpretation of a decision tree is intuitive and it also allows ranking of process features according to their relevance.

Support vector machines (SVM) [63]
Based on the structural risk minimization principle, SVMs learn a 'decision boundary' that maximizes the separation between runs from the two groups. The training runs that are closest to the decision boundary and hence most informative are called support vectors. The decision function is calculated based on these support vectors only; the runs distant from the boundary are ignored. Mathematically, SVM is formulated as a convex optimization problem. A soft-margin approach, where violations of the decision boundary are penalized with a cost function, generally provides a more robust solution. SVMs also present a well-suited method for kernel-based learning. One-class [64] and multiclass [65] extensions of SVMs have considerably broadened their applications.

regression method based on partial least squares (PLS) has been used to identify predictive correlations between output parameters and process parameters to characterize the process and detect process abnormalities. Furthermore, PLS-based assessment of the similarity of the temporal parameter profiles for process runs at two different reactor scales (2L and 2000L) suggested process comparability at different scales [33].

Recent advances in predictive methods have significantly enhanced their applicability for process data mining. The development of the Vapnik-Chervonenkis theory has laid the foundations of the structural risk

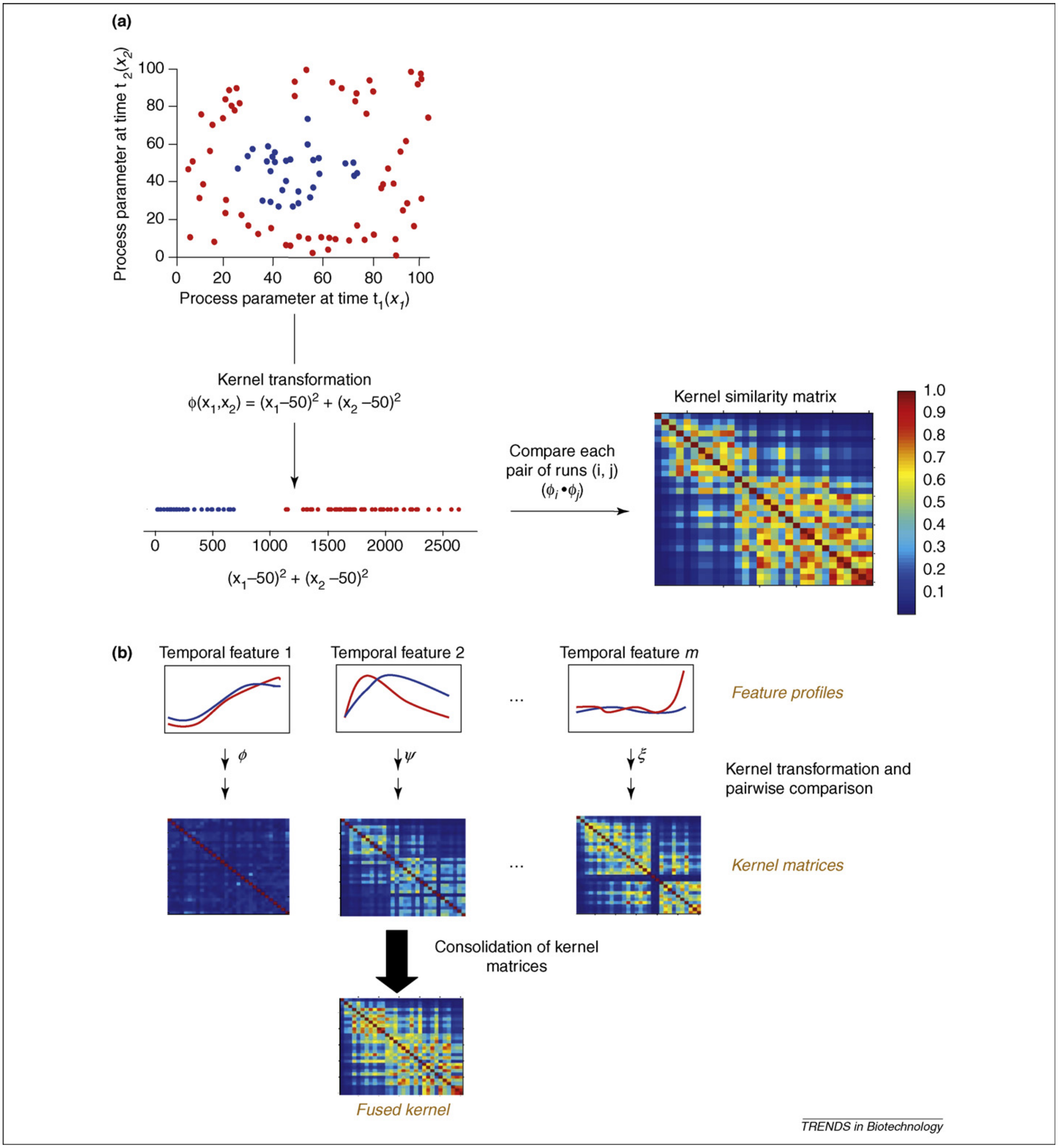
(a)
Process parameter at time t$_2$(x_2)
100
80
60
40
20
0
0
20
40
60
80
100
Process parameter at time t$_1$(x_1)
Kernel transformation
$\phi(x_1,x_2) = (x_1-50)^2 + (x_2-50)^2$
0
500
1000
1500
2000
2500
$(x_1-50)^2 + (x_2-50)^2$
Compare each pair of runs (i, j)
($\phi_i \bullet \phi_j$)
Kernel similarity matrix
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
(b)
Temporal feature 1
Temporal feature 2
...
Temporal feature m
Feature profiles
ϕ
ψ
ξ
Kernel transformation and pairwise comparison
...
Kernel matrices
Consolidation of kernel matrices
Fused kernel
TRENDS in Biotechnology

Figure 4. A kernel-based learning approach. **(a)** A simplified scheme of the approach is illustrated. Process data of a single parameter at two different time points is shown for a set of runs categorized into two classes based on process outcome: high (in blue) or low (in red). The distinction between the two classes is immediately obvious after the data have been transformed using a specifically designed kernel function (ϕ), which in this example results in a visible 'separation' of the runs. Thereafter, a kernel matrix is obtained by computing the similarity between each pair of run parameters on a scale from dissimilar (0) to identical (1). Note that the diagonal entries in the kernel matrix are 1, i.e. a run is identical to itself. **(b)** Several different kernel transformations can be performed to compare different temporal features. The resulting kernel matrices for individual features can then be combined to obtain a fused kernel that can be used for model construction.

minimization (SRM) principle [34,35], which derives the upper limit on the *generalization error* of a classifier. This upper limit is optimized by classifiers that maximize the separation (called margin) between instances from two (or more) classes. Due to its strong mathematical foundations and intuitive appeal, the idea of maximizing the separation between two groups has gained immense popularity and has been successfully used to improve the predictive robustness of several well-known classification methods, such as ANN [36], *k*-nearest neighbors [37], and regression.

Another major development was the introduction of kernel-based learning that decouples the optimization step in many classification approaches from any data modeling aspects. Kernel-based methods employ a kernel function, which measures the similarity between each pair of runs (Figure 4a). A pair-wise comparison of all the runs results in a kernel matrix, which is then used to construct the model. Kernels also provide an elegant solution for addressing the heterogeneity of process data. Multiple kernels can be used, where each kernel serves to compare one temporal process feature (e.g. oxygen uptake rate, osmolarity) over different runs. Kernel functions that quantify linear or nonlinear relationships, or even empirically defined functions based on process knowledge and/or historical data, can be used to compute the pair-wise similarities of a particular process feature across different runs. Individual kernels can then be compiled into a 'fused' kernel (Figure 4b). Furthermore, the individual features (or their kernels) can be differentially weighted in such a way that the features that are more predictive of the process outcome have higher contribution to the final fused kernel. This step of sorting different features according to their relative importance can be incorporated in the process of model construction. The weights of different features can be 'learned' from the data in such a way that the predictability of the model is maximal [38,39]. The SRM principle and kernel-based learning also form the basis of support vector machines (SVM) (Box 3), a relatively novel method that has already been widely used to analyze several data-rich applications, such as gene expression analysis [40,41], text classification [42], and image retrieval [43].

Model validation and interpretation

Discovery of a model pattern or trend must be followed by subsequent evaluation and expert interpretation. In descriptive methods, it is important to examine whether a pattern or a cluster represents a genuine relationship between the performances of different process runs or is simply the outcome of a spurious trend. In addition, noise in process measurements can obscure the interpretation of a discovered pattern. Furthermore, many clustering algorithms are designed to find a set of clusters that are only locally optimized. For example, the initial assignment of the runs to clusters (which is often random) may have an effect on the final clustering, and different initial assignments may lead to different groupings of the runs. Resampling-based approaches have been proposed to evaluate the reproducibility of a set of clusters [44,45]. In these procedures, a subset of runs can be sampled from the original dataset and clustering performed. This process is repeated multiple times and the agreement of the resulting clusters is compared across all the subsets and is used to assign a confidence term for the clustering.

Predictive methods run the risk of constructing an *overfitted* model. Datasets where the number of process features is much higher than the number of runs used for model construction are particularly vulnerable to overfitting. To avoid this, it is essential to assess the predictive ability of a model for new runs. A subset of runs (*training set*) is used for model construction and the remaining runs (*test set*) are used for model evaluation. Error rates are calculated based on the number of test runs misclassified by the model. For datasets with finite or few runs, cross-validation and resampling schemes (e.g. bootstrap) can be used, where the dataset is divided into multiple training and test subsets to obtain an average estimate of the error [46].

The introduction of a 'selection bias' is another relevant issue for generating models based on a subset of features (selected from the entire feature set). This bias is introduced if all runs (including test set runs) are involved in the feature selection process, and the test set is used merely to validate the model build on the preselected features. Both feature selection and model construction must be implemented on the training subset only, without any input from the test set [47]. Although feature selection strategies have been used in previous reports on process data mining, it is unclear whether these examples involved test objects in the feature selection process [9,10,20].

Concluding remarks

Modern production plants are equipped with sophisticated control systems to ensure high consistency and robustness of production. Nevertheless, fluctuations in process performance invariably occur. Understanding the cause of these fluctuations can greatly enhance process outcome and help to achieve higher performance levels. Given the vast amount of archived process data in a typical modern production plant, the opportunities for unveiling any hidden patterns within the data and recognizing the key characteristics for process enhancement are enormous. The ultimate aim of mining bioprocess data is to gain insights for process advancement or even process innovation. Interpretation by process experts is essential to relate the discovered patterns to cellular physiology, which in turn can generate hypotheses for experimental verification. In a bioreactor operation, ultimately it is the physiological state of the cells that determines the process outcome.

We believe that the benefits to be gained from mining bioprocess data will be immense. These opportunities are met with major advances in data mining tools that have become available in the past decade. The application of these tools to explore bioprocess data will be highly rewarding in the near future.

Acknowledgements

The support for bioprocess data mining work to W.S.H. and G.K. from Genentech, Inc. is gratefully acknowledged. S.C. would like to thank Huzefa Rangwala for useful discussions and Eamonn Keogh for the MatLab codes for SAX and APCA methods.

References

1 Walsh, G. (2006) Biopharmaceutical benchmarks 2006. *Nat. Biotechnol.* 24, 769–776

2 Fayyad U.M., *et al.* (1996) From data mining to knowledge discovery: an overview. In *Advances in Knowledge Discovery and Data Mining*, pp. 1–34, American Association for Artificial Intelligence

3 Cheung, J.T.Y. and Stephanopoulos, G. (1990) Representation of process trends- Part II. The problem of scale and qualitative scaling. *Comput. Chem. Eng.* 14, 511–539

4 Cheung, J.T.Y. and Stephanopoulos, G. (1990) Representation of process trends–part I. A formal representation framework. *Comput. Chem. Eng.* 14, 495–510

5 Bakshi, B.R. and Stephanopoulos, G. (1994) Representation of process trends. 4. Induction of real-time patterns from operating data for diagnosis and supervisory control. *Comput. Chem. Eng.* 18, 303–332

6 Bakshi, B.R. and Stephanopoulos, G. (1994) Representation of process trends—3. Multi-scale extraction of trends from process data. *Comput. Chem. Eng.* 18, 267–302

7 Moult, J. (2006) Rigorous performance evaluation in protein structure modelling and implications for computational biology. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 361, 453–458

8 Tompa, M. *et al.* (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.* 23, 137–144

9 Huang, J. *et al.* (2002) Classification of fermentation performance by multivariate analysis based on mean hypothesis testing. *J. Biosci. Bioeng.* 94, 251–257

10 Kamimura, R.T. *et al.* (2000) Mining of biological data I: identifying discriminating features via mean hypothesis testing. *Metab. Eng.* 2, 218–227

11 Sakoe, H. and Chiba, S. (1978) Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Trans. Acoust. Speech Signal Process.* 26, 43–49

12 Keogh, E. *et al.* (2001) Locally adaptive dimensionality reduction for indexing large time series databases. *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data* 30, 151–162

13 Keogh, E. and Ratanamahatana, C.A. (2005) Exact indexing of dynamic time warping. *Knowl. Inf. Syst.* 7, 358–386

14 Buck, K.K. *et al.* (2002) Identification of critical batch operating parameters in fed-batch recombinant *E. coli* fermentations using decision tree analysis. *Biotechnol. Prog.* 18, 1366–1376

15 Coleman, M.C. *et al.* (2003) An integrated approach to optimization of *Escherichia coli* fermentations using historical data. *Biotechnol. Bioeng.* 84, 274–285

16 Stephanopoulos, G. *et al.* (1997) Fermentation database mining by pattern recognition. *Biotechnol. Bioeng.* 53, 443–452

17 Tai, Y.C. and Speed, T.P. (2006) A multivariate empirical Bayes statistic for replicated microarray time course data. *Ann. Stat.* 34, 2387–2412

18 Storey, J.D. *et al.* (2005) Significance analysis of time course microarray experiments. *Proc. Natl. Acad. Sci. U. S. A.* 102, 12837–12842

19 Bar-Joseph, Z. *et al.* (2003) Comparing the continuous representation of time-series expression profiles to identify differentially expressed genes. *Proc. Natl. Acad. Sci. U. S. A.* 100, 10146–10151

20 Kamimura, R.T. *et al.* (2000) Mining of biological data II: assessing data structure and class homogeneity by cluster analysis. *Metab. Eng.* 2, 228–238

21 Agrawal, R. and Srikant, R. (1994) Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases, VLDB* 1215, 487–499

22 Seno, M. and Karypis, G. (2001) LPMiner: an algorithm for finding frequent itemsets using length-decreasing support constraint. *Proceedings of the 2001 IEEE International Conference on Data Mining* 505–512

23 Ahlberg, C. (1996) Spotfire: an information exploration environment. *SIGMOD Rec.* 25, 25–29

24 D'Haeseleer, P. (2005) How does gene expression clustering work? *Nat. Biotechnol.* 23, 1499–1501

25 Lapointe, J. *et al.* (2004) Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *Proc. Natl. Acad. Sci. U. S. A.* 101, 811–816

26 Duda, R.O. *et al.* (2000) *Pattern Classification,* Wiley-Interscience

27 Slonim, N. *et al.* (2005) Information-based clustering. *Proc. Natl. Acad. Sci. U. S. A.* 102, 18297–18302

28 Glassey, J. *et al.* (1994) Enhanced supervision of recombinant *E. coli* fermentations via artificial neural networks. *Process Biochem.* 29, 387–398

29 Glassey, J. *et al.* (1994) Artificial neural network based experimental design procedures for enhancing fermentation development. *Biotechnol. Bioeng.* 44, 397–405

30 Bachinger, T. *et al.* (2000) Electronic nose for estimation of product concentration in mammalian cell cultivation. *Bioprocess Biosyst. Eng.* 23, 637–642

31 Vlassides, S. *et al.* (2001) Using historical data for bioprocess optimization: modeling wine characteristics using artificial neural networks and archived process information. *Biotechnol. Bioeng.* 73, 55–68

32 Coleman, M.C. and Block, D.E. (2006) Retrospective optimization of time-dependent fermentation control strategies using time-independent historical data. *Biotechnol. Bioeng.* 95, 412–423

33 Kirdar, A.O. *et al.* (2007) Application of multivariate analysis toward biotech processes: case study of a cell-culture unit operation. *Biotechnol. Prog.* 23, 61–67

34 Vapnik, V.N. (1998) *Statistical Learning Theory,* Wiley-Interscience

35 Vapnik, V.N. (2000) *The Nature of Statistical Learning Theory,* Springer

36 Li, Y. and Long, P.M. (2002) The relaxed online maximum margin algorithm. *Mach. Learn.* 46, 361–387

37 Weinberger, K. *et al.* (2006) Distance metric learning for large margin nearest neighbor classification. *Adv. Neural Inf. Process. Syst.* 18, 1473–1480

38 Lanckriet, G.R. *et al.* (2004) A statistical framework for genomic data fusion. *Bioinformatics* 20, 2626–2635

39 Lanckriet, G.R.G. *et al.* (2004) Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.* 5, 27–72

40 Brown, M.P. *et al.* (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. U. S. A.* 97, 262–267

41 Charaniya, S. *et al.* (2007) Transcriptome dynamics-based operon prediction and verification in *Streptomyces coelicolor*. *Nucleic Acids Res.* 35, 7222–7236

42 Tong, S. and Koller, D. (2002) Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* 2, 45–66

43 Tong, S. and Chang, E. (2001) Support vector machine active learning for image retrieval. *Proceedings of the Ninth ACM Internatioinal Conference on Multimedia* 9, 107–118

44 Kerr, M.K. and Churchill, G.A. (2001) Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. *Proc. Natl. Acad. Sci. U. S. A.* 98, 8961–8965

45 Monti, S. *et al.* (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression Microarray data. *Mach. Learn.* 52, 91–118

46 Kohavi, R. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* 2, 1137–1145

47 Ambroise, C. and McLachlan, G.J. (2002) Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl. Acad. Sci. U. S. A.* 99, 6562–6566

48 Lio, P. (2003) Wavelets in bioinformatics and computational biology: state of art and perspectives. *Bioinformatics* 19, 2–9

49 Bakshi, B.R. and Stephanopoulos, G. (1996) Compression of chemical process data by functional approximation and feature extraction. *AIChE J.* 42, 477–492

50 Lin, J. *et al.* (2003) A symbolic representation of time series, with implications for streaming algorithms. *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* 2–11

51 Guyon, I. and Elisseeff, A. (2003) An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3, 1157–1182

52 Golub, T.R. *et al.* (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537

53 Ringner, M. (2008) What is principal component analysis? *Nat. Biotechnol.* 26, 303–304

54 Lee, D.D. and Seung, H.S. (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 788–791

55 Han, J. *et al.* (2004) Mining frequent patterns without candidate generation: a frequent-pattern tree approach. *Data Min. Knowl. Discov.* 8, 53–87

56 Jain, A.K. *et al.* (1999) Data clustering: a review. *ACM Comput. Surv.* 31, 264–323

57 Eisen, M.B. *et al.* (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. U. S. A.* 95, 14863–14868

58 Zhao, Y. and Karypis, G. (2003) Clustering in life sciences. In *Functional Genomics: Methods and Protocols* (Brownstein, M.J. and Khodursky, A., eds), pp. 183–218, Humana Press

59 Fix, E. and Hodges, J.L. (1951) *Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties*, F School of Aviation Medicine, (U. S. A)

60 Needham, C.J. *et al.* (2006) Inference in Bayesian networks. *Nat. Biotechnol.* 24, 51–53

61 Krogh, A. (2008) What are artificial neural networks? *Nat. Biotechnol.* 26, 195–197

62 Quinlan, J.R. (1990) Decision trees and decision-making. *IEEE Trans. Syst. Man Cybern.* 20, 339–346

63 Noble, W.S. (2006) What is a support vector machine? *Nat. Biotechnol.* 24, 1565–1567

64 Scholkopf, B. *et al.* (2001) Estimating the support of a high-dimensional distribution. *Neural Comput.* 13, 1443–1471

65 Weston, J. and Watkins, C. (1999) Support vector machines for multi-class pattern recognition. *Proceedings of the Seventh European Symposium on Artificial Neural Networks* 4, 219–224