AI Magazine Volume 25 Number 1 (2004) (© AAAI)

Using Machine Learning to Design and Interpret Gene-Expression Microarrays

Michael Molla, Michael Waddell, David Page, and Jude Shavlik

■ Gene-expression microarrays, commonly called *gene chips,* make it possible to simultaneously measure the rate at which a cell or tissue is expressing—translating into a protein—each of its thousands of genes. One can use these comprehensive snapshots of biological activity to infer regulatory pathways in cells; identify novel targets for drug design; and improve the diagnosis, prognosis, and treatment planning for those suffering from disease. However, the amount of data this new technology produces is more than one can manually analyze. Hence, the need for automated analysis of microarray data offers an opportunity for machine learning to have a significant impact on biology and medicine. This article describes microarray technology, the data it produces, and the types of machine learning tasks that naturally arise with these data. It also reviews some of the recent prominent applications of machine learning to gene-chip data, points to related tasks where machine learning might have a further impact on biology and medicine, and describes additional types of interesting data that recent advances in biotechnology allow biomedical researchers to collect.

Almost every cell in the body of an organism has the same deoxyribonucleic acid (DNA). Genes are portions of this DNA that code for proteins or (less commonly) other large biomolecules. As Hunter covers in his introductory article in this special issue (and, for completeness, we review in the next section of this article), a gene is expressed through a two-step process in which the gene's DNA is first transcribed into ribonucleic acid (RNA), which is then translated into the corresponding protein. A novel technology of gene-expression microarrays—whose development started in the second half of the 1990s and is having a revolutionary impact on molecular biology—allows one to monitor the DNA-to-RNA portion of this fundamental biological process.

Why should this new development in biology interest researchers in machine learning and other areas of AI? Although the ability to measure transcription of a single gene is not new, the ability to measure at once the transcription of all the genes in an organism is new. Consequently, the amount of data that biologists need to examine is overwhelming. Many of the data sets we describe in this article consist of roughly 100 samples, where each sample contains about 10,000 genes measured on a gene-expression microarray. Suppose 50 of these patients have one disease, and the other 50 have a different disease. Finding some combination of genes whose expression levels can distinguish these two groups of patients is a daunting task for a human but a relatively natural one for a machine learning algorithm. Of course, this example also illustrates a challenge that microarray data pose for machine learning algorithms—the dimensionality of the data is high compared to the typical number of data points.

The preceding paragraph gives one natural

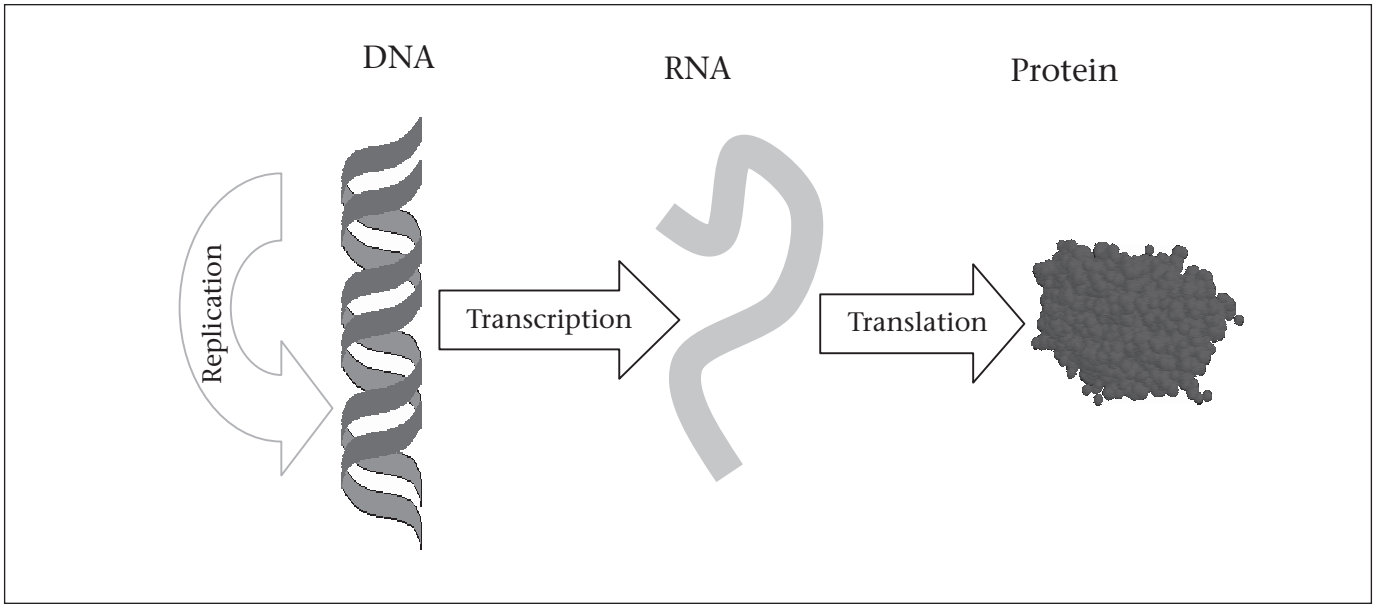


Figure 1. The Central Dogma of Molecular Biology.

When a gene is expressed, it is first transcribed into an RNA sequence, and the RNA is then translated into a protein, a sequence of amino acids. DNA is also replicated when a cell divides, but this article only focuses on the DNA-to-RNA-to-protein process.

example of how one can apply machine learning to microarray data. There are many other tasks that arise in analyzing microarray data and correspondingly many ways in which machine learning is applicable. We present a number of such tasks, with an effort to describe each task concisely and give concrete examples of how researchers have addressed such tasks, together with brief summaries of their results.[1] Before discussing these particular tasks and approaches, we summarize the relevant biology and biotechnology. This article closes with future research directions, including the analysis of several new types of high-throughput biological data, similar to microarray data, that are becoming available based on other advances in biotechnology.

Some Relevant Introductory Biology

The method by which the genes of an organism are expressed is through the production of proteins,[2] the building blocks of life. This process of gene expression occurs in all organisms, from bacteria to plants to humans. Each gene encodes a specific protein,[3] and at each point in the life of a given cell, various proteins are being produced. It is through turning on and off the production of specific proteins that an organism responds to environmental and biological situations, such as stress, and different developmental stages, such as cell division.

Genes are contained in the DNA of the organism. The mechanism by which proteins are produced from their corresponding genes is a two-step process (figure 1). The first step is the transcription of a gene from DNA into a temporary molecule known as RNA. During the second step—translation—cellular machinery builds a protein using the RNA message as a blueprint. Although there are exceptions to this process, these steps (along with DNA replication) are known as the *central dogma* of molecular biology.

One property that DNA and RNA have in common is that each is a chain of chemicals known as bases.[4] In the case of DNA, these bases are adenine, cytosine, guanine, and thymine, commonly referred to as *A, C, G,* and *T,* respectively. RNA has the same set of four bases, except that instead of thymine, RNA has uracil—commonly referred to as *U*.

Another property that DNA and RNA have in common is called *complementarity.* Each base only binds well with its complement: *A* with *T* (or *U*) and *G* with *C*. As a result of complementarity, a strand of either DNA or RNA has a strong affinity for what is known as its *reverse complement,* which is a strand of either DNA or RNA that has bases exactly complementary to the original strand, as figure 2 illustrates. (Just like in English text, there is a directionality for reading a strand of DNA or RNA. Hence in figure 2, the DNA would be read from left to right, whereas the RNA would be read from right to left, which is why *reverse* is in the phrase *reverse complement.*)

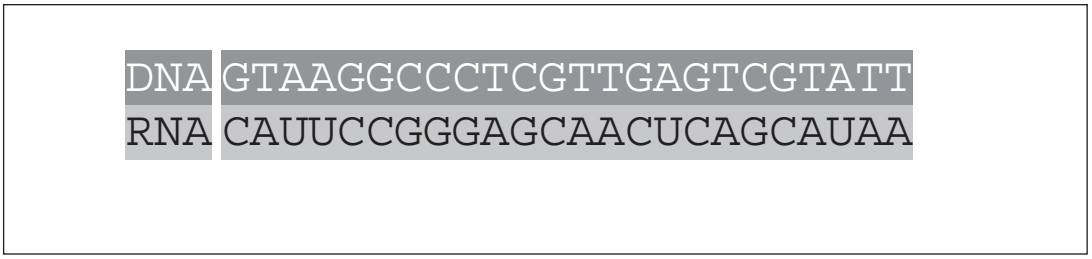


Figure 2. Complementary Binding between DNA and RNA Sequences.

Complementarity is central to the double-stranded structure of DNA and the process of DNA replication. It is also vital to transcription. In addition to its role in these natural processes, molecular biologists have, for decades, taken advantage of complementarity to detect specific sequences of bases within strands of DNA and RNA. One does this detection by first synthesizing a *probe,* a piece of DNA that is the reverse complement of a sequence one wants to detect and then introducing this probe to a solution containing the genetic material (DNA or RNA) to be searched.[5] This solution of genetic material is called the *sample.* In theory, the probe will bind to the sample if and only if the probe finds its complement in the sample (but as we later discuss in some detail, this does not always happen in practice, and this imperfect process provides an excellent opportunity for machine learning). The act of binding between probe and sample is called *hybridization.* Prior to the experiment, one labels the probes using a fluorescent tag. After the hybridization experiment, one can easily scan to see if the probe has hybridized to its reverse complement in the sample. In this way, the molecular biologist can determine the presence or absence of the sequence of interest in the sample.

What Are Gene Chips?

More recently, DNA probe technology has been adapted for detection of not just one sequence but tens of thousands simultaneously. This is done by synthesizing a large number of different probes and either carefully placing each probe at a specific position on a glass slide (so-called *spotted arrays*) or by attaching the probes to specific positions on some surface. Figure 3 illustrates attaching the probes, which has become the predominant approach as the technology has matured. Such a device is called a *microarray* or *gene chip.*[6]

Utilization of these chips involves labeling the sample rather than the probe, spreading thousands of copies of this labeled sample across the chip and washing away any copies of the sample that do not remain bound to some probe. Because the probes are attached at specific locations on the chip, if the labeled sample is detected at any position on the chip, one can determine which probe has hybridized to its complement.

The most common use of gene chips is to measure the *expression level* of various genes in an organism, and in this article we focus on that task (however, the reader should be aware that novel uses of microarrays will be continually devised, offering new opportunities for machine learning). An expression level measures the rate at which a particular gene is being transcribed, which is used as a proxy measure for the amount of corresponding protein that is being produced within an organism's cells at a given time.

Ideally, biologists would measure the protein production rate directly, but doing so is currently very difficult and impractical on a large scale. One instead measures the expression level of various genes by estimating the amount of RNA for that gene that is currently present in the cell. Because the cell degrades RNA very quickly, this level will accurately reflect the rate at which the cell is producing the corresponding protein. To find the expression level of a group of genes, one labels the RNA from a cell or a group of cells and spreads the RNA across a chip that contains probes for the genes of interest. A single gene chip can hold enough probes to monitor tens of thousands of genes.

Data Collection and Preprocessing

When one runs a microarray experiment, an optical scanner records the *fluorescence-intensity values*—the level of fluorescence at each spot on the gene chip. (Machine learning might also be able to improve this image-processing step, but we do not address that task in this article.) In the case of gene-expression arrays, typically many experiments measure the same set of genes under various circumstances (for example, when the conditions are normal, when the

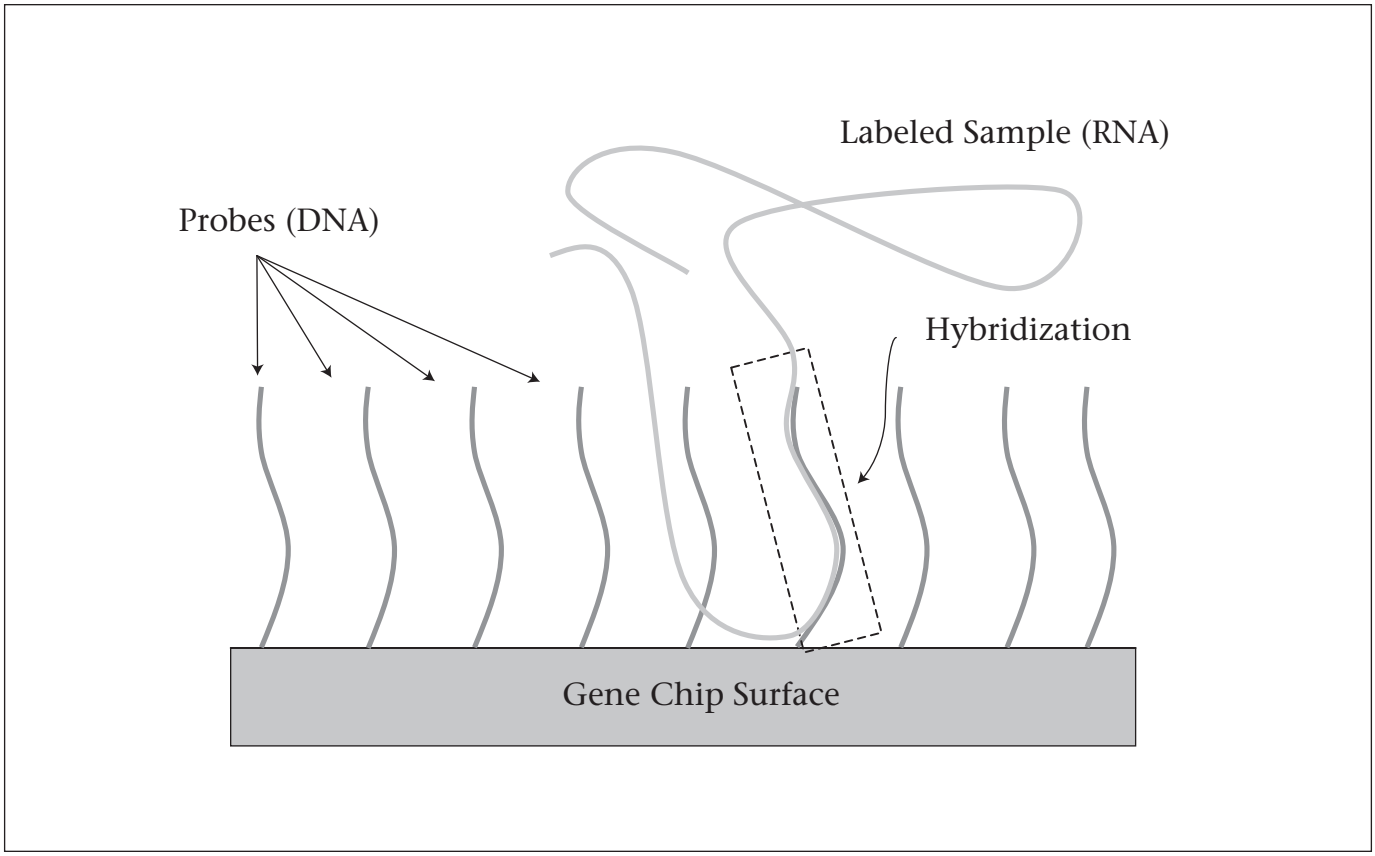


Figure 3. Hybridization of Sample to Probe.

Probes are typically on the order of 25 bases long, whereas samples are usually about 10 times as long, with a large variation as a result of the process that breaks up long sequences of RNA into small samples (one way this is done is by *sonication*, the use of sound waves).

cell is heated up or cooled down, or when some drug is added) or at various time points (such as 5, 10, and 15 minutes after adding an antibiotic; because the steps one needs to manually perform to produce an RNA sample, subminute resolution is not currently feasible).

From the perspective of machine learning, one can organize the measured expression values in several ways, as table 1 illustrates. Tables 1a and 1c show that one can view each gene as an example; here the expression levels measured under various conditions constitute each example's features. Alternatively (table 1b and 1d), one can view each experiment as an example; in this case, the features are the expression values for all the genes on the microarray. In either case, the examples can be unlabeled (tables 1a and 1b) or labeled (tables 1c and 1d) according to some category of interest; for example, some sets of measurements might come from normal cells, and the others from cancerous cells. As we discuss throughout this article, the specific learning task of interest will dictate which among these scenarios gives the most appropriate perspective on the data. We describe, for each of the four scenarios shown in table 1, at least one published project that views microarray data according to that scenario.

To this point, we have presented the process of measuring gene-expression levels as simply creating one probe for each gene and then computing how much RNA is being made by measuring the fluorescence level of the probe-sample hybrid. Not surprisingly, there are complications, and the remainder of this section summarizes the major ones.

Probes on gene chips (figure 3) are typically on the order of 25 bases long because synthesizing longer probes is not practical. Genes are on the order of a 1000 bases long, and although it might be possible to find a unique 25-base-long probe to represent each gene, most probes do not hybridize to their corresponding sample as well as one would like. For example, a given probe might partially hybridize to other samples, even if the match is not perfect, or the sample might fold up and hybridize to itself. For these reasons, microarrays typically use about a dozen or so probes for each gene, and an algorithm combines the measured fluorescence levels for each probe in this set to estimate the expression level for the associated gene.

A

Examples	Features			
	Experiment 1	Experiment 2	…	Experiment *N*
Gene 1	1083	1464	…	1115
Gene 2	1585	398	…	511
…	…	…	…	…
Gene M	170	302	…	751

B

Examples	Features			
	Gene 1	Gene 2	…	Gene *M*
Experiment 1	1083	1585	…	170
Experiment 2	1464	398	…	302
…	…	…	…	…
Experiment *N*	1115	511	…	751

C

Examples	Features				
	Experiment 1	Experiment 2	…	Experiment *N*	Category
Gene 1	1083	1464	…	1115	Y
Gene 2	1585	398	…	511	X
…	…	…	…	…	…
Gene *M*	170	302	…	751	X

D

Examples	Features				
	Gene 1	Gene 2	…	Gene *M*	Category
Experiment 1	1083	1585	…	170	B
Experiment 2	1464	398	…	302	A
…	…	…	…	…	…
Experiment *N*	1115	511	…	751	B

Table 1. Different Ways of Representing Microarray Expression Data for Machine Learning.

A. In this panel, each example contains the measured expression levels of a single gene under a variety of conditions. B. In this panel, each example contains the measured expression levels of thousands of genes under one condition. C, D. These panels illustrate that one can also associate categories with each example, such as the type of cell from which the genes came (for example, normal versus diseased). A and B illustrate the structure of data sets for unsupervised learning, and C and D illustrate the structure of data sets for supervised learning.

Because of the nature of these experiments, including the fact that microarrays are still a nascent technology, the raw signal values typically contain a great deal of *noise*. Noise can be introduced during the synthesis of probes, the creation and labeling of samples, or the reading of the fluorescent signals. Ideally, the data illustrated by table 1 will include replicated experiments. However, each gene-chip experiment can cost several hundred dollars, so in practice, one only replicates each experiment a very small number of times (and, unfortunately, often no replicated experiments are done).

Currently, it is not possible to accurately estimate the absolute expression level of a given gene. One workaround is to compute the ratio of fluorescence levels under some experimental condition to those obtained under normal or control conditions. For example, one might compare gene expression under normal circum-

URL (viable as of 2003)	Brief Description
www.ebi.ac.uk/arrayexpress/	EBI microarray data repository
www.ncbi.nlm.nih.gov/geo/	NCBI microarray data repository
genome-www5.stanford.edu/MicroArray/SMD/	Stanford microarray database
rana.lbl.gov/EisenData.htm	Eisen-lab's yeast data (Spellman et al. 1998)
www.genome.wisc.edu/functional/microarray.htm	University of Wisconsin *E. coli* Genome Project
llmpp.nih.gov/lymphoma/data.shtml	Diffuse large B-cell lymphoma (Alizadeh et al. 2000)
llmpp.nih.gov/DLBCL/	Molecular profiling (Rosenwald et al. 2002)
www.rii.com/publications/2002/vantveer.htm	Breast cancer prognosis (Van't Veer et al. 2002)
www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi	MIT Whitehead Center for Genome Research, including data in Golub et al. (1999)
lambertlab.uams.edu/publicdata.htm	Lambert Laboratory data for multiple myeloma
www.cs.wisc.edu/~dpage/kddcup2001/	KDD Cup 2001 data; Task 2 includes correlations in genes' expression levels
www.biostat.wisc.edu/~craven/kddcup/	KDD Cup 2002 data; Task 2 includes gene-expression data
clinicalproteomics.steem.com/	Proteomics data (mass spectrometry of proteins)
snp.cshl.org/	Single nucleotide polymorphism data

Table 2. URLs for Some Publicly Available Microarray Data Sets.

stances to when the cell is heated to a higher than normal temperature (so called *heat shock*); experimenters might say, "when *E. coli* is heated, gene *X* is expressed at twice its normal rate." When dealing with such ratios, the problem of noise is exacerbated, especially when the numerator and denominator are small numbers. Newton et al. (2001) have developed a Bayesian method for more reliably estimating these ratios. In some studies, the numbers in table 1 are gene-expression ratios, hopefully corrected to minimize the problems that arise from creating ratios of small, noisy numbers.

Another approach is to partner each probe with one or more *mismatch probes*; these are probes that have different bases from the probe of interest in one or more positions. Each gene's expression score is then a function of the fluorescence levels of the dozen or so match and mismatch probes (Li and Wong 2001).

Table 2 contains World Wide Web URLs for some freely available, gene-expression data sets, many of which we discuss further in this article.

Machine Learning to Aid the Design of Microarrays

As described in the previous section, one typically uses a dozen or so probes to represent one gene because the probe-sample binding process is not perfect (Breslauer et al. 1986). If one did a better job of picking good probes, one could not only use fewer probes for each gene (and, hence, test for more genes for each microarray) but also get more accurate results.

Tobler et al. (2002) have used machine learning to address the task of choosing good probes. It is easy to get training examples for this task; simply place all possible probes for a given set of genes (for example, every 24-base subsequence of each gene) on a microarray and see which probes produce strong fluorescence levels when the corresponding gene's RNA is in the sample applied to the gene chip. Figure 4 shows a portion of the data that Tobler and colleagues used, and table 3 illustrates how they cast probe selection as a machine learning task.

Tobler et al. (2002) used a microarray supplied by NimbleGen Systems (Nuwaysir et al. 2002), a microarray company, containing all possible probes from eight different bacterial genes. They exposed that chip to a sample of RNA known to contain all eight of those genes. They then measured the fluorescence level at each location on the chip. If the probes all hybridized equally well, then there would be a uniformly high signal across the entire chip. However, as is clear in figure 4, this is not the case. Instead, some probes hybridize well, and others do not. They used 67 features (see table 4) to represent each probe and used several

well-known learning algorithms to learn how to predict whether a candidate probe sequence is likely to be a good one.

Tobler et al. (2002) found that of the 10 probes predicted by a trained neural network to be the best for each gene, over 95 percent satisfy their definition for being a good probe. When randomly selecting probes, only 13 percent satisfy their good-probe definition.

Machine Learning in Biological Applications of Microarrays

In this section, we provide some examples of the use of microarrays to address questions in molecular biology, focusing on the role played by machine learning. We cover both supervised and unsupervised learning as well as discuss some research where microarray data are just one of several types of data given to machine learning algorithms. Most of these studies involve what are called *model organisms* (bacteria, yeast, fruit flies, and so on), on which it is much easier to perform carefully controlled experiments. The subsequent section mainly addresses the application of microarrays to human data.

Supervised Learning and Experimental Methodology

Supervised learning methods train on examples whose categories are known to produce a model that can classify new examples that have not been seen by the learner. Evaluation of this type of learner is typically done through the use of a method called *N-fold cross-validation,* a form of hold-out testing. In hold-out testing, some (for example, 90 percent) of the examples are used as the training data for a learning algorithm, and the remaining ("held aside") examples are used to estimate the future accuracy of the learned model. In *N*-fold cross-validation, the examples are divided into *N* subsets, and then each subset is successively used as the held-aside test set, and the other (*N*–1) subsets are pooled to create the training set. The results of all *N* test-set runs are averaged to find the total accuracy. The typical value for *N* is 10. In fact, the probe-selection project the previous section describes is an application of supervised learning, and the described results are measured on held-aside data. (In that project, there were eight genes and eight times the learning algorithms trained on seven genes, and the resulting models were tested on the held-out gene.)

Another application of supervised learning (Brown et al. 2000) deals with the functional classifications of genes. They use a representation of the data similar to the one pictured in table 1c. Genes are the examples, and functional classifications are the classes. The features are the gene-expression values under various experimental conditions. Functional classifications are simply the classes of genes, defined by the genes' function, that have been described by biologists over the years through various methods. Given expression profiles, across multiple experiments, of multiple genes whose functional class is known, Brown et al. train a learner to predict the functional classification of genes whose functional class is not known—see table 4. To do this training, they use a ma-

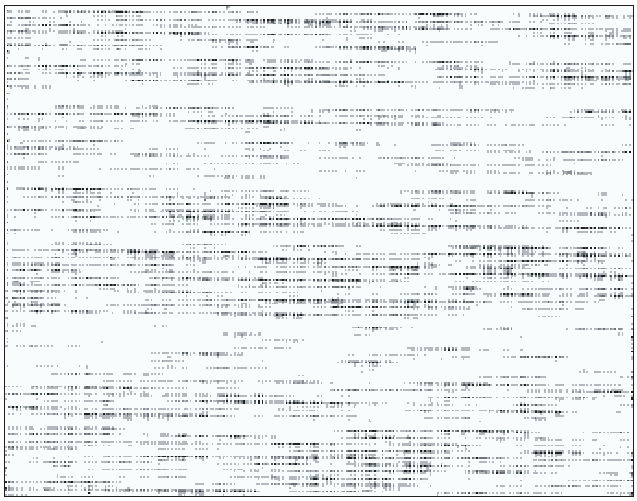


Figure 4. The Result of an Actual Microarray Experiment Where All Possible 24-Base-Long Probes from Eight Bacterial Genes Are on the Chip.

We show one quadrant of the chip. The darker the point, the greater the fluorescence was in the original sample. In the ideal case, all the points would have equally strong fluorescence values; one can use these mappings from probe sequence to fluorescence value as training examples for a machine learning system. *These data were supplied through the courtesy of NimbleGen Systems, Inc.*

Given	A set of probes, each associated with a fluorescence value. Tobler et al. (2002) represent each probe as a vector of 67 feature values: the specific base at each of the 24 positions in the probe sequence; the pair of adjacent bases at each of 23 positions in the probe (for example, the first two bases in a probe might be *AG*); the percentage of *A*s, *C*s, *G*s, and *T*s in the probe; and the percentage of each of the 16 possible pairs of adjacent bases in the probe. They discretize the fluorescence values into three groups: (1) good, (2) ambiguous, and (3) bad (they discard ambiguous probes during training but group them with bad probes during testing).
Do	Learn to choose the best among the possible probes one could use for a new gene.

Table 3. Probe-Quality Prediction.

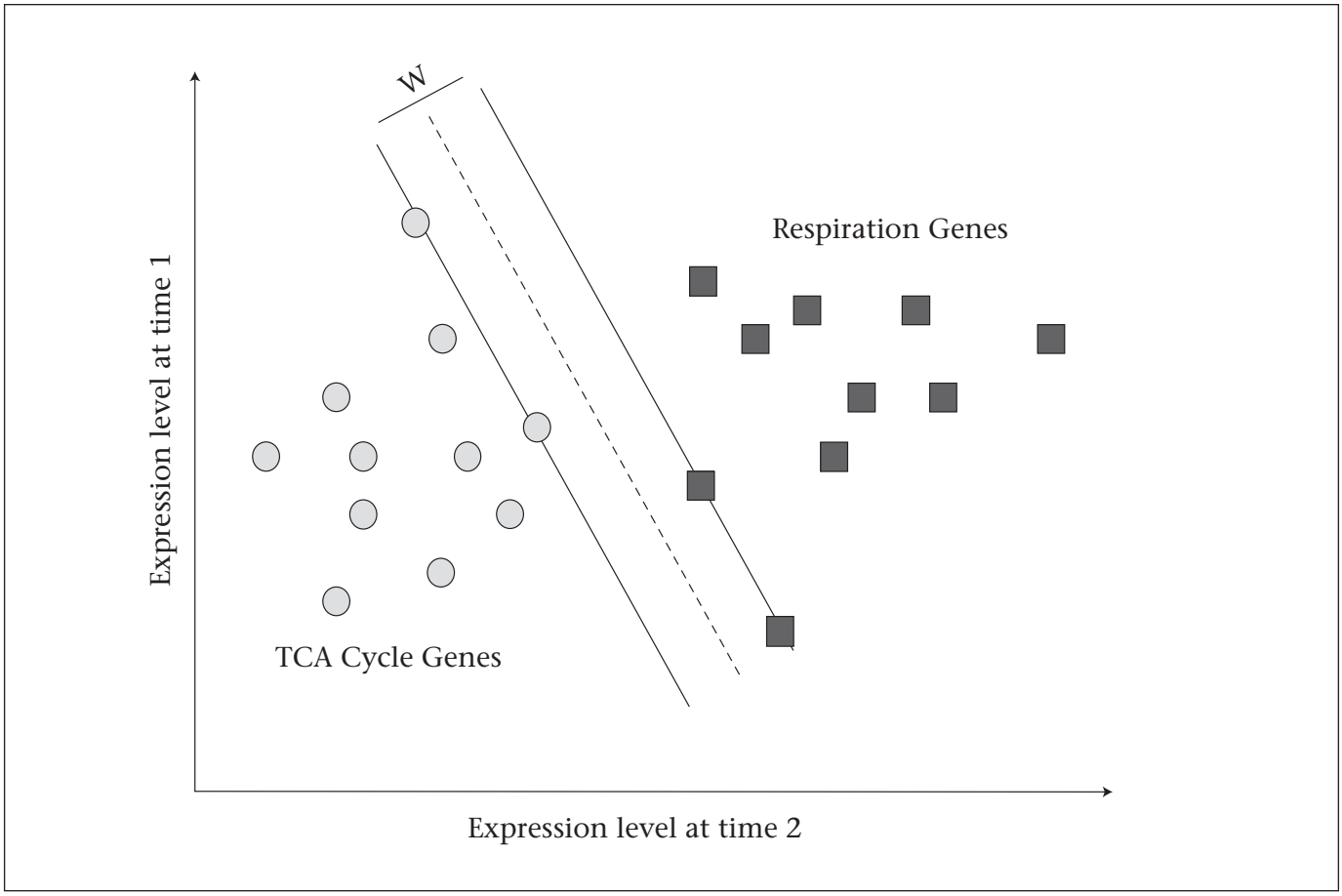


Figure 5. A Support Vector Machine for Differentiating Genes Involved in Respiration from Those Involved in the TCA Cycle by Maximizing the Margin, W.

This is done in the *N*-dimensional space defined by the expression levels of the genes across *N* experimental conditions. In this simple example, there are only two experimental conditions: time 1 and time 2; so, $N = 2$. Normally, however, *N* would be much greater. For example, in the paper by Brown et al. (2000), $N = 79$. The number of genes to categorize would also be much higher. In the Brown et al. paper, the number of genes is 2,467.

Given	A set of genes represented similarly to table 1c. Each gene is an example, whose features are the numeric expression levels measured under multiple experimental circumstances. These experimental conditions include stresses such as temperature shock, change in pH, or the introduction of an antibiotic; other experimental circumstances include different developmental stages of the organism or time points in a series. The category of each gene is simply that gene's functional category. One possible set of functional categories contains the TCA cycle, respiration, cytoplasmic ribosome, proteasome, histone, and helix-turn-helix (see Brown et al. [2000] for explanations of these classes).
Do	Learn to predict the functional category of additional genes given a vector of expression levels under the given set of experimental conditions.

Table 4. Predicting a Gene's Biological Function.

chine learning technique known as a *support vector machine* (SVM).

In its simplest form, an SVM is an algorithm that attempts to find a linear separator between the data points of two classes, as figure 5 illustrates. SVMs seek to maximize the *margin,* or separation between the two classes, to improve the chance of accurate predictions on future data. Maximizing the margin can be viewed as an optimization task solvable using linear or quadratic programming techniques. Of course, in practice there might be no good linear separator of the data. SVMs based on kernel functions can efficiently produce separators that are nonlinear.

Often, kernel functions improve the accuracy of SVMs; however, Brown and colleagues empirically found that for their gene-expression data, simple linear SVMs produce more accurate predictions. Linear SVMs also generalize better than non–SVM supervised learning methods on their data. For example, of the

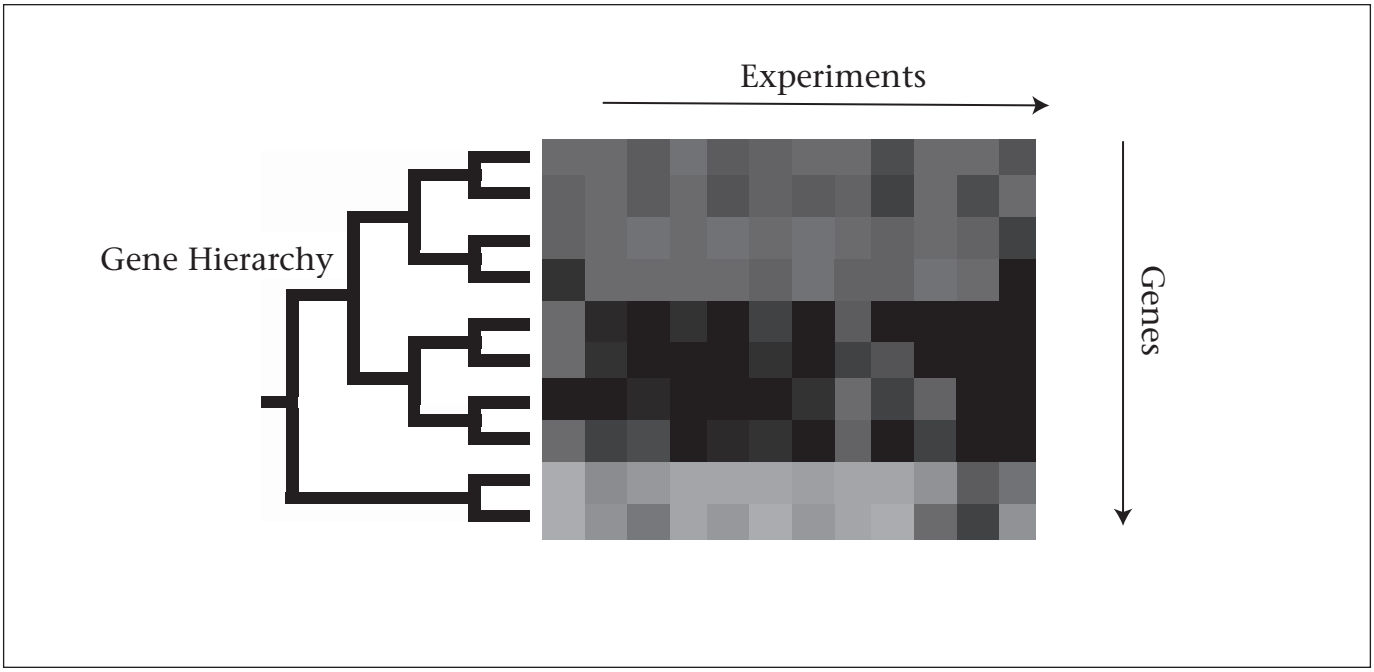


Figure 6. The Graphic Output of a Cluster Analysis.

It is similar to the representation in table 1a, where integers are represented by gray-scale intensity. However, unlike table 1a, the genes here are sorted by similarity (more similar genes, with respect to their vector of expression values, are grouped together). For a more realistic diagram made from real data, see Eisen et al. (1998).

2,467 genes in the data set, the trained SVM correctly identifies 116 of the 121 ribosomal proteins and only produces 6 false positives. The next-best supervised learner correctly identifies the same number but produces eight false positives.

Unsupervised Learning

Unsupervised learning is learning about a set of examples from their features alone; no categories are specified for the examples. Examples of this type are commonly called *unlabeled examples*. Thus, in the context of gene chips, learning models of biological processes and relationships among genes are based entirely on their expression levels without being able to improve models by checking the learners' answers against some sort of externally provided ground truth.

Clustering Methods Many successful efforts in unsupervised learning involve clustering algorithms, including much of the work in the algorithmic analysis of microarray data. Because of the nature of evolution, clustering of biological data makes sense, and this task has a long history in computational biology (in the past, individual protein or DNA sequences were most commonly clustered). Clustering algorithms group, or cluster, examples based on the similarity of their feature values, such as gene-expression values.

Given	A set of genes in an organism represented similarly to table 1a. Each gene is an example. An example's features are the gene's numeric expression levels under various experimental circumstances (environmental stresses, developmental stage, and so on).
Do	Cluster genes based on the similarity of their expression values.

Table 5. Clustering Genes Based on Their Expression Levels.

Eisen et al. (1998) describe one such method. Table 5 presents the problem that they address.

For example, Eisen et al. clustered the expression patterns across a number of experiments of all the genes of the yeast *Saccharomyces cerevisiae* (Spellman et al. 1998). Some of these experiments measure the genetic response to environmental stresses such as cold shock. Others measure transcription during various stages in the life cycle of the organism, such as cell division. Each gene is an example, and the measured expression levels of the gene during each of the experiments are the *features* (that is, the data are in the format of table 1a). They use a standard statistical technique to describe the similarity between any two examples in terms of these features and use that as their distance metric.

More specifically, Eisen et al. perform *hierarchical clustering*. Their algorithm clusters by re-

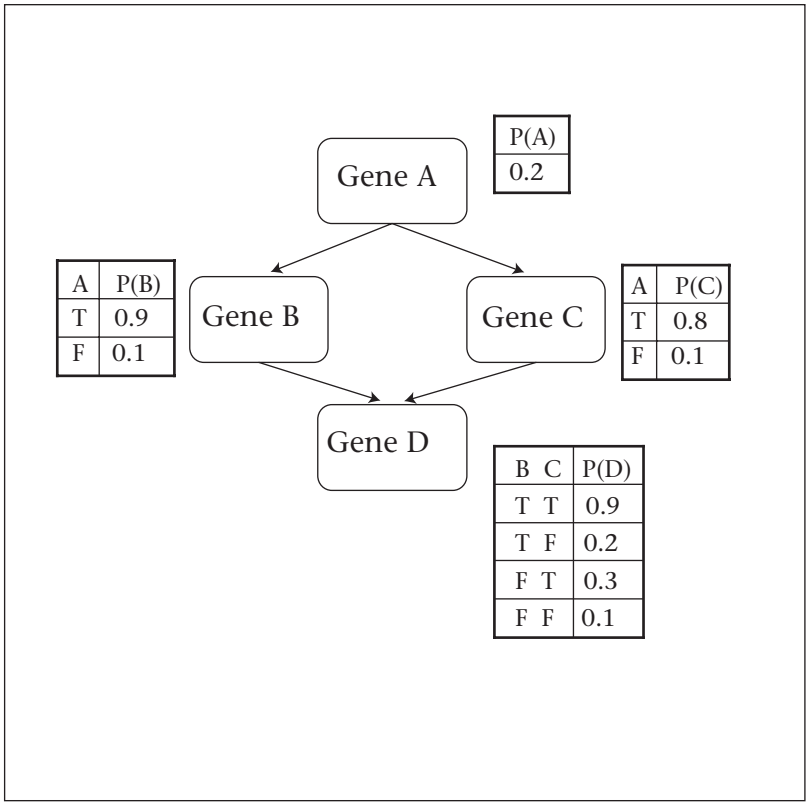


Figure 7. A Simple Bayesian Network.

This illustrative example of a Bayes's network describes the relationships between four hypothetical genes. Each of the probabilities *P*(*X*) refers to the probability that the gene *X* is expressed. Note that the conditional probabilities rely only on the parent variables (that is, other gene's expression levels). For simplicity, in this figure, we consider genes to be either expressed or not expressed. In a richer model, the variables could correspond to a numeric expression level.

Given	A set of microarray experiments represented similarly to table 1b. Each experiment is an example. For example, in Thomas et al. (2001), each experiment involves subjecting mice to one toxic compound. An example's features are the numeric expression levels of the microarray's genes.
Do	Cluster experimental conditions based on the similarity of the gene-expression vectors they produce.

Table 6. Clustering Experimental Conditions Based on Gene-Expression Levels They Produce.

peatedly pairing the two most similar examples, removing those two from the data set, and adding their average to the set of examples. Their method pairs examples and can then later pair these "average" examples, producing a hierarchy of clusters.

Figure 6 shows a hypothetical output of such a hierarchical clustering algorithm. The x-axis spans the experimental conditions, whereas the y-axis spans the genes. The measured expression level of the gene during that experiment relative to that of the organism under normal conditions dictates the shading of the graph; the higher the expression level, the lighter the point. The genes are ordered so that similar genes, with regard to these experimentally derived values, are grouped together visually. The result is an intuitive visual guide for the researcher to quickly discern the blocks of similar genes with regard to a set of experiments.

Because of their flexibility and intuitive nature, clustering methods have proven popular among biologists. In many laboratories that conduct microarray experiments, clustering of genes in microarray experiments is now a standard practice. Clustering of experiments is also a common practice (table 6). For example, Thomas et al. (2001) ran microarrays on RNA from mice subjected to a variety of toxic compounds, with one microarray for each compound. They hierarchically clustered the microarray experiments and found that the clusters correspond closely to the different toxicological classes of the compounds (Thomas et al. also report some supervised learning experiments).

Bayesian Networks Another unsupervised learning algorithm used for the analysis of microarray data is known as the Bayesian network, or Bayes's net. A *Bayes's net* is a directed acyclic graph that specifies a joint-probability distribution over its variables. Arcs between nodes specify dependencies among variables, and the absence of arcs can be used to infer conditional independencies; figure 7 contains a simple example. By capturing conditional independence where it exists, a Bayes's net can provide a much more compact representation of the joint-probability distribution than a full joint table. Every node in a Bayes's net has an associated conditional probability table that specifies the probability distribution for that variable (*A*) given the values of its parents (values of the set of nodes with arcs going to *A*, denoted by *Pa*(*A*)). The probability distribution specified by a Bayes's net over variables $X_1,\ldots,X_p$ is defined as

$$P\left(X_1 = x_1 \ldots, X_p = x_p\right) = \prod_i P\left(X_1 = x_i \middle| Pa\left(X_i\right)\right)$$

Friedman and Halpern (1999) were the first to use this technique in the area of microarray expression data. Using the same *S. cerevisiae* data as were used by Eisen et al. for clustering, Friedman et al. show that using statistical methods, a Bayes's network representing the observed relationships between the expression levels of different genes can be learned automatically from the expression levels of the genes across a variety of experiments (table 7).

The application of learning Bayes's nets to gene-expression microarray data is receiving a great deal of attention because the resulting

Bayes's nets potentially provide insight into the interaction networks within cells that regulate the expression of genes. Others have since developed other algorithms to construct Bayes's network models from data and have also had substantial success.

One might interpret the graph in figure 7 to mean that gene *A* causes gene *B* and gene *C* to be expressed, in turn influencing gene *D*. However, caution must be exercised in interpreting arcs as specifying causality in such automatically constructed models. The presence of an arc merely represents correlation—that one variable is a good predictor of another. This correlation can arise because the parent node influences the behavior of the child node, but it can also arise because of a reverse influence or an indirect chain of influence involving other features.

One method for addressing causality in Bayes's net learning is to use genetic mutants, in which some gene is "knocked out." Pe'er et al. (2001) use this approach to model expression in *S. cerevisiae* (that is, bakers' yeast). For 300 of the genes in *S. cerevisiae,* biologists have created a *knockout mutant*, or a genetic mutant lacking that gene. If the parent of a gene in the Bayes's net is knocked out, and the child's status remains unchanged, then it is unlikely that the arc from parent to child captures causality. A current limitation of this approach is that no other organism has such an extensive set of knockout mutants.

Another method for addressing the issue of causality, explored by Ong, Glasner, and Page (2002), is through the use of time-series data. *Time-series data* are simply data from the same organism at various time points. Ong et al. use time-series data from the tryptophan regulon of *E. coli* (Khodursky et al. 2000). A *regulon* is a set of genes that are coregulated. The *tryptophan regulon* regulates the metabolism of the amino acid tryptophan in the cell. Ong et al. use these data to infer a temporal direction for gene interactions, thereby suggesting possible causal relations. To model this temporal directionality, they use a representation known as a dynamic Bayesian network. In a *dynamic Bayesian network*, genes are each represented, not by only one node, but by T nodes, where T is the number of time points. Each of these T nodes represents the gene's expression level at a different time point. This way, the algorithm can learn relationships between genes at time t and at time $t + 1$. It is also possible for the network to identify *feedback loops,* cases where a gene either directly or through some chain of influence actually influences its own regulation. Feedback loops are common in gene regulation.

Given	A set of genes in an organism represented similarly to table 1a. Each of these genes is an example. Each example's numeric expression levels under various experimental circumstances (environmental stresses, developmental stage, and so on) are its features.
Do	Learn a Bayesian network that captures the joint probability distribution over the expression levels of these genes.

Table 7. Learning Bayes's Networks.

Using Additional Sources of Data A recent trend in computational biology is to use more than just microarray data as the source of input to a learning algorithm. In this subsection, we briefly describe a few such investigations.

Some recent approaches to clustering genes rely not only on the expression data but also on background knowledge about the problem domain. Hanisch et al. (2002) present one such approach. They add a term to their distance metric that represents the distance between two genes in a known biological-reaction network. A *biological-reaction network* is a set of proteins, various intermediates, and reactions among them; together these chemicals carry out some cooperative function, such as cell respiration or metabolism. They can function like assembly lines where one protein turns chemical X into chemical Y by adding or removing atoms or changing its conformation; the next protein turns chemical Y into chemical Z in a similar fashion, and so on. One often depicts the entities in such biological networks as edges in a graph and the reactions among them as vertexes. Biologists have discovered many of these networks through other experimental means, and some of these networks are now well understood. Genes that are nearer to one another in such a biological network can be considered, for the purposes of clustering, more similar than genes that are farther apart.

The BioLingua system of Shrager, Langley, and Pohorille (2002) also uses a network graph describing a known biological pathway and updates it using the results of microarray experiments. Their algorithm adds and removes links in the biological pathway based on each link's experimental support in the microarray data, which is a form of theory revision, a small subtopic within machine learning (see chapter 12 of Mitchell [1997]). The network structures in BioLingua are similar to a dynamic Bayes's network in that the links imply causality—not just correlation—between the expression of one particular gene and another. Shrager et al. achieve this perspective through a combination of domain knowledge and their use of time-series data; if there is a causal connection

between two events, they require that it can only go in the forward temporal direction. One way that their representation differs from Bayesian approaches is that BIOLINGUA's links are *qualitative* rather than quantitative. Instead of a joint statistical distribution on probabilities between linked nodes, their algorithm uses a qualitative representation that simply specifies influences as either positive or negative. Along with the causal links, their representation mirrors the type of network description that biologists are familiar with, thereby making the resulting model more useful.

Another source of data is the DNA sequence itself. Over 60 organisms, including *E. coli,* fruit fly, yeast, mouse, and humans, have already been (nearly) completely sequenced; in other words, the sequence of the entire string of the millions to billions of bases constituting their genomes is known. Many others, although not complete, are in progress and have large amounts of data available. The DNA sequence surrounding a gene can have an impact on its regulation and, through this regulation, its function. Craven et al. (2000) use machine learning to integrate *E. coli* DNA sequence data, including geometric properties such as the spacing between adjacent genes and the predicted DNA binding sites of important regulatory proteins, with microarray expression data to predict operons. An *operon* is a set of genes that are transcribed together. Operons provide important clues to gene function because functionally related genes often appear together in the same operon.

DNA sequence information is also used in a method that Segal et al. (2001) developed. Their goal is to jointly model both gene-expression data and transcription factor binding sites. *Transcription factors* are proteins that bind to a subsequence of the DNA before a gene and encourage the start of transcription. The subsequence to which a transcription factor binds is called the *transcription factor binding site.* If two genes have similar expression profiles, it is likely that they are controlled by the same transcription factor and therefore have similar transcription factor binding sites in the sequence preceding them. To model both gene-expression information and sequence information jointly, Segal et al. use what are known as probabilistic relational models (PRMs). A PRM can be thought of as a Bayesian network whose variables are fields in a relational database. The strength of this representation is that PRMs can be learned from a relational database with multiple relational tables, whereas learning algorithms for ordinary Bayes's nets require the data to be in a single table. The different tables can be used to represent different types of data, for example, sequence data and expression data. The approach of Segal et al. uses an expectation-maximization algorithm to learn a PRM that models both clusters of genes and, for each such cluster, the likely transcription factor binding sites in front of those genes in the DNA.

Another excellent source of supplementary material is the large amount of human-produced text about the genes on a microarray (and their associated proteins) that is contained in biomedical digital libraries and the expert-produced annotations in biomedical databases. Molla et al. (2002) investigate using the text in the curated SWISSPROT protein database (Bairoch and Apweiler 2000) as the features characterizing each gene on an *E. coli* microarray. Using these text-based features, they utilize a machine-learning algorithm to produce rules that "explain" which genes' expression levels increase when *E. coli* is treated with an antibiotic.

There is a wealth of data—known reaction pathways, DNA sequences, genomic structure, information gleaned from protein-DNA and protein-protein binding experiments, carefully annotated databases and the scientific literature, and so on—that one can use to supplement table 1's meager representation of microarray experimental data. Exploiting such richness offers an exciting opportunity for machine learning.

Machine Learning in Medical Applications of Microarrays

Having seen how both supervised and unsupervised learning methods have proven useful in the interpretation of microarray data in the context of basic molecular biology, we next turn to the application of microarrays in medicine. Microarrays are improving the diagnosis of disease, facilitating more accurate prognosis for particular patients, and guiding our understanding of the response of a disease to drugs in ways that have already improved the process of drug design. It is quite possible that these technologies could someday even lead to medicines personalized at the genetic level (Mancinelli, Cronin, and Sadee 2000). In this section, we attempt to provide a sense of the large number of future opportunities for machine learning as the medical applications of microarray technology expand.

Disease Diagnosis

A common issue in medicine is to distinguish accurately between similar diseases to make an

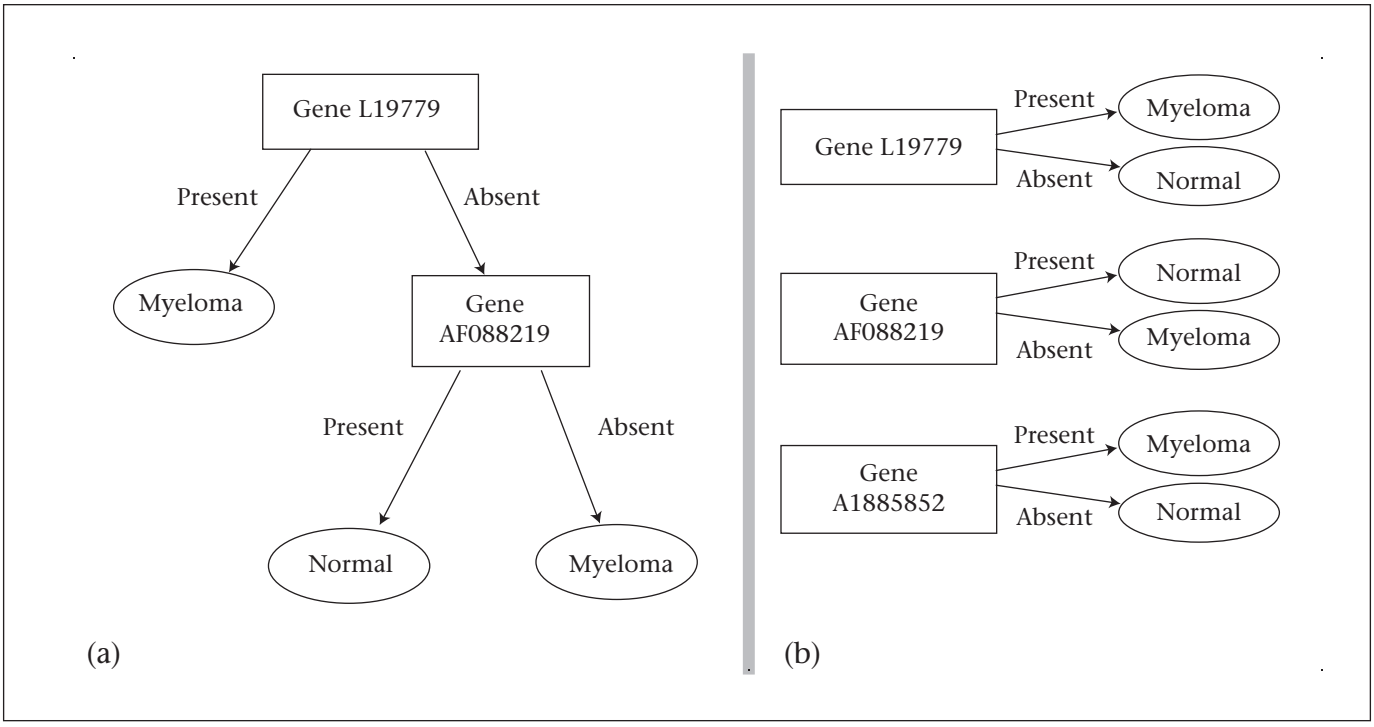


Figure 8. Comprehensible Models for Disease Diagnosis.

A. A two-level decision tree for discriminating between Myeloma cells and normal cells, based on the gene-expression levels from those cells. B. An ensemble of voting decision *stumps* (one-level decision trees) for the same task. In the case of unweighted voting, each decision stump is given a single vote, and a simple majority vote is taken to distinguish Myeloma cells from normal cells. In the case of weighted voting, some decision stumps have their votes counted more than others. One can choose from a variety of methods to determine how to weight the votes.

accurate diagnosis of a patient. Molecular-level classification using gene microarrays has already proven useful for this task. This technique has been used in two tasks that we discuss in the context of cancer diagnosis: (1) class discovery and (2) class prediction. *Class discovery* (table 8) is the task of identifying new classes of cancer; *class prediction* (table 9) is the task of assigning a new tumor to a known class. Accurate diagnosis is crucial for obtaining an accurate prognosis as well as assigning appropriate treatment for the disease.

Golub et al. (1999) use microarray technology for class discovery and class prediction on two types of closely related cancers: (1) acute lymphoblastic leukemia (ALL) and (2) acute myeloid leukemia (AML). The distinction between these two cancers has long been well established, but no single test is sufficient to accurately diagnose between them. Current medical practice is to use a series of separate, highly specialized tests. When combined, the results of these tests are fairly accurate, but misdiagnoses do occur.

The Golub group uses microarrays to address this diagnostic issue by analyzing samples from patients' tumors. Until this time, microarrays had been used primarily on highly purified cell lines grown in laboratories. When using microarrays to analyze samples taken directly from patients, the "noise" because of the genetic variation between the patients can obscure the results. For this reason, when working with samples from patients, it is very important to have a large number of patients from which to sample so that the genetic variation unrelated to the disease does not obscure the results.

One can use any of the many supervised learning techniques to induce a diagnosis model from the gene-expression data of a number of patients and the associated disease. Once an accurate predictive model is obtained, new patients—and those who were previously undiagnosable—can be classified. Using an ensemble of 50 weighted voters (figure 8b) on this AML/ALL diagnosis task, Golub et al. were able to correctly classify 29 of the 34 samples in their test set. Their ensemble rejects the other five samples in the test set as "too close to call."

This same type of gene microarray data can also be used in a class-discovery task. Commonly, one discovers classes by using an unsupervised learning technique to cluster the examples. One then matches the clusters produced with known disease types and con-

Given	A set of microarray experiments, each done with cells from a different patient. These data are represented similarly to table 1d. The patients have a group of closely related diseases. Each patient's numeric expression levels from the microarray experiment constitute the features of an example. The corresponding disease classification for each patient is that patient's category.
Do	Using clustering (ignoring the disease category), find those cells that do not fit well in their current disease classification. Assume these cells belong to new disease classifications.

Table 8. Discovering New Disease Classes.

Given	The same data as in table 8.
Do	Learn a model that can accurately classify a new cell into its appropriate disease classification.

Table 9. Predicting Existing Disease Classes.

siders any remaining clusters as new, unstudied disease classes. The primary challenge in class discovery is ensuring that the clustering is biologically meaningful. Because unsupervised learning is done without considering the current disease classification of the example, it is very possible that the clustering will be based on the wrong variations among patients. For example, when performing unsupervised learning on a group of patients with similar cancers, obtaining a clustering based on the ethnicity of the patients could result. Although this grouping might be optimal according to the algorithm used, it offers no insight into the diseases being studied. A second important challenge when doing unsupervised learning, which can also significantly affect the usefulness of the results obtained, is the granularity at which the examples are clustered. Because one can find an optimal clustering for any specified number of clusters, it is important to find a clustering that accurately captures the level of differentiation sought—in this case, the distinction among diseases.

Whenever gene-microarray technology is used on patient samples, instead of on highly purified laboratory samples, one must exercise caution to validate the results obtained to ensure that the genes chosen as predictors are biologically relevant to the process being studied, which is especially relevant in solid-tumor analysis. Because of the method in which they are obtained, tumor-biopsy specimens can have large variations in the amount of the surrounding connective tissue that is obtained along with the tumor cells.[7] Applying class discoveries or predictions made on the data from these cells, without first analyzing the learned predictive model, can result in making decisions using the wrong basis—such as the skill of the person who performed the biopsy—instead of the desired basis—the underlying tumor biology. For this reason, those learning techniques that create directly comprehensible models (such as decision trees, figure 8a; ensembles of voters, figure 8b; and Bayesian networks) are, in these types of applications, preferred to those whose induced models cannot be as easily comprehended by humans (such as neural networks and support vector machines).

Although primarily used for diagnosis, molecular-level classification is not limited simply to distinguishing among diseases. The methods of class prediction and class discovery can also be used to predict a tumor's site of origin, stage, or grade.

Disease Prognosis

As we saw when discussing molecular-level classification, one can use supervised learning to more accurately diagnose a patient who might have one of a set of similar diseases. These same types of techniques can also be used to predict the future course and outcome, or *prognosis,* of a disease. Making an accurate prognosis can be a complicated task for physicians because it depends on a very large number of factors, some of which might not be known by the physician at the time of diagnosis. By more accurately diagnosing the disorder and, as we see later, predicting the response that the disorder will have to particular drugs, we can make a more accurate prognosis for a patient.

Microarray analysis is already being used to predict the prognosis of patients with certain types of cancer. Investigators have chosen to study cancer as a model disease using gene- expression microarrays for a variety of reasons. First, the prognosis for a patient with cancer is highly dependent on whether the cancer has metastasized. Second, it has been shown that important components of the biology of a malignant cell are inherited from the type of cell that initially gave rise to the cancer and the life-cycle stage at which that cell was in during at the time of its transformation; figure 9 illustrates this process. Finally, providing an accurate prognosis to a patient is crucial in deciding how aggressive a treatment should be used. Because of these reasons, researchers typically utilize supervised learning techniques to address this problem (table 10).

One group to use this supervised learning approach for prognosis prediction is Van't Veer et al. (2002). They utilize an ensemble of voters

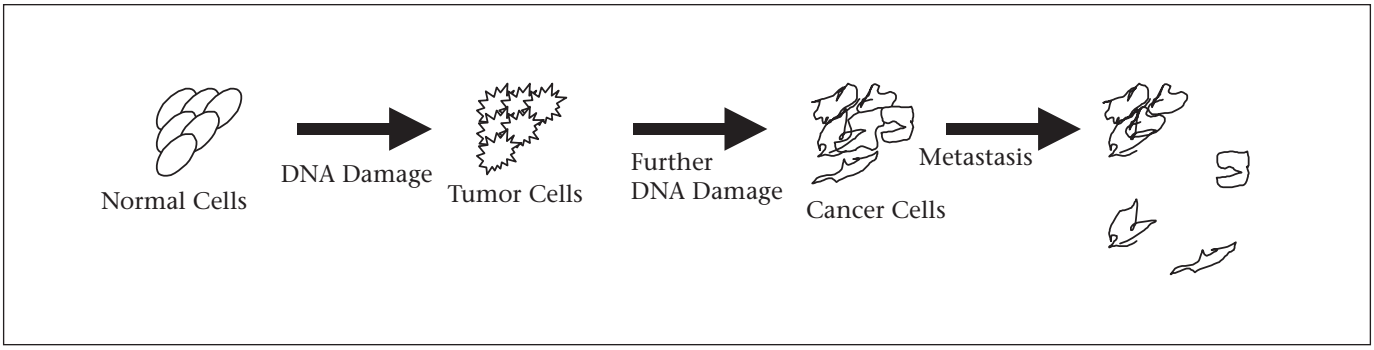


Figure 9. Transformation: The Development of Cancerous Cells from Normal Cells.

In the first step of this transformation, DNA damage causes normal cells to keep multiplying uncontrollably, forming a benign tumor. If further DNA damage occurs, these cells convert from benign to cancerous. The final stage of this progression is the cells' *metastasis*, which is the process whereby the cancer gains the ability to spread to other locations within the body.

to classify breast cancer patients into two groups: (1) good prognosis (no metastasis within five years after initial diagnosis) and (2) poor prognosis (distant metastases found within five years). To begin, they select those 231 genes from the 25,000 genes on the microarray with the highest degree of association with the disease outcome (calculated by correlation coefficient over the full set of 78 examples). They then rank these genes by their correlation coefficients. They repeat "leave-one-out" cross-validation over all 78 examples using various ensemble sizes. They found that an ensemble size of 70 genes gives the best cross-validated accuracy (83 percent).

Their methodology contains two errors from the perspective of current machine learning practice. First, they chose the 231 features using the entire set of 78 examples, which constitutes "information leakage" because all 78 of the examples—including those that will later appear in test sets during the cross-validation—are used to guide the selection of these 231 features. Second, they report the best ensemble size by seeing which size works best in a cross-validation experiment. This again constitutes information leakage because they optimized one of the parameters of the learning system—namely, the size of the ensemble—using examples that will appear in the test sets. These two errors mean that their estimated accuracy is likely to be an overestimation because they "overfit" their test data. A better methodology is to separately select parameters for each fold during their *N*-fold cross-validation experiments. Recognizing these issues after publication, Van't Veer et al. reported a modified version of their algorithm in the online supplement to their article to address these two concerns; their changes reduced the cross-validated accuracy from 83 percent to 73 percent (and one might still question whether their revised approach leads to an overestimate of future accuracy).

Given	A set of microarray experiments, each done with cells from a different patient. These data are represented similarly to table 1d. All these patients have the same type of cancer but are in different stages of progression. Each patient is an example, and the numeric expression levels for all the genes on the microarray are the features. The true prognosis of that patient is that patient's category. (*Note:* Because the true prognosis of a patient might not be known for years, collecting labeled examples can be a challenging task. The fact that the gene-expression measurement technology is rapidly changing also complicates the creation of good training sets for prognosis tasks.) Possible categories include whether or not a cancer is likely to metastasize and what the prognosis of that patient is (for example, will the patient survive for at least five years). One could also formulate this as a real-valued prediction task, such as years until recurrence of the cancer.
Do	Learn a model that accurately predicts which category new patients belong to.

Table 10. Predicting the Prognosis for Cancer Patients.

Although prognosis prediction is commonly thought of as a supervised learning task, valuable information about a disease can also be gained through unsupervised learning. Alizadeh et al. (2000) utilized unsupervised learning techniques to cluster patients with diffuse large B-cell lymphoma into two clusters. They discovered that the average 5-year survival for the patients in 1 cluster was 76 percent compared to 16 percent in the other cluster (average 5-year survival for all patients was 52 percent). These results illustrate that the clusters found through unsupervised learning can be biologically and medically relevant. However, before (solely) employing clustering algorithms, users of machine learning should consider whether their task can be cast in the form

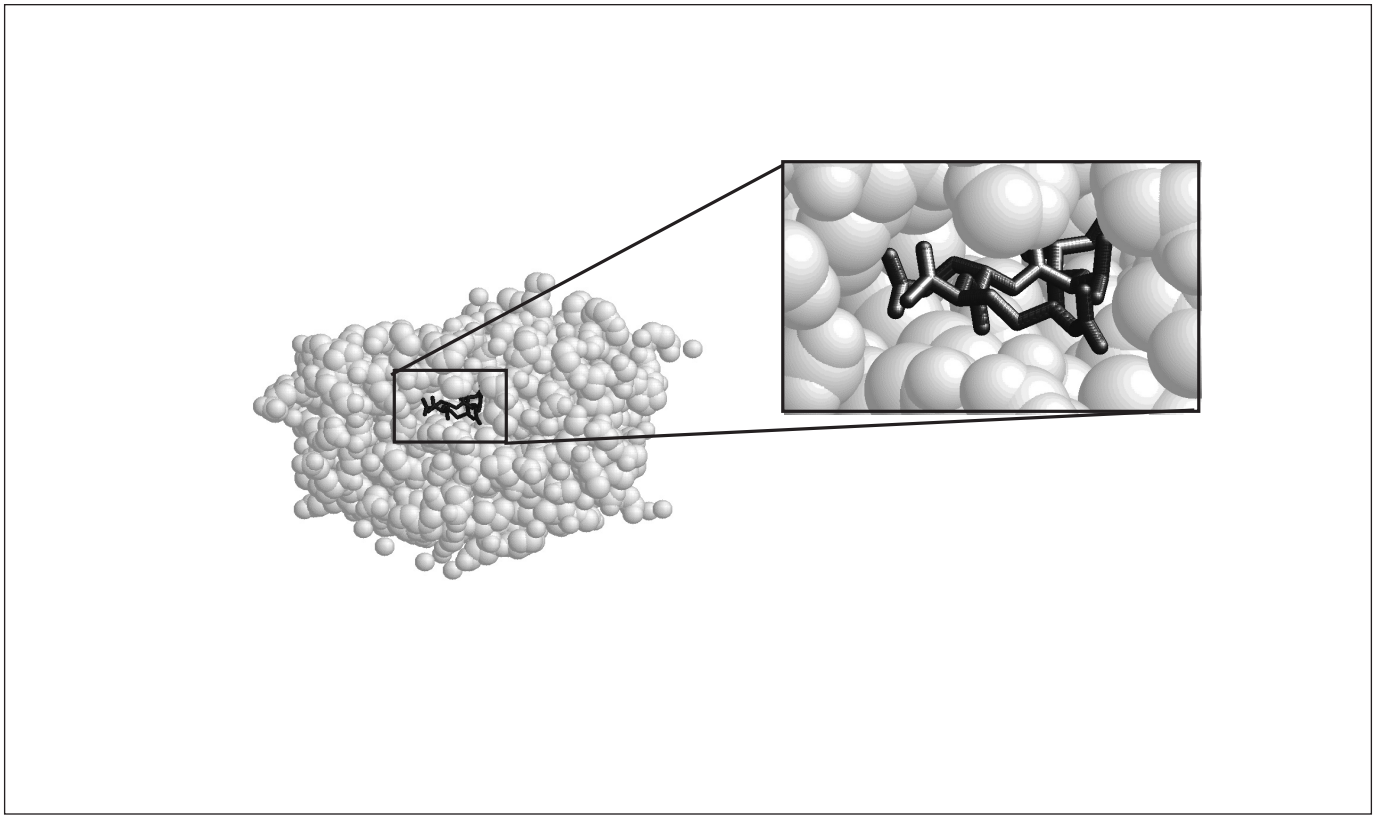


Figure 10. A Drug Binding to a Protein.

Inhibitor Drug U-89360E (shown as a stick model in dark gray) bound to protein HIV-1 Protease mutant G48H (shown as a space-filling model in lighter gray). We generated this image from data publicly available at the Protein Data Bank (www.rcsb.org/pdb/).

of the more directed supervised learning, where training examples are labeled with respect to an important property of interest.

Response to Drugs Drugs are typically small molecules that bind to a particular protein in the body and act to inhibit or activate its activity; figure 10 contains an example. Currently, pharmaceutical companies are limited to designing drugs that have a high level of success and a low level of side effects when given to the "average" person. However, the way that an individual responds to a particular drug is very complex and is influenced by his/her unique genetic makeup, as figure 11 summarizes. Thus, there are millions of cases annually of adverse reactions to drugs and far more cases where drugs are ineffective.[8] The field of pharmacogenomics addresses this tight interrelation between an individual's genetic makeup and his/her response to a particular drug—see table 11 to see how microarrays can play a role.

An area related to pharmacogenomics is molecular-level profiling. The main difference between these two fields is that although pharmacogenomics deals with finding genetic variations among individual people that predict an individual person's response to a particular drug, the goal of molecular-level profiling is to find genetic variations among individual diseased cells that predict that cell's response to a particular drug. Analyzing specific cells is important for predicting drug response because—as the result of the highly variable nature of cancer—significant variation exists among tumors of the same type of cancer, just as significant variation exists between organisms of the same species.

Molecular-level profiling has been found to be effective in treating certain types of cancers. A recent example is Rosenwald et al.'s (2002) lymphoma/leukemia project. This study investigated large-B-cell lymphoma, a type of cancer curable by chemotherapy in only 35 to 40 percent of patients. It is thought that large-B-cell lymphoma is not a single disease but actually a class that contains several different diseases that, although morphologically the same, differ in response to certain types of therapy.

By analyzing gene-expression profiles of cells from different large–B-cell lymphoma tumors, Rosenwald et al. developed a method to

predict the survival rates of diffuse large–B-cell lymphoma based on this microarray data. Using training data from 160 patents whose outcomes on anthracycline-based chemotherapy are known, they predicted which of 80 held-out test-set patients would respond well to this type of chemotherapy. The actual five-year survival rate among those who were predicted to respond was 60 percent. Those who were predicted not to respond had an actual 5-year survival rate of only 39 percent.

Currently, this investigation into large–B-cell lymphoma has yielded prognosis information only. However, this type of insight into how the genetic variations between cells can affect their response to particular drugs will eventually suggest new drugs to treat the types of cells that currently do not respond to chemotherapy and can also lead to the deeper understanding of a disease's mechanism.

As we gain a deeper insight into the diseases that we study, the lines among molecular-level classification, pharmacogenomics, and molecular-level profiling will blur. More accurate subtyping of a single disease can ultimately lead to it being considered as two separate diseases. A deeper understanding of the underlying mechanisms of diseases can lead to the discovery that two previously distinct diseases are different manifestations of the same underlying disease. Personalized medicine could eventually lead not just to classifying patients based on the drug that will work best for them but to designing a drug specifically tailored to a patient's exact disorder and genetic makeup.

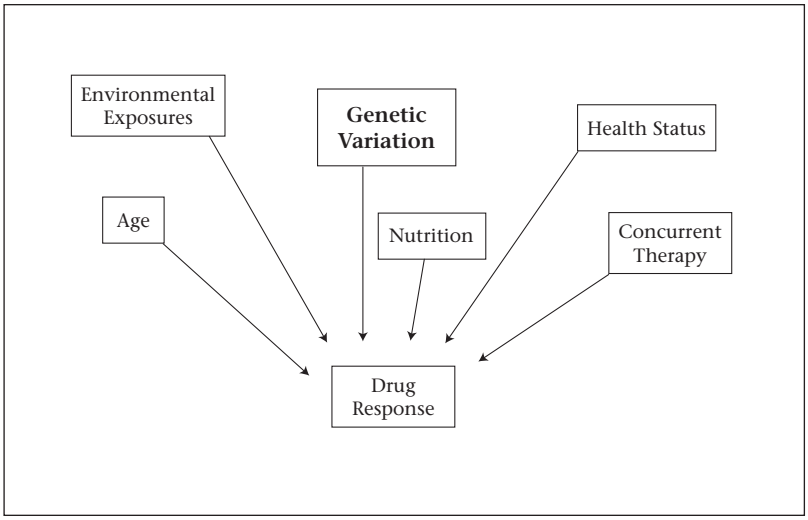


Figure 11. The Major Factors That Affect a Person's Response to a Drug.

Given	A set of microarray experiments, each done with cells from a patient infected with a given disease. These data are represented similarly to table 1d. Each microarray experiment is an example, with each gene's numeric expression level during that experiment serving as a feature. (One might want to augment the gene-expression features with additional features such as the age, gender, and race of each patient.) The drug-response classification of each patient is that example's category. Typical categories are good response (that is, improved health), bad response (that is, bad side-effects), and no response.
Do	Build a model that accurately predicts the drug response of new patients.

Table 11. Predicting the Drug Response of Different Patients with a Given Disease.

New Data Types from High-Throughput Biotechnology Tools

In this section, we briefly discuss three other novel types of high-throughput, molecular-level biological data to which machine learning is applicable. (*High-throughput techniques* are those that permit scientists to make thousands of measurements from a biological sample in about the time and effort it traditionally took to make at most a handful of measurements.) Data sets arising from these additional techniques are similar to gene microarrays in that they have a similar tabular representation and high dimensionality.

Single-Nucleotide Polymorphisms (SNPs)

Genome researchers have learned that much of the variation between individuals is the result of a number of discrete, single-base changes in the human genome. Since that discovery, there has been intense effort to catalog as many of these discrete genetic differences as possible. These single positions of variation in DNA are called *single-nucleotide polymorphisms* (SNPs) and are illustrated in figure 12. Although it is currently infeasible to obtain the sequence of all the DNA of a patient, it is feasible to quickly measure that patient's *SNP pattern,* the particular DNA bases at a large number of these SNP positions.

Machine learning can be applied to SNP data in a manner similar to its application to microarray data. For example, given an SNP data file as in table 12, one can utilize supervised learning to identify differences in SNP patterns between people who respond well to a particular drug versus those who respond poorly. If the data points are classified instead by diseased versus healthy, one can use supervised learning to identify SNP patterns predictive of disease. If the highly predictive SNPs appear

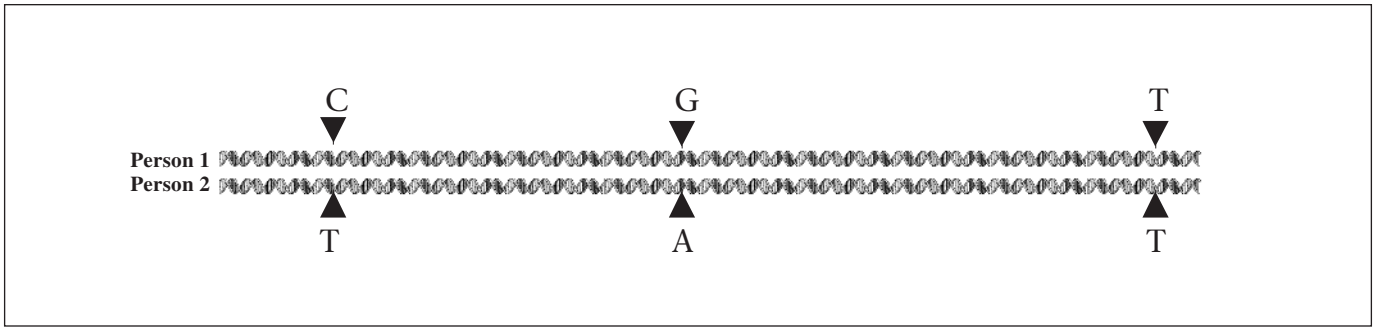


Figure 12. Single-Nucleotide Polymorphism.

The differences between the genomes of two individuals are generally discrete, single-base changes. Shown is a simplified example of what the corresponding genomes of two people might look like. The differences are highlighted—all other DNA bases are identical between the two sequences.

	SNP 1	SNP 2	…	SNP *M*	Response
Person 1	C T	A G	…	T T	Positive
Person 2	C C	A A	…	C T	Negative
…	…	…	…	…	…
Person *N*	T T	A G	…	C C	Positive

Table 12. A Sample Single-Nucleotide Polymorphism Data File.

Because humans have paired chromosomes, one needs to record the base on each chromosome at an SNP position (notice that each of the two chromosomes contains a pair of DNA strands—the famous double-helix—but because of the complementarity of these paired strands, there is no need to record all four bases at a given SNP position). Although biologists have already identified over a million SNP positions in the human genome, currently, a typical SNP data file will contain only thousands of SNPs because of the cost of data gathering.

within genes, these genes can be important for conferring disease resistance or susceptibility, or the proteins they encode can be potential drug targets.

One challenge of SNP data is that it is collected in unphased form. For example, suppose that instead of coming from two different people, the two DNA strands in figure 12 refer to the two copies of chromosome 1 in a single person (humans haves two copies of each chromosome). Current SNP technology would return the first row of table 12; it would not provide any information about which SNP variants are on which chromosome. Should this "phase" information be necessary for the particular prediction task, the machine learning algorithm will be unsuccessful.

Proteomics

Gene microarrays measure the degree to which every gene is being transcribed. This measure is a useful surrogate for gene expression (that is, the complete process of transcription followed by translation), particularly because protein levels are more difficult to measure than RNA levels. Nevertheless, increased transcription does not always mean increased protein production. Therefore, it is desirable to instead measure protein directly; this process is called *proteomics* in contrast to *genomics,* which is the rubric under which gene microarrays fall. An organism's *proteome* is its full complement of proteins.

Mass spectrometry makes it possible to detect the presence of various proteins in a sample. The details of mass spectrometry are beyond the scope of this article; however, figure 13 provides a sense of this type of data. To convert such an example into a feature vector, it is common to perform some type of "peak picking." The result of picking peaks in mass-spectrometry data is a feature vector of *x*-*y* pairs, where each entry corresponds to a mass-to-charge ratio (the *x* axis) and the associated peak height (the *y* axis).

Mass-spectrometry data present at least three major challenges. First, in raw form, the peaks typically correspond to pieces of proteins—*peptides*—rather than entire proteins. One can either work with these features or preprocess the data by attempting to map from a set of peaks to a (smaller) set of proteins. Second, currently, mass spectrometry is extremely poor at giving quantitative values; peak heights are not calibrated from one sample to another. Hence, although the normalized peak height at a particular mass-to-charge ratio can be much greater in example 1 than example 2, the amount of protein at that ratio actually might be greater in example 2. Therefore, often

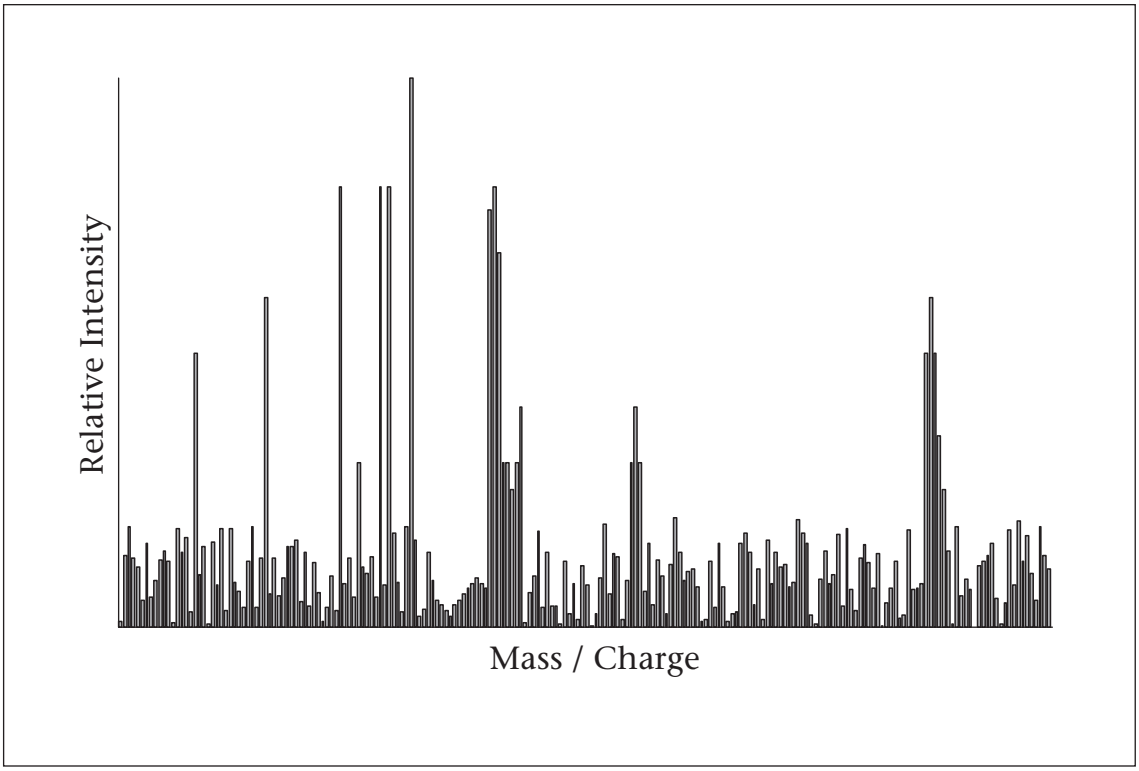


Figure 13. Sample Mass-Spectrometry Output.

Different protein fragments appear at different mass-charge values on the horizontal axis. The vertical axis reflects the amount of the protein fragment in the sample. The plotted peak heights are typically normalized relative to the highest intensity.

it is desirable to use binary features instead of continuous ones—at a particular mass-to-charge ratio, either there is a peak, or there is not. The third major challenge of mass spectrometry data is that peaks from lower-concentration proteins cannot be distinguished from the background noise.

Although this discussion has focused on mass-spectrometry data because of its similarities to gene-microarray data, the phrase *proteomics* actually refers to a broader range of data types. Most significantly, it also includes data on protein-protein interactions. Such data also pose interesting opportunities and challenges for machine learning. KDD CUP 2001 (Cheng et al. 2002) contained one challenging task involving protein-protein interaction data.

Metabolomics

It is tempting to believe that with data about DNA (SNPs), RNA (microarrays), and proteins (mass spectrometry), one has access to all the important aspects of cell behavior. However, in fact, many other aspects remain unmeasured with these high-throughput techniques. These aspects include posttranslational modifications to proteins (for example, phosphorylation), cell structure, and signaling among cells. For most such aspects, there currently exist no high-throughput measurement techniques. Nevertheless, some insight into these other aspects of cell behavior can be obtained by examining the various small molecules (that is, those with low molecular weight) in the cell. Such molecules are often important input and output of metabolic pathways in the cell. High-throughput techniques for measuring these molecules exist. The area of studying data on these molecules is called *metabolomics* (Oliver et al. 1998). High-throughput metabolomics data can be represented naturally in feature vectors in a manner similar to gene-microarray data and mass-spectrometry data. In metabolomics data, the features correspond to small molecules, and each feature takes a value that expresses the quantity of that molecule in a given type of cell.

Systems Biology

Additional forms of high-throughput biological data are likely to become available in the future. Much of the motivation for these developments is a shift within biology toward a systems approach, commonly referred to as

systems biology. As Hood and Galas (2003, p. 447) note, whereas in the past biologists could study a "complex system only one gene or one protein at a time," the "systems approach permits the study of all elements in a system in response to genetic (digital) or environmental perturbations." They go on to state,

> The study of cellular and organismal biology using the systems approach is at its very beginning. It will require integrated teams of scientists from across disciplines—biologists, chemists, computer scientists, engineers, mathematicians and physicists. New methods for acquiring and analyzing high-throughput biological data are needed (Hood and Galas 2003, p. 448).

Constructing models of biological pathways or even an entire cell—an *in silico cell*—is a goal of systems biology. Perhaps the preeminent example to date of the systems approach is a gene-regulatory model that Davidson et al. (2002) developed for embryonic development in the sea urchin. Nevertheless, this model was developed over years using data collected without the benefit of high-throughput techniques. Machine learning has the potential to be a major player in systems biology because learning algorithms can be used to construct or modify models based on the vast amounts of data generated by high-throughput techniques.

Conclusion

Machine learning has much to offer to the revolutionary new technology of gene microarrays. From microarray design itself to basic biology to medicine, researchers have utilized machine learning to make gene chips more practical and useful.

Gene chips have already changed the field of biology. Data that might have taken years to collect now take a week. Biologists are aided greatly by the supervised and unsupervised learning methods that many are using to make sense of the large amount of data now available to them, and additional challenging learning tasks will continue to arise as the field further matures. As a result, we have seen a rapid increase in the rate at which biologists are able to understand the molecular processes that underlie and govern the function of biological systems.

Although their impact will progress more slowly in medicine than in molecular biology, microarray technology, coupled with machine learning, is also being used for a variety of important medical applications: diagnosis, prognosis, and drug response. These applications are similar in that they all deal with predicting some aspect of a disease by differentiating at the molecular level among individuals in a population—either patients or cells. The difference among these applications concerns what is being predicted. In disease classification, one focuses on distinguishing among cells with different, but possibly related, diseases. In disease prognosis, one is predicting long-range results. In pharmacogenomics and molecular profiling, one uses molecular-level measurements to differentiate among patients or cells with the same disease based on their reaction to particular drugs.

As our vast amount of genomic and similar types of data continues to grow, the role of computational techniques, especially machine learning, will grow with it. These algorithms will enable us to handle the task of analyzing these data to yield valuable insight into the biological systems that surround us and the diseases that affect us.

Acknowledgments

The writing of this article was partially supported by grants from the National Institutes of Health, 2 R44 HG02193-02, 2 P30 CA14520-29, and 5 T32 GM08349; the National Library of Medicine, 1T15LM007359-01 and 1 R01 LM07050-01; and the National Science Foundation, 9987841.

Notes

1. While this article was in production, an especially relevant special issue of the journal *Machine Learning* was published ("Machine Learning in the Genomics Era," Volume 52, Numbers 1–2, 2003). Interested readers can find additional concrete examples in that special issue's eight articles.

2. Some organisms produce what are known as *small RNA* (sRNA) molecules. This RNA that is not translated into protein. These RNA molecules play roles in the cell different from what is shown in figure 1. Rather, the RNA molecule itself takes on a shape suitable to perform some function in the cell.

3. This is not strictly true. Because of a process in higher organisms called *alternate splicing*, a single gene can encode multiple proteins. However, for the purposes of gene detection by microarrays, each part of such genes (called *exons*) can be detected separately. We do not discuss the detection of splice variants in this article.

4. Although it is often useful to think of DNA and RNA as chains of bases, technically, they are chains of sugars. In the case of DNA, the sugar is deoxyribose, and in the case of RNA, it is ribose; hence, the full names: deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). The bases are actually attached to the sugars.

5. One could also make probes from RNA, but they tend to degrade much faster.

6. The word *chip* might be confusing to those famil-

iar with integrated circuits. Microarrays can be about the size of a computer chip, and some approaches for creating them do use the masking technology used for etching integrated circuits. However, a single gene chip is typically only used once, unlike a computer chip. It might be better to conceptually view a gene chip as holding thousands of miniature test tubes. (One should also not confuse gene chips with DNA computing, where one uses DNA to solve computational tasks such as the traveling salesman problem. In this article, we address using computer science to solve biomedical tasks, rather than using molecular biology processes to solve computational tasks.)

7. This problem is specific to the collection of specimens from solid tumors and is not the case when dealing with cancers of the blood. For this reason, higher accuracies are generally found when using machine learning on cancers of the blood than on solid tumor cancers.

8. In 1994, there were over 2.2 million serious cases of adverse drug reactions and over 100,000 deaths in the United States (Lazarou, Pomeranz, and Corey 1998).

References

Alizadeh, A.; Eisen, M.; Davis, R.; Ma, C.; Lossos, I.; Rosenwald, A.; Boldrick, J.; Hajeer, S.; Tran, T.; Yu, X.; Powell, J.; Yang, L.; Marti, G.; Moore, T.; Hudson, J., Jr; Lu, L.; Lewis, D.; Tibshirani, R.; Sherlock, G.; Chan, W.; Greiner, T.; Weisenburger, D.; Armitage, J.; Warnke, R.; Levy, R.; Wyndham Wilson, W.; Grever, M.; Byrd, J.; Botstein, D.; Brown, P.; and Staudt, L. 2000. Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene-Expression Profiling. *Nature* 403(6769): 503–511.

Bairoch, A., and Apweiler, R. 2000. The SWISS-PROT Protein Sequence Database and Its Supplement TrEMBL in 2000. *Nucleic Acids Research* 28(1): 45–48.

Breslauer, K.; Frank, R.; Blocker, H.; and Marky, L. 1986. Predicting DNA Duplex Stability from the Base Sequence. *Proceedings of the National Academy of Sciences* 83(11): 3746–3750.

Brown, M.; Grundy, W.; Lin, D.; Cristianini, N.; Sugnet, C.; Furey, T.; Ares, M., Jr.; and Haussler, D. 2000. Knowledge-Based Analysis of Microarray Gene-Expression Data by Using Support Vector Machines. *Proceedings of the National Academy of Sciences* 97(1): 262–267.

Cheng, J.; Hatzis, C.; Hayashi, H.; Krogel, M.; Morishita, S.; Page, D.; and Sese, J. 2002. Report on KDD Cup 2001. *SIGKDD Explorations* 3(2): 47–64.

Craven, M.; Page, D.; Shavlik, J.; Bockhorst, J.; and Glasner, J. 2000. Using Multiple Levels of Learning and Diverse Evidence Sources to Uncover Coordinately Controlled Genes. In *Proceedings of the Seventeenth International Conference on Machine Learning,* 199–206. San Francisco, Calif.: Morgan Kaufmann.

Davidson, E.; Rast, J.; Oliveri, P.; Ransik, A.; Calestani, C.; Yuh, C.; Amore, G.; Minokawa, T.; Hynman, V.; Arenas-Mena, C.; Otim, O.; Brown, C.; Livi, C.; Lee, P.; Revilla, R.; Alistair, R.; Pan, Z.; Schilstra, M.; Clarke, P.; Arnone, M.; Rowen, L.; Cameron, R.; McClay, D.; Hood, L.; and Bolouri, H. 2002. A Genomic Regulatory Network for Development. *Science* 295(5560): 1669–1678.

Eisen, M.; Spellman, P.; Brown, P.; and Botstein, D. 1998. Cluster Analysis and Display of Genome-Wide Expression Patterns. *Proceedings of the National Academy of Science* 95(25): 14863–14868.

Friedman, N., and Halpern, J. 1999. Modeling Beliefs in Dynamic Systems. Part II: Revision and Update. *Journal of Artificial Intelligence Research* 10:117–167.

Golub, T.; Slonim, D.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J.; Coller, H.; Loh, M.; Downing, J.; Caligiuri, M.; Bloomfield, C; and Lander, E. 1999. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene- Expression Monitoring. *Science* 286(5439): 531–537.

Hanisch, D.; Zien, A.; Zimmer, R.; and Lengauer, T. 2002. Co-Clustering of Biological Networks and Gene-Expression Data. *Bioinformatics* (Supplement) 18:S145–S1554.

Hood, L., and Galas, D. 2003. The Digital Code of DNA. *Nature* 421(6921): 444–448.

Hughes, T.; Matthew, J.; Marton, M.; Jones, A.; Roberts, C.; Stoughton, R.; Armour, C.; Bennett, H.; Coffey, E.; Dai, H.; He, Y.; Kidd, M.; King, A.; Meyer, M.; Slade, D.; Pek, Y.; Lum, P.; Stepaniants, S.; Shoemaker, D.; Gachotte, D.; Chakraburtty, K.; Simon, J.; Bard, M.; and Friend, S. 2000. Functional Discovery via a Compendium of Expression Profiles. *Cell* 102(1): 109–126.

Khodursky, A.; Peter, B.; Cozzarelli, N.; Botstein, D.; Brown, P.; and Yanofsky, C. 2000. DNA Microarray Analysis of Gene Expression in Response to Physiological and Genetic Changes That Affect Tryptophan in *Escheria Coli*. *Proceedings of the National Academy of Science* 97(22): 12170–12175.

Lazarou, J.; Pomeranz, B.; and Corey, P. 1998. Incidence of Adverse Drug Reactions in Hospitalized Patients. *Journal of the American Medical Association* 279(15): 1200–1205.

Li, C., and Wong, W. 2001. Model-Based Analysis of Oligonucleotide Arrays: Expression Index Computation and Outlier Detection. *Proceedings of the National Academy of Science* 98(1): 31–36.

Mancinelli, L.; Cronin, M.; and Sadee, W. 2000. Pharmacogenomics: The Promise of Personalized Medicine. *AAPS PharmSci* 2(1).

Mitchell, T. 1997. *Machine Learning*. Boston, Mass.: McGraw-Hill.

Molla, M; Andrae, P.; Glasner, J.; Blattner, F.; and Shavlik, J. 2002. Interpreting Microarray Expression Data Using Text Annotating the Genes. *Information Sciences* 146:75–88.

Newton, M.; Kendziorski, C.; Richmond, C.; Blattner, F.; and Tsui, K. 2001. On Differential Variability of Expression Ratios: Improving Statistical Inference about Gene Expression Changes from Microarray Data. *Journal of Computational Biology* 8(1): 37–52.

Nuwaysir, E. F.; Huang, W.; Albert, T.; Singh, J.; Nuwaysir, K.; Pitas, A.; Richmond, T.; Gorski, T.; Berg, J.; Ballin, J.; McCormick, M.; Norton, J.; Pollock, T.; Sumwalt, T.; Butcher, L.; Porter, D.; Molla, M.; Hall, C.; Blattner, F.; Sussman, M.; Wallace, R.; Cerrina, F.;

and Green, R. 2002. Gene- Expression Analysis Using Oligonucleotide Arrays Produced by Maskless Lithography. *Genome Research* 12(11): 1749–1755.

Oliver, S.; Winson, M.; Kell, D.; and Baganz, F. 1998. Systematic Functional Analysis of the Yeast Genome. *Trends in Biotechnology* 16(9): 373–378.

Ong, I.; Glasner, J.; and Page, D. 2002. Modeling Regulatory Pathways in *E.coli* from Time-Series Expression Profiles. *Bioinformatics* (Supplement) 18:S241–S248.

Pe'er, D.; Regev, A.; Elidan, G.; and Friedman, N. 2001. Inferring Subnetworks from Perturbed Expression Profiles. *Bioinformatics* (Supplement) 17:S215–S224.

Rosenwald, A.; Wright, G.; Chan, W.; Connors, J.; Campo, E.; Fisher, R.; Gascoyne, R.; Muller-Hermelink, H.; Smeland, E.; and Staudt, L. 2002. The Use of Molecular Profiling to Predict Survival after Chemotherapy for Diffuse Large-B-Cell Lymphoma. *New England Journal of Medicine* 346(25): 1937–1947.

Segal, E.; Taskar, B.; Gasch, A.; Friedman, N.; and Koller, D. 2001. Rich Probabilistic Models for Gene Expression. *Bioinformatics* 17(1): 1–10.

Shrager, J.; Langley, P.; and Pohorille, A. 2002. Guiding Revision of Regulatory Models with Expression Data. In *Proceedings of the Pacific Symposium on Biocomputing,* 486–497. Lihue, Hawaii: World Scientific.

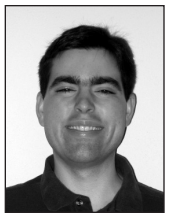
Spellman, P.; Sherlock, G.; Zhang, M.; Iyer, V.; Anders, K.; Eisen, M.; Brown, P.; Botstein, D.; and Futcher, B. 1998. Comprehensive Identification of Cell Cycle–Regulated Genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization. *Molecular Biology of the Cell* 9(12): 3273–3297.

Thomas, R.; Rank, D.; Penn, S.; Zastrow, G.; Hayes, K.; Pande, K.; Glover, E.; Silander, T.; Craven, M.; Reddy, J.; Jovanovich, S.; and Bradfield, C. 2001. Identification of Toxicologically Predictive Gene Sets Using cDNA Microarrays. *Molecular Pharmacology* 60(6): 1189–1194.

Tobler, J.; Molla, M.; Nuwaysir, E.; Green, R.; and Shavlik, J. 2002. Evaluating Machine Learning Approaches for Aiding Probe Selection for Gene-Expression Arrays. *Bioinformatics* (Supplement) 18:S164–S171.

Van't Veer, L.; Dai, H.; van de Vijver, M.; He, Y.; Hart, A.; Mao, M.; Peterse, H.; van der Kooy, K.; Marton, M.; Witteveen, A.; Schreiber, G.; Kerkhoven, R.; Roberts, C.; Linsley, P.; Bernards, R.; and Friend, S. 2002. Gene-Expression Profiling Predicts Clinical Outcome of Breast Cancer. *Nature* 415(6871): 530–536.

Michael Molla has been a Ph.D. student in computer sciences at the University of Wisconsin at Madison since 1999. He also works part time as a paid consultant for Nimblegen Systems, a local microarray manufacturer. He received his BA in computer science from Brandeis University in 1992 and an MEng from the University of Colorado at Boulder. Molla worked as a programmer at IBM from 1993 to 1997. He worked on the Human Genome Project at the MIT/Whitehead Institute Genome Sequencing Center from 1997 to 1999. His research interests include machine learning, computational biology, and bioinformatics. His e-mail address is molla@cs.wisc.edu.



Michael Waddell is a Ph.D. student at the University of Wisconsin at Madison in the Department of Computer Sciences and the Department of Biostatistics and Medical Informatics. He received his BS in mathematics, biochemistry, and molecular biology from the University of Wisconsin at Madison in 2000. His research interests include data mining, expert systems, and human-computer collaboration. His e-mail address is mwaddell@cs.wisc.edu.

David Page received his Ph.D. in computer science from the University of Illinois at Urbana-Champaign in 1993. He was a research scientist in the Oxford University Computing Laboratory from 1993 to 1997, where he also served as a visiting member of the mathematics faculty from 1995 to 1997. Before joining the University of Wisconsin's Department of Biostatistics and Medical Informatics and Department of Computer Sciences in 1999, Page also served as an assistant professor at the University of Louisville, where he was a founding member of the Institute for Molecular Diversity and Drug Design. Page's research interests are in machine learning and data mining, in particular in techniques that work with relational data and in applications of these techniques to biomedical data. His e-mail address is page@biostat.wisc.edu.

Jude Shavlik is a professor of computer sciences and biostatistics and medical informatics at the University of Wisconsin at Madison. He has been at Wisconsin since 1988, following the receipt of his Ph.D. from the University of Illinois for his work on explanation-based learning. His current research interests include machine learning, computational biology, and intrusion detection. For three years, he was editor in chief of *AI Magazine* and currently serves on the editorial boards of several journals. He chaired the 1998 International Conference on Machine Learning, cochaired the First International Conference on Intelligent Systems for Molecular Biology in 1993, cochaired the First International Conference on Knowledge Capture in 2001, and was conference chair of the 2003 IEEE Conference on Data Mining. He coedited, with Tom Dietterich, Readings in Machine Learning (Morgan Kaufmann, 1990). His e-mail address is shavlik@cs.wisc.edu.