Identifying Predictive Features in Drug Response Using Machine Learning: Opportunities and Challenges

Mathukumalli Vidyasagar

Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, Texas 75080; email: M.Vidyasagar@utdallas.edu

Annu. Rev. Pharmacol. Toxicol. 2015. 55:15–34

First published online as a Review in Advance on November 12, 2014

The *Annual Review of Pharmacology and Toxicology* is online at pharmtox.annualreviews.org

This article's doi: 10.1146/annurev-pharmtox-010814-124502

Keywords

cancer biology, machine learning, SVMs, regression, LASSO, EN algorithm, neural networks, PAM, SAM, GSEA, *k*-means clustering, precision medicine, prediction in pharmacology

Abstract

This article reviews several techniques from machine learning that can be used to study the problem of identifying a small number of features, from among tens of thousands of measured features, that can accurately predict a drug response. Prediction problems are divided into two categories: sparse classification and sparse regression. In classification, the clinical parameter to be predicted is binary, whereas in regression, the parameter is a real number. Well-known methods for both classes of problems are briefly discussed. These include the SVM (support vector machine) for classification and various algorithms such as ridge regression, LASSO (least absolute shrinkage and selection operator), and EN (elastic net) for regression. In addition, several well-established methods that do not directly fall into machine learning theory are also reviewed, including neural networks, PAM (pattern analysis for microarrays), SAM (significance analysis for microarrays), GSEA (gene set enrichment analysis), and *k*-means clustering. Several references indicative of the application of these methods to cancer biology are discussed.

INTRODUCTION

Cancer is the second leading cause of death in both advanced countries and emerging economies, accounting for roughly 13% of all deaths. Siegel et al. (1) estimated that there would be 1,665,540 new cases of cancer and 585,720 deaths in the United States during 2014. In the United Kingdom, the latest available figures (for the year 2011) show that there were 331,487 new cases of cancer and 159,178 deaths (2). Worldwide, cancer led to about 8.2 million deaths in 2012, the latest year for which data are available from the World Health Organization (3). One of the biggest challenges faced by cancer researchers is the sheer diversity of the disease. No two manifestations of cancer are exactly alike, even when they appear in the same site. We can paraphrase the opening sentence of Leo Tolstoy's *Anna Karenina* and say, "Normal cells are all alike. Every malignant cell is malignant in its own way." Therefore, the ultimate objective in both cancer research and cancer treatment should be to customize the therapeutic regimen to each individual patient, so-called personalized medicine.

Truly personalized therapy is still some distance away. However, this review argues that it is possible, in a systematic and data-driven fashion, to identify a handful of biomarkers for each form of cancer for which sufficient data are available. These biomarkers would be highly predictive of drug response but may not always have biological significance. By overlaying the data on top of inferred gene (or other) regulatory networks, biomarkers can be not only highly predictive but also biologically meaningful in the context of the network. Such a machine learning–based analysis would lead to verifiable biological and/or clinical hypotheses that could then be tested in the laboratory and/or clinic. The outcomes of the experiments and/or trials would generate fresh data that would in turn be fed back into the machine learning analysis, thus completing a virtuous cycle. Ultimately, the validated biomarkers could be used to separate cancer patients into several groups, and therapy could be tailored to each group. Such an approach is often erroneously referred to as personalized medicine. It would be more accurate to refer to this as patient stratification, which is an important intermediate milestone on the road to truly personalized medicine.

Recent advances in experimental biology make it feasible to generate an enormous amount of high-quality raw data at a relatively affordable cost. The Cancer Genome Atlas (TCGA) project has already come out with large databases, comprising molecular measurements as well as clinical information, for several forms of cancer including ovarian (4), colorectal (5), breast (6), lung (7), and brain (8). In addition to TCGA, there are other large projects under way, including the International Cancer Genomics Consortium (**http://icgc.org/**). The data generated by these large, publicly funded efforts are complemented by smaller proprietary databases created by individual research groups around the world.

The availability of so much high-quality raw data offers a unique opportunity to analyze the data with the objective of identifying the most predictive biomarkers in a given clinical application. In short, the challenge is to turn this raw data into information and then knowledge. Typically, when data are being collected, the researchers measure everything they can, as it is not always possible to ascertain beforehand which measurements will hold the key to predicting patient clinical parameters. The sheer volume of the data would require the use of some automated procedures, which would complement and not supplant biological insights and prior knowledge.

We often hear the phrases big data and data mining used in connection with biological data analysis. This masks the fact that biological data sets are not particularly big! Researchers have an unfortunate tendency to measure the size of biological data sets by the totality of raw data that is stored in various repositories such as the Gene Expression Omnibus. In reality, however, the input to any one algorithmic study is but a tiny fraction of this entire data set and is thus not particularly large. A typical whole-genome cancer study might measure 20,000 or so genes via

40,000 to 50,000 probes across a few hundred patients for a total of a few million entries. A typical biomarker study analyzes a data set of this size. In contrast, about 10,000 credit card transactions are processed around the world every second. Thus, the available data for devising algorithms to identify fraudulent credit card transactions would consist of nearly a billion records per day, or 300 billion records per year. Any researcher who wishes to develop an algorithm for fraud detection must analyze this large data set. At the same time, biological data analysis typically does not fit the paradigm of looking for a needle in a haystack that often characterizes data mining applications. Persisting with the credit card fraud example, according to the European Central Bank, only about 0.040% of all transactions—about 1 in 2,500 transactions—are fraudulent (9). Thus, any automated approaches for detecting credit card fraud must either be enormously powerful or risk committing massive errors in terms of false positives or false negatives. Fortunately, in cancer biology, the sizes of the groups to be distinguished are more evenly balanced, such as responders versus nonresponders to therapy or those at risk for metastasis versus those not at risk.

Given the scope and nature of biological data analysis, it would be fruitful to apply the methods of machine learning theory, which is based on sound statistical foundations. This is preferable to relying upon on ad hoc approaches that may or may not work. The review focuses on three subareas within the broad domain of machine learning: sparse feature selection for classification, sparse feature selection for regression, and network inference. Classification refers to the problem of assigning a given sample to one of two (and sometimes more) categories on the basis of various molecular measurements. Typical applications include predicting whether a patient is likely to respond to standard front-line therapy or whether a patient is at risk for metastasis. The expected outcome in each of the above examples is binary, that is, yes or no. Regression refers to the problem of predicting a real number, such as the expected time of survival or time to tumor recurrence. The word regression is potentially a source of confusion, as it has nothing to do with the regression of a tumor. In mathematics, regression essentially means curve-fitting. Both of these problems are widely studied in machine learning. Network inference, which really belongs to a different branch of statistics, nonetheless complements the other two areas by providing context to the process of selecting biomarkers. Whereas sparse feature selection leads to automated procedures for identifying biomarkers, network inference provides both a priori guidance to the biomarker selection algorithms and a posteriori biological context to the identified biomarkers.

The relative size of features versus samples is a key attribute that distinguishes biological data sets from conventional data sets in machine learning. In a typical machine learning application, such as detecting credit card fraud or recognizing faces, the data usually consist of a very large number of samples, each of which consist of relatively few features. A credit card transaction can be described by about ten parameters, such as the size of the transaction, the geographic location, the time interval between the last and the current transaction, and the geographic distance between the location of the last and current transaction. So the number of features is about ten, whereas the number of samples that can be used to train a good fraud detector can be in the millions, if not higher. Similarly, an image of a face can be represented as a 10×10 image, making the number of features equal to 100, whereas the number of facial images that can be used to train the recognition system is often in the hundreds of thousands, if not millions. In sharp contrast, in a typical cancer database, the measurements on each tumor sample would consist of whole-genome expression levels, ranging from 12,000 to 20,000 genes and up to 50,000 probes, depending on the apparatus used; against this, the number of distinct samples for which data are available would at best be a few hundred. This inversion between the relative sizes of the number of features and the number of samples has a major impact on the type of machine learning algorithms that are appropriate for use in biological applications. Only during the past ten years or so has the machine learning research community begun to develop algorithms that are specifically tailored

to situations in which the number of features vastly exceeds the number of samples. As this is precisely the situation that prevails in biology, the present review focuses on this class of machine learning algorithms.

The remainder of the article is organized as follows: The next section briefly reviews some well-established algorithms that are widely used in feature selection, although only some of these techniques could be called machine learning. The following three sections introduce three different topics, namely classification, regression, and network inference. Throughout these four sections, several references to the biology literature are given that illustrate the application of these techniques, although clearly these references are only indicative and not exhaustive. In the final section, some challenges in applying machine learning methods to biological data are discussed.

SOME ESTABLISHED ALGORITHMS

In this section, we briefly review some well-established algorithms that are widely used in biomarker discovery.

Neural Networks

The subject of (artificial) neural networks has a long history, but its recent revival began during the mid-1980s. A typical neural network contains many more adjustable parameters than inputs, representing the number of features used by the predictor. Researchers must therefore choose the features before embarking upon training the neural network. In other words, neural networks are not a feature selection methodology but rather a methodology for building a good predictor using a feature set selected through some other mechanism. In feature selection, we measure a great many quantities because the relevant features are unknown. Neural networks are not an appropriate tool for such situations. Nevertheless, they are mentioned here in the interest of completeness.

The most popular type of neural network is the multilayer perceptron network. As the name implies, the basic building block in such a network is a perceptron, which is a binary device whose output is $+1$ if the weighted sum of its inputs exceeds a threshold and -1 (or 0) if the weighted sum does not exceed the threshold. In mathematical terms, if there are l inputs to the perceptron denoted by $x_1, \ldots, x_l$ and the output is denoted by y, then the input-output relation of the perceptron is given by

$$y = \begin{cases} +1 & \text{if} \quad \sum_{i=1}^{l} w_i x_i - \theta \geq 0, \\ -1 & \text{if} \quad \sum_{i=1}^{l} w_i x_i - \theta < 0, \end{cases}$$

where $w_1, \ldots, w_l$ are the weights and θ is the threshold. **Figure 1*a*** depicts the input-output relationship of a perceptron. In some applications, the perceptron is replaced by a so-called sigmoidal nonlinearity, whose input-output relationship is given by

$$y = \frac{e^z}{1 + e^z}, \quad \text{where} \quad z = \sum_{i=1}^{l} w_i x_i.$$

Figure 1*b* depicts the input-output relationship of a sigmoidal nonlinearity.

Neural networks have been used to predict the response to radiation therapy and androgen-deprivation therapy in prostate cancer (10). The inputs to the neural network are measurements of four parameters at three different points in time, or twelve parameters in all. This is a typical use of neural networks, wherein the human domain experts select the appropriate features. In

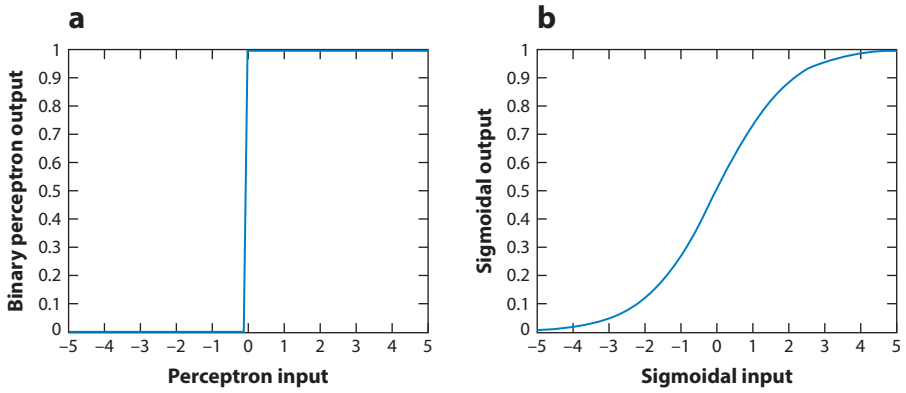


Figure 1

Input-output maps of (*a*) a perceptron and (*b*) a sigmoidal nonlinearity.

contrast, Menden et al. (11) use a neural network with 827 input parameters, consisting of 689 chemical descriptors and 138 genomic features, to predict 38,930 IC_{50} values on cancer cell lines. It is rather unusual to use neural networks with such a large number of inputs.

There are several excellent textbooks on neural networks that are addressed to a mathematically trained readership, and far fewer that address a nonmathematical user community. However, there are software packages available for training neural networks in various programming languages, and these can be used by the nonexpert. The only caution is that the user must ensure that neural networks are an appropriate methodology for the problem at hand. As stated above, neural networks are not a suitable tool for feature selection; rather, they should be used when the key features have already been chosen based on some other consideration, such as the users' expertise.

Unsupervised Clustering

Suppose we are given a collection of vectors $x_1, \ldots, x_m$, where each vector x_i is n-dimensional. In biological terms, the integer m denotes the number of samples on which measurements are taken, and n denotes the number of features that are measured on each sample. Therefore, the vector x_i represents the feature vector corresponding to the i-th sample. Unsupervised clustering refers to any approach in which only some general rules are specified, and the various samples are grouped into clusters following the rules.

Hierarchical clustering, also known as connectivity-based clustering, is a common method. In this approach, all the m samples are connected via a dendrogram, in which a node is more related to other nodes that are close by than to other nodes that are far away. Different definitions of relatedness lead to different dendrograms. The algorithm begins by measuring the distance between feature vectors, and the Euclidean distance is a popular, although by no means the only, choice. Once this step is executed, there are several options. The user can opt for single linkage clustering, in which the minimum of object distances is used; complete linkage clustering, in which the maximum of object distances is used; or average distances are used. The last approach is usually referred to by its acronym UPGMA (unweighted pair group method with average). In addition, we can either begin with individual nodes (samples) and grow them into clusters, which is known as agglomerative clustering, or begin with the entire set of samples in one large cluster and then divide them, which is known as divisive clustering. For all these approaches to clustering, efficient algorithms are available.

The well-known Mammaprint panel of 70 genes for predicting the likelihood of distant metastasis in breast cancer patients (12–14) is obtained using unsupervised hierarchical clustering. Another popular method of unsupervised clustering is k-means clustering, which seeks to detect similarities among the various samples by clustering the m vectors into k different clusters, where k is specified by the user. We can associate a centroid $\bar{x}_i$ to each cluster, which is just the average of all the feature vectors within that cluster. Note that each $\bar{x}_i$ is also an n-dimensional vector. The clusters are chosen in such a way that each feature vector within a particular cluster is closer to the centroid of that cluster than it is to the centroid of any other cluster. In this definition, the most popular choices for measuring the distance between two feature vectors are the Euclidean norm, or mean-squared norm, and the ℓ_1-norm, which is the sum of the absolute values of a vector. This technique was first introduced by MacQueen (15).

If the user insists on obtaining an exact solution, then the problem is NP-hard. This means there are no efficient methods either for solving the problem exactly or for verifying that a proposed candidate solution is indeed a solution, provided that a technical condition written as $P \neq NP$ holds. It is widely believed that $P \neq NP$, although this has not yet been proven. For this reason, all available algorithms for solving the k-means clustering problem are suboptimal, meaning they return a clustering in which most, although not necessarily all, vectors within a particular cluster are closer to the centroid of that cluster than they are to the centroid of any other cluster. The main reason for using these suboptimal algorithms is that they can be efficiently implemented.

In the k-means problem, the number of clusters must be specified by the user. Also, because available algorithms are based on randomization, even for a fixed value of k, the algorithms need to be rerun several times with different initializations, before the best output is chosen. The k-means clustering problem is an example of an unsupervised learning problem in that the natural groupings arise out of the algorithm and are not specified beforehand.

Once all the available feature vectors are clustered, the clustering can be used for classification purposes. Specifically, suppose a new test vector z is given, corresponding to the n-dimensional feature vector of a new sample; it is then assigned to the cluster whose centroid is closest to it. In other words, the new feature vector z is assigned to the i-th cluster if the distance between z and the centroid $\bar{x}_i$ is minimal. The main advantage of k-means clustering is that, as stated above, the process is unsupervised and thus not biased by the user. The main disadvantage is that, to implement it, the distance between feature vectors of very large dimension must be computed. For instance, in genomics studies, it becomes necessary to compute pairwise distances between vectors of dimension 20,000 or more. The method discussed below tries to simplify this aspect of the problem.

Pattern Analysis for Microarrays

The pattern analysis for microarrays (PAM) method is also referred to as the method of shrunken centroids, as described in a paper by Tibshirani et al. (16). Only the basics of the method are sketched here, and the reader is directed to the original publication or to section 2.4 of Reference 17 for a fuller treatment.

Recall how clustering is used to classify new samples. The first step is to cluster the given data consisting of m feature vectors $x_1, \ldots, x_m$, each of which is an n-dimensional vector. Let $\bar{x}_1, \ldots, \bar{x}_k$ denote the centroids of the resulting clusters. If a new feature vector z is given, it is classified to the cluster whose centroid is closest to it. In other words, for each cluster, we compute the squared Euclidean distance

$$d^2(z, \bar{x}_i) = \sum_{j=1}^{n} |z_j - \bar{x}_{i,j}|^2$$

and then choose the index i for which the above quantity is minimal. Now observe that, if for some index j, the coordinates of all k centroids are roughly the same, then that index can be omitted from the above calculation, as that term is just (nearly) a constant offset that does not affect the final choice of the closest centroid. Of course, the i-th coordinate of all centroids will never be exactly the same; rather, we can hope that they are all nearly the same so that the j-th index can be omitted from the summation. This can be accomplished by setting a threshold Δ and omitting the j-th index if the maximum variation of the j-th coordinate of all k centroids is smaller than Δ. If we set Δ too high, then a larger number of indices can be dropped from the above summation, but the accuracy of the classification decreases. If we set Δ too low, then there would not be much simplification. Tibshirani et al. (16) suggest using cross-validation to choose Δ. These authors describe an application to cancer in which an initial feature set of 2,308 genes is reduced to just 43 genes.

Significance Analysis for Microarrays

In a binary classification problem, the data consist of labeled vectors $\{(x_1, y_1), \ldots, (x_m, y_m)\}$, where each x_i is an n-dimensional vector denoting the n features measured on the i-th sample, and the label y_i is binary, e.g., positive or negative. To determine whether a particular feature is useful for distinguishing between the two classes, we can compute the mean value for the positively and negatively labeled samples for that feature and then apply the well-known Student t test to determine whether the difference between the two means is statistically significant. Such an exercise would lead to the retention of very few of the original features as being useful. However, this by itself is not necessarily a meaningful approach. Even if the data had been generated totally at random, the nature of the t test is that, at a 95% significance level, roughly 5% of the features would be found to show a significant difference between the means of the two classes. It is therefore necessary to look further.

Significance analysis for microarrays (SAM) may be used to identify, from within the features that pass the t test, a still more significant feature set (18). The idea is as follows: We choose a fairly large number of permutations of the integers 1 through m, such that a fixed number of samples from each of the two classes have their labels swapped. Each permutation is then applied to the samples. We then compute the average of the t-test statistic over all permutations. If this particular feature has been chosen as an artifact of the noise in measurements, the average of the t-test statistic over all permutations would be roughly the same as that of the original set of labels. If so, such a feature is rejected. The only features retained are those for which the t-test statistic with the natural labeling differs significantly from the average over all permutations. Tusher et al. (18) found only 46 out of an original set of 6,800 genes to be significant. Note that, without the SAM procedure, we would expect roughly 5% of 6,800 genes, or 340 genes, to pass the t test. Thus, SAM results in an order of magnitude reduction in the number of features chosen to be significant.

The SAM procedure is also discussed in detail in section 2.2 of Reference 17. Readers interested in finding out more information can also consult that source.

Gene Set Enrichment Analysis

The SAM procedure discussed above aims to unearth individual features that show a significant difference in means between two classes that is not an artifact of noisy data. The gene set enrichment analysis (GSEA) goes further and tries to look for collections of features that show such a difference. This approach is suggested by Efron & Tibshirani (19) and builds on earlier efforts by Subramanian et al. (20). The main difficulty with the t test is that, although two individual

features might not show a statistically significant difference between the means of the two classes, a linear combination of the two features might. Indeed, there is nothing special about the number of features being two. Given k features, none of which shows a statistically significant difference between the means of the two classes, it is possible to determine an optimal linear combination of all the k features for which the t-test statistic is maximized. Thus, in GSEA, we not only permute labels as in SAM but also replace chosen significant features with other features chosen at random and tests to see whether the difference is still significant. The details are rather mathematical, and the interested reader is referred to the original references (19, 20) or to the discussion in section 2.3 of Reference 17.

SPARSE CLASSIFICATION

In a typical classification problem, the data consist of a set of labeled samples in the form $\{(x_i, y_i), i = 1, \ldots, m\}$, where m denotes the number of samples for which data are available, for example the number of tumors; each x_i is an n-dimensional vector of real numbers corresponding to the molecular measurements of n features on sample i; and y_i is the label of sample i, assuming just two possible values, often denoted by $+1$, -1, or 0,1, or just the symbols P, N for positive and negative. It is also possible to consider multiclass problems where y_i can assume more than two possible values, but such a discussion is not possible within the scope of this review. Note that if y_i is a real number, then the problem becomes one of regression, as discussed in the section titled Sparse Regression.

The customary approach to binary classification problems is to define a so-called discriminant function f that associates with each feature vector x_i a real number $f(x_i)$. If the discriminant $f(x_i)$ is positive, then the sample i is assigned the label P (or $+1$), whereas if the discriminant $f(x_i)$ is negative, then the sample i is assigned the label N (or 0 or -1). The most common discriminant functions are linear, that is,

$$f(x) = \sum_{j=1}^{n} w_j x_j - \theta,$$

where $w_1, \ldots, w_n$ are the weights and θ is the threshold. Note that, in the above summation, it is possible to replace the measured feature component x_j by some preprocessed version of it, such as $\log(x_j)$ or $e^{x_j}/(1 + e^{x_j})$. Thus, although the discriminant function is linear, there is actually a considerable amount of flexibility in how it is chosen. For instance, if the argument to the function f consists of the logarithm of some values x_i, then f is the logarithm of the product of the corresponding features. The main reason for choosing linear discriminant functions is that they are easy to analyze from a theoretical standpoint and are also biologically realistic.

In the current context, in which the number of features far exceeds the number of samples, it is desirable to impose a further constraint on the discriminant function f, namely that the weights w_i should be nonzero for only a few values of the index i. In other words, the linear discriminant function should make use of a very small number of features; this is known as sparse feature selection. The remainder of this section describes some methods for solving this problem.

One of the most popular and widely used linear classifiers is known as the support vector machine (SVM) (21), which can be described as follows: Given the labeled feature vectors $(x_1, y_1), \ldots, (x_m, y_m)$, we find a hyperplane that separates the positively labeled samples from the negatively labeled samples while ensuring that the closest point in each class is as far away as possible from the hyperplane. This situation is depicted in **Figure 2**.

At this point, the reader might wonder whether there exists a hyperplane that separates the two classes of points, let alone an optimal separating hyperplane. Wenocur & Dudley (22) provide an answer in their well-known theorem in machine learning, which states that if the number of

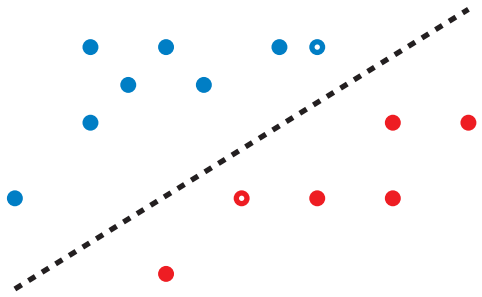


Figure 2

Optimal separating hyperplane. The blue and red dots denote the feature vectors belonging to the two classes, while the hollow dots denote the closest points within each class to the optimal separating hyperplane.

features n exceeds $m - 1$, then generically it is always possible to separate the positively labeled samples from the negatively labeled samples, for each of the 2^m possible ways of assigning labels to points. Because in biology the number of features vastly exceeds the number of samples, the existence of a separating hyperplane is assured.

A search of the PubMed database (**http://www.ncbi.nlm.nih.gov/pubmed**) reveals hundreds of applications of the SVM algorithm to cancer. The approach has clearly found favor within the research community. The main shortcoming of the traditional SVM methodology is that the optimal separating hyperplane in general uses all n features, which is certainly undesirable. Therefore, one approach is to determine the optimal set of weights, discard the feature with the smallest weight (in absolute value), and repeat. This approach, described in a paper by Guyon et al. (23), is known as recursive feature elimination (RFE). The SVM with RFE proposed in that paper is claimed to work well for a leukemia data set and to result in just two features being finally retained. Be that as it may, SVM with RFE generally behaves somewhat erratically and is computationally intensive.

Bradley & Mangasarian (24) propose a more promising approach to choosing relatively few features via a modification of the standard SVM, known as the ℓ_1-norm SVM. The ℓ_1-norm of a vector is the sum of the absolute values of its components and is an alternative to the more widely used Euclidean norm. The main advantage of the ℓ_1-norm SVM, as shown in Reference 24, is that the optimal separating hyperplane uses no more than m of the n features. In problems such as those in biology, where m is two or even three orders of magnitude smaller than n, this is a huge advantage.

Despite this, there are applications where even m biomarkers are too many, especially in large studies with hundreds of samples. It is therefore desirable to combine RFE with the ℓ_1-norm SVM to make the number of chosen features substantially smaller than m. The resulting algorithm is referred to as the lone-star algorithm (25). This approach is described in greater detail in the perspective article by Vidyasagar (26). In an application to endometrial cancer, the lone-star algorithm is able to find 13 key microRNAs that can predict lymph node metastasis. Despite this success, as of now there is no theoretical analysis to explain why the lone-star algorithm seems to use far fewer features than the number of samples. The theoretical research community needs to study this problem, whose resolution would have an immediate application to biology.

SPARSE REGRESSION

The most familiar example of linear regression is least-squares error minimization, which was invented by Gauss and Legendre more than two hundred years ago and has been a staple technique

ever since. To state this problem formally, suppose, as before, that we have a set of m labeled samples, in the form $(x_1, y_1), \ldots, (x_m, y_m)$, where each x_i is an n-dimensional feature vector corresponding to the i-th sample, and y_i is a real number. We define a regressor function f that associates a real number $f(x_i)$ with the feature vector x_i, and the hope is that $f(x_i)$ is a reasonable approximation of the label y_i for each sample index i. To measure how good a regressor the function f is, we define the familiar least-squares error

$$I = \sum_{i=1}^{m} |y_i - f(x_i)|^2,$$

and then choose the function f to minimize the objective function I. As with classification, the most commonly used regressor functions are linear regressors of the form

$$f(x) = \sum_{j=1}^{n} w_j x_j - \theta.$$

The fact that the label is a real number distinguishes regression from classification, where the label assumes just two possible values. Note that, in classification problems also, the discriminant function f associates a real number $f(x_i)$ with each feature vector x_i. However, the difference is that, in the case of classification problems, the actual value of the discriminant function $f(x_i)$ is unimportant—so long as $f(x_i)$ has the same sign as the binary label y_i, we are content. In contrast, in regression problems, it is not just the sign of the discriminant function $f(x_i)$ that is important but also its magnitude. This is why regression is a more difficult problem than classification.

If we substitute the form of the regressor function into the error function I, we observe that I is a quadratic function of the weights w_i and threshold θ. In the case where the number of samples m exceeds the number of features n, the problem is overdetermined. As a result, there is a unique choice of weights and threshold that achieves the minimum of the error I. However, when the situation is reversed and the number of features exceeds the number of samples, the problem is underdetermined. Thus, there are infinitely many choices of the weights and threshold that achieve the minimum of the error I. To disambiguate among them and arrive at a unique minimizer, the usual procedure is to add a penalty term to the error I; this procedure is known as regularization. Different choices of the penalty term lead to different algorithms, which have their own advantages and disadvantages. Some of the more widely used algorithms are described in this section.

In 1943, Tikhonov (27) developed one of the very first approaches to regularization, in which the error function I is modified to

$$J_{\text{ridge}} = \sum_{i=1}^{m} \left| y_i - \sum_{j=1}^{n} w_j x_{j,i} - \theta \right|^2 + \lambda \sum_{i=1}^{n} |w_i|^2,$$

where λ is a user-defined parameter, often referred to as a Lagrange parameter. Note that J is a sum of the original error I and the Euclidean norm of the n-dimensional weight vector; note also that the threshold is not a part of the penalty term. In other words, the intent is to penalize overly large weights but not to worry about the magnitude of the threshold. This technique was later rediscovered by Hoerl & Kennard (28) and has come to be known as ridge regression. The main advantage of ridge regression is that, for each choice of the parameter λ, there is a unique optimal choice of weights, even when $n \gg m$. However, the main disadvantage of ridge regression is that, in general, every component of the optimal weight vector is nonzero. In other words, while ridge regression produces a unique optimal classifier corresponding to every choice of the parameter λ,

the corresponding regressor function makes use of every single feature. When n is in the tens of thousands, this is clearly a most undesirable property for an algorithm to have.

A major advance in regression took place with the publication of Tibshirani's 1996 paper (29), in which the penalty was changed from the Euclidean norm of the weight vector to the sum of the absolute values of its components, that is, the so-called ℓ_1-norm. This method is known as LASSO (least absolute shrinkage and selection operator). In LASSO, the objective function to be minimized is

$$J_{\text{LASSO}} = \sum_{i=1}^{m} \left| y_i - \sum_{j=1}^{n} w_j x_{j,i} - \theta \right|^2 + \lambda \sum_{i=1}^{n} |w_i|,$$

where, as before, λ is a Lagrange multiplier. The main advantage of the LASSO algorithm over ridge regression is that, in almost all cases, the optimal weight vector has no more than m nonzero components (30). Thus, even if there are 50,000 features and 100 samples, the optimal weight vector will have no more than 100 nonzero components. The flip side of this, of course, is that which m features get selected depends very much on the data; consequently, small perturbations of the data result in drastically different sets of features being selected in the optimal weight vector.

To alleviate this difficulty, Zou & Hastie (31) introduced an algorithm known as the elastic net (EN). The objective function to be optimized in the EN algorithm is

$$J_{\text{EN}} = \sum_{i=1}^{m} \left| y_i - \sum_{j=1}^{n} w_j x_{j,i} - \theta \right|^2 + \lambda \left[(1-\mu) \sum_{i=1}^{n} |w_i| + \mu \sum_{i=1}^{n} |w_i|^2 \right],$$

where, as before, λ is the Lagrange multiplier, and μ is another parameter that lies between 0 and 1. It is obvious that if $\mu = 0$, then EN becomes LASSO, whereas if $\mu = 1$, then EN becomes ridge regression. In this manner, the EN algorithm smoothly interpolates between LASSO and ridge regression. Unlike with LASSO, there is no prior limit on how many nonzero components the optimal weight vector can have. In particular, in numerical examples, generally the number of nonzero components exceeds the number of samples m. However, the main benefit of EN over LASSO is a property established by Zou & Hastie (31): If two feature vectors are highly correlated, then their weights are nearly equal. This is in sharp contrast to LASSO, which would choose one of the two and ignore the other. This makes the EN algorithm relatively less sensitive than LASSO to noise in the data. A detailed discussion of these algorithms can be found in Reference 32.

When applying regression methods to biological problems, it is not always appropriate merely to count the number of nonzero components of the weight vector or, equivalently, the number of features that are used by the regressor function. It is necessary to go into the finer structure of the features, which is provided by the underlying regulatory network. Regulatory networks and how to infer them are covered below in the section titled Network Inference. But to set the stage for that discussion, consider the very simple regulatory network shown in **Figure 3*a***. Suppose that there are two regressor functions, one of which uses features 1, 2, and 4, while the other uses features 1, 2, and 6. Which one is preferable? Recall that algorithms such as EN assign nearly equal weights to highly correlated features. Features 1, 2, and 4 lie in the same pathway, so it can be surmised that these three features are chosen because their values are highly correlated. In other words, the regressor that uses features 1, 2, and 4 is in effect telling us that the pathway $1 \rightarrow 2 \rightarrow 4$ is important, and in reality, any one of these features could probably be used as a biomarker. In contrast, features 1, 2, and 6 belong to two different pathways. Therefore, the regressor that uses these three features actually paints a more confusing picture than does the regressor that uses features 1, 2, and 4.

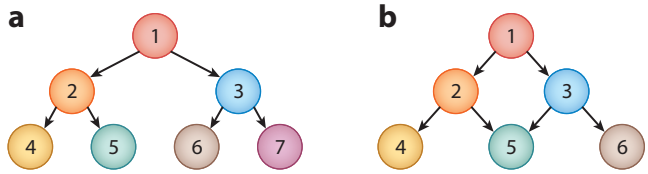


Figure 3

Two regulatory networks. (*a*) A network without overlapping groups and (*b*) one with overlapping groups.

A more interesting example is the following: Suppose one regressor chooses features 1, 2, and 4, while the other regressor chooses features 2 and 6. Because 1, 2, and 4 lie in a common pathway whereas 2 and 6 lie in different pathways, we should probably prefer the regressor that uses features 1, 2, and 4, even though the other regressor uses fewer features. In other words, ideally we should strive to choose a regressor that chooses features from the smallest number of pathways, as opposed to choosing a regressor that chooses the smallest number of features.

In short, it is necessary to go beyond LASSO- or EN-like algorithms that merely try to minimize the number of distinct features chosen. There are several variants of LASSO that try to accommodate the finer structure of the feature set, of which only two are mentioned here, namely the group LASSO and the sparse group LASSO. The starting point of both algorithms is a partitioning of the feature set $\{1, \ldots, n\}$ into g nonoverlapping subsets, which we will call $G_1, \ldots, G_g$. For instance, the various subsets can consist of all genes that are downstream of a master regulator. In the group LASSO algorithm, introduced by Yuan & Lin (33), the objective function is

$$J_{\mathrm{GL}} = \sum_{i=1}^{m} \left| y_i - \sum_{j=1}^{n} w_j x_{j,i} - \theta \right|^2 + \lambda \sum_{l=1}^{g} \sum_{i \in G_l} |w_i|,$$

whereas in the sparse group LASSO, introduced in References 34 and 35, the objective function is

$$J_{\mathrm{EN}} = \sum_{i=1}^{m} \left| y_i - \sum_{j=1}^{n} w_j x_{j,i} - \theta \right|^2 + \lambda \left[(1-\mu) \sum_{l=1}^{g} \sum_{i \in G_l} |w_i| + \mu \sum_{l=1}^{g} \sum_{i \in G_l} |w_i|^2 \right].$$

The group LASSO algorithm attempts to choose features from as few distinct sets G_l as possible, whereas the sparse group LASSO algorithm goes further, trying not only to choose features from as few distinct sets as possible but also to choose as few features as possible from within those sets. In the world of optimization, the analysis of these algorithms is still at a nascent stage. Therefore, it is hardly surprising that there are relatively few applications of these algorithms to biological problems.

NETWORK INFERENCE

The above section shows that, if some information is available about the biological pathways of all the genes in a study, then this information can be incorporated into feature selection via algorithms such as group LASSO or sparse group LASSO. If this is done, then the set of features selected by the regression algorithm will have some biological plausibility in addition to being sparse. The main difficulty in adopting such an approach is that, in any given application, very few pathways are known, and usually the number of known pathways is too small to have any impact on the feature selection algorithms. One way to remedy the situation is to augment the known pathways with inferred pathways. Methods for inferring whole-genome regulatory networks, which would

in turn lead to a great many inferred pathways on top of the known pathways, constitute the topic of this section.

The deduction of biological pathways, in terms of gene regulatory networks or protein-protein interaction networks, has been an active area of research for several decades. In earlier years, these interactions were determined through experimental techniques, a few at a time. Since then, several perturbation studies have shown that suppressing one gene causes the activity of many other genes to go up or down, and an explanation is sought for a set of such observations. Recently developed, large-scale experimental methods such as ChIP-seq (chromatin immunoprecipitation sequencing) result in a huge number of false positives, that is, predicted interactions that do not really exist. With recent advances in high-throughput data generation, there has been some interest in automated methods for inferring regulatory networks from data using statistical methods.

At present, there are several databases, either in the public domain or through commercial vendors, that are compilations of reported interactions between various genes and/or proteins. In addition, there are tools such as MiMI (36), which is a Cytoscape plug-in that can query multiple repositories of interaction databases such as IntAct (37), KEGG (38), MINT (39), and Reactome (40) and that can extract biomolecular interaction networks. The interaction type can also be specified, e.g., protein, DNA, or gene. Therefore, it is possible to generate a first-cut regulatory network from available data for each situation of interest. However, a database consisting of a mere compilation of available and proven interactions is often unsatisfactory for at least a couple of reasons. First, each interaction is established under one specific set of experimental conditions. Thus, it can be highly misleading to collect together, in a common database, a list of interactions established under widely disparate experimental conditions. Second, and perhaps more important, the networks of interactions so obtained often lack full coverage, in the sense that only a fraction of the genes in a study are present in such networks. Thus, there is a fundamental incompatibility between the expression data consisting of measurements on all these genes and the networks that include only a subset of genes. It is therefore imperative to use expression data to reverse-engineer whole-genome networks and not be content with networks that have only partial coverage. It is also essential to use only data from a common set of experimental conditions to construct the network, so as to ensure that the resulting network is context specific.

Ideally, the reverse-engineered network should have the following properties:

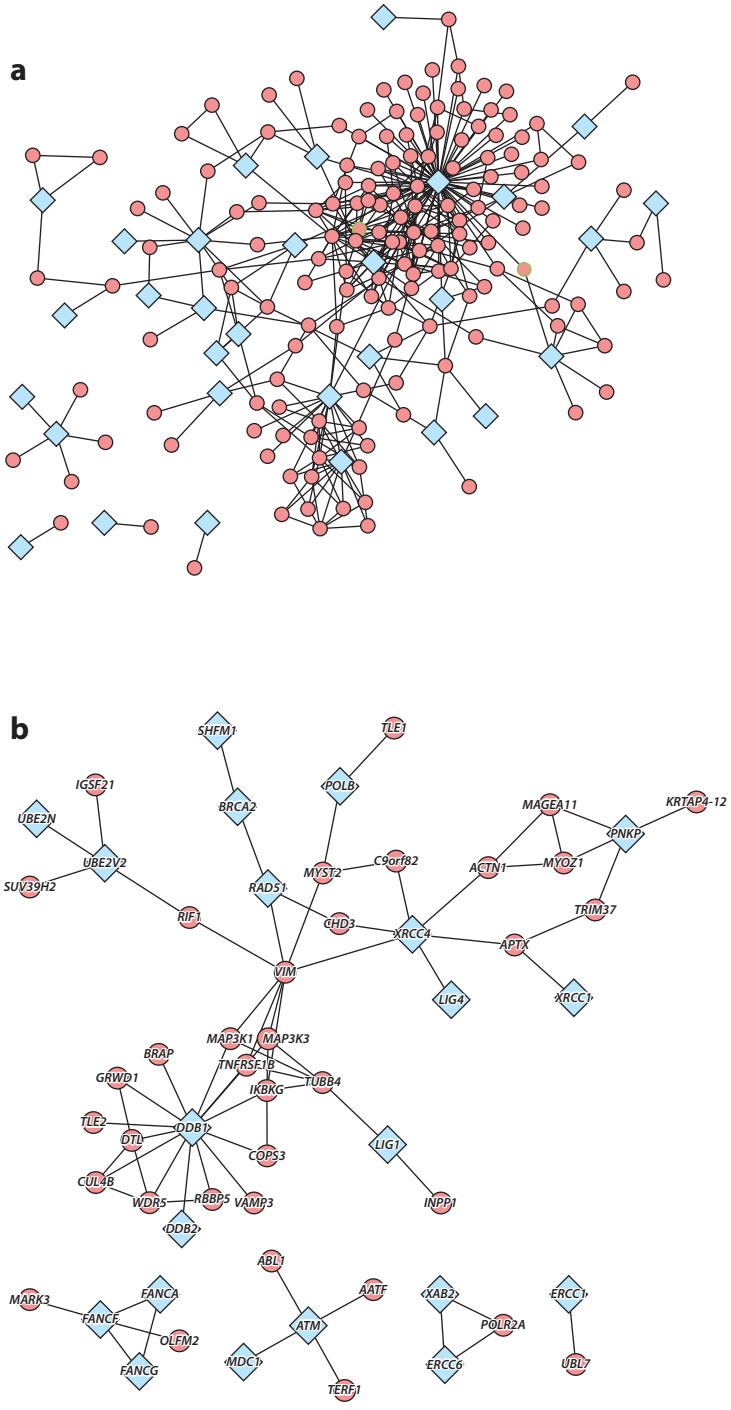
- The network should contain all the genes in a study as nodes, while interactions between genes are denoted by edges.
- The edges should be both (*a*) directed, so that whether gene *A* affects gene *B* or vice versa can be inferred, and (*b*) weighted, so that the extent of the influence can be quantified.
- The network should be strongly connected, so that there is a path between every pair of nodes; in other words, there should not be any dangling or isolated nodes.
- It should be straightforward to harmonize known interactions into the network.

Methods for constructing such networks form the topic of the remainder of this section.

As a concrete illustration of the need for such methods, a network was extracted from IntAct (37) using the BROCA biomarker panel (41) as the seed. The resulting network is shown in **Figure 4*a***. Note that the network is islanded and consists of several subgraphs that do not intersect. A similar exercise was undertaken in the case of DNA repair genes. Kang et al. (42) mention 151 DNA repair genes, of which only 135 are measured by the Affymetrix U133A platform. Another network was extracted from IntAct using these 135 genes as the seed. The resulting network is shown in **Figure 4*b***. Note that, unlike with the BROCA panel, only 62 of the 135 DNA repair genes are present in the IntAct database, and only 41 other genes in that database interact with the

Figure 4

Subgraphs generated using the MiMI utility and Cytoscape (36). Light blue diamonds denote genes of interest, and pink circles denote other genes that intermediate between these. (*a*) All 51 genes in the BROCA panel (41) and their intermediate genes from IntAct (37). (*b*) Some of the 151 DNA repair genes (42) and their intermediate genes from IntAct. Note that many DNA repair genes are absent from the IntAct database.



DNA repair genes. Thus, in both cases, there are serious deficiencies in the extracted networks, in terms of a lack of coverage as well as a lack of connectivity.

In the remainder of this section, we describe several methods for inferring whole-genome regulatory networks using statistical methods. Suppose that there are n genes in a study and that the expression level of each gene is measured in each of m samples. In the statistical approach, the expression level of gene i is viewed as a random variable X_i, and the measurements on each sample j, which we will call $x_{j1}, \ldots, x_{jn}$, are viewed as a joint sample of the set of n random variables $X_1, \ldots, X_n$. Note that these random variables are not assumed to be independent. Indeed, the objective of the exercise is to unearth interdependences among these random variables. Because $m \ll n$ in biological studies, there is no hope of being able to infer the joint distribution function of all n random variables based on m sets of measurements. Therefore, we look for some network model that is consistent with the data. In principle, we could compute the Pearson correlation coefficient between each pair of random variables (gene expression levels). However, the main difficulty with the Pearson correlation coefficient is that if the expression levels are processed via some nonlinear transformation—for example replacing the raw values by their logarithm after centering and scaling—then the correlation coefficient would change in general. In short, the Pearson correlation coefficient is invariant under linear transformations but not nonlinear transformations. Thus, we need to find some other measure of interaction or interdependence that remains invariant even under nonlinear transformations of the data.

In order to explain various statistical modeling methods, we introduce a bare minimum of statistical formalism. Suppose a random variable X is finite-valued, assuming values $a_1, \ldots, a_k$ with probabilities $\mu_1, \ldots, \mu_k$, respectively. In the context of gene regulatory networks, the values a_i would be real numbers; however, for the purposes of the discussion below, this is not necessary, as the values can be just abstract symbols. The entropy of the random variable X is denoted by the symbol $H(X)$ and is defined as

$$H(X) = -\sum_{i=1}^{k} \mu_i \log \mu_i.$$

The entropy of a random variable is a measure of just how uncertain it is—the greater the entropy, the more difficult it is to predict. Now suppose X and Y are random variables whose values range over $a_1, \ldots, a_k$ and $b_1, \ldots, b_l$, respectively. Again, in the present application, these would be real numbers, but for the definition below, they could be abstract symbols. The number

$$\psi_{ij} = \Pr\{X = a_i \text{ and } Y = b_j\}$$

is called the joint probability of the two random variables. The two random variables are said to be independent if

$$\Pr\{X = a_i \text{ and } Y = b_j\} = \Pr\{X = a_i\} \cdot \Pr\{Y = b_j\}.$$

Now the joint random variable (X, Y) has its own entropy, given by

$$H(X, Y) = -\sum_{i=1}^{k} \sum_{j=1}^{l} \psi_{ij} \log \psi_{ij}.$$

The mutual information between the two random variables is denoted by $I(X, Y)$ and is defined by

$$I(X, Y) = H(X) + H(Y) - H(X, Y).$$

The mutual information is a nonnegative number that can be interpreted as a quantitative measure of how much Y tells us about X and vice versa. In particular, if X and Y are independent random

variables, then $I(X, Y) = 0$; in other words, measuring X does nothing to reduce the uncertainty of Y and vice versa. Note that mutual information is symmetric; that is, $I(X, Y) = I(Y, X)$. One of the main advantages of mutual information compared to other symmetrical measures of dependence such as the Pearson correlation coefficient is that mutual information is invariant under any monotone transformation of the data, such as centering, scaling, and taking logarithms or exponentials.

One of the very useful properties of mutual information is known as the data processing inequality, and an intuitive (although imprecise) explanation of this property is given below. Suppose that in a graph containing n nodes, every path from node i to node k must pass through node j. Then

$$I(X_i, X_k) \leq \min\{I(X_i, X_j), I(X_j, X_k)\}.$$

Now we are in a position to discuss various approaches to statistical network inference. Perhaps the first such approach is that of influence networks introduced by Butte & Kohane (43). In this approach, we construct a graph with n nodes, where n is the number of genes in the study. The expression levels of all genes are discretized by partitioning the range of values into a finite number of bins and assigning each sample to the corresponding bin. This turns a real-valued expression level into a discretized (or finite-valued) random variable. The mutual information between each pair of genes, say X_i and X_j, is then computed using the formula given above. If the mutual information exceeds a user-defined threshold, an edge is introduced between nodes i and j. Note that the edge is undirected because mutual information is a symmetric quantity.

The main difficulty with influence networks as defined by Butte & Kohane (43) is that they are overly dense. The reason is that the recipe of introducing an edge between two nodes whenever their mutual information is sufficiently high does not distinguish between direct and indirect interactions. To get around this limitation, Margolin et al. (44) introduced an algorithm called ARACNE in which one begins with a so-called complete graph with undirected edges between every pair of vertices. Then, for each triplet of indices i, j, k, one checks whether the data processing inequality

$$I(X_i, X_k) \leq \min\{I(X_i, X_j), I(X_j, X_k)\}$$

is satisfied. If so, the edge between nodes i and k is pruned. A minor detail is that, whereas the mutual information is computed in Reference 43 by discretizing the random variables into a finite number of bins, the mutual information is computed in Reference 44 by approximating the observed values by Gaussian kernels; however, this need not concern us here. The network that results from applying the ARACNE algorithm is less dense than the influence network generated using Butte & Kohane's (43) approach. However, the network is both undirected and unweighted.

An algorithm developed by Vidyasagar and colleagues (45) is briefly described here. The above discussion makes clear that mutual information is not a very good measure of interaction between random variables, as it is symmetric and can thus lead only to undirected networks. Instead, we need to look for another measure of interaction between random variables that is not symmetric but satisfies the data processing inequality. Ahsen & Vidyasagar (46) show that the so-called ϕ-mixing coefficient has all the desired properties. A precise definition of the ϕ-mixing coefficient would require a lot of mathematical details; hence, its key properties are summarized below:

- If X, Y are any two random variables, then $\phi(X|Y)$, read as "phi of X given Y," is a number between 0 and 1.
- The ϕ-mixing coefficient is invariant under any monotone transformation of the data, such as centering, scaling, and taking logarithms or exponentials.
- In general, $\phi(X|Y) \neq \phi(Y|X)$.
- $\phi(X|Y) = 0$ if and only if the random variables are independent.

- There is an easily computable formula for the ϕ-mixing coefficient.
- The ϕ-mixing coefficient satisfies the data processing inequality.

Because of these properties, it is possible to define an algorithm that produces weighted and directed graphs as follows: Start with a complete directed graph of n nodes, with one node for each gene. Introduce a directed edge between every pair of genes. Then, for each triplet of indices i, j, k, check whether the data processing inequality

$$\phi(X_i|X_k) \leq \min\{\phi(X_i|X), \phi(X_j|X_k)\}$$

is satisfied. If so, prune the edge from node i to node k. As mentioned above, the regulatory network produced by this algorithm has weighted and directed edges and does not have any islands. Therefore, the pathways in the network can be used together with feature selection algorithms such as group LASSO or sparse group LASSO to identify a sparse set of features that are also biologically meaningful.

CHALLENGES IN APPLYING MACHINE LEARNING TECHNIQUES

The field of machine learning has already proved its mettle in solving difficult engineering problems such as recognizing handwritten characters and recognizing faces from a photograph. Therefore, there is no doubt that the methodology is very sound. However, some serious difficulties arise when it comes to applying machine learning techniques to problems of predicting drug response. These difficulties can be summed up in one phrase: the lack of reliability and repeatability of biological data. Two distinct sources of unreliability are often confused. The first is the inherent nonrepeatability of experimental conditions and/or biological phenomena. This by itself does not limit the applicability of machine learning methods, as there are analogs of this nonrepeatability in the engineering world as well. For instance, a machine learning algorithm for recognizing faces from photographs has to work under a wide variety of lighting conditions and camera angles. Therefore, it is possible to factor in this type of variability while designing algorithms. The second and more insidious factor is the apparently poor quality control on the extremely costly equipment used to make biological measurements. Our experience has been that, even when allegedly the same platform was used in two different laboratories, the resulting data sets were not always compatible, indicating a serious lack of standardization of operating conditions for the equipment and perhaps even a serious lack of any attempt at standardization. For instance, the probe sets on two data sets that ostensibly used the same microarray platform do not always match, and when the probe sets match, the statistical distributions of the measurements do not always match. In the world of engineering, it is pretty much taken for granted that the measurement apparatus is thoroughly dependable and produces repeatable measurements. If this cornerstone of data acquisition is removed, it is not clear how we can proceed.

We conclude with some observations about the nature of biomarkers identified via machine learning algorithms. Suppose that we begin with a data set consisting of expression values of 20,000 genes and 500 patient samples and identify a small number (say 30) of biomarkers that are highly predictive of clinical response. Now the question arises: How reliable are these biomarkers? In other words, if all the expression values were to be perturbed up or down at random by 10% or less, would the machine learning algorithm return the same 30 biomarkers? Obviously, the answer is no. While a given set of 30 biomarkers might be optimal for a given data set, many other biomarker panels that are nearly as good are bound to exist, and which set of biomarkers finally gets chosen by the machine learning algorithm is significantly affected by small changes in the data. The key point here is that the predictor based on the unperturbed data would still be an excellent predictor of the perturbed data. Therefore, the quality of the predictions is quite robust against

perturbations in the data, in the sense that an optimal set of biomarkers for the original data would still be nearly optimal for the perturbed data. In other words, so long as we do not try to imbue too much biological significance into the identified biomarkers, machine learning algorithms will produce excellent predictors. The above discussion on inferring regulatory networks and using an inferred network to guide the choice of biomarkers will address this concern, but only to a limited extent. To identify biomarkers that are both highly predictive and biologically meaningful, it is necessary to undertake some basic research into algorithm development wherein we distinguish between known interactions, inferred interactions, and other possible interactions. The algorithms developed via such research need to be constantly validated in actual clinical applications, leading to considerable benefit to both the research and clinical communities.

DISCLOSURE STATEMENT

The author is not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

LITERATURE CITED

1. Siegel R, Ma J, Zou Z, Jamal A. 2014. Cancer statistics, 2014. *CA Cancer J. Clin.* 64(1):9–29
2. Cancer Research UK. 2013. *Cancer statistics report: cancer mortality in the UK in 2011*. Rep., Cancer Res. UK, London, UK. **http://publications.cancerresearchuk.org/downloads/Product/CS_CS_MORTALITY.pdf**
3. World Health Organization. 2014. *Cancer*. World Health Organ., Cancer Control Programme, Geneva, Switz. **http://www.who.int/cancer/en/**
4. Cancer Genome Atlas Res. Netw. 2011. Integrated genomic analyses of ovarian carcinoma. *Nature* 474:609–15
5. Cancer Genome Atlas Netw. 2012. Comprehensive molecular characterization of human colon and rectal cancer. *Nature* 487:330–37
6. Cancer Genome Atlas Netw. 2012. Comprehensive molecular portraits of human breast tumours. *Nature* 490:61–70
7. Cancer Genome Atlas Res. Netw. 2012. Comprehensive genomic characterization of squamous cell lung cancers. *Nature* 489:519–25
8. Cancer Genome Atlas Res. Netw. 2008. Comprehensive genomic characterization defines human glioblastoma and core pathways. *Nature* 455:1061–68
9. European Central Bank. 2012. *Report on card fraud*. Rep., July, Eur. Cent. Bank, Frankfurt am Main, Ger. **http://www.ecb.europa.eu/pub/pdf/other/cardfraudreport201207en.pdf**
10. Røe K, Kakar M, Seierstad T, Ree AH, Olsen DR. 2011. Early prediction of response to radiotherapy and androgen-deprivation therapy in prostate cancer by repeated functional MRI: a preclinical study. *Radiat. Oncol.* 6:65
11. Menden MP, Iorio F, Garnett M, McDermott U, Benes CH, et al. 2013. Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties. *PLOS ONE* 8(4):e61318
12. van 't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AAM, et al. 2002. Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415:530–37
13. Mook S, Schmidt MK, Weigelt B, Kreike B, Eekhout I, et al. 2010. The 70-gene prognosis signature predicts early metastasis in breast cancer patients between 55 and 70 years of age. *Ann. Oncol.* 21:717–22
14. Kok M, Koornstra RH, Mook S, Hauptmann M, Fles R, et al. 2012. Additional value of the 70-gene signature and levels of ER and PR for the prediction of outcome in tamoxifen-treated ER-positive breast cancer. *Breast* 21:769–78
15. MacQueen JB. 1967. Some methods for classification and analysis of multivariate observations. *Proc. Fifth Berkeley Symp. Math. Stat. Probab.*, pp. 281–97. Berkeley: Univ. Calif. Press

16. Tibshirani R, Hastie T, Narasimhan B, Chu G. 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl. Acad. Sci. USA* 99(10):6567–72
17. Vidyasagar M. 2012. *Computational Cancer Biology: An Interaction Network Approach*. London: Springer
18. Tusher VG, Tibshirani R, Chu G. 2001. Significance analysis of microarrays applied to the ionizing radiation responses. *Proc. Natl. Acad. Sci. USA* 98(9):5116–21
19. Efron B, Tibshirani R. 2007. On testing the significance of a set of genes. *Ann. Appl. Stat.* 1(1):107–29
20. Subramanian A, Tamayo P, Mootha V, Mukherjee S, Ebert BL, et al. 2005. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* 102(43):15545–50
21. Cortes C, Vapnik VN. 1995. Support vector networks. *Mach. Learn.* 20:273–97
22. Wenocur RS, Dudley RM. 1981. Some special Vapnik-Chervonenkis classes. *Discret. Math.* 33:313–18
23. Guyon I, Weston J, Barnhill S, Vapnik V. 2002. Gene selection for cancer classification using support vector machines. *Mach. Learn.* 46:389–422
24. Bradley PS, Mangasarian OL. 1998. Feature selection via concave minimization and support vector machines. In *ICML '98 Proc. Fifteenth Int. Conference Mach. Learn.*, pp. 82–90. San Francisco: Morgan Kaufmann
25. Ahsen ME, Singh NK, Boren T, Vidyasagar M, White MA. 2012. A new feature selection algorithm for two-class classification problems and application to endometrial cancer. In *Proc. IEEE 51st Conf. Decis. Control, Maui, HI, Dec. 10–13*, pp. 2976–82
26. Vidyasagar M. 2014. Machine learning methods in the computational biology of cancer. *Proc. R. Soc. A* 470:20140081
27. Tikhonov AN. 1943. On the stability of inverse problems. *Dokl. Akad. Nauk SSSR* 39(5):195–98
28. Hoerl AE, Kennard RW. 1970. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 12(1):55–67
29. Tibshirani R. 1996. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B* 58(1):267–88
30. Osborne MR, Presnell B, Turlach BA. 2000. On the LASSO and its dual. *J. Comput. Graph. Stat.* 9:319–37
31. Zou H, Hastie T. 2005. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B* 67:301–20
32. Hastie T, Tibshirani R, Friedman J. 2011. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer. 2nd ed.
33. Yuan M, Lin Y. 2006. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B* 68:49–67
34. Friedman J, Hastie T, Tibshirani R. 2010. *A note on the group lasso and sparse group lasso*. Dep. Stat., Stanford Univ., Stanford, CA. **http://statweb.stanford.edu/~tibs/ftp/sparse-grlasso.pdf**
35. Simon N, Friedman J, Hastie T, Tibshirani R. 2013. A sparse-group lasso. *J. Comput. Graph. Stat.* 22:231–45
36. MiMI. 2013. Cytoscape plugin for MiMI. Univ. Mich., Ann Arbor, MI. **http://mimiplugin.ncibi.org/**
37. IntAct. 2014. *IntAct molecular interaction database*. EMBL-EBI, Wellcome Trust Genome Campus, Hinxton, Cambridgeshire, UK. **http://www.ebi.ac.uk/intact/**
38. KEGG. 2014. *KEGG database*. Kanehisa Laboratories, Inst. Chem. Res., Kyoto Univ., Kyoto, Jpn. **http://www.genome.jp/kegg/kegg1.html**
39. MINT. 2014. *MINT: Molecular INTeraction database*. **http://mint.bio.uniroma2.it/mint/Welcome.do**
40. Reactome. 2013. *Reactome Pathway Browser*. Reactome, **http://www.reactome.org/PathwayBrowser/**
41. BROCA. 2013. *BROCA—Cancer Risk Panel*. Lab. Med., Univ. Wash., Seattle, WA. **http://tests.labmed.washington.edu/BROCA**
42. Kang J, D'Andrea AD, Kozono D. 2012. A DNA repair pathway–focused score for prediction of outcomes in ovarian cancer treated with platinum-based chemotherapy. *J. Natl. Cancer Inst.* 104(9):670–81
43. Butte AJ, Kohane IS. 2000. Mutual information relevance networks: functional genomic clustering using pairwise entropy measures. *Pac. Symp. Biocomput.* 2000:418–29

44. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, et al. 2008. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a cellular context. *BMC Bioinform.* 7(Suppl. 1):S7
45. Singh NK, Ahsen ME, Mankala SK, Vidyasagar M, White MA. 2012. Inferring weighted and directed gene interaction networks from gene expression data using the phi-mixing coefficient. *Proc. 2012 IEEE Int. Workshop Genomic Signal Process. Stat.* (*GENSIPS'12*), *Washington, DC, Dec. 2–4*, pp. 168–71
46. Ahsen ME, Vidyasagar M. 2014. Mixing coefficients between discrete and real random variables: computation and properties. *IEEE Trans. Autom. Control* 59(1):34–47