Journal of Computational Biology, in press, March 2004.

A Bayesian Network Classification Methodology for Gene Expression Data

Paul Helman,[1] Robert Veroff,[1] Susan R. Atlas,[2] and Cheryl Willman[3]

Abstract

We present new techniques for the application of a Bayesian network learning framework to the problem of classifying gene expression data. The focus on classification permits us to develop techniques that address in several ways the complexities of learning Bayesian nets. Our classification model reduces the Bayesian network learning problem to the problem of learning multiple subnetworks, each consisting of a class label node and its set of parent genes. We argue that this classification model is more appropriate for the gene expression domain than are other structurally similar Bayesian network classification models, such as Naive Bayes and Tree Augmented Naive Bayes (TAN), because our model is consistent with prior domain experience suggesting that a relatively small number of genes, *taken in different combinations*, is required to predict most clinical classes of interest. Within this framework, we consider two different approaches to identifying parent sets which are supported by the gene expression observations and any other currently available evidence. One approach employs a simple greedy algorithm to search the universe of all genes; the second approach develops and applies a gene selection algorithm whose results are incorporated as a prior to enable an exhaustive search for parent sets over a restricted universe of genes. Two other significant contributions are the construction of classifiers from multiple, competing Bayesian network hypotheses and algorithmic methods for normalizing and binning gene expression data in the absence of prior expert knowledge. Our classifiers are developed under a cross validation regimen and then validated on corresponding out-of-sample test sets. The classifiers attain a classification rate in excess of 90% on out-of-sample test sets for two publicly available data sets. We present an extensive compilation of results reported in the literature for other classification methods run against these same two data sets. Our results are comparable to, or better than, any we have found reported for these two sets, when a train-test protocol as stringent as ours is followed.

1. Introduction

The advent of high-density microarray technology for gene expression profiling on the genomic scale (Schena *et al.*, 1995; Lockhart *et al.*, 1996; DeResi *et al.*, 1997; Brown and Botstein, 1999) has opened new avenues of research in data analysis and knowledge discovery. With the huge quantities of data now being generated, the opportunities, as well as the challenges, appear almost limitless.

Recent literature explores several types of analyses of gene expression data:

- gene clustering, in which subsets of genes exhibiting similar expression patterns across *cases* (e.g., patients, experimental conditions, points of a time-series) are identified (Eisen *et al.*, 1998; Tavazoie *et al.*, 1999; Getz *et al.*, 2000; Rigoutsos *et al.*, 2000; Ben-Dor *et al.*, 2001);
- case clustering, in which sets of cases that exhibit similar gene expression patterns are identified (Alizadeh *et al.*, 2000; Getz *et al.*, 2000; Rigoutsos *et al.*, 2000; Bhattacharjee *et al.*, 2001);

[1] Computer Science Department, University of New Mexico, Albuquerque, NM 87131.

[2] Department of Physics and Astronomy and Center for Advanced Studies, University of New Mexico, Albuquerque, NM 87131.

[3] Department of Pathology and UNM Cancer Research and Treatment Center, UNM School of Medicine, University of New Mexico, Albuquerque, NM 87131.

- case classification, in which the value of one or more attributes external to expression data (e.g., disease subtype, treatment response, prognosis) is predicted from gene expression levels (Alon *et al.*, 1999; Golub *et al.*, 1999; Ben-Dor *et al.*, 2000; Ben-Dor *et al.*, 2001; Khan *et al.*, 2001; Ibrahim *et al.*, 2002; Pomeroy *et al.*, 2002; van't Veer *et al.*, 2002); and

- gene network reconstruction, in which models of the gene regulatory system are built (Friedman *et al.*, 1999; Murphy and Mian, 1999; Tobin *et al.*, 1999; Friedman *et al.*, 2000; D'haeseleer, 2000; Woolf and Wang, 2000; Pe'er *et al.*, 2001). This objective can be viewed as subsuming the others, provided that the external classification variables are included as nodes in the network.

Two factors influence a researcher's focus: the questions of interest in a given setting and the nature of the data sets available. Each of the goals sketched above is of great import, and, in fact, advances in one area often contribute to advances in the others. For example, the identification of strong gene clusters, in addition to indicating potentially significant biological relationships (e.g., co-regulation), in some instances may allow a set of genes to be collapsed into a single abstract unit, thereby reducing problem dimensionality and allowing other objectives to be more successfully addressed.

The data sets available may or may not include information to support classification. Training data that is labeled—associating with each training case the class to which it belongs—supports statistical methods for constructing a classifier. After training on a collection of labeled data, a classifier is constructed which, when presented with new query cases, predicts a class label from gene expression levels and other possibly relevant information which may be associated with a case. Without class-labeled data, genes and cases can be clustered but not classified. Often, however, an effort is made after the fact to construe biological significance for the clusters formed; the success of such clustering methods depends critically on there being a relationship between the measure of similarity used to perform clustering and actual biological similarity. Techniques that attempt to classify after training on labeled data are referred to as *supervised*, while those that do not utilize labels in training (e.g., many techniques for gene and case clustering) are known as *unsupervised*.

Additionally, various amounts of prior information (e.g., expert knowledge, such as previously known or suspected functional relationships) can be associated with gene expression data in an attempt to guide the analysis methods toward better results. Again, the amount of information available—and the degree of belief in this information—determines what information can be utilized and how it can be utilized. Little is understood regarding how such information can best be represented and applied within a rigorous and consistent framework. Such a framework will become of ever increasing importance as our biological knowledge base grows and as our objectives increase in their scope and complexity.

Our group at the University of New Mexico (UNM) is fortunate to have unusually large microarray data sets, with a substantial amount of associated clinical information. This clinical information can be utilized both as additional input and to establish classification criteria. For example, clinical history might be available that allows us to search for correlations between environmental factors and gene expression levels and, ultimately, biological manifestation (e.g., disease). In the realm of classification, we expect to have several interesting class labels to associate with our gene expression data, thus allowing us to explore a variety of supervised classification problems. Information that will be available to us includes disease absence or presence, disease type (e.g., leukemia subtypes), response to treatment, relapse / nonrelapse information, and karyotype.

Consequently, we are motivated to concentrate on the development of methodologies that can exploit the unusually rich amount of information to be associated with our gene expression data and to develop techniques particularly well suited to classification in this context. At the same time, we anticipate soon extending our objectives to include the construction of gene regulatory networks and wish also to be able to integrate in a rigorous way external information, such as prior identification of key controlling genes, causal relationships between genes, and known or hypothesized gene clusters. As is argued in the sections to follow, we believe that the mathematically grounded framework of *Bayesian networks* (Bayesian nets)—for example, Pearl (1988) and Heckerman *et al.* (1995)—is uniquely suited to meet these objectives. Furthermore, the ability of Bayesian nets to integrate prior knowledge with observational evidence potentially provides researchers with the ability to build incrementally solutions to problems of increasing scope and complexity. The primary contribution of the current work is the development of a Bayesian network classification model that is customized for the characteristics of gene expression data. In particular, we propose a Bayesian network structure whose relative simplicity allows the computational effort to be focused on the very high dimensionality inherent in gene expression data. This strategy is designed specifically to exploit certain domain-specific beliefs regarding gene and class label interactions. The initial experimental results reported here bear out the validity of this approach. Further, by operating within the Bayesian framework, the aforementioned capabilities—such as the ability to capture the inter-gene relationships of regulatory networks—remain available to the model in the form of future enhancements.

The remainder of this paper is organized as follows. Section 2 briefly reviews some of the most successful Bayesian network classification methods reported in the literature, details the key elements of our approach, and offers a motivation for our approach in the context of clinical classification from gene expression data. Section 3 presents alternative search methodologies which we have utilized in Bayesian net classifier construction. Section 4 describes our experimental design and Section 5 presents a suite of results. Since we began developing and implementing our techniques prior to the production of microarray data at UNM, the experimental results reported here are against two publicly available Affymetrix data sets:[4]

- MIT leukemia data (Golub *et al.*, 1999), for samples of two types, ALL and AML, of leukemia. This data set is available at `http://www-genome.wi.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43`.
- Princeton colon cancer data (Alon *et al.*, 1999), for normal and tumor tissue samples (available at `http://microarray.princeton.edu/oncology/affydata/index.html`).

For purposes of comparison, an appendix presents an extensive compilation of results reported in the literature for these two datasets, generated using a broad range of classification methodologies.

At the time of this writing, some of the UNM data has begun to become available. As is reported in a series of papers (Mosquera-Caro *et al.*, 2003a; Mosquera-Caro *et al.*, 2003b) our classification methodology continues to perform well on these data sets as compared with other classification methods such as support vector machines (Vapnik, 1998) and discriminant analysis (Bishop, 1996; Duda *et al.*, 2000), though we have discovered that some clinical classification tasks (e.g., prognosis prediction) are inherently more difficult than are such tasks as classification by disease subtype.

2. Bayesian Nets for the Classification of Gene Expression Data

A Bayesian net (Pearl, 1988; Heckerman *et al.*, 1995) is a graph-based model for representing probabilistic relationships between random variables. The random variables, which may, for example, represent gene expression levels, are modeled as graph nodes; probabilistic relationships are captured by directed edges between the nodes and conditional probability distributions associated with the nodes. A Bayesian net asserts that each node is statistically independent of all its nondescendants, once the values of its parents (immediate ancestors) in the graph are known, i.e., a node *n*'s parents render *n* and its nondescendants *conditionally independent*. It follows from these conditional independence assertions and the laws of probability that once a conditional distribution is associated with each node, specifying the probability that the node assumes a given value conditioned on the values assumed by the node's parents, a joint distribution for the entire set of random variables is uniquely determined. Algorithms and software packages (Lauritzen and Spiegelhalter, 1988; Jensen *et al.*, 1990; Shafer and Shenoy, 1990; Dawid, 1992; Dechter, 1996; Madsen and Jensen, 1999; Cozman, 2001; Jensen, 2001) have been

[4] These sets were produced using the analysis algorithms of the Affymetrix Microarray Suite (MAS) Version 4.0. Future data sets will be based on the newer statistical algorithms provided by MAS Version 5.0. See `http://www.netaffx.com/index.affx`.

developed to help the analyst visualize and query Bayesian nets, making this a very convenient representational tool.

While Bayesian nets have found much use as a representational tool for modeling known probabilistic relationships, from the perspective of the gene expression analysis tasks of current interest, their primary utility lies in the fact that they also are a powerful learning paradigm. A body of work has evolved—see, for example, Buntine (1991, 1996), Dawid and Lauritzen (1993), Friedman and Goldszmidt (1996a, 1996b), Heckerman *et al.* (1995), Lam and Bacchus (1994), Pearl and Verma (1991), and Spiegelhalter *et al.* (1993)—in which statistical machine learning techniques utilize a combination of data (observations) and prior domain knowledge to direct a search for Bayesian nets which best explain the current state of knowledge embodied by these inputs. This makes Bayesian nets an attractive framework for gene expression analysis, since they can methodically hypothesize and test gene regulatory models, and other such relationships, using the rigorous methods of classical probability theory and statistics.

Not surprisingly then, others—for example, Friedman *et al.* (1999), Friedman *et al.* (2000), and Pe'er *et al.* (2001)—have successfully applied Bayesian nets to the domain of gene expression analysis. Approaches reported in those works differ from those reported here both with respect to goals (e.g., the identification of gene relationships versus our classification objectives) and with respect to the heuristics employed in an attempt to tame the complexities of the problem. The three cited papers, for example, focus on reconstructing regulatory networks by identifying network relationships most strongly supported by the data and develop heuristics for construction of Bayesian nets that reveal such structure.

The construction of regulatory networks is an eventual goal of our work as well. Hence, the natural applicability of Bayesian networks to regulatory network construction provides one of our motivations for tackling with Bayesian networks the specific problem of immediate interest, clinical classification from gene expression data. The literature contains several different Bayesian network classification models. Friedman *et al.* (1997) describe an approach, Tree Augmented Naive Bayes (TAN), to using Bayesian nets in classification as a way of improving upon the classification approach known as *Naive Bayes* (Duda and Hart, 1973; Langley *et al.*, 1992). Madden (2002) describes a heuristic for building a Markov blanket classifier (see, for example, Cheng and Greiner (1999)) that focuses search on only those network nodes which are relevant to determining the class label's probability distribution, thus making the search over the space of full, unconstrained Bayesian net classifiers more computationally effective. Buntine (1992) develops classification trees in a Bayesian framework. Friedman and Goldszmidt (1996b) and Chickering *et al.* (1997) develop extensions to the Bayesian network model in which local structure between variables can be captured and exploited by importing the structure of decision trees and graphs. To our knowledge, however, these approaches have not been applied in the context of classification problems of such high dimensionality as the problem of clinical classification from gene expression data.

The approach we have chosen to take, rather than starting with these most general and potentially complex Bayesian models that have been developed as general-purpose classification methods, is to attempt to utilize a modest amount of domain knowledge and develop a model that allows the computational effort to be focused where that domain knowledge suggests the most benefit will result. Consequently, a primary contribution of the current work is the development of a Bayesian network classification model that is customized for the characteristics of gene expression data.

The most significant aspects of the customizations presented here involve approaches to cope with the very high dimensionality (i.e, large number of genes, each of which assumes a wide range of values) inherent in gene expression data by exploiting the belief that a relatively small number of genes, *taken in different combinations*, is actually required to predict most clinical classes of interest. This prior belief regarding the nature of these gene expression classification tasks has led us to a rather simple Bayesian network classification structure that, in its initial tests, has performed quite well in comparison with other state-of-the-art learning schemes when applied to several gene expression classification tasks. See the Appendix and (Mosquera-Caro *et al.*, 2003; Kang and Atlas, 2003) for detailed comparisons.

In the following, we introduce our method as an alternative to the existing Bayesian net classifier models, and then briefly contrast the method with the structurally similar methods of Naive Bayes and TAN. We believe this comparison will motivate our approach as a particularly effective, yet equally compact, alternative for problem domains of extremely high dimensionality, such as gene expression data. The experimental results reported in Section 5 bear out the merit of our approach in the gene expression domain. While it appears that the success of our model structure stems from its focusing of the search on those dimensions of the model space from which the greatest gain often is found, our classification model is nevertheless amenable to future extensions with techniques that can utilize and/or discover additional local structure between the genes (Friedman and Goldzmidt, 1996b; Chickering *et al.*, 1997) and to model averaging techniques (for example, Han and Carlin (2000) and Madigan and York (1995)) for augmenting the distribution blending methods presented in Section 3.5.

The work presented here provides an alternative formulation and solution to the classification problem, a formulation which appears to be particularly well suited to classification based on gene expression data. While the focus of our current research is to extend our methods to other biologically important problems, such as the construction of regulatory networks, in this article we do not consider problems beyond classification. In this context, our work is most appropriately compared with other Bayesian network-based classification schemes, such as Naive Bayes, TAN, and Markov blanket classifiers; with other related classification methods, such as Bayesian classification trees; and, in general, with other classification methods, such as support vector machines and boosting.

A Bayesian Net Classification Model for Gene Expression Data

We view each gene as a random variable, with the class label as an additional random variable. The genes assume expression levels (which we shall bin into a small number of distinct values), and the label assumes values such as “cancer” or “no-cancer,” type of cancer, or response to treatment. $< e >$ denotes a vector of expression levels assumed by the set *genes* of all genes in a single case, and c_k denotes a value assumed by the class label. The classification problem can be stated as learning the posterior conditional distribution of the class label C, conditioned on the gene expression levels, that is, the collection of conditional probabilities

$$Pr\{C = c_k \mid genes =< e >,\ current\ knowledge\},$$

one for each c_k and $< e >$ combination.

The *current knowledge* appearing in the conditioning event of the above probability generally includes both a training set of cases and prior distributions over the random variables. These prior distributions may capture, for example, prior beliefs regarding biological mechanisms. From this perspective, classification can be solved as a problem of statistical density estimation. After viewing the training set—a sample of vectors of expression values with an associated class label, drawn from the same distribution as the query cases we later will be asked to classify—we apply laws of probability to update our priors and “learn” this common distribution. We then are able to estimate the probability that query q’s class label $q[C]$ is c_k, given that q’s expression vector $q[genes]$ is $< e >$.

The main difficulty in this learning problem is that the huge dimensionality of $< e >$ implies that any realistically-sized sample will provide only extremely sparse coverage of the sample space. For example, even if continuous expression levels are partitioned into 2 or 3 discrete bins, each of the $number\_of\_bins^{number\_of\_genes}$ combinations of (binned) expression levels of the several thousand genes which appear in the training data typically appears only once, and combinations in the query cases typically have not appeared at all in the training data. Consequently, estimation of the conditional distributions from simple joint frequencies observed in the sample is impossible.

We consider Bayesian nets in which each gene is a node, and the class label is an additional node having no children. Associated with each node n is a conditional distribution, a set of $\theta_{n=v,par=<p>} \equiv Pr\{n = v \mid Par(n) =< p >\}$, specifying a conditional probability for each value v of n, conditioned on each combination of values $< p >$ of the parents of n. Note that a Bayesian net is a pair (G,Θ), where G is a directed acyclic graph (DAG), and Θ supplies a conditional probability $\theta_{n=v,par=<p>}$ for every node value, parent set-combination implied by

G. Such a pair (G, Θ) compactly encodes a unique joint distribution over the nodes of G; this joint distribution $Pr\{genes = < e >, C = c_k\}$, and any conditional distribution over the random variables represented by the nodes, can be recovered via various known graph traversal algorithms (Lauritzen and Spiegelhalter, 1988; Jensen *et al.*, 1990; Shafer and Shenoy, 1990; Dawid, 1992; Dechter, 1996; Madsen and Jensen, 1999; Cozman, 2001; Jensen, 2001).

If we had a fixed Bayesian net that encoded the true distribution from which each case is drawn, we could extract a classifier, namely the subgraph defined by the class label node C and its parent set $Par(C)$, along with the associated conditional distributions $\theta_{C=c_k, par=<p>} = Pr\{C = c_k \mid Par(C) = < p >\}$. Note that the conditional independence assertion associated with (leaf) node C implies that the classification of case q depends only on the expression levels of the genes in $Par(C)$, i.e., the distribution $Pr\{q[C] \mid q[genes]\}$ is identical to the distribution $Pr\{q[C] \mid q[Par(C)]\}$. Note, in particular, that the classification does not depend on other aspects (other than the parent set of C) of the graph structure of the Bayesian net. Hence, once given a parent set, density estimation becomes far more tractable. Rather than being concerned with combinations of all the genes, we are concerned only with combinations of the parent set, and hence a training sample will generally provide much better coverage of this reduced space.

Given a fixed Bayesian net of the form described, the classification rule it induces is simply a table associating values $< p >$ of the parent variables and $C = c_k$ of the class label with the network's induced conditional probability $\theta_{C=c_k, par=<p>}$. However, we are not given the "true" Bayesian net, but, rather, a collection of training cases, plus, possibly, some accumulation of prior knowledge, and our task is to build a classifier to fit these data. How the classifier is constructed to fit the data is what primarily distinguishes methods, ultimately determining success or failure. Two central aspects of the construction enabled by operating within the Bayesian framework are:

> *The use of a Bayesian metric in controlling the complexity of the classification rule*. While it often is observed that tables representing complex classification rules (complex conditions over many attributes) overfit to the training data, our use of the *BD* metric (Heckerman *et al.* 1995) as described in Section 3.1 balances in a principled way the gain in adding new conditions with the complexity of the rule. MacKay (1995) has formalized how a Bayesian metric inherently embodies Occam's razor, favoring simple rules unless sufficient gain in information is realized by the addition of conditions. Hence, a stopping condition for rule refinement is not *ad hoc*, but part of the Bayesian metric evaluation.[5] Further, when prior information is available, it is incorporated naturally into the evaluation. It also is possible to incorporate into the model local structure, as described in Friedman and Goldzmidt (1996b) and Chickering *et al.* (1997). While this has not been necessary for the classification tasks undertaken to date, future work will explore the utility of this model extension in the gene expression domain.
>
> *The blending of distributions in a principled way*. As is detailed in Section 4, we search a space of networks for a collection of networks that score well under the *BD* metric. If we chose the single best scoring network (the maximal posterior method) we would utilize as our posterior the single conditional distribution this network induces. Our approximation of the posterior is more sophisticated in that it is capable of blending the distributions of a possibly large number of networks. The blending is based directly on the mathematics of the Bayesian analysis. In one of our two search methods, the blending is over the highest *a posterior* probability networks of an *exhaustively* searched model space. In the second search method, a larger model space is sampled by means of a greedy search. Future extensions to the sampling methods utilizing MCMC averaging techniques (Han and Carlin, 2000; Madigan and York, 1995) would be quite natural.

Comparison With Existing Bayesian Net Classification Models

[5] Similarly, Buntine (1992) applies a Bayesian metric in connection with classification tree construction; alternatively, the *MDL* evaluation criterion—which includes an explicit penalty term for model complexity—has been used quite successfully in Bayesian network learning (Friedman *et al.*, 1997). In a separate work (Ding, 2003), we are comparing the *MDL* and *BD* metric in the gene expression domain.

As was reviewed earlier in this section, several Bayesian network classification models have been proposed. In terms of its simplicity of structure, our model most resembles Naive Bayes and its generalization known as TAN. However, unlike these existing models, our model was conceived specifically to address key characteristics of the gene expression application domain. In particular, our model is customized to application domains having very high dimensionality (e.g., many genes), while at the same time exhibiting a dependency structure which implies that a relatively small number of features, *taken in different combinations of several features at a time*, is required to predict the class label of interest. These are characteristics consistent with prior experience with gene expression data and which translate to dependency structures which Naive Bayes or TAN are incapable of capturing. After contrasting our model with Naive Bayes and TAN, we briefly consider the potential for extending our model with techniques that have proven successful in other application domains.

A Naive Bayesian classifier (Duda and Hart, 1973; Langley *et al.*, 1992) assumes independence of the features (genes), given the value of the class label. Under this assumption, the conditional probability $Pr\{q[C] \mid q[genes]\}$ can be computed from the product $\prod_{g_i \in genes} Pr\{q[g_i] \mid q[C]\}$ of the marginal conditional probabilities. The Naive Bayesian model is equivalent to a Bayesian net in which no edges exist between the genes, and in which an edge exists from the class label into each gene. Friedman *et al.* (1997) introduces Tree Augmented Naive Bayes (TAN), which relaxes somewhat the independence assumption of a Naive Bayesian classifier by allowing each gene to have an incoming edge from at most one other gene, while maintaining an edge from the class label into each gene. This approach yields good improvements over Naive Bayesian classifiers in the experiments—which are over application domains other than gene expression data—reported in Friedman *et al.* (1997).

By contrast, our modeling assumes neither an edge between each gene and the class label, nor concerns itself with gene interaction. Rather, we are able to ignore the issue of what edges may exist between the genes, and compute $Pr\{q[C] \mid q[genes]\}$ as $Pr\{q[C] \mid q[Par(C)]\}$, an equivalence that is valid regardless of what edges exist between the genes, provided only that $Par(C)$ is a set of genes sufficient to render the class label conditionally independent of the remaining genes. This modeling is in response to a prior belief (supported by experimental results reported here and in other gene expression analyses) that for the gene expression application domain, only a small number of genes, *taken in combination*, is necessary to render the class label (practically) conditionally independent of the remaining genes. This both makes learning parent sets $Par(C)$ tractable, and generally allows the quantity $Pr\{q[C] \mid q[Par(C)]\}$ to be well estimated from a training sample.

Each of these two simple models for Bayesian network classifiers—TAN and the model presented here—has advantages over the other in certain situations. Specifically, because our parent sets, in principle, allow an arbitrary number of genes to interact in combination, any conditional distribution for the class label can be modeled exactly. This is in contrast to TAN, where no term of the joint probability distribution may involve combinations of more than two genes. (Terms of the joint distribution, expanded according the conditional independence assertions implied by any TAN network, are of one of the following three forms: $P\{C\}$, $P\{g \mid C\}$, or $P\{g \mid C, g'\}$.) Consequently, it is a simple matter to identify families of underlying distributions over n random variables (for any $n \geq 3$) where every TAN network is necessarily asymptotically incorrect, while instances of our model are asymptotically correct. That is, for these families of distributions, as the sample grows to reflect the actual distribution, any TAN network misclassifies some (possibly large) fraction of the query cases, whereas our model approaches perfect classification when all the relevant variables are included in the parent set. In practice with gene expression data, it has been our experience that, typically, combinations of between 2 and 5 binary-binned genes determine the class label—that is, render the class label conditionally independent of the other genes—while combinations of up to 5–7 binary-valued genes can reasonably be evaluated (for example, by the *BD* metric) and distributions learned from data sets of the sizes with which we have been working.

TAN's advantage may be seen when the sample is sparse relative to the number of genes necessary to render the class label approximately conditionally independent of the other genes. In such a case, if the true distribution obeys or approximates the conditional independence assertions of TAN, the TAN model is capable of learning the correct distribution, and will converge to this distribution faster as a function of sample size than will our model.

Our network blending (see Section 3.5) can somewhat mitigate the problem for some distributions, and, further, in some instances it may be desirable to augment our model with local structure, allowing our density estimates to converge to the true distribution even for sparse samples. (Note that the incorporation of local structure would not address the inaccuracies of TAN when its conditional independence assertions are violated.)

One can envision a hybrid search, where the *BD* metric evaluates the fit of networks from both classification models, choosing the best fitting model, or possibly even blending their distributions. In the limiting case, of course, one could consider unconstrained and full Bayesian nets, using the Markov blankets they define as the classifier (Cheng and Greiner, 1999; Madden, 2002). While this is the most general of the modeling approaches, it is very much an open question (especially for applications with the characteristics described here) whether or not the gain in modeling generality is actually an advantage, given the fact that the search over the more constrained network space implied by our model structure (possibly combined with TAN structures) may focus on that task—construction of good parent sets, expected to be of small cardinality—most likely to determine classifier quality. Similarly, it is not clear whether, in the gene expression domain, the diversion of search time to include consideration of local structure would generally be beneficial or not. As indicated, current gene expression data sets do yield sufficient coverage of the small number (e.g., often less than 5) of binary-binned genes that our experience indicates are required for the class label's parent sets, and focusing search on the selection of such subsets of genes often may be the most fruitful utilization of search time. Future experiments will explore these issues further.

3. Additional Model Details

Our approach requires that we address the following issues, which are considered in this and the sections to follow.

- What does it mean for a Bayesian net to be plausible?
- What do we do with multiple plausible Bayesian nets?
- How do we find (the parent sets $Par(C)$ in) plausible Bayesian nets?
- How do we specify prior distributions?
- How do bin the continuous gene expression data?
- How do we preprocess (e.g., normalize) the gene expression data?

3.1. Scoring the Nets

The derivations in this and the following sections summarize and adapt to our context the work appearing in Heckerman *et. al.* (1995), and we implicitly accept the set of assumptions made there.

Bayesian net structures are hypotheses. Each network structure G hypothesizes a collection of conditional independence assertions. Were hypothesis G true with probability 1, the assertions it encodes, plus the priors and observations D, would induce via the laws of probability a posterior distribution $f(\Theta \mid G, D, prior)$ over the space of conditional distributions for G, where each Θ in the space contains conditional distributions $\theta_{n=v,par=<p>}$ for each node n in G. Of particular interest are expectations under this distribution of the form

$$E(\theta_{n=v,par=<p>} \mid G, D, prior) = \int f(\Theta \mid G, D, prior) \times \theta_{n=v,par=<p>} d\Theta,$$

as this is $Pr\{n = v \mid Par(n) =< p >, G, D, prior\}$. For classification, of course, the desired quantity is

$$\begin{aligned} & E(\theta_{C=c_k,par=<p>} \mid G, D, prior) \\ & \quad = \quad Pr\{C = c_k \mid Par(C) =< p >, G, D, prior\} \\ & \quad = \quad Pr\{C = c_k \mid \ < e >, G, D, prior\}, \end{aligned}$$

for any full expression vector $< e >$ whose projection onto the parent set $Par(C)$ is $< p >$. (Recall that class label C is constrained to have no children in the network.)

In a learning context, we generally never obtain a single net structure G with certainty, but rather obtain a collection of plausible G_i. Therefore, it is desirable to employ a probabilistically-based scoring function, both to guide our exploration of nets, and to specify how to blend the distributions they induce. In a Bayesian framework, one scores how well a hypothesis G_i fits $\{D, prior\}$ by computing

$$Pr\{D \mid G_i, prior\} = \int Pr\{D \mid \Theta\} \times f(\Theta \mid G_i, prior) d\Theta.$$

Then, from priors $P(G_i)$ over network structures, we can obtain $Pr\{G_i \mid D, prior\}$. Such a scoring function is known as a *Bayesian metric*.

If we evaluated all possible structures G_i in this manner, the posterior distribution over joint distributions Θ_j of the nodes in the networks is computed by

$$f(\Theta_J \mid D, prior) = \sum_{G_i} f(\Theta_J \mid G_i, D, prior) \times Pr\{G_i \mid D, prior\}.$$

The classification probabilities

$$Pr\{q[C] = c_k \mid q[genes] =< e >, D, prior\}$$

of interest then are the expectations

$$E(\theta_{q[C]=c_k, q[genes]=<e>} \mid D, prior) \tag{1}$$

under this distribution and are obtained as a weighted sum of expectations, namely

$$\sum_{G_i} E(\theta_{q[C]=c_k, par=<p>} \mid G_i, D, prior) \times Pr\{G_i \mid D, prior\}, \tag{2}$$

where each parent vector $< p >$ is the projection of $< e >$ onto the parent set par of C in each G_i. That is, the probability each G_i assigns to $q[C]$ given $q[genes]$ is weighted by the posterior $Pr\{G_i \mid D, prior\}$. In principle, if we could evaluate this sum over all G_i, we would have an exact posterior—and hence classifier—given the current state of knowledge represented by our priors and the observed cases. The more peaked is the distribution $Pr\{q[C] = c_k \mid q[genes] =< e >, D, prior\}$ about its mode class c^*, the higher is the probability that the classification provided for query q is correct.

3.2. Computational Considerations

Our task can be viewed as approximating expression (2) by finding a set of nets whose respective contributions dominate (e.g., because they have relatively high posterior weights $Pr\{G_i \mid D, prior\}$) the evaluation of this sum. Some empirical studies (Cooper and Herskovita, 1992; Heckerman *et al.*, 1995) indicate that, in a variety of contexts, only a relatively small number of the nets considered (e.g., often 1) have weights large enough to materially influence the evaluation, since the weights drop off quickly as edges which represent necessary dependencies are omitted or edges which represent unnecessary dependencies are added. The experimental results reported in Section 5 explore the effect of varying the number of nets used in this approximation. One important conclusion we draw is that, in the context of high-dimensional gene expression data, the inclusion of more nets than is typical appears to yield better results. Our experiments indicate this to be the case both because the "polling" provided by a large number of nets is more accurate than that provided by a small number, and because a large number of nets often provides better coverage of the expression value combinations observed in the query cases (that is, the inclusion of more nets increases the chances that query *q*'s binned expression levels projected onto some included parent sets have been observed in the training sample).

On the surface, the evaluation of even a single *G* seems a formidable task; both the expectations (1) and the Bayesian metric require an integration over potentially arbitrary distributions for Θ. However, following the work of Heckerman *et al.* (1995), we assume that a prior distribution is specified in terms of a complete net and is Dirichlet. Intuitively, such a prior can be equated with an imaginary sample of joint observations of the random variables that represents the analyst's beliefs—both in terms of relative frequency counts (corresponding to prior probabilities) and absolute size (corresponding to degree of belief)—prior to observing the sample cases. This prior distribution on the nodes of a complete net induces on the nodes of any net a unique prior distribution consistent with a modest set of assumptions. Then, for any *G* and this induced prior distribution, plus a set of observed cases, the calculations reduce to a closed form.

In particular, the closed form for the expectation is

$$\begin{aligned} & E(\theta_{n=v,par=<p>} \mid G, D, prior) \\ = \quad & \int f(\Theta \mid G, D, prior) \times \theta_{n=v,par=<p>} d\Theta \\ = \quad & (\alpha_{pv} + N_{pv})/(\alpha_p + N_p), \end{aligned} \tag{3}$$

where N_p is the number of cases observed in *D* in which $Par(n) =< p >$; N_{pv} is the number of cases observed in *D* in which $Par(n) =< p >$ and $n = v$; and α_p and α_{pv} are derived from prior probabilities for these combinations of values and, under our prior assignments, are extremely small (see Section 3.3 and Heckerman *et al.* (1995)). The closed form for the Bayesian metric is

$$\begin{aligned} & Pr\{D \mid G, prior\} \\ = \quad & \int Pr\{D \mid \Theta\} \times f(\Theta \mid G, prior) d\Theta \\ = \quad & \prod_n \prod_p \frac{\Gamma(\alpha_p)}{\Gamma(\alpha_p + N_p)} \prod_v \frac{\Gamma(\alpha_{pv} + N_{pv})}{\Gamma(\alpha_{pv})}, \end{aligned}$$

where

Γ is the Gamma function;

n ranges over the nodes in *G*;

p ranges over values $< p >$ of $Par(n)$ for the node *n* fixed by the outermost $\prod$;

v ranges over the values of the node *n* fixed by the outermost $\prod$; and

$\alpha_p, \alpha_{pv}, N_p, N_{pv}$ are as defined above, with respect to the node *n* fixed by the outermost $\prod$.

The above expression for $Pr\{D \mid G, prior\}$, which assumes a Dirichlet prior, is known as the *BD (Bayesian-Dirichlet) metric*. (Technically, the *BD* metric is more commonly defined in terms of the joint posterior probability $Pr\{D,\ G \mid prior\}$, which is simply the above expression multiplied by the network prior $P(G)$.)

Further simplifying the computational task is the observation that the scoring function is decomposable; it can be expressed as the product of scores over the nodes, where a node's score depends only on its parent set. In our restricted context of classification, this means we can ignore the score of every node except the label, effectively using the *BD* metric as an evaluator of potential parent sets. More precisely, the *BD* evaluation of a parent set $Par(C)$ is node *C*'s contribution to the *BD* score of *any* Bayesian net containing this subgraph. In particular (in contrast to a Naive Bayesian classifier, in which there must be no edges between genes), the decomposability of the *BD* score allows the hypothesis represented by parent set $Par(C)$ to be evaluated in isolation of the question of what other edges may exist in the network. Similarly, since the expectation of interest depends only on frequencies of node *C* and of its parent set, the remainder of the network can be ignored in our context.

3.3. Specification of Priors

In each of the experiments reported, we choose an uninformed prior over the distributions that can be associated with any given network structure. In particular, we employ an extremely small equivalent sample size (Heckerman *et al.*, 1995) of 0.001, and assign each joint combination of variable values equal probability. There then is a simple translation of this prior to priors over the possible conditional distributions in any given network structure, yielding the α_{pv} and α_p values appearing in expression (3). Our choice of prior minimizes its impact on posterior calculations, allowing the data to dominate.

The network structures *G* are assigned a uniform prior also, but after various prunings (see Section 4) have been imposed. In the context of our minimal-knowledge greedy algorithm, a prior which assigns equal probability to each DAG in which the class label has $\mathcal{M}$ or fewer parents (and zero probability to all other DAGs) is used, for some specified maximum cardinality choice $\mathcal{M}$. In the context of the external gene selection algorithms, a prior which assigns equal probability to each DAG in which the class label has $\mathcal{M}$ or fewer parents, each of which is a member of the selected set of genes (and zero probability to all other DAGs), is used.

Current research is considering how various types of expert biological information can be incorporated into priors and utilized by our methods. This is an area we believe to be critically important to future advances.

3.4. Binning Issues

Though Bayesian nets can be utilized to represent continuous distributions, most Bayesian net procedures assume that the random variables take on only a small number (e.g., 2 or 3) of discrete values. This requires procedures to discretize (i.e., collapse) typically continuous gene expression values. We describe in Section 4 the two relatively simple approaches we have used with our current search procedures. The first method bins expression values into "low," "medium," and "high" based on the distance of a particular expression value from the gene's mean expression value. The second method is more closely coupled with our external gene selection method and produces a binary binning based on a maximal "point of separation" in the training data between the classes.

While these simple methods have produced good classification results, we point out here that there are many interesting avenues of research in which the binning procedure is more integrated with the search for good Bayesian nets, and candidate binnings are evaluated in the same framework as are other aspects of the nets (see, for example, Fayyad (1993) and Friedman and Goldszmidt (1996a)). We consider this to be an important avenue for future research.

3.5. A Multi-Parent-Set Classifier

We have indicated how a parent set of the class label corresponds to the relevant (for classification) subgraph of a Bayesian net and, with expression (2), how the class distributions associated with each parent set in a collection of parent sets are combined by means of the *BD* scoring metric. Our method then is to build a classifier from some number $\mathcal{PS}$ of parent sets that score high under the *BD* metric. That is, we perform some form of search (see the next section), selecting the $\mathcal{PS}$ top scoring parent sets, and these are the sets whose distributions contribute the terms for our approximation of the expression (2). We see from expression (3) that the individual probabilities contributed are simply of the form $(\alpha_{pv} + N_{pv})/(\alpha_p + N_p)$.

An important phenomenon results from the sparseness of the data, especially in the high dimensional space of microarray data. It is possible that the combinations of values appearing in $q[par_i]$ for some of the parent sets par_i are not seen in training or seen only minimally (for example, one or two occurrences). The distributions yielded by such nets will then reflect only the prior, which (as we shall generally assume) is uninformed, yielding equal class probabilities, or will be determined by the handful of training cases with this par_i combination. It is important to note that this is the correct posterior distribution under the hypothesis of this parent set and given current knowledge and should not be interpreted as a “weak” or “missing” distribution simply because it is based on a small or empty sample. The strength of this distribution as it contributes to (2) is determined solely by the *BD* fit. A dispersed distribution (e.g., uniform) learned from a small sample and a peaked distribution learned from a large sample contribute their expectation in the same way, their relative contributions to the posterior affected only by their *BD* fit.[6]

Is it appropriate to treat the sparse-sample based distributions on equal footing with large-sample based distributions? We consider the variance of the distribution. Variance reflects, among other characteristics, how much the distribution may be expected to change if more data is observed. In the case of high variance, it is not unlikely that new data will shift the distribution dramatically.

The variance of the posterior $Pr\{C = c_k | Par(C) =< p >, G, D, prior\}$ of a binary-valued class label, being a Dirichlet distribution, is

$$(Pr\{C = c_k | Par(C) =< p >\} \times (1 - Pr\{C = c_k | Par(C) =< p >\}))/(\alpha_p + N_p + 1).$$

So, an interpretation is, when the “sample size” N_p is small, or when the probability is spread evenly across the classes, variance is relatively high, and the distribution is possibly “unstable” in the presence of additional observations. While the posterior distribution it yields is undeniable given the current state of knowledge, it is not unlikely to change dramatically given new data. In this sense, it is less “reliable”.

We have experimented with two heuristics for adjusting a parent set’s contribution to the evaluation of a query case in order to address the issue of the variance of the distribution. Note that unlike a set’s *BD* score, which is used in parent set selection as well as for a weight in the posterior computation (2), this adjustment is *query specific*, reflecting the amount of variance $var(q)$ in the distribution of a particular query q’s (unknown) label. The two adjustments considered are:

- When evaluating a query q, set to zero the weight in (2) of any parent set par_i such that $q[par_i]$ has no occurrences in the training sample. Then renormalize the remaining *BD* weights to sum to 1.
- Generalize the above so that $1/var(q)$ is the adjustment factor of each set par_i, and then renormalize $BD/var(q)$.

[6] Though peaked distributions which fit a large sample well tend to have better scores than dispersed distributions that fit small samples well.

A variant of the second adjustment strategy, in which an adjustment factor of zero is used when N_p is zero, improved performance in cross validation experiments on the gene data training sets by preventing a large number of parent sets, each yielding few observations on a query case, from unduly influencing the classification. This method is what is used in the experiments reported in this paper. More sophisticated adjustments tied to Bayes risk are the subject of current research.

4. Search

The research presented in the following sections explores two alternative methods of building the type of Bayesian classifier described in the previous sections.

The first method utilizes minimal prior knowledge regarding good parent sets for the class label and, within the Bayesian net framework, performs a simple greedy search over the entire set of genes to construct $\mathcal{PS}$ good parent sets. The second method utilizes gene selection external to the Bayesian net framework to produce a small set S of "good genes" (like the *informative genes* of Ben-Dor *et al.* (2000) and Ben-Dor *et al.* (2001)), and then, within the Bayesian net framework, performs an exhaustive search of this set to find the best $\mathcal{PS}$ subsets of S (each subset up to a specified maximum cardinality $\mathcal{M}$).

4.1. Minimal-Knowledge Greedy Building Methods

This family of methods ignores essentially all prior knowledge, including, in the experiments reported here, prior knowledge of which genes are "control" or "housekeeping" genes, which expression values are deemed reliable (in particular, as indicated by the P, M, and A values in Affymetrix data), and biologically known relationships between genes. We do utilize a biological "prior" that deems it likely that only a small number of genes is necessary to classify the cases, that is, that only a small number of genes is required to render the class label conditionally independent of the remaining genes. This biological prior is necessary for any frequency-based classification method to go forward, due to sample size issues, and makes both the greedy and exhaustive searches computationally feasible. This prior is in fact supported by experiments with the current data sets in which performance—both *BD* and our actual classification rates—begins to diminish after a cardinality of roughly > 6. This is not quite conclusive proof, as improvement might follow disimprovement (e.g., as is exploited by simulated annealing), but this seems unlikely, especially in light of sample size issues (e.g., statistically meaningful numbers of observations of any combination of more than six gene's expression levels is unlikely).

The version of greedy employed here proceeds in the following manner. On a designated training set (see details of the methodology in Section 5.1):

1. Use some algorithm to bin the gene expression data.

2. Determine a number $\mathcal{K}$ of seeds, a number $\mathcal{PS}$ of parent sets, and a maximum cardinality $\mathcal{M}$ for the parent sets.

3. Select $\mathcal{K}$ seed genes, based on some "goodness" criterion.

4. For each seed gene g_{seed},

   a. Initialize the parent set to the singleton set $\{g_{seed}\}$. Consider the parent set $\{g_{seed}\}$ for inclusion in the list of the best $\mathcal{PS}$ parent sets evaluated so far.

   b. Iteratively build the set to cardinality $\mathcal{M}$ by adding one gene g at a time, chosen from the universe of all genes to maximize the *BD* score of $\{\text{current set}\} \cup \{g\}$. Consider each such parent set $\{\text{current set}\} \cup \{g\}$ for inclusion in the list of the best $\mathcal{PS}$ parent sets evaluated so far, resulting in the inclusion of zero or more of these parent sets $\{\text{current set}\} \cup \{g\}$. The single best of these extensions to the previous $\{\text{current set}\}$ then becomes the new current parent set and is similarly extended at the next iteration. Continue iterating until parent sets of cardinality $\mathcal{M}$ genes are evaluated and considered for inclusion in the list of the best $\mathcal{PS}$ parent sets evaluated so far.

5. Construct a $\mathcal{PS}$-parent-set Bayesian net classifier from the list of selected parent sets (each of cardinality between 1 and $\mathcal{M}$) as described in Section 3.5.

In Section 5.1, we specify the binning and seed selection methods used in the experiments reported in this paper.

Note that every set the greedy method evaluates, starting from each of its seeds, is a candidate for ultimate selection as one of the $\mathcal{PS}$ parent sets—even those sets of smaller than the maximum cardinality $\mathcal{M}$. In particular, at every iteration, in going from cardinality c to $c+1$, every extension of the best parent set of cardinality c gets a chance to be on the list of top parent sets. Consequently, some seeds may contribute more than one parent set; others may not contribute any parent sets at all.

This simple greedy method was implemented initially as a proof of concept; we suspected it would have many flaws and that we would soon replace it with more sophisticated search methods. However, it performed surprisingly well, as is attested to by both the *BD* scores of the best sets it finds and by the performance on our cross validation tests of the classifiers it produced (see the results in Section 5). This is not to say that avenues of potential improvements are not apparent. For example, there often is a great deal of overlap in the membership of the parent sets produced. Two or three genes tend to be present in a large fraction of the $\mathcal{PS}$ parent sets selected. This is not necessarily a problem, but it might indicate that a nonrepresentative subspace of the set of all possible parent sets is being searched. As is discussed in Sections 5.2 and 5.4, this effect could explain why a relatively small number of high quality parent sets are found by the algorithm.

An alternative heuristic search would mimic classical integral approximation techniques (Gander and Gautschi, 2000). In a similar learning context (Helman and Bhangoo, 1997; Helman and Gore, 1998), we employ with some success a Monte Carlo sampling method to approximate an integral representing Bayes risk. Such methods are designed to approximate an integral by sampling from regions in proportion to the amount of density contained in the region and may be adaptable to the current approximation problem. Additionally, we will consider utilizing the more sophisticated MCMC averaging techniques (Han and Carlin, 2000; Madigan and York, 1995) in this context.

4.2. External Gene Selection Methods

A second family of methods utilizes gene selection algorithms that have been developed in other contexts. This is both a promising approach to the classification problem and is indicative of how the Bayesian framework can be used to incorporate expert prior knowledge of a variety of types. As is the case with the minimal-knowledge greedy methods, we currently do not utilize prior domain knowledge about the genes; such information may, however, be discovered by our external gene selection and normalization methods and then incorporated into the framework in the form of gene selections, normalization, and binning.

The objective of external gene selection is to identify a small set of genes from which good parent sets can be constructed within a Bayesian net search procedure. By severely limiting *a priori* the size of the universe of genes to be searched for good parent sets and the maximum cardinality of the resulting parent sets, an exhaustive search for the $\mathcal{PS}$ *best* parent sets (under the *BD* metric) can feasibly be performed. Thus, whereas the greedy method described in the previous section heuristically builds $\mathcal{PS}$ *good* subsets of the universe of all genes, the external method finds the $\mathcal{PS}$ *best* subsets of an intelligently restricted universe of genes.

We are studying a number of different methods for selecting genes whose expression values are strong indicators of a case's classification. The results reported in this paper are based on a strategy that computes a *separation quality* value for each gene and orders the genes accordingly. We then, for example, can select the genes that are the best separators.

Our separation measure is similar to Ben-Dor's *TNoM* score, described in Ben-Dor *et al.* (2000) and Ben-Dor *et al.* (2001). Both methods consider partitionings of the cases into two sets; the difference between the two methods is in how the partitions are evaluated. Where $TNoM$ compares the *number* of cases from each class in each of the two partitions, we account for the sizes of the two classes by comparing the *fraction* of cases from each class. The two methods can result in different gene selections, and we claim that the relative score is well justified when, for example, the underlying classes differ significantly in size. We have experimented with both measures and don't find either to be uniformly better than the other. Consequently, for a given application, we currently allow experimental cross validation results against a training set to guide our choice of measure.

Let $E_1, E_2, ..., E_n$ be the expression values for a given gene across the n cases of a training set, and let $L_1, L_2 ... L_n$ be the corresponding class labels. Without loss of generality, we assume that the expression values are ordered $E_1 \leq E_2 \leq ... \leq E_n$ so that L_i is the class label of the i^{th} smallest expression value. The separation quality value of a gene is intended to indicate to what extent identical class labels are grouped together in $L_1, L_2, ..., L_n$ as a consequence of the ordering of the E_i values. Separation is considered to be perfect, for example, if the L_i labels are completely "sorted".

Under the assumption that there are exactly two class labels, A and B, we compute separation quality as follows. Let $Acount(i)$ be the number of A labels in $L_1, L_2 ..., L_i$, and let $Bcount(i)$ be the number of B labels in $L_1, L_2 ..., L_i$. For each position $0 \leq i \leq n$, we can quantify the relative separation of the class labels if we were to split into the two sets $L_1, L_2, ..., L_i$ and $L_{i+1}, L_{i+2}, ..., L_n$:

$$Separation(i) = \left| \frac{Acount(i)}{Acount(n)} - \frac{Bcount(i)}{Bcount(n)} \right|$$

We then define separation quality to be the best of these values:

$$SeparationQuality = \max_{1 \leq i \leq n} Separation(i)$$

Genes can be ordered by their $SeparationQuality$ values, so we can talk about the k best or the k worst separators.

The computed values have the following properties.

- $Acount(0) \ = \ Bcount(0) \ = \ 0$
- $Bcount(i) \ = \ i \ - \ Acount(i)$, for $0 \leq i \leq n$
- $Separation(0) \ = \ Separation(n) \ = \ 0$
- $SeparationQuality \ = \ 1$ indicates perfect separation.
- $SeparationQuality$ necessarily is $> \ 0$, since $Separation(1)$ is $1/Acount(n)$ or $1/Bcount(n)$, depending on whether L_1 is A or B, and we take the maximum of the $Separation$ values.
- We get the same $SeparationQuality$ value if we define $Acount$ and $Bcount$ in terms of $L_{i+1}, L_{i+2}, ..., L_n$ instead of $L_1, L_2 ..., L_i$.

We note that if the gene expression values are not distinct, then the ordering of E_i values is not unique, and the computed separation quality value will depend on the procedure used to break ties. We are considering a number of ways to pin down the ordering in the case of ties—specifically, to determine an appropriate separation quality value. We currently break these ties arbitrarily.

In addition to computing a separation quality value, we can use the same computation to propose a binning of each gene's expression values into two bins. Let max be the i value that maximizes $Separation(i)$, and compute

$$BinValue = \frac{E_{max} + E_{max+1}}{2},$$

which is a gene expression value that lies between the separated E_i values in the best separation. The computed *BinValue* can be used as a boundary between bins.

We note that the maximizing i value is not necessarily unique, even if the E_i values are distinct; we currently break these ties arbitrarily. We also note that L_{max} and L_{max+1} necessarily are different labels; otherwise, $SeparationQuality$ could be increased by increasing or decreasing *max* by 1.

This binning strategy is motivated by a prior belief, shared by many domain experts, that a gene is well-modeled as a binary switch. This belief appears to be supported by preliminary analyses against the data sets considered here, as well as against additional data sets (Mosquera-Caro *et al.*, 2003a; Mosquera-Caro *et al.*, 2003b) reported elsewhere. The method whose analyses are for setting bin boundaries described above is quite natural, as it selects a point yielding bins that maximally distinguish the classes (with respect to SeparationQuality), and thus is highly analogous to the boundaries suggested by Ben-Dor's $TNOM$-based method. The binning procedure also is similar to the initial binning policy of the procedures described in Fayyad (1993) and Friedman and Goldszmidt (1996a)—though they consider a variety of policy evaluation metrics—in which an initial binary binning is heuristically refined into a finer-grained discretization. We have conducted extensive experiments (Ding, 2003) using each of the *MDL* and *BD* evaluation criteria to determine stopping conditions for refinement, and, for a large fraction of genes (i.e., $>$ 90% of the genes on the Princeton data set), refinement of the initial binary binning is not supported by these measures. Section 5.1 describes an alternative tertiary binning strategy we considered in the context of the uninformed greedy method.

4.3 Preprocessing the Data (Normalization)

One of the advantages of the Bayesian approach is that it provides a natural mechanism to account for special domain knowledge in the construction of a classifier. Nevertheless, in our first round of experiments, we are focusing on the gene expression data, making use of minimal prior knowledge. One of the issues we are addressing in this simplified context is the preprocessing (normalization) of gene expression data before the application of our classification procedures. Because of variabilities in gene expression measurements and uncertainties about the processing done by the tools used to generate the data,[7] we decided to include the effect of normalization as part of our studies. Specifically, for each data set we study, we attempt to learn via cross validation the most effective of a family of normalization parameters.

Our approach to normalization is to consider, for each case, the average expression value over some designated set of genes, and to scale each case so that this average value is the same for all cases. This approach allows our analysis to concentrate on relative gene expression values within a case by standardizing a reference point between cases. For example, if the expression value within a case of certain genes g_i *relative* to the expression value of some reference point gene g is an effective class discriminator, then it suffices simply to consider these g_i values, provided cases have first been normalized to a common g value. The key difference between the normalization strategies we considered is the choice of the reference point gene g, or, more generally, the choice of a set R of reference point genes. While selecting an appropriate set R could provide a good opportunity to take advantage of special knowledge of the underlying domain, consistent with our desire to focus first on raw data in the absence of prior knowledge, we use here a simple selection method based on the $SeparationQuality$ value already discussed. In particular, we set R to be the k *worst* separators—that is, genes with the lowest $SeparationQuality$ values—for some number k. The motivation for this choice of R is that, as our experiments indicate, a suitable reference point can be found as the average of the expression values of genes that are independent of the class label for which we are trying to develop a classifier. Further, normalizing with respect to such genes will not discard information that might be valuable in class discrimination. Choosing the k worst separators for normalization is a heuristic for identifying genes likely to be independent of the class label. While many factors (e.g., noise) could mislead this measure into selecting inappropriate reference point genes, it seems reasonable that, in the absence of additional information, genes that appear in the data to be bad separators are good candidates to serve as reference point

[7] Affymetrix Microarray Suite (MAS) Version 4.0.

genes. Indeed, our experimental results support this as a reasonable normalization strategy.

In summary, the normalization algorithm we used is as follows.

1. Let *R* consist of the *k* worst separator genes, as described above.

2. Let *A* represent the target average value for the genes in *R*; *A* may be chosen arbitrarily, since its value does not affect any aspects of the computation.

3. For each case *C*,

    a. Compute the average value, $Ave_{R,C}$, of the expression values in case *C* for the genes in *R*.

    b. Multiply *every* expression value of case *C* by the scaling factor $A/Ave_{R,C}$.

We took *k* to be a parameter to be learned in the course of training and experimented with several different values accordingly. The results of these experiments against training data are reported in Section 5.3; Section 5.4 reports how well a choice of *k* made against training data generalizes to an out-of-sample test set.

5. Results

The MIT leukemia data (Golub *et al.*, 1999) and the Princeton colon cancer data (Alon *et al.*, 1999) are considered. The MIT data consists of 7,129 gene expression values per case. The Princeton data is provided in a heavily pruned form, consisting of only 2,000 genes per case.

5.1. Experimental Methodology

In order to avoid any possibility of overfitting our results to specific data sets, we set aside from each data set a fraction of cases, forming a *test set*. For the MIT data set, a partition into 38 training cases and 34 test cases (our set aside, out-of-sample cases) is provided on the MIT Web site. The Princeton Web site provides a single data set of 62 cases. We randomly partitioned this data set into 38 training cases and 24 set aside test cases. The test sets were not examined *at all* in the course of algorithm development, nor to establish parameter settings, and were considered only after our best methods, along with their parameter settings, were identified through a cross validation methodology (detailed below) on the training sets. Results of our best method—as identified against the training sets only—run against the set aside test sets are reported in Section 5.4.

We now describe the cross validation methodology that was applied to the 38-case training sets in order to develop our methods and to indicate which techniques would be the most promising to pursue. In particular, our initial evaluation of a classifier building method under development employed “leave one out” (*LOO*) cross validation. On each experiment, a method would train on 37 cases, building a classifier to be used to classify the single left out query case; the build/evaluate cycle is repeated 38 times, once for each “fold” created by leaving out of the training a different query case.

Care must be taken during development that the methods used in the classifier construction process not exploit *any* knowledge of the left out query case it is to be evaluated on. That is, any method applied to build the classifier must be applicable when we turn attention to the set aside test set (or to an actual set of query cases for which a classification is desired), at which time knowledge of the query’s class label, of course, is unavailable.

This requirement implies, for example:

- Gene selection by external means must be repeated on each of the 38 folds, without being exposed to the left out case to be used as a query in the evaluation.

- Similarly, if normalization or binning is to use label knowledge, it must not be exposed to the left out case, and hence must be repeated for each fold. If, however, a binning algorithm does not use knowledge of labels (as is the case of the algorithm used in connection with the greedy construction), it may inspect the entire training set, since in an actual classification application, the binning algorithm could inspect the non-label fields (genes) of the cases to be classified at the time these cases are presented for analysis.

Greedy Parent Set Construction

The LOO cross validation setup for the greedy method takes the following form:

1. Let T represent the full training set (e.g., of 38 cases).
2. Bin T, without using label knowledge.
3. For each $q_i \in T$, define fold $F_i = (T - \{q_i\})$,
   a. Select $\mathcal{K}$ seeds against F_i.
   b. Use the greedy method to construct $\mathcal{PS}$ good sets (under *BD*) up to cardinality $\mathcal{M}$ against F_i, starting from each seed.
   c. Compute the variance in each set's induced distribution of q_i's unknown label, and adjust the *BD* score of each set to form a $\mathcal{PS}$-set classifier.
   d. Classify $q_i[C]$ as the most likely value, given $q_i[genes]$ under the classifier's distribution.
   e. Compute the error and uncertainty in the classification for fold F_i.
4. Report the average error and uncertainty rates across the folds.

The information reported in Step 4 is derived from the constructed classifiers' induced distributions. In particular, the classifier constructed for each fold F_i specifies a conditional posterior distribution $Pr\{q[C] = c_k \mid q[genes] =< e >\}$ for a query case's class label. In the current experiments, the class label is binary, and q is classified as belonging to the class with higher posterior; if value = 0.5, no classification is possible. An error occurs if $q[C]$ is the lower probability class. The TER (*total error rate*) values reported in Tables 3–6 are based on the combined number of misclassifications and no classifications.

Uncertainty is a measure of the strength of a classification. If $Pr\{q[C] = c_k \mid q[genes] =< e >\}$ is near 1.0, the classification is strong, whereas if it is near 0.5, it is weak. On each fold, we compute the "probability of error" as well as the 0/1 misclassification indicator. In particular, probability of error is given by (1.0 $-$ (the probability the classifier assigns to the true class $q[C]$)). The APE (*average probability of error*) values reported in Tables 3–6 are averages over this quantity.

For the experiments reported in Section 5.2 and 5.4, we utilized the following relatively simple binning (Step 2) and seed selection (Step 3.a) techniques.

Binning: As is indicated in Section 3.1, practical Bayesian net methods require a discretization of the expression values. Following many gene expression researchers, we partition values into three ranges: "under-", "average-", and "over-" expressed. Our partitioning method for greedy creates a tertiary binning for each gene g as

$$\begin{gathered}(-\infty, (mean(g) - n_{low} \times \sigma(g)),\\ [mean(g) - n_{low} \times \sigma(g), mean(g) + n_{high} \times \sigma(g)],\\ (mean(g) + n_{high} \times \sigma(g), \infty),\end{gathered}$$

where the mean $mean(g)$ and standard deviation $\sigma(g)$ of each gene's g expression values are computed over all cases. The choices of n_{low} and n_{high} are made through experimentation on the training data. Once selected, these are fixed and used without modification on the set aside test data; otherwise, we would run the risk of overfitting to the data. For the MIT data, setting $n_{low} = n_{high} = 1.0$ worked well, and there was little sensitivity in the cross validation results. In the Princeton data, there was far more sensitivity in the cross validation, and a limited search arrived at the settings $n_{low} = 1.25$ and $n_{high} = 0.4$. Subsequent analysis indicates that a more extensive search for these parameter settings often results in overfitting to the data. In fact, it appears that the tertiary binning considered here is generally inferior to the binary binning described in Section 4.2 in conjunction with the external gene selection methods.

Seed selection: Singleton parent sets $\{\mathrm{g}\}$ are formed for each gene g and the *BD* score obtained. The genes corresponding to the $\mathcal{K}$ highest scoring parent sets are used as seeds.

External Gene Selection Plus Exhaustive Parent Set Construction

The LOO cross validation setup for external gene selection takes the following form:

1. Let T represent the full training set (e.g., of 38 cases).
2. For each fold defined by $F_i = (T - \{q_i\})$,
   a. Use an external method against F_i to normalize expression values and select a set S of $\mathcal{N}$ genes.
   b. Bin F_i, possibly using information returned by gene selection.
   c. Exhaustively search the set S for the best $\mathcal{PS}$ subsets (of cardinality up to $\mathcal{M}$) under the *BD* scoring metric.
   d. Compute the variance in each set's induced distribution of q_i's unknown label, and adjust the *BD* score of each set to form a $\mathcal{PS}$-set classifier.
   e. Classify $q_i[C]$ as the most likely value, given $q_i[genes]$ under the classifier's distribution.
   f. Compute the error and uncertainty in the classification for fold F_i.
3. Report the average error and uncertainty rates across the folds.

In our experiments, we employed the external gene selection, normalization, and binning methods described in Section 4.2. In particular, the external gene selection algorithm is invoked on each fold with the following effect:

- The algorithm normalizes the cases in F_i using the k genes with the lowest $SeparationQuality$ as controls.
- The algorithm returns the $\mathcal{N}$ genes with the highest $SeparationQuality$.
- The algorithm returns a binary bin boundary for each selected gene, corresponding to where the maximum separation value is obtained.

Once results of the external gene selection algorithm are returned for a fold, an exhaustive search is performed (on a normalized and binned F_i) for the best $\mathcal{PS}$ parent sets, from which the Bayesian net classifier is formed.

Note that the instantiation of the steps of either methodology with specific algorithms defines a classifier building method. When run on a specific training set (or fold of a training set), it yields a $\mathcal{PS}$-set classifier, which in turn yields a posterior class distribution. This distribution can then be used to classify query cases with unknown labels, assuming that the query cases are drawn from the same distribution which underlies the training set. We emphasize that it is the building method, not the particular classifiers built on a run against a training set (or fold of a training set), that is being assessed.

5.2. Cross Validation Results with Greedy

In tests of the greedy method, we studied the effects of varying the number $\mathcal{PS}$ of sets used in the classifier. We held fixed at $\mathcal{M} = 5$ the maximum cardinality and, due to computational considerations, the number of seeds at $\mathcal{K} = 60$.

The following two tables summarize, respectively, results with the Princeton and MIT training sets. Each row of the tables summarizes, for a fixed $\mathcal{PS}$, the LOO cross validation test results for the 38 cases of the respective training set. The table entries *MIS*, *ERR*, and TER tally the number of misclassifications and nonclassifications as described in the legand below. *APE*–average probability of error per fold—captures the uncertainty in the classifications. Since the classification is based on the posterior probability of a class, a posterior near 1.0 or 0.0 is a confident prediction (which may be either correct or incorrect), while a posterior near 0.5 is a prediction with low confidence (when the posterior is approximately 0.5, no classification is made). The error in a prediction is 1.0 minus the posterior probability assigned by the classifier to the true class, and APE is the average over these errors. The *qMax* result appearing at the end of each table is discussed below.

Legend:

$\mathcal{PS}$: Number of parent sets used.
APE : Average probability error per fold.
MIS : Number of misclassifications.
ERR : Total error count (misclassifications + nonclassifications).
TER : Total error rate (including both misclassifications and nonclassifications).

$\mathcal{PS}$	APE	MIS	ERR	TER
1	0.184212	4	10	0.263158
5	0.169929	7	7	0.184211
10	0.259123	12	12	0.315789
20	0.312331	14	14	0.368421
60	0.329858	13	13	0.342105
300	0.340612	13	13	0.342105
500	0.346113	14	14	0.368421
$qMax$	0.289474	11	11	0.289474

Table 1. Princeton training data ($n_{low} = 1.25$, $n_{high} = 0.4$).

$\mathcal{PS}$	APE	MIS	ERR	TER
1	0.315791	0	24	0.631579
5	0.193975	1	14	0.368421
10	0.140994	1	9	0.236842
20	0.067464	2	3	0.078947
60	0.070245	3	3	0.078947
300	0.089030	3	3	0.078947
500	0.118584	5	5	0.131579
$qMax$	0.157897	6	6	0.157897

Table 2. MIT training data ($n_{low} = 1.0$, $n_{high} = 1.0$).

The tables indicate an initial increase in quality as $\mathcal{PS}$ increases, then a leveling off and ultimate decrease in quality. The most interesting result is the significant increase in quality over just a single set ($\mathcal{PS} = 1$, the *maximum a posteriori solution*), which is a prevalent Bayesian net methodology for learning distributions. As predicted from the discussion in Section 3.3, a single parent set does not provide adequate coverage of gene expression combinations in the query case, leading to a large number of non classifications.

To establish that the polling effect noted in Section 3.3 is real and significant, we also conducted experiments labeled "*qMax*". Here, 500 sets are built as with $\mathcal{PS} = 500$, but for each query case q, the single parent set with the highest variance adjusted score is used to classify q. Note that this query-specific set selection from the 500 always selects (if available, which is the case in all our cross validation runs) a set in which q's combination of expression values appears in the training set, eliminating the no-classification errors. That this method underperforms the best $\mathcal{PS} > 1$ methods indicates that the blending of distributions contributes to the quality of the classification. Examination of the details of the computations performed by the classifier also indicates that, in many cases, the distributions induced by the parent sets exert competing effects on the classification, and that the weighting resolution generally leads to a correct classification.

We speculate that the degradation in classification quality for $\mathcal{PS}$ above a threshold is caused by the potentially unrepresentative search performed by our simple greedy algorithm, as alluded to in Section 4.1—greedy, being unable to construct enough high scoring sets, must "fill" the classifier with many low scoring (and, hence, worse fitting to the observational data) sets which contribute inaccurate distributions. This explanation is supported by the near monotonic increase in quality reported in Section 5.3 for the exhaustive search following external gene selection. This suggests that refinements to greedy as proposed in Section 4.1 could well obtain overall improvements, especially as is noted in Section 5.4 when we discuss the results of the greedy-built classifiers against the out-of-sample test set.

5.3. Cross Validation Results with External Gene Selection

In tests of the external gene selection methods, we studied the effects of varying both $\mathcal{PS}$ and the fraction $\mathcal{W}$ of genes used as controls in normalization. As with greedy, we held fixed the maximum cardinality $\mathcal{M}$ at 5. For computational reasons, the number of genes selected was fixed at 30.

The following two tables summarize, respectively, results with the Princeton and MIT training sets. Each row of the tables summarizes, for a fixed $\mathcal{W}$ and $\mathcal{PS}$, the LOO cross validation test results for the 38 cases of the respective training set. As is the case for Tables 1 and 2, the *qMax* result at the end of each of Tables 3 and 4 is for 500 available parent sets and with $\mathcal{W}$ set at a value which produced generally good results across the $\mathcal{PS}$ values for the multi-set classifiers.

Legend:

$\mathcal{PS}$: Number of parent sets used.
$\mathcal{W}$: Fraction of genes used as controls for normalization.
APE : Average probability error per fold.
MIS : Number of misclassifications.
ERR : Total error count (misclassifications + nonclassifications).
TER : Total error rate (including both misclassifications and nonclassifications).

$\mathcal{PS}$	$\mathcal{W}$	APE	MIS	ERR	TER
1	0.000000	0.394735	15	15	0.394737
1	0.100000	0.480700	17	20	0.526316
1	0.250000	0.328950	8	17	0.447368
1	0.400000	0.302636	8	15	0.394737
1	0.550000	0.328950	7	18	0.473684
1	0.700000	0.263162	7	13	0.342105
1	0.850000	0.499997	18	20	0.526316
1	1.000000	0.499994	16	22	0.578947
5	0.000000	0.393831	14	14	0.368421
5	0.100000	0.376276	14	14	0.368421
5	0.250000	0.287669	9	11	0.289474
5	0.400000	0.267520	9	11	0.289474
5	0.550000	0.241729	9	10	0.263158
5	0.700000	0.261501	9	10	0.263158
5	0.850000	0.455024	17	17	0.447368
5	1.000000	0.333537	10	13	0.342105
10	0.000000	0.377660	15	15	0.394737
10	0.100000	0.398858	15	15	0.394737
10	0.250000	0.334334	12	12	0.315789
10	0.400000	0.261875	9	11	0.289474
10	0.550000	0.221307	8	9	0.236842
10	0.700000	0.270484	9	9	0.236842
10	0.850000	0.410469	14	14	0.368421
10	1.000000	0.303383	10	11	0.289474

Table 3. Princeton training data.

$\mathcal{PS}$	$\mathcal{W}$	APE	MIS	ERR	TER
20	0.000000	0.377660	15	15	0.394737
20	0.100000	0.402184	16	16	0.421053
20	0.250000	0.302113	11	11	0.289474
20	0.400000	0.251675	9	9	0.236842
20	0.550000	0.215504	7	8	0.210526
20	0.700000	0.265321	9	9	0.236842
20	0.850000	0.361076	12	12	0.315789
20	1.000000	0.325153	11	12	0.315789
60	0.000000	0.350131	12	12	0.315789
60	0.100000	0.375262	14	14	0.368421
60	0.250000	0.290695	10	10	0.263158
60	0.400000	0.233612	9	9	0.236842
60	0.550000	0.204675	7	7	0.184211
60	0.700000	0.249359	8	8	0.210526
60	0.850000	0.358279	12	12	0.315789
60	1.000000	0.286617	10	11	0.289474
300	0.000000	0.344514	13	13	0.342105
300	0.100000	0.358541	14	14	0.368421
300	0.250000	0.297478	11	11	0.289474
300	0.400000	0.223621	7	7	0.184211
300	0.550000	0.204802	7	7	0.184211
300	0.700000	0.237995	8	8	0.210526
300	0.850000	0.317356	12	12	0.315789
300	1.000000	0.249347	9	9	0.236842
500	0.000000	0.341484	13	13	0.342105
500	0.100000	0.351571	14	14	0.368421
500	0.250000	0.293802	12	12	0.315789
500	0.400000	0.218802	7	7	0.184211
500	0.550000	0.206535	6	6	0.157895
500	0.700000	0.231278	8	8	0.210526
500	0.850000	0.301052	11	11	0.289474
500	1.000000	0.251559	9	9	0.236842
qMax	0.550000	0.210529	8	8	0.210526

Table 3. Princeton training data (continued).

$\mathcal{PS}$	$\mathcal{W}$	APE	MIS	ERR	TER
1	0.000000	0.065801	1	4	0.105263
1	0.100000	0.052644	1	3	0.078947
1	0.250000	0.065801	1	4	0.105263
1	0.400000	0.078959	1	5	0.131579
1	0.550000	0.078959	1	5	0.131579
1	0.700000	0.078959	1	5	0.131579
1	0.850000	0.065802	1	4	0.105263
1	1.000000	0.078959	1	5	0.131579
5	0.000000	0.072555	3	3	0.078947
5	0.100000	0.053353	2	2	0.052632
5	0.250000	0.072555	3	3	0.078947
5	0.400000	0.080379	3	3	0.078947
5	0.550000	0.080379	3	3	0.078947
5	0.700000	0.080379	3	3	0.078947
5	0.850000	0.061176	2	2	0.052632
5	1.000000	0.080378	3	3	0.078947
10	0.000000	0.072554	3	3	0.078947
10	0.100000	0.053351	2	2	0.052632
10	0.250000	0.072554	3	3	0.078947
10	0.400000	0.080378	3	3	0.078947
10	0.550000	0.080378	3	3	0.078947
10	0.700000	0.080378	3	3	0.078947
10	0.850000	0.061175	2	2	0.052632
10	1.000000	0.083038	3	3	0.078947

Table 4. MIT training data.

$\mathcal{PS}$	$\mathcal{W}$	APE	MIS	ERR	TER
20	0.000000	0.072553	3	3	0.078947
20	0.100000	0.053351	2	2	0.052632
20	0.250000	0.072553	3	3	0.078947
20	0.400000	0.080377	3	3	0.078947
20	0.550000	0.080377	3	3	0.078947
20	0.700000	0.080377	3	3	0.078947
20	0.850000	0.061174	2	2	0.052632
20	1.000000	0.084275	3	3	0.078947
60	0.000000	0.070544	3	3	0.078947
60	0.100000	0.051839	2	2	0.052632
60	0.250000	0.071990	3	3	0.078947
60	0.400000	0.069324	3	3	0.078947
60	0.550000	0.070813	3	3	0.078947
60	0.700000	0.070774	3	3	0.078947
60	0.850000	0.050437	2	2	0.052632
60	1.000000	0.069833	3	3	0.078947
300	0.000000	0.057444	2	2	0.052632
300	0.100000	0.059049	2	2	0.052632
300	0.250000	0.072465	3	3	0.078947
300	0.400000	0.074483	3	3	0.078947
300	0.550000	0.074229	3	3	0.078947
300	0.700000	0.075196	3	3	0.078947
300	0.850000	0.056310	2	2	0.052632
300	1.000000	0.050879	2	2	0.052632
500	0.000000	0.065868	2	2	0.052632
500	0.100000	0.068942	2	2	0.052632
500	0.250000	0.080150	3	3	0.078947
500	0.400000	0.079164	3	3	0.078947
500	0.550000	0.078501	3	3	0.078947
500	0.700000	0.078683	3	3	0.078947
500	0.850000	0.074403	2	2	0.052632
500	1.000000	0.068241	2	2	0.052632
$qMax$	0.100000	0.052644	2	2	0.052632

Table 4. MIT training data (continued).

Unlike the case for greedy selection, the results of Tables 3 and 4 demonstrate that there is a steady improvement for the Princeton data as $\mathcal{PS}$ increases, and near flat behavior for the MIT data for $\mathcal{PS} \geq 60$. Again, the *qMax* experiments (for the Princeton data) and inspection of the detailed results provide further evidence that the blending provided by a large number of parent sets has a positive impact on classifier quality.

The tables indicate different best values across the two training sets for the fraction $\mathcal{W}$ of control genes used in expression-level normalization, and a greater sensitivity to this value in the Princeton training data. This may be indicative of differences in experimental conditions, analysis preprocessing, and so forth. That we can, without the benefit of descriptive procedural information as input, discover through methodical application of cross validation good normalization parameters for each data set is a significant finding. The results against the test set presented in the following section indicate that these findings are not simply an overfitting to the training data, but truly a learning of the underlying processes that generalizes well.

5.4. Out-of-Sample Test Set Results

Only after running the above experiments on the training sets did we turn attention to the test sets. Our primary interest is to select the *single method* which performed best (lowest total error rate, TER) in the cross validation experiments and assess its classification rate on the out-of-sample test sets. In this way, we avoid a "selection effect" in which one of several methods run against the test set performs well.

Inspection of the tables of Sections 5.2 and 5.3 identifies the external gene selection method as being preferable to the minimal knowledge greedy method in building parent sets for the Bayesian net classifier. Since we have data from two different experimental contexts, it is proper to select the parameters for the selected method (i.e., $\mathcal{PS}$ and $\mathcal{W}$) based on performance in the cross validation trials on each training set; such parameter setting would of course be performed in an actual classification application in which we had access to training, but not query, cases in advance.

External Gene-Selection Method Against Test Data

Inspection of the tables in Section 5.3 indicates that, against the Princeton training set, the best setting is $\mathcal{PS} = 500$ (number of parent sets to be used in the Bayesian net classifier) and $\mathcal{W}$=0.55 (control list fraction for normalization). Against the MIT training set, several parameter settings resulted in the minimal TER of 0.052632. Somewhat arbitrarily, we selected $\mathcal{PS}$=300 and $\mathcal{W} = 0.85$.[8] Using *only these settings*, we built the classifiers by training against the 38 cases of each of the two training sets and used the resulting classifiers to classify the cases of the respective test sets.

The results are exhibited in Table 5 and are extremely good. The classifier had nearly identical error rates against the MIT training and test sets (0.05 for training versus 0.06 for test) and a significantly lower error rate against the Princeton test set (0.16 for training versus 0.08 for test). The results strongly suggest that our multi-parent-set Bayesian net classifiers employing external gene selection and normalization algorithms are able to learn from training data underlying distributions which generalize extremely well to out-of-sample query cases whose classifications are of biological and clinical significance.

Comparison with Other Published Results

The Appendix contains an extensive compilation of results reported in the literature for the MIT and Princeton datasets, generated using a broad range of classification methodologies. The high accuracies achieved in our results are particularly noteworthy given the stringent nature of our 'one-shot' testing approach: we have used

[8] While we chose our single run to be made against the test set with $\mathcal{PS} = 300$ and $\mathcal{W} = 0.85$, in order to assess the sensitivity of the results to this somewhat arbitrary choice of settings from among settings achieving equally good TER, we later ran against the test set with several other settings which achieved the same TER against the training data. The majority of those settings tried also incurred the same number 2 of misclassification errors as those reported here, while a few others incurred 3 misclassifications errors.

parameter settings determined from cross-validation results for the training dataset only, followed by a single pass at classifying the held-out test set. Our results compare favorably even with those obtained using rather less stringent training/testing protocols.

Test Set	Cases	APE	MIS	ERR	TER
Princeton	24	0.142092	2	2	0.083333
MIT	34	0.085831	2	2	0.058824

Table 5. Out-of-sample results with external gene selection.

Minimal-Knowledge Greedy Methods Against Test Data

After obtaining the results reported in the previous subsection for the external methods, we decided also to run our greedy methods against the test sets. Since the greedy method's results in the cross validation experiments were almost as good as the external gene selection methods, we consider this to be an interesting avenue of research as well. We report the results here in order to indicate potential directions for future work.

Table 6 reports the results against the two test sets of Bayesian net classification using the greedy construction method. The only parameter considered in the cross validation against the training set was $\mathcal{PS}$, with the best settings found to be $\mathcal{PS} = 20$ for the MIT training set and $\mathcal{PS} = 5$ for the Princeton training set.

Test Set	Cases	APE	MIS	ERR	TER
Princeton	24	0.145834	1	6	0.250000
MIT	34	0.279412	7	12	0.352941

Table 6. Out-of-sample results with greedy selection.

Against the Princeton test set, the error rate was similar to the rate against the training set (0.18 for training versus 0.25 for test), but it was significantly higher against the MIT test set (0.08 for training versus 0.35 for test). We speculate that two sources of this lack of generalization, especially in the MIT data, are our failure to normalize the data for the greedy experiments and the use of an overly rigid binning method. This conjecture is consistent with the high number of "nonclassifications" against the test sets. Note also that the MIT data was provided as two distinct data sets. Procedural differences in experimental preparation and processing of the output between the sets as is described in (Golub *et al.*, 1999) may have hampered the greedy method because it fails to normalize across the sets. In the case of the Princeton data, where a single data set is randomly split, performance against the test set was much more comparable to that of the training set.

Consequently, one avenue of future research is to include in the greedy method a normalization procedure similar to that employed by the external gene selection method. Also, as noted in Section 4.1, there is a concern that the greedy search may not provide a good representation of the space of possible parent sets. We speculated that this might be the cause of the degradation observed in the cross validation experiments for large values of $\mathcal{PS}$. Note that the exhaustive (and, hence, completely representative) search of the universe of externally selected genes resulted in large $\mathcal{PS}$s performing best. The greedy method's use of small values of $\mathcal{PS}$, in combination with the failure to normalize, certainly contributes to the large number of non-classifications in the test set. Hence, modifying the search to be more representative, as discussed in Section 4.1, potentially could give minimal-knowledge searches such as greedy access to more good parent sets, thereby addressing the large number of failure-to-classify errors that were observed.

6. Summary and Future Work

We have presented a methodology for applying Bayesian nets to the problem of classifying clinical cases from their gene expression profiles. While Bayesian nets have been applied previously to identify relationships among genes and have been proposed as classifiers for other problem domains, we have outlined new methods for classification particularly well suited to gene expression data. Through a systematic experimental design, we demonstrated that these classifiers, trained by means of a cross-validation methodology, generalize extremely well to out-of-sample test data. In particular, we achieved error rates of 92% and 94% on out-of-sample partitions of the MIT leukemia and Princeton colon cancer data sets, respectively. These results are comparable to or better than those reported previously using other classification approaches, even in those instances when less stringent train/test procedures were utilized.

Our Bayesian net classifiers are built by constructing alternative parent sets for the class label node and use a posterior probability and variance-weighted blending of the resulting distributions. This blending of the distributions induced by the competing hypotheses embodied by the alternative parent sets was seen in our experimental results to yield improvements over the so called *maximum a posteriori solution*, in which only the single most likely hypothesis is used. We experimented with two methods for searching for good parent sets: a simple greedy search of the universe of all genes and an exhaustive search of a universe of genes selected by a separation heuristic. The latter method produced better performing parent sets in the experiments reported here. This method also employs a novel expression-level normalization scheme based on algorithmically discovered control genes. Current work is considering improvements to both methods for parent set construction and to normalization. We are exploring also how other aspects of the problem—value binning and gene clustering, for example—can be studied within the framework.

It also is possible to incorporate into the model local structure, as described in Friedman and Goldzmidt (1996b) and Chickering *et al.* (1997). While this has not been necessary for the classification tasks undertaken to date, future work will explore the utility of this model extension in the gene expression domain.

We believe that Bayesian approaches to gene expression analysis, such as those described here and in Friedman *et al.* (1999), Friedman *et al.* (2000) and Pe’er *et al.* (2001), have enormous potential, not simply because of the quality of the results achieved so far, but also because the mathematically-grounded formalism provides the opportunity to expand systematically the range of problems treated, integrating newly developed algorithmic techniques with an ever-increasing base of domain knowledge. Thus, results such as those reported here, while significant in their own right, are only the first steps toward the ultimate construction of rigorous and comprehensive models that promise to be of great scientific and clinical import.

Acknowledgments

This work has been supported by grants from the D.H.H.S. National Institutes of Health/National Cancer Institute (CA88361), the W.M. Keck Foundation, and National Tobacco Settlement funds to the State of New Mexico Provided to UNM for Genomics and Bioinformatics. We are grateful for the computational support that has been provided by the UNM High Performance Computing Education and Research Center (HPCERC). We are grateful for the comments of the anonymous referee, which greatly improved this work.

A. COMPILATION OF PUBLISHED CLASSIFICATION RESULTS

In this Appendix, we list the feature selection and classification methodologies, testing procedures, and accuracies that have been reported in the literature to date for the two datasets considered in this work. Results for the MIT dataset are given in Table 7, and for the Princeton dataset, in Table 8. Within each table, results are listed in order of *decreasing* testing protocol stringency (most stringent appear first), and within stringency sub-category, in order of decreasing classification success rate. The stringency hierarchy used to order the results is *Separate Train/Test* followed by *One-Shot Test?* followed by *# Features*. Due to the varied nature of the classification testing protocols utilized in the literature, the detailed ordering of results in cases where multiple attempts were made is necessarily somewhat arbitrary.

Table 7. MIT dataset.

CLASSIFIER	SEPARATE TRAIN/TEST?	ONE-SHOT TEST?	FEATURE SELECTION METHOD	# FEATURES	TESTING ERRORS	REF.
NN with $[N(\mu,\nu)]^{1/2}$ metric and Euclidean distance kernel	Y	Y	maximized between-class distance, with $[N(\mu,\nu)]^{1/2}$ metric and Euclidean distance kernel[1]	10	2	[1]
Bayesian network	Y	Y	Separation measure of Section 4.2, present work	30	2	Present work
neighborhood analysis (weighted voting scheme)	Y	Y	MIT “F(x)” score[2]	50	10	[2]
linear SVM	Y	Y	MIT “F(x)” score[2]	7129	1	[3]
linear SVM	Y	Y	none	7129	1	[4]
PCI	Y	Y	none	7129	2	[5]
linear SVM	Y	Y	none	7129	5	[6]
MIT linear discriminant (“baseline”) classifier	Y	Y	none	7129	5	[6]
Bayesian (max-likelihood)	Y[3]	N	ARD (I) ARD (II)	6.98 7.09	2.04 2.90	[7]
SVM	Y[3]	N	RFE Fisher score	14.41	2.84 2.68	[7]
linear SVM	Y	N (*f*)	R^2W^2 minimization	5; **20**	1;**0**	[4]
linear SVM	Y	N (*f*)	Fisher score	5; **20**	5;**3**	[4]
discriminant analysis[4]	Y[5]	N (*a*)	BSS/WSS ratio	40	3 **(a)** 0 **(b)** 1 **(c)** 1 **(d)**	[8]; [9]
classification trees[6]	Y[5]	N (*a*)	BSS/WSS ratio	40	3 **(a)** 2 **(b)** 1 **(c)** 1 **(d)**	[8]; [9]
linear SVM	Y	N (*f*)[7]	*K=2* superv. NBGR **(a)** MVR **(b)** MAR **(c)**	50	2 **(a)** 1 **(b)** 1 **(c)**	[10]

[1] 10-fold cross-validated search; search repeated 50 times with 10,000 iterations each.

[2] See [1] for definition, also referred to as Mean Aggregate Relevance (MAR) [10].

[3] 100 trials of random 36/36 splits; # of features and testing errors reported for each feature selection algorithm are averages computed over 100 trials.

[4] Discriminant analysis methods: **(a)** FLDA; **(b)** DLDA; **(c)** method of [GST99]; **(d)** DQDA.

[5] 200 trials of different 48/24 (2:1) train/test splits; testing errors for methods **(a)**–**(d)** are for *median quartile* over 200 trials.

[6] Classification tree methods: **(a)** single CART tree with pruning by 10-fold cross-validation; **(b)** 50 bagged exploratory trees; **(c)** 50 boosted (using *Arc-fs*) exploratory trees; **(d)** 50 bagged exploratory trees with convex pseudodata.

[7] Three distinct sets of “top 50” genes were generated using the NBGR, MVR, and MAR feature selection methods. Testing errors are correspondingly labeled **(a)**, **(b)**, **(c)**.

Table 7. MIT dataset, continued.

CLASSIFIER	SEPARATE TRAIN/TEST?	ONE-SHOT TEST?	FEATURE SELECTION METHOD	# FEATURES	TESTING ERRORS	REF.
nonlinear (radial basis function) SVM, data-dependent range	Y	N (*f*)[7]	*K=2* superv. NGBR **(a)** MVR **(b)** MAR **(c)**	50	2 **(a)** 1 **(b)** 1 **(c)**	[10]
KNN/Euclidean metric	Y	N (*p*)	GA/KNN with *z*-score	50	1–2	[11]
nonlinear (radial basis function) SVM, data-dependent range	Y	N (*f*)[7]	*K=2* superv. NGBR **(a)** MVR **(b)** MAR **(c)**	50	4 **(a)** 1 **(b)** 2 **(c)**	[10]
multiclass[8] nonlinear (Gaussian) SVM, GACV tuning	Y	N (*f*,*p*,*a*)	BSS/WSS ratio	**40**; **50**; 100	**1**; **1**; 4	[12]
multiclass[8] linear SVM, LOOCV tuning	Y	N (*f*,*p*,*a*)	BSS/WSS ratio	**40**; 50; 100	**1**; 2.25; 4	[12]
multiclass[8] linear SVM, GACV tuning	Y	N (*f*,*p*,*a*)	BSS/WSS ratio	**40**; 50; 100	**1**; 2; 4	[12]
multiclass[8] nonlinear (Gaussian) SVM, LOOCV tuning	Y	N (*f*,*p*,*a*)	BSS/WSS ratio	**40**; 50; 100	**0.8**; 1; 6	[12]
linear SVM	Y	N (*f*)	MIT "F(x)" score[2]	49; **99**; **999**	2;**0**;**0**	[3]
linear SVM	Y	N (*f*)	MIT "F(x)" score[2]	25; 250; 500; 1000	2–4	[13]
LD	Y	N (*f*)	*t*-score + PLS	**50**; 100; 500 1000; 1500	**1**;2;3;3;3	[14]
LD	Y	N (*f*)	*t*-score + PC	**50**; 100; 500 1000; 1500	**1**;2;3;3;4	[14]
LD	Y	N (*f*)	nested expressed genes (100%; 75%; 50%; 25%; at least 1 array) + PLS	246; **662**; **864** **1076**; **1554**	7;**3**;**3**;**3**;**3**	[14]
LD	Y	N (*f*)	nested expressed genes (100%; 75%; 50%; 25%; at least 1 array) + PC	246; **662**; **864** **1076**; **1554**	12;**9**;**9**;**9**;**9**	[14]
QDA	Y	N (*f*)	*t*-score + PLS	50; 100; **500** 1000; 1500	6;5;**2**;3;4	[14]
QDA	Y	N (*f*)	*t*-score + PC	**50**; **100**; 500 1000; 1500	**4**;**4**;6;6;6	[14]
QDA	Y	N (*f*)	nested expressed genes (100%; 75%; 50%; 25%; at least 1 array) + PLS	246; 662; 864 **1076**; 1554	10?;3;3;**2**;3	[14]
QDA	Y	N (*f*)	nested expressed genes (100%; 75%; 50%; 25%; at least 1 array) + PC	246; **662**; 864 1076; 1554	17;**4**;6;6;6	[14]
linear SVM	Y	N (*f*)	SVM RFE	1; 2; 4; **8**; **16**; 32 64 128; 256; 512 1024 2048; 4096	7;4;3;**0**;**0**;1;2 1;2;4;2 5;10	[6]
linear SVM	Y	N (*f*)	MIT "F(x)" score[2]	1; 2; 4; 8; 16; 32 **64**; 128; 256; 512 1024 2048; 4096	7;7;4;3;2;3;**1** 2;3;2;2 5;9	[6]
MIT linear discriminant ("baseline") classifier	Y	N (*f*)	SVM RFE	1; 2; 4; 8; 16; 32 **64**; 128; 256; 512 1024 2048; 4096	7;4;4;1;2;2;**0** 2;2;3;2 4;4	[6]

[8] The three classes are: B-ALL (38), T-ALL (9), AML (25).

Table 7. MIT dataset, continued.

CLASSIFIER	SEPARATE TRAIN/TEST?	ONE-SHOT TEST?	FEATURE SELECTION METHOD	# FEATURES	TESTING ERRORS	REF.
MIT linear discriminant (“baseline”) classifier	Y	N (*f*)	MIT “F(x)” score[2]	1; 2; 4; 8; 16; 32 **64**; 128; 256; 512 1024 2048; 4096	7;6;4;3;2;2;**1** 2;3;2;2 5;5	[6]
linear SVM	N[9]	Y	none	7129	0	[13]
modified perceptron	N[9]	Y[10]	none	7129	3.4	[13]
quadratic SVM	N[11]	Y	none	7129	4	[15]
linear SVM	N[11]	Y	none	7129	5	[15]
NN/Pearson correlation metric	N[11]	Y	none	7129	6	[15]
Emerging patterns	N[11]	N (*f*)	entropy-based discretization	1	3	[16]
multiclass[8] PD	N[11]	N (*f,a*)[12]	PAMD/CS + MPLS	94; **813** (**A0**) 94; 813 (**A1**) 69-90-100 (**A2**) 710-804-850 (**A2**)	4; **3** 4; 4 4 4	[17]
multiclass[8] PD	N[11]	N (*f,a*)[12]	PAMD/CS + PC	**94**; **813** (**A0**) 94; 813 (**A1**) 69-90-100 (**A2**) 710-804-850 (**A2**)	**4**; **4** 8; 11 8 12	[17]
multiclass[8] QDA	N[11]	N (*f,a*)[12]	PAMD/CS + MPLS	94; **813** (**A0**) 94; 813 (**A1**) 69-90-100 (**A2**) 710-804-850 (**A2**)	1; **0** 2; 2 4 3	[17]
multiclass[8] QDA	N[11]	N (*f,a*)[12]	PAMD/CS + PC	94; **813** (**A0**) 94; 813 (**A1**) 69-90-100 (**A2**) 710-804-850 (**A2**)	3; **2** 16; 30 17 31	[17]
multiclass[8] DQDA	N[11]	N (*f,a*)[12]	PAMD/CS + MPLS	94; **813** (**A0**) 94; 813 (**A1**) 69-90-100 (**A2**) 710-804-850 (**A2**)	2; **1** 2; 2 4 3	[17]
multiclass[8] DQDA	N[11]	N (*f,a*)[12]	PAMD/CS + PC	**94**; **813** (**A0**) 94; 813 (**A1**) 69-90-100 (**A2**) 710-804-850 (**A2**)	**3**; **3** 13; 23 19 22	[17]
multiclass[8] DLDA	N[11]	N (*f,a*)[12]	PAMD/CS + MPLS	94; **813** (**A0**) 94; 813 (**A1**) 69; 90; 100 (**A2**) 710-804-850 (**A2**)	2; **1** 3; 3 4 4	[17]

[9] Leave-one-out classification results are total # errors for training set only (MIT) and full train+test set (Princeton).

[10] Result is average over 5 runs, each with a new classifier constructed for a new sample-shuffled dataset, since the perceptron method is sensitive to sample order.

[11] Leave-one-out classification results are total errors for train+test set (*N*=72, MIT; *N*=62, Princeton).

[12] Feature lists are labeled according to the algorithm variant with which they were used (**A0**, **A1**, **A2**). Note that for **A2**, the values x-y-z correspond to the min-mean-max of the *N*-fold gene re-selections.

Table 7. MIT dataset, continued.

CLASSIFIER	SEPARATE TRAIN/TEST?	ONE-SHOT TEST?	FEATURE SELECTION METHOD	# FEATURES	TESTING ERRORS	REF.
multiclass[8] DLDA	N[11]	N (*f*,*a*)[12]	PAMD/CS + PC	**94**; **813** (**A0**) **94**; 813 (**A1**) 69-90-100 (**A2**) 710-804-850 (**A2**)	**3**; **3** **3**; 6 8 8	[17]
linear SVM	N[11]	N (*f*)[13]	*K=2* supervised NBGR	4;5;11;**25**;50 **100**; 7070	5;12;2;**1**;2 **1**;2	[10]
linear SVM	N[11]	N (*f*)[13]	MVR	4;5;11;25;50 **100**; 7070	23;3;2;2;2 **1**;2	[10]
nonlinear (radial basis function) SVM, data-dependent range	N[11]	N (*f*)[13]	*K=2* supervised NBGR	4;5;11;**25**;50 **100**; **7070**	8;9;3;**2**;4 **2**;**2**	[10]
nonlinear (radial basis function) SVM, data-dependent range	N[11]	N (*f*)[13]	MVR	4;5;11;**25**;**50** 100; 7070	33;8;3;**1**;**1** 2;2	[10]
nonlinear (radial basis function) SVM, fixed range	N[11]	N (*f*)[13]	*K=2* supervised NBGR	4;5;**11**;25;**50** 100; 7070	19;19;**2**;4;**2** 4;27	[10]
nonlinear (radial basis function) SVM, fixed range	N[11]	N (*f*)[13]	MVR	4;5;11;25;50 **100**; 7070	31;11;8;7;6 **4**;27	[10]
boosting with decision stump learner	N[11]	N (*p*)[14]	none	7129	3 **(a)** 3 **(b)** 3 **(c)**	[15]

[13] Prior to feature selection and classification, the initial genelists were filtered from 7129→7070 (MIT) and 2000→1998 (Princeton). For #genes = 7070 (MIT) and 1998 (Princeton), no feature selection was performed; however, for convenience, the results are reported together with the feature-selected results for each classification methodology.

[14] Testing errors are labeled **(a)**, **(b)**, **(c)** corresponding to the number of boosting iterations (100, 1000, 10,000 respectively).

Table 8. Princeton dataset.

CLASSIFIER	SEPARATE TRAIN/TEST?	ONE-SHOT TEST?	FEATURE SELECTION METHOD	# FEATURES	TESTING ERRORS	REF.
Bayesian network	Y[15]	Y	Separation measure of Section 4.2, present work	30	2	Present work
linear SVM	Y[16]	Y	none	2000	5	[6]
MIT linear discriminant classifier	Y[16]	Y	none	2000	7	[6]
KNN/Euclidean metric	Y[17]	Y	none	2000	10	[11]
linear SVM	Y[16]	N (*f*)	SVM RFE	8	3	[6]
linear SVM	Y[16]	N (*f*)	MIT "F(x)" score[2]	8	5	[6]
linear SVM	Y[18]	N (*a*)	R^2W^2 min. **(a)** Pearson corr. coeff. **(b)** Fisher score **(c)** KS **(d)**	15	1.5 **(a)** 2.0 **(b)** 2.3 **(c)** 2.3 **(d)**	[4]
MIT linear discriminant classifier	Y[16]	N (*f*)	MIT "F(x)" score[2]	16	6	[6]
MIT linear discriminant classifier	Y[16]	N (*f*)	SVM RFE	32	4	[6]
KNN/Euclidean metric	Y[19]	N(*p*)	GA/KNN with *z*-score	50	0 **(a)** 0 **(b)** 1 **(c)**	[18]
KNN/Euclidean metric	Y[17]	N(*f*)	GA/KNN with *z*-score	1; 5; **25** **50**; **100**; 500	9;7;**5** **5**;**5**;7	[11]
MIT linear discriminant classifier	N[9]	Y	MIT "F(x)" score[2]	16	6	[6]
Emerging patterns	N[11]	Y	entropy-based discretization	35	5	[16]
modified perceptron	N[9]	Y[9]	none	2000	7.5	[13]
clustering-based classifier (CAST + maximized compatibility score)	N[11]	Y	none	2000	7	[15]
NN/Pearson correlation metric	N[11]	Y	none	2000	12	[15]
linear SVM	N[11]	Y	none	2000	14	[15]
quadratic SVM	N[11]	Y	none	2000	16	[15]
recursive partitioning	N[20]	N (*a,p*)	entropy measure of node purity[21]	3	4–5	[19]

[15] Random partitioning, 38/24 (see Section 5.1, present work).

[16] Random split, 31/31.

[17] 42/20 split, first 42 + remaining 20 samples.

[18] 50 trials of different 50/12 splits; testing errors for feature selection algorithm **(a)**–**(d)** are averages computed over 50 trials.

[19] Five samples (N34, N36, T30, T33, and T36), deemed "likely to have been contaminated" were removed, leaving 57 samples. Three different train/test sets (40/17 split) were defined: **(a)** *original:* the first 40 samples were placed in the training set, the remainder in the test set; **(b)** *random:* 40 samples were randomly assigned to the training set, the remainder to the test set; **(c)** *discrepant:* the last 40 samples were placed in the training set, the rest in the test set. Testing errors are correspondingly labeled **(a)**, **(b)**, **(c)**.

[20] Five-fold cross-validation on full (*N*=62) dataset.

[21] [19] also tried a variant gene selection approach, "competitive node splits," but since this method is not explained or referenced in the paper, and gave inferior results, we note only that it was tried ("*a*" notation under "One-Shot Test") but do not report cross-validation results.

Table 8. Princeton dataset, continued.

CLASSIFIER	SEPARATE TRAIN/TEST?	ONE-SHOT TEST?	FEATURE SELECTION METHOD	# FEATURES	TESTING ERRORS	REF.
supervised naïve Bayes model	N[9]	N (*f*)	fold-independent subsets of *K*-class unsupervised NBGR ranking	2; **5**; 10 25; 50; 100	27;**13**;14 17;20;20	[20]
nonlinear (radial basis function) SVM	N[9]	N (*f*)	fold-independent subsets of *K*=2 supervised NBGR ranking	2; 5; 10 **25**; 50; 100	42;24;15 **14**;16;20	[20]
supervised naïve Bayes model	N[9]	N (*f*)	fold-independent subsets of *K*=2 supervised NBGR ranking	2; 5; 10 25; **50**; 100	29;33;29 28;**26**;28	[20]
linear SVM	N[9]	N (*f*)	SVM RFE	16–256	0	[6]
LD	N[9]	N (*f*)	*t*-score + PC	**50**; 100 500; 1000	**8**;9;9;10	[14]
LD	N[9]	N (*f*)	*t*-score+ PLS	**50**; **100** 500; 1000	**4**;**4**;6;5	[14]
QDA	N[9]	N (*f*)	*t*-score + PC	**50**; 100; 500; **1000**	**8**;10;9;**8**	[14]
QDA	N[9]	N (*f*)	*t*-score+ PLS	**50**; 100 **500**; 1000	5;6;**5**;6	[14]
nonlinear (radial basis function) SVM	N[9]	N (*f*)	fold-independent subsets of *K*-class unsupervised NBGR ranking	2; 5; 10 25; 50; 75; 100 150; **200**; 300; 400 500; 1000; 1250 1500; 1750	10; 10; 9 10; 7; 8; 7 7; **6**; 7; 7 10; 8; 8 7;7	[20]
nonlinear (radial basis function) SVM	N[9]	N	none	1988	7	[20]
linear SVM	N[9]	N (*f*)	MIT “F(x)” score[2]	1000; 2000	6;6	[13]
supervised naïve Bayes model	N[9]	N (*f*)	fold-independent subsets of *K*-class unsupervised NBGR ranking	2; **5**; 10 25; 50; 100	27;**13**;14 17;20;20	[20]
boosting with decision stump learner	N[11]	N (*p*)[14]	none	2000	17 **(a)** 17 **(b)** 18 **(c)**	[15]
linear SVM	N[11]	N (*f*)[13]	*K=2* supervised NBGR	4;5;11;25;50 100; **1988**	24;17;12;15;13 16;**9**	[10]
linear SVM	N[11]	N (*f*)[13]	MVR	4;**5**;11;25;50 100; **1988**	10;**9**;13;16;14 11;**9**	[10]
nonlinear (radial basis function) SVM, data-dependent range	N[11]	N (*f*)[13]	*K=2* supervised NBGR	4;5;11;25;50 **100**; **1988**	20;11;10;9;9 **8;8**	[10]
nonlinear (radial basis function) SVM, data-dependent range	N[11]	N (*f*)[13]	MVR	4;5;11;25;50 100; **1988**	12;13;12;12;10 10;**8**	[10]
nonlinear (radial basis function) SVM, fixed range	N[11]	N (*f*)[13]	*K=2* supervised NBGR	4;5;11;25;**50** 100; **1988**	22;20;8;9;**7** 9;**7**	[10]
nonlinear (radial basis function) SVM, fixed range	N[11]	N (*f*)[13]	MVR	4;5;11;25;**50** **100**; **1988**	8;10;9;10;**7** **7**;**7**	[10]

References

[1] Szabo, A., Boucher, K, Carroll, W. L., Klebanov, L.B., Tsodikov, A.D., and Yakovlev, A. Y. 2002. Variable selection and pattern recognition with gene expression data generated by the microarray technology. *Mathematical Biosciences* 176, 71–98.

[2] Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537.

[3] Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J. P., and Poggio, T. 1999. Technical Report AI Memo 1677, Massachussetts Institute of Technology (Cambridge, MA).

[4] Weston, J., Mikherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. Feature selection for SVMs. 2001. In: *Advances in Neural Information Processing Systems (NIPS)* 13, 668–674.

[5] Korenberg, M.J., 2002. Prediction of treatment response using gene expression profiles. *J. of Proteome Research 1*, 55–61.

[6] Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 389–422.

[7] Li, Y., Campbell, C., and Tipping, M. 2002. Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics* 18(10) 1332–1339.

[8] Dudoit, S., Fridlyand, J., and Speed, T. P. 2000. Comparison of discrimination methods for the classification of tumors using gene expression data. Technical Report #576, June, 2000, Department of Statistics, University of California, Berkeley (Berkeley, CA).

[9] Dudoit, S., Fridlyand, J., and Speed, T. P. 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc*. 97, 77-87 (short/edited version of [8]).

[10] M. L. Chow, E. J. Moler, and I. S. Mian. 2001. Identifying marker genes in transcription profiling data using a mixture of feature relevance experts. *Physiol. Genomics* 5, 99–111.

[11] Li, L., Darden, T. A., Weinberg, C. R., and Pedersen, L. G. 2001. Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method. *Combinatorial Chemistry and High-Throughput Screening* 4(8), 727–739.

[12] Lee, Y. and Lee, C.-K. 2002. Classification of multiple cancer types by multicategory support vector machines using gene expression data. Technical Report No. 1051r, Department of Statistics, University of Wisconsin (Madison, WI).

[13] Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., and Haussler, D. 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16(10), 906–914.

[14] Nguyen, D.V., and Rocke, D.M., 2002. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* 18(1) 39–50.

[15] Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., Yakhini, Z. 2000. J. Comp. Biol. 7(3,4), 559–584

[16] Li, J. and Wong, L. 2002. Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. Bioinformatics 18(5), 725–734.

[17] Nguyen, D.V., and Rocke, D.M., 2002. Multi-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics* 18(9) 1216–1226.

[18] Li, L., Weinberg, C. R., Darden, T. A., and Pedersen, L. G. 2001. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics* 17(12) 1131–1142.

[19] Zhang, H.P., Yu, C., Singer, B., and Xiong, M. 2001. Recursive partitioning for tumor classification with gene expression microarray data. *Proc. Natl. Acad. Sci. USA* 98, 6730–6735.

[20] Moler, E. J., Chow, M. L., and Mian, I. S. 2000. Analysis of molecular profile data using generative and discriminative methods. *Physiol. Genomics* 4, 109–126 (2000).

Key/Notes

1. Separate train/test = N ⇒ classification is via LOO training on full train+test dataset, followed by classification of the left-out sample.

2. One-shot test = Y ⇒ a single 'best shot' classifier is evaluated against the test set, after having been selected from various possible combinations of classifiers/feature sets developed using the training set only. If at *any* point in the algorithm (feature selection, *f*; classifier parameter tuning, *p*; classifier algorithm details, *a*) the test set is included in the procedure, this is considered to be "One-shot test = N." Test set results for multiple feature sets without an *a priori* (test-set-blind) method to select among feature sets are reported as N(*f*).

3. "Testing errors" should be interpreted as applying to the dataset implied by the "Separate train/test?" column. The errors include actual errors + samples characterized as 'unclassifiable.'

4. Train/test set is Golub *et al*. 38/34 (MIT) and 62 train+test (Princeton), unless otherwise indicated.

5. Boldface #genes/error #s correspond to best results in cases where one-shot test = N, and multiple results are reported.

Abbreviations

ARD	automatic relevance determination
BSS	between classes sum of squares
CART	classification and regression trees
DQDA	diagonal quadratic discriminant analysis
DLDA	diagonal linear discriminant analysis
DQDA	diagnonal quadratic discriminant analysis
FLDA	Fisher's linear discriminant analysis
GACV	generalized approximate cross validation
KS	Kolmogorov-Smirnov
LD	logistic discrimination
LDA	linear discriminant analysis
LOOCV	leave-one-out cross validation
MAR	mean aggregate relevance
MPLS	multivariate partial least squares
MVR	median vote relevance
NBR	naïve Bayes relevance
NBGR	naïve Bayes global relevance
NN	nearest neighbor
PAMD/CS	pairwise absolute mean differences/critical score (multiclass generalization of *t*-score)
PC	principal components
PCI	parallel cascade information
PD	polychotomous discrimination (multiclass generalization of logistic discrimination)
PLS	partial least squares
QDA	quadratic discriminant analysis
SVD	singular value decomposition
SVM	support vector machine
WSS	within class sum of squares

References

Alizadeh, A., Eisen, M., Davis, R., Ma, C., Lossos, I., Rosenwald, A., Boldrick, J., Sabet, H., 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., 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, 503–511.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *Proc. National Academy of Sciences USA* 96(12), 6745-6750.

Bhattacharjee, A., Richards, W., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E., Lander, E., Wong, W., Johnson, B., Golub, T., Sugarbaker, D., and Meyerson, M. 2001. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. In *Proc. National Academy of Sciences* 98(24), 13790–13795.

Ben-Dor, A., Bruhn, B., Friedman, N., Nachman, I., Schummer, M., and Yakhini, Z. 2000. Tissue classification with gene expression profiles. *J. Computational Biology* 7(3,4), 559–584.

Ben-Dor, A., Friedman, N., and Yakhini, Z. 2001. Class discovery in gene expression data. In *Proc. 5th Annual International Conference on Computational Biology*, 31–38, ACM Press.

Bishop, C. 1996. *Neural Networks for Pattern Recognition.* Oxford University Press, New York.

Brown, P., and Botstein, D. 1999. Exploring the new world of the genome with DNA microarrays. *Nature Genetics* 21, 33–37.

Buntine, W. 1991. Theory refinement for Bayesian networks. In *Proc. 7th Conference on Uncertainty in Artificial Intelligence (UAI)*, 52–60, Morgan Kaufmann.

Buntine, W. 1992. Learning classification trees. *Statistics and Computing* 2, 63–73.

Buntine, W. 1996. A guide to the literature on learning probabilistic networks from data. *IEEE Trans. on Knowledge and Data Engineering* 8, 195–210.

Cheng, J. and Greiner, R. 1999. Comparing Bayesian network classifiers. In *Proc. 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, 101–108, Morgan Kaufmann.

Chickering, D., Heckerman, D., and Meek, C. 1997. A Bayesian approach to learning Bayesian Networks with local structure. In *Proc. 13th Conference on Uncertainty in Artificial Intelligence (UAI)*, 80–89, Morgan Kaufmann.

Chow, M., Moler, E., and Mian, I. 2001. Identifying marker genes in transcription profiling data using a mixture of feature relevance experts. *Physiol. Genomics* 5, 99–111.

Cooper, G., and Herskovita, E. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* 9, 309–347.

Cozman, F. 2001. `http://www-2.cs.cmu.edu/~javabayes/Home/`.

D'haeseleer, P. 2000. Reconstructing gene networks from large scale gene expression data. Ph.D. dissertation. Computer Science Department, University of New Mexico. `http://www.cs.unm.edu/~patrik`.

Dawid, A., 1992. Applications of a general propagation algorithm for a probabilistic expert system. *Statistics and Computing* 2, 25–36.

Dawid, A., and Lauritzen, S. 1993. Hyper Markov laws in the statistical analysis of decomposable graphical models. *Annals of Statistics* 21, 1272–1317.

Dechter, R. 1996. Bucket elimination: a unifying framework for probabilistic inference. In *Proc. 12th Conference on Uncertainty in Artificial Intelligence (UAI)*, 211–219, Morgan Kaufmann.

DeRisi, J., Iyer, V., and Brown, P. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* 278, 680–686.

Ding, J. 2003. An analysis of binning heuristics using MDL and the BD metric. MS thesis in progress.

Duda, R., and Hart, P. 1973. *Pattern classification and scene analysis.* John Wiley and Sons, New York.

Duda, R., Hart, P., and Stork, D. 2000. *Pattern Classification (2nd Edition).* Wiley-Interscience, New York.

Dudoit, S., Fridlyand, J., and Speed, T. 2000. Comparison of discrimination methods for the classification of tumors using gene expression data. Technical Report 576, Department of Statistics, University of California, Berkeley.

Dudoit, S., Fridlyand, J., and Speed, T. 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc.* 97, 77–87.

Eisen, M., Spellman, P., Botstein, D., and Brown, P. 1998. Cluster analysis and display of genome-wide expression patterns. In *Proc. National Academy of Sciences USA* 95, 14863–14867.

Fayyad, U., and Irani, K. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, 1022–1027.

Friedman, N., and Goldszmidt, M. 1996a. Discretizing continuous attributes while learning Bayesian networks. In *Proc. 13th International Conference on Machine Learning (ICMS)*, 157–165.

Friedman, N., and Goldszmidt, M. 1996b. Learning Bayesian networks with local structure. In *Proc. 12th Conference on Uncertainty in Artificial Intelligence (UAI)*, 211–219, Morgan Kaufmann.

Friedman, N., Geiger, D., and Goldszmidt, M. 1997. Bayesian network classifiers. *Machine Learning* 29, 131–163.

Friedman, N., Nachman, I., and Pe'er, D. 1999. Learning Bayesian network structure from massive datasets: the sparse candidate algorithm. In *Proc. 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, 196–205, Morgan Kaufmann.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. 2000. Using Bayesian networks to analyze expression data. *J. Computational Biology* 7(3,4), 601–620.

Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., and Haussler, D. 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16(10), 906–914.

Gander, W., and Gautschi, W. 2000. Adaptive Quadrature Revisited. *BIT* 40(1), 84–101.

Getz, G., Levine, E., and Domany. E. 2000. Coupled two-way clustering analysis of gene microarray data. In *Proc. National Academy of Sciences* 97(22), 12079–12084.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caliguiri, M., Bloomfield, C., and Lander, E. 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 389–422.

Han, C., and Carlin, B. 2000. MCMC methods for computing Bayes factors: a comparative review. Technical Report, Division of Biostatistics, University of Minnesota.

Heckerman, D., Geiger, D., and Chickering, D. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 20, 197–243.

Helman, P., and Bhangoo, J. 1997. A statistically based system for prioritizing information exploration under uncertainty. *IEEE Trans. on Systems, Man, and Cybernetics* 27(4), 449–466, 1997.

Helman, P., and Gore, R. 1998. Prioritizing information for the discovery of phenomena. *J. of Intelligent Information Systems* 11(2), 99–138.

Ibrahim, J., Chen, M., and Gray, R. 2002. Bayesian models for gene expression with DNA microarray data. *J. American Statistical Association* 97(457), 88–99.

Jensen, F., Lauritzen, S., and Olesen, K. 1990. Bayesian updating in causal probabilistic networks by local computations, *Computational Statistics Quarterly* 4, 269–282.

Jensen, F., 2001. *Bayesian networks and decision graphs.* Springer-Verlag, New York.

Kang, H., and Atlas, S. 2003. Methods for evaluating performance of a class predictor. Technical Report, High Performance Computing Education and Research Center, University of New Mexico.

Khan, J., Wei, J., Ringner, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C., Peterson, C., and Meltzer, P. 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* 7, 673–679.

Korenberg, M. 2002. Prediction of treatment response using gene expression profiles. *J. of Proteome Research* 1, 55–61.

Lam, W., and Bacchus, F. 1994. Learning Bayesian belief networks: an approach based on the MDL principle, *Computational Intelligence* 10, 269–293.

Langley, P., Iba, W., and Thompson, K. 1992. An analysis of Bayesian classifiers. In *Proc. 10th National Conference on Artificial Intelligence*, 223–228, AAAI Press.

Lauritzen, S., and Spiegelhalter, D. 1988. Local computations with probabilities on graphical structures and their applications to expert systems, *J. Royal Statistical Society* Series B 50, 157–224.

Lee, Y., and Lee, C. 2002. Classification of multiple cancer types by multicategory support vector machines using gene expression data. Technical Report No. 1051r, Department of Statistics, University of Wisconsin, Madison.

Li, J., and Wong, L. 2002. Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. *Bioinformatics* 18(5), 725–734.

Li, L., Darden, T., Weinberg, C., and Pedersen, L. 2001a. Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method. *Combinatorial Chemistry and High-Throughput Screening* 4(8), 727–739.

Li, L., Weinberg, C., Darden, T., and Pedersen, L. 2001b. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics* 17(12), 1131–1142.

Li, Y., Campbell, C., and Tipping, M. 2002. Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics* 18(10), 1332–1339.

Lockhart, D., Dong, H., Byrne, M., Follettie, M., Gallo, M., Chee, M., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., and Brown, E. 1996. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology* 14(12), 1675.

MacKay, D. 1995. Probable networks and plausible predictions—a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems* 6, 469–505.

Madden, M. 2002. Evaluation of the performance of the Markov Blanket Bayesian Classifier Algorithm. Technical Report NUIG-IT-0110002, Dept. of Information Technology, National University of Ireland, Galway.

Madigan, D., and York, J. 1995. Bayesian graphical models for discrete data. *International Statistics Review* 63, 215–232.

Madsen, A., and Jensen, F. 1999. Lazy propagation: a junction tree inference algorithm based on lazy evaluation, *Artificial Intelligence* 113, 203–245.

Moler, E., Chow, M., and Mian, I. 2000. Analysis of molecular profile data using generative and discriminative methods. *Physiol. Genomics* 4, 109–126.

Mosquera-Caro, M., *et al.* 2003a. Gene expression profiling for molecular classification and outcome prediction in infant leukemia reveals novel biologic clusters, etiologies, and pathways for treatment failure, in preparation.

Mosquera-Caro, M., *et al.* 2003b. Heterogeneity of gene expression profiles in MLL-associated infant leukemia: identification of distinct expression profiles and novel therapeutic targets for each MLL translocation variant, in preparation.

Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J. P., and Poggio, T. 1999. Support vector machine classification of microarray data. Technical Report AI Memo 1677, Massachusetts Institute of Technology, Cambridge.

Murphy, K., and Mian, S. 1999. Modelling gene expression data using dynamic Bayesian networks. Technical Report, Computer Science Division, University of California, Berkeley. `http://www.cs.berkeley.edu/~murphyk/Papers/ismb99.ps.gz`.

Nguyen, D., and Rocke, D. 2002. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* 18(1), 39–50.

Nguyen, D., and Rocke, D. 2002. Multi-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics* 18(9), 1216–1226.

Pearl, J. 1988. *Probabilistic reasoning for intelligent systems.* Morgan Kaufmann, San Francisco.

Pearl, J., and Verma, T. 1991. A theory of inferred causation. In *Knowledge Representation and Reasoning: Proc. 2nd International Conference*, 411–452, Morgan Kaufmann.

Pe'er, D., Regev, A., Elidan, G., and Friedman, N. 2001. Inferring subnetworks from perturbed expression profiles. *Bioinformatics* 17, S215–S224.

Pomeroy, S., Tamayo, P., Gassenbeek, M., Sturla, L., Angelo, M., McLaughlin, M., Kim, J., Goumnerova, L., Black, P., Lau, C., Allen, J., Zagzag, D., Olson, J., Curran, T., Wetmore, C., Blegel, J., Pogglo, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, G., Louis, D., Mesirov, J., Lander, E., and Golub, T. 2002. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature* 415, 436–442.

Rigoutsos, I., Floratos, A., Parida, L., Gao, Y., and Platt, D. 2000. The emergence of pattern discovery techniques in computational biology. *Metabolic Engineering* 2(3), 159–177.

Schena, M., Shalon, D., Davis, R., and Brown, P. 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science, New Series* 270(5235), 467–470.

Shafer, G., and Shenoy, P. 1990. Probability propagation. *Annals of Mathematics and Artificial Intelligence* 2, 327–352.

Spiegelhalter, D., Dawid, D., Lauritzen, S., and Cowell, R. 1993. Bayesian analysis in expert systems. *Statistical Science* 8, 219–292.

Szabo, A., Boucher, K, Carroll, W., Klebanov, L., Tsodikov, A., and Yakovlev, A. 2002. Variable selection and pattern recognition with gene expression data generated by the microarray technology. *Mathematical Biosciences* 176, 71–98.

Tavazoie, S., Hughes, J., Campbell, M., Cho, R., and Church, G. 1999. Systematic determination of genetic network architecture. *Nature Genetics* 22, 281–285.

Tobin, F., Damian-Iordache, V., and Greller, L. 1999. Towards the reconstruction of gene regulatory networks. In *Technical Proc. 1999 International Conference on Modeling and Simulation of Microsystems*, San Juan, Puerto Rico.

van't Veer, L., Dal, 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, 530–536.

Vapnik, V. 1998. *Statistical Learning Theory.* Wiley-Interscience, New York.

Weston, J., Mikherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. 2001. Feature selection for SVMs. In Advances in Neural Information Processing Systems (NIPS) 13, 668–674.

Woolf, P., and Wang, Y. 2000. A fuzzy logic approach to analyzing gene expression data. *Physiol. Genomics* 3, 9–15.

Zhang, H., Yu, C., Singer, B., and Xiong, M. 2001. Recursive partitioning for tumor classification with gene expression microarray data. *Proc. National Academy of Science USA* 98, 6730–6735.