1 The Use of Emerging Patterns in the Analysis of Gene Expression Profiles for the Diagnosis and Understanding of Diseases

GUOZHU DONG

Wright State University

Email:gdong@cs.wright.edu

JINYAN LI

Institute for Infocomm Research

Email:jinyan@i2r.a-star.edu.sg

LIMSOON WONG

Institute for Infocomm Research

Email:limsoon@i2r.a-star.edu.sg

1.1 INTRODUCTION

Microarrays are glass surfaces bearing arrays of DNA fragments at discrete addresses. These DNA fragments on the microarray are hybridized to a complex sample of fluorescently labeled DNA or RNA in solution. After a washing and staining process, the addresses at which hybridization has taken place can be determined and the expression level of the corresponding genes derived. Today, a single microarray can contain several tens of thousands of DNA fragments. Thus, microarrays are a technology for simultaneously profiling the expression levels of tens of thousands of genes in a patient sample.

It is hopeful that better diagnosis methods and better understanding of disease mechanisms can be derived from a careful analysis of microarray measurements of gene expression profiles. This chapter discusses several types of analysis of such gene expression profiles using a form of supervised learning based on the idea of emerging patterns. The types of analysis discussed are (a) diagnosis of disease state or subtype, (b) derivation of disease treatment plan, and (c) understanding of gene interaction networks.

The first type of analysis mentioned above reasonably postulates that the expression levels of various genes in a patient are dependent on his/her disease state and/or subtype. Therefore, by a careful analysis of gene expression profiles, we can hopefully determine the signature pattern of gene expression profiles associated with specific disease states and/or subtypes that are useful for diagnosis purposes. The second and third types of analysis mentioned above suggest the converse that the state of a disease can be affected by the expression levels of certain genes. That is, the improper expression of these genes is the cause of the disease. Hence, by a careful analysis of gene expression profiles, one might be able to infer such "causal" genes and to plan a course of treatment to modulate these genes.

This chapter is organized as follows. Section 1.2 discusses a method of classification/prediction, called PCL, that uses collective likelihood based on emerging patterns. Section 1.3 deals with the selection of relevant genes. Section 1.4 considers the diagnosis of disease states or subtypes using PCL. Section 1.5 presents an

approach to deriving treatment plans based on emerging patterns. Section 1.6 discusses approaches to understanding molecular circuits. Section 1.7 provides closing remarks.

1.2 PREDICTION BY COLLECTIVE LIKELIHOOD BASED ON EMERGING PATTERNS

In the field of machine learning, there are many good prediction methods including k-nearest neighbours (k-NN) [23], C4.5 [100], support vector machines (SVM) [17], Naive Bayes (NB) [65], etc. C4.5 is a widely used learning algorithm that induces from training data rules that are easy to comprehend. However, it may not have good performance if real decision boundary underlying the data is not linear. The Naive Bayes model uses Bayesian rules to estimate probabilistic score for each class. When given a test sample, NB uses the probabilistic scores to rank the classes, and assigns the sample to the highest scoring class. An important assumption used in NB is that the underlying features are statistically independent. However, this is not appropriate for gene expression data analysis as genes involved in an expression profile are often closely related and appear not to be independent. The k-nearest neighbour method assigns a test sample the class of its nearest training sample in terms of some distance functions. Even though the k-NN is intuitive and has good performance, it is not helpful for understanding complex cases in depth. The support vector machines use non-linear kernel functions to construct a complicated mapping between samples and their class labels. SVM has good performance, but it functions as a black box. Similarly, traditional datamining methods that look for high frequency patterns are frequently not useful on gene expression data. We are therefore motivated to seek a classification method that enjoys the advantages of both high accuracy and high comprehensibility. In this section, we describe the method of Prediction by Collective Likelihood based on emerging patterns (PCL).

PCL [71, 67] is a classification method that we have been developing during the last couple of years. This method focuses on (a) fast techniques for identifying patterns whose frequencies in two classes differ by a large ratio [26], which are the

so-called **emerging patterns**; and on (b) combining these patterns to make decisions. A pattern is a set of expression conditions of the form $c_{i1} \leq G_i < c_{i2}$ if feature G_i is numerical, and of the form $G_i = c_i$ if feature G_i is discretized. Often we discretize numerical features first. Example emerging patterns are given in Section 1.4. Note that a pattern is still emerging if its frequencies are as low as 1% in one class and 0.1% in another class, because the ratio indicates a 10 times difference. However, for the purpose of PCL, we use only emerging patterns that are most general and have infinite ratio. That is, we use only emerging patterns that occur in one class of data but not the other class and that do not contain any subsets which are also infinite-ratio emerging patterns. From now on by emerging patterns we mean infinite-ratio most-general emerging patterns.

Basically, the **PCL classifier** has two phases. Given two training datasets $D^{\mathcal{A}}$ and $D^{\mathcal{B}}$ (instances of classes $\mathcal{A}$ and $\mathcal{B}$, resp.), PCL first discovers two groups of most general emerging patterns from $D^{\mathcal{A}}$ and $D^{\mathcal{B}}$. Denote the most general emerging patterns of $D^{\mathcal{A}}$ as, $EP_1^{\mathcal{A}}$, $EP_2^{\mathcal{A}}$, $\cdots$, $EP_i^{\mathcal{A}}$, in descending order of frequency. Similarly denote the most general emerging patterns of $D^{\mathcal{B}}$ as $EP_1^{\mathcal{B}}$, $EP_2^{\mathcal{B}}$, $\cdots$, $EP_j^{\mathcal{B}}$. Let T be a test sample. Suppose T contains the following most general emerging patterns of $D^{\mathcal{A}}$: $EP_{i_1}^{\mathcal{A}}$, $EP_{i_2}^{\mathcal{A}}$, $\cdots$, $EP_{i_x}^{\mathcal{A}}$, $i_1 < i_2 < \cdots < i_x \leq i$, and the following most general emerging patterns of $D^{\mathcal{B}}$: $EP_{j_1}^{\mathcal{B}}$, $EP_{j_2}^{\mathcal{B}}$, $\cdots$, $EP_{j_y}^{\mathcal{B}}$, $j_1 < j_2 < \cdots < j_y \leq j$. Next, PCL calculates two scores for predicting the class label of T. Suppose we use k ($k \ll i$ and $k \ll j$) top-ranked most general emerging patterns of $D^{\mathcal{A}}$ and $D^{\mathcal{B}}$. Then we define the score of T in the $D^{\mathcal{A}}$ class as

$$score(T, D^{\mathcal{A}}) = \sum_{m=1}^{k} \frac{frequency(EP_{i_m}^{\mathcal{A}})}{frequency(EP_m^{\mathcal{A}})},$$

and the score in the $D^{\mathcal{B}}$ class is similarly defined in terms of $EP_{j_m}^{\mathcal{B}}$ and $EP_m^{\mathcal{B}}$. The use of summation allows us to combine signals captured by different emerging patterns, and the use of ratio allows us to somehow normalize the scores for situations where one class has many strong (high frequency) emerging patterns but another class has very few or even no strong emerging patterns. If $score(T, D^{\mathcal{A}}) > score(T, D^{\mathcal{B}})$, then T's predicted class is $D^{\mathcal{A}}$. Otherwise its predicted class is $D^{\mathcal{B}}$. We use the size

of $D^{\mathcal{A}}$ and $D^{\mathcal{B}}$ to break tie. The PCL classifier has proved to be a good tool for analysing gene expression data and proteomic data [67, 131, 71, 73, 68, 74].

Clearly PCL requires an efficient method to discover emerging patterns that are most general (and that have infinite ratio). Observe that there are 2^n possible patterns contained in each tuple of $D^{\mathcal{A}}$ and $D^{\mathcal{B}}$, if $D^{\mathcal{A}}$ and $D^{\mathcal{B}}$ have n attributes. Hence naive methods to extract emerging patterns would be too expensive. More efficient methods for extracting emerging patterns are therefore crucial to the operation of PCL. We briefly discuss here an approach for developing more practical algorithms for finding such emerging patterns. Let us begin with a theoretical observation:

Proposition 1.2.1 (Cf. [69]) *The collection of all (infinite-ratio) emerging patterns from $D^{\mathcal{A}}$ to $D^{\mathcal{B}}$ form a convex space, i.e., for all emerging patterns X and Y and for each Z such that $X \subseteq Z \subseteq Y$, Z is also an emerging pattern.* ∎

A convex space C can be represented by a pair of borders $\langle L, R\rangle$, so that (a) both L and R are anti-chains, (b) each $X \in L$ is more general than some $Z \in R$ (i.e., Z is a superset of X), (c) each $Z \in R$ is more specific than some $X \in L$, and (d) $C = \{Y \mid \exists X \in L, \exists Z \in R, X \subseteq Y \subseteq Z\}$. Actually, L consists of the most general, and R the most specific, patterns in C. We can write $[L, R]$ for C. Observe that

Proposition 1.2.2 *Suppose $D^{\mathcal{A}}$ (respectively, $D^{\mathcal{B}}$) has no duplicate and its tuples are of the same dimension. Then the set of emerging patterns of $D^{\mathcal{A}}$ is given by $[\{\{\}\}, D^{\mathcal{A}}] - [\{\{\}\}, D^{\mathcal{B}}]$, and the set of emerging patterns of $D^{\mathcal{B}}$ is given by $[\{\{\}\}, D^{\mathcal{B}}] - [\{\{\}\}, D^{\mathcal{A}}]$.* ∎

Having reduced emerging patterns to this border formulation, we can derive a more efficient approach to discovering them. Let $A_1, \ldots, A_n$ be the tuples of $D^{\mathcal{A}}$ that do not occur in $D^{\mathcal{B}}$, and let $D^{\mathcal{B}} = \{B_1, \ldots, B_m\}$. Then

$$\begin{array}{rl} & [\{\{\}\}, D^{\mathcal{A}}] - [\{\{\}\}, D^{\mathcal{B}}] \\ = & [\{\{\}\}, \{A_1, \ldots, A_n\}] - [\{\{\}\}, \{B_1, \ldots, B_m\}] \\ = & [L, \{A_1, \ldots, A_n\}] \end{array}$$

where

$$L = \bigcup_{i}^{n} min\{\{s_1, \ldots, s_m\} | s_j \in A_i - B_j, 1 \leq j \leq m\}.$$

Citations [26, 69, 132] give fairly efficient methods to compute L, using novel border-based algorithms and constraint-based algorithms. Citation [127] gives results showing that emerging pattern mining is hard theoretically.

In the remainder of this chapter, we discuss the use of emerging patterns and PCL in the analysis of microarray gene expression profiles.

1.3 SELECTION OF RELEVANT GENES

probe	pos	neg	pairs in avg	avg diff	abs call	Description
...	...	...	...	...	...	...
106_at	4	1	15	1527.6	A	Z35278 Human PEBP2aC1 ...
107_at	4	4	15	3723.3	A	Z95624 Human DNA from ...
108_g_at	5	2	15	1392.4	A	Z95624 Human DNA ...
109_at	6	2	16	2274.7	M	Z97074 Human mRNA for ...
...	...	...	...	...	...	...

Fig. 1.1 A partial example of a processed microarray measurement record of a patient sample using the Affymetrix U95A Gene Chip. Each row represents a probe. Typically each probe represents a gene. The U95A Gene Chip contains more than 12,000 probes. The 5th column contains the gene expression measured by the corresponding probe. The 2nd, 3rd, 4th, and 6th columns are quality control data. The 1st and last columns are the probe identifier and a description of the corresponding gene.

A single microarray experiment can measure the expression level of tens of thousands of genes simultaneously [76, 102]. In other words, the microarray experiment record of a patient sample—see Figure 1.1 for an example—is a record having tens of thousands of features or dimensions. This extremely high dimensionality causes many problems to existing datamining and machine learning methods. One such problem is that of efficiency because most datamining and machine learning methods have time complexity that are high with respect to the number of dimensions [47]. Another such problem is that of noise because most datamining and machine learning methods suffer from the "curse of dimensionality"—these methods typically require

an exponential increase in the number of training samples with respect to an increase in the dimensionality of the samples in order to uncover and learn the relationship of the various dimensions to the nature of the samples [48].

Let us assume that we have two classes $\mathcal{A}$ and $\mathcal{B}$ of microarray gene expression profiles of patient samples. For example, $\mathcal{A}$ could be gene expression profiles of colon tumour cells and $\mathcal{B}$ could be gene expression profiles of normal (matching) cells. Then a feature—in this case, a gene—is relevant if it contributes to separating samples in $\mathcal{A}$ from those in $\mathcal{B}$. Conversely, a feature may be irrelevant if it does not contribute much to separating samples in $\mathcal{A}$ from those in $\mathcal{B}$. In order to alleviate the impact of the problems caused by high dimensionality mentioned above, it is desirable to first discard as many features that are irrelevant as possible. In this section, we present several techniques for deciding whether a feature is relevant, viz. t-statistics, signal-to-noise, and entropy measures.

A basic approach for selecting relevant features is the following: if the values of a feature in samples in $\mathcal{A}$ are significantly different from the values of the same feature in samples in $\mathcal{B}$, then the feature is likely to be more relevant than a feature that has similar values in $\mathcal{A}$ and $\mathcal{B}$.

One concept to capture significant difference among feature values in multiple classes is to use the difference between mean values of a feature in the different classes. However, we caution that the mean difference itself is not good enough for selecting relevant features. Indeed, if the values of a feature f varies greatly within the same class of samples, even if $\mu_f^{\mathcal{A}}$ differs greatly from $\mu_f^{\mathcal{B}}$, the feature f is still not a reliable one.

The deficiency of the mean difference concept leads us to a second concept to capture significant difference among feature values in multiple classes: the standard deviation $\sigma_f^{\mathcal{A}}$ of f in $\mathcal{A}$ and the standard deviation $\sigma_f^{\mathcal{B}}$ of f in $\mathcal{B}$. We will also use the variance $(\sigma_f^{\mathcal{A}})^2$ and $(\sigma_f^{\mathcal{B}})^2$ which can be derived from the standard deviation.

One way to combine these two concepts is the **signal-to-noise measure**, proposed in the first paper [45] that applied gene expression profiling for disease diagnosis,

$$s(f, \mathcal{A}, \mathcal{B}) = \frac{|\mu_f^{\mathcal{A}} - \mu_f^{\mathcal{B}}|}{\sigma_f^{\mathcal{A}} + \sigma_f^{\mathcal{B}}}$$

However, the statistical property of $s(f, \mathcal{A}, \mathcal{B})$ is not fully understood. Subsequently, a second and older way—the t-test—to combine these two concepts was rediscovered. The classical t-test statistical measure [10, 18] is known to follow a Student distribution with $(h(\mathcal{A})+h(\mathcal{B}))^2/((h(\mathcal{A})^2/(n^{\mathcal{A}}-1))+(h(\mathcal{B})^2/(n^{\mathcal{B}}-1)))$ degrees of freedom, where $h(\mathcal{C}) = (\sigma_f^{\mathcal{C}})^2/n^{\mathcal{C}}$, $n^{\mathcal{A}}$ and $n^{\mathcal{B}}$ are respectively the number of samples in $\mathcal{A}$ and $\mathcal{B}$. The **t-test statistical measure** is given by,

$$t(f, \mathcal{A}, \mathcal{B}) = \frac{|\mu_f^{\mathcal{A}} - \mu_f^{\mathcal{B}}|}{\sqrt{(\sigma_f^{\mathcal{A}})^2/n^{\mathcal{A}} + (\sigma_f^{\mathcal{B}})^2/n^{\mathcal{B}}}}$$

Both of these measures are easy to compute and thus are straightforward to use. However, these measures have three significant deficiencies in the context of gene expression profiles. Firstly, in gene expression profile experiments, the population sizes $n^{\mathcal{A}}$ and $n^{\mathcal{B}}$ are often as small as 2 or 3. These small population sizes can lead to significant underestimates of the standard deviations and variances. Secondly, due to some technological limitations of microarrays, there is no guarantee that two measurements of gene expression values taken from the same sample will agree with each other. That is, the value of a gene f may be different in these two microarray measurements taken from the same sample. However, if the ranges of f in $\mathcal{A}$ and $\mathcal{B}$ do not overlap, then the variances with respect to $\mathcal{A}$ and $\mathcal{B}$ should not matter all that much. Unfortunately, both $t(f, \mathcal{A}, \mathcal{B})$ and $s(f, \mathcal{A}, \mathcal{B})$ are sensitive to small changes in the values of f. Thirdly, the t-test statistical measure requires the gene expression values to follow the Student's distribution or a nearly normal distribution. For some experiments, the gene expression values may not follow such distributions.

So, one should consider alternative statistical measures that are less sensitive to changes in the value of f that are unimportant in the sense that they do not shift the value of f from the range in $\mathcal{A}$ into the range of $\mathcal{B}$. One such idea is the entropy measure [34]. Let $P(f, \mathcal{C}, S)$ be the proportion of samples whose feature f has value in the range S and are in class $\mathcal{C}$. The *class entropy* of a range S with respect to feature f and a collection of classes $\mathcal{U}$ is defined as $Ent(f, \mathcal{U}, S) = -\sum_{\mathcal{C}\in\mathcal{U}} P(f, \mathcal{C}, S) log(P(f, \mathcal{C}, S))$. Let T partitions the values of f into two ranges S_1 (of values less than T) and S_2 (of values at least T). We sometimes refer to

T as the **cutting point** of the values of f. The **entropy measure** $e(f, \mathcal{A}, \mathcal{B})$ of a feature f is then defined as $\min\{E(f, \{\mathcal{A}, \mathcal{B}\}, S_1, S_2) \mid (S_1, S_2)$ is a partitioning of the values of f in $\mathcal{A}$ and $\mathcal{B}$ by some point $T\}$. Here, $E(f, \{\mathcal{A}, \mathcal{B}\}, S_1, S_2)$ is the *class information entropy* of partition (S_1, S_2). The definition is given below, where $n(f,\mathcal{U}, S)$ denotes the number of samples in the classes in $\mathcal{U}$ whose feature f has value in the range S,

$$E(f,\mathcal{U}, S_1, S_2) = \sum_{i=1}^{2} \frac{n(f,\mathcal{U}, S_i)}{n(f,\mathcal{U}, S_1 \cup S_2)} Ent(f,\mathcal{U}, S_i)$$

A refinement of the entropy measure is to recursively partition the ranges S_1 and S_2 until some stopping criteria is reached [34]. A commonly used stopping criteria is the so-called minimal description length principle. Another refinement is the $\mathcal{X}^2$ measure [75].

All of the preceding measures provide a rank ordering of the features in terms of their relevance to separating $\mathcal{A}$ and $\mathcal{B}$. One would rank the features using one of these measures and select the top n features. However, one must appreciate that there may be a variety of independent reasons why a sample is in $\mathcal{A}$ or is in $\mathcal{B}$. For example, there can be a number of different pathways via which a cell becomes cancerous and there can be a number of different pathways via which a disease cell becomes of a specific subtype. If a primary pathway involves n genes, the procedure above may select only these n genes and may ignore genes in other secondary pathways. Consequently, concentrating on such top n features may cause us to lose sight of the secondary pathways underlying the disease.

This issue above calls for a different approach to feature selection: select a group of features that are correlated with separating $\mathcal{A}$ and $\mathcal{B}$ but are not correlated with each other. The cardinality in such a group may suggest the number of independent factors that underly the separation of $\mathcal{A}$ and $\mathcal{B}$. A well-known technique that implements this feature selection strategy is the Correlation-based Feature Selection (CFS) method [46]. Rather than scoring and ranking individual features, the CFS method scores and ranks the worth of subsets of features. As the feature subset space is usually huge, CFS uses a best-first-search heuristic. This heuristic algorithm takes into account the usefulness of individual features for predicting the class along with the

level of intercorrelation among them. CFS first calculates a matrix of feature-class and feature-feature correlations from the training data. Then a score of a subset of features is assigned by a heuristic. CFS starts from the empty set of features and uses the best-first-search heuristic with a stopping criterion of 5 consecutive fully expanded non-improving subsets. The subset with the highest merit found during the search will be selected.

Note that even if each gene selected by CFS is associated with a distinct pathway underlying the separation of $\mathcal{A}$ and $\mathcal{B}$, there is no guarantee that these genes will lead to good results by themselves. Each pathway involves multiple genes acting in a coordinated fashion. In methods such as the entropy measure, one is more likely to select all the genes in a primary pathways and neglect those of secondary pathways. In methods such as CFS, one is more likely to select the more important gene in each pathways and neglect the secondary genes. However, to get the best analysis results and to achieve the best understanding, it is crucial to know all of the relevant pathways and all of the genes relevant in each pathway. This ideal remains a significant challenge in research in feature selection methods.

Nevertheless, empirical evidence [68] suggests that so long as $\mathcal{A}$ and $\mathcal{B}$ are relatively homogeneous, the entropy measure and its refinements can make a good selection of relevant genes from microarray gene expression profiles. For example, comparing Figure 1.3 and Figure 1.4 in the next section, we see that prediction errors are significantly reduced by selecting relevant genes as described earlier. Furthermore, this reduction in prediction errors is universal across all the prediction algorithms used.

We want to make a final note of caution in performing feature selection to microarray gene expression profiles. The collection of gene expression profiles should be divided into a training set and a testing set. Selection of relevant genes should be made on the basis of the training set only. If there is insufficient gene expression profiles to divide into separate training and testing sets, then a k-fold cross validation strategy should be used and a fresh selection should be made for each fold using the training portion of that fold. To appreciate the importance of this caution, let us visit a simulation experiment reported by Miller et al. [88]. They constructed an artificial

data set with 100 samples. Each sample contains 100,000 random expression values and has a randomly assigned class. They then selected the 20 genes with the smallest p values determined by the Wilcoxon rank sum test. They evaluated the accuracy of using these 20 genes in class prediction by leave-one-out cross validation. The resultant estimated accuracy was 88%. However, as the data are derived from random assignments, the true accuracy must be only 50%. This is clearly inappropriate. Ambroise and McLachlan [5] provide additional examples that illustrate this issue.

1.4 DIAGNOSIS OF DISEASE STATE OR SUBTYPE

A major excitement generated by microarrays in the biomedical world is the possibility of using microarrays to diagnose disease states or disease subtypes in a way that is more efficient and more effective than conventional techniques [45, 131, 3, 40, 96]. Let us consider the diagnosis of childhood leukaemia subtypes as an illustration. Childhood leukaemia is a heterogeneous disease comprising more than 10 subtypes, including T-ALL, E2A-PBX1, TEL-AML1, BCR-ABL, MLL, Hyperdiploid$>$50, and so on. The response of each subtype to chemotherapy is different. Thus the optimal treatment plan for childhood leukaemia depends critically on the subtype. Conventional childhood leukaemia subtype diagnosis is a difficult and expensive process [131]. It requires intensive laboratory studies comprising cytogenetics, immunophenotyping, and molecular diagnostics. Usually, these diagnostic approaches require the collective expertise of a number of professionals comprising hematologists, oncologists, pathologists, and cytogeneticists. Although such combined expertise is available in major medical centers in developed countries, it is generally unavailable in less developed countries. It is therefore very exciting if microarrays and associated automatic gene expression profile analysis can serve as a single easy-to-use platform for subtyping of childhood leukaemia. This section applies the ideas of emerging patterns and PCL on a large childhood leukaemia dataset to perform subtype diagnosis.

We show the results of PCL on the dataset reported in Yeoh et al. [131]. The whole dataset consists of gene expression profiles of 327 childhood accute lymphoblastic

leukaemia (ALL) samples. These profiles were obtained by hybridization on the Affymetrix U95A GeneChip containing probes for 12558 genes. The data contain all the known acute lymphoblastic leukemia subtypes, including T-ALL, E2A-PBX1, TEL-AML1, BCR-ABL, MLL, and Hyperdiploid>50. The data were divided by Yeoh et al. into a training set of 215 instances and an independent test set of 112 samples. There are 28, 18, 52, 9, 14, and 42 training instances and 15, 9, 27, 6, 6, and 22 test samples respectively for T-ALL, E2A-PBX1, TEL-AML1, BCR-ABL, MLL, and Hyperdiploid>50. There are also 52 training and 27 test samples of other miscellaneous subtypes. The original training and test data were layered in a tree-structure, as shown in Figure 1.2. Given a new sample, we first check if it is T-ALL. If it is not classified as T-ALL, we go to the next level and check if it is a E2A-PBX1. If it is not classified as E2A-PBX1, we go to the third level and so on.

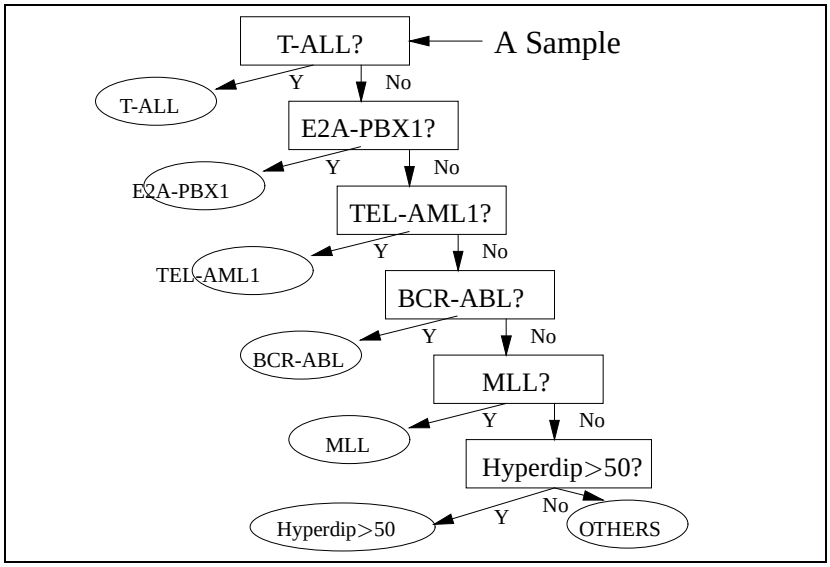

Fig. 1.2 The classification of the ALL subtypes is organized in a tree. Given a new sample, we first check if it is T-ALL. If it is not classified as T-ALL, we go to the next level and check if it is a E2A-PBX1. If it is not classified as E2A-PBX1, we go to the third level and so on.

In applying PCL to this dataset, at each level of the tree, we first use the entropy measure described in the previous section to select the 20 genes that have the lowest

entropy in that level's training data. Then we extract emerging patterns of that level involving just these 20 genes using the training set of that level. After the discretization of these top-ranked genes, we use border-based algorithms [26, 69] to discover the most general emerging patterns. Then these emerging patterns are used by PCL to predict the subtypes of test instances of that level. For comparison, we have also applied several popular classification methods—C4.5, SVM, and Naive Bayes(NB)—to the same datasets after filtering using the same selected genes. In each of these comparison methods, the default settings of the weka package (`http://www.cs.waikato.ac.nz/ml/weka`) was used. In the PCL case, the parameter k was set to 20. The number of false predictions on the test instances, after filtering by selecting relevant genes as described above, at each level of the tree by PCL, as well as those by C4.5, SVM, and NB, is given in Figure 1.3. The results of the same algorithms but without filtering by selecting relevant genes beforehand is given in Figure 1.4. The number of false predictions by PCL is less than that made by the other methods. We have also tried using different number of genes and different selection methods and different values of the parameter k in PCL, the number of false predictions by PCL is consistently less than that made by other methods [68]. Similar results are also obtained when a parallel classification scheme is used in place of the tree-structured scheme [67].

PCL has high accuracy and the underlying emerging patterns identified can also be translated into highly comprehensible rules. Let us illustrate this point about comprehensibility using some of the top emerging patterns from the ALL study above. In the prediction of the substype E2A-PBX1 versus other subtypes, the gene 32063_at has perfect entropy measure when its expression range is partitioned at the point 4068.7. The two emerging patterns induced by this partitioning have very high support: the expression of 32063_at in all E2A-PBX1 samples are greater than 4068.7 and the expression of 32063_at in all other subtypes are less than 4068.7. In other words, the emerging pattern, $\{32063\_\text{at} \geq 4068.7\}$, and the emerging pattern, $\{32063\_\text{at} < 4068.7\}$, yield two rules that are 100% valid:

if 32063_at $\geq$ 4068.7, then the sample is E2A-PBX1;

if 32063_at $<$ 4068.7, then the sample is OTHERS2.

Testing Data	Error rate of different models			
	C4.5	SVM	NB	PCL ($k = 20$)
T-ALL vs OTHERS1	0:1	0:0	0:0	0:0
E2A-PBX1 vs OTHERS2	0:0	0:0	0:0	0:0
TEL-AML1 vs OTHERS3	1:1	0:1	0:1	1:0
BCR-ABL vs OTHERS4	2:0	1:1	2:2	1:1
MLL vs OTHERS5	1:1	0:0	0:0	0:0
Hyperdiploid>50 vs OTHERS	1:6	0:2	0:2	0:2
Total Errors	14	5	7	5

Fig. 1.3 The error counts of various classification methods on the blinded ALL test samples are given in this figure. PCL is shown to make considerably less misclassifications. The OTHERSi class contains all those subtypes of ALL below the ith level of the tree depicted in Figure 1.2.

Testing Data	Error rate of different models		
	C4.5	SVM	NB
T-ALL vs OTHERS1	0:1	0:0	13:0
E2A-PBX1 vs OTHERS2	0:0	0:0	9:0
TEL-AML1 vs OTHERS3	2:4	0:9	20:0
BCR-ABL vs OTHERS4	1:3	2:0	6:0
MLL vs OTHERS5	0:1	0:0	6:0
Hyperdiploid>50 vs OTHERS	4:10	12:0	7:2
Total Errors	26	23	63

Fig. 1.4 The error counts of various classification methods on the blinded ALL test samples without filtering by selecting relevant genes are given in this figure. The OTHERSi class contains all those subtypes of ALL below the ith level of the tree depicted in Figure 1.2.

In some other subtypes, no single gene can yield such reliable rules, and we must thus look at rules involving co-ordinated gene expression. In the prediction of the subtype TEL-AML1, two of the top genes are 38652_at and 36937_s_at. The expression range of 38652_at is partitioned at the point 8997.35 and the expression range of 36937_at is partitioned at the point 13617.05. These partitioning induces 4 candidate patterns and one of them (38652_at $\geq$ 8997.35 and 36937_s_at $<$ 13617.05) is a

top emerging pattern that appears in 92.31% of TEL-AML1 training samples but never in OTHERS3. This suggests that 38652_at and 36937_s_at are co-ordinated in TEL-AML1 and induces a rule that has an estimated validity of 92.31%:

> if 38652_at $\geq$ 8997.35 and 36937_s_at $<$ 13617.05 in a sample, then the sample is TEL-AML1.

These rules and other additional ones on the childhood leukaemia dataset are discussed in more detail in Li et al [67].

1.5 DERIVATION OF TREATMENT PLAN

In the previous sections, we see that the entropy measure can be used to identify genes that are relevant to the diagnosis of disease states and subtypes. We also saw that the top emerging patterns are suggestive of coordinated gene groups in particular disease states and subtypes: (i) For each gene in such a coordinated gene group there is a pre-determined interval of gene expression level, as specified by the corresponding emerging pattern. (ii) Particular disease states and subtypes can often be characterized by one or more such emerging patterns, in the sense that a large portion of the cases in the given disease state (or subtype) match the corresponding emerging patterns and the cases in other disease states (or subtypes) never match the same emerging patterns. Based on these patterns, we conjecture the possibility of a personalized "treatment plan" which converts tumor cells into normal cells by modulating the expression levels of a few genes.

We use the colon tumour dataset of Alon et al. [4] to demonstrate our idea in this section. This dataset consists of 22 normal tissues and 40 colon tumor tissues. We begin with finding out which intervals of the expression levels of a group of genes occur only in cancer tissues but not in the normal tissues and vice versa. Then we attempt an explanation of the results and suggest a plan for treating the disease.

We use the entropy measure [34] described earlier to induce a partition of the expression range of each gene into suitable intervals. As discussed, this method partitions a range of real values into a number of disjoint intervals such that the entropy of the partition is minimal. For the colon cancer dataset, of its 2000 genes,

only 135 genes can be partitioned into 2 intervals of low entropy [72, 70]. The remaining 1865 genes are ignored by the method. Thus most of the genes are viewed as irrelevant by the method. For the purpose of this chapter we further concentrate on the 35 genes with the lowest entropy measure amongst the 135 genes. These 35 genes are shown in Figure 1.5. This gives us an easy platform where a small number of good diagnostic indicators are concentrated. For simplicity of reference, the index numbers in the first column of Figure 1.5 are used to refer to the two expression intervals of the corresponding genes. For example, the index 1 means M26338 $<$ 59.83 and the index 2 means M26383 $\geq$ 59.83.

Next, we use an efficient border-based algorithm [26, 69] to discover emerging patterns based on the selected 35 genes and the partitioning of their expression intervals induced by the entropy measure. The emerging patterns are thus combinations of intervals of gene expression levels of these relevant genes. A total of 10548 emerging patterns are found, 9540 emerging patterns for the normal class and 1008 emerging patterns for the tumour class. The top several tens of the normal class emerging patterns contain about 8 genes each and can reach a frequency of 77.27%, while many tumour class emerging patterns can reach a frequency of around 65%. These top emerging patterns are presented in Figure 1.6 and Figure 1.7. Note that the numbers in the emerging patterns in these figures, such as $\{2, 10\}$ in Figure 1.7, refer to the index numbers in Figure 1.5. Hence, $\{2, 10\}$ denotes the pattern $\{$M26383 $\geq$ 59.83, H08393 $\geq$ 84.87$\}$. The emerging patterns that are discovered are most general ones, and they occur in one class of data but do not occur in the other class. The discovered emerging patterns always contain only a small number of the relevant genes. This result reveals interesting conditions on the expression of these genes that differentiate between two classes of data.

Each emerging pattern with high frequency is considered as a common property of a class of cells. Based on this idea, we propose a strategy for treating colon tumors by adjusting the expression level of some improperly expressed genes. That is, we increase or decrease the expression levels of some particular genes in a cancer cell, so that it has the common properties of normal cells and no properties of cancer cells. As a result, instead of killing the cancer cell, it is "converted" into a normal one. We

Our list	accession number	cutting points	Name
1,2	M26383	59.83	monocyte-derived neutrophil-activating protein mRNA
3,4	M63391	1696.22	Human desmin gene
5,6	R87126	379.38	myosin heavy chain, nonmuscle (Gallus gallus)
7,8	M76378	842.30	Human cysteine-rich protein (CRP) gene, exons 5 and 6
9,10	H08393	84.87	COLLAGEN ALPHA 2(XI) CHAIN (Homo sapiens)
11,12	X12671	229.99	heterogeneous nuclear ribonucleoprotein core protein A1
13,14	R36977	274.96	P03001 TRANSCRIPTION FACTOR IIIA
15,16	J02854	735.80	Myosin regulatory light chain 2, smooth muscle isoform
17,18	M22382	447.04	Mitochondrial matrix protein P1 precursor (Human)
19,20	J05032	88.90	Human aspartyl-tRNA synthetase alpha-2 subunit mRNA
21,22	M76378	1048.37	Human cysteine-rich protein (CRP) gene, exons 5 and 6
23,24	M76378	1136.74	Human cysteine-rich protein (CRP) gene, exons 5 and 6
25,26	M16937	390.44	Human homeo box c1 protein mRNA
27,28	H40095	400.03	Macrophage migration inhibitory factor (Human)
29,30	U30825	288.99	Human splicing factor SRp30c mRNA
31,32	H43887	334.01	Complement Factor D Precursor
33,34	H51015	84.19	Proto-oncogene DBL Precursor
35,36	X57206	417.30	1D-myo-inositol-trisphosphate 3-kinase B isoenzyme
37,38	R10066	494.17	PROHIBITIN (Homo sapiens)
39,40	T96873	75.42	Hypothetical protein in TRPE 3'region (S. aurantia)
41,42	T57619	2597.85	40S ribosomal protein S6 (Nicotiana tabacum)
43,44	R84411	735.57	Small nuclear ribonucleoprotein assoc. protein B and B'
45,46	U21090	232.74	Human DNA polymerase delta small subunit mRNA
47,48	U32519	87.58	Human GAP SH3 binding protein mRNA
49,50	T71025	1695.98	Human (HUMAN)
51,52	T92451	845.7	Tropomyosin, fibroblast and epithelial muscle-type
53,54	U09564	120.38	Human serine kinase mRNA
55,56	H40560	913.77	THIOREDOXIN (HUMAN)
57,58	T47377	629.44	S-100P PROTEIN (HUMAN)
59,60	X53586	121.91	Human mRNA for integrin alpha 6
61,62	U25138	186.19	Human MaxiK potassium channel beta subunit mRNA
63,64	T60155	1798.65	ACTIN, AORTIC SMOOTH MUSCLE (HUMAN)
65,66	H55758	1453.15	ALPHA ENOLASE (HUMAN)
67,68	Z50753	196.12	H.sapiens mRNA for GCAP-II/uroguanylin precursor
69,70	U09587	486.17	Human glycyl-tRNA synthetase mRNA

Fig. 1.5 The 35 top-ranked genes by the entropy measure. The index numbers in the first column are used to refer to the two expression intervals of the corresponding genes. For example, the index 1 means M26338 $<$ 59.83 and the index 2 means M26383 $\geq$ 59.83.

show later that almost all "adjusted" cells are predicted as normal cells by a number of good classifiers that were trained to distinguish normal from colon tumor cells.

Emerging patterns	Count & Freq. (%) in normal tissues	Count & Freq. (%) in cancer tissues
$\{25, 33, 37, 41, 43, 57, 59, 69\}$	17(77.27%)	0
$\{25, 33, 37, 41, 43, 47, 57, 69\}$	17(77.27%)	0
$\{29, 33, 35, 37, 41, 43, 57, 69\}$	17(77.27%)	0
$\{29, 33, 37, 41, 43, 47, 57, 69\}$	17(77.27%)	0
$\{29, 33, 37, 41, 43, 57, 59, 69\}$	17(77.27%)	0
$\{25, 33, 35, 37, 41, 43, 57, 69\}$	17(77.27%)	0
$\{33, 35, 37, 41, 43, 57, 65, 69\}$	17(77.27%)	0
$\{33, 37, 41, 43, 47, 57, 65, 69\}$	17(77.27%)	0
$\{33, 37, 41, 43, 57, 59, 65, 69\}$	17(77.27%)	0
$\{33, 35, 37, 41, 43, 45, 57, 69\}$	17(77.27%)	0
$\{33, 37, 41, 43, 45, 47, 57, 69\}$	17(77.27%)	0
$\{33, 37, 41, 43, 45, 57, 59, 69\}$	17(77.27%)	0
$\{13, 33, 35, 37, 43, 57, 69\}$	17(77.27%)	0
$\{13, 33, 37, 43, 47, 57, 69\}$	17(77.27%)	0
$\{13, 33, 37, 43, 57, 59, 69\}$	17(77.27%)	0
$\{13, 32, 37, 57, 69\}$	17(77.27%)	0
$\{33, 35, 37, 57, 68\}$	17(77.27%)	0
$\{33, 37, 47, 57, 68\}$	17(77.27%)	0
$\{33, 37, 57, 59, 68\}$	17(77.27%)	0
$\{32, 37, 41, 57, 69\}$	17(77.27%)	0

Fig. 1.6 The top 20 emerging patterns, in descending frequency order, in the 22 normal tissues. The numbers in the emerging patterns above refer to the index numbers in Figure 1.5.

As shown in Figure 1.6, the frequency of emerging patterns can reach a very high level such as 77.27%. The conditions implied by a highly frequent emerging pattern form a common property of one class of cells. Using the emerging pattern $\{25, 33, 37, 41, 43, 57, 59, 69\}$ from Figure 1.6, we see that each of the 77.27% of the normal cells simultaneously expresses the eight genes— M16937, H51015, R10066, T57619, R84411, T47377, X53586, and U09587 referenced in this emerging pattern—in such a way that each of the eight expression levels is contained in the corresponding interval—the 25th, 33th, 37th, 41st, 43rd, 57th, 59th, and 69th—as indexed in Figure 1.5. Although a cancer cell may express some of the eight genes in a similar manner as normal cells do, according to the dataset, a cancer cell can never express all of the eight genes in the same way as normal cells do. So, if the expression levels of those improperly expressed genes can be adjusted, then the cancer cell can be made to have one more common property that normal cells exhibit.

Emerging patterns	Count & Freq. (%) in normal tissues	Count & Freq. (%) in cancer tissues
$\{2, 10\}$	0	28 (70.00%)
$\{10, 61\}$	0	27 (67.50%)
$\{10, 20\}$	0	27 (67.50%)
$\{3, 10\}$	0	27 (67.50%)
$\{10, 21\}$	0	27 (67.50%)
$\{10, 23\}$	0	27 (67.50%)
$\{7, 40, 56\}$	0	26 (65.00%)
$\{2, 56\}$	0	26 (65.00%)
$\{12, 56\}$	0	26 (65.00%)
$\{10, 63\}$	0	26 (65.00%)
$\{3, 58\}$	0	26 (65.00%)
$\{7, 58\}$	0	26 (65.00%)
$\{15, 58\}$	0	26 (65.00%)
$\{23, 58\}$	0	26 (65.00%)
$\{58, 61\}$	0	26 (65.00%)
$\{2, 58\}$	0	26 (65.00%)
$\{20, 56\}$	0	26 (65.00%)
$\{21, 58\}$	0	26 (65.00%)
$\{15, 40, 56\}$	0	25 (62.50%)
$\{21, 40, 56\}$	0	25 (62.50%)

Fig. 1.7 The top 20 emerging patterns, in descending frequency order, in the 40 cancer tissues. The numbers in the emerging patterns refer to the index numbers in Figure 1.5.

Conversely, a cancer cell may exhibit an emerging pattern that is a common property of a large percentage of cancer cells and is not exhibited in any of the normal cells. Adjustments should also be made to some genes involved in this pattern so that the cancer cell can be made to have one less common property that cancer cells exhibit. A cancer cell can then be iteratively converted into a normal one as described above.

As there usually exist some genes of a cancer cell which express in a similar way as their counterparts in normal cells, less than 35 genes' expression levels are required to be changed. The most important issue is to determine which genes need an adjustment. Our emerging patterns can be used to address this issue as follows. Given a cancer cell, we first determine which top emerging pattern of normal cells has the closest Hamming distance to it in the sense that the least number of genes need to be adjusted to make this emerging pattern appear in the adjusted cancer cell. Then we proceed to adjust these genes. This process is repeated several times until the

adjusted cancer cell exhibits as many common properties of normal cells as a normal cell does. The next step is to look at which top emerging pattern of cancer cells that is still present in the adjusted cancer cell has the closest Hamming distance to a pattern in a normal cell. Then we also proceed to adjust some genes involved in this emerging pattern so that this emerging pattern would vanish from the adjusted cancer cell. This process is repeated until all top emerging patterns of cancer cells disappear from our adjusted cancer cell. It is possible to choose genes to adjust following the spirit above, but in a different way so that the number of gene adjustments needed is minimized. We leave the diligent readers to devise a more optimal strategy.

We use a cancer cell (T1) of the colon tumor dataset as an example to show how a tumor cell is converted into a normal one. Recall the emerging pattern $\{25, 33, 37, 41, 43, 57, 59, 69\}$ is a common property of normal cells. The eight genes involved in this emerging pattern are M16937, H51015, R10066, T57619, R84411, T47377, X53586, and U09587. Let us list the expression profile of these eight genes in T1:

genes	expression levels in T1
M16937	369.92
H51015	137.39
R10066	354.97
T57619	1926.39
R84411	798.28
T47377	662.06
X53586	136.09
U09587	672.20

However, 77.27%—17 out of 22 cases—of the normal cells have the following expression intervals for these 8 genes:

genes	expression interval
M16937	$<$390.44
H51015	$<$84.19
R10066	$<$494.17
T57619	$<$2597.85
R84411	$<$735.57
T47377	$<$629.44
X53586	$<$121.91
U09587	$<$486.17

Comparing T1's gene expression levels with the intervals of normal cells, we see that 5 of the 8 genes—H51015, R84411, T47377, X53586, and U09587—of the cancer cell T1 behave in a different way from those the 22 normal cells commonly express. However, the remaining 3 genes of T1 are in the same expression range as most of the normal cells. So, if the 5 genes of T1 can be down regulated to scale below those cutting points, then this adjusted cancer cell will have a common property of normal cells. This is because $\{25, 33, 37, 41, 43, 57, 59, 69\}$ is an emerging pattern which does not occur in the cancer cells. This idea is at the core of our suggestion for this treatment plan.

Interestingly, the expression change of the 5 genes in T1 leads to a chain of other changes. These include the change that 9 extra top-ten EPs of normal cells are contained in the adjusted T1. So all top-ten EPs of normal cells are contained in T1 if the 5 genes' expression level are adjusted. As the average number of top-ten EPs contained in normal cells is 7, the changed T1 cell will now be considered as a cell that has the most important features of normal cells. Note that we have adjusted only 5 genes' expression level so far.

We also need to eliminate those common properties of cancer cells that are contained in T1. By adjusting the expression level of 2 other genes, M26383 and H08393, the top-ten EPs of cancer cells all disappear from T1. According to our colon tumor dataset, the average number of top-ten EPs of cancer cells contained in a cancer cell is 6. Therefore, T1 is converted into a normal cell as it now holds the common properties of normal cells and does not hold the common properties of cancer cells.

By this method, all the other 39 cancer cells can be converted into normal ones after adjusting the expression levels of 10 genes or so, possibly different genes from person to person. We conjecture that this personalized treatment plan is effective if the expression of some particular genes can be modulated by suitable means.

We next discuss a validation of this idea. The "adjustments" we made to the 40 colon tumour cells were based on the emerging patterns in the manner described above. If these adjustments had indeed converted the colon tumour cells into normal cells, then any good classifier that could distinguish normal vs colon tumour cells on the basis of gene expression profiles would classify our adjusted cells as normal

cells. So, we established a SVM model using the original entire 22 normal plus 40 cancer cells as training data. The code for constructing this SVM model is available at `http://www.cs.waikato.ac.nz/ml/weka`. The prediction result is that all of the adjusted cells were predicted as normal cells. Although our "therapy" was not applied to the real treatment of a patient, the prediction result by the SVM model partially demonstrates the potential biological significance of our proposal.

1.6 UNDERSTANDING OF MOLECULAR CIRCUIT

A large number of genes can be differentially expressed in a microarray experiment. Such genes can serve as markers of the different classes—such as tumour vs. normal—of samples in the experiment. Some of these genes can even be the primary cause of a sample being tumour. However, on the basis of gene expression alone, it is not possible to decide which gene is part of the primary cause and which gene is merely a down-stream effect. In order to separate the former from the latter, it is necessary to consider the underlying molecular network or biological pathway [43]. In this section, we first briefly discuss four approaches to this issue that do not rely on microarray gene expression experiments, and then we discuss a fifth approach to this issue that does rely on microarray gene expression experiments.

There are four approaches to constructing a database of molecular network that do not rely on microarray gene expression experiments. The first approach is that of hand curation from literatures that discuss pathways of protein inhibition, activation, and other interactions. KEGG [60], MPW [33], and CSNDB [119] are examples of such hand curated databases of molecular network. However, this approach is laborious and is unlikely to scale. The second approach is that of conducting high throughput experiments such as yeast two-hybrid assay [55] and mass spectrometry [84]. This is also very time consuming, costly, and is also not free of errors [24]. Furthermore, the number of interactions determined by recent large-scale experiments is still small relative to the large number of interactions reported from the thousands of individual small experiments over the years [129]. This leads to the development of the third approach which is based on natural language processing. In this approach, a

large collection of abstracts and texts from biological research papers are collected. Algorithms are then applied to recognize names of proteins and other molecules in these texts. These algorithms are typically based on special characteristics of protein names such as the occurrence of uppercase letters, numerals, and special endings [39]. Then sentences containing multiple occurrences of protein names and some action words—such as "inhibit" and "activate"—are extracted. Such sentences are then analysed by natural language parsers to determine the exact relationships between the proteins mentioned [97, 93]. Lastly, these relationships are assembled into a network, so that we know exactly which protein is acting directly or indirectly on which other proteins and in what way [128]. However, this third approach can only identify previously reported protein interactions. In order to obtain additional interactions, it is necessary to computationally infer them in a fourth approach. One of the important methods to infer interactions is the Rosetta stone method [85, 32]. Under this method, two proteins are assumed to interact if they are fused into a third protein in another organism. Fusion is typically detected by a sequence comparison and alignment program. There are three caveats. (1) This technique cannot infer the direction of the interaction. (2) It cannot deal with proteins having promiscuous domains such as SH3 and ATP-binding cassettes. (3) It can produce false positives.

Networks constructed using the second and fourth approaches above lack information on the direction of interactions, and thus they are less useful for inferring genes that are responsible for the primary effect. Nevertheless, as networks constructed using the other two approaches do not have this deficiency, they are extremely useful for another purpose. Specifically, these networks can be used to perform functional annotation [41], because proteins near each other in the network can be assumed to be in the same pathway and thus have related functions. Networks constructed using the second and fourth approaches may also contain a large number of false positives and false negatives, as demonstrated [24] by the small overlap among protein interactions determined by several experiments [55, 123, 38, 90]. In order to improve their fidelity, additional filtering by computational methods is necessary. Deane et al [24] described two such assessment techniques. The first technique is to compare the RNA expression profiles of a pair of proteins that have been identified

as interacted by the second or the fourth approach to the RNA expression profiles of pairs of known interaction proteins and pairs of known non-interacting proteins. The second technique is to check whether a pair of proteins, that have been identified as interacting by the second or fourth approach, have paralogs that are known to interact. In addition, subcellular co-localization information can also be used to assess a putative interaction [111].

A fifth approach to constructing a database of molecular network that does rely on microarray gene expression experiments is possible. Let us recall that in analysing microarray gene expression output in the last two sections, we first identify a number of candidate genes by feature selection. Do we know which ones of these are causal genes and which are mere surrogates? Genes are "connected" in a "circuit" or network. The expression of a gene in a network depends on the expression of some other genes in the network. Can we reconstruct the gene network from gene expression data? For each gene in the network, can we determine which genes affect it? and how they affect it—positively, negatively, or in more complicated ways? There are several techniques to reconstructing and modeling molecular networks from gene expression experiments. Some techniques that have been tried are Bayesian networks [36], Boolean networks [2, 1], differential equations [21], association rule discovery [92], as well as classification-based methods [116]. We devote the rest of section to describe the classification-based method of Soinov et al [116].

Let a collection of n microarray gene expression output be given. For convenience, this collection can be organized into a gene expression matrix X. Each row of the matrix is a gene, each column is a sample, and each element x_{ij} is the expression of gene i in sample j. Then the basic idea of the method of Soinov et al [116] is as follows. First determine the average value a_i of each gene i as $(\sum_j x_{ij})/n$. Next, denote s_{ij} as the state of gene i in sample j, where $s_{ij} = up$ if $x_{ij} \geq a_i$, and $s_{ij} = down$ if $x_{ij} < a_i$. Then, according to Soinov et al [116], to see whether the state of a gene g is determined by the state of other genes G, we check whether $\langle s_{ij} \,|\, i \in G \rangle$ can predict s_{gj}. If it can predict s_{gj} with high accuracy, then we conclude that the state of the gene g is determined by the states of other genes G. Furthermore, any classifier can be used to see if such predictions can be made reliably, such as

C4.5, PCL, and SVM. Then, to see how the state of a gene g is determined by the state of other genes, we apply C4.5, or PCL, or other rule-based classifiers to predict s_{gj} from $\langle s_{ij} |\ i \in G\rangle$ and extract the decision tree or rules used.

This interesting method has a few advantages: It can identify genes affecting a target genes in an explicit manner, it does not need a discretization threshold, each data sample is treated as an example, and explicit rules can be extracted from a rule-based classifier like C4.5 or PCL. For example, we generate from the gene expression matrix a set of n vectors $\langle s_{ij} |\ i \neq g\rangle \Rightarrow s_{gj}$. Then C4.5 (or PCL) can be applied to see if $\langle s_{ij} |\ i \neq g\rangle$ predicts s_{gj}. The decision tree (or emerging patterns, respectively) induced would involve a small number of s_{ij}. Then we can conclude that those genes corresponding to these small number of s_{ij} affect gene g.

One other nice advantage of the Soinov method [116] is that it is easily generalizable to time series. Suppose the matrices X^t and X^{t+1} correspond to microarray gene expression measurements taken at time t and $t+1$. Suppose s_{ij}^t and s_{ij}^{t+1} correspond to the expression of gene i in sample j at time t and $t+1$. Then to find out whether the state of a gene g is affected by other genes G in a time-lagged manner, we check whether $\langle s_{ij}^t |\ i \in G\rangle$ can predict s_{gj}^{t+1}. The rest of the procedure is as before.

Of course, there is a major caveat that this method as described assumes that a gene g can be in only two states, viz. $s_{gj} = up$ or $s_{gj} = down$. As cautioned by Soinov et al [116], it is possible for a gene to have more than two states and thus this assumption may not infer the complete network of gene interactions. Another caution is that if the states of two genes g and h are strongly co-related, the rules $s_{hj} \Rightarrow s_{gj}$ and $s_{gj} \Rightarrow s_{hj}$ saying that h depends on g and g depends on h are likely to be both inferred, even though only one of them may be true and the other false. Hence, further confirmation by gene knock-out or other experiments is advisable.

We do not have independent results on this approach to reconstructing molecular networks. However, we refer the curious reader to Soinov et al [116] for a discussion on experiments they have performed to verify the relevance of this method.

1.7 CLOSING REMARKS

Microarrays are a technology for simultaneously profiling the expression levels of tens of thousands of genes in a patient samples. It is increasingly clear that better diagnosis methods and better understanding of disease mechanisms can be derived from a careful analysis of microarray measurements of gene expression profiles. This chapter discussed several types of analysis of such gene expression profiles, including (a) diagnosis of disease state and subtype, (b) derivation of disease treatment plan, and (c) understanding of gene interaction networks. In the course of this discussion, we have surveyed techniques for gene selection from microarray gene expression profiles, including signal-to-noise measure, t-test, entropy measure, and CFS. We have also introduced the emerging patterns-based classification method called PCL.

Let us end this chapter with a brief mention of a number of other data mining tools and their applications in the biomedical arena in the context of classification and prediction. The most popular classification technique is the idea of decision tree induction. We already briefly discussed the C4.5 method in Section 1.4. Other algorithms for decision tree induction include CART [14], ID3 [99], SLIQ [87], FACT [78], QUEST [77], PUBLIC [103], CHAID [59], ID5 [124], SPRINT [114], and BOAT [42]. This group of algorithms are most successful for analysis of clinical data and for diagnosis from clinical data. Some examples are diagnosis of central nervous system involvement in hematooncologic patients [80], prediction of post-traumatic acute lung injury [101], identification of acute cardiac ischemia [113], prediction of neurobehavioral outcome in head-injury survivors [120], and diagnosis of myoinvasion [79].

Another important group of techniques [9, 28, 89, 58, 50, 57, 108, 66] are based on the Bayes theorem. The theorem states that $P(h|d) = P(d|h) * P(h)/P(d)$, where $P(h)$ is the prior probability that a hypothesis h holds, $P(d|h)$ is the probability of observing data d given some world that h holds, and $P(h|d)$ is the posterior probability that h holds given the observed data d. Let H be all the possible classes. Then given a test instance with feature vector $(f_1, \ldots, f_n)$, the most probable classification is $\text{argmax}_{h_j \in H} P(h_j|f_1, \ldots, f_n)$. Using the Bayes

theorem, this is rewritten to argmax$_{h_j \in H} P(f_1, \ldots, f_n|h_j) * P(h_j)/P(f_1, \ldots, f_n)$ = argmax$_{h_j \in H} P(f_1, \ldots, f_n|h_j) * P(h_j)$. However, estimating $P(f_1, \ldots, f_n|h_j)$ accurately may not be feasible unless the training set is sufficiently large. So, the Naive Bayes method mentioned in Section 1.4 assumes that the effect of a feature value on a given class is independent of the values of other features. This assumption is called class conditional independence. It is made to simplify computation and it is in this sense that Naive Bayes is considered to be "naive." Under this class conditional independence assumption, argmax$_{h_j \in H} P(f_1, \ldots, f_n|h_j) * P(h_j)$ = argmax$_{h_j \in H} \prod_i P(f_i|h_j) * P(h_j)$. $P(h_j)$ and $P(f_i|h_j)$ can often be estimated reliably from typical training sets. Some example applications of Bayesian classifiers in the biomedical context are mapping and controlling of a genetic trait [44], screening for macromolecular crystallization [51], classification of cNMP-binding proteins [86], prediction of carboplatin exposure [54], prediction of prostate cancer recurrence [25], prognosis of femoral neck fracture recovery [64], and prediction of protein secondary structure [61, 118, 6].

Related to the Bayesian classifiers are the Hidden Markov Models or HMMs [9, 62, 29, 30]. A HMM is a stochastic generative model for sequences defined by a finite set S of states, a finite alphabet A of symbols, a transition probability matrix T, and an emission probability matrix E. The system moves from state to state according to T while emitting symbols according to E. In an n-th order HMM, the matrices T and E depend on all n previous states. HMMs have been applied to a variety of problems in sequence analysis, including protein family classification and prediction [11, 8, 63], tRNA detection in genomic sequences [81], methylation guide snoRNA screening [82], gene finding and gene structure prediction in DNA sequences [13, 12, 7, 62, 109], protein secondary structure modeling [35], and promoter recognition [130, 94].

Artificial neural networks [107, 9, 19] are another important approach to classification that have high tolerance to noisy data. Feed-forward multi-layer neural networks have received the greatest attention, in part because of their universal approximation capability [20, 53] as well as simple and effective training algorithms [107]. Successful applications of artificial neural networks in the biomedical context include protein

secondary structure prediction [105, 106, 98], signal peptide prediction [22, 91, 31], gene finding and gene structure prediction [122, 115], T-cell epitope prediction [52], RNA secondary structure prediction [117], toxicity prediction [16], disease diagnosis and outcome prediction [126, 112, 121], as well as protein translation initiation site recognition [95, 49].

Last but not least, support vector machines are another approach to the classification problem that has clear connections to statistical learning theory [17, 125]. We have also briefly seen SVM in Section 1.4. An SVM selects a small number of critical boundary samples from each class and builds a linear discriminant function that separates them as widely as possible. In the case that no linear separation is possible, the technique of "kernel" is used to automatically inject the training samples into a higher-dimensional space, and to learn a separator in that space. An SVM is largely characterized by the choice of its kernel function. Thus SVMs connect the problem they are designed for to a large body of existing research on kernel-based methods [104, 125, 17]. Some recent applications of SVM in the biomedical context include protein homology detection [56], microarray gene expression data classification [15], breast cancer diagnosis [83, 37], as well as protein translation initiation site recognition [133, 134].

REFERENCES

1. T. Akutsu, S. Miyano, and S. Kuhara. Algorithms for inferring qualitative models of biological networks. In *Proc. of Pacific Symposium on Biocomputing 2000*, pages 293–304, 2000.

2. T. Akutsu and S. Miyano. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Proc. of Pacific Symposium on Biocomputing 1999*, pages 17–28, 1999.

3. A. A. Alizadeh, M. B. Eisen, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

4. U. Alon, N. Barkai, D. A. Notterman, K. Gish, S.Y.D. Mack, and J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96:6745–6750, 1999.

5. C. Ambroise and G. J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl. Acad. Sci. USA*, 14:6562–6566, 2002.

6. G.E. Arnold, A.K. Dunker, S. L. Johns, and R.J. Douthart. Use of conditional probabilities for determining relationships between amino acid sequence and protein secondary structure. *Proteins*, 12(4):382–399, April 1992.

7. P. Baldi, S. Brunak, Y. Chauvin, and A. Krogh. Hidden Markov models for human genes: Periodic patterns in exon sequences. In *Theoretical and Computational Methods in Genome Research*, pages 15–32, 1997.

8. P. Baldi and Y. Chauvin. Hidden Markov models of G-protein-coupled receptor family. *Journal of Computational Biology*, 1:311–335, 1994.

9. P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, MA, 1999.

10. P. Baldi and A. D. Long. A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics*, 17(6):509–519, 2001.

11. A. Bateman, E. Birney, R. Durbin, S. R. Eddy, R. D. Finn, and E. L. L. Sonnhammer. Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins. *Nucleic Acids Research*, 27(1):260–262, 1999.

12. M. Borodovsky and J. D. McIninch. GENEMARK: Parallel gene recognition for both DNA strands. *Computers and Chemistry*, 17(2):123–133, 1993.

13. M. Borodovsky, J. D. McIninch, E.V. Koonin, K.E. Rudd, C. Medigue, and A. Danchin. Detection of new genes in a bacterial genome using Markov models for three gene classes. *Nucleic Acids Research*, 23:3554–3562, 1995.

14. L. Breiman, L. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1984.

15. M.P. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares Jr, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*, 97(1):262–267, 2000.

16. F. R. Burden and D. A. Winkler. A quantitative structure-activity relationships model for the acute toxicity of substituted benzenes to *tetrahymena pyriformis* using Bayesian-regularised neural networks. *Chemical Research in Toxicology*, 13:436–440, 2000.

17. C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

18. M. Caria. *Measurement Analysis: An Introduction to the Statistical Analysis of Laboratory Data in Physics, Chemistry, and the Life Sciences*. Imperial College Press, London, 2000.

19. Y. Chauvin and D. Rumelhart. *Backpropagation: Theory, Architectures, and Applications*. Lawrence Erlbaum, Hillsdale, NJ, 1995.

20. Chen, T., & Chen, H. Universal approximation to non-linear operators by neural networks with arbitrary activation functions and its application to dynamically systems. *IEEE Transactions on Neural Networks*, 6:911–917, 1995.

21. T. Chen, H.L. He, and G.M. Church. Modeling gene expression with differential equations. In *Proc. of Pacific Symposium on Biocomputing 1999*, pages 29–40, 1999.

22. M.G. Claros, S. Brunak, and G. von Heijne. Prediction of n-terminal protein sorting signals. *Current Opinion in Structural Biology*, 7:394–398, 1997.

23. T.M. Cover and P.E. Hart. Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

24. C.M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg. Protein interactions: Two methods for assessment of the reliability of high-throughput observations. *Molecular and Cellular Proteomics*, 1:349–356, 2002.

25. J. Demsar, B. Zupan, M.W. Kattan, J.R. Beck, and I. Bratko. Naive Bayesian-based nomogram for prediction of prostate cancer recurrence. *Studies in Health Technology Informatics*, 68:436–441, 1999.

26. G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. of 5th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 15–18, San Diego, August 1999.

27. G. Dong, J. Li, and X. Zhang. Discovering jumping emerging patterns and experiments on real datasets. In *Proc. of 9th International Database Conference on Heterogeneous and Internet Databases*, pages 15–17, Hong Kong, July 1999.

28. R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

29. R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, editors. *Biological Sequence Analysis: Probabilistics Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

30. S.R. Eddy. Hidden Markov models. *Current Opinion in Structural Biology*, 6:361–365, 1996.

31. O. Emanuelsson, H. Nielsen, and G. von Heijne. ChloroP, a neural network-based method for predicting chloroplast transit peptides and their cleavage sites. *Protein Science*, 8(5):978–984, May 1999.

32. A. J. Enright, I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis. Protein interaction maps for complete genomes based on gene fusion events. *Nature*, 402:86–90, November 1999.

33. E. Selkov Jr., Y. Grechkin, N. Mikhailova, and E. Selkov. MPW: The metabolic pathways database. *Nucleic Acids Research*, 26(1):43–45, 1998.

34. U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of 13th International Joint Conference on Artificial Intelligence*, pages 1022–1029, 1993.

35. V. Di Francesco, J. Granier, and P.J. Munson. Protein topology recognition from secondary structure sequences—applications of the hidden Markov models to the alpha class proteins. *Journal of Molecular Biology*, 267:446–463, 1997.

36. N. Friedman, m. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyse expression data. *Journal of Computational Biology*, 7:601–620, 2000.

37. T.-T. Friess, N. Cristianini, and C. Campbell. The kernel adatron algorithm: A fast and simple learning procedure for support vector machines. In *Proc. of 15th International Conference on Machine Learning*, 1998.

38. M. Fromont-Racine, A.E. Mayes, A. Brunet-Simon, J.C. Rain, A. Colley, I. Dix, N. Joly, J.D. Beggs, and P. Legrain. Genome-wide protein interaction screens reveal functional networks involving Sm-like proteins. *Yeast*, 17:95–110, 2000.

39. K. Fukuda, A. Tamura, T. Tsunoda, and T. Takagi. Toward information extraction: Identifying protein names from biological papers. In *Proc. of Pacific Symposium on Biocomputing'98*, pages 707–718, Maui, Hawaii, January 1998.

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

41. M. Y. Galperin and S. E. Brenner. Using metabolic pathway databases for functional annotation. *Trends in Genetics*, 14(8):332–333, August 1998.

42. J. Gehrke, V. Ganti, R. Ramakrishnan, and W. Y. Loh. BOAT—optimistic decision tree construction. In *Proc. of ACM-SIGMOD International Conference on Management of Data*, pages 169–180, Philadelphia, PA, June 1999.

43. D. Gerhold, T. Rushmore, and C. T. Caskey. DNA chips: promising toys have become powerful tools. *Trends in Biochemical Sciences*, 24(5):168–173, May 1999.

44. S. Ghosh and P.P. Majumder. Mapping a quantitative trait locus via the EM algorithm and Bayesian classification. *Genetic Epidemiology*, 19(2):97–126, September 2000.

45. T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Misirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(15):531–537, 1999.

46. M.A. Hall. *Correlation-based feature selection machine learning*. PhD thesis, Department of Computer Science, University of Waikato, New Zealand, 1998.

47. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco, CA, 2000.

48. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.

49. A. G. Hatzigeorgiou. Translation initiation start prediction in human cDNAs with high accuracy. *Bioinformatics*, 18(2):343–350, February 2002.

50. D. Heckerman. Bayesian networks for knowledge discovery. In *Advances in Knowledge Discovery and Data Mining*, pages 273–305, Cambridge, MA, 1996. MIT Press.

51. D. Hennessy, B. Buchanan, D. Subramanian, P.A. Wilkosz, and J.M. Rosenberg. Statistical methods for the objective design of screening procedures for macromolecular crystallization. *Acta Crystallogr. D Biol. Crystallogr.*, 56(7):817–827, July 2000.

52. M. C. Honeyman, V. Brusic, N. Stone, and L. C. Harrison. Neural network-based prediction of candidate T-cell epitopes. *Nature Biotechnology*, 16(10):966–969, 1998.

53. Hornik, K. Some new results on neural network approximation. *Neural Networks*, 6:1069–1072, 1993.

54. A.D. Huitema, R.A. Mathot, M.M. Tibben, J.H. Schellens, S. Rodenhuis, and J.H. Beijnen. Validation of techniques for the prediction of carboplatin exposure: application of Bayesian methods. *Clinical Pharmacology Therapeutics*, 67(6):621–630, June 2000.

55. T. Ito et al. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA*, 98:4569–4574, 2001.

56. T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7((1-2)):95–114, 2000.

57. F. V. Jensen. *An Introduction to Bayesian Networks*. Springer-Verlag, New York, 1996.

58. G. H. John. *Enhancements to the Data Mining Process*. PhD thesis, Stanford University, 1997.

59. G. V. Kaas. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29:119–127, 1980.

60. M. Kanehisa, S. Goto, S. Kawashima, and A. Nakaya. The KEGG database at GenomeNet. *Nucleic Acids Research*, 30(1):42–46, 2002.

61. S. Kasif and A. L. Delcher. Modeling biological data and structure with probabilistic networks. In *Computational Methods in Molecular Biology*, pages 335–352, 1998.

62. A. Krogh. An introduction to hidden Markov models for biological sequences. In *Computational Methods in Molecular Biology*, pages 45–62, 1998.

63. A. Krogh, M. Brown, I. S. Mian, K. Sjolander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531, 1994.

64. M. Kukar, I. Kononenko, and T. Silvester. Machine learning in prognosis of the femoral neck fracture recovery. *Artificial Intelligence Medicine*, 8(5):431–451, October 1996.

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

66. S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201, 1995.

67. J. Li, H. Liu, J. R. Downing, A. E.-J. Yeoh, and L. Wong. Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients. *Bioinformatics*, 19:71–78, 2003.

68. J. Li, H. Liu, and L. Wong. A comparative study on feature selection and classification methods using a large set of gene expression profiles. In *Proc. of 13th International Conference on Genome Informatics*, pages 51–60, Tokyo, Japan, December 2002.

69. J. Li, K. Ramamohanarao, and G. Dong. The space of jumping emerging patterns and its incremental maintenance algorithms. In *Proc. of 17th International Conference on Machine Learning*, pages 551–558, San Francisco, June 2000. Morgan Kaufmann.

70. J. Li and L. Wong. Corrigendum: Identifying good diagnostic genes or genes groups from gene expression data by using the concept of emerging patterns. *Bioinformatics*, 18:1407–1408, 2002.

71. J. Li and L. Wong. Geography of differences between two classes of data. In *Proc. 6th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 325–337, Helsinki, Finland, August 2002.

72. J. Li and L. Wong. Identifying good diagnostic genes or genes groups from gene expression data by using the concept of emerging patterns. *Bioinformatics*, 18:725–734, 2002.

73. J. Li and L. Wong. Solving the fragmentation problem of decision trees by discoverying boundary emerging patterns. In *Proc. of IEEE International Conference on Data Mining*, pages 653 – 656, Maebashi, Japan, 2002.

74. J. Li and L. Wong. Using rules to analyse bio-medical data: A comparison between C4.5 and PCL. In *Proc. of the Fourth International Conference on Web-Age Information Management (WAIM)*, To appear, Chengdu, China, 2003. Springer.

75. H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proc. of IEEE 7th International Conference on Tools with Artificial Intelligence*, pages 338–391, 1995.

76. D. J. Lockhart, H. Dong, M.C. Byrne, M.T. Follettie, M.V. Gallo, M.S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, E.L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14:1675–1680, 1996.

77. W. Y. Loh and Y. S. Shih. Split selection methods for classification trees. *Statistica Sinica*, 7:815–840, 1997.

78. W. Y. Loh and N. Vanichsetakul. Tree-structured classification via generalized discriminant analysis. *Journal of American Statistical Association*, 83:715–728, 1988.

79. T.A. Longacre, M.H. Chung, D.N. Jensen, and M.R. Hendrickson. Proposed criteria for the diagnosis of well-differentiated endometrial carcinoma. A diagnostic test for myoinvasion. *American Journal of Surgical Pathology*, 19(4):371–406, April 1995.

80. I.S. Lossos, R. Breuer, O. Intrator, and A. Lossos. Cerebrospinal fluid lactate dehydrogenase isoenzyme analysis for the diagnosis of central nervous system involvement in hematooncologic patients. *Cancer*, 88(7):1599–1604, April 2000.

81. T. M. Lowe and S. R. Eddy. tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Research*, 25(5):955–964, 1997.

82. T. M. Lowe and S. R. Eddy. A computational screen for methylation guide snoRNAs in yeast. *Science*, 283:1168–1171, February 1999.

83. O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, 1995.

84. M. Mann, R.C. Hendrickson, and A. Pandey. Analysis of proteins and proteomes by mass spectrometry. *Annual Review of Biochemistry*, 70:437–473, 2001.

85. E. M. Marcotte, M. Pellegrini, H.-L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285:751–753, July 1999.

86. L. A. McCue, K. A. McDonough, and C. E. Lawrence. Functional classification of cnmp-binding proteins and nucleotide cyclases with implications for novel regulatory pathways in mycobacterium tuberculosis. *Genome Research*, 10(2):204–219, February 2000.

87. M. Mehta, R. Agrawal, and J. Rissanen. SLIQ: A fast scalable classifier for data mining. In *Proc. of International Conference on Extending Database Technology*, pages 18–32, Avignon, France, March 1996.

88. L. D. Miller, P. M. Long, L. Wong, S. Mukherjee, L. M. McShane, and E. T. Liu. Optimal gene expression analysis by microarrays. *Cancer Cell*, 2:353–361, November 2002.

89. T.M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

90. J.R. Newman, E. Wolf, and P.S. Kim. From the cover: A computationally directed screen identifying interacting coiled coils from Saccharomyces cerevisiae. *Proc. Natl. Acad. Sci. USA*, 97:13203–13208, 2000.

91. H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Science*, 3:3–14, 1994.

92. T. Oyama, K. Kitano, K. Satou, and T. Ito. Extraction of knowledge on protein-protein interaction by association rule discovery. *Bioinformatics*, 18(5):705–714, 2002.

93. J.C. Park, H.S. Kim, and J.J. Kim. Bidirectional incremental parsing for automatic pathway identification with combinatory categorial grammar. In *Proc. of Pacific Symposium on Biocomputing*, pages 396–407, 2001.

94. A.G. Pedersen, P. Baldi, S. Brunak, and Y. Chauvin. Characterization of prokaryotic and eukaryotic promoters using hidden Markov models. *Intelligent Systems for Molecular Biology*, 4:182–191, 1996.

95. A. G. Pedersen and H. Nielsen. Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis. *Intelligent Systems for Molecular Biology*, 5:226–233, 1997.

96. C. M. Perou, T. Sorlie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, C. A. Rees, J. R. Pollack, D. T. Ross, H. Johnsen, L. A. Akslen, et al. Molecular portraits of human breast tumours. *Nature*, 406:747–752, 2000.

97. J. Putejovsky and J.M. Castano. Robust relational parsing over biomedical literature: Extracting inhibit relations. In *Proc. of Pacific Symposium on Biocomputing*, pages 362–373, 2002.

98. N. Qian and T.J. Sejnowski. Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, 202:865–884, 1988.

99. J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

100. J. R. Quinlan. *C4.5: Program for Machine Learning*. Morgan Kaufmann, 1993.

101. T.H. Rainer, P.K. Lam, E.M. wong, and R.A. Cocks. Derivation of a prediction rule for post-traumatic acute lung injury. *Resuscitation*, 42(3):187–196, November 1999.

102. G. Ramsay. DNA chips: State-of-the art. *Nature Biotechnology*, 16:40–44, January 1998.

103. R. Rastogi and K. Shim. Public: A decision tree classifier that integrates building and pruning. In *Proc. of 24th International Conference on Very Large Data Bases*, pages 404–415, New York, August 1998.

104. S. Raudys. How good are support vector machines? *Neural Network*, 13(1):17–19, January 2000.

105. S.K. Riis and A. Krogh. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *Journal of Computational Biology*, 3:163–183, 1996.

106. B. Rost and C. Sander. Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins*, 19:55–72, 1994.

107. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

108. S. Russell, J. Binder, D. Koller, and K. Kanazawa. Local learning in probabilistic networks with hidden variables. In *Proc. of 14th Joint International Conference on Artificial Intelligence, volume 2*, pages 1146–1152, Montreal, Canada, August 1995.

109. S. L. Salzberg, A. L. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated Markov models. *Nucleic Acids Research*, 26(2):544–548, 1998.

110. B. Scholkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

111. B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature Biotechnology*, 18:1257–1261, 2000.

112. J.A. Scott, E. L. Palmer, and A.J. Fischman. How well can radiologists using neural network software diagnose pulmonary embolism? *AJR Am. J. Roentgenol.*, 175(2):399–405, August 2000.

113. H.P. Selker, J.L. Griffith, S. Patil, W.J. Long, and R.B. D'Agostino. A comparison of performance of mathematical predictive methods for medical diagnosis: identifying acute cardiac ischemia among emergency department patients. *J. Investig. Med.*, 43(5):468–476, October 1995.

114. J. Shafer, R. Agrawal, and M. Mehta. SPRINT: A scalable parallel classifier for data mining. In *Proc. of 22nd International Conference on Very Large Data Bases*, pages 544–555, Bombay, India, September 1996.

115. E. E. Snyder and G. D. Stormo. Identification of protein coding regions in genomic DNA. *Journal of Molecular Biology*, 248:1–18, 1995.

116. L. A. Soinov, M. A. Krestyaninova, and A. Brazma. Towards reconstruction of gene networks from expression data by supervised learning. *Genome Biology*, 4(1):R6, 2003.

117. E. W. Steeg. Neural networks, adaptive optimization, and RNA secondary structure prediction. In *Artificial Intelligence and Molecular Biology*, pages 121–160, 1993.

118. C.M. Stultz, R. Nambudripad, R.H. Lathrop, and J.V. White. Predicting protein structure with probabilistic models. In *Protein Structural Biology in Biomedical Research*, pages 447–506, 1997.

119. T. Takai-Igarashi and T. Kaminuma. A pathway finding system for the cell signaling networks database. *In Silico Biology*, 1:129–146, 1999.

120. N.R. Temkin, R. Holubkov, J.E. Machamer, H.R. Winn, and S.S. Dikmen. Classification and regression trees (CART) for prediction of function at 1 year following head trauma. *Journal of Neurosurgery*, 82(5):764–771, May 1995.

121. E.P. Turton, D.J. Scott, M. Delbridge, S. Snowden, and R.c. Kester. Ruptured abdominal aortic aneurysm: A novel method of outcome prediction using neural network technology. *European Journal of Vascular and Endovascular Surgery*, 19(2):184–189, February 2000.

122. E. C. Uberbacher and R. J. Mural. Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proc. Natl. Acad. Sci. USA*, 88:11261–11265, December 1991.

123. P. Uetz, L. Giot, G. Cagney, T.A. Mansfield, R.S. Judson et al. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403:623–627, 2000.

124. P. E. Utgoff. An incremental ID3. In *Proc. of 5th International Conference on Machine Learning*, pages 107–120, San Mateo, CA, 1988.

125. V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

126. J.L. Vriesema, H.G. van der Poel, F.M. Debruyne, J.A. Schalken, L.P. Kok, and M.E. Boon. Neural network-based digitized cell image diagnosis of bladder wash cytology. *Diagnostic Cytopathology*, 23(3):171–179, September 2000.

127. L. Wang, H. Zhao, G. Dong, and J. Li. On the Complexity of Computing Emerging Patterns. Manuscript. 2003.

128. L. Wong. PIES, a protein interaction extraction system. In *Proc. of Pacific Symposium on Biocomputing*, pages 520–531, January 2001.

129. I. Xenarios and D. Eisenberg. Protein interaction databases. *Current Opinion on Biotechnology*, 12:334–339, 2001.

130. T. Yada, M. Ishikawa, H. Tanaka, and K. Asai. Extraction of hidden Markov model representations of signal patterns in DNA sequences. In *Proc. of Pacific Symposium on Biocomputing*, pages 686–696, 1996.

131. E.-J. Yeoh, M. E. Ross, S. A. Shurtleff, W. K. William, D. Patel, R. Mahfouz, F. G. Behm, S. C. Raimondi, M. V. Reilling, A. Patel, C. Cheng, D. Campana,

D. Wilkins, X. Zhou, J. Li, H. Liu, C.-H. Pui, W. E. Evans, C. Naeve, L. Wong, and J. R. Downing. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1:133–143, March 2002.

132. X. Zhang, G. Dong, and K. Ramamohanarao. Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In *Proc. of 6th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 310–314, Boston, USA, August 2000.

133. A. Zien, G. Raatsch, S. Mika, B. Schoelkopf, C. Lemmem, A. Smola, T. Lengauer, and K.R. Mueller. Engineering support vector machine kernels that recognize translation initiation sites. In *Proc. of German Conference on Bioinformatics*, pages 37–43, Hanover, Germany, 1999.

134. A. Zien, G. Raatsch, S. Mika, B. Schoelkopf, T. Lengauer, and K.R. Mueller. Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, 16(9):799–807, 2000.