

## Gene expression

# Simple decision rules for classifying human cancers from gene expression profiles

Aik Choon Tan<sup>1,\*</sup>, Daniel Q. Naiman<sup>1,2</sup>, Lei Xu<sup>1</sup>, Raimond L. Winslow<sup>1</sup> and Donald Geman<sup>1,2</sup><sup>1</sup>Center for Cardiovascular Bioinformatics and Modeling, Whitaker Biomedical Engineering Institute, 3400 N. Charles Street, Baltimore, MD 21218, USA and <sup>2</sup>Department of Applied Mathematics and Statistics, Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218, USAReceived on May 9, 2005; revised on July 28, 2005; accepted on August 14, 2005  
Advance Access publication August 16, 2005**ABSTRACT**

**Motivation:** Various studies have shown that cancer tissue samples can be successfully detected and classified by their gene expression patterns using machine learning approaches. One of the challenges in applying these techniques for classifying gene expression data is to extract accurate, readily interpretable rules providing biological insight as to how classification is performed. Current methods generate classifiers that are accurate but difficult to interpret. This is the trade-off between credibility and comprehensibility of the classifiers. Here, we introduce a new classifier in order to address these problems. It is referred to as *k*-TSP (*k*-Top Scoring Pairs) and is based on the concept of ‘relative expression reversals’. This method generates simple and accurate decision rules that only involve a small number of gene-to-gene expression comparisons, thereby facilitating follow-up studies.

**Results:** In this study, we have compared our approach to other machine learning techniques for class prediction in 19 binary and multi-class gene expression datasets involving human cancers. The *k*-TSP classifier performs as efficiently as Prediction Analysis of Microarray and support vector machine, and outperforms other learning methods (decision trees, *k*-nearest neighbour and naïve Bayes). Our approach is easy to interpret as the classifier involves only a small number of informative genes. For these reasons, we consider the *k*-TSP method to be a useful tool for cancer classification from microarray gene expression data.

**Availability:** The software and datasets are available at <http://www.ccbm.jhu.edu>

**Contact:** [actan@jhu.edu](mailto:actan@jhu.edu)

## 1 INTRODUCTION

Many different tumors have a similar appearance when observed using routine histological techniques and are therefore difficult to distinguish. Histological approaches for tumor classification are also labor intensive. With advances in microarray technology, it is now possible to monitor global gene expression profiles of cancer tissues and compare them with corresponding normal tissues. Extracting accurate and simple decision rules from such microarray data for classification and prediction tasks is of great interest for biomedical applications. Accurate decision rules are essential for diagnostic purposes, as the treatment options, responses to therapy

and prognoses vary depending on the type, staging and grouping of tumors. However, accurate classification of microarray data poses several challenges to machine learning methods. In particular, we are faced with the ‘small *N*, large *P*’ problem of statistical learning, as the number of genes *P* is typically much larger than the number of samples *N*.

The Top Scoring Pair (TSP) classifier was introduced by Geman *et al.* (2004) as a new classification technique for microarray data based entirely on relative gene expression values, specifically pairwise comparisons between two gene expression levels. The TSP classifier is a parameter-free, data-driven machine learning method, which avoids over-fitting by eliminating the need to perform specific parameter tuning, as in other learning techniques [e.g. support vector machines (SVMs) and neural networks (NN)]. In addition, the TSP classifier provides decision rules that (i) involve very few genes; (ii) are both accurate and transparent; (iii) are largely invariant to any monotonic transformation of the data, as is typical of most data normalization methods; and (iv) provide specific hypotheses for follow-up studies (Geman *et al.*, 2004).

In this paper, we present a new ensemble method, *k*-TSP, a refinement of the original TSP algorithm, which uses exactly *k* pairs of genes for classifying gene expression data. When *k* = 1, this algorithm, referred to simply as TSP necessarily selects a unique pair of genes. More generally, both TSP and *k*-TSP may be seen as special cases of a new classification methodology based on the concept of ‘relative expression reversals.’

We also extend the TSP and *k*-TSP techniques beyond binary classification to the multi-class setting. Three different multi-class decomposition methods are employed in this study, namely (i) One-vs-Others (1-vs-r); (ii) One-vs-One (1-vs-1) and (iii) Hierarchical Classification (HC). We investigate the performance of TSP-family classifiers (TSP and *k*-TSP) on both binary and multi-class data, assessing their credibility and comprehensibility on 19 different human cancer microarray gene expression datasets. We show that our ensemble method (*k*-TSP) performs as efficiently as state-of-the-art methods in classifying microarray data, is generally more efficient in terms of the number of genes employed and provides biologically meaningful decision rules.

## 2 TSP-FAMILY CLASSIFIERS

Consider a gene expression profile consisting of *P* genes {1, ..., *P*} and suppose there are *N* profiles or arrays,  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , available for

\*To whom correspondence should be addressed.

training, these data can then be represented as a matrix of dimension  $P \times N$  in which the expression value of the  $i$ -th gene,  $i \in \{1, \dots, P\}$ , from the  $n$ -th sample is denoted by  $x_{i,n}$ . The column vector  $\mathbf{x}_n = (x_{1,n}, \dots, x_{P,n})$  represents the  $P$  expression values for the  $n$ -th sample.

Let  $(y_1, \dots, y_N)$  be the vector of class labels for the  $N$  samples, where  $y_n \in C = \{C_1, \dots, C_M\}$ , the set of possible class labels. We begin by assuming  $M = 2$ ; for example,  $C_1$  refers to normal tissues and  $C_2$  to cancer tissues. The labeled training set is then  $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ , where  $\mathbf{x}_n$  is the  $n$ -th column vector of the matrix of gene expression profiles. As usual, we regard the expression profile and its class label as random variables, denoted by  $\mathbf{X}$  and  $Y$ , respectively, and we assume that the elements of  $S$  represent independent and identically distributed samples from the underlying probability distribution of  $(\mathbf{X}, Y)$ .

The TSP classifiers are rank-based, meaning that the decision rules only depend on the relative ordering of the gene expression values within each profile. This should not be confused with rank-based methods for determining differentially regulated genes in which the expression values for each fixed gene are ordered within samples. Here, in contrast, the expression values of the  $P$  genes are ordered (most highly expressed, second most highly expressed, etc.) within each fixed profile. Let  $R_{i,n}$  denote the rank of the  $i$ -th gene in the  $n$ -th array (profile). Replacing the expression values  $x_{i,n}$  by their ranks  $R_{i,n}$  results in a new data matrix  $\mathbf{R}$  in which each column is a permutation of  $\{1, \dots, P\}$ .

## 2.1 The TSP classifier

*Learning the TSP classifier* Formulation of the TSP classifier has been described previously (Geman *et al.*, 2004). In essence, we will exploit discriminating information contained in the  $\mathbf{R}$  matrix by focusing on ‘marker gene pairs’  $(i, j)$ , for which there is a significant difference in the probability of the event  $\{R_i < R_j\}$  across the  $N$  samples from class  $C_1$  to  $C_2$ . Here, the quantities of interest are  $p_{ij}(C_m) = \text{Prob}(R_i < R_j \mid Y = C_m)$ ,  $m = \{1, 2\}$ , i.e. the probabilities of observing  $R_i < R_j$  (equivalently,  $x_i < x_j$ ) in each class. These probabilities are estimated by the relative frequencies of occurrences of  $R_i < R_j$  within profiles and over samples. Let  $\Delta_{ij}$  denote the ‘score’ of the gene pair  $(i, j)$ , where  $\Delta_{ij} = |p_{ij}(C_1) - p_{ij}(C_2)|$ . We compute the score  $\Delta_{ij}$  for every pair of genes  $i, j \in \{1, \dots, P\}$ ,  $i \neq j$ . Obviously, pairs of genes with high scores are viewed as most informative for classification. In fact, the TSP classifier defined in Geman *et al.* (2004) depends only on those pairs of genes that achieve the largest score, denoted  $\Delta_{\max}$ .

It is possible for multiple gene pairs to achieve the same top score. In order to eliminate ties and select a unique pair from the TSPs, we use a secondary score based on the rank differences in each sample in each class. For each top-scoring gene pair  $(i, j)$ , we compute the ‘average rank difference’  $\gamma_{ij}$  in class  $C_m$ , defined as

$$\gamma_{ij}(C_m) = \frac{\sum_{n \in C_m} (R_{i,n} - R_{j,n})}{|C_m|}, \quad m = \{1, 2\}, \quad (1)$$

where  $|C_m|$  denotes the number of samples in  $C_m$ . The ‘rank score’ of the gene pair  $(i, j)$  is then defined to be  $\Gamma_{ij} = |\gamma_{ij}(C_1) - \gamma_{ij}(C_2)|$ . We then choose the pair with the largest rank score from those pairs with the score  $\Delta_{\max}$ . The motivation behind using the rank score to break ties is that it incorporates a measure of the magnitude to which

inversions of gene expression levels occur from one class to the other within a pair of genes.

*Prediction with the TSP classifier* If  $(i, j)$  is the unique, distinguished pair selected according to the criterion described above and suppose  $p_{ij}(C_1) > p_{ij}(C_2)$ , the TSP classifier  $h_{\text{TSP}}$  is then defined as follows: let  $\mathbf{x}_{\text{new}}$  represent a new profile, Then

$$y_{\text{new}} = h_{\text{TSP}}(\mathbf{x}_{\text{new}}) = \begin{cases} C_1, & \text{if } R_{i,\text{new}} < R_{j,\text{new}}, \\ C_2, & \text{Otherwise.} \end{cases} \quad (2)$$

On the other hand, if  $p_{ij}(C_2) \geq p_{ij}(C_1)$ , then the decision rule is reversed. Put differently, the TSP classifier chooses the class for which the observed ordering between the expression levels of genes  $i$  and  $j$  is the most likely. It is also noteworthy that the sum of misclassification probabilities over the two classes can be expressed as  $1 - \Delta_{ij}$ , which provides a natural justification for score maximization. The TSP algorithm is illustrated in Supplementary Figure 1. It has been shown to perform well in classifying binary class gene expression data (Geman *et al.*, 2004).

## 2.2 The $k$ -TSP classifier

In some instances, the TSPs may change when the training data are perturbed by adding or deleting a few examples (Geman *et al.*, 2004). Here, we introduce the  $k$ -TSP classifier, which extends the TSP classifier, and is designed to deal with this problem, as well as increase the accuracy of the TSP classifier, by generating a more stable classifier. This is accomplished by basing the classification on the  $k$  disjoint Top Scoring Pairs ( $k$ -TSP) of genes that achieve the best combined score. We can view the  $k$ -TSP as an ensemble learning approach where the intention is to combine the discriminating power of many ‘weaker’ rules to make more reliable predictions. In this case, there are  $k$  ‘weaker’ rules, one for using each of the  $k$ -TSP to classify according to Equation (2).

*Learning the  $k$ -TSP classifier* The learning algorithm of  $k$ -TSP is similar to that of TSP. It consists of first forming a list of gene pairs, sorted from the largest to the smallest according to their original scores  $\Delta_{ij}$ , and then breaking ties by sorting within those achieving the same score  $\Delta$  using the secondary score  $\Gamma_{ij}$ . The  $k$ -TSP classifier uses the  $k$  top scoring *disjoint* gene pairs from this list. The procedure is straightforward: take the first pair  $(i_1, j_1)$ , then go down the list until the first pair  $(i_2, j_2)$  that does not involve either  $i_1$  or  $j_1$  is arrived at, and continue in this manner until the  $k$ -th disjoint pair  $(i_k, j_k)$  is reached. The parameter  $k$  is determined by cross-validation, with the restriction that  $k$  does not exceed 10 in this study and is an odd number in order to break ties in the majority voting procedure. Figure 1 illustrates the  $k$ -TSP learning algorithm.

In order to accelerate cross-validation, we have devised an algorithm that employs an efficient computational shortcut to calculate the cross-validation error. This shortcut creates a pruned list consisting of all the pairs that could possibly be identified among the TSPs and  $k$ -TSPs, no matter which of the original  $N$  samples are removed during a loop of the cross-validation. In brief, for every gene pair, a lower bound and upper bound for the score that could be achieved for that pair, no matter which samples are removed, is calculated. Next, after initializing  $\Theta$  (Fig. 1) to be the list of pairs ordered according to the score lower bound, a list  $\Theta$  is created by applying Step 2d of the  $k$ -TSP algorithm

***k*-TSP Algorithm****Input:** Training sample  $S$  of  $P$  genes and  $N$  arrays.**Output:**  $k$ -TSP classifier  $h_{k\text{-TSP}}$ .

1. Set an upper bound ( $k_{\max}$ ) on the number of top scoring pairs to be included in the final  $k$ -TSP classifier ( $h_{k\text{-TSP}}$ ). ( $k_{\max} = 10$  in this study.)
2. (Cross-validation) Repeat  $m$  times:
  - a. Leave out  $n$  arrays from the training set  $S$ . ( $n = 3$  and  $m = N/3$  in this study.)
  - b. Compute the score  $\Delta_{ij}$  and the rank score  $\Gamma_{ij}$  on the current, reduced training set for every pair of genes  $(i, j)$ ,  $1 \leq i \neq j \leq P$ .
  - c. Make an ordered list  $O$  of all of the gene pairs  $(i, j)$  from largest to smallest using the lexicographic ordering defined by setting  $(i, j) > (i', j')$  whenever either  $\Delta_{ij} > \Delta_{i'j'}$  or  $\Delta_{ij} = \Delta_{i'j'}$  and  $\Gamma_{ij} > \Gamma_{i'j'}$ .
  - d. Initialize  $\Theta$  at the empty list and perform the following steps for  $k=1, 2, \dots, k_{\max}$ :
    - i. Add the top pair  $(i, j)$  in the list  $O$  to  $\Theta$ .
    - ii. Remove every pair from  $O$  that involves either  $i$  or  $j$ .
    - iii. If  $k$  is odd, compute the error rate for the classifier based on the  $k$  pairs in  $\Theta$ .
3. Select the (odd) value of  $k$  whose average classification rate over the  $m$  loops in Step 2 is optimal and compute the classifier  $h_{k\text{-TSP}}$  based on the top  $k$  scoring pairs as follows:
4. Make an ordered list  $O$  of gene pairs as in Steps 2b and 2c using the entire training set.
  - a. Initialize  $\Theta$  at the empty list.
  - b. Repeat  $k$  times:
    - i. Add the top pair  $(i, j)$  in  $O$  to  $\Theta$ .
    - ii. Remove every pair from  $O$  that involves either  $i$  or  $j$ .
5. Return  $h_{k\text{-TSP}}$ .

**Fig. 1.** Description of the  $k$ -TSP algorithm.

$2k_{\max}$  times. Finally, letting  $L$  denote the lower bound for the score of the last pair of  $\Theta$ , the pruned list consists of those pairs whose score upper bound exceeds  $L$ . Even though this algorithm is exact, in the sense that the same  $TSPs$  are chosen with or without it, the amount of computation necessary for cross-validation is greatly reduced; details can be found in the Supplementary Material, Section 2.

**Prediction with the  $k$ -TSP classifier** Given a new profile  $\mathbf{x}_{\text{new}}$ , each gene pair  $(i_u, j_u)$ ,  $u = 1, \dots, k$ , determines an individual classifier  $h_u(\mathbf{x}_{\text{new}})$  according to the decision rule in Equation (2). This yields  $k$  predictions of  $y_{\text{new}}$ . The  $k$ -TSP classifier  $h_{k\text{-TSP}}$  employs an unweighted majority voting procedure to obtain the final prediction of  $y_{\text{new}}$ ; in other words, the  $k$ -TSP classifier simply chooses the class receiving the most votes:

$$y_{\text{new}} = h_{k\text{-TSP}}(\mathbf{x}_{\text{new}}) = \arg \max_{C=C_1, C_2} \sum_{u=1}^k I(h_u(\mathbf{x}_{\text{new}}) = C), \quad (3)$$

where

$$I(h_u(\mathbf{x}_{\text{new}}) = C) = \begin{cases} 1 & \text{if } h_u(\mathbf{x}_{\text{new}}) = C, \\ 0 & \text{otherwise} \end{cases}, \quad C = \{C_1, C_2\}. \quad (4)$$

### 2.3 Multi-class classification

Some classification methods, e.g. SVMs, TSP and  $k$ -TSP, are designed for binary classification problems and others, e.g. nearest-neighbors, decision trees and variants of linear discriminant analysis (LDA), apply immediately to any number of classes. In the former cases, multi-class problems are usually addressed by training and combining a family of binary classifiers dedicated to various binary sub-classification problems. In this study, we investigate the performance of the TSP and  $k$ -TSP classifiers for three different schemes for differentiating among  $M$  classes.

**2.3.1 One-vs-Others (1-vs- $r$ ) scheme.** Given multiple classes  $C = \{C_1, C_2, \dots, C_M\}$ , the One-vs-Others approach decomposes the original problem into a set of  $M$  two-class problems (Supplementary Figure 3a). For each class  $m = 1, \dots, M$ , a classifier is constructed for distinguishing between the individual class  $C_m$  and the composite class  $C \setminus C_m$  consisting of all other classes. To predict the class of a new sample, each of these  $M$  classifiers is evaluated, leading to a set of  $M$  predictions, each of which consists of either a single class or a set of  $M - 1$  classes. The final prediction is chosen from among the *single* classes identified by the  $M$  classifiers; in other words, classifiers that vote for a set of  $M - 1$  classes are ignored. If no single classes are chosen, or if ties occur (i.e. if more than one single class is identified), then the final output is the class with the largest number of training samples.

**2.3.2 One-vs-One (1-vs-1) scheme.** Another well-known approach for extending binary to multi-class classification is the One-vs-One method [also known as pairwise coupling (Hastie and Tibshirani, 1997) or Round Robin ensemble (Furnkranz, 2002)]. A binary classifier  $h_{lm}$  is constructed for each distinct pair of classes  $C_l, C_m \in C, C_l \neq C_m$ , using only the training samples for those classes. Consequently, this approach generates  $M(M - 1)/2$  binary classifiers (Supplementary Figure 3b), each predicting exactly one of the  $M$  classes. In this scheme, the classifiers can be combined by simple voting: the final prediction is the class that appears most often among the  $M(M - 1)/2$  decisions.

**2.3.3 Hierarchical Classification (HC) scheme.** Hierarchical classification is a sequential procedure in which a binary classifier is associated with each internal node of a binary decision tree and a class label is assigned to each leaf of the tree. The classifier  $h_1$  at the root is designed to distinguish between the largest class and the other classes combined ('composite class 1'); it is trained using all of the training samples. If  $h_1$  chooses the largest class, the procedure terminates and this becomes the final prediction. Otherwise, i.e. if  $h_1$  chooses composite class 1, the second classifier,  $h_2$ , is applied, which is dedicated to separating the second largest class from 'composite class 2', consisting of all classes combined except the largest and second largest;  $h_2$  is trained from all examples whose class labels belong to composite class 1. This procedure iterates until all the leaves in the decision-tree are labeled with a unique class (Supplementary Figure 3c). The final prediction for this scheme is obtained by traversing the decision-tree in a top-down fashion and returning the class label of the leaf node that is reached.

### 2.4 Implementation of TSP and $k$ -TSP

The core TSP and  $k$ -TSP programs are written in C++, and wrapped by the multi-class decomposition scheme, which is written in Perl v5.8.0. The program utilizes some UNIX commands for data parsing and manipulation. The software has been tested on three different operating systems: a RedHat 9.0 Linux box of dual 500 MHz Pentium III processors with 1 GB memory; an IBM cluster of 20 1.1 GHz Power4 processors with 24 GB memory running AIX 5.1 and a Windows XP machine of 2.8 GHz dual Xeon processors with 2 GB memory. The software is available at our website (<http://www.ccbm.jhu.edu>).

**Table 1.** Binary class gene expression datasets

Dataset	Platform	No. of genes ( $P$ )	No. of samples ( $N$ )		Reference
			$C_1$	$C_2$	
Colon	cDNA	2000	40 (T)	22 (N)	(Alon <i>et al.</i> , 1998)
Leukemia	Affy	7129	25 (AML)	47 (ALL)	(Golub <i>et al.</i> , 1999)
CNS	Affy	7129	25 (C)	9 (D)	(Pomeroy <i>et al.</i> , 2002)
DLBCL	Affy	7129	58 (D)	19 (F)	(Shipp <i>et al.</i> , 2002)
Lung	Affy	12 533	150 (A)	31 (M)	(Gordon <i>et al.</i> , 2002)
Prostate1	Affy	12 600	52 (T)	50 (N)	(Singh <i>et al.</i> , 2002)
Prostate2	Affy	12 625	38 (T)	50 (N)	(Stuart <i>et al.</i> , 2004)
Prostate3	Affy	12 626	24 (T)	9 (N)	(Welsh <i>et al.</i> , 2001)
GCM	Affy	16 063	190 (C)	90 (N)	(Ramaswamy <i>et al.</i> , 2001)

**Table 2.** Multi-class gene expression datasets

Dataset	Platform	No of classes	No of genes ( $P$ )	No. of samples ( $N$ )		Reference
				Training	Testing	
Leukemia1	Affy	3	7129	38	34	(Golub <i>et al.</i> , 1999)
Lung1	Affy	3	7129	64	32	(Beer <i>et al.</i> , 2002)
Leukemia2	Affy	3	12 582	57	15	(Armstrong <i>et al.</i> , 2002)
SRBCT	cDNA	4	2308	63	20	(Khan <i>et al.</i> , 2001)
Breast	Affy	5	9216	54	30	(Perou <i>et al.</i> , 2000)
Lung2	Affy	5	12 600	136	67	(Bhattacharjee <i>et al.</i> , 2001)
DLBCL	cDNA	6	4026	58	30	(Alizadeh <i>et al.</i> , 2000)
Leukemia3	Affy	7	12 558	215	112	(Yeoh <i>et al.</i> , 2002)
Cancers	Affy	11	12 533	100	74	(Su <i>et al.</i> , 2001)
GCM	Affy	14	16 063	144	46	(Ramaswamy <i>et al.</i> , 2001)

### 3 MICROARRAY DATA AND EVALUATION METHODS

In the following sections we investigate the performance of TSP-family classifiers on both binary and multi-class expression datasets. For this purpose, we have collected 19 publicly available microarray datasets, with sample sizes ranging from 33 to 327 and numbers of genes ranging from 2000 to 16 063. All of the datasets, which are summarized in Tables 1 and 2, are related to studies of human cancer, including colorectal, leukemia, lung, prostate, breast, central nervous system, lymphoma, bladder, melanoma, renal, uterus, pancreas, ovary and mesothelioma. Further information can be obtained from the related publications.

#### 3.1 Other machine learning methods

We compare the performance of TSP-family classifiers with five well-known machine learning methods: C4.5 decision trees (DT), Naïve Bayes (NB),  $k$ -nearest neighbor ( $k$ -NN), Support Vector Machines (SVM) and prediction analysis of microarrays (PAM). We use the DT, NB,  $k$ -NN and SVM implemented in the WEKA machine learning package (Witten and Frank, 2000) and the PAM Windows version 1.22 program (Tibshirani *et al.*, 2002) for all the experiments in this study.

Since DT and NB directly handle multi-class problems, we use the default parameters for these techniques. For  $k$ -NN, the number of neighbors  $k$  is determined using cross-validation on the training set. The SVMs are trained using sequential minimal optimization with a linear kernel and extended to multi-class problems using both the (i) 1-vs-1 and (ii) 1-vs-r schemes.

PAM is a variation of diagonal LDA and one of the most popular statistical methods for analyzing gene expression data. PAM is a statistical

technique developed by Tibshirani *et al.* (2002) based on the nearest shrunken centroids approach. We perform cross-validation on the training set to determine the optimal amount of shrinkage (tuning parameter of PAM) for each dataset. Other than that, we apply the default parameters of the PAM program.

#### 3.2 Estimation of classification rate

*Leave-One-Out Cross-Validation (LOOCV) for binary class problems.* In order to estimate the classification error rate for the binary classification problems listed in Table 1 we use standard LOOCV. Hence, for each sample  $\mathbf{x}_n$  in the training set  $S$ , we train a classifier based on the remaining  $N - 1$  samples in  $S$  and use that classifier to predict the label of  $\mathbf{x}_n$ . The LOOCV estimate of the classification rate is the fraction of the  $N$  samples that are correctly classified.

*Independent test set for multi-class problems.* In order to evaluate the performance of the multi-class problems in this study (Table 2), we use the test sets provided from the original references when available. Otherwise, we randomly partition each dataset into a training set and a test set. We train the classifiers on the training set and evaluate their performance on the independent test set.

*Estimation of  $k$  in  $k$ -TSP.* The parameter  $k$  in the  $k$ -TSP classifier is determined by cross-validation, as described in Fig. 1. This requires a double loop of cross-validation in the case of estimating the classification rate from LOOCV in the binary classification problems, an outer loop for estimating the generalization error and an inner loop for estimating  $k$ . Only a single loop of cross-validation is necessary when there is an independent test set available (multi-class datasets).

**Table 3.** LOOCV accuracy of classifiers for binary class expression datasets

Method	Leukemia	CNS	DLBCL	Colon	Prostate1	Prostate2	Prostate3	Lung	GCM	Average
TSP	93.80	77.90	<b>98.10</b>	<b>91.10</b>	<b>95.10</b>	67.60	97.00	98.30	75.40	88.26
<i>k</i> -TSP	95.83	<b>97.10</b>	97.40	90.30	91.18	75.00	97.00	98.90	85.40	<b>92.01</b>
DT	73.61	67.65	80.52	80.65	87.25	64.77	84.85	96.13	77.86	79.25
NB	<b>100.00</b>	82.35	80.52	58.06	62.75	73.86	90.91	97.79	84.29	81.17
<i>k</i> -NN	84.72	76.47	84.42	74.19	76.47	69.32	87.88	98.34	82.86	81.63
SVM	98.61	82.35	97.40	82.26	91.18	76.14	<b>100.00</b>	<b>99.45</b>	<b>93.21</b>	91.18
PAM	97.22	82.35	85.71	85.48	91.18	<b>79.55</b>	<b>100.00</b>	<b>99.45</b>	79.29	88.91

The best prediction rate for each particular data set is highlighted in boldface.

**Table 4.** Accuracy of classifiers for the independent test set for multi-class expression datasets

Method	Leuk1	Lung1	Leuk2	SRBCT	Breast	Lung2	DLBCL	Leuk3	Cancers	GCM	Average
HC-TSP	<b>97.06</b>	71.88	80.00	95.00	66.67	83.58	83.33	77.68	74.32	52.17	78.17
HC- <i>k</i> -TSP	<b>97.06</b>	78.13	<b>100.00</b>	<b>100.00</b>	66.67	94.03	83.33	82.14	82.43	<b>67.39</b>	85.12
DT	85.29	78.13	80.00	75.00	73.33	88.06	86.67	75.89	68.92	52.17	76.35
NB	85.29	81.25	<b>100.00</b>	60.00	66.67	88.06	86.67	32.14	79.73	52.17	73.20
<i>k</i> -NN	67.65	75.00	86.67	30.00	63.33	88.06	93.33	75.89	64.86	34.78	67.96
1-vs-1-SVM	79.41	<b>87.50</b>	<b>100.00</b>	<b>100.00</b>	83.33	97.01	<b>100.00</b>	84.82	83.78	65.22	<b>88.11</b>
PAM	<b>97.06</b>	78.13	93.33	95.00	<b>93.33</b>	<b>100.00</b>	90.00	<b>93.75</b>	<b>87.84</b>	56.52	<b>88.50</b>

The best prediction rate for each dataset is highlighted in boldface.

## 4. RESULTS

### 4.1 The *k*-TSP classifier performs comparably to PAM and SVM for the binary classification problems

Table 3 summarizes the results of LOOCV using the seven different classifiers on the nine binary classification problems. In this case, the estimated classification rate is  $(TP + TN)/N$ , where TP denotes the number of correctly classified  $C_1$  samples, TN denotes the number of correctly classified  $C_2$  samples and  $N$  is the total sample size.

Our results show that the seven classification methods can be roughly divided into two groups. Averaged over the nine problems, the top tier classifiers (*k*-TSP, SVM, TSP and PAM) achieve accuracies in the vicinity of 90% and the second-tier classifiers (*k*-NN, NB and DT) in vicinity of 80%. In this study (Table 3), *k*-TSP outperforms PAM in four cases (CNS, DLBCL, Colon and GCM), PAM outperforms *k*-TSP in four cases (Leukemia, Prostate2, Prostate3 and Lung) and they perform the same in one case (Prostate1). SVM is superior to *k*-TSP in classifying five datasets (Leukemia, Prostate2, Prostate3, Lung and GCM), inferior in two cases (CNS and Colon) and the same in two cases (DLBCL and Prostate1).

The best classifier based on the average accuracy for the nine binary classification problems used in this study is *k*-TSP (92.01%), followed by SVM (91.18%), PAM (88.91%) and TSP (88.26%). We do not consider these differences in accuracy as noteworthy and conclude that all four methods perform similarly. However, in terms of efficiency and simplicity, one can argue that the TSP method is superior since it uses a single

pair of genes and an elementary decision rule based solely on expression inversion. These results confirm the findings in Geman *et al.* (2004) that TSP-family classifiers can generate accurate and interpretable decision rules for classifying microarray data.

### 4.2 The HC-*k*-TSP classifier performs comparably to SVM and PAM in multi-class problems

Table 4 summarizes the performance of the 7 methods for the 10 multi-class problems. Recall that DT, NB, *k*-NN and PAM directly handle multiple classes. In order to simplify the presentation of the results for TSP, *k*-TSP and SVM, in Table 4 we only present the multi-class scheme that performs best for each of these methods, 1-vs-1 for SVMs and HC for TSP and *k*-TSP; the full results for all multi-class schemes are available in Supplementary Table 1. One general observation from the multi-class experiments is that the accuracy of the classifiers decreases as the number of classes increases. This is due, at least in part, to the small number of training samples for many of the classes, which makes learning more difficult.

From the results (Table 4), PAM (88.50%) and 1-vs-1-SVM (88.10%) yield the best performance averaged over the 10 problems and each outperforms the other in 5 of the 10 cases. HC-*k*-TSP achieves an average accuracy of 85.12%. Compared with PAM, HC-*k*-TSP is superior in three cases (Leukemia2, SRBCT and GCM), inferior in five cases (Breast, Lung2, DLBCL, Leukemia3 and Cancers) and the same in two cases (Leukemia1 and Lung1). The situation is approximately the same when HC-*k*-TSP is compared with 1-vs-1-SVM.

**Table 5.** Number of genes used in the classifiers for binary class expression datasets

Method	Leukemia	CNS	DLBCL	Colon	Prostate1	Prostate2	Prostate3	Lung	GCM
TSP	2	2	2	2	2	2	2	2	2
<i>k</i> -TSP	18	10	2	2	2	18	2	10	10
DT	2	2	3	3	4	4	1	3	14
PAM	2296	4	17	15	47	13	701	9	47

**Table 6.** Number of genes used in the classifiers for multi-class expression datasets

Method	Leuk1	Lung1	Leuk2	SRBCT	Breast	Lung2	DLBCL	Leuk3	Cancers	GCM
HC-TSP	4	4	4	6	8	8	10	12	20	26
HC- <i>k</i> -TSP	36	20	24	30	24	28	46	64	128	134
DT	2	4	2	3	4	5	5	16	10	18
PAM	44	13	62	285	4822	614	3949	3338	2008	1253

## 5. DISCUSSION

### 5.1 Number of genes

We have shown that TSP-family classifiers are comparable in accuracy with state-of-the-art classification methods such as PAM and SVM in a variety of cancer classification problems using gene microarray data. One interesting observation is that the number of genes involved in both the TSP and *k*-TSP classifiers is significantly lower in most cases than the number used by PAM (which is chosen by cross-validation); the SVM classifier uses all the genes.

Tables 5 and 6 show the number of genes used by the TSP, *k*-TSP, DT and PAM classifiers for binary and multi-class datasets, respectively. Since NB, *k*-NN and SVMs perform classification using all the genes, these methods are not included in Tables 5 and 6.

For binary classification, the TSP classifier is naturally the most efficient since, by construction, it only uses a single discriminating pair of genes. The number of genes used in *k*-TSP is (by design) fewer than 20, yet it achieves superior performance in the binary class expression problems.

Although DT also uses a relatively small number of genes for classification, its performance is significantly worse than TSP and *k*-TSP (Table 3), suggesting that the chosen features may overfit the training data and may be sensitive to noise. In fact, DT is known to be a sensitive classifier; small perturbations in the training samples lead to large differences in its tree-structure (Dietterich, 2000; Tan and Gilbert, 2003).

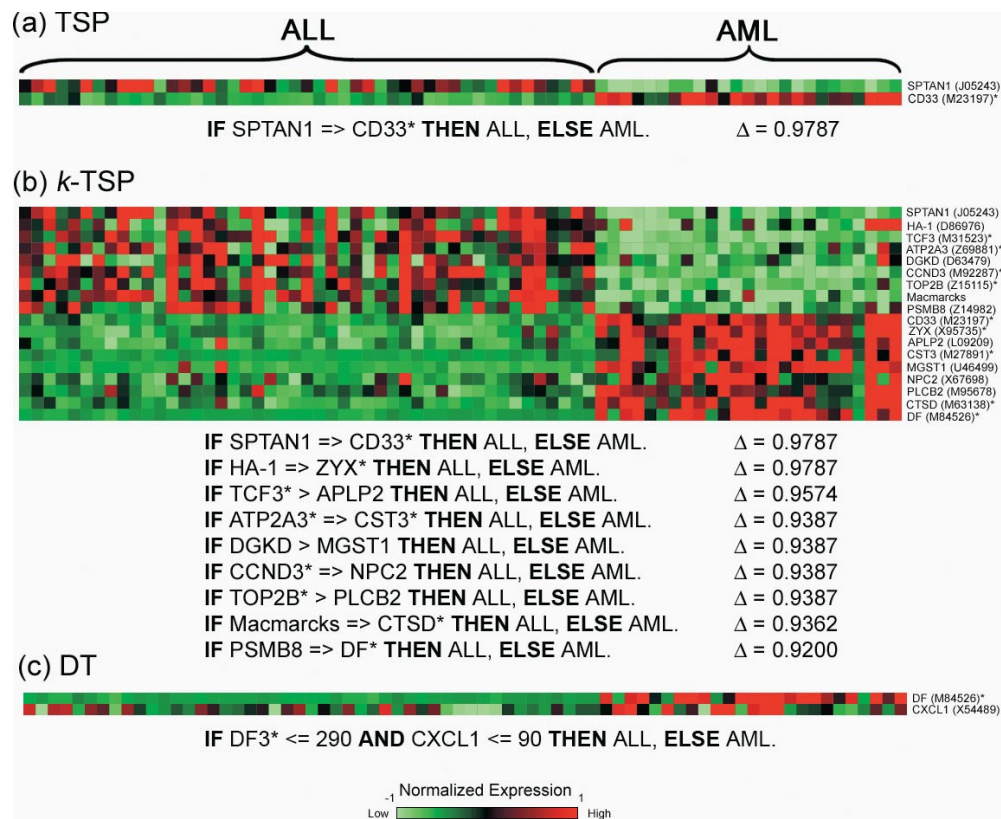
For the multi-class problems, DT utilizes the smallest number of genes. As expected, for the TSP-family classifiers, the number of genes increases according to the number of classifiers used in the hierarchical scheme (see Section 2.3), which is smaller than the number of classifiers in the 1-vs-r and 1-vs-1 schemes (Supplementary Table 2). Eventhough the number of genes increases relative to the binary case, the TSP-family classifiers still maintain reasonably transparent results. In contrast, PAM and SVM achieved slightly higher accuracy in these problems, the potential for post-analysis study and biological interpretation is questionable. For PAM, the concept of the nearest centroid has intuitive appeal, the number of

genes that figure in the decision rule can far exceed one thousand, as shown in Table 6; in the case of SVM, the decision boundary is a linear function of the entire input vector  $\mathbf{x}_{\text{new}}$  and many support vectors from  $S$  may participate in determining the coefficients.

Several studies have shown that it is possible to reduce the number of genes by using gene selection methods before training a classifier. The simplest way of doing gene filtering is to introduce a requirement of statistical significance of individual genes based on measurements such as *t*-test or the commonly used signal-to-noise ratio (Golub *et al.*, 1999; Li *et al.*, 2004). An alternative approach to gene selection is to apply filtering and subset selection algorithms from machine learning (Bø and Jonassen, 2002; Guyon *et al.*, 2002). Gene filtering can improve the accuracy of classification. However the performance of a gene selection method may depend on the nature of the classifier, the criterion for selection and the number of genes selected (Dudoit and Fridlyand, 2003; Li *et al.*, 2004). As opposed to most other gene selection approaches, the choice of the number of gene pairs in the *k*-TSP classifier is systematically determined by an internal cross-validation loop in the training step.

### 5.2 Invariance to platforms and pre-processing

The TSP and *k*-TSP decision rules only use the ordering of the expression values within profiles; in fact, only a selected number of pairwise comparisons are utilized. However, other methods rely on the actual expression values and are therefore sensitive to pre-processing, such as scaling and normalization, as well as to the manner in which the data are collected. For example, the decision rules derived from the DT classifier (Fig. 2c) are based on comparing individual expression values to a fixed threshold. As a result, the expression values will typically vary according to the particular pre-processing methods employed in different studies and experiments, rendering it difficult to apply conventional decision rules, such as those found in decision trees, to other technologies or studies. In contrast, the TSP decision rule can be readily applied in clinical settings across different technology platforms since the outcome of gene-to-gene comparisons will usually be independent of pre-processing based on scaling and other forms of normalizing microarrays.



**Fig. 2.** Genes that distinguish ALL from AML. Each row corresponds to a gene and each column corresponds to a sample array. Genes labeled with an asterisk (\*) were identified in Golub *et al.* (1999). This heat map is generated by using the matrix2png software (Pavlidis and Noble, 2003). The expression level for each gene is normalized across the samples such that the mean is 0 and the standard deviation (SD) is 1. Genes with expression levels greater than the mean are colored in red and those below the mean are colored in green. The scale indicates the number of SDs above or below the mean. In (a–c), the discriminative genes and decision rules in three cases are shown: (a) TSP Classifier, (b) *k*-TSP Classifier and (c) Decision tree (DT) classifier.

### 5.3 Characterization of *k*-TSP as an ensemble method

Various empirical observations and studies have shown that it is unusual for a single learning algorithm to outperform other learning methods in all problem domains. Random Forests (Amit and Geman, 1997; Breiman, 2001), bagging (Breiman, 1996) and boosting (Freund and Schapire, 1996, 1997) represent recent success stories of ensemble methods, and all have been shown to perform well in classifying different microarray datasets (Dudoit *et al.*, 2002; Dettling and Buhlmann, 2003; Long and Vega, 2003; Tan and Gilbert, 2003).

The *k*-TSP method can be seen as a straightforward extension of the TSP classifier to an elementary ensemble approach in which the ‘base classifiers’ are the TSP classifiers for the top-scoring *k* disjoint pairs of genes. Consequently, the *k*-TSP classifier maintains interpretability at the same time often improving the accuracy of the TSP classifier by recruiting additional ‘weaker’ classifiers in the final decision-making process.

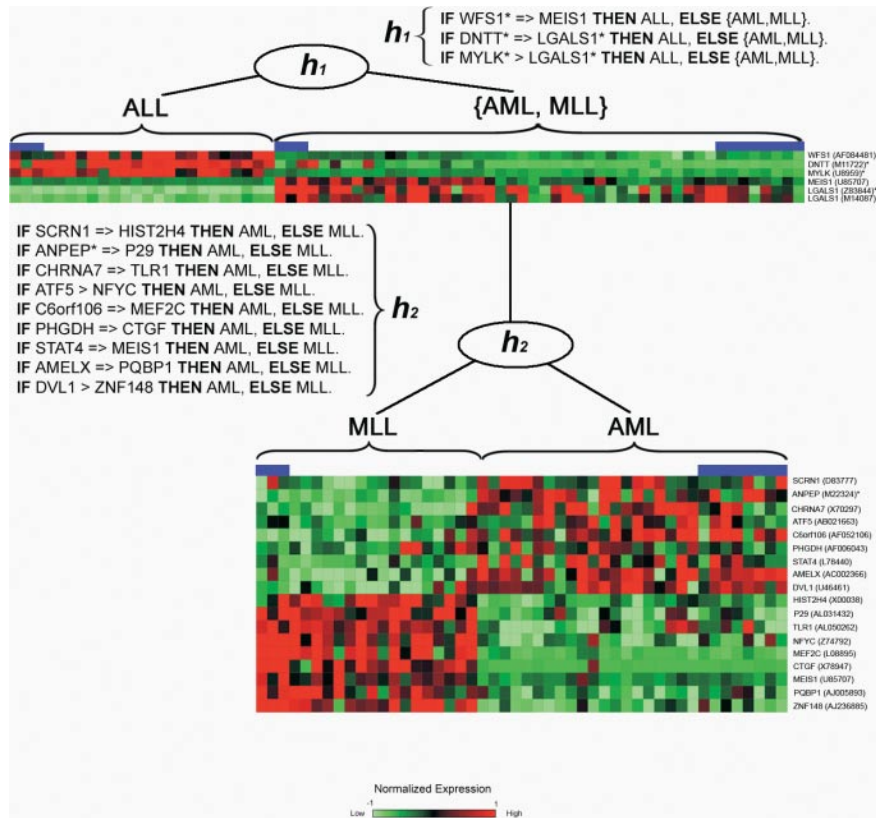
### 5.4 Interpretation and biological significance of the TSP-family classifiers

*Interpretation of TSP.* The TSP classifier can be easily translated into a set of IF-ELSE decision rules describing the relationship between the relative expression levels of the informative genes

and the class labels, as illustrated in Figure 2a for the Leukemia dataset (Golub *et al.*, 1999). The gene pair (SPTAN1, CD33) is induced by the TSP learning algorithm for distinguishing ALL (acute lymphoblastic leukemia) from AML (acute myeloid leukemia). The corresponding decision rule is

$$\text{IF SPTAN1} \geq \text{CD33 THEN ALL; ELSE AML.}$$

In words: *if the expression of SPTAN1 is greater than or equal to CD33, then the sample is classified as ALL, otherwise AML.* This simple decision rule has an estimated accuracy of 93.80% (using LOOCV). CD33 is one of the genes listed in the ALL vs AML predictor in Golub *et al.* (1999), which is based on fifty genes. CD33 encodes a cell surface protein and SPTAN1 is involved in secretion and it interacts with calmodulin in a calcium-dependent manner. Early studies (Griffin *et al.*, 1983; Bernstein *et al.*, 1992) have identified CD33 as a cell surface marker for AML, while several studies have successfully demonstrated the use of monoclonal antibodies in discriminating AML from ALL (Golub *et al.*, 1999), indicating that CD33 may be a therapeutic target for AML. In another study using gene expression data to distinguish subtypes of leukemia (Armstrong *et al.*, 2002), SPTAN1 is found to be over-expressed in ALL compared with AML. These findings confirm the biological significance of the genes identified by the TSP.



**Fig. 3.** Hierarchical classification of leukemia subtypes ALL, AML and MLL, using  $k$ -TSP. Rows and columns correspond to genes and samples, respectively. Genes labeled with an asterisk (\*) were previously identified as discriminating genes for this problem in Armstrong *et al.* (2002). The blue panel denotes the independent test samples. HC- $k$ -TSP consists of sequentially applying two  $k$ -TSP decision rules: the first classifier  $h_1$  distinguishes ALL from {AML, MLL} based on three (top-scoring) pairs of genes and the second classifier  $h_2$  discriminates MLL from AML using nine pairs. The heat maps generated the same way as in Fig. 2.

**Interpretation of  $k$ -TSP.** In Figure 2b, we illustrate the decision rules derived by the  $k$ -TSP classifier using the Leukemia dataset. The  $k$ -TSP classifier that distinguishes ALL from AML contains 9 modular rules, involving 18 genes. Of these 18 genes, 9(CD33, ZYX, TCF3, CST3, ATP2A3, CCND3, TOP2B, CTSD and DF) are among the 50 singled out in Golub *et al.* (1999) for distinguishing ALL from AML. Recall that  $k$  is chosen by cross-validation in contrast to the arbitrary choices for some of the parameters in many other methods.

These nine genes have known biological correlation with cancer pathogenesis. Genes CCND3 and ZYX are involved in cell development and adhesion, respectively. CD33 is a specific marker for AML, TCF3 is a known oncogene and TOP2B is a target of the anti-leukemia drug etoposide (Golub *et al.*, 1999). In addition, we found that other genes used in the  $k$ -TSP classifier, such as HA-1 and APLP2, have been linked with leukemia (Mutis *et al.*, 1999; Yang, 2004).

**Interpretation of HC- $k$ -TSP.** Finally, consider the example of using the  $k$ -TSP classifier to distinguish among three subtypes of leukemia. Armstrong *et al.* (2002) identified specific genes involved in chromosomal translocation of the human acute leukemia known as the mixed-lineage leukemia (MLL). This subtype of leukemia is aggressive and is associated with poorer prognosis compared with ALL and AML. Using gene expression profiling techniques, they

have identified discriminative groups of genes that are useful in classifying these leukemia subtypes. Here, using  $k$ -TSP in the context of hierarchical classification (HC- $k$ -TSP), it gives 100% accuracy when tested on 15 independent test leukemia samples (4 ALL, 3 MLL and 8 AML), as does SVM (Table 4). Figure 3 illustrates the decision rules of the HC- $k$ -TSP classifier learned from the leukemia subtypes dataset (Leukemia2 dataset in Tables 2 and 4).

Investigating the genes appearing in the HC- $k$ -TSP class reveals that 7 out of the total of 24 were also listed by Armstrong *et al.* (2002). DNTT, WFS1 and MYLK have been identified by Armstrong *et al.* (2002) as the top 100 under-expressed genes in MLL as compared with ALL. Similarly, two different probe sets of LGALS1 were listed in the top 100 over-expressed genes in MLL compared with ALL, by Armstrong *et al.* (2002), and ANPEP is highly expressed in AML and is included in the list of 45 genes in distinguishing ALL-AML-MLL, by Armstrong *et al.* (2002). In addition, MEIS1, a cofactor of HOX, is found to be over-expressed in MLL in two independent gene expression studies (Yeoh *et al.*, 2002; Tsutsumi *et al.*, 2003). Yeoh *et al.* (2002) suggest that MEIS1 may be directly involved in MLL-mediated alterations in the growth of the leukemia cells. P29 is thought to be related to the functional regulation of GCIP, a protein that is involved in cell cycle progression and the regulation of transcriptional factors (Chang *et al.*, 2000).



## 5.5 Comparing multi-class schemes for the TSP-family classifiers

In general, our results show that the HC scheme is the best. One difficulty with the 1-vs-r scheme is the unbalanced sample sizes when a small class is trained against all others combined, perhaps resulting in over-sensitivity to the examples in the small class. One difficulty with the 1-vs-1 scheme is the number of classifiers that must be trained, which grows quadratically with  $M$ , the number of classes and the corresponding loss of interpretability. The HC scheme is less sensitive to sample imbalance and maintains better interpretability since the number of classifiers is linear in  $M$ .

## 6 CONCLUSIONS

In this paper, we have introduced two examples of classification based on relative expression reversals: a version of the original TSP classifier, which provides a decision rule based on exactly two genes and an extension of this classifier to learn a decision rule based on  $k$ -disjoint pairs of genes. All classifiers in this TSP-family are based entirely on the ordering of the gene expression values within profiles and hence are largely invariant to pre-processing. We have compared TSP-family classifiers with 5 different machine learning methods on 19 gene expression datasets involving human cancers, comprising both binary and multi-class classification problems. From our results, TSP and  $k$ -TSP perform approximately the same as the PAM and SVM classifiers on these data, but provide decision rules that usually involve many fewer genes and are far easier to interpret. Finally, the genes appearing in these TSP decision rules, which are automatically selected by the TSP learning algorithms, have clear biological connections to their corresponding cancer types.

## ACKNOWLEDGEMENTS

This work was supported by the Falk Medical Trust and NIH RO1-HL72488.

*Conflict of interest:* none declared.

## REFERENCES

- Alizadeh, A.A. et al. (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.
- Alon, U. et al. (1998) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA*, **96**, 6745–6750.
- Amit, Y. and Geman, D. (1997) Shape quantization and recognition with randomized trees. *IEEE Trans. Pattern Anal. Machine Intell.*, **19**, 1300–1305.
- Armstrong, S. et al. (2002) MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat. Genet.*, **30**, 41–47.
- Beer, D.G. et al. (2002) Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat. Med.*, **8**, 816–824.
- Bernstein, I. et al. (1992) Differences in the frequency of normal and clonal precursors of colony-forming cells in chronic myelogenous leukemia and acute myelogenous leukemia. *Blood*, **79**, 1811–1816.
- Bhattacharjee, A. et al. (2001) Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl Acad. Sci. USA*, **98**, 13790–13795.
- Bø, T.H. and Jonassen, I. (2002) New feature subset selection procedures for classification of expression profiles. *Genome Biol.*, **3**, research0017.0011–0017.0011.
- Breiman, L. (1996) Bagging predictors. *Mach. Learn.*, **24**, 123–140.
- Breiman, L. (2001) Random forests. *Mach. Learn.*, **45**, 5–32.
- Chang, M.-S. et al. (2000) p29, a novel GCIP-interacting protein, localizes in the nucleus. *Biochem. Biophys. Res. Commun.*, **279**, 732–737.
- Detting, M. and Buhlmann, P. (2003) Boosting for tumor classification with gene expression data. *Bioinformatics*, **19**, 1061–1069.
- Dieterich, T.G. (2000) An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Mach. Learn.*, **40**, 139–157.
- Dudoit, S. and Fridlyand, J. (2003) Classification in microarray experiments. In Speed, T.P. (ed.), *Statistical Analysis of Gene Expression Microarray Data*. Chapman & Hall/CRC, pp. 93–158. In pp.
- Dudoit, S.J. et al. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc.*, **97**, 77–87.
- Freund, Y. and Schapire, R.E. (1996) Experiments with a new boosting algorithm. Morgan Kaufmann, In *Proceedings of the 13th International Conference on Machine Learning (ICML 96)*, 148–156.
- Freund, Y. and Schapire, R.E. (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, **55**, 119–139.
- Furnkranz, J. (2002) Round robin classification. *J. Mach. Learn. Res.*, **2**, 721–747.
- Geman, D. et al. (2004) Classifying gene expression profiles from pairwise mRNA comparisons. *Stat. Appl. Genet. Mol. Biol.*, **3**, Article 19.
- Golub, T.R. et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.
- Gordon, G.J. et al. (2002) Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Res.*, **62**, 4963–4967.
- Griffin, J.D. et al. (1983) Surface marker analysis of acute myoblastic leukemia: identification of differentiation-associated phenotypes. *Blood*, **62**, 557–563.
- Guyon, I. et al. (2002) Gene selection for cancer classification using support vector machines. *Mach. Learn.*, **46**, 389–422.
- Hastie, T. and Tibshirani, R. (1997) Classification by pairwise coupling. In Jordan, M.I., Kearns, M.J. and Solla, S.A. (eds), *Advances in Neural Information Processing Systems 10*. MIT Press, pp. 507–513.
- Khan, J. et al. (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med.*, **7**, 673–679.
- Li, T. et al. (2004) A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, **20**, 2429–2437.
- Long, P.M. and Vega, V.B. (2003) Boosting and microarray data. *Mach. Learn.*, **52**, 31–44.
- Mutis, T. et al. (1999) Feasibility of immunotherapy of relapsed leukemia with *ex vivo*-generated cytotoxic T lymphocytes specific for hematopoietic system-restricted minor histocompatibility antigens. *Blood*, **93**, 2336–2341.
- Pavlidis, P. and Noble, W.S. (2003) Matrix2png: a utility for visualizing matrix data. *Bioinformatics*, **19**, 295–296.
- Perou, C.M. et al. (2000) Molecular portraits of human breast tumours. *Nature*, **406**, 747–752.
- Pomeroy, S.L. et al. (2002) Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, **415**, 436–442.
- Ramaswamy, S. et al. (2001) Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl Acad. Sci. USA*, **98**, 15149–15154.
- Shipp, M.A. et al. (2002) Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nat. Med.*, **8**, 68–74.
- Singh, D. et al. (2002) Gene expression correlates of clinical prostate cancer behaviour. *Cancer Cell*, **1**, 203–209.
- Stuart, R.O. et al. (2004) In silico dissection of cell-type-associated patterns of gene expression in prostate cancer. *Proc. Natl Acad. Sci. USA*, **101**, 615–620.
- Su, A.I. et al. (2001) Molecular classification of human carcinomas by use of gene expression signatures. *Cancer Res.*, **61**, 7388–7393.
- Tan, A.C. and Gilbert, D. (2003) Ensemble machine learning on gene expression data for cancer classification. *Appl. Bioinformatics*, **2**, S75–S83.
- Tibshirani, R. et al. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl Acad. Sci. USA*, **99**, 6567–6572.
- Tsutsumi, S. et al. (2003) Two distinct gene expression signatures in pediatric acute lymphoblastic leukemia with MLL rearrangements. *Cancer Res.*, **63**, 4882–4887.
- Welsh, J.B. et al. (2001) Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Res.*, **61**, 5974–5978.
- Witten, I.H. and Frank, E. (2000) *Data Mining: Practical machine learning tools and techniques with java implementations*. Morgan Kaufmann.
- Yang, X.-J. (2004) The diverse superfamily of lysine acetyltransferases and their roles in leukemia and other diseases. *Nucl. Acids Res.*, **32**, 959–976.
- Yeoh, A.E.-J. et al. (2002) Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, **1**, 133–143.