BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 5 2005, pages 631–643
doi:10.1093/bioinformatics/bti033

*Gene expression*

# A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis

Alexander Statnikov[1,*], Constantin F. Aliferis[1], Ioannis Tsamardinos[1], Douglas Hardin[2] and Shawn Levy[1]

[1]Department of Biomedical Informatics and [2]Department of Mathematics, Vanderbilt University, Nashville, TN, USA

Received on January 21, 2004; revised on July 29, 2004; accepted on September 10, 2004
Advance Access publication September 16, 2004

**ABSTRACT**

**Motivation:** Cancer diagnosis is one of the most important emerging clinical applications of gene expression microarray technology. We are seeking to develop a computer system for powerful and reliable cancer diagnostic model creation based on microarray data. To keep a realistic perspective on clinical applications we focus on multicategory diagnosis. To equip the system with the optimum combination of classifier, gene selection and cross-validation methods, we performed a systematic and comprehensive evaluation of several major algorithms for multicategory classification, several gene selection methods, multiple ensemble classifier methods and two cross-validation designs using 11 datasets spanning 74 diagnostic categories and 41 cancer types and 12 normal tissue types.

**Results:** Multicategory support vector machines (MC-SVMs) are the most effective classifiers in performing accurate cancer diagnosis from gene expression data. The MC-SVM techniques by Crammer and Singer, Weston and Watkins and one-versus-rest were found to be the best methods in this domain. MC-SVMs outperform other popular machine learning algorithms, such as *k*-nearest neighbors, back-propagation and probabilistic neural networks, often to a remarkable degree. Gene selection techniques can significantly improve the classification performance of both MC-SVMs and other non-SVM learning algorithms. Ensemble classifiers do not generally improve performance of the best non-ensemble models. These results guided the construction of a software system GEMS (Gene Expression Model Selector) that automates high-quality model construction and enforces sound optimization and performance estimation procedures. This is the first such system to be informed by a rigorous comparative analysis of the available algorithms and datasets.

**Availability:** The software system GEMS is available for download from http://www.gems-system.org for non-commercial use.

**Contact:** alexander.statnikov@vanderbilt.edu

## 1 INTRODUCTION

An important emerging medical application domain for microarray gene expression profiling technology is clinical decision support in the form of diagnosis of disease as well as the prediction of clinical outcomes in response to treatment. The two areas in medicine that currently attract the greatest attention in this respect are management of cancer and infectious diseases (Fortina *et al*., 2002; Ntzani and Ioannidis, 2003).

A necessary prerequisite for the creation of clinically successful microarray-based diagnostic models is a solid understanding of the relative strengths and weaknesses of available classification and related methods (i.e. gene selection and cross-validation). Although prior research has established the feasibility of creating accurate models for cancer diagnosis, the corresponding studies conducted limited experiments in terms of the number of classifiers, gene selection algorithms, number of datasets and types of cancer involved (e.g. Yeo and Poggio, 2001; Su *et al*., 2001; Ramaswamy *et al*., 2001; Yeang *et al*., 2001; Lee and Lee, 2003). In addition, the results of these studies cannot be combined into a comprehensive comparative meta-analysis because each study follows different experimental protocols and applies learning algorithms differently. Thus, it is not clear from the literature which classifier, if any, performs best among the many available alternatives. It is also currently poorly understood what are the best combinations of classification and gene selection algorithms across most array-based cancer datasets.

Another major methodological concern is the problem of *overfitting*; that is creating diagnostic models that may not generalize well to new data from the same cancer types and data distribution despite excellent performance on the training set. Since many algorithms are highly parametric and datasets consist of a relatively small number of high-dimensional samples, it is easy to overfit both the classifiers and the gene selection procedures especially when using intensive model search and powerful learners. Indeed recently, a number of reports appeared in the literature raising doubts about the generalization ability of classifiers produced by major studies in the field (Schwarzer and Vach, 2000; Reunanen, 2003; Guyon *et al*., 2003, http://www.clopinet.com/isabelle/Papers/RFE-erratum.html). In recent meta-analytic assessment of 84 published microarray cancer outcome predictive studies (Ntzani and Ioannidis, 2003), it was found that 74% of the studies did not perform independent validation or cross-validation of proposed findings, 13% applied cross-validation in an incomplete fashion and only 13% performed cross-validation correctly. On the other hand, when building a diagnostic model, one should avoid *underfitting* as well, which

*To whom correspondence should be addressed.

results in the classifiers that are not optimally robust due to limited experimentation. In particular, this is manifested by the application of a specific learning algorithm without consideration of alternatives, or use of parametric learners with certain values of parameters without searching for the best ones.

At the heart of the present work lies the thesis that *an automated system can help with the creation of high-quality and robust diagnostic and prognostic models*. For such a system to be successful, it must implement the best possible classification and gene selection algorithms for the domain and guide model selection by enforcing sound principles of model building and data analysis. Hence, to inform the development of such a system, the goals of the present work are to (1) investigate which one among the many powerful classifiers currently available for gene expression diagnosis performs the best across many cancer types; (2) how classifiers interact with existing gene selection methods in datasets with varying sample size, number of genes and cancer types; (3) whether it is possible to increase diagnostic performance further using meta-learning in the form of ensemble classification; and (4) how to parameterize the classifiers and gene selection procedures so as to avoid overfitting. The ultimate goal is to utilize the knowledge gleaned from the above experiments to create a fully automated software platform that creates high-quality, if not optimal, diagnostic models for use in clinical applications. A first incarnation of such a system is introduced as a result of the reported experiments.

## 2 MATERIALS AND METHODS

### 2.1 Support vector machine-based classification methods

Support vector machines (SVMs) (Vapnik, 1998) are arguably the single most important development in supervised classification of recent years. SVMs often achieve superior classification performance compared to other learning algorithms across most domains and tasks; they are fairly insensitive to the curse of dimensionality and are efficient enough to handle very large-scale classification in both sample and variables. In clinical bioinformatics, they have allowed the construction of powerful experimental cancer diagnostic models based on gene expression data with thousands of variables and as little as few dozen samples (e.g. Furey *et al*., 2000; Guyon *et al*., 2002; Aliferis *et al*., 2003a). Moreover, several efficient and high-quality implementations of SVM algorithms (e.g. Joachims, 1999; Chang and Lin, 2003, http://www.csie.ntu.edu.tw/∼cjlin/libsvm) facilitate application of these techniques in practice. The first generation of SVMs could only be applied to binary classification tasks. Yet, most real-life diagnostic tasks are not binary. Moreover, all other things being equal, multicategory classification is significantly harder than binary classification (Mukherjee, 2003). Fortunately, several algorithms have emerged during the last few years that allow multicategory classification with SVMs. The preliminary experimental evidence currently available suggests that some multicategory SVMs (MC-SVMs) perform well in isolated gene expression-based cancer diagnostic experiments (Yeo and Poggio, 2001; Su *et al*., 2001; Ramaswamy *et al*., 2001; Yeang *et al*., 2001; Lee and Lee, 2003).

We outline the principles behind SVM algorithms used in the study below. Full technical descriptions can be found in the references provided in the text. A detailed review of binary SVMs, exact mathematical formulations of both binary and multiclass SVM algorithms, and an illustration of MC-SVMs methods via a solution of example cancer diagnostic problem are presented in Appendices A, B and C, respectively, which are available online (A.Statnikov, C.Aliferis, I.Tsamardinos, D.Hardin and S.Levy, http://www.gems-system.org). In the description of methods below, $k$ is the number of classes or distinct diagnostic categories and $n$ is the number of samples or patients in the training dataset.

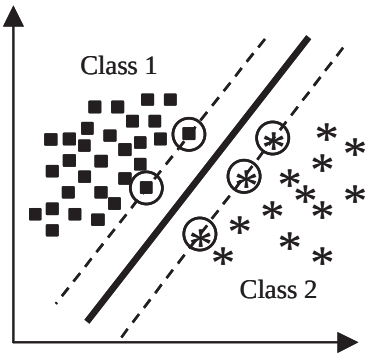


**Fig. 1.** A binary SVM selects a hyperplane (bold line) that maximizes the width of the 'gap' (margin) between the two classes. The hyperplane is specified by 'boundary' training instances, called support vectors shown with circles. New cases are classified according to the side of the hyperplane they fall into.

*2.1.1 Binary SVMs* The main idea of binary SVMs is to implicitly map data to a higher dimensional space via a kernel function and then solve an optimization problem to identify the maximum-margin hyperplane that separates training instances (Vapnik, 1998). The hyperplane is based on a set of boundary training instances, called *support vectors*. New instances are classified according to the side of the hyperplane they fall into (Fig. 1). The optimization problem is most often formulated in a way that allows for non-separable data by penalizing misclassifications.

*2.1.2 Multiclass SVMs: one-versus-rest (OVR)* This is conceptually the simplest multiclass SVM method (for details see Kressel, 1999). Here, we construct $k$ binary SVM classifiers: class 1 (positive) versus all other classes (negative), class 2 versus all other classes, . . ., class $k$ versus all other classes (Fig. 2a). The combined OVR decision function chooses the class of a sample that corresponds to the maximum value of $k$ binary decision functions specified by the furthest 'positive' hyperplane. By doing so, the decision hyperplanes calculated by $k$ SVMs 'shift', which questions the optimality of the multicategory classification.

This approach is computationally expensive, since we need to solve $k$ quadratic programming (QP) optimization problems of size $n$. Moreover, this technique does not currently have theoretical justification such as the analysis of generalization, which is a relevant property of a robust learning algorithm.

*2.1.3 Multiclass SVMs: one-versus-one (OVO)* This method involves the construction of binary SVM classifiers for all pairs of classes; in total there are $\binom{k}{2} = [k(k-1)]/2$ pairs (Fig. 2b and Kressel, 1999). In other words, for every pair of classes, a binary SVM problem is solved (with the underlying optimization problem to maximize the margin between two classes). The decision function assigns an instance to a class that has the largest number of votes, so-called *Max Wins strategy* (Friedman, 1996). If ties still occur, each sample will be assigned a label based on the classification provided by the furthest hyperplane.

One of the benefits of this approach is that for every pair of classes we deal with a much smaller optimization problem, and in total we need to solve $k(k-1)/2$ QP problems of size *smaller than* $n$. Given that QP optimization algorithms used for SVMs are polynomial to the problem size, such a reduction can yield substantial savings in the total computational time. Moreover, some researchers postulate that even if the entire multicategory problem is non-separable, while some of the binary subproblems are separable, then OVO can lead to the improvement of classification compared with OVR (Kressel, 1999). Unlike the OVR approach, here tie-breaking plays only a

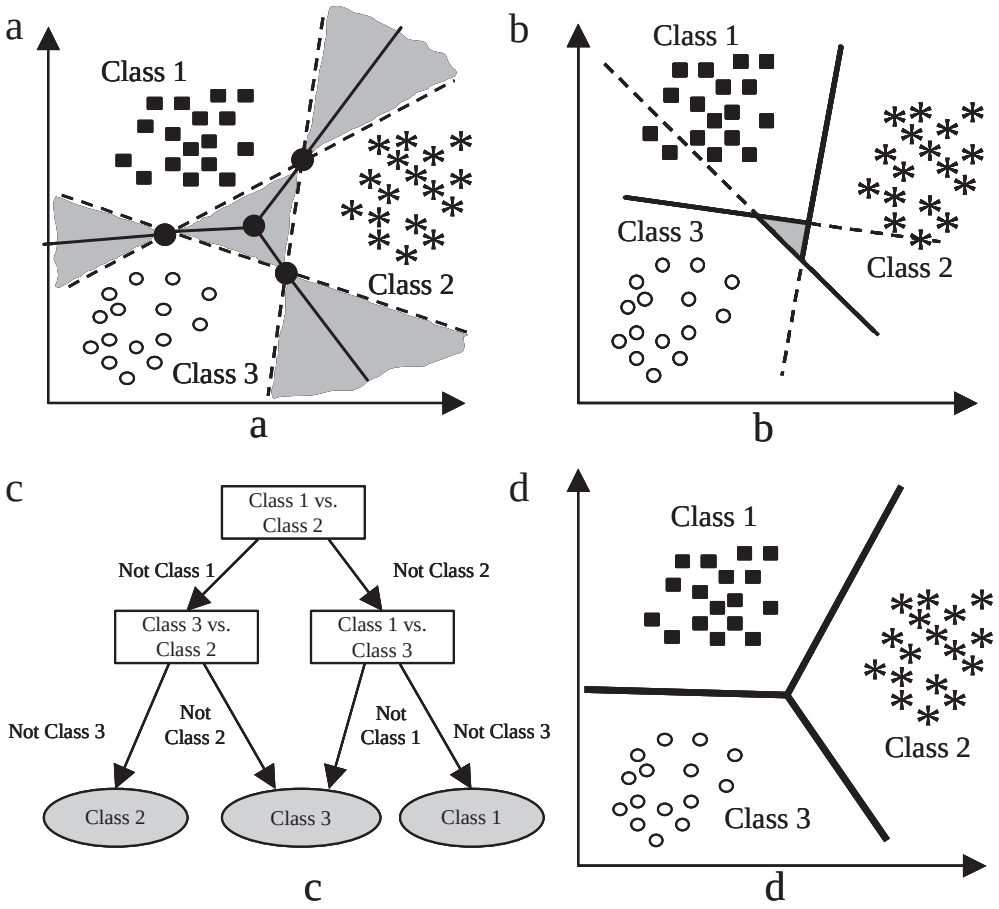


**Fig. 2.** MC-SVM algorithms applied to a three-class diagnostic problem. (**a**) MC-SVM OVR constructs three classifiers: (1) class 1 versus classes 2 and 3; (2) class 2 versus classes 1 and 3; and (3) classes 3 versus classes 1 and 2. (**b**) MC-SVM OVO constructs three classifiers: (1) class 1 versus class 2; (2) class 2 versus class 3; and (3) class 1 versus class 3. (**c**) MC-SVM DAGSVM constructs a decision tree on the basis of OVO SVM classifiers. (**d**) MC-SVM methods by Weston and Watkins and by Crammer and Singer construct a single classifier by maximizing margin between all classes simultaneously.

minor role and does not affect the decision boundaries significantly. On the other hand, similar to OVR, OVO does not currently have established bounds on the generalization error.

*2.1.4 Multiclass SVMs: DAGSVM* The training phase of this algorithm is similar to the OVO approach using multiple binary SVM classifiers; however, the testing phase of DAGSVM requires the construction of a rooted binary decision directed acyclic graph (DDAG) using $\binom{k}{2}$ classifiers (Fig. 2c and Platt *et al*., 2000). Each node of this graph is a binary SVM for a pair of classes, say $(p, q)$. On the topologically lowest level there are $k$ leaves corresponding to $k$ classification decisions. Every non-leaf node $(p, q)$ has two edges—the left edge corresponds to decision 'not $p$' and the right one corresponds to 'not $q$'. The choice of the class order in the DDAG list can be arbitrary as shown empirically in Platt *et al*. (2000).

In addition to inherited advantages from the OVO method, DAGSVM is characterized by a bound on the generalization error.

*2.1.5 Multiclass SVMs: method by Weston and Watkins* (*WW*) This approach to multiclass SVMs is viewed by some researchers as a natural extension of the binary SVM classification problem [Fig. 2d; Hsu and Lin (2002) and Weston and Watkins (1999)]. Here, in the $k$-class case one has to solve a single quadratic optimization problem of size $(k-1)n$ which is identical to binary SVMs for the case $k = 2$. In a slightly different formulation of QP problem, a bounded formulation, decomposition techniques can provide a significant speed-up in the solution of the optimization problem (Hsu and Lin, 2002; Platt, 1999). This method does not have an established bound on the generalization error, and its optimality is not currently proved.

*2.1.6 Multiclass SVMs: method by Crammer and Singer* (*CS*) This technique is similar to WW [Fig. 2d; Hsu and Lin (2002) and Crammer and Singer (2000)]. It requires the solution of a single QP problem of size $(k-1)n$, however uses less slack variables in the constraints of the optimization problem, and hence it is cheaper computationally. Similar to WW, the use of decompositions can provide a significant speed-up in the solution of the optimization problem (Hsu and Lin, 2002). Unfortunately, the optimality of CS, as well as the bounds on generalization has not yet been demonstrated.

## 2.2 Non-SVM classification methods

In addition to five MC-SVM methods, three popular classifiers, $K$-nearest neighbors (KNNs), backpropagation neural networks (NNs) and probabilistic neural networks (PNNs), were also used in this study. These learning methods have been extensively and successfully applied to gene expression-based cancer diagnosis (e.g. Khan *et al*., 2001; Ramaswamy *et al*., 2001; Pomeroy *et al*., 2002; Nutt *et al*., 2003; Singh *et al*., 2002; Berrar *et al*., 2003).

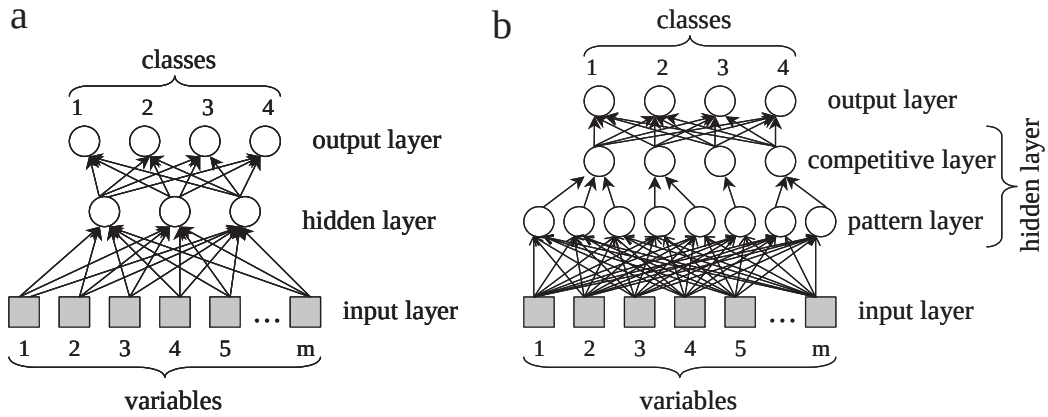


**Fig. 3.** Simplified illustration of the design of neural networks for a four-category diagnostic problem with $m$-dimensional samples of variables (genes) and training set containing $N$ samples. (**a**) Backpropagation neural network contains inputs for $m$ variables (genes); hidden layer with three units (this number is usually determined by cross-validation); and output layer with a unit for each diagnostic category (1-of-$n$ encoding scheme). (**b**) Probabilistic neural network contains inputs for $m$ variables (genes); pattern layer with $N$ units (a unit for each training instance); competitive layer with four units (a unit for each diagnostic category) and receive inputs only from pattern units that are associated with the category to which the training instance belongs; and output layer with a unit for each diagnostic category.

*2.2.1 K-nearest neighbors* The main idea of KNN is that it treats all the samples as points in the $m$-dimensional space (where $m$ is the number of variables) and given an unseen sample $x$, the algorithm classifies it by a vote of $K$-nearest training instances as determined by some distance metric, typically Euclidean distance (Mitchell, 1997).

*2.2.2 Backpropagation neural networks* NNs are feed-forward neural networks with signals propagated only forward through the layers of units. These networks are comprised of (1) an input layer of units, which we feed with gene expression data; (2) hidden layer(s) of units; and (3) an output layer of units, one for each diagnostic category, so-called *1-of-n encoding* (Fig. 3a and Mitchell, 1997). The connections among units have weights and are adjusted during the training phase (epochs of a neural network) by backpropagation learning algorithm. This algorithm adjusts weights by propagating the error between network outputs and true diagnoses backward through the network and employs gradient descent optimization to minimize the error function. This process is repeated until we find a vector of weights that best fits the training data. When training of a neural network is complete, unseen data instances are fed to the input units, propagated forward through the network and the network outputs classifications.

*2.2.3 Probabilistic neural networks* PNNs belong to the family of Radial Basis Function (RBF) neural networks (Mitchell, 1997). RBF networks are feed-forward neural networks with only one hidden layer. The primary difference between an NN with one hidden layer and an RBF network is that for the latter one, the inputs are passed directly to the hidden layer *without weights*. The Gaussian density function is used in a hidden layer as an activation function. The weights for the connections among the hidden and the output layer are optimized via a least squares optimization algorithm. A key advantage of RBF networks is that they are trained much more efficiently than NNs.

PNNs are made up of (1) an input layer; (2) a hidden layer consisting of a pattern layer and a competitive layer; and (3) an output layer [Fig. 3b; Demuth and Beale (2001) and Specht (1990)]. The pattern layer contains one unit for each sample in the training dataset. Given an unseen training sample $x$, each unit in the pattern layer computes a distance from $x$ to a specific training instance and applies a Gaussian density activation function. The competitive layer contains one unit for each diagnostic category, and these units receive inputs only from pattern units that are associated with the category to which the training instance belongs. Each unit in the competitive layer sums over the outputs of the pattern layer and computes a probability of $x$ belonging to a specific diagnostic category. Finally, the output unit corresponding to a maximum of these probabilities outputs 1, while those remaining output 0.

## 2.3 Ensemble classification methods

Given that learners used in this study are different in a sense that they give preference to the different models, the final classification performance may be improved via the use of algorithms that combine outputs of individual classifiers, so-called *ensembles of classifiers*. This idea has received much attention in machine learning literature (e.g. Ho *et al*., 1994; Sharkey, 1996) and has been recently applied to the gene expression domain (Dudoit *et al*., 2002; Valentini *et al*., 2003). Learning how to combine classifiers to further improve the performance is an additional meta-learning problem. Since there is no consensus on the methods, which are the best in ensembling classifiers, we considered a number of techniques: the most common approach by majority voting (Freund, 1995) and more complex approaches such as Decision Trees (DT) (Murthy, 1998) and MC-SVM methods (OVR, OVO and DAGSVM). When algorithms were applied for ensembling of classifiers, the input dataset consisted of attributes corresponding to the outputs of classifiers (either SVM or both SVM and non-SVM algorithms) and the original class labels. Combining classifiers by DT or MC-SVM methods could yield majority voting for some cases, but DT or MC-SVMs allow many more ways to construct ensemble of classifiers.

## 2.4 Parameters for the classification algorithms

Parameters for the classification algorithms were chosen by nested cross-validation procedures to optimize performance while avoiding overfitting as described in the experimental design subsection.

For all five MC-SVM methods we used a polynomial kernel $K(x, y) = (\gamma \cdot x^{\mathrm{T}} y + r)^p$, where $x$ and $y$ are samples with gene expression values and $p$, $\gamma$, $r$ are kernel parameters. We performed classifier optimization over the set of values of cost $C$ (the penalty parameter of SVMs) $= \{0.0001, 0.01, 1, 100\}$ and $p = \{1, 2, 3\}$. The kernel parameters $\gamma$ and $r$ were set to default values as in Chang and Lin (2003): $\gamma$ = 1/number of variables and $r = 0$. For NNs, we performed optimization by implementing early stopping regularization techniques following Goodman and Harrell (2004) http://brain.cs.unr.edu/publications/NevPropManual.pdf on top of the Matlab NN toolbox with parameter selection in a nested cross-validation fashion

**Table 1.** Cancer-related human gene expression datasets used in this study

| Dataset name | Diagnostic task | Number of | | | | Max. prior (%) | Reference |
|---|---|---|---|---|---|---|---|
| | | Samples | Variables (genes) | Categories | Variables/ samples | | |
| *11_Tumors* | 11 various human tumor types | 174 | 12 533 | 11 | 72 | 15.5 | Su *et al*. (2001) |
| *14_Tumors* | 14 various human tumor types and 12 normal tissue types | 308 | **15 009** | 26 | 49 | 9.7 | Ramaswamy *et al*. (2001) |
| *9_Tumors* | 9 various human tumor types | 60 | **5726** | 9 | 95 | 15.0 | Stuanton *et al*. (2001) |
| *Brain_Tumor1* | 5 human brain tumor types | 90 | **5920** | 5 | 66 | 66.7 | Pomeroy *et al*. (2002) |
| *Brain_Tumor2* | 4 malignant glioma types | 50 | **10 367** | 4 | 207 | 30.0 | Nutt *et al*. (2003) |
| *Leukemia1* | Acute myelogenous leukemia (AML), acute lymphoblastic leukemia (ALL) B-cell and ALL T-cell | 72 | **5327** | 3 | 74 | 52.8 | Golub *et al*. (1999) |
| *Leukemia2* | AML, ALL and mixed-lineage leukemia (MLL) | 72 | **11 225** | 3 | 156 | 38.9 | Armstrong *et al*. (2002) |
| *Lung_Cancer* | 4 lung cancer types and normal tissues | 203 | 12 600 | 5 | 62 | 68.5 | Bhattacherjee *et al*. (2001) |
| *SRBCT* | Small, round blue cell tumors (SRBCT) of childhood | 83 | 2308 | 4 | 28 | 34.9 | Khan *et al*. (2001) |
| *Prostate_Tumor* | Prostate tumor and normal tissues | 102 | **10 509** | 2 | 103 | 51.0 | Singh *et al*. (2002) |
| *DLBCL* | Diffuse large B-cell lymphomas (DLBCL) and follicular lymphomas | 77 | **5469** | 2 | 71 | 75.3 | Shipp *et al*. (2002) |

In addition to nine multicategory datasets, two datasets with two diagnoses were included to empirically confirm that the MC-SVM methods behave as well as binary SVMs in binary classification tasks as expected theoretically. The column 'Max. prior' indicates the prior probability of the dominant diagnostic category.

in order to avoid overfitting. In particular, we used feed-forward NN with one hidden layer and the number of units chosen from the set {2,5,10,30,50} based on cross-validation error. We employed gradient descent with adaptive learning rate backpropagation, mean squared error performance goal set to $10^{-8}$ (an arbitrary value very close to zero), fixed momentum of $10^{-3}$ and an optimal number of epochs in the range [100, 10000] based on the early stopping criterion of Goodman and Harrell (2004). For PNNs, we optimized the smoothing factor $\sigma$, a parameter of the Gaussian density function, over 100 different values ranging from 0.01 to 1.00. The parameter $\sigma$ was set the same for all diagnostic categories. Similarly, we performed a thorough optimization of the KNN classifier over all possible numbers of neighbors $K$ ranging from 1 to the total number of instances in the training dataset based on cross-validation error.

## 2.5 Datasets and data preparatory steps

The datasets used in this work are described in Table 1. In addition to nine multicategory datasets, which were most of the multicategory cancer diagnosis datasets in humans found in the public domain at the time when this study was initiated, two binary datasets (i.e. with two diagnoses), *DLBCL* and *Prostate_Tumor*, were also included to empirically confirm that the employed MC-SVM learners behave well in binary classification tasks as expected theoretically.

The studied datasets were produced primarily by oligonucleotide-based technology. Specifically, in all datasets except for *SRBCT*, RNA was hybridized to high-density oligonucleotide Affymetrix arrays HG-U95 or Hu6800, and expression values (average difference units) were computed using Affymetrix GENECHIP analysis software. The *SRBCT* dataset was obtained by using two-color cDNA platform with consecutive image analysis performed by DeArray Software and filtering for a minimal level of expression (Khan *et al*., 2001).

The genes or oligonucleotides with 'absent' calls in all samples were excluded from the analysis to reduce the amount of noise in the datasets (Lu *et al*., 2002; Wouters *et al*., 2003), and if this was the case, the number of genes is listed in bold-face in Table 1. While setting up datasets for experiments, we took advantage of all available documentation in order to increase the number of categories or diagnoses for the outcome variable. For example, the original *Brain_Tumor1* data analysis had only two categories—glioblastomas and anaplastic oligodendrogliomas. Instead of a binary classification problem, we solved a diagnostic problem with four outcomes: classic glioblastomas, non-classic glioblastomas, classic anaplastic oligodendrogliomas and non-classic anaplastic oligodendrogliomas.

In summary, the 11 datasets had 2–26 distinct diagnostic categories, 50–308 samples (patients) and 2308–15 009 variables (genes) after the data preparatory steps outlined above. All datasets are available for download (A.Statnikov, C.Aliferis, I.Tsamardinos, D.Hardin and S.Levy, http://www.gems-system.org).

We note that no new methods to preprocess gene expression data were invented. We relied instead on standard normalization and data preparatory steps performed by the authors of the primary dataset studies. In addition to that, we performed a simple rescaling of gene expression values to be between 0 and 1 for speeding up SVM training. The rescaling was performed based on the training set in order to avoid overfitting.

## 2.6 Experimental designs for model selection and evaluation

Two experimental designs were employed to obtain reliable performance estimates and avoid overfitting. Both experimental designs are based on two loops. The inner loop is used to determine the best parameter of the classifier (i.e. values of parameters yielding the best performance on the validation dataset). The outer loop is used for estimating the performance of the classifier

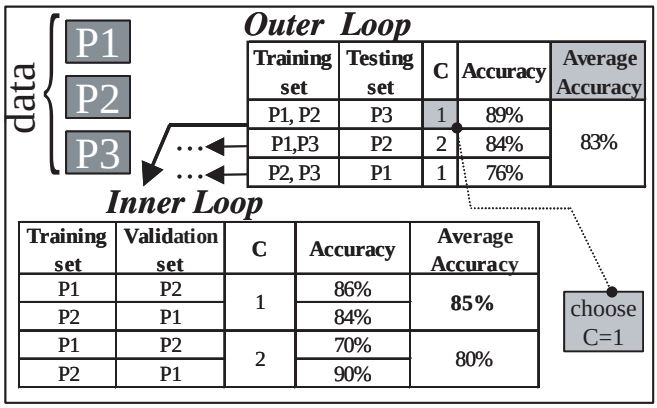


**Fig. 4.** Pictorial simplified example of Design I. Data are split into mutually exclusive sets P1, P2 and P3. The performance is estimated in the outer loop by training on all splits but one, and using the remaining one for testing. The average performance over testing sets is reported. The inner loop is used to determine the optimal value of parameter C (in a cross-validated fashion) for training in the outer loop.

built using the previously found best parameters by testing on *an independent set of patients*. Design I uses a stratified 10-fold cross-validation in the outer loop and a stratified 9-fold cross-validation in the inner loop (Weiss and Kulikowski, 1991). It is often referred to as *nested stratified 10-fold cross-validation*. Fig. 4 shows a simplified pictorial example of a 3-fold Design I applied to three patient groups (P1, P2 and P3) with the optimization of parameter $C$ (which takes values '1' and '2') of some classifier. Note that in reality we do not optimize just one parameter but, rather, a large set of combined parameters. Design II uses leave-one-out cross-validation (LOOCV) in the outer loop and a stratified 10-fold cross-validation in the inner loop. We chose to employ both designs because there exists contradictory evidence in the machine learning literature regarding whether $N$-fold cross-validation provides more accurate performance estimates than LOOCV and vice versa for zero-one loss classification (Kohavi, 1995).

Building the final diagnostic model involves: (1) finding the best parameters for the classification algorithm using a single loop of cross-validation analogously to the inner loop in Designs I and II; (2) building the classifier on all data using the previously found best parameters; and (3) estimating a conservative bound on the classifier's future accuracy by running either Design I or II.

## 2.7 Gene selection

To study how dimensionality reduction can improve the classification performance, we applied all classifiers with subsets of 25, 50, 100, 500 and 1000 top-ranked genes, following the example set by Furey *et al*. (2000). Genes were selected according to four gene selection methods/metrics: (1) ratio of genes between-categories to within-category sums of squares (BW) (Dudoit *et al*., 2002); (2–3) signal-to-noise (S2N) scores (Golub *et al*., 1999) applied in a OVR (S2N-OVR) and in OVO (S2N-OVO) fashion; and (4) Kruskal–Wallis non-parametric one-way ANOVA (KW) (Jones, 1997). The ranking of the genes was performed based on the training set of samples to avoid overfitting.

## 2.8 Performance metrics

We used two classification performance metrics. The first metric is accuracy since we wanted to compare our results with the previously published studies that also used this performance metric. Accuracy is easy to interpret and simplifies statistical testing. On the other hand, accuracy is sensitive to the prior class probabilities and does not fully describe the actual difficulty of the decision problem for highly unbalanced distributions. For example, it is more difficult to achieve an accuracy of 50% for a 26-class dataset *14_Tumors* with prior probability of the major class = 9.7% compared to an accuracy of 75% for a binary dataset *DLBCL* with prior of the major class = 75.3%.

The second metric is relative classifier information (RCI), which corrects for differences in prior probabilities of the diagnostic categories, as well as the number of categories. RCI is an entropy-based measure that quantifies how much the uncertainty of a decision problem is reduced by a classifier relative to classifying using only the priors (Sindwani and *et al*., 2001).

## 2.9 Overall research design

To maintain the feasibility of this study, we pursued a staged factorial design: in Stage I, we conducted a fully factorial design involving datasets and classifiers without gene selection; in Stage II, we focused on the datasets for which the full gene sets yielded poor performance and applied gene selection in a factorial fashion. In addition, we optimized algorithms using accuracy only and limited the possible cardinalities of selected gene sets to only five choices as described in the subsection on gene selection.

Although the above choices restricted the number of models generated, the resulting analyses still generated more than $2.6 \times 10^6$ diagnostic models. The total time required was four single-CPU months using Intel Xeon 2.4 GHz platform. Out of this set of models, only one model was selected for each combination of algorithm and dataset.

Not that, despite the very large number of examined models, the final performance estimates are not overfitted. This is because only one model is selected per split for the estimation of the final performance and it is applied to previously unseen cases. Thus, regardless of how much performance is overestimated in the inner loop (which, in the worst case, may result in not choosing the best possible parameters' combination), the outer loop guarantees proper estimation of the performance.

## 2.10 Statistical comparison among classifiers

To test that differences in accuracy between the best method (i.e. one with the largest average accuracy) and all remaining algorithms are non-random, we need a statistical comparison of observed differences in accuracies.

In machine learning, the major study about the comparison of supervised classification learning algorithms is that of Dietterich (1998) who suggests using $N$-fold cross-validated paired $t$-test for the comparison of $N$-fold accuracy estimates for a single dataset. However, the author clearly admits that this test violates independence and, even more importantly, does not address how this procedure is applied to a multitude of datasets. That is why we decided to use random permutation testing that does not rely on independence assumptions and can be straightforwardly applied to several datasets (Good, 2000). For every algorithm X, other than the best algorithm Y, we performed the following steps. (1) We defined the null hypothesis $H_0$ to be: classification algorithm X is as good as Y, i.e. the accuracy of the best algorithm Y minus the accuracy of algorithm X is zero. (2) We obtained the permutation distribution of $\Delta_{XY}$, the estimator of the true unknown difference between accuracies of the two algorithms, by repeatedly rearranging the outcomes of X and Y at random. (3) We computed the cumulative probability ($P$-value) of $\Delta_{XY}$ being greater than or equal to observed difference $\hat{\Delta}_{XY}$ over 10 000 permutations. If the $P < 0.05$, we rejected $H_0$ and concluded that the data support that algorithm X is not as good as Y in terms of classification accuracy, and this difference is not due to sampling error. To increase the resolution of simulated sampling distribution, we computed a single value of accuracy over all samples from all the datasets. In other words, we treated classifier's predictions from all 11 datasets as if we had one large dataset with samples from all individual datasets.

# 3 IMPLEMENTATIONS

We used the MC-SVM algorithms implemented by the LibSVM team (Chang and Lin, 2003), since they use state-of-the-art optimization methods SMO (Platt, 1999) and TRON (Lin and Moré, 1999) for the solution of MC-SVM problems. The implementation of NN and

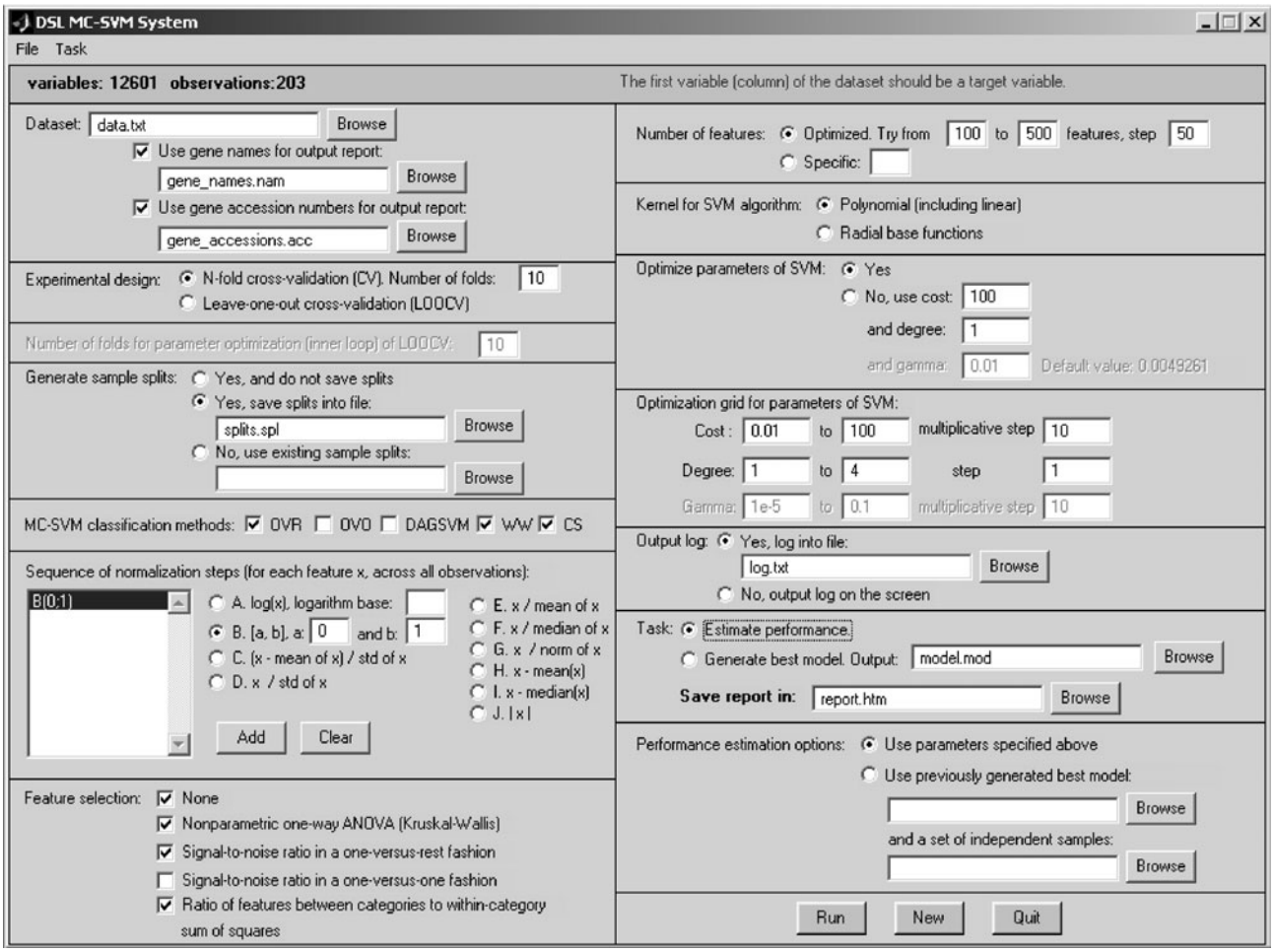


**Fig. 5.** Screenshot of the GEMS system. Many fields are automatically filled out with default values. Most experiments in this study can be replicated using the system with a few clicks of the mouse.

PNN classifiers was based on the Matlab Neural Networks toolbox (Demuth and Beale, 2001). We applied Matlab R13 implementation of the CART algorithm (Murthy, 1998) for DT, and we used our own implementations of KNN, ensemble classification, gene selection as well as statistical comparison algorithms.

The prototype analysis system GEMS (Gene Expression Model Selector) based on the results and analyses reported here, was built using Matlab R13 and MS Visual C++ 6. GEMS has a graphics user interface consisting of a single form (Fig. 5) and is freely available for download (http://www.gems-system.org). The user's manual for the system is provided in Appendix D, which is available online (http://www.gems-system.org).

## 4 RESULTS AND ANALYSES

### 4.1 Classification without gene selection

The performance results of experiments without gene selection obtained using Design I (nested stratified 10-fold cross-validation) with accuracy and RCI as a performance metric are shown in Tables 2 and 3, respectively. The results for Design II are almost identical and are provided only in Appendix E, section 1, which is available online (http://www.gems-system.org). The fact that we obtained similar results with two different experimental designs is evidence in favor of the reliability of performance estimation procedures.

Notably, RCI performance metric revealed different results compared to accuracy. For example, the best RCI for *14_Tumors* dataset is 90.96% and for *Prostate_Tumor* is 71.14%. In contrast, when accuracy was employed, we obtained 76.60% in *14_Tumors* and 92% in *Prostate_Tumor*. The difference can be explained by the difficulties of the classification problems—*14_Tumors* is much harder (it has 26 classes with prior of the most frequent class 9.7%; Table 1) than *Prostate_Tumor* (it is a binary problem with prior 51%; Table 1).

According to Table 2, in 8 out of 11 datasets, MC-SVMs perform cancer diagnoses with accuracies >90%. The results for RCI performance metric are similar (Table 3): in 7 out of 11 datasets, MC-SVMs yield diagnostic performance with RCI > 90%. Overall, all MC-SVMs outperform KNN, NN and PNN significantly. The only exception is KNN and PNN applied to *14_Tumors* dataset which outperformed OVO and DAGSVM, but still were unable to perform better than more robust MC-SVM techniques, OVR, WW and CS. The superior classification performance of the SVM-based methods

**Table 2.** Performance results (accuracies) without gene selection obtained using a nested stratified 10-fold cross-validation design (Design I)

| Methods | Multicategory classification (%) | | | | | | | | | Binary classification (%) | | Averages (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 9_Tumors | 11_Tumors | 14_Tumors | Brain_Tumor1 | Brain_Tumor2 | Leukemia1 | Leukemia2 | Lung_Cancer | SRBCT | Prostate_Tumor | DLBCL | |
| MC-SVM | | | | | | | | | | | | |
| OVR | 65.10 | 94.68 | 74.98 | **91.67** | 77.00 | **97.50** | **97.32** | 96.05 | **100.00** | **92.00** | **97.50** | **89.44** |
| OVO | 58.57 | 90.36 | 47.07 | 90.56 | **77.83** | 97.32 | 95.89 | 95.59 | **100.00** | **92.00** | **97.50** | 85.70 |
| DAGSVM | 60.24 | 90.36 | 47.35 | 90.56 | **77.83** | 96.07 | 95.89 | 95.59 | **100.00** | **92.00** | **97.50** | 85.76 |
| WW | 62.24 | 94.68 | 69.07 | 90.56 | 73.33 | **97.50** | 95.89 | 95.55 | **100.00** | **92.00** | **97.50** | 88.03 |
| CS | **65.33** | **95.30** | **76.60** | 90.56 | 72.83 | **97.50** | 95.89 | **96.55** | **100.00** | **92.00** | **97.50** | 89.10 |
| Non-SVM | | | | | | | | | | | | |
| KNN | 43.90 | 78.51 | 50.40 | 87.94 | 68.67 | 83.57 | 87.14 | 89.64 | 86.90 | 85.09 | 86.96 | 77.16 |
| NN | 19.38 | 54.14 | 11.12 | 84.72 | 60.33 | 76.61 | 91.03 | 87.80 | 91.03 | 79.18 | 89.64 | 67.73 |
| PNN | 34.00 | 77.21 | 49.09 | 79.61 | 62.83 | 85.00 | 83.21 | 85.66 | 79.50 | 79.18 | 80.89 | 72.38 |

These results are further improved by gene selection (Fig. 6). The last column in the bottom table reports average performance computed over datasets. Numbers in bold correspond to the best classifications for each dataset.

**Table 3.** Performance results (RCI) without gene selection obtained using a nested stratified 10-fold cross-validation design (Design I)

| Methods | Multicategory classification (%) | | | | | | | | | Binary classification (%) | | Averages (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 9_Tumors | 11_Tumors | 14_Tumors | Brain_Tumor1 | Brain_Tumor2 | Leukemia1 | Leukemia2 | Lung_Cancer | SRBCT | Prostate_Tumor | DLBCL | |
| MC-SVM | | | | | | | | | | | | |
| OVR | 77.00 | 95.80 | 90.53 | **82.31** | 77.49 | **93.90** | **94.42** | 89.45 | **100.00** | **71.14** | **90.91** | **87.54** |
| OVO | 78.24 | 92.24 | 64.99 | 80.77 | **80.27** | 93.05 | 92.35 | 87.95 | **100.00** | **71.14** | **90.91** | 84.72 |
| DAGSVM | **78.67** | 92.24 | 65.64 | 80.77 | **80.27** | 90.16 | 92.35 | 87.95 | **100.00** | **71.14** | **90.91** | 84.55 |
| WW | 76.22 | 95.80 | 86.30 | 80.77 | 74.75 | **93.90** | 91.90 | 87.46 | **100.00** | **71.14** | **90.91** | 86.29 |
| CS | 77.25 | **96.20** | **90.96** | 80.77 | 74.44 | **93.90** | 91.90 | 91.40 | **100.00** | **71.14** | **90.91** | 87.17 |
| Non-SVM | | | | | | | | | | | | |
| KNN | 63.38 | 83.93 | 82.73 | 67.86 | 64.48 | 64.45 | 76.95 | 68.48 | 80.71 | 51.09 | 63.08 | 69.74 |
| NN | 65.57 | 67.80 | 16.24 | 61.42 | 62.49 | 53.06 | 78.02 | 64.97 | 87.50 | 33.25 | 58.36 | 58.97 |
| PNN | 55.59 | 81.39 | 81.40 | 43.86 | 61.73 | 68.85 | 73.51 | 59.72 | 68.92 | 39.22 | 38.23 | 61.13 |

These results are further improved by gene selection (Fig. 7). The last column in the bottom table reports average performance computed over datasets. Numbers in bold correspond to the best classifications for each dataset.

compared to KNN, NN and PNN reflects that SVMs are less sensitive to the curse of dimensionality and more robust to a small number of high-dimensional gene expression samples than other non-SVM techniques (Aliferis *et al*., 2003b). A more detailed explanation of this matter follows in the next section.

Among MC-SVMs, OVR, WW and CS yield the best results and are not statistically significant from each other at the 0.05 level (Appendix E, section 2). On the other hand, OVO, DAGSVM, KNN, PNN and NN have poorer performance than the above methods to a statistically significant degree. OVO and DAGSVM perform very similar, which is due to the fact that both MC-SVM methods use the same binary SVM classifiers. We conjecture that OVO and DAGSVM perform worse than other MC-SVM methods because both algorithms are based on one-versus-one binary classifiers that use only a fraction of total training samples at a time (samples that belong to two classes) and ignore information about the distribution of the remaining examples that may be significant for the classification. In case of large sample sizes, we expect MC-SVMs OVO and DAGSVM to perform as good as WW, CS and OVR (e.g. see Hsu and Lin, 2002).

According to Tables 2 and 3 and the results of application of the binary SVM implementation SVMLight (Joachims, 1999) to *DLBCL* and *Prostate_Tumor* datasets (data not shown), we conclude that employed implementations of MC-SVM algorithms perform the same classifications as binary SVMs and, hence, handle binary diagnostic problems appropriately as expected.

We tried to explain the classification performance of the best MC-SVM algorithms OVR, WW and CS by fitting inverse power curves motivated by the ideas described previously (Cortes *et al*., 1993). We found that in high-dimensional spaces of microarray gene expression data, the number of samples divided by the product of the number of variables times the number of categories explains observed classification accuracies in the datasets. When we reduced dimensionality by gene selection, or employed RCI performance metric, or used other classification algorithms, this behavior disappeared. More details can be found in Appendix E, section 3. It is important to note that curve fitting procedure used in this study is very simplistic since it does not incorporate predictors describing degree of biological difficulty and assumes that datasets and learning tasks used in this study are representative.

Finally, we also analyzed execution time for all learning algorithms applied without gene selection (Table 4). The fastest MC-SVM methods CS and WW took 7.95 and 7.88 h for Design I and 289.01 and 290.77 h for Design II, respectively. The slowest MC-SVM technique OVR completed within 19.28 h for Design I and 772.43 h for Design II. This technique is slowest among the MC-SVM algorithms since it constructs several classifiers repeatedly employing all samples from the training dataset. The fastest overall algorithm KNN took 3.40 h for Design I and 109.60 h for Design II, while the slowest overall algorithms NN and PNN took 195.68 h and 186.19 h, respectively, for Design I. All experiments were executed in the Matlab R13 environment on eight Intel Xeon 2.4 GHz dual-CPU workstations connected in a cluster.

**Table 4.** Total time of classification experiments without gene selection for all 11 datasets and two experimental designs

| Methods | Time (h) Design I | Design II |
|---|---|---|
| MC-SVM | | |
| OVR | 19.28 | 772.43 |
| OVO | 9.86 | 388.11 |
| DAGSVM | 9.93 | 390.97 |
| WW | 7.95 | 290.77 |
| CS | 7.88 | 289.01 |
| Non-SVM | | |
| KNN | 3.40 | 109.60 |
| NN | 195.68 | N/A |
| PNN | 186.19 | N/A |

## 4.2 Classification with gene selection

The summary of application of the four gene selection methods, BW, S2N-OVR, S2N-OVO and KW, to the four most 'challenging' datasets, *9_Tumors*, *14_Tumors*, *Brain_Tumor1* and *Brain_Tumor2*, using accuracies and RCI as a performance metric is presented in Figures 6 and 7, respectively. It should be noted that a more rigorous way to do gene selection with the validation of number of genes and gene selection method is implemented in GEMS software system and may be very expensive computationally (that is why it was not pursued here as explained in the Materials and methods section).

The results show that gene selection significantly improves the classification performance of non-SVM learners. In particular, for some datasets, accuracy is improved by up to 14.97, 59.78 and 22.67% and RCI is improved by up to 19.52, 69.95 and 34.98% for KNN, NN and PNN, respectively. Gene selection also improves the accuracy of MC-SVMs up to 9.53% and, hence, improves the accuracy of the overall best classifier. Although KNN, NN and PNN perform closer to MC-SVMs, three MC-SVM algorithms, OVR, WW and CS, still outperform non-SVM methods in most of the cases. We also found that these three MC-SVM methods are not statistically significant from each other and NN at the 0.05 level (Appendix E, section 2). The remaining algorithms, MC-SVMs OVO and DAGSVM, KNN and PNN, have statistically significant poorer performance. Finally, none of the four gene selection methods performs significantly better than the other ones.

As we have empirically found, the non-SVM methods KNN, PNN and NN benefit significantly more than MC-SVMs from gene selection. A number of observations can explain this behavior: in high-dimensional spaces, KNN has high variance of the prediction since all the training points are located close to the edge of the sample (Hastie *et al*., 2001). Furthermore, many irrelevant variables in the data dominate distances between samples which presents a significant problem for the prediction (Mitchell, 1997). PNN encounter problems similar to KNN, in particular because they rely on Parzen windows for density estimation that generally require exponential sample to the data dimensionality (Duda *et al*., 2001). NNs are sensitive to high dimensionality for at least two reasons: first, note the larger the number of variables, the larger is the number of weights in this type of neural network. Because of this, (1) there may be more local minima in the error landscape and it is thus more probable for backpropagation to get 'trapped' in one of them, and (2) the model space becomes exponentially larger with the addition of each weight, and therefore, it becomes harder to identify a model that generalizes. In comparison, the family of SVMs allows for effective optimization search procedure by utilizing convex formulation with

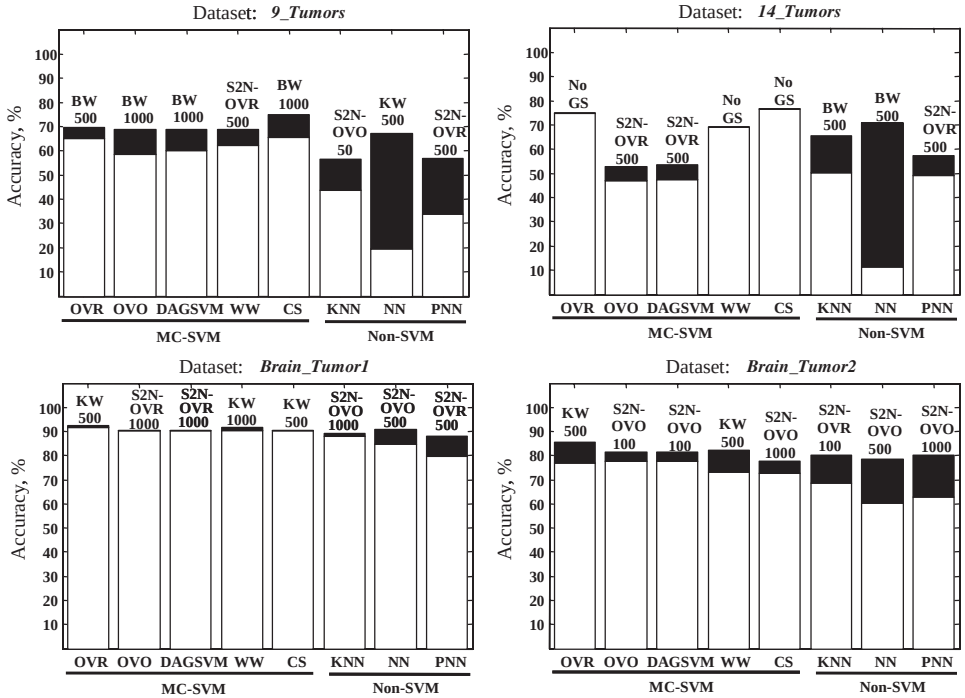


**Fig. 6.** Performance results (accuracies) of the classification experiments with gene selection obtained using a nested stratified 10-fold cross-validation design (Design I) for four datasets: *9_Tumors*, *14_Tumors*, *Brain_Tumor1* and *Brain_Tumor2*. The white bars correspond to the classification results without gene selection. The black bars show improvement of the results by gene selection. The text above each bar indicates the optimal combination of gene selection method and the number of genes for a specific classifier. The abbreviation 'No GS' stands for 'No gene selection'.

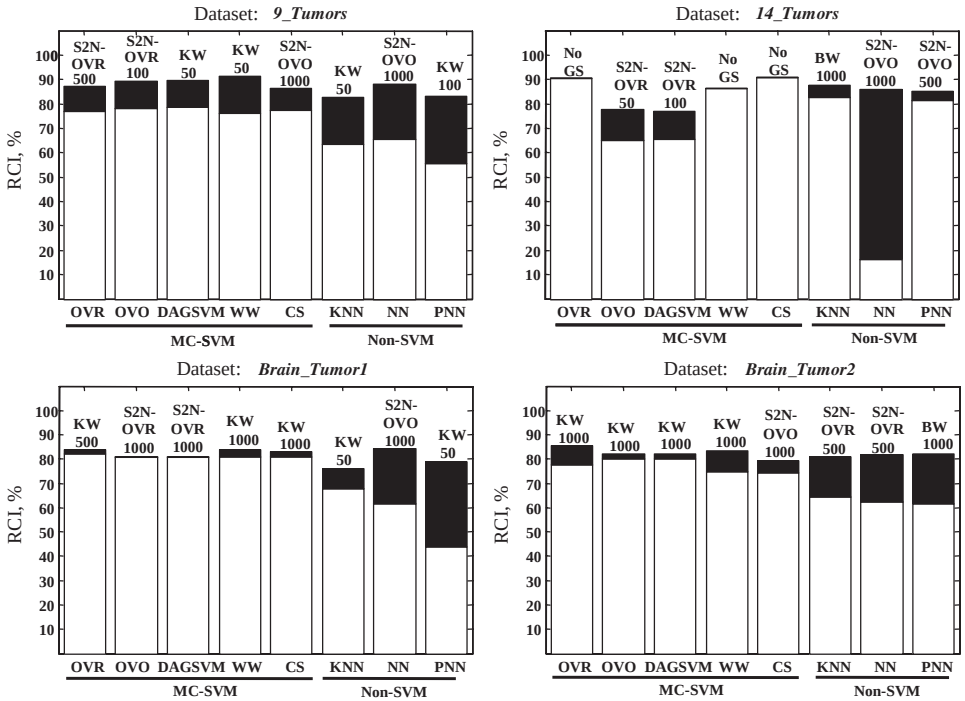


**Fig. 7.** Performance results (*RCI*) of the classification experiments with gene selection obtained using a nested stratified 10-fold cross-validation design (Design I) for four datasets: *9_Tumors*, *14_Tumors*, *Brain_Tumor1* and *Brain_Tumor2*. The white bars correspond to the classification results without gene selection. The black bars show improvement of the results by gene selection. The text above each bar indicates the optimal combination of gene selection method and the number of genes for a specific classifier. The abbreviation 'No GS' stands for 'No gene selection'.

a single optimum justified by Statistical Learning Theory (Vapnik, 1998). Furthermore, SVMs seem relatively insensitive to the curse of dimensionality, possibly due to the specific regularization mechanism they employ. In particular, this is reflected by the following: (1) many established generalization bounds do not depend on the data dimensionality (Herbrich, 2002) and (2) even linear SVMs assign zero weights to irrelevant variables (Hardin *et al*., 2004). On the other hand, the SVM algorithm may assign non-zero weights to weakly relevant variables (Hardin *et al*., 2004), which explains why effective variable selection can still improve the SVM classification.

### 4.3 Ensemble classification

For the case when no gene selection was performed, ensembles do not outperform the best non-ensemble methods with the exception of the DT ensemble classifier for *Brain_Tumor2* dataset, which improves the classification accuracy by 1.67%. Other ensembles often achieve similar performance to the best non-ensemble methods (Appendix E, section 4).

Next, we considered three datasets, *9_Tumors*, *Brain_ Tumor1* and *Brain_Tumor2*, where we previously observed an improvement in the classification performance by gene selection. For each dataset, we selected a subset of genes yielding the best classification performance (over all gene selection methods, subsets of genes and learning algorithms) and constructed combined classifiers. According to the results, ensembles perform worse than the best non-ensemble models (Appendix E, section 4).

We believe that in our study, ensemble classifiers did not improve the final classification performance for the following two reasons: first, samples misclassified by non-SVM algorithms are almost always a strict superset of samples misclassified by MC-SVM algorithms. Second, SVM algorithms are fairly stable in a sense that small changes in the training data do not result in large changes in the predictive model's behavior (Kutin and Niyogi, 2002), and according to Dudoit *et al*. (2002) stable algorithms do not usually tend to benefit from the ensemble classification.

### 4.4 Comparison with previously published results

Most of the results from this study are not exactly comparable with the analyses provided in the original studies due to differences in the setup of dataset/learning task, experimental design, gene selection, classifiers and so on that vary from study to study. However, the reported results in the literature confirm that MC-SVMs as applied here perform equally as well, or even better, compared to previously published models on the same datasets (Appendix E, section 5).

## 5 DISCUSSION AND LIMITATIONS

One of the limitations of the present study is that we use accuracy and RCI as our performance measures. These metrics do not incorporate information about confidence of the predictions as well as different misclassification costs of diagnostic categories. On the other hand, accuracy was used in the published studies and it is easy to interpret and simplifies statistical comparison, while RCI is insensitive to prior class probabilities and accounts for the difficulty of the learning problem. There are currently no mature performance metrics applicable for multiclass domains and suitable for our classifiers with both confidence information and consideration of misclassification costs. Initial attempts were introduced by Lee and Lee (2003), Mossman (1999) and Ferri *et al*. (2003); however, much needs to be done before we obtain a workable metric for experiments such as those presented here.

As we mentioned above, the choice of KNN, NN and PNN classifiers as the baseline techniques was grounded on prior successful applications to gene expression-based cancer diagnosis (e.g. Khan *et al*., 2001; Ramaswamy *et al*., 2001; Pomeroy *et al*., 2002; Nutt *et al*., 2003; Singh *et al*., 2002; Berrar *et al*., 2003). We have also experimented with other non-SVM classifiers, such as DT (Murthy, 1998) and Weighed Voting (WV) classifiers applied both in OVR and OVO fashion (Golub *et al*., 1999; Ramaswamy *et al*., 2001; Yeang *et al*., 2001). We found that both with and without gene selection, DT perform significantly worse than MC-SVMs, worse than KNN, and similarly or worse than NN and PNN. Likewise, WV classifiers are significantly outperformed by MC-SVMs, KNN, NN and PNN. More details about these additional experiments with DT and WV classifiers can be found in Appendix E, section 6.

A particularly interesting direction for future research is to improve our existing gene selection procedures with the selection of 'optimal' number of genes by cross-validation[1]. Furthermore, we are interested in applying various multivariate Markov blanket and local neighborhood algorithms that have been previously successfully applied to cancer gene expression and several other domains and do guarantee efficient identification of a set of relevant attributes under fairly broad assumptions (Aliferis *et al*., 2003c; Tsamardinos *et al*., 2003).

We plan to extend our comparative analyses with new MC-SVM methods as they become available. In particular, we plan to use an MSVM[2] (for details, see Lee and Lee, 2003), which has promising theoretical properties. Namely, an MSVM employs a certain loss function for the multicategory classification problem under which the solution to the multicategory problem resembles Bayes rule asymptotically. Moreover, this framework easily allows accounting for unequal misclassification and distortion of class proportions. In addition to new MC-SVM algorithms, a promising set of new methods is error correcting output codes for the solution of multiclass problems by reducing them to binary problems[3] that can be solved using binary SVMs. This approach is very promising in that some researchers proved a general empirical multiclass error loss bound given empirical loss of individual binary classifiers (Allwein *et al*., 2000).

The emergence of new cancer gene expression datasets in our institution and elsewhere will allow us to conduct a prospective evaluation of the GEMS system to study its ability to facilitate creation of powerful diagnosis models. We are also working on augmenting the preliminary version of the system with wizard-like graphics user interface that will make GEMS usable by researchers with limited expertise in data analysis.

To the best of our knowledge, currently there exists only one work aimed at the evaluation of MC-SVM algorithms (Hsu and Lin, 2002). This study is outside the realm of biomedicine since Hsu and

[1]The functionality to cross-validate number of genes is already implemented in the software system.

[2]Although the codes of MSVM were kindly provided by its author, the implementation of this algorithm cannot currently handle problems with very large number of categories and/or large sample size. Given the excellent theoretical properties of the MSVM, we hope that this issue will be solved in the near future so this algorithm can be applied to the present problem domain.

[3]In the present study, we have already employed three approaches to reduce multiclass problems to binary—OVR, OVO and DAGSVM.

Lin (2002) considered such classification tasks as wine recognition, letter recognition, shuttle control and so on with the number of variables ranging from 4 to 180 and sample sizes greater than 500 in the majority of tasks, which is not typical for microarray cancer gene expression datasets. However, it is worthwhile to mention the major conclusions of that evaluation. The authors empirically found the following: (1) using a Gaussian radial basis kernel, all MC-SVM methods perform similarly; (2) DAGSVM and OVO have the fastest training time; and (3) for problems with large sample size, WW and CS yield fewer support vectors compared to OVR, OVO and DAGSVM. The work by Hsu is complementary to ours and is not overlapping due to significant differences in the problem domain and dataset characteristics. For example, in our experiments, MC-SVM methods OVO and DAGSVM achieved inferior classification performance compared to other MC-SVM algorithms.

Finally, two recent bioinformatics studies have also performed comparative analyses of multicategory classification algorithms in cancer gene expression domain (Berrar *et al*., 2003; Romualdi *et al*., 2003). Unfortunately, neither study optimized parameters of the classifiers for all datasets, which is likely to result in suboptimal application of the learning methods. Therefore, no study can convincingly answer the central question of this research—what is the best learning algorithm for multicategory cancer diagnosis based on gene expression data?

## 6 CONCLUSIONS

The contributions of the present study are two-fold. The first contribution is that we conducted the most comprehensive systematic evaluation to date of multicategory diagnosis algorithms applied to the majority of multicategory cancer-related gene expression human datasets publicly available. Based on the results of this evaluation, the following conclusions can be drawn:

- MSVMs are the best family of algorithms for these type of data and medical tasks. They outperform other popular non-SVM machine learning techniques by a large margin.
- Among MC-SVM methods, the ones by Crammer and Singer, Weston and Watkins and OVR have superior classification performance.
- The performance of both MC-SVM and non-SVM methods can be moderately (for MC-SVMs) or significantly (for non-SVM) improved by gene selection.
- Ensemble classification does not further improve the classification performance of the best MC-SVM models.

We believe that practitioners and software developers should take note of these results when considering construction of decision support systems in this domain, or when selecting algorithms for inclusion in related analysis software.

The second contribution is that we created the fully automated software system GEMS that automates the experimental procedures described in this paper to (1) develop optimal classification models for the domain of cancer diagnosis with microarray gene expression data and (2) estimate their performance in future patients. The results obtained by the system in a labor-efficient manner appear to be on par with or better than previously published results in the literature on the same datasets. Although several commercial and academic software tools exist for gene expression classification (e.g. Reich *et al*., 2004), to the best of our knowledge GEMS treats the task in the most comprehensive manner and is the first such system to be informed after a rigorous analysis of the available algorithms and datasets. We hope that the methodology presented in the present paper may encourage similar principled treatment of other software development efforts in clinical bioinformatics. The system is freely available for download (http://www.gems-system.org) for non-commercial use and its users manual is provided in Appendix D.

## ACKNOWLEDGEMENTS

This research was supported by NIH grants RO1 LM007948-01 and P20 LM 007613-01.

## REFERENCES

Aliferis,C.F., Tsamardinos,I., Massion,P., Statnikov,A., Fananapazir,N. and Hardin,D. (2003a) Machine learning models for classification of lung cancer and selection of genomic markers using array gene expression data. In *Proceedings of the 16th International Florida Artificial Intelligence Research Society (FLAIRS) Conference*, St Augustine, FL, May 12–14. AAAI Press, pp. 67–71.

Aliferis,C.F., Tsamardinos,I., Massion,P., Statnikov,A. and Hardin,D. (2003b) Why classification models using array gene expression data perform so well: a preliminary investigation of explanatory factors. In *Proceedings of the 2003 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS)*, Las Vegas, NV, June 23–26. CSREA Press.

Aliferis,C.F., Tsamardinos,I. and Statnikov,A. (2003c) HITON, a novel Markov blanket algorithm for optimal variable selection. In *Proceedings of the 2003 American Medical Informatics Association (AMIA) Annual Symposium*, Washington, DC, November 8–12, pp. 21–25.

Allwein,E.L., Schapire,R.E. and Singer,Y. (2000) Reducing multiclass to binary: a unifying approach for margin classifiers. *J. Mach. Learn. Res.*, **1**, 113–141.

Armstrong,S.A., Staunton,J.E., Silverman,L.B., Pieters,R., den Boer,M.L., Minden,M.D., Sallan,S.E., Lander,E.S., Golub,T.R. and Korsmeyer,S.J. (2002) MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat. Genet.*, **30**, 41–47.

Berrar,D., *et al*. (2003) Multiclass cancer classification using gene expression profiling and probabilistic neural networks. In *Proceedings of the Pacific Symposium on Biocomputing (PSB)*, Lihue, Hawaii, January 3–7.

Bhattacherjee,A., Richards,W.G., Staunton,J., Li,C., Monti,S., Vasa,P., Ladd,C., Beheshti,J., Bueno,R., Gillete,M. *et al*. (2001) Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl Acad. Sci. USA*, **98**, 13790–13795.

Chang,C.-C. and Lin,C.-J. (2003) LIBSVM: a library for support vector machines.

Cortes,C., Jackel,L.D., Solla,S.A., Vapnik,V. and Denker,J.S. (1993) Learning curves: asymptotic values and rate of convergence. In *Advances in Neural Information Processing Systems (NIPS)*, Denver, CO, pp. 327–334.

Crammer,K. and Singer,Y. (2000) On the learnability and design of output codes for multiclass problems. *Proceedings of the Thirteen Annual Conference on Computational Learning Theory (COLT 2000)*, Standford University, Palo Alto, CA, June 28–July 1.

Demuth,H. and Beale,M. (2001) Neural network toolbox user's guide. Matlab user's guide. The MathWorks Inc., Natick, MA.

Dietterich,T.G. (1998) Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.*, **10**, 1895–1923.

Duda,R.O., Hart,P.E. and Stork,D.G. (2001) *Pattern Classification*, 2nd edn. John Wiley, NY.

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

Ferri,C., Hernández-Orallo,J. and Salido,M.A. (2003) Volume under the ROC surface for multi-class problems. In *Proceedings of the 14th European Conference on Machine Learning (ECML'03)*, Cavtat-Dubrovnik, Croatia, September 22–26, LNAI 2837. Springer-Verlag, pp. 108–120.

Fortina,P., Surrey,S. and Kricka,L.J. (2002) Molecular diagnostics: hurdles for clinical implementation. *Trends Mol. Med.*, **8**, 264–266.

Freund,Y. (1995) Boosting a weak learning algorithm by majority. *Inform. Comput.*, **121**, 256–285.

Friedman,J. (1996) Another approach to polychotomous classification. *Technical Report*, Stanford University, CA.

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

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

Good,P.I. (2000) *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*, 2nd edn. Springer-Verlag, NY.

Goodman,P.H. and Harrell,F.E. (2004) NevProp Manual with Introduction to Artificial Neural Networks Theory.

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

Guyon,I., Weston,J., Barnhill,S. and Vapnik,V. (2003) Erratum: gene selection for cancer classification using support vector machines.

Hardin,D., Tsamardinos,I. and Aliferis,C.F. (2004) A theoretical characterization of linear SVM-based feature selection. In *Twenty-First International Conference on Machine Learning (ICML 2004)*, Banff, Alberta, Canada, July 4–8.

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

Herbrich,R. (2002) *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, Cambridge, MA, USA.

Ho,T.K., Hull,J.J. and Srihari,S.N. (1994) Decision combination in multiple classifier systems. *IEEE Trans. Pattern Anal. Machine Intell.*, **16**, 66–76.

Hsu,C.-W. and Lin,C.-J. (2002) A comparison of methods for multi-class support vector machines. *IEEE Trans. Neural Netw.*, **13**, 415–425.

Joachims,T. (1999) Making large-scale SVM learning practical. In Schölkopf,B., Burges,C. and Smola,A. (ed.), *Advances in Kernel Methods—Support Vector Learning*. MIT Press, Cambridge, MA, USA.

Jones,B. (1997) Matlab Statistics Toolbox. The MathWorks, Inc., Natick, MA.

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

Kohavi,R. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI 1995)*, Montreal, Quebec, Canada, August 20–25. Morgan Kaufmann Publishers, pp. 1137–1145.

Kressel,U. (1999) Pairwise classification and support vector machines. In *Advances in Kernel Methods: Support Vector Learning*, (Chapter 15.) MIT Press, Cambridge, MA, USA.

Kutin,S., and Niyogi,P. (2002) Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-2002)*, University of Alberta, Edmonton, Canada, August 1–4. Morgan Kaufmann Publishers, pp. 275–282.

Lee,Y. and Lee,C.-K. (2003) Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*, **19**, 1132–1139.

Lin,C.-J. and Moré,J.J. (1999) Newton's method for large bound-constrained optimization problems. *SIAM J. Optimization*, **9**, 1100–1127.

Lu,J., Hardy,S., Tao,W.L., Muse,S., Weir,B. and Spruill,S. (2002) Classical statistical approaches to molecular classification of cancer from gene expression profiling. In Lin,S.M. and Johnson,K.F. (eds), *Methods of Microarray Data Analysis: Papers from CAMDA'00*. Kluwer Academic Publishers, Boston/Dordrecht/London, pp. 97–107.

Mitchell,T.M. (1997) *Machine Learning*. McGraw-Hill, New York, NY, USA.

Mossman,D. (1999) Three-way ROCs. *Med. Decis. Making*, **19**, 78–89.

Mukherjee,S. (2003) *Classifying Microarray Data Using Support Vector Machines, Understanding And Using Microarray Analysis Techniques: A Practical Guide*. Kluwer Academic Publishers, Boston, MA.

Murthy,S.K. (1998) Automatic construction of decision trees from data: a multi-disciplinary survey. *Data Min. Knowl. Discov.*, **2**, 345–389.

Ntzani,E.E. and Ioannidis,J.P. (2003) Predictive ability of DNA microarrays for cancer outcomes and correlates: and empirical assessment. *Lancet*, **362**, 1439–1444.

Nutt,C.L., Mani,D.R., Betensky,R.A., Tamayo,P., Cairncross,J.G., Ladd,C., Pohl,U., Hartmann,C., McLaughlin,M.E., Batchelor,T.T. *et al*. (2003) Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer Res.*, **63**, 1602–1607.

Platt,J. (1999) Fast training of support vector machines using sequential minimal optimization. In Schölkopf,B., Burges,C. and Smola,A. (ed.), *Advances in Kernel Methods—Support Vector Learning*. MIT Press, Cambridge, MA, USA.

Platt,J.C., Cristianini,N. and Shawe-Taylor,J. (2000) Large margin DAGS for multiclass classification. In *Advances in Neural Information Processing Systems 12*. MIT Press, pp. 547–553.

Pomeroy,S.L., Tamayo,P., Gaasenbeek,M., Sturla,L.M., Angelo,M., McLaughlin,M.E., Kim,J.Y., Goumnerova,L.C., Black,P.M., Lau,C. *et al*. (2002) Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, **415**, 436–442.

Ramaswamy,S., Tamayo,P., Rifkin,R., Mukherjee,S., Yeang,C.H., Angelo,M., Ladd,C., Reich,M., Latulippe,E., Mesirov,J.P. *et al*. (2001) Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl Acad. Sci. USA*, **98**, 15149–15154.

Reich,M., Ohm,K., Angelo,M., Tamayo,P., Mesirov,J.P. (2004) GeneCluster 2.0: an advanced toolset for bioarray analysis. *Bioinformatics*, **20**, 1797–1798.

Reunanen,J. (2003) Overfitting in making comparisons between variable selection methods. *J. Machine Learn. Res.*, **3**, 1371–1382.

Romualdi,C., Campanaro,S., Campagna,D., Celegato,B., Cannata,N., Toppo,S., Valle,G. and Lanfranchi,G. (2003) Pattern recognition in gene expression profiling using DNA array: a comparative study of different statistical methods applied to cancer classification. *Hum. Mol. Genet.*, **12**, 823–836.

Schwarzer,G. and Vach,W. (2000) On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology. *Stat. Med.*, **19**, 541–561.

Sindwani,V., *et al*. (2001) Information theoretic feature crediting in multiclass support vector machines. *First SIAM International Conference on Data Mining (ICDM'01)*, Chicago, IL, April 5–7.

Singh,D., Febbo,P.G., Ross,K., Jackson,D.G., Manola,J., Ladd,C., Tamayo,P., Renshaw,A.A., D'Amico,A.V., Richie,J.P. *et al*. (2002) Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 203–209.

Sharkey,A.J.C. (1996) On combining artificial neural net. *Connection Sci.*, **8**, 299–314.

Shipp,M.A., Ross,K.N., Tamayo,P., Weng,A.P., Kutok,J.L., Aguiar,R.C., Gaasenbeek,M., Angelo,M., Reich,M., Pinkus,G.S. *et al*. (2002) Diffuse large B-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning. *Nat. Med.*, **8**, 68–74.

Specht,D.F. (1990) Probabilistic neural network. *Neural Networks*, **3**, 109–118.

Staunton,J.E., Slonim,D.K., Coller,H.A., Tamayo,P., Angelo,M.J., Park,J., Scherf,U., Lee,J.K., Reinhold,W.O., Weinstein,J.N. *et al*. (2001) Chemosensitivity prediction by transcriptional profiling. *Proc. Natl Acad. Sci. USA*, **98**, 10787–10792.

Su,A.I., Welsh,J.B., Sapinoso,L.M., Kern,S.G., Dimitrov,P., Lapp,H., Schultz,P.G., Powell,S.M., Moskaluk,C.A., Frierson,H.F., Jr and Hampton,G.M. (2001) Molecular classification of human carcinomas by use of gene expression signatures. *Cancer Res.*, **61**, 7388–7393.

Tsamardinos,I., Aliferis,C.F. and Statnikov,A. (2003) Time and sample efficient discovery of Markov blankets and direct causal relations. *The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2003)*, Washington, DC, August 24–27.

Valentini,G., Muselli,M. and Ruffino,F. (2003) Bagged ensembles of SVMs for gene expression data analysis. In The IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN2003), Portland, OR.

Vapnik,V. (1998) *Statistical Learning Theory*. Wiley-Interscience, New York, NY, USA.

Weiss,S.M. and Kulikowski,C.A. (1991) *Computer Systems that Learn*. Morgan Kaufmann, San Mateo, CA, USA.

Weston,J. and Watkins,C. (1999) Support vector machines for multi-class pattern recognition. In *Proceedings of the Seventh European Symposium On Artificial Neural Networks* (ESANN 99), Bruges, April 21–23.

Wouters,L., Gohlmann,H.W., Bjinens,L., Kass,S.U., Molenberghs,G. and Lewi,P.J. (2003) Graphical exploration of gene expression data: a comparative study of three multivariate methods. *Biometrics*, **59**, 1131–1139.

Yeang,C., Ramaswamy,S., Tamayo,P., Rifkin,R., Mukherjee,S., Angelo,M., Reich,M., Lander,E., Mesirov,J. and Golub,T. (2001) Molecular classification of multiple tumor types. In *Proceedings of the Ninth International Conference on Intelligent Systems in Molecular Biology*, Copenhagen, Denmark, July 21–25, pp. 316–322.

Yeo,G. and Poggio,T. (2001) Mutliclass classification of SRBCT tumors. *Technical Report AI Memo 2001-018 CBCL Memo 206*, MIT Press.