Multiclass Cancer Classification Using Gene Expression Profiling and Probabilistic Neural Networks

D.P. Berrar, C.S. Downes, W. Dubitzky

MULTICLASS CANCER CLASSIFICATION USING GENE EXPRESSION PROFILING AND PROBABILISTIC NEURAL NETWORKS

DANIEL P. BERRAR, C. STEPHEN DOWNES, WERNER DUBITZKY

School of Biomedical Sciences, University of Ulster at Coleraine,
BT521SA, Northern Ireland
E-mail: {dp.berrar, cs.downes, w.dubitzky}@ulster.ac.uk

Gene expression profiling by microarray technology has been successfully applied to classification and diagnostic prediction of cancers. Various machine learning and data mining methods are currently used for classifying gene expression data. However, these methods have not been developed to address the specific requirements of gene microarray analysis. First, microarray data is characterized by a high-dimensional feature space often exceeding the sample space dimensionality by a factor of 100 or more. In addition, microarray data exhibit a high degree of noise. Most of the discussed methods do not adequately address the problem of dimensionality and noise. Furthermore, although machine learning and data mining methods are based on statistics, most such techniques do not address the biologist's requirement for sound mathematical confidence measures. Finally, most machine learning and data mining classification methods fail to incorporate misclassification costs, i.e. they are indifferent to the costs associated with false positive and false negative classifications. In this paper, we present a probabilistic neural network (PNN) model that addresses all these issues. The PNN model provides sound statistical confidences for its decisions, and it is able to model asymmetrical misclassification costs. Furthermore, we demonstrate the performance of the PNN for multiclass gene expression data sets. Here, we compare the performance of the PNN with two machine learning methods, a decision tree and a neural network. To assess and evaluate the performance of the classifiers, we use a *lift*-based scoring system that allows a fair comparison of different models. The PNN clearly outperformed the other models. The results demonstrate the successful application of the PNN model for multiclass cancer classification.

1 Introduction

The diagnosis of complex genetic diseases such as cancer has traditionally been made on the basis of non-molecular criteria such as tumor tissue type, pathological features, and clinical stage. In the past several years, DNA microarray technology has attracted tremendous interest in both the scientific community and in industry. Several studies have recently reported on the application of microarray gene expression analysis for molecular classification of cancer [1,2,3]. Microarray analysis of differential gene expression has been used to distinguish between different subtypes of lung adenocarcinoma [4] and colorectal neoplasm [5], and to predict clinical outcomes in breast cancer [6,7] and lymphoma [8]. J. Khan et al. used an artificial neural network approach for the classification of microarray data, including both tissue biopsy material and cell lines [9]. Various machine learning methods have been investigated for the analysis of microarray data [10,11]. The combination of gene microarray technology and machine learning methods promises new insights into mechanisms of living systems. An application area where these

techniques are expected to make major contributions is the classification of cancers according to clinical stage and biological behavior. Such classifications have an immense impact on prognosis and therapy. In our opinion, a classifier for this task should address the following issues: (1) The classifier should provide an easy-to-interpret measure of confidence for its decisions. Thereby, the final diagnosis rests with the medical expert who judges if the confidence of the classifier is high enough. In one scenario, a classification that relies on a confidence of 75% might be acceptable, whereas in another, the medical expert only accepts classifications of at least 99%. (2) The classifier should take into account asymmetrical misclassification costs for false positive and false negative classifications. For example, suppose a tissue sample is to be classified as either benign or malign. A false positive classification may result in further clinical examinations, whereas a false negative result is very likely to have severe consequences for the patient. Consequently, the classifier should ideally be very "careful" when classifying a sample to the class "benign". The misclassification costs depend on the problem at hand and have to be evaluated by the medical expert. Machine learning methods that are able to address both issues are very rare. In this paper, we present a model of a probabilistic neural network for the classification of microarray data that addresses both issues.

Many publications report on cancer classification problems where the number of classes is rather small. For example, the classification problem of J. Khan et al. comprised four cancer classes [9], and the classification problem of T. Golub et al. comprised only two classes [1]. However, multiclass distinctions are a considerably more difficult task. S. Ramaswamy et al. recently reported on the successful application of *support vector machines* (SVM) for multiclass cancer diagnosis [2].

2   Probabilistic neural networks

Probabilistic neural networks (PNNs) belong to the family of radial basis function neural networks. PNN are based on Bayes' decision strategy and Parzen's method of density estimation. In 1990, D. Specht proposed an artificial neural network that is based on these two principles [12]. This model can compute nonlinear decision boundaries that asymptotically approach the Bayes' optimal. Bayesian strategies are decision strategies that minimize the expected risk of a classification. The Bayesian decision theory is the basis of many important learning schemes such as the naïve Bayes classifier, Bayesian belief networks, and the EM algorithm. The optimum decision rule that minimizes the average costs of misclassification is called *Bayes' optimal decision rule*. It can be proven that no other classification method using the same hypothesis space and the same prior knowledge can outperform the Bayes' optimal classifier on average [13]. The following definition is adapted from T. Masters [14]:

Definition 1: *Bayes' optimal classifier*

Given a collection of random samples from *n* populations. The prior probability that a sample belongs to population *k* is denoted as h_k. The costs associated with a misclassification of a sample belonging to population *k* is denoted as c_k. The conditional probability that a specific sample belongs to population *k*, $p(k \mid \ )$, is given by the probability density function $f_k(\ )$. An unknown sample is classified into population *i* if

for all populations .

   We refer to this decision criterion as *Bayes' decision criterion*. This criterion favors a class if the costs associated with its misclassification are high (c_i). Furthermore, the rule favors a class if it has a high prior probability (h_i). Finally, the rule favors a class if it has high density in the vicinity of the unknown sample ($f_i(\ )$). The prior probabilities *h* are generally known or can be estimated. The misclassifications costs *c* rely on a subjective evaluation. The probability density functions, however, are unknown in real-world applications and have to be estimated. D. Specht proposed to use Parzen's method for non-parametric estimation of the density using the set of training samples. D. Parzen proved that the estimated univariate probability density converges asymptotically to the true density as the sample size of the training data increases [15]. The estimator for the density function contains a weighting function that is also known as *kernel function* or *Parzen window*. The kernel is centered at each training sample. The estimated density is the scaled sum of the kernel function for all training samples. Various kernel functions are possible [16], but the most common kernel is the Gaussian function [14]. The scaling parameter σ defines the width of the bell curves and is also referred to as *window width*, *bandwidth*, or *smoothing factor* (the latter one is most commonly used in the context of PNNs). Equation 1 shows the estimated density for the multivariate case and the Gaussian as kernel function:

$$ \tag{1} $$

where   :   estimated density for the *j*-th class

    :   test case

    :   *i*-th training sample of the *j*-th population / class

dim :   dimensionality of

σ: smoothing factor

T: transpose

m_j :   number of training cases in the *j*-th class

D. Specht proposed a four-layered feed-forward network topology that implements Bayes' decision criterion and Parzen's method for density estimation. The operation of the basic PNN is best illustrated on a simple architecture as depicted in Figure 1:
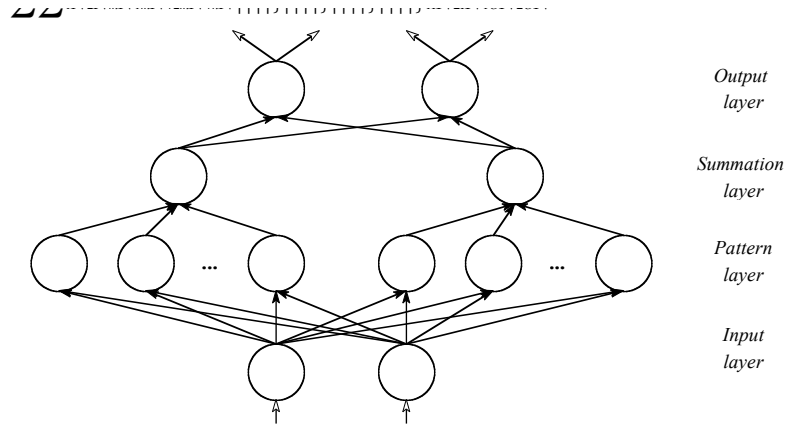
Figure 1: Architecture of a four-layered PNN for *n* training cases of 2 classes.

The *input layer* of the PNN in Figure 1 contains two input neurons, N_{I1} and N_{I2}, for the two test cases, and . The *pattern layer* contains one *pattern neuron* for each training case, with an exponential activation function. A pattern neuron N_i computes the squared Euclidean distance

between a new input vector and the *i*-th training vector of the *j*-th class. This distance is then transformed by the neuron's activation function (the exponential). In the PNN of Figure 1, the training set comprises cases belonging to two classes, *A* and *B*. In total, *m* training cases belong to class *A*. The associated pattern neurons are $N_1...N_m$. For example, the neuron N_3 contains the third training case of class *A*. Class *B* contains $n - m$ training cases; the associated pattern neurons are $N_{m+1}...N_n$. For example, the neuron N_{m+2} contains the second training case of class *B*. For each class, the *summation layer* contains a *summation neuron*. Since we have two classes in this example, the PNN has two summation neurons. The summation neuron for class *A* sums the output of the pattern neurons that contain the training cases of class *A*. The summation neuron for class *B* sums the output of the pattern neurons that contain the training cases of class *B*. The activation of the summation neuron for a class is equivalent to the estimated density function value of this class. The summation neurons feed their result to the output neurons in the *output layer*. These neurons are threshold discriminators that implement Bayes' decision criterion. The output neuron N_{O1} generates two outputs: the estimated conditional probability that the test case belongs to class *A*, and the estimated conditional probability that this case belongs to class *B*. The output neuron N_{O2} generates the respective estimated probabilities for the test case . Unlike other feed-forward neural networks, e.g., multi-layer perceptrons (MLPs), all hidden-to-output weights are equal to 1 and do not vary during processing. For the present study, we use the same smoothing factor σ for all classes. Whereas the choice of the kernel function

has no major effect on the performance of the PNN, the choice of σ has a significant influence. The smaller the smoothing factor, the more influence have individual training samples. The larger the smoothing factor, the more blurring is induced. It has been shown that neither limiting case provides optimal separation of class distributions [12]. Clearly, averaging multiple nearest neighbors results in a better generalization than basing the decision on the first nearest neighbor only. On the other hand, if too many neighbors are taken into account, then the PNN generalizes weakly as well. The optimal smoothing factor can be determined through cross-validation procedures. However, the choice of the smoothing factor always implies a trade-off between the variance and the bias of a kernel-based classifier. Further techniques for adapting σ and for improving the basic model of the PNN can be found in [17,18].

3 Analysis of the leukemia data set

The leukemia data set includes expression profiles of 7,129 human DNA probes spotted on Affymetrix Hu6800 microarrays of 72 patients with either acute myeloid leukemia (*AML*) or acute lymphocytic leukemia (*ALL*) [1]. Tissue samples were collected at time of diagnosis before treatment, taken either from bone marrow (62 cases), or peripheral blood (10 cases) and reflect both childhood and adult leukemias. Furthermore, a description of cancer subtypes, treatment response, gender, and source (laboratory) was given. RNA preparation, however, was performed using different protocols. The gene expression profiles of the original data set are represented as $\log_{10}$ normalized expression values. This data set was used as a benchmark for various machine learning techniques at the First Critical Assessment of Microarray Data Analysis at the Duke University in October 2000 (CAMDA 2000). The data set was divided into a training and a validation set. Table 1 shows the number of cases in the data sets:

Table 1. Distribution of cancer primary classes (*AML* and *ALL*) and subclasses in the training and the test set (*N/a*: no cancer subclass specified).

Primary class	*ALL*			*AML*				
Subclass	*B-cell*	*T-cell*	*M1*	*M2*	*M4*	*M5*	*N/a*	Σ
# of cases in training set	19	8	3	5	1	2	0	38
# of cases in validation set	19	1	1	5	3	0	5	34

The original data set of 7,129 genes contains some control genes that we excluded from further analysis. After this preprocessing, each sample consists of a row vector of 7,070 expression values. The classification of the leukemia subclasses is an even more challenging task than the classification of the primary classes (*ALL* and *AML*) in the CAMDA 2000, because the subclass distributions are very skewed in the training and the validation set. In a leave-one-out cross-validation procedure, we tested different values for the smoothing factor. We initialized σ with 0.01. The first case of the training set,   , was used as the hold-out case, and the remaining

37 cases were forwarded to the pattern layer. We assume equal misclassification costs for all cancer classes and classify   using Bayes' decision criterion (cf. Definition 1). Then, the second case was retained as the hold-out case, and the remaining cases were moved to the pattern layer. This procedure was repeated for 100 different values for the smoothing factor, ranging from 0.01 to 1.00. After all cases had been classified in turn, we performed a sensitivity analysis. Ideally, the sensitivity for each class should be maximal. Consequently, the optimal smoothing factor maximizes the sum of all sensitivities. Based on this criterion, the PNN performed best for a smoothing factor of 0.03 on the training set. Therefore, we chose this smoothing factor to classify the cases of the validation set. Table 2 shows the resulting confusion matrix for the classification of the cancer subclasses.

Table 2. Confusion matrix for the classification of the leukemia subclasses.

		M1	*M2*	*M4*	*M5*	*B-cell*	*T-cell*	*N/a*	*Σ*
						Real class			
	M1	1	1	-	-	3	-	1	6
	M2	-	4	1	-	-	-	-	5
	M4	-	-	-	-	-	-	-	-
Classification	*M5*	-	-	-	-	-	-	2	2
	B-cell	-	-	2	-	16	1	2	21
	T-cell	-	-	-	-	-	-	-	-
	N/a	-	-	-	-	-	-	-	-
	Σ	1	5	3	-	19	1	5	34
	sensitivity	*1.00*	*0.80*	*0.00*	*-*	*0.84*	*0.00*	*0.00*	
	specificity	*0.85*	*0.97*	*1.00*	*0.94*	*0.67*	*1.00*	*1.00*	

The PNN is very sensitive to the class *M1* and *M2*, but fails to classify the *M4* cases correctly. This can be explained by the fact that only one case of this subclass is contained in the training set. The sensitivity for the subclass *B-cell* is relatively high (0.84). However, the PNN misclassified the *T-cell* case a *B-cell* case, although the training set contained 8 *T-cell* cases. Given this relatively large number of cases of *T-cell* cases in the training set, this result is rather disappointing. In the validation set, 5 cases are of type *AML*, but no further subclass specification is given. In the training set, this class is not present, thus the PNN is not able to predict this class. Interestingly, 3 of these 5 cases are correctly classified as cases of type *AML* (1 case is classified as *M1*, 2 cases are classified as *M5*).

4   Analysis of the NCI60 data set

The NCI60 data set comprises 1,416 gene expression profiles of 60 cell lines [19]. The data set includes nine different cancer classes: central nervous system (*CNS*, 6 cases), breast (*BR*, 8 cases), renal (*RE*, 8 cases), lung (*LC*, 9 cases), melanoma (*ME*, 8 cases), prostate (*PR*, 2 cases), ovarian (*OV*, 6 cases), colorectal (*CO*, 7 cases), and leukemia (*LE*, 6 cases). The gene expression data comprise mainly ESTs of known and unknown function given by the negative logarithm of the ratio between the red and green fluorescence of the signals. The 60   1,416 microarray matrix contains

2,033 missing values in total. Different methods for missing value imputation in the context of microarrays have been discussed. We propose the following missing value imputation method: Let $v(c_i, g)$ denote the gene expression value for case c_i and gene g. If $v(c_i, g)$ is a missing value, then replace it by the mean of all values $v(c_j, g)$ where the cancer class of c_i and c_j is the same. This method makes explicit use of the class membership of each sample and is based on the following rational: O. Troyanskaya et al. resumed that k-nearest neighbor (kNN) methods provide for the best estimation of missing values in microarrays [20]. A major problem with kNN methods is the adequate choice of the number of neighbors (k) to be taken into account. It is probable that a gene is similarly expressed in samples of the same cancer type. For missing value imputation, we therefore consider only the neighbors that belong to the same class. For some genes, our imputation method was not possible. For example, the expression values of *topoisomerase II alpha-log* are missing for both cases of class *PR*. In total, the missing value imputation was not possible with the described method for 11 genes. These genes were excluded from further analysis.

Feature selection and dimension reduction techniques are both used to remove features that do not provide significant incremental information. In the context of microarray data, such features can be redundant genes. For example, if two genes are similarly co-regulated, then they provide the same basic information, and the removal of one of these genes does, in general, not result in a loss of information for a classifier. Numerous studies have revealed that in high-dimensional microarray data, feature selection and dimension reduction methods are essential to improve the performance of a classifier (for a general discussion, see [21]). Many publications report on dimension reduction techniques such as principal component analysis (PCA) that is based on singular value decomposition [22]. To assess the performance of our model, we tested the PNN in a leave-one-out cross-validation procedure (1) on the original data set, and (2) on a reduced data set, comprising only a set of principal components. We compared the performance of the PNN with the performance of two other machine learning methods: the decision tree C5.0 [23], and a neural network: the multi-layer feedforward perceptron (MLP), trained with the backpropagation algorithm [24]. The training of the MLP was stopped when no further optimization was possible. The MLP comprised one hidden layer, containing 7 neurons for classifying the original data, and 4 neurons for classifying the reduced data. We applied the leave-one-out cross-validation procedure as described above to all models; i.e. each model is trained on all but one sample (*hold-out* case), and then we used the model to predict the class of the hold-out case. We iterated this procedure until each case was used as hold-out case.

4.1 Analysis of the NCI60 original data set

After data cleansing, the original data set consisted of 60 cell-line samples (9 cancer classes), and 1,405 features (expression values of genes and ESTs). We assumed equal misclassification costs for all classes. Given the relatively small number of

cases per class, we chose a relatively small value for the smoothing factor. Table 3 shows the confusion matrix for $\sigma = 0.3$.

Table 3. Confusion matrix for the NCI60 original data set.

		Real class									
		CNS	BR	RE	LC	ME	PR	OV	CO	LE	Σ
Classification	CNS	5	1	-	-	-	-	-	-	-	6
	BR	1	5	1	1	-	-	1	-	-	9
	RE	-	-	7	2	1	-	-	-	-	10
	LC	-	1	-	5	-	1	1	-	-	8
	ME	-	-	-	-	7	-	-	-	-	7
	PR	-	-	-	-	-	-	-	-	-	-
	OV	-	1	-	-	-	-	4	-	-	5
	CO	-	-	-	1	-	1	-	7	-	9
	LE	-	-	-	-	-	-	-	-	6	6
	Σ	6	8	8	9	8	2	6	7	6	60
	sensitivity	*0.83*	*0.63*	*0.88*	*0.56*	*0.86*	*0.00*	*0.67*	*1.00*	*1.00*	
	specificity	*0.98*	*0.92*	*0.98*	*0.94*	*1.00*	*1.00*	*0.98*	*0.96*	*1.00*	

The sensitivity and specificity for the classes *CO* and *LE* are very high, whereas the PNN was not able to classify the *PR* cases. This can be explained by the leave-one-out cross-validation procedure: When a *PR* case is used as the hold-out case, the training set comprises only one *PR* case. In total, the PNN misclassified 14 cases (23.3%). However, if we accept only those classifications that rely on a confidence of at least $r = 0.8$, then the model misclassifies only 2 cases. Both C5.0 and MLP performed very weakly on the original data set (the respective confusion matrices are not shown). Their classification performance improved significantly on the reduced data set. Table 4 summarizes the performance of the three models on both the original and the reduced data set.

4.2   Analysis of the NCI60 reduced data set

We used PCA without mean centering. In our analysis, we used the first 23 principal components as hybrid variables; these variables explain more than 75% of the total variance. The sensitivities and specificities of the PNN are very similar to those that resulted from the original data set and are therefore not shown.

So far, we evaluated the performance of the PNN on the basis of its classification accuracy. However, accuracy-based evaluation metrics alone are inadequate to evaluate the performance of a classifier. A tacit assumption in the use of these accuracy measures is that the class distributions among the cases are constant and relatively balanced. The *lift* is a measure that takes different class distributions into account and is the preferred method for evaluating a classifier's performance [21].

Definition 2: *class lift* and *total lift*
Given the set of class labels, $C = \{c_1, c_2, ..., c_m\}$ and the set of cases, $S = \{x_1, x_2, ..., x_n\}$. Let *act*($x_j$) denote the actual class (label) of case x_j and *prd*(x_j)

the class (label) predicted for x_j by a classifier. Then the *class lift* for a particular class c_i, $lift(c_i)$, is measured by the prior probability, $p(act(x_j) = c_i)$, of class c_i occurring in S, and the conditional probability, $p(act(x_j) = c_i \mid prd(x_j) = c_i)$ of class $act(x_j) = c_i$ given the prediction, $prd(x_j) = c_i$, as follows:

Table 4 shows the *lifts* resulting from the three models for the original and the reduced data set.

Table 4. *Lifts* for the classification of the NCI60 data set (p.c.: principal component).

Class	Maximum lift	Class lift of PNN		Class lift of C5.0		Class lift of MLP	
		All data	23 p.c.	All data	23 p.c.	All data	23 p.c.
CNS	10.00	8.33	8.33	1.67	8.33	0.00	2.00
BR	7.50	4.17	3.75	2.14	3.75	1.67	1.25
RE	7.50	5.25	5.83	1.67	3.21	0.00	1.89
LC	6.67	4.17	5.56	2.50	1.03	0.00	1.82
ME	7.50	6.56	6.56	3.75	5.63	1.07	3.75
PR	30.00	0.00	0.00	0.00	0.00	0.00	0.00
OV	10.00	8.00	8.33	0.00	5.56	0.00	1.67
CO	8.57	6.67	7.50	3.43	6.43	1.43	3.43
LE	10.00	10.00	10.00	10.00	8.57	1.00	6.67
Total lift	10.86	6.01	6.21	2.80	4.72	0.57	2.50

The *lift* can be interpreted as a *score*: the more difficult the classification of a case, the higher the potential score for the classifier. Table 4 also shows the *maximum lift*, i.e. the highest score that a classifier can obtain. Although the decision tree and the neural network performed much better on the reduced data set than on the original data set, the PNN still outperformed both models. However, it should be noted that other feature selection methods might significantly improve the performance of the decision tree and the neural network. But it is interesting that the PNN performs similarly on both the original and the reduced data set. It seems that – compared with the other models – the PNN is less sensitive to noise.

5   Discussion

We consider the ability to provide sound confidence levels and the ability to model asymmetrical misclassification costs as the two most important qualities of PNN in the context of microarray analysis. PNN have shown excellent classification performance in other applications, and perform equally or better than other types of artificial neural networks (ANNs). In contrast to other types of ANNs, e.g. MLPs,

PNN are not "black boxes": The contribution of each pattern neuron to the outcome of the network is explicitly defined and accessible, and has a precise interpretation. The training of MLPs involves heuristic searches like the *steepest descent* method. These heuristics involve small modifications of the network parameters that result in a gradual improvement of system performance. Heuristic approaches are associated with long training times with no guarantee of converging to an acceptable solution within a reasonable timeframe. The training of PNN involves no heuristic searches, but consists essentially of incorporating the training cases into the pattern layer. However, finding the best smoothing factor for the training set remains an optimization problem. PNNs tolerate erroneous samples and outliers. Sparse samples are adequate for the PNN. Other types of ANN and many traditional statistical techniques are hampered by outliers. Finally, when new training data become available, PNN do not need to be reconfigured or retrained from scratch; new training data can be incrementally incorporated in the pattern layer.

A disadvantage of PNNs is the fact that all training data must be stored in the pattern layer, requiring a large amount of memory. But in general, today's standard PCs have a sufficiently large main memory capacity for an efficient implementation of PNN. In applications where large amounts of training cases are available, this argument against PNNs becomes relevant. But the problem can be circumvented by using cluster centroids as training cases, or by resorting to a parallel processor implementation.

Although the output of the PNN is probabilistic, we should keep in mind that the probabilities are estimates and conditional on the learning set. Future work will focus on an exhaustive comparison of state-of-the-art classifiers in multiclass cancer classification problems.

References

1. 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., Lander E.S., Molecular classification of cancer class discovery and class prediction by gene expression monitoring. *Science* **286**:531-537, (1999).
2. Ramaswamy S., Tamayo P., Rifkin R., Mukherjee S., Yeang C.H., Angelo M. Ladd C., Reich M., Latulippe E., Mesirov J.P., Poggio T., Gerald W., Loda M., Lander E.S., Golub T.R., Multiclass cancer diagnosis using tumor gene expression signatures, *Proc. Natl. Acad. Sci. USA.* **98**(26):15149-15154, (2001).
3. Tibshirani R., Hastie T., Narasimhan B., Chu G., Diagnosis of multiple cancer types by shrunken centroids of gene expression, *Proc. Natl. Acad. Sci. USA.* **99**(10):6567-6572, (2002).
4. Bhattacharjee A., Richards W.G., Staunton J., Li C., Monti S., Vasa P., Ladd C., Beheshti J., Bueno R., Gillette M., Loda M., Weber G., Mark E.J., Lander E.S., Wong W., Johnson B.E., Golub T.R., Sugarbaker D.J., Meyerson M., Classification of human lung carcinomas by mRNA expression

profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl. Acad. Sci. USA* **98**(24):13790-13795, (2001).

5. Selaru F.M., Xu Y., Yin J., Zou T., Liu T.C., Mori Y., Abraham J.M., Sato F., Wang S., Twigg C., Olaru A., Shustova V., Leytin A., Hytiroglou P., Shibata D., Harpaz N., Meltzer S.J., Artificial neural networks distinguish among subtypes of neoplastic colorectal lesions. *Gastroenterology* **122**:606-613, (2002).

6. West M., Blanchette C., Dressman H., Huang E., Ishida S., Spang R., Zuzan H., Olson J.A., Marks J.R., Nevins J.R., Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl. Acad. Sci. USA* **98**(20):11462-11467, (2001).

7. van't Veer L.J., Dai H.Y., van de Vijver M.J., He Y.D.D., Hart A.A.M., Mao M., Peterse H.L., van der Kooy K., Marton M.J., Witteveen A.T., Schreiber G.J., Kerkhoven R.M., Roberts C., Linsley P.S., Bernards R., Friend S.H., Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415**:530-536, (2002).

8. Shipp M.A., Ross K.N., Tamayo P., Weng A.P., Kutok J.L., Aguiar R.C.T., Gaasenbeek M., Angelo M., Reich M., Pinkus G.S., Ray T.S., Koval M.A., Last K.W., Norton A., Lister T.A., Mesirov J., Neuberg D.S., Lander E.S., Aster J.C., Golub T.R., Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine* **8**:68-74, (2002).

9. Khan J., Wei J.S., Ringnér M., Saal L.H., Ladanyi M., Westermann F., Berthold F., Schwab M., Antonescu C.R., Peterson C., Meltzer P.S., Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* **7**(6):673-679, (2001).

10. Lin S.M. and Johnson K.F. (eds.), *Methods of Microarray Data Analysis*. Kluwer Academic Publishers, Boston, (2002).

11. Berrar D., Dubitzky W., Granzow M. (eds.), *A Practical Approach to Microarray Data Analysis*, Kluwer Academic Publishers, Boston, Dec.2002.

12. Specht D.F., Probabilistic Neural Networks. *Neural Networks*, vol. 3, (1990) pp. 109-118.

13. Mitchell T.M., *Machine Learning*. McGraw-Hill Book Co., Singapore, (1997) pp. 174-175.

14. Masters T. *Advanced Algorithms for Neural Networks*. John Wiley & Sons, Academic Press, (1995).

15. Parzen E. On Estimation of a Probability Density Function and Mode. *Ann. Math. Stat.* **33**, (1962), pp.1065-1076.

16. Silverman B.W., *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability 26, Chapman & Hall, (1986).

17. Specht D.F., Enhancements to the probabilistic neural networks. Proc. of the IEEE Int. Joint Conf. on Neural Networks, Baltimore, MD., vol. 1, (1992) pp. 761-768.

18. Zaknich A., A vector quantisation reduction method for the probabilistic neural network. IEEE Proc. of the Int. Conf. on Neural Networks (ICNN), Houston/Texas, USA, (1997) pp. 1117-1120.

19. Scherf U., Ross D., Waltham M., Smith L., Lee J., Tanabe L., Kohn K., Reinhold W., Myers T., Andrews D., Scudiero D., Eisen M., Sausville E., Pommier Y., Botstein D., Brown P., Weinstein J., A gene expression database for the molecular pharmacology of cancer. *Nature Genetics* **24**(3):236-244, (2000).

20. Troyanskaya O., Botstein D., Altman R., "Missing value estimation", in: Berrar D., Dubitzky W., Granzow M. (eds.): *A Practical Approach to Microarray Data Analysis*, Kluwer Academic Publishers, Boston, Dec.2002.

21. Dudoit S. and Fridlyand, "Introduction to classification in microarray experiments", in: Berrar D., Dubitzky W., Granzow M. (eds.): *A Practical Approach to Microarray Data Analysis*, Kluwer Academic Publishers, Boston, Dec.2002.

22. Wall M.E., Rechtsteiner A., Rocha L.M.: "Singular value decomposition and principal component analysis", in: Berrar D., Dubitzky W., Granzow M. (eds.): *A Practical Approach to Microarray Data Analysis*, Kluwer Academic Publishers, Boston, Dec.2002.

23. RuleQuest Research Data Mining Tools. http://www.rulequest.com

24. Bishop C.M., *Neural Networks for Pattern Recognition*, Oxford, Oxford University Press, (1995).