BIOINFORMATICS ORIGINAL PAPER
Vol. 21 no. 19 2005, pages 3755–3762
doi:10.1093/bioinformatics/bti429

Gene expression

A protocol for building and evaluating predictors of disease state based on microarray data

Lodewyk F. A. Wessels[1,2,*], Marcel J. T. Reinders[1], Augustinus A. M. Hart[3], Cor J. Veenman[1], Hongyue Dai[4], Yudong D. He[4] and Laura J. van't Veer[2]

[1]Department of Mediamatics, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands, [2]Department of Pathology and [3]Department of Radiotherapy, The Netherlands Cancer Institute, Plesmanlaan 121, 1066 CX Amsterdam, The Netherlands and [4]Rosetta Inpharmatics LLC (a wholly owned subsidiary of Merck & Co., Inc.), 401 Terry Avenue N. Seattle, Washington 98109, USA

Received on November 2, 2004; revised and accepted on April 2, 2005
Advance Access publication April 7, 2005

ABSTRACT

Motivation: Microarray gene expression data are increasingly employed to identify sets of marker genes that accurately predict disease development and outcome in cancer. Many computational approaches have been proposed to construct such predictors. However, there is, as yet, no objective way to evaluate whether a new approach truly improves on the current state of the art. In addition no 'standard' computational approach has emerged which enables robust outcome prediction.

Results: An important contribution of this work is the description of a principled training and validation protocol, which allows objective evaluation of the complete methodology for constructing a predictor. We review the possible choices of computational approaches, with specific emphasis on predictor choice and reporter selection strategies. Employing this training-validation protocol, we evaluated different reporter selection strategies and predictors on six gene expression datasets of varying degrees of difficulty. We demonstrate that simple reporter selection strategies (forward filtering and shrunken centroids) work surprisingly well and outperform partial least squares in four of the six datasets. Similarly, simple predictors, such as the nearest mean classifier, outperform more complex classifiers. Our training-validation protocol provides a robust methodology to evaluate the performance of new computational approaches and to objectively compare outcome predictions on different datasets.

Contact: l.f.a.wessels@ewi.tudelft.nl

Supplementary information: http://ict.ewi.tudelft.nl/pub/wessels/wessels_protocol_29-10-2004_Supplemental.pdf

INTRODUCTION

Microarray gene expression profiling has become a widely used tool to identify particular disease subpopulations and to perform diagnostic and prognostic predictions (Van 't Veer *et al*., 2002; van de Vijver *et al*., 2002; Golub *et al*., 1999; Hedenfalk *et al*., 2001; Khan *et al*., 2001; Huang *et al*., 2003; Gruvberger *et al*., 2001). Typically the primary goal is to construct a predictor based on the measured gene expression dataset, such that the outcome (e.g. prognosis, disease type) on unseen cases can be predicted as accurately as possible. Since this prediction becomes, in general, more reliable if only a subset of the complete set of measured genes is employed by the predictor, a by-product of this process is a genetic profile (a set of marker genes) associated with the prediction problem.

The methodology for constructing a prediction rule based on microarray data involves several steps. These include selection of a quality filtered gene set, choosing a predictor (classifier), selection of the reporter genes in the final profile and independent validation of the rule. Many different variations on this basic methodology have been proposed, mostly involving different combinations of reporter selection strategies and predictors. Dudoit *et al*. (2002) compared different predictors and concluded that simple predictors, such as the diagonal linear discriminant classifier (DLDC), perform well on gene expression datasets. Recently, Romualdi *et al*. (2003) performed a comparison of predictors and dimensionality reduction techniques, with an emphasis on investigating the effect of certain experimental variables in a multi-class setting. Li *et al*. (2004) also compared predictors in multi-class problems, with an emphasis on issues relating to the combination of binary classifiers in multi-class settings. Inza *et al*. (2004) compared filter and wrapper reporter selection strategies on two gene expression datasets.

Here we review the basic methodology (from normalized array data to the final predictor) and evaluate the merits of the possible choices in each step. We investigate different types of predictors, chosen to cover a range of predictor types and to be representative of earlier employed rules. We also evaluate different strategies for selecting the reporters for the final profile. We compare different predictor–selector combinations on six gene expression datasets with varying degrees of difficulty in predictability: breast cancer metastasis (Van 't Veer *et al*., 2002; van de Vijver *et al*., 2002), colon cancer (Alon *et al*., 1999), leukemia (Golub *et al*., 1999), diffuse-large-B-cell-lymphoma (Alizadeh *et al*., 2000), prostate cancer (Singh *et al*., 2002) and central nervous system embryonic treatment outcome (Pomeroy *et al*., 2002). This study shows that 'simple' predictors (e.g. the nearest mean classifier) and simple reporter selection strategies (forward filtering) perform above expectations,

*To whom correspondence should be addressed at Information and Communication Theory Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, P.O.Box 5031, 2600 GA Delft, The Netherlands. Tel.: +31 15 2785114; fax: +31 15 2781843.

while forward filtering outperforms partial least squares as a dimensionality-reduction algorithm.

An important contribution of this work is the description of a training and validation protocol that allows objective evaluation of the complete methodology for constructing a predictor. Since the protocol reduces the effect of variance in the performance estimates and completely decouples the training from the validation steps, more reliable performance estimates are obtained. Consequently the protocol is a useful tool to evaluate different configurations of reporter selectors and predictors, and to determine whether a given novel methodology truly improves on existing cutting-edge approaches.

METHODS

In this section we outline the basic predictor construction methodology, which inputs a gene expression dataset and outputs a trained prediction rule. This rule predicts the relevant outcome based on a specified set of genes (reporters).

Most approaches follow the same basic steps, namely (1) reducing the initial complete set (tens of thousands) of genes to a smaller quality filtered set of genes (thousands); (2) selecting a suitable prediction rule and reporter selection algorithm; (3) training the rule, i.e. selecting the optimal set of reporters and tuning the parameters in the prediction rule itself; (4) estimating the expected generalization performance of the rule; (5) constructing the final predictor and (6) validating the performance of the final predictor on an independent validation series.

Quality filtering

The first step involves the reduction of the number of genes from the number on the array (tens of thousands) to several thousand. Often, heuristic approaches are followed by selecting genes with an N-fold variation (ratio of maximal and minimal expression) and an absolute difference (difference of maximal and minimal expression) which exceeds a given threshold.

Quality filtering should ideally be performed based on the quality of the measurements, as expressed objectively in an error model (Roberts *et al*., 2000). Such an error model provides p-values expressing the evidence that a particular gene is expressed differentially with respect to the control (employing the measurement error as yardstick). Consequently, genes are included that have been present ($p < p_{\min}$) in at least $N_{\min}$ tumors. This is a one-time selection, which is performed prior to reporter selection and predictor training, and is not repeated during later steps. Note that this step eliminates genes purely based on the quality of the measurement—it does not employ any information related to the outcome.

Predictor selection

In molecular classification we are faced with a situation where the number of samples (arrays), n, is much smaller than the number of genes, p. Since perfect prediction of the outcome for the arrays in the training set is likely to occur when employing all genes, care must be taken not to over-train the predictor. An over-trained predictor predicts the outcome of the training set systematically better than the outcome of unseen arrays. Several measures, such as the application of cross-validation during training and the reduction of the set of reporters through unbiased reporter selection, can be taken to prevent over-training (see next section). Choosing a predictor with a limited complexity also constrains the training process and is therefore an important tool to prevent over-training and ensure robust prediction.

For our study, predictors were selected based on their low complexity and/or the fact that these predictors were successfully applied in published studies. We also strived for a combination of nonlinear and linear predictors. In addition, all selected methods can be employed when $p > n$. We have included the following predictors:

(1) The nearest mean classifier (NMC) with cosine correlation as distance measure, previously employed on gene expression data in Van't Veer *et al*., (2002).

(2) Diagonal linear discriminant classifier (DLDC) (Dudoit *et al*., 2002). The expression ratios within the classes are assumed to be similarly distributed around the class means. The DLDC is, together with the NMC, the predictor with the lowest complexity.

(3) Simple Bayes Gaussian classifier (SBGC) (Domingos and Pazzani, 1997; Dudoit *et al*., 2002), also referred to as a quadratic linear discriminant, is a quadratic predictor, with a higher complexity than the NMC and DLDC predictors. This predictor is based on the assumption that the genes are independent and that the distribution of the expression ratios within each of the classes can be modeled by a Gaussian distribution. It was applied to molecular data by Dudoit *et al*. (2002) and Wessels *et al*. (2002a).

(4) k-Nearest neighbor predictor (k-NN) (Barnard, 1935). This predictor is a very simple, intuitive predictor frequently included in gene expression analyses (Dudoit *et al*., 2002; Ben-Dor *et al*., 2000). By varying the value of k, i.e. the number of neighbors employed to determine the class label of a new array, more robustness against noise can be introduced. In this study we varied k across the following values: $k \in \{1, 5, 9\}$.

(5) Regularized Fisher linear discriminant (RFLD) (Fisher, 1936). A classical, linear, low-complexity predictor of which the intrinsic complexity can be further reduced through regularization. We varied the regularization parameter across the following set of values: $\lambda \in \{0, 1, 10\}$.

(6) Linear support vector machine (linSVC) (Vapnik, 1999). A linear predictor specifically designed for the $p > n$ situation. Support vector machines were employed, amongst others, by Ben-Dor *et al*. (2000) and Guyon *et al*. (2002).

Reporter selection

Some methodologies perform a mapping of genes to meta-genes prior to training a predictor. For example, Huang *et al*. (2003) do a correlation-based k-means clustering and select the dominant principal component to be the meta-gene for each cluster. Gruvberger *et al*. (2001) and Khan *et al*. (2001) directly perform a principal component analysis to reduce the dimensionality from thousands of genes to tens of principal components. The principal components or 'meta-genes' are, in general, less interpretable abstractions of the real genes. However, meta-genes may, in some cases, be linked to underlying biological processes [see, e.g. Huang *et al*. (2003)]. Recently, Romualdi *et al*. (2003) showed that a dimensionality reduction method known as partial-least-squares (PLS)—which was first applied to gene expression data by Nguyen and Rocke (2002)—had either no effect or improved predictor performance on some datasets. For this reason we include this dimensionality reduction method in this study. However, it should immediately be noted that these methods do not reduce the number of reporter genes that will be employed in the final predictor. This has obvious drawbacks in cases where a custom diagnostic array should be designed containing only the marker genes required for correct classification.

Techniques that explicitly select a subset of reporter genes typically do so by searching for a set of reporter genes that optimize a given criterion. In the machine learning community a distinction is frequently made between filter and wrapper approaches (Kohavi and John, 1997). Wrapper approaches explicitly employ the prediction rule to determine the order in which a gene is added to the selected reporter set. Filter approaches, on the other hand, employ a criterion separate from the prediction rule to determine the order in which genes are added. Feature selection strategies can also be distinguished based on the *direction* of the search. Forward selection strategies start with an empty set and add reporters until no improvement in prediction performance results; backward selection starts with the complete set of genes and removes reporters until the prediction performance starts decreasing. In earlier studies (Wessels *et al*., 2002a,b) it was found that, in general, forward filter approaches performed at least as good as, and frequently even better than, forward wrapper approaches employing a greedy forward selection strategy. In addition, wrapper approaches are computationally far more

expensive. Therefore, we will not include forward wrapper approaches in this study.

Forward filtering involves two steps. First the reporters are ranked based on their capability to separate the outcome groups (classes). Then, starting with the best gene (largest separation between classes), the set of chosen reporters is expanded by adding the next best reporter. Each time a gene is added, the classification performance of the selected set of reporters is estimated by computing the cross-validation performance of a predictor employing these reporters. This process of adding a gene and testing is terminated when the cross-validation performance reaches a maximum.

In order to perform filtering, a ranking criterion needs to be selected. Several criteria can be employed for continuous-valued data, such as signal-to-noise ratio (SNR), *t*-test, Mann–Whitney *u*-test and Mahalanobis distance. In our experience (data not shown) these measures perform very similarly and therefore the choice of ranking criterion is not critical. We have chosen the SNR as the univariate ranking criterion for its simplicity, which has the added benefit of considerable computational speed-up. Unlike the other methods, it does not vary systematically with sample size, which may also be regarded as an advantage.

Two backward selection approaches are included. These were selected since these approaches were, apart from forward filtering and wrapping, the most frequently employed reporter selection approaches. There approaches are: (1) recursive feature elimination (RFE) (Guyon *et al*., 2002) and (2) shrunken centroids (SC) (Tibshirani *et al*., 2002). Strictly speaking, SC is usually performed in a backward fashion, i.e. gradually removing genes, but it could also be performed in a forward fashion, since the direction does not influence the course of the search. In contrast to the filter and wrapper approaches where the predictor and selection procedure can be independently selected, RFE and SC represent cases where the predictor and the selection algorithm are tightly integrated. The RFE method is a selection method designed to operate in conjunction with the support vector machine (SVM). An SVM is trained on the complete set of genes and assigns a weight to each gene. Only those genes with the largest absolute weights are retained for the next iteration. Typically, only half of the genes are retained. The cross-validation performance of every reporter set is registered at every elimination step. The process of training and removal is repeated on the remaining genes until the performance drops or a single gene is retained. Shrunken centroids assign a univariate distance score to every gene, which reflects the capacity of that gene to separate the classes in the dataset. This distance is gradually shrunk (reduced) to zero, and at every shrinkage step a linear predictor is constructed based on the shrunken centroids (class means reconstructed based on the shrunken distance scores). The cross-validation performance of this predictor is employed as a measure of the quality of the set of reporters employed. When a distance becomes zero for a particular gene, that gene does not participate in the prediction, and is removed from the reporter set. The shrinkage process stops when the performance reaches a maximum or when all genes have been shrunk away.

All the predictors discussed in the section entitled 'Predictor selection' were employed in conjunction with (1) filtering as a reporter selection algorithm with SNR as the ranking criterion and (2) PLS as a dimensionality reduction algorithm. In addition, the linSVC–RFE combination and SC were included as instances of integrated selector–predictor approaches.

Training-validation protocol

A frequently employed approach during predictor construction is to split the available samples into a training set and a validation set. Various rules are maintained for the sizes of these sets. The training set is employed to construct a predictor, and the validation set is employed to estimate the performance of the predictor on unseen samples. However, depending on the particular way in which the dataset is split (amongst a very large set of possibilities) the validation performance may vary widely. For the six datasets employed in this study, the best and worst validation performances obtained during many random splits are as follows: breast cancer (81.8%, 44.8%); colon (100%, 59.3%); leukemia (100%, 78.1%); DLBCL (100%, 51.8%); prostate (100%, 71.7%); and CNS (85.7%, 29.7%). These ranges were obtained for the NMC (similar ranges are obtained for the other predictors). From these results, it is quite apparent that the specific choice of split may greatly influence the estimate of the predictor's performance. To avoid such random variations, we therefore prefer an approach where the dataset is split multiple times, and the validation performance of each split is aggregated. This ensures a more complete description of the validation performance, since a good approximation of the *distribution* of the performance is obtained, from which the mean and the variance can be estimated. This repeated splitting of the dataset, training on one part and testing on the remaining part and afterwards aggregating the results from the different splits, boils down to performing cross-validation repeatedly. Typically, the training step also employs cross-validation to optimize certain parameters associated with the predictor. Consequently the training-validation protocol is characterized by a 'double-cross-validation-loop': an inner training loop and an outer validation loop.

Cross-validation is a powerful, frequently employed tool to train predictors and estimate the generalization performance (performance on unseen cases). It is generally accepted (Ambroise and McLachlan, 2002) that leave-one-out-cross-validation (LOOCV) is unbiased with respect to the training set, but highly variable (Efron, 1983). Ten-fold cross-validation (10FCV)—which leaves out 10% of the data—is, on the other hand, more biased, but has lower variance. Based on this fact, as well as empirical evidence (albeit in $p < n$ settings) (Kohavi and John, 1995), 10FCV is frequently the method of choice. Here we employ a smaller fold, since a larger proportion of the dataset is available for testing, resulting in more reliable estimates of differences between different approaches. More specifically, we employ three-fold cross-validation (3FCV) in the outer or validation loop to have a larger proportion of the data for validation [see also Dudoit *et al*. (2002)], and we employ 10FCV in the inner or training loop, since a larger proportion of the data is then employed to optimize the predictor.

The cross-validation error rate is employed to guide the training process. Often, the total error rate is employed, which is the sum of the false negatives (FN) and false positives (FP). In microarray datasets, the positive class (tumor versus normal tissue, recurrence versus no recurrence) frequently consists of fewer samples than the negative class. It is often desirable to identify the positive cases as accurately as possible. Under these conditions, the total error rate has the disadvantage that a predictor which always guesses the majority class will achieve a low total error rate, while it is, in fact, misclassifying all the positive cases. For this reason, we employ the 'average-true-positive-true-negative-performance' which is given by (TP/P +TN/N)/2, with TP, TN, P and N denoting true positives, true negatives, number of positives and number of negatives, respectively. In diagnostic terms, this is the average of sensitivity and specificity.

A training protocol delivers, given a training dataset, a trained predictor and an estimate of the performance of this predictor on unseen data. We therefore split the training-validation protocol into two parts: a training procedure and a validation step. Figure 1 provides a simplified schematic overview of the complete protocol. The first step is to split the dataset, X, in a training and validation set. When N-fold cross-validation is employed in the outer loop, the dataset is split into N folds. During one cross-validation iteration, all folds but fold j is employed as training set, denoted by $X_{(-j)}$ and fold j is employed as the validation set, denoted by $X_{(j)}$. Since we employ 3FCV for the outer loop, the dataset is split into three folds, or a ratio of 1:2. It is always ensured that each of the splits is properly stratified with respect to the classes in the dataset; i.e. that all classes are represented in the same ratios in each fold as they are represented in the complete dataset.

The purpose of the training procedure (Block 4 in Fig. 1) is to set all parameters in a given predictor employing the training set. For example, for the nearest mean predictor and forward filtering as reported selector, the training procedure optimizes the number of reporters to employ and the values of the centroids given these reporters. We employ 10FCV to perform this optimization; i.e. we split the training set, $X_{(-j)}$, in ten different training-test-folds, determine the centroids and optimal reporter set on the training fold and estimate the performance on the test fold (Block 1 in Fig. 1). At the end of a 10FCV run we have created ten predictors with their associated reporter sets.

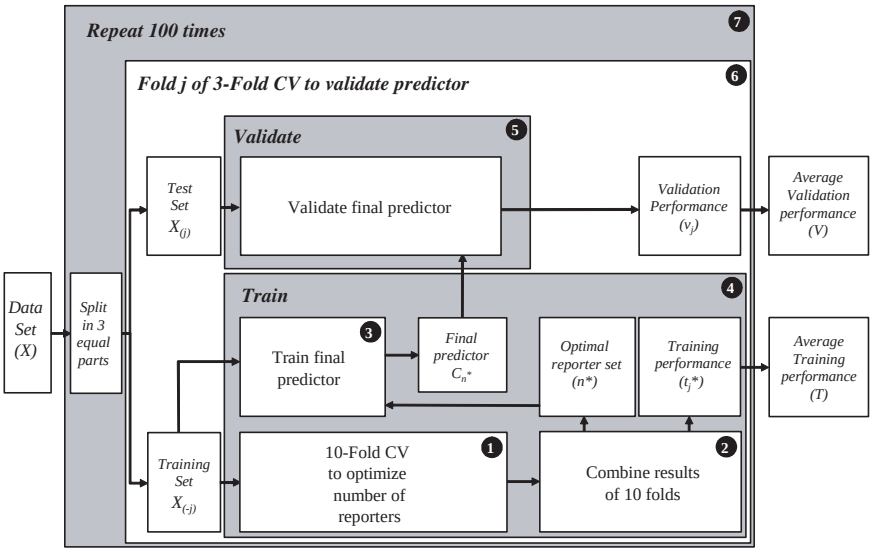

Fig. 1. The train-validation protocol in a simplified schematic format. The input is a labeled dataset and the output is an estimate of the training and validation performance. The most important steps in the protocol are (1) the training step (Block 4) consisting of the optimization of the number of reporters in a ten-fold cross-validation procedure (Block 1), combining these results (Block 2) and training the final predictor (Block 3); and (2) the validation step (Block 5) where this final predictor is validated on a completely independent validation set. These steps are performed within a three-fold cross-validation procedure which is repeated 100 times, each time for a different split of the dataset.

In Block 2, these ten predictors and reporter sets are combined into a single final predictor. The output of this combining process is an estimate of the optimal number of reporter genes to employ, n^*, and the training performance, t_j^*. The final step of the training procedure is performed in Block 3, where the complete training set, $X_{(-j)}$, and the optimal number of reporters, n^*, is employed to train the final predictor, C_{n^*}. In summary, the training procedure inputs the training set and outputs the training performance and a trained predictor with the optimal parameter settings.

In the validation step, the final trained predictor, C_{n^*}, is validated on the validation set, $X_{(j)}$ (the fold of X that was not employed in the training process). The validation step (Block 5) estimates the performance of the trained predictor on new, previously unseen samples. This simulates, for example, the situation in the clinic where the final predictor is employed to predict the disease state of a new patient entering the clinic for the first time. It is important to note that the samples in the validation set were *never* involved in the optimization of *any* of the parameters of the final predictor, and can therefore be employed to perform a truly independent performance estimate, denoted by v_j. The training and validation steps (Block 6) are repeated for each fold (indexed by j) and the validation performances of the different folds are averaged. Finally, the complete outer cross-validation (Block 7) is repeated 100 times, and the resulting cross-validation performances ($v_1, \ldots, v_{300}$) are averaged to obtain a final validation performance estimate, V. The complete protocol is described in greater detail in Figure 2 in the Supplementary information.

Ambroise and McLachlan (2002) pointed out the pitfalls associated with biased gene selection. In the cases they discuss, the bias is introduced as follows. The *complete* dataset is employed to select a subset of reporter genes. Then the expected performance of the predictor with the selected subset of reporter genes is estimated in a cross-validation procedure. This cross-validation performance estimate is upwardly biased since the tumors present in the validation sets in each of the cross-validation folds were also employed to select the subset of genes. Since gene selection is part of the training process, this amounts to training on the validation set, and hence the bias. This effect was also demonstrated for random datasets by Simon *et al*. (2003). The bias is removed by performing both the gene selection and predictor training only on the training set of a particular fold of the cross-validation process. In our protocol, this ensures that the construction of the predictor is completely independent from the evaluation thereof in the cross-validation procedure.

DATASETS

In this study, we revisit our published breast cancer series (Van 't Veer *et al*., 2002; van de Vijver *et al*., 2002). We also selected five publicly available datasets, the colon cancer (Alon *et al*., 1999), leukemia (Golub *et al*., 1999), DLBCL (Alizadeh *et al*., 2000), prostate cancer (Singh *et al*., 2002) and CNS (Pomeroy *et al*., 2002) datasets. For the colon, leukemia, DLBCL and prostate cancer datasets, the purpose is to predict the disease/tissue type. For the breast cancer and CNS datasets, the purpose is to predict *future* events: whether metastasis will occur (breast cancer) and the effectiveness of treatment (CNS). The latter type of problem (outcome prediction) is, generally, significantly more difficult than type-prediction. This is also reflected in the performance levels typically achieved. For this reason, the selection of datasets provides a good test of different prediction methodologies. All datasets are two-class prediction problems, and the datasets and quality filtering strategies are described in detail in the Supplementary information.

EXPERIMENTS

Each of the nine predictors (NMC; DLDC, SBGC; 1-NN; 5-NN; 9-NN; RFLD[0]; RFLD[1]; RFLD[10]; linSVC) was paired with both filtering (as reporter selector) and PLS, as the dimensionality reduction scheme. In addition, linSVC was paired with RFE, while SC is by design an integrated predictor and selector. This results in a total of 22 predictor–selector combinations. For each of these combinations, and each of the six datasets (breast cancer, colon, leukemia, DLBCL, prostate, CNS) the training-validation protocol was applied. The 10FCV employed during training was repeated five times while the 3FCV employed for validation was repeated

Table 1. Validation, V, and training, T, performance of the different predictors and reporter selection strategies on the breast cancer dataset[a]

Reporter selection	Predictor	T (%) Mean	T (%) SD	V (%) Mean	V (%) SD	k*	W	D	L
Filter	NMC	64.3	3.8	62.7	3.1	92	152	6	142
	DLDC	62.3	3.9	60.6	3	88	119	9	172
	SBGC	60.8	4.2	59.7	3	69	103	6	191
	1NN	61.2	4	60.3	3.5	95	109	4	187
	5NN	61.2	4	59.3	3.5	102	108	4	188
	9NN	60.7	4.2	58.8	3.3	88	90	4	206
	RFLD[0]	61.1	5.6	59.2	4	96	97	3	200
	RFLD[1]	61.8	5.6	60.7	3.8	93	117	4	179
	RFLD[10]	63.4	4	61.8	3.1	86	132	4	164
	LinSVC	60.6	4.3	60.6	3.6	102	111	2	187
PLS	NMC	62.5	3.1	61.7	2.2	12.4	138	4	158
	DLDC	61.6	3.6	60.1	3.2	12.4	116	3	181
	SBGC	61.2	3.9	58	3.6	10.7	92	2	206
	1NN	56.6	2.9	51.9	3.6	10.1	29	1	270
	5NN	56.7	3	52.7	3	8.9	29	1	270
	9NN	56	3	52.5	2.7	8.4	30	0	270
	RFLD[0]	59.5	3.5	55.7	3.5	9.2	67	2	231
	RFLD[1]	59.5	3.5	55.7	3.4	9.2	66	3	231
	RFLD[10]	60.8	3.6	58.6	3.2	11.6	89	4	207
	LinSVC	59.2	3.7	56.4	3.2	12.1	64	2	234
SC	**SC**	**65**	**3.4**	**62.9**	**1.9**	**909**	—	—	—
RFE	LinSVC	62.8	3.8	59.8	3.4	648	107	4	189

[a]The best predictor–selector combination is indicated in bold. The column marked 'k*' contains the average number of factors selected during training for the cases where PLS is employed for dimensionality reduction, and the optimal number of genes selected in the remaining cases. The columns 'W', 'D' and 'L' represent the number of times a particular selector–predictor combination won (better validation performance), drew (same performance) and lost (worse performance) from the best combination across all 300 validation folds.

100 times. In order to efficiently compare the results, each validation fold contained exactly the same objects for all selector–predictor combinations.

Strictly speaking, parameters such as the value of k (number of neighbors) in the nearest neighbor predictor and λ in the RFLD should also be optimized in the training procedure, since a choice of these parameters based on the validation performance also introduces a slight bias. However, we have chosen to isolate these parameters in order to highlight their effect on the validation performance of the predictors. In general, this bias is much smaller than the bias which Ambroise and McLachlan (2002) warn against.

RESULTS

The results of applying the training and validation protocol for the 22 selector–predictor combinations on the breast cancer dataset are depicted in Table 1 (Supplementary Tables 4–8 contain similar results for the other datasets). In these tables, the first four columns list the means and standard deviations (across the 100 repetitions) of the training, T, and validation, V, performances, respectively. The three validation performances associated with a particular 3-fold run were averaged, and the standard deviation of these averages is listed in the tables. The next column contains the value of k*, which represents the average number of genes selected by a particular selector–predictor combination during training. More specifically, this is the average value of n* across the validation repeats. For the cases where PLS is employed for dimensionality reduction, this column represents the average number of factors selected during the training runs. The next three columns represent a breakdown of the results obtained for each of the validation folds. Note that there are 300 folds in total: 100 repetitions of 3FCV. This breakdown reflects the performance of each selector–predictor combination with respect to the combination with the best overall validation performance. The columns 'W', 'D' and 'L' represent the number of times a particular combination won (better validation performance), drew (same performance) and lost (worse performance) from the best combination for a particular dataset across all 300 validation folds. Note that in each of the 300 trials a direct comparison can be made between the different combinations, since exactly the same samples were employed in each split for every classifier. The 'won–drawn–lost' representation provides additional information to interpret differences in average performance between two combinations. In the clinical setting, this has the following implication. Given two predictors, P_a and P_b, where P_a is better than P_b, it implies that if both predictors are employed to predict the disease state of a very large number of patients, the average performance of P_a across all patients will exceed that of P_b. However, it is quite possible that for some of these patients predictor P_b will predict correctly, while P_a is in error. However, we expect P_a to win a majority of the times. The won–drawn–lost decomposition helps to interpret the margin of difference between the average performances. For example, for the breast cancer dataset (Table 1) SC performs slightly better than filter-NMC in terms of validation performance. This small margin is also reflected in the won–drawn–lost breakdown: filter-NMC wins 152 times and loses 142 times from SC

Table 2. Comparison of forward filtering and PLS on the different datasets[a]

	Breast cancer	Colon	Leukemia	DLBCL	Prostate	CNS
NMC	−1.0	−4.7	1.8	−4.6	−20.9	−1.7
DLDC	−0.5	−4.7	1.5	−4.2	−17.6	−0.1
SBGC	−1.7	−7.2	5.5	−7.4	−25.2	1.3
1NN	−8.4	−3.5	1.8	−12.3	−8.1	−2.5
5NN	−6.6	−8.7	1.2	−7.1	−11.3	−3.8
9NN	−6.3	−11.2	1.0	−7.8	−9.7	−3.5
RFLD[0]	−3.5	−0.2	2.3	−32.9	2.8	−11.8
RFLD[1]	−5.0	−1.8	1.1	−29.5	2.8	−9.2
RFLD[10]	−3.2	−1.6	1.9	−4.2	2.5	−9.5
linSVC	−4.2	3.6	2.6	−4.6	2.8	−3.7

[a]Each entry in this table represents the absolute change in average validation performance (in percentage) when substituting filtering with PLS with the same predictor. Unshaded cells indicate an improvement with PLS, while lightly shaded cells represent a degradation in forward filtering performance due to PLS.

(even though filter-NMC has a smaller average performance; it wins more often; however, when it loses, it loses by a larger margin). On the other hand, filter-RFLD[1] performs worse than SC but still wins 117/300 times. In the next subsections we will emphasize specific aspects of the results.

Filter versus PLS

An interesting question we could pose is whether filtering as reporter selector outperforms the dimensionality reduction approach PLS. Table 2 provides a summary of the validation performances of the different predictors on the different datasets with these two selector approaches. Each entry in this table represents the absolute change in average validation performance when substituting filtering with PLS with the same predictor. Entries colored light gray indicate cases where PLS causes a decrease in performance, while unshaded entries indicate an improvement due to PLS. Partial least squares causes an improvement for all predictors on the leukemia dataset and improves the RFLD[λ] and linSVC to such an extent on the prostate dataset that it results in the best overall validation performance on prostate. On all the other datasets PLS causes a (in some cases dramatic) decrease in performance. Why this happens is not completely clear at this stage. The performance of PLS does not seem to be correlated with the type of microarray employed to measure the data (one-color Affymetrix or two-color cDNA/oligo/Agilent) and will be a topic for further research. The experiments show, contrary to the results presented in Romualdi *et al*. (2003), that PLS does not improve performance *in general*.

Training and validation performance

An inspection of the average training performance (T) and average validation performance (V) in Table 1 for all the selector–predictor combinations reveals two striking characteristics in the data. (Similar characteristics are revealed in the results for the other datasets presented in the Supplementary information.) First, the training and validation performance is well correlated. This is to be expected, since the training performance aims at producing a good predictor; therefore performance on the training data should be *indicative* of validation performance. Second, the average training set performance is generally an over-estimate of the final (most unbiased) validation set performance. This clearly indicates the benefit of adding an independent validation loop to the training-validation protocol to remove any biases that may be present in the training performance. Interestingly enough, this bias need not always be upward. When 2FCV is employed during the training phase, training performance *underestimates* the final validation performance (results not shown). This is probably caused by the fact that in 10FCV the training performance is based on the predictor extracted from 90% of the training set, while in 2FCV it is based on 50% of the training set, leading to a lower performance. In contrast, the validation performance is based on the predictor extracted from the complete training set in both cases. Quite reassuring is the fact that the validation performances obtained when 10FCV and 2FCV are employed in the training step correlate very well with each other, with Pearson correlation coefficients ranging from 0.88 to 0.96. The correction of the performance estimate obtained during the training phase when the predictor is evaluated on an independent validation set in the outer loop emphasizes the importance of this double loop construction in estimating predictor performance.

Global comparison

Table 3 contains a global summary of all the results contained in Table 1 and Supplementary Tables 4–8. Each column of this table presents the difference in validation performance of a particular selector–predictor combination and the best combination on a particular dataset. The second-to-last column contains the median validation performance difference across all datasets (median of that row). The top five average performances are shaded gray and the worst five are colored dark gray. From this global ranking it is quite clear that PLS in combination with the nearest neighbor classifiers is a particularly poor match, with very poor average differences. Among all combinations employing filtering as selector, the SVC and 1-NN perform the worst. The SVC did perform slightly better with RFE, a reporter selection strategy tailor-made for this classifier. On the positive side, the filter-NMC, filter–DLDC, filter-RFLD[10] combinations and SC perform the best. Predictors such as the DLDC, RFLD (with high degree of regularization) and the classifier in SC are closely related to the NMC, and their similar performances are therefore not completely surprising.

The last column indicates the largest absolute performance difference across a row. Here DLDC has the lowest value; i.e. it performs consistently well, with SC and NMC performing slightly worse. With filter-NMC, filter–DLDC and SC scoring well in terms of the average and maximal difference, these are the selector–predictor combinations of choice.

DISCUSSION

In this paper we make a detailed proposal for a training-validation protocol that can be employed to create a predictor of outcome (e.g. disease state) based on a dataset consisting of a series of (gene) expression microarrays derived from a labeled set of exemplars, e.g. tumor samples. The most important aspect of this protocol is that it ensures that all training steps, such as reporter set selection and optimization of the remaining parameters of the classifier, are performed completely independently from the final validation step, where the performance of the predictor is estimated. This ensures

Table 3. Comparison of the selector–predictor combinations on the different datasets[a]

Reporter selection	Classifier	Breast cancer	Colon	Leukemia	DLBCL	Prostate	CNS	Median performance difference	Maximal performance difference
Filter	NMC	0.2	0	1.9	0.7	4.6	0	0.45	4.6
	DLDC	2.3	0.8	1.6	0.8	1.6	1.1	1.35	2.3
	SBGC	3.2	0.6	5.8	1	2.4	5	2.8	5.8
	1NN	2.6	8	2.8	0.8	5.8	4.6	3.7	8
	5NN	3.6	1.5	1.4	0.7	2.5	2.6	2	3.6
	9NN	4.1	0.6	1.3	0.5	2.4	3.3	1.85	4.1
	RFLD[0]	3.7	2.6	3	3.9	2.8	2.4	2.9	3.9
	RFLD[1]	2.2	0.7	1.6	1.3	2.8	5	1.9	5
	RFLD[10]	1.1	0.9	1.9	0	2.5	4.7	1.5	4.7
	linSVC	2.3	6.8	3.2	1.5	3	3.7	3.1	6.8
PLS	NMC	1.2	4.7	0.1	5.3	25.5	1.7	3.2	25.5
	DLDC	2.8	5.5	0.1	5	19.2	1.2	3.9	19.2
	SBGC	4.9	7.8	0.3	8.4	27.6	3.7	6.35	27.6
	1NN	11	12	1	13.1	13.9	7.1	11.25	13.9
	5NN	10.2	10	0.2	7.8	13.8	6.4	9	13.8
	9NN	10.4	12	0.3	8.3	12.1	6.8	9.35	12.1
	RFLD[0]	7.2	2.8	0.7	36.8	0	14.2	5	36.8
	RFLD[1]	7.2	2.5	0.5	30.8	0	14.2	4.85	30.8
	RFLD[10]	4.3	2.5	0	4.2	0	14.2	3.35	14.2
	linSVC	6.5	3.2	0.6	6.1	0.2	7.4	4.65	7.4
SC	SC	0	1.7	0.7	1	4.3	0.3	0.85	4.3
RFE	LinSVC	3.1	6.2	1	5.3	1.2	1.2	2.15	6.2

[a]Each column of this table presents the difference in validation performance (in percentage) of a particular selector–predictor combination and the best combination on a particular dataset. The second-to-last column contains the median validation performance difference across all datasets (median of that row). The top five average performances are shaded gray and the worst five are colored dark gray.

that this performance estimate is not an upwardly biased (i.e. optimistic) estimate of the performance of the predictor. Another important characteristic of this protocol is that the steps of independent training and validation are repeated frequently, to overcome the sampling effects that are strongly present in microarray datasets. We advocate this approach over an approach where the dataset is split once in a training and validation set, and the performance on the validation set is employed as the true performance estimate of the predictor trained on the training set. This preference stems from the fact that the validation set is typically small (∼20% of the total dataset) which increases the variability in the validation estimate (up to 50% in performance difference), and could, therefore, in some cases result in an optimistic performance estimate. A validation set of substantial size (e.g. more than twice the training set size) obviously reduces the variance in the validation performance estimate. However, it suffers from the obvious drawback that the majority of the measured samples are not employed during the training process. In this study we proposed using 3FCV in the outer validation loop. Leave-one-out cross-validation is frequently employed in microarray studies, and could also be employed in the outer validation loop. It has the advantage that the performance estimates have a smaller bias, but suffer from larger variance (Efron, 1983). Results with LOOCV employed in the outer loop, for filtering in combination with all predictors, indicated that the relative performances of the combinations remain largely the same, with a slight performance increase (∼3%) on the prostate and CNS datasets (results not shown). The classifiers that showed the least sensitivity to a change in the fold of the outer loop are also the best performing classifiers, namely the filter-NMC and filter–DLDC. Similarly, experimental evidence (not shown here) indicated that when we switched from 10FCV to 2FCV in the inner loop, no noticeable differences with respect to the validation performance as measured in the outer loop were observed. Nevertheless, in the case of 2FCV, the choice of the number of reporters would be based on only 1/3 of the full sample and this might cause problems in the case of small sample sizes. Employing 3FCV in the validation loop has the following advantages. In a comparative study it puts more emphasis on the differences between the performances of the predictors [see also Dudoit *et al*. (2002)]. In addition, many three-fold loops can be performed to overcome the sampling effects. Finally, 3FCV has the advantage that a relatively large portion of the samples are set aside for testing during each loop, while the training set remains relatively large. This results in conservative performance estimates and is preferable since one then errs on the side of caution.

Given such a training-validation protocol, it is possible to objectively evaluate different strategies for constructing predictors. The most important components of this multi-step strategy that were evaluated here are the types of reporter selectors and predictors being employed. Here we compared 22 selector–predictor combinations on six gene expression datasets. The most important conclusions are that PLS slightly improved the performance on two datasets (leukemia when combined with any predictor and prostate when combined with RFLD) while resulting in an often dramatic deterioration in performance on all the other datasets. Even in cases where PLS is the best dimensionality reduction technique, the number of genes

required to make the final prediction is not reduced: in order to classify a new case, all genes need to be measured, and mapped to the optimal PLS subspace. From a cost perspective, this might be a drawback, since complex arrays are then required for diagnostic tests. Generally speaking, simple approaches performed very well. The best combinations are filter-NMC, filter–DLDC and SC. This can be explained by the fact that these are known to be most tolerant to noisy measurements and to datasets with potentially many irrelevant features.

The results of two studies seem to contradict some of our own conclusions. Inza *et al*. (2004) found that wrapper methods generally outperform filter methods, while Romualdi *et al*. (2003) found that PLS resulted in a better performance of the classifier compared to PCA and shrunken centroids. One important difference between the current paper and the work of Inza *et al*. is the definition of the term 'filtering'. Rather than employing the approach we followed, i.e. to employ the classifier to choose the optimal number of top-ranked reporters, Inza *et al*. fix the number of top-ranked genes in advance. We prefer our approach since the estimation of the number of genes to employ in the final predictor is exactly one of the objectives when constructing diagnostic classifiers from microarray data. In addition, fixing the number of reporters in advance results (as Inza *et al*. state themselves) in sub-optimal performance and partially explains why their filter approach is outperformed by their wrapper approach. However, apart from these and other dissimilarities between our study and those by Inza *et al*. and Romualdi *et al*. that make comparisons difficult, in both studies only a single cross-validation loop was employed both to construct the classifier and to estimate its performance. It can be shown that in both papers the resulting bias especially favored the methods which turned out as optimal. Since similar forms of bias are also present in other studies [see e.g. (Ambroise and McLachlan, 2002) for additional examples], a standardized robust training and evaluation protocol becomes a necessity to ensure fair comparisons between studies.

The proposed protocol can be further refined. The most important (possible) improvements involve the way in which the results obtained during the different cross-validation training runs can be combined to produce the final trained predictor. Currently, e.g. for filtering, the performance versus number of reporter curves are *averaged* and maximized to determine the optimal number of reporters to employ. Other approaches, such as a voting-combination of the actual predictors created during all the folds of the training process, or a selection of a subset of reporters occurring most frequently in the top of the rankings associated with each fold, can be employed. Another possible improvement involves a more refined optimization of the predictor with respect to sensitivity and specificity by employing ROC curves during the training process. For some predictors it is not the average sensitivity and specificity rates that are important but rather the lowest false negative rate, and the methodology can be adapted accordingly (Van 't Veer *et al*., 2002).

REFERENCES

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

Alon,U. *et al*. (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA*, **96**, 6745–6750.

Ambroise,C. and McLachlan,G.J. (2002) Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl Acad. Sci. USA*, **99**, 6562–6566.

Barnard,M. (1935) The secular variations of skull characters in four series of Egyptian skulls. *Ann. Eugenics*, **6**, 352–371.

Ben-Dor,A. *et al*. (2000) Tissue classification with gene expression profiles. *J. Comput. Biol.*, **7**, 559–583.

Domingos,P. and. Pazzani,M.J. (1997) On the Optimality of the simple Bayesian classifier under zero-one loss. *Mach. Learn.*, **29**, 103–130.

Duda,R.O., Hart,P.E. and Stork,D.G. (2000) *Pattern Classification*, 2d ed. Wiley, New York, NY.

Dudoit,S. *et al*. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *JASA*, **97**, 77–87.

Efron,B. (1983) Estimating the error rate of a prediction rule: improvements in cross-validation. *JASA*, **72**, 316–331.

Fisher,R.A. (1936) The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, **7**, 179–188.

Golub,T. *et al*. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Gruvberger,S. *et al*. (2001) Eastrogen receptor status in breast cancer is associated with remarkbly distinct gene expression patterns. *Cancer Res.*, **61**, 5979–5984.

Guyon,I. *et al*. (2002) Gene selection for cancer classification using support vector machines. *Mach. Learn.*, **46**, 389–422.

Hedenfalk,I. *et al*. (2001) Gene-expression profiles in hereditary breast cancer. *N. Engl. J. Med.*, **344**, 539–548.

Huang,E. *et al*. (2003) Gene expression predictors of breast cancer outcomes. *Lancet*, **361,** 1590–1596.

Inza,I. *et al*. (2004) Filter versus wrapper gene selection approaches in DNA microarray domains. *Artif. Intell. Med.*, **31**, 91–103.

Khan,J. *et al*. (2001) Classificaiton and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med.*, **7**, 673–679.

Kohavi,R. and John,G.H. (1997) Wrapper for feature subset selection. *Artif. Intell.*, **97,** 273–324.

Kohavi,R. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *IJCAI-95*, San Mateo, pp. 1137–1143.

Li,T. *et al*. (2004) A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, **20**, 2429–2437.

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

Pomeroy,S.L. *et al*. (2002) Prediction of central nervous system embryonal tumor outcome based on gene expression. *Nature*, **415**, 436–442.

Roberts,C.J. *et al*. (2000) Signaling and circuitry of multiple MAPK pathways revealed by a matrix of global gene expression profiles. *Science*, **287**, 873–880.

Romualdi,C. *et al*. (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.

Simon,R. *et al*. (2003) Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *J. Natl Cancer Inst.*, **95**, 14–18.

Singh,D. *et al*. (2002) Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, **1**, 203–209.

Tibshirani,R. *et al*. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl Acad. Sci. USA*, **99**, 6567–6572.

van de Vijver, M.-J. *et al*. (2002) A gene-expression signature as a predictor of survival in breast cancer. *N. Engl. J. Med.*, **347**, 1999–2009.

Van 't Veer,L.J. *et al*. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Vapnik,V. (1999) *Statistical Learning Theory*. John Wiley and Sons, New York.

Wessels,L.F.A., Reinders,M.J.T., van Welsem,T. and Nederlof,P.M. (2002a) Representation and classification for high-throughput data. *Proceedings of SPIE*, Vol. 4626, BIOS2002, San Jose, CA.

Wessels,L.F.A. *et al*. (2002b) Molecular classification of breast carcinomas by comparative genomic hybridisation a specific somatic genetic profile for BRCA1 tumors. *Cancer Res.*, **62**, 7110–7117.