Available online at www.sciencedirect.com
ScienceDirect
International Journal of Approximate Reasoning
47 (2008) 58–69

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

Integrating gene expression profiling and clinical data

Silvano Paoli, Giuseppe Jurman, Davide Albanese, Stefano Merler, Cesare Furlanello *

FBK-irst, via Sommarive 18, I-38100 Povo, Trento, Italy

Received 3 April 2006; received in revised form 17 October 2006; accepted 15 March 2007
Available online 21 April 2007

Abstract

We propose a combination of machine learning techniques to integrate predictive profiling from gene expression with clinical and epidemiological data. Starting from BioDCV, a complete software setup for predictive classification and feature ranking without selection bias, we apply semisupervised profiling for detecting outliers and deriving informative subtypes of patients. During the profiling process, sampletracking curves are extracted, and then clustered according to a distance derived from dynamic time warping. Sampletracking allows also the identification of outlier cases, whose removal is shown to improve predictive accuracy and stability of derived gene profiles. Here we propose to employ clinical features to validate the semisupervising procedure. The procedure is demonstrated in the analysis of a liver cancer dataset of 213 samples described by 1993 genes and by pathological features.

Keywords: Statistical learning; Classification; Feature selection; SVM; Functional genomics; BioDCV; DNA microarray; Biomarkers

1. Introduction

Microarray and other high-throughput technologies are being extensively used to evaluate genetic markers potentially associated with onset and progression of disease. To determine the role of different genetic profiles on disease outcomes, it is crucial to devise new machine learning methods supporting the identification of patterns from a variety of genomic, biological, and clinical data sources. In most cases, deciphering the physiological basis of complex diseases requires to clarify the heterogeneity of the illness by subgrouping the disease population [1], and possibly to exploit annotation data present in gene-centered corpora [2].

In the last few years, a growing interest has arises for integrating genomic and clinical data [3–5]. For instance, a very recent work [6] in this field proposes a strategy based on Bayesian networks to treat clinical and microarray data on an equal footing. In this probabilistic strategy, the integration of the two data sources may be obtained at different levels of complexity to understand the underlying model structure and parameters.

* Corresponding author. Tel.: +39 0461 314592; fax: +39 0461 302040.
E-mail addresses: silpaoli@itc.it (S. Paoli), jurman@itc.it (G. Jurman), albanese@itc.it (D. Albanese), merler@itc.it (S. Merler), furlan@itc.it (C. Furlanello).

0888-613X/$ - see front matter
doi:10.1016/j.ijar.2007.03.012

In our approach we choose not to include the clinical features straightforwardly inside the classification/ranking process: instead, the clinical features are used downstream in the procedure to validate the biological consistency of the resulting prediction. We first search for a profiling process improved both in terms of predictive accuracy as well as of stability of the gene signature. Then we analyze the subtypes derived from the expression data in conjunction with the available pathological information. In particular, we propose to employ clinical features to validate the results of the semisupervised procedure.

We base this procedure on BioDCV (Biodata Distributed Complete Validation), a software setup for predictive molecular profiling. Complete validation is an experimental scheme for the correct assessment of predictive accuracy in gene expression studies. In order to control for selection bias [7], it requires intensive resampling and replication of the classification processes. To overcome computing limits of standard resources, BioDCV was designed to implement complete validation schemes on distributed computing resources such as clusters and virtual GRID facilities [8]. The profiles considered in this study are based on the method recently introduced in [9] for semisupervised pattern discovery from functional genomics data. The method has been proposed to obtain subtyping and outlier detection from diagnostic functions derived from the complete validation scheme applied to a target classification task.

As an example, we apply the BioDCV system to Liver cancer profiling, considering a dataset of relatively large size and endowed with description of pathological features [10]. The dataset of Sese's study includes more than one hundred of positive samples of different age, sex and previous exposure to diseases or physiological states potentially correlated to liver cancer. In the case of the Sese dataset, the original target function is the discrimination of liver cancer patients from control cases. Answering to this biological question is a relatively easy task, on this dataset, as estimated predictive error is close to 3%. Questions such as subtyping for response to treatment are typically much more complex to answer, and semisupervised learning has been proposed in particular for predicting survival [11].

The structure of the paper is as follows. In Section 2, we describe first the core classification and feature selection procedures used in the BioDCV system. In the same Section, we define the complete validation schema, the semisupervised approach and the outlier detection strategy. In Section 3, we present the application on the liver cancer dataset: classification results, sampletracking analysis, outlier detection and stability of obtained biomarkers are discussed, on both the complete and the shaved dataset. The integration with the clinical features is then presented in the dedicated Section 3.3.

2. Methods

The study of gene expression patterns is expected to enable significant advances in disease diagnosis and prognosis. Generally, biomaterials (such as tissues) from different phenotypes are analyzed for an automatic discrimination based on their gene expression profile, trying to highlight which are the most important features (genes) supporting this classification. Therefore, we need:

(i) a *classifier* to train the model;
(ii) a *ranking method* to find the more important genes;
(iii) a *complete validation* procedure to protect from selection bias in estimating predictive error and biomarker list.

In Section 2.1 we outline the selection bias problem, while in 2.2 and 2.3 we briefly introduce SVM and feature selection methods; in Section 2.4 we detail the employed procedure designed to avoid the selection bias. In Section 2.5 we describe the procedure employed for the detection of subtypes and outlier removal. Clinical features are then used to validate the results.

2.1. Selection bias

As reported in [12], a serious procedural problem affects a number of results in the literature in gene profiling methodology. Initial studies on microarray data proposed classification models defined by very few genes and resulting in negligible or zero error rates. As discussed in [13,14], the problem is that the feature-

selection process has to be separated from the classification accuracy assessment to avoid uncorrected estimates of the prediction error. This flaw in methodology is know as "selection bias". While the problem may be reproduced with any wrapper algorithm, selection bias is a specific risk for recursive gene selection procedures, and especially for systems based on the RFE–SVM pair.

2.2. Support vector machines

Support vector machines (SVM) [15], which are considered a performing classification method for gene-expression data, were soon embedded with feature selection procedures. A backward selection approach like the recursive feature elimination (RFE) procedure is often adopted with SVM [16] (see Section 2.3).

Let F: $X \to \{-1,1\}$ be an unknown function and let $D = \{p_i \equiv (x_i, y_i)\}_{i=1}^{N}$ be a set of training examples, where $x_i \in X$ and $y_i \in \{-1,1\}$. In order to approximate the function F the following algorithm can be considered:

- Choose a Mercer kernel, i.e., a continuous, symmetric and positive definite function $K : X \times X \to \mathbb{R}$. Examples of such Mercer kernels are the Linear kernel $K(x,x') = \langle x, x' \rangle$, the Gaussian kernel $K(x,x') = \mathrm{e}^{-\|x-x'\|^2/\sigma}$ and the polynomial kernel $K(x,x') = 1 + \langle x, x' \rangle^d$. Observe that for any given $\bar{x}$, the function of a single variable $K_{\bar{x}}(x) = K(\bar{x}, x)$ can be defined.
- Choose the regularization parameter $C \in (0, +\infty)$.
- Define $f_C : X \to \mathbb{R}$ as follows:

$$f_C(x) = \sum_{i=1}^{N} c_i K_{x_i}(x), \tag{1}$$

where $c_i = y_i \alpha_i$ and the vector α is the solution to the quadratic programming problem:

$$\begin{cases} \max\limits_{\alpha \in \mathbb{R}^N} & \sum\limits_{i=1}^{N} \alpha_i - \frac{1}{2} \sum\limits_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \text{subject to} & 0 \leqslant \alpha_i \leqslant C \quad i = 1, \ldots, N. \end{cases}$$

- Define the classifier $\tilde{f}_C(x) = \mathrm{sign}(f_C(x))$. This is the parametric approximation of the unknown function $F(x)$.

Typically, the regularization parameter C is optimized by employing statistical model selection procedures, e.g., cross-validation. In the linear case, approximation (1) reduces to $f_C(x) = \langle w, x \rangle + b$.

2.3. Feature selection

The recursive feature elimination (RFE) is a well-known feature selection method for support vector machines firstly introduced in [16]. This method has been evaluated in experimental analysis and it is considered effective for gene selection and classification on microarrays. The idea is to define the importance of a feature for SVM in terms of its contribution to a cost function $J(\alpha)$. At each step of the RFE procedure, a SVM is trained on the given data set, J is computed and the feature that less contributes to J is discarded. In the case of linear SVM, the difference due to the elimination of the ith feature is

$$\delta J(i) = w_i^2,$$

and in the non-linear case is

$$\delta J(i) = \frac{1}{2} \alpha^t Z \alpha - \frac{1}{2} \alpha^t Z(-i) \alpha,$$

where $Z_{i,j} = y_i y_j K(x_i, x_j)$.

The heavy computational cost of RFE is a function of the number of original variables, because a SVM must be trained each time a variable is removed. The elimination of a single variable at each step (as in the

basic RFE procedure) is, however, inefficient. Indeed, at the first loops of the RFE algorithm, many weights are generally similar and concentrated nearby zero. The removal of a group of variables at every loop represents a feasible approach, and it was suggested in [16].

Entropy-based recursive feature elimination (E-RFE), is a non-parametric procedure for gene ranking, which accelerates – without reducing accuracy – the standard recursive feature elimination (RFE) method for SVM. This strategy was introduced in [17]. The aim of the E-RFE procedure is to provide a more flexible feature elimination mechanism in which the ranking is obtained by discarding groups of genes which contribute least to the SVM classifier. In E-RFE we cautiously discard, according to the entropy of the weight distribution, several (possibly many) genes at each step to drive the weight distribution in a high-entropy structure of a few equally important variables.

2.4. The complete validation system: BioDCV

The complete validation methodology involves three different procedures. The method is composed of three main procedures, organized as in Fig. 1:

ONF (optimal number of features); the procedure computes the optimal number of features (n^*).
OFS-M (optimal feature set-model); the procedure trains the model with the first n^* ranked features. The model is tested on a test portion.
VAL (validation); the procedure validates the OFS-M procedure over B replicates according to a resampling scheme.

In summary, given a dataset (matrix of gene expressions), the VAL procedure analyses B replicated experiments (runs) according to a resampling scheme. At each run, a training/test split ($\mathrm{TR}^b, \mathrm{TS}^b$) is created and only the training portion is used by OFS-M procedure. The ONF procedure identifies an optimal feature subset and the corresponding model is constructed in OFS-M procedure. The model is tested on the test portion TS^b (unused in the development of the model). Thus an average test (predictive) error can be computed from the B test error values TE^b in the VAL procedure.

More in detail, given a training set TR^b, the ONF procedure is applied to select the optimal number of features based on a ranking method. A resampling procedure is iterated K times, producing each time a $(\mathrm{TR}^b_k, \mathrm{TS}^b_k)$ split of TR^b. A feature ranking is applied to TR^b_k. Then, n subsets are created with the first F_i features of the *feature list* (i.e., $F_1 = 1, F_2 = 5, F_3 = 10, \ldots, F_n = 1000$). Therefore, for each k a model family ($M^b_{ki}, \quad i = 1, \ldots, n$) is produced, one for each increase of F_i. The M^b_{ki} models are evaluated on the TS^b_k test data, computing TE^b_{ki} test errors, and we obtain the average error curve $\mathrm{TE}^b_i = \frac{1}{K}\sum_{k=1}^{K}\mathrm{TE}^b_{ki}$. An exponential fit is applied to TE^b_i, and the n^* value leading to saturation in terms of the exponential curve is returned as the ONF result.

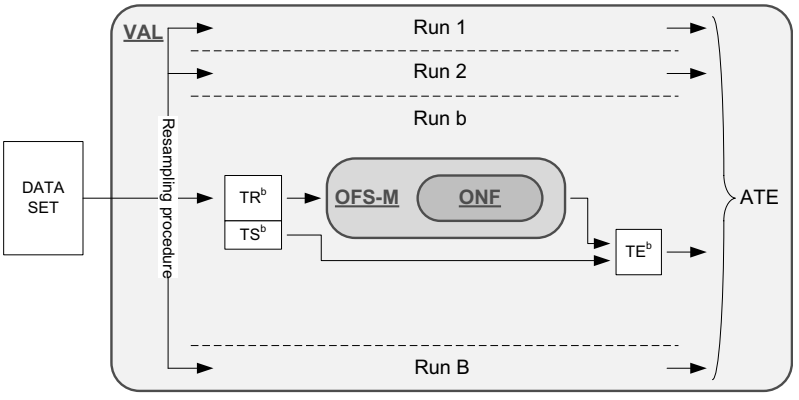


Fig. 1. The complete validation setup.

Given a training set TR^b, a feature ranking method produces a list of ranked features, from which an optimal feature set OFS of size n^* is selected. Based on ONF procedure, a model M^b is developed by a suitable learning method. The accuracy of OFS-M is validated by the VAL procedure on B replicated experiments (runs) using a resampling scheme. The model with n^* features is tested on the test set, in order to minimize risk of data overfitting, obtaining a TE^b error. The procedure returns the expected test error

$$\mathrm{ATE} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{TE}^b.$$

Together with the average test error, a bootstrap confidence interval at 95% is usually computed for each point throughout all the replicated experiments.

2.4.1. BioDCV implementation

The BioDCV system (http://biodcv.itc.it) implements the above described complete validation setup. BioDCV is portable from single workstations to local Linux clusters and virtual GRID facilities. It is written in C and interfaced with the SQLite database management library (http://www.sqlite.org). Support to concurrent access and transactions is used to store and manage results and parameters during the learning, tuning and evaluation tasks, which may be replicated for up to a few millions of models in a distributed environment. The system is distributed under GPL.

The main engine of BioDCV is the `libml` library, a C toolbox for learning problems, including the support vector machine (SVM) applied in this study. BioDCV runs also within the Egrid (http://www.egrid.it) computational infrastructure, based on Globus/EDG/LCG2 middleware and integrated as an independent virtual organization within the Grid.it, the INFN production grid. Part of the computation described in this paper was performed on a local computing facility (an Open Mosix cluster of 26 bi-processors units and 1 data server).

2.5. Semisupervised profiling and outlier shaving

The semisupervised procedure [9] implemented in BioDCV is based on an analysis of the effect of the feature selection and ranking process for each individual sample. Given a complete validation setup (such as the one described in Section 2.4), for each sample s, we count the number $N(s)$ of runs in which s is extracted in a test set and, for each feature step size f_s also the number of times $W(s,f_s)$ that s is wrongly classified when in the test set. The sequences $E_s(f_s) = W(s,f_s)/N(s)$ may be studied as an estimate of the classification error as a function of the size of the feature set. We call $E_s(f_s)$ the sampletracking profile (curve) of the sample: easily classifiable points correspond to curves reaching zero, while curves not far from the no-information error rate (the

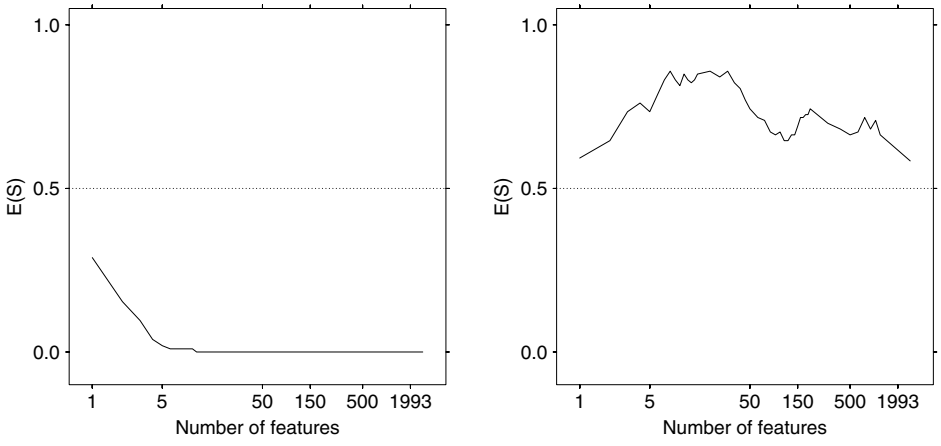


Fig. 2. Examples of sampletracking profiles from the liver cancer dataset (see Section 3): easy point (left) and outlier (right).

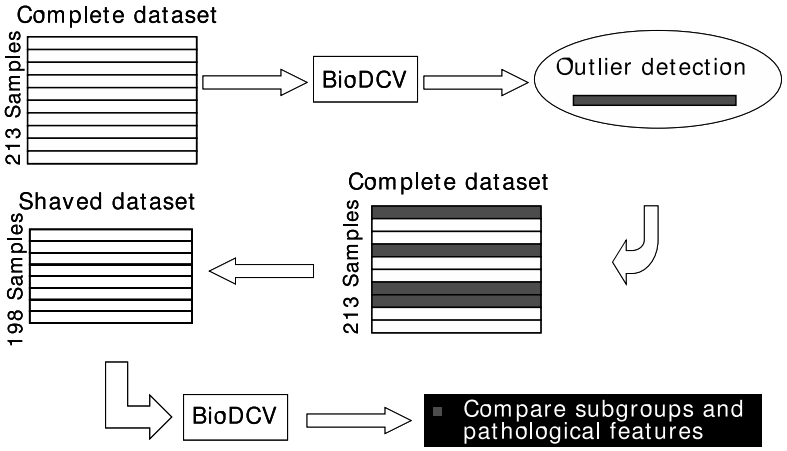


Fig. 3. Block diagram of the BioDCV profiling. To identify subgroups and study the association with clinical features, BioDCV is run twice, first on the complete dataset and then on its shaved version (outliers removed).

prior of the class with smaller cardinality) should belong to points hard to classify; a profile lying in the highest error zone indicates a typical outlier behaviour. Examples for the liver cancer dataset used in this study are shown in Fig. 2.

The response to the supervised classification task is then used to drive a secondary unsupervised pattern discovery process. Similar classification responses may be aggregated by a hierarchical clustering technique, which we apply with a $(1, 2, 1)$ dynamic time warping (DTW) distance. Aach and Church [18] proposed DTW for use on gene time series data in expression studies. This distance is more suited than Euclidean metric in curves comparison, because it takes into account morphology instead of just evaluating the pointwise distance of the vectors.

The described methodology is summarized in Fig. 3. BioDCV is first applied to the entire data set and the sampletracking profiles are analyzed for outlier detection and removal. Then, BioDCV is again applied on the outlier shaved dataset. The clusters of the sampletracking profiles resulting from the semisupervised procedure, defining potential subtypes, are finally compared with pathological features to detect common relevant patterns.

3. Results

We applied the semi-supervised profiling procedures to the liver cancer dataset originally analyzed in [10]. The dataset consists of 213 cases described by 1993 values of ATAC-PCR gene expression. The basic task is the predictive discrimination of the 107 samples extracted from tumors (liver cancer patients) from the 106 control samples. Several clinical features are available and of interest for the pathology, as displayed in Table 1. The pathological information is encoded as boolean variables, i.e., by binary features.

3.1. Semisupervised analysis

Following the pipeline in Fig. 3, the BioDCV system was applied to the complete liver cancer dataset. Linear SVM models were used for classification, with regularization parameter $C = 100$. Classification was coupled to the E-RFE ranking method, reverting to standard RFE in the last 100 steps. Average test error (ATE) results are reported in Table 2 for increasing number of features. Results are obtained by averaging over 400 replicated experiments.

The sampletracking curves (examples are shown in Fig. 4) were then obtained and hierarchically clustered with respect to DTW distance by the `hclust` algorithm (average link) from the `stats` package in the *R* statistical system [19].

Table 1
Summary of pathological features and their binary encoding

Abbr.	Feature	Value	Binary
V-B	Hepatitis B	Positive	1
		Negative	0
V-C	Hepatitis C	Positive	1
		Negative	0
A	Age	Over 65 years old	1
		Not over 65 years old	0
S	Sex	Male	1
		Female	0
C-A	Child score A	High	1
		Low	0
C-B	Child score B	High	1
		Low	0
Cir	Cirrhosis	Present	1
		Absent	0

Table 2
Predictive error (ATE: average test error) for the liver cancer dataset with the full set of 213 samples and after outlier removal (shaved)

Feat.	Complete dataset		Shaved dataset	
	ATE	CI	ATE	CI
1	27.7	(26.9, 28.6)	24.3	(23.6, 25.0)
2	24.3	(23.5, 25.3)	20.4	(19.5, 21.6)
3	21.8	(21.0, 22.8)	15.4	(14.6, 16.4)
4	18.4	(17.8, 19.1)	11.4	(10.9, 11.9)
5	16.6	(16.1, 17.3)	9.9	(9.4, 10.4)
10	12.7	(12.3, 13.1)	5.9	(5.6, 6.2)
20	8.9	(8.6, 9.3)	3.4	(3.2, 3.7)
50	5.8	(5.5, 6.1)	1.8	(1.6, 2.0)
100	4.8	(4.5, 5.0)	1.5	(1.3, 1.6)
500	3.4	(3.1, 3.6)	1.5	(1.3, 1.7)
1000	3.1	(2.9, 3.4)	1.5	(1.4, 1.7)
1993	3.2	(3.0, 3.5)	1.5	(1.3, 1.7)

Studentized bootstrap confidence intervals (CI .95 level) are included.

A group of 15 samples were detected as outliers: two of them are displayed in Fig. 4. The detection was derived from their sampletracking patterns and further confirmed by the cluster structure.

The BioDCV system was applied again after removing the outliers. The original and new accuracy scores are compared in Table 2: for every feature set size, the ATE values improve after removing the outliers. In particular, a minimum error is achieved at 100 features (ATE = 1.5%). With only 20 genes, the ATE = 3.4% is obtained on the shaved dataset, while 500 genes were required for the same ATE on the complete dataset. The dendrogram obtained by clustering the sampletracking curves of the cancer cases is shown in Fig. 5. The rightmost arm of the dendrogram (denoted by C1) is particularly well separated from the other positive samples, providing indication for further analysis (see Section 3.3).

3.2. Stability analysis

We compared the ranked gene lists before and after the removal of outliers (stability analysis). For the two cases, the genes ranked in the top k positions in the 400 lists were listed and ordered for decreasing multiplicity of extractions (*Exts*). The lists computed profiling the shaved dataset are shorter: for instance, at $k = 5$ and $k = 20$, respectively, 129 versus 228 genes and 330 versus 427 genes were extracted at least once. The top genes extracted in the shaved dataset are consistently more important for classification. In fact, the best genes are

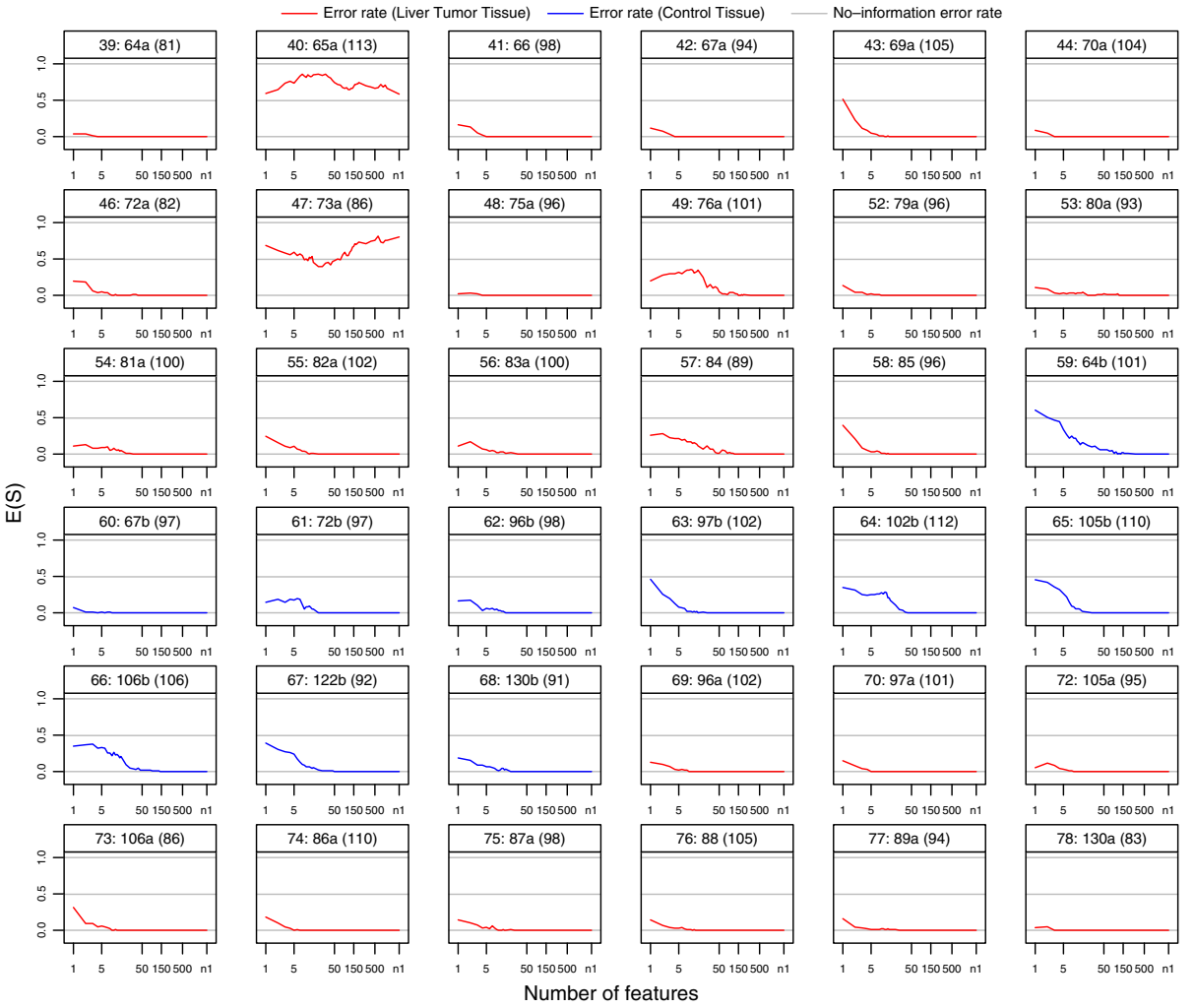


Fig. 4. Sampletracking profiles for disease (red) and control (blue) samples from the liver cancer dataset. Sample 40 and sample 47 are two outliers. (For better interpretation of the figure in colour, the reader is referred to the web version of this article.)

extracted more frequently, and their positions have smaller mean and standard deviation. A comparison for the best ranked 10 genes is detailed in Table 3 for $k = 5$ and $k = 20$.

We then compared the distribution of the mutual distances among the lists (1993 genes, Canberra distance) in the unshaved and shaved versions of the dataset (Fig. 6). The distribution of the distances in the reduced dataset has lower mean (437.9 versus 487.7) and standard deviation (41.4 versus 42.8), indicating that the 400 lists of the shaved dataset are mutually more similar than those of the complete dataset.

In summary, the structure of the two sets of ranked lists are different. The analysis points out that the outlier shaving has improved both ATE and the stability of the features (genes) that are relevant for the classification problem.

3.3. Integration with clinical data

The semisupervised procedure in Section 3.1 evidenced an interesting subgroup of 8 liver cancer samples (C1). As a case study, C1 was then analyzed in terms of the relationships between pathological features and gene expressions. In Fig. 7, the structure of C1 is paired with the clinical data.

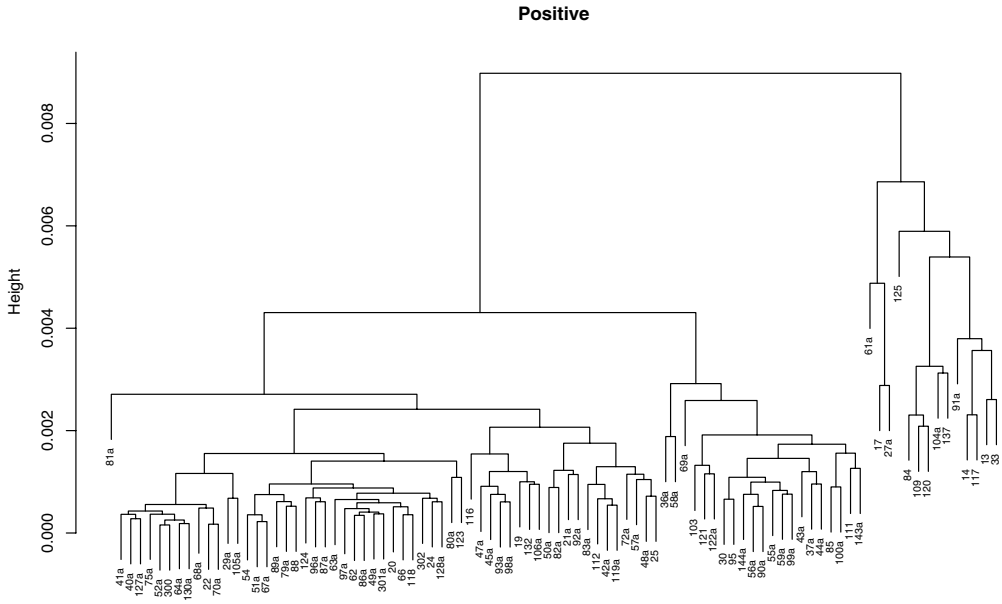


Fig. 5. Dendrogram of the DTW-distance clustering from semisupervised analysis of the liver cancer cases (after outlier removal).

Table 3
Summary of the properties of the best 10 genes extracted in the top k positions ($k = 5$, $k = 20$) in both experiments

Pos.	Gene	Exts.	Mean	SD	Pos.	Gene	Exts.	Mean	SD
$k = 5$ − complete					$k = 5$ − shaved				
1	GS1686	120	1.8	1.2	1	GS3244	217	2.8	1.1
2	GS201	90	3.3	1.3	2	GS6094	196	3	1.2
3	GS11954	87	2.3	1.4	3	GS1686	155	2.2	1.4
4	GS1324	85	2.5	1.4	4	GS11954	120	2.8	1.3
5	GS6094	78	2.4	1.3	5	GS2303	106	2.3	1.4
6	GS3244	74	2.7	1.3	6	GS10759	86	3.5	1.2
7	GS6487	70	2.4	1.2	7	GS12544	78	2	1.4
8	GS10588	56	2.1	1.1	8	GS201	75	3.6	1.2
9	GS11601	56	2.7	1.3	9	GS2131	67	2.5	1.4
10	GS5927	52	2.8	1.4	10	GS2954	61	3.6	1.1
$k = 20$ − complete					$k = 20$ − shaved				
1	GS201	260	9	5.5	1	GS3244	356	5.6	4.4
2	GS1324	212	8.4	5.8	2	GS6094	336	5.9	4.3
3	GS1686	204	6.2	6.1	3	GS201	276	8.9	4.8
4	GS3244	198	8.7	5.9	4	GS1686	264	5.8	5.2
5	GS6094	194	8.7	6.1	5	GS10759	248	8.6	5.1
6	GS11601	194	9.7	5.9	6	GS1710	235	10.1	5.4
7	GS11954	187	7.5	5.9	7	GS11954	232	7.2	5.5
8	GS3097	163	11.3	4.9	8	GS2954	232	9.6	4.9
9	GS2375	152	10.8	4.8	9	GS1324	214	10.4	6
10	GS10424	145	11.2	5.2	10	GS2303	212	7.5	6.2

The last two rows in the table (Fig. 7, right) provide a comparison of incidence of the pathological feature in the cluster and in the subset Pos of all the 98 liver cancer cases. All subjects in C1 are positive to Virus C and negative to Virus B. They belong mostly to the elderly group. Note that they are all males but for sample 116: this subject is detected as a singleton by the DTW-based clustering focused on this subgroup.

It is interesting to backtrack to the gene expressions for the samples in the identified subgroup. In particular we may study the best ranked genes according the BioDCV profiling for C1 in comparison to the dataset strat-

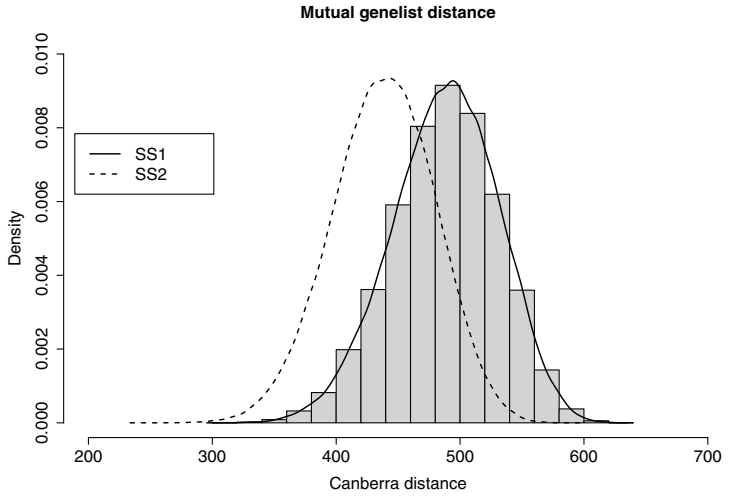
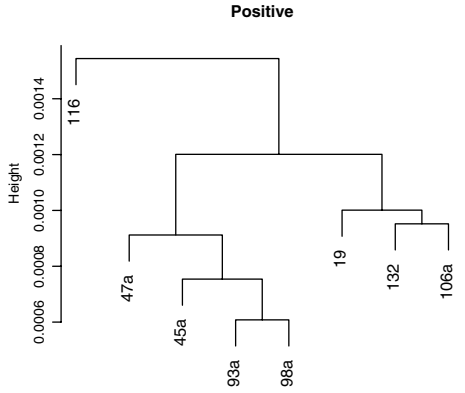


Fig. 6. Histogram and density plots of the mutual Canberra distances among the 400 gene lists obtained on the complete dataset SS1 (solid line) and on the outlier shaved dataset SS2 (dashed line).



n	v-b	v-c	a	s	c-a	c-b	cir
132	0	1	1	1	1	0	0
19	0	1	1	1	0	1	1
45a	0	1	1	1	0	1	1
47a	0	1	1	1	1	0	1
106a	0	1	0	1	0	1	1
93a	0	1	0	1	1	0	0
98a	0	1	1	1	1	0	1
116	0	1	0	0	1	0	1
C1 %	0	**100**	62	88	62	38	75
Pos %	18	65	47	80	73	26	51

Fig. 7. The dendrogram of the C1 subgroup (left) and table of clinical features of the corresponding samples (right): sample label (*n*), virus-type b (v-b) and c (v-c), age > 65 (a), sex (s), child score a (c-a) and b (c-b), and presence of cirrhosis (cir). The last two lines show percentage of values equal to 1 in the C1 subgroup and in the set Pos of all the cancer samples (shaved dataset).

ification. In Fig. 8 the expressions of 5 of the best 20 genes (top-20 list) are considered aggregating on control cases (blue), cancer cases (red), C1 (orange), and those (yellow) sharing some of the same clinical features (VB = 0, VC = 1, $S = 1$) of the samples belonging to C1. The genes GS3244, GS3309, GS6487, GS11817, GS189 provide a better separation between C1 and controls than for all cancer cases. Finally, the subset defined by the clinical feature pattern is closer to C1 than the set of all cancer cases for all these top ranked genes.

4. Conclusions

In this study, we have shown that the high-throughput structure of complete validation schemes for gene profiling may support new modes of semisupervised analysis based on the concept of sampletracking profiles. With the availability of covariate features of pathological relevance, there is a challenging opportunity for subtype discovery methods, with applications for the search of biomarkers in a very interesting class of studies.

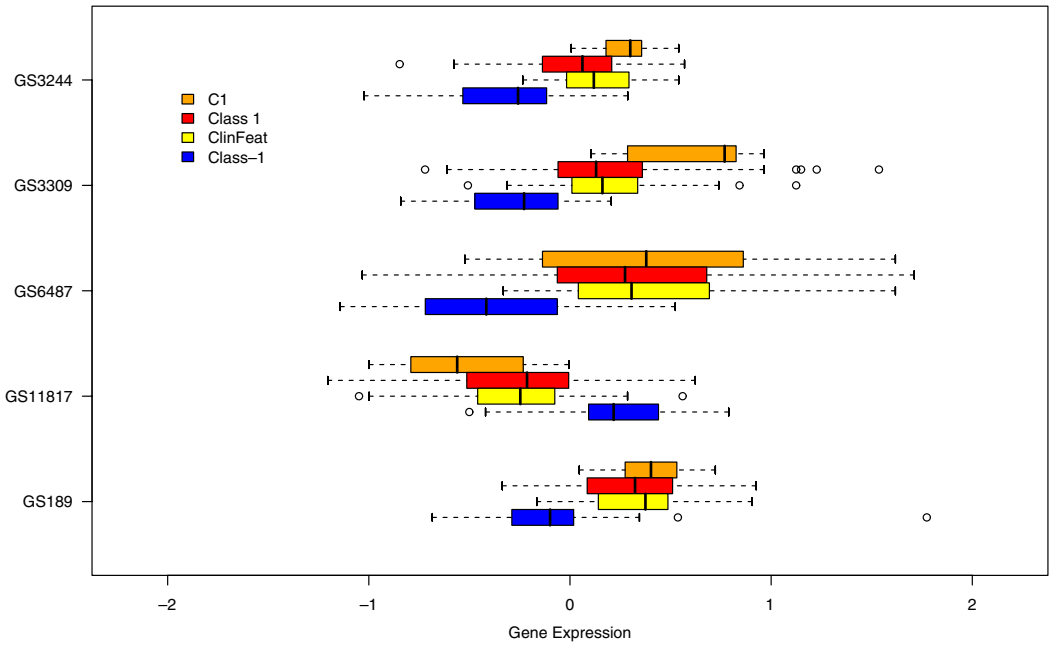


Fig. 8. Comparing the expressions of 5 top ranked genes for different subgroups: C1 (orange), all liver cancer (red, class 1), those sharing some a pattern of pathological features with C1 (yellow), and controls (blue, class-1). (For better interpretation of the figure in colour, the reader is referred to the web version of this article.)

Acknowledgements

Research partially financed by the AIRC-BICG grant. We thank S. Dzerosky for helpful indication of this application. We particularly thank the Egrid Project at ICTP Trieste and R. Flor at FBK-irst for guidance in developing the Grid implementation of BioDCV.

References

[1] T. Whistler, E. Unger, R. Nisenbaum, S. Vernon, Integration of gene expression, clinical, and epidemiologic data to characterize Chronic Fatigue Syndrome, J. Translation. Med. 1 (10) (2003), doi:10.1186/1479–5876–1–10.
[2] J. Semeiks, A. Rizki, M. Bissell, I. Mian, Ensemble attribute profile clustering: discovering and characterizing groups of genes with similar patterns of biological features, BMC Bioinf. 7 (147) (2006), doi:10.1186/1471–2105–7–147.
[3] T. Venkatesh, H. Harlow, Integromics: challenges in data integration, Genome Biology 3 (8) (2002) Reports 4027.1–Reports4027.3.
[4] M. Tyers, M. Mann, From genomics to proteomics, Nature 422 (2003) 193–197.
[5] E. Schadt, S. Monks, S. Friend, A new paradigm for drug discovery: integrating clinical, genetic, genomic and molecular phenotype data to identify drug targets, Biochem. Soc. Trans. 31 (2003) 437–443.
[6] O. Gevaert, F. Smet, D. Timmerman, Y. Moreau, B. Moor, Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks, Bioinformatics 22 (14) (2006) 184–190.
[7] R. Simon, M. Radmacher, K. Dobbin, L. McShane, Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification, J. Natl. Cancer Inst. 95 (2003) 14–18.
[8] D. Albanese, Biodcv: A distributed computing system for the complete validation of gene profiles, M.Sc. Thesis. University of Trento (2005).
[9] C. Furlanello, M. Serafini, S. Merler, G. Jurman, Semi-supervised learning for molecular profiling, IEEE/ACM Trans. Comput. Biol. Bioinf. 2 (2) (2005) 110–118.
[10] J. Sese, Y. Kurokawa, M. Monden, K. Kato, S. Morishita, Constrained clusters of gene expression profiles with pathological features, Bioinformatics 20 (17) (2004) 3137–3145.
[11] E. Bair, R. Tibshirani, Semi-supervised methods to predict patient survival from gene expression data, PLoS Biol. 2 (4) (2004), doi:10.1371/journal.pbio.0020108.
[12] C. Furlanello, M. Serafini, S. Merler, G. Jurman, Entropy-based gene ranking without selection bias for the predictive classification of microarray data, BMC Bioinf. 4 (54) (2003), doi:10.1186/1471–2105–4–54.

[13] R. Simon, L. McShane, G. Wright, E. Korn, Y. Radmacher, M. Amd Zhao, Design and Analysis of DNA Microarray Investigations, Statistics for Biology and Health, Springer, 2004.
[14] C. Ambroise, G. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proc. Natl. Acad. Sci. USA (99) (2002) 6562–6566.
[15] C. Cortes, V. Vapnik, Support–vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[16] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.
[17] C. Furlanello, M. Serafini, S. Merler, G. Jurman, An accelerated procedure for recursive feature ranking on microarray data, Neural Networks 16 (5–6) (2003) 641–648.
[18] J. Aach, G. Church, Aligning gene expression time series with time warping algorithms, Bioinformatics 17 (6) (2001) 495–508.
[19] R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, available from: http://www.R-project.org (2005).