Journal of Biological Systems, Vol. 20, No. 4 (2012) 349–375

DOI: 10.1142/S0218339012400025

World Scientific
www.worldscientific.com

GENE EXPRESSION DATA CLASSIFICATION COMBINING HIERARCHICAL REPRESENTATION AND EFFICIENT FEATURE SELECTION

MATTIA BOSIO, PAU BELLOT, PHILIPPE SALEMBIER
and ALBERT OLIVERAS-VERGÉS

Department of Signal Theory and Communications, Technical University of Catalonia UPC, Campus Diagonal Nord, building D4 Jordi Girona 1-3 08034 Barcelona
mattia.bosio@upc.edu

A general framework for microarray data classification is proposed in this paper. It produces precise and reliable classifiers through a two-step approach. At first, the original feature set is enhanced by a new set of features called metagenes. These new features are obtained through a hierarchical clustering process on the original data. Two different metagene generation rules have been analyzed, called *Treelets* clustering and *Euclidean* clustering. Metagenes creation is attractive for several reasons: first, they can improve the classification since they broaden the available feature space and capture the common behavior of similar genes reducing the residual measurement noise. Furthermore, by analyzing some of the chosen metagenes for classification with gene set enrichment analysis algorithms, it is shown how metagenes can summarize the behavior of functionally related probe sets. Additionally, metagenes can point out, still undocumented, highly discriminant probe sets numerically related to other probes endowed with prior biological information in order to contribute to the knowledge discovery process.

The second step of the framework is the feature selection which applies the Improved Sequential Floating Forward Selection algorithm (IFFS) to properly choose a subset from the available feature set for classification composed of genes and metagenes. Considering the microarray sample scarcity problem, besides the classical error rate, a reliability measure is introduced to improve the feature selection process. Different scoring schemes are studied to choose the best one using both error rate and reliability. The Linear Discriminant Analysis classifier (LDA) has been used throughout this work, due to its good characteristics, but the proposed framework can be used with almost any classifier. The potential of the proposed framework has been evaluated analyzing all the publicly available datasets offered by the Micro Array Quality Control Study, phase II (MAQC). The comparative results showed that the proposed framework can compete with a wide variety of state of the art alternatives and it can obtain the best mean performance if a particular setup is chosen. A Monte Carlo simulation confirmed that the proposed framework obtains stable and repeatable results.

Keywords: Microarray classification; metagenes; hierarchical representation; Treelets; feature selection; LDA.

1. Introduction

Microarrays are an important technology for the exploration of biological data, developed to allow researchers to gather a very large number of gene expressions

simultaneously. Microarrays are commonly used to study the transcriptional responses of cells and tissues. By measuring the mRNA concentration levels, inference about the phenomena inside a cell can be made[27]. A typical microarray experiment measures thousands, or tens of thousands, gene expressions with a relatively small sample number, at most a few hundreds. The relative difference between the sample number and the gene expression number makes the microarrays a typical example of sample scarcity, and it is commonly addressed as the curse of dimensionality[1].

Microarray classification is a complicated task, not only due to the high dimensionality of the feature set (thousands of gene expressions), but also due to the lack of a known data structure which limits the applicability of signal processing techniques, and due to residual measurement noise even after data normalization[11,16].

In this article the microarray classification task is studied and a novel global approach is proposed. It tackles the problems that make the prediction difficult with a two-step framework to obtain a precise and reliable classifier.

A plethora of algorithms has been presented in the literature to address the classification problems and produce reliable classifiers[27,8]. In almost every case, feature selection algorithms have been applied to reduce the impact of the feature number. In the proposed framework, three main classification issues (high feature number, lack of structure and noise) are addressed by a two-step approach. In a first phase, a structure from the data is inferred by the application of a hierarchical clustering algorithm inspired by Lee's work in[18], assigning a binary tree structure to unordered data. Afterwards, the original feature set is enriched by newly created variables called *metagenes*, one for each node of the tree structure. Objective of this first step is to give the data a structure and, from that, to produce new features summarizing the common traits of gene clusters with a noise reducing effect.

The second step in the proposed framework is the development of an effective wrapper feature selection method to choose a small set of features, including genes and *metagenes*, with which a final classifier is trained. As a result, the final classifier only relies on a reduced set of features and must be able to catch the key elements that differentiate between the sample classes. The benefit of hierarchical clustering by extracting interesting new variables to be used for classification has been studied in different works in the literature[12,7,18]. In[18] the *Treelets* multi-scale representation is used for microarray classification as a feature extraction and dimensionality reduction tool prior to classification combined with a filter feature selection phase, obtaining interesting results. In this paper, the produced *metagenes* are added to the original feature set as a new group of features available for selection in the classifier building process. Furthermore, the proposed framework offers additional benefits concerning the results interpretability, the hypothesis generation and the model flexibility. The results interpretation is eased by the inferred structure which can group various genes with common numerical behavior and common shared biological knowledge, after an analysis with enrichment tools such as DAVID[14] or Gene Set Enrichment Analysis[25] (GSEA). The same enrichment analysis can help

in hypothesis generation to promote further studies to find the biological relevance of selected features without previous knowledge (e.g. a chosen probe set with no associated gene symbol in DAVID or GSEA analysis). Finally the model flexibility is an advantage if the chosen features are unavailable for further clinical testing, for example due to economic reasons or due to antibodies unavailability for immunohistochemistry [26]. The proposed framework allows easy finding of alternative features by looking at the inferred data structure.

A suitable wrapper feature selection algorithm has been chosen in this work, which is called improved sequential floating forward selection (IFFS)[21]. The proposed framework introduces two main contributions for the feature selection task. The first one is the introduction of a reliability parameter, which is combined with the commonly used error rate to evaluate the predictive properties of a single classifier. The second contribution concerns the use of both the error rate and the reliability at the same time to evaluate the predictive performance of a classifier. In[2], a two-level lexicographic sorting has been adopted and it showed good results when applied on small samples datasets without an independent validation test set. In this work, three scoring criteria have been evaluated on bigger datasets, endowed with an independent validation dataset, to more precisely assess the predictive ability of the proposed approach. As well as the lexicographic sorting, two additional scoring rules have been tested. Both combine the two sources of information, error rate and reliability, into a scalar value representative of the classification performance.

The potential of the proposed framework, considering the setup variants depending on the scoring rule and the *metagene* generation process, has been evaluated on four publicly available datasets, classified with eight different endpoints. The datasets are provided by the Micro Array Quality Control study, phase II, (MAQC) as a common ground to test classification algorithms[24]. The analyzed data are all the publicly available datasets of the MAQC II study data, for more details about all the datasets analyzed in the MAQC study, refer to[24].

This paper is organized as follows: in Section 2, the feature set enhancement algorithm is introduced and the two implemented alternatives are detailed. In Section 3, the feature selection algorithm is explained. The reliability measure and the applied scoring rules are described there. In Section 4, the MAQC datasets are described and the whole experimental protocol is detailed. In Section 5, the experimental results are reported and discussed, while in Section 6, the conclusions of this work are reported.

2. Feature set enhancement

The feature set enhancement is the first step in the proposed framework. The aim of this phase is to apply an unsupervised learning technique to infer a hierarchical structure from the original data. Then, such structure is used to produce new features called *metagenes* that will be added to the original gene expression values.

This processing step is included because the original data have no a priori structure and suffer from residual noise on the measured values.

The newly created *metagenes* expand the feature space and can improve, after a proper feature selection process, the classification ability. *Metagenes* can reduce the residual noise by summarizing local clusters of similar genes. In order to infer a hierarchical structure from the data, a hierarchical clustering algorithm has been applied based on Lee's work in[18]. Each resulting *metagene* is obtained as a linear combination of the original feature set. A hierarchical clustering algorithm has been chosen for organizing and grouping the dataset variables into an easily interpretable description of the data structure.

This elaboration step focuses its attention on extracting new variables from the gene expression values. This approach obtains relations among gene groups while building new features from the original data. This strategy has already been adopted by algorithms like *Tree Harvesting*[12], or *Pelora*[7], where the benefit of hierarchical clustering to extract interesting new variables to expand the original feature set is highlighted. The possibility to summarize gene clusters in a single variable representing the common behavior as input for the classifier brings many advantages. At first, the summarizing feature is easily interpreted as a combination of the genes in the cluster. Relations among these genes can be inferred, for example two genes may be involved in the same biological process due to their high numerical similarity. One of the most important advantages is the noise reduction as a side effect of the linear combination of similar genes particularly at the lower levels of the hierarchical tree. The common behavior of a gene cluster is encoded into a representing *metagene*. The *metagene* profile emphasizes the common traits of a gene group, simultaneously reducing the residual noise on the measured values. Furthermore, *metagenes* are more robust to chance because a *metagene* useful for classification is less likely to be a product of chance rather than an individual gene.

The requirements to produce a hierarchical structure, organized into a tree, are to define an aggregation rule to form the clusters (i.e. a similarity metric) and to specify a generation rule for the *metagene* calculation as a combination of individual genes. In this work, the chosen clustering process is a bottom-up, pairwise hierarchical clustering based on Lee's work in[18], where an adaptive method for multi-scale representation and eigen-analysis of the data called *Treelets* is presented.

2.1. *Metagene generation algorithm*

The feature set enhancement phase is based on a bottom-up, pairwise hierarchical clustering algorithm, whose general pseudocode is outlined in Figure 1. The algorithm on which it is based is called *Treelets*. It is an iterative process in which, at each level, the two most similar features are replaced by two newly created features, a coarse-grained approximation feature and a residual detail feature. Such method outputs a multi-scale representation of the original data allowing a perfect reconstruction of the original signal. In our case, the main interest lies in finding rep-

Original feature set $\underline{\underline{G}}_0 = \{\underline{g}_1, \ldots, \underline{g}_p\}$
Active feature set $\underline{\underline{F}} = \underline{\underline{G}}_0$
Metagene set $\underline{\underline{M}} = \emptyset$

For $i = 1 : p\text{-}1$

1. *Calculate pairwise similarity metric* $d(\underline{f}_a, \underline{f}_b)$ *for all features in* $\underline{\underline{F}}$
2. *Find a,b* : $d(\underline{f}_a, \underline{f}_b) = \max(d(\cdot, \cdot))$
3. *New metagene* $\underline{m}_i = g(\underline{f}_a, \underline{f}_b)$ *generation:*

$$\underline{m}_i = \alpha_a \underline{f}_a + \alpha_b \underline{f}_b = \sum_{i=1}^{p} \beta_i \underline{g}_i;$$
$$\underline{\alpha} \in \Re^2 \quad \underline{\beta} \in \Re^p$$

   Each metagene can be seen either as a combination of its two child features $\{\underline{f}_a, \underline{f}_b\}$ *or as a linear combination of all the original features* $\underline{g}_i$
4. *Add the new metagene to the active feature set*

$$\underline{\underline{F}} := \underline{\underline{F}} \cup \{\underline{m}_i\}$$

5. *Remove the two features* $\underline{f}_a, \underline{f}_b$ *from the active feature set*

$$\underline{\underline{F}} := \underline{\underline{F}} \backslash \{\underline{f}_a, \underline{f}_b\}$$

6. *Join the metagene* $\underline{m}_i$ *to the metagene set*

$$\underline{\underline{M}} := \underline{\underline{M}} \cup \{\underline{m}_i\}$$

end

Define the new expanded feature set:
$\underline{\underline{F}} = \underline{\underline{G}}_0 \cup \underline{\underline{M}}$ *as the union of metagenes and original gene expression profiles.*

Fig. 1. Pseudocode for the feature set enhancement algorithm adopted in this work.

resentative features for gene clusters and in generating a hierarchical tree structure. To this end, at each level, the detail feature is discarded, while the coarse-grained approximation feature is chosen as the *metagene*. Afterwards, the newly created *metagene* is used as a feature to be compared in the next iterations. As outlined in Figure 1, the two main elements defining the final output are the similarity metric $d(\underline{f}_a, \underline{f}_b)$ and the *metagene* generation rule $g(\underline{f}_a, \underline{f}_b)$. In the current work, the generation rule adopts the strategy proposed in[18], by applying a local Principal Component Analysis (PCA) on the two child nodes for each *metagene* to be calculated.

Two variants of the *metagene* generation algorithm have been studied in this work. They are called *Treelets* clustering and *Euclidean* clustering, and are respectively detailed in 2.1.1 and 2.1.2.

2.1.1. *Treelets clustering*

The first studied alternative is called *Treelets* clustering and it owes its name to the *Treelets* algorithm from[18]. As well as the basic hierarchical clustering algorithm, it shares the similarity metric with Lee's work in[18]. The feature similarity is measured in terms of the Pearson correlation. It is a normalized correlation measure between two features, it is defined as in Eq.(2.1) for two generic feature vectors $\underline{f}_a$ and $\underline{f}_b$.

$$d(\underline{f}_a, \underline{f}_b) = \frac{< \underline{f}_a, \underline{f}_b >}{\|\underline{f}_a\|_2 \|\underline{f}_b\|_2} \quad \in [-1 \dots 1] \tag{2.1}$$

This criterion captures the profile shape similarity of two features, so it is invariant to any positive scale factor: $d(\underline{f}_a, \underline{f}_b) = d(k\ \underline{f}_a, \underline{f}_b)$ with $k \in \Re^+$. The Pearson correlation assumes value equal to 1 when two features have the same exact pattern. A correlation value of -1 implies a perfect profile anticorrelation, defining the farthest possible point in the similarity space spanned by the Pearson correlation.

An observation must be done about the *metagene* generation. *Metagenes* are produced via a PCA, which is a unitary transform and it generates an orthonormal basis for the spanned space. The coefficient vector has a l^2 norm equal to one, meaning that $\|\underline{\alpha}\|_2 = 1$ and $\|\underline{\beta}\|_2 = 1$, where $\underline{\alpha}$ is the coefficient vector applied to the sons of a *metagene*, while $\underline{\beta}$ is the coefficient vector which can generate the same *metagene* as a linear combination of all the individual genes (with nonzero values for the leaves of the subtree hanging from the *metagene* node). This translates into producing *metagenes* of growing dynamic range as the number of clustered genes grows as showed in Figure 2 where, a toy example with an initial feature set of three identical genes is considered. In this example, as $\underline{f}_1 = \underline{f}_2$, we can see that $\underline{m}_1 = \sqrt{2}\underline{f}_1$ and $\underline{m}_2 = \sqrt{3}\underline{f}_1$. The growing dynamic range in *Treelets* clustering is due to the PCA characteristic of being energy conservative and to the choice of the approximation feature as *metagene*.

2.1.2. *Euclidean clustering*

The second *metagene* generation technique is called *Euclidean* clustering. The iterative process is the same as in Figure 1 but it introduces some changes in both the similarity metric $d(\underline{f}_a, \underline{f}_b)$ and in the generation rule $g(\underline{f}_a, \underline{f}_b)$ with respect to the *Treelets* clustering.

The *Euclidean* clustering adopts the negative Euclidean distance as similarity metric defined as in Eq.(2.2). Such measure has its maximum in zero when the two compared features are equal, while its minimum is unlimited. It has been chosen because it represents a different point of view with respect to the Pearson correlation. The Euclidean distance captures the point-wise closeness rather than the profile shape similarity.

$$d(\underline{f}_a, \underline{f}_b) = -\|\underline{f}_a - \underline{f}_b\|_2 \tag{2.2}$$

Changing the similarity measure implies a modification in the *metagene* generation rule too. Due to the modification of the dynamic range introduced by the iterative application of the PCA transformation, the produced *metagenes* have an energy amount proportional to the number of clustered genes. This *metagenes* increasing energy is a problem if the Euclidean distance is considered because $d(\underline{f}_a, \underline{f}_b) \neq d(k\,\underline{f}_a, \underline{f}_b)$, thus biasing the merging phase to initially join genes with genes and only afterwards to include the *metagenes* in the merging process. To properly compare genes with *metagenes*, the latter must be a pure weighted average of the clustered genes. As a result, when a *metagene* $\underline{m}_x$ is created, two versions of it are defined. The first one, $\underline{m}_x$, is the one described in the *Treelets* case, that is the first principal component from the local PCA transformation, while the second, $\underline{m}_{xscaled}$ is a scaled version of the former. The scale factor is the l^1 norm of the coefficient vector $\underline{\beta}$: $\underline{m}_{xscaled} = \underline{m}_x / \|\underline{\beta}\|_1$. In this way, the scaled version of the *metagene* results to be a pure weighted average of the merged genes and it is used for the pairwise similarity calculation as *metagene*. Note however that the non-scaled version of the *metagene* is not eliminated. It is maintained and used when a new *metagene* is built from $\underline{m}_x$ to preserve the energy distribution among the individual components.

An illustration of the *metagene* generation process is included in Figure 2. The obtained *metagenes* through the sole use of the local PCA transformation are scaled weighted averages of the original features. As discussed in Section 2.1.1, the scaling factor is proportional to the number of joined genes. It can be observed how the introduction of the scaled versions $\underline{m}_{xscaled}$ allows a direct comparison between the *metagene* and the original features. For example, one can see how the non-scaled *metagene* does not produce a zero Euclidean distance $d(\underline{m}_1, \underline{f}_1) = (\sqrt{2} - 1)\|\underline{f}_1\|_2$, while $d(\underline{m}_{1scaled}, \underline{f}_1) = 0$. The desired feature, in this case, must be an exact replica of the original features, as they are all equal to each other. In Figure 2, the usefulness of preserving the $\underline{m}_x$ version is showed in the creation of the second *metagene*. Without this non-scaled *metagene*, the obtained feature would have been $\underline{m}_2 = 1/2\,\underline{m}_1 + 1/2\,\underline{f}_3$, obtaining an undesired bias towards the $\underline{f}_3$ feature. If this process is repeated in a real-case scenario, the result will be *metagenes* resembling more and more the last added components rather than the first joined ones, frustrating the original meaning of the whole process to capture the common behavior of gene clusters.

With both the *Euclidean* and *Treelets* clustering techniques, a new feature set is produced. It is obtained as a hierarchical tree, in which the non-leaf nodes are formed by *metagenes*. Subsequently, the newly created features are added to the original feature to improve the prediction capabilities. The *metagenes* can improve the next steps in the classification framework in different ways. They expand the feature space, thus increasing the probability to find suitable features for classification and they reduce the noise impact when clustering groups of similar genes.

Initial feature set of three equal genes

$$\underline{\underline{F}}_0 = \{\underline{f}_1, \underline{f}_2, \underline{f}_3\} \text{ with } \underline{f}_1 = \underline{f}_2 = \underline{f}_3$$

Two metagenes will be created

1. *metagene* $\underline{m}_1$ *joining* $\underline{f}_1$ *and* $\underline{f}_2$

$$\underline{m}_1 = \sqrt{1/2}\underline{f}_1 + \sqrt{1/2}\underline{f}_2$$
$$\underline{m}_{1scaled} = 1/2\underline{f}_1 + 1/2\underline{f}_2$$

2. *metagene* $\underline{m}_2$ *joining* $\underline{m}_1$ *and* $\underline{f}_3$

$$\underline{m}_2 = \sqrt{2/3}\underline{m}_1 + \sqrt{1/3}\underline{f}_3$$
$$\underline{m}_2 = \sqrt{1/3}\underline{f}_1 + \sqrt{1/3}\underline{f}_2 + \sqrt{1/3}\underline{f}_3$$
$$\underline{m}_{2scaled} = 1/3\underline{f}_1 + 1/3\underline{f}_2 + 1/3\underline{f}_3$$

Scaled versions $\underline{m}_{1scaled}$ *and* $\underline{m}_{2scaled}$ *are used to define the similarity with the Euclidean distance because they preserve the components dynamic ranges. These versions are then used as metagenes, enhancing the original feature set.*

The non-scaled versions, $\underline{m}_1$ *and* $\underline{m}_2$ *preserve the energy distribution among the elementary components. They are used as child nodes when a new metagene is built either from* $\underline{m}_{1scaled}$ *or* $\underline{m}_{2scaled}$.

Fig. 2. Example of metagene creation with *Euclidean* clustering.

3. Feature selection algorithm

After the metagene creation step, the feature set has been enriched with a whole new group of alternatives. The main problem now is to choose an appropriate subset for classification. This task is more compelling after the metagene introduction, which has almost doubled the features number, while the sample number is still the same.

In this work, a wrapper algorithm has been implemented, because of its multivariate potential and of its ability to be used with virtually any classifier[17,8]. The chosen algorithm is called Improved Sequential Floating Forward Selection (IFFS)[21]. It is an evolution of the Sequential Floating Forward Selection (SFFS)[23] which adds a replacing step to the original algorithm. The choice has fallen on the IFFS algorithm since our aim was to find an algorithm to choose good subsets and able to reproduce its results over time if the initial conditions do not change. IFFS adds more evolution ability to the SFFS algorithm and evolution has proven to be useful in feature selection when dealing with high dimensional feature sets. Evolutionary search algorithms like the genetic algorithm[15], genetic programming[10] or NSGAA II[6] have shown good predictive power thanks to the mutation possibility of the selected feature set throughout the learning process. However, these algorithms

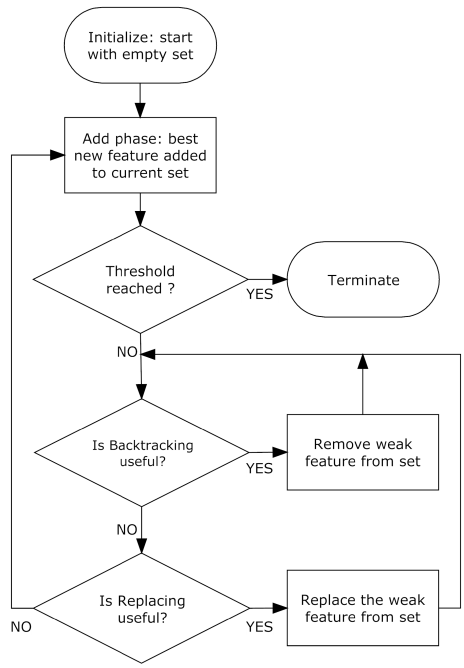

Fig. 3. IFFS feature selection algorithm.

are for their own nature random and strongly depend on the initial population, thus limiting the solution space. Many parallel runs are then needed to obtain a final solution as stated in[23], where it is also noticed that evolutionary algorithms performances tend to degrade as the feature number increases.

In Figure 3, the flowchart of the IFFS algorithm is illustrated. The search process starts with an empty set and ends when a threshold value is reached. The threshold is either the maximum accepted number of features or a maximum number of iterations in case the algorithm has entered in an infinite loop. After the initialization, the selection process enters in a loop of tasks. At first there is the add phase, where all those features not yet selected are added to the current set one at a time. For each one, a classifier is trained and the correspondent classification score $J(\cdot)$ is calculated. The feature obtaining the best $J(\cdot)$ is added to the current set. Then, if the threshold has not been reached yet, the algorithm starts a backtracking phase. In this step the weakest feature in the subset (i.e. the feature whose elimination implies the minimum performance loss or the maximum performance gain) can be eliminated. If the elimination improves $J(\cdot)$, the weak feature is removed and a new backtracking phase is performed. Otherwise, the algorithm looks for substituting one feature in the replacing phase. In this last phase, a substitute is chosen for each feature in the current set via an analysis similar to the add phase. If the best substitution has proven useful, (i.e. the $J(\cdot)$ value with the substitution is better

than without), the current set is updated and a new backtracking phase takes place. Otherwise, the algorithm goes back to the add phase.

3.1. *Feature ranking criterion*

Throughout the feature selection process a major role is played by the $J(\cdot)$ score. It estimates the classification ability of the proposed strategy. Being IFFS a wrapper algorithm, the classifier rule is iteratively applied in each step. The chosen classifier in this work is the Linear Discriminant Analysis (LDA), due to its good properties of simplicity, interpretability and precision[3,24]. During the feature selection LDA is applied multiple times and, in every case, a $J(\cdot)$ score is extracted.

The most popular $J(\cdot)$ score is the classification error rate. When sample scarcity is not a problem, usually the error rate is estimated on thousands of samples and a reliable (generalizable) estimation is obtained. In the current microarray classification scenario, no such sample abundance is available, so different error rate estimation techniques have been developed like cross-validation, bagging, boosting or bolstered resubstitution[3,9,5]. Among the possible alternatives, the ten times five fold stratified cross validation estimator[20] has been chosen for the training phase.

3.1.1. *The reliability parameter*

To integrate the error rate as a fitness estimator in a small sample scenario, an additional value is introduced to define the $J(\cdot)$ score: the reliability. This parameter considers that a feature obtaining well separated classes is better than a feature in which the class separation is very thin. It tries to transfer the univariate t-tests concept (i.e. give more importance to features having large mean class separation and small intra class variance) to a multivariate scenario by considering the classifier point of view.

The reliability parameter, r, measures a weighted sum of sample distances from the classification boundary as goodness estimation. It is calculated on the test set samples and the final value is the mean through the cross validation iterations. Inside each iteration, the reliability parameter is defined as in Eq.(3.1) for a binary classifier. In Eq.(3.1), n_{test} is the test set dimension, c_l is the class of sample l (it can be 1 or 2), and $p(c_l)$ is the probability of class c_l in the test set. The value d_l is the Euclidean distance of sample l from the classifier boundary with positive sign in case of correct classification or negative sign otherwise.

$$r \;=\; \frac{1}{n_{test}\cdot \hat{\sigma}_d} \sum_{l=1}^{n_{test}} \frac{d_l}{p(c_l)} \tag{3.1}$$

Finally, $\hat{\sigma}_d \;=\; \sqrt{\frac{\hat{\sigma}_1^2}{n_1} + \frac{\hat{\sigma}_2^2}{n_2}}$, is an estimation of intra class variance of the sample distances from the classification boundary. In order to get a more complete estimation, the intra-class variance is estimated using all the samples from both the training and the test sets; n_1 and n_2 are the number of samples in class 1 and 2 respectively.

The $\hat{\sigma}_d$ parameter is defined as in the independent two-sample t-test denominator with classes of different size and variance. In detail $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the estimated variances of sample distances from the boundary for all the samples of class 1 and 2 respectively. Dividing r by $\hat{\sigma}_d$ guarantees that it is invariant to a scaling factor, thus $r(\underline{f}_a) = r(k\underline{f}_a)\ \ k \in \Re^+$. Dividing by $p(c_l)$ assigns to each class the same relative weight and it is useful when the class distribution is highly skewed. The reliability value, $r \in [-\infty\ \infty]$, is positively influenced by large mean class separation in the perpendicular direction to the classifier boundary, and by small intra class data variance. On the other hand, it is negatively influenced by the error number and intensity, making greater errors inducing greater penalizations.

3.1.2. *Scoring rules*

The final $J(\cdot)$ value is determined by both the reliability measure introduced in the previous section and the error rate along the cross validation iterations. A particular classifier is ranked to be better than another if its $J(e, r)$ score is higher, where e is the error rate and r is the reliability. The score definition is the key point for the feature selection. An effective scoring rule using error rate and reliability can highly improve the final classifier. In this work, three different scoring rules have been studied to explore several combinations of e and r.

The first scoring scheme is a two-step ranking process introduced in[2]. Features are firstly sorted by increasing error rate value, and then, reliability is taken into account to break ties among features with equal error rate. This criterion produces a lexicographic sorting of the features in which the reliability parameter has a secondary role. The lexicographic sorting has obtained interesting results classifying small, publicly available datasets[2], reducing the number of needed features to get a 0 estimated error rate with respect to state of the art alternatives. Nevertheless, the lexicographic sorting is a rigid scheme. The benefits of the introduction of the reliability parameter can fade when the test set cardinality grows. In such a case it is less probable to have error ties, thus reducing the reliability contribution.

To make better use of the reliability information, two scoring rules have been studied. Both of them unify in a scalar value the two sources of information. The proposed score definition rules are influenced by error rate and reliability simultaneously, allowing a feature with higher reliability and slightly higher error rate to be considered better than another with poor reliability but with a smaller error rate. This flexibility can be useful for small sample datasets like microarrays. The first of the two rules compares features in terms of the reliability value, properly penalized depending on the estimated error rate. The aim of the penalization is to introduce a fixed penalization factor to the reliability value for a constant error difference. Such a behavior is obtained with an exponential penalization to the reliability value as detailed in Eq.(3.2) where r is the reliability value, e is the error rate value, and α is a penalization parameter.

$$J\ =\ r \cdot \exp\left(-sign(r) \cdot \frac{100}{\alpha} \cdot e\right) \tag{3.2}$$

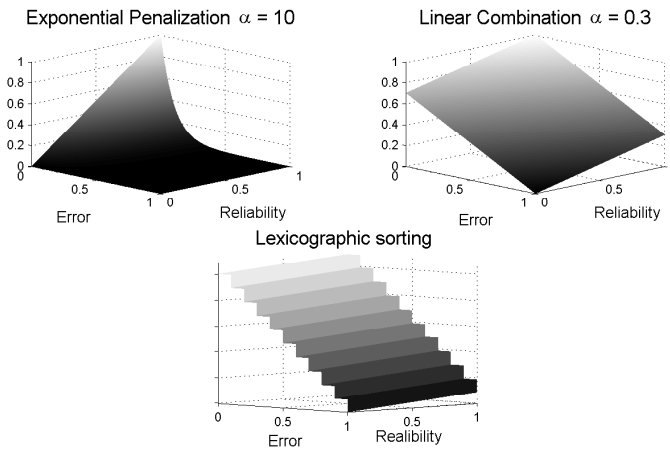

Fig. 4. Score surfaces in the error-reliability space depending on the three scoring rules.

$J(e, r)$ is a product of the reliability value with a penalization coefficient ≤ 1 with an exponential behavior depending on the error rate value. The $-sign(r)$ factor in the exponent has been included to highly penalize features with negative reliability, while the α parameter defines the steepness of the penalization. The α value defines the e^{-1} penalization interval: between two features with equal reliability value, a $\alpha\%$ difference in the error rate induces a e^{-1} penalization factor in the final score. So, when α is small, the dominant parameter is the error rate (an extreme case is when $\alpha \to 0$ the reliability has no influence at all), while when α is large the dominant parameter becomes the reliability (when $\alpha \to \infty$ the error rate is not taken into account).

The last introduced scoring rule is a linear combination of error rate and normalized reliability. The linear combination score is defined in Eq.(3.3) as a weighted sum of error rate e and a normalized reliability value $r_n = (r - \min(r))/\max(r)$, where $\min(r)$ and $\max(r)$ are respectively the minimum and the maximum values of the reliability obtained in the current search iteration. The α parameter is bounded between 0 and 1 and it defines the relative importance of reliability with respect to the error rate.

$$J = \alpha \cdot r_n + (1 - \alpha) \cdot (1 - e) \quad \alpha \in [0, 1] \tag{3.3}$$

The scoring surface has a linear trend both in the error rate and in the reliability direction. The main change with respect to the exponential penalization scoring is that, here, a constant penalization is added (not multiplied) to a constant error rate increase. To visualize how the possible score surface changes with the scoring rule, Figure 4 is introduced. It shows the assigned score values to points in the Error-Reliability space for the exponential combination, the linear combination and the lexicographic sorting.

Table 1. Microarray datasets used for classification.

Dataset	Endpoint description		Affymetrix Platform	Training set			Validation set		
				Tot	Pos.	Neg.	Tot	Pos.	Neg.
Hamner	Lung tumorigen vs. non tumorigen	A	Mouse 430.2.0	70	26	44	88	28	60
NIEHS	Liver toxicant vs. non toxicant	C	Rat 230.2.0	214	79	135	204	78	126
Breast cancer	Pre operative treatment response	D	Human U133A	130	33	97	100	15	85
	Estrogen receptor status	E		130	80	50	100	61	39
Multiple Myeloma	Overall survival milestone outcome	F	Human U133Plus2.0	340	51	289	214	27	187
	Event-free survival milestone outcome	G		340	84	256	214	34	180
	Sex of the patient	H		340	194	146	214	140	74
	Negative control, random assignation	I		340	200	140	214	122	92

From Figure 4 it can be observed how in the exponential combination case, the score has an exponential decrease along the Error dimension, while it has a linear trend in the Reliability dimension. The scores with the linear combination rule lie onto a rotated plane in the space with the rotation axis passing through the (0, 1) and (1, 0) points. It shows linear trends in both dimensions (Error and Reliability) with slopes equal to $1 - \alpha$ and α respectively. The lexicographic scoring in Figure 4 is visualized in a very coarse scenario formed by only 10 allowed error values (imagine a test set composed of 10 samples only) in order to show its behavior. It is a stairway-like surface showing how the main dimension is the Error value. Only if two features share the same error value the reliability is taken into account (it shows a linear trend in the reliability direction). Otherwise the score of a feature with smaller error rate is higher, regardless of the reliability value. From Figure 4 it can be observed how both the scoring rules combining reliability and error rate in a scalar value radically change the score surface. From a stairway-like surface (with discontinuities between possible error rates), the score surface is transformed into a continuous surface in which the reliability gains more decisional power. This change is more noticeable when the test set cardinality grows. In such a scenario, the lexicographic scoring would be like a stairway with many small steps, making the reliability parameter almost useless.

4. Experimental protocol

In this section, the datasets used to evaluate the predictive potential of classifiers built with the proposed framework are presented. Afterwards, the experimental protocol is detailed, from the data preprocessing steps to the adopted parameter setup for the classification analysis.

4.1. *Data*

The predictive properties of the proposed framework have been evaluated on a subset of the six datasets provided by the MAQC consortium[24]. There, six datasets containing 13 preclinical and clinical endpoints coded from A through M have been made available to a selected group of analysis teams[24]. Each coded endpoint represents a different sample classification implying that the same dataset can be classified following various criteria like treatment, outcome, sex, random, etc.

Out of the six original datasets, four of them have been considered in this work, corresponding to endpoints A, C to I. These datasets have been chosen because they are publicly available at http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE16716, and a detailed explanation of the endpoint composition is given in Table 1. The remaining dataset have not been analyzed because, up to now, not all the necessary data are publicly available.

The MAQC data are highly reliable, selected after a quality control process, with the specific aim to provide a common test ground for prediction algorithms. For each endpoint, two datasets are provided. The first one called the Training set is meant to be used for the classifier training, while the second one is named Validation set and it is to be used as an independent test set to validate the prediction performance. An additional advantage when analyzing the MAQC data is that a diverse collection of analysis teams has worked on the same data, following the same evaluation procedure and publishing their results[24,22,19]. An accurate benchmark of a new algorithm can be done to understand how well it performs when compared to a considerable number of state of the art alternatives.

4.2. *Experimental protocol*

The experimental setup to validate the potential of the proposed framework is outlined here. The first step is the input data preprocessing. It consists in verifying that the data are already in a logarithmic scale and, if not, simply computing the base two logarithm of the data. After that, a minimum threshold equal to $log_2 10$ is applied in order to remove small valued probe sets since they are considered unreliable[11]. Subsequently, each probe set is forced to have a zero mean.

Afterwards, each training dataset is processed to create the metagenes, generated either with *Treelets* clustering, or with *Euclidean* clustering as explained in Section 2.1. Finally, the enriched training dataset is then analyzed by the feature selection algorithm to properly choose a small number of features, genes and meta-

genes, with the algorithm explained in Section 3 to produce a prediction model to be tested on the corresponding validation dataset.

The performances have been evaluated in terms of the Matthews Correlation Coefficient (MCC) and in terms of prediction accuracy. The accuracy measures the proportion of correctly predicted samples. In is defined by:

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

where TP is the number of the true positives identified by the classifier, TN are the true negatives, FP are the false positives and FN are the false negatives. With true positive it is meant a sample categorized as positive, P, in Table1 and correctly classified as positive by the classifier. The remaining values of TN, FP and FN are consequently defined. Whilst the accuracy is the most common performance indicator, it can be not informative enough when the class distribution is highly skewed. High accuracy values can be obtained by assigning all the samples to one class only. On the contrary, the MCC is not influenced by the class distribution skewness and, for this reason it has been adopted as the principal performance measure in the MAQC study[24]. The MCC is defined by:

$$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.2}$$

The MCC can assume values from 1 (perfect classification) to -1 (perfect inverse classification).

In the final performance evaluation step, the proposed algorithm is applied to develop prediction models over the training datasets. The predictive ability is measured over the independent validation sets extracting the respective MCC and accuracy values. Inside each training phase, a ten times five-fold stratified cross validation has been adopted[20]. Furthermore, in order to reduce the cross validation variance, a different dataset partition has been applied at each iteration of the feature selection (*add, replacing or subtracting* phase).

The proposed framework has been tested in different configurations to assess its predictive potential, to confirm that the metagene addition is beneficial for classification and to verify that a scoring rule, $J(\cdot)$, using both the error rate and the reliability can improve the final result. Experiments have been conducted applying the *Treelets* and the *Euclidean* clustering algorithms. These results have also been compared with the case in which no metagenes are added to the original feature set. In this way it is possible to evaluate the best enrichment technique and to quantify the improvements resulting from the use of metagenes. For each case, the three scoring rules $J(e, r)$ have been studied alternatively. For the exponential penalization rule and the linear combination rule, three different α parameter values have been tested. The three values are chosen after a previous optimization study. In detail, for the exponential penalization case, the chosen α values are [5 10 15], while for the linear combination rule, α can be [0.05 0.10 0.15].

5. Results and discussion

The experimental results following the protocol defined in Section 4 are presented and discussed here. The proposed framework has been tested in different configurations and compared with state of the art alternatives from MAQC study[24].

In Table 2, the mean MCC and accuracy results across the analyzed endpoints from[24], A C D E F G H I, are showed. Each *datXX* expression identifies a different classifier developed by a different research group involved in the MAQC study. The *datXX* values are those whose results are reported in[24]. As it can be observed, the MCC results in Figure 2 span a range from 0.284 corresponding to *dat3*, to the 0.490 obtained by *dat24* group, while the accuracy values span from 65.43% of *Dat3* to 83.86% of *Dat20*. The best alternative is different depending on the chosen measure. This variation is linked to the class distribution skewness which can lead an algorithm to have a high accuracy but a very low or null MCC value. This is exactly what happens to *Dat20* analyzing endpoint F: it has 87.38% accuracy while MCC= 0 because it considers all the samples pertaining to a single class which corresponds to 87.38% of the validation set samples. The MCC value better evaluates the performances of the scheme, particularly in cases of uninformative classification. The I endpoint is not considered in the mean calculations because it is a negative control dataset on which algorithms should produce bad results because class memberships have been randomly defined (see Table 1). Results in Table 2 are organized by increasing MCC value along each column.

Table 2. MAQC mean MCC and mean Accuracy results

Group	*MCC*	*Accuracy*	*Group*	*MCC*	*Accuracy*
dat3	0.284	65.43%	*dat11*	0.453	75.59%
dat33	0.300	66.04%	*dat36*	0.457	79.18%
dat7	0.307	71.04%	*dat10*	0.458	78.39%
dat19	0.384	79.52%	*dat4*	0.468	81.49%
dat29	0.397	81.78%	*dat12*	0.476	82.54%
dat35	0.419	77.69%	*dat25*	0.477	80.81%
dat18	0.428	77.29%	*dat13*	0.488	80.67%
dat32	0.431	78.89%	*dat24*	0.490	81.13%
dat20	0.443	83.86%			

In Tables 3, 4 and 5 the results applying the proposed framework on the datasets from Table 1 are presented. Each table includes the results pertaining to a different scoring rule: the lexicographic sorting, the exponential penalization or the linear combination.

In Table 3, the mean MCC and accuracy values with the lexicographic scoring are showed. In each column the results corresponding to a different metagene generation method are reported: *Treelets* clustering, *Euclidean* clustering, or *None*. The *None* column corresponds to the results when no metagene has been considered. As for

Table 3. Mean results adopting the lexicographic scoring scheme

Lexicographic sorting					
Treelets		*Euclidean*		*None*	
MCC	Accuracy	MCC	Accuracy	MCC	Accuracy
0.423	77.46%	0.418	76.18%	0.381	75.48%

Table 4. Mean results adopting the exponential penalization scoring scheme

Exponential penalization						
α	*Treelets*		*Euclidean*		*None*	
	MCC	Accuracy	MCC	Accuracy	MCC	Accuracy
5	0.475	84.02%	0.457	81.57&	0.442	82.99%
10	0.495	83.95%	0.460	83.61%	0.421	82.66%
15	0.483	83.67%	0.451	83.187%	0.457	83.30%

the method reported in[24], the I endpoint is not considered in the mean calculation due to its random nature. As can be seen, the introduction of metagenes allows obtaining higher mean MCC and accuracy values, thus producing better classifiers. With the lexicographic sorting the best MCC result is 0.423, with 77.46% accuracy, if *Treelets* clustering as metagene generation method is chosen.

Table 4 contains the collected values applying the exponential penalization scoring rule. Results are organized in four columns. The left column specifies the α parameter, while the remaining three columns are organized as in Table 3. Changing the scoring rule leads to remarkably better results than those in Table 3. The simultaneous use of both the error rate and the reliability allows us to reach better performances. Here also, results with metagenes are better than without and the best result is obtained when *Treelets* clustering is adopted and α is equal to 10. Finally, the best mean MCC value is even higher than the best one of Table 2 from *Dat24*. There, the best MCC is 0.490, while here 0.495 is reached, supporting the proposed framework as an excellent alternative to state of the art methods. Concerning the accuracy values, with *Treelets* clustering and $\alpha = [5, 10]$, better results than those in Table 2 are obtained. The highest accuracy value is 84.02%, obtained with $\alpha = 5$.

In Table 5, the results relative to the linear combination score are showed. The organization is the same as in Table 4. In this case too, the metagenes have confirmed

Table 5. Mean results adopting the linear combination scoring scheme.

Linear combination						
α	*Treelets*		*Euclidean*		*None*	
	MCC	Accuracy	MCC	Accuracy	MCC	Accuracy
0.05	0.483	83.45%	0.437	81.58%	0.444	81.46%
0.10	0.468	83.31%	0.486	83.60%	0.444	81.46%
0.15	0.469	83.25%	0.486	83.19%	0.444	81.46%

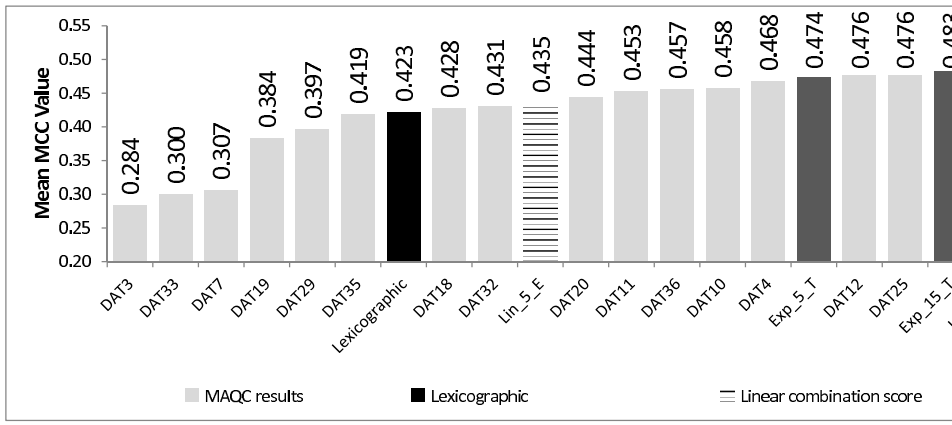

Fig. 5. Mean MCC values comparison between MAQC results and the best alternatives for the different scoring techniques adopted.

to be useful for classification because the results obtained with *Treelets* or *Euclidean* clustering are better than without. A comparison with the lexicographic sorting shows how, generally, the mean results are higher. In this case, the best mean MCC is 0.486 when *Euclidean* clustering is adopted and the α parameter is between 0.1 and 0.15, while the highest accuracy value is 83.60% when α is set to 10. Observing the results using both linear combination and exponential penalization rule, the MCC values are quite stable to small variation of the α parameter. This is a good property because there is no need to precisely optimize the alpha value.

To visualize the proposed algorithm performance in comparison with the state of the art alternatives from[24], Figure 5 and 6 are introduced. In Figure 5, the results are sorted by increasing mean MCC value and are represented as columns. The MCC value for each alternative is printed above each column, and below the corresponding method is indicated. In Figure 6, the accuracy values are presented, sorted by increasing values. All the results from Table 2 are included and painted as uniform light gray bars. For space and clarity reasons, not all the results obtained with the proposed framework are included. A selection of them is proposed representing only the best three results for the exponential penalization and for the linear combination rule, and the overall best result with the lexicographic sorting. The result from the lexicographic sorting scheme is painted as a black bar and is identified by the *Lexicographic* label. Results applying the linear combination scheme are highlighted by a black and white horizontal lines pattern. The labels start with *lin_xx_E*, where *_xx* is the α value multiplied by 100 and *_E* indicates that the *Euclidean* clustering has been used. The values corresponding to the exponential penalization scoring rules are coded as dark gray columns. The labels are coded by *exp_xx_T*, where *_xx* is the α value and *_T* indicates that the *Treelets* clustering has been adopted. As can be observed in Figure 5, the proposed framework obtains results comparable to the best state of the art alternatives when the linear combination scoring or the

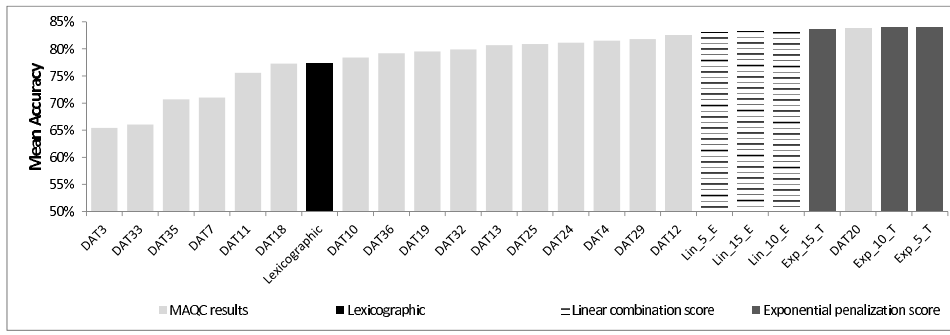

Fig. 6. Mean accuracy values comparison between MAQC results and the best alternatives for the different scoring techniques adopted.

exponential penalization rule are used. Furthermore, the *exp_10_T* obtains the best overall mean MCC value. From Figure 6 it can be observed how both *exp_10_T* and *exp_5_T* obtain better values than the compared state of the art alternatives. Furthermore, it is shown how the accuracy value too is robust to small α variations.

The mean number of chosen features by all the presented alternatives spans between 2.14 of *exp_10_T* to 3.43 of *lin_10_E*. As can be seen, the metagene creation process has almost doubled the number of features compared to the original number of genes, but the final classifier actually uses a very low number of features to perform the classification.

The proposed framework provides competitive performances with respect to the state of the art alternatives. To validate this result, a further study has been performed to assess the robustness of the obtained performance. The study consists in a 50 runs Monte Carlo analysis of the classification endpoints. This 50 run setup has been proposed to have a broader range of experiments to assess the performance stability linked to the use of cross validation as performance estimation method, which is known to have a large variance[4]. In each run, the framework setup is the same as the best alternative: *Treelets* clustering as metagene generation method and exponential penalization with $\alpha = 10$ as scoring rule for feature selection.

The results are shown in Figure 7 as a boxplot, and some results statistics are presented in Table 6. Each column in Figure 7 corresponds to a different endpoint, labeled along the x axis. For each column, an asterisk identifies the MCC value obtained in the previous study (the values used to obtain the mean MCC value in Figure 5), whose values are included in the last column of Table 6 under the label of *run 0*.

The values are collected in the same way as the *run 0* iteration, for each endpoint, classifiers have been built up to five features and the best one is then considered in the mean calculation. Results for each endpoint are presented separately to better identify how the algorithm performance can change depending on the analyzed data.

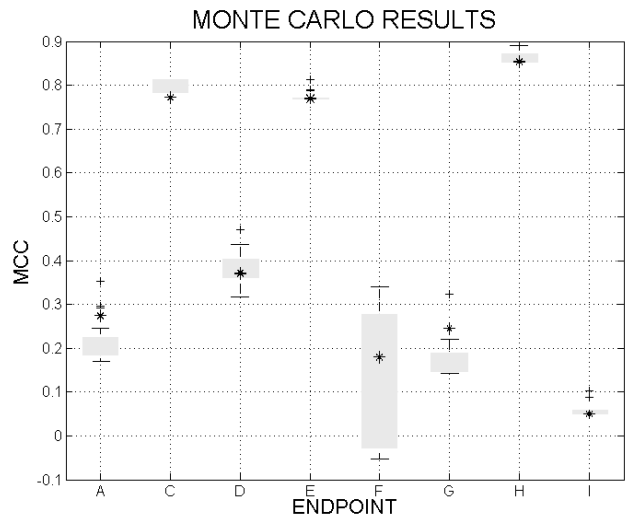

Fig. 7. Boxplot of the obtained results along the 50 independent runs. Each column corresponds to a different endpoint.

Table 6. Statistical properties of the Monte Carlo simulation.

Endpoint	*MCC*	*Accuracy*	*Run 0 MCC*	*Run 0 Accuracy*
A	0.2176	67.37%	0.2750	65.91%
C	0.7949	90.25%	0.7700	89.22%
D	0.3869	80.49%	0.3690	80.00%
E	0.7732	89.17%	0.7680	89.00%
F	0.1147	86.3%	0.1800	87.85%
G	0.1723	79.57%	0.2430	82.71%
H	0.8609	93.21%	0.8550	92.99%
I	0.0564	55.14%	0.0510	52.68%

What can be observed from both Figure 7 and Table 6 is that the results show a high robustness in the analysis of most endpoints. The obtained values are tight around their mean value for the endpoints A,C,D,E,H,G and I. The mean values are very close to the *run 0* results. The mean values in all these endpoints are slightly higher than the *run 0* results, except for the G and A endpoints, where the *run 0* results are well above the mean in the upper tail of the MCC distribution. About the F endpoint, it shows a considerably higher variability in the MCC distribution and this is mainly due to the class distribution skewness. In this endpoint, the positive class represents about the 15% of the training set. This eases the choice of uninformative features during the feature selection phase. The choice of an uninfor-

mative feature (a feature classifying all the samples to one class) biases the feature selection process and can lead to very different results. The Monte Carlo simulation confirmed that the predictive power is mainly determined by the analyzed dataset[24]. About the G endpoint, the Monte Carlo results are quite robust but inferior to the *run 0* result. This lead to think that the formerly obtained MCC value is due to some fortunate cross validation partition that allowed the selection of more useful features. Such lucky case is not unique as other runs may obtain even better values in the 50 run simulation (marked by the crosses outside the box). These results can be interpreted as "outliers" in the population distribution which underlines how the cross validation partition can change the final results. The MCC and accuracy values for the remaining endpoints are good and very consistent. This is an important feature of the proposed framework because it produces robust results. The results stability seems to be connected with the class distribution skewness which can lead to the choice of uninformative features.

To analyze how the class distribution skewness influences the prediction performances, a final test with synthetic data has been performed. The experimental process follows the protocol introduced in[13], limiting the total feature number to 1000 and the Monte Carlo iterations to 30 due to calculation time reasons. Moreover, a skewness dimension has been introduced. In[13], the two classes have the same number of samples, while here three different setups have been tested. Class 1 may represent 50%, 70% or 90% of all the available samples. For each Monte Carlo iteration, a different dataset is built, the hierarchical tree is built with *Treelets* clustering and classifiers up to 10 features are trained with the exponential penalization rule and $\alpha = 10$. For each iteration, the best classifier in terms of MCC is used for the following analysis.

Table 7 contains the summary of the study based on synthetic data. Results are organized in three subtables, one for each data generation model. In[13], three data generation models have been proposed: *Redundant, Synergetic* and *Marginal*, producing data with different distributions. Each subtable presents the results organized by size of the training set because it is an important variable about the possible overfitting, and organized by skewness, which is the main variable in this study. For each, skewness-training set size, the mean MCC value, the standard deviation of the MCC (Std), and the mean feature number (# F) are presented. The mean is calculated not only along the Monte Carlo iterations, but also along the other varying parameters. This is done for sake of synthesis and because the focus is on the skewness. In a complete algorithm assessment, many more results should be presented taking into account dependencies while varying each one of the possible parameters.

Analyzing the results in Table 7, it can be observed how in both the *Redundant* and the *Synergetic* models, the high class skewness has a negative effect over the MCC value. When the training set size is not too small, 120 and 180 samples, the MCC and skewness are inversely proportional along all the studied values. The

Table 7. Results of the study based on synthetic data. The three subtables correspond to the three different data distributions. Each subtable is organized showing the values depending on the skewness value and the different size of the training set. The *Train* column contains the size of the training set, the *MCC* columns shows the mean MCC value across the different experimental conditions and Monte Carlo iterations while *Std* and *#F* columns contain the MCC standard deviation and the mean number of selected features respectively.

Skewness - Class 1 percentage -									
	50%			*70%*			*90%*		
Redundant model									
Train	MCC	Std	# F	MCC	Std	# F	MCC	Std	# F
60	0.509	0.120	4.50	0.431	0.140	4.83	0.319	0.193	3.58
120	0.532	0.086	2.58	0.468	0.117	4.67	0.323	0.143	3.50
180	0.545	0.071	2.75	0.492	0.086	3.33	0.346	0.120	7.33
Synergetic model									
Train	MCC	Std	# F	MCC	Std	# F	MCC	Std	# F
60	0.343	0.184	4.58	0.315	0.187	2.92	0.325	0.239	1.83
120	0.431	0.133	5.42	0.351	0.143	5.83	0.266	0.221	4.75
180	0.475	0.108	5.50	0.407	0.109	5.92	0.257	0.189	6.50
Marginal model									
Train	MCC	Std	# F	MCC	Std	# F	MCC	Std	# F
60	0.509	0.159	6.58	0.555	0.150	3.25	0.490	0.193	2.17
120	0.549	0.148	7.50	0.610	0.135	4.92	0.542	0.211	2.92
180	0.570	0.139	7.75	0.631	0.137	8.17	0.572	0.193	4.00

marginal model instead presents a different behavior in which the best MCC values are constantly obtained with the intermediate skewness value and in one case, 180 training samples, the 50% case is the one obtaining the, slightly, worse performance. It can be stated how the skewness negatively influences the performance when the data have a redundant or synergetic model distribution, while with data represented by the marginal model, such direct relation does not hold.

What holds throughout all the results in Table 7 is the mean standard deviation of the MCC values. When the distribution is highly skewed (the 90% case) the standard deviation is always higher than the other cases, regardless of the mean MCC, whether it is better or worse. This is similar to what has been observed in the MAQC Monte Carlo study where the F endpoint results showed a much higher variability than any other endpoint.

About the mean selected feature number, the values span from 1.83 to 8.17. The best classifiers obtained by the proposed algorithm use also a reduced number of features in the synthetic case. This behavior helps in the training phase since the maximum feature number can be bounded by values of small magnitude.

The proposed analysis framework offers additional benefits other than the prediction accuracy thanks to the introduction of the hierarchical metagene structure. These benefits can make the use of this framework even more appealing from an analysis and hypothesis generation point of view. To illustrate them, a more detailed look to the *Run 0* results is provided. From Tables 3, 4 and 5, it is shown how using metagenes improves the classification results. Almost 15% of the chosen features in

Run 0 and in the Monte Carlo simulation are metagenes. These features extract the common behavior of gene clusters, reducing the noise thanks to the linear combination of individual genes. An example comes from the E endpoint classification, obtained applying LDA on two features: an individual probe set, 205225_at, and a metagene merging three probe sets named: 213462_at, 39548_at and 39549_at. These three probes show high pairwise correlation, higher than 90%, and, after a gene list analysis with DAVID[14] and GSEA[25], all refer to the neuronal PAS domain protein 2 (NPAS2). The chosen metagene is a summary of the NPAS2 behavior by merging three different probes expressing the same biological element.

Furthermore, the metagene structure can be useful for hypothesis formulation to infer biological relations between probe sets. An example of this potential in *Run 0* is the endpoint C analysis where the chosen metagene is formed by two elements, 13763271_at and 1379381_at. The first one, 13763271_at, corresponds to the tumor necrosis factor receptor superfamily member 14, (TNFRSF14), while no additional information can be found about the 1379381_at probe set neither in DAVID nor in GSEA. As a result, this metagene may suggest that further analysis and experiments on the 1379381_at probe set in relation with the tumor necrosis factor receptor superfamily could be initiated.

Finally, the proposed framework offers a model flexibility to deal with unpredictable problems during the numerical analysis and feature selection such as the probe set availability for further validating experiments. A practical case is when one of the chosen features is not available for a further validation with immunohistochemistry (IHC), due to the unavailability of the respective antibodies[26]. In that condition, the inferred hierarchical structure offers an efficient way to find alternatives to the best proposed model. Two cases are discussed here:

1. One of the metagene components is not available for validation;
2. An individual gene is not available for validation.

Both cases are studied analyzing the *Run 0* results about the E endpoint classification. The final system is a two dimensional classifier composed of a metagene and the 205225_at probe set. In the first scenario, assume that the metagene cannot be used because one of its three probe sets is unavailable for validation. In such case, the chosen metagene could be substituted by any of the available descendants in the hierarchical tree without loosing too much in terms of the prediction performance: at worst, an error rate of 11% and an MCC = 0.770 can be obtained instead that an error rate of 9% and MCC = 0.812 (see Figure 8). The second scenario is complementary to the first one. In this case, assume that the unavailable feature for validation is an individual probe set, 205225_at, used jointly with the previously chosen metagene. In this case, the hierarchical structure may be used to find the closest available nodes to the originally selected feature. The obtained results are shown in Figure 9. As can be seen, the best results (obtained with the 205225_at probe set) correspond to MCC = 0.812 and error rate = 9 %. The best alternative is obtained with the *brother* node which is a metagene. It gives a MCC = 0.756

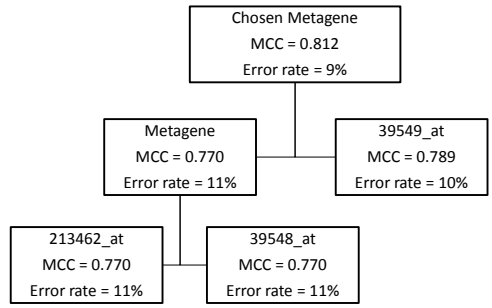

Fig. 8. Hierarchical structure with the chosen metagene as root. In each node, the obtained MCC value and error rate are showed when the node is used instead of the chosen metagene. The best values are obtained with the original feature, root node, but the substitution with one of its descendant does not severely degrade the performances.

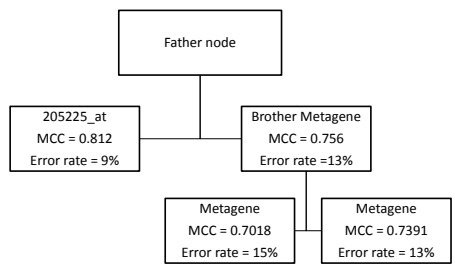

Fig. 9. Substitution results for the 205225_at probe set. In each node the obtained MCC value and error rate are showed when the node is used instead of the chosen probe set. The best values are obtained with the original feature, 205225_at and the best substitution is with the brother node, *Brother Metagene*. The root node has no available values because it cannot be chosen as a substitute for the 205225_at node.

and an error rate = 13%. The *brother* node is a metagene composed of five probe sets, 209602_at, 209603_at, 209604_at, 212956_at and 212960_at and obtains better performance than any of its descendants in the hierarchical structure.

6. Conclusions

In this work, a new classification framework to analyze microarray data has been studied. It has been developed as an improvement to the microarray classification method presented in[2]. There are two key elements in the proposed framework. The feature set enhancement through hierarchical clustering and the introduction of the reliability in the score evaluation during the feature selection stage.

The feature set enhancement with the addition of metagenes has confirmed its usefulness. The mean predictive performance has always improved when metagenes are used. Concerning the two studied metagene generation techniques, the *Treelets*

clustering has allowed to reach the best results even when compared with a wide variety of state of the art alternatives. Moreover, a deeper look at the chosen metagenes composition showed how metagenes can also be useful in summarizing common behavior of probe sets sharing the same biological function. Furthermore, the metagenes can help in the knowledge/hypothesis generation, focusing the attention for a further study on probe sets not endowed with previous biological knowledge but that are useful for classification.

The feature selection phase improvements are related to the introduction of the reliability measure, defined in[2]. Scoring rules making a simultaneous use of both the classification error rate and the reliability in the training phase allowed to reach better results than using the lexicographic sorting introduced in[2]. When compared to the results obtained with the lexicographic criterion, both the linear combination rule and the exponential penalization rule improve the prediction results.

In the linear combination and the exponential penalization scores there is a dependence on the α parameter which weights the relative importance of the error rate and the reliability. It has been shown that the mean results are robust with respect to small α parameter changes.

A Monte Carlo study has been performed to analyze the repeatability of the best results. It showed how the results are consistent with those previously obtained. The performance distribution is well concentrated around the mean values in all the cases where the class distribution is not highly skewed. This behavior has been studied analyzing synthetic data. This study has confirmed how the results variability increases when the class distribution skewness is very high.

The proposed framework has proven to be a valid alternative for microarray classification. It has a good predictive power, competing with a wide variety of state of the art techniques and is consistent along repeated runs. Finally, an efficient way to produce alternative classifiers to the proposed one is given by the inferred structure, in case the chosen features are unavailable for further validation studies or clinical applications.

The proposed algorithm has the potential to be straightforwardly applied to the analysis of epigenetic data like methylation arrays, or to the analysis of gene expression values obtained with Next Generation Sequencing: RNA-seq. In the latter case, the input data must be the numerical values obtained after the application of data processing techniques to assign expression values to individual genes. Future studies will assess the predictive power of the proposed algorithm applied in both scenarios in order to have a broader evaluation of its potential.

Acknowledgements

This work has been partially financed by "Fundació privada CELLEX"; and by the "Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya".

References

1. R. E. Bellman, *Dynamic programming*, Dover Publications, Incorporated, 2003.
2. M. Bosio, P. Bellot Pujalte, P. Salembier, and A. Oliveras, *Feature set enhancement via hierarchical clustering for microarray classification*, IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS) (San Antonio TX, USA), IEEE, December 2011.
3. U. M. Braga-Neto, *Fads and fallacies in the name of small-sample microarray classification - a highlight of misunderstanding and erroneous usage in the applications of genomic signal processing*, Signal Processing Magazine, IEEE **24** (2007), no. 1, 91 –99.
4. U. M. Braga-Neto and E. R. Dougherty, *Is cross-validation valid for small-sample microarray classification?*, Bioinformatics **20** (2004), no. 3, 374–380.
5. L. Breiman, *Bagging predictors*, Machine Learning **24** (1996), 123–140, 10.1007/BF00058655.
6. K. Deb and A. Reddy, *Reliable classification of two-class cancer data using evolutionary algorithms*, BioSystems (2003).
7. M. Dettling and P. Bühlmann, *Finding predictive gene groups from microarray data*, J. Multivar. Anal. **90** (2004), 106–131.
8. R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern classification*, Wiley, 2001.
9. B. Efron, *Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods*, Biometrika **68** (1981), no. 3, 589–599.
10. P.G. Espejo, S. Ventura, and F. Herrera, *A survey on the application of genetic programming to classification*, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **40** (2010), no. 2, 121 –144.
11. H. Gohlmann and W. Talloen, *Gene expression studies using affymetrix microarrays*, (2009).
12. T. Hastie, R. Tibshirani, D. Botstein, and P. Brown, *Supervised harvesting of expression trees*, Genome Biology **2** (2001), no. 1.
13. J. Hua, W. Tembe, and E. R. Dougherty, *Performance of feature-selection methods in the classification of high-dimension data*, Pattern Recognition **42** (2009), no. 3, 409–424.
14. D. W. Huang, B. T. Sherman, and R. A. Lempicki, *Systematic and integrative analysis of large gene lists using david bioinformatics resources.*, Nature Protocols **4** (2009), no. 1, 44–57.
15. E. Keedwell and A. Narayanan, *Genetic algorithms for gene expression analysis*, Applications of Evolutionary Computing, Lecture Notes in Computer Science, vol. 2611, Springer Berlin / Heidelberg, 2003, 10.1007/3-540-36605-9_8, pp. 191–192.
16. L. Klebanov and A. Yakovlev, *How high is the level of technical noise in microarray data*, Biol. Direct (2007), 9.
17. R. Kohavi and G. H. John, *Wrappers for feature subset selection*, Artif. Intell. **97** (1997), 273–324.
18. A. B Lee, B. Nadler, and L. Wasserman, *Treelets - an adaptive multi-scale basis for sparse unordered data*, Annals of Applied Statistics **2** (2008), no. 2, 435–471.
19. Q. Liu, A. H. Sung, Z. Chen, J. Liu, X. Huang, and Y. Deng, *Feature selection and classification of maqc-ii breast cancer and multiple myeloma microarray gene expression data*, PLoS ONE **4** (2009), no. 12, e8250.
20. G. J. McLachlan, K.A. Do, and C. Ambroise, *Analyzing microarray gene expression data / geoffrey j. mclachlan, kim-anh do, christopher ambroise*, Wiley-Interscience, Hoboken, N.J. :, 2004 (English).

21. S. Nakariyakul and D. P. Casasent, *An improvement on floating search algorithms for feature subset selection*, Pattern Recognition **42** (2009), no. 9, 1932 – 1940.
22. R.M. Parry, W. Jones, T.H. Stokes, J.H. Phan, R.A. Moffitt, H. Fang, L. Shi, A. Oberthuer, M. Fischer, W. Tong, and M.D. Wang, *k-nearest neighbor models for microarray gene expression analysis and clinical outcome prediction.*, Pharmacogenomics J **10** (2010), no. 4, 292–309.
23. P. Pudil, J. Novovičová, and J. Kittler, *Floating search methods in feature selection*, Pattern Recogn. Lett. **15** (1994), no. 11, 1119–1125.
24. L. Shi, G. Campbell, W. D. Jones, F. Campagne, and Z. Wen, *The microarray quality control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models.*, Nature biotechnology **28** (2010), 827–38.
25. A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, *Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles*, Proceedings of the National Academy of Sciences of the United States of America **102** (2005), no. 43, 15545–15550.
26. S. Suster and C. A Moran, *Applications and limitations of immunohistochemistry in the diagnosis of malignant mesothelioma.*, Adv Anat Pathol **13** (2006), no. 6, 316–29.
27. W. K. Yip, S. B. Amin, and C. Li, *A survey of classification techniques for microarray data analysis*, Handbook of Statistical Bioinformatics (Henry Horng-Shing Lu, Bernhard Schölkopf, and Hongyu Zhao, eds.), Springer Handbooks of Computational Statistics, Springer Berlin Heidelberg, 2011, 10.1007/978-3-642-16345-6_10, pp. 193–223.