Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

Microarray classification with hierarchical data representation and novel feature selection criteria

Mattia Bosio, Pau Bellot, Philippe Salembier, 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.
E-mail: mattia.bosio@upc.edu

***Abstract*—Microarray data classification is a challenging problem due to the high number of variables compared to the small number of available samples. An effective methodology to output a precise and reliable classifier is proposed in this work as an improvement of the algorithm in [1]. It considers the sample scarcity problem and the lack of data structure typical of microarrays. Both problem are assessed by a two-step approach applying hierarchical clustering to create new features called metagenes and introducing a novel feature ranking criterion, inside the wrapper feature selection task. The classification ability has been evaluated on 4 publicly available datasets from *Micro Array Quality Control study phase II* (MAQC) classified by 7 different endpoints. The global results have showed how the proposed approach obtains better prediction accuracy than a wide variety of state of the art alternatives.**

***Index Terms*—Microarray classification; metagenes; hierarchical representation; Treelets; feature selection; LDA; wrapper.**

I. Introduction

Gene expression microarrays are a powerful high-throughput technology which offers the ability to simultaneously measure thousands of gene expression values, thereby providing a significant amount of multivariate data with which it is possible to produce classifiers. The typical microarray analysis setting constitutes an extreme case of high-dimensionality (or sample scarcity) as there is a very large number of available features with respect to the sample number. In such circumstances, a feature selection process to produce reliable classifiers is necessary as stated in [2], [3].

In this paper, an efficient and reliable microarray classifier, able to reach the smallest prediction error using as few features as possible to reduce the overfitting risk is proposed. It is an evolution of the method presented in [1]. The literature provides a vast number of microarray classifiers as remarked in [4], among which evolutionary algorithms have obtained good results [5], thanks to the mutation possibility of the feature set during the train phase. A drawback of evolutionary algorithms is stated in [6], which discusses how performance tends to decrease when the feature set dimension grows to numbers comparable to those of microarray datasets. On the other hand, algorithms like *Tree Harvesting* [7], or *Pelora* [8], highlight the usefulness of hierarchical clustering as a method to extract interesting new variables to expand the original feature set. The possibility to summarize groups of genes with similar expression pattern in a single feature as input for the classifier has many advantages. First, the interpretability of the selected feature as a combination of correlated genes that may be involved in the same biological process. Second, the robustness to noise or random fluctuations because a group of correlated genes useful for classification is less likely to be due to chance than an individual gene. Third and last, classifying with a cluster-representing features can highlight linear relations among groups of correlated genes. The benefits of an expanded feature set and of a flexible feature selection algorithm are pursued in this paper through a novel classification scheme. The proposed algorithm is an enhanced version of the two-step process in [1]. At first, the original data are enriched via a hierarchical clustering method. New features called metagenes that are linear combination of the original gene expression are added. Each metagene is a synthesis of a gene cluster representing the common trend in a group of correlated genes.

The second step consists in the application of a flexible wrapper feature selection process called *Improved Sequential Floating Forward Selection* [6]. In this phase, a reliability measure is introduced due to the microarray data characteristic of sample scarcity. This reliability parameter increases the information amount obtained with the commonly used error rate estimation, gaining more insights about the actual data distribution from the classifier point of view. Inside the feature selection process, error rate and reliability are combined into a final score to determine the predictive power of each candidate. The key point in the current paper contribution with respect to [1] is the score definition rule. The new rules make better use of the reliability parameter, assigning more importance in the decisional process and improving the feature selection.

The prediction performance of the proposed algorithm is compared on four publicly available datasets, classified following seven different endpoints. The datasets are available at http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE16716 and are a subset of the six datasets utilized in the *Micro Array Quality Control study phase II* (MAQC) [9]. Results from this paper are then compared to results obtained in MAQC study following the same evaluation procedure, where more than 30.000 models were built using many combination of analytical methods.

This paper is organized as follows: in Section II, the metagene creation process is described, while in Section III the feature selection procedure is presented with particular attention to the reliability parameter definition and to the scoring rules. In Section IV, the experimental protocol is

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-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:*
$$\underline{m}_i = \alpha_a \underline{f}_a + \alpha_b \underline{f}_b = \sum_{i=1}^{p} \beta_i \underline{g}_i;$$
Each metagene is equivalently a linear combination of $\underline{f}_a$ *and* $\underline{f}_b$ *and a linear combination of all original features* $\underline{g}_i$
4) $\underline{\underline{F}} := \underline{\underline{F}} \cup \{\underline{m}_i\}$: *add new metagene to active feature set*
5) $\underline{\underline{F}} := \underline{\underline{F}} \setminus \{\underline{f}_a, \underline{f}_b\}$: *remove the two features* $\underline{f}_a, \underline{f}_b$ *from the active feature set*
$\underline{\underline{M}} := \underline{\underline{M}} \cup \{\underline{m}_i\}$: *include metagene* $\underline{m}_i$ *into the metagene set*

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. General clustering algorithm.

detailed. The classification results are presented in Section V, compared to many state of the art alternatives using the same experimental protocol. A discussion about the utility and efficiency of the proposed method is presented in Section VI.

II. Feature set enhancement

In the expansion of the original feature set process, the clustering operation is not performed to find gene clusters, but to generate a new set of features, each of which summarizes in itself a cluster of genes. The expected result from a metagene is a noise reduction with respect to individual genes thanks to the filtering effect of the linear combination. The objective is to highlight the common behavior of a gene cluster and reproduce it into a metagene.

The chosen global approach is a bottom up, pairwise hierarchical clustering described by the pseudo code of Figure 1. It is an iterative algorithm that, starting from individual genes, merges the pair of most similar features at each step. The newly created metagene is then added to the feature set whereas the pair of most similar features are removed from it. At the end of the process, the initial feature set of *p* genes is expanded with *p-1* metagenes. The key points in the metagene creation are the similarity metric: $d(\cdot,\cdot)$ and the generation rule: $g(\cdot,\cdot)$. Any change in one of these two functions implies the generation of a different metagene set. Two metagene generation methods have been studied in this work as in [1].

A. Treelets clustering

The first technique is based on Lee's work in [10], where an adaptive method for multi-scale representation and eigen-analysis of data called *Treelets* is presented. This method produces a clustering tree in which, at each level, the two most similar features are chosen and replaced by a coarse-grained approximation feature and a residual detail feature. In *Treelets*, the Pearson correlation is chosen as similarity measure: $d(\underline{f}_a, \underline{f}_b) = \langle \underline{f}_a, \underline{f}_b \rangle / (\|\underline{f}_a\| \cdot \|\underline{f}_b\|)$, where $\underline{f}_a$ is the sequence of gene expression values for all the samples. The two newly created features, approximation and detail, are obtained through a local Principal Component Analysis (i.e. PCA) on the two child nodes: the coarse-grained approximation is defined as the first local principal component, while the detail is the second one. In this work the approximation feature is chosen as metagene at each level in the iterative process.

B. Euclidean clustering

In the second technique, called *Euclidean* clustering, the negative Euclidean distance, $d(\underline{f}_a, \underline{f}_b) = -\|\underline{f}_a - \underline{f}_b\|_2$ is chosen, reaching a maximum of zero when the two features are equal. This choice slightly modifies the clustering process in the metagene generation rule. The selection of the first PCA component as metagene implies that it is a scaled weighted average of the genes. An illustrative example where all features are equal is presented in Figure 2. In this case, using the first PCA component produces metagenes which are not pure weighted average of genes. There is a scaling factor that moreover depends on the number of genes in the cluster. This phenomenon does not affect the Pearson correlation thus it is irrelevant in the *Treelets* case, but it is definitely an issue when a point-wise similarity is considered.

To correctly compare genes with metagenes, the latter should be a pure weighted average of genes. To obtain that, when a metagene $\underline{m}_x$ is created, two versions of it are used. The first one is the same as in the *Treelets* case, $\underline{m}_x$, while the second is a scaled version of the former: $\underline{m}_{xscaled} = \underline{m}_x / \|\underline{\beta}\|_1$. In this way, the scaled version, $\underline{m}_{xscaled}$, is a pure weighted average of the corresponding genes, thus it is used in the pairwise distance calculation and it is chosen as metagene. From Figure 2, it can be seen how the scaled versions are correctly comparable with the individual genes in terms of Euclidean distance, obtaining $d(\underline{m}_{1scaled}, \underline{f}_i) = 0$. Using the non scaled version, instead, would produce undesired results like $d(\underline{m}_1, \underline{f}_i) = (\sqrt{2} - 1) \cdot \|\underline{m}_i\|_2$. The non scaled version is maintained because it is used to preserve the energy distribution among the elementary components when a new metagene is built from $\underline{m}_x$ as showed in Figure 2. Without this approach, the $\underline{m}_{2scaled}$ in Figure 2 would become $\underline{m}_{2scaled} = 1/4 \cdot \underline{f}_1 + 1/4 \cdot \underline{f}_2 + 1/2 \cdot \underline{f}_3$, giving more weight to the last added feature.

III. Feature selection

The metagene creation process enriches the initial feature set with a whole new group of possibilities. The metagenes can improve classification because they expand the available feature space. Each one of them is the representation of the common behavior of a gene cluster, thus can benefit from a noise filtering effect derived from the linear combination. The main problem now is to choose an appropriate feature subset to train a precise and reliable classifier.

A. The IFFS algorithm

For the feature selection, we have chosen a deterministic approach to avoid evolutionary algorithms problems of

Feature set $\underline{\underline{F}}_0 = \{\underline{f}_1, \underline{f}_2, \underline{f}_3\}$ *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* $\underline{f}_2$

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

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

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

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

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

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

Scaled versions $\underline{m}_{1scaled}$ *and* $\underline{m}_{2scaled}$ *are used for Euclidean clustering because they preserve the components dynamics. The scaled versions will expand the feature set.*

Non scaled versions $\underline{m}_1$ *and* $\underline{m}_2$ *are used in the construction phase with PCA as they preserve the energy distribution among the components*

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

precision loss when the feature set dimension grows, [6]. The chosen algorithm tries to preserve the advantages of an evolutionary search allowing mutations of previous choices. Figure 3 illustrates the algorithm flowchart. It is a modification of the Sequential Floating Forward Selection algorithm (SFFS) [11], with the introduction of a replacing step when backtracking does not work. It is called *Improved sequential Floating Forward Selection* (IFFS) and has proven to get better or equal prediction results than SFFS [6]. In each step the current feature subset is updated by choosing the best available alternative depending on a $J(\cdot)$ performance metric. For each alternative, the $J(\cdot)$ is obtained by training a classifier on the candidate subset and evaluating its predictive ability.

B. Feature ranking criterion

The IFFS algorithm has been adopted for feature selection, so a classifier is applied inside the selection phase. The Linear Discriminant Analysis (LDA) [12] has been used in this study because its simple classification rule makes the results easier to interpret and more robust to overfitting [13]. Throughout the feature selection process, the criterion $J(\cdot)$ has a determinant role in the feature selection process. It sorts the features and decides which is the best one, and it is an estimation of the classifier predictive power with the current feature set. To obtain a reliable estimation, a 10 times 5-fold stratified cross validation process has been used.

Due to the microarray data characteristic involving few samples and many dimensions, a $J(\cdot)$ criterion based only on error rate may not be enough in ranking features. Indeed, it is common to have a group of features with the same error rate from which only one feature has to be selected. Furthermore, due to the chosen 10 times 5 fold cross validation, slight error rate differences can be due to an unfortunate dataset partition: if a specific sample is included in the test set more times than another it gains more relative weight in the error rate.

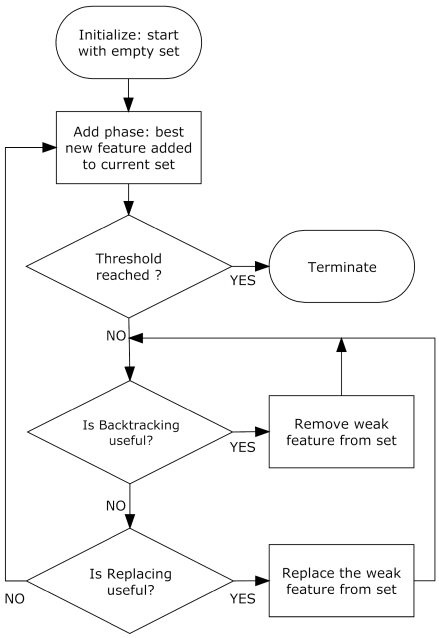

Fig. 3. IFFS feature selection algorithm.

1) The reliability parameter: To overcome the error rate limitation as a fitness estimator in a small sample scenario, an additional value is included to define the $J(\cdot)$ score: the reliability. It takes into account that a feature that can obtain well separated classes (i.e. high margin) is better than a feature in which the two classes are separated only by a very thin margin.

The reliability parameter r quantifies the estimation goodness as a weighted sum of sample distances from the classification boundary. It is calculated on the test set samples and the final value is the mean through the cross validation iterations. The reliability is calculated inside a cross-validation iteration for a two-class problem. It is defined in (1), where 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.

$$r = \frac{1}{n_{test} \cdot \hat{\sigma}_d} \left[\sum_{l=1}^{n_{test} \in c_1} \frac{d_l}{p(c_1)} + \sum_{l=1}^{n_{test} \in c_2} \frac{d_l}{p(c_2)} \right] \quad (1)$$

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 $\hat{\sigma}_d$ definition recalls the independent two-sample t-test denominator with classes of different size and variance, as it is the most general case for a two-class problem. In detail $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the estimated variances of sample distance from boundary for all samples of class 1 and 2 respectively. Dividing by $\hat{\sigma}_d$ guarantees that r is invariant to a scaling factor, thus obtaining the same value for metagenes that are perfect scaled replicas

of genes. Dividing by $p(c_l)$ assigns to each class the same relative weight and it is useful when the test set distribution is highly skewed. 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. It is penalized by a factor proportional to error value so that greater errors produce greater penalties, allowing discrimination among features with equal error rates.

2) Score calculation: The final $J(e, r)$ value is a combination of the mean error rate and the mean reliability parameter along the cross validation iterations. A feature is ranked to be better than another if its $J(e, r)$ score is higher. This is a crucial point in the feature selection because changing the $J(e, r)$ definition rule highly affects the chosen subset. The score definition a key innovation to the algorithm presented in [1], where the features were ranked following a two-level lexicographic criterion. Features were ranked in terms of error rate at first, thereafter, reliability was considered only to break ties among features obtaining the same error rate. With such a criterion, reliability influence loses importance as the test set cardinality grows because the probability to obtain equal error rates decreases proportionally To overcome that limitation and to make more use of the information proceeding from reliability too, new score rules are here proposed to unify in a scalar value both error rate and reliability.

The objective is to propose a softer combination rule allowing reliability comparison not only among features with equal error rate. The first idea is to compare features by the reliability value, properly penalized in terms of the error rate to induce a fixed penalization factor for a constant error rate difference. Such a behavior can be obtained introducing an exponential penalization to the reliability value as detailed in (2). For each feature, the $J(e, r)$ score is obtained as in (2), where r is the reliability value, e is the error rate value, and α is a penalization parameter.

$$J \;=\; r \cdot \exp\left(-sign(r) \cdot \frac{100}{\alpha} \cdot e\right) \tag{2}$$

$J(e, r)$ is a product of the reliability value with a penalization coefficient ≤ 1 with 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 values, while the α parameter defines the steepness of the penalization. The α value defines the e^{-1} penalization interval: between two feature with equal reliability value, an $\alpha\%$ difference in the error rate induces a e^{-1} penalization 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 great the dominant parameter becomes the reliability (when $\alpha \to \infty$ the error rate is not taken into account).

The proposed score is influenced both by error rate and reliability, 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 is useful for small sample datasets like microarrays because it takes into account the data distribution seen from the classifier point of view, thus giving a higher score to features showing high mean class separation and small intra-class variation.

The exponential penalization is not the only score considered in this paper, a linear combination of error rate and normalized reliability has also been considered. The linear combination score is obtained as in (3) as a weighted sum of error rate e and a normalized reliability value $r_n = (r - \min(r))/\max(r)$. The α parameter is bounded between 0 and 1 and it defines the relative weight of reliability with respect to the error rate.

$$J \;=\; (\alpha) \cdot r_n \;\; + (1 - \alpha) \cdot (1 - e) \;\; \alpha \in [0, 1] \tag{3}$$

This simple scoring rule allows a more flexible comparison of reliability values among features with different error rates. It has a linear trend both in the error rate and in the reliability direction. With respect to the former exponential penalization scoring, here, a constant penalization is added (not multiplied) to a constant error rate increase.

Throughout the experiments, both the exponential penalization and the linear combination criteria are compared for microarray classification to highlight how differences in the score definition imply performance changes.

IV. Experimental Protocol

The analyzed data are a subset of the provided datasets by the MAQC II consortium: six datasets containing 13 preclinical and clinical endpoints coded A through M; for more information refer to [9]. Each endpoint corresponds to a different sample classification so that the same dataset can be classified following different criteria (e.g. treatment, outcome, sex, random, etc.).

In this work, four out of six datasets have been used, corresponding to endpoints A,C to H endpoints of [9], available at http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE16716. A detailed explanation of the endpoint composition is included in Table I. These data have been chosen because they are highly reliable, selected after a quality control process in order to provide a common test ground and because for each endpoint both a training set and an independent validation set are provided [9]. Furthermore, many different laboratories have tested their algorithm on the same datasets with the same evaluation protocol (i.e. train the classifiers on the training set with performance assessment on the validation dataset) and published their final outcome [9], [14], [15], thus an accurate benchmark can be performed to understand how well does a proposed algorithm perform with respect to a large number of state of the art alternatives. The experimental setup is a sequence of five main steps: data preprocessing, metagene creation, α optimization on small datasets, full-data analysis with the chosen α values and a final performance assessment.

The data preprocessing step for all the datasets consists in setting the minimum value to $log_2 10$ to not consider small valued probe sets followed by a mean removal operation along the samples direction (i.e. each feature is set to have zero mean). The metagenes are built as explained in Section II.

TABLE I
MICROARRAY DATASETS USED FOR CLASSIFICATION.

Dataset	Endpoint description		Microarray platform	Training set			Validation set		
				Samples	Positives	Negatives	Samples	Positives	Negatives
Hamner	Lung tumorigen vs. non tumorigen	A	Affymetrix Mouse 430.2.0	70	26	44	88	28	60
NIEHS	Liver toxicant vs. non toxicant	C	Affymetrix Rat 230.2.0	214	79	135	204	78	126
Breast cancer	Pre operative treatment response	D	Affymetrix 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	Affymetrix Human U133Plus 2.0	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

In the next steps the predictive performance of the proposed method is measured. The objective is to see if the changes in the feature selection phase improve the prediction ability with respect to the algorithm presented in [1] and to benchmark the current performance with state of the art alternatives. As shown in subsection III-B2, both the exponential penalization and the linear combination depend on a α parameter, so the algorithm has been tested on multiple α values. An optimization phase has been added to choose a small set of α values for time reasons. The chosen performance metric is the Matthews Correlation Coefficient (MCC) [16], since, as stated in [9] it is informative when the distribution of the two classes is highly skewed, it is simple to calculate and available for all models with which the proposed method has been compared to. MCC values range from -1 (i.e. perfect inverse prediction) to 1 (perfect prediction). The linear combination rule has been tested a range of α values between 0.05 and 1 with 0.05 interval, the best selected value is 0.15. About the exponential combination a range of α values from 5 to 100 with 5 interval has been tested, choosing $\alpha = 10$ for further application in real world scenarios. In both cases, the results are quite stable to small α variations as long as it maintains small values.

Once the α values have been chosen, the same analysis is performed on the complete dataset (genes and metagenes) applying the feature selection algorithm to train classifiers up to five dimensions. Results are collected and the classifier obtaining the best MCC value is considered as the measure of the prediction potential of the method.

The complete analysis has been applied on all the datasets, for all the endpoints in Table I, to find the predictive potential of the proposed method. To have a complete comparison about the feature enrichment techniques, experiments have been performed adopting *Treelets, Euclidean* clustering and without any metagene. In this way it is possible to evaluate which is the best enrichment technique and quantify its benefits with respect to the initial gene set. A set of experiments has also been performed applying the lexicographic algorithm of [1], to evaluate the improvements induced by the scoring techniques.

V. RESULTS AND DISCUSSION

The collected experimental results are presented in Figure 4. They represent the mean MCC value across the classified endpoints. Data in Figure 4 are represented as columns with the MCC value indicated above each bar. In Figure 4, bars are named depending on the adopted classification method and are sorted by increasing MCC value. Results coded by *dat XX* prefix are extracted from +30.000 models evaluated in [9]. More information about which laboratory or academic institution is represented by *dat XX* identification can be found in [9] supplementary material. It can be seen how they span a range from 0.284 for *dat 3*, to a 0.490 for *dat 24* in terms of MCC.

The black column, labeled as *Lexicographic* is the best result of applying the algorithm proposed in [1] to the MAQC databases classification, which leads to a 0.423 mean MCC value with *Treelets* clustering. Such not so outstanding prediction ability is mainly due to the rigidness of the scoring system in the feature selection phase which makes the feature ranking utterly sensitive to slight error rate differences connected to the cross validation process.

Results are highlighted depending on the adopted feature enrichment technique: those adopting *Treelets* clustering are represented as dark gray bars with the *T_xx* prefix; those adopting the *Euclidean* clustering are represented by bars filled with black and white horizontal lines and are coded by the *E_xx* prefix; finally the results obtained without adding any metagene to the initial dataset are visualized with a black and white mosaic pattern and with the *N_xx* prefix. For each subgroup the score definition rules are coded as: *exponential penalization* = *_exp* or *linear combination* = *_lin*. Finally, labels include the α value. The exponential penalization obtained better results for the *Treelets* clustering case and the gene only case; on the contrary, the best result with the linear combination is reached adopting the *Euclidean* clustering.

Analyzing the results, a general improvement has been obtained with respect to the lexicographic scoring [1]. The results using the exponential penalization or the linear combination

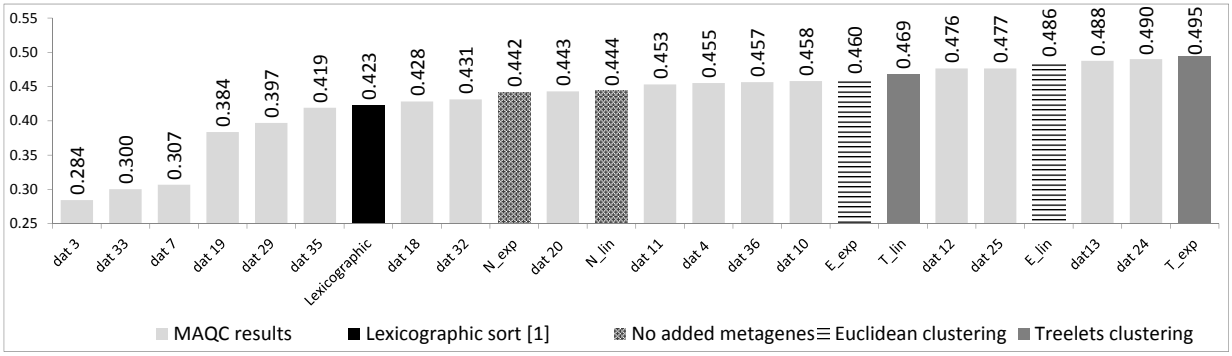

Fig. 4. Mean MCC values obtained by classifying endpoints A,C,D,E,F,G,H provided by MAQC study [9]. Results are sorted by increasing MCC value. Results from the MAQC study are the light gray *dat_xx* columns. The black *"Lexicographic"* column shows the best result with algorithm from [1]. Results with the current framework are highlighted depending on the feature enrichment method adopted: dark gray columns for the *Treelets* clustering, black and white horizontal pattern for the *Euclidean* clustering, and a mosaic pattern if no metagenes are added to the original set.

reach higher mean performance rates.

In two out of three cases, *No metagenes* and *Treelets clustering*, applying the exponential penalization rule allows better results, while when the *Euclidean clustering* is chosen the best alternative is to apply the linear combination rule.

The metagene creation phase is useful to classification because better results can be obtained applying the *Treelets* or the *Euclidean clustering* than without any metagene addition. The best results are obtained applying the *Treelets* hierarchical clustering with exponential penalization scoring. In this case, the obtained MCC mean value is the highest among all the alternative in Figure 4, thus highlighting the predictive potential of the developed algorithm.

A summary of the observed results is that the changes in the score definition have significantly improved the prediction performance. Metagenes have proven useful for classification and, if *Treelets* is used with exponential penalization rule it is possible to reach a mean MCC value higher than a wide variety of state of the art alternatives.

VI. Conclusion

In this paper, improvements to the microarray classification method presented in [1] have been studied. The key contribution has been to change the score definition inside the feature selection phase: it allows a better use of the reliability information, thus overcoming the limitations of the original lexicographic sorting. The metagene creation process induced benefits within this framework too, considerably improving the mean performance with respect to solutions involving only genes. Between the two proposed score metrics, linear combination or exponential penalization, the latter has proven to get the best results if *Treelets* is applied as feature enrichment. Furthermore, in this case the mean MCC value is better than all the state of the art alternatives compared in this study.

The proposed classification method has produced very interesting classifiers with mean MCC value close to, and superior to the best methods in [9]. The score definition rule with exponential penalization allows to reach the best performance.

Acknowledgements

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

References

[1] M. Bosio, P. Bellot Pujalte, P. Salembier, and A. Oliveras, "Feature set enhancement via hierarchical clustering for microarray classification," in *IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS)*. San Antonio TX, USA: IEEE, December 2011.

[2] S. Dudoit and J. Fridlyand, "Classification in microarray experiments," *Statistical analysis of gene expression microarray data*, pp. 93–158, 2003.

[3] J. Hua, W. Tembe, and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," *Pattern Recognition*, vol. 42, no. 3, pp. 409–424, 2009.

[4] W.-K. Yip, S. B. Amin, and C. Li, "A survey of classification techniques for microarray data analysis," in *Handbook of Statistical Bioinformatics*, ser. Springer Handbooks of Computational Statistics, H. H.-S. Lu, B. Schölkopf, and H. Zhao, Eds. Springer Berlin Heidelberg, 2011, pp. 193–223.

[5] K. Deb and A. Reddy, "Reliable classification of two-class cancer data using evolutionary algorithms," *BioSystems*, 2003.

[6] S. Nakariyakul and D. Casasent, "An improvement on floating search algorithms for feature subset selection," *Pattern Recogn.*, 2009.

[7] T. Hastie, R. Tibshirani, D. Botstein, and P. Brown, "Supervised harvesting of expression trees," *Genome Biology*, vol. 2, no. 1, 2001.

[8] M. Dettling and P. Bühlmann, "Finding predictive gene groups from microarray data," *J. Multivar. Anal.*, vol. 90, pp. 106–131, July 2004.

[9] L. Shi *et al.*, "The microarray quality control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models." *Nature biotechnology*, vol. 28, pp. 827–38, 2010 Aug 2010.

[10] A. B. Lee, B. Nadler, and L. Wasserman, "Treelets - an adaptive multi-scale basis for sparse unordered data," *Annals of Applied Statistics*, vol. 2, no. 2, pp. 435–471, 2008.

[11] P. Pudil, J. Novovivcova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recogn. Lett.*, 1994.

[12] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley, 2001.

[13] U. Braga-Neto, "Fads and fallacies in the name of small-sample microarray classification," *Signal Processing Magazine, IEEE*, vol. 24, no. 1, pp. 91 –99, jan. 2007.

[14] R. Parry *et al.*, "k-nearest neighbor models for microarray gene expression analysis and clinical outcome prediction." *Pharmacogenomics J*, vol. 10, no. 4, pp. 292–309, 2010.

[15] Q. Liu *et al.*, "Feature selection and classification of maqc-ii breast cancer and multiple myeloma microarray gene expression data," *PLoS ONE*, vol. 4, no. 12, p. e8250, 12 2009.

[16] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme." *Biochimica et Biophysica Acta*, vol. 405, no. 2, pp. 442–451, 1975.