A distributed feature selection algorithm based on distance correlation with an application to microarrays

Aida Brankovic, *Student Member, IEEE,* Marjan Hosseini, and Luigi Piroddi, *Member, IEEE*

Abstract—DNA microarray datasets are characterized by a large number of features with very few samples, which is a typical cause of overfitting and poor generalization in the classification task. Here we introduce a novel feature selection (FS) approach which employs the distance correlation (dCor) as a criterion for evaluating the dependence of the class on a given feature subset. The dCor index provides a reliable dependence measure among random vectors of arbitrary dimension, without any assumption on their distribution. Moreover, it is sensitive to the presence of redundant terms. The proposed FS method is based on a probabilistic representation of the feature subset model, which is progressively refined by a repeated process of model extraction and evaluation. A key element of the approach is a distributed optimization scheme based on a vertical partitioning of the dataset, which alleviates the negative effects of its unbalanced dimensions. The proposed method has been tested on several microarray datasets, resulting in quite compact and accurate models obtained at a reasonable computational cost.

Index Terms—DNA microarrays, Feature selection, Classification, Model selection, Randomized methods, Distance correlation.

1 Introduction

The high dimensional nature of bioinformatic data poses a severe challenge on machine learning methods. For example, microarrays allow to simultaneously measure the expression levels of a large number of genes, so that the resulting datasets are characterized by a large number of features (more than 50 thousand genes) and a very limited sample size [1]. Most of the genes provide little or no information useful for classification purposes, and it is particularly important to detect the smallest subset of features (referred to as *biomarkers*), that provide sufficient information to separate the classes represented in the dataset (which could distinguish cancerous and noncancerous samples, or identify different types of cancer [2]). This highly crucial task is referred to as feature selection (FS), which is a combinatorial optimization problem aiming at selecting from a set of available features only the relevant ones, in order to build a classifier with the required performance. FS reduces the computational cost of the classifier design and simplifies its structure, thus facilitating model interpretation and data understanding, and ultimately improving both accuracy and robustness of the designed classifier [3]. Indeed, the presence of redundant features may adversely affect the classification accuracy, as they can add more noise than useful information [4].

The highly unbalanced dimensions of microarray datasets greatly complicate the FS task, and unsatisfactory classification performances are often reported with standard methods [43], [6]. Indeed, large feature vectors significantly slow down the learning process, since the complexity of the FS problem grows exponentially with the number of features. At the same time, the small number of samples may cause the classifier to overfit the training data, thus compromising model generalization [4]. Besides their unbalanced dimensions, microarray data are often affected by noise, which further aggravates the analysis. For all these reasons, specialized FS techniques must be developed to appropriately handle this type of datasets.

FS methods can be characterized as filter, wrapper or embedded methods. Filter methods select features based only on data-related properties, *i.e.* independently of the classifier design. Wrapper methods are more costly but potentially more accurate than filter-based ones, as they condition the FS process to the performance of the resulting classifier. Finally, embedded methods combine the benefits of both explained approaches: a feature screening is initially performed using a filter-based approach, followed by the application of a wrapper method to refine the final solution. In the following we focus on filter methods, which are the predominant choice in microarray problems. Indeed, the added cost of classifier design may be significant for large size problems. In addition, the classifier bias resulting from the relatively small number of samples can negatively affect the FS process [7].

Univariate filter methods are a common choice in view of their computational advantages. These methods are based on individual feature assessment, *i.e.* they rank the features based on their individual capabilities to discriminate among the classes (see, *e.g.*, [8], [9], [10]). Once the features have been ranked, the top ones in the ranking are selected. As interactions among features (in our case, the correlations among genes) are not taken into account, it is not infrequent that redundant terms might be selected in this way [11]. Furthermore, features that are individually not significant are discarded, although they may actually reveal strong discriminatory power in combination with others [4].

• *A. Brankovic, M. Hosseini, and L. Piroddi are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy. E-mail: marjan.hosseini@mail.polimi.it, {aida.brankovic, luigi.piroddi}@polimi.it*

Multivariate filter methods overcome this problems by evaluating subsets of features according to some scoring function. Multivariate methods pose greater complexity than univariate ones in that, besides requiring a method to evaluate groups of features, they also involve a search mechanism in the space of all possible feature subsets. Regarding the first issue, many works employ correlation-oriented criteria based on the concept of mutual information (MI) (see, *e.g.*, [12], [13], [8], [14], [15], [16]). Indeed, the MI between the features and the output reveals their discriminating capabilities, whereas the correlation among features indicates possible redundancy issues (see, *e.g.*, the Minimal Redundancy Maximal Relevance (MRMR) algorithm [14], and the Correlation Based Filtering (CFS) method [17]). It is important to note that the mentioned correlation-based criteria operate on pairs of variables, so that their usage to assess subsets of features of arbitrary size requires some form of aggregation of the pairwise computed indices (*e.g.*, averaging), which does not necessarily capture the actual value of a given subset [16]. Methods using this kind of approximated calculation of the group mutual information are proposed, *e.g.*, in [18], [19], [20], [21], [17]. From the above discussion, it is apparent that ranking criteria naively designed for groups of variables of arbitrary size, as opposed to pairs, are highly desirable for the problem at hand.

The second crucial element in multivariate filter methods is the strategy for selecting feature subset candidates to be evaluated and ranked. Indeed, in view of the exponential complexity of the underlying combinatorial problem, the exhaustive approach is barely applicable with large feature sets. The space of feature subsets is typically explored with heuristic rules. A typical choice is the incremental strategy, due to its simplicity. For example, the sequential FS (SFS) approach incrementally builds the model, by adding at each step the feature that yields the maximum marginal improvement (see, *e.g.*, [14], [17], [22]). This strategy has several drawbacks both conceptual and computational. First of all, the decision on which feature to add or remove at a given step of the selection process depends locally on the currently selected feature subset. In this respect, it can be easily seen that the marginal utility of a feature can greatly vary depending on the feature subset with respect to which it is evaluated. In other words, the relevance of a specific feature is not evaluated as a global property, but rather as a local one. This may stray the selection process from the optimal path. Also, what is optimized at every step is only the local improvement of the current feature subset with an elementary feature variation. In this way selection errors are propagated throughout the process. Finally, the incremental strategy depends critically on the threshold adopted as a stopping criterion. For all these reasons, methods based on greedy policies such as the SFS are subject to redundancy and overfitting issues, especially if applied to datasets with extremely unbalanced dimensions such as microarrays [4].

We here propose a novel multivariate filter-based FS method that can effectively tackle the two mentioned issues and is therefore suitable for classification problems with high data dimensionality and complex data distributions. The proposed method is based on the combination of the following three factors:

(i) A selection criterion based on the distance correlation (dCor);
(ii) A distributed combinatorial optimization approach;
(iii) A randomized FS procedure;

which are briefly explained below.

The dCor index [23], [24] provides an ideal criterion for the evaluation of feature subsets. Indeed, the dCor is a generalization of the correlation concept that provides a reliable dependence measure between random vectors of arbitrary dimension (not just pairs of random variables), without any assumption on their distribution. The higher the correlation between vectors, the higher the dependence measure (in case of linear dependence it equals 1, while it is 0 for independent vectors). In the presented approach the dCor is employed to evaluate a feature subset by measuring the correlation of the latter with the classification target output. As will be shown in the paper, the dCor is inherently robust to redundancy and overfitting issues, and provides satisfactory performance even in the presence of nonlinear dependencies [25]. The dCor has been studied for variable selection in regression problems [25], where it was employed in combination with an incremental model building strategy. It has also been applied for feature screening purposes in ultrahigh-dimensional data [26], where it proved more effective than a classical screening procedure based on the classical Pearson's correlation coefficient.

The distributed combinatorial optimization scheme allows to efficiently tackle the prohibitive complexity of the combinatorial problem underlying the classification task on microarrays, as a result of the large number of features combined with the small number of samples. It is based on a *divide et impera* strategy that breaks the FS problem into smaller and more balanced subproblems, which are typically more tractable by classification methods. More in detail, the original set of features is partitioned into several smaller subsets (denoted feature bins) and an FS algorithm is run independently on each of them. Then, the features belonging to the best among the obtained local solutions are added to all feature bins, and the local FS processes are repeated. This sharing of the most promising features with all the local FS problems allows each of them to improve the local solution by combining the old features with the new ones. The algorithm stops when all local problems converge over the same solution. A noticeable benefit of the suggested distributed approach is the inherent parallelizability of the procedure.

The third contribution of the presented approach is to employ a randomized FS method to address the local FS problems, in which the utility of each feature is evaluated in a global fashion, as opposed to the local evaluation adopted in incremental methods. More in detail, the FS selection problem is reformulated in a probabilistic framework, where a probability distribution characterizes the likelihood that each feature belongs to the target model. The FS procedure alternates a generation phase, where different feature subsets are extracted from the current distribution, to an assessment phase, where the distribution is updated based on an aggregate performance analysis carried out for each feature over all the extracted subsets. Features appearing more often in highly ranked feature subsets are re-enforced, and viceversa. Unlike incremental selection strategies the proposed method is occasionally capable of escaping local

optima, and is also reported to be less prone to redundancy and overfitting issues [27].

Randomized algorithms for FS are not new as such. For example, several works in the literature consider genetic algorithms (GA) for this purpose (see, *e.g.*, [43], [44], [45], [46], [47]). These methods also exploit randomization in the selection of potential features and process populations of feature subsets. However, while GA methods are based on enforcing the fittest models, our method is oriented towards the selection of the fittest *features*. For this purpose, we apply an aggregate evaluation of the population of models to assess the importance of a feature, rather than simply evaluating individually the models. It is also worth mentioning that the mechanisms adopted in GA-based methods to update the population of individuals are random and blind (there is no criterion guiding the crossover and mutation operators towards model improvement), which may easily lead to redundancy problems and extremely long convergence processes. Indeed, with such methods the feature set must be reduced to an order 10^2 [47], which makes their applicability to microarrays somewhat awkward.

In summary, the proposed distributed FS algorithm displays the following features:

(i) The method combines – for the first time, to the authors' best knowledge – various ideas, namely vertical partitioning, information sharing, iteration of the local FS processes, and the dCor criterion.
(ii) Feature distribution enforces a huge reduction in the problem complexity, thus enabling the applicability of the FS method to large problems such as microarrays.
(iii) This reduction in the combinatorial complexity does not jeopardize accuracy, since the information feedback mechanism ensures that promising features are visible by *all* local FS processes.
(iv) Operating separately on smaller feature sets generally leads to a more accurate functioning of the FS algorithm employed by the local processors, since the solution space is smaller. It also helps preventing overfitting and redundancy.
(v) The described iterative process allows combinations of features to emerge even if their components are originally scattered among the local FS search spaces, and so it results in a "deeper" space search overall.
(vi) The evaluation of feature subsets is carried out using the dCor criterion, which has several interesting properties for FS: it works for arbitrarily sized random vectors, does not depend on their distributions, and generally tends to avoid redundancy.
(vii) The assessment of the importance of the features is based on a *global* evaluation of a population of models.
(viii) The method tends to provide very compact models compared to other filter-based methods. This is a crucial property in order to establish the really important features for diagnostic purposes.

The rest of the paper is organized as follows. Section 2 provides the problem formulation and the relevant notation, and briefly reviews the related literature. Section 2.2 introduces the dCor index, emphasizing the properties that make it particularly suited to the FS task. The proposed method is introduced in Section 3. Section 4 provides different experimental studies carried on well-known microarray datasets from the literature. Finally, Section 5 presents some concluding remarks.

2 Preliminaries

2.1 The classification problem: definition and notation

We here consider the classification problem in the framework of supervised learning. Let $\mathcal{D} = \{d^{(1)}, \ldots, d^{(N)}\}$ be a set of N available observations, each consisting of an input-output pair $d^{(k)} = (\boldsymbol{f}^{(k)}, c^{(k)})$, where $\boldsymbol{f} = [f_1, \ldots, f_{N_f}]$ denotes the vector of features, and $c \in \{1, \ldots, N_c\}$ the class, with $k = 1, \ldots, N$. $\mathcal{D}$ is used to build a classifier, capable of predicting the class label of previously unseen samples of the features. The general form of the classifier is thus given as:

$$\hat{c} = h(\boldsymbol{f}), \tag{1}$$

where $\hat{c}$ denotes the predicted class associated to the vector of features $\boldsymbol{f}$ and h is a suitable function of the feature values.

Classifiers can be evaluated by means of the classification error rate, denoted PE (for percentage error), defined as the ratio of misclassified samples over the total number of tested samples. Equivalently, the performance index $J = 1 - PE$ can be employed. For binary classification problems, the performance index can be defined as:

$$J = \frac{TP + TN}{N}, \tag{2}$$

where TP and TN denote the number of correctly classified samples of classes 1 and 2, respectively. The total number of samples equals the sum of misclassified and correctly classified samples of both classes, *i.e.* $N = TP + TN + FP + FN$, where FP and FN are the misclassified samples of class 1 and 2, respectively.

2.2 The distance correlation index

Various statistical tests have been developed in the literature to test the dependence of random vectors. We here employ the one proposed by Szekely *et al.* [23], based on the concept of dCor. It is applicable to both discrete and continuous random variables, and does not require any *a priori* assumption on their distribution. For the sake of completeness, we here briefly report the main results of [23].

2.2.1 The basic dCor index

Let $\boldsymbol{x} = [x_1, \ldots, x_p]^T$ and $\boldsymbol{y} = [y_1, \ldots, y_q]^T$ be two random vectors, such that $\mathbb{E}(\|\boldsymbol{x}\| + \|\boldsymbol{y}\|) < \infty$, where $\|\cdot\|$ denotes the Euclidean norm. Let also $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$ be N i.i.d. realizations of $\boldsymbol{x}$, and $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}$ the corresponding i.i.d. realizations of $\boldsymbol{y}$. Now, the empirical distance covariance (briefly, dCov) is defined as

$$\nu_N^2(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{N^2} \sum_{k,l=1}^{N} A_{kl} B_{kl}, \tag{3}$$

where

$$A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot},$$
$$B_{kl} = b_{kl} - \bar{b}_{k\cdot} - \bar{b}_{\cdot l} + \bar{b}_{\cdot\cdot},$$

with

$$a_{kl} = \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(l)}\|, \quad b_{kl} = \|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(l)}\|,$$

and

$$\bar{a}_{k\cdot} = \frac{1}{N}\sum_{l=1}^{N} a_{kl}, \quad \bar{a}_{\cdot l} = \frac{1}{N}\sum_{k=1}^{N} a_{kl}, \quad \bar{a}_{\cdot\cdot} = \frac{1}{N^2}\sum_{k,\,l=1}^{N} a_{kl},$$

$$\bar{b}_{k\cdot} = \frac{1}{N}\sum_{l=1}^{N} b_{kl}, \quad \bar{b}_{\cdot l} = \frac{1}{N}\sum_{k=1}^{N} b_{kl}, \quad \bar{b}_{\cdot\cdot} = \frac{1}{N^2}\sum_{k,\,l=1}^{N} b_{kl}.$$

Then, the empirical dCor is the square root of

$$\mathcal{R}_N^2(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \frac{\nu_N^2(\boldsymbol{x},\boldsymbol{y})}{\sqrt{\nu_N^2(\boldsymbol{x})\nu_N^2(\boldsymbol{y})}}, & \nu_N^2(\boldsymbol{x})\nu_N^2(\boldsymbol{y}) > 0 \\ 0, & \nu_N^2(\boldsymbol{x})\nu_N^2(\boldsymbol{y}) = 0 \end{cases} \tag{4}$$

In the assumption that $\mathbb{E}(\|\boldsymbol{x}\| + \|\boldsymbol{y}\|) < \infty$, it holds that the sampled version of the dCor tends to the corresponding probabilistic quantity, denoted $\mathcal{R}$:

$$\lim_{N\to\infty} \mathcal{R}_N^2(\boldsymbol{x},\boldsymbol{y}) = \mathcal{R}^2(\boldsymbol{x},\boldsymbol{y}). \tag{5}$$

It also holds that $0 \le \mathcal{R}(\boldsymbol{x},\boldsymbol{y}) \le 1$, and $\mathcal{R}(\boldsymbol{x},\boldsymbol{y}) = 0$ iff $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent. Similarly, $0 \le \mathcal{R}_N(\boldsymbol{x},\boldsymbol{y}) \le 1$, and if $\mathcal{R}_N(\boldsymbol{x},\boldsymbol{y}) = 1$, then there exists a vector ζ, a nonzero real number τ and an orthogonal matrix C such that $\boldsymbol{Y} = \zeta + \tau\boldsymbol{x}C$.

In view of the last property, $\mathcal{R}_N(\boldsymbol{x},\boldsymbol{y})$ can be indeed used as a measure of linear dependence between random vectors. Fortunately, it can be verified that the proposed index is also sensitive to nonlinear input-output relationships.

2.2.2 The unbiased dCor index

It is worth mentioning that the bias of the dCor index increases with the dimension of the random vectors dimensions. As discussed in [24], for fixed number of samples N the dCor tends to 1 as $p, q \to \infty$. Thus, it might be hard to interpret the obtained index in high dimensional cases. This problem is investigated in [24] where an unbiased version of the dCor index is introduced, which is amenable for high dimensional problems. Here, the following quantities A_{kl}^* and B_{kl}^* are used instead of A_{kl} and B_{kl}:

$$A_{kl}^* = \begin{cases} \frac{N}{N-1}(A_{kl} - \frac{a_{kl}}{N}), k \neq l \\ \frac{N}{N-1}(\bar{a}_{k\cdot} - \bar{a}_{\cdot\cdot}), k = l \end{cases} \tag{6}$$

$$B_{kl}^* = \begin{cases} \frac{N}{N-1}(B_{kl} - \frac{b_{kl}}{N}), k \neq l \\ \frac{N}{N-1}(\bar{b}_{k\cdot} - \bar{b}_{\cdot\cdot}), k = l \end{cases} \tag{7}$$

Let

$$\mathcal{U}_N^*(\boldsymbol{x},\boldsymbol{y}) = \sum_{k\neq l}^{N} A_{kl}^* B_{kl}^* - \frac{2}{N-2}\sum_{k=1}^{N} A_{kk}^* B_{kk}^*. \tag{8}$$

The modified dCov and dCor indices are given respectively by:

$$\nu_N^*(\boldsymbol{x},\boldsymbol{y}) = \frac{\mathcal{U}_N^*(\boldsymbol{x},\boldsymbol{y})}{N(N-3)}, \tag{9}$$

$$\mathcal{R}_N^*(\boldsymbol{x},\boldsymbol{y}) = \frac{\nu_N^*(\boldsymbol{x},\boldsymbol{y})}{\sqrt{\nu_N^*(\boldsymbol{x})\nu_N^*(\boldsymbol{y})}}. \tag{10}$$

For simplicity, in the rest of paper we will drop the asterisk symbol and use the notation $\mathcal{R}_N$ to denote the unbiased dCor index.

2.2.3 Sensitivity of the dCor to redundant terms

We next present some illustrative simulations that emphasize the robustness of the dCor index in the presence of redundant terms. Let $\boldsymbol{x} = [x_1, \ldots, x_6]^T$ be a random vector and $y = 3x_1$, and assume that N i.i.d. realizations of both $\boldsymbol{x}$ and y are available. All elements of the $\boldsymbol{x}$ vector are independently drawn from the same distribution. Table 1 reports the dCor value calculated for different subsets of inputs on average over 1000 Monte Carlo tests performed for data generated with different distributions (normal, Poisson and lognormal). The evaluated input subsets are $\{x_1, \ldots, x_{1+k}\}$, for $k = 0, \ldots, 5$, corresponding to the exact model and 5 redundant models with increasing number of redundant terms. While the dCor equals 1 for the true model (including only x_1), its value decreases as we introduce further terms, regardless of the distribution of the data.

TABLE 1
Average dCor measure over 1000 Monte Carlo tests for increasingly redundant models (true model: $y = 3x_1$).

Number of redundant terms	Data distribution		
	Normal	Poisson	Lognormal
0	1.0000	1.0000	1.0000
1	0.9873	0.9835	0.9765
2	0.9778	0.9729	0.9573
3	0.9697	0.9640	0.9406
4	0.9623	0.9560	0.9262
5	0.9555	0.9488	0.9130

A similar result holds even if the input-output relationship is nonlinear, *e.g.* $y = 3x_1^2$, although this time the dCor associated to the model containing only x_1 is less than 1: any further term added to the model decreases the dCor. The results are reported in Table 2.

Inspecting the results presented in Tables 1-2 leads to the conclusion that the dCor index is highly sensitive to the presence of redundant terms, and is maximal in the absence of redundant terms. This property proves to be crucial for the detection of redundant terms in the FS task.

3 The Proposed Method

3.1 Distributed optimization scheme

The FS is a combinatorial problem whose complexity grows exponentially with the number of features (the number of

TABLE 2
Average dCor measure over 1000 Monte Carlo tests for increasingly redundant models (true model: $y = 3x_1^2$).

Number of redundant features	Data distribution		
	Normal	Poisson	Lognormal
0	0.5731	0.9682	0.9221
1	0.5447	0.9570	0.9106
2	0.5261	0.9490	0.8997
3	0.5120	0.9419	0.8897
4	0.5008	0.9352	0.8808
5	0.4916	0.9289	0.8716

possible feature subsets is equal to $2^{N_f} - 1$). For these reasons the search strategy adopted to explore the feature subset space is crucial. We here employ a distributed combinatorial optimization scheme first employed in [28] that breaks the complexity of the FS task into smaller, less dimensionally unbalanced problems, and iteratively repeats the optimization of the latter after an information exchange stage. Besides allowing the dCor to be used in more favorable conditions, the distributed scheme facilitates the search over the feature subset space and improves the computational efficiency of the FS task.

A sketch of the proposed scheme is depicted in Fig. 1. In the first step, the data are vertically partitioned, *i.e.* the full feature set $\mathcal{F} = \{f_1, ..., f_{N_f}\}$ is (randomly) divided into a number of non-overlapping subsets (denoted feature bins in the sequel) $\mathcal{F}_b^{(0)}$, $b = 1, \ldots, N_b$ of approximately the same size. The number of feature bins N_b is a critical parameter, especially regarding the robustness with respect to the overfitting issue. In this work, following [29], we set $N_b = 2N_f/N$, which results in $N/2$ features for each bin.

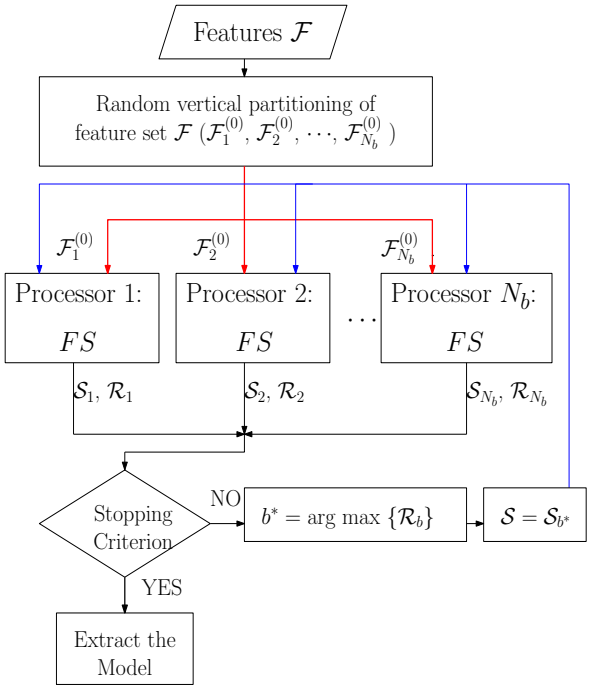


Fig. 1. Flowchart of the proposed distributed scheme.

An independent FS task is carried out on each feature bin $\mathcal{F}_b = \mathcal{F}_b^{(0)}$, according to the selection strategy of choice. The obtained solutions are compared and the one with the best performance (denoted $\mathcal{S}^*$) is shared among all feature bins. In other words, $\mathcal{F}_b = \mathcal{F}_b^{(0)} \cup \mathcal{S}^*$, *i.e.* each feature bin is reset to its initial state $\mathcal{F}_b^{(0)}$ and then augmented with the features corresponding to the current local best solution. The procedure is then repeated iteratively, alternating the execution of the independent FS tasks with the information exchange phase until convergence.

The information exchange phase guarantees that each feature bin contains the features of the best solution found so far, so that the new solution can only improve over the previous best (at least in principle). At the same time the dimension of the feature bins is kept low during all the selection process, so that the individual FS problems have appropriate feature-sample balancing. The procedure terminates when it is not possible to find a better solution than the previous best in any feature bin and all problems yield the same solution (alternative termination conditions can be applied, as explained later on).

In this work we used the dCor as a criterion for selecting the local best solution over the iterative procedure. A pseudocode of the distributed scheme is provided below under the name D^{2}CORFS (Distributed dCor-based FS, see algorithm 1).

Algorithm 1 D^{2}CORFS

Input: $\mathcal{D}$, $\mathcal{F}$, N_b, N_r, N_i, N_p, $\mu^{(0)}$, $\bar{\mu}$, ϵ.
Output: $\mathcal{S}^\star$, $\mathcal{R}^\star$.

1: $\mathcal{F} = \mathcal{F}_1^{(0)} \cup \cdots \cup \mathcal{F}_{N_b}^{(0)}$
2: $\mathcal{S} = \emptyset$, $\mathcal{S}^\star = \emptyset$, $\mathcal{R}^\star = 0$
3: **for** $r = 1$ **to** N_r **do**
4: $\quad \mathcal{S} \leftarrow \mathcal{S}^\star$
5: $\quad$ **for** $b = 1$ **to** N_b **do**
6: $\quad\quad \mathcal{F}_b = \mathcal{F}_b^{(0)} \cup \mathcal{S}$
7: $\quad\quad (\mathcal{S}_b, \mathcal{R}_b)$ = DCORFS(D, $\mathcal{F}_b$, N_i, N_p, $\mu^{(0)}$, $\bar{\mu}$, ϵ)
8: $\quad\quad$ **if** $\mathcal{R}_b > \mathcal{R}^\star$ **then**
9: $\quad\quad\quad \mathcal{S}^\star \leftarrow \mathcal{S}_b$, $\mathcal{R}^\star \leftarrow \mathcal{R}_b$
10: $\quad\quad\quad$ **if** $\mathcal{R}^\star = 1$ **then return end if**
11: $\quad\quad$ **end if**
12: $\quad$ **end for**
13: $\quad$ **if** $\cup_b \mathcal{S}_b = \cap_b \mathcal{S}_b$ **then return end if**
14: $\quad \mathcal{R}^\star_{vec}(r) = \mathcal{R}^\star$
15: $\quad$ **if** $r = N_r$ **then**
16: $\quad\quad$ **return**
17: $\quad$ **else if** $r \geq 3$ **then**
18: $\quad\quad$ **if** $(\mathcal{R}^\star_{vec}(r-1) = \mathcal{R}^\star_{vec}(r-2) = \mathcal{R}^\star)$ **then**
19: $\quad\quad\quad$ **return**
20: $\quad\quad$ **end if**
21: $\quad$ **end if**
22: **end for**

The algorithm employs the procedure DCORFS (see Section 3.2) to solve the local FS problems on the feature bins. Input parameters N_p, N_i, $\mu^{(0)}$, $\bar{\mu}$, and ϵ are actually arguments of the DCORFS function and will be explained later. The other inputs are the set of input/output observation pairs $\mathcal{D}$, the full set of features $\mathcal{F}$, the number of feature bins N_b, and the maximum number of allowed rounds N_r. The algorithm returns the selected feature subset $\mathcal{S}^\star$, along with its dCor value $\mathcal{R}^\star$. Notice that the procedure is terminated if a model with the maximum possible dCor is obtained (line 10), or if all local models are equal (line 13), or if the maximum number of rounds has been reached (line 15), or finally if no improvement is achieved for 3 consecutive rounds (line 18).

3.2 The DCORFS algorithm

In principle, any filter/wrapper method could be implemented to solve the local FS problems in the distributed scheme explained in the previous section, the most popular choice probably being a (multivariate) filter method for

computational reasons, based on a sequential search strategy. As already mentioned, however, sequential strategies have significant drawbacks, essentially originating from the fact that model variations are enforced based on a local assessment of the features (*e.g.*, a new term is added because it significantly improves the *current* model). For this reason, we here introduce a novel multivariate filter FS algorithm based on the unbiased dCor criterion which employs a different search strategy, in that it implements model variations based on a *global* assessment of the features.

The FS problem amounts to solving an optimization problem, whose objective is to find the subset of features $\mathcal{S} \subseteq \mathcal{F}$ that maximizes the dCor index $\mathcal{R}_N(\boldsymbol{f}_\mathcal{S}, c)$. A convenient way to tackle the problem above exploits the probabilistic reformulation of [30] (used to develop a wrapper FS method in [27]) obtained by associating a discrete random variable ϕ to the feature subsets $\mathcal{S}$ according to a probability distribution $\mathcal{P}_\phi$, which expresses the probability of each feature subset $\mathcal{S}$ to coincide with the target one. Accordingly, the dCor index becomes a function of $\boldsymbol{f}_\phi$ and is therefore a random variable with expectation given by

$$\mathbb{E}[\mathcal{R}_N(\boldsymbol{f}_\phi, c)] = \sum_{\mathcal{S}\subseteq\mathcal{F}} \mathcal{R}_N(\boldsymbol{f}_\mathcal{S}, c)\mathcal{P}_\phi(\mathcal{S}). \tag{11}$$

The expected value (11) is maximal if the mass of distribution $\mathcal{P}_\phi$ is all concentrated on the feature subset with highest dCor $\mathcal{S}^\star$. Therefore, the problem of finding $\mathcal{S}^\star$ can be reformulated as that of finding the target limit distribution

$$\mathcal{P}^\star_\phi = \arg\max_{\mathcal{P}_\phi} \mathbb{E}[\mathcal{R}_N(\boldsymbol{f}_\phi, c)], \tag{12}$$

such that $\mathcal{P}^\star_\phi(\mathcal{S}^\star) = 1$.

To model the probability that a feature $f_j \in \mathcal{S}^\star$, we parameterize $\mathcal{P}_\phi$ by associating a Bernoulli random variable ρ_j to each feature f_j:

$$\rho_j \sim \text{Be}(\mu_j), \quad \mu_j \in [0, 1]$$

$j = 1, \ldots, N_f$, where μ_j denotes the feature inclusion probability (FIP) of the jth feature. Initially, the FIPs are set to values which may reflect the prior knowledge on the most promising features or simply assign an equal probability to all of them. Then, the distribution is iteratively refined by taking into account the information gathered by sampling it. More in detail, at every iteration a population of feature subsets is extracted using the current Bernoullian distributions and each feature subset is evaluated with the dCor criterion. Then, all features are assessed individually using (a sampled version of) the index $\mathcal{I}_j$ given by

$$\mathcal{I}_j = \mathbb{E}[\mathcal{R}_N(\boldsymbol{f}_\phi, c)|f_j \in \phi] - \mathbb{E}[\mathcal{R}_N(\boldsymbol{f}_\phi, c)|f_j \notin \phi], \tag{13}$$

for $j = 1, \ldots, N_f$. Index $\mathcal{I}_j$ compares the dCor criterion of the features subsets that include f_j with that of the remaining ones and thus can be interpreted as a global measure of the feature's importance. Finally, the probability distribution is updated according to the update rule given by

$$\mu_j(i+1) = \text{sat}(\mu_j(i) + \gamma\mathcal{I}_j) \tag{14}$$

where i is the current iteration and $\text{sat}(\cdot)$ is a saturating function that ensures that μ_j remains within the $[0, 1]$ interval. Parameter γ in (14) is an adaptive step-size defined as:

$$\gamma = \frac{1}{\lambda(\mathcal{R}_{\max} - \bar{\mathcal{R}}) + 0.1}, \tag{15}$$

where λ is a design coefficient and $\mathcal{R}_{\max}$ and $\bar{\mathcal{R}}$ are the maximum and the average of the dCor values of the extracted feature subsets. The rationale behind γ is that it should be larger if the averaged index $\mathcal{I}_j$ is reliable (small variance of the dCor values), and smaller otherwise.

The iterative procedure terminates upon convergence of the probability distribution (or if the maximum number of iterations is exceeded). The selected feature subset is given by $\mathcal{S}^\star = \{f_j | \mu_j \geq \bar{\mu}\}$, where $\bar{\mu}$ is the prescribed acceptance threshold. A pseudocode of the proposed DCORFS algorithm is given below (see Algorithm 2).

Algorithm 2 DCORFS

Input: $\mathcal{D}$, $\mathcal{F}_b$, N_i, N_p, $\mu^{(0)}$, $\bar{\mu}$, ϵ
Output: $\mathcal{S}^\star$, $\mathcal{R}^\star$

for $j = 1$ **to** $|\mathcal{F}_b|$ **do**
  $\mu_j \leftarrow \mu^{(0)}$ FIP initialization
end for
for $i = 1$ **to** N_i **do**
  for $p = 1$ **to** N_p **do**
    $\phi_p \sim \mathcal{P}_\phi$ Extract sample feature subset
    $\mathcal{R}^p \leftarrow \mathcal{R}_N(\boldsymbol{f}_{\phi_p}, c)$ Compute dCor with (10)
  end for
  $\mathcal{R}_{\max} \leftarrow \max(\mathcal{R}^1, \ldots \mathcal{R}^{N_p})$
  $\bar{\mathcal{R}} = \frac{1}{N_p}\sum_{p=1}^{N_p} \mathcal{R}^p$
  $\gamma = \frac{1}{\lambda(\mathcal{R}_{\max} - \bar{\mathcal{R}}) + 0.1}$
  for $j = 1$ **to** $|\mathcal{F}_b|$ **do**
    $\mathcal{I}_j \leftarrow \frac{\sum_{p|f_j \in \phi_p} \mathcal{R}^p}{\sum_{p|f_j \in \phi_p} 1} - \frac{\sum_{p|f_j \notin \phi_p} \mathcal{R}^p}{\sum_{p|f_j \notin \phi_p} 1}$
    $\mu_j \leftarrow \text{sat}(\mu_j + \gamma\mathcal{I}_j)$ FIP update
  end for
  if $\max_{j=1,\ldots,|\mathcal{F}_b|} |\mu_j(i) - \mu_j(i-1)| \leq \epsilon$ **then**
    break
  end if
end for
$\mathcal{S}^\star \leftarrow \emptyset$
for $j = 1$ **to** $|\mathcal{F}_b|$ **do**
  if $\mu_j \geq \bar{\mu}$ **then** $\mathcal{S}^\star \leftarrow \mathcal{S}^\star \cup \{f_j\}$ **end if**
end for
$\mathcal{R}^\star = \mathcal{R}_N(\boldsymbol{f}_{\mathcal{S}^\star}, c)$

The required inputs are the observations $\mathcal{D}$, the set of features $\mathcal{F}_b$ on which to perform the search, the maximum number of iterations N_i, the number of feature subsets to be extracted from the current distribution at each iteration N_p, the initial value of the FIPs $\mu^{(0)}$, the acceptance threshold $\bar{\mu}$, and a convergence threshold ϵ. The algorithm returns the selected feature subset $\mathcal{S}^\star$, along with its dCor value $\mathcal{R}^\star$.

3.3 Feature screening using dCor

A high-dimensional feature space negatively affects the selection process in many regards, including computational efficiency, statistical accuracy, and algorithm stability [31]. For this reason, an independence feature screening [31] is common practice to reduce the dimensionality to a more

convenient size before the application of the actual FS procedure. We here perform such feature screening by employing again the dCor index to test the dependence of the output on every individual feature. The statistical test proposed in [23] rejects the independence hypothesis if

$$\frac{N\nu_N^2(f_j,c)}{\bar{a}..\bar{b}..} > \mathcal{N}^{-1}(1-\frac{\alpha_d}{2})^2, \tag{16}$$

where $\mathcal{N}(\cdot)$ denotes the normal cumulative distribution function and α_d is the significance level of the test. Inequality (16) is tested for every feature f_j, and only features with enough statistical evidence to reject the independence hypothesis are held for the FS task.

4 Experimental Study

In this section we report the results of a series of experiments carried out to assess the performance of the proposed algorithm on eight well known microarray benchmarks: Breast, CNS (Central Nervous System), Colon, DLBCL (Diffuse Large B-Cell Lymphoma), Leukemia, Lung, Ovarian and Prostate cancers. The Breast dataset provides gene information retrieved from tumor material belonging to breast cancer patients, distinguishing those that developed metastases within 5 years from the other ones. The CNS dataset documents both failed and succeeded treatment cases of embryonal CNS tumors. The Colon dataset contains expression levels of 2000 genes for several colon tissue samples including both normal and cancerous ones. Gene analysis of diagnostic tumor specimens from DLBCL patients having received a specific chemotherapy treatment is reported in the DLBCL dataset, distinguishing between cured and fatal or refractory disease cases. The Leukemia dataset contains gene information extracted from bone marrow and peripheral blood samples of several leukemia patients, corresponding either to Acute Lymphoblast Leukemia (ALL) or Acute Myeloid Leukemia (AML). The Lung dataset provides genetic information regarding both Malignant Pleural Mesothelioma (MPM) and lung ADenoCArcinoma (ADCA) cases. The Ovarian dataset aims to identify ovarian cancer from proteomic patterns in serum. Finally, the Prostate dataset contains the expression level of 12600 genes for more than 100 tissue samples, a part of which are taken from prostate tumors. The main characteristics of the considered microarray datasets are summarized in Table 3 (see [32] for a comprehensive review of these and other microarray datasets and specific references).

All eight datasets are biclass problems. The number of original features ranges from a few thousands to almost 25000. To reduce the feature search space, a dCor-based feature screening (with $\alpha_d \geq 0.9$ [1]) was applied as a preprocessing step to all the datasets except Colon and DLBCL, that already have a sufficiently small feature set. The number of samples is generally relatively small, with the exception of the Ovarian cancer dataset. Table 3 also reports the distribution of the samples over the classes both for the training test and the test set, when applicable (NP and NN are the total number of samples belonging to class 1 and 2, respectively). The class imbalance, measured as the skew ratio $\sigma = \frac{NP}{NN}$, is also given for both the training (σ_{tr}) and the test (σ_{te}) data, respectively. This information is important, since it is related to the achievable accuracy and reliability of classification algorithms across classes [33].

Half of the datasets (Breast, Leukemia, Lung and Prostate) are provided with a given training/test data subdivision, while the CNS, Colon, DLBCL and Ovarian datasets are not. For this reason, we analyzed first the former group of datasets with a Hold-Out Cross Validation (HOCV) method, using the training data to learn the model and the test data for its evaluation. Though the HOCV method is in principle applicable also to the other datasets, the results would be impossible to compare with the literature, in the absence of a nominal training-test data subdivision (subsection 4.4 discusses the sensitivity of the identification results to variations of the data subdivision). Therefore, a second analysis is performed, this time evaluating all datasets with a Leave-One-Out Cross Validation (LOOCV) approach, which is a particular case of k-Folds Cross Validation (k-FCV), with $k = N$. Briefly, the dataset is split into k equal (or, at least, balanced in size) and non-overlapping subsets (folds), possibly uniformly representative of all classes. Then, $k-1$ folds are used for training and the remaining ones for testing, the procedure being repeated k times so that all folds are left once for testing. The algorithm performance is finally computed as the average over the k independent runs.

The original features have been normalized in the $[0,1]$ range according to:

$$\bar{f}_j^{(k)} = \frac{f_j^{(k)} - f_{p_{min}}}{f_{j_{max}} - f_{j_{min}}}, \tag{17}$$

for $k = 1,\ldots,N$, $j = 1,\ldots,N_f$, where $\bar{f}_j^{(k)}$ is the normalized numeric value of the kth observation of the jth feature in a given dataset, and $f_{j,\text{max}}$ and $f_{j,\text{min}}$ denote the maximum and minimum values of the same feature in the dataset, respectively.

To evaluate the performance of the proposed FS method we trained different classifiers on the selected features, namely a support vector machine (SVM) with linear decision boundaries, a k-nearest neighbor (kNN, with $k = 5$) and a naive Bayes (NB) classifier.

We employed various evaluation criteria especially designed to account for class imbalanced data. The sensitivity of the classifier is measured by the true positive rate $TPR = \frac{TP}{TP+FN}$, *i.e.* the ratio of the correctly classified positive samples over the total number of positive samples. Conversely, the specificity is captured by the true negative rate $TNR = \frac{TN}{TN+FP}$, *i.e.* the ratio of the correctly classified negative samples over the total number of negative samples. The *Gmean* $G = \sqrt{TPR \cdot TNR}$ and *Fscore* $F = 2\frac{TPR \cdot TNR}{TPR+TNR}$ indices combine both criteria.

The initial parameter setup for the D^{2}CORFS in the experiments is as follows: a maximum of $N_r = 5$ rounds is allowed for the distributed search scheme and the size of the feature bins is set as close as possible to $N/2$, so as to have ideally balanced datasets in the local FS problems. As for the DCORFS algorithm operating on each feature bin, the number of iterations is limited to $N_i = 100$, the num-

1. Different values of α_d were used in the feature screening process depending on the adopted validation method, since the latter influences the distribution of the samples.

TABLE 3
Main characteristics of the considered microarray datasets.

Dataset	# features	after screening		class labels	Training set			Test set			σ_{tr}	σ_{te}
	total	HOCV	LOOCV		total	NP	NN	total	NP	NN		
Breast	24481	4990	2132	Relapse/non-Relapse	78	34	44	19	12	7	1.29	0.58
CNS	7129	–	1415	Class0/Class1	60	21	39	–	–	–	1.86	
Colon	2000	–	–	*not available*	62	22	40	–	–	–	1.82	
DLBCL	4026	–	–	Cured/Fatal	77	58	19	–	–	–	0.33	
Leukemia	7129	2688	2812	ALL/AML	38	13	25	34	10	24	1.92	2.40
Lung	12533	1872	2585	Mesothelioma/ADCA	32	16	16	149	15	134	1.00	8.93
Ovarian	15154	–	3368	Cancer/Normal	253	162	91	–	–	–	0.57	
Prostate	12600	2053	2472	Relapse/non-Relapse	102	52	50	34	25	9	0.96	0.36

Note. ALL = Acute Lymphoblastic Leukemia, AML = Acute Myeloid Leukemia.

ber of feature subset extractions at each iteration is set to $N_p = 100$, the initial FIPs are set to $\mu_0 = 1/|\mathcal{F}_b|$, $\epsilon = 0.001$, and the acceptance threshold is $\bar{\mu} = 0.98$. The proposed algorithm was implemented in Matlab (version 2016a) and executed on an Intel(R) Core i7-3630QM machine, with 2.4GHz CPU, 8GB of RAM, and a 64-bit Operating System.

4.1 Performance analysis of D^{2}CORFS with HOCV

We first analyze the four datasets with explicit training-test division which can be addressed with the HOCV approach using the native training and test sets. Table 4 reports the best subset of features selected for each case, as well as the performances obtained with linear SVM, kNN and NB classifiers. The results are assessed in terms of the classification accuracy on the training (J_{tr}) and the test data (J_{te}), as well as TPR, TNR, G, and F. Apparently, the FS procedure selected very compact models in all cases, with 4 features at most, and a high classification accuracy was obtained (the results compare quite favorably with the literature, as shown later in Table 8). Interestingly enough, though the Leukemia data are imbalanced in favor of negative samples, the obtained classifiers score better on the TPR index, than on the TNR. The computational time is sufficiently low, the lowest computational cost having being observed for the Lung dataset. Indeed, the Lung dataset has the smallest number of samples, which in turn causes the size of the feature bins to be particularly small resulting in a very high computational efficiency.

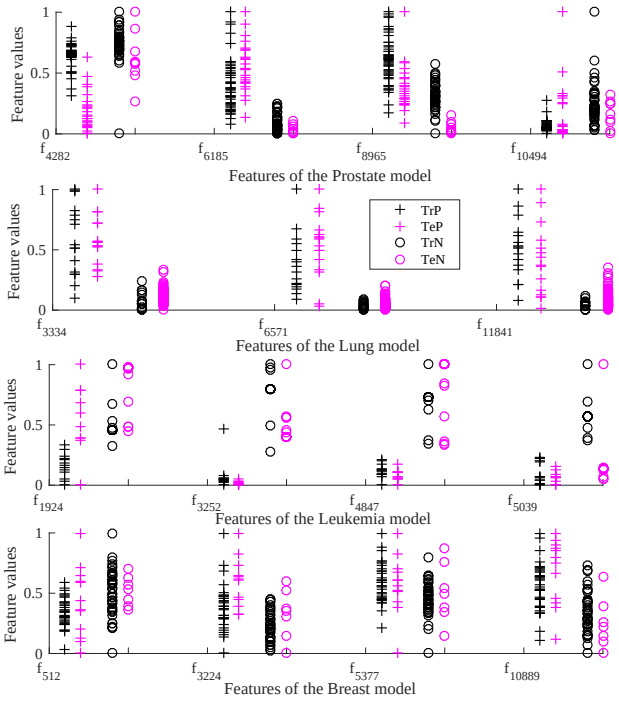


Fig. 2. Distribution of the feature values of the models presented in Table 4.

4.2 Distribution of the feature values

A more detailed analysis of the obtained models, with focus on the selected features, reveals several interesting aspects. Fig. 2 shows the values of the selected features for all samples, divided by class and training/test subset.

Models characterized by perfect performance on the training set have features with little or no overlap between different classes (see, *e.g.*, Lung and Leukemia datasets). This indicates that a perfectly legitimate model selection was operated based on the available information (training set). Unfortunately, the feature value distributions over classes turns out to be different on the test set, typically resulting in some classification errors. Such imprecision could not have been avoided based on the information gathered from the training set, if not by luck. In other words, if a subset of features provides good class discrimination on the training set, it will provide good generalization only if the feature value distributions on the training and the test subsets are similar. It may well happen that better generalization is achieved through a model which is not optimal on the training set.

4.3 Redundancy analysis of obtained models

We performed an *a posteriori* analysis on the obtained models, both in terms of the dCor measure and the classifier performance, to investigate the presence of redundant biological information. The test is performed by removing one gene at a time from $\mathcal{S}^\star$, and re-evaluating the reduced feature subset. Table 5 reports the obtained results.

By inspecting Table 5, it is apparent that the dCor index indicates the absence of redundant terms in all selected models (the full model has the highest dCor). If the performance index on the training set J_{tr} were used as a selection criterion (as would happen, *e.g.*, with a wrapper method), a smaller model would have been selected in the Prostate

TABLE 4
Performance of the best models obtained with D^2CORFS and HOCV.

Dataset	Best model ($\mathcal{S}^\star$)	Time [s]	Classifier	J_{tr}	J_{te}	TNR_{te}	TPR_{te}	G_{te}	F_{te}
Breast	$\{f_{512}, f_{3224}, f_{5377}, f_{10889}\}$	386.29	SVM	0.8077	0.8947	0.8571	0.9166	0.8864	0.8859
			KNN	0.8462	0.8421	0.8571	0.8333	0.8451	0.8450
			NB	0.8590	0.8947	0.8571	0.9166	0.8864	0.8859
Leukemia	$\{f_{1924}, f_{3252}, f_{4847}, f_{5039}\}$	154.01	SVM	1.0000	0.9118	0.8750	1.0000	0.9354	0.9333
			KNN	1.0000	0.9118	0.8750	1.0000	0.9354	0.9333
			NB	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Lung	$\{f_{3334}, f_{6571}, f_{11841}\}$	51.72	SVM	0.9688	0.9933	1.0000	0.9333	0.9661	0.9655
			KNN	0.9688	1.0000	1.0000	1.0000	1.0000	1.0000
			NB	1.0000	0.9128	0.9030	1.0000	0.9503	0.9490
Prostate	$\{f_{4282}, f_{6185}, f_{8965}, f_{10494}\}$	295.78	SVM	0.9216	0.9706	1.0000	0.9600	0.9798	0.9796
			KNN	0.9510	0.9118	0.8889	0.9200	0.9043	0.9042
			NB	0.9314	0.8529	1.0000	0.8000	0.8944	0.8889

TABLE 5
Redundancy analysis on the best models (see Table 4) for data with given training and test set.

Dataset	Classifier	Feature subset	$\mathcal{R}_N$	J_{tr}	J_{te}
Breast	NB	$\mathcal{S}^\star$	0.6126	0.8590	0.8947
		$\mathcal{S}^\star \setminus \{f_{512}\}$	0.6014	0.8333	0.7895
		$\mathcal{S}^\star \setminus \{f_{3224}\}$	0.5858	0.7821	0.6842
		$\mathcal{S}^\star \setminus \{f_{5377}\}$	0.5958	0.8205	0.8421
		$\mathcal{S}^\star \setminus \{f_{10889}\}$	0.5450	0.7949	0.7895
Leukemia	NB	$\mathcal{S}^\star$	0.9782	1.0000	1.0000
		$\mathcal{S}^\star \setminus \{f_{1924}\}$	0.9749	1.0000	0.9118
		$\mathcal{S}^\star \setminus \{f_{3252}\}$	0.9734	1.0000	0.8824
		$\mathcal{S}^\star \setminus \{f_{4847}\}$	0.9770	1.0000	0.9412
		$\mathcal{S}^\star \setminus \{f_{5039}\}$	0.9707	1.0000	1.0000
Lung	KNN	$\mathcal{S}^\star$	0.9150	0.9688	1.0000
		$\mathcal{S}^\star \setminus \{f_{3334}\}$	0.8833	0.9688	0.9732
		$\mathcal{S}^\star \setminus \{f_{6571}\}$	0.8990	0.9688	0.9933
		$\mathcal{S}^\star \setminus \{f_{11841}\}$	0.8755	0.9688	1.0000
Prostate	SVM	$\mathcal{S}^\star$	0.8216	0.9216	0.9706
		$\mathcal{S}^\star \setminus \{f_{4282}\}$	0.8108	0.9216	0.8529
		$\mathcal{S}^\star \setminus \{f_{6185}\}$	0.7848	0.9020	0.9118
		$\mathcal{S}^\star \setminus \{f_{8965}\}$	0.8008	0.9412	0.9706
		$\mathcal{S}^\star \setminus \{f_{10494}\}$	0.8145	0.9020	0.9412

case, but without any improvement on the test data. Observe that in all four cases the model with the highest dCor achieves the best test performance. In general, the dCor-based filter method provides a good guess of the optimal model both in terms of size and performance, although additional accuracy improvements could occasionally be obtained complementing it with a wrapper method that optimizes directly on the classifier performance.

4.4 Model sensitivity on the data subdivision

To analyze the model sensitivity on the data subdivision in training and test data, we took the best model (see Tab. 4) obtained using the nominal training-test subdivision of the Leukemia dataset, and evaluated its performance with a Monte Carlo test over a 1000 random training-test data subdivisions (generated so as to preserve the distribution among classes). On each run, the selected features are the same but the classifier is re-estimated on the corresponding training subset and evaluated on the test subset. The results are presented in Fig. 3, and show a non-neglectable sensitivity to the training-test data subdivisions. Indeed, the same performance of the nominal case is re-obtained less than 50% of the times, and on almost 10% of the runs a performance as low as $J_{te} = 0.91$ is achieved (corresponding to 3 errors over the 34 test samples). One possible explanation of this phenomenon is that there are some isolated samples which cannot be learnt by the model if they fall in the test portion of the data.

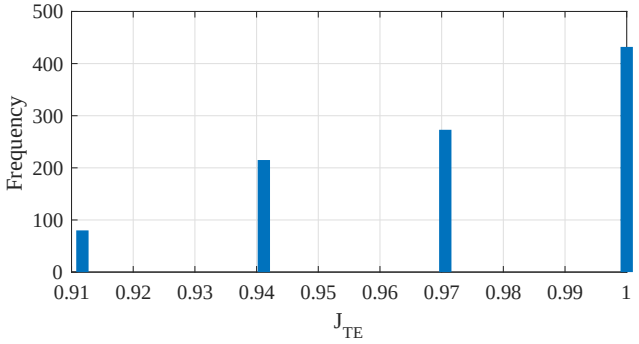


Fig. 3. Accuracy of the Leukemia model obtained with the nominal training-test data splitting over 1000 alternative data-splittings.

4.5 Performance analysis of D^{2}CORFS with LOOCV

The previous analysis confirms that the data subdivision is extremely critical and can greatly affect the quality of the results and ultimately the assessment of an algorithm. Applying the HOCV approach in the absence of a nominal training-test data subdivision would lead to results of questionable objectivity, and difficult to compare with the existing literature. For this reason, we carried out a different analysis on the available datasets using LOOCV, which is a Cross Validation method that does not depend on a specific data subdivision (as HOCV). Table 6 presents the results obtained following the LOOCV approach on all the datasets (the results are averaged over 10 repetitions). For each dataset, the best selected model structure is reported, together with the average computation time. Both the performance of the best model overall and the average performance of the best models over 10 runs are given, for the three types of classifiers considered.

It is interesting to note that where both the HOCV and LOOCV methods have been applied (*i.e.*, for Breast, Leukemia, Lung, and Prostate), the obtained models have a scarce overlap in terms of model structure. More specifically, the models obtained for the Breast dataset have 2 mutual

features (but extremely different size), and the Lung and Prostate models have just one feature in common, while a totally different model structure was obtained in the remaining case. This is yet another indication of the impact that data subdivision can have on the results.

4.6 Diversity analysis of high performance models

Due to the discrete nature of the classification problem, multiple optimal (*i.e.*, with the same maximal accuracy) models can be obtained, which can make the model interpretation awkward. As an example, we explore this phenomenon with reference to the Leukemia dataset, where a model with 0 classification errors was previously obtained with LOOCV and an NB classifier (see Table 6). We repeated the selection process multiple times, each time forcing the exclusion of one of the regressors belonging to one of the previously selected models. This procedure enforces that at each repetition of the algorithm a different best model will be obtained. Table 7 shows several instances of models with different structure but equal accuracy that are obtained in this way. These models contain combinations of two or three regressors taken from a restricted set of seven. Notice that some of the models have no common regressor. This suggests that there are different groups of the genes which contain the same amount of useful information to distinguish among the classes. This phenomenon could have several explanations, ranging from the high correlation of features, to the insufficient information carried by the training set.

4.7 Complexity analysis

We here analyze the computational complexity of the proposed algorithm as a function of the problem size (*i.e.*, the number of features N_f and samples N), and of some crucial design parameters (*e.g.*, the number of rounds of the D^{2}CORFS algorithm N_r, the number of iterations of the DCORFS algorithm N_i, and the number of the feature bins N_b). Let $\mathcal{F}$ be the full set of N_f features. Clearly, the model space to be explored grows exponentially with the number of features (the number of possible non-empty subsets of $\mathcal{F}$ is $2^{N_f}-1$. At every iteration, each processor executes the DCORFS algorithm on its feature bin, which performs three tasks: feature subset extraction and evaluation, regressor evaluation, and RIP update. The first task requires $N_p N'_f$ operations, where N_p is the number of feature subsets to be evaluated, and $N'_f \simeq N_f/N_b$ is the number of features in each feature bin. The evaluation of a feature subset by means of Equation (10) is of order $O(N^2 N'_f)$, whereas the calculation of all the indices $\mathcal{I}_j$, $j=1,\ldots,N'_f$ requires an order of $N_p N'_f$ operations. Finally, the RIP update is linear in the number of features in the bin, *i.e.* $O(N'_f)$. The complexity of the DCORFS is then $O(N_i N'_f(N^2+N_p))$, which is typically dominated by the first term. The complexity of the overall distributed scheme D^{2}CORFS is dominated by its main cycle which repeats up to N_r times the DCORFS on N_b feature bins, for an overall complexity of $O(N_r N_b N_i N'_f(N^2+N_p)) \simeq O(N_r N_f N_i N^2))$.

An experimental characterization of the algorithm time complexity has also been carried out, the results of which are shown in Fig. 4. More precisely, Fig. 4 (top) reports the elapsed time for the DCORFS algorithm, averaged over ten runs (such that the features are reshuffled at random in each run), as a function of the number of features in the bin and the number of iterations. The curves are characterized by an initial increase of the computational time with the growth of the feature space, followed by a saturation associated with the reaching of the maximum allowed number of iterations. Indeed, as N_i increases, the saturation point shifts to the right. These curves can be used to properly set N_i with respect to the number of features in the bin, in order to obtain convergence prior to the saturation point. Fig. 4 (bottom) analyzes the elapsed time of the overall D^{2}CORFS algorithm, as a function of the problem sizes N_f and the number of bins N_b. As can be seen from the figure, it is quite apparent that the execution time decreases rapidly as the number of bins increases, but at a certain point it starts increasing again, though at a slower rate. This result emphasizes the importance of the N_b design parameter. If the number of bins is chosen too sparingly, the bin size will be too large, thus leading to insufficient search space reduction. This ultimately defies the very purpose of the distributed scheme, *i.e.* to break the problem complexity, and thus slows down the convergence of the algorithm. Conversely, if one employs too many small bins, most of these will initially not contain any useful feature and will presumably return inaccurate results, whereas only the processors associated to bins that contain features of the true model will typically produce meaningful results. As a consequence of this, the algorithm will require more rounds and thus more time to converge.

4.8 Comparative analysis with results in the literature

As already commented, a meaningful comparison can be obtained only if the same training-test data distribution is employed. For this reason we divided this comparative analysis into two parts depending on the cross validation method employed. First, we consider the Breast, Leukemia, Lung, and Prostate datasets using a HOCV approach. A reliable comparison is possible, since these datasets are provided with a nominal training-test distribution. Then we consider all eight databases of Table 3 using a LOOCV approach, which employs the data for training and testing in a unique and consistent way.

Table 8 reports a comparison with the results documented in [29], [34] and [35], which consider the Breast, Leukemia, Lung, and Prostate datasets using a HOCV approach based on the nominal training-test distribution of the data. To account for the randomized nature of the D^{2}CORFS algorithm (due to the random distribution of the features in the bins and to the nature of the DCORFS algorithm employed on each local FS sub-problem), we present the averaged results of 5 independent runs besides the best ones. Both the classification accuracy on the test set and the model size are reported ($\bar{J}_{te}$ and $\overline{|\mathcal{S}|}$ denote the averages, and $J^{\star}_{te}$ and $|\mathcal{S}^{\star}|$ the values associated to the best models, respectively). Apparently, the proposed method achieves comparable performance with respect to the best of the competitor methods. Moreover, the obtained models are extremely compact in terms of the number of selected features, which indicates the effectiveness of the FS

TABLE 6
Performance of the best models obtained with D^2CORFS and LOOCV.

Dataset	Best model ($\mathcal{S}^\star$)	Time [s]	Classifier	$J^\star_{te}$	J_{te}
Breast	$\{f_{512}, f_{1205}, f_{1872}, f_{3232}, f_{3773}, f_{4382}, f_{5098}, f_{6859}, f_{7127}, f_{7997}, f_{8776}, f_{10827}, f_{10889}, f_{12275}, f_{12437}, f_{12572}, f_{13800}, f_{17881}, f_{19694}, f_{19906}, f_{20437}, f_{22422}, f_{23322}\}$	8137.83	SVM	0.8969	0.8598
			KNN	0.8454	0.8392
			NB	0.8041	0.8083
CNS	$\{f_{320}, f_{1054}, f_{2496}, f_{2513}, f_{3320}, f_{3731}, f_{4484}, f_{4509}\}$	123.89	SVM	0.9000	0.8583
			KNN	0.8167	0.8350
			NB	0.8500	0.8450
Colon	$\{f_{249}, f_{377}, f_{765}, f_{1482}, f_{1644}, f_{1772}\}$	584.08	SVM	0.8871	0.8823
			KNN	0.8710	0.8581
			NB	0.9194	0.9065
DLBCL	$\{f_{57}, f_{209}, f_{1807}, f_{2115}, f_{2208}\}$	588.29	SVM	0.9870	0.9597
			KNN	0.9740	0.9584
			NB	0.9740	0.9468
Leukemia	$\{f_{2288}, f_{6041}\}$	170.02	SVM	0.9722	0.9722
			KNN	0.9722	0.9722
			NB	1.0000	1.0000
Lung	$\{f_{3334}, f_{4336}, f_{7200}, f_{8370}\}$	2239.40	SVM	1.0000	0.9939
			KNN	0.9945	0.9923
			NB	0.9779	0.9751
Ovarian	$\{f_{182}, f_{1680}, f_{2236}\}$	3383.12	SVM	1.0000	1.0000
			KNN	1.0000	1.0000
			NB	1.0000	1.0000
Prostate	$\{f_{5314}, f_{6185}, f_{9850}, f_{11052}\}$	4042.18	SVM	0.8456	0.7728
			KNN	0.9338	0.8765
			NB	0.8162	0.7559

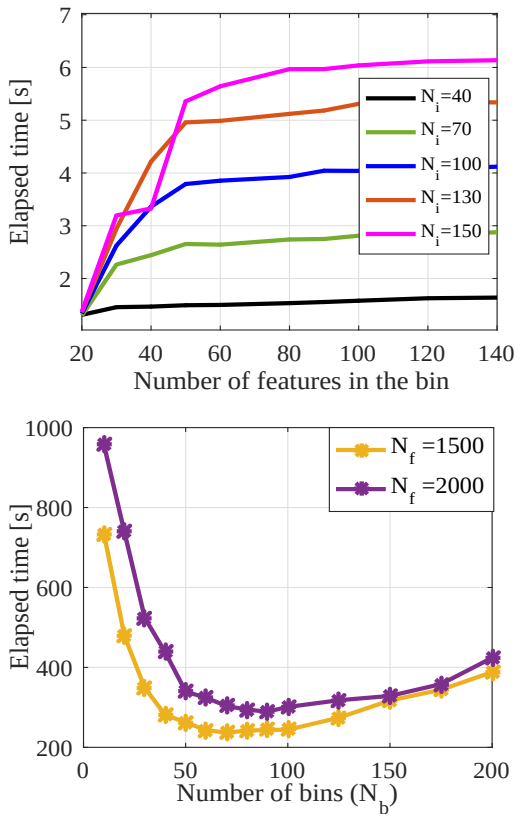


Fig. 4. Complexity analysis: Dependency of the DCORFS execution time on the number of features in the bin and the number of iterations (top), dependency of the D^{2}CORFS execution time on the problem size and the number of bins (bottom).

TABLE 7
Diversity analysis of the models with maximum accuracy (0 classification errors) for the Leukemia dataset, obtained with LOOCV and a NB classifier.

Feature	$\mathcal{S}^*$	$\mathcal{S}^*_1$	$\mathcal{S}^*_2$	$\mathcal{S}^*_3$	$\mathcal{S}^*_4$	$\mathcal{S}^*_5$
f_{2288}	✓					✓
f_{4052}		✓	✓	✓	✓	
f_{4167}		✓			✓	
f_{4230}				✓		
f_{4328}			✓			
f_{4847}					✓	✓
f_{6041}	✓	✓	✓	✓		

approach in pointing out to the expert the really important features.

Table 9 provides a comparison with the methods documented in the literature that study all the eight datasets considered in this work, using a LOOCV approach. The proposed method systematically provides a promising performance, scoring better or equivalently to the competitor methods on five out of eight datasets, and generally ranking among the best methods. In the case for which more documented results can be found in the literature (Leukemia), it obtains perfect performance both on the training and the test set, using only 2 features. It is also confirmed that the method tends to provide a good trade-off between accuracy and compactness of the selected model, which is important both for the robustness of the classifier and for model interpretation purposes.

5 Conclusions

A novel FS method has been developed that is especially designed for large and dimensionally unbalanced classification problems, such as those that arise in connection with

TABLE 8
Comparative analysis with the HOCV approach.

Dataset	Method	$\bar{J}_{te}$	$J^{\star}_{te}$	$\overline{\lvert\mathcal{S}\rvert}$	$\lvert\mathcal{S}^{\star}\rvert$
Breast	D^{2}CORFS	0.67	0.89	6.8	4
	DRF+SVM [29]	–	0.84	–	97
	DRF+kNN [29]	–	0.79	–	52
	DRF+NB [29]	–	0.79	–	40
Leukemia	D^{2}CORFS	0.93	1.00	3.2	4
	DRF+SVM [29]	–	0.91	–	15
	DRF+kNN [29]	–	0.94	–	4
	DRF+NB [29]	–	0.94	–	6
	ABC+DANN [35]	0.88	0.94	3.0	3
	L1-norm SVM [34]	0.84	–	24.9	–
	Elastic Net [34]	0.84	–	36.7	–
	PAEN [34]	0.85	–	21.9	–
	DrSVM [34]	0.85	–	67.7	–
	WDRSVM [34]	0.86	–	19.9	–
	GA+NN [47]	–	0.97	–	2
Lung	D^{2}CORFS	0.97	1.00	2.9	3
	DRF+SVM [29]	–	0.96	–	2
	DRF+kNN [29]	–	0.98	–	4
	DRF+NB [29]	–	0.99	–	8
	L1-norm SVM [34]	0.84	–	29.1	–
	Elastic Net [34]	0.84	–	39.4	–
	PAEN [34]	0.85	–	26.3	–
	DrSVM [34]	0.86	–	54.4	–
	WDRSVM [34]	0.86	–	23.8	–
Prostate	D^{2}CORFS	0.90	0.97	3.8	4
	DRF+SVM [29]		0.97	–	30
	DRF+kNN [29]		0.62	–	35
	DRF+NB [29]		0.26	–	12

TABLE 9
Comparative analysis with the LOOCV approach.

Dataset	Method	$\bar{J}_{te}$	$J^{\star}_{te}$	$\overline{\lvert\mathcal{S}\rvert}$	$\lvert\mathcal{S}^{\star}\rvert$
Breast	D^{2}CORFS	0.84	0.90	21.9	23
	Local-learning based [21]	–	0.78	–	4
	α DD [36]	0.69	0.88	–	–
CNS	D^{2}CORFS	0.84	0.90	6.7	8
	α DD [36]	0.72	0.90	–	–
Colon	D^{2}CORFS	0.88	0.92	8.0	6
	α DD [36]	0.83	0.92	–	–
	Filter mRMR+SVM [15]	–	0.89	–	4
	Filter mRMR+RVM [15]	–	0.94	–	7
	RMIFS+NB [16]	–	0.97	–	6
	RMIFS+ID3 [16]	–	0.95	–	6
	RMIFS+Logistic [16]	–	1.00	–	6
	ERGS [39]	–	0.84	–	100
	HGA-SVM [45]	0.99	1.00	15	10
	GA+SVM [46]	–	0.93	–	12
DLBCL	D^{2}CORFS	0.95	0.99	7.1	5
	Local-learning based [21]	–	0.97	–	10
	α DD [36]	0.71	0.88	–	–
	MOBBBO [37]	1.00	1.00	5.7	5
	LSLS [38]	–	0.82	–	10
Leukemia	D^{2}CORFS	0.98	1.00	2.0	2
	α DD [36]	0.92	0.97	–	–
	Filter mRMR+SVM [15]	–	0.97	–	4
	Filter mRMR+RVM [15]	–	1.00	–	3
	RMIFS+NB [16]	–	1.00	–	4
	RMIFS+ID3 [16]	–	1.00	–	4
	RMIFS+Logistic [16]	–	1.00	–	4
	LSLS [38]	–	0.81	–	50
	ERGS [39]	–	1.00	–	80
	SL-RFE [40]	–	0.94	–	20
	FS-RFE [40]	–	0.94	–	40
	MRMR [14]	1.00	–	6.0	–
	CFS [41]	0.91	–	1.0	–
	MBF [42]	1.00	–	9.0	–
	R1-GA-kNN [43]	0.98	1.00	50	50
	HGA-SVM [45]	1.00	1.00	32.8	25
	GA+SVM [46]	–	1.00	–	6
Lung	D^{2}CORFS	0.99	1.00	4.4	3
	α DD [36]	0.98	1.00	–	–
	LSLS [38]	–	0.99	–	30
	ERGS [39]	–	1.00	–	100
Ovarian	D^{2}CORFS	1.00	1.00	3.0	3
Prostate	D^{2}CORFS	0.80	0.93	3.3	4
	Local-learning based [21]	–	0.84	–	6
	α DD [36]	0.91	0.96	–	–
	MOBBBO [37]	0.98	1.00	11.9	12
	LSLS [38]	–	0.74	–	25
	ERGS [39]	–	0.94	–	10
	GA+SVM [44]	0.77	–	–	–
	GA+kNN [44]	0.84	–	–	–
	GA+NB [44]	0.76	–	–	–
	HGSA+SVM [44]	0.88	–	–	–
	HGSA+kNN [44]	0.86	–	–	–
	HGSA+NB [44]	0.80	–	–	–

microarrays. Its strength resides on three pillars, namely an evaluation criterion for candidate feature subsets based on the distance correlation concept, a distributed optimization approach, and a randomized selection procedure. The dCor index appears to be a particularly robust criterion with respect to overfitting and redundancy issues, which are common with multivariate filter methods. The distributed combinatorial optimization scheme is used to handle the severe asymmetry of microarray datasets, by dividing the feature set into several feature bins and running independently the FS algorithm on each of them. The best solutions are retained and shared among the feature bins and the procedure is iterated until convergence. Thanks to this "divide et impera" approach, the FS algorithm is always employed on small and dimensionally balanced datasets, for better accuracy and reliability of the results, as well as a reduced computational complexity. The FS algorithm at the core of the method introduces another factor that improves the reliability of the method, in that it re-enforces the probability to select a feature based on an aggregate performance evaluation of a population of feature subsets, which allows for a more reliable assessment of the importance of that particular feature. The overall method has been tested on several microarray benchmark datasets, with quite promising results. Indeed, the resulting classifiers achieve high accuracy levels while using information only from an extremely small number of features.

REFERENCES

[1] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[2] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Advances in bioinformatics*, vol. 2015, pp. 1–13, 2015. Article ID 198363.

[3] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 1, pp. 131–156, 1997.

[4] S. Kotsiantis, "Feature selection for machine learning classification problems: a recent overview," *Artificial Intelligence Review*, pp. 1–20, 2011.

[5] T. Jirapech-Umpai and S. Aitken, "Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes," *BMC bioinformatics*, vol. 6, no. 1, p. 148, 2005.

[6] E. Alba, J. Garcia-Nieto, L. Jourdan, and E.-G. Talbi, "Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms," in *IEEE Congress on Evolutionary Computation (CEC*

2007), pp. 284–290, 2007.

[7] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe, "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1106–1119, 2012.

[8] P. E. Meyer and G. Bontempi, "On the use of variable complementarity for feature selection in cancer classification," in *Workshops on Applications of Evolutionary Computation*, pp. 91–102, 2006.

[9] W. Duch, T. Winiarski, J. Biesiada, and A. Kachel, "Feature selection and ranking filters," in *Int. Conf. on Artificial Neural Networks (ICANN) and Int. Conf. on Neural Information Processing (ICONIP)*, (Istanbul, Turkey), pp. 251–254, June 2003.

[10] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Machine learning*, vol. 53, no. 1-2, pp. 23–69, 2003.

[11] Y. Leung and Y. Hung, "A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 1, pp. 108–117, 2010.

[12] H. Peng and F. Long, "An efficient max-dependency algorithm for gene selection," in 36^{th} *Symposium on the interface: Computational Biology and Bioinformatics*, vol. 57, (Baltimore (Maryland), USA), p. 65, May 26–29 2004.

[13] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[14] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of bioinformatics and computational biology*, vol. 3, no. 2, pp. 185–205, 2005.

[15] M. Soltani, M. H. Shammakhi, S. Khorram, and H. Sheikhzadeh, "Combined mRMR filter and sparse Bayesian classifier for analysis of gene expression data," in *Int. Conf. on Signal Processing and Intelligent Systems (ICSPIS)*, pp. 1–5, 2016.

[16] J. Tang and S. Zhou, "A new approach for feature selection from microarray data based on mutual information," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 6, pp. 1004–1015, 2016.

[17] M. A. Hall, *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[18] D. Koller and M. Sahami, "Toward optimal feature selection," tech. rep., Stanford InfoLab, 1996.

[19] D. A. Bell and H. Wang, "A formalism for relevance and its application in feature subset selection," *Machine learning*, vol. 41, no. 2, pp. 175–195, 2000.

[20] Y. Sun and J. Li, "Iterative RELIEF for feature weighting," in *Proceedings of the* 23^{rd} *Int. Conf. on Machine learning*, pp. 913–920, 2006.

[21] Y. Sun, S. Todorovic, and S. Goodison, "Local-learning-based feature selection for high-dimensional data analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1610–1626, 2010.

[22] D. Shalon, S. J. Smith, and P. O. Brown, "A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization.," *Genome research*, vol. 6, no. 7, pp. 639–645, 1996.

[23] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *The Annals of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[24] G. J. Székely and M. L. Rizzo, "The distance correlation t-test of independence in high dimension," *Journal of Multivariate Analysis*, vol. 117, pp. 193–213, 2013.

[25] C. D. Yenigün and M. L. Rizzo, "Variable selection in regression using maximal correlation and distance correlation," *Journal of Statistical Computation and Simulation*, vol. 85, no. 8, pp. 1692–1705, 2015.

[26] R. Li, W. Zhong, and L. Zhu, "Feature screening via distance correlation learning," *Journal of the American Statistical Association*, vol. 107, no. 499, pp. 1129–1139, 2012.

[27] A. Brankovic, A. Falsone, M. Prandini, and L. Piroddi, "A feature selection and classification algorithm based on randomized extraction of model populations," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–12, 2017.

[28] A. Brankovic and L. Piroddi, "A distributed feature selection scheme with partial information sharing," *submitted paper, available on request*, 2017.

[29] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Distributed feature selection: An application to microarray data classification," *Applied soft computing*, vol. 30, pp. 136–150, 2015.

[30] A. Falsone, L. Piroddi, and M. Prandini, "A randomized algorithm for nonlinear model structure selection," *Automatica*, vol. 60, pp. 227–238, 2015.

[31] J. Fan, R. Samworth, and Y. Wu, "Ultrahigh dimensional feature selection: beyond the linear model," *Journal of Machine Learning Research*, vol. 10, pp. 2013–2038, Sep 2009.

[32] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Information Sciences*, pp. 111–135, 2014.

[33] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[34] J. Li, Y. Wang, Y. Cao, and C. Xu, "Weighted doubly regularized support vector machine and its application to microarray classification with noise," *Neurocomputing*, vol. 173, pp. 595–605, 2016.

[35] B. A. Garro, K. Rodríguez, and R. A. Vazquez, "Designing artificial neural networks using differential evolution for classifying DNA microarrays," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2767–2774, 2017.

[36] X. Wang and O. Gotoh, "A robust gene selection method for microarray-based cancer classification," *Cancer informatics*, vol. 9, p. 15, 2010.

[37] X. Li and M. Yin, "Multiobjective binary biogeography based optimization for feature selection using gene expression data," *IEEE Transactions on NanoBioscience*, vol. 12, no. 4, pp. 343–353, 2013.

[38] B. Liao, Y. Jiang, W. Liang, W. Zhu, L. Cai, and Z. Cao, "Gene selection using locality sensitive Laplacian score," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 6, pp. 1146–1156, 2014.

[39] B. Chandra and M. Gupta, "An efficient statistical feature selection approach for classification of gene expression data," *Journal of biomedical informatics*, vol. 44, no. 4, pp. 529–535, 2011.

[40] D. Wei, S. Li, and M. Tan, "Graph embedding based feature selection," *Neurocomputing*, vol. 93, pp. 115–125, 2012.

[41] Y. Wang, I. V. Tetko, M. A. Hall, E. Frank, A. Facius, K. F. Mayer, and H. W. Mewes, "Gene selection from microarray data for cancer classification–a machine learning approach," *Computational biology and chemistry*, vol. 29, no. 1, pp. 37–46, 2005.

[42] E. P. Xing, M. I. Jordan, R. M. Karp, *et al.*, "Feature selection for high-dimensional genomic microarray data," in *Proceedings of the* 18^{th} *Int. Conf. on Machine Learning (ICML '01)*, vol. 1, pp. 601–608, 2001.

[43] T. Jirapech-Umpai, and S. Aitken,"Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes",*BMC bioinformatics*, vol. 6, no. 1, pp. 148, 2005.

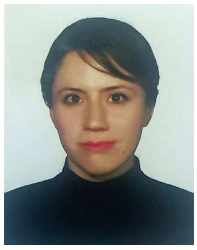
[44] M. Perez, and M. Tshilidzi, "Microarray data feature selection using hybrid genetic algorithm simulated annealing", *Electrical and Electronics Engineers in Israel (IEEEI)*, 2012 IEEE 27th Convention, pp. 1–5, 2012.

[45] E. B. Huerta, B. Duval, and J. -K. Hao, "A hybrid GA/SVM approach for gene selection and classification of microarray data."*Workshops on Applications of Evolutionary Computation*, pp. 34-44. Springer, Berlin, Heidelberg, 2006.

[46] S. Peng, Q. Xu, X. B. Ling, X. Peng, W. Du, and L. Chen, " Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines", FEBS letters, vol. 555, no. 2, pp.358-362, 2003.

[47] J. M. Deutsch, "Evolutionary algorithms for finding optimal gene sets in microarray prediction." Bioinformatics, vol. 19, no. 1, pp. 45-52, 2003.

Aida Brankovic was born in Sarajevo (Bosnia and Herzegovina), in 1987. In 2009 she received her bachelor degree and in 2011 her master degree, both in Automation, Control and Electronics from University of Sarajevo. From February to September 2013 she worked as a teaching assistant at the Electrical Faculty of University of Sarajevo, and from September to November 2013 as research assistant in the MOVE research group of the Politecnico di Milano. In November 2013 she started her PhD at the Dipartimento di Elettronica, Informazione e Bioingegneria of the Politecnico di Milano, Systems and Control section. Her current interests include nonlinear model identification, randomized algorithms and supervised machine learning.



Marjan Hosseini was born in Tehran, Iran. She received the bachelor degree in Computer Engineering from Shomal University, Amol, Iran, in 2006. She is currently pursuing the master degree in Computer Science and Engineering at the Politecnico di Milano, Milan, Italy. Her research interests are machine learning, signal and image processing.

Luigi Piroddi (M'07) was born in London, U.K., in 1966. He received his laurea degree in Electrical Engineering and the Ph.D. degree in Computer Science and Control Theory from the Politecnico di Milano, Milano, Italy, in 1990 and 1995, respectively. Between 1994 and 1999, he was a Professor of fundamentals of automation with the Universitá degli Studi di Bergamo, Bergamo, Italy. From 1999 to 2004, he was an Assistant Professor with the Politecnico di Milano. From 2004 to 2015 he has been an Associate Professor, and from 2016 he is Full Professor with the same institution, where he holds various courses in the systems and control area. His research interests include nonlinear model identification, Petri nets, modeling, and control of manufacturing processes. He currently serves on the editorial board of the IEEE Transactions on Automation Science and Engineering.