A Filter-based Evolutionary Approach for Selecting Features in High-Dimensional Micro-array Data

Laura Maria Cannas, Nicoletta Dessì and Barbara Pes

Università degli Studi di Cagliari, Dipartimento di Matematica e Informatica,
Via Ospedale 72, 09124 Cagliari, Italy
{lauramcannas, dessi, pes}@unica.it

Abstract. Evolutionary algorithms have received much attention in extracting knowledge on high-dimensional micro-array data, being crucial to their success a suitable definition of the search space of the potential solutions. In this paper, we present an evolutionary approach for selecting informative genes (features) to predict and diagnose cancer. We propose a procedure that combines results of filter methods, which are commonly used in the field of data mining, to reduce the search space where a genetic algorithm looks for solutions (i.e. gene subsets) with better classification performance, being the quality (fitness) of each solution evaluated by a classification method. The methodology is quite general because any classification algorithm could be incorporated as well a variety of filter methods. Extensive experiments on a public micro-array dataset are presented using four popular filter methods and SVM.

Keywords: Evolutionary algorithms**,** Feature Selection, Micro-array Data Analysis.

1 Introduction

Evolutionary strategies are now an active area of research and a lot of studies demonstrate the advantages of their use in several knowledge extraction tasks. In particular, recent literature [1][2][3][4] demonstrates their success on micro-array data analysis. The micro-arrays provide a view onto cellular organization of life through quantitative data on gene expression levels and it is expected that knowledge gleaned from micro-array data will contribute significantly to advances in fundamental questions in biology as well as in clinical medicine. In particular, these data may be used to extract knowledge on the molecular variation among cancer i.e. to build a model, namely a classifier, capable of discriminating between different clinical outcomes in order to make accurate prediction and diagnosis of cancer. Building such a classifier is somewhat problematic since in micro-array datasets the number of samples collected is small compared to the number of genes per sample which are usually in the thousands. Since it is highly unlikely that thousands of genes have the information related to the cancer and using all the genes results in too big dimensionality, it is necessary to select some genes highly related to particular classes for classification, which are called informative genes. This process is referred to as gene selection. It is also called feature selection in machine learning.

In this paper, we attempt to move away from strictly statistical and data mining methods that seem to dominate the current state of art in this area, and try to explore how knowledge extraction from gene expressions can be successful carried out by an evolutionary strategy. Our approach to micro-array data classification can be viewed as a two-stage procedure.

First, we try to break the barrier of feature selection. We adopt filters, which are commonly used in the field of data mining and pattern recognition, for ranking features in terms of the mutual information between the features and the class label. Then, we combine ranking results in small subsets of predictive genes substantially reducing the number of features. These subsets are input to the second stage that adopts an evolutionary approach to further select features and precisely classify cancer. Specifically, feature selection is formulated as an optimization problem for which it is to find the genes that guarantee maximum accuracy in a given classification task. A Genetic Algorithm (GA) is used to explore the feature space defined in the first stage and look for solutions (i.e. gene subsets) with better classification performance. The quality (fitness) of each solution is evaluated by an SVM classifier (but any classification algorithm could be incorporated in our approach). As a test-bed for evaluating the proposed methodology we choose the Leukemia dataset, publicly available at [5]. We demonstrate, with results, that our approach is highly effective in selecting small subsets of predictive genes while it allows saving time and alleviating computational load.

Although a number of recent works address the problem of gene selection using a GA in conjunction with some classifier [2][6][7], our approach is innovative: instead of exploring the whole dataset, the GA looks for solutions in the small gene spaces that we defined in the first stage. This way, we can analyze the gene information very rapidly.

The paper is organized as follows. In Section 2, we discuss some related works. Section 3 describes the proposed approach, while experiments are reported in Section 4. In Section 5, we discuss the results and present some concluding remarks.

2 Related Work

Recent studies [1][2][3][4] address the problem of gene selection using a standard GA which evolves populations of possible solutions, the quality of each solution being evaluated by an SVM classifier. GAs have been employed in conjunction with different classifiers, such as k-Nearest Neighbor in [6] and Neural Networks in [7]. Moreover, evolutionary approaches enable the selection problem to be treated as a multi-objective optimization problem, minimizing simultaneously the number of genes and the number of misclassified examples [3][4][8].

Recent literature [4] shows that evolutionary approaches may benefit of a preliminary feature selection step when applied to high dimensional problems such as micro-array data analysis. A number of hybrid approaches have been proposed [2][3][4] that apply some pre-filtering technique to define suitable gene spaces to be further refined by an evolutionary algorithm. Yang et al [9] and Forman [10] conducted comparative studies on filter methods, and they found that Information

Gain and Chi-square are among the most effective methods of feature selection for classification.

3 The Proposed Approach

We define extracting knowledge from micro-array data the process that selects discriminative genes related to classification, trains a classifier and then classifies new data using the learned classifier. As previously mentioned, our knowledge extraction process has two stages that we describe in the following.

First Stage: the Search Space Definition. It is common to use some techniques to generate a small list of important features in order to learn classifiers that use only a small subset of the original dataset. A popular method, which is named filter, is to define the feature selection as a preprocessing step that is independent from classification. In more detail, a filter method computes a score (ranking) for each feature and then selects features according to the scores. However, each filter method is able to point out only a peculiar character of the information contained in the data at hand, resulting in a feature list that may be not nearly informative. For overcoming this problem, we propose constructing M lists of features, that we call *Feature Pools* (FPs), via the combination of M different filter methods. The final objective is to have different lists (i.e. FPs) of candidate genes, to be further refined by a genetic algorithm. Inspired by our previous work [4], the construction of FPs is carried out according the following steps:

1. M filter processes are carried out separately on the original dataset. This results in M lists of ranked features each containing all the features in descending order of relevance.
2. According to a fixed threshold T, we cut the previous lists and consider only the T top-ranked features from each list.
3. To absorb useful knowledge from the above lists, we fuse their information by considering the features they share. Specifically, we build M nested feature pools FP_1 Í FP_2 … Í FP_M, where FP_1 contains the features shared by all the M lists, FP_2 the features shared by at least M-1 of the M lists, …, FP_{M-1} the features shared by at least 2 of the M lists. Finally, FP_M contains all the features belonging to the M lists.

Second Stage: GA-based Gene Selection and Classification. The second stage considers two aspects: how the mechanism of feature selection works and how the classifier accuracy is affected by the mechanism. The evolutionary approach we propose here is intended for two distinct purposes:

1. Effective use of a GA that provides rapid local-search capabilities in the search space defined at the first stage.
2. Effective use of SVM that provides high-quality classifiers.

The key idea is to obtain the benefits from both GA and SVM: the former is used to explore the input search space and to discover promising subsets of features (i.e. genes) while the latter evaluates them by classification.

With the GA, individuals are small sets of important features, typically represented by a string or a binary array. A population of individuals is randomly initialized at the start of the GA. This population undergoes mutation (a bit in an instance is flipped) and crossover (two instances create two new instances by splitting both parent bit-strings) operators, creating a collection of new individuals. This evolution process is repeated until a pre-defined number of generations G is reached, resulting in a "best" individual that represents the most informative feature subset.

Our evolutionary strategy considers to separately apply this process on each FP. Accordingly, each individual is a binary vector (whose maximum size is M*T), where the values "1" and "0" respectively mean that the feature is included or not in the individual. Genetic operations are carried out by roulette wheel selection, single point crossover, and bit-flip mutation. Taking into consideration previous research [4], we assume as fitness function the accuracy of the SVM classifier learnt on the individual. With regard to SVM classifier, error estimation is made by using leave-one-out cross validation (LOOCV). This choice is justified by the will to pay great attention to the classifier accuracy, even if the required computational load is greater than using other evaluation methods.

4 Experimental Analysis

We report on the successful application of the proposed approach to Leukemia dataset [5], which contains 7129 gene expression levels from 72 samples, among which 25 samples are collected from acute myeloid leukemia (AML) patients and 47 samples are from acute lymphoblastic leukemia (ALL) patients. The overall analysis has been implemented using the Weka library [11].

First Stage. We set the number of filter methods M = 4, the threshold T = 20 and choose the following ranking methods: Information Gain, Chi-squared, Symmetrical Uncert, and One Rule. The feature selection process (section 3) results in the following feature pools: FP_1 (composed of 12 features), FP_2 (18 features), FP_3 (21 features), and finally FP_4 (29 features).

Second Stage. Each FP_i (i = 1, 2, …, 4) is used as input to the GA. In order to find an efficient setting of the algorithm in the considered domain, we operated a performance analysis by considering different values of the following parameters: (i) number of generations, (ii) population size, (iii) probability of crossover, and (iv) probability of mutation. Specifically, the analysis was carried on according to two distinct phases:

A. We test the GA/SVM behavior as parameters (i) and (ii) change, while parameters (iii) and (iv) assume values consistent with the literature;
B. We test the GA/SVM behavior as parameters (iii) and (iv) change, while parameters (i) and (ii) assume the best results found in the previous phase A.

This pairing is justified because, in the literature, wide discordances can be found between the values chosen for parameters (i) and (ii). As well, parameters (iii) and (iv) typically assume values in a range that we consider in our analysis. Since the evolutionary algorithm performs a stochastic search, in both phases we consider the average results over a number P = 10 of trials.

Phase A. We tested the performance of GA/SVM as the parameters (i) number of generations and (ii) population size change, by considering each combination of the values of these two parameters. Specifically, values considered for parameters are as follows: (i) number of generations: 10, 20, 30, 50, and 100; (ii) population size: 10, 20, 30, and 50; (iii) probability of crossover = 1; (iv) probability of mutation = 0.01. Tables (1-4) show results on each FPi, in terms of classification accuracy and feature subset size (in brackets). Derived from Tables (1-4), Figures (1-4) show the interpolation surface expressing the global trend of the average accuracy and average subset size (y-axis) vs. the number of generation (x-axis) and the population size (z-axis). Different colours indicate different ranges of values (shown in the enclosed legends) in order to better evaluate changes respectively occurring in the average accuracy and in the average subset size. With regard to computational load, we don't show the relative results explicitly, but we consider them in the subsequent discussion.

Table 1. Performance of GA/SVM on feature pool FP_1

FP1					
Avg. accuracy (avg size)		Population size			
		10	20	30	50
Generations	10	0.9639 (6)	0.9778 (5.4)	0.9806 (4.8)	0.9861 (4.6)
	20	0.9694 (6.2)	0.9778 (5.6)	0.9819 (4.5)	0.9861 (4)
	30	0.9694 (6.2)	0.9778 (5.4)	0.9819 (4.3)	0.9861 (4)
	50	0.9681 (5.1)	0.9806 (4.7)	0.9819 (4.3)	0.9861 (4)
	100	0.9694 (6.2)	0.9778 (5.4)	0.9833 (4)	0.9861 (4)

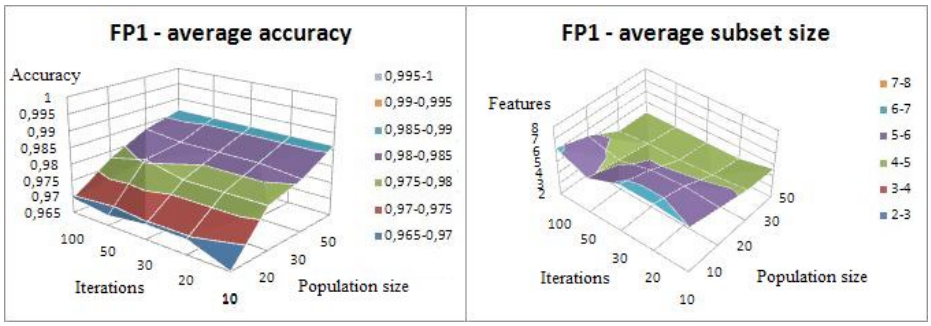

Fig. 1. Performance of GA/SVM on feature pool FP_1

As regards the number of generations, we notice significant results for 30-50 iterations. Going on up to 100 iterations causes some improvement only in one run out of 10, while computational load increases in accordance to the number of generations. Considering the population size, we observe that best results are obtained when the value assumed by this parameter is 30 or 50. Values less than 30 make the algorithm to converge to a local optimum, while values greater than 50 were not considered for two reasons: the average accuracy and average size of the subset seem to stabilize when the value assumed by this parameter is 30 as well, exceeding 30, computational load increases considerably.

Table 2. Performance of GA/SVM on feature pool FP_2

FP2					
Avg. accuracy (avg size)		Population size			
		10	20	30	50
Generations	10	0.9778 (8)	0.9889 (7.6)	0.9875 (6.3)	0.9861 (5.8)
	20	0.9778 (8)	0.9889 (6.6)	0.9889 (5.5)	0.9889 (5.6)
	30	0.9778 (8)	0.9889 (6.2)	0.9903 (5.1)	0.9889 (5.6)
	50	0.9778 (8)	0.9875 (5.3)	0.9917 (4.7)	0.9903 (5)
	100	0.9778 (8)	0.9917 (5)	0.9917 (4.6)	0.9917 (4.8)

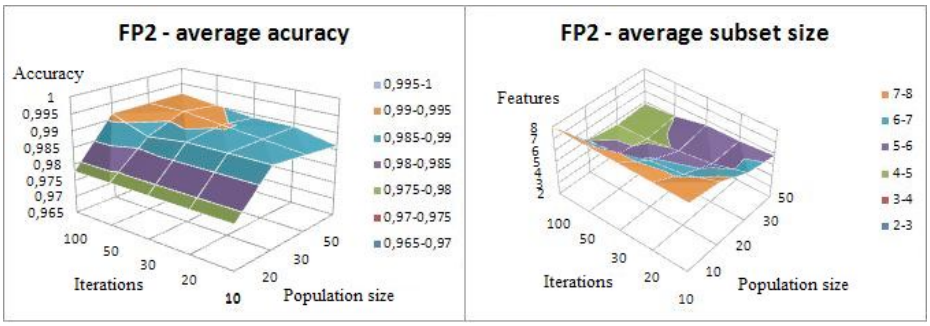

Fig. 2. Performance of GA/SVM on feature pool FP_2

Table 3. Performance of GA/SVM on feature pool FP_3

FP3					
Avg. accuracy (avg size)		Population size			
		10	20	30	50
Generations	10	0.9833 (9.6)	0.9889 (7.8)	0.9986 (6.3)	0.9972 (5.8)
	20	0.9833 (9.2)	0.9889 (7)	0.9986 (5.5)	1 (5.8)
	30	0.9833 (9.2)	0.9889 (6.8)	0.9986 (5)	1 (5.4)
	50	0.9806 (8.5)	0.9875 (5.8)	0.9986 (4.3)	0.9972 (4.3)
	100	0.9833 (9.2)	0.9889 (6.6)	1 (4.3)	1 (3.6)

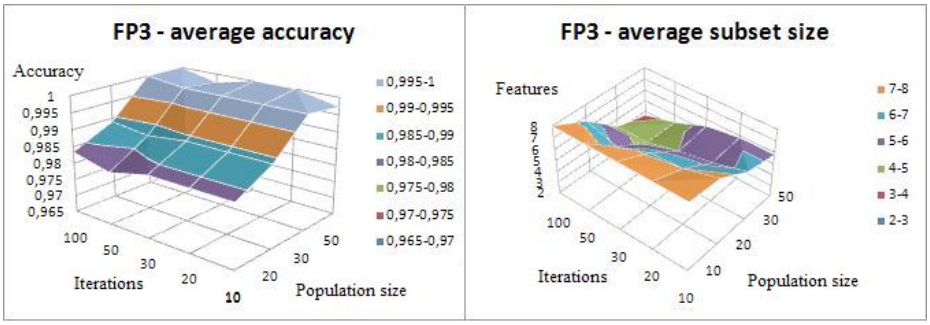

Fig. 3. Performance of GA/SVM on feature pool FP_3

Table 4. Performance of GA/SVM on feature pool FP_4

FP4					
Avg. accuracy (avg size)		Population size			
		10	20	30	50
Generations	10	0.9861 (9.4)	0.9861 (7.4)	0.9889 (9.2)	1 (9.2)
	20	0.9889 (9.2)	0.9861 (7.2)	0.9889 (9.2)	1 (7.6)
	30	0.9889 (9.2)	0.9861 (7)	0.9889 (9.2)	1 (6.8)
	50	0.9875 (10.1)	0.9889 (9.2)	0.9903 (5.9)	0.9958 (6.1)
	100	0.9889 (9.2)	0.9889 (6.6)	0.9903 (5.7)	1 (4.8)

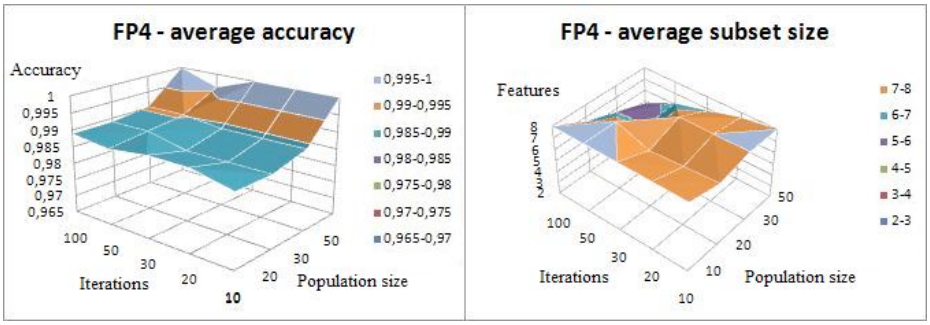

Fig. 4. Performance of GA/SVM on feature pool FP_4

Phase B. We tested the performance of GA/SVM as the parameters (iii) probability of crossover and (iv) probability of mutation change, by considering each combination of the values of these two parameters. Values considered for parameters (iii) and (iv) are respectively: (iii) 0.6, 0.8, 1 and (iv) 0.005, 0.01, 0.02, 0.03. According to the results obtained in the phase A, we set (i) number of generations = 50 and (ii) population size = 30.

Tables (5-8) show results on each FP, in terms of classification accuracy and feature subset size (in brackets). Again, figures (5-8) show the same results using charts (average accuracy and average subset size on the y-axis, probability of mutation on the x-axis, and probability of crossover on the z-axis).

Considering the parameter probability of crossover, we did not achieve significant variations as values change; however we find the best results in correspondence to value 1. Finally, as regards the parameter probability of mutation, we notice that increasing values correspond to better results on average. In particular, the value 0.02 gives good results considering both accuracy and dimensionality and, in addition, exceeding 0.02 computational load increases considerably.

Table 5. Performance of GA/SVM on feature pool FP_1

FP1	
Avg. accuracy	Probability of crossover

(avg size)		0.6	0.8	1
mutationProb. of	0.005	0.9778 (4.2)	0.9833 (5.4)	0.9833 (4.2)
	0.01	0.9806 (4)	0.9861 (4.6)	0.9819 (4.3)
	0.02	0.9861 (4)	0.9861 (4)	0.9861 (4)
	0.03	0.9861 (4)	0.9861 (4)	0.9861 (4)

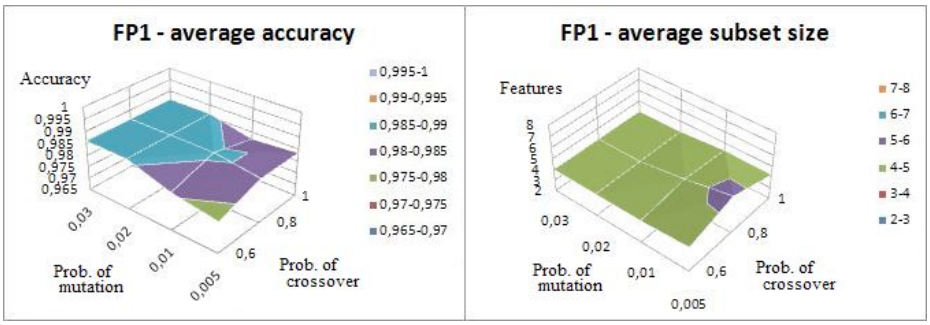

Fig. 5. Performance of GA/SVM on feature pool FP_1

Table 6. Performance of GA/SVM on feature pool FP_2

FP2				
Avg. accuracy (avg size)		Probability of crossover		
		0.6	0.8	1
mutationProb. of	0.005	0.9861 (6.4)	0.9889 (7.8)	0.9889 (5)
	0.01	0.9917 (6.4)	0.9889 (5.8)	0.9917 (4.7)
	0.02	0.9917 (5)	0.9944 (5)	0.9972 (4.6)
	0.03	0.9944 (4.2)	0.9972 (5.8)	0.9944 (4.4)

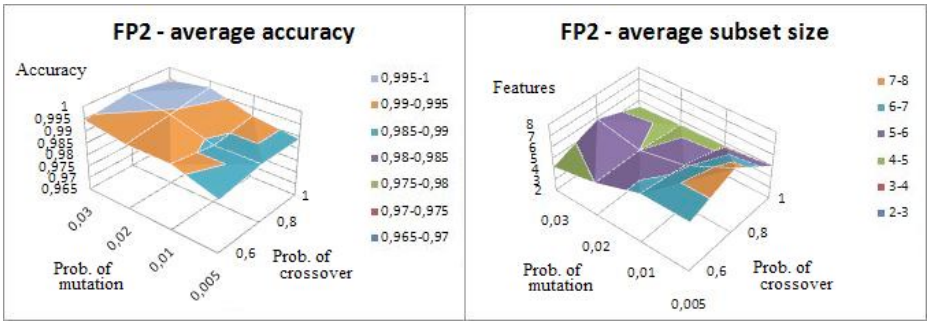

Fig. 6. Performance of GA/SVM on feature pool FP_2

Table 7. Performance of GA/SVM on feature pool FP_3

FP3				
Avg. accuracy (avg size)		Probability of crossover		
		0.6	0.8	1
mutationProb. of	0.005	0.9944 (5.8)	0.9917 (5.4)	0.9917 (5.2)
	0.01	0.9972 (4.4)	0.9972 (4.6)	0.9986 (4.3)
	0.02	0.9972 (4.4)	0.9972 (5.4)	0.9944 (5.4)
	0.03	0.9972 (5.2)	0.9944 (4.4)	0.9972 (4.2)

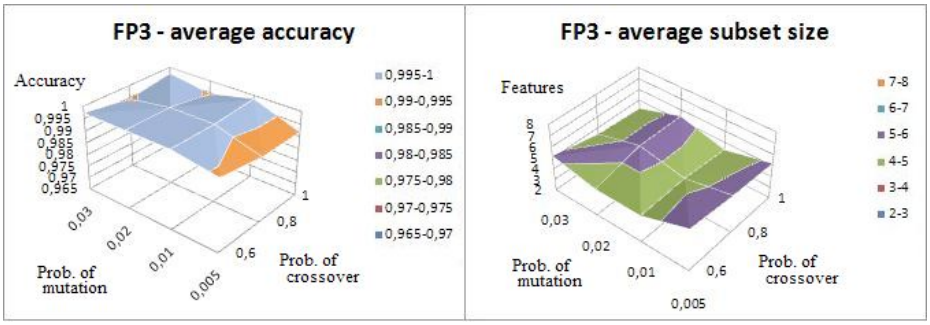

Fig. 7. Performance of GA/SVM on feature pool FP_3

Table 8. Performance of GA/SVM on feature pool FP_4

FP4				
Avg. accuracy (avg size)		Probability of crossover		
		0.6	0.8	1
mutationProb. of	0.005	0.9917 (8.2)	0.9944 (6.8)	0.9944 (7.6)
	0.01	0.9944 (6.6)	0.9972 (6.2)	0.9903 (5.9)
	0.02	0.9972 (6.4)	0.9944 (5.8)	0.9972 (5.6)
	0.03	0.9972 (6.2)	1 (5.6)	0.9972 (6.2)

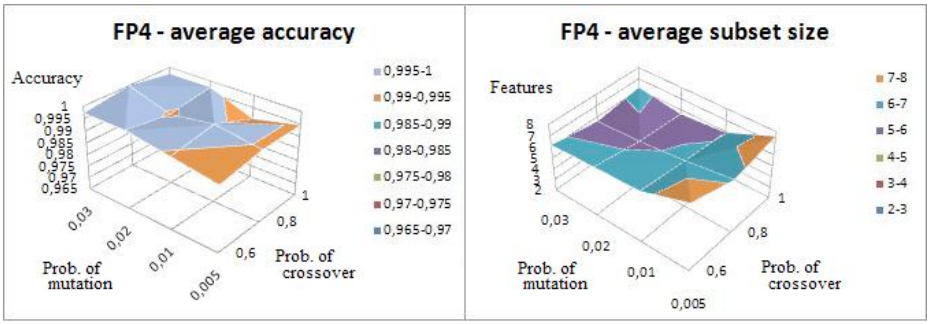

Fig. 8. Performance of GA/SVM on feature pool FP_4

5 Discussion and Concluding Remarks

First, it is important to notice that the parameter values we consider as optimal, especially regarding the number of generations and the population size, are smaller than the values commonly used in other methods discussed in the literature, with consequent time saving and computational load saving. Because our experiments result in excellent fitness, we can assert that the evolutionary approach we propose allows us to use a GA in a both effective and efficient manner: small subsets of predictive genes are selected with a reduced computational load. This validates the process of building FPs that reduce the dimensionality of the initial problem by discarding redundant or irrelevant features.

With regard to FPs construction, a basic question is how defining the most effective search space for the GA. Combining valuable results from different ranking methods allows us to achieve good results. However, features common to all ranking methods (i.e. the features belonging to FP_1) define a search space that is too small: the performance of GA/SVM achieves 98,6% of accuracy and does not increase when the search is refined by an additional number of generations. When this search space is enlarged by adding genes selected by three, two and just one method, our approach shows an excellent performance, not only at providing a very good average accuracy, but also with respect to the number of selected features and the computational cost. In particular, the pool FP_3 seems to define the most effective search space for the GA.

A further question we want to point out is that, as presented in Table 1-8, we consider the average results obtained in the analysis. But, during our study, we noticed that the difference between average values and best values was very scanty, and it means that results are not outcomes of a particularly lucky run, but they derive from a valid and effective behavior of the evolutionary method.

Table 9 summarizes the results we obtained using the proposed approach with the results of three state-of-art methods that use a GA as feature selection technique. To evaluate the results we use the conventional criteria, that is the classification accuracy in terms of the rate of correct classification (first number) and the size of the subset i.e. the number of selected genes (the number in parenthesis). For our approach, we choose to present the data obtained using FP_3. The maximum classification rate we

obtain is 1 using 3 genes while the corresponding average classification rate is 1 and the corresponding average dimension is 3.6 (see Table 3 for details). The same performance is achieved by [1] [2] [8], even if the number of genes selected by [1] [2] [8] is greater than the one obtained by our method.

As feature work, we plan to extend our study by considering different ranking methods, as well as different values of the threshold used to cut-off each ranked list, in order to gain more insight on the evolutionary search space definition. Moreover, the proposed approach will be validated on different micro-array datasets.

Table 9. The proposed method versus three state of the art methods.

Studies	Classification rate	Subset size
The proposed method	1	(3)
[1]	1	(6)
[8]	1	(4)
[2]	1	(25)

References

1. Peng S., et al., Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. FEBS Letter, 555(2):358–362, 2003.
2. Huerta E.B., Duval B., Hao J.K., A Hybrid GA/SVM Approach for Gene Selection and Classification of Microarray Data, EvoWorkshops 2006, LNCS 3907, pp. 34–44, 2006.
3. Tan F., Fu X., Zhang T., Bourgeois A.G., Improving Feature Subset Selection Using a Genetic Algorithm for Microarray Gene Expression Data, 2006 IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, July 16-21, 2006.
4. N. Dessì, B. Pes. "An Evolutionary Method for Combining Different Feature Selection Criteria in Microarray Data Classification". Journal of Artificial Evolution and Applications Volume 2009, Article ID 803973, 10 pages doi:10.1155/2009/803973.
5. http://datam.i2r.a-star.edu.sg/datasets/krbd/
6. Li L., Weinberg C. R., Darden T.A., Pedersen L.G., Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics, 17(12):1131–1142, 2001.
7. Bevilacqua V., et al., Genetic Algorithms and Artificial Neural Networks in Microarray Data Analysis: a Distributed Approach, Engineering Letters, 13:3, EL_13_3_14, 2006.
8. Reddy A. R., Deb K., Classification of two-class cancer data reliably using evolutionary algorithms. Technical Report. KanGAL, 2003.
9. Y. Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. ICML 1997.
10. G. Forman**.** An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research, 2003.
11. Witten, I. H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Second edition, Elsevier, 2005.