Hybrid DE-SVM Approach for Feature Selection: Application to Gene Expression Datasets

José García-Nieto, Enrique Alba
Dept. de Lenguajes y Ciencias de la Computación,
University of Málaga, ETSI Informática,
Campus de Teatinos, Málaga - 29071, Spain.
Email: jnieto,eat@lcc.uma.es

Javier Apolloni
LIDIC - Departamento de Informática,
University of San Luis
Ejército de los Andes 950, 5700, Argentina
Email: javierma@unsl.edu.ar

***Abstract*—The efficient selection of predictive and accurate gene subsets for cell-type classification is nowadays a crucial problem in Microarray data analysis. The application and combination of dedicated computational intelligence methods holds a great promise for tackling the feature selection and classification. In this work we present a Differential Evolution (DE) approach for the efficient automated gene subset selection. In this model, the selected subsets are evaluated by means of their classification rate using a Support Vector Machines (SVM) classifier. The proposed approach is tested on DLBCL Lymphoma and Colon Tumor gene expression datasets. Experiments lying in effectiveness and biological analyses of the results, in addition to comparisons with related methods in the literature, indicate that our DE-SVM model is highly reliable and competitive.**

I. Introduction

DNA Microarrays (MA) [13] allow the scientists to simultaneously analyze thousands of genes, and thus giving important insights about cell's function, since changes in the physio-logy of an organism are generally associated with changes in gene ensembles of expression patterns. The vast amount of data involved in a typical Microarray experiment usually requires to perform a complex statistical analysis, with the important goal of making the classification of the dataset into correct classes. The key issue in this classification is to identify significant and representative gene subsets that may be used to predict class membership for new external samples. Furthermore, these subsets should be as small as possible in order to develop fast and low consuming processes for the future class prediction. The main difficulty in Microarray classification versus other domains is the availability of a relatively small number of samples in comparison with the number of genes in each sample (between 2,000 and more than 10,000 in MA). In addition, expression data are highly redundant and noisy, and of most genes are believed to be uninformative with respect to studied classes, as only a fraction of genes may present distinct profiles for different classes of samples.

In this context, machine learning techniques have been applied to handle with large and heterogeneous datasets, since they are capable to isolate the useful information by rejecting redundancies. Concretely, feature selection is often considered as a necessary preprocess step to analyze large datasets, as this method can reduce the dimensionality of the datasets and often conducts to better analyses [9].

Feature selection (gene selection in Biology) for gene expression analysis in cancer prediction often uses wrapper classification methods to discriminate a type of tumor [9], [11], to reduce the number of genes to investigate in case of a new patient, and also to assist in drug discovery and early diagnosis. The formal definition of the feature selection problem that we consider here is given as follows:

Definition Let $F = \{f_1, ..., f_n\}$ be a set of features; find a subset $F' \subseteq F$ that maximizes a scoring function $\Theta : \Gamma \rightarrow G$ such that $F' = argmax_{G \subset \Gamma}\{\Theta(G)\}$; where Γ is the space of all possible feature subsets of F and G a subset of Γ. ■

Optimal feature selection is a complex problem proved to be NP-hard [12]. Therefore, we need efficient automated approaches to tackle it. Metaheuristics algorithms have proved to be adequate tools for this matter, since they are capable of solving the feature selection accurately and efficiently for the large dimensions needed in Biology. For example, Evolutionary Algorithms (EAs) and, specifically, Genetic Algorithms (GAs) have been successfully used in the past to tackle the gene selection of Microarrays [1], [7], [10]. A recent EA is Differential Evolution (DE) which has been proven to be a very efficient optimizer [14], specially when dealing with continuous landscapes. DE was firstly used for finding informative gene subsets in Microarray data in [15]. In that work, the authors used a version of DE for integer representation (of genes) which applied a Feedforward Neural Network (FNN) as the classifier of the selected subsets for Colon [3] dataset. However, the reported classification rates in [15] were always lower than 80% in Colon which has room of improvement.

Our aim here lies in using a binary version of DE, embedding a feature selection mechanism with SVM as classifier, for the efficient gene selection of high dimensional Microarray datasets. The reported solutions, codifying gene subsets, will be evaluated by means of their classification accuracy. They will be validated inside of the same process (evaluation) by means of a *10-fold cross-validation* procedure, and finally tested with external test datasets. The contributions are noticeable since the hybridization of DE with SVM will be shown to improve existing algorithms in terms of accuracy. Besides, the gene ensembles are interpreted in the light of results

from Biology to show their actual impact far from more test datasets. The effectiveness of our algorithm is analyzed on two well-known public datasets: DLBCL Lymphoma [6] and Colon Tumor [3] discovering newly and biologically challenging gene subsets, and identifying specific genes that our work suggests as significant ones. Comparisons with several recent state-of-the-art methods show the effectiveness of our results according to biological and computational criteria.

The remaining of this paper is organized as follows. Section II introduces the binary DE used in this work. In Section III, our DE-SVM model is described. Experimental results and comparisons are presented in Sections IV and V, including performance and biological analyses. Conclusions and further work are finally given in Section VI.

II. Differential Evolution: Background

Differential Evolution (DE) was introduced by Storn and Price [14] as an efficient EA initially designed for global optimization problems over continuous search spaces. DE uses a population (P) of N real vectors (individuals) representing solutions which evolves simultaneously along with the increasing generation number g. An individual is a vector $v_g^i = (v_g^i(1), v_g^i(2), \ldots, v_g^i(D))$ where $v_g^i(j) \in \mathbb{R}$ $(1 \leq i \leq N, 1 \leq j \leq D)$ and D is the number of variables. A *mutant individual* w_g^i is generated by the following equation:

$$w_g^i = v_g^{r1} + \mu \cdot (v_g^{r2} - v_g^{r3}) \tag{1}$$

where $r1, r2, r3 \in \{1, 2, \ldots, i-1, i+1, \ldots, N\}$ are random integers mutually different and different from the index i, the mutation constant $\mu > 0$ stands for the amplification of the difference between the individuals v_g^{r2} and v_g^{r3}, and avoid the stagnation of the search process.

In order to keep the diversity in the population, each mutated individual undergoes a crossover operation with the *target individual* v_g^i, by means of which a *trial individual* u_g^i is generated. We use here the *binomial crossover* [14] (Equation 2).

$$u_g^i(j) = \begin{cases} w_g^i(j) & \text{if } r(j) \leq Cr \text{ or } j = j_r, \\ v_g^i(j) & \text{otherwise.} \end{cases} \tag{2}$$

The binomial crossover operator randomly chooses a uniformly distributed integer value j_r in $\{1, \ldots, D\}$ and a random real number r in $(0, 1)$ also uniformly distributed for each component j $(j = 1, \ldots, D)$ of the trial individual u_g^i. Then, the crossover probability Cr and r are compared just like j and j_r. If r is less than or equal to Cr or j is equal to j_r then we select the j^{th} element of the mutant individual to be allocated in the j^{th} element of the trial individual u_g^i. Otherwise, the j^{th} element of the target individual v_g^i becomes the j^{th} element of the trial individual. Finally, a selection operator decides the acceptance of the trial individual for the next generation if and only if it yields an improvement in the value of the evaluation function (Equation 3).

$$v_{g+1}^i = \begin{cases} u_g^i & \text{if } f(u_g^i) \; is \; better \; than \; f(v_g^i), \\ v_g^i & \text{otherwise.} \end{cases} \tag{3}$$

Algorithm 1 shows the pseudocode of the canonical DE. After the initialization of individuals (line 1), DE starts the evolution process performing differential mutation (line 5) and binomial crossover (lines 6 to 10). Finally, the selection of trial/target individual is carried out (lines 12 to 16). This generation procedure is repeated until reaching the stop condition.

Algorithm 1 Canonical DE

```
1: initialize(P)
2: while not stop condition do
3:   for each individual v^i of P do
4:     for each variable j of individual v^i do
5:       w_g^i(j) ← v_g^r1(j) + μ · (v_g^r2(j) − v_g^r3(j))
6:       if r(j) ≤ Cr || j = jr then
7:         u_g^i(j) ← w_g^i(j)
8:       else
9:         u_g^i(j) ← v_g^i(j)
10:      end if
11:    end for
12:    if f(u_g^i) is better than f(v_g^i) then
13:      v_{g+1}^i ← u_g^i
14:    else
15:      v_{g+1}^i ← v_g^i
16:    end if
17:  end for
18: end while
19: Output: Best solution found
```

A. Binary Differential Evolution

As previously mentioned, DE was designed for global optimization problems over continuous spaces. In order to make DE capable of dealing with binary encoding problems (which is the case in this work) we have performed some modifications to the canonical DE following the specifications defined in Gong et al. [8]. These modifications are basically made on the operations of multiplication, addition, and subtraction in the mutation operator.

First of all, the individuals are initialized by randomly (uniformly) assigning values from {0,1} to the variables. The individuals will suffer transformations through the evolutionary process but the binary values must be kept in all the changes. Thus, the result of the subtraction $(v_g^{r2} - v_g^{r3})$ is obtained by computing the Hamming distance (d) between these two vectors (individuals). We have considered all decision variables being in a single dimension as explained in [8]. As an example, in Table I, the distance d between v_g^{r2} and v_g^{r3} for the single dimension becomes 3. Assuming a scaling factor μ of 0.6 and according to the Equation 4, the scaled difference d' is 1.8. Therefore, we need to interpret the real-valued scaled distance to a valid scaled distance between binary vectors.

$$d' = \mu \cdot d \tag{4}$$

There are two options to round d' to its nearest integer: $\lfloor d' \rfloor$ and $\lceil d' \rceil$. We can obtain the probability of rounding d' to each option as follows:

$$P(d' \rightarrow \lceil d' \rceil) = d' - \lfloor d' \rfloor, \quad (5)$$

$$P(d' \rightarrow \lfloor d' \rfloor) = 1 - (d' - \lfloor d' \rfloor). \quad (6)$$

Thus, the mutant vector can be generated as shown in Equation 7.

$$D(w_g^i, v_g^{r1}) = \begin{cases} \lceil d' \rceil & \text{if } rand < d' - \lfloor d' \rfloor \\ \lfloor d' \rfloor & \text{otherwise.} \end{cases} \quad (7)$$

where $rand$ $(0 \leq rand \leq 1)$ is a uniform randomly distributed value.

Continuing with the example above, a scaled distance d' of 1.8 leads to a probability of 0.8 to be rounded to 2 and a probability of 0.2 to be rounded to 1. Assuming that $rand$ $(UN(0,1))$ is smaller than 0.8, the distance applied to the base individual v_g^{r1} to produce the mutant individual should be 2. Applying any 2 *mutually exclusive* (according to strict k-change [8]) changes in the base individual will yield a possible mutant individual. As shown in the sp row in Table I, the positions marked with $\diamond$ are randomly uniformly selected to be changed in order to produce the mutant individual.

Considering all decisions variables in the solutions as a single dimension, we have derived a mutation operator that applies the rounded scaled-distance to the base individual by flipping the values for some randomly selected decision variables. This operator is called *any-change DE mutation* (M_{any}), where changes can be made to any decision variables.

TABLE I
ANY-CHANGE DE BINARY MUTATION: OPERATION EXAMPLE

	$v(1)$	$v(2)$	$v(3)$	$v(4)$	$v(5)$
v_g^{r2}	1	1	0	1	1
v_g^{r3}	0	1	1	1	0
d	1	0	1	0	1
v_g^{r1}	0	1	0	1	1
sp		$\diamond$		$\diamond$	
M_{any}	0	0	0	0	1

The crossover operator implemented in this work is similar as explained in Equation 2 but we extended the operator to reinforce the search process and increase the elimination of features. Therefore, after applying the implementation of equation 2 and with a probability τ $(0 \leq \tau \leq 1)$, the trial individual can still undergo another transformation by means of the *extended-binomial DE crossover* shown in Equation 8.

$$u_g^i(j) = \begin{cases} 0 & \text{if } u_g^i(j) = 1 \wedge UN(0,1) \leq \gamma_1, \\ 1 & \text{if } u_g^i(j) = 0 \wedge UN(0,1) \leq \gamma_2, \\ 1 - u_g^i(j) & \text{otherwise.} \end{cases} \quad (8)$$

where $1 \leq j \leq D$ and γ_1 and γ_2 $(0 \leq \gamma_1, \gamma_2 \leq 1)$ are parameters associated to this operator and used to control the level of modification of the components of the trial individual, being $\gamma_1 = 0.9$ the parameter associated to the elimination of features and $\gamma_2 = 0.3$ the potential inclusion of a feature.

III. HYBRID DE-SVM APPROACH FOR FEATURE SELECTION

Following the basic scheme of solution encoding used in feature selection, our binary DE provides a binary encoded solution (vector) where each bit represents a gene in the dataset. If a bit is 1, it means that this gene is kept in the reduced subset, while 0 indicates that the gene is not included. Therefore, the individual length is equal to the number of genes in the initial Microarray dataset.

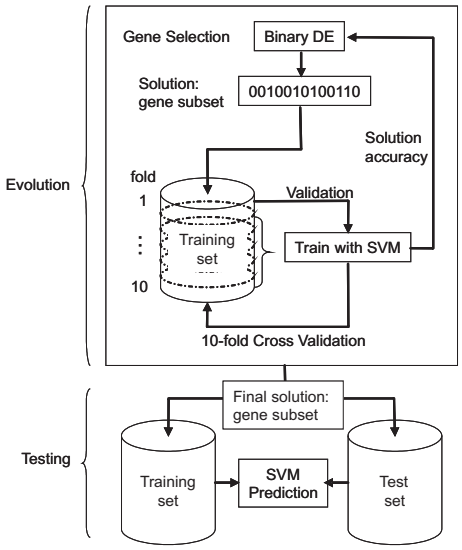

Fig. 1. General scheme of our DE-SVM for gene selection and classification of DNA Microarrays

As illustrated in Fig. 1, where a general model of our DE-SVM is provided, the individuals representing gene subsets, are evaluated by means of a SVM classifier and validated using *10-fold cross-validation* as follows: each gene subset (codified by an individual) is divided into ten subsets of samples, nine of them constituting the training set and the remaining one used as the validation set. The SVM is trained using the training set and then, the accuracy obtained (number of correct sample classifications by the SVM once trained) is evaluated on the validation set [5]. This evaluation in repeated ten times, each one alternating the used validation set. This method reinforces the validation process, so that the final accuracy value is the resulted average of the ten validation folds. Such a strong validation is necessary when the number of samples is low regarding the number of features, which is the case in this work. Once the evolution process has finished, the resulted subset solution is evaluated on the external test set obtaining the accuracy.

A. Fitness Function

Each individual, representing a given gene subset, is evaluated by means of its classification accuracy and the reduction percentage $(reduct)$ it obtained regarding the whole dataset. The fitness function is then as follows:

$$f(v) = \alpha \cdot accuracy + (1 - \alpha) \cdot reduct \quad (9)$$

being

$$reduct = 100 \cdot \frac{\#total\_genes - \#genes\_in\_subset}{\#total\_genes} \quad (10)$$

where $\alpha \in \mathbb{R}$ $(0 \leq \alpha \leq 1)$ is a value to focus the importance in one of the terms of the sum. In our case $\alpha = 0.2$ since we look for reducing as much as possible the dimensionality of the subset. The reduction percentage is calculated by the Equation 10 where $\#total\_genes$ is the total number of genes in the original dataset and $\#genes\_in\_subset$ is the number of genes in the current subset. This way, our objective is to maximize both, accuracy and reduction percentages.

IV. Experiments

We have implemented the proposed binary DE algorithm for gene selection in C++ following the *skeleton* architecture of the MALLBA library [2]. For the SVM classifier we have used a set of object classes provided by the LIBSVM [4] library for training, validation, and testing. These classes were coupled with those of the DE for the evaluation phase.

A. Datasets

The used instances are classified into two well-known datasets got from real-word Microarray experiments. They were taken from the public repository of Kent Ridge Biomedical Dataset in URL http://datam.i2r.a-star.edu.sg/datasets/krbd/index.html. In particular:

- The Colon tumor dataset consists of 62 Microarray experiments collected from colon-cancer patients with 2,000 gene expression levels. Among them, 40 tumor biopsies are from *tumors* and 22 (*normal*) biopsies are from healthy parts of the colons of the same patients.
- *Types of Diffuse Large B-cell Lymphoma* dataset consists of 47 tissue samples, 24 of them are from *germinal centre B-like group* while the rest 23 are *activated B-like group*. Each sample is described by 4,026 genes.

In both Lymphoma and Colon datasets, only training sets are available in the original repositories, and for this reason, new test and training sets were generated for this datasets by randomly (uniformly) extracting samples from the original one, resulting: 70% for training set and 30% for testing set.

Expression levels of training and test sets were normalized separately in order to scale the intensities, thus enabling the comparison between the different datasets previously introduced. Each attribute was scaled to $[-1, 1]$ by using:

$$a'_j(x_i) = 2\frac{a_j(x_i) - min_j}{max_j - min_i} - 1, \quad (11)$$

where max_j and min_j correspond to the maximum and minimum gene expression values for attribute a_j over all samples. Reductions of genes by removing them according to thresholds were not made previously, and so we make the task of correct classification harder by leaving even clearly non-functional genes for the algorithm to remove. Uninformative genes to the classifier could nevertheless be informative ones to the metaheuristic algorithm, since bad solutions are quite important for avoiding guiding the search towards low quality regions of the search space.

B. Parameter Setting

All experiments were carried out using a cluster of PCs with Linux O.S. (Suse 9.0 with kernel 2.4.19) and a Pentium IV 2.8GHz processor, with 1.5GB of RAM. The DE-SVM algorithm was independently executed 30 times on the previously described Microarray datasets in order to have statistically meaningful conclusions. Each one of these executions DE-SVM performed 4,000 iterations.

In the parameter setup, an optimal configuration of the SVM classifier is crucial, since it influences the training effectiveness. Therefore, the main Kernel (Linear) parameter (the C coefficient [5]) was systematically optimized in a preprocessing phase using grid-search with cross-validation. Basically, it consists of identifying the best option of the C coefficient in a rank of bounded values (for example, $C = 2^{-5}, 2^{-5}, \cdots 2^{-15}$). The resulting parameters are shown in Table II.

TABLE II
SVM Parameter Settings

Datasets	C	Ac. (%)
Lymphoma	0.03125	85.71
Colon	0.03125	87.50

These values were set using SVM independently of the DE, in order to obtain an accuracy as higher as possible (Ac.).

The parameters of our binary DE were set using the previously tuned SVM classifier. We have used a population of 50 individuals with which we have carried out a systematic parametrization of the scaling factor μ and the crossover rate Cr, since the success of the performance of DE is related to a complex interaction of these two parameters. In this case we have executed our DE by combining the two parameters in order to identify the best options. Each combination was made using μ and Cr values in the rank of $(1...9) \cdot \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ resulting in 1,296 independent runs. The best parametrization found was $\mu = 0.05$, and $Cr = 0.08$, obtaining validation accuracies close to 100% in both Lymphoma and Colon datasets.

V. Results

In this section we analyze the results obtained by DE-SVM in terms of performance and biological relevance. Comparisons with other works in the related literature are also made.

A. Performance Analysis

Table III shows the results obtained by DE-SVM with Colon and Lymphoma (Lymph.) datasets. The initial accuracy percentage obtained by SVM (before reduction) with the complete training and testing sets is showed in column 2 (Ac_{Init}). This way we can analyze the effects of the reductions performed in the final subsets obtained by DE. In columns 3 and 4 are located the mean percentages of accuracies and

standard deviations obtained by the resulted subsets (after reduction) with 10-fold cross-validation (Ac_{CV}) and with testing validation (Ac_{Test}), respectively. The last column contains the mean and standard deviation of the reduction percentage (Rd.) obtained by our proposal regarding the complete datasets.

TABLE III
RESULTS OBTAINED BY OUR DE-SVM FOR GENE SELECTION

Dataset	SVM	DE-SVM		
	Ac_{Init} (%)	Ac_{CV} (%)	Ac_{Test} (%)	Rd. (%)
Lymph.	85.71	98.78±1.70	**87.14±5.10**	99.75±0.06
Colon	87.50	95.79±0.79	81.25±0.00	99.27±0.09

As we can observe in this table, our DE-SVM obtains reduction percentages close to 99% of the complete datasets for both Lymphoma and Colon instances, while keeping high classification accuracies. We can notice that the accuracy percentage obtained in the cross-validation process (Ac_{CV}) is always higher than 95%, and near to 99% in the case of Lymphoma. Therefore, we can now classify these Microarray datasets by using few genes (less than 20 in colon and 40 in Lymphoma) with effective precisions.

When comparing the classification accuracy reached in the final testing process (Ac_{Test}) with the one obtained before reduction (Ac_{Init}), we can observe two different results depending on the Microarray we are tackling with. In Colon, our proposal obtains a classification accuracy of 81.25% which is relatively lower than the one obtained by the SVM (87.50%) without reduction of the dataset. This is an expected result since in the former we are working with a extremely low number of genes (1% of the complete dataset). Nevertheless, in Lymphoma, our DE-SVM obtains 87.14% of accuracy which is higher than 85.71% in the case of the complete dataset. The true improvement of our proposal is then observed in Lymphoma where we can classify more precisely even with a really short number of genes (lower than 10). Therefore, this result leads us to believe that our DE-SVM selects the most representative genes ignoring the uninformative ones.

B. Comparison with other approaches

In this section we first compare our DE-SVM with a baseline method for gene selection in order to have some insights into the power of the metaheuristic method (DE) we have used. This method runs a *K*-Means procedure for clustering, in which we have selected the same number of genes of the final subsets obtained by DE-SVM as representative centroids (*K*=number of genes). Then, each resulting subset (gene centroids) is used to train the SVM classifier (configured as explained in Section IV-B), and cross-validated with an external test set.

For this purpose, we used the *K*-Means procedure available in Weka tools for data mining [17]. Table IV shows the results obtained by *K*-Means-SVM and DE-SVM in terms of the accuracy percentage. We have used the number of genes obtained in the best results of DE-SVM for being initialized as centroids in the *K*-Means procedure (column 2).

We can observe in this table that our proposal clearly outperforms the *K*-Means procedure in the two datasets. Specifically, the difference regarding the accuracy percentage in Lymphoma, 50.00% of *K*-Means-SVM in contrast with 62.24% of DE-SVM, gives us some insights into the power of our proposal. Nevertheless, we would expect these differences in results since *K*-Means is a naive method without any information about the problem in its procedure.

TABLE IV
COMPARISON WITH BASE-LINE METHOD: *K*-MEANS CLUSTERING

Dataset	*Number of genes*	*K-Means-SVM Ac (%)*	*DE-SVM Ac (%)*
Lymph.	6	50.00	**92.24**
Colon	12	68.75	**81.25**

In this sense, we compare now our DE-SVM with the other DE approach (found in the literature) for gene selection [15]. In this work, the authors used a special version of DE for integer encoding in which the solutions represent subsets in terms of the positions of genes in the complete datasets. It uses a Feedforward Neural Network (FNNs) as classifier which is trained with the selected subsets for Colon dataset.

If we focus on Colon, we can observe in [15] that DE-FNN never obtains mean values of classification accuracies higher than 80%. Concretely, in subsets with reductions percentages lower than 99% (e. g. 20 genes) the mean classification accuracy is always lower than 70% which is noticeably worse than 81.25% obtained by DE-SVM. We can suggest the use of our new version of DE for future studies in gene selection since it outperforms the previous work in the state-of-the-art.

A further comparison lies in analyzing the results of a different metaheuristic algorithm which uses a similar model of feature selection and classification. In [1], a binary version of Particle Swarm Optimization, called Geometric PSO (GPSO-SVM), was used to extract informative genes from Colon dataset and classify it be means of *10-fold cross-validation* and SVM. In this work, the classification rates were presented in terms of the cross-validation accuracy (Ac_{CV}). Compared to our DE-SVM (column 3 in Table III), the mean percentage of correct classification obtained by GPSO-SVM was also higher than 95% with a similar percentage of reduction (close to 99% in both algorithms). These results reinforce the possibility of using well adapted binary versions of metaheuristic approaches (originally designed for continuous spaces) to tackle the feature selection problem.

C. Biological Analysis

Finally, in this section we provide a biological analysis of the computed gene subsets. We will show here the real impact of our technique, capable of computing biological ensembles of genes that have been also suggested in relevant works in the domain [9], [16].

In Fig. 2, a graphic of the most selected genes in 30 independent runs of DE-SVM dealing with Lymphoma dataset is shown. This way, we can have an approximated idea of the most representative genes our approach can obtain

since they are the most frequently selected in the final subsets through all the experiments. The most relevant genes in Lymphoma: GENE3332X, GENE3261X, GENE3329X, GENE3330X, GENE3327X, and GENE3608X were selected in more than five independent runs (up to dotted line in Fig. 2) belonging to all the final subsets. Concretely, the genes GENE3261X, GENE3329X, and GENE3330X were also ranked as the most important ones in Wang et al. [16], where a set of feature filters methods were used to distinguish subtypes of DLBCL Lymphoma.

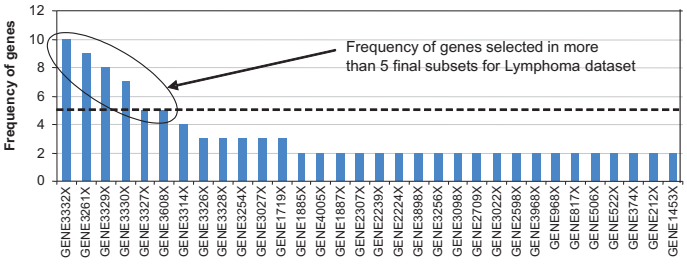

Fig. 2. Frequency of genes in 30 independent runs of DE-SVM in Lymphoma

For Colon dataset, the most frequently selected genes by DE-SVM in the experiments are shown in Fig. 3. In this case we can notice that the frequency of genes is significantly high. Specifically, the genes R88740, H64489, T71025, and M80815 were selected in more than 25 final subsets out of 30. Furthermore, the same 15 genes were selected in more than 10 final subsets (up to dotted line in Fig. 3). This robust behavior of our proposal allows it to accurately isolate the most representative genes through the experimentation. In addition, R88740 was also selected as one of the most representative genes in Guyon et al. [9], where the well-known RFE-SVM (Recursive Feature Elimination with Support Vector Machines) method was used for selecting genes in Colon dataset.

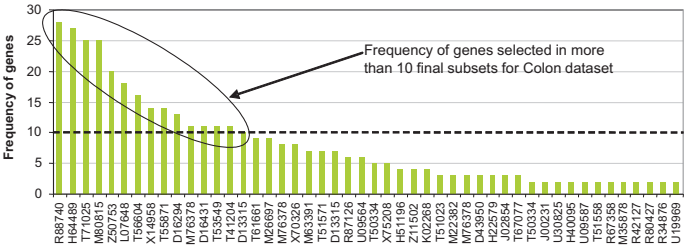

Fig. 3. Frequency of genes in 30 independent runs of DE-SVM in Colon

VI. Conclusions

In this paper, we propose the use of a binary version of DE, embedding a feature selection mechanism with SVM as classifier for the efficient gene selection of high dimensional Microarray datasets. The reported solutions, codifying gene subsets, are evaluated by means of their classification accuracy and percentage of reduction. The contributions are noticeable since the hybridization of DE with SVM is capable of improving broadly used methods in gene selection such as K-Means clustering, as well as different existing versions of DE. Besides, the effectiveness of our algorithm is analyzed on two well-known public datasets: DLBCL Lymphoma and Colon Tumor discovering newly challenging gene subsets, and identifying specific genes that our work, fitting in with known studies in the literature, suggest as highly significant.

As further work, we are interested in evaluating our algorithm in new Microarray datasets, and plan to test and compare different evolutionary approaches in order to offer new performance studies.

Acknowledgment

Authors acknowledge funds from the Spanish Ministry of Sciences and Innovation European FEDER under contract TIN2008-06491-C04-01 (M* project `http://mstar.lcc.uma.es`) and CICE, Junta de Andalucía under contract P07-TIC-03044 (DIRICOM project `http://diricom.lcc.uma.es`). J. Apolloni under research contract MAEC (BOE 157/2008) acknowledge funds from the Spanish Agency for Int. Cooperation and Development (AECID)

References

[1] E. Alba, J. García-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-07*, pages 284–290, Singapore, Sep 2007.

[2] E. Alba, G. Luque, J. García-Nieto, G. Ordonez, and G. Leguizamón. Mallba: a software library to design efficient optimisation algorithms. *Int. J. of Innovative Computing and Applications*, 1(1):74–85, 2007.

[3] Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci*, 96:6745–6750, 1999.

[4] C-C. Chang and C-J. Lin. LIBSVM: A library for support vector machines, 2002.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[6] Alizadeh AA et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 11:403–503, 2000.

[7] J. García-Nieto, E. Alba, L. Jourdan, and E. Talbi. Sensitivity and specificity based multiobjective approach for feature selection: Application to cancer diagnosis. *Information Processing Letters*, April 2009. In Press.

[8] Tao Gong and Andrew L. Tuson. *Differential Evolution for Binary Encoding*, pages 251–262. Advances in Soft Computing. 2007.

[9] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[10] J. Hernandez, B. Duval, and J-K. Hao. A genetic embedded approach for gene selection and classification of microarray data. In E. Marchiori el al, editor, *LNCS of EvoBio*, pages 90–101, 2007.

[11] J. Kohavi and G. H. John. The wrapper approach. In *Feature Selection for Knowledge Discovery and Data Mining*, pages 33–50, 1998.

[12] M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Trans. on Computer*, 26:917–922, 1977.

[13] A.C. Pease, D. Solas, E. Sullivan, M. Cronin, C. P. Holmes, and S. Fodor. Light-generated oligonucleotide arrays for rapid dna sequence analysis. In *Proc. Natl. Acad. Sci.*, volume 96, pages 5022–5026, USA, 1994.

[14] K. V. Price, R. Storn, and J. Lampinen. *Differential Evolution: A practical Approach to Global Optimization*. Springer, London, 2005.

[15] D. K. Tasoulis, V. P. Plagianakos, and M. N. Vrahatis. *Differential Evolution Algorithms for Finding Predictive Gene Subsets in Microarray Data*, volume 204 of *IFIP International Federation for Information Processing*, pages 484–491. 2006.

[16] Y. Wang, I. V. Tetko, M. A. Hall, E. Frank, A. Facius, and H. Mewes K. Mayer. Gene selection from microarray data for cancer classification–a machine learning approach. *Computational Biology and Chemistry*, 29(1):37 – 46, 2005.

[17] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. M. Kaufmann, 2005.