УДК 519.7
V. Lytvynenko
Kherson National Technical University ,
Dept. of Informatics & Computer Science

HYBRID SWARM NEGATIVE SELECTION ALGORITHM FOR DNA-MICROARRAY DATA CLASSIFICATION

In the paper, a classification method is proposed. It is based on Combined Swarm Negative Selection Algorithm, which was originally designed for binary classification problems. The accuracy of developed algorithm was tested in an experimental way with the use of microarray data sets. The experiments confirmed that direction of changes introduced in developed algorithm improves its accuracy in comparison to other classification algorithms.

Key words: Negative Selection Algorithm, Swarm Selection Algorithm, Classifier, DNA-Microarray Data, Principal Component Analysis, Wavelet transformation, Feature reduction, Feature selection.

1.Introduction

DNA microarray technology, introduced in 1995–1996, allows the measurement of thousands of gene expression values simultaneously, providing insight into the global gene expression patterns of cells (tissues) being studied [1,2,3]. Despite the need for further technological developments with microarray assays [4], the approach remains powerful for studying the myriad of transcription-related pathways involved in cellular growth, differentiation, and transformation in various organisms. In particular, the ability to measure thousands of gene expressions simultaneously using DNA microarrays has made it possible to investigate genome-wide objective approaches to molecular cancer classification[5]. Empirical microarray data produce large datasets having expression levels of thousands of genes with a very few numbers (upto hundreds) of samples which leads to a problem of "curse of dimensionality". Due to this high dimension the accuracy of the classifier decreases as it attains the risk of overfitting. As the microarray data contains thousands of genes, hence a large number of genes are not informative for classification because they are either irrelevant or redundant. Hence to derive a subset of informative or discriminative genes from the entire gene set is necessary and a challenging task in microarray data analysis. The purpose of gene selection or dimension reduction is to simplify the classifier by retaining small set of relevant genes and to improve the accuracy of the classifier. For this purpose, researchers have applied a number of test statistics or discriminant criteria to find genes that are differentially expressed between the investigated classes [15].A typical DNA microarray data set in tumor tissue c classification studies consists of expression measurements on thousands of genes over a small number of known tumor tissue samples ($p \gg N$). However, many standard statistical methodologies for classification and prediction require more samples than predictors. For example, in regression, $N < p$ leads to an ill-posed problem because the ordinary least squares (OLS) solution is not unique. Another example is Fisher's discriminant analysis, where the covariance matrix is singular when $N < p$ [5]. It is challenging to use gene expression data for cancer classification because of the following two special aspects of gene expression data. First, gene expression data are usually very high dimensional. The dimensionality ranges from several thousands to over ten thousands. Second, gene expression data sets usually contain relatively small numbers of samples, e.g., a few tens. If we treat this pattern recognition problem with supervised machine learning approaches, we need to deal with the shortage of training samples and high dimensional input features. Recent approaches to solve this problem include artificial neural networks [7], an evolutionary algorithm[8], nearest shrunken centroids [9], and a graphical method [10]. A number of recent publications report on the successful application of support vector machines (SVMs) to the classification of high-dimensional microarray data [11-13]. Therefore, high-dimensional microarray data

present a major challenge for these classifiers. However, the algorithms of Artificial Immune System (AIS) have not been widely explored for cancer classification with microarray data. Yet there exist in literature only very few studies in which AIS were applied to microarray classification. Therefore, this study introduced an artificial immune system approach for cancer detection based on negative selection algorithm (NSA) and Particle Swarm optimization (CSA) named Hybrid Swarm Negative Algorithm (HSNA).

2. MATERIALS AND METHODS

In this study, the microarray data classification was performed in three stages: dimensionality reduction using the Principal Component analysis, Feature extraction using the discrete wavelet transform and classification using hybrid swarm negative algorithm (HSNA).

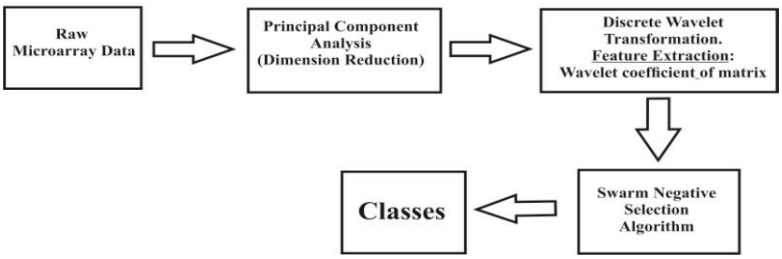


Figure 1. Structure of the HCNA Classifier

2.1 Dataset

Microarray datasets take the form of expression data matrix where rows represent the genes and columns represent the samples. Each cell in this data matrix is a gene expression value which expresses the gene intensity in the corresponding sample. The expression data matrix will be finally dealt with in the form X_{ij} where; $0 < i \leq n_g, 0 < j \leq n_s$ and n_s, n_g are the total number of genes, total number of samples respectively as in figure 2. Each expression data matrix will be further divided into two matrices; training data matrix Y_{ik} and test data matrix Z_{ip} where k, p are the number of samples used in the training process, test process respectively and $p + k = n_s$. The training data matrix will be used to train all the used classifiers and their performance will be evaluated using the test data matrix only [16] .

$$x_{ij} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n_s} \\ x_{21} & x_{22} & & x_{2n_s} \\ & & & \\ x_{n_g 1} & & & x_{n_g n_s} \end{pmatrix}$$

Figure 2. Expression data matrix

In this section, the cancer gene expression data sets used for the study are described. These datasets are also summarized below.

ALL/AML Leukemia Dataset. The dataset consists of two distinctive acute leukemias, namely AML and ALL bone marrow samples with 7129 probes from 6817 human genes. The training dataset consists of 3B8 samples (27 ALL and 11 AML) and the test dataset consists of 34 samples (20 ALL and 14 AML).

Colon Dataset. The dataset consists of 62 samples from 2000 genes. The training dataset consists of 42 samples where (30 class1, 12 class2) and the test data set consists of 20 samples (10 class1, 10 class2).
Prostate cancer. Prostate cancer data contains training set of 52 prostate tumor samples and 50 nontumor (labeled as "Normal") prostate samples with 12 600 genes. An independent set of test samples is also prepared, which is from a different experiment. The test set has 25 tumor and 9 normal samples.

2.2 Dimensionality reduction using the Principal Component analysis

Principal component analysis (PCA) is used to search new abstract orthogonal principal components (eigenvectors) which explain most of the data variation in a new coordinate system [17]. Classical PCA is based on the decomposition of a covariance/ correlation matrix by eigenvalue (spectral) decomposition (EVD) or by the decomposition of real data matrixes using SVD [18].

PCA is a multivariate procedure aimed at reducing the dimensionality of multivariate data while accounting for as much of the variation in the original data set as possible.

This technique is especially useful when the variables within the data set are highly correlated and when there is a higher than normal ratio of explanatory variables to the number of observation. Principal components seeks to transform the original variable to a new set of variables that are (1) linear combinations of the variables in the data set, (2) uncorrelated with each other, and (3) ordered according to the amount of variation of the original variables that they explain [17,19]

PCA is a well-known method of dimension reduction [20]. The basic idea of PCA is to reduce the dimensionality of a data set, while retaining as much as possible the variation present in the original predictor variables. This is achieved by transforming the p original variables $X = \lfloor x_1, x_2, \ldots, x_p \rfloor$ to a new set of K predictor variables, $T[t_1, t_2, \ldots, t_K]$, which are linear combinations of the original variables. In mathematical terms, PCA sequentially maximizes the variance of a linear combination of the original predictor variables,

$$u_K = \arg\max_{u'u} Var(Xu) \tag{1}$$

subject to the constraint $u_i' S_X u_j = 0$, for all $1 \le i \le j$. The orthogonal constraint ensures that the linear combinations are uncorrelated, i.e. $Cov(Xu_i, Xu_j) = 0, i \ne j$. These linear combinations

$$t_i = Xu_i \tag{2}$$

are known as the principal components (PCs) [21]. Geometrically, these linear combinations represent the selection of a new coordinate system obtained by rotating the original system. The new axes represent the directions with maximum variability and are ordered in terms of the amount of variation of the original data they account for. The first PC accounts for as much of the variability as possible, and each succeeding component accounts for as much of the remaining variability as possible. Computation of the principal components reduces to the solution of an eigenvalue-eigenvector problem. The projection vectors (or called the weighting vectors) u can be obtained by eigenvalue decomposition on the covariance matrix S_X,

$$S_X u_i = \lambda_i u_i \tag{3}$$

where λ_i is the *i*-th eigenvalue in the descending order for $i = 1, \ldots, K$, and u_i is the corresponding eigenvector. The eigenvalue λ_i measures the variance of the *i*-th PC and the eigenvector u_i provides the weights (loadings) for the linear transformation (projection). The maximum number of components K is determined by the number of nonzero eigenvalues, which is the rank of S_X, and $K \le \min(n, p)$. The computational cost of PCA, determined by the number of original predictor variables p and the number of samples n, is in the order of $\min(np^2 + p^3, pn^2 + n^3)$. In other words, the cost is $O(pn^2 + n^3)$ when $p > n$ [22].

2.3 Discrete wavelet transform-feature extraction

Suppose that the vector $\overline{\xi}_1$ has a sequence consisting of the 2^n points, for some integer n> 0. This sequence can be identified with the next function in the space V^n of piecewise constant functions at equidistant intervals of length $1/2^n$:

$$f(t) = x_1\phi_{n,0}(t) + ... + x_{2^n}\phi_{n,2^n-1}(t) \quad (4)$$

where $\phi(t)$- scaling functions of space V^n. The first step in calculating the wavelet decomposition of the sequence $\{x_1, x_2, ..., x_{2^n}\}$ is the decomposition of $f(t)$ on the alternative basis of the space V^n, which constitute half of the wavelets $\psi(t)$:

$$f(t) = A_{n-1,0}\phi_{n-1,0}(t) + ... + A_{n-1,2^{n-1}-1}\phi_{n-1,2^{n-1}-1}(t) + D_{n-1,0}\psi_{n-1,0}(t) + ... + D_{n-1,2^{n-1}-1}\psi_{n-1,2^{n-1}-1}(t) \quad (5)$$

where A - approximation coefficients, defining coarse low-frequency component of the original signal, D - detail coefficients, defining the high-frequency component of the original signal. The next step of the conversion process is the use of the same basic conversion the members of (2), containing the approximation coefficients. Detail coefficients at the same time remain unchanged. Block diagram of the wavelet decomposition is presented in Figure 3.:

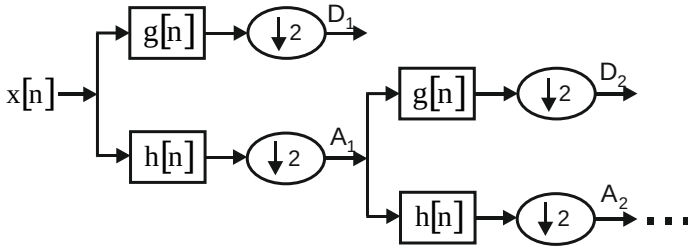


Figure.3. Structural diagram of a discrete wavelet decomposition of the signal; $g[n]$ - high frequency transmit filter, $h[n]$- low frequency filter transmitting.

The Data used in this research were analyzed into the details D1-D2 and one final approximation, A2. Our previous studies [23] have shown that the smoothing feature of the Daubechies wavelet of order 13 (db13) made it more suitable to detect changes of the microarray data. Hence, in our research, we used the db13 to compute the wavelet coefficients of the microarray data. A detailed description of the method of obtaining the approximation coefficients described by us in [32].

The computed discrete wavelet coefficients provide a compact representation. In order to further decrease the dimensionality of the extracted feature vector, statistics over the set of the wavelet coefficients are used. The following statistical features were used to represent the time-frequency distribution of the microarray data:

- Maximum of the wavelet coefficients in each subband
- Minimum of the wavelet coefficients in each subband
- Mean of the wavelet coefficients in each sub-band
- Standard deviation of the wavelet coefficients in each sub-band.

2.4 Artificial immune algorithms

In the 1990s, Artificial Immune System (AIS) emerged as a new computational research filed inspired by simulation of biological behavior of Natural Immune System (NIS). The NIS is a very complex biological network with rapid and effective mechanisms for defending the body against a specific foreign body material or pathogenic material called antigen .

The Artificial Immune Systems, as defined by de Castro and Timmis [24] are: "Adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving". However AIS are one of many types of algorithms inspired by biological systems, such as neural networks, evolutionary algorithms and swarm intelligence. There are many different types of algorithms within AIS and research to date has focused primarily on the theories of immune networks, clonal selection and negative selection. These theories have been abstracted into various algorithms and applied to a wide variety of application areas such as anomaly detection, pattern recognition, learning and robotics [25].

Negative selection algorithm. The negative selection of T-cells is responsible for eliminating the T-cells whose receptors are capable of binding with self-peptides presented by self-MHC molecules. This process guarantees that the T-cells that leave the thymus do not recognize any self-cell or molecule. Forrest et al. [26] proposed a change detection algorithm inspired by the negative selection of T-cells within the thymus. This procedure was named as negative selection algorithm and was originally applied in computational security. A single type of immune cell was modelled: T-cells were represented as bit strings of length *L*. The negative selection algorithm of Forrest and collaborators is simple [26]. Given a set of self-peptides, named self-set S, the T-cell receptors will have to be tested for their capability of binding the self-peptides. If a T-cell recognizes a self-peptide – it is discarded, else it is selected as an immune-competent cell and enters the available repertoire A.

The idea of negative selection algorithm is to generate a set of detectors in a complementary set of **N** and then to use these detectors for binary classification as "Self" or "Non-Self". Formally, the negative selection algorithm can be represented as [27-28]:

$$NegA\lg = \left(\Sigma^{L}, L, \mathbf{S}, \mathbf{N}, r, n, s, pr\right) \tag{6}$$

where Σ^{L} denotes shape-space; L is receptor length; **S** is "Self" detector set; **N** is "Non-Self" detector set; r denotes cross-reactive threshold; n is total number of appointed detectors; s is detector set size; pr denotes rule matching rows in adjacent positions.

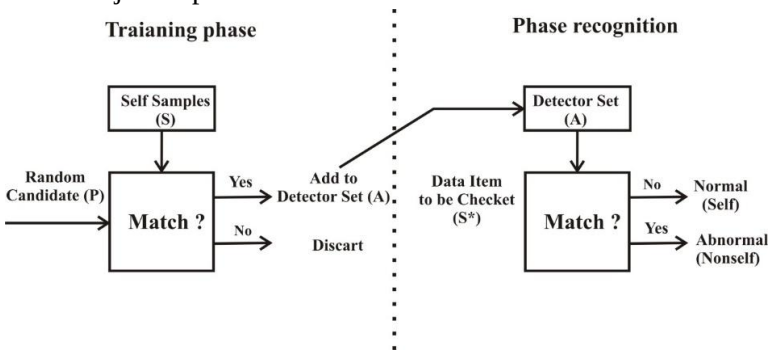


Figure 4. Negative selection algorithm [29]

The negative selection algorithm can be summarized as follows:

- *Initialization:* randomly generate strings and place them in a set **P** of immature T-cells, assuming all the molecules (receptors and self-peptides) are represented as binary strings of the same length ***L***.
- *Affinity evaluation:* determine the affinity of all T-cells in V with all elements of the self set **S**.
- *Generation of the available repertoire:* if the affinity of an immature T –cell with at least one self-peptide is greater than or equal to a give cross reactive threshold, then the T-cell recognizes this self-peptide and has to be eliminated (negative selection); else the T-cell is introduced into the available repertoire **A**.

The process of generating the available repertoire in the negative selection algorithm was termed learning phase. The algorithm is also composed of a monitoring phase. In the monitoring phase, a set $\mathbf{S}^{*}$ of protected strings is matched against the elements of the available repertoire A. The set **S*** might be the own set **S**, a completely new set, or composed of elements of **S**. If recognition occurs, then a non-self pattern (string) is detected.

It is well known, that the algorithm of negative selection (NS) has the some restrictions and limitations [29]. When it is not appropriate, for example, the number of self samples is small and sparse.
Some limitations of the binary-string representation in NS algorithms are as follows:

- binary matching rules are not able to capture the semantics of some complex self/non-self spaces,
- it is not easy to extract meaningful domain knowledge,
- in some cases a large number of detectors are needed to guarantee better coverage (detection rate),
- it is difficult to integrate the NS algorithm with other immune algorithms,
- the crisp boundary of "self" and "non-self" may be very hard to define.

In real-valued representation the detectors are represented by hyper-shapes in n-dimensional space. The algorithms use geometrical spaces and use heuristics to distribute detectors in the non-self space.
Some limitations of the real-valued representation in NS algorithms are:

- the issue of holes in some geometrical shapes, and may need multi-shaped detectors,
- curse of dimensionality,
- the estimation of coverage,
- the selection of distance measure.

During our experiments it has been established that generation of set of detectors in at training phase occurs casually owing to what it is in advance impossible to define is minimum necessary quantity of detectors which will provide the maximum quality of recognition. The increase in quantity of detectors conducts to delay of a phase of recognition, and its reduction – to deterioration of work of algorithm since the probability of formation of the "cavities" which are areas in space of "Non-self" which are not distinguished by any of detectors increases. Thus, a problem of the given research is working out of an advanced method of generation of the detectors, capable to adaptive selection of their options, quantity and an arrangement.

Particle swarm optimization (**PSO**) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-spaceaccording to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position but, is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.
PSO is originally attributed to Kennedy, Eberhart and Shi [24,25] and was first intended for simulating social behaviour,[26] as a stylized representation of the movement of organisms in a bird flock or fish school. The algorithm was simplified and it was observed to be performing optimization. The book by Kennedy and Eberhart[33] describes many philosophical aspects of PSO and swarm intelligence. An extensive survey of PSO applications is made by Poli [34,35].
PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found. More specifically, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient descent and quasi-newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc.
The PSO algorithm is population-based: a set of potential solutions evolves to approach a convenient solution (or set of solutions) for a problem. Being an optimization method, the aim is finding the global optimum of a real-valued function (fitness function) defined in a given space (search space).
The social metaphor that led to this algorithm can be summarized as follows: the individuals that are part of a society hold an opinion that is part of a "belief space" (the search space) shared by every possible individual. Individuals may modify this "opinion state" based on three factors:

- The knowledge of the environment (its fitness value)
- The individual's previous history of states (its memory)
- The previous history of states of the individual's neighborhood

An individual's neighborhood may be defined in several ways, configuring somehow the "social network" of the individual. Several neighborhood topologies exist (full, ring, star, etc.) depending on whether an individual interacts with all, some, or only one of the rest of the population.
Following certain rules of interaction, the individuals in the population adapt their scheme of belief to the ones that are more successful among their social network. Over the time, a culture arises, in which the individuals hold opinions that are closely related. The particle swarm optimization is based on a set of individuals originally randomly arranged and homogeneous. Therefore we call it particles, which move in the hyperspace of research and are each a potential solution. Each particle has a memory about his best seen as the ability to communicate with the particles forming around it. From this information, the particle will follow a trend made, from one side, willingness to return to its optimal solution, and from the other side, his mimicry in relation to the solutions found in its vicinity. From local optima and empirical, all particles will normally converge to the global optimum solution of the addressed problem.

The process of finding the particles is based on two rules:

1) Each particle has a memory that can store the best point by which it has already passed and it tends to return to this point;
2) Each particle is informed of the best known point in its neighborhood and it`will tend to move towards this point.

Each particle moves according to a compromise between the three following trends:

a) Repeat its previous motion;
b) Move towards its best previous position;
c) Move towards the best position (past) of its group of informants.

Each agent tries to modify its position based on the following information:

a) Current positions (x, y)
b) Current velocities (v_x, v_y)
c) Distance between the current position and **pbest**
d) Distance between the current position and **gbest**

Thus, the velocity of the particle i is updated using the following equation:

$$v_i^{k+1} = wv_i^k + c_1 rand_1\left(pbest - S_i^k\right) + c_2 rand_2\left(gbest - S_i^k\right) \tag{7}$$

Where w the inertia weight, c_1 and c_2 are the acceleration constants $(c_1 + c_2 \le 4)$, $rand_1$ and $rand_2$ are random numbers in the interval $[0\ 1]$.

Then the position of the particle S_i^k is modified from the current position and a new speed is calculated v_i^{k+1}:

$$S_i^{k+1} = S_i^k + v_i^{k+1} \tag{8}$$

The weight w is given by the following equation:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} iter \tag{9}$$

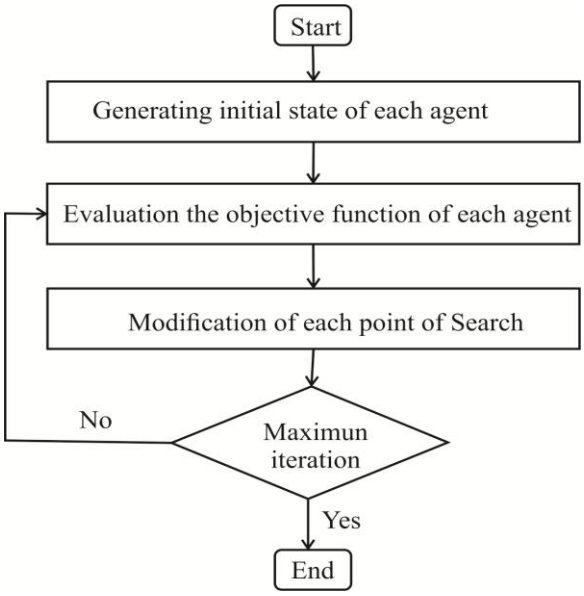
$w = [0.4 - 0.9]$ during the search procedure gives better results.

The right choice of parameters will allow the rapid convergence and minimizes the computation time. The first rule stipulates that c_1 must have an absolute value less than 1 in practice, this coefficient should be neither too small on PSO recommended that it be equalized to 0.7 or 0.8. The parameter c_2 should not be too large, a value of about 1.5 to 1.7 being regarded as effective in the majority of cases. The pairs of values (0.7 1.47) and (0.8 1.62) are indeed correct. The following figure shows the general flowchart of PSO. The steps involved in the optimization algorithmic of the particle swarm are as follows:

Figure 5 General flowchart of the PSO algorithm [36]

Step 1: Select several parameters of PSO

Step 2: Initialize a population of particles with random position and velocities in the problem search



space;

Step 3: Evaluate the ability of optimization for a desired personal touch to each particle;

Step 4: For each individual particle, compare the value of the particle with its ability ***pbest*** , If the current value is better than the evaluated ***pbest*** for the agent i ;

Step 5: Identify the particle that has the best value of its fitness function which will be identified as ***gbest***;

Step 6: Calculate the new speed and position of particles using equations 15 and 16.

Step 7: Repeat steps 3-6 until the stopping criterion is met.

The generalized flowchart of the PSO algorithm is presented on a figure 5.

Combined particle swarm optimization and negative selection algorithm. The classifier presented in this paper is based on the hybridization process of negative selection with clonal selection, and was designed to solve problems of classification to many classes. Concept of classification is used in terms of supervised learning, which allows categorizing objects into known groups using training set prepared beforehand. The main task of every classifier based on supervised learning is to create an internal representation of classes (in the form of a function, set of rules or any other). It acquires it during training. When the training is completed the classifier is ready to produce an answer to any (known or unknown) pattern given subsequently.

In this study the efficiency of immune classifiers is researched, when as the classifier, in general, is a function that for attributes vector of object shall decide to which class it belongs [27]:

$$F : \Re^n \to Y . \tag{10}$$

The function *F* represents the space of sign vectors in the space of the class labels *Y*. In the case of two classes $Y = \{0,1\}$, '1' corresponding case of the detection event, '0' - the event is not detected. We consider the variant of training with a teacher (supervised learning), when the classifier training available to us a set of vectors $\{x\}$ for which is known their valid membership in one of the classes.

Negative selection algorithms and swarm optimization particles, described in detail in [27-29]. In this work we give an algorithm to modify the learning phase in the negative selection algorithm. The main idea is a modification of the learning phase in the negative selection algorithm. Generalized flowchart of this algorithm is shown in Figure 6. Each particle D_i has a velocity vector and the coordinates, and is the immune detector. Cross-reactive threshold of this detector is used to separate the population of antigens into two subsets - "Self" and "Non-self". The learning process is to optimize of this algorithm by using PSO detector location, resizing, changing their cross-reactive thresholds.

For evaluation criteriause values of affinity to antigens, and the detectors together. The flowchart of the learning phase the proposed algorithm is shown in Figure 9.

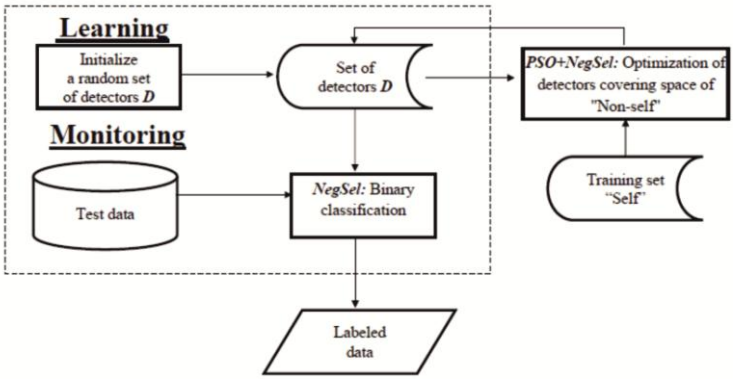


Figure 6. Flowchart learning phase, the proposed algorithm

Step by step description of the algorithm that implements the proposed method:

Step 1. At this step, initializes the population of particles (detectors) with randomly chosen positions and random velocity vectors. The size of this set remains constant. In the learning process only change the properties of its elements.

Step 2. A loop is created for all elements the learning set "Self" (set of antigens). Further actions are performed sequentially for each antigen.

Step 3. A loop is created for each detector from the population.

Step 4. To calculate the affinity values of the relationship “antigen-antibody” is used the ratio:

$$f_{Ab-Ag} = \frac{k_\varepsilon}{\varepsilon} + D_{E(Ab-Ag)} \to \min \qquad (11)$$

where r is the cross-reactive antibodies (detector) threshold, k_r - coefficient of importance of cross-reactive threshold (parameter settings). Parameter k_r is very important study parameter. He manages the robustness of the resulting solution. Increasing this parameter causes the immune network to maintain a larger radius detectors, what gives a coarser, but more sustainable solutions. However, excessive increase of k_r affects negative to the accuracy of solutions. The figure 7 demonstrates the impact parameter k_r on

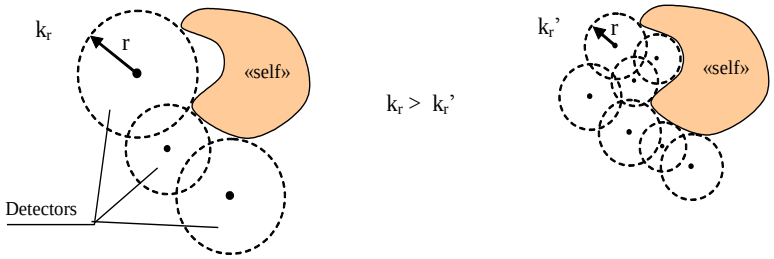


Figure 7. The influence of parameter value k_r on the way of detector generation

the way of detector generation.

Steps 5,6. Based on the calculated values of the affinity, the best particles are calculated in the

Step 7. According to estimates computed values is refreshed values of the velocity vectors and the coordinates of the particles in the swarm. In this case, the ratio used, the algorithm used swarms of particles.

To calculate the values of affinity relationship "antibody-antibody" proposed the following formula:

$$f_{Ab-Ab} = -\frac{D_{E(Ab_1-Ab_2)} - (\varepsilon_{Ab_1} + \varepsilon_{Ab_2})}{2 \cdot \min(\varepsilon_{Ab_1}, \varepsilon_{Ab_2})}, \qquad (12)$$

This possible interpretation of such values f_{Ab-Ab}: ≤ 0 - recognizing hyper sphere detectors do not overlapped. This option does not require compression, because the antibodies do not recognize each other; $(0,1)$ - hyper sphere overlapped by shells, namely the value is the degree of overlap. This compression is dependent on the value of the parameter of threshold compression σ_s, which is a parameter of study; ≥ 1 - hyper sphere smaller radius (r) is completely inside hyper sphere larger radius. In this case, compression is definitely needed, because there is redundancy recognizing elements.

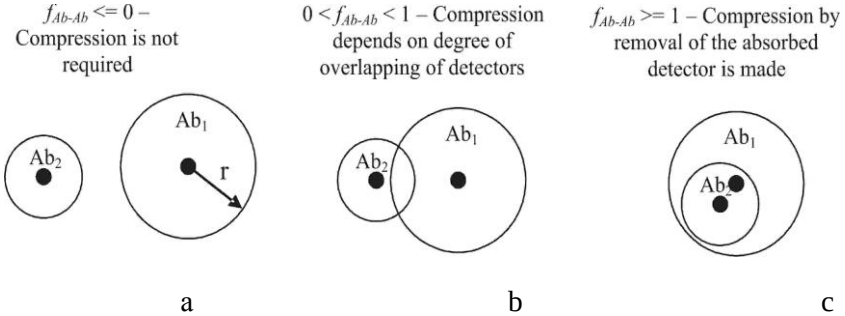


Figure 8. Different mutual arrangement of the recognizing hyper sphere of detectors depending on the value f_{Ab-Ab}

Step 10. Based on the values calculated estimates relations "detector-detector" is corrected values of the velocity vectors and the coordinates of each particle in the population.

Step 11. Found the best particle returns to a population of memory cells, replacing a disposal corresponding in order to preserve a permanent composition of the population.

Steps 12, 13, 14. This process repeated as long until certain conditions are fulfilled stop (for example, the quality of classification). At the same time output of this phase is learning the population of memory cells (detectors).

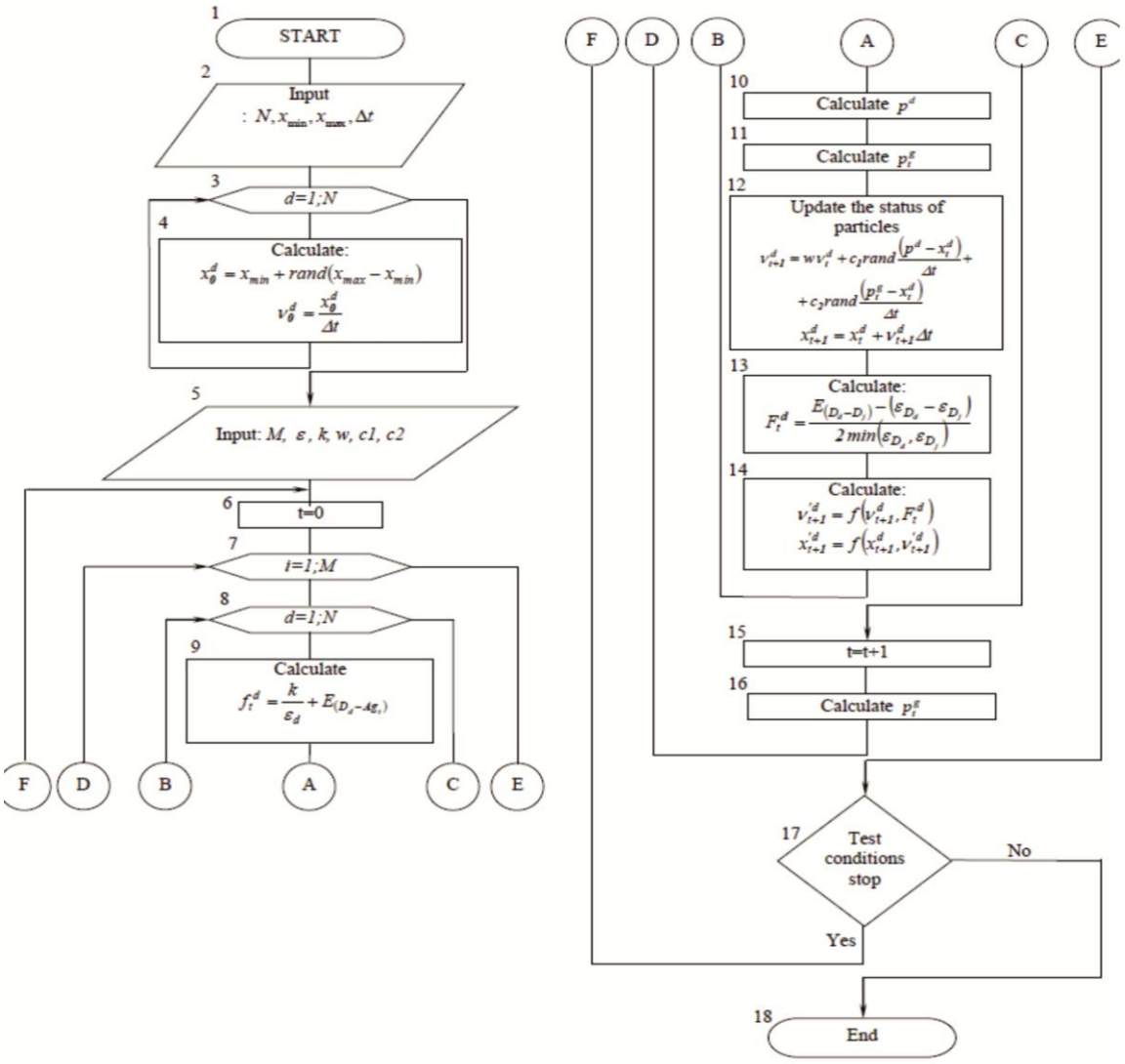


Figure 9 . Flowchart of the learning phase the hybrid algorithm developed

3. RESULTS AND DISCUSSION

It is common practice in machine learning and data mining to perform k-fold cross-validation to assess the performance of a classification algorithm. K-fold cross validation is used among the researchers, to evaluate the behavior of the algorithm in the bias associated with the random sampling of the training data. In k-fold cross-validation, the data is partitioned into k subsets of approximately equal size. Training and testing the algorithm is performed k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Thus, k different test results exist for the algorithm. However, these k results are used to estimate performance measures for the classification system.

The common performance measures used in medical diagnosis tasks are accuracy, sensitivity and specificity. Accuracy measured the ability of the classifier to produce accurate diagnosis. The measure of the ability of the model to identify the occurrence of a target class accurately is determined by sensitivity. Specificity is determined the measure of the ability of the algorithm to separate the target class. The classification accuracies for the datasets are calculated as in Eq. 15:

$$Accuracy(Z) = \frac{\sum_{i=1}^{|z|} Assess(z_i)}{|Z|} \quad (13)$$

while

$$Assess(z) = \begin{cases} 1, & if\ \ calssify(z) = z.c \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where z denotes the patterns in testing set to be classified, $z.c$ is the class of pattern z, *classify*(z) returns the classification of z by classification algorithm.
For sensitivity and specificity analysis, the following equations can be used:

$$Sensitivity = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{15}$$

$$Specificity = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{16}$$

where TP, TN, FP i FN denote respectively true positive, true negative, false positive and false negative classification.

In order to compare the efficiency of the proposed method in predicting the class of the cancer microarray data we have used three standard datasets such as All/AML Leukemia, Colon Dataset, Prostate cancer. All the datasets is binary class datasets. The feature selection process proposed in this paper has two steps. First the microarray data is decomposed by factor analysis optimally choose the discriminate feature set then using Discrete wavelet transform into level 4 using db13 wavelet to get the approximation coefficients as the extracted feature set. The performance of the proposed feature extraction method is analyzed with the well studied neural network classifiers such as MLP and RBFNN. The leave one out cross validation (LOOCV) test is conducted by combining all the training and test samples for both the classifiers with all the three datasets and the results are listed in Table 1. For this Data the performance of HSNA is comparable to RBFN and MLP.

Table 1
Comparison study of classification accuracy, sensitivity and specificity of HCNA with MLP and RBFN classifiers

Dataset	Method	Classification Accuracy	Sensitivity	Specificity
ALL/AML Leukemia	MLN	91.3%	98.21	97.01
	RBFN	98.4%	98.55	97.25
	HSNA	100%	99.16	99.20
Colon Dataset	MLN	94.5%	97.44	96.25
	RBFN	97.6%	98.10	97.10
	HSNA	99.7%	98.80	99.45
Prostate cancer	MLN	98.8%	98.03	97.21
	RBFN	97.2%	98.80	98.00
	HSNA	100%	99.10	99.60

The performance of the proposed method is also compared with those obtained by the recently reported methods and the results are listed in Table 2-4. The existing methods also used the cross validation test on the datasets. From Tables 2-4 it reveals that our method is equivalent to the counterparts with the advantage of reduced computational load. Weka [31]. Table 5 shows the decomposition stages upto 4th level by using db13 in discrete wavelet transform.

Table 2.
Comparison study of accuracy of Colon Dataset

Methods	Classification accuracy
Bayes Network	85.3%

Naive Bayes classifier	60.1 %
Multinomial logistic regression model	74.2%
Support Vector Classifier	94.3%
Class for doing classification using regression methods	91.8%
Simple Decision Table Majority Classifier	89.3%
1R classifier	73.9%
C4.5 decision tree	94.5%
Forest of Random Trees	97.4%
Factor Analysis + Wavelet + HCNA	99.7%
Factor Analysis + Wavelet + HSNA	100%

Table 3.

Comparison study of accuracy of ALL/AML Leukemia dataset

Methods	Classification accuracy
Bayes Network	87.6%
Naive Bayes classifier	64.2 %
Multinomial logistic regression model	77.4%
Support Vector Classifier	91.3%
Class for doing classification using regression methods	97.6%
Simple Decision Table Majority Classifier.	92.5%
1R classifier	70.9%
C4.5 decision tree	96.5%
Forest of Random Trees	97.6%
Factor Analysis + Wavelet + HCNA	100%
Factor Analysis + Wavelet + HSNA	100%

Table 4.

Comparison study of accuracy of Prostate cancer

Methods	Classification accuracy
Bayes Network	91.7%
Naive Bayes classifier	69.2 %
Multinomial logistic regression model	80.5%
Support Vector Classifier	99.4%
Class for doing classification using regression methods	91.7%
Simple Decision Table Majority Classifier.	89.4%
1R classifier	69.2%
C4.5 decision tree	97.1%
Forest of Random Trees	98.8%
Factor Analysis + Wavelet + HCNA	100%
Factor Analysis + Wavelet + HSNA	100%

Table 5

Reduction details of the dataset

Dataset	**Original Dimension**	**Factor Analysis**	**DWT Db 13 Level 4**
Colon	62×2000	62×700	62×180
ALL/AML Leukemia	72×7129	72×700	72×180
Prostate cancer	136×12600	136×700	136×180

4. Conclusion

In this paper we have presented a hybrid feature extraction method using the Factor analysis in conjunction with wavelet transform to effectively select the discriminative genes on microarray data. A simple HSNA based classifier has also been introduced to classify the microarray samples efficiently. The comparison results elucidated that the proposed approach is an efficient method which performs better than the existing methods. Besides it has reduced computational complexity.

1. Lockhart, D.J., Dong, H., Byrne, M.C., Follettie, M.T., Gallo, M.V., Chee, M.S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., Brown, E.L., 1996. Expression monitoring by hybridization to high-density oligonucleotide arrays. Nat. Biotechnol. 14, 1675–1680.; 2. Schena, M., Shalon, D., Davis, R.W., Brown, P.O., 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270, 467–470.; 3. Schena, M., Shalon, D., Heller, R., Chai, A., Brown, P.O., Davis, R.W., 1996. Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. Proc. Natl. Acad. Sci. USA 93, 10614–10619. 4. Nguyen, D.V., Arpat, A.B., Wang, N., Carroll, R.J., 2002d. DNA microarray experiments: biological and technological aspects. Biometrics 58, 701–717.; 5. D.V. Nguyen, D.M. Rocke On partial least squares dimension reduction for microarray-based classi'cation: a simulation study / Computational Statistics & Data Analysis 46 (2004) 407 – 425.;6 Jayakishan M., Barik R. C., Ranjani P. M., S. K. Pradhan C.F. G. Dash Analysis and Wavelet Transform Method for Tumor Classification Using Gene Expression Data/ I.J. Information Technology and Computer Science, 2012, 9, 73-79.; 7. J. Khan, J. S. Wei, M. Ringner, L. H. Saal and M. Ladanyi et al., Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nature Medicine 7 (2001) 673–679.;8. J. M. Deutsch, Evolutionary algorithms for finding optimal gene sets in microarray prediction, Bioinformatics19 (2003) 45–52.; 9. R. Tibshirani, T. Hastie, B. Narashiman and G. Chu, Diagnosis of multiple cancer types by shrunken centroids of gene expression, in Proc. Natl. Acad. Sci. USA, Vol. 99 (2002), pp. 6567–6572.;10. E. Bura and R. M. Pfeiffer, Graphical methods for class prediction using dimension reduction techniques on DNA microarray data, Bioinformatics 19 (2003) 1252–1258.; 11. Ramaswamy S., Tamayo P., Rifkin R., Mukherjee S., Yeang C.H., Angelo M. Ladd C., Reich M., Latulippe E., Mesirov J.P., Poggio T., Gerald W., Loda M., Lander E.S., Golub T.R., Multiclass cancer diagnosis using tumor gene expression signatures, Proc. Natl. Acad. Sci. USA. 98(26):15149-15154, (2001).; 12. Brown M.P.S., Grundy W., Lin D., Cristianini N., Sugnet C., Furey T., Ares M., Jr., Haussler D., Knowledge-based analysis of microarray gene expression data by using support vector machines, Proc. Natl. Acad. Sci. USA. 97(1):263-267, (2000). ; 13. Mukherjee S., Classifying microarray data using support vector machines, in Berrar D., Dubitzky W., Granzow M. (eds.), A Practical Approach to Microarray Data Analysis, Kluwer Academic Publishers, Boston, pp. 166-185, (2002).; 14. Berrar, D.P., Downes, C.S., Dubitzky, W.: A Probabilistic Neural Network for Gene Selection and Classification of Microarray Data. ;In IC-AI(2003)342-352. ; 15. Meher J., Barik R. C., Panigrahi M. R., Pradhan S.K., Dash G. Cascaded Factor Analysis and Wavelet Transform Method for Tumor Classification Using Gene Expression Data / MECS I.J. Information Technology and Computer Science, 2012, 9, 73-79; 16. D. A. Salem, R.A. A. Seoud, H. A. Ali MGS-CM: A Multiple Scoring Gene Selection Technique for Cancer Classification using Microarrays /International Journal of Computer Applications (0975 – 8887) Volume 36– No.6, December 2011 p. 30 – 37.; 17. Dong Hoon Lim Principal Component Analysis using Singular Value Decomposition of Microarray Data/ World Academy of Science, Engineering and Technology 81, 2013, p.56-58.; 18. Wall M.E., Dyck P.A., Brettin T.S.(2001). SVDMAN – singular value decomposition analysis of microarray data. Bioinformatics 17:566-68.;19. Everitt, B. and T Hothorn. (2011). An Introduction to Applied Multivariate Analysis with R (Use R!). Springer, New York, NY.; 20. Jolliffe, I.T. (1986). Principal Component Analysis. Springer, New York.; 21. Massey, W.F. (1965) Principal components regression in exploratory statistical research. Journal of American Statistical Association, 60, 234-246. ; 22. Jian J. Dai, Linh Lieu, David Rocke Dimension Reduction for Classification with Gene Expression Microarray Data/ Stat Appl Genet Mol Biol. 2006;5:Article6. Epub 2006 Feb 24. 23. Babichev S.A., Babenko N.I., Dydik A.A. Lytvynenko V.I., Fefelov A.A., SV Shkurdoda Filtration chromatogram with pomoshchju veyvelet-analysis with Using Criteria entropyy (in Russion)/ Regional mizhvuzovskyy collection of scientific papers Issue 6 (71) Dnipropetrovsk 2010 C-17-3224. ; 24. Kennedy, J., Eberhart, R. (1995). "Particle Swarm

Optimization". Proceedings of IEEE International Conference on Neural Networks ***IV****. pp. 1942–1948.doi:10.1109/ICNN.1995.488968.; 25. Shi, Y.; Eberhart, R.C. (1998). "A modified particle swarm optimizer". Proceedings of IEEE International Conference on Evolutionary Computation. pp. 69–73.; 26. Forrest, S., A.S. Perelson, L. Allen and R. Cherukuri, 1994. Self-nonself discrimination in a computer. Proceedings of the Symposium IEEE Computer Society on Research in Security and Privacy, May 16-18, IEEE Xplore Press, Oakland, CA., USA., pp: 202-212. DOI: 10.1109/RISP. 1994.296580; 27. Lytvynenko V. I.: Comparative experimental study of a modified negative selection algorithm and clonal selection algorithm negative for solving classification (in Russian), Vestnik Kherson National Technical University, nr 4(33), 2008, pp. 7-14. ;28. Lytvynenko V. I.: Immune classifier for solving binary classification - Theoretical Aspects (in Russian), System technologies, nr 1(42), 2006, pp. 32-47. ; 29. D. Dasgupta. Advances in Artificial Immune Systems. IEEE Computational Intelligence Magazine. November, 2006. - P.40-49.; 30. Kennedy, J. "The particle swarm: social adaptation of knowledge". Proceedings of IEEE International Conference on Evolutionary Computation. (1997). pp. 303–308.; 31. http://www.cs.waikato.ac.nz/ml/weka/; Lytvynenko V.I. Application of Clonal Negative Algorithm to Cancer Classification with DNA-Microarray Data// System technologies. – №6 (83). – Dnepropetrovsk, 2012. – P. 72 – 90. 32.Kennedy, J., Eberhart, R.C. (2001). Swarm Intelligence. Morgan Kaufmann. ISBN 1-55860-595-9. ; 33.Poli, R. (2007). "An analysis of publications on particle swarm optimization applications". Technical Report CSM-469 (Department of Computer Science, University of Essex, UK).; 34. Poli, R. (2008). "Analysis of the publications on the applications of particle swarm optimisation". Journal of Artificial Evolution and Applications* ***2008****: 1–10.doi:10.1155/2008/685175.; 35. Mekhanet, Med.; Mokrani, L.; Lahdeb, Med. Comparison between Three Metaheuristics Applied to Robust Power System Stabilizer Design// Acta Electrotehnica . 2012, Vol. 53 Issue 1, p41-49. 9p.*