Microarray Classification and Gene Selection with FS-NEAT

Bruno Iochins Grisci*, Bruno César Feltes† and Márcio Dorn‡
Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil
*bigrisci@inf.ufrgs.br, †bcfeltes@inf.ufrgs.br, ‡mdorn@inf.ufrgs.br

***Abstract*—The analysis of microarrays has the potential to identify and predict diseases predisposition, such as cancer, opening a new path to better diagnosis and improved treatments. Additionally, microarrays can help to find genetic biomarkers, which are genes whose expressions are related to a specific disease stage or condition. But due to the huge number of genes present in microarray experiments, and the small number of available samples, computational methods that deal with such techniques need to overcome difficulties in both classification and feature selection tasks. This paper presents adaptations for the use of FS-NEAT, an evolutionary algorithm that creates and optimizes neural networks through genetic algorithms, as a tool that can satisfactorily perform both tasks simultaneously and automatically. The method is tested with a Leukemia dataset containing six imbalanced classes, compared with other classifiers, and the selected genes are biologically validated.**

I. Introduction

Microarrays are arrays experiments designed for nucleic-acid hybridization [1]. Each microarray experiment requires a special chip, with thousands of probes, where each of these probes contains a nucleic acid sequence. Usually, microarrays function as a tool to identify expression of genes present in a given biological sample, derived from RNA extraction of a target tissue or cell culture. In this sense, target RNAs are codified to complementary DNA (cDNA) using the Reverse-Transcriptase Polymerase Chain Reaction (RT-PCR) technique, which will then hybridize with the nucleic acid sequence of the probe and emit a signal that can be translated as a wavelength, indicating if the target gene is present in a given sample or not [1], [2]. Microarrays have been used to analyze a wide variety of diseases, such as cancer [3], [4], [5], [6]. However, despite their enormous potential, microarrays require the use of Bioinformatics tools to analyze and give sense to the large amount of biological data [7], [8], [9].

A common application of microarray data in Bioinformatics is its use for the creation of classifiers in the hopes of future use in medical diagnosis. Using gene expression profiles of predefined sample groups, for example, a control group and a disease group, it is possible to train supervised learning methods to assign to a new sample its correct label. This approach has great potential in clinical diagnostics and has been successfully tested with different algorithms in the last decades [10]. Different studies have already tested the efficacy of several machine learning techniques in the task of microarray classification with different datasets, exploring methods such as artificial neural networks (ANN), support vector machines (SVM), k-nearest neighbors (k-NN), and random forest (RF) [11], [12], [13], [14].

Another important aspect of working with microarray data is dimensionality reduction. Known as the ”curse of dimensionality”, this major concern refers to when the data has a large number of dimensions, which is associated with overfitting [15], increased computational run time and memory consumption, and interpretability impairment. Datasets with many dimensions but a small number of samples are also affected by the ”*large p, small n*” problem, that is often the case with microarray data. Machine learning algorithms, deep learning especially, rely in sets with thousands or even millions of samples, what can be considered a rarity with this kind of data.

Since the number of samples from microarray datasets is lower than the available number of genes (features), dimensionality reduction is a fundamental step of the process [16]. While popular methods of feature extraction, like principal component analysis (PCA), could be used, it is desirable that the selected features are not a combination of the dimensions of the data, but the dimensions themselves (e.g., the expressions of single genes). Thus, it is possible to reduce the number of features while also retrieving the information of which genes have a greater impact in the classification, finding genes that could have a high probability of being associated to a given disease.

The group of algorithms capable of performing this dimensionality reduction by selecting subgroups of features from the whole data is known as feature selection (FS) and comprises several methods that remove irrelevant, redundant or noisy features. FS has the advantage of providing a more satisfactory interpretation of the results [17], and decreasing computational cost, besides improving the accuracy of different classification methods [18].

Many FS models were proposed for microarray data, that is often noisy and contain irrelevant and redundant expressions. One example is the Minimal Redundancy and Maximum Relevancy (mRMR), a method based on Mutual Information (MI) as a measure of relevancy and redundancy, where the redundancy of a feature subset is the aggregate MI measure between all pairs of features in the subset, and the relevancy is the aggregate MI measure between all features and one specific class [19]. This algorithm has already been applied to genomic data [20], [21]. A complete review on the topic of FS and microarray can be found in the work of Ang *et*

978-1-5090-6017-7/18/$31.00 ©2018 IEEE

al. [22]. Nevertheless, this remains an open problem, with a large variety of new algorithms arising [23], [24], [16].

Besides the computational benefits, FS has the potential to aid biomarkers identification research by finding the subset of genes that best represents the whole data and increases the classification accuracy. In a nutshell, biomarkers are biological signatures found in tissues or body fluids, that can be used to identify a particular pathological or physiological process. There are several types of biomarkers, derived from a broad range of biomolecules, such as DNA, RNA, proteins, miRNA, among others. These molecules can be used for cancer detection, diagnosis, prognosis, treatment choice, or identify tumours stage [25]. The gene expression data derived from microarray experiments can aid in the identification of genes electable as possible biomarkers since microarray technology made possible the analysis of large datasets derived from various biological experiments [26].

Among the promising methods of Artificial Intelligence and Machine Learning that can be applied in the tasks of classification and FS, stands Evolutionary Computation (EC). EC borrows key concepts from evolutionary biology, such as inheritance, random variation, and selection, and adapts them to solve computational problems. EC has been used for a wide range of applications, Bioinformatics among them, and has many important benefits over popular deep learning methods. It does not require a large amount of data to solve a problem, is easily parallelized, and can give solutions based on any fitness function [27]. EC can also work well in hybrid frameworks with other machine learning algorithms [27]. For instance, Neuroevolution is a family of training methods for neural networks to obtain theirs weights, bias, and overall topology by using EC [28]. One example is the NeuroEvolution of Augmenting Topologies (NEAT) [29] that incorporates Genetic Algorithms (GA) into training.

This kind of evolutionary or constructive ANN has already been tested for the classification of microarray data. Garro *et al.* made a study combining Artificial Bee Colony (ABC) for FS and ANNs designed by Differential Evolution (DE) for classification. The ABC algorithm was used to select a more useful set of genes to discriminate a disease subtype, and this was used as input in neural networks created with DE that were free to choose their topology and activation functions. The method was tested in a Leukemia DNA microarray dataset with two classes (AML and ALL), 38 bone marrow samples, and 6817 human genes [30]. Another study by Luque-Baena *et al.* uses a genetic algorithm and C-Mantec (Competitive Majority Network Trained by Error Correction), a neural network constructive algorithm, to select a predictor profile. The approach was tested in six cancer databases [31]. Both methods, however, depend on other algorithms to perform the gene selection before the classification, and on human knowledge to define the number and criteria of selected genes.

In this sense, the NEAT algorithm is an interesting option to be explored due to its automaticity and extensibility. Using GA to create ANNs from minimalist topologies, it grows the network structure adding hidden nodes and connections. More important is that NEAT can be expanded to perform FS while evolving its networks for the classification problem. Feature Selective NEAT (FS-NEAT) is a good example because it starts with networks without any connection and lets the evolutionary algorithm choose which inputs should be connected to the other nodes [32]. This kind of technique can be applied to microarray classification problems - at the same time that it learns how to classify new samples, it selects the fundamental genes for the task that can be then submitted to a biological validation.

The main contribution of this paper is the design of a method capable of automatically performing microarray classification and gene selection at once, with the aim of identifying new biomarkers for diseases, and new ways to use FS-NEAT for the task of classifying imbalanced class datasets. This approach was evaluated with a multiclass Leukemia dataset and compared with other popular classifiers: MLP, SVM, and decision tree. We also present a biological validation of the selected genes obtained through our method, to check if the results match the biological studies. In summary, this paper is organized as follows: Section II reviews the technologies and algorithms used in the proposed method; Section III details the new algorithm for classification and gene selection; Section IV presents the experiments and analysis of the results; and Section V discusses the work and future improvements.

II. Materials and Methods

A. NEAT

Usually, when working with ANNs, a fixed topology (e.g., number of nodes, layers, and connections) is chosen, and the weights and biases of the network are determined by an algorithm such as backpropagation [33]. One of the issues that arise from this approach is how to find the best topology for a given problem since this structure can have a great impact on the learning performance of the network and its final accuracy. This can be a challenge in Bioinformatics since many of the concepts underlying biological process are only partially known [34].

NeuroEvolution of Augmenting Topologies (NEAT) is an algorithm that addresses this problem by creating and evolving ANNs using GA [29]. It is not only capable of automatically finding values for weights and biases, but also the overall topology of a network. It starts by setting a population in which individuals are ANNs sharing the same minimal topology, i.e., input and output neurons fully-connected without hidden nodes and with random weights. The minimalist start is employed to assure that only additions to the topology of a network that were beneficial to its results will be kept, barring useless complexity.

New populations are created iteratively from this first population with traditional GA operators. The crossover operation selects two individuals from the current population, generating a new individual that is a combination of both. The mutation operation can change the values of the network weights, or add new hidden nodes or a new connection between existing nodes. It can also flip a ”disable” bit that activates or deactivates a

connection. These operators are how the topology of the ANNs grows and complexifies over the generations of the GA [29].

The main challenge of implementing this method is that the crossover operation can create defective ANNs when combining two random individuals, since their topologies may not allow a direct exchange of connections and nodes. To solve this problem, NEAT uses a historical marking - a numerical value assigned to new pieces of structure, like a new connection, found through the modifications. This value is determined linearly by when in the evolutionary process the new structure first appeared and is passed as it is to new individuals during the crossover. Hence, NEAT is capable of perfectly matching the same structures in two different topologies by aligning the ones with equal historical markings, creating a new functional ANN that has the same building blocks of its predecessors. Fig. 1 illustrates how the genome codification of NEAT translates to a functional ANN.

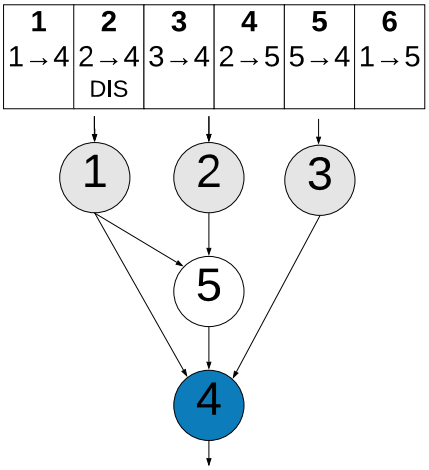


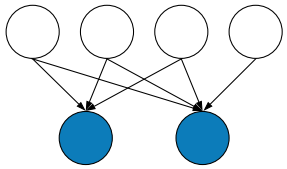
Fig. 1: Genome representation for an individual in the NEAT population. The bold number in the top line of each gene is the historical marker used to identify new structural transformations. The second line informs the link between two nodes. The third line is the disable bit (DIS) that when active means that the corresponding connection is ignored. Adapted from *Stanley and Miikkulainen* [29].

Adding new structure to an ANN without optimizing its weights and biases is usually disadvantageous to its results, making it difficult for an evolutionary algorithm to select individuals with new topologies. In contrast, to give individuals the time to adapt instead of just discarding them when they first show up, NEAT adopts speciation (or niche), and the individuals compete only within groups of similar ANNs. The individuals are divided into niches using the historical markings. For a complete description of NEAT, please refer to *Stanley and Miikkulainen* [29].

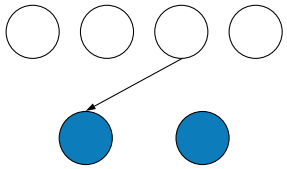
B. FS-NEAT

The evolutionary and constructive model of NEAT has been explored for the task of FS by several studies [35], [36], [37]. In this sense, one of the principal algorithms is FS-NEAT [32], that although simple has shown to have good performance in FS [38], [39], [40]. Being an extension of NEAT, FS-NEAT takes advantage of all the innovations of that method but changes the original population initialization. The minimalist start of NEAT is not as minimalist as it could be and assumes that all available inputs are useful by starting with fully connected networks.

For many datasets, however, this is not the case, and some of the inputs do not contribute to the desired behavior of the ANN. FS-NEAT addresses this problem by connecting, in each individual, one random input to one random output, instead of creating a fully connected topology, as can be seen in Fig. 2. The algorithm then behaves like regular NEAT. These minimal ANNs will most certainly lack the needed structure to have good performance, but the evolutionary algorithm will guide the complexification towards ANNs with the best set of inputs, topology, and weights. Finally, in the end, inputs not connected to an output are discarded. This way, FS-NEAT is capable of simultaneous and automatically performing FS and evolve neural networks, without requiring meta-learning, labeled data, or human expertise. By using only a subset of all the inputs, FS-NEAT is also often less costly than regular NEAT.



(a) NEAT



(b) FS-NEAT

Fig. 2: Examples of initial network topologies for (a) NEAT and (b) FS-NEAT. In regular NEAT, the initial networks have input and output layers fully connected, while in FS-NEAT, the initial population has networks with one link connecting a randomly selected input and a randomly selected output. Adapted from *Whiteson et. al* [32].

III. Proposed Methodology

We use the concept of FS-NEAT to develop a method to simultaneously solve the problems of microarray classification and gene selection and to create new network topologies that can be inspected for more insights about the data. Furthermore, FS-NEAT has the promising feature of selecting genes automatically, without the need for human set thresholds on how many genes to choose or for a filter method before the main algorithm. We start with a preprocessing step in which the microarray data is standardized according to Equation 1,

where x is a feature, and μ and σ are the mean and the standard deviation of that feature over all the samples, respectively. The labels of each sample are one-hot encoded, so for a problem with Q different classes, each class is encoded as an array of Q elements set as zero, except the element with an index corresponding to that class, that is set as one.

$$x_{new} = \frac{x - \mu}{\sigma} \tag{1}$$

As already discussed, FS-NEAT uses GA to evolve ANNs from minimalist topologies. The initial population is created without hidden nodes and connecting random input neurons to random output neurons. In this case, the input neurons are the genes expressions (after the standardization), and the output neurons are the classes. The first set of weights and biases is randomly determined from a distribution with mean equals zero, and standard deviation equals one. The outputs from the neural networks also pass through a softmax layer, described by Equation 2, that scales an array Z with length p and returns the array $\phi(Z)$ with positive elements and sum equal to one, in which e stands for the Euler's number. At the end of the evolutionary process, given a set of genes expressions, this pattern is classified as the class corresponding to the output neuron that produces the larger value. A gene is considered "selected" by the neural network when its input node has a direct or indirect (through hidden nodes) connection to one or more output nodes.

$$\phi(Z) = \frac{\mathrm{e}^{Z_i}}{\sum_{i=1}^{p} \mathrm{e}^{Z_i}} \tag{2}$$

As in most evolutionary algorithms, a cost function (or fitness) is needed to evaluate the models and guide the optimization process. A popular cost function for supervised classification tasks, the cross entropy, was chosen. Cross-entropy compares the softmax outputs from a neural network with the one-hot encoded classes that would represent the correct answer to a given set of input and averages the differences. This is the expression between curly brackets in Equation 3a, in which n is the number of samples, p is the number of outputs, y are the desired outputs, and a are the outputs from the model. Note that this expression is nonnegative. Since many microarray datasets have many imbalanced classes, there is a large risk for the model to not learn correctly how to classify the classes with fewer samples, giving more importance the larger classes. To work around this problem, we added the rest of the Equation 3a, where q is a class, n^q is the number of samples of the class q, $y_j i$ is the jth element of the ith sample of the desired output from class q, and $a_j i$ is the jth element of the ith output from the network from class q, so the cross entropy cost is computed for each class individually and is then summed, so all classes have the same contribution to the final cost, regardless of the number of samples.

Another major concern is overfitting, which happens when the model performs well on the training data, but fails to generalize and has poor performance when faced with new data. One way to avoid this problem is to expand the dataset, which regarding microarrays would mean to make new experiments, what is expensive and not always possible. A variation of this, popular with image datasets, is the addition of artificially generated data, that are often real samples slightly modified. This approach, however, is not advised when dealing with experimental data, since it would add arbitrary changes to values that should represent a real-world phenomena.

L2 regularization, also known as weight decay, is another commonly used technique to mitigate the problem of overfitting [41]. Its effect is to make the optimization prefer networks with smaller weights, what make simpler models, usually capable of better generalization. This is the term in Equation 3b, with n being the number of samples, c the number of connections, w_k the weight of connection k, and λ the regularization parameter, that must be a positive value set by the programmer. The term $\frac{1}{c}$ did not come from the canonical L2 regularization but was added since we are dealing with FS-NEAT and the number of connections is not fixed, and without it, the regularization would have an undesirable impact in the addition of new links. The cost function to be minimized by the evolutionary process is the sum of the cross-entropy cost and the L2 regularization, defined by Equation 3. Also relevant is the fact that, due to its minimalist start, FS-NEAT does not demand a component in the cost function dealing with the minimization of the number of features selected, like the one present in [31].

$$\sum_{q} \left\{ -\frac{1}{n^q} \sum_{i=1}^{n^q} \sum_{j=1}^{p} [y_{ji} \ln a_{ji} + (1 - y_{ji}) \ln(1 - a_{ji})] \right\} \tag{3a}$$

$$+ \frac{\lambda}{2n} \frac{1}{c} \sum_{k=1}^{c} w_k^2 \tag{3b}$$

Finally, it is needed to address the structure of the individual neurons of the neural networks. All hidden and output neurons added by the evolutionary algorithm follow the formula presented in Equation 4. It is a standard model for artificial neurons, where y_h is the output, m_h is the number of inputs of the neuron, w_{hj} is the weight of the input j, x_j is the input j, and b_h is the bias of the neuron h, respectively.

$$y_h = \max(0, \frac{1}{m_h} \sum_{j=1}^{m_h} w_{hj} x_j + b_h) \tag{4}$$

There are two main considerations to be made about Equation 4. The first one is that the neurons in our method use the rectified linear unit (ReLU) [42] as activation function, which has been found useful in many deep learning applications. The second is that the aggregation function is not the summation, as it is commonly used in neural networks, but the mean, hence the $\frac{1}{m_h}$ component in the formula. This choice was made to provide more stability during the learning process since, unlike a MLP or deep learning model, the number of inputs of a neuron in FS-NEAT can change over time. The

use of the mean instead of the summation causes less abrupt modifications in the output of the neuron when a connection is added, smoothing the initial impact of these transformations.

Regarding the GA that evolves the neural networks, it uses the crossover and mutation operators. The mutation can add a new node, add a new connection between nodes, and change the network weights, besides flipping the disable bit. The diversity control is obtained through speciation. Because the topology of the network is also created by the GA, FS-NEAT provides a way to inspect the existing connections between artificial neurons, allowing more direct inspection of the influence of the inputs on the outputs. In the experimental results, for instance, it is reported how certain genes had a clear preference for connections to specific classes.

IV. Experiments and Results

The algorithm described in this work was coded in Python and ran in an Intel Xeon E5-2650V4 30 MB, 4 CPUs, 2.2Ghz, 48 cores/threads, 128G, 4TB. In order to test our method, we used the data described by Yeoh *et al.* [43]. This dataset represents a microarray study of 327 bone marrow samples of pediatric patients with acute lymphoblastic leukemia (ALL). By employing an unsupervised two-dimensional hierarchical clustering algorithm the authors identified six known leukemia subtypes: (i) T-cell acute lymphoblastic leukemia (T-ALL); (ii) hyperdiploid (Hyperdip); (iii) BCR-ABL, which is a fusion of two genes, BCR and ABL, in chronic myelogenous leukemia (BCR); (iv) E2A-PBX1, which is also a fusion between two genes, normally related to adult ALL (E2A); (v) TEL-AML1, that, similarly to the previous two types, is a gene fusion, frequently found in childhood acute lymphoblastic leukemia (TEL); and (vi) Mixed-lineage leukemia (MLL). The details of the dataset are presented in Table I. This data can be found at the Cancer Program Legacy from the Broad Institute[1].

TABLE I: Detailed description of the Leukemia microarray dataset used.

Dataset	St. Jude Leukemia	
Source	[43], [44]	
Chip type	U95	
# Features	985	
# Samples	248	
# Classes	6	
Class	Name	# Samples
	BCR	15
	E2A	27
	Hyperdip	64
	MLL	20
	T-ALL	43
	TEL	79

Since the data is composed of six different classes, this is a considerably harder problem than binary classification, as it is the case of datasets divided into samples with a condition or without it. The difference in the size of each class also motivates the formulas chosen in the last Section. Following our method, the data was standardized and classified by FS-NEAT with the parameters listed in Table II. To get the accuracy of the model we used stratified 10-fold cross-validation, in which the data was divided into ten folds that preserve the total distribution of samples by class. For each iteration of the cross-validation, nine folds were used as training set, and the remaining one was used as testing set. The main advantage of cross-validation is an effectively unbiased error estimate [22]. For each iteration of the cross-validation the whole FS-NEAT evolutionary process was performed.

TABLE II: List of parameters used for the FS-NEAT evolutionary process in this experiment.

Parameter	Value
Population size	2000
Number of generations	200
Aggregation function	mean
Activation function	ReLU
λ	1.0
Probability of mutation adding connection	0.8
Probability of mutation adding node	0.15
Probability of mutation changing weight	0.05
Probability of mutation flipping disable bit	0.05

We used stratified 10-fold cross-validation to compare FS-NEAT with other three widely used classifiers for microarray data: (i) MLP with one hidden layer with five nodes, (ii) SVM with RBF kernel, and (iii) CART decision tree [45]. The accuracy of each classifier is reported in Table III, with the average and standard deviation number of features selected when applicable. FS-NEAT was close to the dedicated classifiers, SVM and MLP, and showed a better predictive power than decision tree, another algorithm capable of selecting features. All the methods had a far better result than the baseline, that would be to predict the label of the largest class (TEL) to all samples. The average number of genes selected by the neural networks created with FS-NEAT represents a reduction of more than 98% of the feature space, so the algorithm is fulfilling its function of dimensionality reduction as well.

TABLE III: Accuracy over the combination of all test sets and average number of selected features (when applicable) with standard deviation for different algorithms with stratified 10-fold cross validation.

Method	Accuracy	Selected features
Baseline	0.32	-
MLP	0.97	-
SVM	0.99	-
Decision Tree	0.83	11.30 ± 1.16
FS-NEAT	0.96	15.50 ± 2.07

The accuracy of FS-NEAT is further detailed in Table IV, a confusion matrix that discriminates the errors by class using the results from the sum of the results from each test set in the stratified 10-fold cross-validation. The diagonal shows the number of correctly classified samples for each class. As can be seen, despite the great imbalance between classes, none of them was poorly classified.

After the predictive power of the algorithm was validated, we evolved 235 artificial neural networks with FS-NEAT using

[1]http://portals.broadinstitute.org/cgi-bin/cancer/publications/view/87

TABLE IV: Confusion matrix expanding the accuracy results of FS-NEAT from Table III. Each row corresponds to the true label of the leukemia classes, and each column corresponds to the predicted labels by the evolved neural networks. The numbers in the diagonal indicate how many samples were correctly predicted by the neural networks.

True\Prediction	BCR	E2A	Hyperdip	MLL	T-ALL	TEL
BCR	12	0	2	1	0	0
E2A	0	27	0	0	0	0
Hyperdip	2	0	61	0	0	1
MLL	0	0	2	18	0	0
T-ALL	0	0	0	0	43	0
TEL	1	0	0	0	0	78

the same set of parameters as before, but this time with all available samples, to analyze the genes being selected. The need for this battery of tests is due to the stochastic nature of FS-NEAT, that may present variable results because of the randomness built into the system. An example of neural network created through this method is shown in Fig. 3.

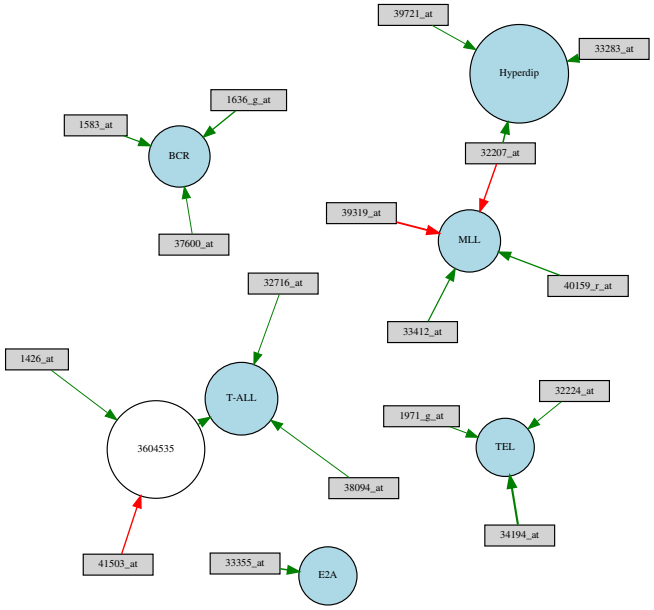


Fig. 3: Example of an ANN created with FS-NEAT using the data from the leukemia dataset. Rectangles are input nodes associated to a specific gene, white nodes are hidden nodes, and colored nodes are output nodes associated to a specific subtype of leukemia. The arrows are connections, with their thickness proportional to the absolute value of their weights.

The list of most frequently selected genes by these networks is presented in Table V. For these genes, their connection with the leukemia subtypes in the generated networks (the presence of direct or indirect connections between the corresponding inputs and outputs) was fairly strong. The most frequent genes always appeared linked to the same subtypes, reinforcing the idea that the networks are indeed encoding possible relations between gene and disease. The apparent low frequency of the genes, for instance 67 for GLUT5 in 235 networks, may be justified by the presence of redundancy and repeated genes under different alias since there are more than one probe for some genes. The sixth most selected gene, with 40 occurrences, was c-ABL, a different alias for the gene ABL. The same happens with the ninth most selected gene, with 37 occurrences, PBX1a, an alias of PBX1. The networks were able to deal with this by not selecting repeated genes, leading to this "fragmented" frequencies. Even then, the results with the alias are coherent, as both are appearing in the top ten, and ABL and c-ABL were always connected to the subtype BCR in the networks, while PBX1 and PBX1a were always connected to the subtype E2A. It is also worth noting that the probability of a gene being randomly selected by a network is $\frac{1}{n}$, n being the number of genes, so in this dataset, it would correspond to $\frac{1}{985} \approx 0.001$. Since the average number of genes by network is 15.5 as detailed in Table III, if the networks were being randomly assembled, for our test with 235 networks we would expect a frequency of 4 occurrences per gene, since $0.001 \times 15.5 \times 235 \approx 4$, far less than the frequencies listed in Table V. This indicates that the genes are indeed being selected due their capacity to better discriminate the data.

TABLE V: The top five most frequently selected genes for the leukemia dataset, with indication of which subtype they were linked to in the networks.

Frequency	Accession number	Gene	Most linked subtype
67	34362_at	GLUT5	BCR
64	33355_at	PBX1	E2A
60	40763_at	MEIS1	MLL
55	1636_g_at	ABL	BCR
47	37600_at	ECM1	BCR

The biological validation shows that the top five genes with the highest frequency among the different studied classes of leukemia were consistent with biological data. The most frequent gene linked to the BCR subtype was the Glucose Transporter-Like Protein 5 (GLUT5). Interestingly, GLUT5 was seen to be overexpressed in acute myeloid leukemia (AML) [46]. AML mice showed increased GLUT5 expression in the bone marrow, and *in vitro* AML-derived human cells also displayed higher expression of GLUT5 [46]. Moreover, consistent with biological data, the Pre-B-Cell Leukemia Transcription Factor 1 (PBX1) was the most frequently linked gene to the E2A type. The E2A-PBX1 gene fusion is frequently seen in patients with ALL, ALL of the central nervous system, and recently was also seen in gastric carcinoma [47], [48], [49]. Furthermore, the Meis Homeobox 1 (MEIS1) is the most frequently associated gene with the MLL class. In agreement with this finding, MEIS1 is commonly upregulated in MLL patients and is directly related to leukemia establishment in both human and mice, in addition to being related to acute leukemia [50], [51]. Also consistent with biological logic, the Proto-Oncogene Tyrosine-Protein Kinase ABL1 (ABL) was also present as the second most associated gene with the BCR class. ABL overexpression and its subsequent fusion with BCR is deeply related to B-cell acute lymphoblastic leukemia (B-ALL), and chronic myelogenous leukemia (CML) [52]

and its direct and indirect inhibition are linked to leukemia treatment [53], [54]. Recently, this protein expression was also related to Parkinson Disease [55]. The Extracellular Matrix Protein 1 (ECM1) was the third more frequently associated gene with the BCR class. Nevertheless, although this gene was not yet related to leukemia, its overexpression was observed in patients with papillary thyroid cancer [56], being a promising candidate for CML or B-ALL studies.

Other genes that appeared among the top ten were also consistent with biological data and showed promising results, such as the Killer Cell Lectin-Like Receptor K1 (NKG2D), which was the second most frequently linked gene with the MLL class. NKG2D overexpression and signaling are already related to MLL [57] and ALL by promoting immune system escape [58]. Finally, in agreement with the scientific literature, Endogolin (CD105), the fifth most frequently connected gene to the BCR class, is already related to both AML and CML, where its overexpression is related to AML progression [59] and CLL poor prognosis [60].

V. Conclusion

This paper described a method for classifying DNA microarrays and selecting genes from their datasets to achieve dimensionality reduction and find possible candidates for biomarkers of diseases. The method explores the FS-NEAT, an evolutionary approach that uses GA to automatically design ANNs capable of gene selection without the need for any human intervention or *a priori* knowledge. We showed how FS-NEAT could be adapted for the task of classification of multiple imbalanced classes, especially by defining the fitness function and artificial neuron structure.

This method was tested with a leukemia microarray dataset containing six subtypes of leukemia with a different number of samples. It achieved 96% accuracy in the stratified 10-fold cross validation, a result close to traditional classifiers known to have good performance with microarray data, and without compromising the classification of any individual class. Moreover, on average, the feature space was reduced by 98% without the need to predetermine the desired number of final genes or to apply other FS algorithms as a first step.

The ANNs created with FS-NEAT are interesting results by themselves since their automatically designed topology has the advantage of showing which gene was linked to which leukemia subtype. The review of the most frequently selected genes revealed consistency between these results and the biological data.

This study can be further developed by testing the method with more datasets and by biologically testing the selected genes as possible biomarkers. Experiments with larger population and number of iterations of FS-NEAT are also a possibility. As it is often the case with population-based optimization heuristics, there is a high computational cost involved, but FS-NEAT has the advantage of being easily parallelized, greatly reducing run time. The exploration of other FS algorithms and filter techniques as a preprocessing step, while not required, could also be considered in the future.

Acknowledgment

This work was partially supported by grants from FAPERGS (16/2551-0000520-6), MCT/CNPq (311022/2015-4), CAPES-STIC AMSUD (88887.135130/2017-01) - Brazil, CAPES-Alexander von Humboldt-Stiftung (AvH) (99999.000572/2016-00) - Germany, CAPES, and by Microsoft Corporation under a Microsoft Azure for Research Award.

References

[1] C. Epstein and R. Butow, "Microarray technology - enhanced versatility, persistent challenge," *Current Opinion in Biotechnology*, vol. 11, no. 1, pp. 36–41, 2000.

[2] D. Blohm and A. Guiseppi-Elie, "New developments in microarray technology," *Current Opinion in Biotechnology*, vol. 12, no. 1, pp. 41–47, 2001.

[3] G. D'Angelo, T. Di Rienzo, and V. Ojetti, "Microarray analysis in gastric cancer: a review," *World Journal of Gastroenterology*, vol. 20, no. 34, pp. 11 972–11 976, 2014.

[4] M. Blumenberg, "Skinomics: past, present and future for diagnostic microarray studies in dermatology," *Expert Review of Molecular Diagnostics*, vol. 13, no. 8, pp. 885–894, 2013.

[5] M. Kittaneh, A. Montero, and S. Glck, "Molecular profiling for breast cancer: a comprehensive review," *Biomarkers in Cancer*, vol. 5, pp. 61–70, 2013.

[6] R. Januchowski, P. Zawierucha, M. Andrzejewska, M. Ruciski, and M. Zabel, "Microarray-based detection and expression analysis of abc and slc transporters in drug-resistant ovarian cancer cell lines," *Biomedicine Pharmacotherapy*, vol. 67, no. 3, pp. 240–245, 2013.

[7] T. Aittokallio, M. Kurki, O. Nevalainen, T. Nikula, A. West, and R. Lahesmaa, "Computational strategies for analyzing data in gene expression microarray experiments," *Journal of Bioinformatics and Computational Biology*, vol. 1, no. 3, pp. 541–586, 2003.

[8] Z. Xiang, Y. Yang, X. Ma, and W. Ding, "Microarray expression profiling: analysis and applications," *Current Opinion in Drug Discovery Development*, vol. 6, no. 3, pp. 384–395, 2003.

[9] B. Karahalil, "Overview of systems biology and omics technologies," *Current Medicinal Chemistry*, vol. 23, no. 37, pp. 4221–4230, 2016.

[10] Y. F. Leung and D. Cavalieri, "Fundamentals of cdna microarray data analysis," *TRENDS in Genetics*, vol. 19, no. 11, pp. 649–659, 2003.

[11] L. E. Peterson, M. Ozen, H. Erdem, A. Amini, L. Gomez, C. C. Nelson, and M. Ittmann, "Artificial neural network analysis of dna microarray-based prostate cancer recurrence," in *Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB'05. Proceedings of the 2005 IEEE Symposium on*. IEEE, 2005, pp. 1–8.

[12] R. Díaz-Uriarte and S. A. De Andres, "Gene selection and classification of microarray data using random forest," *BMC Bioinformatics*, vol. 7, no. 1, p. 3, 2006.

[13] A. Statnikov, L. Wang, and C. F. Aliferis, "A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification," *BMC Bioinformatics*, vol. 9, no. 1, p. 319, 2008.

[14] M. Pirooznia, J. Y. Yang, M. Q. Yang, and Y. Deng, "A comparative study of different machine learning methods on microarray gene expression data," *BMC genomics*, vol. 9, no. 1, p. S13, 2008.

[15] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction." in *IWANN*, vol. 5. Springer, 2005, pp. 758–770.

[16] B. A. Garro, K. Rodríguez, and R. A. Vázquez, "Classification of dna microarrays using artificial neural networks and abc algorithm," *Applied Soft Computing*, vol. 38, pp. 548–560, 2016.

[17] J. Miao and L. Niu, "A survey on feature selection," *Procedia Computer Science*, vol. 91, pp. 919–926, 2016.

[18] E. M. Karabulut, S. A. Özel, and T. Ibrikci, "A comparative study on the effect of feature selection on classification accuracy," *Procedia Technology*, vol. 1, pp. 323–327, 2012.

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

[20] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of Bioinformatics and Computational Biology*, vol. 3, no. 02, pp. 185–205, 2005.

[21] M. Radovic, M. Ghalwash, N. Filipovic, and Z. Obradovic, "Minimum redundancy maximum relevance feature selection approach for temporal gene expression data," *BMC Bioinformatics*, vol. 18, no. 1, p. 9, 2017.

[22] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, "Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 13, no. 5, pp. 971–989, 2016.

[23] F. Martina, M. Beccuti, G. Balbo, and F. Cordero, "Peculiar genes selection: A new features selection method to improve classification performances in imbalanced data sets," *PloS One*, vol. 12, no. 8, p. e0177475, 2017.

[24] H. Lu, J. Chen, K. Yan, Q. Jin, Y. Xue, and Z. Gao, "A hybrid feature selection algorithm for gene expression data classification," *Neurocomputing*, 2017.

[25] V. Das, J. Kalita, and M. Pal, "Predictive and prognostic biomarkers in colorectal cancer: A systematic review of recent advances and challenges," *Biomedicine & Pharmacotherapy*, vol. 87, pp. 8–19, 2017.

[26] G. B. Whitworth, "An introduction to microarray data analysis and visualization," *Methods in enzymology*, vol. 470, pp. 19–50, 2010.

[27] M. Sipper, R. S. Olson, and J. H. Moore, "Evolutionary computation: the next major transition of artificial intelligence?" p. 26, 2017.

[28] S. Ding, H. Li, C. Su, J. Yu, and F. Jin, "Evolutionary artificial neural networks: a review," *Artificial Intelligence Review*, pp. 1–10, 2013.

[29] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

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

[31] R. Luque-Baena, D. Urda, J. Subirats, L. Franco, and J. Jerez, "Analysis of cancer microarray data using constructive neural networks and genetic algorithms," in *Proceedings of the IWBBIO, international work-conference on bioinformatics and biomedical engineering*, 2013, pp. 55–63.

[32] S. Whiteson, P. Stone, K. O. Stanley, R. Miikkulainen, and N. Kohl, "Automatic feature selection in neuroevolution," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM, 2005, pp. 1225–1232.

[33] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 1998, pp. 9–50.

[34] B. Grisci and M. Dorn, "Neat-flex: Predicting the conformational flexibility of amino acids using neuroevolution of augmenting topologies," *Journal of Bioinformatics and Computational Biology*, p. 1750009, 2017.

[35] S. Sohangir, S. Rahimi, and B. Gupta, "Neuroevolutionary feature selection using neat," *Journal of Software Engineering and Applications*, vol. 7, no. 07, p. 562, 2014.

[36] ——, "Optimized feature selection using neuroevolution of augmenting topologies (neat)," in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*. IEEE, 2013, pp. 80–85.

[37] M. Tan, M. Hartley, M. Bister, and R. Deklerck, "Automated feature selection in neuroevolution," *Evolutionary Intelligence*, vol. 1, no. 4, pp. 271–292, 2009.

[38] E. Papavasileiou and B. Jansen, "An investigation of topological choices in fs-neat and fd-neat on xor-based problems of increased complexity," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 2017, pp. 1431–1434.

[39] ——, "A comparison between fs-neat and fd-neat and an investigation of different initial topologies for a classification task with irrelevant features," in *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*. IEEE, 2016, pp. 1–8.

[40] A. Ethembabaoglu, S. Whiteson *et al.*, "Automatic feature selection using fs-neat," *IAS technical report IAS-UVA-08-02*, 2008.

[41] A. Y. Ng, "Feature selection, l 1 vs. l 2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 78.

[42] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," *arXiv preprint arXiv:1611.01491*, 2016.

[43] E.-J. Yeoh, M. E. Ross, S. A. Shurtleff, W. K. Williams, D. Patel, R. Mahfouz, F. G. Behm, S. C. Raimondi, M. V. Relling, A. Patel *et al.*, "Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling," *Cancer cell*, vol. 1, no. 2, pp. 133–143, 2002.

[44] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data," *Machine learning*, vol. 52, no. 1, pp. 91–118, 2003.

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[46] W. Chen, Y. Wang, A. Zhao, L. Xia, G. Xie, M. Su, L. Zhao, J. Liu, C. Qu, R. Wei, C. Rajani, Y. Ni, Z. Cheng, Z. Chen, S. Chen, and W. Jia, "Enhanced fructose utilization mediated by slc2a5 is a unique metabolic feature of acute myeloid leukemia with therapeutic potential," *Cancer Cell*, vol. 30, no. 5, pp. 779–791, 2016.

[47] J. Duque-Afonso, C. Lin, K. Han, M. Wei, J. Feng, J. Kurzer, C. Schneidawind, S. Wong, M. Bassik, and M. Cleary, "E2a-pbx1 remodels oncogenic signaling networks in b-cell precursor acute lymphoid leukemia," *Cancer Research*, vol. 76, no. 23, pp. 6937–6949, 2016.

[48] A. Alsadeq and D. Schewe, "Acute lymphoblastic leukemia of the central nervous system: on the role of pbx1," *Haematologica*, vol. 102, no. 4, pp. 611–613, 2017.

[49] C. He, Z. Wang, L. Zhang, L. Yang, J. Li, X. Chen, J. Zhang, Q. Chang, Y. Yu, B. Liu, and Z. Zhu, "A hydrophobic residue in the tale homeodomain of pbx1 promotes epithelial-to-mesenchymal transition of gastric carcinoma," *OncoTargets and Therapy*, vol. 8, no. 29, 2017.

[50] J. Roychoudhury, J. Clark, G. Gracia-Maldonado, Z. Unnisa, M. Wunderlich, K. Link, N. Dasgupta, B. Aronow, G. Huang, J. Mulloy, and A. Kumar, "Meis1 regulates an hlf-oxidative stress axis in mll-fusion gene leukemia," *Blood*, vol. 125, no. 16, pp. 2544–2552, 2015.

[51] Q. Wang, Y. Li, J. Dong, B. Li, J. Kaberlein, L. Zhang, F. Arimura, R. Luo, J. Ni, F. He, J. Wu, R. Mattison, J. Zhou, C. Wang, S. Prabhakar, M. Nobrega, and M. Thirman, "Regulation of meis1 by distal enhancer elements in acute leukemia," *Leukemia*, vol. 28, no. 1, 2014.

[52] S. Reckel and O. Hantschel, "Bcr-abl: one kinase, two isoforms, two diseases," *OncoTargets and Therapy*, vol. 8, no. 45, 2017.

[53] Z. Tan, A. Peng, J. Xu, and M. Ouyang, "Propofol enhances bcr-abl tkis' inhibitory effects in chronic myeloid leukemia through akt/mtor suppression," *BMC Anesthesiology*, vol. 17, no. 1, p. 132, 2017.

[54] Y. Sun, N. Zhao, H. Wang, Q. Wu, Y. Han, Q. Liu, M. Wu, Y. Liu, F. Kong, H. Wang, Y. Sun, D. Sun, L. Jing, G. Tang, Y. Hu, D. Xiao, H. Luo, Y. Han, and Y. Peng, "Ct-721, a potent bcr-abl inhibitor, exhibits excellent in vitro and in vivo efficacy in the treatment of chronic myeloid leukemia," *Journal of Cancer*, vol. 8, no. 14, pp. 2774–2784, 2017.

[55] S. Brahmachari, S. Karuppagounder, P. Ge, S. Lee, V. Dawson, T. Dawson, and H. Ko, "c-abl and parkinson's disease: Mechanisms and therapeutic potential," *Journal of Parkinson's Disease*, vol. 7, 2017.

[56] M. Vriens, W. Moses, J. Weng, M. Peng, A. Griffin, A. Bleyer, B. Pollock, D. Indelicato, J. Hwang, and E. Kebebew, "Clinical and molecular features of papillary thyroid cancer in adolescents and young adults," *Cancer*, vol. 117, no. 2, pp. 259–267, 2011.

[57] B. Poppe, J. Vandesompele, C. Schoch, C. Lindvall, K. Mrozek, C. Bloomfield, H. Beverloo, L. Michaux, N. Dastugue, C. Herens, N. Yigit, A. De Paepe, A. Hagemeijer, and F. Speleman, "Expression analyses identify mll as a prominent target of 11q23 amplification and support an etiologic role for mll gain of function in myeloid malignancies," *Blood*, vol. 103, no. 1, pp. 229–235, 2004.

[58] M. Tang, D. Acheampong, Y. Wang, W. Xie, M. Wang, and J. Zhang, "Tumoral nkg2d alters cell cycle of acute myeloid leukemic cells and reduces nk cell-mediated immune surveillance," *Immunologic Research*, vol. 64, no. 3, pp. 754–764, 2016.

[59] Z. Chakhachiro, Z. Zuo, T. Aladily, H. Kantarjian, J. Cortes, K. Alayed, M. Nguyen, L. Medeiros, and C. Bueso-Ramos, "Cd105 (endoglin) is highly overexpressed in a subset of cases of acute myeloid leukemias," *American Journal of Clinical Pathology*, vol. 140, no. 3, 2013.

[60] F. Vrbacky, J. Nekvindova, V. Rezacova, M. Simkovic, M. Motyckova, D. Belada, U. Painuly, Z. Jiruchova, J. Maly, J. Krejsek, P. Zak, M. Cervinka, and L. Smolej, "Prognostic relevance of angiopoietin-2, fibroblast growth factor-2 and endoglin mrna expressions in chronic lymphocytic leukemia," *Neoplasma*, vol. 61, no. 5, pp. 585–592, 2014.