A deep learning and genetic algorithm based feature selection processes on Leukemia Data.

Rita Francese, Maria Frasca, Michele Risi and Genoveffa Tortora
Department of Computer Science
University of Salerno, Italy
{francese, mfrasca, mrisi, tortora}@unisa.it

Abstract—Acute Leukemia is classified in terms of two distinct classes: Acute Lymphoblastic Leukemia (ALL) and Acute Myeloid Leukemia (AML). This paper aims at defining a feature selection analysis process mainly based on Deep Learning for classifying the acute leukemia type. The considered dataset consists in data of patients affected by both the leukemia types. Both the leukemia types are characterized by a list of identical genes for all the patients. The analysis exploits feature selection techniques for reducing the consistent number of variables (genes). To this aim, we use linear models for differential expression for microarray data, and an autoencoder based unsupervised deep learning model to simplify and speed up the classification. Then, classification models have been implemented with the use of a deep neural network (DNN), obtaining an accuracy of approximately 92%. Moreover, the results have been compared with the ones provided by an approach based on support vector machines (SVM), giving an accuracy of 87.39%. Another feature selection approach based on genetic algorithms has been experimented, with worse performances. We also conducted a gene enrichment analysis based on the functional annotation of the differentially expressed genes. As a result, a differentially expressed pathway between the two pathologies has been detected.

Index Terms—Feature selection, Neural Network, Genomic data, Leukemia, Pathways

I. INTRODUCTION

In the last years, different Machine Learning and Deep Learning methodologies have been successfully proposed to detect patterns from bioinformatics big data [13], [16]. Among the human tumors, Leukemia is one of the most relevant: in 2020 it has been the cause of worldwide death for 311,594 people, while the new leukemia cases have been 474,5191. For leukemias we mean a heterogeneous group of neoplastic diseases, which foresee any process of proliferative alteration of a progressive and irreversible nature of the bone cells of the bone marrow. Leukemias originate from the malignant transformation of hematopoietic stem progenitor cells, with alteration of the proliferation and differentiation of the same cells. In this paper, we define a process aiming at detecting a set of differentially expressed genes in terms of methylation level, i.e., genes that in different conditions have an expression level significantly different in the AML and ALL cases, and their characteristic pathways. The detection of gene expression data samples involves feature selection and classification.

1https://gco.iarc.fr/today/data/factsheets/cancers/36-Leukaemia-factsheet.pdf

In particular, for feature selection we apply two different approaches: i) the first one, consisting of two steps is carried out by using statistical and artificial intelligence techniques; ii) the second one is based on the use of a genetic algorithm (GA). For the classification, two different approaches have experimented: a deep neural network (DNN) and support vector machines (SVM). Finally, we conduct a pathway analysis on the reduced dataset to identify therapeutic targets of leukemia.

The paper is structured as follows: Section II outlines related works, and Section III describes the leukemia dataset. Section IV presents the analysis process proposed based on the Bayesian method and autoencoders for feature selection, whilst DNN and SVM for the classification. Section V shows the analysis process when feature selection is based on a genetic algorithm, while Section VI discusses the results and their implication in therm of pathway analysis. Finally, Section VII concludes the paper.

II. RELATED WORK

Recently, different approaches have been developed to perform gene selection on a genetic dataset.

Saini et al. [20] proposed a gene masking derived from the genetic algorithm. They identified an optimal gene mask is searched providing the largest performance gain by removing the largest number of features for the chosen classification algorithm.

Lv et al. [15] applied a multi-objective model following the analytic hierarchy process that gives more importance to the detection accuracy than the feature size to build a model such as the multi-objective optimization algorithm (MOEDA). This solution is based on a type of distribution estimation algorithm (EDA) that guides the search for the optimum by building and sampling explicit probabilistic models.

Othman et al. [17] proposed and developed multi-objective hybrid cuckoo research with evolutionary operators for gene selection. The evolutionary operators used are double mutation and simple crossover operators. The results of the experiments revealed that the developed algorithm, multi-objective cuckoo search with evolutionary operators, outperformed the cuckoo and multi-objective search algorithms with less significant selected genes.

Cahyaningrum et al. [6] proposed a technique based on Principal Components Analysis (PCA) to select the most relevant features. Moreover, they proposed the use of ANN
and e GA Hybrid Intelligence (GAHI) for cancer detection. Although ANN is recognized as one of the methods to classify microarray data, GA is used in this case to optimize the ANN architecture. Also Wu et al. [23] proposed an ANN classifier. To initialize the structure, an algorithm was used to choose input variables on layered links and different activation functions for different nodes. Then, a hybrid method integrating GA and particle swarm optimization (PSO) algorithms were used to identify an optimal structure with the parameters encoded in the classifier.

III. DATASET

We extracted the dataset from the GEO platform public database containing genomic data, available on the NCBI website2. It consists of biomedical data of 556 patients, where 233 were affected by AML and 323 by ALL. Specifically, for each patient, the dataset contains the detected CpG probes described by their methylation value. These data are related to the Infinium Human Methylation 450k BeadChip microarray, a popular technology to explore DNA methylomes [10].

IV. THE ANALYSIS PROCESS BASED ON BAYESIAN AND AUTOENCODERS TECHNIQUES

To perform feature selection we applied two different approaches: (i) Bayesian and autoencoders and (ii) GA. In this section, we describe the former that adopts statistical and artificial intelligence techniques. Figure 1 depicts this analysis process consisting of the following six steps.

A. Step 1 - Data Cleaning

The considered samples were extracted from the GEO database and were related to the Illumina Human 450k microarray; they are methylation data related to the GPL13534 platform series. Our initial dataset consisted of 556 samples coming from several microarray experiments. The number of probes among the samples is different; to make the samples uniform and to be able to analyze them we standardized the number of probes among the samples by difference, obtaining 451,308 CpG probes per sample. Then, we performed on our dataset a preprocessing phase by removing the cross-reactive probes, the SNPs probes, and the probes related to our dataset a preprocessing phase by removing the cross-reactive probes, the SNPs probes, and the probes related to 451,308 CpG probes per sample. Then, we performed on our dataset a preprocessing phase by removing the cross-reactive probes, the SNPs probes, and the probes related to the Illumina Human Methylation 450k BeadChip microarray, a popular technology to explore DNA methylomes [10].

413

2https://www.ncbi.nlm.nih.gov/gds

because we will focus our analysis only on autosomal genes (not related to sex); this is because there is an imbalance of methylation on sex chromosomes. At this point, we obtained a dataset composed of 556 samples and 434,917 CpG probes.

On these values we performed a gene sets enrichment analysis operation to obtain the related genes: the resulting dataset was composed of 556 samples and 19,340 genes. A further cleanup eliminated the missing values, as they could generate errors in the measurement and understanding of the relationships between the variables, reducing the genes to 17,996.

B. Step 2 - Batch Effect Removal

The batch effect is a source of variability that has been added to the samples during manipulation, consisting in the introduction of non-biological variability in an experiment [12].

In this step, the batch effect is identified and removed. First, we identified and evaluated the Genomic Spatial Event (GSE) batch variables, i.e., the type of experiment to which each sample refers, through the correlation between the variables. Batch GSEs were identified by statistical analysis of batch medians. This method compares the distribution of each GSE in a single lot to its distribution in all the other lots using the Kolmogorov-Smirnov (KS) non-parametric test that verifies the form of the distributions [1]. We deleted the batch effect on the GSE by using an empirical Bayesian framework and then we validated the results: no further effect was detected.

C. Step 3 - Normalization

The sample data have been normalized to remove systematic variation in a microarray experiment that affects the measured gene expression levels. One of the objectives of DNA microarray analysis is to compare the levels of gene expression in two or more pathological conditions to identify their peculiarities. For our dataset, we have adopted quantile normalization, whose aim is to make equal the empirical intensity distributions of all arrays.

The quantile normalization transforms the intensity distributions of each specific array. In particular, it assigns to each intensity the same value to the quantile to which it belongs. Thus, each intensity has the same distribution in all arrays.

The quantile normalization transforms the intensity distributions of each specific array. In particular, it assigns to each intensity the same value to the quantile to which it belongs. Thus, each intensity has the same distribution in all arrays. This method is based on the consideration that a quantile-quantile graph is a line perfectly coinciding with the diagonal if and only if the distribution of the two data vectors is the same [5].

This means that it is possible to give all arrays the same distribution by replacing the values of the original dataset with the average quantile by applying the following normalization algorithm.

Let \( M \) be a matrix of \( ng \) genes (rows) and \( n \) arrays (columns) representing the number of patients:

1. Sort each column of \( M \), obtaining \( M_{\text{sort}} \);
2. to each element of the \( k_{th} \) row in \( X_{\text{sort}} \) assign the average value of that row, obtaining \( M'_{\text{sort}} \);
3. calculate \( M_{\text{normalized}} \) by reordering each column of \( M'_{\text{sort}} \) according to the original order.
A negative aspect of this method is that it forces the quantile values to be all the same. This could be a problem in the distribution queues, where it is possible for a gene to have the same value on all arrays, even if this situation rarely occurs [5].

D. Step 4 - Bayesian Feature Selection

In this step, we identified the differentially expressed genes between the two pathological classes AML and ALL by using a Bayesian feature selection technique. Bayesian methods are suitable to study multidimensional inference problems, so it naturally applies to microarray data. Unlike the methods that apply classical inference separately for each gene, the Bayesian analysis exploits information sharing between the genes. We adopted Limma (Linear Models for MicroArray data)\(^3\) to identify these differentially expressed genes through the use of the empirical Bayesian method. In the following, we describe the adopted feature selection procedure.

Two matrices are obtained from the dataset: the design matrix, containing the samples in the array, and the contrast matrix, which specifies the comparisons to be performed on the samples. The design matrix is specified as follows:

1. the rows represent samples;
2. the columns represent groups. In our case, there exist two columns representing ALL and AML samples, respectively;
3. for each sample, the column corresponding to its group has a coefficient equal to 1, otherwise 0.

The contrast matrix represents the difference between the columns.

A first fitting function is applied to the design matrix and data from an experiment involving a series of microarrays with the same set of probes. The function adopts multiple linear models for weighted or generalized minimum squares. Thus, the linear model is adapted to the expression data (by gene) for each probe. A second fitting function is applied to the output of the first regression and the contrast matrix. Given the linear model previously computed, the fitting function calculates the estimated coefficients and standard errors for a given set of contrasts. The function re-orientates the adapted model from the original matrix design to any set of contrasts of the original coefficients. We then applied an empirical Bayesian model for linear data regression, which dynamically borrows information between genes. This function is used to classify genes in order of evidence based on their differential expression; the fact that the same linear model is adapted to each gene allows us to borrow the relationships between genes to moderate residual variances. We get an estimate of the prior distribution from Bayes’ empirical moderation, resulting in an increase in statistical power to detect differential expression [19].

A table of top-ranked genes from the adapted linear model is extracted. This table contains various summary statistics for the top-ranked genes and the selected contrast. In particular, it contains the variables \(\text{logFC}\) and \(\text{adj.P.Val}\), adopted to perform the feature selection. \(\text{logFC}\) provides the value of the contrast; usually, this represents a change of \(2^{\text{logFC}}\) times between two or more experimental conditions, While \(\text{adj.P.Val}\) is the distribution of \(p\)-values adjusted by Benjamini-Hochberg correction [3], which introduced FDR, i.e., the expected proportion of the number of false-positive results on the total of all the positive results, and represents the number of null hypotheses wrongly refused on the total of those rejected. For this reason, we initially extracted the genes with a value of \(\text{adj.P.Val} < 0.01\) and with a \(|\text{logFC}| > 2\), thus obtaining 1,118 genes for 556 samples.

E. Step 5 - Feature Selection with Autoencoders

To further reduce our features we used autoencoders. An autoencoder is an unsupervised artificial feed-forward neural network. Conceptually, it is similar to PCA and can be used to reduce the dimensional space.

The autoencoders compress the input data by forcing the network to use a low-dimensional representation of data. They can to reconstruct the original input. An autoencoder consists of 3 layers: an input layer, a hidden layer and an output layer. The number \(n\) of input nodes of this type of network is equal to the number of output nodes (the number of genes). In our case, the hidden layer is composed of two nodes. An autoencoder is divided into two parts: the encoder that learns the mapping between the unlabeled high-dimensional \(I_{[1:n]}\) input data and the low-dimensional representations (in the bottleneck layer), and the decoder that learns the mapping from the

\(^3\)https://bioconductor.org/packages/release/bioc/html/limma.html
intermediate layer representation to the output reconstructed in a high dimension \( O[1 : n] \). Autoencoder-based approaches learn to reconstruct input samples by optimizing the Root Mean Squared Error (RMSE) objective function [18].

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (I[i] - O[i])^2}{n}} \tag{1}
\]

For autoencoders, we have chosen the batch size equal to the number of genes in our dataset (1,118 genes), Adam’s optimization type, an algorithm for the optimization of the gradient of the first order of the stochastic objective functions, based on adaptive estimates of moments of lower order [14]. As activation functions we used "tanh", "linear", "tanh", respectively for the three layers. At the end of the process we obtained 28 abnormal genes that we removed, obtaining a final dataset composed of 1,090 genes.

**F. Step 6 - Classification**

For the classification, we experimented two different techniques: a deep neural network (DNN) and support vector machines (SVM), as described in the following.

1) DNN: We adopted the DNN for classification. The size of the input is equal to the number of genes. The input layer is composed of 30 units with the "relu" activation function. The 4 hidden layers transform the representation of the previous layer into a more abstract form. They are composed of 22, 15, 9, 5 respectively, with "relu" activation function. The final classification is performed by the last layer (the output layer), composed of 2 units with the "softmax" activation function. The final classification is performed by the last layer (the output layer), composed of 2 units with the "softmax" activation function. To classify the class of patients (i.e., AML or ALL) we adopted the "categorical cross entropy" loss function. To set up the hyperparameters we selected the "adam" optimizer. We adopted the k-fold cross-validation, with \( k = 5 \).

For the test set we achieved 0.2499 of loss and 91.89% of accuracy.

2) SVM: We also experimented the use of a support vector machine (SVM) [11] as a classifier. SVM are supervised learning models and a powerful technique for classification and regression, with associated learning algorithms. Given a set of training examples, each one belonging to one of two different categories, a training algorithm creates a model that assigns new examples to one or the other category; the model is then used to make predictions for a set of test examples. We got an accuracy of 87.39% by using the same training and test sets adopted with the DNN.

V. THE ANALYSIS PROCESS BASED ON A GENETIC ALGORITHM

In this section, we present the second feature selection process we experimented with, based on a genetic algorithm (GA), widely used for this purpose in the literature (e.g., [2], [21]). In particular, we followed the analysis process shown in Fig. 3 that differs from the one in Fig. 1 for the red step. Therefore, in the following, we describe only the Feature Selection Genetic Algorithm step and the final classification results.

Our dataset consists of 16,408 individuals (i.e., genes). We adopted a GA for feature selection starting from a binary array that represents a chromosome, where genes are the array elements. A chromosome is generally encoded with a bit or character vector. In our case, each element is set to 1 if the biological gene is not expressed (i.e., its value is less equal than 0.5), 0 otherwise. The population is a set of solutions (chromosomes) related to the considered problem. The adopted GA, instantiated with the parameters in Table I.

The first step of this algorithm creates the initial population (i.e., 100 individuals) randomly setting the binary values of genes, while the next phases are repeated with each generation and are associated with the principle of natural selection or genetics. The selection of the best individuals is performed by combining or modifying the characteristics that identify an individual. From genetics, the new chromosomes are obtained by recombing their genetic heritage or by changing the genes with the mutation and crossover operators. For each combination of genes, it is possible to calculate a value called fitness which indicates the ability with which the chromosome or solution can solve the problem. Concerning the third step (selection), which favors the selection of better individuals of a population to influence the next generation, we adopted a tournament mechanism. A small number of individuals (i.e., tournament size = 3) is choose randomly with replacement. We keep the fittest one. This is done again and again until you have got 100 individuals. The fitness function resolves an optimization problem by maximizing the cross-validation accuracy score with the minimum number of selected genes. The score is the accuracy of training data using only the values of the selected genes. In particular, we divided every dataset into 5 equal parts to calculate the fitness value. Then, we selected one of the mentioned parts as a test set and the rest as a training set. We repeated this action five times for every separate part. In the Crossover phase, the generation of offspring occurs starting from the parents previously selected in the selection phase. The Crossover operator randomly selects a pair of individuals from the pool of solutions for reproduction, with crossover probability (i.e., 0.8); the values of the two solutions are exchanged to generate two new solutions (i.e., the offspring). Crossover aims to generate two new solutions starting from the combination of two previous ones.

The Mutation phase creates a new population starting from the solutions identified in the previous step. Mutation aims to prevent the locking up at a local minimum and to explore the

| Parameter                  | Value |
|----------------------------|-------|
| Population size            | 100   |
| Number of generations      | 100   |
| Fold for cross validation  | 5     |
| Crossover probability      | 0.8   |
| Probability of mutation    | 0.1   |
| Independent probability of crossover | 0.8 |
| Independent probability of mutation | 0.08 |
| Tournament size            | 3     |
Fig. 2. The DNN architecture, where \( n \) represents the number of genes.

Fig. 3. The data analysis process based on Genetic Algorithm feature selection.

| TABLE II | COMPARISON WITH THE ADOPTED FEATURES SELECTION APPROACHES. |
|----------------|-----------------------------|-----------------------------|
| Dataset (rows x columns) | Approach | Obtained dataset (rows x columns) |
| 2*17,996 x 556 | Feature Selection Limma + autoencoders | 2*17,090 x 556 |
| 2*17,996 x 556 | Feature Selection GA | 2*770 x 556 |
| Classifier | DNN | SVM | DNN | SVM |
| Result accuracy | 91.86% | 87.39% | 60.36% | 80.63% |

entire research space when each individual in the population reaches a level of fitness close to the average. This operator randomly flips (i.e., one becomes zero or vice-versa) some elements of the offspring. Like crossover, there is a mutation probability (set to 0.1). If a randomly selected floating-point value is less than the mutation probability, the mutation is performed on the offspring; otherwise, no mutation occurs. The mutation is performed by randomly selecting (with an independent mutation probability set to 0.08) a gene in the offspring’s chromosome and generating a new value uncorrelated to the previous one.

The GA algorithms repeat the process from step 2 until the number of generations (i.e., 100) is reached.

As output, we tried to obtain a better set of individuals that, with the advancement of generations, contains the subsets of genes involved in both AML and ALL. In particular, we selected the genes set to 1. We obtained a reduced dataset containing 6,240 features (genes).

For the classification, we used the DNN and the SVM as we did previously for the first process. By applying the DNN to the GA results we obtained 60.36% in accuracy and loss of 0.671. While applying the SVM on the same data we obtained an accuracy of 19.96%.

VI. RESULTS AND IMPLICATIONS

In this section, we compare the results of the processes described in Section IV, examine the relevance of the selected genes by applying the pathway analysis, and further validate the results.

A. Comparison

The obtained classification results are summarized in Table II, where we reported for each adopted feature selection process (Limma and the autoencoders vs. GA), the number of selected genes and the classification results obtained with both DNN and SVM classifiers. Results show that feature selection using Limma and the autoencoders performs better with both the classifiers, but the deep neural network (DNN) reached higher accuracy (91.86%) than SVM (87.39%).

B. Applying pathways analysis

Pathways analysis provides a means to map key biological processes into important clinical features in disease. It is mainly adopted for predicting cancer outcomes through genome-wide characterizations. In this section, we describe the pathway analysis conducted on the results of the feature selection process based on Limma and autoencoders that reached the best accuracy results and reduced the gene numbers from 19,340 to 1,090. For confirming the relevance at biological level of these results we performed the Gene Sets Enrichment Analysis (GSEA) for the interpretation of gene expression data, which highlights groups of genes that share biological functions (i.e., pathway), chromosomal position or common regulation. In this analysis, we referred to the Kyoto Encyclopedia of Genomes (KEGG)\(^4\), one of the most used databases for pathway knowledge. The procedure\(^5\) takes a character vector of significant CpG sites, maps the CpG sites to Entrez Gene IDs to test for GO or KEGG pathway enriched using a hypergeometric test, taking into account the number of CpG sites for gene on EPIC array. In particular, statistical approaches to identify significantly overexpressed CpG groups, by examining p-value and FDR are used. Finally,

\(^4\)https://www.genome.jp/kegg

\(^5\)https://bioconductor.org/packages/release/bioc/html/missMethyl.html
we have extracted only the pathways with $FDR < 0.05$ [4].

As a result, the analysis detected the "RNA degradation" pathway (see Fig. 4) from 20 genes of the 1,090 differentially methylated genes individuated by the feature selection process based on autoencoders.

RNA degradation in eukaryotic cells plays a very important role in gene expression, as it balances the transcription rate and also serves to rapidly eliminate transcripts that are no longer needed. Furthermore, RNA degradation plays a controlling role by eliminating RNA molecules that are considered non-functional or abnormal if they lack sequences or characteristic changes necessary for their functions.

The detected pathway denotes the deregulation of transcription, which is an important factor in the development of leukemia. In particular, in T-cell acute lymphoblastic leukemia (T-ALL) it identifies mutations in the RNA decay factors, including mutations in the CNOT3 gene, which is part of the CCR4-NOT complex that regulates gene expression transcriptionally and post-transcriptionally [8]. This gene is included in the 1,090 we detected and that seems to be involved in mRNA deadenylation. When errors occur in this process, there are quality control mechanisms that detect and eliminate defective transcripts that can lead to dysfunctional or toxic protein. However, these mechanisms do not only ensure the fidelity of RNA transcripts but also perform important regulatory tasks by allowing rapid modulation of steady-state RNA levels in response to changes in the intracellular or extracellular environment [22]. However, it remains unclear how mutations in RNA processing may contribute to the development of leukemia [9].

VII. Conclusion

In this paper, we presented a process to detect a set of differentially expressed genes in the two acute form of leukemia: ALL and AML. To this aim, we adopted feature selection techniques and classifier methods. The analysis has been performed on a dataset consisting of samples from people suffering from AML or leukemia of different forms: ALL and AML. We also experimented a feature selection algorithm to compute gene centric methylation from probe level methylation data. Plos One, 11(2):1–19, 2016.

REFERENCES

[1] R. Akulenko, M. Merl, and V. Helms. BEclear: batch effect detection and adjustment in DNA methylation data. PloS One, 11(8):1–17, 2016.

[2] B. Baur and S. Bozdag. A feature selection algorithm to compute gene centric methylation from probe level methylation data. PloS One, 11(2):1–19, 2016.

[3] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society: series B (Methodological), 57(1):289–300, 1995.

[4] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. Annals of Statistics, pages 1165–1188, 2001.

[5] B. M. Bolstad, R. A. Irizarry, M. Åstrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics, 19(2):185–193, 2003.

[6] K. Cahan et al. Microarray gene expression classification for cancer detection using artificial neural networks and genetic algorithm hybrid intelligence. In Proc. of the International Conference on Data Science and Its Applications (ICoDSA), pages 1–7. IEEE, 2020.

[7] Y.-a. Chen, M. Lemire, S. Choufani, D. T. Butcher, D. Grafodatskaya, B. W. Zanke, S. Gallinger, T. J. Hudson, and R. Welsberg. Discovery of cross-reactive probes and polymorphic CpGs in the Illumina Infinium HumanMethylation450 microarray. Epigenetics, 8(2):203–209, 2013.

[8] M. A. Collart and O. O. Panasenko. The Ccr4–not complex. Gene, 492(1):42–53, 2012.

[9] J. Cools. ART: aberrant RNA degradation in T-cell leukemia. https://cordis.europa.eu/project/rcn/185655/factsheet, 2014–2019. [Online].

[10] S. Dedeurwaerder, M. Defrance, M. Bizet, E. Calonne, G. Bontempi, and F. Fuks. A comprehensive overview of Infinium HumanMethylation450 data processing. Briefings in Bioinformatics, 15(6):929–941, 2014.

[11] C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al. A practical guide to support vector classification, 2003.

[12] W. E. Johnson, C. Li, and A. Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. Biostatistics, 8(1):118–127, 2007.

[13] H. Kashyap, H. A. Ahmed, N. Hoque, S. Roy, and D. K. Bhattacharyya. Big data analytics in bioinformatics: architectures, techniques, tools and issues. Network Modeling Analysis in Health Informatics and Bioinformatics, 5(1):1–28, 2016.

[14] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[15] J. Lv, Q. Peng, X. Chen, and Z. Sun. A multi-objective heuristic algorithm for gene expression microarray data classification. Expert Systems with Applications, 59:13–19, 2016.

[16] S. Min, B. Lee, and S. Yoon. Deep learning in bioinformatics. Briefings in Bioinformatics, 18(5):851–869, 2017.

[17] M. S. Othman, S. R. Kumaran, and L. M. Yusuf. Gene selection using hybrid multi-objective Cuckoo search algorithm with evolutionary operators for cancer microarray data. IEEE Access, 8:186348–186361, 2020.

[18] A. Pumsirirat and L. Yan. Credit card fraud detection using deep learning based on auto.encoder and restricted boltzmann machine. International Journal of Advanced Computer Science and Applications, 9(1):18–25, 2018.

[19] E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth. Limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Research, 43(7):e47–e47, 2015.

[20] H. Saint, S. P. Lal, V. V. Naidu, V. W. Pickering, G. Singh, T. Tsunoda, and A. Sharma. Gene masking—a technique to improve accuracy for cancer classification with high dimensionality in microarray data. BMC Medical Genomics, 9(3):261–269, 2016.

[21] S. Sayed, M. Nassef, A. Badr, and I. Farag. A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets. Expert Systems with Applications, 121:233–243, 2019.

[22] K. Weskamp and S. J. Barnada. RNA degradation in neurodegenerative disease. RNA Metabolism in Neurodegenerative Diseases, pages 103–142, 2018.

[23] P. Wu and D. Wang. Classification of a DNA microarray for diagnosing cancer using a complex network based method. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 16(3):801–808, 2018.