UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

BRUNO IOCHINS GRISCI

N3O: A NEAT expansion for improving classification and feature selection applied to microarray data

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Advisor: Prof. Dr. Márcio Dorn
Coadvisor: Prof. Dr. Mario Inostroza-Ponta

Porto Alegre
July 2018

CIP — CATALOGING-IN-PUBLICATION

Iochins Grisci, Bruno

N3O: A NEAT expansion for improving classification and feature selection applied to microarray data / Bruno Iochins Grisci. – Porto Alegre: CIC da UFRGS, 2018.

148 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Curso de Ciência da Computação, Porto Alegre, BR–RS, 2018. Advisor: Márcio Dorn; Coadvisor: Mario Inostroza-Ponta.

1. Machine learning, neuroevolution, feature selection, classification, supervised learning, NEAT, microarray, gene expression, gene selection. I. Dorn, Márcio. II. Inostroza-Ponta, Mario. III. Título.

“The intentions of a tool are what it does.
A hammer intends to strike,
a vise intends to hold fast, a lever intends to lift.
They are what it is made for.
But sometimes a tool may have other uses that you don’t know.
Sometimes in doing what you intend,
you also do what the knife intends, without knowing.”’
— PHILIP PULLMAN, HIS DARK MATERIALS

ACKNOWLEDGEMENTS

This work was 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, Alexander von Humboldt-Stiftung (AvH) [BRA 1190826 HFSTCAPES-P] - Germany, CAPES, and by Microsoft Corporation under a Microsoft Azure for Research Award. Asociación de Universidades Grupo Montevideo (AUGM), Pró-Reitoria de Pós-Graduação - UFRGS (PROPG), Pró-Reitoria de Pesquisa - UFRGS (Propesq) and Universidad de Santiago de Chile (USACH) also provided funds for activities related to this project.

I would like to thank the examining board, Dr. Adriano Velasque Werhli from Centro de Ciências Computacionais (C3) at Universidade Federal do Rio Grande (FURG), Dr. Guido Lenz from Centro de Biotecnologia (Cbiot) at UFRGS, and Dr. André Grahl Pereira from INF at UFRGS, for the granted time and attention, and the critical evaluation of this work.

Special thanks to my advisor, Dr. Márcio Dorn, for all the support and guidance while developing and reporting this work. To my coadvisor, Dr. Mario Inostroza-Ponta, for the useful insights and welcoming me at USACH. To my researcher colleague, Dr. Bruno César Feltes, for the inspiring conversations, overall help with all the biological analyses and writing. To Dr. Manuel Villalobos-Cid for the aid with the statistical filtering, plotting tools, and the hospitality at Santiago de Chile. To Eduardo Bassani Chandelier for the work data conversion. To Elisa John for the support and text revision. To Gabriel Toschi for the aid with binomial distribution. To my colleagues at Structural Bioinformatics and Computational Lab for helpful advice. Finally, to my family and my friends for believing in me and in my work.

ABSTRACT

Microarrays are one of the major techniques employed in the study of genes expression, but the identification of expression patterns from microarray datasets is a significant challenge to overcome. In this work, besides reviewing the application of machine learning in the tasks of microarray classification and gene selection, a new approach using Neuroevolution, a machine learning field that combines neural networks and evolutionary computation, is proposed for simultaneously classifying microarray data and autonomously selecting the subset of more relevant genes. The algorithm FS-NEAT was adapted by the addition of three new structural operators designed for better exploring this high dimensional space. A rigorous filtering and preprocessing protocol was also employed to select quality microarray datasets for the experiments, selecting 13 datasets from three different cancer types (breast, colorectal, and leukemia). The results from different experiments show that the proposed method was able to successfully classify microarray samples when compared with other alternatives in the literature, including regular FS-NEAT and SVM, while also finding subsets of genes that can be generalized for other algorithms and carry relevant biological information. This approach detected 177 genes capable of differing classes, 82 of them already being associated to their respective cancer types in the literature and 44 being associated to other types of cancer, becoming potential targets to be explored as cancer biomarkers.

Keywords: Machine learning, neuroevolution, feature selection, classification, supervised learning, NEAT, microarray, gene expression, gene selection.

N3O: Uma expansão de NEAT para melhorar a classificação e seleção de características aplicada a dados de microarranjo

RESUMO

Microarranjos são uma das principais técnicas empregadas no estudo de expressão gênica, mas a identificação de padrões de expressão a partir de conjuntos de dados de microarranjo é um desafio significativo a se superar. Neste trabalho, além de revisar a aplicação de aprendizado de máquina nas tarefas de classificação de microarranjos e seleção de genes, uma nova técnica utilizando Neuroevolução, um campo do aprendizado de máquina que combina redes neurais e computação evolutiva, é proposta para simultaneamente classificar dados de microarranjo e automaticamente selecionar o subconjunto de genes mais relevantes. O algoritmo FS-NEAT foi adaptado através da adição de três novos operadores estruturais projetados para melhor explorar este espaço de busca de alta dimensionalidade. Um rigoroso protocolo de filtragem e preprocessamento foi empregado para selecionar conjuntos de dados de microarranjo de qualidade para os experimentos, selecionando 13 conjuntos de dados de três tipos diferentes de câncer (mama, colorretal e leucemia). Os resultados de diferentes experimentos mostram que o método proposto foi capaz de classificar amostras de microarranjos satisfatoriamente quando comparado com outras alternativas da literatura, incluindo FS-NEAT padrão e SVM, enquanto também encontrando subconjuntos de genes que podem ser generalizados para outros algoritmos e carregam informação biológica relevante. Esta abordagem detectou 177 genes capazes de diferenciar classes, dos quais 82 já foram associados aos seus respectivos tipos de câncer na literatura e 44 foram associados a outros tipos de câncer, tornando-se alvos em potencial a serem explorados como biomarcadores de câncer.

Palavras-chave: aprendizado de máquina, neuroevolução, seleção de características, classificação, aprendizado supervisionado, NEAT, microarranjo, expressão gênica, seleção de genes.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ABC	Artificial Bee Colony
ALL	Acute lymphoblastic leukemia
AML	Acute myeloid leukemia
ANN	Artificial neural network
ANOVA	One-way analysis of variance
ATL	Adult T-Cell Leukemia/Lymphom
cDNA	Complementary DNA
CD34	Cluster of Differentiation 34
C-Mantec	Competitive Majority Network Trained by Error Correction
CRC	Colorectal Cancer
DAVID	Database for Annotation, Visualization and Integrated Discovery
DE	Differential Evolution
DEG	Differentially expressed gene
DL	Deep Learning
DNA	Deoxyribonucleic acid
EC	Evolutionary computation
Evo-Devo	Evolutionary developmental biology
FS	Feature selection
FS-NEAT	Feature Selective NEAT
GA	Genetic algorithm
GEO	Gene Expression Omnibus
GO	Gene ontology
GSE	Gene Expression Omnibus Series
GSEA	Gene Set Enrichment Analysis

HER	Breast Cancer - HER Status
JMML	Juvenile myelomonocytic Leukemia
KO	Knockdown
KW	Kruskal-Wallis H Test
lncRNA	Long non-coding RNA
Max	Maximum
Min	Minimum
miRNA	Micro RNA
mRNA	Messenger RNA
MLP	Multilayer perceptron
N3O	Three new operators
NEAT	Neuroevolution of Augmenting Topologies
PB	Peripheral blood
PBSC	Peripheral Blood Stem Cell
PCA	Principal Component Analysis
RMA	Robust multichip average
RNA	Ribonucleic acid
siRNA	Small interfering RNA
STD	Standard deviation
SVM	Support vector machine
tanh	Hyperbolic tangent

LIST OF SYMBOLS

Φ	Activation function of an artificial neuron
!	Factorial
μ	Mean
σ	Standard deviation
$\sum$	Summation
e	Euler’s number
$\forall$	For all
$\in$	Is an element of
$\lvert A\rvert$	Length of vector A
$\overline{A}$	Mean of vector A
$\ln$	Natural logarithm
$\log_n$	Logarithm with base n

CONTENTS

1 INTRODUCTION

Microarray technology allows the study of several biological questions: it can aid in the understanding of the basic functionalities of an organism, or the behavior of complex diseases. However, despite the large number of available tools for microarray gene expression analysis, the identification of gene expression patterns is still a significant challenge (WALSH et al., 2015).

Machine learning algorithms have been employed in microarray data analysis in order to help to make sense of the large volume of data, often with the two objectives of sample classification and gene selection. The first is a supervised learning task: given a gene expression pattern, it aims to identify its label correctly. For instance, it may be used for creating a classifier able to tell a normal tissue apart from a tumoral tissue. This approach has many applications in clinical diagnostics and has been successfully tested with different algorithms in the past years (LEUNG; CAVALIERI, 2003).

The other task, gene selection, is a subdivision of the more general problem of feature selection (MIAO; NIU, 2016), a form of dimensionality reduction. Gene selection can improve the classification result and is also useful in the biological context by aiding in biomarkers identification as it finds subsets of genes that have a better discriminatory capacity. This work describes the design and application of a new extension of the Neuroevolution algorithm known as NEAT, as a tool to perform classification and identify gene expression patterns in microarray data autonomously.

1.1 Thesis overview

To facilitate the understanding of the work being presented, the next two chapters of this thesis are introductory reviews. Chapter 2 offers a background in the statistical and computational methods referred in this work, with the emphasis in Neuroevolution. Chapter 3 introduces microarray experiments and discusses in more detail the tasks of microarray classification and gene selection.

The next segment of the thesis is focused on the proposed method itself. Chapter 4 explains the process of obtaining and dealing with microarray data, and Chapter 5 describes the design of the computational method being proposed. In Chapter 6 the experiments and results are explained and discussed, and Chapter 7 wraps up the work.

2 STATISTICAL AND COMPUTATIONAL METHODS

Before explaining the problems being tackled by this work in Chapter 3, an understanding of the available computational tools needed for the different tasks is necessary. This chapter is a brief introduction to one statistical test and several machine learning and optimization algorithms that will be later employed. Special focus is given to the topic of Neuroevolution, the main computational strategy behind the new method described in Chapter 5.

2.1 Kruskal–Wallis one-way analysis of variance

The Kruskal-Wallis one-way analysis of variance (KRUSKAL; WALLIS, 1952), also referred as Kruskal–Wallis H test, is a nonparametric statistical test for discovering if samples originate from the same distribution. It compares two or more groups of equal or different sample sizes. Since it is nonparametric there is no need to assume the normal distribution of the data, unlike the one-way analysis of variance (ANOVA) (ARMSTRONG; SLADE; EPERJESI, 2000), its parametric equivalent. The null hypothesis is that there is no difference between the distribution of the groups being tested.

The test is computed by ranking all samples from all groups together, and then calculating the H value using the Equations 2.1, 2.2, and 2.3.

$$H = (N-1)\frac{\sum_{i=1}^{g} n_i(\overline{r}_i - \overline{r})^2}{\sum_{i=1}^{g}\sum_{j=1}^{n_i}(r_{ij} - \overline{r})^2} \tag{2.1}$$

$$\overline{r}_i = \frac{\sum_{j=1}^{n_i} r_{ij}}{n_i} \tag{2.2}$$

$$\overline{r} = \frac{1}{2}(N+1) \tag{2.3}$$

In which N is the total number of samples in all groups, g is the number of groups, n_i is the number of samples in the group i, r_{ij} is the rank of sample j from group i considering the rank among all samples, $\overline{r}_i$ is the average rank of samples in group i, and $\overline{r}$ is the average of all r_{ij}. The p-value is approximated using chi-squared by $\mathbf{Pr}(\chi^2_{g-1} \geq H)$.

2.2 Support vector machine

Support vector machine (SVM) (CORTES; VAPNIK, 1995) is a classical supervised learning method for classification that works by finding the hyperplane (being just a line in 2D or a plane in 3D) capable of splitting data points into different classes. This separating hyperplane acts as a decision boundary, and the "learning" consists in finding a separating hyperplane that maximizes the distance between itself and the closest data points from each class, called the support vectors. In the cases in which the data is not linearly separable, kernels are used to transform the data by mapping it to higher dimensions where a separating hyperplane can be found (HARRINGTON, 2012).

SVM usually performs well on new datasets without the need for modifications. It is also not computationally expensive, has low generalization errors and, in the case of the low dimensionality of the data, is interpretative. It is, however, sensitive to kernel choice and parameter tuning, and only capable of performing binary classification without algorithmic extensions (HARRINGTON, 2012).

2.3 Genetic algorithms

Stochastic methods comprise the class of algorithms that make use of randomness to find optimal or near-optimal solutions for hard problems (LUKE, 2009). Metaheuristics are a general subdivision of such algorithms, applied to a large number of different types of problems. Among them, populational methods keep sets of possible candidate solutions for a given problem, and these solutions are gradually changed until it converges to a local solution (LUKE, 2009).

Many populational methods are inspired by Biology, among them evolutionary algorithms, that incorporate concepts from genetics and evolution. Since the decade of 1970, Genetic Algorithms (GA) are some of the most important algorithms of this kind (LUKE, 2009). A GA operates iteratively, setting "fitness" values for the solutions (called "individuals"), as a way to define how fit they are, and then selects and changes them, creating a new population (LUKE, 2009).

The individuals are represented as a "genome", in which each gene correspond to a particular attribute of the solution. There are several genome representations, two of the most common being binary or real values vectors (KUTHAN; LANSKY, 2007). A popular method for selection in GA is the "tournament selection", that selects individuals in the population by promoting "tournaments" between k randomly picked individuals and selecting the winner of the tournaments (the individual with the best fitness among the participants) (MILLER; GOLDBERG et al., 1995). Experimental tests showed that it can outperform other methods of selection (ZHANG; KIM, 2000; SHARMA; WADHWA, 2014).

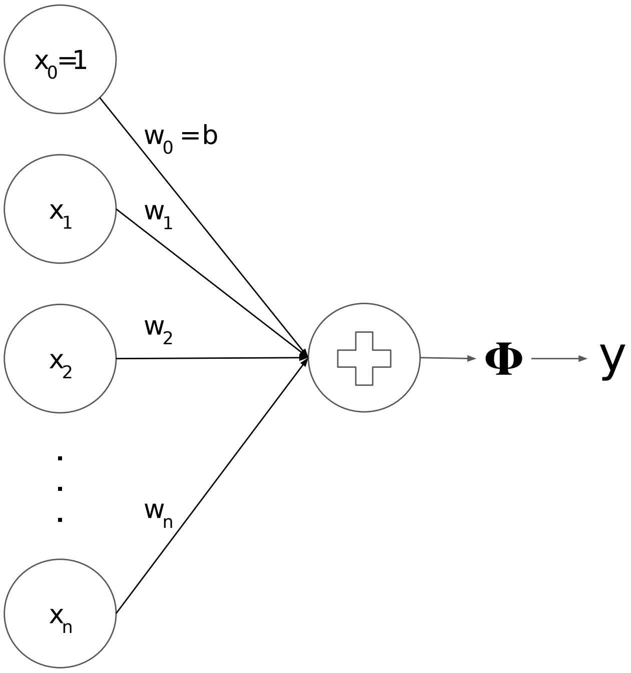
The principal change operators in GA are mutation and crossover. There are several ways to mutate an individual, but it is basically a random perturbation in its genome. The crossover combines two individuals, called "parents", exchanging the genes in their genomes and creating a new individual with characteristics from both, called "offspring", that possibly has a better fitting than the parents, being a better solution for the task at hand (GOLDBERG, 1989). The core idea is that the fittest individuals in each iteration, called a "generation", are selected for crossover, generating better solutions, while the bad solutions are removed from the population, and the random mutation operator improves the exploration of the search space (LUKE, 2009). Another common operator is "elitism", that simply copies to the next generation a fraction of the best individuals in the current generation, in order to preserve the best historical solutions found in the course of the algorithm (BALUJA; CARUANA, 1995).

2.4 Artificial neural networks

Artificial neural networks (ANNs) are a group of machine learning methods inspired by the flow of information in the biological brain, used for estimating or approximating any function (HORNIK, 1991), being largely used in classification and regression tasks. They are formed by neurons, the processing unity, linked by connections usually forming a layering structure. A neuron is composed of three basic elements: (i) a set of input connections, defined by a weight that multiplies the input signal; (ii) an aggregation function, that combines all input signals multiplied by the connections weights; (iii) an activation function that transforms the output of the neuron and usually introduces non-linearity.

Equation 2.4 represents a neuron k, with x_0, x_1, ..., x_m being the input signals plus a bias b_k, that can increase or decrease the threshold for activation, w_{k0}, w_{k1}, ..., w_{km} are the weights of the neuron connections, Φ is the activation function, and y_k is the output of the neuron (HAYKIN, 1998). Fig. 2.1 illustrates this model of a neuron. A neural network can be seen as an oriented graph through which flows a signal from the input nodes to the

Figure 2.1: Model of an artificial neuron.



Source: Adapted from Haykin (1998)

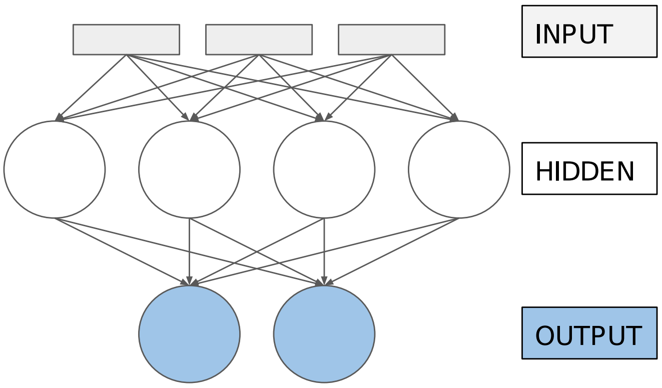
computing nodes.

$$y_k = \Phi\left(\sum_{j=0}^{m} w_{kj}x_j + b_k\right) \tag{2.4}$$

Neural networks with multiple layers are often called a multilayer perceptron (MLP), and in case they do not have any loop on their structures they are also referred as feed-forward networks. These MLP consist of an input layer that receives the real data values, a succession of one or more layers of neurons called the hidden layers, and an output layer. Usually, the outputs of neurons in one layer are fully connected to the inputs of the neurons in the next layer. This architecture is illustrated in Fig. 2.2, representing a generic MLP with three input nodes, four hidden nodes, and two output nodes. MLP are commonly used in supervised learning, in which the algorithm "learns" by facing labeled data and performing small changes in its internal parameters (weights and biases) in order to produce outputs closer to the real label. One of the most common algorithms for training a neural network is the error back-propagation (LINNAINMAA, 1976; HAYKIN, 1998).

This algorithm can be divided into two steps: feedforward and backpropagation. In the first one, the input vector is passed by the input layer to the next layer, and the

Figure 2.2: Model of a multilayer perceptron.



values are propagated forward until the last layer is reached. In the second step, the error (the difference between the correct answer and the network output) is sent back through the network, that adjusts its weights and biases to approximate the output and the correct answer (HAYKIN, 1998; LINNAINMAA, 1976). The most used algorithms for performing this parameter optimization are gradient descent variations (KIEFER; WOLFOWITZ, 1952).

One of the biggest challenges in designing a MLP is to define its topology, i.e., the number of neurons, layers, and connections. The creation of the network structure is one of the most important factors for its success, and generally, it needs to be planed with a specific task and data previously defined. Carelessly creating a network structure can cause inefficient and inadequate results (CURTEANU; CARTWRIGHT, 2011). Large neural networks are also computationally expensive to train and needs large amounts of data, besides being prone to overfitting.

Overfitting happens when the model performs well on the training data, but fails to generalize and has poor performance when facing new data (test data). This means that the ANN, instead of learning about the true and generic characteristics of the task, simply "memorized" the patterns it was exposed to in the training. One way to avoid this problem is to expand the dataset, for example performing new experiments or observations, but this can be expensive and not always possible depending on the kind of data. A variation of this 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 real-world phenomena.

Another strategy commonly used with ANNs is the incorporation of regularization in the training of the network. L2 regularization, also known as weight decay, is one of the most popular techniques to mitigate overfitting (NG, 2004). It makes the optimization prefer networks with smaller weights, what makes the model simpler, usually capable of better generalization, by penalizing networks with large weights and biases with the term in Equation 2.5.

$$Reg = \frac{\lambda}{2n} \sum_{k=1}^{c} w_k^2 \tag{2.5}$$

In Equation 2.5, n is the number of samples in the dataset, c is the number of connections, w_k is the weight of connection k, and λ is the regularization parameter, that must be a positive value set by the programmer. This value is added to the cost function being minimized.

2.5 Neuroevolution

Neuroevolution is the process of creating and training neural networks with Evolutionary Computation (EC). EC makes use of concepts borrowed from biology, such as inheritance, selection, and random variation, adapting them to solve computational problems. EC is easily parallelized, does not require a large amount of data, and can create solutions based on any fitness criteria (SIPPER; OLSON; MOORE, 2017). Neuroevolution uses EC to find the best neural networks, and is more efficient than other methods for problems with continuous state space and high dimensionality, besides having better memory representation (GOMEZ; MIIKKULAINEN, 1999; GOMEZ; MIIKKULAINEN, 2002).

Many neuro evolutive methods only operate over neural networks with fixed topology, usually with few hidden layers and a full-connected architecture, and optimizes the weights and biases. The topology of a network, however, also affects its functionality and could be part of the evolutive process (ANGELINE; SAUNDERS; POLLACK, 1993; BRANKE, 1995; GRUAU F.; PYEATT, 1996; YAO, 1999).

2.5.1 NeuroEvolution of Augmenting Topologies

NeuroEvolution of Augmenting Topologies (NEAT) is an algorithm for building and training neural networks using GA to evolve the topology and weights. This method is adequate for problems without a known satisfactory network structure (STANLEY;

MIIKKULAINEN, 2002). The networks are encoded as a "genome", that keeps the connection information (Fig 2.3a).

NEAT starts with a random initial population, in which each individual is a neural network, and all individuals share the same minimal topology, i.e., input nodes, output nodes, and connections between them with random weights. This initial condition is important to not introduce useless complexity in the solutions since only the new structures that benefit the networks fitness are kept. If the algorithm was initialized with random structures, neurons and connections not needed could be present since the beginning, and it would be impossible to remove them, what could cause a negative impact on the evolution. This approach also produces smaller and simpler results (STANLEY; MIIKKULAINEN, 2002).

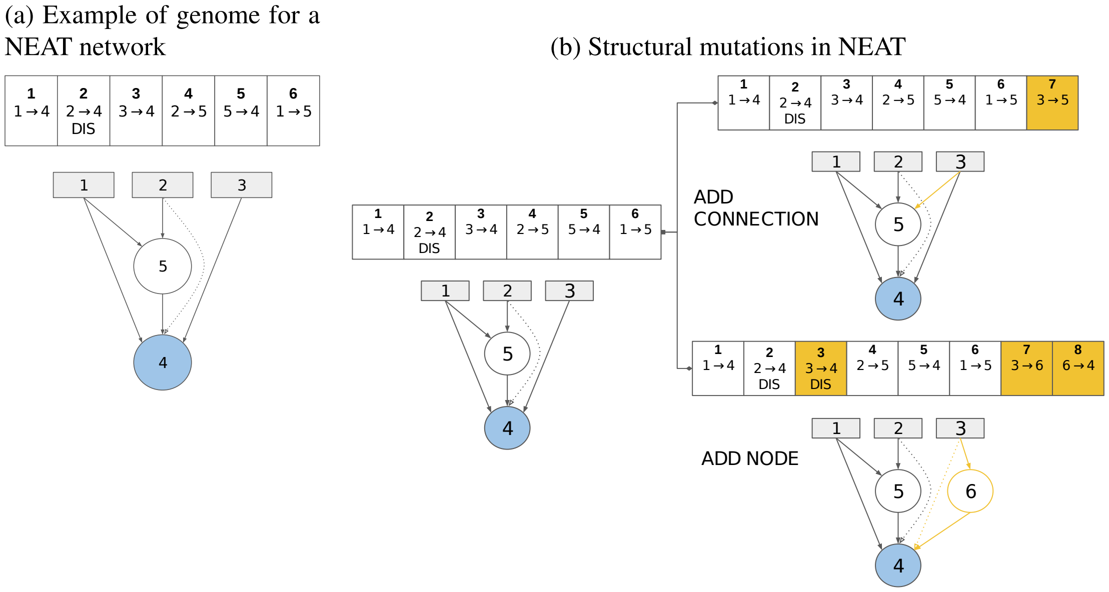
From this initial population of neural networks, new ones are iteratively created through the traditional GA operators, especially mutation, that can change the weight of a connection or the bias of a neuron, add new hidden neurons, or add a new connection between two neurons (Fig. 2.3b), and crossover, that combines two individuals (Fig 2.4). Mutation in NEAT never removes a neuron or connection because it could cause inconsistencies during the evolution. Instead, a marker can be defined to ignore a connection between neurons (STANLEY; MIIKKULAINEN, 2002).

The biggest challenge in this strategy is how to combine two different neural networks during crossover without producing a defecting network, since the topologies of the parents could be not directly compatible with neurons and connections exchange. Because of this, NEAT uses historical markers, a numerical value given to each new structure that appears during the evolutive process and that is passed along without changes in the crossover. They allow the method to correctly align the same parts of the topology of two distinct networks, creating a new functional neural network that does not break its parents structural organization. Historical markers are set at the end of each generation, so if two identical new structures arise in different individuals theirs markers have the same value (STANLEY; MIIKKULAINEN, 2002).

The last problem of implementing NEAT is that adding new structures to existing neural networks, without any adjustment, is usually prejudicial to them. In this case, neural networks would receive new neurons and connections that would produce unfavorable results immediately, causing them to be discarded from the population, even if their novelties are beneficial in the long term. Due to this, NEAT uses speciation, also known as niche, a technique that clusters the individuals by their structural similarity using the his-

Figure 2.3: (a) Representation of the genome of one individual (neural network) in a population in NEAT. The numbers at the top in bold are the historical markers used for identifying the structural novelties. The second information is the connection between two nodes. "DIS" indicates if the particular gene is enabled or disabled (in this case it is ignored). The values of connections weights and neurons biases are omitted for clarity. Gray rectangles represent input nodes, white circles represent hidden nodes, blue circles represent output nodes, arrows represent a connection between nodes, and dotted arrows represent disabled connections. (b) Illustration of the two possible structural mutations in NEAT. "Add connection" adds a connection with random weight between two randomly selected nodes in the network, in this case, nodes 3 and 5, and generates a new historical marker. "Add node" creates a new node with random bias in the place of an existing connection, that is disabled, and creates two new connections, one from the disabled connection origin node and the new node, that receives the weight value of the disabled connection, and one from the new node to the disabled connection destination node, that receives a random weight value. In the example, the new node 6 is added between nodes 3 and 4, that were already connected. The changes are coloured in yellow.

Source: Adapted from Stanley and Miikkulainen (2002)

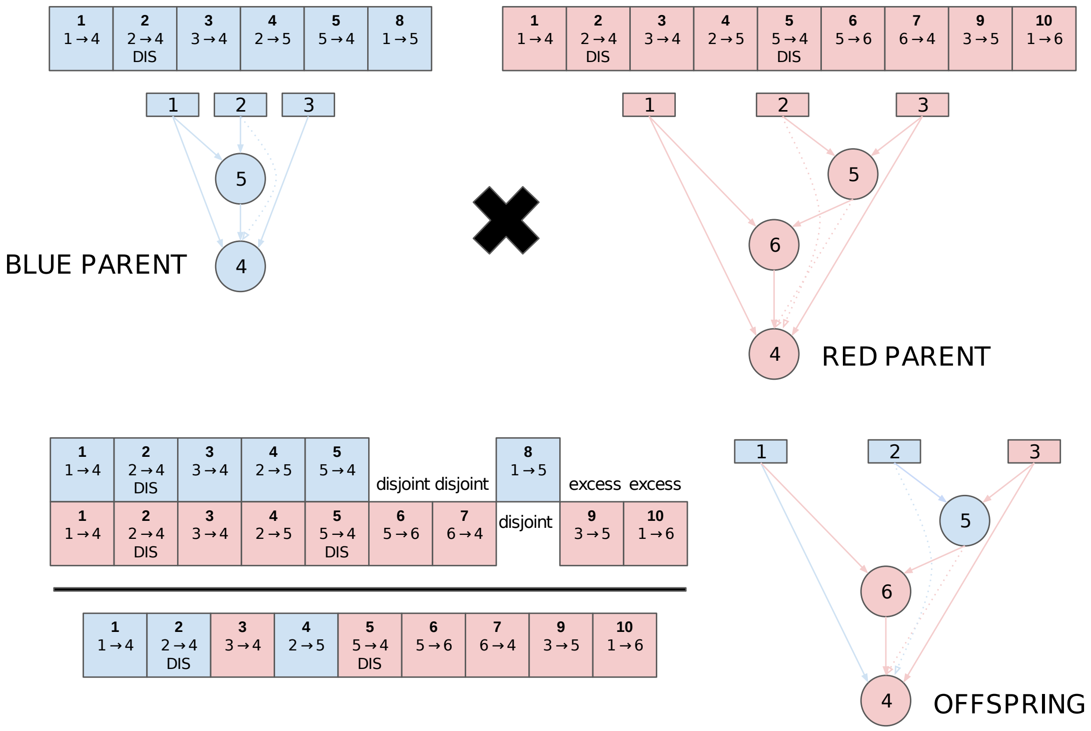

(a) Example of genome for a NEAT network

(b) Structural mutations in NEAT

torical markers, and promotes competition only inside the same niche. The compatibility between individuals is computed using the Equation 2.6 (STANLEY; MIIKKULAINEN, 2002) in which c_1, c_2, and c_3 are coefficients set by the user, N is the number of structures in the largest network, E is the number of excess structures, D is the number of disjoint structures, and $\bar{W}$ is the average weight differences of matching structures. This way, the networks have time to adjust, not being simply discarded as soon as they are created (STANLEY; MIIKKULAINEN, 2002).

$$d = c_1 \frac{E}{N} + c_2 \frac{D}{N} + c_3 \bar{W} \tag{2.6}$$

Figure 2.4: Example of NEAT crossover between two individuals, in which the red parent has better fitness than the blue parent. Their genes are aligned using the historical marker (numbers in bold) in order to avoid structural errors. The offspring receives genes with equal probability from any of the parents if they are present in both, or from the parent with better fitness if they are disjoint or excessive.



Source: Adapted from Stanley and Miikkulainen (2002)

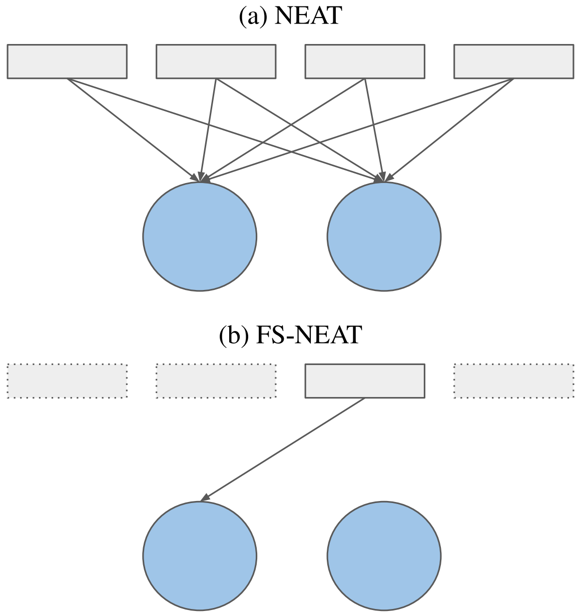
NEAT is a powerful tool for the artificial evolution of neural networks, and experiments have shown that it is more efficient than other neuro evolutive methods. Evolving the topology with the connection weights is an advantage, optimizing and complexifying the solutions simultaneously (STANLEY; MIIKKULAINEN, 2002).

2.5.2 Feature Selective NEAT

NEAT has been used for the task of feature selection by several studies (SOHANGIR; RAHIMI; GUPTA, 2014; SOHANGIR; RAHIMI; GUPTA, 2013; TAN et al., 2009), one of the most relevant being Feature Selective NEAT (FS-NEAT) (WHITESON et al., 2005), that is both a simple and efficient alternative for FS (PAPAVASILEIOU; JANSEN, 2017b; PAPAVASILEIOU; JANSEN, 2016; ETHEMBABAOGLU; WHITESON et al., 2008). In FS-NEAT the minimalist start of NEAT is changed, and instead of a fully connected topology, only one random input is connected to one random output in each individual in the first generation. This difference is illustrated in Fig. 2.5. A new mutation operator is also included, that adds inputs to a network by connecting it to an output (Fig. 2.6).

Figure 2.5: Examples of initial topology for (a) NEAT and (b) FS-NEAT. The first population of networks in regular NEAT has input and output layers fully connected, while FS-NEAT has networks with randomly selected connections between one input and one output. Dotted inputs are currently not selected by the network.

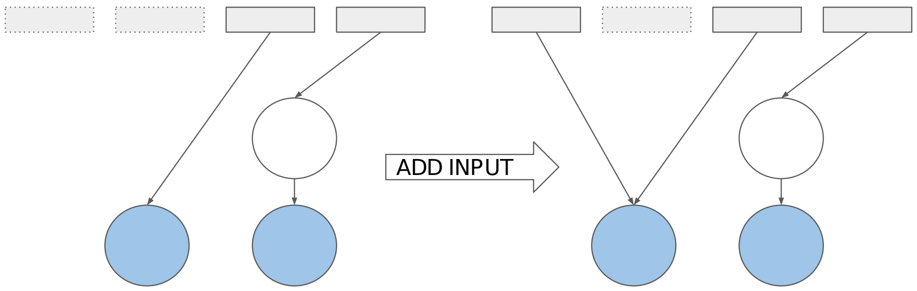
Source: Adapted from Whiteson et al. (2005)



(a) NEAT

(b) FS-NEAT

Figure 2.6: Example of the extra FS-NEAT structural mutation, that adds a new input in a network by creating a connection with random weight value between the input being added and one output.



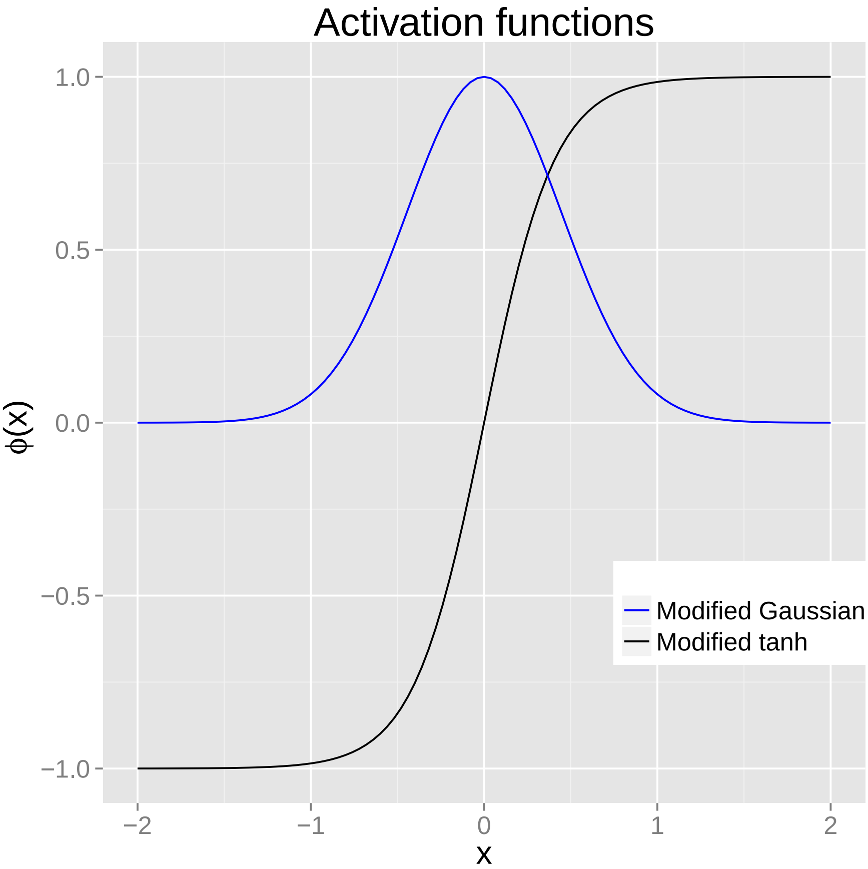
This kind of network will lack the needed structure for a favorable result, but, guided by the evolutive algorithm, it will grow in complexity towards networks capable of solving the task without all the inputs. At the end, the inputs that are not directly or indirectly connected to an output node are discarded, since their information is not being used. This way, FS-NEAT automatically does FS without meta-learning, while creating smaller networks.

Further studies concluded that FS-NEAT outperforms other neuro evolutive methods for FS that start with all inputs selected if the majority of the inputs are irrelevant or redundant (PAPAVASILEIOU; JANSEN, 2017b; PAPAVASILEIOU; JANSEN, 2016). It was also observed that the use of the modified hyperbolic tangent (tanh) (Equation 2.7) as activation of the hidden nodes and the modified Gaussian function (Equation 2.8) as activation of the output nodes makes the algorithm converge faster, improves the FS and accuracy, and generates smaller networks in comparison with other combinations of activation functions (PAPAVASILEIOU; JANSEN, 2017a). The behavior of both activation functions is illustrated in Fig. 2.7.

$$\Phi(x) = tanh(4.9 \times 0.5x) \tag{2.7}$$

$$\Phi(x) = exp(-\frac{5(x-\mu)^2}{2\sigma^2}), \mu = 0, \sigma = 1 \tag{2.8}$$

Figure 2.7: Activation functions for FS-NEAT



This may be related to a tradeoff between the global approximation in the hidden layers and refined local search in the output layer (PAPAVASILEIOU; JANSEN, 2017a).

This could be due to the “smooth” transitions produced by the Gaussian function, and its catchment region that, even for vectors far from the center, will always be larger than zero (KRUSE et al., 2016; PAPAVASILEIOU; JANSEN, 2017a).

2.6 Machine learning for biological data

Although this chapter is focused on the computational aspects of the cited algorithms, it is relevant for this work to understand how the field of machine learning is being applied in biological research. In a previous work, that can be read in the Appendix A, we present a full review on the use of machine learning methods in Evolutionary developmental biology (Evo-Devo) studies, including many of the aforementioned techniques. While out of the scope of this specific work, genomics and microarray data are also extensively discussed in this review, corroborating with key ideas of Chapter 3.

2.7 Chapter conclusion

This chapter introduces several computational and statistical methods in a broad and general way. The Kruskal-Wallis one-way analysis of variance is a nonparametric test that allows the determination if two or more groups of samples came from the same distribution. SVM is a simple but effective algorithm for data classification, and ANNs can be created and used in the task of FS by incorporating ideas from GA. All this information will be essential for chapters 5 and 6, in which it will be used for the construction of the proposed method. Moreover, statistical tests, such as the Kruskal-Wallis one-way analysis of variance, or classifiers, such as SVM and ANN, can be important tools when dealing with gene expression data from microarrays experiments, as will be seen in the next chapter.

3 MICROARRAY DATA ANALYSIS

As the last chapter, the aim of this one is to briefly introduce key topics that will later be useful for the comprehension of the method being proposed and the found results. This chapter reviews some biological concepts about gene expression, and then focus in the microarray technology and how this kind of data can be analyzed, especially how it can be used for the creation of classifiers or the identification of informative genes.

3.1 Gene expression

The DNA contains the codification for all RNA and protein molecules needed for the construction of an organism's cells, and the complete DNA sequence (ranging from millions to billions of nucleotides, depending on the organism) is present in all cells. Even so, the structure and function of different cell types of the same multicellular organism can be remarkably different, while the genome remains the same (ALBERTS, 2015; GILBERT, 2000).

These differences are due to different sets of genes (specific segments of DNA) in each cell type being expressed, i.e., the information from genes being transcribed in RNA (and often producing proteins as the final result). For the majority of genes, the most important control of expression is the beginning of RNA transcription, but this can be changed by the environment, for instance, due to signals from other cells (ALBERTS, 2015).

Many processes are common to all cells in the same organism, resulting in the same gene products, such as DNA repair enzymes and structural proteins of chromosomes, so the set of expressed genes is the same in all cells. On the other hand, some RNAs and proteins are found only in specialized cells and not anywhere else, meaning the correspondent genes are only being expressed in that specific cell type. A classic example is hemoglobin, exclusively expressed in red blood cells. A typical human cell has around 30,000 genes, of which 30% to 60% are expressed simultaneously at some level. Comparing patterns of RNA expression in different human cell types reveals a variation in expression from one type to another in nearly all genes, even though this variation is often small (TAO et al., 2017; ALBERTS, 2015).

Knowing a gene's expression can be useful to predict its function by identifying which other genes share the same expression pattern. If a set of genes are expressed with high correlation under different situations, they probably are coordinately regulated and act together in the cell, for example encoding proteins that are involved in the same coordinated activity (ALBERTS, 2015). It can also be used to study the differences in expression between cells or tissues of the same type under two different conditions, for instance a disease state, providing a method for understanding its mechanisms (TAO et al., 2017).

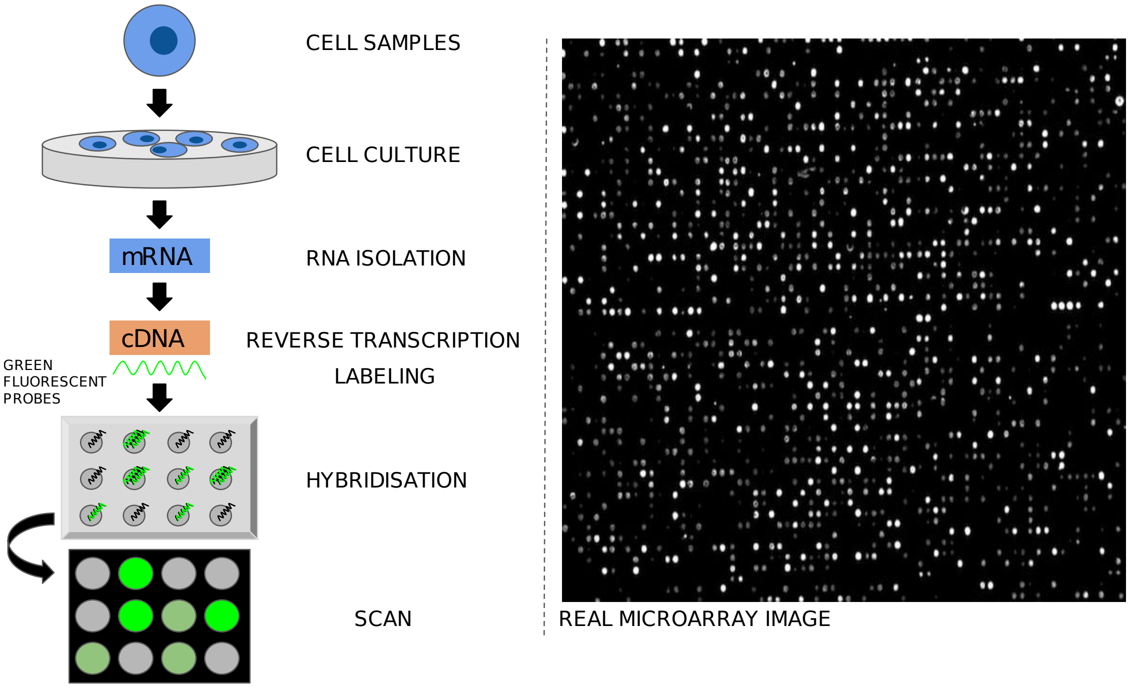
A possible way for discovering which genes among the thousands in the cell genome are being differentially expressed between different cell types, environments, or conditions, is measuring the amount of messenger RNA (mRNA) being produced. DNA microarray, described in the next section, was the first technology to allow the analysis of thousands of different RNAs at the same time (ALBERTS, 2015).

3.2 Microarray experiments

The field of functional genomics requires the analysis of large amounts of information from several biological experiments, such as evaluating the expression levels of thousands of different genes under some specific condition. This large-scale gene expression analysis was made possible to a great extent by the advent of the microarray technology (WHITWORTH, 2010). Microarrays are available from different platforms, providing information on mRNA, micro RNA (miRNA), long non-coding RNA (lncRNA), and exon arrays (TAO et al., 2017; WANG et al., 2014; GORRETA; CARBONE; BARZAGHI, 2012; DENIZ; ERMAN, 2017). This allowed the study of several biological questions, from the basic functionality of an organism to the understanding of complex diseases, including cancer (TAO et al., 2017). The use of microarrays to analyze mRNA, however, continues to be the most common of those, and the identification of expression patterns is still a challenge (WALSH et al., 2015). RNAseq is another technology available for those studies, and is gradually replacing microarray as the major technique applied to gene expression analysis. While this work focuses on microarray, the use of data coming from RNAseq is also viable.

A microarray experiment consists of a glass slide with DNA molecules on it. The molecules are fixed in a specific order and locations, that are called spots and contain millions of copies of identical DNA molecules, per spot, that correspond to just one specific gene, named probes (MURPHY, 2002; WHITWORTH, 2010). Usually these molecules are mRNA molecules extracted from the cells and transcribed into cDNA labeled with fluorescent dyes. This way, cDNA sequences in the studied sample will hybridize to the specific spots in the glass slide that have their complementary sequence, so that the amount of dye in each spot will be proportional to the amount of that particular cDNA sequence in sample. Finally, the spots are excited by a laser, allowing the detection of the wavelengths of the dyes. The amount of the emitted fluorescence corresponds to the amount of nucleic acid expressed in the sample (MURPHY, 2002; WHITWORTH, 2010). This is the procedure in a single-channel experiment, that only uses one dye color, but there is also the possibility of a dual-channel experiment, which uses two dyes, one for each sample group, allowing a more direct comparison between them. This process is showed in Fig. 3.1.

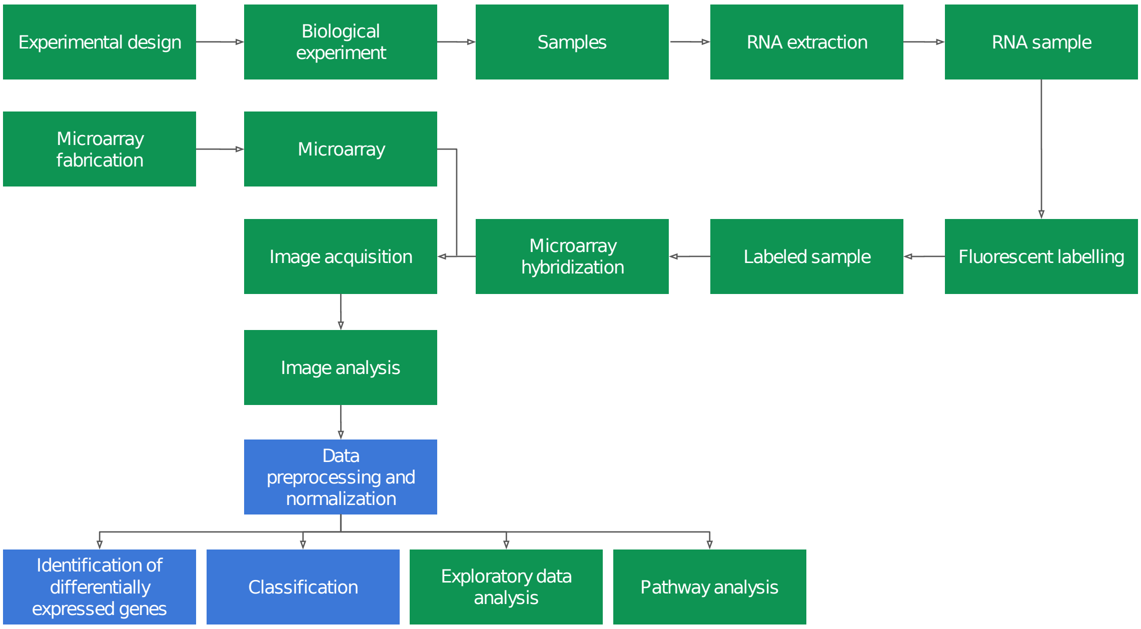
Figure 3.1: **Diagram of a single-channel microarray experiment.** The cell samples can come from patient tissues, animal model tissues, etc. The RNA is extracted from the samples, labeled with the fluorescent dyes, and hybridized to a microarray. The image is then scanned from the microarray.



Source: Adapted from BioNinja[1] and Laboratory-Equipment.com[2]
[1] <http://ib.bioninja.com.au/standard-level/topic-3-genetics/35-genetic-modification-and/cdna-and-microarrays.html> [2] <https://www.laboratory-equipment.com/pba/spotlight-2-turbo-microarray-fluorescence-scanner-arrayit.php>

Since the result of the experiment is a color image of the spots in the glass slide, and the color intensity of each spot is the gene expression information, the next step is image processing and analysis to retrieve the expressions values. The image processing detects the spots and filters noisy signals, then determines the spot area and discovers the signal intensity by comparing it against the background intensity (WHITWORTH, 2010). The raw data is not the best option for biological knowledge discovery, mostly due to the presence of noise and the variability between the different technologies involved (RESSOM et al., 2009). Due to that, many methods of background correction, data transformation, data normalization, as well as statistical validation are available and should be applied (WHITWORTH, 2010; RESSOM et al., 2009). The final result can be represented as a 2D matrix, in which rows represent the probe names (genes) and columns represent the samples of the experiment, usually from distinct conditions, in the case of single-channel, or the whole experiment, in the case of dual-channel. Each element in the matrix is a continuous numerical value indicating the expression of that particular gene for that particular sample (RESSOM et al., 2009). A gene expression matrix from a regular microarray experiment will commonly have thousands of rows and dozens or hundreds of columns (RESSOM et al., 2009).

Figure 3.2: **Flowchart of a microarray experiment.** Expansion of the pipeline showed in Fig. 3.1. The image scanned from the microarray is analysed, resulting in the raw signal data that is filtered and normalized. Finally, the data can be further investigated in order to create classifiers, discover informative genes, find new knowledge, perform pathway analysis, among other analysis (LEUNG; CAVALIERI, 2003). The steps approached in this work are highlighted in blue.



Source: Adapted from Leung and Cavalieri (2003)

Once the gene expression data from the microarray experiment is available and analyzed, the result is a set of differentially expressed genes (DEG) that can be further investigated. Two of the most common applications, classification and gene selection, will be described in the next sections. The application of unsupervised learning (clustering algorithms) (Fig. 3.2 - *Exploratory data analysis*) is also widespread, but not without facing some criticism. Allison et al. (2006) point out that there are problems with reproducibility, validation, and biological relevance of these experiments.

3.3 Microarray classification

One of the most relevant possibilities of microarray analysis is class prediction (Fig. 3.2 - *Classification*). This supervised learning task refers to the use of a classifier capable of labeling new samples based on their genes expression (LEUNG; CAVALIERI, 2003). First, a set of samples whose original group (for example control and disease) is known, called the "training set", is used to train a model in assigning to each of those samples its correct group. Once the training is finished, the model is evaluated with new samples (the "testing set") to check its predictive abilities (ALLISON et al., 2006).

Microarray classifiers are promising in clinical diagnosis (LEUNG; CAVALIERI, 2003; QUACKENBUSH, 2001), with successful results (KHAN et al., 2001; SHIPP et al., 2002). The main objective would be the creation of general classifiers capable of being a trustworthy routine diagnostic tool for cases that are difficult to differentiate with other available techniques (LEUNG; CAVALIERI, 2003). Several studies have already tested the efficacy of different machine learning algorithms for this task, such as ANNs, SVM, k-Nearest Neighbors, and Random Forest (PETERSON et al., 2005; DÍAZ-URIARTE; ANDRES, 2006; STATNIKOV; WANG; ALIFERIS, 2008; PIROOZNIA et al., 2008).

There is no single microarray prediction method considered optimal, nor consensus in which one is superior (ALLISON et al., 2006). Comparison studies, however, point to SVM and random forest as being more efficient methods (LEE et al., 2005; PIROOZNIA et al., 2008), with SVM having the upper hand (STATNIKOV; WANG; ALIFERIS, 2008). SVM also seems to be the supervised learning method of choice in works of gene selection (ANG et al., 2016).

The field of deep learning (DL), that uses MLPs with several layers, has recently attracted attention as a powerful tool for the analysis of a wide variety of biological data (PARK; KELLIS, 2015; ANGERMUELLER et al., 2016; MAMOSHINA et al., 2016; MIN; LEE; YOON, 2017; CHING et al., 2018). The performance of neural networks in the task of microarray classification, however, is surprisingly lagging behind older machine learning algorithms (PIROOZNIA et al., 2008; LEE et al., 2005). A possible explanation has been attributed to the use of gradient-based optimization methods and backpropagation (GUPTA et al., 2015). It is also believed that for the sample sizes available in microarray experiments, simpler methods outperform the more complex algorithms (ALLISON et al., 2006). The advantage of SVMs over ANNs would be their better generalization ability that can be obtained with few training samples and scales the importance of outliers, associated with they being well suited for high-dimensional data and faster to train (RESSOM et al., 2009).

Neuroevolution, discussed in the last chapter, avoids some of the pitfalls encountered by the need of having a fixed topology and backpropagation (MARCUS, 2018; MORSE; STANLEY, 2016; SUCH et al., 2017). The minimalist structure of NEAT (Section 2.5.1), for instance, creates smaller and simpler neural networks. Regarding microarray data, some of Neuroevolution components have already been used in the task of classification with promising results (GARRO; RODRÍGUEZ; VAZQUEZ, 2017; GUPTA et al., 2015; LUQUE-BAENA et al., 2013), making this group of strategies a promising focus for future research.

One of the greatest challenges in microarray classification is the aforementioned problem of overfitting (Section 2.4) and the balance between accuracy and generalizability (LEUNG; CAVALIERI, 2003; ALLISON et al., 2006). This is especially true for microarray data, that usually have few samples but thousands of features (RESSOM et al., 2009), since with fewer samples there is a larger chance of algorithms clinging to specific patterns of the training set, losing generalizability (ALLISON et al., 2006).

3.4 Gene selection

Overfitting is closely associated with the "curse of dimensionality" (VERLEYSEN; FRANÇOIS, 2005), when the data have a large number of dimensions, increasing computational processing time, memory consumption, and causing interpretability impairments. Datasets that only possess a small number of samples are also affected by the "large p, small n" problem, as is the case of most microarray experiments. The DL methods mentioned in the last section, for example, rely on training sets with thousands or millions of samples, very rare numbers for microarrays. Taking this in consideration, the application of methods for dimensionality reduction becomes imperative (GARRO; RODRÍGUEZ; VÁZQUEZ, 2016; ANG et al., 2016).

Dimensionality reduction is the process of lowering the number of features of the data, i.e., the number of genes for microarray data. This is not only important from the computational and the classification perspective, but also from the point of view of biological research and useful information extraction. The discovery of genes capable of differentiating samples from different populations (different target annotations, i.e., the samples classes) is an important aspect of microarray data analysis (Fig. 3.2 - *Identification of differentially expressed genes*) (LAZAR et al., 2012). Finding these genes, that are sometimes referred to as biomarkers, or informative genes, is used as means to help in the precise identification of diseases or as potential drugs targets (LAZAR et al., 2012).

The major group of algorithms for dimensionality reduction is feature extraction, a set of methods that transforms the original feature space into a different space with a new set of axis by combining its features and finding the ones that most preserve the original information (VARSHAVSKY et al., 2006). This new feature space often has better discriminatory power, but the extracted features lack physical or biological meaning for better interpretation (ALELYANI; TANG; LIU, 2013; KRIZEK, 2008; ANG et al., 2016). Some examples of feature extraction methods are Principal Component Analysis (PCA) (JOLLIFFE; CADIMA, 2016), Singular Value Decomposition (KLEMA; LAUB, 1980), Factor Analysis (FRUCHTER, 1954), and t-Distributed Stochastic Neighbor Embedding (MAATEN; HINTON, 2008). ANNs are also known for their abilities in feature extraction due to the unsupervised feature learning, autoencoders being the best example (HINTON; SALAKHUTDINOV, 2006).

While feature extraction can be useful from the computational view, its lack of interpretability leaves it with little use for the discovery of informative genes. A subgroup of techniques, called feature selection (FS), however, solves this problem by choosing small subsets of features instead of combining them, usually through the removal of irrelevant, redundant, or noisy features. This is better suited for biological data since it leads to better performance and model interpretability (MIAO; NIU, 2016). Examples of such methods are Minimum Redundancy Maximum Relevance (DING CAND PENG, 2005), Information Gain (ALHAJ et al., 2016), Chi-Square (JIN et al., 2006), Fisher Score (GU; LI; HAN, 2012), Relief (KIRA; RENDELL, 1992), and Lasso (FONTI; BELITSER, 2017). Compared with feature extraction, FS is less general and may provide less discriminatory power, but once again, in the context of informative genes discovery, the lack of physical meaning in feature extraction is prohibitive (ALELYANI; TANG; LIU, 2013; KRIZEK, 2008; ANG et al., 2016).

Gene selection is the name given to the application of FS in microarray data, with the objective of discovering subsets of genes capable of separating samples from different populations. It is necessary when dealing with data that contains noisy, irrelevant, or redundant gene expressions, and can be effectively used for tumor detection at early stages and more reliable cancer diagnosis, prognosis, or clinical treatment (BOULESTEIX et al., 2008; ANG et al., 2016).

This is not a trivial task. Identifying the subset of genes across all samples with the best discriminative power is only possible in the classification of pre-identified groups, since it is usually the case of disease prediction (LAZAR et al., 2012). The set of DEGs, however, does not always provide the best predictive power, and there is no method considered superior (ALLISON et al., 2006). Moreover, a single feature, when observed alone, may be irrelevant, but in combination with other features it may become highly relevant (GHEYAS; SMITH, 2010; ANG et al., 2016). Ideally, the selected features should be all the strongly relevant and sometimes weakly relevant, meaning that features that are useful for improving accuracy prediction when they are non-redundant and do not cause a negative impact in the evaluation measures, while the noisy, redundant, or irrelevant features are discarded (ANG et al., 2016). Redundant features should be discarded due to their significant statistical relations with other features, not for only having worthless information (ANG et al., 2016), what is coherent for the computation, but can cause the solution to miss informative genes with highly correlated expressions. The setting of a threshold for considering a feature relevant or not is also a difficult task. It is needed to balance the false positives and false negatives, and account for the multiple hypothesis-testing problem when statistical tests are performed for thousands of genes (LEUNG; CAVALIERI, 2003).

Gene selection is still an open problem with many challenges and new alternatives emerging. There are already several methods for FS and DEGs discovery, and many algorithms are only slightly different among them (LAZAR et al., 2012). The literature usually groups the methods for gene selection in four types, as presented below.

Filter: as the first methods to be used, they only consider the intrinsic attributes of the data combined with an evaluation criteria (distance, information, dependency, consistency, etc.) and are not necessarily used as classifiers. Most filters are univariate and considers the problem as a ranking problem. They are independent of specific learning algorithms, providing more general solutions that can be used by different classifiers. Filters are also known to be faster and more computationally efficient than the other groups of methods. The drawback is that they ignore the relationships between different features and the effects that they have when combined. They also ignore the interaction between the selected features and the classifier, leading to varying prediction power (LAZAR et al., 2012; ANG et al., 2016).

Wrapper: the selection is made using some optimization algorithm, and then wrapping a classifier around the selected features, using its accuracy as the criteria of evaluation. The set of most discriminative features is found by the minimization of the classification error, what often gives better results than filters. Wrappers, however, are highly dependent on the learning algorithm being used as classifier, and the solutions are not general, meaning that there is no guarantee that the quality of performance of the selected features will be transferable for other classifiers. Wrappers are also more likely to suffer from overfitting and present huge computational costs, since the training of the classifier needs to be performed again for each new subset being evaluated, making them a less common choice than filters (LAZAR et al., 2012; ANG et al., 2016).

Embedded: the FS is built-in the learning algorithm, so the selection and classification are performed together. When compared with wrappers, this approach is more efficient, because it avoids the repetition of training a classifier, and is less prone to overfitting, while achieving similar performance and considering the interactions between features. Nevertheless, the computational complexity in high-dimensional data is still a challenge (ANG et al., 2016).

Hybrid: together with embedded methods, they are novelties in FS. Hybrid methods are a combination of different methods (that can be or not of the same group), different selection algorithms, or different criteria, in an attempt to merge their distinct strengths. The most common combination is between filters and wrappers (ANG et al., 2016).

Most studies on the gene selection subject are focused in filters, due to their generability and computational efficiency (ANG et al., 2016). While the evaluation of filters is independent of any classifier, wrapper and embedded methods use the classifier accuracy itself, despite also requiring strategies to search the feature space to perform the selection (LAZAR et al., 2012). Based in different studies, however, the hybrid methods are the ones with better results, by combining the strengths of the other approaches, reducing the computational costs by narrowing the total search space, and lowering the risk of overfitting (ANG et al., 2016).

Nevertheless, many challenges persist. Technical defects in the experiments, for example, scanning errors, can cause samples to be mislabeled (ANG et al., 2016). Microarrays also suffer from class imbalance problems, when each class has a distinct number of samples, making the accuracy less informative (POWERS; GOLDSZMIDT; COHEN, 2005; LAZAR et al., 2012). The difference between the several technologies and analyses standards of each microarray platform makes cross-platform comparison problematic. More important, the retrieval of biological information from the gene expressions is not an easy task, and the determination of the relevancy or the redundancy of a gene is difficult, leading to unexpected biases and mistakes in conclusions. Despite the large number of methods available in the literature, there is still a lot of room for improvements and innovations (ANG et al., 2016).

3.5 Microarray and Neuroevolution

Regarding microarray classification and gene selection, a few works have used some of Neuroevolution components, making this group of strategies a promising focus for future research, because they avoid some of the problems of traditional DL and through algorithms, like NEAT, can produce simpler models, better suited for dealing with overfitting. Luque-Baena et al. (2013) performed gene selection using the Welch t-test as a filter and then GA to choose the subsets of features, combining mutual information and classification models to predict the outcome of cancer data. The main innovation was the use of the C-Mantec (Competitive Majority Network Trained by Error Correction) algorithm as a classifier, a constructive neural network model.

Gupta et al. (2015) used GA to evolve neural networks for the task of breast cancer diagnosis, combining it with backpropagation to perform local search. Garro, Rodríguez and Vazquez (2017) focused on microarray classification by designing a strategy that first uses Artificial Bee Colony (ABC) optimization for FS, and then creates an ANN through Differential Evolution (DE), an algorithm akin to GA, to be used as classifier. Withal these works used ideas from Neuroevolution for building classifiers for microarray data, without incorporating the gene selection as part of the evolutive process.

Despite not being directly related to these works, it is also worth mentioning that NEAT has already been employed in the creation of artificial gene regulatory networks (CUSSAT-BLANC; HARRINGTON; POLLACK, 2015).

3.6 Chapter conclusion

This chapter started with a brief overview of gene expression, to then explain the process of a microarray experiment and how the expression data of thousands of genes can be analyzed by several algorithms. The use of machine learning for the creation of classifiers useful as diagnostic tools and the importance of FS in order to find informative genes were discussed to a greater extent. At the end, some applications of Neuroevolution in these tasks were presented. The next chapter describes how microarray data can be obtained and preprocessed before further analysis.

4 DATA OBTAINMENT AND PREPROCESSING

Before describing the proposed method itself in Chapter 5, it is important to take a moment to explain how the microarray data for this research was obtained and manipulated, in order to provide a diversity of test cases for the computational method while also ensuring a high quality control with a rigorous filtering and preprocessing pipeline. For this work the focus was on human cancer experiments, due to their relevance and to the abundance of public available data. The literature also contains vast material on cancer studies, allowing further validation of the results obtained in Chapter 6 by comparing them with finds in published experiments. In order to provide more diversity in the datasets, three cancer types were chosen as targets: breast, colorectal, and leukemia. All the major steps in this and the following chapters are illustrated in Fig. 4.1, that summarizes the whole method.

4.1 Data obtainment

To obtain multiple microarray datasets (GSEs), the raw data of leukemia, breast, and colorectal cancers were downloaded from the GEO (Gene Expression Omnibus[1]) database using the *GEOquery* package (DAVIS; MELTZER, 2007) for the R platform[2] (Fig. 4.1 - *Data Obtainment*). GEO is an international public repository for high-throughput functional genomics data, including different types of microarrays.

With the aim of selecting the most homogeneous and reliable datasets (Fig. 4.1 - *Preprocessing*), several criteria were adopted:

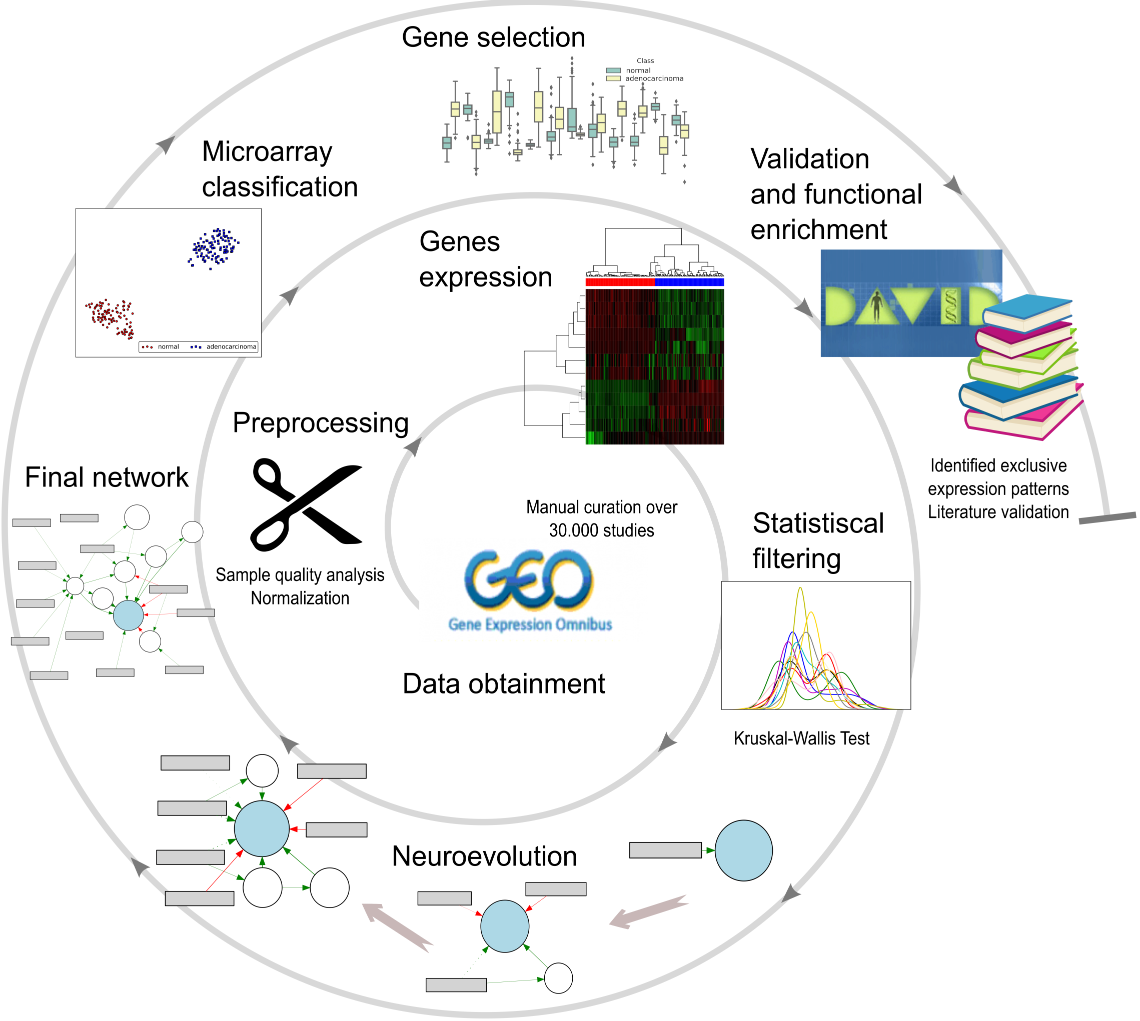
1. Exclusion of studies that used chemotherapics, any kind of gene therapies, or that employed interfering molecules as miRNA and small interfering RNA (siRNA), since those molecules are usually employed to impair the given expression of target genes - thus, altering the expression profile of the chosen experiment;

2. Exclusion of studies that used any kind of xenograft technique. Xenograft studies, in this case, are those in which the human tumoral tissues are transplanted into another organism, usually mice or rat models, to evaluate growth and effect patterns. Hence, they were excluded to assure that no bias from another organism biochemi-

[1] <https://www.ncbi.nlm.nih.gov/geo/>

[2] <[www.r-project.org]>

Figure 4.1: **Summary of the methodological steps taken in this work.** After data obtainment, the microarray datasets were normalized and the low quality samples were excluded. The genes were filtered using the Kruskal-Wallis H Test (Section 2.1) and the remaining data was employed in the Neuroevolution process, from which the best neural network was chosen. Finally, the neural networks were used to perform the microarray classification, and its inputs used for gene selection. The final selected genes, which represent distinct expression patterns, were submitted to a functional enrichment analysis. Additionally, an extensive search in the scientific literature was conducted to see the types of cancer that the selected genes were associated to.



cal profile could alter the results;

3. Exclusion of microarrays that used any form of Knockdown (KO) cultures, or specifically selected mutations. KO cultures are those that delete a specific gene from the genome, so the lack of such molecule could be studied. Therefore, it is essential to excluded such cases, because the lack of a gene will alter the expression profile of a given microarray;

4. Selection of studies performed exclusively on *Homo sapiens*;

5. Selection of datasets only with at least six normal (control) samples and six experimental (tumoral) samples. This was a technical decision needed for the correct use of the cross-validation described in Section 6.1.2;

6. Selection of studies with a clear description of the protocols followed in the experiments. It is common for some studies to be uploaded to the GEO database without proper description, since GEO is a free public database, with no clear control of uploads. These studies were ignored because their origins could be considered doubtful.

Besides that, only data generated with microarray chips from the company Affymetrix[3] were selected, with the goal of keeping the data as consistent as possible. Among the major companies that offer microarray technologies, like Illumina[4] and Agilent[5], Affymetrix has the most standardized probe names and raw data, enhancing the reliability of the subsequent analyzes. In addition to the selected GSEs, the original microarray dataset from Golub et al. (1999)[6] with AML and ALL leukemia subtypes was included in order to provide a comparison with other methods in the literature.

4.2 Preprocessing steps

After data obtainment, background correction and Robust Multichip Average (RMA) normalization of all selected GSEs were performed by the R package *affy* (GAUTIER et al., 2004). After normalization, datasets were analyzed by the R package *arrayQualityMetrics* (KAUFFMANN; GENTLEMAN; HUBER, 2009), to access the sample quality of the selected microarrays. Samples that displayed low quality in at least half of any parameters measured by *arrayQualityMetrics* were discarded from the final pool. Table 4.1 summarizes the chosen GSEs, their specifications, and the number of excluded samples.

The normality of the distribution of the genes expression in the datasets listed in Table 4.1 was tested with the D'Agostino and Pearson's test that combines skew and kurtosis (D'AGOSTINO, 1971; D'AGOSTINO; PEARSON, 1973). It was observed that most of the genes did not follow a normal distribution with p-value < 0.01.

[3] <http://www.affymetrix.com/analysis/index.affx>

[4] <https://www.illumina.com/>

[5] <https://www.agilent.com/>

[6] <https://github.com/ramhiser/datamicroarray>

Table 4.1: **List of GSEs and datasets employed in this work.**

Datasets	Cancer Type	Samples	Excluded Samples*	Genes	Classes
GSE42568	Breast	121	5	54675	2
GSE45827	Breast	155	4	54675	6
GSE10797	Breast	66	None	22277	3
GSE44076	Colorectal	246	52	49386	2
GSE44861	Colorectal	111	6	22277	2
GSE8671	Colorectal	64	1	54675	2
GSE21510	Colorectal	148	105	54675	2
GSE32323	Colorectal	44	11	54675	2
GSE41328	Colorectal	20	2	54675	2
GSE9476	Leukemia	64	None	22283	5
GSE14317	Leukemia	26	1	22277	2
GSE63270	Leukemia	104	3	54675	2
GSE71935	Leukemia	51	6	54675	2
Golub et al. (1999)	Leukemia	72	NA	7129	2

*Includes: (i) samples excluded prior to the analysis, due to the presence of one or more samples that didn't met the criteria described on Section 4.1; (ii) samples that could generate a bias in the analysis due to treatment, tissue origin or platform mix; (iii) file corruption and errors; and (iv) samples excluded due to low quality. NA = Not Applicable.

4.3 One-vs-All classification

Another consideration about the creation of the datasets to be used by the method described in the next chapter is that it uses One-vs-All classification for multiclass classification problems. This means that if a dataset has more than two classes, as for example the dataset GSE45827 described in Table 4.1, each class is classified separately against all the other classes combined. While most ANNs can handle multiclass data without problems, including the proposed method, the One-vs-All approach was chosen due to four assumptions:

- Most of the selected microarray datasets are binary (Table 4.1 - *Classes*).
- One-vs-All allows the use of only one output neuron in each ANN, simplifying their structures and the evolutionary search.
- It makes easier to interpret the FS results since for each subset of selected features only one class was considered against the remainder, so these features are responsible from the classification of that particular class.
- The major drawback of One-vs-All classification is the creation of size imbalance among classes, but for many microarray experiments the data is already imbalanced

and, in fact, some times becomes more balanced with the splits created with One-vs-All.

Taking this in consideration, the datasets GSE45827, GSE10797, and GSE9476 from Table 4.1 are transformed in six, three, and five datasets, respectively, as shown in Table 4.2.

Table 4.2: **List of classes division for each dataset employed in this work, considering One-vs-All classification.**

	Class A		**Class B**	
Datasets	**Type**	**Samples**	**Type**	**Samples**
GSE42568	Tumoral	101	Normal	15
GSE45827	Basal	41	Remainder	110
	HER	30	Remainder	121
	Cell Line	14	Remainder	137
	Luminal A	29	Remainder	122
	Luminal B	30	Remainder	121
	Normal	7	Remainder	144
GSE10797	Cancer Epithelial	28	Remainder	38
	Cancer Stroma	28	Remainder	38
	Normal	10	Remainder	56
GSE44076	Adenocarcinoma	97	Normal	97
GSE44861	Tumoral	52	Normal	53
GSE8671	Adenoma	31	Normal	32
GSE21510	Tumoral	18	Normal	25
GSE32323	Tumoral	16	Normal	17
GSE41328	Tumoral	8	Normal	10
GSE9476	AML	26	Remainder	38
	Bone Marrow	10	Remainder	54
	Bone Marrow CD34	8	Remainder	56
	PB	10	Remainder	54
	PBSC CD34	10	Remainder	54
GSE14317	ATL	18	Normal	7
GSE63270	AML	60	Normal	41
GSE71935	JMLL	37	Normal	9
Golub et al. (1999)	AML	47	ALL	25

Class A = Class being discriminated if the original dataset has more than two classes; Remainder = union of samples of all classes except the discriminated class if the original dataset has more than two classes; Normal = control group; CRC = Colorectal Cancer; AML = Acute Myeloid Leukemia; ALL = Acute lymphoblastic leukemia; ATL = Adult T-Cell Leukemia/Lymphoma; JMML = Juvenile myelomonocytic Leukemia; HER = Breast Cancer - HER Status; PB = Peripheral blood; PBSC = Peripheral Blood Stem Cell; CD34 = Cluster of Differentiation 34.

4.4 Chapter conclusion

This chapter described the criteria for choosing the data presented in this work, as well as the steps it goes through before it can be analyzed. The resultant expression matrices are then used as inputs for the proposed computational method (Fig. 4.1 - *Genes Expression*), as explained in the next chapter.

5 PROPOSED METHOD

After the steps listed in the last chapter, the data is ready for the main analysis. In this chapter, the main computational method is described with the aim of tackling the tasks described in Chapter 1. It is a hybrid approach to gene selection (Section 3.4), combining filtering and embedded algorithms, based in the Kruskal-Wallis H Test (Section 2.1) and FS-NEAT (Section 2.5.2), and capable of autonomously performing the tasks of microarray classification (Fig. 4.1 - *Microarray Classification*) and gene selection (Fig. 4.1 - *Gene Selection*), without the need for specifying how many genes should be selected at the end.

5.1 Filtering and preprocessing

Due to the presence of thousands of genes in each microarray dataset (the smallest one in our list has 7129 genes), before starting the evolutive process, the data is filtered using the Kruskal-Wallis H Test (Section 2.1). This is achieved by comparing the expression of each gene among the two classes (One vs. All classification - Section 4.3) and removing all genes that presented no difference between the two classes (p-value ≥ 0.01) (Fig. 4.1 - *Statistical Filtering*). The Kruskal-Wallis H Test is nonparametric and does not assume a normal distribution, what is in agreement with the normality analysis result described in Section 4.2. The Kruskal-Wallis H Test has already been used in the study microarray data (LAN; VUCETIC, 2011), and the use of statistical methods as a preprocessing filtering step is standard practice in the literature (LEUNG; CAVALIERI, 2003; LUQUE-BAENA et al., 2013).

After the application of the Kruskal-Wallis H Test, around 13% of the total amount of genes is kept for the next steps. The final preprocessing step is to normalize the expression of the genes, using the mean normalization as described in Equation 5.1, with x being a feature, and μ, x_{max}, and x_{min} being the mean, maximum value and minimum value of that feature over all the samples, respectively. Each feature is normalized independently.

$$x_{new} = \frac{x - \mu}{x_{max} - x_{min}} \tag{5.1}$$

5.2 Training

The next step is the Neuroevolution itself (Fig.4.1 - *Neuroevolution*). The output of the networks is a value between 0 and 1 that predicts to which class a sample belongs, and the inputs are the normalized values of the expression of the genes. The population of the first generation is created by connecting one random input to the output for each individual, and the initial weights and biases are randomly determined from a normal distribution with mean equal to zero and standard deviation equal to one. Since the algorithm deals with higher dimensions than usually used with FS-NEAT, it was modified to better explore the input space with the inclusion of three new operators (N3O):

Additive crossover operator: it works similarly to the NEAT crossover operator (Fig. 2.4), but if the parent with lower fitness has an input that the parent with better fitness does not possess, and this input is connected to a node present in the parent with better fitness, there is a 50% chance of the offspring inheriting that input (Fig. 5.1).

Swap input mutation: a new structural mutation operator that randomly swaps one of the network inputs by another input not present in the ANN (Fig. 5.2 - *Swap input*).

Guided add input mutation: the p-values from the Kruskal-Wallis H Test described in Section 5.1 are transformed by the formula $-\log_{10}(p)$ and scaled by the softmax function (Equation 5.2, with Z being a vector of probabilities z). The outputs are probabilities that are larger for smaller p-values. They are used as the probability of an input being selected by the "add input mutation", meaning that the genes that showed the largest difference between classes are more likely to be selected by the mutation (Fig. 5.2 - *Guided add input*).

$$softmax(Z) = \mathrm{e}^{z} \div \sum_{z' \in Z} \mathrm{e}^{z'}, \forall z \in Z \tag{5.2}$$

The additive crossover operator (Fig. 5.1) substitutes the original crossover operator used by NEAT and FS-NEAT (Fig. 2.4). This change allows the combination and integration of the features selected by two ANNs, what is not permitted by the original crossover, since the offspring will always have the same FS as the parent with better fitness. The original NEAT does not suffer from this because all the inputs are always connected to the outputs. In the case of a FS algorithm dealing with high dimensional data, however, it should be beneficial to combine possible good selections.

Figure 5.1: **The proposed crossover operator**. Given two parents, red (better fitness) and blue (lower fitness), the offspring ANN will be a combination of the two, inheriting the structures from both randomly when both have it, and from red otherwise. The major difference from FS-NEAT is that if there is an input in blue that is not connected to red, and this input in blue is connected to a node that is in red, the offspring has fifty percent of chance of inheriting it as well, here represented by input "D".

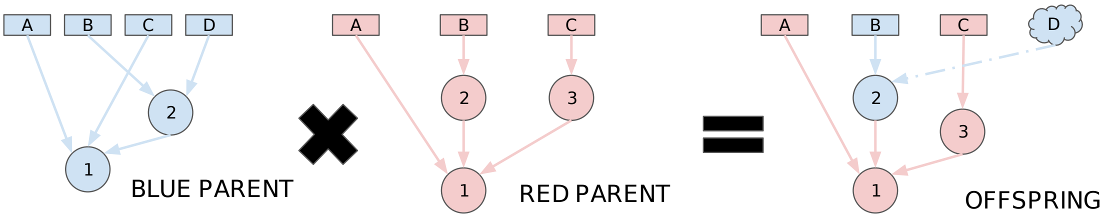
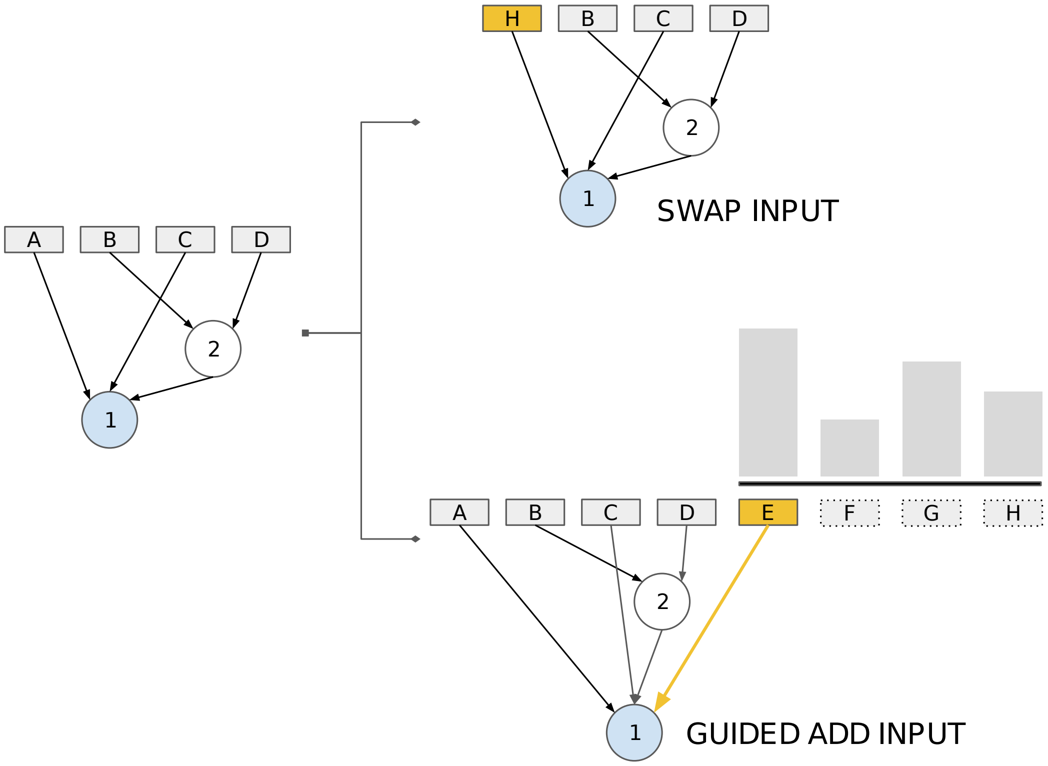


Figure 5.2: **The two new possible structural mutations for the proposed method.** Rectangles represent inputs, blue circles indicates outputs, white circles represent hidden nodes, and arrows are the connections between nodes. The new structures are marked in gold. The histogram above the network that had an added input represents the probabilities of each new input being selected by this operator.



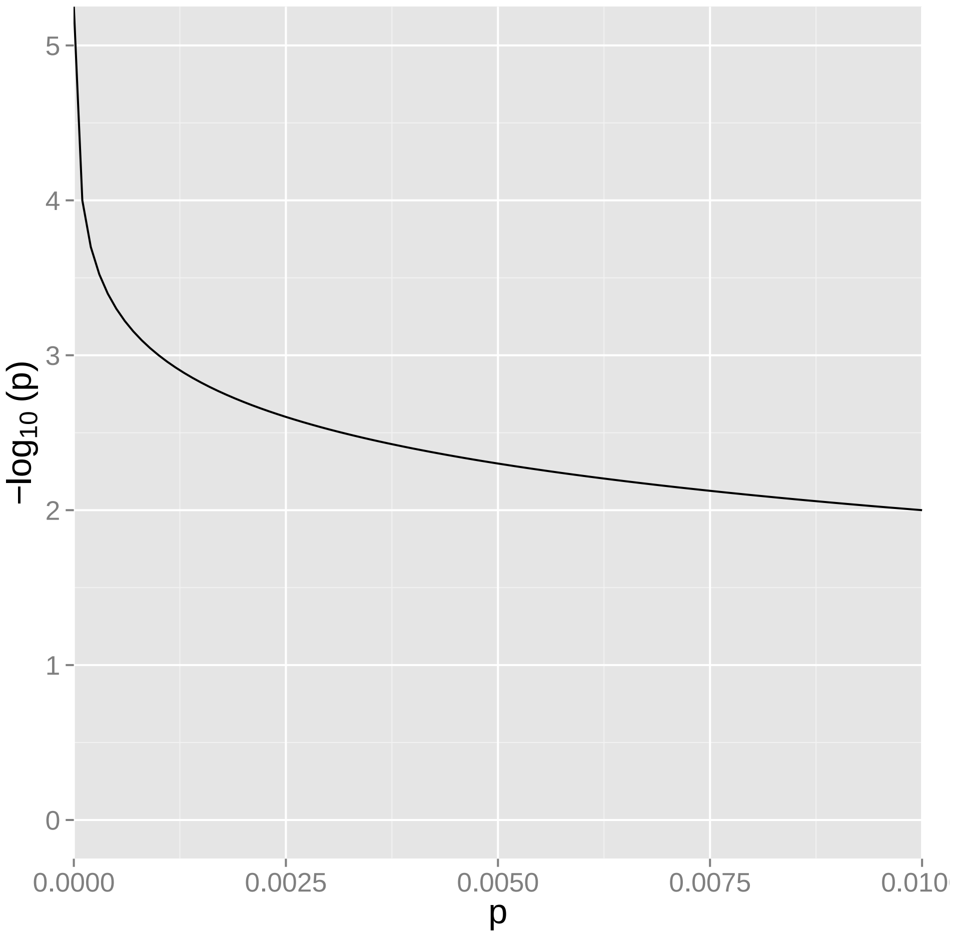
Also motivated by the goal of better exploring the input search space, the swap input mutation (Fig. 5.2 - *Swap input*) was added. This mutation allows the algorithm to explore the use of new possible features without increasing the ANNs complexity or the number of features selected, while also exploiting the already existing network structure.

Finally, it was considered the high cost associated with randomly searching all the input space to find new better features when dealing with thousands of inputs. Thus, the mutation from FS-NEAT that simply added an input to a network (Fig. 2.6) was modified to take advantage of the already computed ranking of genes created when the Kruskal-Wallis H Test was applied. The results from the test are a p-value for each gene indicating which genes have expressions less likely to differ between the classes only by chance (LAZAR et al., 2012). This value can be used to guide the add input mutation (Fig. 5.2 - *Guided add input*), making more likely to select the genes with lowest p-values. This probability of being selected is determined by using the softmax function, which receives as input a vector of values and outputs a vector of the same length, whose sum is equal to one and has only positive values proportional to their inputs. The input of the softmax function, in this case, is the vector of $-\log_{10}$ of the p-values, so that smaller p-values become larger probabilities, but the distribution is smoother to not bias the selection too much towards the genes with smallest p-values, still allowing further exploration. The transformation of the p-values is illustrated in Fig. 5.3.

The fitness function that guides the evolutive process is the cross-entropy, also known as the log loss (GOODFELLOW et al., 2016; BOER et al., 2005), in its binary form (for two classes classification). This is a popular cost function for supervised learning but does not account for data imbalance, which is common in microarray data. Thus, it was altered so that it is computed individually for each class q and then averaged, as shown in Equation 5.3a, in which n^q is the number of samples in the class q, y_i is the true label of the i^{th} sample, and a_i is the ANN output for the i^{th} sample. This way, all classes have the same importance independently of their sizes.

The second term of the fitness function, given by Equation 5.3b, stands for the L2 regularization described in Section 2.4. The L2 regularization penalizes networks with large absolute weights and biases values, under the assumption that simpler models are better in generalizing. As can be observed, however, Equation 5.3b differs from the original expression in Equation 2.5. Since the number of inputs of a neuron can change during the evolution, the term $\frac{1}{c}$ was added, so that the regularization would not have a negative impact in the addition of new connections and nodes. The c is the number of connections and biases, n is the number of samples, w_k is the weight or bias of the connection or node k, and λ is the regularization parameter. Due to the minimalist start of FS-NEAT, this method does not require a component to minimize the number of features selected, as used in Luque-Baena et al. (2013) for instance, making for easier fitness

Figure 5.3: **The p-values $-\log_{10}$ transformation.** The p-values obtained from the Kruskal-Wallis H Test are transformed by this operation, in order to be used in the softmax function (Equation 5.2). All the p-values are between 0 and 0.01, since this step is after the statistical filtering.



function design. The fitness also does not account for the redundancy of the selected features, under the assumption that two genes with highly correlated expressions that are deemed relevant by the network should both be returned as the solution.

$$fitness = \frac{1}{|Q|}\sum_{q \in Q}\left\{-\frac{1}{n^q}\sum_{i=1}^{n^q}[y_i \ln a_i + (1-y_i)\ln(1-a_i)]\right\} \tag{5.3a}$$

$$+\frac{\lambda}{2n}\frac{1}{c}\sum_{k=1}^{c} w_k^2 \tag{5.3b}$$

The fitness is a numerical value that measures the error between the network output and the true sample label, proportionally penalized by large network weights and biases absolute values. The output of Equation 5.3 will always be positive, so the optimization should be a minimization problem (lowest $fitness$ is the best). Because GA usually deals with maximization problems, it is also possible to maximize the expression $fitness' = -fitness$ (largest $fitness'$ is the best). Either way, the best theoretical fitness value would be zero, but it would require an ANN with all weights and biases also equal to zero, thus making achieving this threshold impossible.

The structure of the neurons in this method also differs from a traditional artificial neuron as showed in Equation 2.4. The modified neuron is described by Equation 5.4, in which a_h is the output, m_h is the number of inputs, b_h is the bias, w_{hj} is the weight of the j^{th} input, and x_{hj} is the j^{th} input of the neuron h, respectively. The aggregation is the mean of the inputs instead of the summation because the number of inputs can vary during the evolution, thus the term $\frac{1}{m_h}$. The Φ stands for the activation function of the neuron, that for the output neuron is the modified Gaussian function (Equation 2.8), and for all the hidden nodes is the modified tanh (Equation 2.7). These two functions combined have shown the best performance in the context of FS-NEAT in comparative studies (PAPAVASILEIOU; JANSEN, 2017a), as discussed in Section 2.5.2.

$$a_h = \Phi\left(\frac{1}{m_h}\sum_{j=1}^{m_h} w_{hj}x_{hj} + b_h\right) \tag{5.4}$$

The GA that evolves the neural networks uses the new operators presented before (Fig. 5.2 and Fig. 5.1), in addition to the listed modifications in the fitness function and neurons structure. The selection for crossover uses tournament, and elitism is adopted to preserve the best individuals from each generation (Section 2.3). The used hyperparameters are listed in Table 5.1 and were chosen based on experimental results and literature revision (PAPAVASILEIOU; JANSEN, 2017a). The selected genes are the subset of features (inputs) directly or indirectly connected to the output node in the neural network with the best fitness at the end of the algorithm.

5.3 Chapter conclusion

In this chapter the neuroevolutive method proposed in this work is presented and explained. First, the data is filtered with the Kruskal-Wallis H Test, then normalized, and finally used in the creation of an ANN with a modified FS-NEAT algorithm that has three new structural operators for better exploring the feature space. The fitness function and artificial neuron were also designed taking in consideration the particularities of the tasks of gene selection and microarray classification. Experiments with this method and their

Table 5.1: **List of used hyperparameters.**

Hyperparameter	**Value**
Population size	1000
Number of generations	100
Aggregation function^1	mean
Activation function^1	tanh, Gaussian
L2 regularization λ^2	0.5
Probability of mutation adding input^3	0.05
Probability of mutation swapping input^3	0.05
Probability of mutation adding connection^4	0.05
Probability of mutation adding node^4	0.03
Probability of mutation changing weight	0.04
Elitism proportion	0.1
k tournament selection	2
Coefficient 1^5	1.0
Coefficient 2^5	1.0
Coefficient 3^5	0.4
Compatibility threshold^5	3.0

^{1} Equation 5.4, ^{2} Equation 5.3b, ^{3} Fig. 5.2, ^{4} Fig. 2.3b, ^{5} Equation 2.6

results are described in the next chapter.

6 EXPERIMENTS AND RESULTS

This chapter aim is to test and discuss the results of applying the aforementioned method to the data from Chapter 4. It starts by briefly introducing some necessary concepts for better understating the experiments, and then illustrates the evolution of a single population of ANNs. In further experiments, the accuracy and FS of the method are analyzed, and the results are compared with other strategies. Finally, the set of selected genes is biologically reviewed.

6.1 Introduction

This section introduces some concepts and tools used for the experiments.

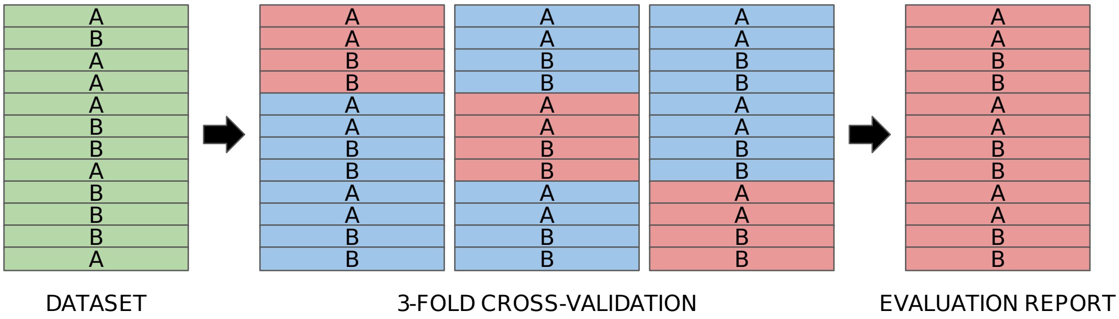
6.1.1 A proof of concept

In a previous work, developed during the time of this research, we already tested the use of FS-NEAT in the tasks of microarray classification and gene selection. While this early work did not use most of the improvements related in Chapters 4 and 5, nor was tested in the same datasets, its results were the starting point for the creation of this new approach, and can be read, in conference paper format, in Appendix B.

6.1.2 Cross-validation

To validate the accuracy of classification and number of features selected by the algorithms, k-fold cross-validation was used. Cross-validation is an efficient and unbiased error estimator, and the most common validation method for microarray datasets (ANG et al., 2016), as a way to check the generalizability of the model. For our experiments, stratified 3-fold cross-validation was adopted (illustrated in Fig. 6.1), so each dataset was split into three random folds (or partitions) of equal size, and each fold preserves the same sample per class ratio of the whole dataset. At each iteration of the cross-validation, one of the folds is used as a testing set, and the remaining folds are used for the training of the algorithm. At the end of the iteration, the algorithm the accuracy of the algorithm in the samples of the testing fold (that it has never seen) is computed. The final accuracy is the weighted average accuracy of the testing folds. The choice of stratified 3-fold cross-validation was made based on the minimum number of samples in each class of the datasets in Table 4.1, the high imbalance in class size, and computational time.

Figure 6.1: **Stratified 3-fold cross-validation.** This example has a dataset with two classes, "A" and "B", of equal sizes, and twelve samples, represented in green. During the cross-validation, this dataset is split in three folds of the same size and each containing the same number of samples from each class. At each iteration, the algorithm being evaluated is trained with the two training folds (in blue) and then tested with the testing fold (in red). This process is repeated for each fold, so at the end, the algorithm was trained and tested three times. The reported performance is the combination of the accuracies at each testing fold.



It is important to note that for each iteration of the cross-validation, it is necessary to repeat all the training steps needed by the algorithm using only the samples from the training folds, including the filtering and normalization of the data. This is fundamental to avoid the introduction of bias from the testing samples into the training, what would impact the results validity.

6.1.3 Computing the baseline

When dealing with classification problems, it is important to know beforehand the baseline accuracy of the datasets, defined here as the expected accuracy from a classifier that always naively predicts that a new sample belongs to the larger class, sometimes also referred as ZeroR[1]. In a binary classification problem with well-balanced classes, the baseline accuracy will be 50%, but this can drastically change for ill-balanced datasets. Equation 6.1 defines the formula used for computing the baseline accuracy for a dataset D with two classes, A and B, $|D_A|$ being the number of samples from D belonging to

[1] <http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/ZeroR.html>

class A.

$$baseline(D) = \max(\frac{|D_A|}{|D|}, \frac{|D_B|}{|D|}) \tag{6.1}$$

If the model accuracy is no better than the baseline, some problems may exist within the algorithm design, implementation, or training. It may be, for instance, just randomly guessing or suffering from overfitting (Section 2.4).

It is also needed to define a baseline of sorts for the FS. In this case, the criteria of comparison is the probability of a gene being randomly selected by the final neural network considering uniform distribution, meaning how likely it is for a gene to be chosen without the need of any optimization. Of course, since each gene can only be selected exactly one time per network, the probability of gene g being selected at random is $\frac{1}{G}$, G being the total number of genes in the dataset. However, we should account for the fact that a single neural network can select more than one gene, and that for multiple experiments of k-fold cross-validation (with the same dataset) several "final" neural networks will be created, one for each run. The total number A of ANNs resulting from an experiment of r runs with k folds each is $A = r \times k$. Each of these A ANNs can have a different number of selected genes, so we average the number of inputs in each ANNs, equal to m. Thus, considering a gene g that was selected s times, i.e., that g appeared as input in s of A ANNs, it is possible to approximate the probability of g being randomly selected at least s times as the binomial distribution in Equation 6.2.

$$p = \frac{m}{G} \tag{6.2a}$$

$$\binom{A}{s} = \frac{A!}{s!(A-s)!} \tag{6.2b}$$

$$P_g[X = s] = \binom{A}{s} p^s (1-p)^{A-s} \tag{6.2c}$$

$$P_g[X \geq s] = 1 - (P_g[X = 0] + P_g[X = 1] + ... + P_g[X = s-1]) \tag{6.2d}$$

6.1.4 Functional enrichment analysis

The final FS obtained by using the proposed method does not deal with the genes separately, like some of the ranking filtering algorithms described in Section 3.4, but as a set of inputs all needed for the correct performance of its corresponding neural network. Thus, an analysis of the classes of found genes needs to focus on the whole set, and not on individual genes. Functional enrichment analysis also referred to as Gene Set Enrichment Analysis (GSEA), is an analytical technique devised for this purpose. It defines the sets of genes based on previously published biological knowledge and determines whether the genes in the set are correlated to specific phenotypic class distinctions (SUBRAMANIAN et al., 2005).

The Database for Annotation, Visualization and Integrated Discovery (DAVID) v 6.8[2] (HUANG; SHERMAN; LEMPICKI, 2009b; HUANG; SHERMAN; LEMPICKI, 2009a) was used to discover the most relevant bioprocesses and to trace the nature of the selected genes from the employed datasets (Fig. 4.1 - *Validation and Functional Enrichment*). The entire list of selected genes was used as input in DAVID, using the Benjamini FDR correction (BENJAMINI; HOCHBERG, 1995) with a significance score of 0.05. The bioprocesses came from Gene Ontology (GO), that provides a unified and structured vocabulary and annotation for genes and gene products (CONSORTIUM, 2007).

6.1.5 Computational resources

All experiments reported in this work ran in an Intel Xeon E5-2650V4 30 MB, 4 CPUs, 2.2Ghz, 48 cores/threads, 128GB, 4TB. The main code was written in the Python 2.7 programming language, with the use of some methods from *Scikit-learn* (PEDREGOSA et al., 2011) and *NEAT-Python*[3] libraries. Some of the analyses were made with the R programming language, including the creation of gene expression heatmaps with the *heatmap.2* method using the correlation as distance function. All analyses of statistical significance were performed with the Kruskal-Wallis posthoc test after Dunn with Bonferroni-type adjustment of p-values from the *PMCMR*[4] R package.

[2] <https://david.ncifcrf.gov/>

[3] <http://neat-python.readthedocs.io/en/latest/>

[4] <https://www.rdocumentation.org/packages/PMCMR/versions/4.3>

6.2 Visualizing the Neuroevolution

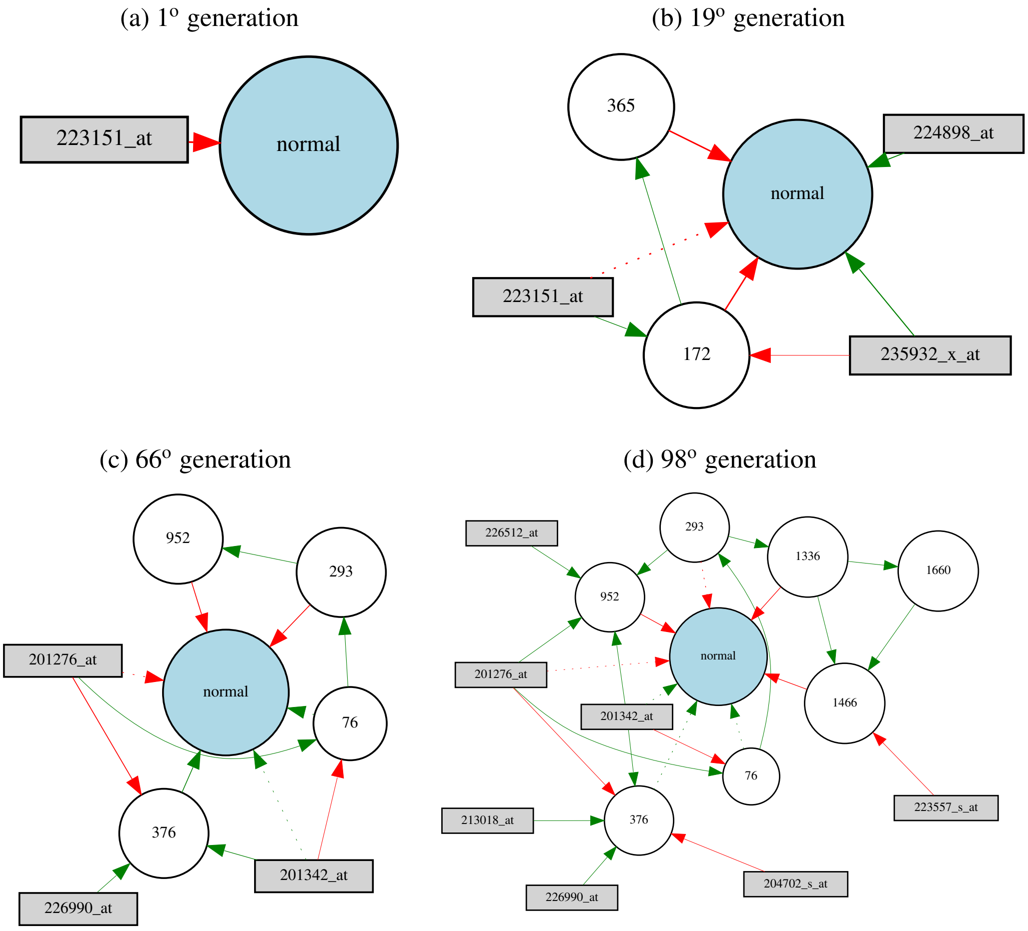
The first experiment is an introduction to better illustrate the working of the proposed method, from now on referred as N3O due to the three new operators already discussed, but also considering all the steps and details described in Chapter 5. When not stated otherwise, the hyperparameters values are the ones from the Table 5.1. The dataset GSE32323 of colorectal cancer was chosen as an example for this experiment, and its details can be found at Tables 4.1 and 4.2.

One single run of the method was performed for dataset GSE32323, using 60% of the samples for the training set and 40% for the testing set. This specific run reported 100% accuracy on both sets (meaning it correctly classified all samples in the training and testing sets) and all images in this section refer to this specific experiment. Fig. 6.2 brings neural networks in four distinct stages in the neuroevolutive process, as a way to better show the growth and complexification of the solutions. These four stages were chosen to show the evolution in a roughly equal time distribution over 100 generations, but also taking into consideration the structural difference between them. As described in Section 2.5.2, all networks in the population are created as Fig. 6.2a, with one random input connected to the output, and from there they grow. These four networks do not necessarily belong to a single line of hereditary, because they can have distinct ancestors. They all were, however, the individual with the best fitness in the population at their respective generations.

From the different examples of network topologies in Fig. 6.2 and other neural networks figures from the next sections, an observation that can be made is that N3O found ANN architectures distinct from traditional MLP models, unlikely to be designed by programmers. Many inputs are directly connected to the output, and the algorithm makes use of gates akin to Highway Networks, usually employed to improve the learning of very deep neural networks allowing information to flow between layers unrestricted (SRIVASTAVA; GREFF; SCHMIDHUBER, 2015).

There are other aspects of the evolution of ANNs with N3O that are worth checking. Fig. 6.3a shows the convergence of the fitness of the best individual in the population, as it approaches the theoretical limit of zero. Since the algorithm employs elitism it is impossible for the fitness curve to regress. Note that in this chart, the fitness is represented as a negative value being maximized, as discussed in Section 5.2. Fig. 6.3b portrays the behavior of the population in regard of speciation, discussed in Section 2.5.1. In this case,

Figure 6.2: **Stages in the evolution of neural networks.** The neural networks with best fitness in the population at generations (a) 1, (b) 19, (c) 66, and (d) 98 for a run of the proposed method with the dataset GSE32323 colorectal cancer. The best networks gradually grow in size, including in number of selected features (genes). Grey rectangles are input nodes, white circles are hidden nodes, and blue circles are output nodes. The number inside the hidden nodes inform the order in which they were created. Arrows are green if they are connections with positive weight, or they are red otherwise. Their thickness is proportional to the absolute values of their weights. Dotted arrows are disabled connections.


(a) 1^{o} generation

(b) 19^{o} generation

(c) 66^{o} generation

(d) 98^{o} generation

the individuals diversified into four distinct species (defined by Equation 2.6) around the 20^{o} generation, whose sizes (number of individuals) stayed stable to the end. As the evolution progresses it becomes harder for a solution to become so distant from the others that it creates a new species, but at the same time the existence of distinct species maintains diversity in the population.

Regarding the gene selection, Fig. 6.3c shows the spread of the genes in the population. As would be expected, at the beginning of the evolution the total number of genes present in at least one individual (the corresponding input is connected to the output) is roughly the size of the population, since one random input is assigned to each individual. Through crossover and mutation, however, the genes selected by the individuals with best fitness spread in the population. It is also possible to see how the preference for certain genes shift during the evolution, how some appear late in the process, and how some become extinct. Considering the high dimensionality of the data, this flow in the genes presence is important as a way to keep the algorithm exploring new solutions and avoiding an early stagnation.

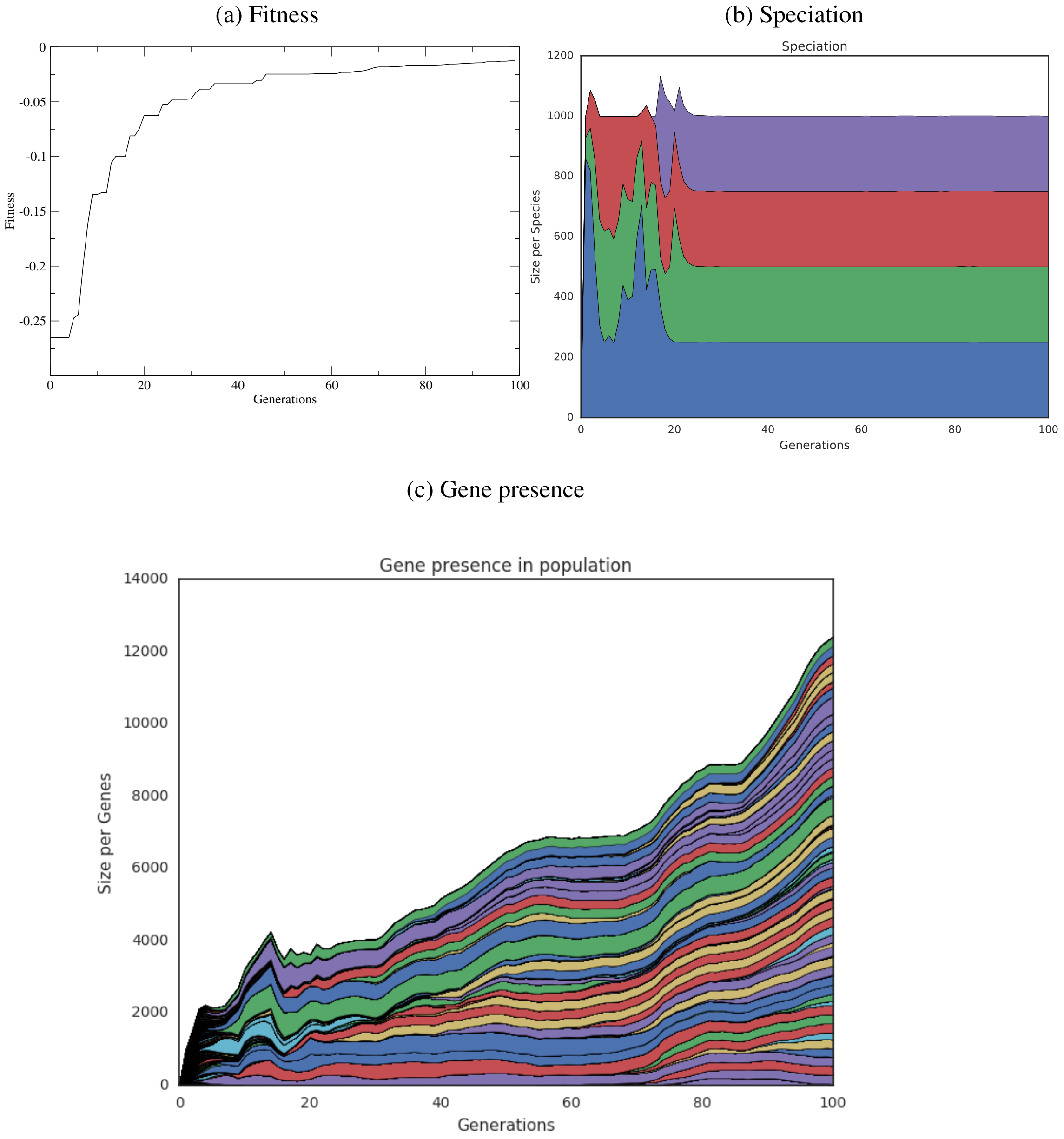
The chart in Fig. 6.4 offers another view of the changes in gene selection over the generations. Looking at the bottom part of the chart, it is possible to see that at the beginning the gene selection was more widespread over the population, with all genes having approximately the same distribution (randomly chosen inputs). Towards the end of the evolution, however, the density has shifted to the left. This is an illustration of the effect of the guided input mutation, which increases the chances of genes with smaller p-values (in the left side of the chart) being selected.

The orange triangles mark which genes were selected by the individual with the best fitness at each generation. During the first generations, only one gene was selected, corresponding to the ANN from Fig. 6.2a. The row of the 100$^{\text{o}}$ generation has six marked genes, corresponding to the six inputs in the ANN from Fig. 6.2d. Once again, it is possible to see that the configuration of the best individual shifts from time to time, most likely due to changes in its weights and biases values, the addition of new nodes or connections, and the interaction between the inputs. From this it is also visible that the selection performed by N3O is not equivalent as just selecting the top-ranked features after filtering with the Kruskal-Wallis H Test, otherwise, at the final generation only the left-most genes would have been selected. The horizontal spread of genes belonging to the best individual suggests that statistical measurements and attributes detectable by filtering methods are not the only factors that should be considered, what is in agreement with the literature (ANG et al., 2016).

Like Fig. 6.3c, this chart also allows the visualization of the selection flow. Tracking the colors of the dots one can see when certain genes started to increase their presence among individuals, or when they vanished from the population. Interestingly, the stochastic nature of the selection allows some genes that have been extinct in the population to come back at a later time, and combined with other genes they can increase their presence.

Finally, Fig. 6.5 overlaps the regularized error (the minimization version of maximizing the fitness) with the number of genes in the population. Both values seem to converge roughly together, around the 50$^{\text{o}}$ generation, but this did not stop the method

Figure 6.3: **Different aspects of the evolution.** For the same experiment shown in Fig. 6.2 with the dataset GSE32323 (colorectal cancer). (a) Convergence of the fitness as the maximization problem described in Section 5.2. (b) Speciation in population, as described in Section 2.5.1. Each color strip corresponds to one species in the population, the height being the number of individuals belonging to it. (c) The presence of genes (features) in the population over the generations. A gene is considered present in the population in a given generation if it is selected by at least one neural network. Each strip corresponds to one gene (colors may be repeated), the height being the number of neural networks selecting the gene.



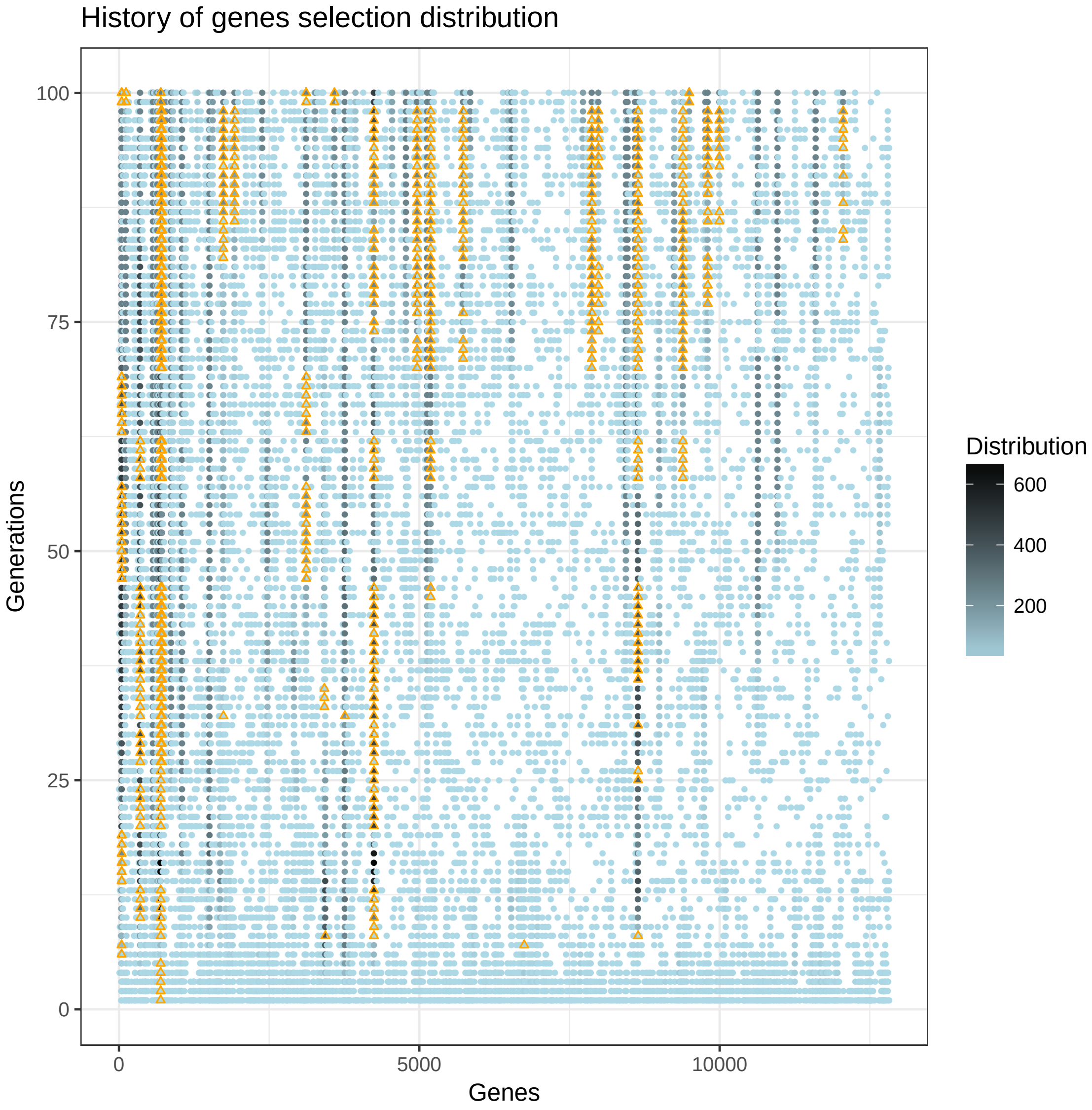
(a) Fitness

(b) Speciation

(c) Gene presence

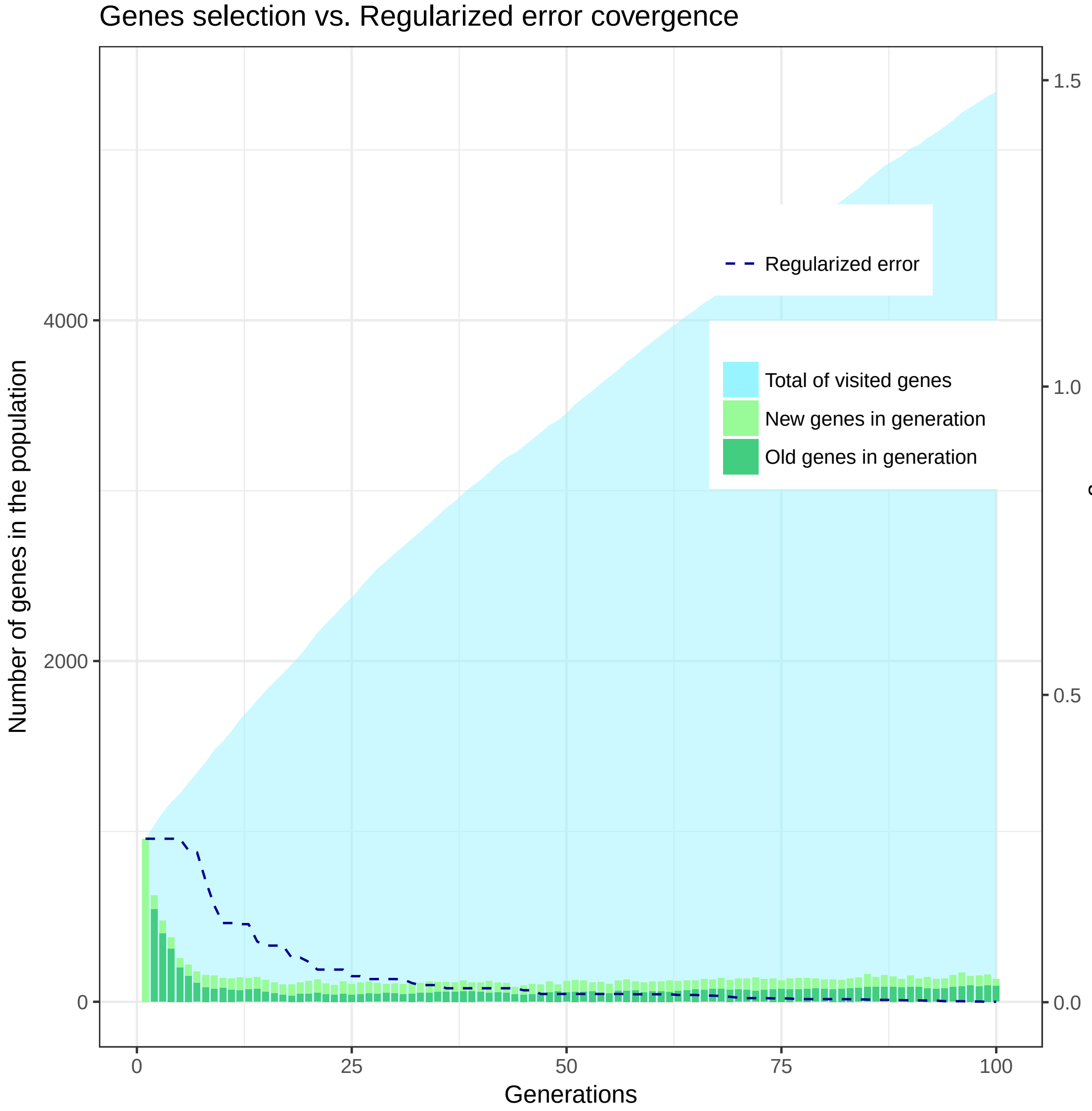
from exploring new solutions, as the total number of genes selected by at least one individual at some point in the evolution kept growing. Despite that, the quantity of genes existing simultaneously in the population dropped sharply during the first generations,

Figure 6.4: **Selection history of candidate genes in the entire population.** For the same experiment in Fig. 6.2 with the dataset GSE32323 (colorectal cancer). This chart brings all the genes allowed to be selected during the evolution (after filtering with Kruskal-Wallis H Test and $p < 0.01$), ordered in the x axis from the smallest p-value (left) to the highest p-value (right), and the generations in the y axis, from the beginning (bottom) to the end (top). If a gene was present in the population (selected by at least one individual) at a given generation, it will be marked with a circle in the respective position. The darker the point, the larger the number of individuals selecting this gene at the same generation. If a point is marked with an orange triangle, that gene, at that generation, was present in the individual with best fitness.



after which the number of new and old genes in the population stabilized and became balanced.

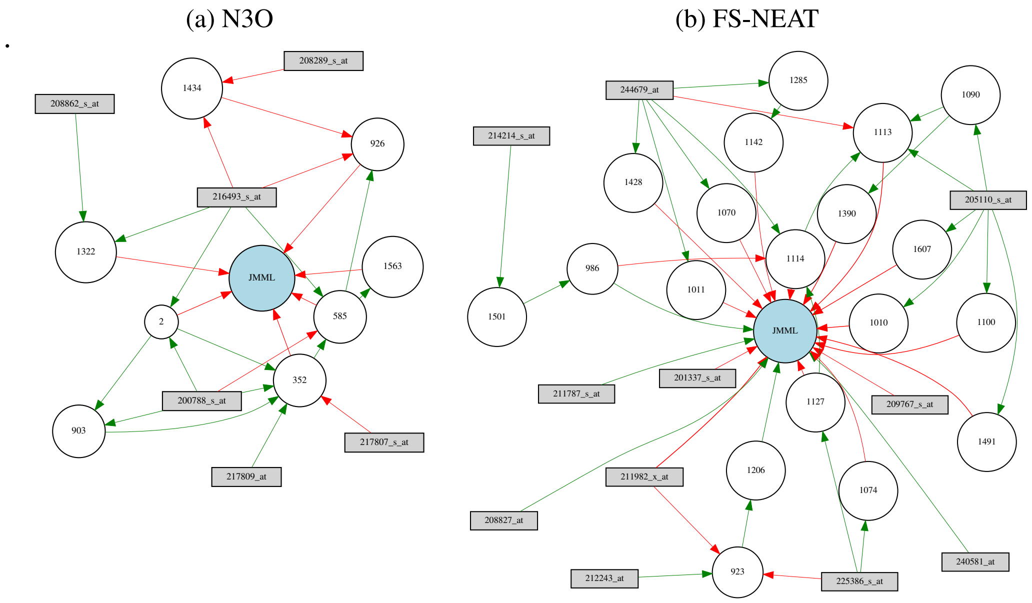
Figure 6.5: **Genes selection and error convergence during evolution.** For the same experiment in Fig. 6.2 with the dataset GSE32323 (colorectal cancer). Chart showing the convergence of both the best regularized error ($-fitness$) in the dashed line, and the selection of genes in the population in green bars. Each green bar represents the number of genes present in at least one individual of the population at a given generation. The darker bar is the number of genes that were already present in the previous generation, and the lighter bar the number of genes new to the population when compared with the previous generation. The blue curve represents the total exploration of genes, counting the number of genes that were present in at least one individual during at least one generation.



6.3 Departing from FS-NEAT

The kind of visualization from the last section can help in understanding the impact of the new structural operators of N3O compared with regular FS-NEAT. Taking as example the dataset GSE71935 (leukemia), it was performed an experiment with N3O and regular FS-NEAT, using the same preprocessing steps, filtering, fitness function, artificial neuron structure, and hyperparameters for both algorithms. The difference was that N3O had the addition of the new crossover operator and the swap input and guided add input mutations. The probability of the add input mutation in regular FS-NEAT happening was doubled, to compensate for the lack of the swap input mutation. Both algorithms used the exact same training (60%) and testing (40%) sets. The reported accuracies for the testing set were 100.0% for N3O and 75.0% for FS-NEAT, selecting 6 and 11 genes, respectively. Fig. 6.6 shows the two best ANNs created with both algorithms.

Figure 6.6: **ANNs created with N3O and FS-NEAT for the same data.** Two neural networks with best fitness in the population at the final generation for a run of N3O and regular FS-NEAT with dataset GSE71935 (leukemia). Details of the ANNs representation as in Fig. 6.2



(a) N3O

(b) FS-NEAT

As can be observed, FS-NEAT required a larger neural network structure than N3O. This is also visible in Fig. 6.7 from the same experiment. Both algorithms showed roughly the same regularized error convergence and total number of genes visited during the evolution, as well as a similar amount of new genes being explored at each generation. FS-NEAT, however, keeps a larger number of genes in the population at each generation, making for larger networks.

The difference between the genes selection of N3O and FS-NEAT is even more visible in Figs. 6.8 and 6.9. Despite visiting the same total amount of genes considering all generations, N3O required less features at each generation, and showed a better spread of genes among the individuals in the population.

Figure 6.7: **Genes selection and error convergence for N3O and FS-NEAT** for a run with dataset GSE71935 (leukemia). Details of the image representation as in Fig. 6.5

(a) N3O (b) FS-NEAT

.

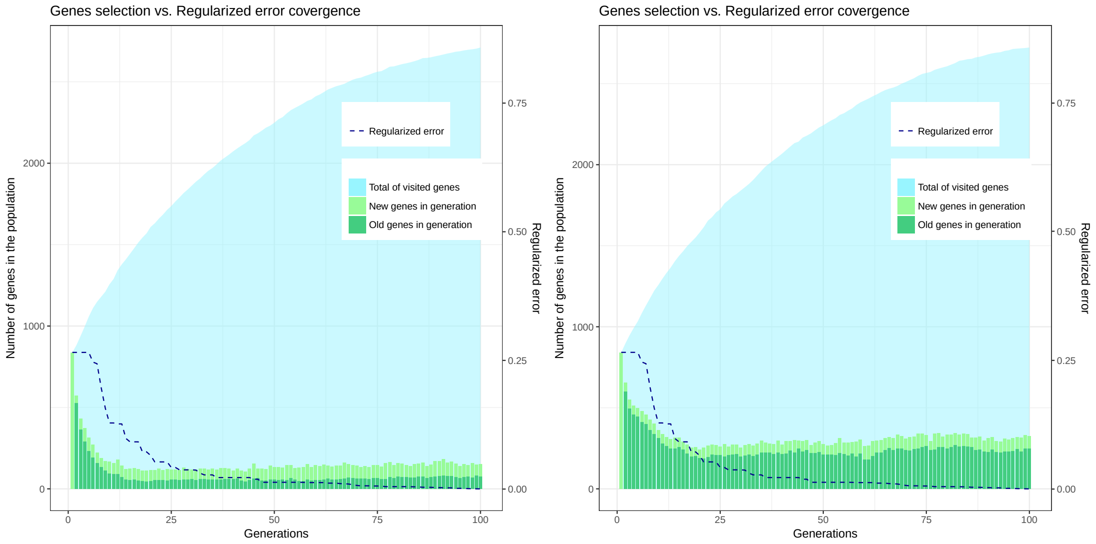
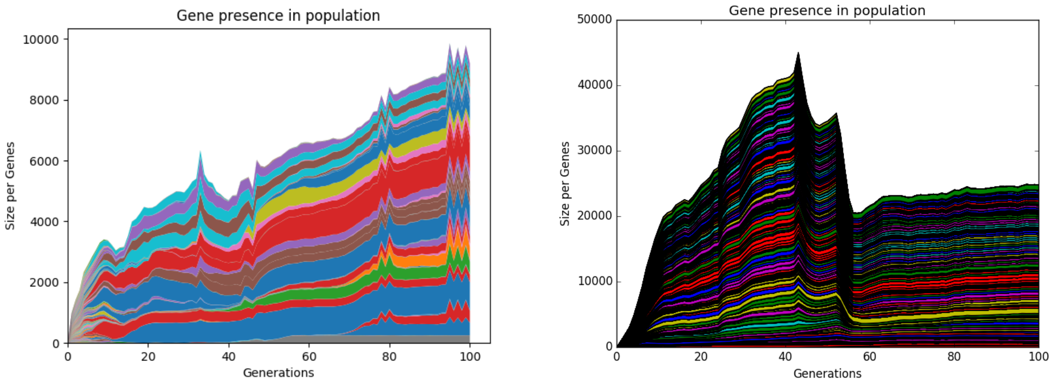


Figure 6.8: **Presence of genes in the population over the generations** for a run with dataset GSE71935 (leukemia). Details of the image representation as in Fig. 6.3c

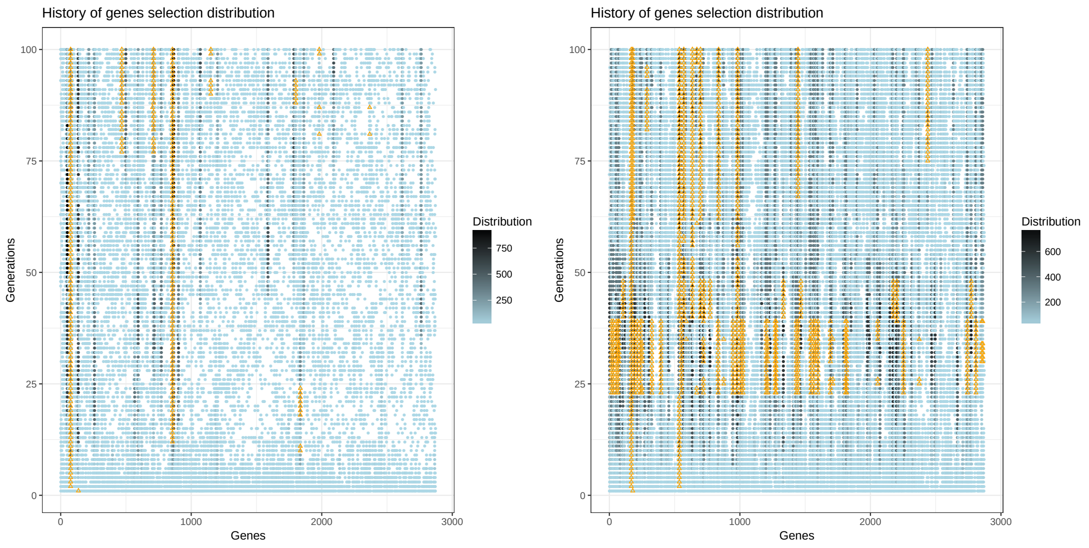
(a) N3O (b) FS-NEAT

.



In order to expand those results, 31 independent runs of stratified 3-fold cross-validation of N3O and FS-NEAT were performed under the same conditions for some of the datasets at Table 4.2, using random folds partitions for each run (the same for both algorithms). The accuracy and number of selected features are compared in Table 6.1. N3O consistently achieved better accuracies than regular FS-NEAT, but for most of the cases there was no statistical difference between the algorithms. Considering the number of selected genes, however, N3O performed better than FS-NEAT and was able to provide smaller solutions with at least the same predictive power. N3O also showed less variance in the number of selected genes than FS-NEAT for all studied cases. It may be

Figure 6.9: **Selection history of candidate genes for N3O and FS-NEAT** for a run with dataset GSE71935 (leukemia). Details of the image representation as in Fig. 6.4


(a) N3O

(b) FS-NEAT

the case that the guided input mutation makes N3O spend fewer resources searching for less favorable areas of the input space, while the swap input mutation promotes diversity in the subset of selected features without increasing it. On the new crossover operator, by allowing the inheritance of inputs from both parents (regular FS-NEAT only allows the inheritance from the parent with the best fitness), it may produce larger neural networks locally. Globally, however, the combination of the inputs from both parents allows the offspring to achieve better fitness without the need of exaggerated growth.

Table 6.1: **Accuracy and FS comparison of N3O with FS-NEAT.**

		Accuracy		**FS**	
Datasets	**Class**	**N3O**	**FS-NEAT**	**N3O**	**FS-NEAT**
GSE10797	Cancer Epithelial	**0.736** ± .058	0.725 ± .043	**13.65** ± 2.36	32.33 ± 10.77
	Cancer Stroma	**0.744** ± .035	0.734 ± .044	**13.85** ± 2.76	37.12 ± 12.53
	Normal	**0.930** ± .024	0.921 ± .024	**12.92** ± 4.19	20.09 ± 9.13
GSE8671		**0.984** ± .018	0.980 ± .020	**15.16** ± 3.99	17.53 ± 7.98
GSE32323		**0.939** ± .040	0.934 ± .043	**15.74** ± 4.02	20.29 ± 8.97
GSE41328		**0.968** ± .045	0.955 ± .071	18.67 ± 6.35	**18.60** ± 9.24
GSE14317		**0.964** ± .040	0.960 ± .044	**14.80** ± 4.76	20.77 ± 9.44
GSE71935		**0.902** ± .046	0.860 ± .047	**14.60** ± 3.42	26.13 ± 11.54
Golub et al. (1999)		0.900 ± .032	**0.901** ± .038	**12.51** ± 2.43	28.58 ± 11.97
Average		**0.896** ± .093	0.886 ± .095	**14.65** ± 1.83	24.60 ± 6.84

Reported values from 31 runs of the stratified 3-fold cross-validation. N3O = average accuracy and FS of the proposed method. FS-NEAT = average accuracy and FS of regular FS-NEAT (same fitness function and neuron structure as N3O). In bold are the best average accuracy and smallest average FS of each dataset. Best results with statistical significance ($p < 0.01$) are marked in blue.

In a final experiment, the genes selected in the runs of Table 6.1 by N3O and regular FS-NEAT were used to train SVMs, considered the state-of-the-art for microarray classification, as discussed in Section 3.3. The aim was to compare the generalizability of the solutions from both algorithms. As before, 31 runs of stratified 3-fold cross-validation were performed, with the same random partitions from the last experiment. The SVMs used RBF kernel and the hyperparameters were tuned with grid search. The results in Table 6.2 showed no statistically significant difference between the algorithms for most of the datasets.

Table 6.2: **Accuracy comparison of N3O and FS-NEAT gene selection applied to SVM.**

Datasets	Class	N3O	FS-NEAT
GSE10797	Cancer Epithelial	**0.850** ± .053	0.807 ± .060
	Cancer Stroma	**0.825** ± .062	0.789 ± .051
	Normal	**0.965** ± .018	0.957 ± .023
GSE8671		0.667 ± .000	**0.671** ± .029
GSE32323		0.686 ± .050	**0.696** ± .056
GSE41328		0.722 ± .000	**0.724** ± .023
GSE14317		**0.996** ± .012	0.982 ± .037
GSE71935		**0.966** ± .030	0.953 ± .042
Golub et al. (1999)		**0.943** ± .028	0.940 ± .039
Average		**0.847** ± .129	0.835 ± .124

The accuracy is the result from 31 runs of the stratified 3-fold cross-validation. All SVM versions used the RBF kernel and had their hyperparameters tuned by grid search. N3O = average accuracy of SVM using only the genes selected by the proposed method; FS-NEAT = average accuracy of SVM using only the genes selected by FS-NEAT. In bold is the best average accuracy of each dataset. Best results with statistical significance ($p < 0.01$) are marked in blue.

6.4 Microarray classification and gene selection

The aim of the next experiments was to characterize the classification and gene selection of N3O. For this, 31 independent runs of stratified 3-fold cross-validation were performed under the same conditions (totaling 93 complete executions of the method) for all datasets at Table 4.2, but with random folds partitions for each run.

The chosen metric for evaluating the classification was the accuracy, defined as the total number of true positives plus true negatives, divided by the total number of samples. Accuracy is the most used metric in gene selection studies (ANG et al., 2016). Due to the high imbalance in class sizes, the baseline for all datasets is also present for comparison. The results are reported in Table 6.3. As can be seen, the average accuracy always beat the baseline. The mean and median values are also close, suggesting the accuracies are well distributed around the mean. All datasets have coefficient of variation (the ratio between the standard deviation and the mean) lower than 0.1, what can be interpreted as low variance (GOMES, 2000). Similar results were obtained for the number of selected features, reported in Table 6.4.

Table 6.3: **Stratified 3-fold cross-validation statistical report of accuracy for N3O.**

Datasets	Class	Baseline	Mean±std	Median	Min-Max
GSE42568		0.87	0.978 ± .011	0.983	0.95 - 0.99
GSE45827	Basal	0.73	0.934 ± .016	0.934	0.89 - 0.97
	HER	0.80	0.946 ± .019	0.947	0.89 - 0.97
	Cell Line	0.91	0.994 ± .006	0.993	0.98 - 1.00
	Luminal A	0.81	0.934 ± .019	0.940	0.90 - 0.97
	Luminal B	0.80	0.890 ± .026	0.894	0.84 - 0.95
	Normal	0.95	0.988 ± .009	0.993	0.97 - 1.00
GSE10797	Cancer Epithelial	0.57	0.736 ± .058	0.727	0.58 - 0.83
	Cancer Stroma	0.57	0.744 ± .035	0.742	0.68 - 0.83
	Normal	0.85	0.930 ± .024	0.924	0.88 - 0.97
GSE44076		0.50	0.982 ± .009	0.985	0.97 - 1.00
GSE44861		0.50	0.823 ± .031	0.829	0.74 - 0.87
GSE8671		0.51	0.984 ± .018	0.984	0.94 - 1.00
GSE21510		0.58	0.956 ± .032	0.953	0.88 - 1.00
GSE32323		0.51	0.939 ± .040	0.939	0.85 - 1.00
GSE41328		0.55	0.968 ± .045	1.000	0.83 - 1.00
GSE9476	AML	0.59	0.901 ± .035	0.891	0.83 - 0.97
	Bone Marrow	0.84	0.989 ± .017	1.000	0.94 - 1.00
	Bone Marrow CD34	0.87	0.963 ± .023	0.969	0.92 - 1.00
	PB	0.84	0.994 ± .009	1.000	0.97 - 1.00
	PBSC CD34	0.84	0.976 ± .022	0.984	0.94 - 1.00
GSE14317		0.72	0.964 ± .040	0.960	0.84 - 1.00
GSE63270		0.59	0.969 ± .022	0.970	0.89 - 1.00
GSE71935		0.80	0.902 ± .046	0.891	0.83 - 0.98
Golub et al. (1999)		0.65	0.900 ± .032	0.903	0.83 - 0.97

Reported values from 31 runs of the stratified 3-fold cross-validation. Baseline computed as in Section 6.1.3. Std = Standard deviation; Min = Minimum value reported in all runs; Max = Maximum value reported in all runs.

An important aspect of FS is whether or not the same features are being selected in different runs. To analyze this, Table 6.5 reports the most selected genes for each dataset, considering the experiments in Table 6.4. It shows which genes appeared as selected the most in the final solutions, how many times this happened, and how many genes appeared in at least 5% of the solutions. The gene ERBB2 (HER2), for instance, was the most selected gene for the dataset GSE45827 - HER (breast cancer), being selected by 90.6% of the neural networks, while the gene SCNN1B was the most selected gene for the dataset GSE8671 (colorectal cancer), but appeared only in 3.1% of the networks.

Table 6.4: **Stratified 3-fold cross-validation statistical report of FS for N3O.**

Datasets	Class	Accuracy	Mean±std	Median	Min-Max
GSE42568		0.978	11.44 ± 3.12	10.67	6.33 - 19.00
GSE45827	Basal	0.934	11.76 ± 2.61	12.00	7.33 - 19.67
	HER	0.946	10.57 ± 2.63	10.33	5.33 - 17.00
	Cell Line	0.994	10.34 ± 3.97	09.67	4.33 - 21.00
	Luminal A	0.934	11.41 ± 2.02	11.33	6.00 - 16.00
	Luminal B	0.890	14.11 ± 2.38	14.00	10.0 - 18.33
	Normal	0.988	13.05 ± 4.51	12.00	7.33 - 26.00
GSE10797	Cancer Epithelial	0.736	13.65 ± 2.36	13.33	9.33 - 18.00
	Cancer Stroma	0.744	13.85 ± 2.76	13.33	7.67 - 20.00
	Normal	0.930	12.92 ± 4.19	13.00	6.67 - 20.33
GSE44076		0.982	09.65 ± 2.66	10.00	4.00 - 15.00
GSE44861		0.823	11.37 ± 2.55	10.67	6.67 - 16.33
GSE8671		0.984	15.16 ± 3.99	15.00	4.00 - 21.67
GSE21510		0.956	13.10 ± 4.47	13.00	3.00 - 22.00
GSE32323		0.939	15.74 ± 4.02	16.00	4.67 - 23.00
GSE41328		0.968	18.67 ± 6.35	18.67	3.00 - 29.33
GSE9476	AML	0.901	13.57 ± 2.80	13.00	8.00 - 19.00
	Bone Marrow	0.989	13.63 ± 3.61	13.67	5.67 - 20.33
	Bone Marrow CD34	0.963	12.52 ± 3.40	12.67	4.00 - 20.67
	PB	0.994	14.41 ± 4.77	13.67	5.00 - 26.33
	PBSC CD34	0.976	12.87 ± 3.62	13.33	7.00 - 19.00
GSE14317		0.964	14.80 ± 4.76	14.00	3.00 - 22.33
GSE63270		0.969	12.03 ± 3.11	12.33	5.33 - 18.00
GSE71935		0.902	14.60 ± 3.42	14.67	7.33 - 22.00
Golub et al. (1999)		0.900	12.51 ± 2.43	12.33	8.00 - 17.00

Reported values from 31 runs of the stratified 3-fold cross-validation. Average accuracy as reported from Table 6.3. Std = Standard deviation; Min = Minimum value reported in all runs; Max = Maximum value reported in all runs.

Even those genes with a small number of repetitions are significant, however, when the probability of it happening at random is considered, what, as shown in the last column of Table 6.5 and discussed in Section 6.1.3, is highly unlikely.

To further validate this selection, Table 6.6 brings a literature review of the most selected genes from Table 6.5, considering the PubMed[5] repository. On total, 44% of those genes were already described in the literature as being relevant for the specific cancer type of their corresponding dataset, 20% were described as relevant for other cancer types, 20% were not described as relevant for any cancer type, and 16% were not yet described in the literature. Interestingly, the aforementioned gene ERBB2 (HER2) was the most selected gene in its dataset among all experiments, while also being described as one of the most relevant genes in breast cancer in general (BORGES et al., 2018; NATTESTAD et al., 2018; AL., 2018; SOARES et al., 2018).

[5] <https://www.ncbi.nlm.nih.gov/pubmed/>

Table 6.5: **Most selected genes by N3O.**

Datasets	**Class**	**Accuracy**	**FS**	**Probe**	**Gene**	**s(d)**	**d** ≥ 0.05	$P_g[X \geq s]$
GSE42568		0.978	11.44	219059_s_at	LYVE1	05 (0.052)	01	$2.05e^{-11}$
GSE45827	Basal	0.934	11.76	218211_s_at	MLPH	21 (0.219)	16	$3.33e^{-16}$
	HER	0.946	10.57	210930_s_at	ERBB2 (HER2)	87 (0.906)	11	0.00
	Cell Line	0.994	10.34	242646_at	AA702946	04 (0.042)	00	$3.68e^{-9}$
	Luminal A	0.934	11.41	228554_at	PGR	13 (0.135)	07	0.00
	Luminal B	0.890	14.11	213557_at	Hs.444858	11 (0.125)	09	$4.66e^{-15}$
	Normal	0.988	13.05	226018_at	C7orf41	03 (0.031)	00	$1.74e^{-6}$
GSE10797	Cancer Epithelial	0.736	13.65	208331_at	BPY2	12 (0.125)	20	$1.55e^{-15}$
	Cancer Stroma	0.744	13.85	212760_at	UBR2	10 (0.104)	05	$2.44e^{-15}$
	Normal	0.930	12.92	205051_s_at	KIT	34 (0.354)	43	$1.11e^{-16}$
GSE44076		0.982	09.65	11730386_at	GREM2	06 (0.062)	01	$4.01e^{-14}$
GSE44861		0.823	11.37	215118_s_at	ENSG00000253701	24 (0.250)	14	0.00
GSE8671		0.984	15.16	205464_at	SCNN1B	03 (0.031)	00	$2.71e^{-6}$
GSE21510		0.956	13.10	221922_at	GPSM2	03 (0.031)	00	$1.76e^{-6}$
GSE32323		0.939	15.74	218513_at	C4orf43	04 (0.042)	00	$1.96e^{-8}$
GSE41328		0.968	18.67	201195_s_at	SLC7A5	04 (0.042)	00	$3.87e^{-8}$
GSE9476	AML	0.901	13.57	212224_at	ALDH1A1	18 (0.187)	23	0.00
	Bone Marrow	0.989	13.63	211820_x_at	GYPA	12 (0.125)	31	$3.33e^{-16}$
	Bone Marrow CD34	0.963	12.52	205347_s_at	TMSB15A	07 (0.073)	09	$1.65e^{-13}$
	PB	0.994	14.41	207387_s_at	GK	08 (0.083)	05	$6.44e^{-15}$
	PBSC CD34	0.976	12.87	213714_at	CACNB2	11 (0.114)	19	0.00
GSE14317		0.964	14.80	203139_at	DAPK1	05 (0.052)	01	$6.41e^{-9}$
GSE63270		0.969	12.03	204294_at	AMT	11 (0.114)	16	$3.33e^{-15}$
GSE71935		0.902	14.60	219295_s_at	PCOLCE2	12 (0.125)	16	0.00
Golub et al. (1999)		0.900	12.51	U46499_at	GST	27 (0.281)	36	$1.11e^{-15}$

Reported values from 31 runs of the stratified 3-fold cross-validation. Average accuracy and average FS as reported from Table 6.4. Probe = the probe number of the selected gene; Gene = gene corresponding to the probe; s(d) = how many times the most selected gene was selected (distribution on all networks). d ≥ 0.05 = how many genes were selected in or more than 5% of the networks. $P_g[X \geq s]$ = probability of the most selected gene being randomly selected s or more times (if 0.00 the system lacked enough float precision to represent the number).

Table 6.6: **Literature review of the most selected genes by N3O.**

Datasets	**Class**	**Gene**	**s**	**References**
GSE42568		🟩 LYVE1	05	(MARTÍNEZ-IGLESIAS et al., 2016; NEWMAN et al., 2012)
GSE45827	Basal	🟩 MLPH	21	(THAKKAR et al., 2010; THAKKAR et al., 2015)
	HER	🟩 ERBB2 (HER2)	87	(BORGES et al., 2018; NATTESTAD et al., 2018)
				(AL., 2018; SOARES et al., 2018)
	Cell Line	AA702946	04	
	Luminal A	🟩 PGR	13	(KUNC; BIERNAT; SENKUS-KONEFKA, 2018)
	Luminal B	Hs.444858	11	
	Normal	C7orf41	03	
GSE10797	Cancer Epithelial	🟦 BPY2	12	(DASARI et al., 2002)
	Cancer Stroma	🟥 UBR2	10	
	Normal	🟥 KIT	34	
GSE44076		🟩 GREM2	06	(LIU et al., 2011)
GSE44861		ENSG00000253701	24	
GSE8671		🟩 SCNN1B	03	(SHANGKUAN et al., 2017b)
GSE21510		🟦 GPSM2	03	(LIU; WANG; SUN, 2015)
GSE32323		🟥 C4orf43	04	
GSE41328		🟩 SLC7A5	04	(KALMAR et al., 2013)
GSE9476	AML	🟩 ALDH1A1	18	(LONGVILLE et al., 2015; GASPARETTO; SMITH, 2017b)
	Bone Marrow	🟩 GYPA	12	(LI et al., 2015)
	Bone Marrow CD34	🟦 TMSB15A	07	(DARB-ESFAHANI et al., 2012)
	PB	🟥 GK	08	
	PBSC CD34	🟦 CACNB2	11	(TOMOSHIGE et al., 2015; CHEN et al., 2016)
GSE14317		🟩 DAPK1	05	(TAO et al., 2015; NG et al., 2014; CELIK et al., 2015)
GSE63270		🟥 AMT	11	
GSE71935		🟦 PCOLCE2	12	(THUTKAWKORAPIN et al., 2016)
Golub et al. (1999)		🟩 GST	27	(LAVROV et al., 2017; TANG et al., 2018)

Reported values from 31 runs of the stratified 3-fold cross-validation. Gene and s as reported in Table 6.5. If gene (i) green: reported in the literature as relevant for the corresponding cancer type of the dataset; (ii) blue: reported as relevant for another cancer type; (iii) red: not reported as relevant for any cancer type; (iv) white: gene not described. The references were obtained from the PubMed repository. Table made in collaboration with Dr. Bruno César Feltes - SBCB Lab, INF-UFRGS.

6.4.1 Comparison with SVM

As mentioned earlier, the literature points to SVM as being the best classifier of microarray data, making a comparison between N3O and SVM relevant. The accuracy of N3O and SVMs with RBF kernel and hyperparameters tuned by grid search were compared in three configurations: (i) over the original dataset (Table 6.7, column 4); (ii) after filtering the genes with Kruskal-Wallis H Test (Table 6.7, column 5); (iii) using only the genes selected by N3O (Table 6.7, column 6). 31 runs of stratified 3-fold cross-validation with random partitions were performed for each configuration, keeping the same partitions over different algorithms. The results are in Table 6.7. N3O showed some competitive results against SVM, especially for the datasets GSE8671, GSE32323, and GSE41328, all of them of colorectal cancer. For most of the datasets, however, SVM remains as the classifier with more predictive power.

From the discussion in Section 3.4, it is known that gene selection performed with a classifier is only specific to that given algorithm, meaning that there is no guarantee that the selected features will have a good performance with other methods (ANG et al., 2016). Furthermore, SVMs are usually insensitive to a large number of irrelevant genes, and FS often biases down their accuracy (STATNIKOV; WANG; ALIFERIS, 2008). Even so, when the genes selected by the N3O were applied to SVM (Table 6.7, column 6), its performance was not hurt, and for most of the datasets, it actually had a slight improvement. This result suggests that the selected genes are not methodological artifacts, and could be generalized and further explored even by different algorithms.

6.4.2 Comparison with another Neuroevolution method

A final experiment was made to compare the results of N3O with the recent Neuroevolution method for microarray classification described in Garro, Rodríguez and Vazquez (2017) and discussed in Section 3.5. This approach used the ABC optimization algorithm to select genes and chose the top three ranked genes to be the inputs of ANNs evolved by DE.

The structure of this experiment is different from the one described in the last sections to be coherent with the methodology described in Garro, Rodríguez and Vazquez (2017). Thus, instead of 31 runs of stratified 3-fold cross-validation, we report the average test accuracy and FS of N3O over 30 independent runs with random partitions (80%

Table 6.7: **Accuracy comparison of N3O and SVM.**

Datasets	Class	N3O	SVM	KW&SVM	N3O&SVM
GSE42568		0.978 ± .011	0.985 ± .007	0.985 ± .006	**0.990** ± .006
GSE45827	Basal	0.934 ± .016	**0.972** ± .003	0.971 ± .004	0.968 ± .012
	HER	0.946 ± .019	0.962 ± .010	0.950 ± .011	**0.973** ± .026
	Cell Line	0.994 ± .006	**1.000** ± .000	**1.000** ± .000	0.999 ± .003
	Luminal A	0.934 ± .019	0.968 ± .014	**0.979** ± .007	0.965 ± .017
	Luminal B	0.890 ± .026	**0.931** ± .013	0.928 ± .016	0.923 ± .024
	Normal	0.988 ± .009	**0.995** ± .003	0.993 ± .000	0.994 ± .005
GSE10797	Cancer Epithelial	0.736 ± .058	**0.857** ± .028	**0.857** ± .028	0.850 ± .053
	Cancer Stroma	0.744 ± .035	0.761 ± .036	0.761 ± .036	**0.825** ± .062
	Normal	0.930 ± .024	0.924 ± .019	0.924 ± .019	**0.965** ± .018
GSE44076		0.982 ± .009	0.983 ± .003	0.984 ± .003	**0.987** ± .008
GSE44861		0.823 ± .031	**0.829** ± .045	**0.829** ± .045	**0.829** ± .059
GSE8671		**0.984** ± .018	0.698 ± .065	0.698 ± .065	0.667 ± .000
GSE21510		0.956 ± .032	**0.986** ± .021	**0.986** ± .021	**0.986** ± .039
GSE32323		**0.939** ± .040	0.692 ± .066	0.692 ± .066	0.686 ± .050
GSE41328		**0.968** ± .045	0.695 ± .061	0.697 ± .040	0.722 ± .000
GSE9476	AML	0.901 ± .035	0.947 ± .016	0.920 ± .019	**0.954** ± .039
	Bone Marrow	0.989 ± .017	0.984 ± .000	**0.998** ± .005	0.997 ± .007
	Bone Marrow CD34	0.963 ± .023	**0.997** ± .007	0.980 ± .019	0.984 ± .018
	PB	0.994 ± .009	0.985 ± .013	**1.000** ± .000	0.999 ± .004
	PBSC CD34	0.976 ± .022	0.984 ± .010	**0.997** ± .006	0.995 ± .012
GSE14317		0.964 ± .040	0.957 ± .044	0.991 ± .025	**0.996** ± .012
GSE63270		0.969 ± .022	**0.999** ± .003	0.998 ± .004	0.991 ± .011
GSE71935		0.902 ± .046	0.896 ± .034	0.923 ± .034	**0.966** ± .030
Golub et al. (1999)		0.900 ± .032	0.961 ± .022	**0.978** ± .012	0.943 ± .028
Average		**0.931** ± .070	0.918 ± .102	0.921 ± .103	0.926 ± .102

The accuracy is the result of 31 runs of the stratified 3-fold cross-validation. All SVM versions used the RBF kernel and had their hyperparameters tuned by grid search. N3O = average accuracy of the proposed method; SVM = average accuracy of SVM; KW&SVM = average accuracy of SVM after filtering the data with Kruskal-Wallis H Test; N3O&SVM = average accuracy of SVM using only the genes selected by the proposed method. In bold is the best average accuracy of each dataset. Best results with statistical significance ($p < 0.01$) are marked in blue.

training, 20% training). The number of generations of N3O was also halved to match the number of fitness evaluations in Garro, Rodríguez and Vazquez (2017). The comparison is showed in Table 6.8. While N3O accuracy was slightly best, the results are inconclusive. Nevertheless, it shows that N3O can match recent results in the literature.

Table 6.8: **Comparison of N3O with another Neuroevolution method.**

Method	Dataset	Accuracy	FS
N3O	Golub et al. (1999)	0.917 ± .095	6.27 ± 2.38
ABC&DE	Golub et al. (1999)	0.912 ± .067	3

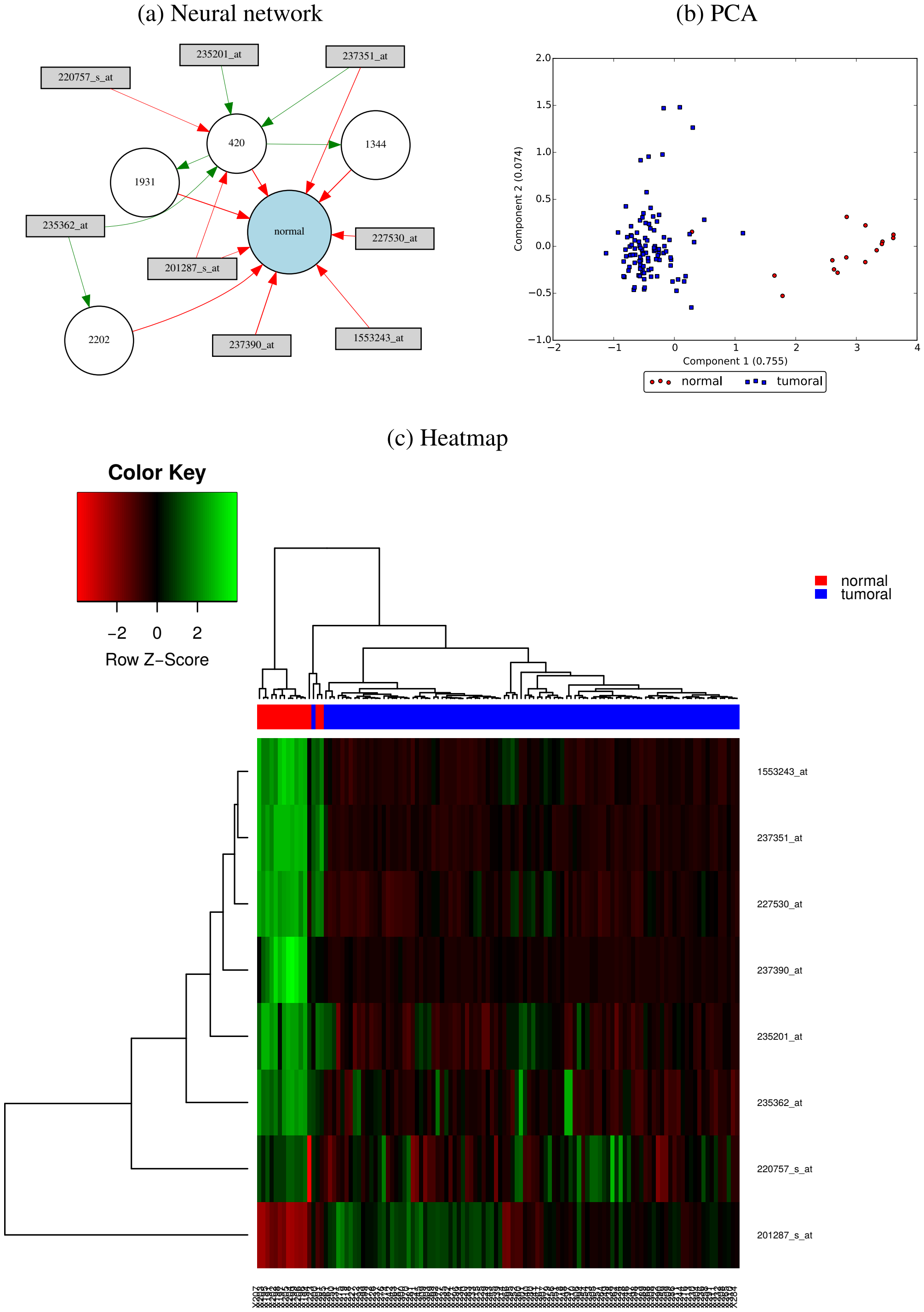
N3O = average accuracy and number of selected features of our method for the testing set (20%) with random partition over 30 repetitions; ABC&DE = accuracy reported by the method from Garro, Rodríguez and Vazquez (2017) for the testing set (20%) with random partition over 30 repetitions; FS = number of selected features.

6.5 Expression Patterns and Gene Selection

After computing the accuracy and FS of N3O over all datasets with several runs of stratified 3-fold cross-validation, a final experiment was performed, consisting of a single run of N3O considering all available samples (no testing set), with the objective of analyzing the gene selection and expression patterns found by the method with all the information available and considering all selected genes together, since an ANN will always use them combined. The set of genes representing an expression pattern was extracted from each GSE based on the best neural network for each run. Figs. 6.10 to 6.26 show these results by representing (a) the best neural network, whose inputs are the selected genes; (b) a 2D vision of the whole dataset considering the selected genes expression by applying PCA to the original data; (c) the heatmap of genes expression, with rows (selected genes expression) and columns (samples) ordered by hierarchical clustering. The class labels in these images are always the original sample labels in the datasets. For the GSEs with more than two classes, only the gene expression patterns that are exclusive to the tumoral classes are discussed.

Table 6.9 lists: (i) the number of genes that were selected for each GSE, per class. In this sense, the algorithm selects the set of genes that differ in a given condition from the other; (ii) the number of genes that were already associated to the GSE's cancer type in the literature; (iii) the quantity of long non-coding RNA (lncRNAs); (iv) the amount of genes that were not found to be related to any type of cancer in the literature, or that don't have a clear described function, such as predicted genes; and (v) the number of genes that were not observed to be related to the GSE's cancer type in the literature, but found in others. The complete list of selected genes with their associated cancer type can be found in Table 6.5. In summary, among the 177 selected genes, 82 were already associated to their given cancer type (lncRNA apply here), 5 were lncRNAs, 44 were not yet related to the GSE's cancer type, but were observed to be altered in other cancer types, and a total of 50 genes didn't return any hits from the scientific literature search, either because they don't possess a clear described function, or were just not related to any tumoral condition (lncRNA apply here). Interestingly, each expression pattern was unique, and only the REC8 Meiotic Recombination Protein (REC8) was common between a set of Leukemia and one of Colorectal cancer (CRC).

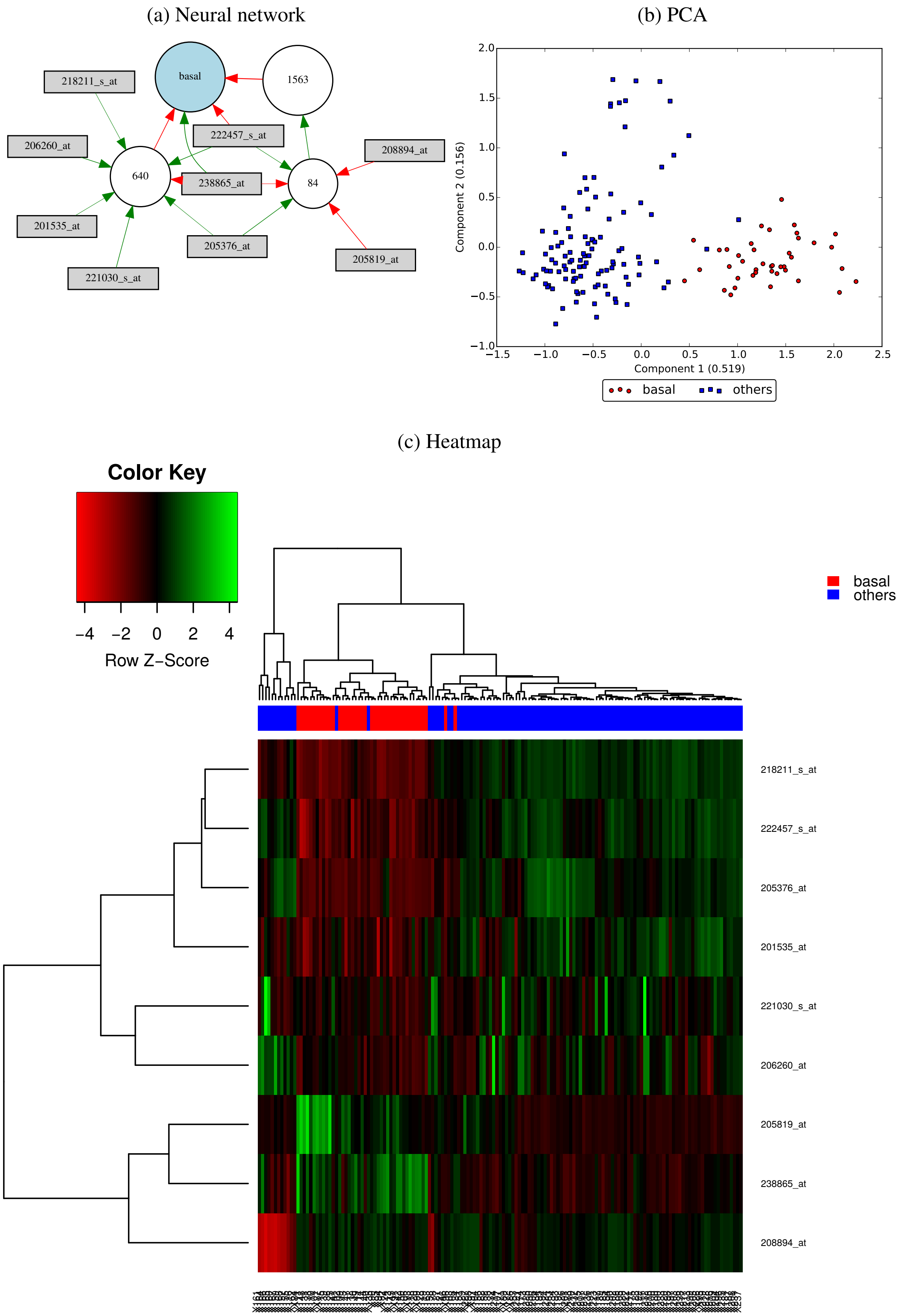
Figure 6.10: **Detailment of gene selection for GSE42568 - Breast Cancer.** (a) The best neural network considering all available samples. The blue circle is the output node, the white circles are hidden nodes, and grey squares are input nodes. Arrows are connections, whose thickness is proportional to the absolute values of their weights. Arrows are red if the weight is negative, or green otherwise. The number inside the hidden nodes inform the order of emergence. (b) Principal component analysis of the expression of genes present in the network. (c) Heatmap of raw gene expression of the selected genes (rows). The red and blue bar at the top is the true label of the samples (columns). The order of samples and genes was determined by hierarchical clustering represented by the dendrograms.


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.11: **Detailment of gene selection for GSE45827 - Breast Basal.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.

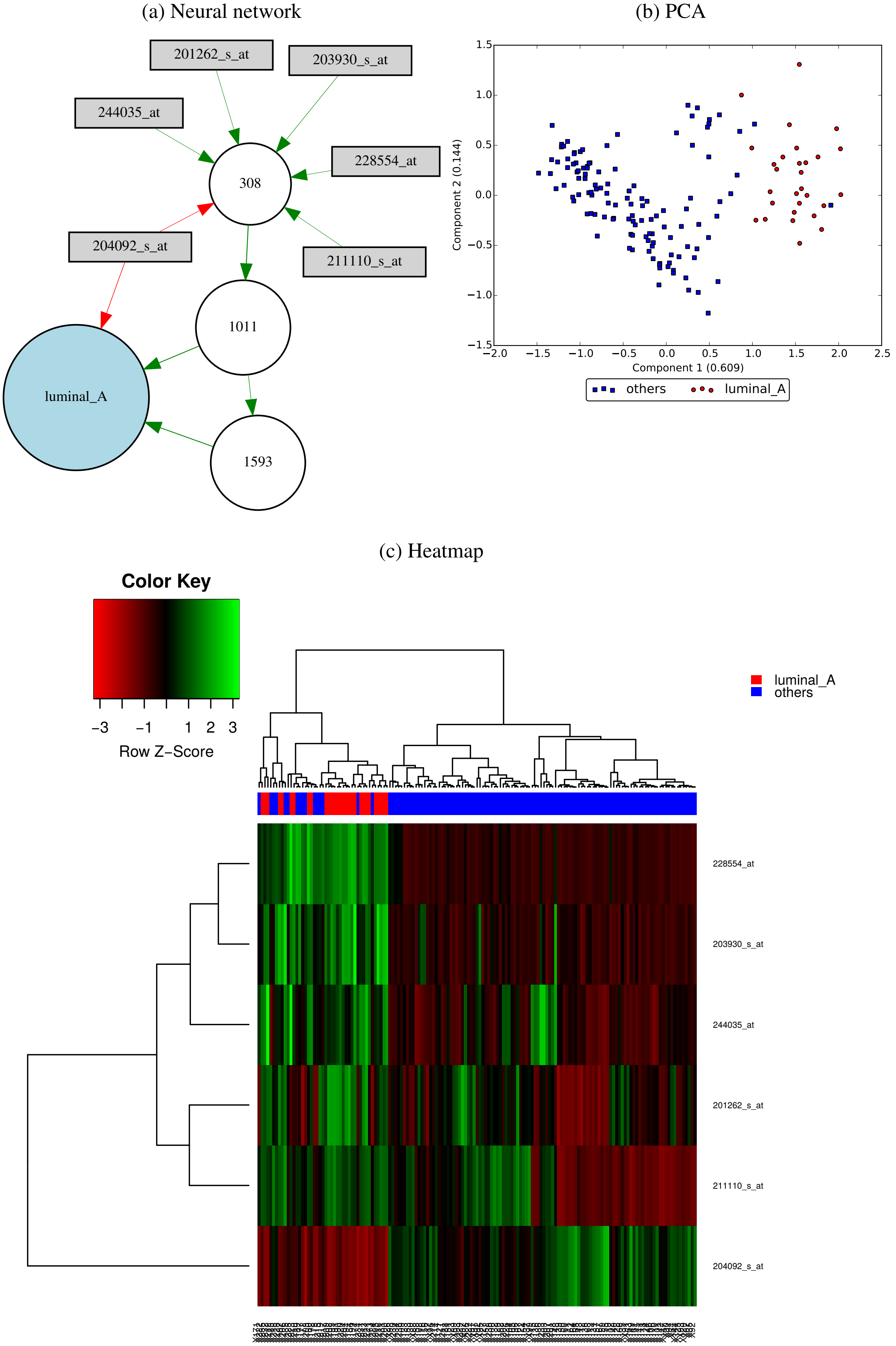


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.12: **Detailment of gene selection for GSE45827 - Breast Luminal A.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.

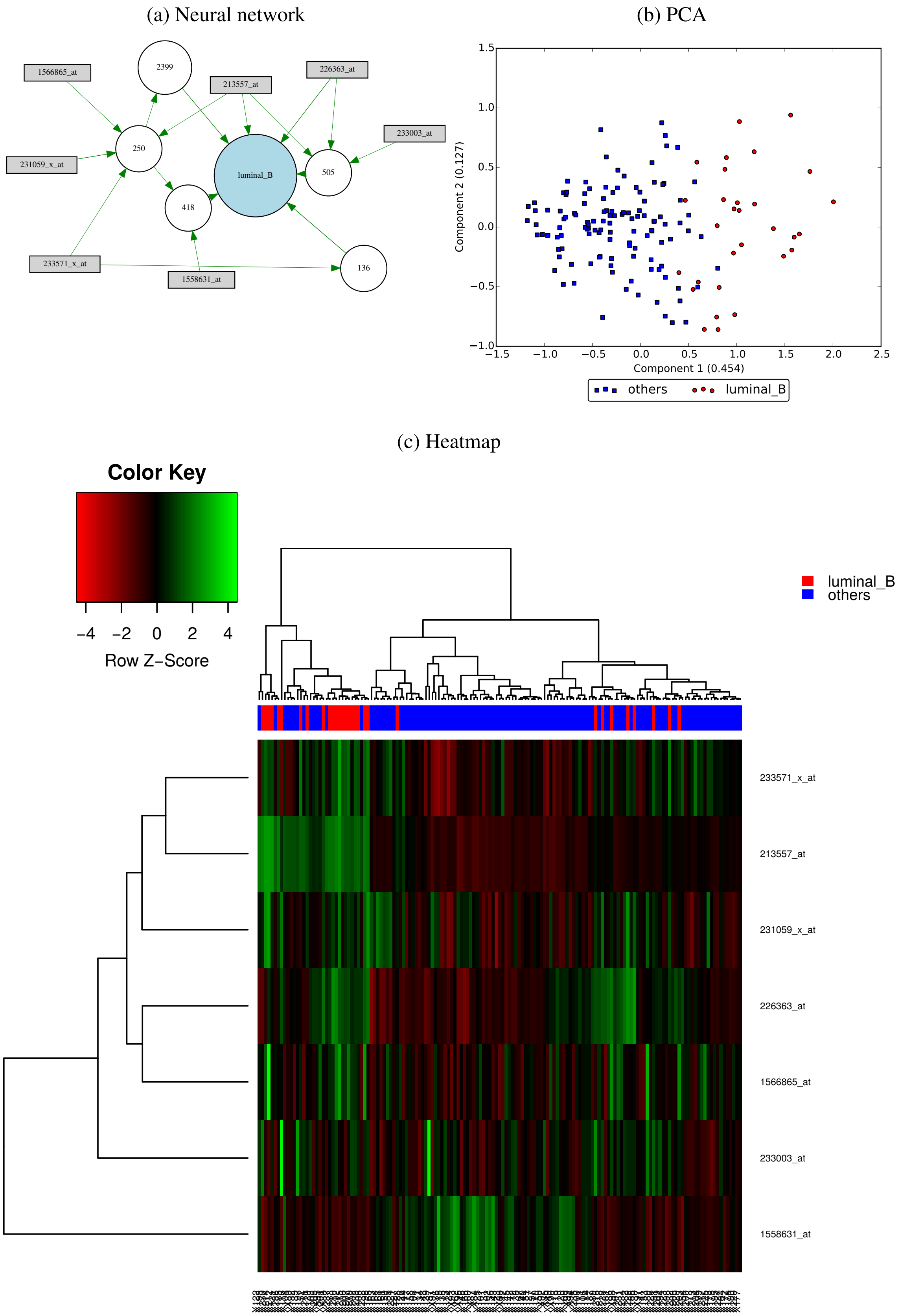


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.13: **Detailment of gene selection for GSE45827 - Breast Luminal B.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.

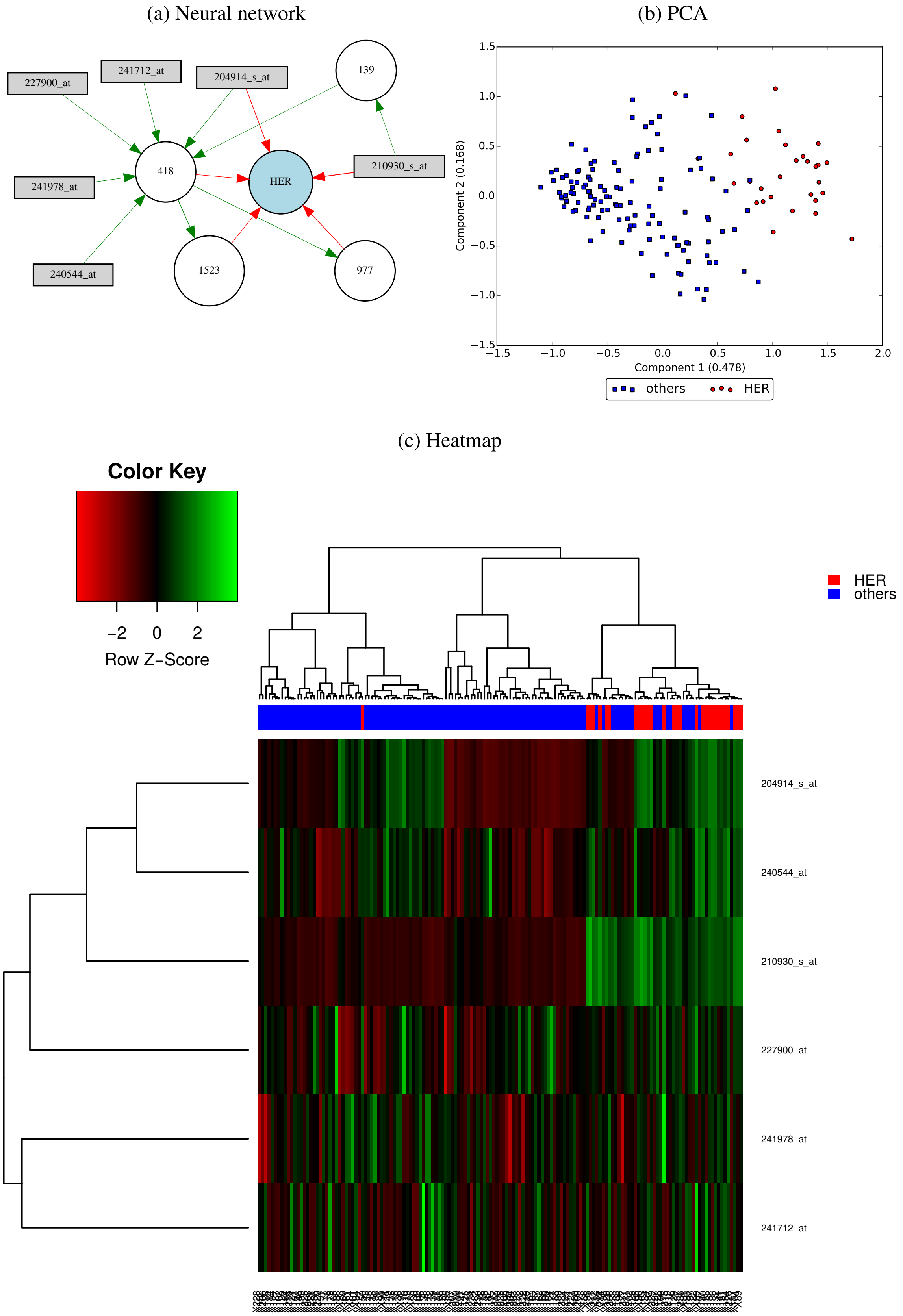


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.14: **Detailment of gene selection for GSE45827 - Breast HER.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.

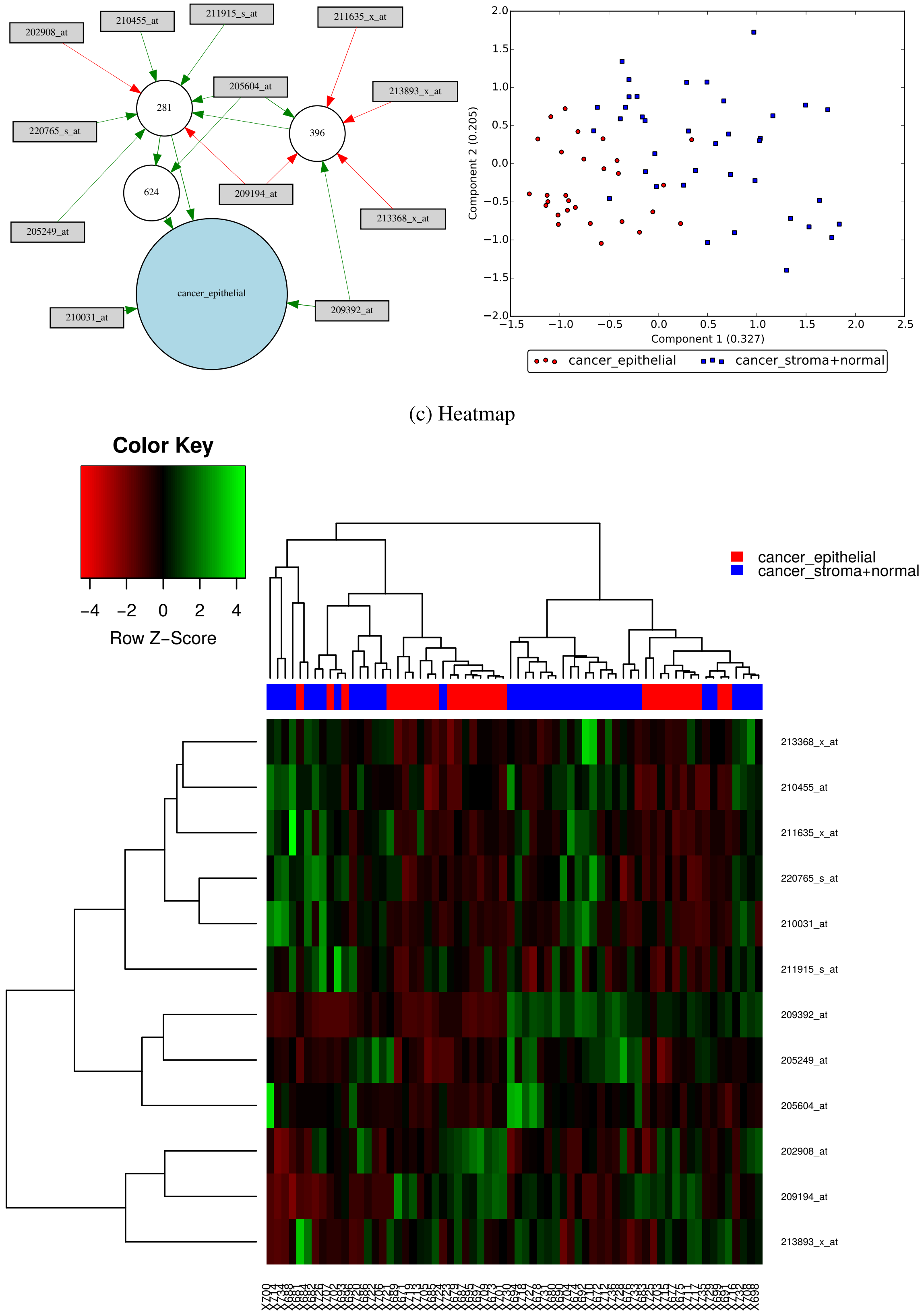


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.15: **Detailment of gene selection for GSE10797 - Breast Epithelium.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.

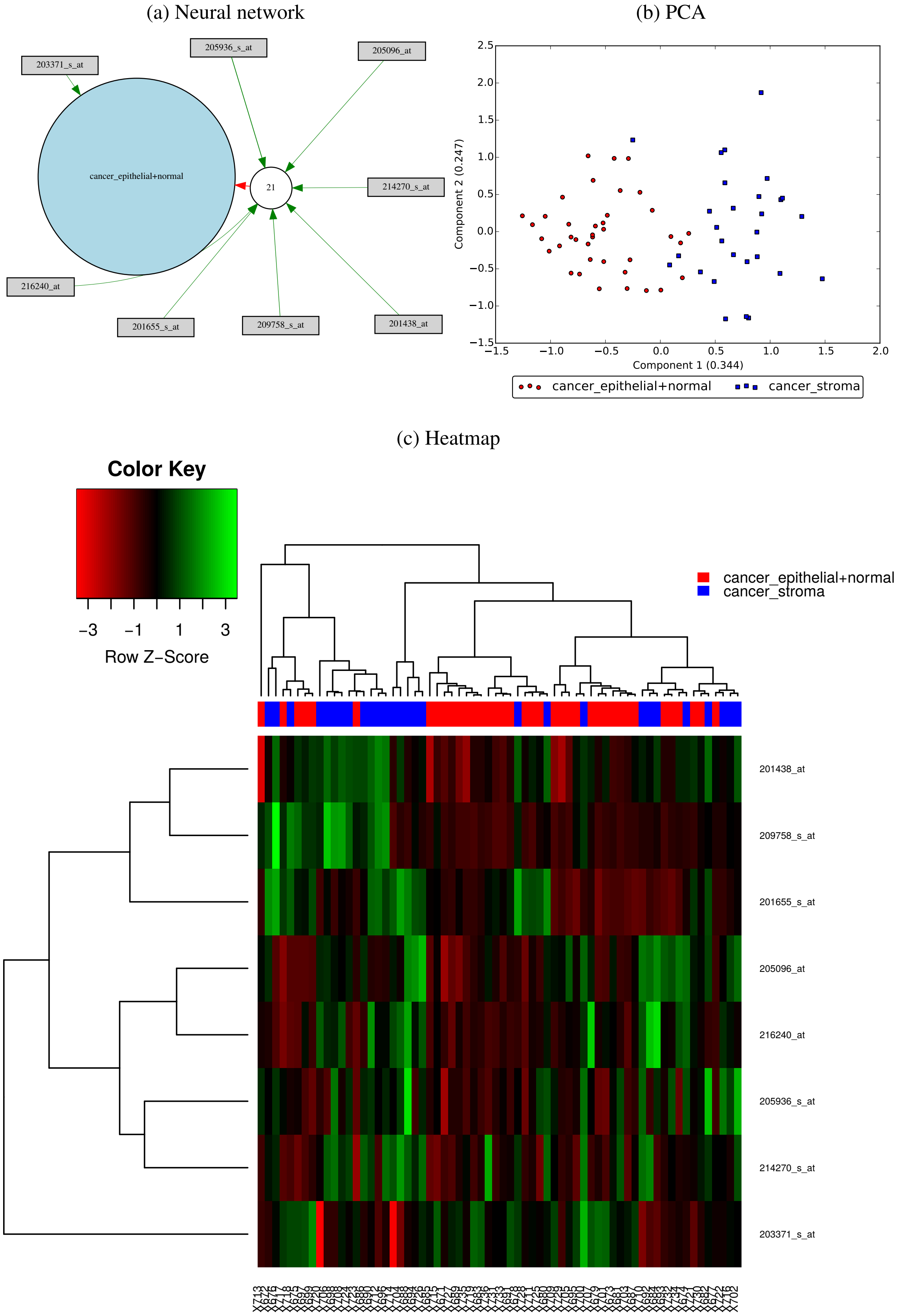


(a) Neural network

(b) PCA

(c) Heatmap

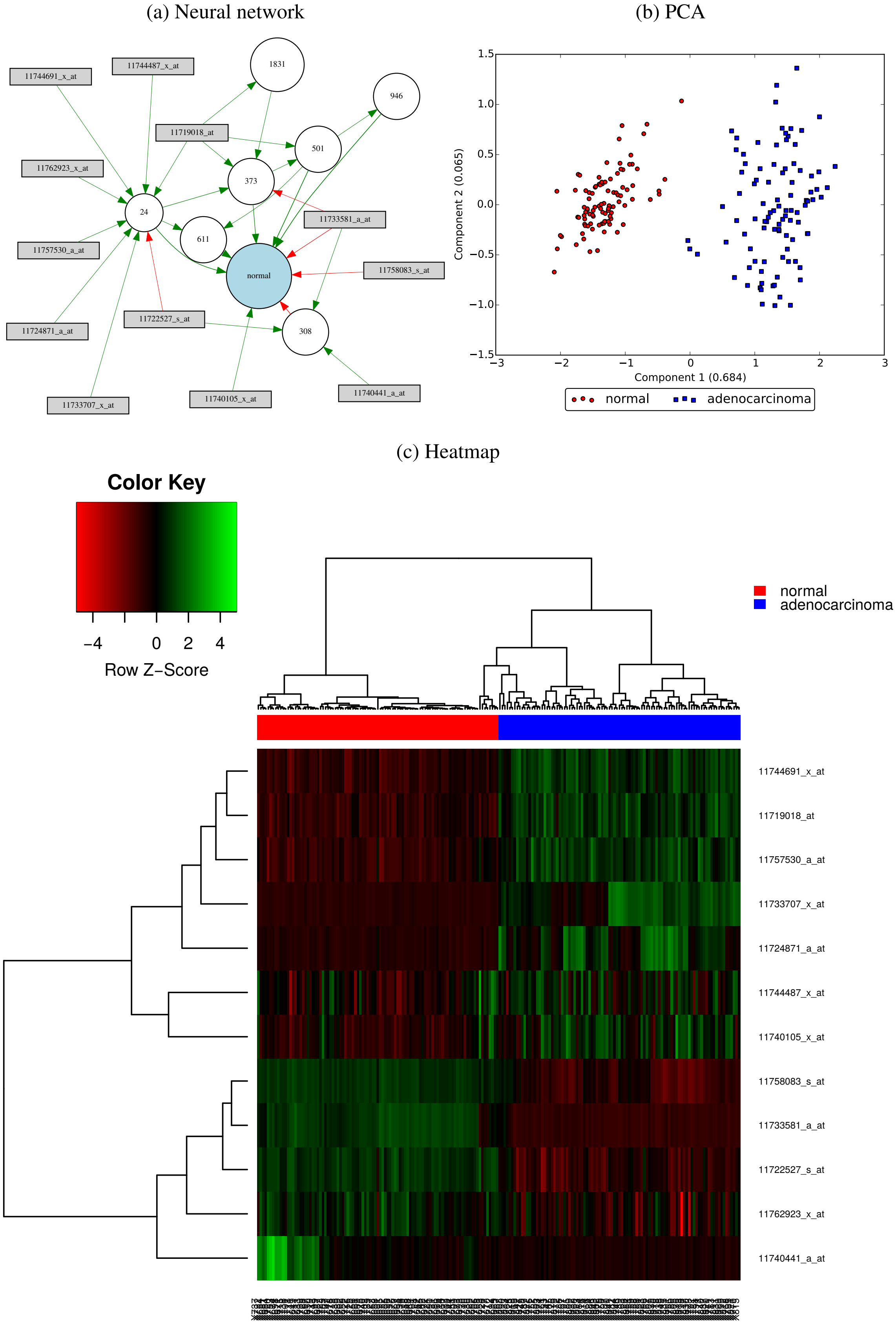
Figure 6.16: **Detailment of gene selection for GSE10797 - Breast Stromal.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.


(a) Neural network

(b) PCA

(c) Heatmap

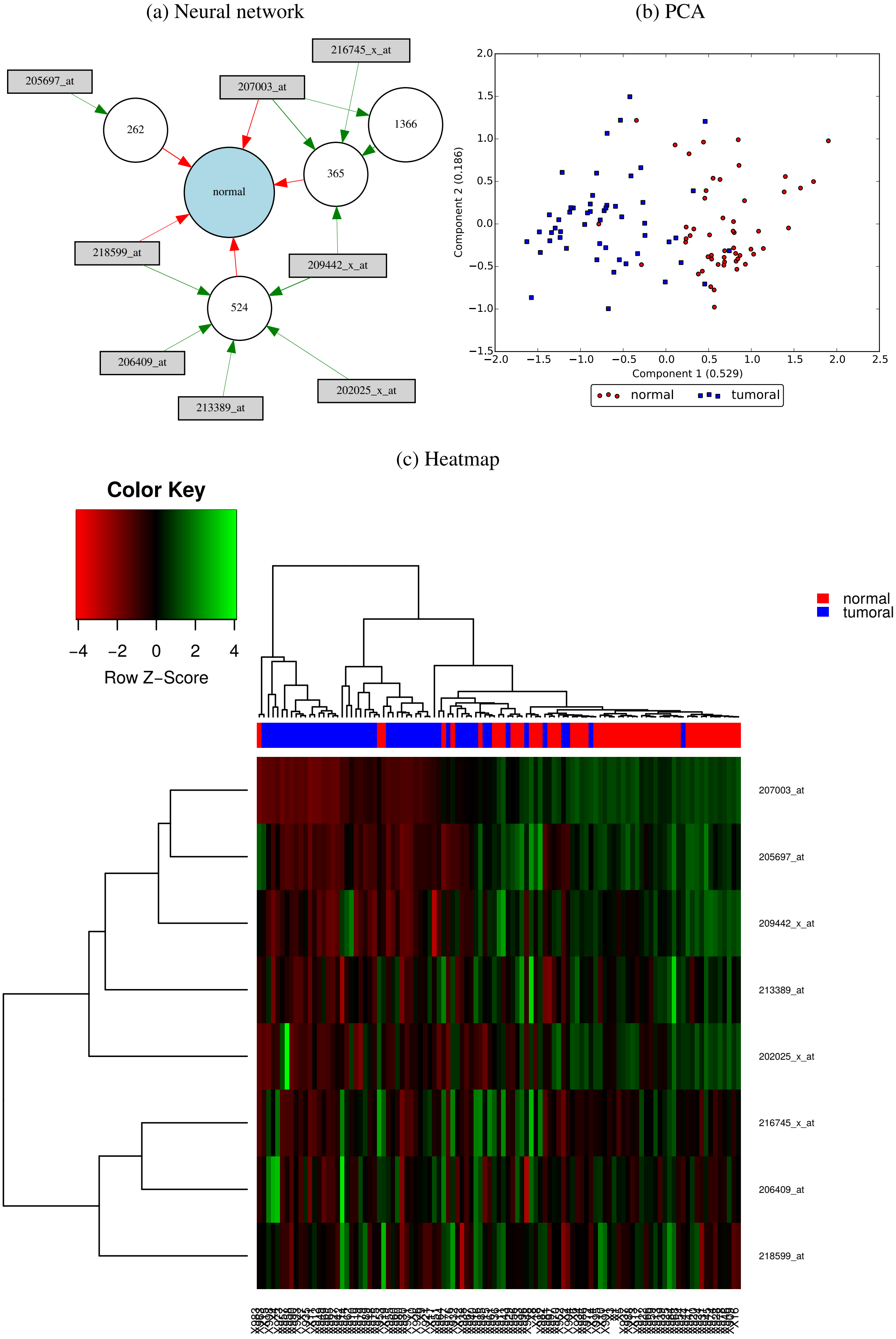
Figure 6.17: **Detailment of gene selection for GSE44076 - CRC Adenocarcinoma.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.


(a) Neural network

(b) PCA

(c) Heatmap

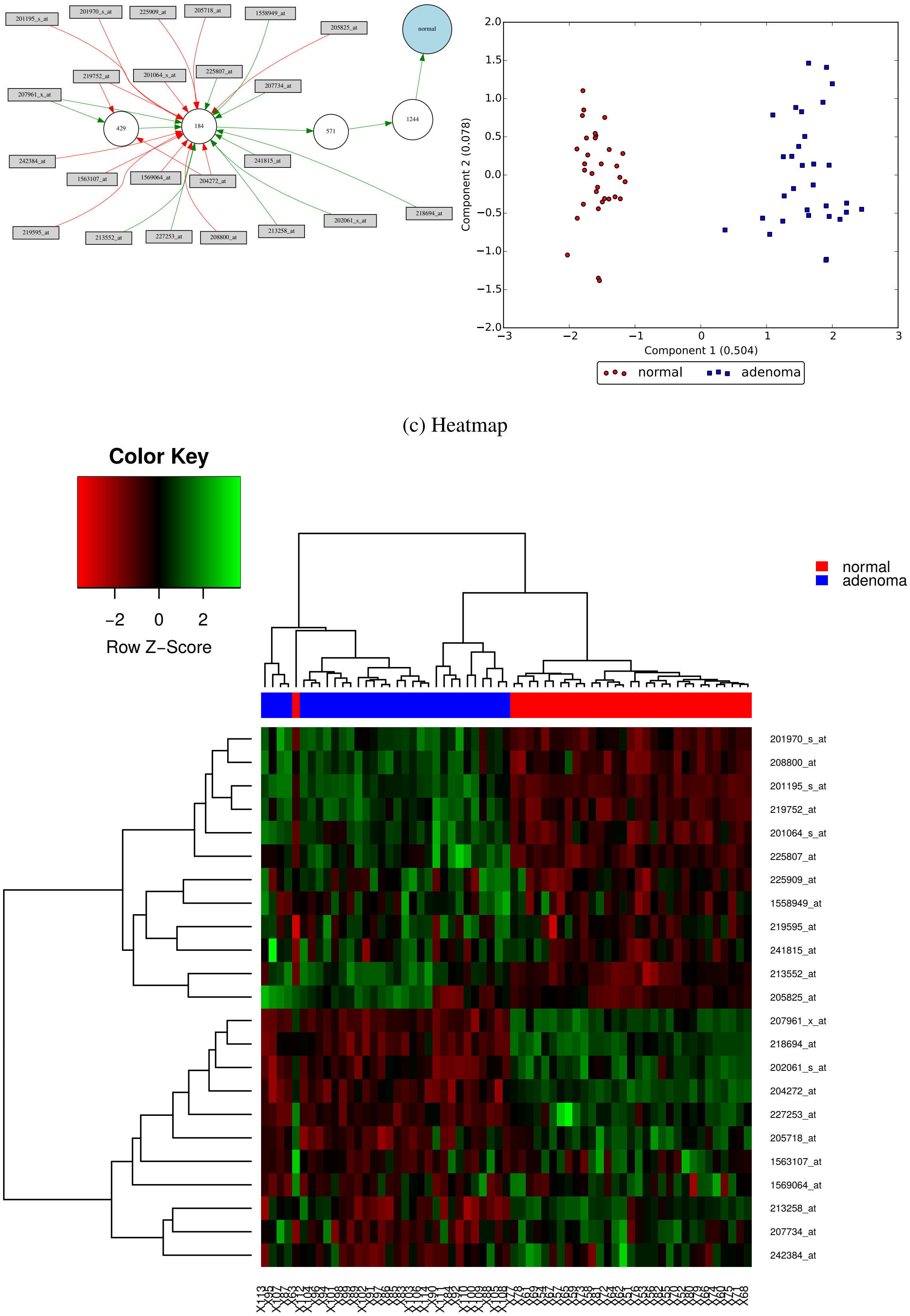
Figure 6.18: **Detailment of gene selection for GSE44861 - CRC.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.19: **Detailment of gene selection for GSE8671 - CRC Adenoma.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.

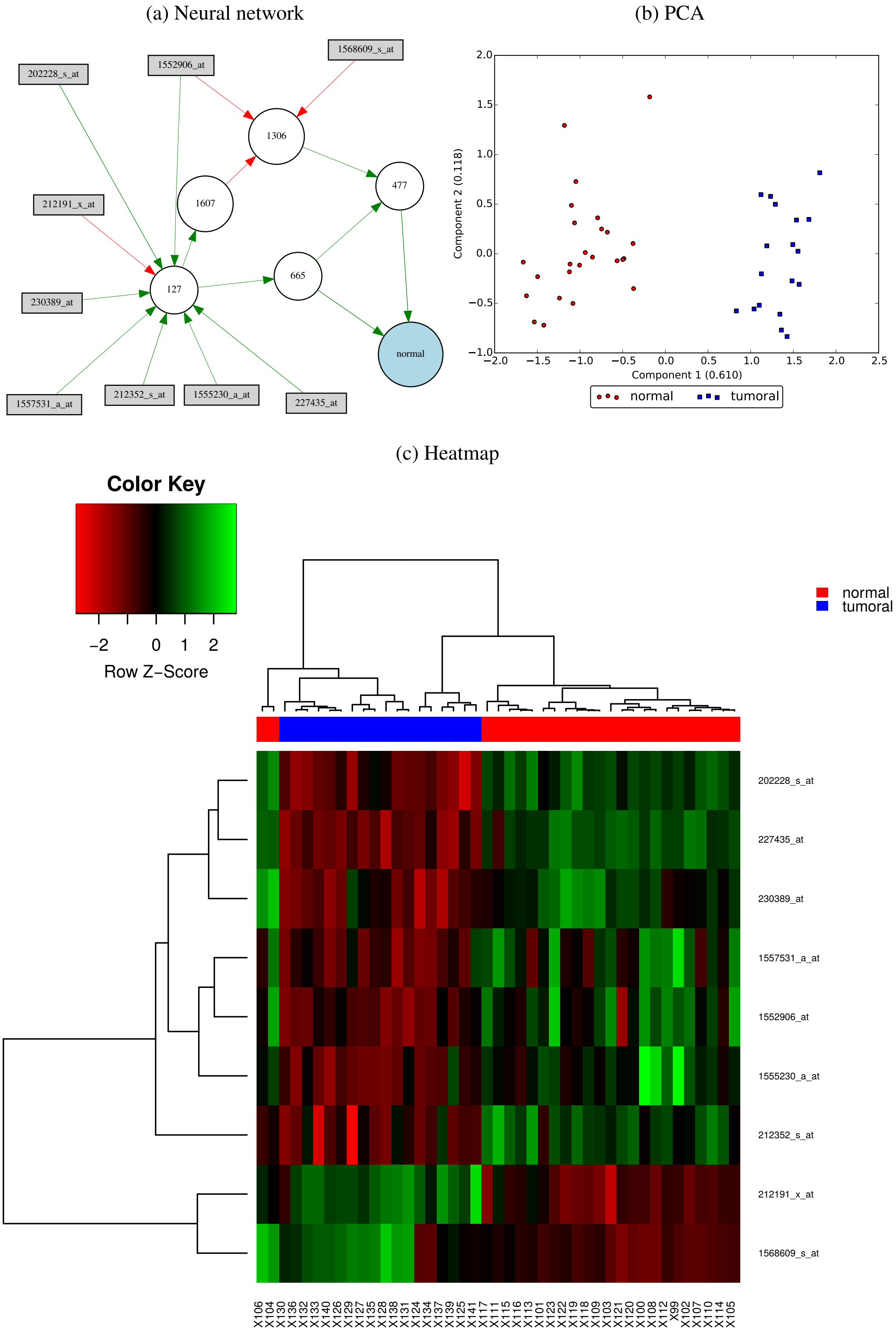


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.20: **Detailment of gene selection for GSE21510 - CRC.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.

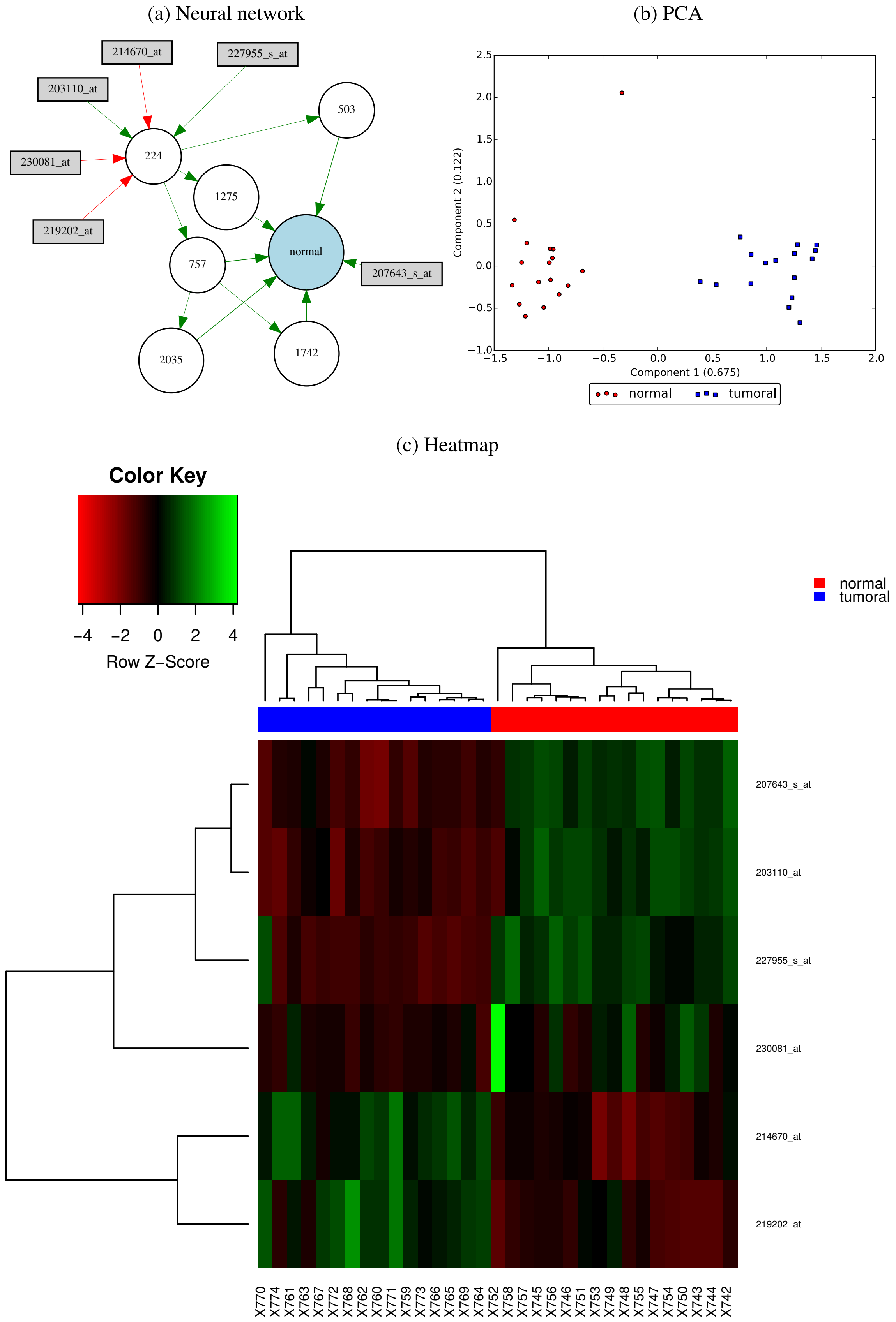


(a) Neural network

(b) PCA

(c) Heatmap

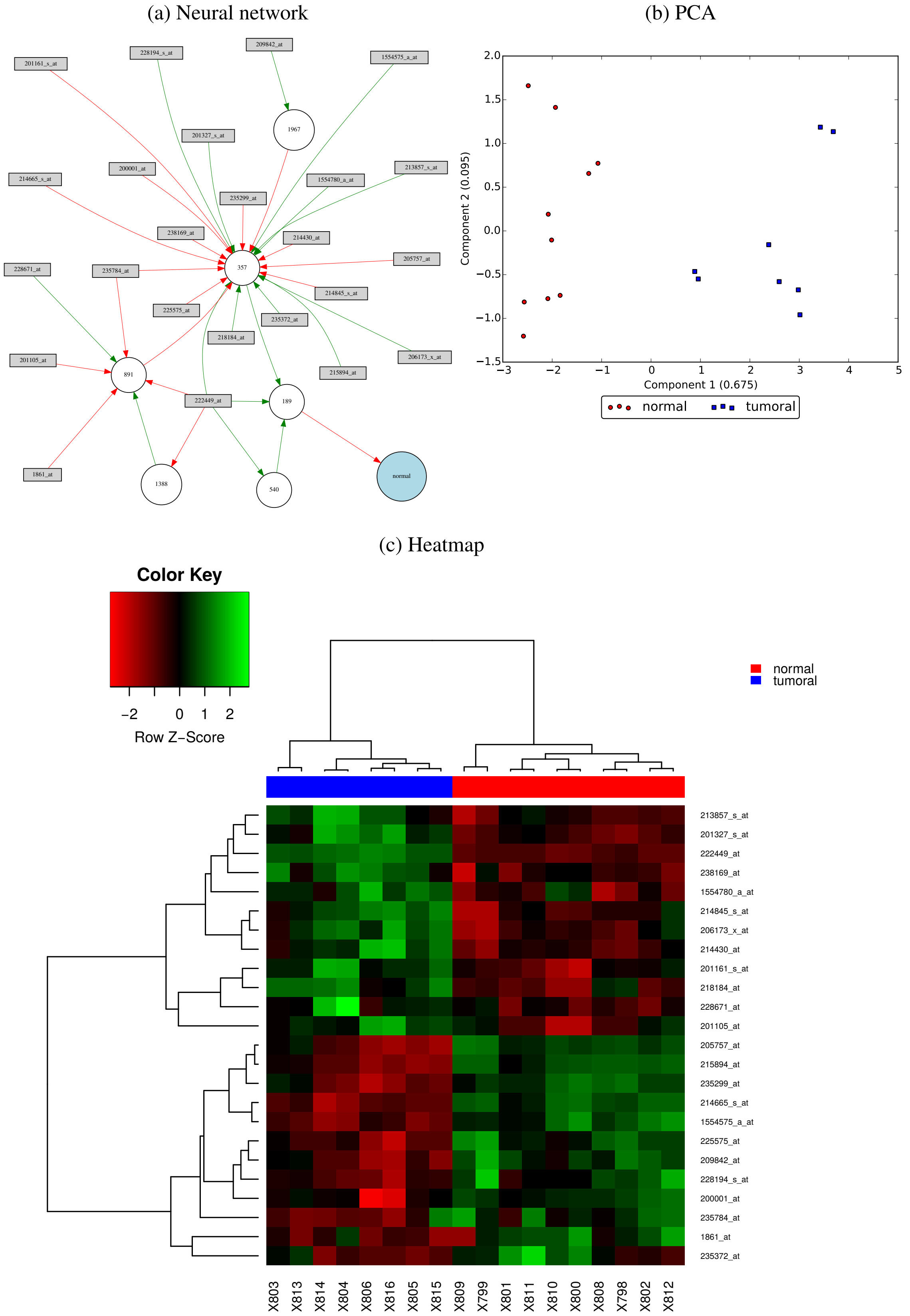
Figure 6.21: **Detailment of gene selection for GSE32323 - CRC.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.


(a) Neural network

(b) PCA

(c) Heatmap

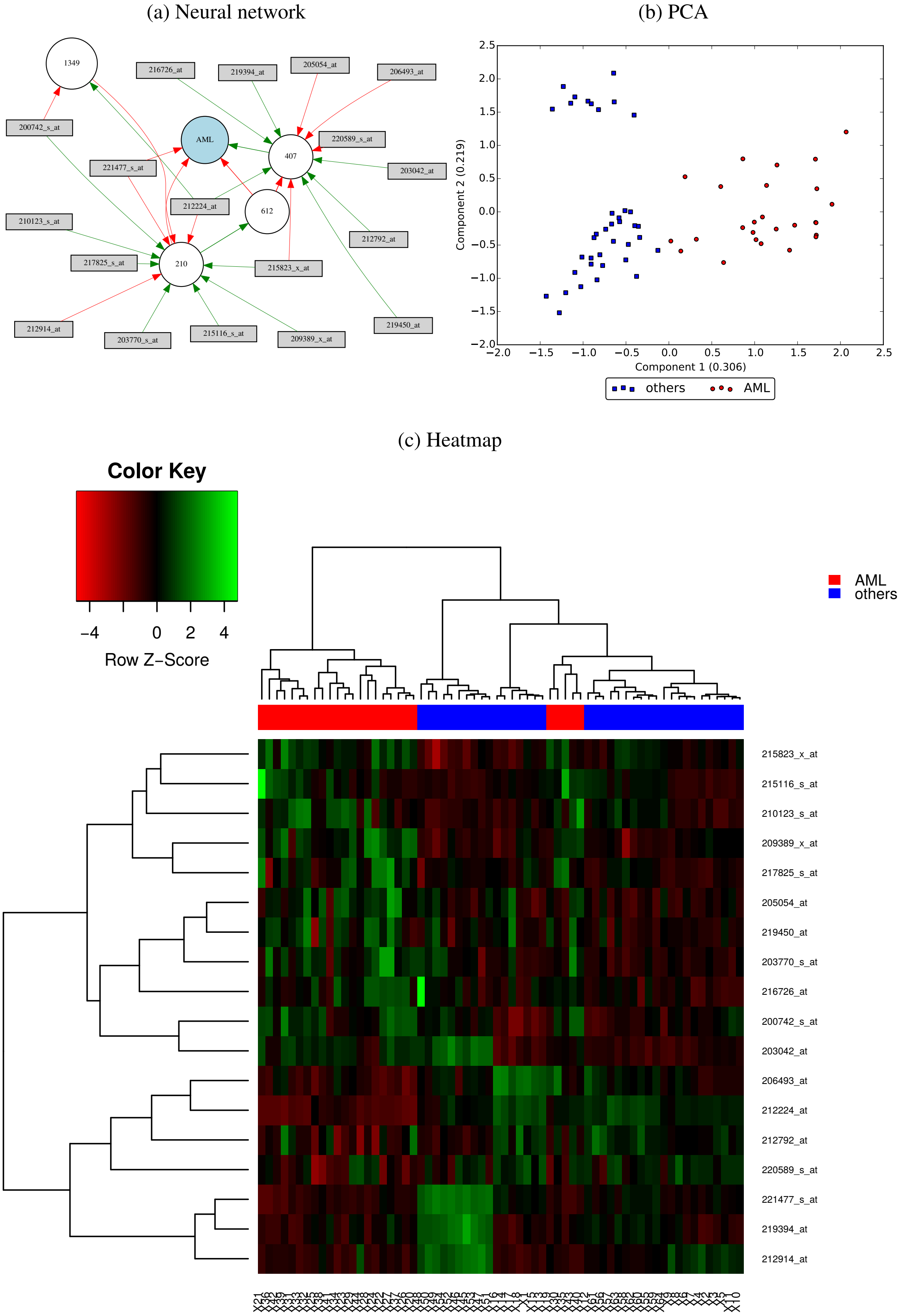
Figure 6.22: **Detailment of gene selection for GSE41328 - CRC Adenocarcinoma.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.23: **Detailment of gene selection for GSE9476 - AML.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.10.

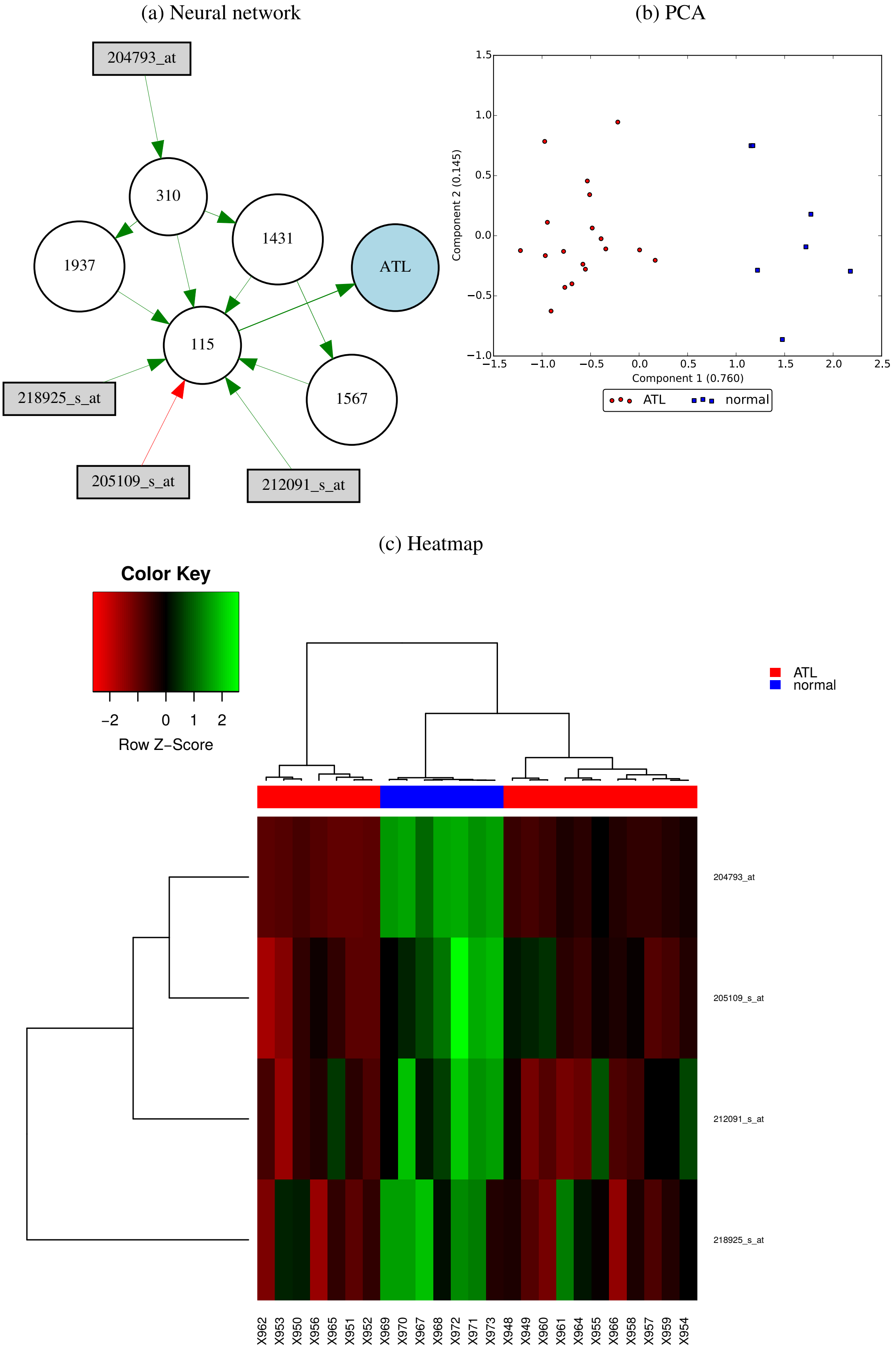


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.24: **Detailment of gene selection for GSE14317 - ATL.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.24.

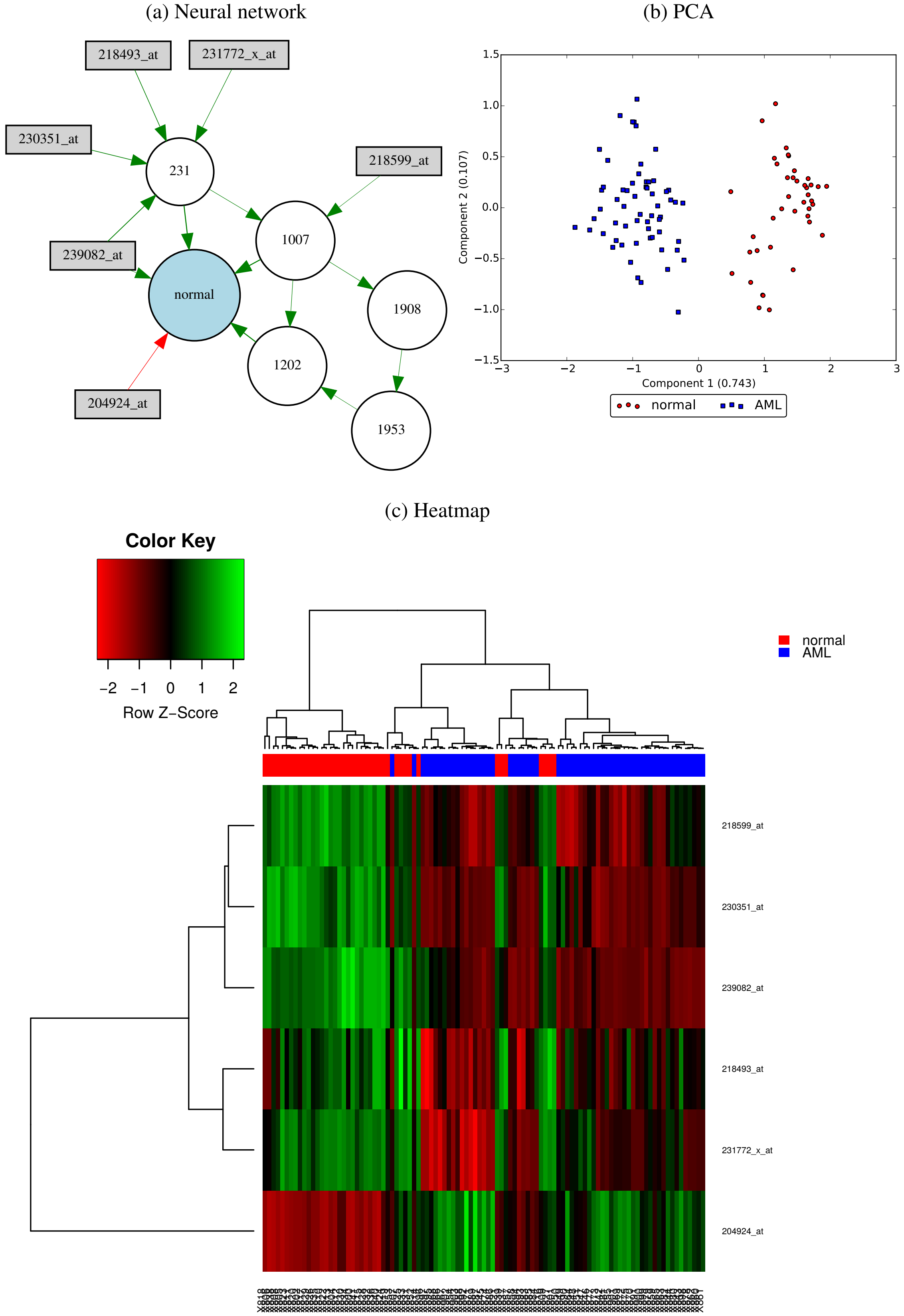


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.25: **Detailment of gene selection for GSE63270 - AML.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.24.

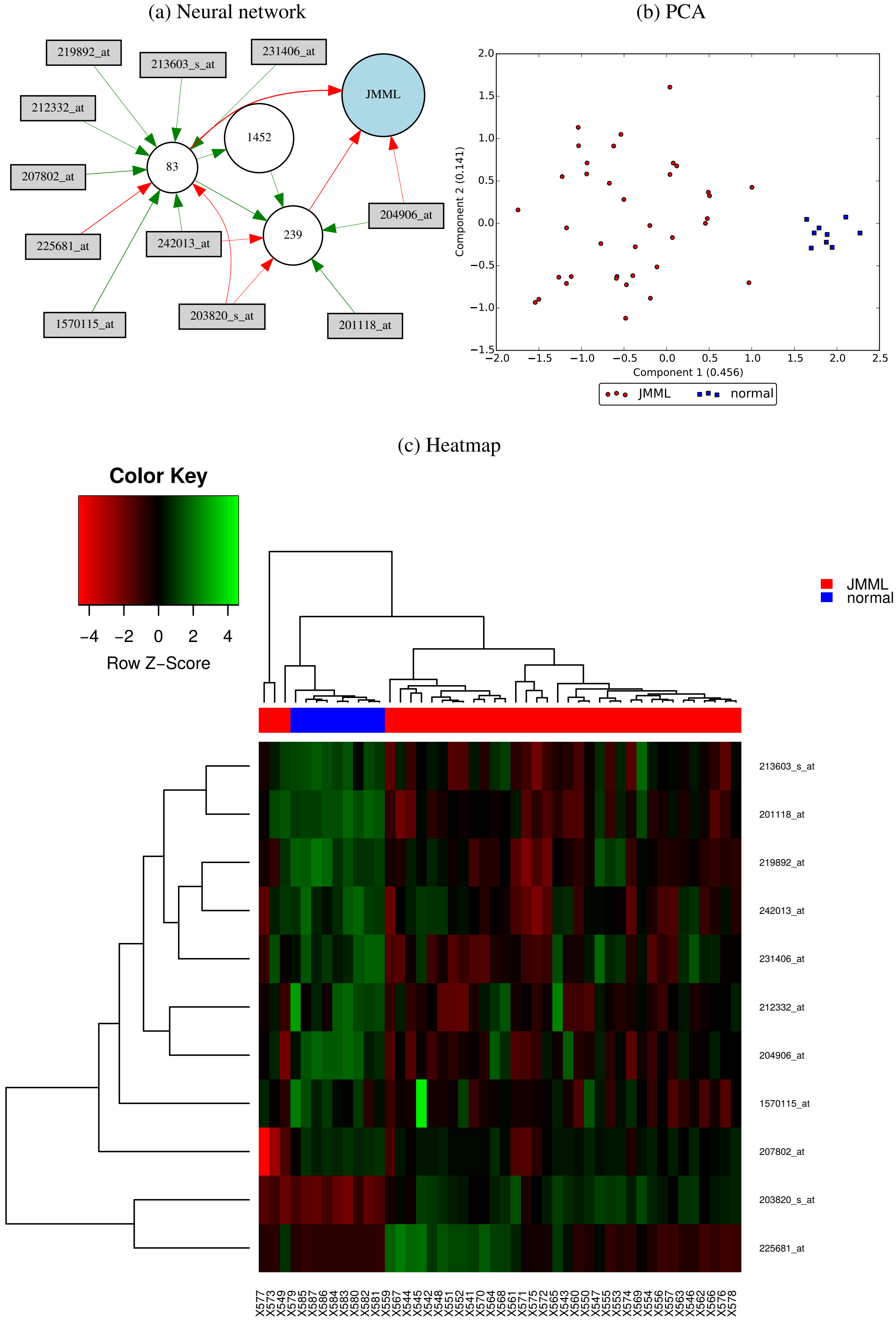


(a) Neural network

(b) PCA

(c) Heatmap

Figure 6.26: **Detailment of gene selection for GSE71935 - JMML.** (a) The best neural network. (b) Principal component analysis of the genes expression. (c) Heatmap of raw gene expression of the selected genes. All details as in Fig. 6.24.


(a) Neural network

(b) PCA

(c) Heatmap

Table 6.9: **Number of associated genes obtained from each GSE.**

GSEs-Cancer	**Genes**	**Hits**	**lncRNA**	**NHF**	**Other**
GSE42568 - Breast Cancer	8	4	1	2	1
GSE45827 - Breast Basal	9	5	NA	2	2
GSE45827 - Breast LuminalA	6	5	1	1	NA
GSE45827 - Breast LuminalB	7	2	NA	5	NA
GSE45827 - Breast HER	6	3	NA	3	NA
GSE10797 - Breast Epithelium	8	5	1	1	2
GSE10797 - Breast Stromal	12	6	NA	3	3
GSE44076 - CRC Adenocarcinoma	12	7	NA	2	3
GSE44861 - CRC	8	5	NA	2	1
GSE8671 - CRC Adenoma	23	10	1	7	6
GSE21510 - CRC	9	1	NA	3	5
GSE32323 - CRC	6	2	NA	1	3
GSE41328 - CRC Adenocarcinoma	24	10	NA	7	7
GSE9476 - AML	18	8	NA	6	4
GSE14317 - ATL	4	1	NA	2	1
GSE63270 - AML	6	3	1	2	1
GSE71935 - JMML	11	5	NA	1	5

Hits = Genes that were already observed to be expressed in the GSE's cancer type; lncRNA = Long non-coding RNA; NHF = No Hits Found. Number of genes that were either not found to be related to any type of cancer in the scientific literature, or that don't have a clear described function so far; Other = Number of genes not observed in the GSE's cancer type, but already found to be expressed in other types of cancer; NA = Not Applicable; CRC = Colorectal Cancer; AML = Acute Myeloid Leukemia; ATL = Adult T-Cell Leukemia/Lymphoma; JMML = Juvenile myelomonocytic Leukemia; HER = Breast Cancer - HER Status. Table made in collaboration with Dr. Bruno César Feltes - SBCB Lab, INF-UFRGS.

Table 6.10: **Table listing all genes that were selected by our approach.** The table brings the probe number, gene symbol, the biochemical function, the cancer type if it was already observed in the literature, the class in which it was selected and their respective references.

Probe	Gene Symbol	Biochemical Function	Cancer Type‡	Class	References*
1553243_at	ITIH5	Inter-alpha-Trypsin inhibitor	BC	BC	(ROSE et al., 2017; BENEZEDER et al., 2017)
201287_s_at	SDC1	Transmembrane Heparan Sulfate Proteoglycan	BC	BC	(CUI et al., 2017)
220757_s_at	UBXN6/UBXD1	UBX Domain Protein	Several	BC	(REZVANI, 2016)
227530_at	AKAP12	A-Kinase Anchoring Protein	BC	BC	(MARINO et al., 2014)
235201_at	FOXP2	Transcription Factor	BC	BC	(WU et al., 2018b; CUIFFO; KARNOUB, 2015)
235362_at	LOC729970	lncRNA	ND	BC	NA
237351_at	ENSG00000232079	ND	ND	BC	NA
237390_at	Hs.52931/ADRA1A	G protein-coupled receptor	HC	BC	(WANG et al., 2013)
202908_at	WFS1	Transmembrane Glycoprotein	HC	BC-Estromal	(HAN et al., 2013)
205249_at	EGR2	Transcription Factor	BC	BC-Estromal	(KENNEDY; HARRIS, 2018; DILLON et al., 2007)
205604_at	HOXD9	Transcription Factor	BC	BC-Estromal	(DEINNOCENTES et al., 2015)
209194_at	CETN2	Calcium-binding protein	BC	BC-Estromal	(HUAN et al., 2014)
209392_at	ENPP2	Pyrophosphatase/Phosphodiesterase	BC	BC-Estromal	(SCHULTE et al., 2012; CASTELLANA et al., 2012)
210031_at	CD247	T-cell receptor zeta	BC	BC-Estromal	(VARCHETTA et al., 2007)
210455_at	C10orf28/R3HCC1L	ND	ND	BC-Estromal	NA
211635_x_at	IGHG1	Immunoglobulin	BC	BC-Estromal	(KABBAGE et al., 2008)
211915_s_at	TUBB7P	Pseudogene	ND	BC-Estromal	NA
213368_x_at	PPFIA3	Tyrosine phosphatase-interacting protein	GC	BC-Estromal	(LI et al., 2016)
213893_x_at	PMS2P5	Pseudogene	ND	BC-Estromal	NA
220765_s_at	LIMS2/PINCH2	Focal adhesion proteins	GC, CRC	BC-Estromal	(PARK et al., 2015; KIM et al., 2006)
201438_at	COL6A3	Collagen	CRC	BC-Epithelium	(LIU et al., 2018)
201655_s_at	HSPG2	Proteoglycan	BC	BC-Epithelium	(VALLADARES et al., 2006)
203371_s_at	NDUFB3	NADH: Ubiquinone Oxidoreductase	ND	BC-Epithelium	NA
205096_at	POM121/P145	Transmembrane Nucleoporin	BC	BC-Epithelium	(LAWRY et al., 1990)
205936_s_at	HK3	Hexokinase	BC	BC-Epithelium	(HARAMI-PAPP et al., 2016)
209758_s_at	MFAP5	Microfibril Associated Protein	BC	BC-Epithelium	(WU et al., 2018c)
214270_s_at	MAPRE3/EBF3	DNA-binding transcription factor	SC, AML	BC-Epithelium	(TAO et al., 2015; ARTOMOV et al., 2017)

Continued on next page

Table 6.10 – *Continued from previous page*

Probe	**Gene Symbol**	**Biochemical Function**	**Cancer Type‡**	**Class**	**References***
216240_at	PVT1	lncRNA	BC	BC-Epithelium	(TANG et al., 2016)
201535_at	UBL3	Membrane-Anchored Ubiquitin-Fold Protein	TC	BC-Basal	(SINGH et al., 2015)
205376_at	INPP4B	Inositol Polyphosphate-4-Phosphatase	BC	BC-Basal	(CROFT et al., 2017; TESSIER-CLOUTIER et al., 2017)
205819_at	MARCO	Macrophage Receptor	ND	BC-Basal	NA
206260_at	TGM4	Transglutaminase	PC	BC-Basal	(SHAIKHIBRAHIM et al., 2011; SHAN et al., 2017)
208894_at	HLA-DRA	Histocompatibility Complex, membrane-bound	BC	BC-Basal	(TRUAX; THAKKAR; GREER, 2012)
218211_s_at	MLPH	Rab effector protein	BC	BC-Basal	(THAKKAR et al., 2010; THAKKAR et al., 2015)
221030_s_at	ARHGAP24	Rho GTPase Activating Protein	BC	BC-Basal	(UEHARA et al., 2017)
222457_s_at	LIMA1/EPLIN	Cytoskeleton-associated Protein	BC	BC-Basal	(JIANG et al., 2008)
238865_at	PABPC4L	Poly(A) Binding Protein Cytoplasmic	ND	BC-Basal	NA
201262_s_at	BGN	Leucine-rich Proteoglycan	BC	BC-LuminalA	(VALLADARES et al., 2006; CASTELLANA et al., 2012)
203930_s_at	MAPT	Microtubule Associated Protein	BC	BC-LuminalA	(LARA-PADILLA et al., 2016)
204092_s_at	AURKA	Kinase	BC	BC-LuminalA	(SANTPERE et al., 2017; LYKKESFELDT et al., 2016)
211110_s_at	AR	Androgen Receptor	BC	BC-LuminalA	(KHATUN et al., 2018)
228554_at	PGR	Progesterone Receptor	BC	BC-LuminalA	(PIROUZPANAH et al., 2018; KUROZUMI et al., 2017)
244035_at	AF086063	lncRNA	ND	BC-LuminalA	NA
1558631_at	PPARA	Peroxisome Proliferator Receptor	BC	BC-LuminalB	(GOLEMBESKY et al., 2008; WU et al., 2012)
1566865_at	FAM200A	ND	ND	BC-LuminalB	NA
213557_at	Hs.444858	ND	ND	BC-LuminalB	NA
226363_at	ABCC5	ATP Binding Transporter	BC	BC-LuminalB	(LAL et al., 2017)
231059_x_at	SCAND1	Zinc Finger proteins	ND	BC-LuminalB	NA
233003_at	Hs.677080	ND	ND	BC-LuminalB	NA
233571_x_at	PPDPF	Cell Differentiation And Proliferation Factor	ND	BC-LuminalB	NA
204914_s_at	SOX11	Transcription Factor	BC	BC-HER	(OLIEMULLER et al., 2017)
210930_s_at	ERBB2	Tyrosine Kinase	BC	BC-HER	(KEUP et al., 2018)
227900_at	CBLB	Adenosyltransferase	BC	BC-HER	(CHEN et al., 2018)
240544_at	N23033	ND	ND	BC-HER	NA
241712_at	Hs.735278	ND	ND	BC-HER	NA

Continued on next page

Table 6.10 – *Continued from previous page*

Probe	**Gene Symbol**	**Biochemical Function**	**Cancer Type‡**	**Class**	**References***
241978_at	Hs.731618	ND	ND	BC-HER	NA
202025_x_at	ACAA1	Acetyl-CoA Acyltransferase	CRC	CRC	(KLIMOSCH et al., 2013)
205697_at	SCGN	Calcium Binding Protein	CRC	CRC	(YANG et al., 2018)
206409_at	TIAM1	Nucleotide exchange factor	CRC	CRC	(YU et al., 2013)
207003_at	GUCA2A	Guanylate Cyclase Activator	CRC	CRC	(LAURIOLA et al., 2010)
209442_x_at	ANK3	Integral membrane protein	CRC	CRC	(YEON et al., 2017)
213389_at	ZNF592	Zinc Finger protein	ND	CRC	NA
216745_x_at	AK024606	ND	ND	CRC	NA
218599_at	REC8	Cohesin	GC, TRC	CRC	(LIU et al., 2015; YU et al., 2017b)
1558949_at	Hs.520638/TNRC18	ND	ND	CRC-Adenoma	NA
1563107_at	ENSG00000233215	lncRNA	ND	CRC-Adenoma	NA
1569064_at	C15orf62	ND	ND	CRC-Adenoma	NA
201064_s_at	PABPC4	Poly(A) Binding Protein Cytoplasmic	PC, LC, BC	CRC-Adenoma	(KHARAZIHA et al., 2015; HSU et al., 2016; KOSTIANETS et al., 2012)
201195_s_at	SLC7A5	Solute Carrier protein	CRC	CRC-Adenoma	(**??**)
201970_s_at	NASP	H1 histone binding protein	GC, OC	CRC-Adenoma	(YU et al., 2017a; ALI-FEHMI et al., 2010)
202061_s_at	SEL1L	Ligase Adaptor Subunit	CRC	CRC-Adenoma	(ASHKTORAB et al., 2012)
204272_at	LGALS4	Beta-galactoside-binding protein	CRC	CRC-Adenoma	(RODIA et al., 2017)
205718_at	ITGB7	Integrin	CRC	CRC-Adenoma	(ORTEGA et al., 2010)
205825_at	PCSK1	Proprotein Convertase	HC, PC, LC	CRC-Adenoma	(RAMALINGAM et al., 2016; MALOUF et al., 2014; DEMIDYUK et al., 2013)
207734_at	LAX1	Lymphocyte Transmembrane protein	CLL	CRC-Adenoma	(JOHNSTON et al., 2018)
207961_x_at	MYH11	Myosin	CRC	CRC-Adenoma	(JO et al., 2018)
208800_at	SRP72	Ribonucleoprotein	PC, TRC	CRC-Adenoma	(CHAI et al., 2016; LYU et al., 2017)
213258_at	TFPI	Serine protease inhibitor	CRC	CRC-Adenoma	(KURER, 2007)
213552_at	GLCE	Glucuronic Acid Epimerase	BC, PC	CRC-Adenoma	(BELYAVSKAYA et al., 2017; SUHOVSKIH et al., 2014)
218694_at	ARMCX1/ALEX1	N-terminal transmembrane protein	CRC	CRC-Adenoma	(ISEKI et al., 2012)
219595_at	ZNF26	Zinc Finger Protein	ND	CRC-Adenoma	NA
219752_at	RASAL1	GTPase-activating protein	CRC	CRC-Adenoma	(AYTEKIN; OZASLAN; CENGIZ, 2010)
225807_at	AJUBA	Complex Adapter protein	CRC	CRC-Adenoma	(JIA et al., 2017; YANG et al., 2017)

Continued on next page

Table 6.10 – *Continued from previous page*

Probe	Gene Symbol	Biochemical Function	Cancer Type‡	Class	References*
225909_at	ZNF775	Zinc Finger Protein	ND	CRC-Adenoma	NA
227253_at	CP	Ferroxidase	CRC	CRC-Adenoma	(MATEO; MARTÍN, 1988)
241815_at	Hs.551393	ND	ND	CRC-Adenoma	NA
242384_at	Hs.605187	ND	ND	CRC-Adenoma	NA
11719018_at	CBFB	Transcription factor	CRC	CRC-Adenocarcinoma	(ANDERSEN et al., 2009)
11722527_s_at	PTPN21	Protein Tyrosine Phosphatase	CRC	CRC-Adenocarcinoma	(KORFF et al., 2008)
11724871_a_at	CLDN2	Integral membrane protein	CRC	CRC-Adenocarcinoma	(BUJKO et al., 2015)
11733581_a_at	CA7	Carbonic Anhydrase	CRC	CRC-Adenocarcinoma	(SHANGKUAN et al., 2017a)
11733707_x_at	COL11A1	Collagen	CRC	CRC-Adenocarcinoma	(ZHANG; ZHU; HARPAZ, 2016)
11740105_x_at	TMEM17	Transmembrane Protein	LC	CRC-Adenocarcinoma	(ZHANG et al., 2017c)
11740441_a_at	APOBEC3A	Cytidine deaminase	LC	CRC-Adenocarcinoma	(WANG et al., 2018)
11744487_x_at	CNBP	Zinc Finger nucleic-acid binding protein	ND	CRC-Adenocarcinoma	NA
11744691_x_at	ARMC10	Transmembrane protein	SC	CRC-Adenocarcinoma	(TURNER et al., 2017)
11757530_a_at	C19orf53	ND	ND	CRC-Adenocarcinoma	NA
11758083_s_at	HPGD	15-Hydroxyprostaglandin Dehydrogenase	CRC	CRC-Adenocarcinoma	(PEREIRA et al., 2016)
11762923_x_at	MT-CO2	Cytochrome C Oxidase	CRC	CRC-Adenocarcinoma	(ERRICHIELLO et al., 2015)
1552906_at	FMR1NB	Cancer/Testis Antigen	TTC	CRC	(CAPPELL et al., 2012)
1555230_a_at	KCNIP2	Potassium Voltage Channel Interacting Protein	ND	CRC	NA
1557531_a_at	C10orf55	ND	ND	CRC	NA
1568609_s_at	ENSG00000232151	ND	ND	CRC	NA
202228_s_at	NPTN	Transmembrane protein	LC	CRC	(KETTUNEN et al., 2017)
212191_x_at	RPL13	Large Ribosomal Subunit	CRC	CRC	(XU et al., 2017)
212352_s_at	TMED10	Transmembrane protein	HC	CRC	(SARAN et al., 2016)
227435_at	KIAA2018	Transcription Factor	LC, TRC	CRC	(NI et al., 2017; RENIERI et al., 2014)
230389_at	FNBP1	Formin-binding-protein	AML	CRC	(KRUMBHOLZ et al., 2015)
1554575_a_at	BPNT1	5'-Bisphosphate Nucleotidase	HNC	CRC-Adenocarcinoma	(AN et al., 2015)
1554780_a_at	PHTF2	Transcription Factor	ND	CRC-Adenocarcinoma	NA
1861_at	BAD	BCL2 Associated Protein	Anti-tumor	CRC-Adenocarcinoma	(STICKLES et al., 2015)

Continued on next page

Table 6.10 – *Continued from previous page*

Probe	**Gene Symbol**	**Biochemical Function**	**Cancer Type‡**	**Class**	**References***
200001_at	CAPNS1	Calcium-Dependent Protease	BC	CRC-Adenocarcinoma	(RAIMONDI et al., 2016)
201105_at	LGALS1	Beta-galactoside-binding protein	CRC	CRC-Adenocarcinoma	(LI et al., 2017)
201161_s_at	CSDA/YBX3	DNA-Binding Protein	RC	CRC-Adenocarcinoma	(DUPASQUIER et al., 2014)
201327_s_at	CCT6A	Chaperone	SC, RC	CRC-Adenocarcinoma	(ZHU et al., 2017; TANIC et al., 2006)
205757_at	ENTPD5	Triphosphate Diphosphohydrolase	CRC	CRC-Adenocarcinoma	(PIZZINI et al., 2013)
206173_x_at	GABPB1	Transcription Factor	SC	CRC-Adenocarcinoma	(ZHANG et al., 2017b)
209842_at	SOX10	Transcription Factor	CRC	CRC-Adenocarcinoma	(TONG et al., 2014)
213857_s_at	CD47	Membrane protein	CRC	CRC-Adenocarcinoma	(THEAN et al., 2018)
214430_at	GLA	Galactosidase	ND	CRC-Adenocarcinoma	NA
214665_s_at	CHP1	Phosphoprotein	CRC	CRC-Adenocarcinoma	(GALAMB et al., 2016)
214845_s_at	CALU	Calcium-binding protein	CRC	CRC-Adenocarcinoma	(TORRES et al., 2013)
215894_at	PTGDR	Prostaglandin D2 Receptor	CRC	CRC-Adenocarcinoma	(**??**)
218184_at	TULP4	ND	ND	CRC-Adenocarcinoma	NA
222449_at	PMEPA1	Transmembrane Protein	CRC	CRC-Adenocarcinoma	(SHEFFER et al., 2009)
225575_at	LIFR	Type I cytokine receptor	CRC	CRC-Adenocarcinoma	(WU et al., 2018a)
228194_s_at	SORCS1	Vacuolar protein receptor	CRC	CRC-Adenocarcinoma	(HUA et al., 2017)
228671_at	TMEM201	Transmembrane Protein	ND	CRC-Adenocarcinoma	NA
235299_at	SLC41A2	Solute Carrier protein	ND	CRC-Adenocarcinoma	NA
235372_at	FCRLA	Fc Receptor-Related Protein	CLL	CRC-Adenocarcinoma	(LI et al., 2008)
235784_at	N32155	ND	ND	CRC-Adenocarcinoma	NA
238169_at	AI307778	ND	ND	CRC-Adenocarcinoma	NA
203110_at	PTK2B	Tyrosine Kinase	CRC	CRC	(OH et al., 2017)
207643_s_at	TNFRSF1A	TNF Receptor	CRC	CRC	(YU et al., 2014)
214670_at	ZKSCAN1	Transcription Factor	HC	CRC	(Z et al., 2017)
219202_at	RHBDF2	ND	GC	CRC	(ISHIMOTO et al., 2017)
227955_s_at	EFNA5	Tyrosine Kinase Ligand	GC, PC	CRC	(ZHU et al., 2015; ROSENBERG et al., 2017)
230081_at	PLCXD3	Phospholipase	ND	CRC	NA
204793_at	GPRASP1	G Protein-Coupled Receptor	ND	LKM-ATL	NA

Continued on next page

Table 6.10 – *Continued from previous page*

Probe	Gene Symbol	Biochemical Function	Cancer Type‡	Class	References*
205109_s_at	ARHGEF4	Rho Guanine Nucleotide Exchange Factor	ALL	LKM-ATL	(LYONS et al., 2010)
212091_s_at	COL6A1	Collagen	PC, RC, CC	LKM-ATL	(ZHU et al., 2015; WAN et al., 2015; HOU et al., 2016)
218925_s_at	C11orf1	ND	ND	LKM-ATL	NA
204924_at	TLR2	Toll-like receptor	AML, CLL	LKM-AML	(ERIKSSON et al., 2017; WILLIAMS; ARIZA, 2018)
218493_at	SNRNP25	Nuclear Ribonucleoprotein	ND	LKM-AML	NA
218599_at	REC8	Meiotic structural protein	GC, TRC	LKM-AML	(LIU et al., 2015; YU et al., 2017b)
230351_at	LOC283481	lncRNA	ND	LKM-AML	NA
231772_x_at	CENPH	Kinetochore protein	MDS	LKM-AML	(LEE et al., 2012)
239082_at	FZD3	Transmembrane Receptor	AML, CLL	LKM-AML	(KAUCKá et al., 2013; ZHANG et al., 2017a)
1570115_at	Hs.684470	ND	ND	LKM-JMML	NA
201118_at	PGD/6PGD	Phosphogluconate Dehydrogenase	AML	LKM-JMML	(BHANOT et al., 2017)
203820_s_at	IGF2BP3	Insulin-Like Growth Factor	ALL	LKM-JMML	(STOSKUS et al., 2011)
204906_at	RPS6KA2	Serine/threonine kinase	CRC, PCC	LKM-JMML	(MILOSEVIC et al., 2013; SLATTERY et al., 2011)
207802_at	CRISP3	Cysteine Rich Secretory Protein	PC	LKM-JMML	(PATHAK et al., 2018)
212332_at	RBL2	Transcriptional Corepressor	ATL	LKM-JMML	(TAKEUCHI et al., 2003)
213603_s_at	RAC2	GTP-metabolizing protein	JMML, CLL	LKM-JMML	(CAYE et al., 2015; NIEBOROWSKA-SKORSKA et al., 2012)
219892_at	TM6SF1	Transmembrane protein	BC	LKM-JMML	(GROOT et al., 2014)
225681_at	CTHRC1	Collagen-associated protein	LC, CRC, HC	LKM-JMML	(HE et al., 2018; LIU et al., 2018; WANG et al., 2018)
231406_at	ORAI2	Calcium-release Channel	AML	LKM-JMML	(DIEZ-BELLO et al., 2017)
242013_at	BCL2L15	ND	Several	LKM-JMML	(NIAVARANI et al., 2018)
200742_s_at	TPP1	Tripeptidyl Peptidase	CLL	LKM-AML	(GUIèZE et al., 2017)
203042_at	LAMP2	Membrane glycoprotein	AML	LKM-AML	(SUKHAI et al., 2013)
203770_s_at	STS	Steroid Sulfatase	AML	LKM-AML	(HUGHES et al., 2005)
205054_at	NEB	Cytoskeleton structural component	ND	LKM-AML	NA
206493_at	ITGA2B	Fibronectin receptor	AML	LKM-AML	(HUANG; LIAO; LI, 2017)
209389_x_at	DBI/ACBP	Diazepam Binding Inhibitor	TC	LKM-AML	(DLAMINI et al., 2017)
210123_s_at	CHRNA7	Cholinergic Receptor	T-ALL	LKM-AML	(CHAKHACHIRO et al., 2013)
212224_at	ALDH1A1	Aldehyde Dehydrogenase	AML	LKM-AML	(GASPARETTO; SMITH, 2017a)

Continued on next page

Table 6.10 – *Continued from previous page*

Probe	Gene Symbol	Biochemical Function	Cancer Type‡	Class	References*
212792_at	DPY19L1	C-mannosyltransferase	CRC	LKM-AML	(MáRQUEZ et al., 2013)
212914_at	CBX7	Polycomb repressive complex component	CML	LKM-AML	(CREA et al., 2015)
215116_s_at	DNM1	GTP-binding protein	BRC	LKM-AML	(PATEL et al., 2013)
215823_x_at	RLIM	E3 ubiquitin protein ligase	BOC, BC	LKM-AML	(JOHNSEN et al., 2009; CHEN et al., 2014)
216726_at	Hs.447377	ND	ND	LKM-AML	NA
217825_s_at	UBE2J1	Ubiquitin Conjugating Enzyme	ND	LKM-AML	NA
219394_at	PGS1	Phosphatidylglycerophosphate Synthase	ND	LKM-AML	NA
219450_at	C4orf19	ND	ND	LKM-AML	NA
220589_s_at	ITFG2	ND	ND	LKM-AML	NA
221477_s_at	SOD2	Superoxide Dismutase	ALL, ABL	LKM-AML	(ALACHKAR et al., 2017; GIRERD et al., 2018)

‡= Other cancer types that the selected genes were observed to be altered in some way; * = when multiple references were available we gave preference to citations from the last 5 years, except when they could be complementary; ABL = Chronic Myeloid Leukemia BCR-ABL fusion; ALL = Acute Myeloid Leukemia; ATL = Acute T-cell Leukemia; BC = Breast Cancer; BOC = Bone Cancer; BRC = Brain Cancer; CC = Cervical Cancer; CLL = Chronic Lymphocytic Leukemia; CRC = Colorectal Cancer; GC = Gastric Cancer; HC = Hepatic Cancer; HNC = Head and Neck Cancer; JMML = Juvenile Myelomonocytic Leukemia; LC = Lung Cancer; MDS = Myelodysplastic Syndrome; NA = Not Applicable; ND = Not Defined; OC = Ovarian Cancer; PC = Prostate Cancer; PCC = Pancreatic Cancer; RC = Renal Cancer; SC = Skin Cancer; T-ALL = T-lymphoblastic Leukemia; TC = Throat Cancer; TRC = Thyroid Cancer; TTC = testicular Cancer. The references were obtained from the PubMed repository. Table made in collaboration with Dr. Bruno César Feltes - SBCB Lab, INF-UFRGS.

As mentioned in Section 6.1.4, DAVID was employed to search for the significant GO and cellular localization of the 177 selected genes, providing a better understanding of the nature of the obtained expression patterns illustrated in Figs. 6.10 to 6.26. Concerning the cellular component, the majority of the genes were related to extracellular exosomes, cell surface, plasma membrane, endoplasmatic reticulum and the cytosol (Fig. 6.27). As for GO, the main bioprocesses were extracellular matrix organization, response to hypoxia, signal transduction, and positive regulation of cell proliferation (Table 6.11).

Figure 6.27: **The number of genes related to the major cellular components.** The five most significant and abundant categories to which the selected genes were classified are related to the extracellular exosomes, cell surface, plasma membrane, endoplasmatic reticulum, and the cytosol. Image made in collaboration with Dr. Bruno César Feltes - SBCB Lab, INF-UFRGS

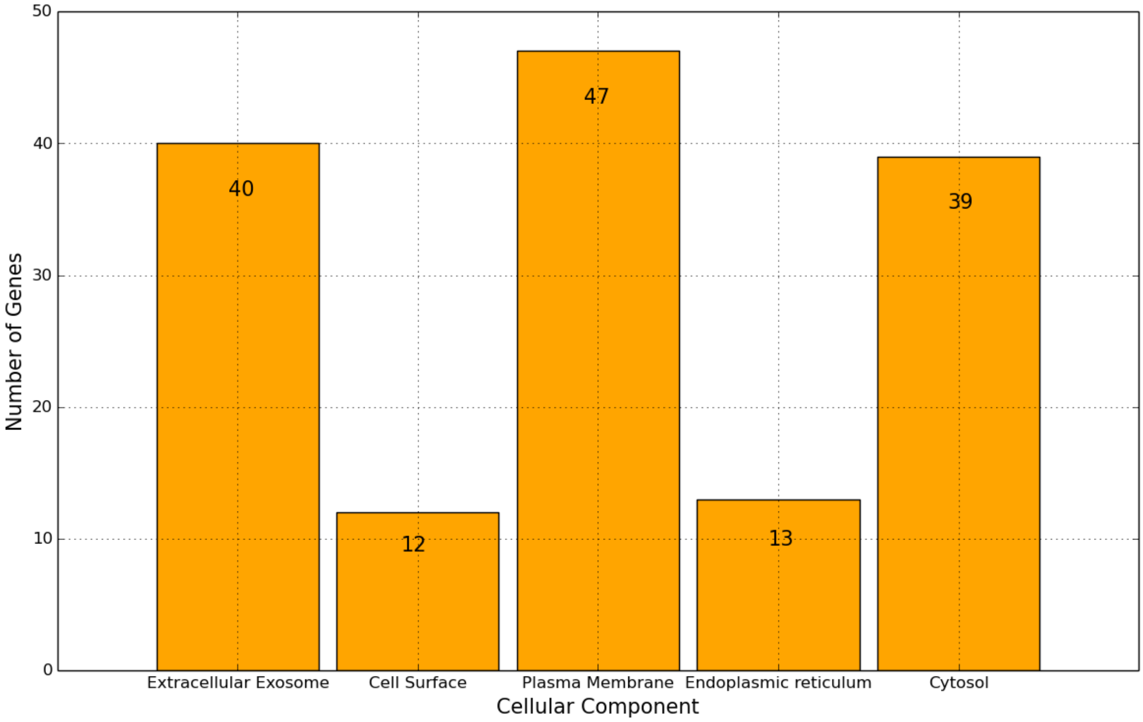


Table 6.11: **Major GO derived from all selected genes.**

Bioprocesses	**Corrected p-value**
Extracellular Matrix Organization	1.9×10^{-1}
Response to Hypoxia	7.7×10^{-1}
Signal Transduction	8.6×10^{-1}
Positive Regulation of Cell proliferation	8.0×10^{-1}

Table made in collaboration with Dr. Bruno César Feltes - SBCB Lab, INF-UFRGS.

6.6 Biological role of selected genes

From the 177 selected genes, 50 genes were either not related to any type of cancer or didn't possess a clear functional description, leaving a total of 127 genes already described in the literature as expressed in some type of cancer. Selecting genes with no described function yet is normal to any expression analysis: there are still many known DNA segments in the human genome with no described function that can impact on cancer biology (TUTAR et al., 2016; EMADI-BAYGI et al., 2017; POLISENO; MARRANCI; PANDOLFI, 2015; SHI et al., 2018; WEDGE et al., 2018), and studies like these have the potential to provide a first glimpse of functional role for such genes. Moreover, among those 127 genes, 82 (64.5%) were related to their specific cancer types, and 44 were observed to be altered in some way in other types, becoming potential targets to be explored in future works. All genes are described in Table 6.5, with their associated cancer types and corresponding references.

Most of the selected genes act in the plasma membrane and extracellular exosomes (Fig. 6.27), known as fundamental aspects of cancer biology (SAITOH, 2018; GKRETSI; STYLIANOPOULOS, 2018; COUTO et al., 2018; MAIA et al., 2018; LIU et al., 2015; FILIPPINI; SICA; D'ALESSIO, 2018; STUELTEN; PARENT; MONTELL, 2018). Another interesting fact is that N3O selected five lncRNAs (Table 6.11). In contrast to mRNAs, lncRNAs do not encode to proteins but are critical transcriptional regulators that modulate gene expression through multiple molecular mechanisms (HU et al., 2018; CHAN; TAY, 2018). Among the five lncRNAs selected by N3O was PVT1 (GSE10797 - Breast cancer), that has already been associated with triple-negative breast cancer (TANG et al., 2018).

6.7 Chapter conclusion

This chapter described several experiments in order to validate N3O, the proposed method. The evolutionary process was illustrated by plots that reveal different aspects of the algorithm. The accuracy and FS of N3O was compared with regular FS-NEAT, SVM, and another neuroevolution method, showing positive results. The generalizability of the genes selected by N3O was also tested by successfully applying them to SVM classification. Finally, the biological role and relevance of the selected genes was assured by a literature review. The next chapter brings a conclusion to this work.

7 CONCLUSION

In this work, a pipeline for microarray classification and gene selection was developed by employing Neuroevolution as a method capable of efficiently performing both tasks autonomously. The preprocessing of microarray datasets prior to the machine learning application to assure better biological results applied in this work was also a highlight. This evolutive method builds upon the FS-NEAT algorithm, adding new operators for better exploration of the search space, and designs neural networks for solving those tasks.

Tested with microarray datasets of three different types of cancer and varying number of samples, features, and classes, it successfully overcame the classification baselines and showed good performance against other algorithms. In the case of SVMs, the use of the features selected by this method did not disturb the classification and, for some cases, even improved it, a result not expected according to the literature and that may show the quality of the selection. The results also pointed to 177 genes involved in specific gene expression patterns that are closely associated to the extracellular matrix, plasma membrane and exosomes, proposing new targets to be explored to uncover the molecular mechanisms underlying colorectal cancer, leukemia and breast cancer. A total of 127 of those genes were already described in the literature as relevant for cancer, 82 of them related to the specific cancer type being analyzed. The successful validation of these targets in the literature also reinforces the efficacy of this approach to correctly classify expression patterns in different types of cancer.

The computational analysis of microarray data remains a challenging task, with several opportunities for further improvements. The problems of overfitting and class imbalance are still hard to overcome, and new strategies, such as the joint analysis of two or more datasets, could help. It is also critical to increase the focus in revealing biological information from the selected genes and to characterize their expressions signature in order to truly provide aid in the biological research or creation of better treatments based on the specific conditions of patients. This work may be improved in the future by the addition of even more structural operators, hyperparameter tuning, and the incorporation of biological information in the fitness function.

REFERENCES

AL., P. L. et. Linc01638 promotes tumorigenesis in her2+ breast cancer. **Current Cancer Drug Targets**, v. 18, p. 1–1, 2018.

ALACHKAR, H. et al. Expression and polymorphism (rs4880) of mitochondrial superoxide dismutase (sod2) and asparaginase induced hepatotoxicity in adult patients with acute lymphoblastic leukemia. **Pharmacogenomics J**, v. 17, n. 3, p. 274–279, 2017.

ALBERTS, B. e. a. **Molecular biology of the cell**. New York: Garland Science, 2015. 1464 p.

ALELYANI, S.; TANG, J.; LIU, H. Feature selection for clustering: A review. **Data Clustering: Algorithms and Applications**, v. 29, p. 110–121, 2013.

ALHAJ, T. A. et al. Feature selection using information gain for improved structural-based alert correlation. **PloS one**, Public Library of Science, v. 11, n. 11, p. e0166017, 2016.

ALI-FEHMI, R. et al. Analysis of the expression of human tumor antigens in ovarian cancer tissues. **Cancer Biomark**, v. 6, n. 1, p. 33–48, 2010.

ALLISON, D. B. et al. Microarray data analysis: from disarray to consolidation and consensus. **Nature reviews genetics**, Nature Publishing Group, v. 7, n. 1, p. 55, 2006.

AN, F. et al. Subpath analysis of each subtype of head and neck cancer based on the regulatory relationship between mirnas and biological pathways. **Oncol Rep**, v. 34, n. 4, p. 1745–1754, 2015.

ANDERSEN, C. et al. Dysregulation of the transcription factors sox4, cbfb and smarcc1 correlates with outcome of colorectal cancer. **Br J Cancer**, v. 100, n. 3, p. 511–523, 2009.

ANG, J. C. et al. Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 13, n. 5, p. 971–989, 2016.

ANGELINE, P. J.; SAUNDERS, G. M.; POLLACK, J. B. An evolutionary algorithm that constructs recurrent neural networks. **IEEE Transactions on Neural Networks**, v. 5, p. 54–65, 1993.

ANGERMUELLER, C. et al. Deep learning for computational biology. **Mol Syst Biol**, v. 12, n. 7, p. 878, 2016.

ARMSTRONG, R. A.; SLADE, S.; EPERJESI, F. An introduction to analysis of variance (anova) with special reference to data from clinical experiments in optometry. **Ophthalmic and Physiological Optics**, Wiley Online Library, v. 20, n. 3, p. 235–241, 2000.

ARTOMOV, M. et al. Rare variant, gene-based association study of hereditary melanoma using whole-exome sequencing. **J Natl Cancer Inst.**, v. 109, n. 12, p. doi: 10.1093/jnci/djx083, 2017.

ASHKTORAB, H. et al. Sel1l, an upr response protein, a potential marker of colonic cell transformation. **Dig Dis Sci**, v. 57, n. 4, p. 905–912, 2012.

AYTEKIN, T.; OZASLAN, M.; CENGIZ, B. Deletion mapping of chromosome region 12q13-24 in colorectal cancer. **Cancer Genet Cytogenet**, v. 201, n. 1, p. 32–38, 2010.

BALUJA, S.; CARUANA, R. Removing the genetics from the standard genetic algorithm. In: **Machine Learning Proceedings 1995**. [S.l.]: Elsevier, 1995. p. 38–46.

BELYAVSKAYA, V. et al. Glce rs3865014 (val597ile) polymorphism is associated with breast cancer susceptibility and triple-negative breast cancer in siberian population. **Gene**, v. 628, p. 224–229, 2017.

BENEZEDER, T. et al. Multigene methylation analysis of enriched circulating tumor cells associates with poor progression-free survival in metastatic breast cancer patients. **Oncotarget**, v. 8, n. 54, p. 92483–92496, 2017.

BENJAMINI, Y.; HOCHBERG, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. **Journal of the royal statistical society. Series B (Methodological)**, JSTOR, p. 289–300, 1995.

BHANOT, H. et al. Acute myeloid leukemia cells require 6-phosphogluconate dehydrogenase for cell growth and nadph-dependent metabolic reprogramming. **Oncotarget**, v. 8, n. 40, p. 67639–67650, 2017.

BOER, P.-T. D. et al. A tutorial on the cross-entropy method. **Annals of operations research**, Springer, v. 134, n. 1, p. 19–67, 2005.

BORGES, V. F. et al. Tucatinib combined with ado-trastuzumab emtansine in advanced erbb2/her2-positive metastatic breast cancer: A phase 1b clinical trial. **JAMA oncology**, 2018.

BOULESTEIX, A.-L. et al. Evaluating microarray-based classifiers: an overview. **Cancer informatics**, SAGE Publications Sage UK: London, England, v. 6, p. CIN–S408, 2008.

BRANKE, J. Evolutionary algorithms for neural network design and training. In: **Proceedings First Nordic Workshop on Genetic Algorithms and their Applications**. Vaasa, Finland: [s.n.], 1995. p. 145–163.

BUJKO, M. et al. Expression changes of cell-cell adhesion-related genes in colorectal tumors. **Oncol Lett**, v. 9, n. 6, p. 2463–2470, 2015.

CAPPELL, K. et al. Multiple cancer testis antigens function to support tumor cell mitotic fidelity. **Mol Cell Biol**, v. 32, n. 20, p. 4131–4140, 2012.

CASTELLANA, B. et al. Aspn and gjb2 are implicated in the mechanisms of invasion of ductal breast carcinomas. **J Cancer**, v. 3, p. 175–183, 2012.

CAYE, A. et al. Juvenile myelomonocytic leukemia displays mutations in components of the ras pathway and the prc2 network. **Nat Genet**, v. 47, n. 11, p. 1334–1340, 2015.

CELIK, S. et al. Methylation analysis of the dapk1 gene in imatinib-resistant chronic myeloid leukemia patients. **Oncology letters**, Spandidos Publications, v. 9, n. 1, p. 399–404, 2015.

CHAI, L. et al. An integrated analysis of cancer genes in thyroid cancer. **Oncol Rep**, v. 35, n. 2, p. 962–970, 2016.

CHAKHACHIRO, Z. et al. Cd105 (endoglin) is highly overexpressed in a subset of cases of acute myeloid leukemias. **American Journal of Clinical Pathology**, v. 140, n. 3, p. 370–378, 2013.

CHAN, J.; TAY, Y. Noncoding rna:rna regulatory networks in cancer. **Int J Mol Sci**, v. 19, n. 5, p. pii: E1310, 2018.

CHEN, D. et al. mir-27b-3p inhibits proliferation and potentially reverses multi-chemoresistance by targeting cblb/grb2 in breast cancer cells. **Cell Death Dis**, v. 9, n. 2, p. 188, 2018.

CHEN, M. et al. Pathway analysis of bladder cancer genome-wide association study identifies novel pathways involved in bladder cancer development. **Genes & cancer**, Impact Journals, LLC, v. 7, n. 7-8, p. 229, 2016.

CHEN, X. et al. Rlim, an e3 ubiquitin ligase, influences the stability of stathmin protein in human osteosarcoma cells. **Cell Signal**, v. 26, n. 7, p. 1532–1538, 2014.

CHING, T. et al. Opportunities and obstacles for deep learning in biology and medicine. **bioRxiv**, Cold Spring Harbor Laboratory, p. 142760, 2018.

CONSORTIUM, G. O. The gene ontology project in 2008. **Nucleic acids research**, Oxford University Press, v. 36, n. suppl_1, p. D440–D444, 2007.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995.

COUTO, N. et al. Exosomes as emerging players in cancer biology. **Biochimie**, pii: S0300-9084, n. 18, p. 30067–1, 2018.

CREA, F. et al. Polycomb genes are associated with response to imatinib in chronic myeloid leukemia. **Epigenomics**, v. 7, n. 5, p. 757–765, 2015.

CROFT, A. et al. Functional identification of a novel transcript variant of inpp4b in human colon and breast cancer cells. **Biochem Biophys Res Commun**, v. 485, n. 1, p. 47–53, 2017.

CUI, X. et al. Clinicopathological and prognostic significance of sdc1 overexpression in breast cancer. **Oncotarget**, v. 8, n. 67, p. 111444–111455, 2017.

CUIFFO, B.; KARNOUB, A. Silencing foxp2 in breast cancer cells promotes cancer stem cell traits and metastasis. **Mol Cell Oncol**, v. 3, n. 3, p. e1019022, 2015.

CURTEANU, S.; CARTWRIGHT, H. Neural networks applied in chemistry. i. determination of the optimal topology of multilayer perceptron neural networks. **Journal of Chemometrics**, Wiley Online Library, v. 25, n. 10, p. 527–549, 2011.

CUSSAT-BLANC, S.; HARRINGTON, K.; POLLACK, J. Gene regulatory network evolution through augmenting topologies. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 19, n. 6, p. 823–837, 2015.

D'AGOSTINO, R.; PEARSON, E. S. Tests for departure from normality. empirical results for the distributions of b 2 and b. **Biometrika**, Oxford University Press, v. 60, n. 3, p. 613–622, 1973.

D'AGOSTINO, R. B. An omnibus test of normality for moderate and large size samples. **Biometrika**, Oxford University Press, v. 58, n. 2, p. 341–348, 1971.

DARB-ESFAHANI, S. et al. Thymosin beta 15a (tmsb15a) is a predictor of chemotherapy response in triple-negative breast cancer. **British journal of cancer**, Nature Publishing Group, v. 107, n. 11, p. 1892, 2012.

DASARI, V. K. et al. Dna methylation regulates the expression of y chromosome specific genes in prostate cancer. **The Journal of urology**, Elsevier, v. 167, n. 1, p. 335–338, 2002.

DAVIS, S.; MELTZER, P. Geoquery: a bridge between the gene expression omnibus (geo) and bioconductor. **Bioinformatics**, v. 14, p. 1846–1847, 2007.

DEINNOCENTES, P. et al. Characterization of hox gene expression in canine mammary tumour cell lines from spontaneous tumours. **Vet Comp Oncol**, v. 13, n. 3, p. 322–336, 2015.

DEMIDYUK, I. et al. Alterations in gene expression of proprotein convertases in human lung cancer have a limited number of scenarios. **PLoS One**, v. 8, n. 2, p. e55752, 2013.

DENIZ, E.; ERMAN, B. Long noncoding rna (lincrna), a new paradigm in gene expression control. **Funct Integr Genomics**, v. 17, n. 2-3, p. 135–143, 2017.

DÍAZ-URIARTE, R.; ANDRES, S. A. D. Gene selection and classification of microarray data using random forest. **BMC bioinformatics**, BioMed Central, v. 7, n. 1, p. 3, 2006.

DIEZ-BELLO, R. et al. Orai1 and orai2 mediate store-operated calcium entry that regulates hl60 cell migration and fak phosphorylation. **Biochim Biophys Acta**, v. 1864, n. 6, p. 1064–1070, 2017.

DILLON, R. et al. An egr2/cited1 transcription factor complex and the 14-3-3sigma tumor suppressor are involved in regulating erbb2 expression in a transgenic-mouse model of human breast cancer. **Mol Cell Biol**, v. 27, n. 24, p. 8648–8657, 2007.

DING CAND PENG, H. Minimum redundancy feature selection from microarray gene expression data. **J Bioinform Comput Biol**, v. 3, n. 02, p. 185–205, 2005.

DLAMINI, Z. et al. Significant up-regulation of 1-acbp, b-acbp and pbr genes in immune cells within the oesophageal malignant tissue and a possible link in carcinogenic angiogenesis. **Histol Histopathol**, v. 32, n. 6, p. 561–570, 2017.

DUPASQUIER, S. et al. Validation of housekeeping gene and impact on normalized gene expression in clear cell renal cell carcinoma: critical reassessment of ybx3/zonab/csda expression. **BMC Mol Biol**, v. 15, p. 9, 2014.

EMADI-BAYGI, M. et al. Pseudogenes in gastric cancer pathogenesis: a review article. **Brief Funct Genomics**, v. 16, n. 6, p. 348–360, 2017.

ERIKSSON, M. et al. Agonistic targeting of tlr1/tlr2 induces p38 mapk-dependent apoptosis and nfb-dependent differentiation of aml cells. **Blood Adv**, v. 1, n. 23, p. 2046–2057, 2017.

ERRICHIELLO, E. et al. Mitochondrial variants in mt-co2 and d-loop instability are involved in mutyh-associated polyposis. **J Mol Med (Berl)**, v. 93, n. 11, p. 1271–1281, 2015.

ETHEMBABAOGLU, A.; WHITESON, S. et al. Automatic feature selection using fs-neat. **IAS technical report IAS-UVA-08-02**, Universiteit van Amsterdam, Informatics Institute, 2008.

FILIPPINI, A.; SICA, G.; D'ALESSIO, A. The caveolar membrane system in endothelium: From cell signaling to vascular pathology. **J Cell Biochem**, p. doi: 10.1002/jcb.26793, 2018.

FONTI, V.; BELITSER, E. Feature selection using lasso. **VU Amsterdam Research Paper in Business Analytics**, 2017.

FRUCHTER, B. Introduction to factor analysis. Van Nostrand, 1954.

GALAMB, O. et al. Aberrant dna methylation of wnt pathway genes in the development and progression of cimp-negative colorectal cancer. **Epigenetics**, v. 11, n. 8, p. 588–602, 2016.

GARRO, B. A.; RODRÍGUEZ, K.; VÁZQUEZ, R. A. Classification of dna microarrays using artificial neural networks and abc algorithm. **Applied Soft Computing**, Elsevier, v. 38, p. 548–560, 2016.

GARRO, B. A.; RODRÍGUEZ, K.; VAZQUEZ, R. A. Designing artificial neural networks using differential evolution for classifying dna microarrays. In: IEEE. **Evolutionary Computation (CEC), 2017 IEEE Congress on**. [S.l.], 2017. p. 2767–2774.

GASPARETTO, M.; SMITH, C. Aldhs in normal and malignant hematopoietic cells: Potential new avenues for treatment of aml and other blood cancers. **Chem Biol Interact**, v. 276, p. 46–51, 2017.

GASPARETTO, M.; SMITH, C. A. Aldhs in normal and malignant hematopoietic cells: Potential new avenues for treatment of aml and other blood cancers. **Chemico-biological interactions**, Elsevier, v. 276, p. 46–51, 2017.

GAUTIER, L. et al. affy - analysis of affymetrix genechip data at the probe level. **Bioinformatics**, v. 20, n. 3, p. 307–315, 2004.

GHEYAS, I. A.; SMITH, L. S. Feature subset selection in large dimensionality domains. **Pattern recognition**, Elsevier, v. 43, n. 1, p. 5–13, 2010.

GILBERT, S. **Differential Gene Expression**. Sunderland (MA): Sinauer Associates, 2000. 695 pages p.

GIRERD, S. et al. Superoxide dismutase 2 (sod2) contributes to genetic stability of native and t315i-mutated bcr-abl expressing leukemic cells. **Biochem Biophys Res Commun**, v. 498, n. 4, p. 715–722, 2018.

GKRETSI, V.; STYLIANOPOULOS, T. Cell adhesion and matrix stiffness: Coordinating cancer cell invasion and metastasis. **Front Oncol**, p. doi: 10.3389/fonc.2018.00145, 2018.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning**. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675.

GOLEMBESKY, A. et al. Peroxisome proliferator-activated receptor-alpha (ppara) genetic polymorphisms and breast cancer risk: a long island ancillary study. **Carcinogenesis**, v. 29, n. 10, p. 1944–1949, 2008.

GOLUB, T. R. et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. **science**, American Association for the Advancement of Science, v. 286, n. 5439, p. 531–537, 1999.

GOMES, F. P. **Curso de estatística experimental**. [S.l.]: Nobel, 2000.

GOMEZ, F.; MIIKKULAINEN, R. Solving non-markovian control tasks with neuroevolution. **Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence**, p. 1356–1361, 1999.

GOMEZ, F.; MIIKKULAINEN, R. Learning robust nonlinear control with neuroevolution. **Technical Report AI02-292, Department of Computer Sciences, The University of Texas at Austin, Austin, Texas.**, 2002.

GOODFELLOW, I. et al. **Deep learning**. [S.l.]: MIT press Cambridge, 2016.

GORRETA, F.; CARBONE, W.; BARZAGHI, D. Genomic profiling: cdna arrays and oligoarrays. **Methods Mol Biol**, v. 823, p. 89–105, 2012.

GROOT, J. de et al. Validation of dna promoter hypermethylation biomarkers in breast cancer–a short report. **Cell Oncol (Dordr)**, v. 37, n. 4, p. 297–303, 2014.

GRUAU F., W. D.; PYEATT, L. A comparison between cellular encoding and direct encoding for genetic neural networks. In: **Genetic Programming, Proceedings of the First Annual Conference**. Cambridge, Massachusetts: [s.n.], 1996. p. 81–89.

GU, Q.; LI, Z.; HAN, J. Generalized fisher score for feature selection. **arXiv preprint arXiv:1202.3725**, 2012.

GUIèZE, R. et al. Telomere status in chronic lymphocytic leukemia with tp53 disruption. **Oncotarget**, v. 7, n. 35, p. 56976–56985, 2017.

GUPTA, A. et al. On the use of local search in the evolution of neural networks for the diagnosis of breast cancer. **Technologies**, Multidisciplinary Digital Publishing Institute, v. 3, n. 3, p. 162–181, 2015.

HAN, H. et al. Altered methylation and expression of er-associated degradation factors in long-term alcohol and constitutive er stress-induced murine hepatic tumors. **Front Genet**, v. 4, p. 224, 2013.

HARAMI-PAPP, H. et al. Tp53 mutation hits energy metabolism and increases glycolysis in breast cancer. **Oncotarget**, v. 7, n. 41, p. 67183–67195, 2016.

HARRINGTON, P. **Machine learning in action**. Shelter Island, NY 11964: Manning Greenwich, CT, 2012.

HAYKIN, S. **Neural Networks: A comprehensive foundation**. 2. ed. New York, USA: Prentice Hall Inc., 1998.

HE, W. et al. Cthrc1 induces non-small cell lung cancer (nsclc) invasion through upregulating mmp-7/mmp-9. **BMC Cancer**, v. 18, n. 1, p. 400, 2018.

HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. **Science**, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006.

HORNIK, K. Approximation capabilities of multilayer feedforward networks. **Neural Networks**, v. 4, n. 2, p. 251—257, 1991.

HOU, T. et al. Expression of col6a1 predicts prognosis in cervical cancer patients. **Am J Transl Res**, v. 8, n. 6, p. 2838–2844, 2016.

HSU, C. et al. Identification and characterization of potential biomarkers by quantitative tissue proteomics of primary lung adenocarcinoma. **Mol Cell Proteomics**, v. 15, n. 7, p. 2396–2410, 2016.

HU, G. et al. Molecular mechanisms of long noncoding rnas and their role in disease pathogenesis. **Oncotarget**, v. 9, n. 26, p. 18648–18663, 2018.

HUA, Y. et al. Abnormal expression of mrna, microrna alteration and aberrant dna methylation patterns in rectal adenocarcinoma. **PLoS One**, v. 12, n. 3, p. e0174461, 2017.

HUAN, J. et al. Screening for key genes associated with invasive ductal carcinoma of the breast via microarray data analysis. **Genet Mol Res**, v. 13, n. 3, p. 7919–7925, 2014.

HUANG, D.; SHERMAN, B.; LEMPICKI, R. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. **Nucleic Acids Res**, v. 37, n. 1, p. 1–13, 2009.

HUANG, D.; SHERMAN, B.; LEMPICKI, R. Systematic and integrative analysis of large gene lists using david bioinformatics resources. **Nature Protoc**, v. 4, n. 1, p. 44–57, 2009.

HUANG, R.; LIAO, X.; LI, Q. Identification of key pathways and genes in tp53 mutation acute myeloid leukemia: evidence from bioinformatics analysis. **Onco Targets Ther**, v. 11, p. 163–173, 2017.

HUGHES, P. et al. 1alpha,25-dihydroxyvitamin d3 stimulates steroid sulphatase activity in hl60 and nb4 acute myeloid leukaemia cell lines by different receptor-mediated mechanisms. **J Cell Biochem**, v. 94, n. 6, p. 1175–1189, 2005.

ISEKI, H. et al. Alex1 suppresses colony formation ability of human colorectal carcinoma cell lines. **Cancer Sci**, v. 103, n. 7, p. 1267–1271, 2012.

ISHIMOTO, T. et al. Activation of transforming growth factor beta 1 signaling in gastric cancer-associated fibroblasts increases their motility, via expression of rhomboid 5 homolog 2, and ability to induce invasiveness of gastric cancer cells. **Gastroenterology**, v. 153, n. 1, p. 191–204, 2017.

JIA, H. et al. The lim protein ajuba promotes colorectal cancer cell survival through suppression of jak1/stat1/ifit2 network. **Oncogene**, v. 36, n. 19, p. 2655–2666, 2017.

JIANG, W. et al. Eplin-alpha expression in human breast cancer, the impact on cellular migration and clinical outcome. **Mol Cancer**, v. 7, p. 71, 2008.

JIN, X. et al. Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles. In: SPRINGER. **International Workshop on Data Mining for Biomedical Applications**. [S.l.], 2006. p. 106–115.

JO, Y. et al. Somatic mutations and intratumoral heterogeneity of myh11 gene in gastric and colorectal cancers. **Appl Immunohistochem Mol Morphol**, p. doi: 10.1097/PAI.0000000000000484, 2018.

JOHNSEN, S. et al. Regulation of estrogen-dependent transcription by the lim cofactors clim and rlim in breast cancer. **Cancer Res**, v. 69, n. 1, p. 128–136, 2009.

JOHNSTON, H. et al. Proteomics profiling of cll versus healthy b-cells identifies putative therapeutic targets and a subtype-independent signature of spliceosome dysregulation. **Mol Cell Proteomics**, v. 17, n. 4, p. 776–791, 2018.

JOLLIFFE, I. T.; CADIMA, J. Principal component analysis: a review and recent developments. **Philos Trans A Math Phys Eng Sci**, v. 374, n. 2065, p. 20150202, 2016.

KABBAGE, M. et al. Protein alterations in infiltrating ductal carcinomas of the breast as detected by nonequilibrium ph gradient electrophoresis and mass spectrometry. **J Biomed Biotechnol**, v. 2008, p. 564127, 2008.

KALMAR, A. et al. Gene expression analysis of normal and colorectal cancer tissue samples from fresh frozen and matched formalin-fixed, paraffin-embedded (ffpe) specimens after manual and automated rna isolation. **Methods**, v. 59, n. 1, p. S16–S19, 2013.

KAUCKá, M. et al. The planar cell polarity pathway drives pathogenesis of chronic lymphocytic leukemia by the regulation of b-lymphocyte migration. **Cancer Res**, v. 73, n. 5, p. 1491–1501, 2013.

KAUFFMANN, A.; GENTLEMAN, R.; HUBER, W. arrayqualitymetrics–a bioconductor package for quality assessment of microarray data. **Bioinformatics**, v. 25, n. 3, p. 415–416, 2009.

KENNEDY, B.; HARRIS, R. Cyclooxygenase and lipoxygenase gene expression in the inflammogenesis of breast cancer. **Inflammopharmacology.**, p. doi: 10.1007/s10787–018–0489–6, 2018.

KETTUNEN, E. et al. Asbestos-associated genome-wide dna methylation changes in lung cancer. **Int J Cancerl**, v. 141, n. 10, p. 2014–2029, 2017.

KEUP, C. et al. Rna profiles of circulating tumor cells and extracellular vesicles for therapy stratification of metastatic breast cancer patients. **Clin Chem**, pii: clinchem.2017, p. 283531, 2018.

KHAN, J. et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. **Nature medicine**, Nature Publishing Group, v. 7, n. 6, p. 673, 2001.

KHARAZIHA, P. et al. Molecular profiling of prostate cancer derived exosomes may reveal a predictive signature for response to docetaxel. **Oncotarget**, v. 6, n. 25, p. 21740–2154, 2015.

KHATUN, A. et al. Transcriptional repression and protein degradation of the ca2+-activated k+ channel kca1.1 by androgen receptor inhibition in human breast cancer cells. **Front Physiol**, v. 9, p. 312, 2018.

KIEFER, J.; WOLFOWITZ, J. Stochastic estimation of the maximum of a regression function. **Ann Math Stat**, p. 462–466, 1952.

KIM, S. et al. The epigenetic silencing of lims2 in gastric cancer and its inhibitory effect on cell migration. **Biochem Biophys Res Commun**, v. 349, n. 3, p. 1032–1040, 2006.

KIRA, K.; RENDELL, L. A. The feature selection problem: Traditional methods and a new algorithm. In: **Aaai**. [S.l.: s.n.], 1992. v. 2, p. 129–134.

KLEMA, V.; LAUB, A. The singular value decomposition: Its computation and some applications. **IEEE Trans Autom Control**, v. 25, n. 2, p. 164–176, 1980.

KLIMOSCH, S. et al. Functional tlr5 genetic variants affect human colorectal cancer survival. **Cancer Res**, v. 73, n. 24, p. 7232–7242, 2013.

KORFF, S. et al. Frameshift mutations in coding repeats of protein tyrosine phosphatase genes in colorectal tumors with microsatellite instability. **BMC Cancer**, v. 8, p. 329, 2008.

KOSTIANETS, O. et al. Immunohistochemical analysis of medullary breast carcinoma autoantigens in different histological types of breast carcinomas. **Diagn Pathol**, v. 7, p. 161, 2012.

KRIZEK, P. **Feature selection: stability, algorithms, and evaluation**. Thesis (PhD) — PhD thesis, Czech Technical University in Prague, 2008. 6, 14, 36, 67, 93, 2008.

KRUMBHOLZ, M. et al. Response monitoring of infant acute myeloid leukemia treatment by quantification of the tumor specific mll-fnbp1 fusion gene. **Leuk Lymphoma**, v. 53, n. 3, p. 793–796, 2015.

KRUSE, R. et al. **Computational intelligence: a methodological introduction**. Springer London Heidelberg New York Dordrecht: Springer, 2016. 490 p.

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. **Journal of the American statistical Association**, Taylor & Francis, v. 47, n. 260, p. 583–621, 1952.

KUNC, M.; BIERNAT, W.; SENKUS-KONEFKA, E. Estrogen receptor-negative progesterone receptor-positive breast cancer–"nobody's land "or just an artifact? **Cancer treatment reviews**, Elsevier, v. 67, p. 78–87, 2018.

KURER, M. Protein and mrna expression of tissue factor pathway inhibitor-1 (tfpi-1) in breast, pancreatic and colorectal cancer cells. **Mol Biol Rep**, v. 34, n. 4, p. 221–224, 2007.

KUROZUMI, S. et al. Power of pgr expression as a prognostic factor for er-positive/her2-negative breast cancer patients at intermediate risk classified by the ki67 labeling index. **BMC Cancer**, v. 17, n. 1, p. 354, 2017.

KUTHAN, T.; LANSKY, J. Genetic algorithms in syllable-based text compression. **Dateso**, p. 21–34, 2007.

LAL, S. et al. Pharmacogenetics of abcb5, abcc5 and rlip76 and doxorubicin pharmacokinetics in asian breast cancer patients. **Pharmacogenomics J**, v. 17, n. 4, p. 337–343, 2017.

LAN, L.; VUCETIC, S. Improving accuracy of microarray classification by a simple multi-task feature selection filter. **International journal of data mining and bioinformatics**, Inderscience Publishers, v. 5, n. 2, p. 189–208, 2011.

LARA-PADILLA, E. et al. Neural transdifferentiation: Maptau gene expression in breast cancer cells. **Asian Pac J Cancer Prev**, v. 17, n. 4, p. 1967–1971, 2016.

LAURIOLA, M. et al. Identification by a digital gene expression displayer (dged) and test by rt-pcr analysis of new mrna candidate markers for colorectal cancer in peripheral blood. **Int J Oncol**, v. 37, n. 2, p. 519–525, 2010.

LAVROV, A. V. et al. Copy number variation analysis in cytochromes and glutathione s-transferases may predict efficacy of tyrosine kinase inhibitors in chronic myeloid leukemia. **PloS one**, Public Library of Science, v. 12, n. 9, p. e0182901, 2017.

LAWRY, J. et al. The value of assessing cell proliferation in breast cancer. **J Microsc**, v. 159, n. p.3, p. 265–275, 1990.

LAZAR, C. et al. A survey on filter techniques for feature selection in gene expression microarray analysis. **IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)**, IEEE Computer Society Press, v. 9, n. 4, p. 1106–1119, 2012.

LEE, J. W. et al. An extensive comparison of recent classification tools applied to microarray data. **Computational Statistics & Data Analysis**, Elsevier, v. 48, n. 4, p. 869–885, 2005.

LEE, S. et al. Identification of genes underlying different methylation profiles in refractory anemia with excess blast and refractory cytopenia with multilineage dysplasia in myelodysplastic syndrome. **Korean J Hematol**, v. 47, n. 3, p. 186–193, 2012.

LEUNG, Y. F.; CAVALIERI, D. Fundamentals of cdna microarray data analysis. **TRENDS in Genetics**, Elsevier, v. 19, n. 11, p. 649–659, 2003.

LI, F. et al. Fcrl2 expression predicts ighv mutation status and clinical progression in chronic lymphocytic leukemia. **Blood**, v. 112, n. 1, p. 179–187, 2008.

LI, H. et al. The integrated pathway of tgf/snail with tnf/nfb may facilitate the tumor-stroma interaction in the emt process and colorectal cancer prognosis. **Sci Rep**, v. 7, n. 1, p. 4915, 2017.

LI, J. et al. Bioinformatics analysis of gene expression profiles in childhood b-precursor acute lymphoblastic leukemia. **Hematology**, Taylor & Francis, v. 20, n. 7, p. 377–383, 2015.

LI, W. et al. Detection of osr2, vav3, and ppfia3 methylation in the serum of patients with gastric cancer. **Dis Markers**, v. 2016, p. 5780538, 2016.

LINNAINMAA, S. Taylor expansion of the accumulated rounding error. **BIT Numerical Mathematics**, v. 16, n. 2, p. 146–160, 1976.

LIU, D. et al. Rec8 is a novel tumor suppressor gene epigenetically robustly targeted by the pi3k pathway in thyroid cancer. **Oncotarget**, v. 6, n. 36, p. 39211–39224, 2015.

LIU, J. et al. Microrna-155 acts as a tumor suppressor in colorectal cancer by targeting cthrc1 in vitro. **Oncol Lett**, v. 15, n. 4, p. 5561–5568, 2018.

LIU, P. et al. Genome-wide association and fine mapping of genetic loci predisposing to colon carcinogenesis in mice. **Molecular cancer research**, AACR, p. molcanres–0540, 2011.

LIU, T. et al. Comparative transcriptomes and evo-devo studies depending on next generation sequencing. **Comput Math Methods Med**, v. 2015, p. 896176, 2015.

LIU, W. et al. Role of col6a3 in colorectal cancer. **Oncol Rep**, v. 39, n. 6, p. 2527–2536, 2018.

LIU, X.; WANG, J.; SUN, G. Identification of key genes and pathways in renal cell carcinoma through expression profiling data. **Kidney and Blood Pressure Research**, Karger Publishers, v. 40, n. 3, p. 288–297, 2015.

LONGVILLE, B. A. et al. Aberrant expression of aldehyde dehydrogenase 1a (aldh 1a) subfamily genes in acute lymphoblastic leukaemia is a common feature of t-lineage tumours. **British journal of haematology**, Wiley Online Library, v. 168, n. 2, p. 246–257, 2015.

LUKE, S. **Essentials of metaheuristics**. 1. ed. [S.l.]: Lulu, 2009. 227 p.

LUQUE-BAENA, R. et al. 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**. [S.l.: s.n.], 2013. p. 55–63.

LYKKESFELDT, A. et al. Aurora kinase a as a possible marker for endocrine resistance in early estrogen receptor positive breast cancer. **Acta Oncol**, v. 57, n. 1, p. 67–73, 2016.

LYONS, R. et al. The rac specific guanine nucleotide exchange factor asef functions downstream from tel-aml1 to promote leukaemic transformation. **Leuk Res**, v. 34, n. 1, p. 109–115, 2010.

LYU, P. et al. Identification of twist-interacting genes in prostate cancer. **Sci China Life Sci**, v. 60, n. 4, p. 386–396, 2017.

MAATEN, L. van der; HINTON, G. Visualizing data using t-sne. **Journal of Machine Learning Research**, v. 9, n. Nov, p. 2579–2605, 2008.

MAIA, J. et al. Exosome-based cell-cell communication in the tumor microenvironment. **Front Cell Dev Biol**, v. 6, p. 18, 2018.

MALOUF, G. et al. Transcriptional profiling of pure fibrolamellar hepatocellular carcinoma reveals an endocrine signature. **Hepatology**, v. 59, n. 6, p. 2228–2237, 2014.

MAMOSHINA, P. et al. Applications of deep learning in biomedicine. **Molecular pharmaceutics**, ACS Publications, v. 13, n. 5, p. 1445–1454, 2016.

MARCUS, G. Deep learning: A critical appraisal. **arXiv preprint arXiv:1801.00631**, 2018.

MARINO, N. et al. Identification and validation of genes with expression patterns inverse to multiple metastasis suppressor genes in breast cancer cell lines. **Clin Exp Metastasis**, v. 31, n. 7, p. 771–786, 2014.

MARTÍNEZ-IGLESIAS, O. et al. The nuclear corepressor 1 and the thyroid hormone receptor β suppress breast tumor lymphangiogenesis. **Oncotarget**, Impact Journals, LLC, v. 7, n. 48, p. 78971, 2016.

MATEO, M. M.; MARTÍN, G. Influence of metallic carcinogenesis in lung and colorectal neoplasia. metals in neoplastic processes. **Clin Physiol Biochem**, v. 6, n. 6, p. 321–326, 1988.

MIAO, J.; NIU, L. A survey on feature selection. **Procedia Computer Science**, Elsevier, v. 91, p. 919–926, 2016.

MILLER, B. L.; GOLDBERG, D. E. et al. Genetic algorithms, tournament selection, and the effects of noise. **Complex systems**, [Champaign, IL, USA: Complex Systems Publications, Inc., c1987-, v. 9, n. 3, p. 193–212, 1995.

MILOSEVIC, N. et al. Synthetic lethality screen identifies rps6ka2 as modifier of epidermal growth factor receptor activity in pancreatic cancer. **Neoplasia**, v. 15, n. 12, p. 1354–1362, 2013.

MIN, S.; LEE, B.; YOON, S. Deep learning in bioinformatics. **Briefings in bioinformatics**, Oxford University Press, v. 18, n. 5, p. 851–869, 2017.

MORSE, G.; STANLEY, K. O. Simple evolutionary optimization can rival stochastic gradient descent in neural networks. In: ACM. **Proceedings of the Genetic and Evolutionary Computation Conference 2016**. [S.l.], 2016. p. 477–484.

MURPHY, D. Gene expression studies using microarrays: principles, problems, and prospects. **Advances in physiology education**, American Physiological Society, v. 26, n. 4, p. 256–270, 2002.

MáRQUEZ, J. et al. Identification of hepatic microvascular adhesion-related genes of human colon cancer cells using random homozygous gene perturbation. **Int J Cancer**, v. 133, n. 9, p. 2113–2122, 2013.

NATTESTAD, M. et al. Complex rearrangements and oncogene amplifications revealed by long-read dna and rna sequencing of a breast cancer cell line. **Genome research**, Cold Spring Harbor Lab, p. gr–231100, 2018.

NEWMAN, B. et al. Possible genetic predisposition to lymphedema after breast cancer. **Lymphatic research and biology**, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 10, n. 1, p. 2–13, 2012.

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

NG, H. Y. et al. Epigenetic inactivation of dapk1, p14arf, mir-34a and-34b/c in acute promyelocytic leukaemia. **Journal of clinical pathology**, BMJ Publishing Group, v. 67, n. 7, p. 626–631, 2014.

NI, Y. et al. Germline compound heterozygous poly-glutamine deletion in usf3 may be involved in predisposition to heritable and sporadic epithelial thyroid carcinoma. **Hum Mol Genet**, v. 26, n. 2, p. 243–257, 2017.

NIAVARANI, A. et al. Pancancer analysis identifies prognostic high-apobec1 expression level implicated in cancer in-frame insertions and deletions. **Carcinogenesis**, v. 39, n. 3, p. 327–335, 2018.

NIEBOROWSKA-SKORSKA, M. et al. Rac2-mrc-ciii-generated ros cause genomic instability in chronic myeloid leukemia stem cells and primitive progenitors. **Blood**, v. 119, n. 18, p. 4253–4263, 2012.

OH, B. et al. Exome and transcriptome sequencing identifies loss of pdlim2 in metastatic colorectal cancers. **Cancer Manag Res**, v. 9, p. 581–589, 2017.

OLIEMULLER, E. et al. Sox11 promotes invasive growth and ductal carcinoma in situ progression. **J Pathol**, v. 243, n. 2, p. 193–207, 2017.

ORTEGA, P. et al. Mmp-7 and sgce as distinctive molecular factors in sporadic colorectal cancers from the mutator phenotype pathway. **Int J Oncol**, v. 36, n. 5, p. 1209–1215, 2010.

PAPAVASILEIOU, E.; JANSEN, B. A comparison between fs-neat and fd-neat and an investigation of different initial topologies for a classification task with irrelevant features. In: IEEE. **Computational Intelligence (SSCI), 2016 IEEE Symposium Series on**. [S.l.], 2016. p. 1–8.

PAPAVASILEIOU, E.; JANSEN, B. The importance of the activation function in neuroevolution with fs-neat and fd-neat. In: IEEE. **Computational Intelligence (SSCI), 2017 IEEE Symposium Series on**. [S.l.], 2017. p. 1–7.

PAPAVASILEIOU, E.; JANSEN, B. An investigation of topological choices in fs-neat and fd-neat on xor-based problems of increased complexity. In: ACM. **Proceedings of the Genetic and Evolutionary Computation Conference Companion**. [S.l.], 2017. p. 1431–1434.

PARK, C. et al. Pinch-2 presents functional copy number variation and suppresses migration of colon cancer cells by paracrine activity. **Int J Cancer**, v. 136, n. 10, p. 2273–2283, 2015.

PARK, Y.; KELLIS, M. Deep learning for regulatory genomics. **Nat Biotechnol**, v. 33, n. 8, p. 825–826, 2015.

PATEL, V. et al. Network signatures of survival in glioblastoma multiforme. **PLoS Comput Biol**, v. 9, n. 9, p. e1003237, 2013.

PATHAK, B. et al. Androgen receptor mediated epigenetic regulation of crisp3 promoter in prostate cancer cells. **J Steroid Biochem Mol Biol**, pii: S0960-0760, n. 18, p. 30108–0, 2018.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PEREIRA, C. et al. Influence of genetic polymorphisms in prostaglandin e2 pathway (cox-2/hpgd/slco2a1/abcc4) on the risk for colorectal adenoma development and recurrence after polypectomy. **Clin Transl Gastroenterol**, v. 7, n. 9, p. e191, 2016.

PETERSON, L. E. et al. Artificial neural network analysis of dna microarray-based prostate cancer recurrence. In: IEEE. **Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB'05. Proceedings of the 2005 IEEE Symposium on**. [S.l.], 2005. p. 1–8.

PIROOZNIA, M. et al. A comparative study of different machine learning methods on microarray gene expression data. **BMC genomics**, BioMed Central, v. 9, n. 1, p. S13, 2008.

PIROUZPANAH, S. et al. Hypermethylation pattern of esr and pgr genes and lacking estrogen and progesterone receptors in human breast cancer tumors: Er/pr subtypes. **Cancer Biomark**, v. 21, n. 3, p. 621–638, 2018.

PIZZINI, S. et al. Impact of micrornas on regulatory networks and pathways in human colorectal carcinogenesis and development of metastasis. **BMC Genomics**, v. 14, p. 589, 2013.

POLISENO, L.; MARRANCI, A.; PANDOLFI, P. Pseudogenes in human cancer. **Front Med (Lausanne)**, v. 2, p. 68, 2015.

POWERS, R.; GOLDSZMIDT, M.; COHEN, I. Short term performance forecasting in enterprise systems. In: ACM. **Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining**. [S.l.], 2005. p. 801–807.

QUACKENBUSH, J. Computational genetics: computational analysis of microarray data. **Nature reviews genetics**, Nature Publishing Group, v. 2, n. 6, p. 418, 2001.

RAIMONDI, M. et al. Calpain restrains the stem cells compartment in breast cancer. **Cell Cycle**, v. 15, n. 1, p. 106–116, 2016.

RAMALINGAM, S. et al. Treatment-related neuroendocrine prostate cancer resulting in cushing's syndrome. **Int J Urol**, v. 23, n. 12, p. 1038–1041, 2016.

RENIERI, A. et al. Oligogenic germline mutations identified in early non-smokers lung adenocarcinoma patients. **Lung Cancer**, v. 85, n. 2, p. 168–174, 2014.

RESSOM, H. W. et al. **Microarray Data Analysis Using Machine Learning Methods**. [S.l.: s.n.], 2009. 32 p.

REZVANI, K. Ubxd proteins: A family of proteins with diverse functions in cancer. **Int J Mol Sci**, v. 17, n. 10, p. pii: E1724, 2016.

RODIA, M. et al. Lgals4, ceacam6, tspan8, and col1a2: Blood markers for colorectal cancer-validation in a cohort of subjects with positive fecal immunochemical test result. **Clin Colorectal Cancer**, pii: S1533-0028, n. 17, p. 30380–30388, 2017.

ROSE, M. et al. Itih5 mediates epigenetic reprogramming of breast cancer cells. **Mol Cancer**, v. 16, n. 1, p. 44, 2017.

ROSENBERG, E. et al. Expression of cancer-associated genes in prostate tumors. **Exp Onco**, v. 39, n. 2, p. 131–137, 2017.

SAITOH, M. Involvement of partial emt in cancer progression. **J Biochem**, p. doi: 10.1093/jb/mvy047, 2018.

SANTPERE, G. et al. Transcriptome evolution from breast epithelial cells to basal-like tumors. **Oncotarget**, v. 9, n. 1, p. 453–463, 2017.

SARAN, S. et al. Depletion of three combined thoc5 mrna export protein target genes synergistically induces human hepatocellular carcinoma cell death. **Oncogene**, v. 35, n. 29, p. 3872–3879, 2016.

SCHULTE, I. et al. Structural analysis of the genome of breast cancer cell line zr-75-30 identifies twelve expressed fusion genes. **BMC Genomics**, v. 13, p. 719, 2012.

SHAIKHIBRAHIM, Z. et al. Analysis of laser-microdissected prostate cancer tissues reveals potential tumor markers. **Int J Mol Med**, v. 28, n. 4, p. 605–11, 2011.

SHAN, M. et al. Molecular analyses of prostate tumors for diagnosis of malignancy on fine-needle aspiration biopsies. **Oncotarget**, v. 8, n. 62, p. 104761–104771, 2017.

SHANGKUAN, W. et al. Risk analysis of colorectal cancer incidence by gene expression analysis. **PeerJ**, v. 5, p. e3003, 2017.

SHANGKUAN, W.-C. et al. Risk analysis of colorectal cancer incidence by gene expression analysis. **PeerJ**, PeerJ Inc., v. 5, p. e3003, 2017.

SHARMA, P.; WADHWA, A. Analysis of selection schemes for solving an optimization problem in genetic algorithm. **International Journal of Computer Applications**, Foundation of Computer Science, v. 93, n. 11, 2014.

SHEFFER, M. et al. Association of survival and disease progression with chromosomal instability: a genomic exploration of colorectal cancer. **Proc Natl Acad Sci U S A**, v. 106, n. 17, p. 7131–7136, 2009.

SHI, Z. et al. Identification and verification of candidate genes regulating neural stem cells behavior under hypoxia. **Cell Physiol Biochem**, v. 47, n. 1, p. 212–222, 2018.

SHIPP, M. A. et al. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. **Nature medicine**, Nature Publishing Group, v. 8, n. 1, p. 68, 2002.

SINGH, V. et al. Esophageal cancer epigenomics and integrome analysis of genome-wide methylation and expression in high risk northeast indian population. **OMICS**, v. 19, n. 11, p. 688–699, 2015.

SIPPER, M.; OLSON, R. S.; MOORE, J. H. **Evolutionary computation: the next major transition of artificial intelligence?** [S.l.]: BioMed Central, 2017.

SLATTERY, M. et al. Genetic variation in rps6ka1, rps6ka2, rps6kb1, rps6kb2, and pdk1 and risk of colon or rectal cancer. **Mutat Res**, v. 706, n. 1-2, p. 13–20, 2011.

SOARES, G. P. et al. Value of systemic staging in asymptomatic early breast cancer. **Revista Brasileira de Ginecologia e Obstetrícia/RBGO Gynecology and Obstetrics**, Thieme Revinter Publicações Ltda, 2018.

SOHANGIR, S.; RAHIMI, S.; GUPTA, B. Optimized feature selection using neuroevolution of augmenting topologies (neat). In: IEEE. **IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint**. [S.l.], 2013. p. 80–85.

SOHANGIR, S.; RAHIMI, S.; GUPTA, B. Neuroevolutionary feature selection using neat. **Journal of Software Engineering and Applications**, Scientific Research Publishing, v. 7, n. 07, p. 562, 2014.

SRIVASTAVA, R. K.; GREFF, K.; SCHMIDHUBER, J. Highway networks. **arXiv preprint arXiv:1505.00387**, 2015.

STANLEY, K. O.; MIIKKULAINEN, R. Evolving neural networks through augmenting topologies. **Evolutionary Computation**, MIT Press, v. 10, n. 2, p. 99–127, 2002.

STATNIKOV, A.; WANG, L.; ALIFERIS, C. F. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. **BMC bioinformatics**, BioMed Central, v. 9, n. 1, p. 319, 2008.

STICKLES, X. et al. Bad-mediated apoptotic pathway is associated with human cancer development. **Int J Mol Med**, v. 35, n. 4, p. 1081–1087, 2015.

STOSKUS, M. et al. Identification of characteristic igf2bp expression patterns in distinct b-all entities. **Blood Cells Mol Dis**, v. 46, n. 4, p. 321–326, 2011.

STUELTEN, C.; PARENT, C.; MONTELL, D. Cell motility in cancer invasion and metastasis: insights from simple model organisms. **Nat Rev Cancer**, v. 18, n. 5, p. 296–312, 2018.

SUBRAMANIAN, A. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 102, n. 43, p. 15545–15550, 2005.

SUCH, F. P. et al. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. **arXiv preprint arXiv:1712.06567**, 2017.

SUHOVSKIH, A. et al. Transcriptional activity of heparan sulfate biosynthetic machinery is specifically impaired in benign prostate hyperplasia and prostate cancer. **Front Oncol**, v. 4, p. 79, 2014.

SUKHAI, M. et al. Lysosomal disruption preferentially targets acute myeloid leukemia cells and progenitors. **J Clin Invest**, v. 123, n. 1, p. 315–328, 2013.

TAKEUCHI, S. et al. Mutations in the retinoblastoma-related gene rb2/p130 in adult t-cell leukaemia/lymphoma. **Leuk Lymphoma**, v. 44, n. 4, p. 699–701, 2003.

TAN, M. et al. Automated feature selection in neuroevolution. **Evolutionary Intelligence**, Springer, v. 1, n. 4, p. 271–292, 2009.

TANG, J. et al. Lncrna pvt1 regulates triple-negative breast cancer through klf5/beta-catenin signaling. **Oncogene**, p. doi:10.1038/s41388–018–0310–4, 2018.

TANG, M. et al. Tumoral nkg2d alters cell cycle of acute myeloid leukemic cells and reduces nk cell-mediated immune surveillance. **Immunologic Research**, v. 64, n. 3, p. 754–764, 2016.

TANG, Y. et al. Benzyl isothiocyanate attenuates the hydrogen peroxide-induced interleukin-13 expression through glutathione s-transferase p induction in t lymphocytic leukemia cells. **Journal of biochemical and molecular toxicology**, Wiley Online Library, p. e22054, 2018.

TANIC, N. et al. Identification of differentially expressed mrna transcripts in drug-resistant versus parental human melanoma cell lines. **Anticancer Res**, v. 26, n. 3A, p. 2137–2142, 2006.

TAO, Y.-F. et al. Early b-cell factor 3 (ebf3) is a novel tumor suppressor gene with promoter hypermethylation in pediatric acute myeloid leukemia. **Journal of Experimental & Clinical Cancer Research**, BioMed Central, v. 34, n. 1, p. 4, 2015.

TAO, Z. et al. Microarray bioinformatics in cancer- a review. **J BUON**, v. 22, n. 4, p. 838–843, 2017.

TESSIER-CLOUTIER, B. et al. Molecular subtyping of mammary-like adenocarcinoma of the vulva shows molecular similarity to breast carcinomas. **Histopathology**, v. 71, n. 3, p. 446–452, 2017.

THAKKAR, A. et al. High expression of three-gene signature improves prediction of relapse-free survival in estrogen receptor-positive and node-positive breast tumors. **Biomarker insights**, SAGE Publications Sage UK: London, England, v. 10, p. BMI–S30559, 2015.

THAKKAR, A. D. et al. Identification of gene expression signature in estrogen receptor positive breast carcinoma. **Biomarkers in cancer**, SAGE Publications Sage UK: London, England, v. 2, p. BIC–S3793, 2010.

THEAN, L. et al. Genome-wide association study identified copy number variants associated with sporadic colorectal cancer risk. **J Med Genet**, v. 55, n. 3, p. 181–188, 2018.

THUTKAWKORAPIN, J. et al. Exome sequencing in one family with gastric-and rectal cancer. **BMC genetics**, BioMed Central, v. 17, n. 1, p. 41, 2016.

TOMOSHIGE, K. et al. Germline mutations causing familial lung cancer. **Journal of human genetics**, Nature Publishing Group, v. 60, n. 10, p. 597, 2015.

TONG, X. et al. Sox10, a novel hmg-box-containing tumor suppressor, inhibits growth and metastasis of digestive cancers by suppressing the wnt/-catenin pathway. **Oncotarget**, v. 5, n. 21, p. 10571–10583, 2014.

TORRES, S. et al. Proteome profiling of cancer-associated fibroblasts identifies novel proinflammatory signatures and prognostic markers for colorectal cancer. **Clin Cancer Res**, v. 19, n. 21, p. 6006–6019, 2013.

TRUAX, A.; THAKKAR, M.; GREER, S. Dysregulated recruitment of the histone methyltransferase ezh2 to the class ii transactivator (ciita) promoter iv in breast cancer cells. **PLoS One**, v. 7, n. 4, p. e36013, 2012.

TURNER, J. et al. Kinase gene fusions in defined subsets of melanoma. **Pigment Cell Melanoma Res**, v. 30, n. 1, p. 53–62, 2017.

TUTAR, Y. et al. Regulation of oncogenic genes by micrornas and pseudogenes in human lung cancer. **Biomed Pharmacother**, v. 83, p. 1182–1190, 2016.

UEHARA, S. et al. Role of arhgap24 in adp ribosylation factor 6 (arf6)-dependent pseudopod formation in human breast carcinoma cells. **Anticancer Res**, v. 37, n. 9, p. 4837–4844, 2017.

VALLADARES, A. et al. Genetic expression profiles and chromosomal alterations in sporadic breast cancer in mexican women. **Cancer Genet Cytogenet.**, v. 170, n. 2, p. 147–151, 2006.

VARCHETTA, S. et al. Elements related to heterogeneity of antibody-dependent cell cytotoxicity in patients under trastuzumab therapy for primary operable breast cancer overexpressing her2. **Cancer Res**, v. 67, n. 24, p. 11991–11999, 2007.

VARSHAVSKY, R. et al. Novel unsupervised feature filtering of biological data. **Bioinformatics**, Oxford University Press, v. 22, n. 14, p. e507–e513, 2006.

VERLEYSEN, M.; FRANÇOIS, D. The curse of dimensionality in data mining and time series prediction. In: SPRINGER. **IWANN**. [S.l.], 2005. v. 5, p. 758–770.

WALSH, C. et al. Microarray meta-analysis and cross-platform normalization: Integrative genomics for robust biomarker discovery. **Microarrays (Basel)**, v. 4, n. 3, p. 389–406, 2015.

WAN, F. et al. Upregulation of col6a1 is predictive of poor prognosis in clear cell renal cell carcinoma patients. **Oncotarget**, v. 6, n. 29, p. 27378–27387, 2015.

WANG, H. et al. Forfeited hepatogenesis program and increased embryonic stem cell traits in young hepatocellular carcinoma (hcc) comparing to elderly hcc. **BMC Genomics**, v. 14, p. 736, 2013.

WANG, Q. et al. Regulation of meis1 by distal enhancer elements in acute leukemia. **Leukemia**, v. 28, n. 1, 2014.

WANG, Y. et al. Interaction analysis between germline susceptibility loci and somatic alterations in lung cancer. **Int J Cancer**, p. doi: 10.1002/ijc.31351, 2018.

WEDGE, D. et al. Sequencing of prostate cancers identifies new cancer genes, routes of progression and drug targets. **Nat Genet**, v. 50, n. 5, p. 682–692, 2018.

WHITESON, S. et al. Automatic feature selection in neuroevolution. In: ACM. **Proceedings of the 7th annual conference on Genetic and evolutionary computation**. [S.l.], 2005. p. 1225–1232.

WHITWORTH, G. B. An introduction to microarray data analysis and visualization. **Methods in enzymology**, Elsevier, v. 470, p. 19–50, 2010.

WILLIAMS, M.; ARIZA, M. Ebv positive diffuse large b cell lymphoma and chronic lymphocytic leukemia patients exhibit increased anti-dutpase antibodies. **Cancers (Basel)**, v. 10, n. 5, p. pii: E129, 2018.

WU, C. et al. Combined effects of peroxisome proliferator-activated receptor alpha and apolipoprotein e polymorphisms on risk of breast cancer in a taiwanese population. **J Investig Med**, v. 60, n. 8, p. 1209–1213, 2012.

WU, H. et al. Lifr promotes tumor angiogenesis by up-regulating il-8 levels in colorectal cancer. **Biochim Biophys Acta**, pii: S0925-4439, n. 18, p. 30174–30171, 2018.

WU, J. et al. Foxp2 promotes tumor proliferation and metastasis by targeting grp78 in triple-negative breast cancer. **Curr Cancer Drug Targets.**, v. 18, n. 4, p. 382–389, 2018.

WU, Z. et al. Mfap5 promotes tumor progression and bone metastasis by regulating erk/mmp signaling pathways in breast cancer. **Biochem Biophys Res Commun**, v. 498, n. 3, p. 495–501, 2018.

XU, K. et al. Suppression subtractive hybridization identified differentially expressed genes in colorectal cancer: microrna-451a as a novel colorectal cancer-related gene. **Tumour Biol**, v. 39, n. 5, p. 1010428317705504, 2017.

YANG, D. et al. Smad1 promotes colorectal cancer cell migration through ajuba transactivation. **Oncotarget**, v. 8, n. 66, p. 110415–110425, 2017.

YANG, X. et al. Overexpression of secretagogin promotes cell apoptosis and inhibits migration and invasion of human sw480 human colorectal cancer cells. **Biomed Pharmacother**, v. 101, p. 342–347, 2018.

YAO, X. Evolving artificial neural networks. In: **Proceedings of the IEEE**. [S.l.: s.n.], 1999. v. 87, n. 9, p. 1423–1447.

YEON, S. et al. Frameshift mutations in repeat sequences of ank3, hacd4, tcp10l, tp53bp1, mfn1, lcmt2, rnmt, trmt6, mettl8 and mettl16 genes in colon cancers. **Pathol Oncol Res**, p. doi: 10.1007/s12253–017–0287–2, 2017.

YU, B. et al. microrna-29c inhibits cell proliferation by targeting nasp in human gastric cancer. **BMC Cancer**, v. 17, n. 1, p. 109, 2017.

YU, J. et al. Rec8 functions as a tumor suppressor and is epigenetically downregulated in gastric cancer, especially in ebv-positive subtype. **Oncogene**, v. 36, n. 2, p. 182–193, 2017.

YU, L. et al. Tiam1 transgenic mice display increased tumor invasive and metastatic potential of colorectal cancer after 1,2-dimethylhydrazine treatment. **PLoS One**, v. 8, n. 9, p. e73077, 2013.

YU, Y. et al. Polymorphisms of inflammation-related genes and colorectal cancer risk: a population-based case-control study in china. **Int J Immunogenet**, v. 41, n. 4, p. 289–297, 2014.

Z, Y. et al. Zkscan1 gene and its related circular rna (circzkscan1) both inhibit hepatocellular carcinoma cell growth, migration, and invasion but through different signaling pathways. **Mol Oncol**, v. 11, n. 4, p. 422–437, 2017.

ZHANG, B.-T.; KIM, J.-J. Comparison of selection methods for evolutionary optimization. **Evolutionary optimization**, v. 2, n. 1, p. 55–70, 2000.

ZHANG, D.; ZHU, H.; HARPAZ, N. Overexpression of 1 chain of type xi collagen (col11a1) aids in the diagnosis of invasive carcinoma in endoscopically removed malignant colorectal polyps. **Pathol Res Pract**, v. 212, n. 6, p. 545–548, 2016.

ZHANG, H. et al. Investigating the microrna-mrna regulatory network in acute myeloid leukemia. **Oncol Lett**, v. 14, n. 4, p. 3981–3988, 2017.

ZHANG, T. et al. Sdhd promoter mutations ablate gabp transcription factor binding in melanoma. **Cancer Res**, v. 77, n. 7, p. 1649–1661, 2017.

ZHANG, X. et al. Tmem17 depresses invasion and metastasis in lung cancer cells via erk signaling pathway. **Oncotarget**, v. 8, n. 41, p. 70685–70694, 2017.

ZHU, H. et al. Genome-wide association pathway analysis to identify candidate single nucleotide polymorphisms and molecular pathways for gastric adenocarcinoma. **Tumour Biol**, v. 36, n. 7, p. 5635–5639, 2015.

ZHU, M. et al. Exome-wide association study identifies low-frequency coding variants in 2p23.2 and 7p11.2 associated with survival of non-small cell lung cancer patients. **J Thorac Oncol**, v. 12, n. 4, p. 644–656, 2017.

ZHU, Y. et al. Reactive stroma component col6a1 is upregulated in castration-resistant prostate cancer and promotes tumor growth. **Oncotarget**, v. 6, n. 16, p. 14488–14496, 2015.

APPENDIX A — PERSPECTIVES AND APPLICATIONS OF MACHINE LEARNING FOR EVOLUTIONARY DEVELOPMENTAL BIOLOGY

Journal Name

ARTICLE TYPE

Cite this: DOI: 10.1039/xxxxxxxxxx

Perspectives and Applications of Machine Learning for Evolutionary Developmental Biology

Bruno César Feltes [a‡], Bruno Iochins Grisci,[a‡], Joice de Faria Poloni [b], and Márcio Dorn [*a]

Received Date
Accepted Date

DOI: 10.1039/xxxxxxxxxx

www.rsc.org/journalname

Evolutionary Developmental Biology (Evo-Devo) is an ever-expanding field that aims to understand how development was modulated by the evolutionary process. In this sense, "omic" studies emerged as a powerful ally to unravel the molecular mechanisms underlying development. In this scenario, bioinformatics tools become necessary to analyze the growing amount of information. Among computational approaches, machine learning stands out as a promising field to generate knowledge and trace new research perspectives for bioinformatics. In this review we aim to expose the current advances of machine learning applied to evolution and development. We draw clear perspectives and argue how evolution impacted machine learning techniques.

Introduction

Evolutionary Developmental Biology (Evo-Devo) is a broad field that seeks to understand the developmental relationship among species, as well as how distinct phenotypes emerged from the evolutionary process[1,2] (Fig. 1). Hence, Evo-Devo encompasses different research approaches to elucidate the physiological, molecular, phylogenetic, and phenotypic aspects of development[1,3,4]. The molecular branching of Evo-Devo officially arose through a budding interest in the experimentation with mutants derived from different model organisms, and kept expanding ever since - from the classical genetic and molecular experiments to phylogenetic and "omic" studies, such as metagenomics, large-scale transcriptomics studies, and next-generation sequencing approaches, the so called "Big-data"[1,5–9]. Due to the inherent complexity of the developmental process together with the wide scope of Evo-Devo research interests, and the fact that such techniques often need the aid of computational methods to preprocess and analyze the massive amount of information, bioinformatics tools become crucial to accelerate and create new knowledge about the developmental aspects of evolution[10].

In the last few years, numerous bioinformatics methods have been developed and applied to molecular biology to cope with the continuous advance of DNA, RNA, and protein data[11–13]. Amidst the bioinformatics "toolkit" to analyze molecular and large-scale data, lies machine learning (ML) techniques. In short, ML is a field of Computer Science that covers several algorithms capable of performing tasks without being explicitly programmed. Being derived from studies of artificial intelligence, pattern recognition, statistics, and optimization, ML techniques "learn" how to make predictions or decisions from data alone. Classification of ML by the tasks or problems it tackles usually divides it in three categories: (i) supervised learning, that uses methods presented with data inputs and the known desired outputs, and learn to map one to another; (ii) unsupervised learning, that promotes information discovery and feature learning from data without any previous labeling, and (iii) reinforcement learning, used for computer agents that act in dynamic environments trying to maximize their rewards in order to find a policy[14] (Fig. 2).

ML has been successfully employed to analyze a broad range of biological data, such as microarray[15–18], RNA-seq[19–21], protein sequence and structural information[22–24], epigenetics[25], and genomic data[26–28]. The major difference of using ML techniques to analyze Big-data, over other computational approaches, is its capacity to extract information from large amounts of raw data and build structural descriptions that can be used for predictions and the creation of new understanding of a given problem[29]. As a matter of fact, biology and computer science are long-term partners, not only in an analytic point of view, but also through the use of metalanguage. For example, the employment of terms such as "hubs" for Systems Biology, which roughly translates to "nodes within a network with above average number of connections"[30], or how we refer to multiple centralities parameters in a biological network, has a strong computational background[31]. In many ways, how we think about a biological problem could be associated to a programming language[32–34].

[a] Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.
[b] Institute of Biotechnology, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.
[*] To whom correspondence should be sent: Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil. Tel: +55 51 3308-6824; E-mail: mdorn@inf.ufrgs.br.
[‡] These authors contributed equally to this work.

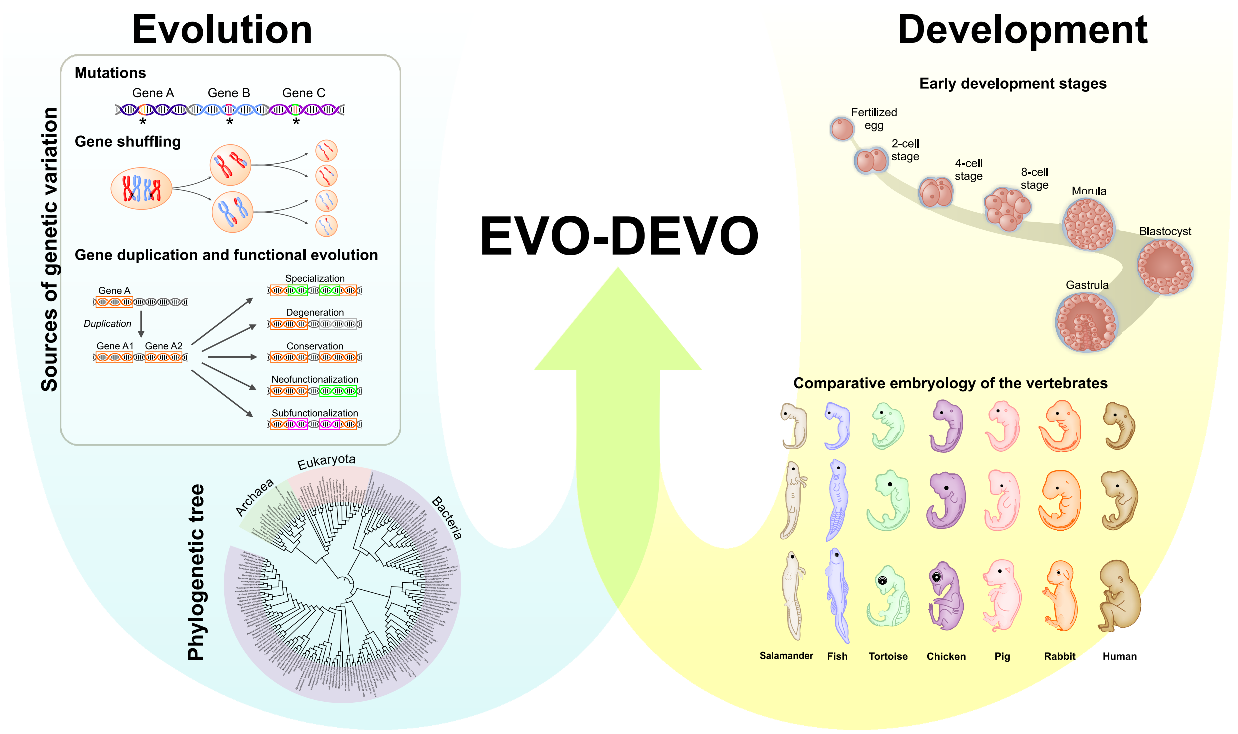


Fig. 1 A simplified illustration of the study of Evo-Devo, representing the integration of developmental processes and the evolutionary origin of phenotypic changes between organisms. The study of development is intrinsically related to the evolutionary process, and evolution-related events, reproduction, and DNA mutations, deeply impact on how an organism develops and what features will create a higher adaptability on the next generation. Hence, Evo-Devo studies encompass a wide variety of research topics and interests that aim to outline how development and evolution shaped the phenotypic variance we witness to this day.

Although ML is already widely explored to analyze Big-data, its applications not only on Evo-Devo, but in developmental and evolutionary studies that employ Big-data, are still scarce, the vast majority we found being from the last three years. Nevertheless, due to the challenges that these studies face when analyzing different types of biological data they could be aided by ML techniques. Thus, the aim of this article is to review the current applications of different ML techniques to developmental and evolutionary studies. We extensively searched the scientific literature for works employing evolutionary and developmental data, or their combination (Evo-Devo). There are extremely few examples of true Evo-Devo studies using ML, thus some studies that would not be considered an Evo-Devo topic, but could be applied to Evo-Devo, are discussed, as well as how evolution shaped ML techniques. We outline new perspectives, discuss the application of ML on different "omic" data, and propose new directions based on current knowledge.

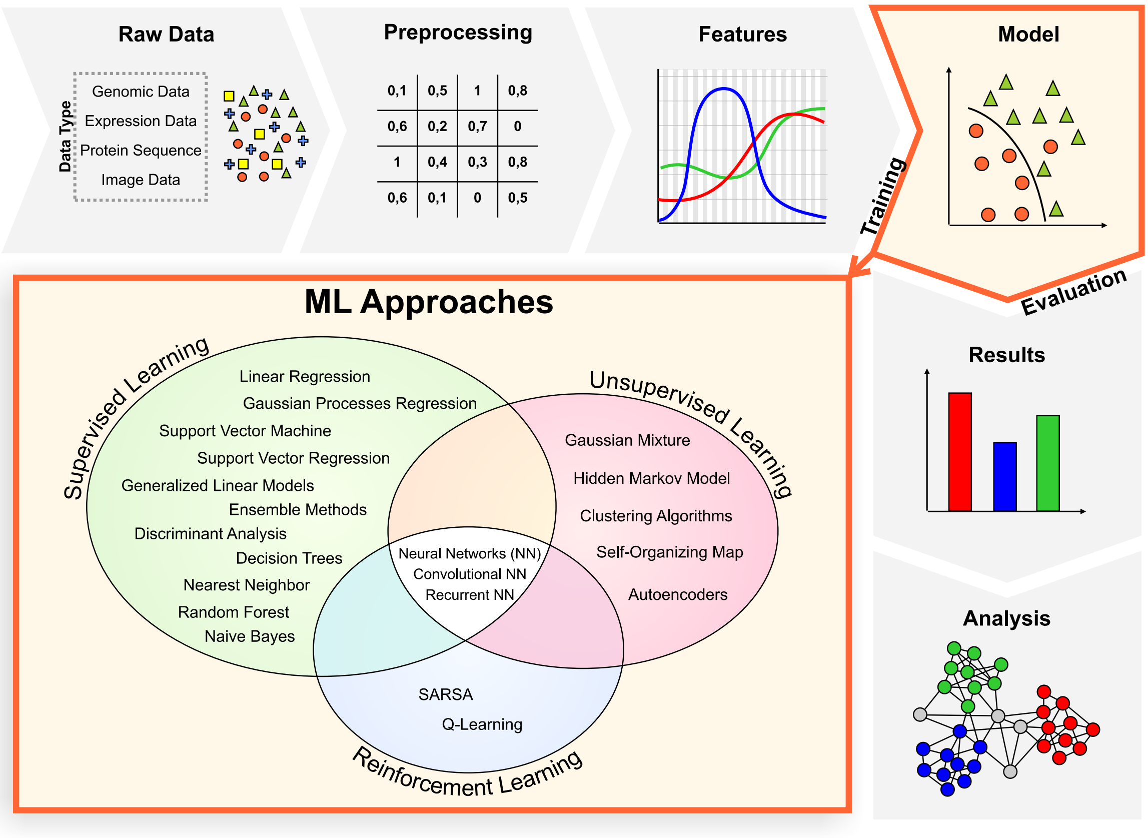
We highlight that the present review has the ultimate goal to guide bioinformatics software developers in the task of enhancing or creating new ML tools to face the technical limitations when working with biological data. We also hope to stimulate biologists to use different bioinformatics approaches when working with evolutionary and developmental "omic" data.

A Glance on Evo-Devo Thinking in the Last Decades

In the early 1980s, Evo-Devo emerged as a new research field, effectively connecting evolution and developmental biology[35]. Hence, Evo-Devo investigates the processes driving organism development and how they are modulated during evolution to create phenotypic diversity[36]. This thought arises from the methodological advances, such as gene cloning and sequencing, that allowed the identification of the conservation of regulatory genes shared by different species during embryogenesis[35]. It was observed that these genes had conserved roles throughout development, indicating developmental body structure homologies of animals with distinct body plans[35].

This knowledge originated one of the most important concepts in Evo-Devo: that the organism possesses a basic collection of genes responsible to control development, called genetic toolkit[37]. Many genes included in this toolkit encode transcription factors responsible for body structures formation[37]. The most known example is *Hox* genes, which act as important determinants of body patterning and tissue differentiation[36]. They were discovered in the fruit fly, *Drosophila melanogaster*, and posteriorly in evolutionary distant species, such as beetles, earthworms, and humans, providing the first insight of direct links between evolution and development[36].

Phenotype is controlled by distinct regulation levels of the ge-



0,1	0,5	1	0,8
0,6	0,2	0,7	0
1	0,4	0,3	0,8
0,6	0,1	0	0,5

Fig. 2 A summary of ML workflow with a schematic of a generic method and its algorithms. The raw data is obtained from measurements and experiments and is preprocessed to be applied in the ML pipeline. This step can involve cleaning, outlier removal, normalization, standardization, frequency balancing, and conversion. Additionally, features for dimensionality reduction or pattern recognition can be extracted from the data. The features are used as input to a giving training model, and its results are evaluated. The Venn-Diagram depicts a list of ML algorithms from the three major categories: supervised, unsupervised and reinforcement learning.

netic material, and this genotype-phenotype relationship is conditioned by evolutionary pressure[38]. In this sense, the majority of heritable phenotypic changes are a consequence of DNA modifications[38]. Mutations observed in *Hox* genes showed aberrant transformations of the body (termed homeosis), such as the development of the a leg pair in the fly antennae[36]. Despite this abnormal morphology, during development, restrictions of the possible phenotypic variability that may evolve occurs, and this concept is called developmental constraints[39]. Different models were proposed to describe the morphological evolution throughout development, where the most known are: (i) the hourglass model, which postulates that embryos are more variable in early development, later converging to a similar morphology during mid-development (a "phylotypic stage") and then progressively diverge; and (ii) the early conservation model, that supports the idea that at the beginning of embryogenesis is more conservative among species[39–41]. At molecular level, Piasecka *et al.* demonstrated that during the mid-development stage, regulatory elements are most conserved for transcription factors, consistent with the hourglass model. However, it was shown that the early stages of embryogenesis are less capable of tolerating gene mutations, duplication and gene introduction[39,41].

Although the field of Evo-Devo has greatly advanced our understanding of development, the question of how the morphologic changes occur at molecular level during evolution is a difficult challenge. Currently, much data about developing phenotype and genotype are available in the different databases, but the link between this information is poorly understood. The integration of information regarding genomic, transcriptomic and proteomic data of developmental and evolutionary studies by bioinformatics tools, specially by approaches that could process large volumes of information with less computational cost, could greatly propel Evo-Devo knowledge.

Brief Overview of Machine Learning Techniques

In this section we briefly explain some of the major ML approaches presented in the works reviewed in the subsequent sections. The aim of this section is not to be an exhaustive review of ML, or to review challenges, perspectives and limitations of such techniques. Its purpose is merely to elucidate some key concepts behind the most used algorithms found in Evo-Devo studies and encourage researchers to further explore this field.

Neural Networks

Artificial Neural Networks (ANN) are classical ML algorithms inspired by biological neural networks. This family of methods can theoretically approximate any continuous function and is used for supervised, unsupervised, and reinforcement learning under different architectures. The building block of any ANN is the artificial neuron, presented in the detail of Fig. 3a. This computing unit receives inputs multiplied by their respective weights, sums them plus a bias, and apply this to a nonlinear activation function. The choice of activation function will depend on the task at hand, but some of the most popular are the sigmoid, the hyperbolic tangent (tanh), and the rectified linear unit (ReLU). An ANN is built by grouping neurons in layers connected to each other, as illustrated in Fig. 3a. The input layer only corresponds to the data values, and the hidden and output layers perform the computation. A neural network with one or more hidden layers is often called a Multilayer Perceptron (MLP). The learning of these algorithms occurs by finding the best set of weights and biases that produces the desired output.

Recently, with the great advances in Big Data, parallel and distributed computing, and new optimization algorithms, we witnessed the rise of deep learning (essentially ANNs with many hidden layers), a branch of ML that became popular after being responsible for major advances in fields such as speech recognition, image recognition, robots control, and bioinformatics. The way it learns is usually by computing an error cost that informs how far the ANN is from the desired answer, and then backpropagates this error through the network[42]. The weights are then updated, often with some variation of the stochastic gradient descent (SGD)[43] algorithm. Different architectures of deep learning have been proposed for different tasks. Fig. 3b and Fig. 3c show two of the most popular: Convolutional Neural Networks (CNN)[44] and Recurrent Neural Network (RNN)[45].

CNNs are successful at analyzing spatial data, being widely used in image recognition due to their local connectivity, invariance to location and to local transition. They are formed by convolution layers, pooling layers, and fully connected layers. RNNs are designed for use with sequential information, such as text, hence the cyclic connections. Nowadays the most popular type of RNN is the long short-term memory (LSTM)[46]. ANNs are powerful algorithms, that were able to improve results in many areas that other approaches struggled for years. However, one needs to be cautious when implementing these models due to their complexity and high number of hyperparameters. Large ANNs are usually computationally expensive to train, rely in large amounts of data, and are prone to overfitting (i.e., they learn how to classify well the training data, but have poor generalization power) if regularization methods are not correctly used. Complete reviews on the topic of deep learning and biological data are found in the works of Angermueller *et al.*[47] and Min *et al.*[48].

Decision Trees

Decision trees[49] are very common classification algorithms, mostly due to their simplicity. In a nutshell, they consist of a hierarchical flowchart that, at each level, has decision blocks that ask something about the data and split it for the next level, or terminal blocks that, when reached, classify the input into the correspondent class. This can be visualized in the dummy example in Fig. 4a, that illustrates how a decision tree would classify some input with two features into four different classes. The learning in this algorithm is the construction of the trees themselves. In this sense, it is needed to find the feature from the data capable of better splitting the dataset, and repeat this process with the splits until all elements in a split belong to the same class. Usually the way to define what is the best split is through information gain, computing the entropy of the split. A high entropy means a more mixed data[50].

Decision trees have many advantages: they are computationally cheap and provide a decision structure that is easy for users to understand. They can also deal with numeric or nominal values. Unfortunately, they are very prone to overfitting[50]. The Random Forest (RF) algorithm, presented in Fig. 4b, was created to deal with this drawback. RF is an ensemble of many different decision trees that promotes a voting between them to select the final class. This greatly increases the accuracy performance of the method, at the expense of making the decision process more opaque to the user[51]. Reviews on decision trees and RF applied to bioinformatics can be found in the works of Chen *et al.*[52] and Qi[53], respectively.

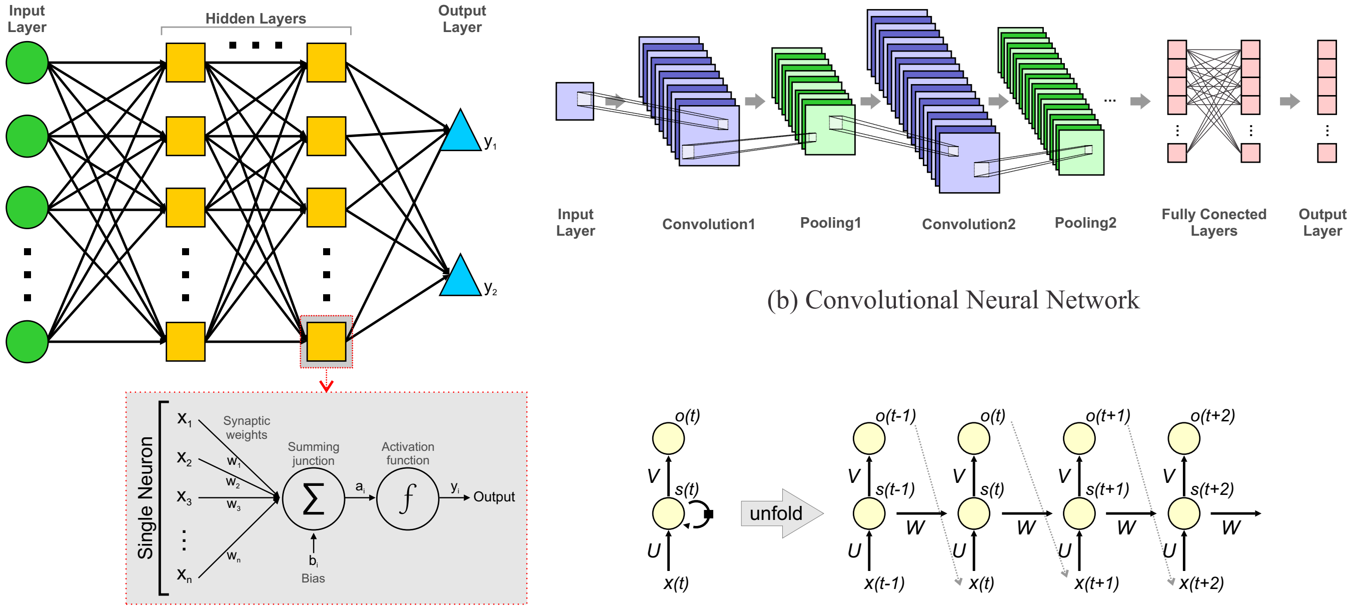
Support Vector Machines

Support Vector Machines (SVM)[54] are classifiers that work by finding the line (in 2D), plane (in 3D), or hyperplane (in larger dimensions) capable of splitting data into distinct classes. This "divider" is called a separating hyperplane and works as a decision boundary, as illustrated in Fig. 5a. The task of the learning algorithm in this case is to find the separating hyperplane that maximizes the margins (the distance between the separating hyperplane and the closest points from each class to it), known as support vectors. For data that is not linearly separable, as shown in Fig. 5b, kernels are used. They transform the data, mapping it to higher dimensions, where the separating hyperplane can be determined[50].

SVM are successful stock classifiers, meaning they perform well on new datasets without the need of being modified. They are usually not computationally costly, have low generalization errors and, for a small number of dimensions, the obtained results are easily interpretative. They have the drawback, however, of being sensitive to kernel choice and tuning parameters, what may demand higher knowledge and tests from the researcher. Besides that, in their basic implementation, SVM are only capable of performing binary classification and more complex tasks require algorithm extensions[50]. A review on bioinformatics applications using SVM is presented in the work of Byvatov and Schneider[55].

Genetic Algorithms

Genetic Algorithms (GA) are a collection of metaheuristics (stochastic methods, that makes use of randomness to find optimal or near optimal solutions for hard problems) that can be applied to several different types of optimization problems[56] -



(b) Convolutional Neural Network

(a) Artificial Neural Network and neuron

(c) Recurrent Neural Network

Fig. 3 (a) Example of an ANN. The input layer receives the numerical values, usually normalized or standardized. The hidden layers and output layer perform the computation. The number of layers, number of neurons per layer, and number of connections must be set by the user. In detail, the schematic of a single artificial neuron, with inputs, weights, bias, summation, and activation function. (b) Model of a generic CNN. The convolution layers build feature maps (groups of local weighted sums), and the pooling layers get the maximum or average sample of regions in the feature maps. (c) Detail of a simple RNN showing its cyclical connections, that allow it to perform analyzes on sequential data.

some being of the most popular options since 1970[56]. They differ from other metaheuristics in being populational methods, meaning they track a set of possible solutions that are gradually changed in order to converge to a local solution[56], and in incorporating concepts from genetics and evolution.

In GA, the candidate solutions are called "individuals" in a "population", and are represented by a "genome" that codifies their attributes. There are several genome representations, two of the most common being binary or real values vectors[57]. All solutions are given a "fitness" value, that is a measurement of their quality, dependent of the specific problem. The GA operate iteratively over the solutions, by selecting which ones will remain in the population, which will be transformed, and which will be discarded (Fig.6). There are several different strategies on how to represent a genome, or how to select individuals. The two major operators in GA, responsible for the modification of existing genomes, are crossover and mutation, and once again there are several distinct options. Crossover combines two individuals, called "parents", thus creating a new individual with characteristics from both parents, the "offspring", that possibly has better fitness[58]. The mutation randomly changes a genome, thus adding diversity and exploration in the algorithm. The core idea is to select the best individuals at each iteration (or "generation"), and combine them to create a new population, with a small chance of random mutations happening, thus converging to better solutions.

Machine Learning Applied to Development and Evolution

Although "omic" studies are broadly employed in developmental and evolutionary research, ML is still a young partner in the pursuit to generate and prospect new knowledge from Big-data in Evo-Devo. Few works mentioned in the next section have an evolutionary or developmental approach - the minority truly combine both aspects in an Evo-Devo topic. This reality is reflected on the fact that Evo-Devo is a broad topic that requires the integration of multiple kinds of biological data, a challenge we still have to overcome. Thus, all studies applied to evolution or development, with a Big-data background, that could be used for Evo-Devo are regarded, as well as other studies outside of these topics. All studies reviewed in this work can be found on Table1. In addition, the major types of data recurrently mentioned in the cited studies and the algorithms that displayed the best performance, or could be considered the best choice to work with such data for newcomers, can be found in Table2. This, however, should be followed just as an initial guidance for newcomers, as many tasks are domain specific and the expected results from some ML algorithms can vary even with the smallest modifications.

Machine Learning, Evo-Devo and Genomics

After the Human Genome Project, the way we see cellular function, evolution and disease completely changed[59]. The massive amount of genetic data accelerated the development of new studies and technologies, opening the way to the "Big-data era", gen-

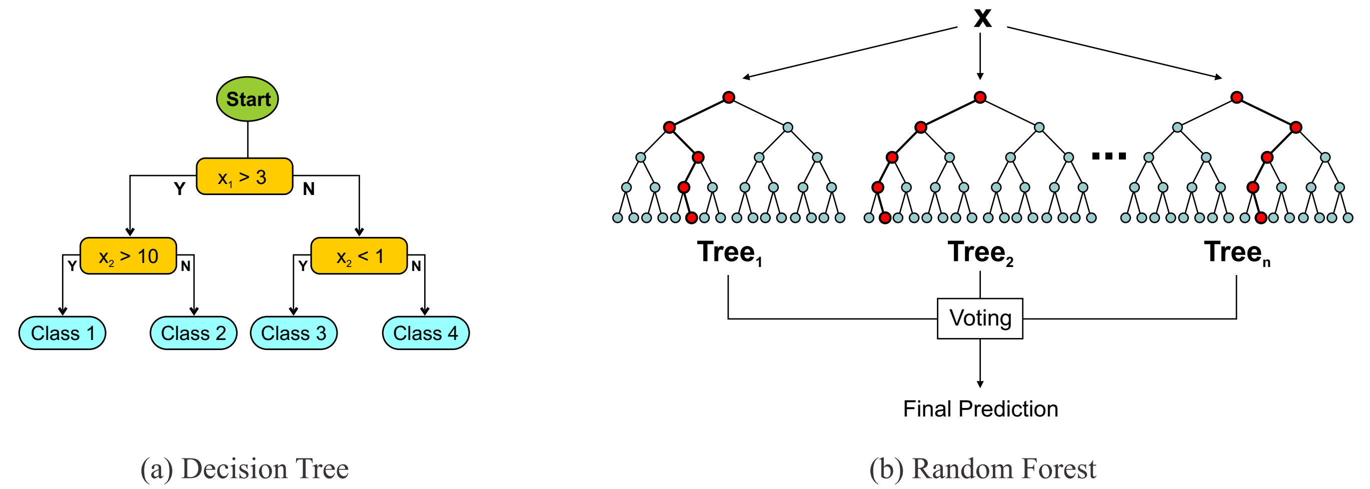


Fig. 4 (a) Dummy decision tree for the classification of data with two numerical features, x_1 and x_2, into four different classes. The branches in the tree are built to better split the data in homogeneous groups. (b) Simplified diagram showing the basic structure of the RF algorithm. For the same dataset, n decision trees are created, and the final prediction is the vote of the outputs from the individual trees.

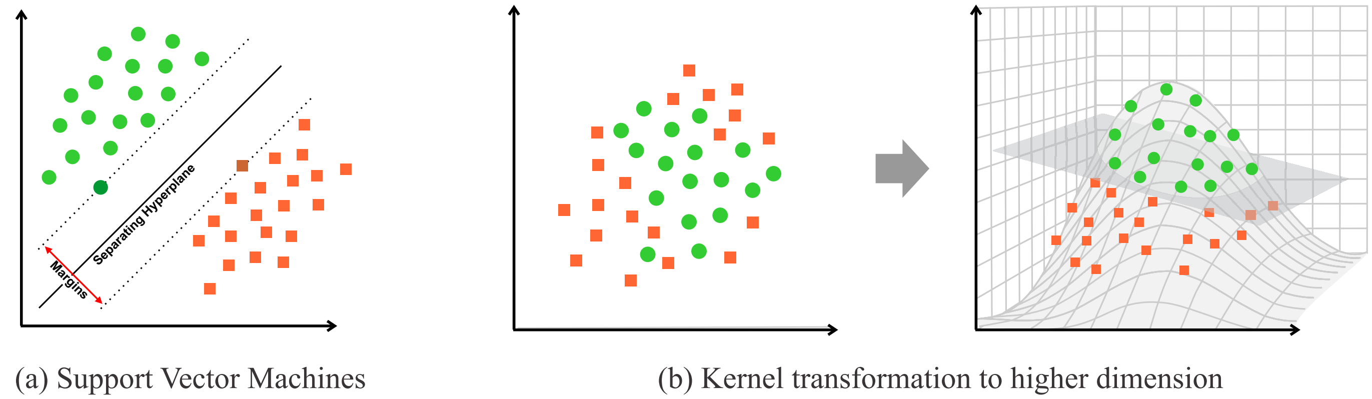

Separating Hyperplane
Margins
(a) Support Vector Machines
(b) Kernel transformation to higher dimension

Fig. 5 (a) Example of a SVM classifying data (represented by dots and squares) in 2D. In this case, the separating hyperplane is the line that best splits the data in two classes. The dots to the left belong to one class, whereas the squares on the right belongs to the other. The closest points to the separating hyperplane are the support vectors. (b) In this case, the data is not linearly separable, so a kernel transformation is applied, mapping it to a higher dimension, where a separating hyperplane exists.

erating large-scale information stored in several databases. Since then, genomic and transcriptomic data continuously expanded, providing a landscape of essential knowledge on DNA and RNA architecture and functionality. Genomic and transcriptomic data are some of the most essential aspects of molecular evolution and are often regarded as basic knowledge to any Evo-Devo study[60], and the availability of whole genome sequences of different organisms offers a robust tool to study evolutionary alterations[61,62]. An exceptional review by Necsulea and Kaessmann explains how the vertebrate transcriptome evolved between different species, organs, and chromosomes, as well as how transcriptomic changes impact on phenotype[63]. The topic of comparative transcriptomics across species is also discussed by Roux *et al.* in[64].

An evolutionary study using transcriptomic data compared developmental stages of distant species (e.g. human, worm, and fly) and revealed conserved cross-species modules enriched in functions such as morphogenesis and chromatin remodeling[65]. It was possible to identify common stage-associated genes between worm and fly for every developmental stage[65]. Interestingly, a transcriptomic meta-analysis study observed the clustering of homologous tissues belonging to distinct species, which is consistent with the concept of developmental conservation of the gene program across species[66].

One of the most crucial biological process that controls embryonic development is the epigenetic program. In this sense, DNA methylation is the best studied epigenetic modification that governs vertebrate development. Methylation patterns are responsible for transcriptional repression, chromatin architecture and cell identity across the vertebrate line, making it a pivotal subject in Evo-Devo[67–69]. An exceptional work by Yan *et al.* used RF to study the relationship between DNA methylation and histone modification in distinct genomic regions in human embryonic stem cells (hESC), fetal fibroblasts (IMR90), and H1-derived neuronal progenitor cultured stem cells (NPC) to understand the mechanisms underlying methylation dynamics on the mentioned cell types[70] (Table 1). WEKA[71] implementation of RF was chosen

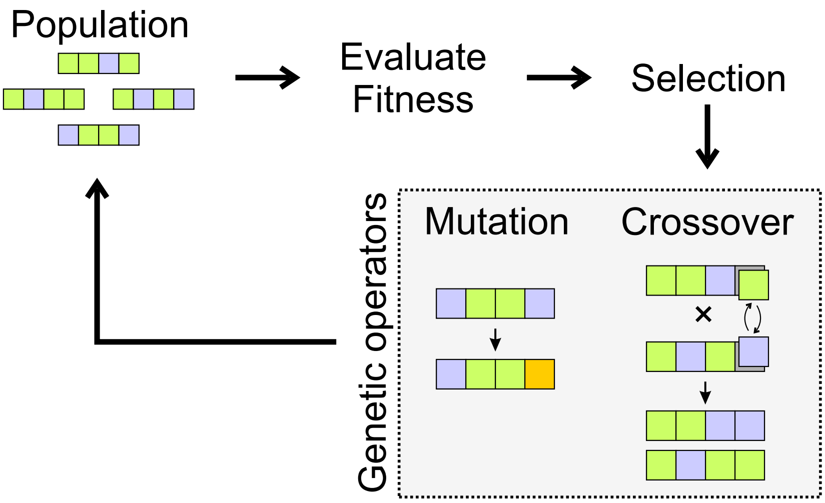


Fig. 6 Schematic of a simple GA pipeline. A population of random individuals is generated, each of them representing a candidate solution. These individuals are evaluated by some domain specific metric and, based on that, selected. The selected individuals can be subjugated to crossover or mutation operators, that create new individuals. A new population is thus created, and the process repeats until the stop criteria is met.

after it obtained the best results on a comparison of the 10-fold cross validation for 10 sampled datasets against other four algorithms: SVM with Radial Basis Function (RBF)[72] as kernel, decision tree J48 (also known as C4.5)[73], naive Bayes[14], and logistic regression[29]. The authors satisfyingly predicted methylation patterns, pointing histone modifications related to specific cell types and genomic regions. During development, chromatin regions display a dynamic and complex regulation that affects the transcriptional expression of patterning genes, especially *HOX*, shaping and modulating tissue and limb development[74]. Predicting methylation patterning shows a promising application for ML in epigenetics, by aiding to unravel chromatin dynamics.

Sheehan and Song described the first use of deep learning in population genetic models by introducing a novel likelihood-free inference framework applied for the problem of jointly inferring natural selection and demographic history[75] (Table 1) with a regular deep neural network model that took advantage of unsupervised pretraining using autoencoders for weights initialization[76]. The model was trained with 345 statistics from simulated data of different demographics for an African population of *D. melanogaster* under distinct selection parameters for each demographic history. The method was used to infer the overall demography and genomic regions under selection for 197 African *D. melanogaster* genomes from Zambia[77], learning about the history of their effective population size and selective landscape. Interestingly, the authors discovered that multiple alleles are more frequently sustained in the genetic pool (balanced selection) near centromeric regions of each chromosome, and that soft sweeps, where a neutral mutation present in a given population can become beneficial for an organism, also occur more frequently in this region.

Still in the topic of natural population genetic studies, Pybus *et al.* proposed the use of ML for the detection of positive selection in genomic regions[78] (Table 1). In this sense, the authors used boosting[79], a supervised classifier capable of maximizing the difference between two groups by estimating linear regressions of input variables. They adopted a framework with sequential consideration of four different boosting functions, creating a hierarchical decision tree, allowing it to discover different polymorphism features expected under the hard sweep model to control the demography as population specific. The algorithm was applied to three human populations from The 1000 Genome Project*, that created a genome-wide classification map of hard selective sweeps. The method achieved a rate of 5.37% sweeps misclassified as complete or incomplete. The complete sweeps were easier to classify: 89.58% were correctly classified, while only 43.41% of incomplete sweeps were correctly classified. Finally, 47.95% of the incomplete sweeps were left unclassified. The authors attribute these results to the fact that the positive selection tests detect beneficial mutations that already reached fixation.

The search for regulatory regions within a genome was always a topic of important discussion in Evo-Devo, since their evolutionary conservation usually implies critical gene expression patterns that must be fine tuned, especially during development, such is the case of Hox genes[74]. Following this line of thought, Congdon *et al.* created GAMI, a program that employed GA to unravel regulatory motifs in non-coding regions in a given genome[80]. GAMI represents the candidate solutions as sequences of nucleotides, that are evaluated with "match count", a measurement of the best consecutive match for the desired motif within the candidate solution sequence, considering forward and reverse-complement matches. The employed GA makes use of elitism and a new mutation operator that truncates one end of a motif and then adds a new base randomly at the other end.

* `http://www.internationalgenome.org`

Machine Learning, Evo-Devo and Protein Data

The understanding of protein molecular behavior, function, and structural changes along the evolutionary process are key concepts in Evo-Devo in different organisms. For example, in plant development it was established that LFY, a key inducer of floral meristemal genes in angiosperm, has a DNA-binding domain that is evolutionary conserved, but retains a nonconserved N-terminal that is likely necessary to allow the interaction of LFY with different protein complexes and promote the expression of different transcription factors[81]. Another study showed that CD24, an important regulator of cell differentiation of multiples tissues in mammals, birds, and reptiles, has an intrinsically disordered state, except in glycosylation regions of protein-ligand interaction, in which it shows evolutionary conservation, indicating that protein function and structure are critical in an evolutionary scenario[82]. Moreover, an excellent review by Londraville *et al.* discussed in detail the evolutionary roles and structural conservation of leptin, a peptide that regulates appetite and metabolic rates in several species, as well as leptin receptors[83]. In this sense, they argue how leptin has several conserved protein-protein interaction (PPI) regions, post-translational modification sites and regions necessary to protein folding[83]. Other concepts and cases of the importance of structural conservation and relation were already discussed in[84] and[85].

Nonetheless, biological phenomena are derived from the interaction of hundreds of pathways, biomolecules and chemical reactions, thus it is plausible to assume that it is virtually impossible to describe the function of a cell through the use of mathematics. However, in molecular biology, the study of protein structure and how a protein behaves is perhaps the most mathematically applicable field in Biology, since it is grounded on thermodynamics, quantum physics, and classical mechanics, and has dozens of techniques developed to study proteins conformational behavior based on their nature[86–88].

In a ML context, several studies using different approaches were applied to protein structural information. In this sense, a recent study by Farhoodi *et al.* implemented Support Vector Regression (SVR)[89], a variation of the SVM adapted to regression problems, using physico-chemical aspects and evolutionary conservation of binding regions, totaling 16 different features to rank PPI regions[90,91] (Table 1). The SVR model was trained using the RBF as kernel, with a training dataset with 6400 complexes and a testing set with 1000 complexes. The SVR approach had better performance than pyDock[92] and ClusPro[93] in identifying top-10 complexes, and achieved lower average ranking error. When compared with RosettaDock[94] the proposed method had worse ranking error by a small difference but was able to identify more top-10 complexes in six out of fifteen test cases, while RosettaDock identified more top-10 complexes in four cases. This approach clearly indicates the usefulness of ML approaches together with evolutionary data, although it doesn't have a developmental background.

Moreover, McSkimming *et al.*[95] recently described a method for protein kinase classification using protein tridimensional data from the eukaryotic lineage (Table 1). The authors created two sets of kinase amino acid chains profiles from the Protein Data Bank[96], one of labeled chains and other of unlabeled chains, with 3,365 and 1,766 elements, respectively. Each chain was defined as an unique vector with the ϕ, ψ, and $\chi 1$ angles at each aligned residue, plus the pseudo-dihedral angle through the alpha carbon of adjacent quads of residues, totalling 961 features per chain. A few feature selection algorithms, such as OneR, chi-squared, ReliefF, Gain-Ratio, and correlation-based feature selection were used together in a training set with 1,000 chains and 10-fold cross validation to select the features that better divided the data in active and inactive structures. These features were used by a RF classifier, that was reported as most accurate in comparison with naive Bayes, ANN, and SVM. All these tested algorithms achieved classification accuracies greater than 97% and could make predictions with missing atoms or residues.

Phylogenetic studies are focused in the comparison of genomic or proteomic data to draw new information about the evolutionary relationship between genes and proteins, and how this association could be related to new functions and accurate classification of gene and protein families. Phylogenetics are not a Big-data issue *per se*, but using phylogenetic concepts is proven to be useful together with structural and ML. For example, Liu successfully applied RNNs in the classification of protein function directly from amino acid sequence without sequence alignment, heuristic scoring, or feature engineering[97] (Table 1). The RNN used common LSTM and was trained on datasets from UniProt, being used in the tasks of predicting different protein functions and out-of-class predictions of phylogenetically distinct protein families that have similar functions, allowing the prediction of remote homologies, that have been highly useful for Evo-Devo studies, especially to trace homologies of development-related proteins. The inputs were the amino acids residues represented by an one-hot vector and were scanned by the forward layer of the RNN from the N- towards the C-terminus and reversed for the backward layer. This architecture allows the use of context from both sides of each position. The method was able to satisfactorily predict four functional classes: iron sequestering proteins, cytochrome P450 proteins, serine and cysteine proteases, and G-protein coupled receptors[97]. The author further tested his functional predictions by testing the iron levels in *Escherichia coli* for the iron sequestering proteins. The results showed a significant decrease in iron levels in all predicted proteins.

Khater and Mohanty took advantage of Hidden-Markov Models (HMM)[98] to identify and classify AMPylation domains in different species[99] (Table 1). HMMs, which are statistical models used for capturing consensus information from a given set, have been used for classification and identification of various protein domains[100–102], and, in this work, remarkably outperformed the results from both standalone SVM with a single feature being used to encode the sequence information, and hybrid SVM using a combination of features, besides being better to overcome insertions and selections than SVMs. The authors argue that a possible explanation for this difference in performance between their method and others is the presence of extra helices and large insertions in members of the Fido family. HMMs models for each family were build using positive datasets and multiple sequence alignment of non redundant set of proteins. The data generated by the authors helped elucidate how protein sequence and function co-evolved and how ML can be applied to both protein and phylogenetic data.

Wan *et al.* combined protein sequence and gene ontology data with RNA-seq expression profile to train a SVM model to enhance protein function identification in *D. melanogaster* development[103] (Table 1). The work makes use of the FFPred server, that inputs a query amino acid sequence to create a set of GO term predictions. After being converted into feature descriptors, this information is screened against a library of SVM. A binary decision indicating if the amino acid sequence should obtain the annotation term is output for each classifier. The GO classes are represented by five SVM classifiers trained with RBF kernels[104]. The classification system proposed by Wan *et al.* could benefit Evo-Devo studies in great length due to the integration of multiple molecular information, an approach more closely related to a developmental reality. Although the authors successfully identified new functions for unannotated proteins and were able to associate them with developmental stages, it should be noted that this was possible due to the high quantity of biological data for *D. melanogaster*.

Not related to Evo-Devo, but with a high potential as a new tool for such studies, Nauman *et al.* proposed DeepSeq, a CNN built to predict protein function[105] (Table 1). The authors used as input protein sequences from 72,945 proteins in *H. sapiens*, with maximum length of 2,000 amino acids, that were classified into five frequent GO classes, namely: (i) ATP binding; (ii) Metal ion binding; (iii) DNA biding; (iv) Zinc ion binding; and (v) Nucleic acid binding. DeepSeq outperformed BLAST, the most common algorithm used for function prediction, mostly because it showed less false positives for proteins with multiple functions, since BLAST transfers the complete annotation in case of high sequence similarity, despite the heterogeneous nature of similar proteins. The model was also reported as being able to localize the residue positions in the amino acid sequence that are involved in particular molecular activities. DeepSeq is a good example of how ML techniques can be efficient as new tools in evolutionary studies using protein sequence. However, it could be interesting to test the authors approach using a more diverse list of GOs, or data from organisms with less protein descriptions and available GOs. A similar CNN application was made for DNA sequences[106].

Finally, another study that used evolutionary information to predict phosphorylation sites was made by Biswas *et al.*[107] (Table 1). The authors created the Phosphorylation PREDictor (PPRED), a SVM classifier with RBF kernel that used sequence information of the PSSM profile employed by PSI-BLAST[108], in addition to phosphorylation information of serine (Ser), threonine (Thr) and tyrosine (Tyr) residues in Phospho.ELM[109]. Since the training data of 5724 phosphorylated proteins was unbalanced in regard of positive and negative sites annotated, the authors performed a change in the ratio of the samples in order to avoid bias in the model. Evaluating an independent benchmark, the proposed method correctly predicted 152, 57, and 74 phosphorylated Ser, Thr, and Tyr sites out of 211, 85, and 97 annotated Ser, Thr, and Tyr sites, respectively. Out of existing prediction systems, PPRED had better performance in terms of Q3 score (accuracy on the classification of the secondary structure in helix, strand, and coil) than five other predictors. The interesting aspect of this work was to predict post-translational modification sites in this particular case: phosphorylation. Nonetheless, other post-translational modifications impact on embryonic development. For example, sumoylation is related to a broad range of cellular function during the embryonic phase, but majorly in the brain[110,111]. Likewise, methylation and acetylation are also tightly associated to brain development[112,113]. Phosphorylation itself is of great importance for multiple aspects of development, as was seen in *D. melanogaster*[114]. Due to their importance, predicting regions of post-translational modifications, particularly for least-known modifications, such as sumoylation, could greatly benefit developmental studies, especially if combined to function prediction and phylogenetic studies.

New grounds to explore: Morphometric data has joined the party

In 1917, D'Arcy Wentworth Thompson published his book termed "On Growth and Form", where he discusses how biological transformations are composed by geometric shapes and governed by "laws of growth"[115]. In his book, it was founded the concept that the morphological shapes of all organisms can be described by physical and mathematical principles[115]. The morphological aspects of an organism and its tissues are the results of generative forces that acted on them, which means that the morphological growth of an organism can be generalized in all individuals within a species or related species[115]. In this sense, body shape is not explained only by a random variation that gives rise to a functional feature[115,116]. In fact, it is accepted that the "laws of growth" are responsible to create, mold and transform the morphology of biological structures, and these structures undergo natural selection, as both basis and subject of evolution[115]. Thus, it is no surprise that this new ideas of how to study the morphological aspects of an organism falls within Evo-Devo interests. A great review by Wanninger comprehensively discusses the new paradigms of the integration of morphological data in Evo-Devo research, called MorphoEvoDevo[117].

Morphogenesis is molded by mechanical forces that stimulate the movement and deformation of an element, according to its resistance[118]. These mechanical forces can be promoted by different sources, such as biophysical alterations in the local environment. Different mechanical forces are involved in development, such as osmotic pressure, shear stress, tensional forces, surface tension and spring forces[119]. Furthermore, the environment offers a great source of variability, and in an ecological context, the major influences, like the developmental temperature, chemical environment, and egg or embryo size, can affect embryonic morphogenesis[118]. These forces drive embryo shape, triggering the deformation of cells and tissues that give rise to the form and phenotype of the organism[120]. Cells are able to sense and respond to external forces and transduce these signals to the molecular machinery, expressing genes that regulate the cell fate[120]. Moreover, the cells that compose an organism are driven by a bioelectric signaling network, and thus are able to regulate pattern formation and direct the growth and form of different tissues[121]. These external influences may be converted into signals and translated to a stimulus that influences morphogenesis in different scales of time and space. The interesting aspect of this new side of morphological studies is its mathematical background, making it a perfect target for ML.

Nowadays, morphological studies are focused on exploring the evolutionary origins, transitions during development, biomechanical functions and understanding the causes and consequences of normal and abnormal variations, but studies focused on development are also being discussed[115,122]. However, the comprehension of morphological patterning and discovering how the biomechanical forces may affect the phenotype may be an important step to bioengineering and to decipher several questions regarding evolution, birth defects and regenerative medicine - and it is in aiding this comprehension that ML can be applied.

Although not Evo-Devo, a work by Masaeli *et al.*[123] shows the potential application of studying morphology to uncover differences in cell types. In this work, the authors use single cells extracts from pluripotent human Embryonic Stem Cells (hESC) and differentiated hESC and evaluate their physical properties using a microfluidic stretching flow field via high-speed microscopy and latter employs SVM to classify the differences in hESC morphologies. The results showed that pluripotent hESC becomes 15% larger, and 20% less deformable morphology after two weeks differentiation. The employed SVM used linear kernel and 5-fold cross validation, and also performed selection over features created with clustering algorithms by hierarchically eliminating features to maximize the classification at each iteration. The authors were also able to observe chromatin modifications, which were considered major players in cell morphology. Although the goal of the study was to discriminate pluripotent cells in mixed cultures, this intention does not fall back of a developmental perspective. In a nutshell, an embryo is a mixed pool of different cell types that only becomes more variate as times goes by. Being able to access and accurately discriminate the morphological changes that each tissue goes by during development, in a time-scale-dependent manner, could be an interesting perspective for Evo-Devo studies, specially by comparing these differences in distinct species.

In a truly interesting evolutionary view, Cai and Ge[124] created a pipeline to improve the discriminative classification of phytoliths at lower taxonomic levels using ML approaches. In this sense, the authors collected 1063 samples from 23 different taxa of the grass family. They measured the major parameters of phytoliths shapes using elliptic Fourier descriptors (EFDs) and applied four different ML algorithms: SVM, Decision Trees (DT), k-nearest neighbors (KNN), and multiple-layer perceptron neural networks (MLP). Although the algorithms are not clearly describe, probably indicating that, in this work, ML was just applied, not developed, their results indicated that SVM had the best accuracy at genus level and the lowest false-positive rates. The authors discuss that their study can be successfully employed to evaluate morphological measures and discriminate between different phytoliths taxa. Although it can be discussed whether one can apply this to non-plant data, the core idea behind this logical thinking has, for sure, a potential positive impact on Evo-Devo studies focused on plants.

The employment of morphometric data on ML studies, and on Evo-Devo works in general, are relatively new, with most works being published in the last 10 years. Taking advantage that these "morphometrics" are mathematical approximations and measures of distinct phenotypes, the application of ML approaches, using this kind of data is an appealing new ground to be explored.

Time, Morphology and *in silico* Predictions: New Paradigms of ML Applications in Evo-Devo

It is a fact that ML can be applied to a vast amount of different types of data, and this versatility could benefit Evo-Devo studies at great length. The following studies employ different types of data, such as images and synthetic predictions, instead of large-scale data as the ones mentioned before (Table 1).

In this sense, Namin *et al.* took advantage of CNN and LSTM algorithms to propose a framework for *Arabdopsis thaliana* from time-lapse videos in order to understand their growing patterns[125] (Table 1). The CNN was used for extracting deep features from the pictures, while the LSTM encoded the growth behavior of the plants over time. The results report that the use of CNN for classification of *A. thaliana* in four different categories (SF-2, CVI, Landsberg, and Columbia) improved the accuracy from 68% when hand-crafted features were used to 76.8% when CNN was used, and the addition of temporal information with the LSTM further improved the accuracy to 93%. This fine-tuning of video data of growing patterns could be applied to other species of plants in response to environmental conditions to simulate ecological disturbances during plant development, allowing an Eco-Evo-Devo approach to ML.

Another system used image segmentation to detect phenotypic differences throughout *Caenorhabditis elegans* embryo development[126] (Table 1). In this case, the system used Differential Interference Contrast (DIC) microscopy images to visualize important cellular functions during development, such as cytokinesis and cell-cell contacts. Therefore, quantitative measurements including the number of cells and time concerning cell division were easily achieved. Most importantly, this system allowed the analysis of specific target gene and how this gene contributes to embryo development. This task was performed by knocking down a gene, or gene set, together with the time-lapse movie record registering the effect of the selected genes knock down in the embryo development. To obtain a more reliable image segmentation, the system was divided in three main modules: (i) a CNN, which classified each pixel into five categories: cell wall, cytoplasm, nucleus membrane, nucleus and extracellular environment; (ii) an Energy-Based Model (EBM), which consist in keeping the label images produced by CNN that are associated to the correct category; and (iii) A set of elastic templates of the embryo development at different stages that are matched to the label images. The CNN was trained with series of overlapping 40 by 40 pixels from the images in the time-lapses, during six epochs, using the tanh function and the mean squared error. The training and testing frames were manually labeled and the pixel-wise error rate was

29.0% on the 30 test frames. However, the elements of embryos were clearly detected, and the nuclei were identified before, during, and after the fusion of the pro-nuclei. The cell wall is also correctly labeled, but the new cell walls created during mitosis were harder to detect[126]. This work is a formidable example of ML applied to developmental studies, and future studies using the same idea, but applied to different organisms, might be a compelling subject.

Although not evolution-related, another interesting study employed a ML model to reverse-engineer a stochastic dynamic model of regulation of melanocyte conversion in *Xenopus laevis* in order to predict the pharmacological perturbations necessary to create a given phenotype[127] (Table 1). For this, it was used a model based on Hill-kinetics with 14 stochastic ordinary differential equations that describe interactions of signaling molecules, pharmacological compounds, and level of melanocyte conversion. This dynamic signaling model of *X. laevis* conversion was introduced in the work of Lobikin *et al.*[128] and uses a genetic algorithm described in[129]. The system was used to identify treatments for wanted outcomes in complex situations, and was validated *in vivo*, confirming the computational discovery of the novel phenotype. The combined use of the three reagents found by this method led to the first predicted partial converted phenotype-animals, with some melanocytes and melanocyte-free regions being normal, and others converted and colonizing ectopic sites. The idea of predicting phenotype by inserting perturbations in regulatory networks could be an ambitious thought for Evo-Devo, by simulating changes in gene regulatory networks and creating "synthetic phenotypes".

In the same line of thinking, focusing on issues permeating the understanding of the developmental process, Spirov and Holloway[130], Aguilar-Hidalgo *et al.*[131] and François[132] provided a comprehensive review on the application of Evolutionary Computation (EC) in the prediction and modeling of Gene Regulatory Networks (GRN), providing intricate details of both methodological and biological backgrounds, as well for implementation strategies. Understanding all aspects of an organism body/structure development, from plants to mammals, is intrinsically related to the study of GRN, since those processes are an orchestra of gene expression patterns that require a delicate regulation[133–135]. It is a fact that more studies that could provide accurate recreations of GRN, taking into consideration spatio-temporal variables, or perturbations, could immensely aid Evo-Devo studies.

One of the most intriguing aspects of development is the spatio-temporal coordination of embryonic development, and understanding this process, which is a result of millions of biological interactions, is one of the major challenges of Evo-Devo. In this sense, a work from Fernández *et al.*, employed an evolutionary algorithm to create a self-regulated model that mimics a developing embryo based on tensegrity graphs, but without genetic regulation[136]. The algorithm only selects individuals and occasionally causes perturbations in theirs "genes", promoting changes in their structure. The evaluation of the individuals is measured based on the system energy. The results showed that, with minimal genetic control, the proposed method was able to create a diversity of morphologies.

Finally, an exciting work by Kriegman *et al.* employed EC to study the morphological changes of soft-robots that evolve in a simulated 3D environment[137]. In this sense, the authors created two different models: (i) the control (i.e. as if "non-treated"), named "Evo", which lacks the developmental variable and is intended to maintain a fixed morphology over its lifespan, and (ii) the experimental model, named "Evo-Devo", in which a developmental program was implemented - thus, it does not sustain a fixed phenotype. The robots "body" was implemented in the open-source soft-body physics simulator *Voxelyze*[138], their controller was a neural network, and the robots were evolved using the Age-Fitness-Pareto Optimization[139] (AFPO) algorithm, with the fitness being the average velocity of locomotion. For development, the authors implemented "ballistic development" and "developmental windows" by embedding in the robots genome intervals of values that some of their components could assume, and making them linearly transit the range of values during their lifespan. This amazing simulation of an "evolvable" organism opens new door on Evo-Devo computational studies. For example, if expression data could be added as an extra variable, modulating new phenotypes, it would greatly benefits the biological background of such studies and amplify their significance.

The idea of more experiments focusing on how to improve the application of ML to more refined models of image analysis, as well as predicting possible phenotypes is, perhaps, the most exciting future application of ML in Evo-Devo because there are few studies of this field applied to the topic, making it an easy target for newer and enhanced algorithms that could detect more accurate morphological transitions and possibly related changes to other variables, such as environmental conditions and gene mutations. The same goes for *in silico* prediction of evolutionary changes. For example, by employing algorithms that can create computational models of evolutionary phenotypical modifications over time, it could be possible to create scenarios where perturbations can be inserted, simulating environmental or genetic events that potentially alters an organism development.

It is dangerous to go alone, take this: Where you can find the data to further your research

One of the major challenges in applying ML to Evo-Devo is finding the data to begin with. Several works create their own data, thus, sometimes they become private, or can simply be found as supplementary information in the journal website. However, most works use public information to benchmark their own data, or simply use as a mean to test their new approaches. In this sense, there are a wide variety of databases where researchers can find different types of data - some extremely popular, other still to be discovered by a broader audience. In this brief section, we provide a list of databases where various types of data can be found, focusing on morphometric and image data, since DNA, RNA and protein sequence information can be obtained in a wide variety of websites. It must be noted that extremely well known databases, such as Gene Expression Omnibus [†], which contains

[†] www.ncbi.nlm.nih.gov/gds/

Table 1 Summary of the ML studies reviewed in this article, contemplating authors, studied organisms, biological background, the type of data used and the applied algorithm.

Reference	Organism	Biological background	Data	Algorithm
Yan *et al.*, 2017[70]	*H. sapiens*	Epigenetics	DNA methylation and Histone modfifications	RF
Sheehan and Song, 2016[75]	*D. melanogaster*	Chromossomic Regions	Genomic Regions/Demographic Distribution	ANN
Pybus *et al.*, 2015[78]	*H. sapiens*	Polymorphism	Genomic	Boosting
Farhoodi *et al.*, 2017[91]	*H. sapiens*	Protein Biding Regions	Protein-Protein Interaction/Sequence Conservation	SVR
Liu, 2017[97]	*H. sapiens*	Protein Function	Amino acid Sequence	RNN
Khater and Mohanty, 2015[99]	*H. sapiens*	Protein Domain	Amino acid Sequence/Post-Translational Mod.	HMM
Wan *et al.*, 2017[103]	*D. melanogaster*	Protein Function	Amino acid Sequence/Gene Ontology	SVM
Nauman *et al.*, 2017[105]	*H. sapiens*	Protein Function	Amino acid Sequence	CNN
McSkimming *et al.*, 2017[95]	*Multiple*	Protein Kinase Conformation	Protein 3D Structure	RF
Biswas *et al.*, 2010[107]	*H. sapiens*	Post-Translational Modifications	Amino acid Sequence/Post-Translational Mod.	SVM
Namin *et al.*, 2017[125]	*A. thaliana*	Plant Growth	Time-lapse Images	CNN
Ning *et al.*, 2005[126]	*C. elegans*	Embryonic Development	Differential Interference Contrast microscopy Images	CNN
Lobo *et al.*, 2017[127]	*X. laevis*	Cellular Phenotype	Hill-kinetics	GA
Congdon *et al.*, 2008[80]	*H. sapiens*	Identification of Regulatory Regions	Genomic	GA
Masaeli *et al.*, 2016[123]	*H. sapiens*	Cellular Morphology	Morphometric parameters	SVM
Cai and Ge, 2017[124]	Multiple	Paleobotany	Morphometric parameters	SVM
Spirov and Holloway, 2013[130]	Not Applicable	Embryonic Development	Not Applicable	GA
Kriegman *et al.*, 2018[137]	Not Applicable	Phenotype Prediction	Not Applicable	ANN, AFPO

Table 2 Summary of the types of data recurrently mentioned in Evo-Devo studies and the respective algorithms that are the possible options for newcomers to work with, according to the cited studies.

Evo-Devo Background	Type of Data	Problem	Algorithms
Genomic/Transcriptomic	DNA	Sequence Pattern Identification	RF, GA
	RNA	Expression Patterns Classification	SVM
Proteomic	Amino Acid Sequence	Structural Conservation Identification	CNN
	Proitein Structure	Protein Function Prediction	RNN, CNN
Phenotype Identification	Images	Visual Patterns Identification	CNN
	Morphometric	Phenotype Analysis	SVM

thousands of large-scale "omic" data from all sorts of studies, the Protein Database[‡], which is the major source of structural data, as well as sites with the same renown were not listed. Due to the massive amount of databases available nowadays and the broad spectrum of data they provide, we focused on less known websites that are more focused on developmental and evolutionary studies (Table 3). Nevertheless, we also listed some sites useful for benchmarking, and other less known repositories. Given the new importance of *in silico* studies, we also mention a physics simulator that can be used for experiments with soft-robots.

The Other Way Around: How Evolution and Development Impact on ML Techniques?

It is clear that ML techniques could be useful tools to analyze a wide variety of data in Evo-Devo studies. However, it is crucial to explain that evolution has its shares of impact on inspiring artificial intelligence algorithms and computational learning approaches. In a nutshell, natural selection is a process that selects features over time, selecting adaptable characteristics that will more likely increase organism survival. This scheme of positive feedback for the organization of a system is analogous to the learning process, and can be applied to ML studies, and the algorithms that employ the use of natural selection concepts are called Evolutionary Algorithms (EAs)[141–143].

There are different approaches in the EAs category: GA[144], that were already described in the section about ML techniques, and Differential Evolution (DE)[145] being two of the most popular. These population-based metaheuristics (algorithms independent of specific problems, capable of creating heuristics that can find solutions in optimization) are often used to solve a range of optimization problems and are loosely inspired by ideas of mutation, crossover, recombination, and selection. In this class of algorithms, a potential solution to a given problem is encoded as a "genome" in a "population", and is combined and altered over generations in order to improve its fitness (or score) value[146].

Moving to ML techniques, Neuroevolution[147] is a family of training methods for neural networks that can be used to obtain theirs weights, biases, and overall topology. Examples of such methods are the NeuroEvolution of Augmenting Topologies (NEAT)[148], the Evolutionary Deep Learning (EDL)[149], and the Evolutionary Deep Networks for Efficient Machine Learning (EDEN)[150], that incorporate GA into training. A review on the subject of Neuroevolution can be seen in the work of Ding[151]. Interestingly, the POET[152] method for optimization of weights of large ANNs is directly inspired by developmental biology. It employed an evolutionary indirect encoding and a novel parameter of search technique using an algorithm called Epigenetic Tracking (ET)[153].

Moreover, inspired by NEAT, Cussat-Blanc *et al.* created a new algorithm for the training of artificial gene regulatory networks (AGRNs), dynamical systems used in the control of agents, called

[‡] www.rcsb.org

Table 3 List of databases containing morphometric, image and genomic data that could be used to explore, benchmark or to be analyzed in ML studies focused on evolutionary and developmental biology, as well as simulators for *in silico* studies.

Name	Website	Type of Data
Reich Lab	`reich.hms.harvard.edu/`	Provide a list of various genomic datasets focused on evolution
SB Morphometrics	`life.bio.sunysb.edu/morph/index.html`	Morphometric data from different species
PRImate Morphometrics Online (PRIMO)	`primo.nycep.org/`	Morphometric studies of primates and evolution
Goldman Osteometric Dataset	`web.utk.edu/~auerbach/GOLD.html`	Osteometrics from human skeletons dating from the Holocene
Peter Brown's Australian and Asian Paleoanthropology	`www.peterbrown-palaeoanthropology.net/index.html`	Skeletal and dental metrics from human and primates
Human Origins Database	`www.humanoriginsdatabase.org/`	Fossil skeletal measurements of hominin and hominoid specimens
Paleo-Org	`www.paleo-org.com/&Morphometric`	Data of skeletal and dental records from modern and ancient humans
Australopithecus	`australopithecus.org/index.html`	Morphometric data on human evolution
Image Data Resource (IDR)[140]	`idr.openmicroscopy.org/about/`	Contains a wide variety of biological image studies
Broad Bioimage Benchmark Collection	`data.broadinstitute.org/bbbc/image_sets.html`	Useful for benchmarking image studies
Voxelyze[138]	`https://github.com/jonhiller/Voxelyze`	Voxel simulation library for static and dynamic analysis

GRNEAT[154]. This approach allowed the design of better AGRNs than regular GA and evolutionary programming strategies for the used benchmarks. Lones has a complete review on the use of AGRNs in computational problems[155].

Compositional pattern-producing networks (CPPNs)[156,157] are another architecture of Neuroevolution that differentiate themselves by adopting aspects of development, since they have the ability to bias evolutionary search to obtain solutions with regular internal structure[158]. Building upon this, Beaulieu *et al.* created a method called developmental compression[158] that explores concepts from Evo-Devo such as developmental mutations to address the problem of catastrophic forgetting, one of the major challenges in training neural networks[159,160].

Cellular Automata[161] is also an area that could benefit from Evo-Devo. The work of Nichele describes an evolutionary and developmental system with incremental evolutionary growth of genomes without any *a priori* knowledge on the necessary genotype size. This incremental growth of genome size could help artificial systems, making them able to avoid the need of knowing a genotype size and providing scalability[162].

A review by Xu[163] explores how the combined ideas from evolutionary developmental psychology, Evo-Devo, and evolutionary cognitive neurosciences are impacting the field known as Evolutionary Development Robotics (Evo-Devo-Robo). Evo-Devo-Robo is the combination of two active research topics in robotics: Evolutionary Robotics (ER), that uses evolutionary computation to create autonomous controllers, and Developmental Robotics (DevRob), with focus on the application of cognitive behaviors, such as language, emotion, and self-motivation[163]. Finally, Kenyon discusses phylogenetic and ontogenetic development as a way to implement artificial intelligence and the relationship between iterative biological development and iterative software development[164].

Perspectives: Where do We Stand, and What Could Benefit ML in Evo-Devo

The number of works applying ML to evolutionary biological data prospered in the last 5 years, with more algorithms adapted and employed to overcome challenging knowledge and technological gaps. Comprehensive reviews by Libbrecht and Noble, and Mckinney *et al.*, discussed the application of ML in genomic data, exemplifying how powerful and flexible ML techniques can be for this kind of data[165,166]. For bioinformaticians that wish to apply ML techniques in a given "omic" data, in terms of microarray data classification, SVM and RF approaches are gaining the upper hand and displaying favorable results[15,16]. Previous research showed that the distributions in microarray classification data are well represented by linear decision functions[167,168], and Statnikov *et al.* argues that SVM could be less sensitive to choice of parameters for those functions[17]. Similarly, deep learning is commonly used to work with image and temporal data, as seen previously in multiple reviews, thanks to its capacity of performing well with spatial (in the case of CNN) and sequential (in the case of RNN) data. Thus, such techniques could be an initial focus for those who are starting to apply ML techniques in biological data.

It is essential to explain that working with Evo-Devo is not an easy task for ML approaches. Most works, as presented in Table 1, are focused in either evolutionary data to answer a given subject, or with developmental data. Combining both fields in a single study requires the knowledge and manipulation of a large set of variables, including spatial-temporal and morphological information, in addition to transcriptomic data. Arbitrarily applying ML in such a complex background as Evo-Devo won't generate useful data. The use of time-lapse image analysis could be an ingenious way to integrate morphological changes, if integrated to the time-equivalent associated transcriptomic profile. Integrating spatial-temporal data would also be an interesting challenge to overcome. However, spatial-temporal analysis would require periodic sample collection that would greatly increase experimental costs. Integrating different "omic" variables, and possibly spatial-temporal data, in the same way Evo-Devo integrates several biological contexts, would be the greatest challenge in this field of research.

In addition, most works in this review used ML to perform supervised learning for classification tasks, and many challenges arise from the use of Evo-Devo data or biological data in general with this goal. One of the major concerns is the "Curse of Dimensionality", when the data has a large number of dimensions, as can be seen in microarray data or collections of pictures and videos. High dimensional data is often associated with overfitting in ML algorithms, higher processing costs and run time, increase in memory consumption, and difficulty in visualization. One way to avoid overfitting is to expand the dataset by performing new experiments, but this can be expensive and time consuming. The addition of artificially generated data should be considered only after great consideration, since it could add arbitrary values that should otherwise represent real-world phenomena. Another option, commonly used with ANNs is the incorporation of some type of regularization in the construction of the method. The works of Gonçalves *et al.* may also provide some guidance in regard of overfitting in evolutionary algorithms[169,170].

There is also the "Large p, Small n" problem for datasets with many dimensions but a small number of samples. Many ML methods, especially in supervised learning like deep learning, thrive when the samples from which they can "learn" are abundant. Successful deep learning applications usually rely in sets of thousands or even millions of examples, but for many evolutionary or developmental applications all that is available are a few dozens.

These kind of concerns should bring to light methods capable of reducing dimensionality. Among them, feature extraction is the major group of techniques capable of transforming the original feature (dimension) space of the data into a different space with a new set of axes[171]. In this case, the transformed feature space does not need to have physical or biological meaning, what can compromise interpretation[172] while providing a better discriminatory ability. Popular examples of methods are Principle Component Analysis (PCA)[173], Singular Value Decomposition (SVD)[174], Factor Analysis (FA)[175], and t-Distributed Stochastic Neighbor Embedding (t-SNE)[176]. Also relevant are autoencoders, which are ANN models used for unsupervised feature learning[76]. Feature selection is a subgroup of feature extraction that instead of transforming the original space, aims to choose a subset of relevant features by the exclusion of the irrelevant, redundant or noisy ones[177]. In many biological applications this approach is better suited since it leads to better model interpretability. An example of such method would be Minimum Redundancy Maximum Relevance (MRMR)[178]. A review of the area and its applications to genomic data can be found in the work of Ang *et al.*[179].

Researchers should also bear in mind the other major areas of ML, namely unsupervised and reinforcement learning, which were less employed in the cited reviews. The use of reinforcement learning has been growing in the past years due to its ability to "learn" without the need of sample data and the satisfactory results achieved in a wide range of applications, such as automation of vehicle and robot control[180], video games[181], and even beating humans in the game of Go[182]. This kind of algorithm shows great promise in 3D manipulation of biomolecules, and could impact Evo-Devo studies. For a complete description of reinforcement learning, refer to[183].

In general, to make life easier for both biologist and biology software developers, the application of ML in biological information can also greatly expand with the generation of more high-throughput data and greater efforts for sharing and standardizing datasets. A review by Li *et al.* discussed in depth the characteristics and application of ML in different types of datasets[184]. In fact, each platform has its unique nomenclature and data organization, which enormously difficult the integration of multiple techniques and datasets for bioinformatics in general. Specifically, one of the main challenges of a researcher that wishes to use ML methods in Evo-Devo is the lack of large, ready-to-use, well-defined sets. Despite the existing difficulties, however, ML and Evo-Devo have already shown to be powerful allies.

Conclusions

Overall, the application of ML in Evo-Devo is still young and, as discussed before, there is a wide research ground to be discovered and challenges to be overcome. The use of well defined omic datasets would greatly improve the life of both biologists and software developers, greatly boosting the application of ML in Evo-Devo. In a subject as broad as evolution and development, the application of different computational tools can propel the knowledge of the evolutionary process and open new pathways to be explored.

Key Points

- A brief explanation of the major thinking behind Evo-Devo and machine learning techniques is provided.
- We review the current works concerning the application of machine learning on evolutionary and developmental data. All types of works that could impact on Evo-Devo were taken into consideration after an extensive review of the literature.
- The selected works are comprehensively reviewed concerning the employed algorithms, biological backgrounds and major results.
- Other works, not necessarily related to Evo-Devo, that could provide new insights on the field and ML applications are also reviewed.
- New perspectives are drawn based on the gathered data for the application of machine learning on Evo-Devo.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This work was supported by grants from FAPERGS [16/2551-0000520-6], MCT/CNPq [311022/2015-4], CAPES-STIC AM-SUD [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.

Notes and references

1 S. Kuraku, N. Feiner, S. D. Keeley *et al.*, *Dev Growth Differ*, 2016, **58**, 131–142.

2 C. S. Campbell, C. E. Adams, C. Bean *et al.*, *Trends Ecol Evol*, 2017, **32**, 746–759.

3 G. B. Müller, *Nat Rev Genet*, 2007, **8**, 943–949.

4 R. Brown, *Entangled Life*, Springer, Dordrecht, Elsevier Inc, First Edition edn, 2014, pp. 237–260.

5 A. M. Cheatle Jarvela and L. Pick, *Curr Top Dev Biol*, Academic Press, Elsevier, Inc, First Edition edn, 2016, vol. 117, pp. 253–274.

6 S. Pantalacci and M. SÃl'mon, *J Exp Zool B Mol Dev Evol*, 2015, **324**, 363–371.

7 J. Alföldi and K. Lindblad-Toh, *Genome Res*, 2013, **23**, 1063–1068.

8 M. Leonardi, P. Librado, C. Der Sarkissian *et al.*, *Syst Biol*, 2017, **66**, e1–e29.

9 T. J. Colston and C. R. Jackson, *Mol Ecol*, 2016, **65**, 3776–3800.

10 P. M. Mabee, *BioScience*, 2006, **56**, 301–309.

11 O. Morozova, M. Hirst and M. A. Marra, *Annu Rev Genomics Hum Genet*, 2009, **10**, 135–151.

12 R. Lowe, N. Shirley, M. Bleackley *et al.*, *PLoS Comput Biol*, 2017, **13**, e100545.

13 A. Oulas, C. Pavloudi, P. Polymenakou *et al.*, *Bioinform Biol Insights*, 2015, **9**, 75–88.

14 S. J. Russell, P. Norvig and E. Davis, *Artificial intelligence: a modern approach*, Pearson Education, Limited, New Jersey, 2016.

15 J. Lee, J. Lee, M. Park *et al.*, *Comput Stat Data Anal*, 2005, **48**, 869–885.

16 M. Pirooznia, J. Y. Yang, M. Q. Yang *et al.*, *BMC Genomics*, 2009, **9**, S13.

17 A. Statnikov, L. Wang and C. Aliferis, *BMC Bioinformatics*, 2008, **9**, 319.

18 Y. Li, A. A. Jourdain, S. E. Calvo *et al.*, *PLoS Comput Biol*, 2017, **13**, e1005653.

19 M. G. Best, N. Sol, I. Kooi *et al.*, *Cancer Cell*, 2015, **25**, 666–676.

20 C. Lin, S. Jain, H. Kim *et al.*, *Nucleic Acids Res*, 2017, **45**, e156.

21 M. K. Leung, H. Y. Xiong, L. J. Lee *et al.*, *Bioinformatics*, 2014, **30**, i121–i129.

22 B. Grisci and M. Dorn, *J Bioinform Comput Biol*, 2017, **15**, 1750009.

23 S. Sønderby and O. Winther, *arXiv:1412.7828*, 2015.

24 M. Dorn, M. E Silva, L. Buriol *et al.*, *Comput Biol Chem*, 2014, **53**, 251–276.

25 C. Angermueller, H. J. Lee, W. Reik *et al.*, *Genome Biol*, 2017, **18**, 67.

26 Y. Park and M. Kellis, *Nat Biotechnol*, 2015, **33**, 825–826.

27 N. Giang Nguyen, V. Tran, D. Ngo *et al.*, *J Biomed Sci Eng*, 2016, **9**, 280–286.

28 Y. Z. Zhang, R. Yamaguchi, S. Imoto *et al.*, *BMC Genomics*, 2017, **18**, 1044.

29 I. H. Witten, E. Frank, M. A. Hall *et al.*, *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, Elsevier, Cambridge, MA, USA, 2016.

30 A. L. Barabási and Z. N. Oltvai, *Nat Rev Genet*, 2004, **5**, 101–113.

31 M. E. J. Newman, *Soc Net*, 2005, **27**, 39–54.

32 A. Livnat and C. Papadimitriou, *Trends Ecol Evol*, 2016, **31**, 894–896.

33 R. A. Watson and E. Szathmáry, *Trends Ecol Evol*, 2016, **31**, 896–898.

34 A. Spirov and D. Holloway, *Evolutionary Computation in Gene Regulatory Network Research*, John Wiley Sons, Inc, Hoboken, NJ, USA, First Edition edn, 2016, pp. 240–268.

35 R. A. Raff, *Nat Rev Genet*, 2000, **1**, 74–79.

36 A. Heffer and L. Pick, *Annu Rev Entomol*, 2013, **58**, 161–179.

37 S. B. Carroll, *Cell*, 2008, **134**, 25–36.

38 P. W. Harrison, A. E. Wright and J. E. Mank, *Semin Cell Dev Biol*, 2012, **23**, 222–229.

39 J. Roux and M. Robinson-Rechavi, *PLoS Genet*, 2008, **4**, e1000311.

40 A. T. Kalinka and P. Tomancak, *Trends Ecol Evol*, 2012, **27**, 385–393.

41 B. Piasecka, P. Lichocki, S. Moretti *et al.*, *PLoS Genet*, 2013, **9**, e1003476.

42 Y. LeCun, L. Bottou, G. B. Orr *et al.*, *Neural networks: Tricks of the trade*, Springer, 1998, pp. 9–50.

43 J. Kiefer and J. Wolfowitz, *Ann Math Stat*, 1952, 462–466.

44 Y. LeCun, Y. Bengio and G. Hinton, *Nature*, 2015, **521**, 436.

45 L. C. Jain and L. R. Medsker, *Recurrent Neural Networks: Design and Applications*, CRC Press, Inc., Boca Raton, FL, USA, 1st edn, 1999.

46 S. Hochreiter and J. Schmidhuber, *Neural Comput*, 1997, **9**, 1735–1780.

47 C. Angermueller, T. Pärnamaa, L. Parts *et al.*, *Mol Syst Biol*, 2016, **12**, 878.

48 S. Min, B. Lee and S. Yoon, *Briefings in bioinformatics*, 2017, **18**, 851–869.

49 C. J. Stone, *Classification and regression trees*, Taylor & Francis Group, LLC, Boca Raton, FL.

50 P. Harrington, *Machine learning in action*, Manning Greenwich, CT, Shelter Island, NY 11964, 2012, vol. 5.

51 L. Breiman, *Machine learning*, 2001, **45**, 5–32.

52 X. Chen, M. Wang and H. Zhang, *Wiley Interdiscip Rev Data Min Knowl Discov*, 2011, **1**, 55–63.

53 Y. Qi, *Ensemble machine learning*, Springer, 2012, pp. 307–323.

54 C. Cortes and V. Vapnik, *Machine learning*, 1995, **20**, 273–297.

55 E. Byvatov and G. Schneider, *Appl Bioinformatics*, 2003, **2**, 67–77.

56 S. Luke, *Essentials of metaheuristics*, Lulu, 1st edn, 2009, p. 227.

57 T. Kuthan and J. Lansky, *Dateso*, 2007, 21–34.

58 D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edn, 1989.

59 E. D. Green, J. D. Watson and F. S. Collins, *Nature*, 2015, **526**, 29–31.

60 A. M. Cheatle Jarvela and V. F. Hinman, *Evodevo*, 2015, **6**, 3.

61 T. Liu, L. Yu, L. Liu *et al.*, *Comput Math Methods Med*, 2015, **2015**, 896176.

62 E. Lécuyer and P. Tomancak, *Curr Opin Genet Dev*, 2008, **18**, 506–512.

63 A. Necsulea and H. Kaessmann, *Nat Rev Genet*, 2014, **15**, 734–748.

64 J. Roux, M. Rosikiewicz and M. Robinson-Rechavi, *J Exp Zool B Mol Dev Evol*, 2015, **324**, 372–382.

65 M. B. Gerstein, J. Rozowsky, K. K. Yan *et al.*, *Nature*, 2014, **512**, 445–448.

66 P. H. Sudmant, M. S. Alexis and C. B. Burge, *Genome Biol*, 2015, **16**, 287.

67 O. Bogdanovic and J. L. Gomez-Skarmeta, *Brief Funct Genomics*, 2014, **13**, 121–130.

68 O. Bogdanović and R. Lister, *Curr Opin Genet Dev*, 2017, **46**, 9–14.

69 Z. D. Smith and A. Meissner, *Nat Rev Genet*, 2013, **14**, 204–220.

70 H. Yan, D. Zhang, H. Liu *et al.*, *Sci Rep*, 2017, **5**, 8410.

71 F. Frank, M. A. Hall and I. H. Witten, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*, Morgan Kaufmann, 50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States, 4th edn, 2016.

72 J.-P. Vert, K. Tsuda and B. Schölkopf, *Kernel Methods in Computational Biology*, MIT Press, Cambridge, MA, 2004, vol. 47, pp. 35–70.

73 J. R. Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann, San Mateo, CA, 2014.

74 J. Deschamps and D. Duboule, *Genes Dev*, 2017, **31**, 1406–1416.

75 S. Sheehan and Y. S. Song, *PLoS Comput Biol*, 2016, **12**, e1004845.

76 G. E. Hinton and R. R. Salakhutdinov, *Science*, 2006, **313**, 504–507.

77 J. B. Lack, C. M. Cardeno, M. W. Crepeau and Tothers, *Genetics*, 2015, **199**, 1229–1241.

78 M. Pybus, P. Luisi, G. M. Dall'Olio *et al.*, *Bioinformatics*, 2015, **31**, 3946–3952.

79 R. E. Schapire, *Machine learning*, 1990, **5**, 197–227.

80 C. Congdon, J. Aman, G. Nava *et al.*, *IEEE/ACM Trans Comput Biol Bioinform*, 2008, **5**, 1–14.

81 C. S. Silva, S. Puranik, A. Round *et al.*, *Front Plant Sci*, 2016, **6**, 1193.

82 D. C. Ayre, N. K. Pallegar, N. A. Fairbridge *et al.*, *Gene*, 2016, **590**, 324–337.

83 R. L. Londraville, J. W. Prokop, R. J. Duff *et al.*, *Front Endocrinol (Lausanne)*, 2017, **8**, 58.

84 A. Andreeva, *Biochem Soc Trans*, 2016, **44**, 937–943.

85 A. Valencia and F. Pazos, *Methods Biochem Anal*, 2003, **44**, 411–426.

86 J. Echave and C. O. Wilke, *Annu Rev Biophys*, 2017, **46**, 85–103.

87 R. C. Bernardi, M. C. Melo and K. Schulten, *Biochim Biophys Acta*, 2015, **1850**, 872–877.

88 J. R. Perilla, B. C. Goh, C. K. Cassidy *et al.*, *Curr Opin Struct Biol*, 2015, **31**, 64–74.

89 H. Drucker, C. J. C. Burges, L. Kaufman *et al.*, author, 1997, pp. 155–61.

90 A. Wilkins, S. Erdin, R. Lua *et al.*, *Methods Mol Biol*, 2012, **819**, 29–42.

91 R. Farhoodi, B. Akbal-Delibas and N. Haspel, Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics: 20-23 August 2017; Boston, Massachusetts, USA, 2017.

92 S. Grosdidier, C. Pons and A. Solernou, *Proteins*, 2007, **69**, 852–858.

93 S. R. Comeau, D. W. Gatchell, S. Vajda *et al.*, *Nucleic Acids Res*, 2004, **1**, 32.

94 S. A. Combs, S. L. Deluca, S. H. Deluca *et al.*, *Nat Protoc*, 2013, **8**, 1277–1298.

95 D. I. McSkimming, K. Rasheed and N. Kannan, *BMC Bioinformatics*, 2017, **18**, 86.

96 H. M. Berman, J. Westbrook, Z. Feng *et al.*, *Nucleic Acids Res*, 2000, **28**, 235–242.

97 X. Liu, *arXiv preprint arXiv:1701.08318*, 2017.

98 L. Rabiner and B. Juang, *IEEE ASSP Mag*, 1986, **3**, 4–16.

99 S. Khater and D. Mohanty, *Sci Rep*, 2015, **5**, 10804.

100 M. Z. Ansari, J. Sharma, R. S. Gokhale *et al.*, *BMC Bioinformatics*, 2008, **9**, 454.

101 K. Blin, M. H. Medema, D. Kazempour *et al.*, *Nucl Acid Res*, 2013, **41**, W204–W212.

102 G. Yadav, R. S. Gokhale and D. Mohanty, *PLoS Comput Biol*, 2009, **5**, e1000351.

103 C. Wan, L. J. G, F. Minneci *et al.*, *PLoS Comput Biol*, 2017, **13**, e1005791.

104 A. E. Lobley, T. Nugent, C. A. Orengo *et al.*, *Nucl Acid Res*, 2008, **36**, W297–W302.

105 M. Nauman, H. U. Rehman, G. Politano *et al.*, *bioRxiv*, 2017.

106 N. G. Nguyen, V. A. Tran, D. L. Ngo *et al.*, *J Biomed Sci Eng*, 2016, **9**, 280–286.

107 A. K. Biswas, N. Noman and A. R. Sikder, *BMC Bioinformatics*, 2017, **11**, 273.

108 S. Kaushik, E. Mutt, A. Chellappan *et al.*, *PLoS One*, 2013, **8**, e56449.

109 H. Dinkel, C. Chica, A. Via *et al.*, *Nucleic Acids Res*, 2011, **Database Issue**, D261–D267.

110 Z. Hannoun, S. Greenhough, E. Jaffray *et al.*, *Toxicology*, 2010, **278**, 288–293.

111 C. Gwizdek, F. Cassé and S. Martin, *Neuromolecular Med*, 2013, **15**, 2677–2691.

112 M. P. Mattson, *Ageing Res Rev*, 2003, **2**, 329–342.

113 A. Tapias and Z. Q. Wang, *CGenomics Proteomics Bioinformatics*, 2017, **15**, 19–36.

114 R. Sopko and N. Perrimon, *Cold Spring Harb Perspect Biol*, 2013, **5**, pii: a009050.

115 A. Abzhanov, *Development*, 2017, **144**, 4284–4297.

116 E. M. De Robertis, Y. Moriyama and C. G, *Dev Growth Differ*, 2017, **59**, 580–592.

117 A. Wanninger, *Frontiers in Ecology and Evolution*, 2015, **3**, 1–9.

118 M. von Dassow and L. A. Davidson, *Phys Biol*, 2011, **8**, 045002.

119 T. Mammoto and D. E. Ingber, *Development*, 2010, **137**, 1407–1420.

120 C. J. Miller and D. L. A, *Nat Rev Genet*, 2013, **14**, 733–744.

121 M. Levin and C. J. Martyniuk, *Biosystems*, 2018, **164**, 76–93.

122 B. Hallgrimsson, C. Percival, R. Green, N. Young, W. Mio *et al.*, *Curr Top Dev Biol*, 2015, **115**, 561–597.

123 M. Masaeli, D. Gupta, S. O'Byrne, H. Tse, D. Gossett *et al.*, *Sci Rep*, 2016, **6**, 37863.

124 z. Cai and S. Ge, *Journal of Systematics and Evolution*, 2017, **55**, 377–384.

125 S. T. Namin, M. Esmaeilzadeh, N. M *et al.*, *bioRxiv*, 2017, **doi: https://doi.org/10.1101/134205**, year.

126 F. Ning, D. Delhomme, Y. LeCun *et al.*, *IEEE Trans Image Process*, 2005, **14**, 1360–1371.

127 D. Lobo, M. Lobikin and M. Levin, *Sci Rep*, 2017, **7**, 41339.

128 M. Lobikin, D. Lobo, D. J. Blackiston *et al.*, *Sci Signal*, 2015, **8**, ra99.

129 D. Lobo and M. Levin, *PLoS Comput Biol*, 2015, **11**, e1004295.

130 A. Spirov and D. Holloway, *Methods*, 2013, **62**, 39–55.

131 D. Aguilar-Hidalgo, M. Lemos and A. Córdoba, *Computation*, 2015, **3**, 99–113.

132 P. FranÃğois, *Semin Cell Dev Biol*, 2014, **35**, 90–97.

133 J. Murray, *Wiley Interdiscip Rev Dev Biol*, 2018, **7**, e314.

134 H. Parker, I. Pushel and R. Krumlauf, *Dev Biol*, 2018, **pii: S0012-1606**, 30597–3.

135 M. Das Gupta and M. Tsiantis, *Curr Opin Plant Biol*, 2018, **45**, 82–87.

136 J. Fernández, F. Vico and R. Doursat, *Complex and diverse morphologies can develop from a minimal genomic model*, 2012.

137 S. Kriegman, N. Cheney and J. Bongard, *arXiv:1711.07387*.

138 J. Hiller and H. Lipson, *Soft robotics*, 2014, **1**, 88–101.

139 M. Schmidt and H. Lipson, *Genetic Programming Theory and Practice VIII*, Springer, 2011, pp. 129–146.

140 E. Williams, J. Moore, S. Li, G. Rustici, A. Tarkowska *et al.*, *Nat Methods*, 2017, **14**, 775–781.

141 K. Kouvaris, J. Clune, L. Kounios *et al.*, *PLoS Comput Biol*, 2017, **13**, e1005358.

142 R. A. Watson, R. Mills, C. L. Buckley *et al.*, *Evol Biol*, 2016, **43**, 553–581.

143 M. Sipper, R. S. Olson and J. H. Moore, *BioData Min*, 2017, **10**, 26.

144 W. Banzhaf, P. Nordin, R. E. Keller *et al.*, *Genetic programming: an introduction*, Morgan Kaufmann, San Francisco, 1998, vol. 1.

145 R. Storn and K. Price, *J Global Opt*, 1997, **11**, 341–359.

146 S. Luke, *Essentials of Metaheuristics*, Lulu, Morrisville, North Carolina, Second Edition edn, 2013.

147 D. Floreano, P. Dürr and C. Mattiussi, *Evol Intel*, 2008, **1**, 47–62.

148 K. O. Stanley and R. Miikkulainen, *Evol Comput*, 2002, **10**, 99.

149 E. Dufourq and B. A. Bassett, *arXiv preprint arXiv:1707.00703*, 2017.

150 E. Dufourq and B. A. Bassett, *arXiv preprint arXiv:1709.09161*, 2017.

151 S. Ding, H. Li, C. Su *et al.*, *Artif Intell Rev*, 2013, 1.

152 A. Fontana, A. Soltoggio and B. Wróbel, *POET: an evo-devo method to optimize the weights of a large artificial neural networks*, 2014.

153 A. Fontana, European Conference on Artificial Life, 2009, pp. 10–17.

154 S. Cussat-Blanc, K. Harrington and J. Pollack, *IEEE Transactions on Evolutionary Computation*, 2015, **19**, 823–837.

155 M. A. Lones, *Evolutionary Computation in Gene Regulatory Network Research*, 2016, 398–424.

156 K. O. Stanley, *Genetic programming and evolvable machines*, 2007, **8**, 131–162.

157 K. O. Stanley, D. B. D'Ambrosio and J. Gauci, *Artificial life*, 2009, **15**, 185–212.

158 S. L. Beaulieu, S. Kriegman and J. C. Bongard, *arXiv preprint arXiv:1804.04286*, 2018.

159 R. M. French, *Trends in cognitive sciences*, 1999, **3**, 128–135.

160 I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville and Y. Bengio, *arXiv preprint arXiv:1312.6211*, 2013.

161 B. Chopard and M. Droz, *Cellular automata*, Springer, Amsterdam, The Netherlands, 1998.

162 S. Nichele, A. Giskeødegård and G. Tufte, *Artif Life*, 2016, **22**, 76–111.

163 B. Xu, H. Min and F. Xiao, *Ind Rob*, 2014, **41**, 527–533.

164 S. H. Kenyon, AAAI Fall Symposium Series, 15-17 November 2013, Arlington, Virginia, 2013.

165 M. W. Libbrecht and W. S. Noble, *Nat Rev Genet*, 2015, **16**, 321–332.

166 B. A. McKinney, D. M. Reif, M. D. Ritchie *et al.*, *Appl Bioinformatics*, 2005, **5**, 77–88.

167 S. Dudoit, J. Fridlyand and T. P. Speed, *Journal of the American statistical association*, 2002, **97**, 77–87.

168 A. Dupuy and R. M. Simon, *Journal of the National Cancer Institute*, 2007, **99**, 147–157.

169 I. Gonçalves, S. Silva, J. B. Melo and J. M. Carreiras, European Conference on Genetic Programming, 2012, pp. 218–229.

170 I. Gonçalves and S. Silva, European Conference on Genetic Programming, 2013, pp. 73–84.

171 R. Varshavsky, A. Gottlieb, M. Linial *et al.*, *Bioinformatics*, 2006, **22**, e507–e513.

172 P. Krızek, *PhD thesis*, PhD thesis, Czech Technical University in Prague, 2008. 6, 14, 36, 67, 93, 2008.

173 I. T. Jolliffe and J. Cadima, *Philos Trans A Math Phys Eng Sci*, 2016, **374**, 20150202.

174 V. Klema and A. Laub, *IEEE Trans Autom Control*, 1980, **25**, 164–176.

175 B. Fruchter, 1954.

176 L. van der Maaten and G. Hinton, *Journal of Machine Learning Research*, 2008, **9**, 2579–2605.

177 J. Miao and L. Niu, *Procedia Computer Science*, 2016, **91**, 919–926.

178 H. Ding, Cand Peng, *J Bioinform Comput Biol*, 2005, **3**, 185–205.

179 J. C. Ang, A. Mirzal, H. Haron *et al.*, *IEEE/ACM transactions on computational biology and bioinformatics*, 2016, **13**, 971–989.

180 S. Gu, E. Holly, T. Lillicrap *et al.*, Robotics and Automation (ICRA), 2017 IEEE International Conference on, 2017, pp. 3389–3396.

181 V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, *Nature*, 2015, **518**, 529.

182 D. Silver, J. Schrittwieser, K. Simonyan *et al.*, *Nature*, 2017, **550**, 354.

183 R. S. Sutton and A. G. Barto, *Reinforcement Learning : An Introduction*, MIT Press, Favoritenstrasse 9/4th Floor/1863, 1998.

184 Y. Li, F. X. Wu and A. Ngom, *Brief Bioinform*, 2016, **pii**, bbw113.

APPENDIX B — MICROARRAY CLASSIFICATION AND GENE SELECTION WITH FS-NEAT

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*

2005 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.

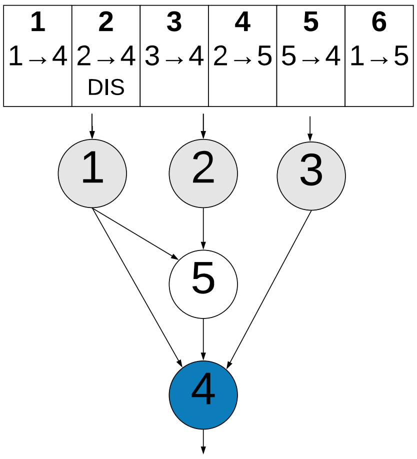


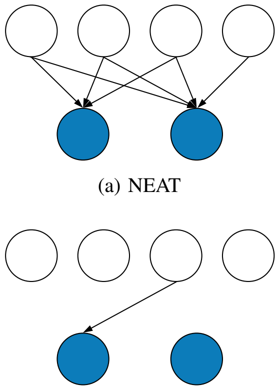
Fig. 1: Genome representation for an individual in the NEAT population. The bold number in the top line of each gene is the historical marker used to identify new structural transformations. The second line informs the link between two nodes. The third line is the disable bit (DIS) that when active means 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_ji is the jth element of the ith sample of the desired output from class q, and a_ji 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.

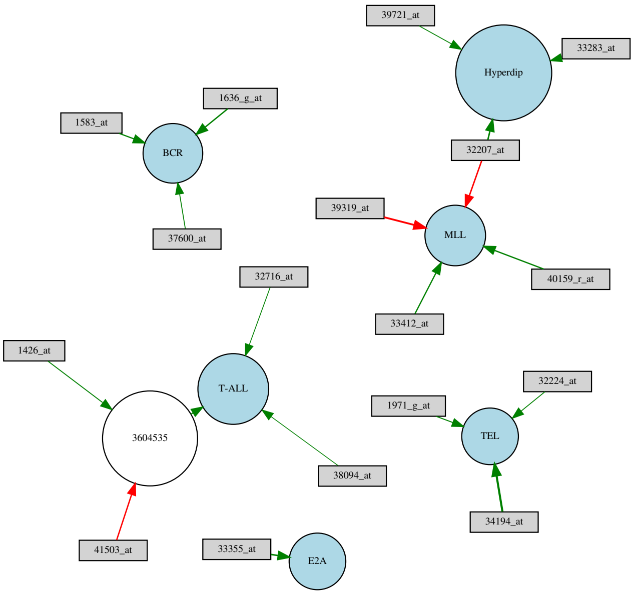


Fig. 3: Example of an ANN created with FS-NEAT using the data from the leukemia dataset. Rectangles are input nodes associated to a specific gene, white nodes are hidden nodes, and colored nodes are output nodes associated to a specific subtype of leukemia. The arrows are connections, with their thickness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