DE GRUYTER OPEN

The EuroBiotech Journal

REVIEW

Bioinformatics / System Biology

Artificial intelligence used in genome analysis studies

Edo D'Agaro

Abstract

Next Generation Sequencing (NGS) or deep sequencing technology enables parallel reading of multiple individual DNA fragments, thereby enabling the identification of millions of base pairs in several hours. Recent research has clearly shown that machine learning technologies can efficiently analyse large sets of genomic data and help to identify novel gene functions and regulation regions. A deep artificial neural network consists of a group of artificial neurons that mimic the properties of living neurons. These mathematical models, termed Artificial Neural Networks (ANN), can be used to solve artificial intelligence engineering problems in several different technological fields (e.g., biology, genomics, proteomics, and metabolomics). In practical terms, neural networks are non-linear statistical structures that are organized as modelling tools and are used to simulate complex genomic relationships between inputs and outputs. To date, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNN) have been demonstrated to be the best tools for improving performance in problem solving tasks within the genomic field.

Keywords: deep learning, NGS, genomics, molecular diagnosis

Department of Agricultural, Food, Environment and Animal Sciences, University of Udine, Italy

*Corresponding author: E. D'Agaro
E-mail: edo.dagaro@uniud.it

DOI: 10.2478/ebtj-2018-0012

Introduction

According to a recent study, artificial intelligence could stimulate employment and increase the general business revenues by 38% in 2020 (1,2). For the global economy, this effect would mean a profit growth of approximately $4.8 trillion and, above all, a profound rethinking of the production system. In this model, the information (input) (e.g., genomic data) is entered into the node (simulated neuron) and processed. This approach leads to an initial transient result that is passed to the upper level, where the process is repeated. From the multiple levels, we arrive at the final stage of information processing (e.g., prediction of gene function and structure). The input (initial information) is defined as specific data, and the output (final information) must be consistent with the input (3- 7).

Artificial intelligence methods

The design of learning elements must consider three fundamental aspects:

- ✓ which components of the executive element (e.g. gene regulation and structures) should be learned;
- ✓ the type of feedback available;
- ✓ the type of representation used.

The feedback step is the most important because it enables us to determine the nature of the problem. We can distinguish three types of learning:

Supervised learning: A number of input independent variables (x) and one dependent variable are supplied to the output (y). Through the use of an algorithm, the programme attempts to apply the function corresponding to the given x, with the appropriate y (8). The goal of this type of learning is to build prediction models under conditions of uncertainty. In such a way, it is possible to predict the variable y when a new input is provided. This process is termed "supervised" because the algorithm iteratively performs the prediction (9). If the prediction is not correct, the system is corrected by a supervisor, until the accuracy of the programme is sufficient. The logical steps of a supervised learning method are as follows:

- prepare the data;
- choose the algorithm;
- adapt a model;
- choose a validation method;
- perform evaluations;
- use the model to make predictions.

Unsupervised learning: An unsupervised machine learning method occurs when the input (x) is supplied without correspondence with the outputs (y) (10). In this case, the goal is to find hidden patterns or intrinsic structures that have been provided in the data. Through this process, it is possible to make inferences regarding data without having an exact answer and a supervisor to correct possible errors (11).

Reinforcement learning: Reinforcement learning differs from the supervised method in that there is no supervisor; hence, the model is based on the reward (measured as the evaluation of the achievied performance) (12). Based on this signal, the algorithm changes its own strategy to achieve the best reward. It is also possible to identify two-way learning methods: passive and active. The passive reinforcement learning method uses a pre-determined fixed action. On the other hand, the active method utilizes a complete model (with all possible results) (13,14).

Deep learning: Deep learning (DL) is an approach developed in recent decades to solve such problems as the increasing size of available datasets. Technological and computing progress have facilitated this development through new processing units, which have significantly reduced the time required to train neural networks (15-17). DL allows data to be represented in a hierarchical way on various levels. DL methods are created and learned automatically through the use of advanced learning algorithms. Input information is manipulated to define the concepts at the highest levels through linear transformations on the lower floors (18-20).

Artificial neurons: The simplest neural network is made up of one neuron and is shown in **Fig.1**. A neuron can be interpreted as a computational unit, which takes inputs x_1, x_2, x_3 and produces an output $h_{w,b}(x)$, called the neuron activation. It can also be noted that there is an additional input, which is a constant +1 (21).

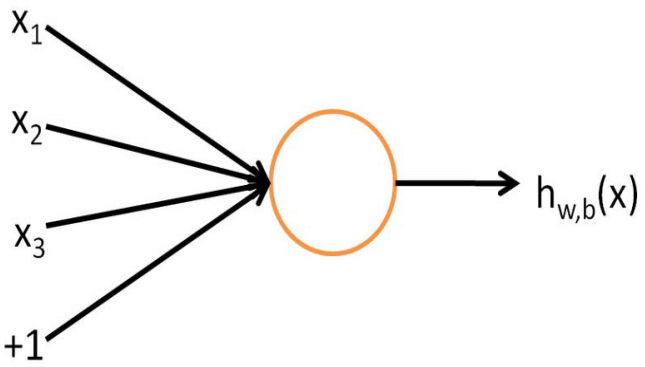

Figure 1. Example of an artificial neuron.

A single neuron receives a numerical input (the weighted sum of different inputs), and it can be activated to process the received value, and its activation can be calculated through a specific function, or it can remain inactive, depending on whether its threshold is exceeded or if it is not activated. Therefore, the neuron will be characterized by a function of activation and an activation threshold. Given a certain function of activation, in a fraction of cases, it may be difficult, if not impossible, to obtain certain output values. Specifically, there may not be a combination of input and weight values for that function that produces the desired output. Therefore, it is necessary to use an additional coefficient b, known as the bias, the purpose of which is to allow the translation of the function activation of the neuron such that it is possible to obtain certain inputs and weight for all desired outputs. We can distinguish two basic types of neural networks: feedforward neural networks and recurrent neural networks (RNN) (also called feedback neural networks) (22). In the former, the connections between neurons never form a cycle and therefore information always travels in one direction. In contrast, recurrent neural networks form a cycle with the creation of an internal state of the network, which enables the performance of dynamic process behaviour over time. The RNN can use its own internal memory to process various types of inputs. It is also possible to distinguish between completely connected networks (fully connected), in which every neuron is connected to all of the others, and stratified networks in which the neurons are organized in layers. In stratified networks, all of the neurons of a layer are connected with all the neurons of the next layer. There are not connections between neurons of one layer, nor are there connections betwee

n neurons of non-adjacent layers. A layered network with three layers is shown in **Fig. 2**.

The left layer in Figure 2 with neurons denoted by blue circles, is generally referred to as an input layer; alternatively, the rightmost layer with a unique orange neuron constitutes the output layer. Finally, the central layer is called hidden layer because there is no connection with the training set.

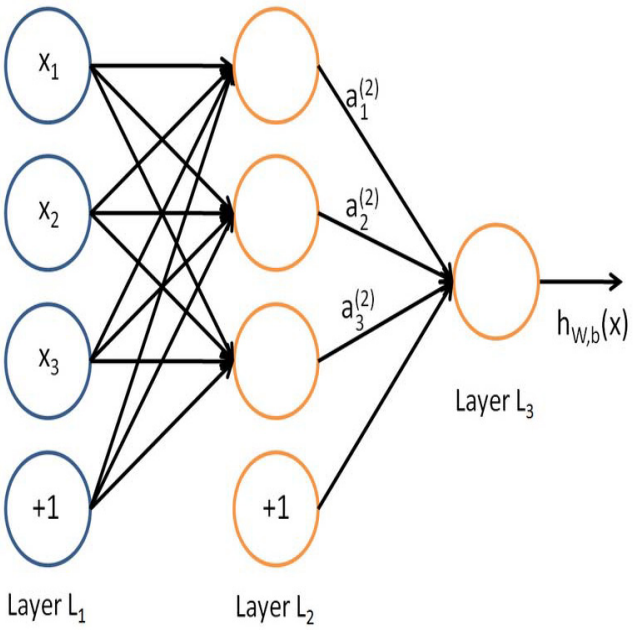

Figure 2. A layered network with three layers.

Training of a neural network: One of the most well-known and effective methods for training neural networks is the so-called error retro-propagation algorithm (error back-propagation), which systematically modifies the weight of the connections between the neurons such that the network's response gets closer to the goal. Training this type of neural network occurs in two different ways: forward propagation and backward propagation. In the first phase, activations of all the neurons of the network are calculated, starting from the first and proceeding to the last layer. During this step, values of the synaptic weights are all fixed (default values) (23). In the second phase, the response of the network or the real output are compared to the desired output, thereby obtaining the network error. The calculated error is propagated in the reverse direction of the first phase. At the end of the second phase, based on the errors, weights are modified to minimize the difference between the current output and the desired output. The whole process is subsequently reiterated, beginning with a forward propagation. In deep learning, all of the layers of the "deep" neurons apply non-linear operations (24,25). Deep networks are characterized by having a number of neurons and layers much greater than the ANNs. Deep networks do not need to work on feature extraction starting from the input data but rather develop a set of criteria during the learning phase to obtain comparable or even superior performance to the classic neural networks. The main types of machine learning tools are briefly described:

- **Decision trees**. This is a technique that makes use of tree graphs (that are equipped with "leaves", which describe states or events associated with a system, and "branches", which represent transitions between states and conditions necessary for such transitions).

- **Bayesian network**. A Bayesian network (BN) is a probabilistic model that represents a set of random variables and their conditional dependences (26,27). Bayesian networks are direct, acyclic graphs whose nodes represent random variables. Therefore, they can be observable quantities, latent variables, unknown parameters or hypotheses. The edges represent conditional dependencies, and nodes, which are not connected, are variables that are conditionally independent of each other. Each node is associated with a probability function that takes as an in input a particular set of variable values from its parental nodes(. The Bayesian Networks that model sequences (for example protein sequences) are called Dynamic Bayesian Networks (28,29). A Bayesian network could be applied to represent probabilistic relationships between diseases and symptoms. Considering the symptoms, the network can be used to calculate the probabilities of the presence of various diseases (30).

- **Hidden Markov model**. A hidden Markov model (HMM) is a statistic model in which the modelled system is a process of Markov (transition between states associated with probability weights) with an un-noticed state (31). An HMM can be considered as the simpler version of the Bayesian network. In a classic Markov model, the state is directly visible to the observer and therefore their transition probabilities are the only parameters. On the other hand, in an HMM the state is not directly visible. The adjective 'hidden' refers to the sequence of states through which the model passes and is not referred to with the parameters.

- **Cluster analysis**. Cluster analysis is a set of multivariate data analyses aimed at the selection and grouping of homogeneous elements in a data set. All clustering techniques are based on the concept of the distance between two elements. The quality of the analysis obtained from clustering algorithms greatly depends on the choice of the metric, and therefore how the distance is calculated. The more common distance functions are the following:

  - ✓ Euclidean distance
  - ✓ Manhattan distance

The Euclidean distance d_2 (x,y) is equal to the square root of the sum of squares of the two coordinate vector differences. The distance of Manhattan is a simple modification of the Euclidean distance. Cluster analysis can use symmetrical or asymmetric distances. Many of the distance functions listed above are symmetrical (the distance between object A and B is equal to the distance from B to A). Clustering techniques can be based mainly on two ways: 1.) from bottom to top (Bottom-Up) and 2.) from top to bottom (Top-Down). In the Bottom-Up method, initially all elements are considered as separate clusters and then the algorithm joins the nearest clusters. The algorithm continues to merge elements to the cluster until a fixed number of clusters is obtained or until the minimum distance between the clusters is achivied. With the Top-Down method, initially

all the elements share a single cluster and then the algorithm begins to divide the cluster into many smaller clusters. The criterion used always tries to obtain homogeneous elements, and proceeds until a predetermined number of clusters is reached.

➢ **Artificial neural network (ANN)**. An artificial neural network or neural network (Neural Network, NN) is a mathematical model based on biological neural networks (32-35). This model consists of information derived from artificial neurons that mimic the properties of living neurons. These mathematical models can be used to solve artificial intelligence engineering problems in different technological fields (e.g., information technology, biology, genomics, proteomics, and metabolomics). In practical terms, neural networks are non-linear statistical structures organized as modelling tools (36,37). The networks can be used to simulate complex relationships between inputs and outputs. An artificial neural network consists of numerous nodes (neurons) connected to each other. There are two types of neurons: input and elaboration neurons. The weight value indicates the synaptic efficacy (ability to increase or decrease its activity) of the input line, and is used to quantify its importance. A very important input variable has a high weight, while a low input has a lower weight. An artificial neural network receives external signals via the input neurons, each of which is connected with numerous internal nodes, organized in additional levels. Each node processes the received signals and transmits the result to the later nodes, and this process continues until the exit level is reached (37-43).

➢ **Deep neural network (DNN)**. The term DNN (deep neural network) refers to deep networks composed of many levels (at least 2 hidden) that are hierarchically organized (44). Hierarchical organization allows for the sharing and reuse of information. The most widely used DNNs consist of a number of levels, between 7 and 50. Deeper networks (100 levels and above) have proven to be able to guarantee slightly better performance, but do so at the expense of efficiency. The number of neurons, connections and weights also characterizes the complexity of a DNN. The greater the number of weights (i.e., parameters to be learned), the higher the complexity of the training (45-50). At the same time, a large number of neurons (and connections) increases the forward and backward propagation, as the number of necessary operations increases. The training of complex models (deep and with many weights and connections) requires high computational power. The availability of Graphics Processing Units (GPUs) with thousands of cores and high internal memory has made it possible to drastically reduce training time (51-61)(**Fig. 3**).

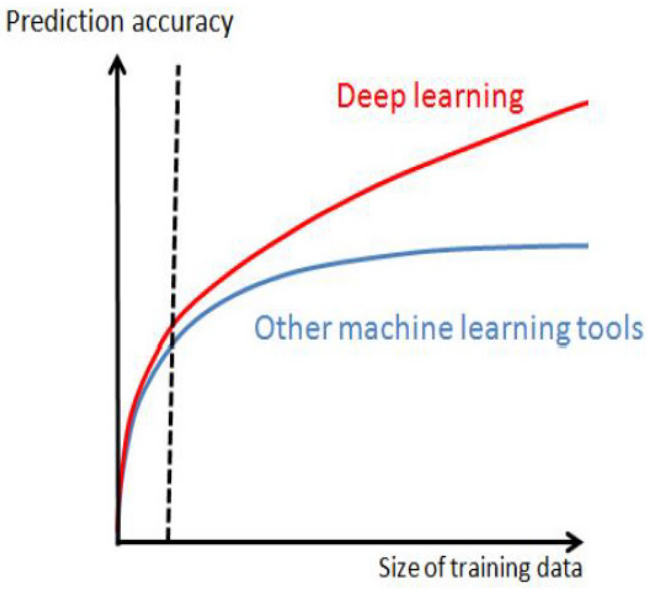

Figure 3. Increase of the prediction accuracy as a function of the data training size.

➢ **Convolutional neural networks (CNN).** Convolutional neural networks (CNN) are a type of artificial network in which neurons are connected to each other by weighted branches (62,63). Therefore, it is always possible to measure the weights and trainable bias. Training of a neural network occurs via forward/backward propagation, and the updating of the weights is also valid in this context. Moreover, a convolutional neural network always uses a single differentiable cost function (a scalar value indicating how good is your model). However, CNN use the specific assumption that the input has a precise data structure and a more efficient forward propagation in order to reduce the amount of parameters of the network. The capacity of a CNN can vary in relation of the number of layers. In addition, a CNN can have multiple layers of the same type. In a convolutional neural network, there are different types of layers, with each having its own specific function. Some of these have trainable parameters (weight and bias), while other layers simply use fixed functions. Usually, a CNN has a series of convolutional layers, the first of these, starting from the input layer and going to the output layer, are used to obtain low-level features, such as horizontal or vertical lines, angles, various contours, etc. (64-66). Features increase with the depth in the network (to the output layer). Generally, the more convolutional layers a network has, the more detailed features it can process (67-69). Compared to a multi-layer perception model (MLP) (a feedforward artificial neural network with non linear activation functions), a CNN method shows substantial changes in the architecture of the convolutional layers (70,71):

✓ **Local processing**. Neurons are only locally connected to the neurons of the previous level, and each neuron then performs local processing. In this manner, there is a strong reduction in the number of connections.

✓ **Shared weights**: Weights are shared in groups, and different neurons of the same level perform the same type of processing on different portions of the input. In this way, there is a strong reduction in the number of weights.

✓ **Spontaneous connectivity**. In a CNN, neurons belonging to different layers of a convolutional layer are connect-

ed to each other through a regular pyramidal architecture. Each neuron of a given layer receives information from a specific number of neurons of the previous layer. Each unit is sensitive only to variations from a specific area of expertise, as shown in **Fig. 4.** This architecture ensures that the patterns learned from the individual units provide strong responses to spatially contained inputs. A sufficient depth of this structure allows local information to be gradually grouped together, leading to the creation of increasingly linear non-linear filters in the layers closer to the output.

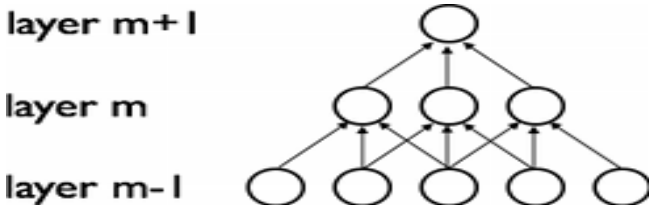

Figure 4. Example of sparse connectivity of neurons between different convolutional layers: the neurons of a layer m are connected only to some of the neurons of the previous layer m-1.

✓ **Weight Sharing**. In a CNN, each layer has the same weights and biases (activation thresholds), and the set of parameters of all similar neurons forms a feature map (see **Fig. 5**), i.e., the set of characteristics common to all neurons present at a certain level of the network. This property is required once the pyramidal architecture described above is set to achieve invariance of the translation network response, i.e., the ability to recognize a target regardless of its position within the scene.

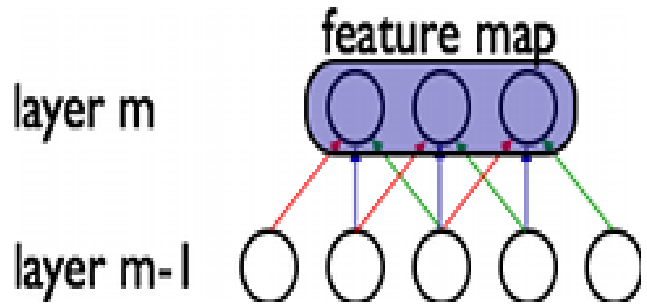

Figure 5. Example of weight sharing (feature map) between different neurons convolutional layers, a link of the same color corresponds to equal weight

✓ **Pooling**. A pooling layer is responsible for aggregating data. Typically, the areas where the pooling is applied are built in such a way as to be partially overlapping in order to preserve local information despite deletions. This operation has two consequences, the first is the reduction of the number of inputs in the next layer, and the second is an increase in the invariance due to the translations introduced by weight sharing. In fact, pooling allows dimensions to be reabsorbed (reducing the computation amount).

✓ **ReLU**. The rectified and linear unit is an activation layer that acts on the entry data in precise terms. The ReLU maintains all the unaltered inputs as positive and multiplies all the negative inputs by a constant (typically 0). It is often used instead of more regular activation functions, such as sigmoid or hyperbolic functions, due to the simplicity of implementation and the nearly negligible computational load introduced by this layer. Since ReLU are inserted between convolution and pooling layers, they help to further decrease the number of calculations performed in the target layers.

Artificial intelligence system applied to NGS studies

In the last ten years, new generation sequencing technologies (NGS) have demonstrated their potential, and with the production of short reads, the throughput process has become increasingly larger NGS techniques, combined with advances in various subjects, ranging from chemistry to bioinformatics, have led to the capability of DNA sequencing at reasonable prices. Specifically, bioinformatics has been fundamental in this development process. Owing to the development of multiple algorithms based on different techniques, such as hash tables, indexes, and spaced-seed, it has been possible to optimize the analysis of increasingly large data sets. NGS technologies are used in a variety of different areas, such as cancer research, human DNA analysis, and animal studies. In **Table 1** are reported several genomic programs concerned with cancer and rare diseases.

Next Generation Sequencing allows for parallel reading of multiple individual DNA fragments, thereby enabling the identification of millions of base pairs in several hours. All of the available NGS platforms have a common technological feature: massive parallel sequencing of clonally amplified DNA molecules or single spatially separated DNA molecules in a flow cell. The main NGS platforms are the following:

- ✓ HiSeq
- ✓ MiSeq
- ✓ Ion Torrent
- ✓ SOLiD
- ✓ Pacbio Rs II and Sequel System.

This strategy represents a radical change with respect to the sequencing method described by Sanger, which is based on the electrophoretic separation of fragments of different lengths obtained by single sequencing reactions (95). Conversely, with NGS technologies, sequencing is performed by repeated cycles of nucleotide extensions by a DNA polymerase or alternatively by iterative cycles of oligonucleotide ligation. Because the procedure is parallel and massive, these platforms allow for the sequencing of hundreds of millions of base pairs (Mb) to billions of base pairs (Gb) of DNA in a single analytical session, depending on the type of NGS technology used. These technologies are versatile because they can be used both for diagnostic and basic and translational research. Traditionally, genomic data analysis is performed with software such as Var Direct, Free Bayes and GATK that are based on statistical analysis (83). Sequencing of genomic sub-regions and gene groups

Table 1. Main genomic programs concerned with cancer and rare diseases

Internet site	Program	Thematic area
www. allofus.nih.gov/	All of Us Research Program	Cancer, rare diseases, complex traits
www.australiangenomics.org.au	Australian Genomics	Cancer, rare diseases, complex traits
www.brcaexchange.org	BRCA Challenge	Cancer, rare diseases, complex traits
www.candig.github.io	Canadian Distributed Infrastructure for Genomics	Cancer, rare diseases, basic biology
www.clinicalgenome.org	Clinical Genome Resource	Rare diseases
www.elixir-europe.org	ELIXIR Beacon	Rare diseases, basic biology
www.ebi.ac.uk	European Genome-Phenome Archive	Rare diseases, basic biology
www.genomicsengland.co.uk	Genomics England	Cancer, rare diseases, complex traits
www.humancellatlas.org/	Human Cell Atlas	Cancer, rare diseases, complex traits, basic biology
www.icgcmed.org	ICGC-ARGO	Cancer
www.matchmakerexchange.org	Matchmaker Exchange	Rare diseases
www.gdc.cancer.gov	National Cancer Institute Genomic Data Commons	Cancer, rare diseases
www.monarchinitiative.org	Monarch Initiative	Rare diseases, complex traits, basic biology
www.nhlbiwgs.org	Trans-Omics for Precision Medicine	Rare diseases, complex traits, basic biology
www.cancervariants.org	Variant Interpretation for Cancer Consortium	Cancer

are currently used to identify polymorphisms and mutations in genes implicated in tumours, and regions of the human genome involved in genetic diseases, through "linkage" and genome-wide association studies (investigation using a panel of genes of different individuals to determine gene variations and associations). These instruments are also used for genetic and molecular analysis in various fields, including the diagnosis of rare genetic diseases, as well as neoplastic and endocrine-metabolic diseases. The field of application of these technologies is expanding and will cover more diagnostic aspects in the future. Several NGS platforms can be used to generate different sources of genomic data:

- ✓ whole genomes;
- ✓ microarrays;
- ✓ RNAseq;
- ✓ capture arrays (exomes) (Illumina TrueSeq Exome Enrichment (62 Mb); Agilent SureSelect (50 Mb);
- ✓ targeted regions;
- ✓ specific genes;
- ✓ chromatin interactions (DNAseq, MNareseq; FAIRE);
- ✓ chip-seq;
- ✓ expression profiles.

Computer programmes that predict genes are becoming increasingly sophisticated. Most software recognizes genes by identifying distinctive patterns in DNA sequences, such as the start and end signals of translation, promoters, and exon-intron splicing junctions (72). However, open reading modules (ORFs) may be difficult to find when genes are short or when they undergo an appreciable amount of RNA splicing with small exons divided by large introns. Prokaryotic genomes are predicted more accurately (sensitivity and specificity > 90%) than eukaryotic genomes. The main features of genomic analysis consist of the following: identification of the gene location and structure, identification of regulatory elements, identification of non-coding RNA genes, prediction of gene function, and prediction of RNA secondary structure. Another approach discovers new genes based on their similarity to known genes. However, this approach is only able to identify new genes when there is an obvious homology with other genes. Eventually, the function of the gene must be confirmed by numerous molecular biology methods. In fact, some genes could be pseudogenes. Pseudogenes have a sequence similar to the normal genes, but usually contain interruptions such as displacements of the reading frame or stop codons in the middle of coding domains. This prevents the pseudogenes from generating a functional product or having a detectable effect on the phenotype. Pseudogenes are present in a wide variety of animals, fungi, plants and bacteria. Predictive methods aim to derive general rules starting from a large number of examples (nucleotide labels within sequences) represented by observations referring to the past, and accumulated in international databases.

Predictive models based on machine learning aim to draw conclusions from a sample of past observations and to transfer

Table 2. Main software tools used for machine learning studies

Internet site	Program name	Thematic area
www.sourceforge.net/p/fingerid	FingerID	Molecular fingerprinting
https://bio.informatik.uni-jena.de/software/sirius/	SIRIUS	Molecular fingerprinting
http://www.metaboanalyst.ca/	Metaboanalyst	Metabolomics analysis
www.knime.com/	KNIME	Machine learning tool
www.cs.waikato.ac.nz/ml/weka/	Weka	Machine learning tool
https://orange.biolab.si/	Orange	Machine learning tool
https://www.tensorflow.org/	TensorFlow	Machine learning tool
http://caffe.berkeleyvision.org/	Caffe	Machine learning tool
http://deeplearning.net/software/theano/	Theano	Machine learning tool
http://torch.ch/	Torch	Machine learning tool

these conclusions to the entire population. Identified patterns can take different forms, such as linear, non-linear, cluster, graph, and tree functions (73-75).

The machine learning workflows are usually organized in four steps:

- ✓ filtering and data pre-processing;
- ✓ feature extraction;
- ✓ model fitting;
- ✓ model evaluation.

Machine learning methods may use supervised or unsupervised systems. The supervised method requires a set of DNA sequences (with all the genetic information including the start and end of the gene, splicing sites, regulatory regions, etc.) for the training step, in order to build the predictive model. This model is then used to find new genes that are similar to the genes of the training set. Supervised methods can only be used if a known training set of sequences is available. Unsupervised methods are used if we are interested in finding the best set of unlabelled sequences that explain the data (76). In **Table 2** are listed the main software tools used for machine learning studies.

Machine learning methodologies have a wide range of application areas:

- ✓ non-coding variants (73,77);
- ✓ identification of protein coding regions and protein-DNA interactions (78,79);
- ✓ identify regulatory regions (e.g., promoters, enhancers, and
- ✓ polyadenylation signals) (80-95);
- ✓ prediction of splice sites (Bayesian classification);
- ✓ identification of functional RNA genes (96);
- ✓ chromatin fragment interactions;
- ✓ histones marks (77);
- ✓ transcriptional factor (TF) binding sites (96);
- ✓ prediction of amino acid sequences and RNA secondary structures (97-106);
- ✓ metabolomics (107-110).

Convolutional neural networks (CNNs) can substantially improve performance in problem sequence solving, compared to previous methods (111). Recurring patterns of sequences in the genome can be efficiently identified by means of CNNs methods (111,112). In this method, the genome sequence is analysed as a 1D window using four channels (A,C,G,T) (113). The protein-DNA interactions are analysed and solved as a two class identification problem (114). Genomic data used in machine learning models should be divided into three proportions training (60%), model testing (30%) and model validation (10%). The main advantages of the CNNs method can be summarized as follows:

- ✓ training on sequence;
- ✓ no feature definition;
- ✓ reduction of number of model parameters
- ✓ use only small regions and share parameters between regions;
- ✓ train wider sequence windows;
- ✓ make *in silico* mutation predictions;

In the CNN method, the following parameters should be optimized:

- ✓ the number of feature map;
- ✓ window size (DNA sequences);
- ✓ kernel size;
- ✓ convolution kernel design;
- ✓ pooling design.

Chen *et al.* (114) used a multi-layer neural model called D-GEX to analyse microarray and RNAseq expression data results. Torracina and Campagne (115) analysed genomic data (variant calling and indel analysis) using deep learning meth-

ods (642 features for each genetic variant) and Popolin *et al.* (116) developed a new software tool called Deep Variant with a precision > 99% (at 90% recall) for SNPs and indel detection. Mutation effects can be predicted using the software Deep Sequence (117), which also uses latent variables (a model using an encoder and decoder network to identify the input sequence).

Recurrent neural networks (RNN) have been proposed to improve the performance of the CNN method. RNN are very useful in the case of ordered sequences (e.g., sequential genomic data). Several applications have recently been reported:

- ✓ base calling (118);
- ✓ non-coding DNA (119);
- ✓ protein prediction (97);
- ✓ clinical medical data (56,120).

Google genomics has recently developed a new software called Deep Variant (Google Brain Team) that uses AI techniques to determine the characteristics of the genome starting from the information of reference sequences. Deep Genomics and WUXI (with offices in Shanghai, Reykiavik and Boston) use the latest generation of AI techniques to study several genetic diseases in an attempt to find possible therapies. AI methods have also been used for metabolomic data analysis including the following:

- ✓ metabolite identification from spectrograms (121,122);
- ✓ metabolite concentration identification using high throughput data (123,124).

Use of CRISPRCas9 for genetic diseases treatments

Attempts to correct the genome for the treatment of genetic diseases have been going on for a long time. Until several years ago, this type of research required long and complex procedures. In 2012, a decisive change came when it was discovered that a protein present in bacteria (Cas9), in association with an RNA sequence, can be used as a device to probe the DNA and identify the point of genetic damage (125). To use Cas9 as a tool of genetic engineering, it is necessary to produce a RNA guide, identify the corresponding DNA segment and delete the gene (125-130). Although this method is highly efficient and rapid compared to the older technologies, the limitation has been its precision. The recognition sequences, i.e., the guides, are small, consisting of approximately twenty nucleotides. In several laboratories around the world, approaches were already underway to make this technology more precise. Recently, Casini *et al.* (130) inserted the protein into specialized yeast cells and later selected the yeasts in which Cas9 made the cut in the most precise way. When tested in human cells, the new method has been shown to reduce mutations by 99%.

Big data analysis

Currently, even the smallest of lab or company can generate an incredibly large volume of data. These data sets are referred to as big data (the term indicates large sets of data, the size of which requires different instruments than those traditionally used) and can be analysed with various techniques, including predictive analysis and data mining. It is important to thoroughly understand what is meant by big data. Since 2011, this term has powerfully entered into common language. For the moment, big data refers to a database that contains data from both structured (from databases) and from unstructured (from new sources as sensors, devices, images) sources, coming from areas internal and external to a facility which can provide valuable information if used and analysed in the correct way. The data can be of two types: structured and unstructured data (131). The former refers to a variety of formats and types that can be acquired from interactions between people and machines, such as all information derived from consumer behaviour on internet sites or users of social networks. Unstructured data, conversely, are usually "text" and consist of information that is not organized or easily interpretable through models. To define the big data more formally, we can use the following four variables (132):

- ✓ Volume: the amount of data has increased in the past years, and it is still increasing significantly. It is extremely easy for a company to store terabytes and petabytes of data.
- ✓ Speed: compared to fifteen years ago, currently data becomes obsolete after a very short period of time. This is one of the reasons why there are no longer long-term plans. New factory leaders should be able to perform a preliminary analysis of the data in order to anticipate trends.
- ✓ Variety: the available data have many formats. Continuous changes of the formats creates many problems in the acquisition and classification of the data.
- ✓ Complexity: being able to manage numerous sources of data is becoming highly difficult.

The analysis of big data is carried out using the following applications:

- ✓ Descriptive Analytics (DA)
- ✓ Prescriptive Analytics (PA)
- ✓ Automated Analytics (AA)

The DA implementation tools aimed at describing the current or past data situation. The PA uses advanced tools that perform data analysis to predict future scenarios. These tools use algorithms that play an active role to find hidden patterns and correlations in the past data, and make prediction models for future use (133-135). PA are advanced tools that, together with data analysis, are able to propose operational/strategic solutions. The AA are tools capable of autonomously implementing proposed actions according to the results. A practical tool that can aid in providing an overview of the benefits that can be derived from a project in the field of big data is the Value Tree. The Value Tree allows tracking of all the benefits that can emerge from a big data project and helps to identify non-pre-

ventable benefits. In the value tree, the benefits are defined as being quantifiable or non-quantifiable, without distinguishing benefits that are directly quantifiable in an economic/financial way. The awareness of the existence of big data have forced several companies to create new effective strategies. Appropriate analytical tools are needed that can translate the data into usable information. However, data mining processes, in graphical or numerical form, large collections of continuous flows of data for the purpose of extracting useful information. The need to anticipate certain situations has become extremely important and can lead to a competitive advantage. Thus, data analysis serves to identify current trends and also to predict future strategies. In particular, predictive models, as well as machine learning and data mining, attempt to identify relationships within the data to identify future trends. Therefore, through the process of automatic learning, the task of the model is to analyse the variables, both individually and together, and provide a predictive score by reducing the margin of error. Predictive Analysis (P.A.) is a science that is already widely used and has been continuously improved in terms of its features and predictions. Persons and companies who use personal and medical data cannot ignore the legal and ethical aspects related to the acquisition and use of such data and information and must comply with current legislation.

Conclusions

Machine learning is the science that enables computers to perform future predictions, and it represents one of the fundamental areas of artificial intelligence. Deep learning is a branch of machine learning and is based on a set of algorithms organized hierarchically with many levels (at least of which are 2 hidden). Generally, these algorithms provide multiple stages of processing (training, model fitting, model evaluation) that often have complex structures and are normally composed of a series of non-linear transformations. Recently, CNN and RNN have been widely used in deep learning, especially for the identification of protein coding regions, protein-DNA interactions, regulatory regions (e.g., promoters, enhancers, and polyadenylation signals), splicing sites and functional RNA gene applications.

References

1. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv 2014: 1409.0473.
2. Hutter F, Hoos HH, Leyton-Brown K. Learning and intelligent optimization. (Berlin: Springer: 2011).
3. Friedman N. Inferring cellular networks using probabilistic graphical models. Science 2004; 303: 799–805.
4. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference and Prediction (Berlin: Springer: 2001).
5. Hamelryck T. Probabilistic models and machine learning in structural bioinformatics. Stat Methods Med Res 2009; 18: 505–526.
6. Zien A. Engineering support vector machine kernels that recognize translation initiation sites. Bioinformatics 2000; 16: 799–807.
7. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv 2015; 1502.03167.
8. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. Pattern Anal Mach Intell IEEE Trans 2013; 35: 1798–1828.
9. Jain V, Murray JF, Roth F, Turaga S, Zhigulin V, Briggman KL, Helmstaedter MN, Denk W, Seung HS. Supervised learning of image restoration with convolutional networks. Int Conf Computer Vision. 2007; 1–8.
10. Day N, Hemmaplardh A, Thurman RE, Stamatoyannopoulos JA, Noble WS. Unsupervised segmentation of continuous genomic data. Bioinformatics 2007; 23: 1424–1426.
11. Hoffman MM. Unsupervised pattern discovery in human chromatin structure through genomic segmentation. Nat Methods 2012; 9: 473–476.
12. Chapelle O, Schölkopf B, Zien A. Semi-supervised Learning (Cambridge Ma: MIT Press: 2006).
13. Ernst J, Kellis M. ChromHMM: automating chromatin-state discovery and characterization. Nat Methods. 2012; 9: 215–216.
14. Chapelle O, Schölkopf B, Zien A. Semi-supervised Learning. (Cambridge MA: MIT Press: 2006).
15. Urbanowicz RJ, Granizo-Mackenzie A, Moore JH. An analysis pipeline with statistical and visualization-guided knowledge discovery for Michigan-style learning classifier systems. IEEE Comput Intell Mag 2012; 7: 35–45.
16. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Josofowicz R, Kaiser L, Kudlur M, Levenberg J. TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXiv 2016; 1603.04467
17. Xiong C, Merity S, Socher R. Dynamic memory networks for visual and textual question answering. arXiv 2016; 1603.01417.
18. Xu R, Wunsch D II., Frank R. Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. IEEE/ACM Trans Comput Biol Bioinformatics 2007; 4: 681–692.
19. Xu Y, Mo T, Feng Q, Zhong P, Lai M, Chang EI. Deep learning of feature representation with multiple instance learning for medical image analysis. IEEE Int Conf Acoustics, Speech, Signal Processing. 2014; 1626–1630.
20. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. (Berlin: Springer: 2014).
21. Ng AY, Jordan MI. Advances in Neural Information Processing Systems. (Cabridge MA: MIT Press: 2002).
22. Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997; 1: 67–82.
23. Boser BE, Guyon IM, Vapnik VN. A Training Algorithm for Optimal Margin Classifiers. (NY: ACM Press: 1992).
24. Noble WS. What is a support vector machine? Nature Biotech 2006; 24: 1565–1567.
25. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. International Conference on Artificial Intelligence and Statistics. 2010; 249–256.
26. Troyanskaya OG, Dolinski K, Owen AB, Altman RB, Botstein DA. Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). Proc Natl Acad Sci USA 2003; 100: 8348–8353.
27. Friedman N, Linial M, Nachman I, Peer D. Using Bayesian networks to analyze expression data. J Comput Biol 2000; 7: 601–620.
28. Koski TJ, Noble J. A review of Bayesian networks and structure learning. Math Applicanda 2012; 40: 51–103.
29. Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian networks to analyze expression data. J Comput Biol 2000; 7: 601–620.
30. Koski TJ, Noble J. A review of bayesian networks and structure learning. Math Applicanda 2012; 40: 51–103.
31. Brown M. Using Dirichlet mixture priors to derive hidden Markov models for protein families. Int Conf Intelligent Systems Mol Biol 1993; 47-55.
32. Keogh E, Mueen A. Encyclopedia of Machine Learning (Berlin: Springer: 2011).
33. Manning CD, Schütze H. Foundations of Statistical Natural Language Processing (Cambridge MA: MIT Press: 1999).
34. Friedman N. Inferring cellular networks using probabilistic graphical models. Science. 2004; 303: 799–805.

35. Hastie T, Tibshirani R.; Friedman, J. The Elements of Statistical Learning: Data mining, Inference and Prediction. (New York NY: Springer: 2001).
36. Yip KY, Cheng C, Gerstein M. Machine learning and genome annotation: a match meant to be? Genome biol 2013; 14:205.
37. Day N, Hemmaplardh A, Thurman RE, Stamatoyannopoulos JA, Noble WS. Unsupervised segmentation of continuous genomic data. Bioinformatics. 2007; 23: 1424–1426.
38. Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. (Pittsburgh, PA: ACM Press: 1992).
39. Noble WS. What is a support vector machine? Nature Biotech 2006; 24: 1565–1567.
40. Hastie T, Tibshirani R, Friedman J, Franklin J. The elements of statistic learning: data mining, inference and prediction. Math Intell 2005; 27: 83–85.
41. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv 2015; 1512.03385.
42. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006; 313: 504–507.
43. Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput 2006; 18: 1527–1554.
44. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015; 521: 436-444.
45. Schmidhuber J. Deep learning in neural networks: An overview. Neural Networks 2015; 61: 85-117.
46. Mamoshina P, Vieira A, Putin E, Zhavoronkov A (2016) Applications of deep learning in biomedicine. Mol Pharm 2016; 13: 1445–1454.
47. Murphy KP (2012) Machine learning: a probabilistic perspective. (Cambridge MA: MIT Press: 2012).
48. Rampasek L, Goldenberg A (2016) TensorFlow: biology's gateway to deep learning? Cell Syst 2016; 2: 12–14.
49. Salakhutdinov R, Hinton G (2012) An efficient learning procedure for deep Boltzmann machines. Neural Comput 2012; 24: 1967–2006.
50. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 2015; 61: 85–117.
51. Snoek J, Larochelle H, Adams RP. Practical bayesian optimization of machine learning algorithms. In Advances in neural information processing systems, pp 2951–2959. (Cambridge MA: MIT Press: 2012).
52. Spencer M, Eickholt J, Cheng J. A deep learning network approach to ab initio protein secondary structure prediction. IEEE/ACM Trans Comput Biol Bioinformatics 2015; 12: 103–112.
53. Eickholt J, Cheng J. Predicting protein residue-residue contacts using deep networks and boosting. Bioinformatics 2012; 28: 3066–3072.
54. Eickholt J, Cheng J. DNdisorder: predicting protein disorder using boosting and deep networks. BMC Bioinformatics 2013; 14: 88.
55. Gawehn E, Hiss JA, Schneider G. Deep learning in drug discovery. Mol Informatics 2016; 35: 3–14.
56. Che Z, Purushotham S, Khemani R, Liu Y. Distilling knowledge from deep networks with applications to healthcare domain. arXiv 2015; 1512.03542.
57. Bastien F, Lamblin P, Pascanu R, Bergstra J, Goodfellow I, Bergeron A, Bouchard N, Warde-Farley D, Bengio Y. Theano: new features and speed improvements. arXiv 2012; 1211.5590
58. Bengio Y. Practical recommendations for gradient-based training of deep architectures. In Neural networks: tricks of the trade, Montavon G, Orr G, Müller K-R (Kelley DR, Snoek J, Rinn J. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Mol Syst Biol. 2016; 12(7): 878.
59. Kingma DP, Welling M. Auto-encoding variational bayes. arXiv 2013; 1312.6114.
60. Kingma D, Ba J. Adam: a method for stochastic optimization. arXiv 2014; 1412.6980.
61. Leung MKK, Xiong HY, Lee LJ, Frey BJ. Deep learning of the tissue-regulated splicing code. Bioinformatics 2014; 30: 121–129.
62. Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv 2013; 1312.6034.
63. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv 2014; 1409.1556.
64. Koh PW, Pierson E, Kundaje A. Denoising genome-wide histone ChIP-seq with convolutional neural networks. Bioinformatics 2017; 33(14): 225–233.
65. Dahl GE, Jaitly N, Salakhutdinov R. Multi-task neural networks for QSAR predictions. arXiv 2014; 1406.1231.
66. Lipton ZC (2015) A critical review of recurrent neural networks for sequence learning. arXiv 2015; 1506.00019.
67. Lipton ZC, Kale DC, Elkan C, Wetzell R (2015) Learning to diagnose with LSTM recurrent neural networks. arXiv 2015; 1511.03677.
68. Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T. Decaf: a deep convolutional activation feature for generic visual recognition. arXiv 2013; 1310.1531.
69. Kraus OZ, Ba LJ, Frey B. Classifying and segmenting microscopy images using convolutional multiple instance learning. arXiv 2015; 1511.05286v1.
70. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015; 521: 436–444.
71. Lee B, Lee T, Na B, Yoon S. DNA-level splice junction prediction using deep recurrent neural networks. arXiv 2015; 1512.05135
72. Park Y, Kellis M (2015) Deep learning for regulatory genomics. Nat Biotechnol 2015;33: 825–826.
73. Libbrecht MW, Noble WS (2015) Machine learning applications in genetics and genomics. Nat Rev Genet 2015; 16: 321–332.
74. Sutskever I, Vinyals O, Le QV. Advances in neural information processing systems. (Cambridge MA: MIT Press: 2014).
75. Wasson T, Hartemink AJ. An ensemble model of competitive multi-factor binding of the genome. Genome Res 2009; 19: 2102–2112.
76. Yip KY, Cheng C, Gerstein M. Machine learning and genome annotation: a match meant to be? Genome Biol 2013; 14: 205.
77. Zhou J, Troyanskaya OG (2015) Predicting effects of noncoding variants with deep learning based sequence model. Nat Methods 2015; 12: 931–934.
78. Swan AL, Mobasheri A, Allaway D, Liddell S, Bacardit J (2013) Application of machine learning to proteomics data: classification and biomarker identification in postgenomics biology. Omics 2013; 17: 595–610.
79. Alipanahi B, Delong A, Weirauch MT, Frey BJ (2015) Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol 2015; 33: 831–838.
80. Zhang J, White NM, Schmidt HK. Integrate: gene fusion discovery using whole genome and transcriptome data. Genome Res 2016; 26(1):108–118.
81. Degroeve S, Baets BD, de Peer YV, Rouz P. Feature subset selection for splice site prediction. Bioinformatics. 2002; 18: S75–S83.
82. Wasson, T., Hartemink, A. J. An ensemble model of competitive multi-factor binding of the genome. Genome Res 2009;19: 2102–2112.
83. Lanckriet GRG, Bie TD, Cristianini N, Jordan MI, Noble WS. A statistical framework for genomic data fusion. Bioinformatics 2004; 20: 2626–2635.
84. Pavlidis P, Weston J, Cai J, Noble WS. Learning gene functional classifications from multiple data types. J Computat Biol 2002; 9: 401–411.
85. Picardi E, Pesole G. Computational methods for ab initio and comparative gene finding. Meth Mol Biol 2010; 609: 269–284.
86. Degroeve S, Baets BD, de Peer YV, Rouzé P. Feature subset selection for splice site prediction. Bioinformatics 2002; 18: S75–S83.
87. Ouyang Z, Zhou Q, Wong HW. ChIP-Seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells. PNAS USa. 2009; 106: 21521–21526.
88. Chen Y, Li Y, Narayan R, Subramanian A, Xie X. Gene expression inference with deep learning Bioinformatics 2016; 32: 1832–1839.
89. Troyanskaya OG, Dolinski K, Owen AB, Altman RB, Botstein D. A Bayesian framework for combining heterogeneous data sources for gene function prediction (in S. cerevisiae). PNAS USA 2003; 100: 8348–8353.
90. Upstill-Goddard R, Eccles D, Fliege J, Collins A. Machine learning approaches for the discovery of gene–gene interactions in disease data. Brief Bioinform 2013; 14: 251–260.

91. Urbanowicz R, Granizo-Mackenzie D, Moore J. An expert knowledge guided michigan-style learning classifier system for the detection and modeling of epistasis and genetic heterogeneity. Proc Parallel Problem Solving From Nature 2012; 12: 266–275.
92. Angermueller C, Lee H, Reik W, Stegle O. Accurate prediction of single-cell DNA methylation states using deep learning. Genome Biol 2017; 18: 67.
93. Ernst J, Kellis M. ChromHMM: automating chromatin-state discovery and characterization. Nature Methods 2012;9: 215–216 (2012).
94. Fraser AG, Marcotte EM. A probabilistic view of gene function. Nature Genet 2004; 36: 559–564.
95. Battle A, Khan Z, Wang SH, Mitrano A, Ford MJ, Pritchard JK, Gilad Y (2015) Genomic variation. Impact of regulatory variation from RNA to protein. Science 2015; 347: 664–667.
96. Kelley DR , Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res 2016; 26: 990-99.
97. Sønderby SK, Winther O. Protein secondary structure prediction with long short term memory networks. arXiv 2014; 1412.78.
98. Beer MA, Tavazoie S. Predicting gene expression from sequence. Cell 2004; 117: 185–198. Heintzman N. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. Nature Genet 2007; 39: 311–318.
99. Pique-Regi R. Accurate inference of transcription factor binding from DNA sequence and chromatin accessibility data. Genome Res. 2011;21: 447–455.
100. Qiu J, Noble WS. Predicting co-complexed protein pairs from heterogeneous data. PLoS Comput Biol 2008; 4: e1000054.
101. Ramaswamy S. Multiclass cancer diagnosis using tumor gene expression signatures. Proc Natl Acad Sci USA 2001; 98: 15149–15154.
102. Saigo H, Vert JP, Akutsu T. Optimizing amino acid substitution matrices with a local alignment kernel. BMC Bioinformatics 2006; 7: 246.
103. Segal E. A genomic code for nucleosome positioning. Nature 2006;44, 772–778.
104. Karlic RR, Chung H, Lasserre J, Vlahovicek K, Vingron M. Histone modification levels are predictive for gene expression. PNAS USA 2010; 107: 2926–2931.
105. Bell JT, Pai AA, Pickrell JK, Gaffney DJ, Pique-Regi R, Degner JF, Gilad Y, Pritchard JK (2011) DNA methylation patterns associate with genetic and gene expression variation in HapMap cell lines. Genome Biol 2011; 12: R10.
106. Cuellar-Partida G, et al. Epigenetic priors for identifying active transcription factor binding sites. Bioinformatics 2011; 28: 56–62.
107. Kell DB (2005) Metabolomics, machine learning and modelling: towards an understanding of the language of cells. Biochem Soc Trans 2005; 33: 520–524.
108. Shen H, Zamboni N, Heinonen M, Rousu J. Metabolite identification through machine learning—Tackling CASMI challenge using fingerID. Metabolites 2013; 3: 484–505.
109. Glaab E, Bacardit J, Garibaldi JM, Krasnogor N. Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data. Plos one. 2012; 7: e39932.
110. Menden MP, Iorio F, Garnett M, McDermott U, Benes CH, Ballester PJ, Saez-Rodriguez J. Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties. PLos one 2013; 8: e61318.
111. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems Lake Tahoe, Nevada 2012: 1097-1105.
112. Lanchantin J, Lin Z, Qi Y. Deep motif: Visualizing genomic sequence classifications. arXiv 2016: 1605.01133.
113. Zeng H, Edwards MD, Liu G, Gifford DK. Convolutional neural network architectures for predicting DNA–protein binding. Bioinformatics 2016; 32(12): 121–127.
114. Chen J, Guo M, Wang X, Liu B. A comprehensive review and comparison of different computational methods for protein remote homology detection. Briefings in bioinformatics 2016; 108:256.
115. Torracinta R, Campagne F. Training genotype callers with neural networks. bioRxiv 2016; 097469.
116. Poplin R, Newburger D, Dijamco J, Nguyen N, Loy D, Gross SS, McLean CY, DePristo MA. Creating a universal SNP and small indel variant caller with deep neural networks. 2018; bioRxiv: doi. org/10.1101/092890.
117. Schreiber J, Libbrecht M, Bilmes J, Noble W. Nucleotide sequence and dnasei sensitivity are predictive of 3d chromatin architecture. bioRxiv; 2017: 103614.
118. Boza V, Brejova B, Vinar T. Deepnano: Deep recurrent neural networks for base calling in minion nanopore reads. Plos one 2017;12(6): e0178751.
119. Quang D , Xie X . Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. Nucleic Acids Res 2016; 44(11): e107–e107. X.
120. Lee T , Yoon S . Boosted categorical restricted boltzmann machine for computational prediction of splice junctions. Int Conf Machine Learning; 2015: 2483–2492.
121. Baumgartner C, Böhm C, Baumgartner D. Modelling of classification rules on metabolic patterns including machine learning and expert knowledge. J Biomed Inform 2005; 38: 89–98.
122. Alakwaa FM, Chaudhary K, Garmire LX. Deep learning accurately predicts estrogen receptor status in breast cancer metabolomics data. J Proteom Res 2018; 17: 337–347.
123. Hao J, Astle W, De Iorio M, Ebbels T. BATMAN—An R package for the automated quantification ofmetabolites from NMR spectra using a Bayesian model. Bioinformatics 2012; 28: 2088–2090.
124. Ravanbakhsh S, Liu P, Bjorndahl TC, Mandal R, Grant JR, Wilson M, Eisner R, Sinelnikov I, Hu X, Luchinat C. Accurate, fully-automated NMR spectral profiling for metabolomics. PLos one 2015; 10: e0124219.
125. Hsu PD, Lander ES, Zhang F. Development and Applications of CRISPR-Cas9 for Genome Engineering. Cell 2014; 157: 1262.
126. Sternberg S, Doudna J. Expanding the Biologist's Toolkit with CRISPR-Cas9.Molecular Cell. 2015; 58: 568.
127. Tsai SQ, Zheng Z, Nguyen NT, Liebers M, Topkar VV, Thapar V, Wyvekens N, Khayter C, Iafrate AJ, Le LP, Aryee MJ, Joung JK. GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR-Cas nucleases. Nat Biotechnol 2015; 33(2): 187.
128. Slaymaker IM et al. Rationally engineered Cas9 nucleases with improved specificity. Science 2016; 351: 84–88.
129. Kim S, Kim D, Cho SW, Kim J, Kim JS. Highly efficient RNA-guided genome editing in human cells via delivery of purified Cas9 ribonucleoproteins. Genome Res 2014; 24 :1012–1019.
130. Casini A, Olivieri M, Petris G, Montagna C, Reginato G, Maule G, Lorenzin F, Prandi D, Romanel A, Demichelis F, Inga A, Cereseto A. A highly specific SpCas9 variant is identified by in vivo screening in yeast. Nature Biotech 2018; 36: 265–271.
131. Wilson H, Elizabeth D, McDonald M. (2002). Factors for success in customer relationship management (CRM) systems. J Marketing Manage 2002; 18(1): 193–219.
132. Costa FF. Big data in genomics: challenges and solutions. GIT Lab J 2012; 11: 1-4.
133. Ward RM, Schmieder R, Highnam G, Mittelman D. Big data challenges andopportunities in high-throughput sequencing. Syst Biomed 2013; 1: 29- 34.
134. Eisenstein M. Big data: The power of petabytes. Nature 2015; 527: S2-S4.
135. Woodco Bacardit J, Llorà X. Large-scale data mining using genetics-based machine learning. Wiley Interdiscip Rev 2013; 3: 37–61.