CANCER GENOMICS & PROTEOMICS *18*: 605-626 (2021)
doi:10.21873/cgp.20284

Review

Machine Learning Approaches on High Throughput NGS Data to Unveil Mechanisms of Function in Biology and Disease

VASILEIOS C. PEZOULAS[1*], ORSALIA HAZAPIS[2*], NEFELI LAGOPATI[2,3*], THEMIS P. EXARCHOS[4,5], ANDREAS V. GOULES[6], ATHANASIOS G. TZIOUFAS[6], DIMITRIOS I. FOTIADIS[1], IOANNIS G. STRATIS[7], ATHANASIOS N. YANNACOPOULOS[8] and VASSILIS G. GORGOULIS[2,3,9,10,11]

[1]*Unit of Medical Technology and Intelligent Information Systems, University of Ioannina, Ioannina, Greece;*
[2]*Molecular Carcinogenesis Group, Department of Histology and Embryology, School of Medicine, National and Kapodistrian University of Athens, Athens, Greece;*
[3]*Biomedical Research Foundation of the Academy of Athens, Athens, Greece;*
[4]*Unit of Medical Technology and Intelligent Information Systems, Department of Materials Science and Engineering, University of Ioannina, Ioannina, Greece;*
[5]*Department of Informatics, Ionian University, Corfu, Greece;*
[6]*Department of Pathophysiology, School of Medicine, National and Kapodistrian University of Athens, Athens, Greece;*
[7]*Department of Mathematics, National and Kapodistrian University of Athens, Athens, Greece;*
[8]*Department of Statistics, and Stochastic Modelling and Applications Laboratory, Athens University of Economics and Business (AUEB), Athens, Greece;*
[9]*Division of Cancer Sciences, Faculty of Biology, Medicine and Health, Manchester Academic Health Science Centre, Manchester Cancer Research Centre, NIHR Manchester Biomedical Research Centre, University of Manchester, Manchester, U.K.;*
[10]*Center for New Biotechnologies and Precision Medicine, Medical School, National and Kapodistrian University of Athens, Athens, Greece;*
[11]*Faculty of Health and Medical Sciences, University of Surrey, Surrey, U.K.*

Abstract. *In this review, the fundamental basis of machine learning (ML) and data mining (DM) are summarized together with the techniques for distilling knowledge from state-of-the-art omics experiments. This includes an introduction to the basic mathematical principles of unsupervised/supervised learning methods, dimensionality reduction techniques, deep neural networks architectures and the applications of these in bioinformatics. Several case studies under evaluation mainly involve next generation sequencing (NGS) experiments, like deciphering gene expression from total and single cell (scRNA-seq) analysis; for the latter, a description of all recent artificial intelligence (AI) methods for the investigation of cell sub-types, biomarkers and imputation techniques are described. Other areas of interest where various ML schemes have been investigated are for providing information regarding transcription factors (TF) binding sites, chromatin organization patterns and RNA binding proteins (RBPs), while analyses on RNA sequence and structure as well as 3D dimensional protein structure predictions with the use of ML are described. Furthermore, we summarize the recent methods of using ML in clinical oncology, when taking into consideration the current omics data with pharmacogenomics to determine personalized*

*These Authors contributed equally to this study.

Correspondence to: Vassilis G. Gorgoulis, Molecular Carcinogenesis Group, Department of Histology and Embryology, School of Medicine, National and Kapodistrian University of Athens, 75 Mikras Asias Str., 11527, Goudi, Athens, Greece. Tel: +30 2107462352, email: vgorg@med.uoa.gr and Athanasios N. Yannacopoulos, Department of Statistics, and Stochastic Modelling and Applications Laboratory, Athens University of Economics and Business (AUEB), 76 Patision Str., 104 34, Athens, Greece. Tel: +30 2108203801, e-mail: ayannaco@aueb.gr

Key Words: Machine learning, supervised-unsupervised learning, NGS, gene expression, scRNA-seq, TFs, RBPs, RNA structure, sequence motifs, review.

treatments. With this review we wish to provide the scientific community with a thorough investigation of main novel ML applications which take into consideration the latest achievements in genomics, thus, unraveling the fundamental mechanisms of biology towards the understanding and cure of diseases.

The majority of the large-scale data in bioinformatics and systems biology include genome wide studies from next generation sequencing (NGS) experiments, such as, studies for deciphering gene expression from total and single cell (scRNA-seq), as well as data that provide information regarding the binding sites of transcription factors (TFs) and RNA binding proteins (RBPs) while incorporating information of the RNA substrate such as, sequence and RNA structure. NGS technology enables the decoding the genome of many organisms, learning the transcriptome and proteome per cell or deciphering differences from genome-wide association studies (GWAS) (1) between different organisms and clarifying the functions and properties of many biological systems. These topics of bioinformatics include a large repertoire of datasets. To analyze and interpret the big biomedical data, efficient algorithms are constantly being developed for processing, building, and matching the genomes (2) or determining the gene expression differences under normal or disease conditions (3). The constant evolution of ML will aid biologists to find patterns and associations in various studies while also enabling them to predict the outcome of biomodels under investigation, uncovering the fundamental mechanisms in biology.

A General Overview of Machine Learning (the Basic Principles)

The general scope of machine learning ML is to devise algorithms that can run in an automated fashion to predict new behavior or classify patterns arising in complex data sets, based on sets of training data. ML consists of a large variety of methods designed to address a wide class of different problems, so in this section we refrain to a selection of methods and problems.

Classification is one of the main tasks in ML with many important applications in a variety of disciplines including medicine. The problem of classification can be abstracted as follows: Given points in a high dimensional space, corresponding to different entities (of different quality) can you separate these points into distinct groups each being homogeneous and comprising of points of the same quality? We will illustrate this analysis with a real example for defining and classifying the distribution of gene expression from RNA-seq experiments and their response upon treatment in different time points such as the distribution shown in Figure 1.

Each distribution can be modeled as an array of n real numbers, say $x=(x_1,\ldots,x_n)$, each considered as a point in the Euclidean space R^n of suitable dimension n. Then, our data can be conceived as collections of points in R^n. The rationale of such a "visualization" is to manage those points related to qualitatively different gene expressions located in distant parts of the underlying Euclidean space where the points are embedded, therefore obtaining different clusters.

One possible separation scheme between the qualitatively different clusters can be in terms of a separating hyperplane, *i.e.*, a subset of R^n defined as the set of points $x=(x_1,\ldots,x_n)$ such that $H=\{x\in R^n | w\bullet x+w_0=0\}$ for a suitable vector $w=(w_1,\ldots,w_n)$ and scalar w_0 with $w \bullet x$ denoting the Euclidean inner product defined by $w\bullet x=\sum_{i=1}^{n} w_i x_i$. This set is called hyperplane since in 2- or 3- dimensions H corresponds to a straight line or a two-dimensional (2D) plane (respectively) separating R^2 or R^3 in two distinct sub-spaces.

The same principle is applied in any dimension n. The subspace H corresponds to an $n-1$ dimensional hyperplane which separates R^n into two distinct parts $H+=\{x\in R^n | w\bullet x+ w_0>0\}$ and $H-=\{x\in R^n | w\bullet x+ w_0<0\}$. Our goal is to find if a separating hyperplane H exists such that all points of class 1, satisfy the condition $x_i\in H_+$ while all points of class 2, *i.e.*, points x_i with $i\in N_2$, satisfy the condition $x_i\in H_-$. If we manage to identify such a hyperplane, then we have a separating rule, that is a function f: $R^l \rightarrow R$, defined by $f(x)=f(x_1,\ldots,x_l):=w\bullet x+w_o$ such that for all points $x_i=(x_{i1},\ldots,x_{il})$, with $i\in N_1$ it holds that $f(x_i)>0$, whereas for all points $x_i=(x_{i1},\ldots,x_{il})$, with $i\in N_2$ it holds that $f(x_i)<0$. The construction of such a function is in the very essence of ML where we will demonstrate the basic supervised and unsupervised learning techniques for solving such task.

Supervised Machine Learning Principles

A supervised machine learning approach infers to a function derived from labeled training data or training set and a desired output value. The algorithm after learning will proceed in trying to correctly determine the class labels for unseen instances with the minimum error.

Support vector machines (SVM). In our previous examples for classifying gene expression data (Figure 1), a possible criterion for assessing the "fitness" of a separating hyperplane in classifying the data (equiv. points) is the distance that such a criterion yields for the most difficult data to be separated, *i.e.*, those data (equiv. points) on the boundaries of the geometric loci of the points in $x_i\in N_1$ and the points in $x_i\in N_2$. Clearly, under this interpretation the optimal hyperplane H would be the one for which the distance between classified points on the boundaries is maximum. In such cases, an optimal hyperplane H, would be the one for which the misspecification error is minimum.

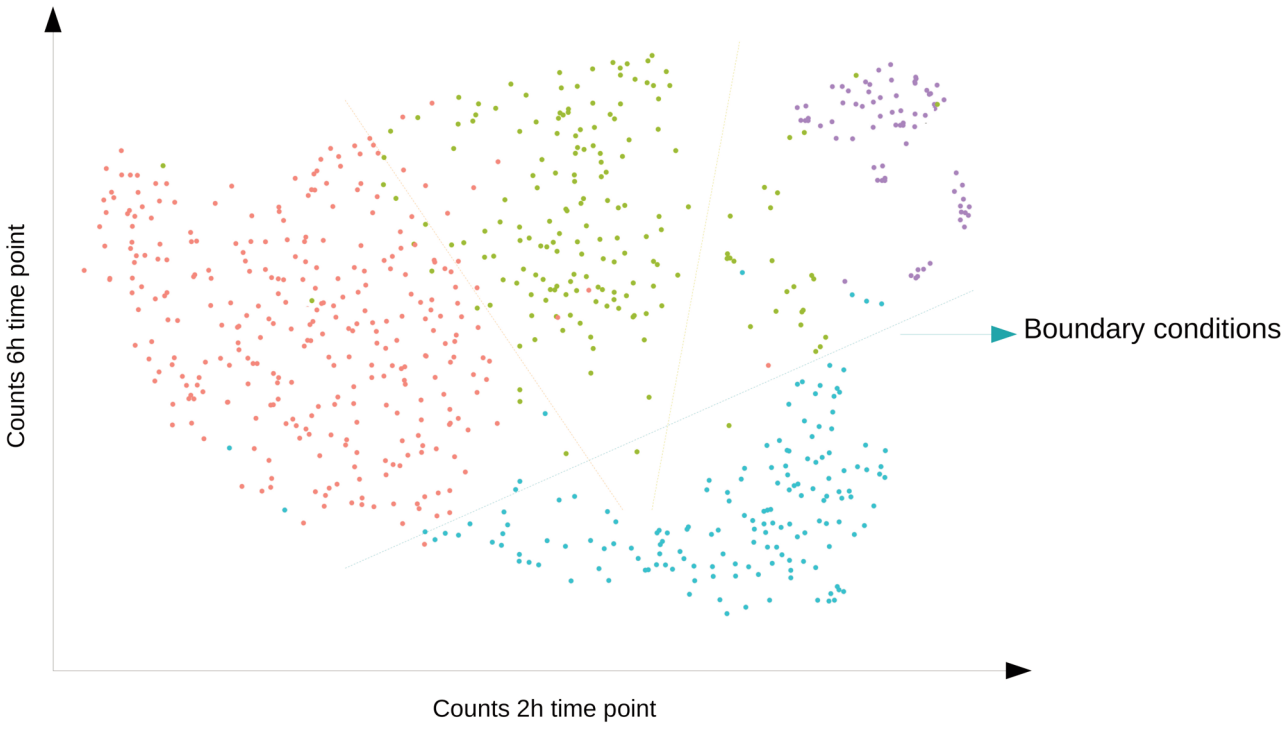

Figure 1. *Read count distribution from RNA seq experiments, where each dot represents a gene classified according to its response in a specific treatment.*

The distance of a point x_i from the hyperplane H can be calculated in terms of the vector **w** and the scalar w_o as shown in Equation 1 (Eq. 1).

$$d(x_i,\mathrm{H}) = \frac{1}{\|w\|}\left|w \cdot x_i + \mathrm{w}_o\right|, \quad [1]$$

where ||w|| denotes the Euclidean norm of the vector **w**, defined by ||w||:=w•w. Let $\delta=\delta(H)=\delta(w,w_0)$ be the minimum distance between the points in class 1 and 2 (Eq. 2).

Let us consider the problem of choosing the hyperplane H which maximizes the distance δ, $\max\delta_{\{\mathrm{w},\mathrm{w}0\}}(\mathrm{w},\mathrm{w}_0)$, subject to the separation conditions as determined *via* Eq. 2 and 3.

$$d(x,\mathrm{H}) = \frac{1}{\|w\|}(w \cdot x_i + \mathrm{w}_o),\ \mathrm{i} \in N_1 \quad [2]$$

$$d(x_j,\mathrm{H}) = \frac{-1}{\|w\|}\left(w \cdot x_j + \mathrm{w}_o\right),\ \mathrm{j} \in N_2 \quad [3]$$

For a fixed ε>0, we can get the best separation result if we choose the separation hyperplane H such that: $w\bullet x_i+w_o>\varepsilon\|w\|$, i∈$N_1$, $w\bullet x_j+w_o<-\varepsilon\|w\|$, j∈$N_2$ where the term on the left-hand sides correspond to the distance of the points from H. Defining the new variables y_i so that y_i=1 if $i\in N_1$ and y_i=-1 if i∈N_2, and considering the problem of maximizing the distance (or equivalently maximizing ε/||w|| and setting without loss of generality ε=1) corresponds to the more standard convex optimization problem (Eq. 4):

$$\min_{\mathrm{w},\mathrm{w}_o} \frac{1}{2}\|w\|^2, \quad [4]$$

such that

$$y_i(w \cdot x_i + \mathrm{w}_o) > 1, \quad \mathrm{i} \in N = N_1 \cup N_2. \quad [5]$$

The classification function may be readressed as $f(x)=sign(w\bullet x+w_o)$, with the classification assignment for any data point x_i given in terms of the response $y_i=f(x_i)$. A geometric viewpoint of the above optimization problem is as trying to fit an empty slab of the maximum possible width between the two data classes. Convex optimization problems

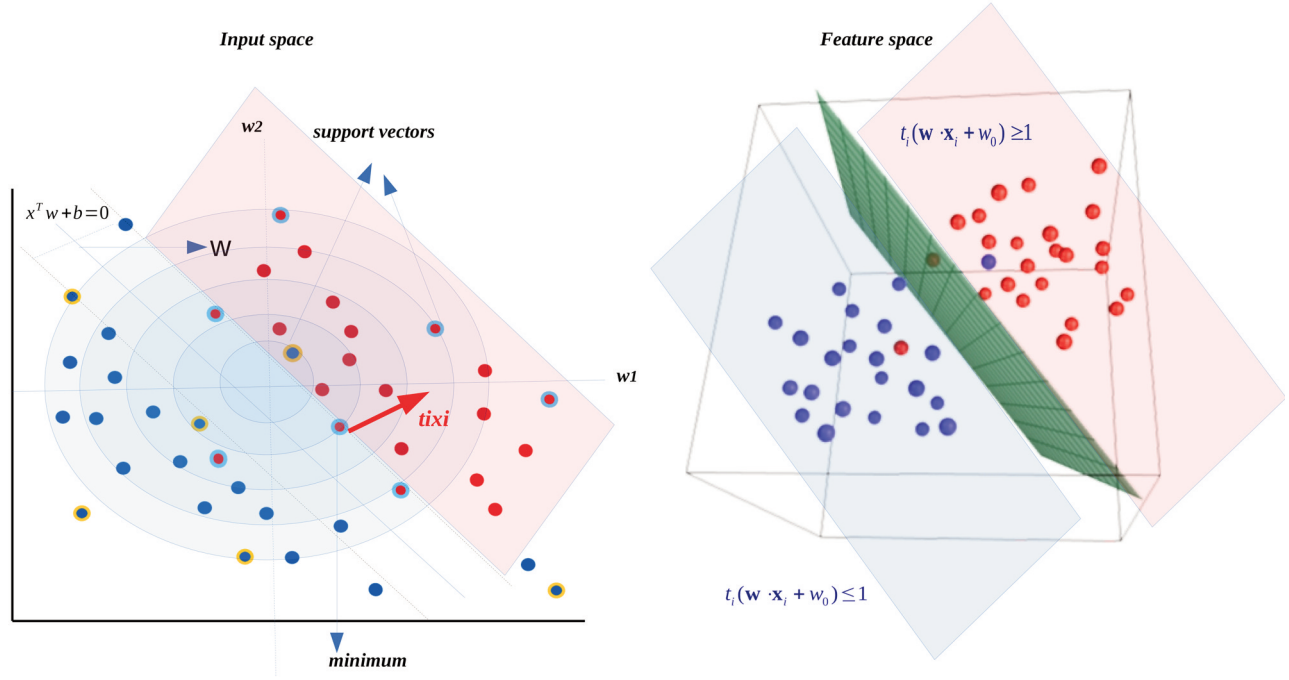

Figure 2. *Description of the boundary conditions for optimal classification. Red and blue dots represent experimental measurements classified in 2 classes. The boundary conditions can be seen as the circled measurements close to a border where in between the 2 classes a hyperplane can be fitted to describe the border based on the equation ti(w*xi+wo)><=1. The classification of the 2 classes happens via adding weights to each point with the maximum goal of making the classes as separable as possible.*

enjoy a long tradition and a powerful arsenal of analytic and numerical techniques that can be used for their treatment (4). One of these is their Lagrangian formulation, intimately related to the powerful and deep dual formulation of such problems. According to this viewpoint, a solution of the above problem can be understood as a saddle point (min-max) of the augmented Lagrangian function defined by Eq. (6):

$$L\left(w_1,\ldots,w_l,w_o,\lambda_1,\ldots\lambda_N\right) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N}\lambda_i\left[y_i(w\cdot x_i+w_o-1)\right] \quad [6]$$

where $\lambda_i \geq 0$, are the generalized Lagrange multipliers. The saddle point conditions for L are the celebrated KKT conditions (5) according to which **w** and λ satisfy the conditions.

$$\frac{\partial}{\partial w_i}L = 0, i = 0,1,\cdots,n,$$

$$\frac{\partial}{\partial \lambda_i}L = 0, i = 1,\cdots,N,$$

$$\lambda_i y_i(w\cdot x_i+w_o-1) = 0, i = 1,\cdots,N \quad [7]$$

The first condition in Eq. (7) can help us identify a linear combination of the support vectors, depending on the Lagrange multipliers, hence the terminology SVM. An interesting alternative formulation of the problem, introduces the concept of the loss function which is fundamental in ML and is the following: Upon defining M as the set of misclassified data points, one possible indicator of misclassification error may be the loss function: $L(H)=L(w,w_0)=-\sum_{\{i\in M\}}y_i\,(w\bullet x_i+w_o)$. The gradient of the loss function is defined as the direction of $n=(\partial L/(\partial w_1),\ldots,\partial L/(\partial w_l),\partial L/(\partial w_o))$, which can be normalized by dividing with ||n|| (if needed) and a typical gradient scheme would be to start at an initial point w_o to $w^k=(w_1^k,\ldots,w_1^k,w_0^k)$ and then create a sequence $w^{(k+1)}=w^k-\varrho n$, k=0,1..n for a scalar ϱ>0, often called the learning rate. If this scheme converges [which does under conditions, as described in (4)] then it converges to a point where the gradient of the loss function vanishes, which is a candidate for a local minimum (and in fact a global minimum if the loss function is convex). Figure 2 best describes the boundary conditions as obtained *via* this condition for having the minimum optimal solution. Figure 3 represents the maximal margin hyperplane in the 2D space.

Often linear separation is not feasible on account of the geometry of the data. In this case, a kernel classifier techniques may prove useful. In such techniques the original feature space X is mapped through a nonlinear mapping *Φ*, into a higher dimensional space Z, where the data may display linear

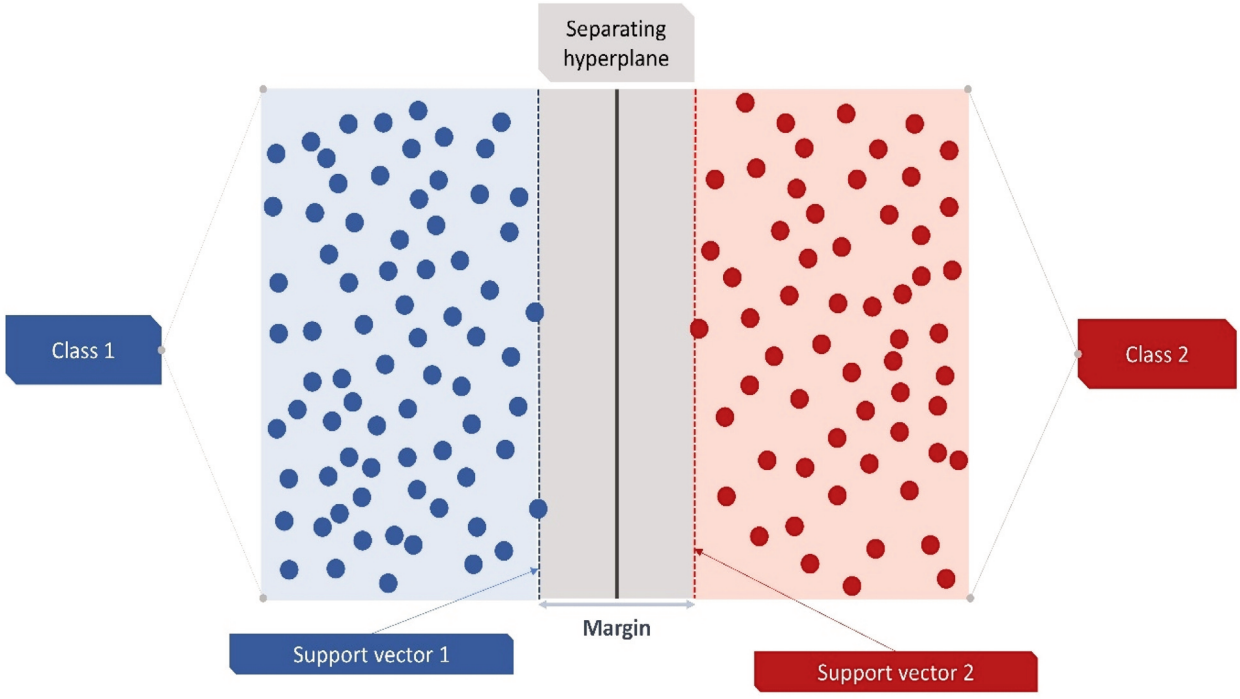

Figure 3. *An illustration of the maximal margin hyperplane in the 2D space. As demonstrated in Figure 1 and illustrated here, the maximum goal of any classification regime is to find the optimal solution which will increase the margin between 2 classes.*

structure and hence linear separation techniques may be used. Convenient feature maps are those related with the so-called kernel functions $K:X\times X\rightarrow R$, in terms of the relation $K(x,x')=<\Phi(x),\Phi(x')>_Z$, where by $<.,.>_Z$ we denote the inner product in the feature space Z. Kernel functions are symmetric functions which quantify similarity between any two data points in the input space x,x', in terms of the value $K(x,x')$. Various choices for kernel function are possible, for example polynomial kernels of degree d are defined as: $K(x,x')=(x\bullet x'+b)^d$, where x and x' are observations vectors, b is a non-negative constant which is used to balance among higher and lower degree polynomial coefficients, and d is the non-negative degree of the polynomial. A d-degree polynomial kernel is used to create decision boundaries of that degree. Other widely adopted kernels are the Gaussian RBF kernels and the radial kernels (6). The nature of SVM classifier which uses multiple features to learn and drive a classification has led to a wide range of applications in biology. This can include image analysis approaches for predicting stages of tumor such as in (7) to more combinatorial methods of combining genomic data sets from gene expression, DNA methylation, GWAs studies and MRI images to drive clinical outcomes and treatments in oncology.

Regression. Regression aims to determine the relationship between a set of input features $x_i\in R^n$ and a continuous or otherwise quantitative outcome variable $y_i\in R^m$, in terms of a model of the form $y_i=f(x_i)+\in i$, where $f:R^n\rightarrow R^m$, is a suitable function to be determined and $\in_i$ are (random) errors modeling fluctuations around the deterministic law $y_i=f(x_i)$. In ML, regression techniques are used to first fit and construct an appropriate function f, that models a set of training input data and corresponding responses, say $(x_i,y_i),i=1,...,N_{train}$, along with a distribution of the errors.

Linear regression (8) is a widely adopted supervised learning technique. Linear regression uses a linear function f, (in the case m=1) of the form $f(x)=b_0+b\bullet x$ for a suitable scalar b_0 and vector $b=(b_1,...,b_n)\in R^n$. By appropriately specifying the parameters of the model from test data, the model can be used to provide predictions for new data. The estimation of these parameter is obtained by minimizing an appropriate loss function. A large variety of models is built around linear regression depending on how the model parameters are determined. A common choice comprising four important models is to choose b so that the following loss function is minimized (Eq. 8):

$$C(b)=\frac{1}{n}\sum_{i=1}^{n}(b\cdot x_i-y_i)^2+ra\sum_{i=1}^{k}|b_i|+a\frac{1-r}{2}\sum_{i=1}^{k}{b_i}^2 \quad [8]$$

where the first term corresponds to the mean square error of the model which focuses on the closeness of the model to the observations, and the rest correspond to regularization terms. The above model when a=0 corresponds to the standard linear regression model, which may give rise to issues related to over-fitting, multicolinearity or high dimensional data that might lead to models which are difficult to interpret. The case where r=0, $\alpha \neq 0$ leads to Ridge Regression, also known as Tikhonov regularization (9), a variation of linear regression which attempts to solve the multi-collinearity problems of the latter approach. The case where r=1, $\alpha \neq 0$ corresponds to least absolute shrinkage and selection operator (Lasso) regression (10) that uses the most important features of the dataset, while the rest are set to zero and consequently are ignored, thus leading to more economical and easier to interpret models. The last variation discussed here is the elastic nets (11) corresponding to $r \in (0,1)$, $\alpha \neq 0$ and can be considered as a combination of the previous two.

Logistic regression (12) estimates the probability that an input belongs to the target class (labeled as '1', called the positive class) or to the other one (labeled as '-1', the negative class –often labeled as 0). The basis for this approach is the logistic function, which connects the probability of assignment to each class with the features vector x connected to each data point, as in Eq. 9:

$$p(x) = \frac{e^{b_0 + b \cdot x}}{1 + e^{b_0 + b \cdot x}} \quad [9]$$

If for a feature vector x, p(x)<1/2 then this is classified as y=–1 and y=1 otherwise (often y=–1 is labeled as y=0). A common alternative form of this model is in terms of the log of the odds $O=p/(1-p)$, which turns out to be a linear function, $log(O)=b_0+b.x$, better known as the logit model. Model fitting is done by using the maximum likelihood estimation (MLE) (13), which intuitively can be seen as a way of determining coefficients so that by inserting them to the logit function which will produce outputs as close to 1 as possible for all data items that are indeed classified as such. Likewise, it should produce a number as close to zero as possible for all items that are classified using the other label. As soon as training is complete and the coefficients determined, predictions are made simply by inserting unknown inputs to the logit function. This model, which is extendible to more than two classes, is related to SVM providing thus a probabilistic based version of them.

Regression techniques have a broad range of applications in bioinformatics. These methods have been widely adopted in normalization procedures of RNA-seq data, such as in (14) where a loess regression has been applied on RPKM values by using a spike-in RNA as control to fit the loess regression model. More recently in (15) the authors make use of Pearson residuals from "regularized negative binomial regression," to estimate for cellular sequence depth in order to be used as a covariate in a generalized linear model, which can efficiently normalize the data but also maintain the biological heterogeneity of scRNA-seq.

Naïve Bayes. Naïve Bayes (16) is a popular probabilistic approach for supervised learning which estimates the conditional probability for a random data point being in class j. Suppose we have data points described by n features, $x=(x_1,x_2...,x_n)$, and a target feature y. This method assigns probabilities that a data point characterized by features x is classified in terms of y using the conditional probability $P(y|x)$ called the posterior probability. This is the desired outcome of the model, which is obtained in terms of information available from the training (*i.e.*, already classified) data, *i.e.* the quantities $P(x|y)$ (the likelihood), and the joint probability P(x) and P(y) of the input and target features respectively. This connection is obtained through Bayes' theorem as $P(y|x)=P(y)P(x|y)/P(x)$. We then classify a data point x, to this class y for which $P(x|y)$ is maximized, leading to an estimation for the classifier in terms of $\hat{y}=argmax_{\{y\}} P(y|x)$, with P(y/x) given in terms of the training data as above. To simplify the optimization problem required for the classification, the likelihood term can be reduced by assuming that the conditional probabilities of each feature given the target feature y are independent (or in some cases conditional independent), so that approximately $P(x|y=\prod_{i=1}^{n} P(x_i|y)$, where upon ignoring the unimportant factor $1/P(x)$, which does not depend on y, we obtain the estimator $\hat{y}=argmax_y (P(y) \prod_{i=1}^{n} P(x_i|y))$, which is simpler to calculate. The maximum a posteriori (MAP) rule (17) can be used to find an estimate, that maximizes the product of the likelihood and the prior probabilities.
Bayesian classification procedures have been applied to a variety of biological problems. These include determining gene expression differences from bulk RNA-seq analysis to the analysis of more complex systems such as in scRNA-seq analysis where probabilistic models are able to learn cell-specific parameters in order to drive normalization (18).

Random forests (RFs). A popular methodology based on decision trees (19) is random forests (RFs). An RF consists of a multitude of decision trees and is essentially an ensemble learning method the outcome of which is determined as either in terms of the majority or as some sort of average of the outcomes of the trees comprising the forest. To construct an RF, a procedure known as bagging is applied where the outcomes from multiple, randomly constructed decision trees are combined to determine the final decision. Bootstrap aggregation (20), which is also referred to as bootstrap aggregating or bagging, is a technique which enables the

reduction of variance of ML methods, considerably improving their accuracy. This technique may prove useful to avoid overfitting. The predictions from each data instance on the corresponding testing data instances are combined through a process known as majority voting to produce the final prediction as the one prediction with the highest frequency across the data instances. In RFs, bagging is applied albeit with a key difference: each decision tree selects a random subset of the features at each split (Figure 4).

The sampling of features enables the random forest to eventually filter out the "weaker" predictors and use those that are strongly related to the outcome. Thus, bagging alongside with random features subsets selection, increases the diversification to produce better outcomes. Having as input the original training data, sampling must be done uniformly to yield the predetermined number of subsets which are to be used to train the same number of instances of the decision trees. The same item may be included to more than one subset, thus bagging sampling is done with replacement. If the training subsets have no overlaps the technique is called pasting. The quality of an overall random forest architecture depends, to a great extent, on the quality of each individual predictor (*i.e.*, each individual decision tree). The correlation between two or more features in the random forest should be minimal. An ensemble of trees using bagging is expected to produce lower quality outcomes in comparison to an equivalent Random Forest.

Decision trees and random forest help drive prediction of complex problems in biology. This includes the prediction and applicability of specific drug treatments in oncology (21) or in combining metabolic labelling of RNA to extract RNA kinetic rates and to use a Random Forest approach to classify the function and use of coding and nc-RNAs in eukaryotes (22).

Artificial neural networks (ANNs). Artificial neural networks (ANNs) are computational systems whose layout and operation were inspired by the way a biological brain works. An ANN consists of a set of interconnected artificial neurons, where each artificial neuron is described as a mathematical function which receives a set of input features that are aggregated in a proper way to produce an output that is inserted into a non-linear activation function that produces outcomes which can be transmitted to other neurons to repeat the same process until the final output layer is reached. Thus, a neuron is inspired by the operation of the nerve cells of a biological brain which operates in a similar manner. ANNs are organized in layers. In its simplest form an ANN consists of three main parts, namely: (i) the input layer, which provides the original inputs to the network, (ii) a single hidden layer which consists of a number of artificial neurons tasked with transforming inputs in order to produce activations, and (iii) an output layer, which produces the results of the network using the activations. Multiple hidden layers can exist between the input and output layers. A fully connected neural network with more than one hidden layer is called a multilayer ANN.

ANNs can modify their internal structure (by using their inputs which consist of actual measurements, weights, and biases) in relation to a set of desirable outcomes. This is the underlying concept of the learning process followed in all implementations. ANNs are used in fields, such as, pattern recognition, classification tasks and natural language processing (NLP). A variety of actual implementations has shown that they are particularly suitable for determining acceptable, almost optimal, solutions to complex non-linear problems. In a feed-forward ANN, the information flow has a single direction. To clarify the operation of an ANN let us start with the simple case of a single layer and k neurons. The key components there are the set of input vectors $x=(x_1,...,x_n)$, associated biases $b=(b_1,...,b_k)$, and weights $W=(w_{ij},i=1,...k,j=1,...n)$ considered as a k×n matrix. Each neuron i takes the input vector x, nonlinearly transforms it in terms of an activation function h and returns an output z_i which depends on the weights and the activation function as $z_i=h_i\ (\Sigma_{j=1}^{n} w_{ij}\ x_j\ +b_i)$, where h_i is the activation function and its argument is often called the activation of the neuron. The vector $z=(z_1,...,z_k)$ is considered as the output of the neurons. Based on training data the parameters of the ANN, which are essentially the weight matrices and the biases at each layer are computed and are minimizers of an appropriate loss function which connects the actual input with the observed final output for the training data. Then, once trained, the ANN can be used to classify or predict new data. The optimization procedure involved in the training of an ANN is complicated and often time consuming but technically simple as it often relies on gradient descent methods or its variants.

In classification scenarios, the output function is usually tasked with mapping numeric parameters to labels (or classes). The number of neurons in the output layer is typically the same as the number of classes. To calibrate a Neural Network, the loss function is required in order to measure the model's error. The primary task of the training process is the minimization of the loss function. This is achieved by re-adjusting weights and biases through a repetitive process which concludes after a set number of iterations is completed and/or in case the loss function produces acceptable (*i.e.*, sufficiently minimized) results. The two most widely adopted loss functions are the mean squared error (MSE) and the cross-entropy loss (CEL). Non-linear activation functions are required to solve complex problems. Among them some of the most prominent are the sigmoid, rectified linear unit (ReLU) (23) and softmax (24). The latter proves particularly suitable for the output layer. There is no intuitive way of determining the number of hidden layers. In many cases this is chosen through a trial-and-error process.

It is generally acceptable that relatively "shallow" ANNs, consisting of up to three layers, produce good quality outcomes. On the other hand, a great number of layers (Figure 5) may lead to overfitting.

These basic principles provide the background for introducing deep learning approaches for machine learning which as we will investigate have been applied to a plethora of applications in biology.

The rise of deep learning. The increasing technological achievements has enabled us to implement network architectures, embedding thousands of neurons. These topologies use "deep" depth neural networks with or without prior knowledge for training and their fundamental principles use the general notion of Hopfield or Perceptron network architectures (25).

To extract the ultimate features that can increase a classification performance, filtering steps can be applied with the use of convolution layers. Convolution layers walk through the matrix of values and filter the important features in one or three dimensions. This can be done *via* adopting a specific kernel function for partitioning or grouping the data. Moreover, sub-sampling techniques can be used for extracting important features *via* the use of maximum pooling layers. The maximum pooling can be evaluated either *via* grouping every n data and extracting either the maximum, median, L2-norm, or average values. The data from all input sources after passing the previous steps can be concatenated and merged into a fully connected neural network. The output will be a set of weights after training where these will be evaluated on a test dataset. Such architectural schemes are defined as convolutional neural networks (CNNs). An example of such a network can be visualized bellow (Figure 6). The filter scanning or windowing can be set accordingly to either 2×2 or 3×3 depending on the data set being used. Other popular deep neural network architectures under investigation are encoders-decoders or autoencoders (Figure 7).

This type of architecture consists of three layers, an input layer, a hidden (encoding) layer, and a decoding layer. Regarding the decoding layer the input is partially reconstructed, based on the important features. The hidden layer serves to learn good representations of the inputs. In an encoder-decoder architecture the input is suppressed and only the features that differ are presented. The convolutional filters are reduced at the different layers. The decoders try to reproduce the input, while the convolution layers are increased oppositely than the encoders. This allows to distinguish the ultimate features that can define a classification regime in every turn of training. Recurrent neural networks (RNN), such as the long short-term memory, or LSTM can learn recursively (Figure 8).

These types of networks make use of an internal state of memory to process sequences of inputs. RNNs are popular for speech or handwriting recognition. Furthermore, this form of architecture makes all the inputs relate to each other. Deep–learning applications have gained a lot of use in bioinformatics and system biology as they can be used either in supervised or unsupervised ML schemes and predict an outcome using a number of features. The increasing amount of biological input makes them suitable in various learning approaches from motif finding of TFs and RBPs to single cell clustering as it will be extensively analyzed in Section 3.

Unsupervised Learning

Unsupervised Learning is an ML technique where the training is done without supervision, instead the model tries to discover patterns through mimicry and tries to build a compact representation of the input data. Known unsupervised learning techniques include the dimensionality reduction techniques PCA, UMAP and k-means while neural networks can be used as well.

k-means. The k-means algorithm (26) is perhaps the most popular and simplest data clustering algorithm. Assume we are given an input vector with N-data points (samples), $x_i \in R^n$ (considered as embedded in some Euclidean space R^n, hence described as elements of R^n and visualized as points in this space) say $X=\{x_1,x_2,...,x_N\}$. Their embedding in R^n implies that data points which are similar (in some qualitative sense) will display this feature as having their Euclidean distance minimized in R^n. Under this perception, one way of obtaining clusters of similar data points is to arrange these data points in sets C_r, $r=1,...,k$, each located around a common "center" m_r with the points in each such set (cluster) being similar in terms of their distances between themselves and the corresponding cluster center. This procedure could be easily visualized if l=2,3 however a more sophisticated abstract formulation is needed in the case of high dimensional data, which is the case of practical interest in most applications. The above variational problem can be solved in terms of the k-means algorithm which is an iterative process which starts by randomly selecting k data points from x that serve as the initial candidate for the clustering centroids. Then, it assigns the remaining data points (*i.e.*, the data points of $x_j \in$X, where $x_j \neq m_r, r=1,...,k$) into the k clusters by: (i) calculating the Euclidean distance between each data point from each clustering centroid, and (ii) assigning a data point x_j to the cluster C_i if the Euclidean distance of x_j from the r-th clustering centroid is smaller than the distance from the rest of the clustering centroids.

This is the main essence of the k-means algorithm, which is an NP hard problem in terms of computational complexity. Furthermore k-means has been used extensively in bioinformatics, one example is in cases of scRNA-seq data

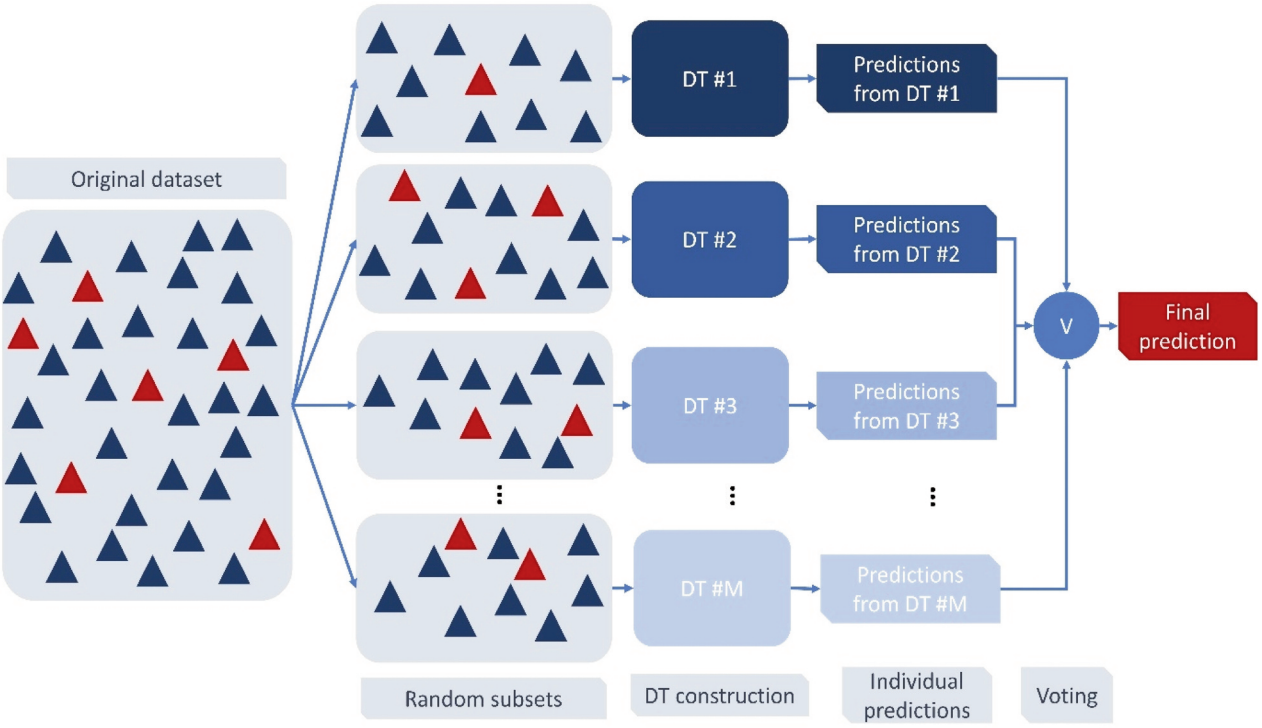

Figure 4. *An illustration of bagging where the outcomes from multiple, decision trees are combined to determine the final decision.*

when creating the different sub-classes of cell types. Such an implementation can be seen in mbkmeans (28) formulation, where the authors make use of sub-sampling approaches with the use of mini-batches, while evaluating k-means for each batch, thus minimizing the memory requirements. Other methods can also include least absolute shrinkage (Lasso) techniques that can cope with the sparse high-dimensional nature of scRNA-seq together with k-means thus to apply feature selection and clustering of the cell sub-types (29).

Dimensionality reduction. The primary objective of all dimensionality reduction techniques is to seek for the optimal data representation model, in terms of data compression, which describes the initial dataset without significant information loss which is feasible since most datasets contain redundancies. To put it differently, some sets of features can be regarded as an indicator of another, initially unobserved, latent feature. This also implies that these sets of original features are correlated. On the other hand, if no redundancies exist within a dataset compression, regardless of the method, is unable to produce significant results. Thus, a dimensionality reduction method reduces an original -dimensional feature space to a -dimensional feature space (where, $n>k$), which can then be used to train a machine learning algorithm in a smaller feature space.

Principal component analysis (linear). Principal component analysis (PCA) is one of the most widely adopted dimensionality reduction methods. PCA aims to create a feature space of reduced dimensions while preserving as much variance as possible in the original dataset. To illustrate the idea, consider a data set consisting of N data points $x_i \in R^n$, $i=1,...,N$, where n is the original dimension of the feature space (equivalently a sample of the vector valued random variable $x=(x_1,...,x_n)$ and which may conveniently be considered as a data matrix $X=[x_1,...,x_N] \in R^{(n \times N)}$. Alternatively, each row j, j=1,...,n. of the matrix can be considered as N observations $(x_{ji}, i=1,...,N)$, of the feature j of the multidimensional data set. Then,

$$E[x_j] \simeq \frac{1}{N}\sum_{i=1}^{\{N\}} x_{ji}$$

is an estimate for the mean value of feature x_j, whereas

$$cov(x_k, x_j) \simeq \frac{1}{N}\sum_{i=1}^{N}(x_{ki} - E[x_k])(x_{ji} - E[x_j])$$

is an estimate of the covariance between the features x_i and x_j, with the matrix $S=(s_{kj})$ called the covariance matrix of the

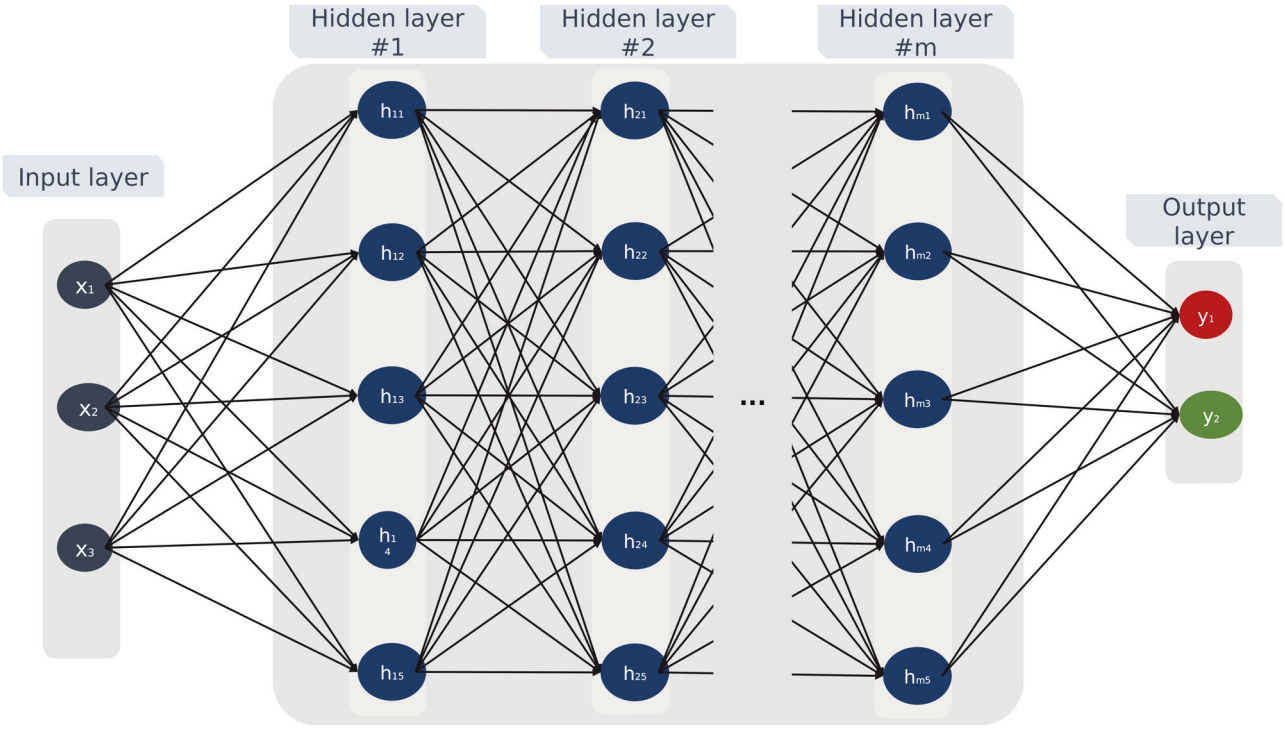

Figure 5. *An illustration of an artificial neural network with multiple hidden layers.*

features. Without loss of generality, we consider that $E[x_j]=0, j=1,...,n$, otherwise we may center the data matrix by subtracting the mean of each feature (row). Note that these important statistical quantities can be expressed directly in terms of the data matrix X. PCA attempts to find linear transformations of the **x**, in terms of n vectors $a_1,...,a_n \in R^n$ such that the new features $z_m=a_m \bullet x$, $m=1,...,n$ can well describe the sample and moreover that the transformation is such that only $k<n$ of the transformed features capture as much as possible of the variance of the original dataset. Using the sample for the vector valued random variable x, we have samples for each of the scalar random variables z_m. Recall that if $z=a \bullet x$ then $var(z)=a \bullet Sa$. An alternative way of looking at this is to consider the new features as a new coordinate system with which we view the original data set, as taking projections of the original data set along directions that minimize the projection error and thus the information loss. These directions define the new coordinate system and will satisfy appropriate orthogonality conditions *i.e.*, $cov(z_m,z_l)=0$, $m \neq l$. The resulting directions (equivalently linear combination of features) will be called principal components and will carry most of the information of the original data set in their first $k<n$ components. The choice of directions will be made as follows: Choose a_1 respectively z_1 such that $var(z_1)$ is maximum. At the r level, $r \leq k$, choose a_r, respectively and z_r such that $var(z_r)$ is maximum, subject to the constraints $cov(z_r,z_m)=0$ $m=1,...,r-1$ *and* $a_r \bullet a_r=1$. These problems can be handled using the technique of Lagrange multipliers, and the solution of these problems reveals that the desired directions a_r are solutions of the eigenvalue problems $(S-\lambda I)a=0$, where $S \in R^{(n \times n)}$ is the covariance matrix of the random variable x, which for centered data is related to $X \bullet X' \in R^{(n \times n)}$. The eigenvalue problem has n solutions for (λ_r,a_r), $r=1,...,n$. All the eigenvalues are positive, and the eigenvector α_1 corresponding to the largest eigen value λ_1 will indicate the first (dominant in terms of capturing the variance) PC, the eigenvector α_2 corresponding to the second largest eigen value λ_2 will correspond to the next (best performing after the dominant in terms of capturing the variance) PC *etc*. The number k of PCs retained is related to the spectral properties of the matrix $X \bullet X' \in R^{(n \times n)}$, and can be found by ordering the eigenvalues of the covariance matrix in descending order and keeping the first k dominant ones. The procedure is closely related to the singular value decomposition SVD. The above procedure could be performed using SVD of the data matrix X, according to which X admits a representation of the form $X=U \Sigma V'$, where $U \in R^{(n \times n)}$, $\Sigma \in R^{(n \times N)}$, $V \in R^{(N \times N)}$ with U, V being orthogonal matrices containing the eigenvectors of $X \bullet X'$ and $X' \bullet X$ respectively, and Σ is a matrix which contains nonzero elements only on the diagonal consisting of the singular values of X. The choice of k can be made by ordering

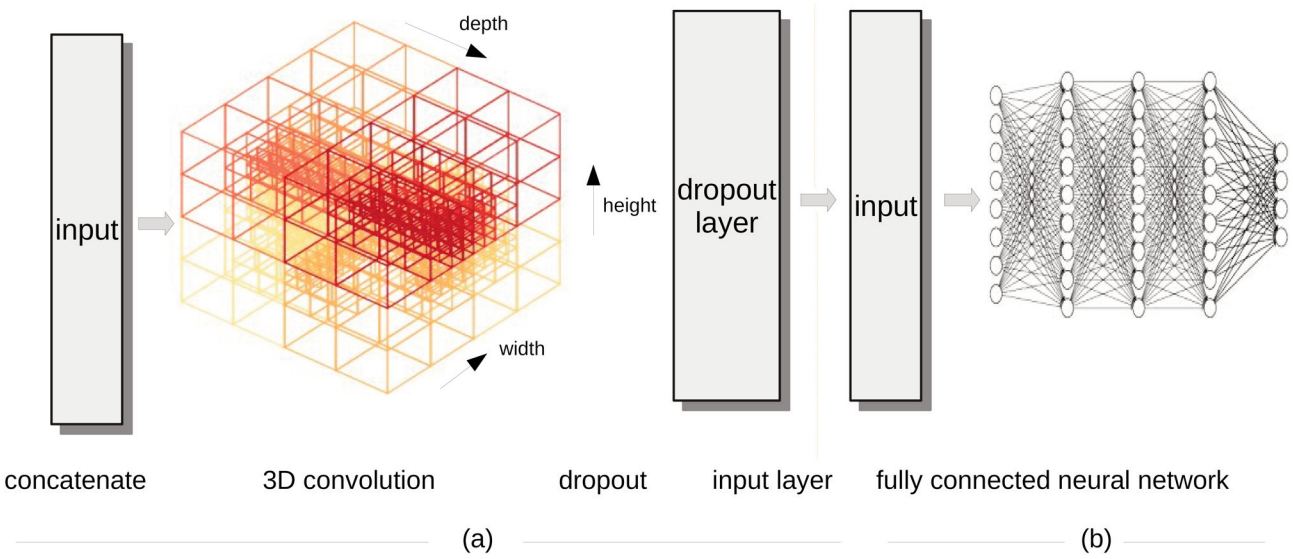

Figure 6. *A convolutional neural network (CNN) architecture schema. In (a) the convolution, concatenation, and dropout layers and in (b) a fully connected neural network is shown.*

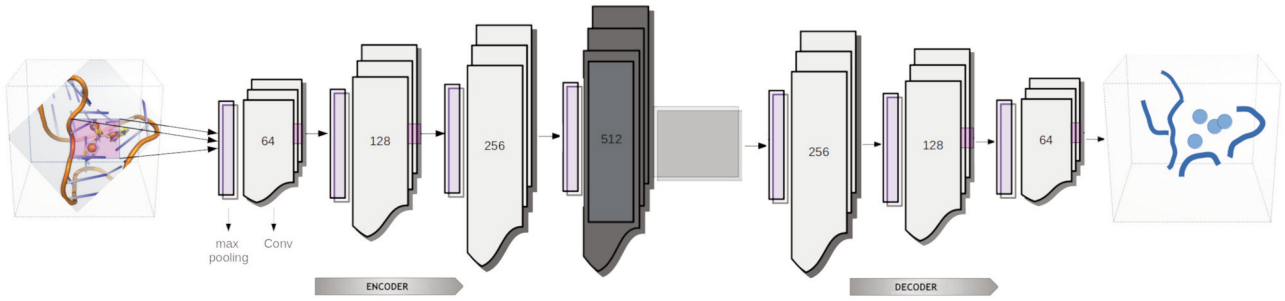

Figure 7. *A 3D Encode-Decode architecture, taking as input information a 3D molecule and extracting the most important information in trying to represent it, while determining the compressed information of it as output; in other words, visualising the most important features that have determined such classification or learning.*

the singular values of the data matrix, while the transformation to principal components can be made using U.

tSNE and UMAP: Two well-known dimensionality reduction techniques. t-SNE is one of the most commonly used dimensionality reduction techniques. It represents high-dimensional data by assigning each datapoint, being in a higher dimension, to a two- or three-dimensional map. This is done by using a Gaussian probability Eq.10 for observing the distance between any two points in the high-dimensional map where an optimal σ (Eq. 11) is defined or so-called perplexity. The goal is to minimize a loss function when projecting from a high dimensional distribution to a lower one *via* a gradient optimization technique.

$$p_{j|i}=\frac{exp\left(-\|x_i-x_j\|^2/2\sigma_i^2\right)}{\sum_k exp\left(-\|x_i-x_k\|^2/2\sigma_i^2\right)}, \quad [10]$$

the symmetrization being used is so to ensure that the any points will be glued efficiently together and is set as p_{ij}=(pilj +pjli)/2N

$$\sigma = 2^{-\sum_j p_{j|i} log 2 p_{j|i}} \quad [11]$$

The student *t*-test distribution can be used in the lower dimension to declare the distances (Eq. 12); thus, the t-SNE after describing the distance of any two points aims to learn the similarities of a d-dimensional map $y_i...y_n$.

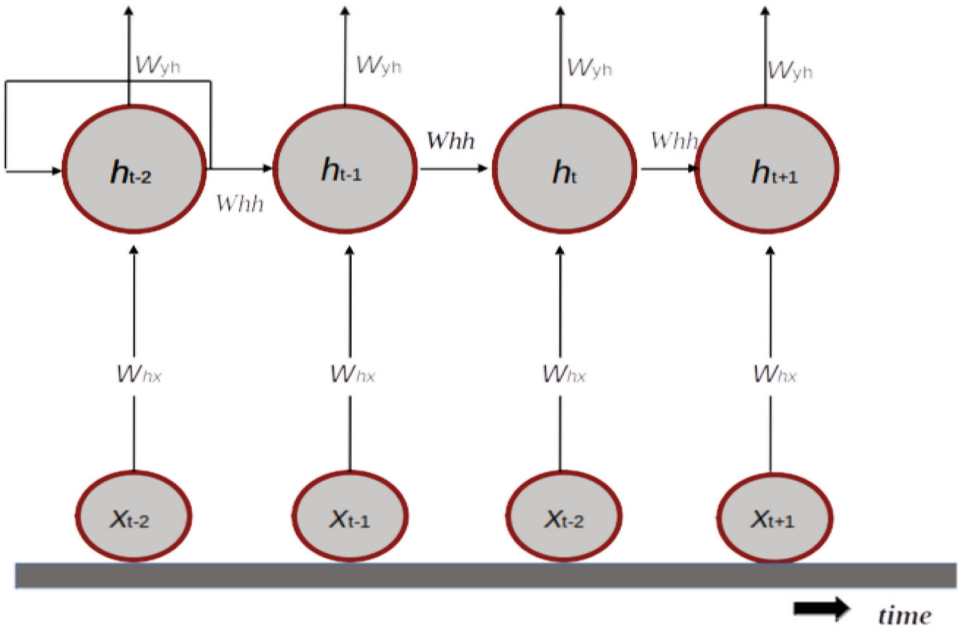

Figure 8. *Basic RNN architecture where the feedback loops define an internal state of memory.*

$$q_{j|i} = \frac{\left(1+\|y_i - y_j\|^2\right)^{-1}}{\sum_{kl}\left(1+\left(\|y_k - y_j\|^2\right)^{-1}\right)} \quad [12]$$

The locations of the point in the map of d-dimension will be learned by estimating the Kullback-Leibler divergence loss function of the distribution P from the distribution Q (Eq. 13)

$$KL(P||Q) = \sum_i p_{ij}\, log\, \frac{p_{ij}}{q_{ij}} \quad [13]$$

UMAP stands for uniform manifold approximation and projection: the algorithm tries to approximate a manifold on which the data lie and constructs a simpler representation of it. UMAP assumes that there exists a Riemannian manifold where the data are uniformly distributed. A simpler version of a manifold is assumed and derived *via* constructing a weighted k-neighbor graph. A fuzzy topological connectivity is applied on the edges of the graph having as a constraint the minimization of the cross entropy to deal with any inherent asymmetry. The algorithm will proceed by iteratively applying attractive and repulsive forces at each edge or vertex. This will converge to a local minimum by dynamically decreasing the attractive and repulsive forces.

Like t-SNE given an input data set as $X=\{x_1,...,x_n\}$, and an input hyper-parameter k, for each x_i we compute the distance of each set of $\{xi_1,...,xi_k\}$ points from the *k*- nearest neighbors. If, for each x_i, we denote by p_i its distance from its first nearest neighbor x_j then:

$$p_{i|j} = e^{\frac{-d(x_i,x_j)-p_i}{\sigma_i}} \quad [14]$$

with the symmetrization here to be $p_{ij}=p_{i|j}+p_{j|i}-p_{i|j}p_{j|i}$ and like t-SNE the optimal σ_i will be

$$\sigma_i = \sum_k^{j=1} exp\left(\frac{-max(0,d(x_i,x_{i,j})-p_i)}{\sigma_i}\right) \quad [15]$$

Since ϱ demonstrated the distance from each i-th data point to its first nearest neighbor this demonstrates a local connectivity of a manifold. The process of UMAP is approximating the number of nearest neighbors *k*, where for these *k* neighbors the UMAP function tries to glue together points with locally varying metrics. UMAP, unlike t-SNE, doesn't use a t-distribution but instead it uses a family of curves which demonstrate the connectivity or strength between any of two points in a manifold as these can be defined as attractive or repelling forces, where a, b are hyper parameters: $1/((1+a \cdot y^{2b}))$. A distance probability $q_{(i,j)}$ is set as $q_{ij}=(1+a(y_i-y_j)^{2b})^{-1}$. The goal is to find the minimum distance between points i,j in a 2D space described by $X=\{x_1...x_i\}$, $Y=\{y_1...y_i\}$, where overall this will lead to

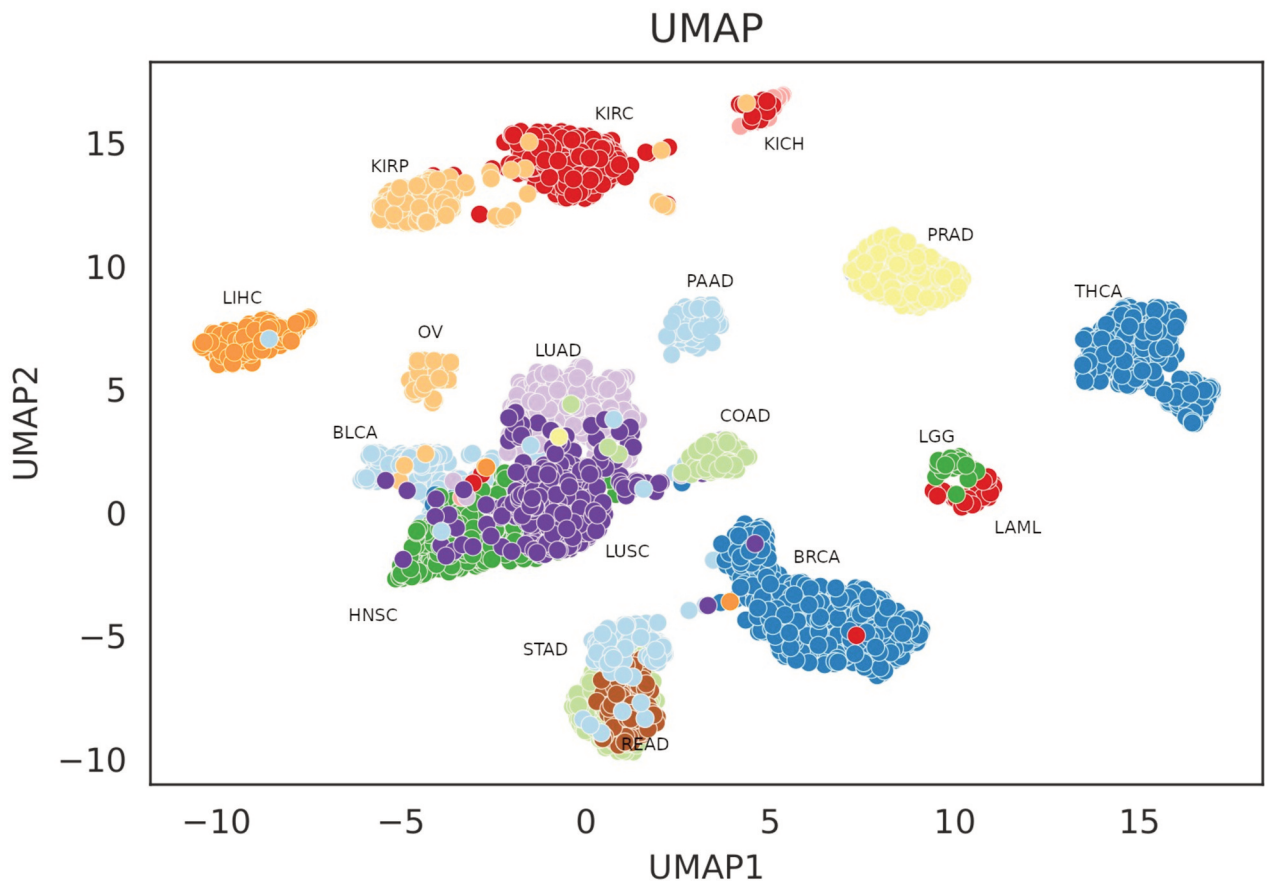

Figure 9. *UMAP example for classifying cancer types based on TCGA gene expression data.*

compact clusters with the lowest cost estimate. Regarding the cost estimate, UMAP sets this as a binary cross entropy CE approximation, as in Eq. 16:

$$CE(X,Y) = \sum_i \sum_j p_{ij}(X) * log\left(\frac{p_{ij}(X)}{q_{ij}(Y)}\right) + \left(1 - p_{ij}(X)\right) * log\left(\frac{1 - p_{ij}(X)}{1 - q_{ij}(Y)}\right) \quad [16]$$

To estimate the cost function, we need to apply the gradient of the cross-entropy *via* applying the gradient descent. Setting the distance $d_{i,j}=y_i-y_j$ *and* $Q(d_{i,j})=1/(1+ab_{ij}^{2b})$ we obtain (Eq. 17):

$$\frac{\theta CE}{\theta y_i} = \sum_i \left[\frac{-P(X)}{Q(\theta_{i,j})} \cdot \frac{\theta Q}{\theta d_{i,j}} + \frac{1 - P(X)}{1 - Q(\delta_{i,j})} \cdot \frac{\theta Q}{\theta d_{i,j}}\right], \quad [17]$$

In general, a Laplacian graph is used by UMAP to initial low dimension coordinates. In this context a graph is first constructed using the kNN algorithm (30), where it is formalized *via* constructing the Laplacian matrix. The eigen-value-decomposition problem is then used to factor the matrix. Figure 9 demonstrates an example of UMAP clustering when classifying gene expression cancer data from TCGA (31).

Machine Learning Applications in Biology and Bioinformatics

In the last decade, the technological breakthroughs of biochemistry and next generation sequencing (NGS) have led to the generation of a huge amount of information. The need for novel ML approaches to decipher fundamental mechanisms of biology has gained great importance (32). This section will provide and describe several ML techniques that have been previously analyzed, while taking advantage of state-of-the-art biochemical methods to generate NGS data. Several techniques will be described for recognizing the functional domains of TFs and RNA binding proteins while taking advantage of neural networks to acquire this information. Then an analysis of the latest methods will be

given on using expression data and genome wide association studies (GWAS) from TCGA and other consortia to learn the pathways that may drive diseases. Also, the latest achievements in scRNA-seq will be examined together with the ML techniques and dimensionality reduction algorithms that may enable us to learn and predict the cell states under normal conditions or in disease.

The art of binding "from transcription factors to RBPs"

Learn the determinants of transcription. A large majority of encoded proteins have functional and regulatory roles. These elements such as TFs can recognize and bind specific regulatory sites of the DNA. These sites are composed of specific nucleotide motifs and recognize a characteristic geometry with high accessibility of the DNA helix. The binding of TFs in the promoter of a gene either facilitates the pre-initiation of the transcription complex with RNA Pol II or the TFs upon binding can recruit transcriptional cofactors to alter the chromatin state. Furthermore, histone methylation such as H3K4me3 contributes to active transcription while H3K27me3 and H3K9me3 have a repressive role. Other histone modifications include phosphorylation, ubiquitination, sumoylation and ADP ribosylation (33).

Chip-seq (34) and ATAC-seq (35) are two established methods that enable the identification of binding sites of TFs. With chromatin immunoprecipitation sequencing (ChIP-seq) we are able to identify the binding of TFs on the DNA as a specific antibody is used to IP the complex. More precisely, crosslinking DNA and proteins uses formaldehyde, followed by sonication of the DNA into smaller fragments usually around 200–600 bp fragments. This is then followed by immunoprecipitation of the DNA-protein complexes of interest with antibodies. The DNA is then uncross-linked, and the DNA is adapter ligated according to the library preparation steps before it is sequenced. The obtained fragments are the binding sites which hold the region of interest. Regarding ATAC-seq the nuclei of cells are isolated and a Tn5 transposase is used while ligated to adapters to identify open chromatin regions which are more accessible. Overall, these techniques have enabled us to identify the regulatory domains that drive transcription.

To predict the binding domains and decipher the regulatory properties of such sites, ML strategies have been employed. Such methods include convolutional neural networks such as in the case of DeepBind (36) which uses the sequence fragments of TFs and the open chromatin regions from experiments such as ATAC-seq or Chip-seq as input. More specifically, DeepBind determines a score for the binding positions in four stages. The convolution stage scans and groups appropriately a set of sequences of length m. There is a motif detector step which is 4 × m matrix, that extracts frequencies and forms position weight matrixes (PWM), which are then fed by a rectification stage where positions with high scores are selected, and all negative values are set to zero. There is a pooling layer which uses maximum and average techniques to identify short-in-long motifs which are then given as input into a nonlinear neural network. Similar studies include more complex architectures such as recurrent neural networks (RNNs) or long short-term memory (LSTM) which can improve the binding accuracy. Other methods such as KEGRU (37) use bidirectional recurrent networks named as b-GRUs which makes use of a k-mer sequence representation in combination with the states of the recurrent method to capture more efficiently the dependencies and thus achieve better performance. Other combinatorial methods such as Janggu (38) use DNA sequences from DNAse tracks as input to a convolution neural network. This method uses a higher order one-hot encoding of the DNA sequence that captures di- or trinucleotide-based motifs. Similar combinatorial methods as in DeepSEA (39), make use of various inputs such as DNase I–hypersensitive sites (DHSs), histone marks and TF binding profiles and have as a major goal the identification of functional effects of noncoding variants. Inputs are genomic sequences of the positions of the marks which are used in a deep neural network architecture. The effects of individual SNPs on TF binding sites have been evaluated with high performance. Similar DNA genomic features that can be investigated with deep neural networks include the investigation of DNA methylation sites and the analysis of chromatin loops from genome-wide interaction matrices from Hi-C experiments (40). The latter uses as input genome wide interaction maps and a set of positive defined and negative training sets to a binary classifier. A hyperparameter search is then followed to find the best random forest model separating the two classes, which can be used to detect loops from genome-wide contact maps. Another interesting approach that uses Hi-C to determine nuclear compartmentalization is presented in (41), where Hi-C data were employed to construct a Hi-C interaction graph whereby graph embedding techniques 1st and 2nd order proximities are derived, thus transforming the graph into a lower-dimensional space where *k*-means clustering is applied to cluster the nodes.

Learning the RNA binding properties. Apart from the DNA binding elements the research community has started to use ML applications to investigate the binding properties of RBPs. This task has gained great attention due to the hard nature and the many functional features of RNA such as the various sequence motifs and RNA structure in 2D and 3D. Unlike DNA, RNA is a molecule of high plasticity which adopts various regulatory structures from its transcription to decay. Furthermore, more than 100 different modifications have been examined and more than 1,800 regulatory RBPs have been discovered. Only recently, biochemistry has enabled us to detect the RNA structure in 2D or even in 3D genome-wide (42) while adopting the RNA backbone

confirmations. These can be accomplished *via* evaluating techniques that enable us to use biochemical reagents that can cause mutations (43) or RT stops (44) upon open accessible regions or that make use of specific cross link techniques with compounds such as AMT (45) that can capture the double stranded RNA regions.

Furthermore, several enzymes such as in (46) can distinguish and cleave either single or double stranded RNA conformations to distinguish RNA structure. In addition, a large plethora of experiments have been developed to identify the binding domains of ribonucleocomplexes (RNPs). These mainly include UV cross linking and IP with a specific antibody and pull down of the complex. Furthermore, in order to increase the sensitivity of the binding position a 4sU analog of uridine can be used to allow for a specific mutation during cDNA synthesis (T>C conversion) upon the binding of the RBP with the RNA (47). Similar methods such as HITS CLIP (48) include mutations on the binding position that can be identified and increase resolution. In iCLIP (49) and upon circular ligation after the IP pull-down, the identification of an RT stop can be seen during cDNA synthesis on the RBP contact point. Thus, the exact location of the binding site irrespective of the sequence composition can be extracted. In addition, if these methods are combined with specific enzymatic cleavage, one can determine the RNA structure composition of the binding domain. Protein occupancy profiles (50) have determined various RNA binding proteins that can crosslink RNA. Moreover, cell fractionation experiments coupled with IPs and specific biochemistry for in vivo identification of RNA structure can be used to determine with more specificity the binding properties per cell compartment. This vast amount of information, if combined with ML, can lead to identification of the binding specificities of RBPs, *i.e.*, the sequence and structural motifs that may drive such binding. DeepBind, as examined earlier, has also investigated the binding properties of RBPs using as input information the sequence of exon-exon junctions and demonstrates the leading binding motifs for RBPs known to regulate splicing such as TIA1, NOVA, PTBP1 or hnRNPC. IDeep (51) uses eCLIP-seq data from ENCODE (52) and also the RNA structure to train a hybrid network with two CNNs and a long-short temporary memory (LSTM) network. The input from sequence and structure is driven to CNNs and the LSTM learns the binding properties in terms of sequences and structures to improve prediction. Other methods such as DeepRipe (53) one-hot encode the sequence or in the case of pysster (54) also include the sequence and RNA 2D structure using an extended alphabet into a convolutional max-pooling and dropout layers. After the dropout layers the information is used by a dense neural network which can be easily tuned. Similarly, DeepCLIP (55) applies a similar network architecture that uses 1D convolutional layers to find and enhance features of a set of presented sequences. This is followed by a bidirectional long, short-term memory (BLSTM) layer which uses the extracted features and contextual information of the sequences to find areas of the RNA-sequences associated with RBP binding. A different approach is being used by a kernel-based model called Graphprot (56). This scheme extracts the structure and sequence information from CLIP-seq data. The structure of the binding sites is calculated *via* RNAshapes (57). The structure and sequence are encoded as a hypergraph, where graph kernels are used to extract features to be set as input information to support vector machine (SVM) and support vector regression (SVR) (58) modules. A graph-kernel can extract large number of features and when comparing bound and unbound regions while using a k-mer similarity, the binding features per RBP can be extracted versus the background noise.

Extracting information of the RNA backbone using 3D RNA modeling has been a great challenge. Molecular dynamics using Monte Carlo techniques (59) can be applied. Usually in this type of modeling the RNA in its 2D form will be placed on a grid. The RNA molecules will be perturbed on the grid and in each position the Poisson Boltzmann equation describing the energy state of each atom will be deciphered having as goal to extract the position with the minimum energy as in (60). ML techniques can be employed to learn the 3D properties such as dihedral angles and total energies per cluster of molecules. One such method is RNA3DCNN (61), where the RNA molecules can be treated as a 3D image or voxels as input to 3D CNNs. The RNA molecule is described using a 3D grid representation of the RNA molecules on a cartesian coordinate system directly as input to the convolutional neural network. This network is arranged using an input layer, of a two-stage convolutional layer, followed by a maxpooling layer and another two-stage convolutional layer following an output layer. The output is a score per nucleotide, defining how a nucleotide fits in its surrounding, taking into account all the conformations. Deepnet-rbp (62) uses information of the tertiary structure motifs as predicted by JAR3D (63) together with the sequence and structure into a multimodal deep learning module to predict RBP binding sites and motifs. JAR3D is a computational framework that extracts probable structural motifs in the hairpins and internal loop regions using RNA 3D Motifs Atlas (R3DMA) (64). The deep learning module uses restricted Boltzmann machines (RBMs) (65) of multi connected layers based on Markov random fields (66) which define the probability distribution of the variables. NucleicNet (67) uses features of the RNA backbone and the physicochemical characteristics of the RBPs such as hydrophobicity, molecular charges and accessibility surfaces calculated from Fpocket (68) with the ultimate goal to predict on each location of the RBP's surface, scores for

RNA interactions. More precisely the surface contact points of the RNPs are extracted using the physiochemistry of the RNA and the RBP's potential contact points. Furthermore, these are clustered into classes that correspond to the bound and non-bound RNA sites. A deep residual network is trained to determine the scores according to the physicochemical properties and the network is optimized through standard back-propagation of the categorical cross entropy loss.

All these methods presented examples of different ML architectures used to decipher mechanisms of fundamental principles of biochemistry which mainly define specific structural and sequence motifs or the biophysical properties that can drive RBPs or TFs to bind and thus contribute to the post transcriptional regulation that determines the cell fate.

Prediction of protein 3D structures using ML. Accurate prediction of 3D protein structures remains a very demanding task in terms of computational power as there can be more than 10^{300} possible different ways that a protein can be folded before setting into a final stable 3D structure manifold (69). The importance however of evaluating an accurate 3D protein folding structure is of great essence in order to investigate protein function. Incorporating ML methods for predicting the 3D protein structure has been recently adopted to improve this task. The most well-known example is the Alphafold algorithm (70) where a neural network was trained to make predictions regarding the distances between pairs of residues and construct a potential mean force, able to define the shape of the protein with high accuracy. The predictions include backbone torsion angles and pairwise distances between residues. A gradient descent method was applied to optimize the predictions.

Machine learning to identify the per cell transcriptome. The recent advances in the field have enabled scientists to distinguish the transcriptome properties per cell using novel sc-RNA seq techniques. This achievement allows for the first time to discover the progression of a disease as the signaling is driven from the cell-to-cell communication and define more precisely marker genes in each cell stage and each cell cycle. The power of ML, and dimensionality reduction techniques is mandatory to extract the driver genes and RNA properties from this ever increasing per cell information. A plethora of methods and analysis exist taking advantage of several dimensionality reduction techniques. These methods as explained previously can demonstrate the various cell types, emerging from a single cell experiment but also examine driver genes of each cellular sub-population. Furthermore, cell deconvolution is an important aspect for such analysis. ML methods such as linear regression can be used on gene expression profiles (GEPs) of specifically expressed genes per cell type to estimate cellular sub-populations. Other methods such as Scaden (71) make use of a deep neural network for cell deconvolution. This method uses gene expression information as input, to a DNN where the hidden layer nodes of the DNN would account for higher-order latent representations of cell types. This seems to be more robust in terms of technical bias and noise. Many methods such as k-means, gaussian mixture models and spectral clustering have been employed for clustering cells from sc-RNAseq experiments.

Recently force directed layout graphs have been applied as another dimensionality reduction technique. FLOW-MAP (72), uses a graph layout analysis and sequential time ordering to extract cellular trajectories in high-dimensional single-cell data-sets. Diffusion maps (73) have also gained interest as they can preserve the relations between the data points and thus are more suitable for re-ordering the differentiating cells and for reconstructing developmental traces. Also, diffusion distance is robust to noise. Other methods to provide an alternative dimensionality reduction approach use the power of neural networks such as in the work of (74) and (75). Variational autoencoders (VAE) together with Bayesian inference have also been used to learn a probabilistic encoding of the data. Dhaka (76) is such a method where the assumption is that the data is coming from a multivariate Gaussian probability distribution. The autoencoder encodes the means (μ_z) and variances (σ_z) of the Gaussian distributions. The sampled latent representation is then passed through a similar decoder network to reconstruct the input.

The major problem to tackle, in scRNA-seq analysis, is the large dropout rates in gene expression. Spectral clustering (SC) (77) is an emerging popular method that combines multiple kernels to learn a distance metric using the KNN algorithm that best fits the structure of the data. To resolve the issue of drop-out in sc-RNAseq data, several imputation methods have been implemented such as scImpute (78), MAGIC (79) and SAVER (80). Deep Impute (81) uses standard deep neural networks to predict the missing values by using correlated genes with high gene expression values. In brief, DeepImpute imputes gene counts in a divide-and-conquer approach, by constructing multiple sub-neural networks where each sub-neural network is used to decipher the relationship of certain category of genes with a subset targeted gene. Each sub neural network can use 512 input genes, a hidden layer of 256 neurons using a Relu activation, a 20% dropout layer and an output dense fully connected network. Similarly, DCA (82) uses a deep autoencoder scheme to denoise scRNA-seq by defining a reconstruction error as the likelihood of the distribution of the noise model instead of reconstructing the input data itself. Probabilistic principal components analysis (PPCA) or factor analysis (FA) (83) can also account for these events and provide another form of clustering. Zero inflated factor analysis (ZIFA) (84) defines a dropout relationship which takes into account the mean level of non-zero expressed genes (log read counts) as μ and the dropout rate P_o for each gene. The relationship can thus be defined as $P_o=exp(-\lambda\mu^2)$, where λ is a fitted parameter, based on a double exponential

function. If one assumes that the separable cell states lie as points in a low dimensional space, these can be transformed and projected into a higher dimensional space using a linear transformation while adding a Gaussian distributes noise. Each point or cell then has the probability of being set to zero based on the dropout relationship.

Trajectory inference methods have enhanced single cell analysis which enables us to determine genes that are associated with specific lineages or are differentially expressed between lineages *via* the use of branching events. Monocole and Beam (85) fit additive models of gene expresion as a function of pseudotime. Bifurcation analysis can be inferred to identify whether a gene expression can be differentially associated with any of two lineages.

Quite recently it was shown that, by a technique named velocity (86), the future states of individual cells can be predicted. In RNA velocity, the time derivative of the gene expression between spliced and unspliced genes can be distinguished from the reads that fall within introns and exons. This can result in an extra layer or feature in a ML scheme to predict the cell states and cell cycle if also coupled with Viterbi (87) or a Hidden Markov algorithm or even if it is introduced in a neural network. In addition, when introducing mRNA labeling techniques such as in SLAM-seq (88) with scRNA-seq this creates an extra dimension for modelling RNA kinetis. Dynamo (89) provides a framework that incorporates intrinsic splicing with mRNA labeling kinetics to determine RNA velocities and extract vector fields that determine and predict future cell fates. To accomplish this, a machine learning scheme is employed which contracts a kernel method to learn a vector field in Hilbert space *via* the use of weighted linear combination of functions that describe the field.

Apart from deciphering the dynamics of the transcriptome per cell type, single cell RNA-seq technologies have advanced and allow us to decipher the binding properties of TFs per cellular sub-type. This can be achieved with scATAC-seq which has emerged as the method of choice to map open chromatin regions, which can be used to infer TF binding events per cell type. ScFAN (90) uses a CNN pretrained on bulk Chip-seq or ATAc-seq data and is used to predict TF binding properties at the single cell level. The data input is a feature vector of 1000 base pair bins from total ATAC-seq, Chip-seq and DNA sequence into a CNN linked to two fully connected layers while using a sigmoid function to make predictions of motifs for TFs. This model is the used on scATAC-seq (91) data to predict the candidate active TFs per cell state.

More recent methods based upon these principles introduce a weighted-nearest neighbor approach as in (92), which enables embedding of multiple datasets to be used for clustering cell sub-populations. Thus, this allows us to consider various single-cell approaches such as scATAC-seq for determining chromatin accessibility or CITE_seq (93) to be combined and thus infer a specific cell sub-population.

In this section, we have studied several applications of scRNA-seq analysis while incorporating ML methods to better project and cluster cells from a high dimensional manifold to a reduced representation. This however, as studied extensively in (94), can result to bias regarding the interpenetration of results. The authors provide an example of a sphere to determine dimensionality reduction loss by comparing the local neighborhood of a point in the sphere with the neighborhood of the same point in the reduced dimensional space using the Jaccard distance as a metric. Implementing AI techniques for such a task such as using a VAE network, can greatly contribute to the dimensionality reduction and loss. Moreover, considering the reduced representation, data from various single cell biochemistries and applying specific weights while building a new graph topology when using all available information can also improve the observed bias.

Training on gene expression patterns and GWAs studies to learn drug targeted therapies. Repositories such as TCGA (31), Cosmic (95), cBioportal (96) and CPTAC (97) have enabled scientists to use a large repertoire of data sets targeted on oncology from samples that are retrieved directly from patients from a variety of cancer types. Drugs targeting specific genes are frequently used in chemotherapy, thus learning the expression values of certain genes, the mutation and methylation profiles in addition to other features such as alternative splicing events or specific regulatory features of the RNA for these transcripts can help determine ultimate treatments and move towards a more personalized medicine approach per patient and per cancel type. Recent advances in this field such as in (98) have used association rule mining techniques (99) to distinguish how mutations in compliance with gene expression can result in chemoresistance; this is done by generating gene association correlations while extracting scores and ranking them to prioritize pathways. Furthermore, by incorporating deep neural networks (DNNs) as in (21), which were trained and optimized on a 1,001 cell-line drug response database, a model was generated which was tested blindly on patient cohorts to determine the best treatment. This approach compared several ML implementations such as random forests (RFs) and elastic nets (Enets) (100) with DNNs to decipher the best performance model, while tested from patients derived from TCGA cohorts and the Multiple myeloma consortium (101). Another approach named DrugCell (102) combines conventional artificial neural networks (ANN) with a visible neural network (VNN) to learn, using as input mutations from GWAS studies per cancer patient, molecular subsystems from 2,086 biological processes of the Gene Ontology (GO) database (Gene Ontology Consortium 2004) and the drug chemical structure encoding the Morgan fingerprint of a drug (103) to learn specific drug treatments per groups of patients.

New sequencing technologies and ML. In the last five years nanopore technology has started to shape a new era into introducing fast and simple sequencing technologies (104). These methods rely on a protein nanopore from which a DNA or RNA molecule with appropriate adapter priming is let through the pore. The interaction of each base with the pore creates a unique current signal whereby the base pair is determined. Areas of high GC content or polyA repeats can cause noise during the read-out of the current; thus, ML is applied to learn and correct for such errors. A hidden Markov model (HMM) based approach, has been applied to detect 5mC DNA in CpG from events of Nanopore reads. Similarly, DeepSignal (105) makes use of convolutional neural networks (CNN) using as input raw electrical signals around methylated bases. In parallel, a sequence feature module uses a bidirectional recurrent neural network (BRNN) that determines features from sequences of signal information. Then, the output features from the two modules (CNN and BRNN) are concatenated and fed into a fully connected neural network. The advances of such technology are many; small size of devise makes it portable but also direct sequencing of RNA molecules to define RNA modifications makes this technology a promising tool for RNA biology.

Concluding Remarks

This work provides an overview of the supervised and unsupervised ML techniques and the dimensionality reduction methods that are widely coupled with the fundamental mathematics behind them. Examples of how these applications can be used in bioinformatics for multiple integration of genomic data was shown regarding various tasks, from deciphering the elements that drive TFs and RBP binding sites to sc-RNAseq applications while leading to more holistic approaches for using such data sets to learn treatments in oncology. Thus, the aim of this review was to cover the importance and the vast applications of ML in biology. As a future perspective, a very promising contribution of AI and ML is strongly related to what we call "precision medicine". The main concept behind this type of approach is to apply principles of medical science tailored according to the needs and the personal characteristics of each patient. The era of personalized medicine, based on omics data, such as genomics or proteomics, has started. T-cell specific immunotherapy (106) *via* seeking for neo-antigens is the next bet of the 21st century. In conclusion, the vast amounts of daily generated medical data, the clinical unmet needs, the complexity of rare and common diseases and the patient's center strategies applied in most biomedical institutions, point out an emerging need for handling medical data in the most efficient way. ML approaches are necessary tools to be employed in every aspect of medical clinical practice where specific research methodologies need to be applied, capable of optimizing the current health policies and promoting the transition towards the precision medicine era.

Supplementary Material

We provide a handbook of extra supplemental material and examples of code in Python for the most common ML and Bioinformatic analysis. This can be found under the URL: https://www.gorgoulis.gr/images/Supplementary_Material_Pezoulas_et_al.pdf

Conflicts of Interest

The Authors declare no conflicts of interest.

Authors' Contributions

V.C.P., O.H., N.L. wrote the original draft and prepared the original figures. V.C.P., O.H., N.L., T.P.E., A.V.G. wrote, reviewed and edited the manuscript as well as assisted in literature search. A.G.T., D.I.F., I.G.S. supervised the preparation of the subsections, A.N.Y. and V.G.G. conseptualized and supervised the process of all the study and the manuscript preparation. V.G.G. achieved funding acquisition. All Authors have read and agreed to the published version of the manuscript.

Acknowledgements

VGG and his colleagues received financial support from the following grants: National Public Investment Program of the Ministry of Development and Investment/General Secretariat for Research and Technology, in the framework of the Flagship Initiative to address SARS-CoV-2 (2020ΣΕ01300001); Horizon 2020 Marie Sklodowska-Curie training program no. 722729 (SYNTRAIN); Welfare Foundation for Social & Cultural Sciences, Athens, Greece (KIKPE); H. Pappas donation; Hellenic Foundation for Research and Innovation (HFRI) grants no. 775 and 3782, NKUA-SARG grant 70/3/8916 and H. Pappas donation. This study was also co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE - INNOVATE (project code: T2EDK-02939).

References

1 GWAS Catalog. Available at: https://www.ebi.ac.uk/gwas/ [Last accessed on July 26, 2021]

2 Voshall A and Moriyama EN: Next-generation transcriptome assembly: strategies and performance analysis. Bioinformatics in the Era of Post Genomics and Big Data. London, IntechOpen, pp. 15-36, 2018.

3 Robinson MD, McCarthy DJ and Smyth GK: edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics *26(1)*: 139-140, 2010. PMID: 19910308. DOI: 10.1093/bioinformatics/btp616

4 Kravvaritis DC and Yannacopoulos AN: Variational methods in nonlinear analysis: with applications in optimization and partial differential equations. Walter De Gruyter Gmbh & Co Kg, 2020. DOI: 10.1515/9783110647389

5 Kuhn HW and Tucker AW: Nonlinear programming. Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, Berkeley and Los Angeles, pp. 481-492, 1951.

6 Steinwart I, Hush D and Scovel C: An explicit description of the reproducing Kernel Hilbert spaces of Gaussian RBF kernels. IEEE Transactions on Information Theory *52(10)*: 4635-4643, 2017. DOI: 10.1109/TIT.2006.881713

7 Brata Chanda P and Kumar Sarkar S: Detection and classification technique of breast cancer using multi kernal SVM classifier approach, 2018. IEEE Applied Signal Processing Conference (ASPCON), 7-9 December, 2018. DOI: 10.1109/ASPCON.2018.8748810

8 Yan X and Su X: Linear regression analysis: theory and computing. World Scientific, New Jersey, USA, 2009. DOI: 10.1142/6986

9 Golub G, Hansen P and O'leary D: Tikhonov regularization and total least squares. SIAM Journal on Matrix Analysis and Applications *21(1)*: 185-194, 2017. DOI: 10.1137/S0895479897326432

10 Vasquez MM, Hu C, Roe DJ, Chen Z, Halonen M and Guerra S: Least absolute shrinkage and selection operator type methods for the identification of serum biomarkers of overweight and obesity: simulation and application. BMC Med Res Methodol *16(1)*: 154, 2016. PMID: 27842498. DOI: 10.1186/s12874-016-0254-8

11 Carreira-Perpinán MA and Goodhill GJ: Generalised elastic nets. arXiv: 1108.2840, 2011.

12 Lever J, Krzywinski M and Altman N: Logistic regression. Nature Methods *13(7)*: 541-542, 2019. DOI: 10.1038/nmeth.3904

13 Maximum Likelihood Estimation (MLE) - MLE under model misspecification, Lecture 16, Introduction to Statistical Inference, Stanford, Autumn 2016.

14 Lovén J, Orlando DA, Sigova AA, Lin CY, Rahl PB, Burge CB, Levens DL, Lee TI and Young RA: Revisiting global gene expression analysis. Cell *151(3)*: 476-482, 2012. PMID: 23101621. DOI: 10.1016/j.cell.2012.10.012

15 Hafemeister C and Satija R: Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. Genome Biol *20(1)*: 296, 2019. PMID: 31870423. DOI: 10.1186/s13059-019-1874-1

16 Sammut C and Webb GI: Naïve Bayes. In: Encyclopedia of Machine Learning. Sammut C, Webb GI (eds). Springer, Boston, MA, USA, pp. 713-714, 2010. DOI: 10.1007/978-0-387-30164-8

17 Leondes CT: The Maximum A Posteriori (MAP) rule. Computer Techniques and Algorithms in Digital Signal Processing: Advances in Theory and Applications, Volume 75, Academic Press, Elsevier, USA, 1996.

18 Azizi E, Prabhakaran S, Carr A and Pe'er D: Bayesian inference for single-cell clustering and imputing. Genomics Comput Biol *3(1)*: e46, 2017.

19 Elmachtoub A, Liang JCN and McNellis R: Decision trees for decision-making under the predict-then-optimize framework. In International Conference on Machine Learning, PMLR, 2858-2867, 2020.

20 Lee T, Ullah A and Wang R: Bootstrap aggregating and random forest. Macroeconomic Forecasting in the Era of Big Data: 389-429, 2019. DOI: 10.1007/978-3-030-31150-6_13

21 Sakellaropoulos T, Vougas K, Narang S, Koinis F, Kotsinas A, Polyzos A, Moss TJ, Piha-Paul S, Zhou H, Kardala E, Damianidou E, Alexopoulos LG, Aifantis I, Townsend PA, Panayiotidis MI, Sfikakis P, Bartek J, Fitzgerald RC, Thanos D, Mills Shaw KR, Petty R, Tsirigos A and Gorgoulis VG: A deep learning framework for predicting response to therapy in cancer. Cell Rep *29(11)*: 3367-3373.e4, 2019. PMID: 31825821. DOI: 10.1016/j.celrep.2019.11.017

22 Mukherjee N, Calviello L, Hirsekorn A, de Pretis S, Pelizzola M and Ohler U: Integrative classification of human coding and noncoding genes through RNA metabolism profiles. Nat Struct Mol Biol *24(1)*: 86-96, 2017. PMID: 27870833. DOI: 10.1038/nsmb.3325

23 Xu B, Wang N, Chen T and Li M: Empirical evaluation of rectified activations in convolutional network. arXiv: 1505.00853, 2015.

24 Gao B and Pavel L: On the properties of the softmax function with application in game theory and reinforcement learning. arXiv: 1704.00805, 2017.

25 Bishop CM: Pattern recognition and machine learning. Springer, New York, USA, 2006.

26 Peña JM, Lozano JA and Larrañaga P: An empirical comparison of four initialization methods for the k-means algorithm. Pattern Recognit Lett *20(10):* 1027-1040, 1999. DOI: 10.1016/S0167-8655(99)00069-0

27 Oguntunde P, Khaleel M, Ahmed M and Okagbue H: The Gompertz FrÃ©chet distribution: Properties and applications. Cogent Mathematics & Statistics *6(1)*: 1568662, 2021. DOI: 10.1080/25742558.2019.1568662

28 Hicks S, Liu R, Ni Y, Purdom E and Risso D: mbkmeans: Fast clustering for single cell data using mini-batch k-means. PLOS Computational Biology *17(1)*: e1008625, 2021. DOI: 10.1371/journal.pcbi.1008625

29 Hua J, Liu H, Zhang B and Jin S: LAK: Lasso and K-Means Based Single-Cell RNA-Seq Data Clustering Analysis. IEEE Access *8*: 129679-129688, 2021. DOI: 10.1109/ACCESS.2020.3008681

30 Altman NS: An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat *46(3)*: 175-185, 1992.

31 Tomczak K, Czerwińska P and Wiznerowicz M: The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. Contemp Oncol (Pozn) *19(1A)*: A68-A77, 2015. PMID: 25691825. DOI: 10.5114/wo.2014.47136

32 Salk JJ, Schmitt MW and Loeb LA: Enhancing the accuracy of next-generation sequencing for detecting rare and subclonal mutations. Nat Rev Genet *19(5)*: 269-285, 2018. PMID: 29576615. DOI: 10.1038/nrg.2017.117

33 Rambout X, Dequiedt F and Maquat LE: Beyond transcription: Roles of transcription factors in pre-mRNA splicing. Chem Rev *118(8)*: 4339-4364, 2018. PMID: 29251915. DOI: 10.1021/acs.chemrev.7b00470

34 Park PJ: ChIP-seq: advantages and challenges of a maturing technology. Nat Rev Genet *10(10)*: 669-680, 2009. PMID: 19736561. DOI: 10.1038/nrg2641

35 Magri MS, Jiménez-Gancedo S, Bertrand S, Madgwick A, Escrivà H, Lemaire P and Gómez-Skarmeta JL: Assaying chromatin accessibility using ATAC-Seq in invertebrate chordate embryos. Front Cell Dev Biol *7*: 372, 2020. PMID: 32039199. DOI: 10.3389/fcell.2019.00372

36 Alipanahi B, Delong A, Weirauch MT and Frey BJ: Predicting the sequence specificities of DNA- and RNA-binding proteins by

deep learning. Nat Biotechnol *33(8)*: 831-838, 2015. PMID: 26213851. DOI: 10.1038/nbt.3300

37 Shen Z, Bao W and Huang DS: Recurrent neural network for predicting transcription factor binding sites. Sci Rep *8(1)*: 15270, 2018. PMID: 30323198. DOI: 10.1038/s41598-018-33321-1

38 Kopp W, Monti R, Tamburrini A, Ohler U and Akalin A: Deep learning for genomics using Janggu. Nat Commun *11(1)*: 3488, 2020. PMID: 32661261. DOI: 10.1038/s41467-020-17155-y

39 Zhou J and Troyanskaya OG: Predicting effects of noncoding variants with deep learning-based sequence model. Nat Methods *12(10)*: 931-934, 2015. PMID: 26301843. DOI: 10.1038/nmeth.3547

40 Meuleman W, Muratov A, Rynes E, Halow J, Lee K, Bates D, Diegel M, Dunn D, Neri F, Teodosiadis A, Reynolds A, Haugen E, Nelson J, Johnson A, Frerker M, Buckley M, Sandstrom R, Vierstra J, Kaul R and Stamatoyannopoulos J: Index and biological spectrum of human DNase I hypersensitive sites. Nature *584(7820)*: 244-251, 2020. PMID: 32728217. DOI: 10.1038/s41586-020-2559-3

41 Ashoor H, Chen X, Rosikiewicz W, Wang J, Cheng A, Wang P, Ruan Y and Li S: Graph embedding and unsupervised learning predict genomic sub-compartments from HiC chromatin interaction data. Nat Commun *11(1)*: 1173, 2020. PMID: 32127534. DOI: 10.1038/s41467-020-14974-x

42 van Berkum NL, Lieberman-Aiden E, Williams L, Imakaev M, Gnirke A, Mirny LA, Dekker J and Lander ES: Hi-C: a method to study the three-dimensional architecture of genomes. J Vis Exp *(39)*: 1869, 2010. PMID: 20461051. DOI: 10.3791/1869

43 Cheng C, Chou F, Kladwang W, Tian S, Cordero P and Das R: MOHCA-seq: RNA 3D models from single multiplexed proximity-mapping experiments. BioRxiv: 004556, 2014. DOI: 10.1101/004556

44 Flynn RA, Zhang QC, Spitale RC, Lee B, Mumbach MR and Chang HY: Transcriptome-wide interrogation of RNA secondary structure in living cells with icSHAPE. Nat Protoc *11(2)*: 273-290, 2016. PMID: 26766114. DOI: 10.1038/nprot.2016.011

45 Sharma E, Sterne-Weiler T, O'Hanlon D and Blencowe BJ: Global mapping of human RNA-RNA interactions. Mol Cell *62(4)*: 618-626, 2016. PMID: 27184080. DOI: 10.1016/j.molcel.2016.04.030

46 Foley SW and Gregory BD: Protein interaction profile sequencing (PIP-seq). Curr Protoc Mol Biol *116*: 27.5.1-27.5.15, 2016. PMID: 27723083. DOI: 10.1002/cpmb.21

47 Hafner M, Landthaler M, Burger L, Khorshid M, Hausser J, Berninger P, Rothballer A, Ascano M Jr, Jungkamp AC, Munschauer M, Ulrich A, Wardle GS, Dewell S, Zavolan M and Tuschl T: Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. Cell *141(1)*: 129-141, 2010. PMID: 20371350. DOI: 10.1016/j.cell.2010.03.009

48 Licatalosi DD, Mele A, Fak JJ, Ule J, Kayikci M, Chi SW, Clark TA, Schweitzer AC, Blume JE, Wang X, Darnell JC and Darnell RB: HITS-CLIP yields genome-wide insights into brain alternative RNA processing. Nature *456(7221)*: 464-469, 2008. PMID: 18978773. DOI: 10.1038/nature07488

49 Huppertz I, Attig J, D'Ambrogio A, Easton LE, Sibley CR, Sugimoto Y, Tajnik M, König J and Ule J: iCLIP: protein-RNA interactions at nucleotide resolution. Methods *65(3)*: 274-287, 2014. PMID: 24184352. DOI: 10.1016/j.ymeth.2013.10.011

50 Schueler M, Munschauer M, Gregersen LH, Finzel A, Loewer A, Chen W, Landthaler M and Dieterich C: Differential protein occupancy profiling of the mRNA transcriptome. Genome Biol *15(1)*: R15, 2014. PMID: 24417896. DOI: 10.1186/gb-2014-15-1-r15

51 Pan X and Shen HB: RNA-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach. BMC Bioinformatics *18(1)*: 136, 2017. PMID: 28245811. DOI: 10.1186/s12859-017-1561-8

52 ENCODE: Encyclopedia of DNA Elements. Available at: https://www.encodeproject.org/ [Last accessed on July 29, 2021]

53 Ghanbari M and Ohler U: Deep neural networks for interpreting RNA-binding protein target preferences. Genome Res *30(2)*: 214-226, 2020. PMID: 31992613. DOI: 10.1101/gr.247494.118

54 Budach S and Marsico A: pysster: classification of biological sequences by learning sequence and structure motifs with convolutional neural networks. Bioinformatics *34(17)*: 3035-3037, 2018. PMID: 29659719. DOI: 10.1093/bioinformatics/bty222

55 Grønning AGB, Doktor TK, Larsen SJ, Petersen USS, Holm LL, Bruun GH, Hansen MB, Hartung AM, Baumbach J and Andresen BS: DeepCLIP: predicting the effect of mutations on protein-RNA binding with deep learning. Nucleic Acids Res *48(13)*: 7099-7118, 2020. PMID: 32558887. DOI: 10.1093/nar/gkaa530

56 Maticzka D, Lange SJ, Costa F and Backofen R: GraphProt: modeling binding preferences of RNA-binding proteins. Genome Biol *15(1)*: R17, 2014. PMID: 24451197. DOI: 10.1186/gb-2014-15-1-r17

57 Janssen S and Giegerich R: The RNA shapes studio. Bioinformatics *31(3)*: 423-425, 2015. PMID: 25273103. DOI: 10.1093/bioinformatics/btu649

58 Smola AJ and Schölkopf B: A tutorial on support vector regression. Stat Comput *14(3)*: 199-222, 2004.

59 Šponer J, Bussi G, Krepl M, Banáš P, Bottaro S, Cunha RA, Gil-Ley A, Pinamonti G, Poblete S, Jurečka P, Walter NG and Otyepka M: RNA structural dynamics as captured by molecular simulations: a comprehensive overview. Chem Rev *118(8)*: 4177-4338, 2018. PMID: 29297679. DOI: 10.1021/acs.chemrev.7b00427

60 Boniecki MJ, Lach G, Dawson WK, Tomala K, Lukasz P, Soltysinski T, Rother KM and Bujnicki JM: SimRNA: a coarse-grained method for RNA folding simulations and 3D structure prediction. Nucleic Acids Res *44(7)*: e63, 2016. PMID: 26687716. DOI: 10.1093/nar/gkv1479

61 Li J, Zhu W, Wang J, Li W, Gong S, Zhang J and Wang W: RNA3DCNN: Local and global quality assessments of RNA 3D structures using 3D deep convolutional neural networks. PLoS Comput Biol *14(11)*: e1006514, 2018. PMID: 30481171. DOI: 10.1371/journal.pcbi.1006514

62 Zhang S, Zhou J, Hu H, Gong H, Chen L, Cheng C and Zeng J: A deep learning framework for modeling structural features of RNA-binding protein targets. Nucleic Acids Res *44(4)*: e32, 2016. PMID: 26467480. DOI: 10.1093/nar/gkv1025

63 Roll J, Zirbel CL, Sweeney B, Petrov AI and Leontis N: JAR3D Webserver: Scoring and aligning RNA loop sequences to known 3D motifs. Nucleic Acids Res *44(W1)*: W320-W327, 2016. PMID: 27235417. DOI: 10.1093/nar/gkw453

64 Parlea LG, Sweeney BA, Hosseini-Asanjan M, Zirbel CL and Leontis NB: The RNA 3D Motif Atlas: Computational methods for extraction, organization and evaluation of RNA motifs. Methods *103*: 99-119, 2016. PMID: 27125735. DOI: 10.1016/j.ymeth.2016.04.025

65 Fischer A and Igel C: An introduction to restricted Boltzmann machines. Progress in Pattern Recognition, Image Analysis,

Computer Vision, and Applications: 14-36, 2021. DOI: 10.1007/978-3-642-33275-3_2

66 Künsch H: Gaussian Markov random fields. Journal of the Faculty of Science, University of Tokyo, Section IA. Math *26(1)*: 53-73, 1979.

67 Lam JH, Li Y, Zhu L, Umarov R, Jiang H, Héliou A, Sheong FK, Liu T, Long Y, Li Y, Fang L, Altman RB, Chen W, Huang X and Gao X: A deep learning framework to predict binding preference of RNA constituents on protein surface. Nat Commun *10(1)*: 4941, 2019. PMID: 31666519. DOI: 10.1038/s41467-019-12920-0

68 Le Guilloux V, Schmidtke P and Tuffery P: Fpocket: an open source platform for ligand pocket detection. BMC Bioinformatics *10*: 168, 2009. PMID: 19486540. DOI: 10.1186/1471-2105-10-168

69 Levinthal C: How to fold graciously. Mossbauer Spectroscopy in Biological Systems: Proceedings of a meeting held at Allerton House, Monticello, Illinois: 22-24. Archived from the original on 2010-10-07, 1969.

70 Senior AW, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, Qin C, Žídek A, Nelson AWR, Bridgland A, Penedones H, Petersen S, Simonyan K, Crossan S, Kohli P, Jones DT, Silver D, Kavukcuoglu K and Hassabis D: Improved protein structure prediction using potentials from deep learning. Nature *577(7792)*: 706-710, 2020. PMID: 31942072. DOI: 10.1038/s41586-019-1923-7

71 Menden K, Marouf M, Oller S, Dalmia A, Magruder DS, Kloiber K, Heutink P and Bonn S: Deep learning-based cell composition analysis from tissue expression profiles. Sci Adv *6(30)*: eaba2619, 2020. PMID: 32832661. DOI: 10.1126/sciadv.aba2619

72 Ko ME, Williams CM, Fread KI, Goggin SM, Rustagi RS, Fragiadakis GK, Nolan GP and Zunder ER: FLOW-MAP: a graph-based, force-directed layout algorithm for trajectory mapping in single-cell time course datasets. Nat Protoc *15(2)*: 398-420, 2020. PMID: 31932774. DOI: 10.1038/s41596-019-0246-3

73 Angerer P, Haghverdi L, Büttner M, Theis FJ, Marr C and Buettner F: destiny: diffusion maps for large-scale single-cell data in R. Bioinformatics *32(8)*: 1241-1243, 2016. PMID: 26668002. DOI: 10.1093/bioinformatics/btv715

74 Gupta A, Wang *H* and Ganapathiraju M: Learning structure in gene expression data using deep architectures, with an application to gene clustering. In 2015 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE, pp. 1328-1335, 2015.

75 Lin C, Jain S, Kim *H* and Bar-Joseph Z: Using neural networks for reducing the dimensions of single-cell RNA-Seq data. Nucleic Acids Res *45(17)*: e156, 2017. PMID: 28973464. DOI: 10.1093/nar/gkx681

76 Rashid S, Shah S, Bar-Joseph Z and Pandya R: Dhaka: Variational autoencoder for unmasking tumor heterogeneity from single cell genomic data. Bioinformatics, 2019. PMID: 30768159. DOI: 10.1093/bioinformatics/btz095

77 Verma D and Meila M: A comparison of spectral clustering algorithms. University of Washington Tech Rep UWCSE030501, 1, 1-18, 2003.

78 Li WV and Li JJ: An accurate and robust imputation method scImpute for single-cell RNA-seq data. Nat Commun *9(1)*: 997, 2018. PMID: 29520097. DOI: 10.1038/s41467-018-03405-7

79 van Dijk D, Sharma R, Nainys J, Yim K, Kathail P, Carr AJ, Burdziak C, Moon KR, Chaffer CL, Pattabiraman D, Bierie B, Mazutis L, Wolf G, Krishnaswamy S and Pe'er D: Recovering gene interactions from single-cell data using data diffusion. Cell *174(3)*: 716-729.e27, 2018. PMID: 29961576. DOI: 10.1016/j.cell.2018.05.061

80 Huang M, Wang J, Torre E, Dueck H, Shaffer S, Bonasio R, Murray JI, Raj A, Li M and Zhang NR: SAVER: gene expression recovery for single-cell RNA sequencing. Nat Methods *15(7)*: 539-542, 2018. PMID: 29941873. DOI: 10.1038/s41592-018-0033-z

81 Arisdakessian C, Poirion O, Yunits B, Zhu X and Garmire LX: DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. Genome Biol *20(1)*: 211, 2019. PMID: 31627739. DOI: 10.1186/s13059-019-1837-6

82 Eraslan G, Simon LM, Mircea M, Mueller NS and Theis FJ: Single-cell RNA-seq denoising using a deep count autoencoder. Nat Commun *10(1)*: 390, 2019. PMID: 30674886. DOI: 10.1038/s41467-018-07931-2

83 Tipping ME and Bishop CM: Probabilistic principal component analysis. J R Stat Soc Series B Stat Methodol *61(3)*: 611-622, 1999.

84 Pierson E and Yau C: ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biol *16*: 241, 2015. PMID: 26527291. DOI: 10.1186/s13059-015-0805-z

85 Qiu X, Hill A, Packer J, Lin D, Ma YA and Trapnell C: Single-cell mRNA quantification and differential analysis with Census. Nat Methods *14(3)*: 309-315, 2017. PMID: 28114287. DOI: 10.1038/nmeth.4150

86 La Manno G, Soldatov R, Zeisel A, Braun E, Hochgerner H, Petukhov V, Lidschreiber K, Kastriti ME, Lönnerberg P, Furlan A, Fan J, Borm LE, Liu Z, van Bruggen D, Guo J, He X, Barker R, Sundström E, Castelo-Branco G, Cramer P, Adameyko I, Linnarsson S and Kharchenko PV: RNA velocity of single cells. Nature *560(7719)*: 494-498, 2018. PMID: 30089906. DOI: 10.1038/s41586-018-0414-6

87 Forney G: The viterbi algorithm. Proceedings of the IEEE *61(3)*: 268-278, 2017. DOI: 10.1109/PROC.1973.9030

88 Herzog VA, Reichholf B, Neumann T, Rescheneder P, Bhat P, Burkard TR, Wlotzka W, von Haeseler A, Zuber J and Ameres SL: Thiol-linked alkylation of RNA to assess expression dynamics. Nat Methods *14(12)*: 1198-1204, 2017. PMID: 28945705. DOI: 10.1038/nmeth.4435

89 Qiu X, Zhang Y, Hosseinzadeh S, Yang D, Pogson A, Wang L, Shurtleff M, Yuan R, Xu S, Ma Y, Replogle J, Darmanis S, Bahar I, Xing J and Weissman J: Mapping transcriptomic vector fields of single cells, 2021. DOI: 10.1101/696724

90 Fu L, Zhang L, Dollinger E, Peng Q, Nie Q and Xie X: Predicting transcription factor binding in single cells through deep learning. Sci Adv *6(51)*: eaba9031, 2020. PMID: 33355120. DOI: 10.1126/sciadv.aba9031

91 Cao J, Cusanovich DA, Ramani V, Aghamirzaie D, Pliner HA, Hill AJ, Daza RM, McFaline-Figueroa JL, Packer JS, Christiansen L, Steemers FJ, Adey AC, Trapnell C and Shendure J: Joint profiling of chromatin accessibility and gene expression in thousands of single cells. Science *361(6409)*: 1380-1385, 2018. PMID: 30166440. DOI: 10.1126/science.aau0730

92 Hao Y, Hao S, Andersen-Nissen E, Mauck WM 3rd, Zheng S, Butler A, Lee MJ, Wilk AJ, Darby C, Zager M, Hoffman P, Stoeckius M, Papalexi E, Mimitou EP, Jain J, Srivastava A, Stuart T, Fleming LM, Yeung B, Rogers AJ, McElrath JM, Blish CA, Gottardo R, Smibert P and Satija R: Integrated analysis of multimodal single-cell data. Cell *184(13)*: 3573-3587.e29, 2021. PMID: 34062119. DOI: 10.1016/j.cell.2021.04.048

93 Stoeckius M, Hafemeister C, Stephenson W, Houck-Loomis B, Chattopadhyay PK, Swerdlow H, Satija R and Smibert P: Simultaneous epitope and transcriptome measurement in single cells. Nat Methods *14(9)*: 865-868, 2017. PMID: 28759029. DOI: 10.1038/nmeth.4380

94 Cooley S, Hamilton T, Ray J and Deeds E: A novel metric reveals previously unrecognized distortion in dimensionality reduction of scRNA-Seq data. BioRxiv, 2020. DOI: 10.1101/689851

95 COSMIC, the Catalogue of Somatic Mutations in Cancer, Available at: https://cancer.sanger.ac.uk/cosmic [Last accessed on July 29, 2021]

96 cBioPortal for cancer genomics. Available at: https://slack.cbioportal.org [Last accessed on July 29, 2021]

97 National Cancer Institute's Clinical Proteomic Tumor Analysis Consortium (CPTAC). Available at: https://proteomics.cancer.gov/programs/cptac [Last accessed on July 29, 2021]

98 Vougas K, Sakellaropoulos T, Kotsinas A, Foukas GP, Ntargaras A, Koinis F, Polyzos A, Myrianthopoulos V, Zhou H, Narang S, Georgoulias V, Alexopoulos L, Aifantis I, Townsend PA, Sfikakis P, Fitzgerald R, Thanos D, Bartek J, Petty R, Tsirigos A and Gorgoulis VG: Machine learning and data mining frameworks for predicting drug response in cancer: An overview and a novel in silico screening process based on association rule mining. Pharmacol Ther *203*: 107395, 2019. PMID: 31374225. DOI: 10.1016/j.pharmthera.2019.107395

99 Kotsiantis S and Kanellopoulos D: Association rules mining: A recent overview. GESTS Int Trans Comput Sci Eng *32(1)*: 71-82, 2006.

100 Zou *H* and Hastie T: Regularization and variable selection *via* the elastic net. J R Stat Soc Series B Stat Methodol *67(2)*: 301-320, 2005.

101 Harrison B, Anderson KC, Raje N, Richardson P, Warren D, Chari A and Giusti K: The Multiple Myeloma Research Consortium (MMRC): A model for accelerating development of novel therapies for multiple myeloma. Blood *122(21):* 5388, 2013, DOI: 10.1182/blood.V122.21.5388.5388

102 Kuenzi BM, Park J, Fong SH, Sanchez KS, Lee J, Kreisberg JF, Ma J and Ideker T: Predicting drug response and synergy using a deep learning model of human cancer cells. Cancer Cell *38(5)*: 672-684.e6, 2020. PMID: 33096023. DOI: 10.1016/j.ccell.2020.09.014

103 Capecchi A, Probst D and Reymond JL: One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. J Cheminform *12(1)*: 43, 2020. PMID: 33431010. DOI: 10.1186/s13321-020-00445-4

104 Bowden R, Davies RW, Heger A, Pagnamenta AT, de Cesare M, Oikkonen LE, Parkes D, Freeman C, Dhalla F, Patel SY, Popitsch N, Ip CLC, Roberts HE, Salatino S, Lockstone H, Lunter G, Taylor JC, Buck D, Simpson MA and Donnelly P: Sequencing of human genomes with nanopore technology. Nat Commun *10(1)*: 1869, 2019. PMID: 31015479. DOI: 10.1038/s41467-019-09637-5

105 Ni P, Huang N, Zhang Z, Wang DP, Liang F, Miao Y, Xiao CL, Luo F and Wang J: DeepSignal: detecting DNA methylation state from Nanopore sequencing reads using deep-learning. Bioinformatics *35(22)*: 4586-4595, 2019. PMID: 30994904. DOI: 10.1093/bioinformatics/btz276

106 Waldman AD, Fritz JM and Lenardo MJ: A guide to cancer immunotherapy: from T cell basic science to clinical practice. Nat Rev Immunol *20(11)*: 651-668, 2020. PMID: 32433532. DOI: 10.1038/s41577-020-0306-5

Received June 25, 2021
Revised July 21, 2021
Accepted August 3, 2021