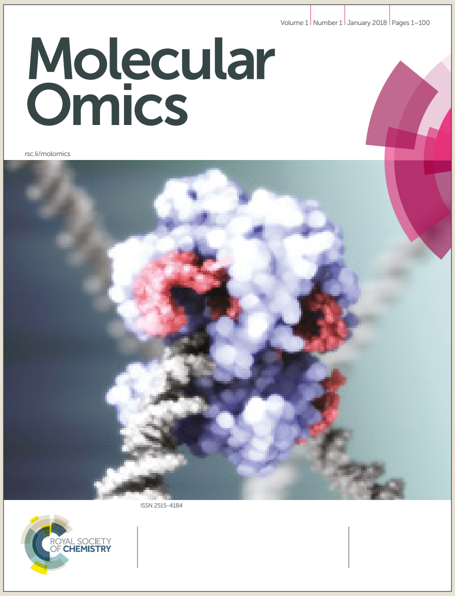
This article can be cited before page numbers have been issued, to do this please use: B. C. Feltes, B. I. Grisci, J. D. F. de Faria Poloni and M. Dorn, *Mol. Omics*, 2018, DOI: 10.1039/C8MO00111A.


Volume 1 | Number 1 | January 2018 | Pages 1–100
Molecular Omics
rsc.li/molomics
ISSN 2515-4184
ROYAL SOCIETY OF CHEMISTRY

View Article Online
DOI: 10.1039/C8MO00111A

Journal Name

ARTICLE TYPE

Cite this: DOI: 10.1039/xxxxxxxxxx

Perspectives and Applications of Machine Learning for Evolutionary Developmental Biology

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

Received Date
Accepted Date

DOI: 10.1039/xxxxxxxxxx

www.rsc.org/journalname

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

Introduction

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

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

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

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

View Article Online
DOI: 10.1039/C8MO00111A

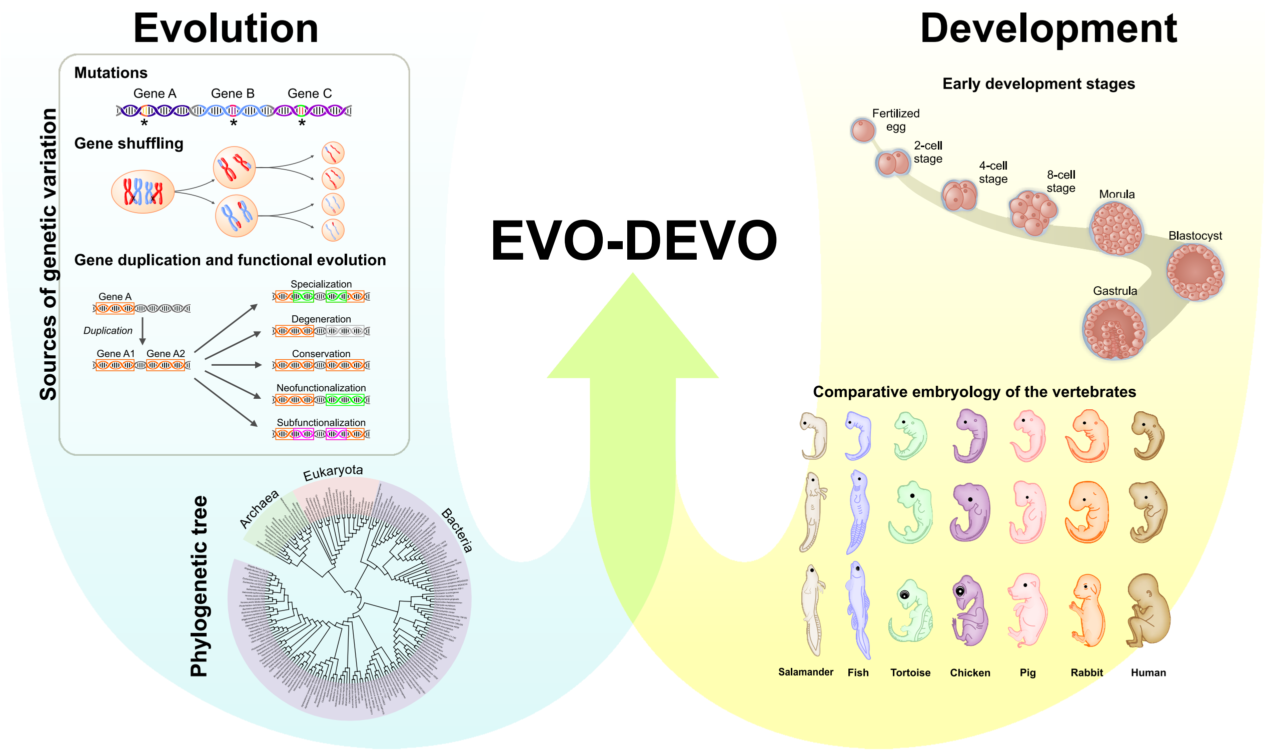


Fig. 1 A simplified illustration of the study of Evo-Devo, representing the integration of developmental processes and the evolutionary origin of phenotypic changes between organisms. The study of development is intrinsically related to the evolutionary process, and evolution-related events, reproduction,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

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

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

A Glance on Evo-Devo Thinking in the Last Decades

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

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

The phenotype is controlled by distinct regulation levels of

View Article Online
DOI: 10.1039/C8MO00111A

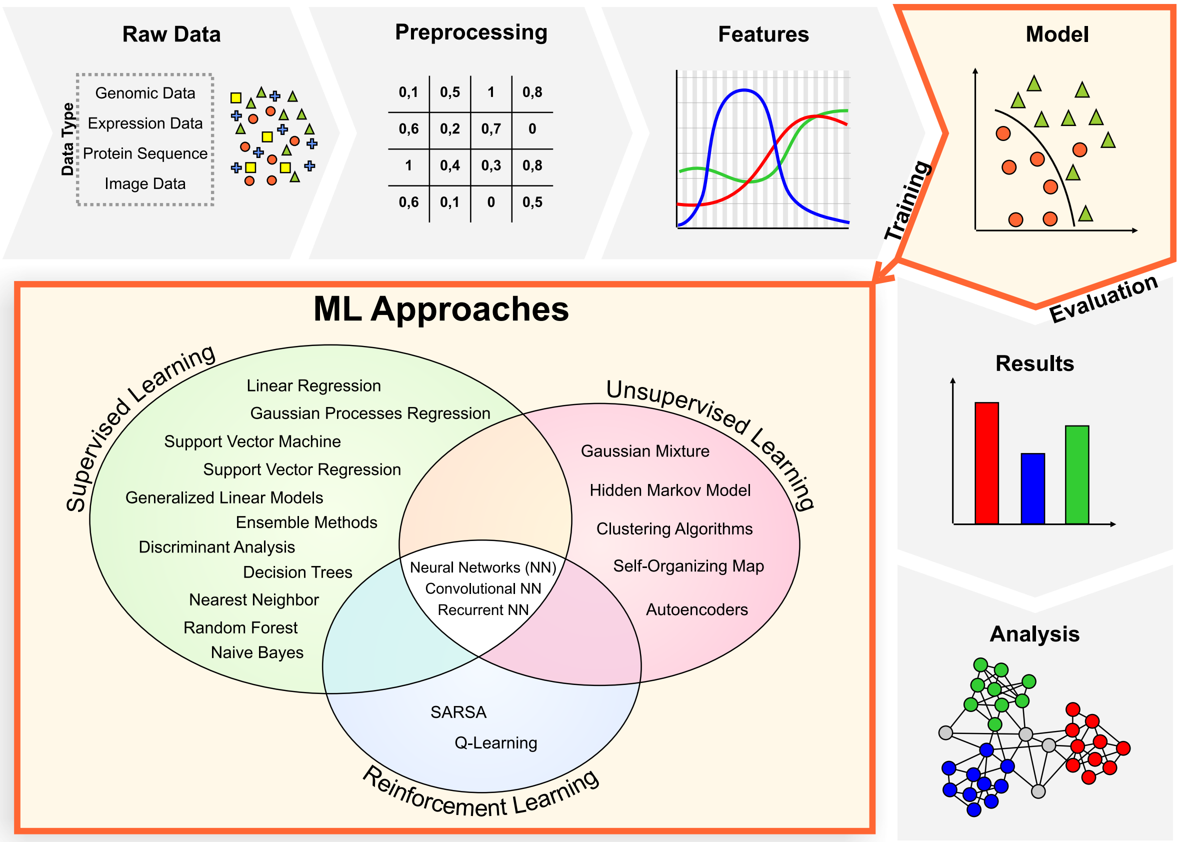


Fig. 2 A summary of ML workflow with a schematic of a generic method and its algorithms. The raw data is obtained from measurements and experiments and is preprocessed to be applied in the ML pipeline. This step can involve cleaning, outlier removal, normalization, standardization, frequency balancing,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

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

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

Brief Overview of Machine Learning Techniques

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

View Article Online
DOI: 10.1039/C8MO00111A

Neural Networks

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

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

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

Decision Trees

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

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

Support Vector Machines

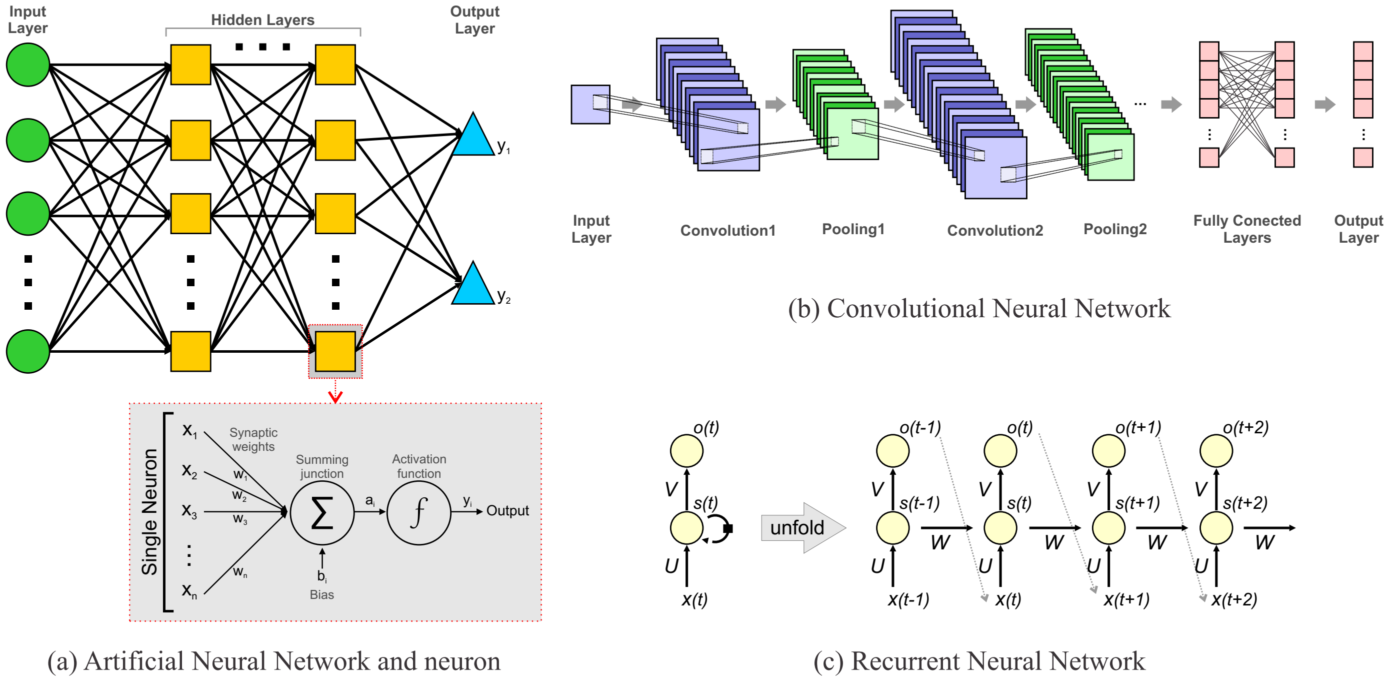
Support Vector Machines (SVM)[54] are classifiers that work by finding the line (in 2D), plane (in 3D), or hyperplane (in larger dimensions) capable of splitting data into distinct classes. This "divider" is called a separating hyperplane and works as a decision boundary, as illustrated in Fig. 5a. The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

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

Genetic Algorithms

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

View Article Online
DOI: 10.1039/C8MO00111A



(b) Convolutional Neural Network

(a) Artificial Neural Network and neuron

(c) Recurrent Neural Network

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

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

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

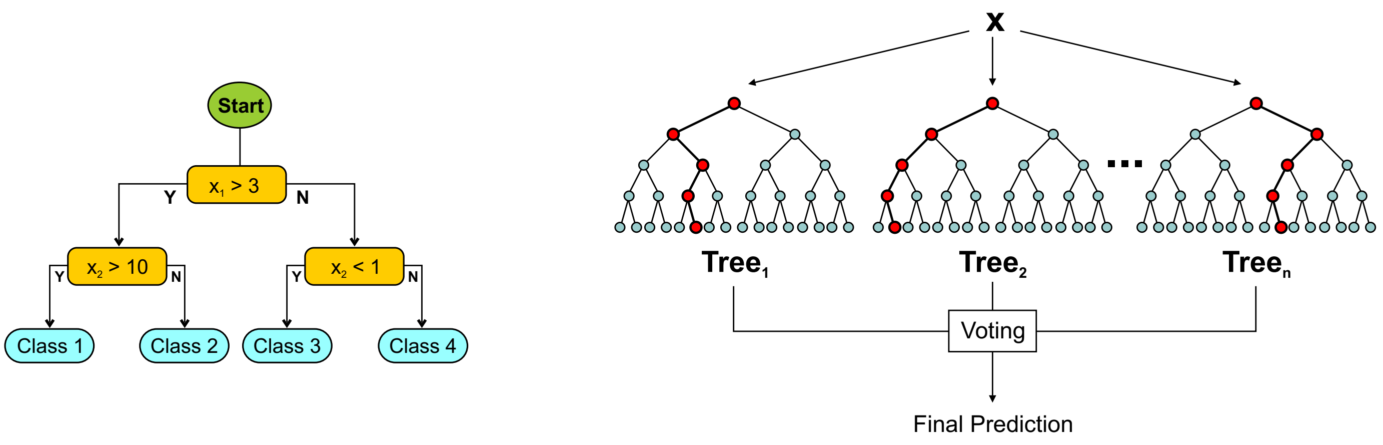
Machine Learning Applied to Development and Evolution

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

Machine Learning, Evo-Devo and Genomics

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

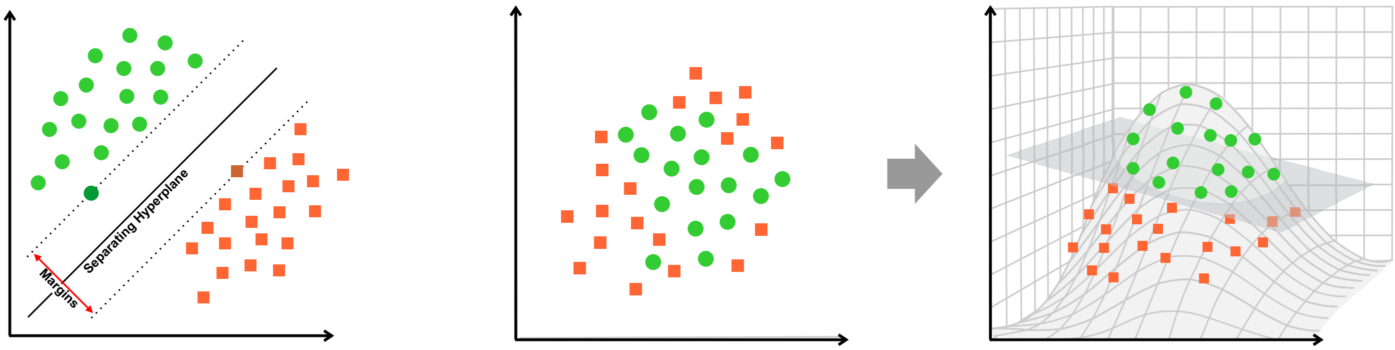
View Article Online
DOI: 10.1039/C8MO00111A



(a) Decision Tree

(b) Random Forest

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



(a) Support Vector Machines

(b) Kernel transformation to higher dimension

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

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

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

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

View Article Online
DOI: 10.1039/C8MO00111A

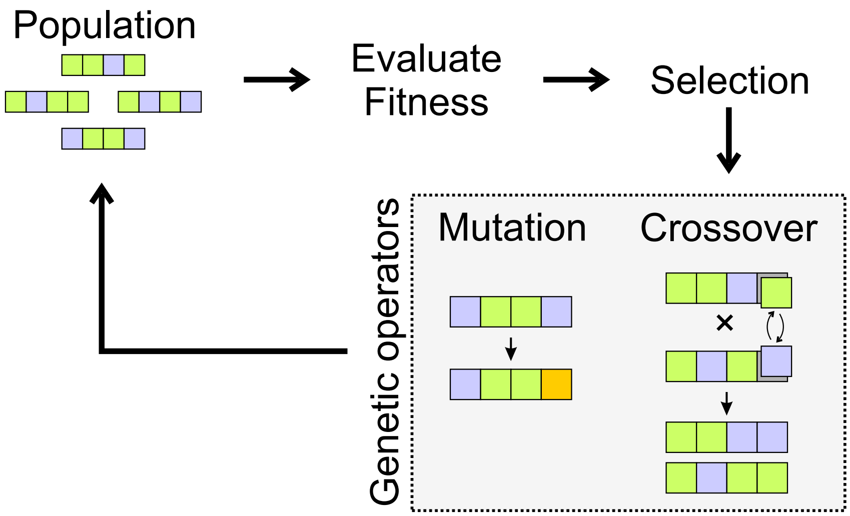


Fig. 6 Schematic of a simple GA pipeline. A population of random individuals is generated, each of them representing a candidate solution. These individuals are evaluated by some domain-specific metric and, based on that, selected. The selected individuals can be subjugated to crossover or mutation operators,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

cell types[70] (Table 1). WEKA[71] implementation of RF was chosen after it obtained the best results on a comparison of the 10-fold cross validation for 10 sampled datasets against other four algorithms: SVM with Radial Basis Function (RBF)[72] as kernel, decision tree J48 (also known as C4.5)[73], naive Bayes[14], and logistic regression[29]. The authors satisfyingly predicted methylation patterns, pointing histone modifications related to specific cell types and genomic regions. During development, chromatin regions display a dynamic and complex regulation that affects the transcriptional expression of patterning genes, especially *HOX*, shaping and modulating tissue and limb development[74]. Predicting methylation patterning shows a promising application for ML in epigenetics, by aiding to unravel chromatin dynamics.

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

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

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

* http://www.internationalgenome.org

View Article Online
DOI: 10.1039/C8MO00111A

Machine Learning, Evo-Devo and Protein Data

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

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

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

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

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

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

View Article Online
DOI: 10.1039/C8MO00111A

alignment of a non-redundant set of proteins. The data generated by the authors helped elucidate how protein sequence and function co-evolved and how ML can be applied to both protein and phylogenetic data.

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

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

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

New grounds to explore: Morphometric data has joined the party

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

Morphogenesis is molded by mechanical forces that stimulate the movement and deformation of an element, according to its resistance[118]. These mechanical forces can be promoted by different sources, such as biophysical alterations in the local environment. Different mechanical forces are involved in development, such as osmotic pressure, shear stress, tensional forces, surface tension and spring forces[119]. Furthermore, the environment offers a great source of variability, and in an ecological context, the major influences, like the developmental temperature, chemical environment, and egg or embryo size, can affect embryonic morphogenesis[118]. These forces drive embryo shape, triggering the deformation of cells and tissues that give rise to the form and phenotype of the organism[120]. Cells are able to sense and respond to external forces and transduce these signals to the molecular machinery, expressing genes that regulate the cell fate[120]. Moreover, the cells that compose an organism are driven by a bioelectric sig-

View Article Online
DOI: 10.1039/C8MO00111A

naling network, and thus are able to regulate pattern formation and direct the growth and form of different tissues[121]. These external influences may be converted into signals and translated to a stimulus that influences morphogenesis in different scales of time and space. The interesting aspect of this new side of morphological studies is its mathematical background, making it a perfect target for ML.

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

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

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

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

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

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

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

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

View Article Online
DOI: 10.1039/C8MO00111A

Published on 13 August 2018. Downloaded on 8/14/2018 12:05:44 AM.

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

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

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

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

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

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

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

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

DOI: 10.1039/C8MO00111A

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

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

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

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

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

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

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

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

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

Moreover, inspired by NEAT, Cussat-Blanc *et al.* created a new

[†] www.ncbi.nlm.nih.gov/gds/

[‡] www.rcsb.org

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

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

algorithm for the training of artificial gene regulatory networks (AGRNs), dynamical systems used in the control of agents, called GRNEAT[154]. This approach allowed the design of better AGRNs than regular GA and evolutionary programming strategies for the used benchmarks. Lones has a complete review on the use of AGRNs in computational problems[155].

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

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

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

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

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

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

In addition, most works in this review used ML to perform supervised learning for classification tasks, and many challenges arise from the use of Evo-Devo data or biological data in general with this goal. One of the major concerns is the "Curse of Dimensionality", when the data has a large number of dimensions, as can be seen in microarray data or collections of pictures and videos. High dimensional data is often associated with overfitting in ML algorithms, higher processing costs and runtime, increase in memory consumption, and difficulty in visualization. One way to avoid overfitting is to expand the dataset by performing new experiments, but this can be expensive and time-consuming. The addition of artificially generated data should be considered only after great consideration since it could add arbitrary values that

View Article Online
DOI: 10.1039/C8MO00111A

should otherwise represent real-world phenomena. Another option, commonly used with ANNs is the incorporation of some type of regularization in the construction of the method. The works of Gonçalves *et al.* may also provide some guidance in regard of overfitting in evolutionary algorithms[169,170].

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

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

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

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

Conclusions

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

Key Points

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

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

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

Notes and references

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

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

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

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

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

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

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

View Article Online
DOI: 10.1039/C8MO00111A

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

View Article Online
DOI: 10.1039/C8MO00111A

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

116 E. M. De Robertis, Y. Moriyama and C. G, *Dev Growth Differ*, 2017, **59**, 580–592.

117 A. Wanninger, *Frontiers in Ecology and Evolution*, 2015, **3**, 1–9.

118 M. von Dassow and L. A. Davidson, *Phys Biol*, 2011, **8**, 045002.

119 T. Mammoto and D. E. Ingber, *Development*, 2010, **137**, 1407–1420.

View Article Online
DOI: 10.1039/C8MO00111A

120 C. J. Miller and D. L. A, *Nat Rev Genet*, 2013, **14**, 733–744.

121 M. Levin and C. J. Martyniuk, *Biosystems*, 2018, **164**, 76–93.

122 B. Hallgrimsson, C. Percival, R. Green, N. Young, W. Mio *et al.*, *Curr Top Dev Biol*, 2015, **115**, 561–597.

123 M. Masaeli, D. Gupta, S. O'Byrne, H. Tse, D. Gossett *et al.*, *Sci Rep*, 2016, **6**, 37863.

124 z. Cai and S. Ge, *Journal of Systematics and Evolution*, 2017, **55**, 377–384.

125 S. T. Namin, M. Esmaeilzadeh, N. M *et al.*, *bioRxiv*, 2017, **doi: https://doi.org/10.1101/134205**, year.

126 F. Ning, D. Delhomme, Y. LeCun *et al.*, *IEEE Trans Image Process*, 2005, **14**, 1360–1371.

127 D. Lobo, M. Lobikin and M. Levin, *Sci Rep*, 2017, **7**, 41339.

128 M. Lobikin, D. Lobo, D. J. Blackiston *et al.*, *Sci Signal*, 2015, **8**, ra99.

129 D. Lobo and M. Levin, *PLoS Comput Biol*, 2015, **11**, e1004295.

130 A. Spirov and D. Holloway, *Methods*, 2013, **62**, 39–55.

131 D. Aguilar-Hidalgo, M. Lemos and A. Córdoba, *Computation*, 2015, **3**, 99–113.

132 P. FranÃğois, *Semin Cell Dev Biol*, 2014, **35**, 90–97.

133 J. Murray, *Wiley Interdiscip Rev Dev Biol*, 2018, **7**, e314.

134 H. Parker, I. Pushel and R. Krumlauf, *Dev Biol*, 2018, **pii: S0012-1606**, 30597–3.

135 M. Das Gupta and M. Tsiantis, *Curr Opin Plant Biol*, 2018, **45**, 82–87.

136 J. Fernández, F. Vico and R. Doursat, *Complex and diverse morphologies can develop from a minimal genomic model*, 2012.

137 S. Kriegman, N. Cheney and J. Bongard, *arXiv:1711.07387*.

138 J. Hiller and H. Lipson, *Soft robotics*, 2014, **1**, 88–101.

139 M. Schmidt and H. Lipson, *Genetic Programming Theory and Practice VIII*, Springer, 2011, pp. 129–146.

140 E. Williams, J. Moore, S. Li, G. Rustici, A. Tarkowska *et al.*, *Nat Methods*, 2017, **14**, 775–781.

141 K. Kouvaris, J. Clune, L. Kounios *et al.*, *PLoS Comput Biol*, 2017, **13**, e1005358.

142 R. A. Watson, R. Mills, C. L. Buckley *et al.*, *Evol Biol*, 2016, **43**, 553–581.

143 M. Sipper, R. S. Olson and J. H. Moore, *BioData Min*, 2017, **10**, 26.

144 W. Banzhaf, P. Nordin, R. E. Keller *et al.*, *Genetic programming: an introduction*, Morgan Kaufmann, San Francisco, 1998, vol. 1.

145 R. Storn and K. Price, *J Global Opt*, 1997, **11**, 341–359.

146 S. Luke, *Essentials of Metaheuristics*, Lulu, Morrisville, North Carolina, Second Edition edn, 2013.

147 D. Floreano, P. Dürr and C. Mattiussi, *Evol Intel*, 2008, **1**, 47–62.

148 K. O. Stanley and R. Miikkulainen, *Evol Comput*, 2002, **10**, 99.

149 E. Dufourq and B. A. Bassett, *arXiv preprint arXiv:1707.00703*, 2017.

150 E. Dufourq and B. A. Bassett, *arXiv preprint arXiv:1709.09161*, 2017.

151 S. Ding, H. Li, C. Su *et al.*, *Artif Intell Rev*, 2013, 1.

152 A. Fontana, A. Soltoggio and B. Wróbel, *POET: an evo-devo method to optimize the weights of a large artificial neural networks*, 2014.

153 A. Fontana, European Conference on Artificial Life, 2009, pp. 10–17.

154 S. Cussat-Blanc, K. Harrington and J. Pollack, *IEEE Transactions on Evolutionary Computation*, 2015, **19**, 823–837.

155 M. A. Lones, *Evolutionary Computation in Gene Regulatory Network Research*, 2016, 398–424.

156 K. O. Stanley, *Genetic programming and evolvable machines*, 2007, **8**, 131–162.

157 K. O. Stanley, D. B. D'Ambrosio and J. Gauci, *Artificial life*, 2009, **15**, 185–212.

158 S. L. Beaulieu, S. Kriegman and J. C. Bongard, *arXiv preprint arXiv:1804.04286*, 2018.

159 R. M. French, *Trends in cognitive sciences*, 1999, **3**, 128–135.

160 I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville and Y. Bengio, *arXiv preprint arXiv:1312.6211*, 2013.

161 B. Chopard and M. Droz, *Cellular automata*, Springer, Amsterdam, The Netherlands, 1998.

162 S. Nichele, A. Giskeødegård and G. Tufte, *Artif Life*, 2016, **22**, 76–111.

163 B. Xu, H. Min and F. Xiao, *Ind Rob*, 2014, **41**, 527–533.

164 S. H. Kenyon, AAAI Fall Symposium Series, 15-17 November 2013, Arlington, Virginia, 2013.

165 M. W. Libbrecht and W. S. Noble, *Nat Rev Genet*, 2015, **16**, 321–332.

166 B. A. McKinney, D. M. Reif, M. D. Ritchie *et al.*, *Appl Bioinformatics*, 2005, **5**, 77–88.

167 S. Dudoit, J. Fridlyand and T. P. Speed, *Journal of the American statistical association*, 2002, **97**, 77–87.

168 A. Dupuy and R. M. Simon, *Journal of the National Cancer Institute*, 2007, **99**, 147–157.

169 I. Gonçalves, S. Silva, J. B. Melo and J. M. Carreiras, European Conference on Genetic Programming, 2012, pp. 218–229.

170 I. Gonçalves and S. Silva, European Conference on Genetic Programming, 2013, pp. 73–84.

171 R. Varshavsky, A. Gottlieb, M. Linial *et al.*, *Bioinformatics*, 2006, **22**, e507–e513.

172 P. Krızek, *PhD thesis*, PhD thesis, Czech Technical University in Prague, 2008. 6, 14, 36, 67, 93, 2008.

173 I. T. Jolliffe and J. Cadima, *Philos Trans A Math Phys Eng Sci*, 2016, **374**, 20150202.

174 V. Klema and A. Laub, *IEEE Trans Autom Control*, 1980, **25**, 164–176.

175 B. Fruchter, 1954.

176 L. van der Maaten and G. Hinton, *Journal of Machine Learning Research*, 2008, **9**, 2579–2605.

177 J. Miao and L. Niu, *Procedia Computer Science*, 2016, **91**, 919–926.

View Article Online
DOI: 10.1039/C8MO00111A

178 H. Ding, Cand Peng, *J Bioinform Comput Biol*, 2005, **3**, 185–205.

179 J. C. Ang, A. Mirzal, H. Haron *et al.*, *IEEE/ACM transactions on computational biology and bioinformatics*, 2016, **13**, 971–989.

180 S. Gu, E. Holly, T. Lillicrap *et al.*, Robotics and Automation (ICRA), 2017 IEEE International Conference on, 2017, pp. 3389–3396.

181 V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, *Nature*, 2015, **518**, 529.

182 D. Silver, J. Schrittwieser, K. Simonyan *et al.*, *Nature*, 2017, **550**, 354.

183 R. S. Sutton and A. G. Barto, *Reinforcement Learning : An Introduction*, MIT Press, Favoritenstrasse 9/4th Floor/1863, 1998.

184 Y. Li, F. X. Wu and A. Ngom, *Brief Bioinform*, 2016, **pii**, bbw113.