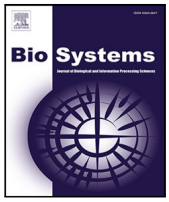
BioSystems 237 (2024) 105164

Contents lists available at ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems



Review article

Synthetic biological neural networks: From current implementations to future perspectives

Ana Halužan Vasle, Miha Moškon *

Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia

ARTICLE INFO

Keywords:
Neural networks
Perceptron
Synthetic biology
Molecular computing
Neuromorphic computing
Modelling and simulation

ABSTRACT

Artificial neural networks, inspired by the biological networks of the human brain, have become game-changing computing models in modern computer science. Inspired by their wide scope of applications, synthetic biology strives to create their biological counterparts, which we denote synthetic biological neural networks (SYNBIONNs). Their use in the fields of medicine, biosensors, biotechnology, and many more shows great potential and presents exciting possibilities. So far, many different synthetic biological networks have been successfully constructed, however, SYNBIONN implementations have been sparse. The latter are mostly based on neural networks pretrained *in silico* and being heavily dependent on extensive human input. In this paper, we review current implementations and models of SYNBIONNs. We briefly present the biological platforms that show potential for designing and constructing perceptrons and/or multilayer SYNBIONNs. We explore their future possibilities along with the challenges that must be overcome to successfully implement a scalable *in vivo* biological neural network capable of *online* learning.

1. Introduction

Synthetic biology has made significant impact in different fields ranging from *living* therapeutics (Cubillos-Ruiz et al., 2021) to sustainable manufacturing (Scown and Keasling, 2022). Using engineered biological parts, (bio)computing has been performed with the employment of different biological processes (Cameron et al., 2014; English et al., 2021; Fink and Jerala, 2022; Kawasaki et al., 2020; Lv et al., 2021; Moškon et al., 2021) as well as beyond conventional binary logic (Teo et al., 2015). One of the key potentials of such parts is the possibility of their direct integration into a biological context, which offers a vast scope of applications, such as industrial fermentation control (TerAvest et al., 2011), drug discovery and delivery (Kis et al., 2015), and in-the-field diagnostics (Vasić et al., 2022).

Artificial neural networks (ANNs) have become one of the key computing models applied in modern machine learning and have found many real-world applications (Abiodun et al., 2018). ANNs have been inspired by the biological networks of human brains and are based on different mathematical models of neurons (Wang, 2003). Even though the first mathematical model of a neuron was introduced in 1943 (McCulloch and Pitts, 1943), the popularity of ANNs has increased significantly only after 2006 (Emmert-Streib et al., 2020). Today, deep learning (using ANNs with many hidden layers — deep ANNs) has become one of the most successful approaches in machine learning and has drastically improved applications of the latter in different domains ranging from speech and visual object recognition to drug discovery and genomics (LeCun et al., 2015). The most applicable ANNs' implementations have been mostly performed *in silico*. However, certain attempts have already been made to implement these using unconventional nonsilicon-based computing platforms (Fernando and Sojakka, 2003; Jones et al., 2007).

Herein, we focus on the current implementations and future potentials of synthetic biological neural networks (SYNBIONNs), i.e., ANNs implemented using engineered biological parts for use in a broad variety of synthetic biology applications. We review current implementations and recent efforts to implement highly scalable and adaptable SYNBIONNs. We present an overview of the different approaches that could be applied to the *in vivo* training of SYNBIONNs. We conclude with future perspectives for the development and applications of neuromorphic computing using engineered biological parts.

2. From biology to computer science and back

The human brain is a complex organ that collects, stores, and processes information from the environment. The elementary computational units of the brain are neurons (Sidiropoulou et al., 2006), which form complex networks. The dynamics of the latter cannot be directly explained by the relatively simple nature of an individual neuron. Therefore, different models of neurons, biological learning, and

* Corresponding author.
E-mail address: miha.moskon@fri.uni-lj.si (M. Moškon).

https://doi.org/10.1016/j.biosystems.2024.105164
Received 14 June 2023; Received in revised form 3 January 2024; Accepted 21 February 2024
Available online 23 February 2024

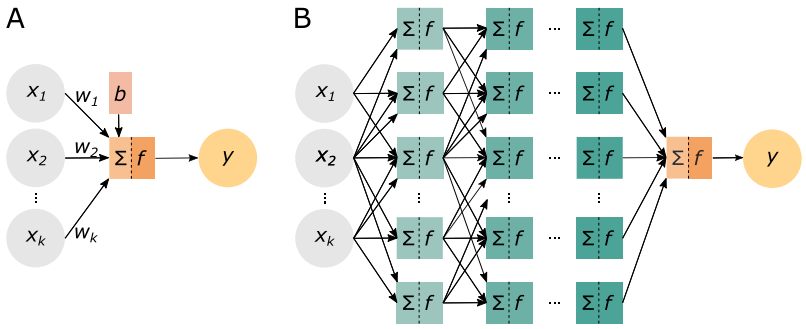


Fig. 1. Basic artificial neural network models. A: A perceptron neural network with k inputs. Each input is multiplied by its weight (w_i, $i \in \{1, 2, \ldots, k\}$), and all the products are summed together with a bias (b). The activation function f translates the weighted sum into an output (y). B: An example of a multilayer neural network with several hidden layers (weights and biases are omitted from the figure). Note that both examples present feedforward neural networks in which inputs are always propagated towards an output. Recurrent edges are possible in recurrent neural networks. In these, the output of a neuron can serve as an input to its preceding neuron.

neural networks have been introduced. Recently, the main applications of these models have pivoted from analysing the dynamics of a natural system to using neural models as a computing platform, known as ANNs, one of the most popular frameworks in machine learning research and applications.

The first computational implementation of an ANN was a perceptron, introduced in 1958 (McCulloch and Pitts, 1943; Rosenblatt, 1958) (Fig. 1A). Since it only has a single layer, it can solely learn to classify linearly separable classes, meaning different classes of inputs can be separated with a linear decision boundary, that is either a line or a plane in 2- or 3-dimensional space, respectively and a hyperplane in higher dimensions. Examples of such classes are AND, OR and NOT functions, however, XOR function is not linearly separable, which is why the perceptron fails in classifying its dynamics. The perceptron model was later extended into a multilayer network (Ivakhnenko, 1968) (Fig. 1B). Moreover, its supervised learning, which relies on using labelled datasets to train the network, was enhanced with an intuitive yet powerful algorithm using backpropagation in combination with gradient descent (Werbos, 1974; Wythoff, 1993).

Until today, different types of ANNs have been proposed and widely applied to different real-world applications ranging from natural language processing (Young et al., 2018) and large language models (Zhao et al., 2023) to speech recognition (Zhang et al., 2018). Deep learning combined with deep neural networks has shown dominance in several domains compared to other machine learning approaches (Alzubaidi et al., 2021). Some of the neural circuit models that have been employed in SYNBIONN implementations so far are Hopfield networks, winner-take-all and loser-take-all networks (see Section 4.1), perceptrons (see Section 4.2), convolutional neural network (see Section 4.3) and reservoir computing (see Section 4.4). Although these models can be applied to perform certain tasks, they mostly lag behind deep ANNs in general-purpose problem solving. The training of described models is mostly based on unsupervised or relatively simple supervised or reinforced training.

Hebbian learning falls under unsupervised learning and follows the principle known as *neurons that fire together, wire together*. This means that the strength of the connection between neurons changes based on how often they are activated. When a connection between two neurons is activated frequently, its weight increases and vice-versa: when the activation is less frequent, the weight weakens (Choe, 2014; Munakata and Pfaffly, 2004). In Hebbian learning, the learning process is driven by the patterns of activity that occur with given inputs, which allows self-organisation of the network and does not require explicit teaching signals (Munakata and Pfaffly, 2004).

In supervised or reinforced training, desirable behaviours are rewarded while undesirable ones are punished. These models may therefore have advantages when implemented on alternative processing platforms such as molecular or DNA-based computing, where weight adaptation cannot rely on fast and accurate backpropagation-based training (Song and Reif, 2019).

A further distinction can be made between offline and online learning approaches to model training. In offline learning, the model is trained with the entire data set in a single batch. In contrast, with online learning, the learning process begins with each newly acquired data point, which enables continuous learning (Bottou, 1999; Fontenla-Romero et al., 2013).

3. Synthetic biology meets molecular computing

Most of the first successful implementations of engineered biological circuits with information processing capabilities were based on transcription logic (Cameron et al., 2014). These include the *repressilator* (Elowitz and Leibler, 2000) and the *toggle switch* (Gardner et al., 2000), which were both introduced already in 2000. Soon, the transcription-based engineered platforms as well as the proposed structures have been extended not only to operate in bacterial cells, but also in other types of cells (e.g., in mammalian cells Ausländer and Fussenegger, 2016 and plants Khan et al., 2022). The first implementations were mainly based on the same set of biological parts, such as repressors lacI and cI, however, different strategies have soon been applied to increase the scalability of engineered biological circuits ranging from the employment of synthetically designed orthogonal transcription factors (Liu et al., 2018), e.g., CRISPR repressors (Yeo et al., 2018), transcription activator-like effectors (Moore et al., 2014), synthetic zinc finger proteins (Gersbach et al., 2014), to the implementation of complex logic circuits using multicellular approaches (Moškon et al., 2021), e.g., see Regot et al. (2011) and Tamsir et al. (2011). Ample research has also been conducted in *de novo* design of orthogonal protein- or peptide-based building blocks. Numerous parts were used in circuits targeting biological applications that carried out amplification, sensing or control of transcription and/or translation. Noteworthy instances of computational systems utilising transcription-based logic include decoder-based structural logic circuits (Lormeau et al., 2021), an analogue computing platform (Teo et al., 2015), mixed-signal computation (Rubens et al., 2016), and sequential logic circuits (Andrews et al., 2018). These structures have also been applied to perform neural-like computation (e.g., see Sarkar et al. (2021)).

Engineered protein circuits present a more recent alternative to transcription-based logic circuits, and might exhibit certain advantages in comparison to the latter. Unlike transcription-based circuits, "ready-made" protein circuits that perform computation via protein-protein interactions or protein modifications can directly interface with endogenous pathways such as cell death, function in different cell types, and generally have faster response times (Chen and Elowitz, 2021; Fink et al., 2019). Several different implementations of protein-based logic circuits have been proposed in recent years (Chen and Elowitz, 2021; Fink et al., 2019; Gao et al., 2018; Chen et al., 2022). These include

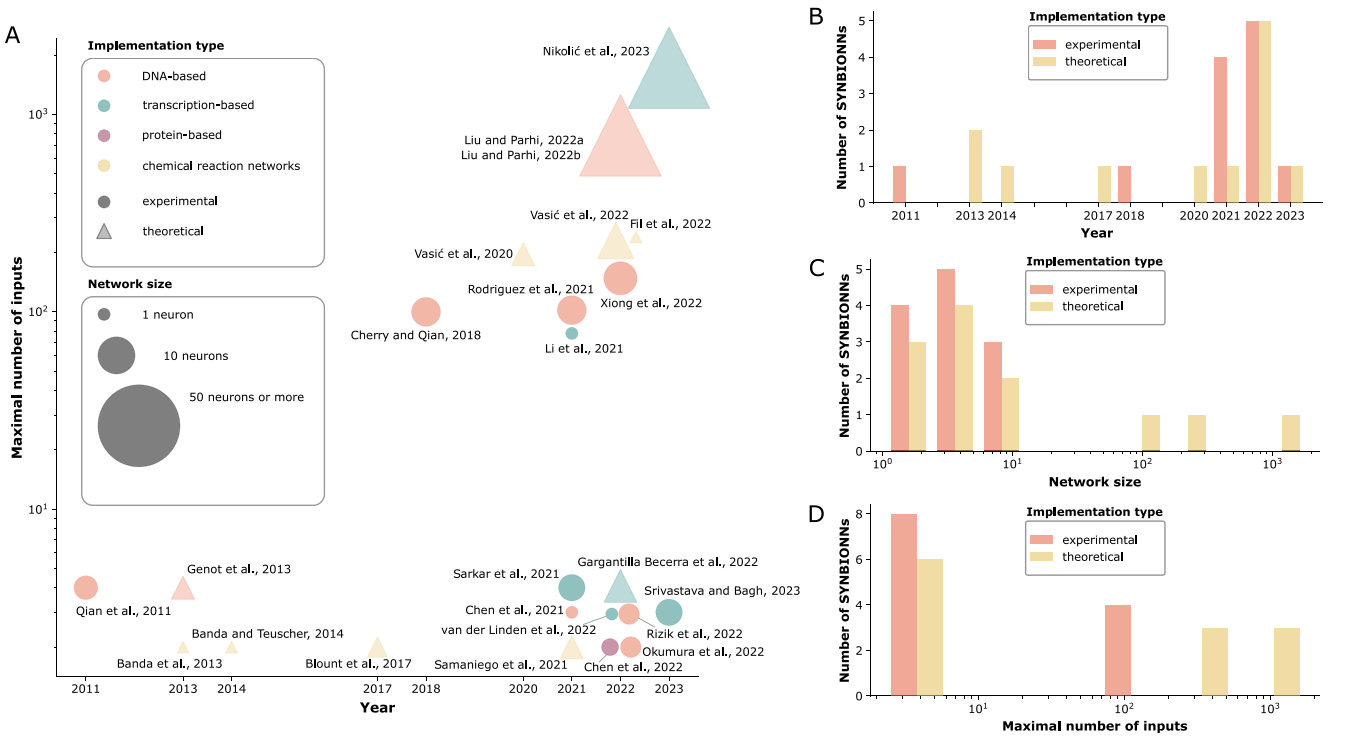


Fig. 2. The evolution of synthetic biological neural networks (SYNBIONNs). A: Types and sizes of different SYNBIONN implementations over the last decade. B: Number of SYNBIONNs reported in each year. C: Number of neurons (i.e., network size) in different SYNBIONN implementations. D: Maximal number of inputs in different SYNBIONN implementations. *The references used in the figure:* Qian et al. (2011), Banda et al. (2013), Genot et al. (2013), Banda and Teuscher (2014), Blount et al. (2017), Cherry and Qian (2018), Vasić et al. (2020), Samaniego et al. (2021), Chen et al. (2021), Rodriguez et al. (2021), Li et al. (2021), Sarkar et al. (2021), Fil et al. (2022), Vasić et al. (2022), Liu and Parhi (2022a,b), Okumura et al. (2022), Rizik et al. (2022), Xiong et al. (2022), Becerra et al. (2022), van der Linden et al. (2022), Chen et al. (2022), Nikolić et al. (2023) and Srivastava and Bagh (2023).

logic gates based on *de novo* designed protein heterodimers (Chen et al., 2020) and protease-based logic circuits, such as *split-protease-cleavable orthogonal-coiled coils-based* (SPOC) logic circuits and *circuits of hacked orthogonal modular proteases* (CHOMP) (Fink and Jerala, 2022; Fink et al., 2019; Gao et al., 2018).

DNA made its debut in the realm of molecular computing already in 1994, marking the inception of the field known as DNA computing (Adleman, 1994). DNA has since then been used as a building block in applications ranging from logic circuits to complex neural networks (Nagipogu et al., 2023). The attributes that make it so advantageous are the highly predictable Watson–Crick base-pairing, data storage capacity, and programmability (Song and Reif, 2019). It has also been applied to engineered nucleic acid logic circuits, which currently present one of the most promising synthetic biological platforms. These were introduced with the application of strand displacement reactions (SDRs), where a single-strand DNA (ssDNA), denoted invading strand, is capable of displacing one of the strands in the double-stranded DNA (dsDNA). After displacement, the invading strand hybridises with the remaining DNA strand, forming dsDNA. SDRs can occur only if the invading strand is complementary to the remaining strand (Lv et al., 2021; Seelig et al., 2006).

Up to this day, multiple versions of SDR designs have been used (for an extensive review see Lv et al. (2021)), one of the most prominent being toehold-mediated strand displacement reactions (TMSDR). TMSDR were proposed in 2006 by Seelig et al. (2006) to simplify experimental conditions and achieve enzyme-free computation. TMSDRs exhibit predictable kinetics and enable easy information encoding (Lv et al., 2021), which are highly desirable features in networks capable of complex computation. The invading and base DNA strands in TMSDR feature complementary single-stranded ends, called toehold domains, that enable their association, after which the invading strand displaces the top strand in the gate dsDNA and hybridises with the base strand to form new dsDNA (Lv et al., 2021; Seelig et al., 2006).

DNA computing structures were additionally expanded with enzymes, among the better-known and widely used examples are the PEN-based circuits (Montagne et al., 2011) and genelets (Kim et al., 2006), the former applying polymerase, endonuclease and nickase, and the latter T7 RNA polymerase and RNase H. Enzymes have been widely used in different circuits due to their speed, versatility, and tight kinetic control.

4. Synthetic biological neural networks

Synthetic biology started combining biological building blocks into programmed circuits for different types of practical applications also in the field of molecular computing. These circuits soon led to the design and implementation of synthetic biological perceptrons and neural networks, which have been applied to perform tasks such as molecular classification (Kieffer et al., 2023) and pattern formation (Vasić et al., 2022). The first neural network architecture constructed with biological parts (Qian et al., 2011) laid the foundation for SYNBIONNs and led to their further evolution (Fig. 2A). Ten years later, in 2021, the golden age of experimental SYNBIONNs began, encouraging a surge in research dedicated to theoretical SYNBIONNs (Fig. 2B). Remarkable progress can be seen in the dimensions of neural networks, ranging from perceptrons to theoretical networks with more than a thousand neurons (Nikolić et al., 2023) – see Fig. 2C. Most SYNBIONNs feature a maximum of two to four inputs, however, a few networks with more than a hundred potential inputs have also been designed (Fig. 2D). In the upcoming chapters, selected representatives of this research field are presented in more detail.

4.1. The first SYNBIONN and its evolution

Hopfield networks present single-layer recurrent ANNs using the sign function as an activation function (Fig. 3). They have been used in

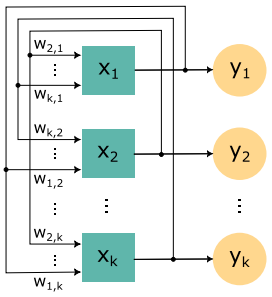


Fig. 3. A Hopfield network consists of a single layer of neurons, with each neuron being recurrently connected to all the other neurons. Connection weights between two neurons i and j are symmetric, i.e. $w_{i,j} = w_{j,i}$.

applications such as associative memory (McEliece et al., 1987) and image restoration (Paik and Katsaggelos, 1992), and can be trained using Hebbian learning (Munakata and Pfaffly, 2004). This ANN architecture was used in the first SYNBIONN, constructed in 2011 by Qian et al. (2011).

To build the Hopfield associative memory, DNA-based SYNBIONN Qian et al. used linear threshold gates consisting of different types of seesaw gates, which are based on TMSDR containing invading ssDNA and gate dsDNA.

Each seesaw gate, regardless of its type, incorporates three types of reactions: seesawing, thresholding and reporting. The first two reactions occur side-by-side, with thresholding being faster than seesawing. Upon the addition of input strands, some are sequestered by the thresholding complex, and the remaining proceed to seesawing, which releases the output strand from the reporter complex (Fig. 4) (Qian et al., 2011). Combining multiplying and threshold gates resulted in amplifying gates, which helped in the reduction of the circuit size. The four-neuron Hopfield network that remembers four patterns was trained in *in silico* and then assembled accordingly to the pre-determined weights without the capability of additional online learning. When presented with a partial or incomplete pattern, it was able to recover the most similar pattern it remembered (Qian et al., 2011).

To further scale-up the capabilities of DNA-based neural networks, Genot et al. (2013) reformed Hopfield associative memory into a winner-take all (WTA) neural network. WTA is computationally more powerful and less constrained by the number of patterns and their complexity. In WTA networks, only the neuron that achieves the highest weighted sum of all inputs remains active (Maass, 2000). While Genot et al. used a combination of DNA and enzymes and only focused on simulation-based analyses, Cherry and Qian (2018) built a DNA-only WTA neural network, which was able to classify patterns of handwritten digits from 1 to 9 into corresponding categories. Each digit was represented with a weight matrix procured by averaging digit patterns from the MNIST database. The output was provided by different types of seesaw gates, which executed five reactions: weight multiplication, summation, comparison, signal restoration, and reporting (Fig. 5A) (Cherry and Qian, 2018).

The reactions provide weighted inputs that are summed and compared to each other via pairwise annihilation. In this process, two competing weighted sum species are bound by an annihilator, forming waste products, until only one species remains. The concentration of the winning species is restored during signal restoration, which is followed by reporting. All the reactions are sequential, save for pairwise annihilation and signal restoration, which occurred side by side, although pairwise annihilation was much faster than signal restoration. This difference in the speed of the two reactions enables only the winning molecule to enter signal restoration (Cherry and Qian, 2018).

The weights in the presented WTA network underwent *in silico* training before being compiled *in vitro*. It could be easily reconfigured for different pattern-recognition tasks by varying the concentration of weight molecules. However, the network is limited to the highest number of distinct patterns remembered, the maximum number of corrupted bits that still allow recognition, and the size of the DNA network (Cherry and Qian, 2018).

Deriving the weights from the average training patterns reduces the accuracy of classification between different classes of patterns that share a high similarity (Cherry and Qian, 2018; Rodriguez et al., 2021). To avoid this, the determination of the least similarity between the pattern and the class is preferable, which can be assessed using a loser-take-all (LTA) neural network. In this network, the neuron with the lowest value among all inputs is activated (Rodriguez et al., 2021).

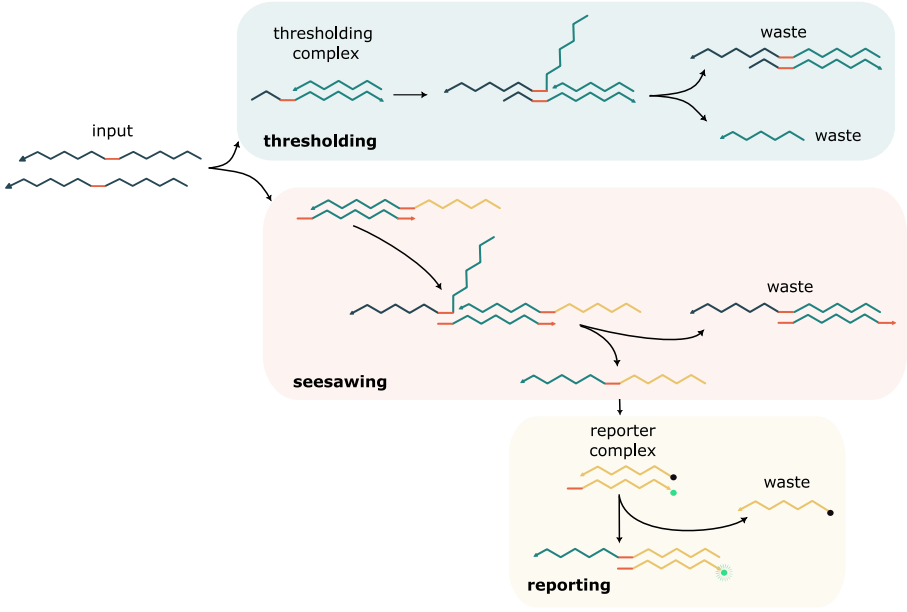


Fig. 4. The three basic types of reactions in a seesaw gate in Hopfield associative memory are seesawing, thresholding, and reporting. Seesawing and thresholding take place side-by-side with thresholding being faster. When the threshold concentration is surpassed, input molecules react in the seesawing reaction and release a strand, capable of displacing the strand with the quencher molecule (dark dot) from the reporter complex, providing an output by the fluorescent probe (green dot).

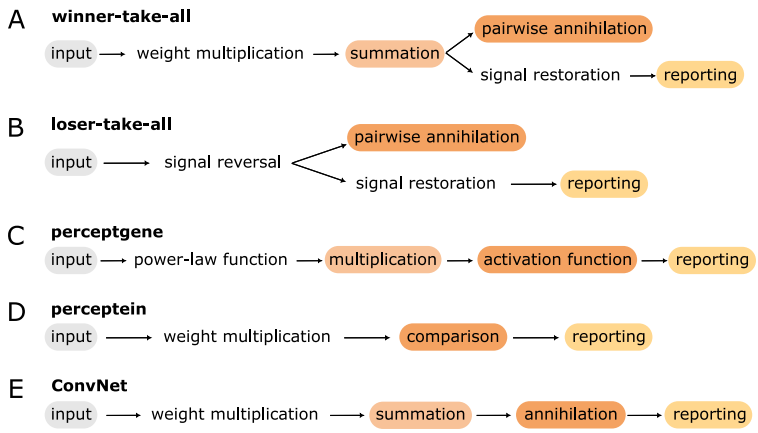


Fig. 5. Basic types of reactions in selected SYNBIONNs. A: The input in the WTA network undergoes weight multiplication, after which the weighted inputs are subjected to summation, followed by pairwise annihilation and signal restoration. Although signal restoration and pairwise annihilation occur side-by-side, the latter has a faster reaction rate, allowing that only the winning molecules to proceed to signal restoration stage. The concentration of the winning molecule is replenished in the course of signal restoration to ensure a measurable output during reporting. B: The LTA network features seesaw gates that execute similar functions to WTA. Each input is first subjected to signal reversal, after which the reversed input undergoes pairwise annihilation. The winning molecule enters signal restoration to provide an output in reporting reactions. C: The perceptgene responds in a non-linear way. The power-law function provides weights for each input, and the analogue value of weighted inputs is combined during the multiplication step. The activation function introduces a threshold which, if exceeded, is followed by reporting. D: The inputs in perceptein undergo weight multiplication, followed by comparison, where the winning molecule is chosen to yield a WTA output in the reporting. E: The switching gate architecture used in ConvNet begins with weight multiplication, the weighted inputs are summed and carry on to annihilation, after which the remaining molecules provide an output.

A DNA-based three-input LTA circuit with nine unique input combinations was designed by Rodriguez et al. (2021). The first reaction in the LTA network is signal reversal, in which each input is reversed, meaning that the smaller the input, the higher its reversal. This is followed by the reactions also executed in the WTA (Fig. 5B). However, unlike the WTA, the inputs in the LTA had two toeholds, which made the signal reversal step irreversible (Rodriguez et al., 2021). Both networks were first subjected to *in silico* training, where the connection weights were determined. This was followed by *in vitro* assembly. Although LTA yields better classification accuracy, it lacks the robustness of the WTA network (Cherry and Qian, 2018; Rodriguez et al., 2021).

4.2. Perceptrons, the perceptgene and the perceptein

Perceptrons are the simplest but nonetheless a non-trivial form of neural networks. Several efforts have been devoted to their implementation as a SYNBIONN. However, none has succeeded in achieving online learning, since the weights in most have been pre-determined with *in silico* learning.

Chen et al. (2021) prepared a neuron model based on the combination of DNAzymes and TMSDR. This improved the efficiency of molecular devices for performing more complex molecular computing. The input DNA molecules were hybridised via toeholds with an inhibitor strand that was hybridised with the catalytic site of the DNAzyme, thus rendering it inactive. Once free of inhibitors, the DNAzyme was active and capable of providing downstream products, which, after undergoing a linear thresholding function, produced fluorescent output. The described perceptron was used for the construction of a voting machine and although it is easily scalable, it does not employ learning (Chen et al., 2021).

Okumura et al. (2022) proposed a two-layer perceptron, comprised of enzymatic neurons with microRNAs as inputs. The tunable weights were positive and negative, the latter suppressing the replication of the signal strand via a drain template. The neuron was considered active if the weighted sum of inputs exceeded a pre-set threshold. The two-layer perceptron recursively partitioned a two-dimensional concentration plane into three nonlinearly separable regions. To achieve this, the perceptron was trained to first decide between two regions, using a hidden neural layer. Then the third region was decided upon with a NOR gate. This perceptron was trained on experimental data (Okumura et al., 2022).

Bactoneuron is a bacterial device, constructed by Sarkar et al. (2021), who mapped an ANN model to an engineered cellular model. Each bactoneuron unit generated a specific nonlinear response based on its inputs. These units were subsequently aggregated into a single-layer ANN bactoneuron. Chemical inputs to the bactoneuron were thus linearly combined. The weights of each input and biases of each bactoneuron were tuned by using different well-characterised synthetic promoters. Nevertheless, fine-tuning of these weights is not possible in this implementation, which also hinders the possibility of learning. The final single-layer bactoneuron ANN could be configured to perform logic functions in accordance with different digital circuits, such as majority gate, decoder, priority encoder, multiplexer and demultiplexer. This showcases the modularity of the presented system, which was also demonstrated in living cells (Sarkar et al., 2021). More recently, Srivastava and Bagh (2023) further advanced the bactoneurons by employing five distinct bactoneuron units to build reversible double Feynman gates with three inputs and three outputs. Similarly to previous work by Sarkar et al. (2021), the final ANN had a single-layer, relied on manual adjustment of weights and biases and had no learning mechanisms incorporated (Srivastava and Bagh, 2023).

Li et al. (2021) proposed neural-like computing in bacterial consortia for pattern recognition. Two types of bacteria were employed, namely, receiver and sender cells, each harbouring different genetic circuits. The weights in the network were implemented with promoters of different strengths. A gradient descent-based algorithm that enables weight optimisation was developed for *in silico* optimisation and offline training of the network. This algorithm was adapted to discrete weight values since promoters exhibit discrete strength levels. The sender bacteria were initially incubated with the corresponding inducer and subsequently combined with the receiver bacteria, which produced fluorescent outputs. The experimental characterisation was focused to 4-bit patterns, while the *in silico* simulations were demonstrated on the classification of 9×9 bit patterns (Li et al., 2021).

Becerra et al. (2022) have proposed an *in silico* learning of a perceptron neural network, producing a structure of the genetic network, which was then analysed with the application of the extended gro simulator (Gutiérrez et al., 2017). Moreover, the proposed network was later projected onto a set of plasmids using Cello (Jones et al., 2022). In *in silico* simulation experiments researchers explored how a bacterial colony derived from a bacterium harbouring the appropriate perceptron, performed in an optimisation task and whether the constructed

network allowed for programming of various behaviours. The latter was evaluated in a perceptron that detected four input signals, which resulted in four behaviours. The colony with such a perceptron formed microbial consortia based on different input signal patterns (Becerra et al., 2022).

The three-input perceptron prepared by van der Linden et al. (2022) takes a DNA input, the transcript of which unfolds the hairpin structure on the RNA toehold switch and allows translation of the downstream encoded protein. The concentration of this protein represents a weighted sum that undergoes a post-translational thresholding reaction, which sequesters or renders a predetermined amount of the first protein inactive. The remaining unsequestered proteins can trigger the expression of downstream genes. Weights in this perceptron were fine-tuned *in silico* and later implemented *in vitro* (van der Linden et al., 2022). This perceptron unravels the potential of the interplay between different biological platforms in biological computation.

Rizik et al. (2022) proposed a *perceptgene* circuit that responds in a non-linear way. The *perceptgene* is based on three functions: power-law, multiplication and activation function (Fig. 5C). First, the power law function assigns weights to inputs and is followed by the multiplication function, which combines analogue values of weighted inputs. Activation function is computed through AND or OR logic, depending on the magnitude of the weight and bias. Weights in this perceptgene take into account both binding affinities between interacting species as well as circuit topology, whereas bias is determined by the ratio between the maximum protein level and corresponding binding affinity. The perceptgene was described to be robust to noise at low signal concentrations; however, it can accept only a limited number of distinct inputs. Connecting different versions of perceptgenes into larger circuits, researchers implemented soft majority function and an analogue-to-digital converter. The optimisation of the perceptron was conducted with backpropagation algorithm *in silico* (Rizik et al., 2022).

The challenge of preparing an *in vivo* protein-based neural network has been undertaken by Chen et al. (2022). In their work, they employed protease-based logic circuits (Fink et al., 2019; Gao et al., 2018; Holt and Kwong, 2020) to implement a *perceptein* circuit in mammalian cells. Perceptein is an artificial perceptron using protein modules that compute the weighted sum of input protein concentrations and produce a WTA output — see Fig. 5D (Chen et al., 2022). Such a network has multiple outputs, whereas only the winner should be active for a given set of inputs. The proposed circuit was demonstrated on a programmable signal classification with tunable decision thresholds. This implementation was based on heterodimers designed *de novo* in a combination with split viral proteases to control protein activities. Maximal protease activation was achieved after approximately 100 h; however, the decision was made within 3 h. Even though there are many benefits of the proposed implementation, such as tunable weights and scalability of number of inputs/outputs, it is limited to a single-layer neural network, i.e. perceptron, which does not support learning.

4.3. The rise of the ConvNet

Xiong et al. (2022) expanded the capabilities of DNA-based SYNBIONNs by constructing a convolutional neural network, denoted ConvNet, which can recognise 32 different handwritten symbols, originating from 4 languages. The presented implementation approached the classification via a two-step process: it determined the language of the symbol presented with logic gates and then the ConvNet classified the specific symbol. The network was first trained *in silico* and constructed according to training results.

ConvNet consists of four layers: the input layer, convolutional layer, non-linear layer, and output layer. The convolutional layer performs multiplication, summation, and reporting which allocate the weights to inputs — see Fig. 5E (Alzubaidi et al., 2021; Xiong et al., 2022). The DNA architecture used was the switching gate, which has two toeholds, one or more recognition domains, and a weight-tuning domain. In the absence of the weight-tuning molecule, the weight-tuning domain forms a loop and enables the corresponding input to dislodge the intermediate product strand from the weight-substrate molecule. The released strand then reacts with the reporter gates. Obtaining two responses from one input, e.g., implementing the sum of positive and negative weights, requires two weight substrate molecules with the same recognition domains but different weight-tuning domains. The classification into 32 categories took more than 36 h, which can be decreased using a freeze-thaw cycle approach, where iterative freezing and thawing of DNA components causes faster dissolution of unproductive complexes (Xiong et al., 2022).

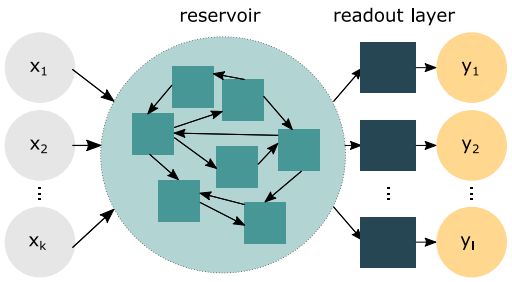


Fig. 6. In reservoir computing a neural network is composed of an input layer, a reservoir, and a readout layer. Reservoir maps a low-dimensional input into a high-dimensional linearly separable feature space. Learning is performed only on the readout layer, and the other two layers are constructed randomly.

4.4. Reservoir computing: Learning simplified

Another architecture that is well suited for use in biological networks is reservoir computing. A reservoir computing system consists of input neurons, a reservoir, a readout layer and output neurons. The *reservoir* is implemented with a randomly generated neural network that responds to the inputs in a nonlinear and high-dimensional manner. Randomised nature of the reservoir circumvents the need for precise characterisation of its building blocks. During training, only the weights in the readout layer are modified, which greatly simplifies the training of the network, since these represent a small fraction of the whole network (Schrauwen et al., 2007) (Fig. 6). Furthermore, the unchangeable nature of the reservoir allows for learning different transformations from the same input signal (Cucchi et al., 2022; Echlin et al., 2018). These characteristics of reservoir computing could greatly facilitate the design of adaptable SYNBIONNs.

In 2022, Liu and Parhi introduced two theoretical reservoir computing systems based on SDR. The first employed molecular oscillators (Liu and Parhi, 2022b) and the second used molecular memristors (Liu and Parhi, 2022a). Both were able to recognise patterns and solve second-order nonlinear prediction tasks. However, they required a large number of reactions. Despite being simulated *in silico*, neither system explored the adaptability and online learning potential of these molecular systems.

Nikolić et al. (2023) devised an *in silico* model of a multicellular reservoir computing system, which reduces the metabolic burden to a single cell and increases modularity, scalability, and robustness of the network. The cells were arranged in a three-dimensional grid of stacked layers, with communication relying on diffusion-based signalling via extracellular signalling molecules. Each cell harboured a gene regulatory network based on a random Boolean network, including a signalling gene, receptor and a secretor gene. The study demonstrated superior accuracy in the multicellular approach compared to single cells, yet the spatial organisation of the cells had a strong influence on the performance of the reservoir (Nikolić et al., 2023).

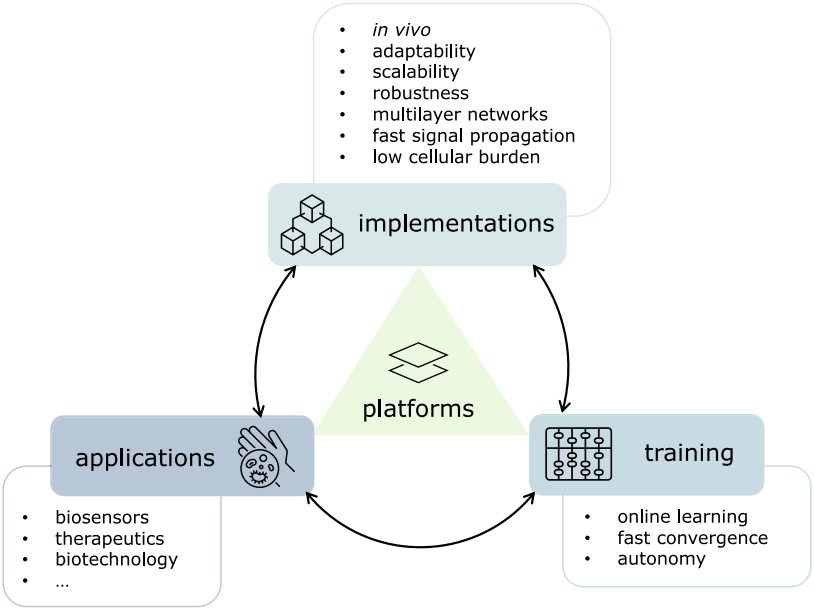


Fig. 7. Key considerations in designing an ideal SYNBIONN. Designing an efficient SYNBIONN first requires a close look at biological building blocks and platforms. The latter are closely connected to the ability of implementing scalability, robustness, multiple layers, and ensuring an *in vivo* operation with fast signal propagation and low cellular burden. Adaptability and autonomous operation can be incorporated through online learning, which also requires a relatively fast convergence rate. So far, SYNBIONNs show greatest potential in the fields of biotechnology, biosensors and personalised medicine.

5. An ideal network: Are we there yet?

The core desired traits of an ideal *in vivo* SYNBIONN are fast signal propagation, circuit compactness, robustness, scalable topology, a certain degree of autonomy, and tunable response, which can be adapted with online learning. The latter represents the capability of the network to adapt its weights even after it has been deployed in the environment. On the way to constructing such a system, the researchers are faced with numerous hindrances, some of which can be foreseen and already accounted for in *in silico* models, whereas others are revealed only during *in vitro/in vivo* experiments. As seen in the previous section, current SYNBIONNs do not yet encompass all of the core desired traits. Some vital design considerations to implement these traits are discussed in the following section and outlined in Fig. 7.

5.1. Considering the building blocks

DNA is the most widely used biological molecule in current SYNBIONNs, nonetheless, there are other promising building blocks for use in SYNBIONNs, including RNA, protein-based platforms, phosphorylation circuits and multicellular systems.

RNA carries genetic information and undergoes Watson–Crick base-pairing. However, unlike DNA, it can also convey a diversity of functions, thus holding the potential for increasing the circuit capabilities. Along with rapid signal propagation, these characteristics make RNA an appealing alternative. Furthermore, post-transcriptional RNA circuits are also a safer choice for use in therapeutic applications, as they avoid entering the nucleus and are thus prone to genomic integration or epigenetic modifications (Kawasaki et al., 2020; Wagner et al., 2018). Some of the parts and platforms based on RNA that have been used in the construction of complex RNA-based synthetic circuits so far are toehold switches (Green et al., 2014), RNA interfaced with CRISPR/Cas technology (Li et al., 2019), and RNA binding proteins (RBP) (Matsuura et al., 2018; Wroblewska et al., 2015).

A fetching stepping stone for further development of not only SYNBIONNs but also other biological circuits are designed proteins and split proteases, such as CHOMP and SPOC (Fink et al., 2019; Gao et al., 2018). Protein-based platforms provide faster response times compared to DNA-based platforms. Furthermore, circuits featuring CHOMP and/or SPOC technologies can be encoded in a single transcript, thus facilitating cell delivery, and are suitable for a wide variety of applications since they can interact with endogenous pathways and avoid genomic integration (Fink et al., 2019).

Another interesting platform possibility that has attracted attention are phosphorylation-driven circuits. In the context of SYNBIONNs, these have been discussed in Samaniego et al. (2021) and Gentili Pier Luigi (2022), however, their applications and experimental characterisation remain limited. Phosphorylation circuits have the potential for reversibility, paving the way to expanding functionalities within already established biological circuits.

AdaptoCell is a recently proposed multicellular sensor system that can be applied to biological implementations of reinforced learning (Prakash et al., 2023). Each AdaptoCell includes two quasi-identical cotransformed plasmids, harbouring a distinct fluorescent protein and antibiotic resistance. The weight of an AdaptoCell is defined by the fraction of the plasmid concentrations. These are regulated through the induction of the expression of a fluorescent protein and an antibiotic resistance gene. However, training of an AdaptoCell needs to be performed with the manual introduction of inducers and antibiotics into the cellular medium (Prakash et al., 2023).

5.2. The challenges of in vivo implementations

The selection of architecture and building blocks for SYNBIONNs marks the initial phase of the design process. This is followed by tackling challenges of an *in vivo* implementation. Among these challenges are successful delivery of building blocks and circuits into cells or engineering their *in situ* preparation, and ensuring the accurate functionality of the desired SYNBIONN.

Delivery of exogenously prepared DNA or RNA circuits into a cell often suffers from low efficiency and potential toxicity of delivery agents (Bae et al., 2021; Groves et al., 2016). Improving the delivery methods and switching to *in situ* preparation of logical gates could help design more sustainable SYNBIONNs. In DNA-based TMSDRs, DNA circuit components must undergo thermal annealing separately from each other to prevent cross-reactions between gates and inputs. This disables the continuous production of circuits at their operating site, particularly in the *in vivo* setting. Bae et al. (2021) approached

the issue of *in situ* circuit preparation by designing a DNA template that provides an ssRNA transcript, containing a self-cleaving ribozyme, which cuts the ssRNA and forms a gate complex. Although the proposed system solved the issue of *in situ* component preparation, the SDRs were unreliable.

Once the circuits are introduced into cells, they are not only limited to single use but are also restricted by the degradation rate of its components (Schaffter and Strychalski, 2022). Groves et al. (2016) successfully addressed this problem by chemically modifying DNA; however, some of the chemical modifications negatively impacted the reaction kinetics (Groves et al., 2016). Exploring the effects of chemical modifications on DNA, RNA, and proteins in synthetic biological circuits could broaden the potential for developing circuits that are more versatile and less susceptible to degradation. It is important to note, though, that altering the degradation rate should be approached with care, as it may have a negative impact on the metabolic burden, especially in larger networks.

To bypass most of the previously mentioned obstacles, Schaffter and Strychalski (2022) developed a similar design of isothermal production of circuit components as Bae et al. (2021), denoted cotranscriptionally encoded RNA strand displacement (ctRSD) circuits. In ctRSD gate transcription templates are DNA-based and consist of two gate strands interrupted by a self-cleaving RNA ribozyme motif. The latter enables the cleavage of such single RNA transcripts into reactive gates. All gates have been constructed using a 3- instead of 4-letter code to decrease incorrect or problematic folding. This constraint poses a limitation in biosensor design, since cellular RNA uses a 4-letter code and must be converted to a 3-letter code for detection with ctRSD circuits. Another issue in ctRSD is that the ribozyme is left on the gates after cleavage, which consequently influences the kinetics of strand displacement. Adding a single-strand spacer between the toehold and the ribozyme binding sequence alleviates this condition; however, the design space of single-strand spacers has not yet been characterised (Schaffter and Strychalski, 2022).

As seen above, the design and construction of highly efficient SYNBIONNs involve the interplay of various scientific fields, necessitating not only the meticulous study of building blocks and ANN architectures but also turning attention to purely methodological aspects.

5.3. Scaling up the networks

While scalability remains a highly sought-after feature in an ideal SYNBIONN, the construction of SYNBIONNs scalable beyond a perceptron still presents a major challenge. Certain approaches have already been taken towards the implementation of multilayer networks (Blount et al., 2017). However, the scalability of these is limited. One way to address this problem is to construct a hybrid ANN combining different platforms that have been used in existing implementations as separate layers. For example, the combination of DNA and RNA platforms or protein modules and post-translational regulation have already been successfully implemented by van der Linden et al. (2022) and Chen et al. (2020) – see Section 4.2. Leveraging distinct biological platforms could therefore facilitate the construction of multiple layers within a single cell.

Further increase of the network complexity is plausible by employing multicellular consortia that leverage distributed computing, a continual topic of discussion regarding biological circuits (Karkaria et al., 2020; Macía et al., 2012; Canadell et al., 2022). While distributed computing allows for higher capabilities and lower metabolic burden to a single cell, the challenge of maintaining distinct population ratios over several generations remains.

The shift towards constructing deep biological ANNs with multicellular consortia requires using engineered intercellular communication mechanisms. Currently, the set of orthogonal molecules used in quorum sensing, a form of intercellular communication in bacteria, is limited. Nevertheless, ongoing research aims to broaden this set, thus advancing the use of intercellular communication in synthetic biology applications (Tekel et al., 2019; Jiang et al., 2020).

In silico research plays an indispensable role in testing new scenarios, providing valuable insights and offering an understanding of system behaviour. For instance, tools like gro have been proposed to test such envisioned scenarios, offering a platform for the simulation and analysis of intricate multicellular dynamics, including the exploration of various types of cell communication mechanisms such as bacterial conjugation (Gutiérrez et al., 2017; Jang et al., 2012).

5.4. From pre-trained to adaptable neural networks

The SYNBIONN implementations discussed in this review have predominantly relied on prior *in silico* weight learning, where pretrained networks were subsequently hardwired into a biological system. Such hardwiring significantly impedes flexibility, adaptability and scalability of a SYNBIONN, diminishing its ability to effectively function in diverse and dynamic environments.

Even though specific architectures require minimal learning (see Section 4.4), the implementation of adaptable response of a network still requires the application of synthetic biological parts and/or circuits with an online tunability (Pei et al., 2010). Moreover, the response of a network needs to be tuned with an implementation of a learning strategy to allow the convergence of weights towards the most optimal solution.

So far, adaptability has been mainly researched in *in silico* chemical reaction networks (CRN). Banda et al. (2014) introduced a theoretical chemical two-input perceptron denoted as *asymmetric signal perceptron* (ASP). The proposed system learns through reinforcement by injecting a *penalty signal* when the output discrepancy is detected. Weights are encoded by separate chemical species, concentrations of which can be adapted during the process of learning. The authors demonstrated the proposed framework on the learning of linearly separable binary functions. The proposed approach was then extended to a multilayer ANN, which supports supervised learning using a chemical analogue of backpropagation (Blount et al., 2017). Additionally, Lakin (2023) devised an abstract chemical reaction network that facilitates supervised learning through backpropagation for nonlinear neurons, employing a leaky ReLU nonlinear transfer function. Although the presented results seem promising, the proposed frameworks applied models of abstract chemical reaction networks, which are yet to be translated to more realistic models and their actual implementations.

Originating directly from biology, Hebbian learning has emerged as an interesting possibility for integrating learning into biological systems. This concept was first explored by Fernando et al. (2009), and the most recent and accomplished work in this field was presented by Fil et al. (2022). The latter introduced a fully autonomous model of a chemical neuron, which was also proposed in the form of DNA SDRs.

The computation begins with the transformation of input molecules into internal state molecules. These initiate the conversion of the learning molecules into their active form, which together with the unconverted input molecules activate the neuronal weight increase. The simultaneous presence of two different molecular species culminating in weight increase is the hallmark of Hebbian learning. The constructed chemical neuron was able to solve different variants of frequency bias tasks and a time correlations task (Fil et al., 2022). The showcased autonomous neuron, capable of Hebbian learning, presents a promising option for implementing this learning mechanism into SYNBIONNs.

Weight learning could also be achieved with the cellular implementation of different metaheuristics, which have already been applied to solve different types of optimisation problems using cellular populations (Ortiz et al., 2021; Gargantilla Becerra et al., 2021; Wakabayashi and Yamamura, 2005). A widely recognised metaheuristic algorithm is the genetic algorithm, inspired by Darwinian evolution. In each iteration, population from the previous step is assessed, and individuals displaying the highest fitness values are chosen for crossing, crossover

and mutation, aimed at procuring an improved population (Katoch et al., 2021). Moškon and Mraz proposed an integration of a heuristic optimisation framework based on genetic algorithms into a cellular population to automatically select the computing parts to execute a specific logic function and thus perform a variation of reinforcement learning (Moškon and Mraz, 2022). Each cell in the population contains two sets of functions. The first set presents functions that are used to compare the cellular response with the required behaviour and leads to apoptosis when the actual response differs from the required. The second set of functions presents basic processing functions which are optimised during the optimisation. These functions are randomly distributed among cells in the beginning. Later on, cells with a larger success rate share their functions among their neighbours through conjugation to allow the convergence of the whole population towards an optimal solution. Even though the framework was only used in the acquisition of different logic functions within a cellular population, this could as well be applied to training of SYNBIONNs in the context of their topology as well as in the context of SYNBIONN weight adaptation.

The implementation of adaptability and online learning into SYNBIONNs present one of the more elusive aspects in this field of research, since many constraints that are absent in computer science must be accounted for in biological systems. As seen throughout this review, overcoming these challenges requires close cooperation of multiple research fields.

6. Conclusions

ANNs present one of the essential frameworks in the modern machine learning arsenal. Their incorporation into (synthetic) biological systems might present a game changer for life sciences. Even though we are still striving towards *in vivo* SYNBIONNs capable of online learning, several approaches have already demonstrated tremendous potential to implement these in the near future. Such implementations will enable new applications in synthetic biology in which systems become more and more susceptible and adaptable to external cues.

All of these exciting new possibilities and applications call for a thorough consideration of possible negative, near-, and far-reaching consequences. A high level of adaptability of synthetic biological systems might cause unpredictable behaviour, which requires defining and standardising stringent biosafety strategies (Pei et al., 2022). As the development of these technologies advances, encouraging discussion of their use, biosafety and ethical aspects is of utmost importance.

We believe that the development of SYNBIONNs and their application will allow us to address the challenges that arise in different fields, particularly biotechnology, personalised medicine, bioremediation and biosensors. Advancing the research on building blocks, architectures, scalability, adaptability, learning in biological systems, and *in vivo* execution, will enable the transition of applications of synthetic biology to neuromorphic computing and the further development of biological artificial intelligence.

Funding

The research was partially supported by the scientific-research program P2-0359 and by the basic research project J1-50024, both financed by the Slovenian Research and Innovation Agency. The research was also supported by the University of Ljubljana, Slovenia Development Fund. The funding sources had no role in the design of the study and collection, analysis, and interpretation of data nor in writing the manuscript.

CRediT authorship contribution statement

Ana Halužan Vasle: Conceptualization, Investigation, Writing – original draft, Writing – review & editing. **Miha Moškon:** Conceptualization, Investigation, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

References

Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A., Arshad, H., 2018. State-of-the-art in artificial neural network applications: A survey. Heliyon 4 (11), e00938.

Adleman, L.M., 1994. Molecular computation of solutions to combinatorial problems. Science 266 (5187), 1021–1024.

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L., 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. J. Big Data 8 (1), 1–74.

Andrews, L.B., Nielsen, A.A., Voigt, C.A., 2018. Cellular checkpoint control using programmable sequential logic. Science 361 (6408), eaap8987.

Ausländer, S., Fussenegger, M., 2016. Engineering gene circuits for mammalian cell–based applications. Cold Spring Harb. Perspect. Biol. 8 (7), a023895.

Bae, W., Stan, G.-B.V., Ouldridge, T.E., 2021. In situ generation of RNA complexes for synthetic molecular strand-displacement circuits in autonomous systems. Nano Lett. 21, 265–271.

Banda, P., Teuscher, C., 2014. Learning two-input linear and nonlinear analog functions with a simple chemical system. In: International Conference on Unconventional Computation and Natural Computation. Springer, pp. 14–26.

Banda, P., Teuscher, C., Lakin, M.R., 2013. Online learning in a chemical perceptron. Artif. Life 19 (2), 195–219.

Banda, P., Teuscher, C., Stefanovic, D., 2014. Training an asymmetric signal perceptron through reinforcement in an artificial chemistry. J. R. Soc. Interface 11 (93), 20131100.

Becerra, A.G., Gutiérrez, M., Lahoz-Beltra, R., 2022. Computing within bacteria: Programming of bacterial behavior by means of a plasmid encoding a perceptron neural network. Biosystems 104608.

Blount, D., Banda, P., Teuscher, C., Stefanovic, D., 2017. Feedforward chemical neural network: An *in silico* chemical system that learns XOR. Artif. Life 23 (3), 295–317.

Bottou, L., 1999. On-Line Learning and Stochastic Approximations. Publications of the Newton Institute, Cambridge University Press, pp. 9–42, Ch. Chapter 2.

Cameron, D.E., Bashor, C.J., Collins, J.J., 2014. A brief history of synthetic biology. Nat. Rev. Microbiol. 12, 381–390.

Canadell, D., Ortiz-Vaquerizas, N., Mogas-Diez, S., de Nadal, E., Macia, J., Posas, F., 2022. Implementing re-configurable biological computation with distributed multicellular consortia. Nucleic Acids Res. 50 (21), 12578–12595.

Chen, Z., Elowitz, M.B., 2021. Programmable protein circuit design. Cell 184 (9), 2284–2301.

Chen, Z., Kibler, R.D., Hunt, A., Busch, F., Pearl, J., Jia, M., VanAernum, Z.L., Wicky, B.I., Dods, G., Liao, H., Wilken, M., Ciarlo, C., Green, S., El-Samad, H., Stamatoyannopoulos, J., Wysocki, V.H., Jewett, M.C., Boyken, S.E., Baker, D., 2020. De novo design of protein logic gates. Science 368 (6486), 78–84.

Chen, Z., Linton, J.M., Zhu, R., Elowitz, M.B., 2022. A synthetic protein-level neural network in mammalian cells. Preprint at https://www.biorxiv.org/content/10.1101/2022.07.10.499405v1.

Chen, C., Wu, R., Wang, B., 2021. Development of a neuron model based on DNAzyme regulation. RSC Adv. 11 (17), 9985–9994.

Cherry, K.M., Qian, L., 2018. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. Nature 559 (7714), 370–376.

Choe, Y., 2014. Hebbian learning. In: Encyclopedia of Computational Neuroscience. Springer Science Business Media, New York, pp. 1–5.

Cubillos-Ruiz, A., Guo, T., Sokolovska, A., Miller, P.F., Collins, J.J., Lu, T.K., Lora, J.M., 2021. Engineering living therapeutics with synthetic biology. Nat. Rev. Drug Discov. 20 (12), 941–960.

Cucchi, M., Abreu, S., Ciccone, G., Brunner, D., Kleemann, H., 2022. Hands-on reservoir computing: A tutorial for practical implementation. Neuromorph. Comput. Eng. 2 (3), 032002.

Echlin, M., Aguilar, B., Notarangelo, M., Gibbs, D.L., Shmulevich, I., 2018. Flexibility of Boolean network reservoir computers in approximating arbitrary recursive and non-recursive binary filters. Entropy 20 (12).

Elowitz, M.B., Leibler, S., 2000. A synthetic oscillatory network of transcriptional regulators. Nature 403 (6767), 335–338.

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S., Dehmer, M., 2020. An introductory review of deep learning for prediction models with big data. Front. Artif. Intell. 3, 4.

English, M.A., Gayet, R.V., Collins, J.J., 2021. Designing biological circuits: Synthetic biology within the operon model and beyond. Annu. Rev. Biochem. 90, 221–244.

Fernando, C.T., Liekens, A.M.L., Bingle, L.E.H., Beck, C., Lenser, T., Stekel, D.J., Rowe, J.E., 2009. Molecular circuits for associative learning in single-celled organisms. J. R. Soc. Interface 6 (34), 463–469.

Fernando, C., Sojakka, S., 2003. Pattern recognition in a bucket. In: European Conference on Artificial Life. pp. 588–597.
Fil, J., Dalchau, N., Chu, D., 2022. Programming molecular systems to emulate a learning spiking neuron. ACS Synth. Biol. 11 (6), 2055–2069.
Fink, T., Jerala, R., 2022. Designed protease-based signaling networks. Curr. Opin. Chem. Biol. 68, 102146.
Fink, T., Lonzarić, J., Praznik, A., Plaper, T., Merljak, E., Leben, K., Jerala, N., Lebar, T., Strmšek, Ž., Lapenta, F., Bečina, M., Jerala, R., 2019. Design of fast proteolysis-based signaling and logic circuits in mammalian cells. Nat. Chem. Biol. 15 (2), 115.
Fontenla-Romero, Ó., Guijarro-Berdiñas, B., Martinez-Rego, D., Pérez-Sánchez, B., Peteiro-Barral, D., 2013. Online machine learning. In: Efficiency and Scalability Methods for Computational Intellect. IGI Global, pp. 27–54.
Gao, X.J., Chong, L.S., Kim, M.S., Elowitz, M.B., 2018. Programmable protein circuits in living cells. Science 361 (6408), 1252–1258.
Gardner, T.S., Cantor, C.R., Collins, J.J., 2000. Construction of a genetic toggle switch in Escherichia coli. Nature 403, 339–342.
Gargantilla Becerra, Á., Gutiérrez, M., Lahoz-Beltra, R., 2021. A synthetic biology approach for the design of genetic algorithms with bacterial agents. Int. J. Parallel Emergent Distrib. Syst. 36 (3), 275–292.
Genot, A.J., Fujii, T., Rondelez, Y., 2013. Scaling down DNA circuits with competitive neural networks. J. R. Soc. Interface 10 (85), 20130212.
Gentili Pier Luigi, S.P., 2022. Chemical neural networks inside synthetic cells? A proposal for their realization and modeling. Front. Bioeng. Biotechnol. 10.
Gersbach, C.A., Gaj, T., Barbas III, C.F., 2014. Synthetic zinc finger proteins: The advent of targeted gene regulation and genome modification technologies. Acc. Chem. Res. 47 (8), 2309–2318.
Green, A.A., Silver, P.A., Collins, J.J., Yin, P., 2014. Toehold switches: De-novo-designed regulators of gene expression. Cell 159 (4), 925–939.
Groves, B., Chen, Y.-J., Zurla, C., Pochekailov, S., Kirschman, J.L., Santangelo, P.J., Seelig, G., 2016. Computing in mammalian cells with nucleic acid strand exchange. Nat. Nanotechnol. 11 (3), 287–294.
Gutiérrez, M., Gregorio-Godoy, P., Perez del Pulgar, G., Muñoz, L.E., Sáez, S., Rodríguez-Patón, A., 2017. A new improved and extended version of the multicell bacterial simulator gro. ACS Synth. Biol. 6 (8), 1496–1508.
Holt, B.A., Kwong, G.A., 2020. Protease circuits for processing biological information. Nature Commun. 11 (1), 1–12.
Ivakhnenko, A.G., 1968. The group method of data handling, a rival of the method of stochastic approximation. Sov. Autom. Control 13 (3), 43–55.
Jang, S.S., Oishi, K.T., Egbert, R.G., Klavins, E., 2012. Specification and simulation of synthetic multicelled behaviors. ACS Synth. Biol. 1 (8), 365–374.
Jiang, W., He, X., Luo, Y., Mu, Y., Gu, F., Liang, Q., Qi, Q., 2020. Two completely orthogonal quorum sensing systems with self-produced autoinducers enable automatic delayed cascade control. ACS Synth. Biol. 9 (9), 2588–2599.
Jones, T.S., Oliveira, S., Myers, C.J., Voigt, C.A., Densmore, D., 2022. Genetic circuit design automation with Cello 2.0. Nat. Protoc. 17, 1097–1113.
Jones, B., Stekel, D., Rowe, J., Fernando, C., 2007. Is there a liquid state machine in the bacterium *Escherichia coli*? In: 2007 IEEE Symposium on Artificial Life. pp. 187–191.
Karkaria, B.D., Treloar, N.J., Barnes, C.P., Fedorec, A.J., 2020. From microbial communities to distributed computing systems. Front. Bioeng. Biotechnol. 8, 834.
Katoch, S., Chauhan, S.S., Kumar, V., 2021. A review on genetic algorithm: Past, present, and future. Multimedia Tools Appl. 80 (5), 8091–8126.
Kawasaki, S., Ono, H., Hirosawa, M., Saito, H., 2020. RNA and protein-based nanodevices for mammalian post-transcriptional circuits. Curr. Opin. Biotechnol. 63, 99–110.
Khan, M.A., Herring, G., Oliva, M., Fourie, E., Zhu, J.Y., Johnston, B., Pflüger, J., Swain, T., Pflüger, C., Lloyd, J., et al., 2022. CRISPRi-based circuits for genetic computation in plants. Preprint at https://www.biorxiv.org/content/10.1101/2022.07.01.498372v1.
Kieffer, C., Genot, A.J., Rondelez, Y., Gines, G., 2023. Molecular computation for molecular classification. Adv. Biol. 7 (3), 2200203.
Kim, J., White, K.S., Winfree, E., 2006. Construction of an in vitro bistable circuit from synthetic transcriptional switches. Mol. Syst. Biol. 2 (1), 68.
Kis, Z., Pereira, H.S., Homma, T., Pedrigi, R.M., Krams, R., 2015. Mammalian synthetic biology: Emerging medical applications. J. R. Soc. Interface 12 (106), 20141000.
Lakin, M.R., 2023. Design and simulation of a multilayer chemical neural network that learns via backpropagation. Artif. Life 29 (3), 308–335.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.
Li, X., Rizik, L., Kravchik, V., Khoury, M., Korin, N., Daniel, R., 2021. Synthetic neural-like computing in microbial consortia for pattern recognition. Nature Commun. 12 (1), 1–12.
Li, Y., Teng, X., Zhang, K., Deng, R., Li, J., 2019. RNA strand displacement responsive CRISPR/Cas9 system for mRNA sensing. Anal. Chem. 91 (6), 3989–3996.
Liu, C.C., Jewett, M.C., Chin, J.W., Voigt, C.A., 2018. Toward an orthogonal central dogma. Nat. Chem. Biol. 14 (2), 103–106.
Liu, X., Parhi, K.K., 2022a. DNA memristors and their application to reservoir computing. ACS Synth. Biol. 11 (6), 2202–2213.
Liu, X., Parhi, K.K., 2022b. Reservoir computing using DNA oscillators. ACS Synth. Biol. 11 (2), 780–787.
Lormeau, C., Rudolf, F., Stelling, J., 2021. A rationally engineered decoder of transient intracellular signals. Nature Commun. 12 (1), 1–8.
Lv, H., Li, Q., Shi, J., Fan, C., Wang, F., 2021. Biocomputing based on DNA strand displacement reactions. ChemPhysChem 22 (12), 1151–1166.
Maass, W., 2000. On the computational power of winner-take-all. Neural Comput. 12 (11), 2519–2535.
Macía, J., Posas, F., Solé, R.V., 2012. Distributed computation: The new wave of synthetic biology devices. Trends Biotechnol. 30 (6), 342–349.
Matsuura, S., Ono, H., Kawasaki, S., Kuang, Y., Fujita, Y., Saito, H., 2018. Synthetic RNA-based logic computation in mammalian cells. Nature Commun. 9 (1), 4847.
McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5 (4), 115–133.
McEliece, R., Posner, E., Rodemich, E., Venkatesh, S., 1987. The capacity of the Hopfield associative memory. IEEE Trans. Inf. Theory 33 (4), 461–482.
Montagne, K., Plasson, R., Sakai, Y., Fujii, T., Rondelez, Y., 2011. Programming an in vitro DNA oscillator using a molecular networking strategy. Mol. Syst. Biol. 7 (1), 466.
Moore, R., Chandrahas, A., Bleris, L., 2014. Transcription activator-like effectors: A toolkit for synthetic biology. ACS Synth. Biol. 3 (10), 708–716.
Moškon, M., Komac, R., Zimic, N., Mraz, M., 2021. Distributed biological computation: From oscillators, logic gates and switches to a multicellular processor and neural computing applications. Neural Comput. Appl. 33, 8923–8938.
Moškon, Mraz, M., 2022. Programmable evolution of computing circuits in cellular populations. Neural Comput. Appl..
Munakata, Y., Pfaffly, J., 2004. Hebbian learning and development. Dev. Sci. 7 (2), 141–148.
Nagipogu, R.T., Fu, D., Reif, J.H., 2023. A survey on molecular-scale learning systems with relevance to DNA computing. Nanoscale 15 (17), 7676–7694.
Nikolić, V., Echlin, M., Aguilar, B., Shmulevich, I., 2023. Computational capabilities of a multicellular reservoir computing system. PLoS One 18 (4), e0282122.
Okumura, S., Gines, G., Lobato-Dauzier, N., Baccouche, A., Deteix, R., Fujii, T., Rondelez, Y., Genot, A., 2022. Nonlinear decision-making with enzymatic neural networks. Nature 610 (7932), 496–501.
Ortiz, Y., Carrión, J., Lahoz-Beltrá, R., Gutiérrez, M., 2021. A framework for implementing metaheuristic algorithms using intercellular communication. Front. Bioeng. Biotechnol. 9, 330.
Paik, J.K., Katsaggelos, A.K., 1992. Image restoration using a modified Hopfield network. IEEE Trans. Image Process. 1 (1), 49–63.
Pei, L., Garfinkel, M., Schmidt, M., 2022. Bottlenecks and opportunities for synthetic biology biosafety standards. Nature Commun. 13 (1), 2175.
Pei, R., Matamoros, E., Liu, M., Stefanovic, D., Stojanovic, M.N., 2010. Training a molecular automaton to play a game. Nat. Nanotechnol. 5 (11), 773–777.
Prakash, S., Racovita, A., Varela, C., Walsh, M., Galizi, R., Isalan, M., Jaramillo, A., 2023. Engineered sensor bacteria evolve master-level gameplay through accelerated adaptation. Preprint at http://dx.doi.org/10.1101/2022.04.22.489191.
Qian, L., Winfree, E., Bruck, J., 2011. Neural network computation with DNA strand displacement cascades. Nature 475 (7356), 368–372.
Regot, S., Macia, J., Conde, N., Furukawa, K., Kjellén, J., Peeters, T., Hohmann, S., De Nadal, E., Posas, F., Solé, R., 2011. Distributed biological computation with multicellular engineered networks. Nature 469 (7329), 207–211.
Rizik, L., Danial, L., Habib, M., Weiss, R., Daniel, R., 2022. Synthetic neuromorphic computing in living cells. Nature Commun. 13 (1), 1–17.
Rodriguez, K.R., Sarraf, N., Qian, L., 2021. A loser-take-all DNA circuit. ACS Synth. Biol. 10 (11), 2878–2885.
Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. Psychol. Rev. 65 (6), 386.
Rubens, J.R., Selvaggio, G., Lu, T.K., 2016. Synthetic mixed-signal computation in living cells. Nature Commun. 7 (1), 1–10.
Samaniego, C.C., Moorman, A., Giordano, G., Franco, E., 2021. Signaling-based neural networks for cellular computation. In: 2021 American Control Conference. ACC, pp. 1883–1890.
Sarkar, K., Bonnerjee, D., Srivastava, R., Bagh, S., 2021. A single layer artificial neural network type architecture with molecular engineered bacteria for reversible and irreversible computing. Chem. Sci. 12, 15821–15832.
Schaffter, S.W., Strychalski, E.A., 2022. Cotranscriptionally encoded RNA strand displacement circuits. Sci. Adv. 8 (12), eabl4354.
Schrauwen, B., Verstraeten, D., Van Campenhout, J., 2007. An overview of reservoir computing: Theory, applications and implementations. In: Proceedings of the 15th European Symposium on Artificial Neural Networks. pp. 471–482, 2007, pp. 471–482.
Scown, C.D., Keasling, J.D., 2022. Sustainable manufacturing with synthetic biology. Nat. Biotechnol. 40 (3), 304–307.
Seelig, G., Soloveichik, D., Zhang, D.Y., Winfree, E., 2006. Enzyme-free nucleic acid logic circuits. Science 314 (5805), 1585–1588.
Sidiropoulou, K., Pissadaki, E.K., Poirazi, P., 2006. Inside the brain of a neuron. EMBO Rep. 7 (9), 886–892.
Song, X., Reif, J., 2019. Nucleic acid databases and molecular-scale computing. ACS Nano 13 (6), 6256–6268.
Srivastava, R., Bagh, S., 2023. A logically reversible double Feynman gate with molecular engineered bacteria arranged in an artificial neural network-type architecture. ACS Synth. Biol. 12 (1), 51–60.

Tamsir, A., Tabor, J.J., Voigt, C.A., 2011. Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'. Nature 469 (7329), 212–215.

Tekel, S.J., Smith, C.L., Lopez, B., Mani, A., Connot, C., Livingstone, X., Haynes, K.A., 2019. Engineered orthogonal quorum sensing systems for synthetic gene regulation in Escherichia coli. Front. Bioeng. Biotechnol. 7.

Teo, J.J., Woo, S.S., Sarpeshkar, R., 2015. Synthetic biology: A unifying view and review using analog circuits. IEEE Trans. Biomed. Circuits Syst. 9 (4), 453–474.

TerAvest, M.A., Li, Z., Angenent, L.T., 2011. Bacteria-based biocomputing with cellular computing circuits to sense, decide, signal, and act. Energy Environ. Sci. 4 (12), 4907–4916.

van der Linden, A.J., Pieters, P.A., Bartelds, M.W., Nathalia, B.L., Yin, P., Huck, W.T., Kim, J., de Greef, T.F., 2022. DNA input classification by a riboregulator-based cell-free perceptron. ACS Synth. Biol. 11 (4), 1510–1520.

Vasić, M., Chalk, C., Khurshid, S., Soloveichik, D., 2020. Deep molecular programming: A natural implementation of binary-weight ReLU neural networks. In: International Conference on Machine Learning. PMLR, pp. 9701–9711.

Vasić, M., Chalk, C., Luchsinger, A., Khurshid, S., Soloveichik, D., 2022. Programming and training rate-independent chemical reaction networks. Proc. Natl. Acad. Sci. USA 119 (24), e2111552119.

Wagner, T.E., Becraft, J.R., Bodner, K., Teague, B., Zhang, X., Woo, A., Porter, E., Alburquerque, B., Dobosh, B., Andries, O., Sanders, N.N., Beal, J., Densmore, D., Kitada, T., Weiss, R., 2018. Small-molecule-based regulation of RNA-delivered circuits in mammalian cells. Nat. Chem. Biol. 14 (11), 1043–1050.

Wakabayashi, K., Yamamura, M., 2005. A design for cellular evolutionary computation by using bacteria. Nat. Comput. 4 (3), 275–292.

Wang, S.-C., 2003. Interdisciplinary Computing in Java Programming. Springer US, Boston, MA, pp. 81–100, Ch. Artificial Neural Network.

Werbos, P., 1974. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences (Ph. D. dissertation). Harvard University.

Wroblewska, L., Kitada, T., Endo, K., Siciliano, V., Stillo, B., Saito, H., Weiss, R., 2015. Mammalian synthetic circuits with RNA binding proteins for RNA-only delivery. Nat. Biotechnol. 33 (8), 839–841.

Wythoff, B.J., 1993. Backpropagation neural networks: A tutorial. Chemom. Intell. Lab. Syst. 18 (2), 115–155.

Xiong, X., Zhu, T., Zhu, Y., Cao, M., Xiao, J., Li, L., Wang, F., Fan, C., Pei, H., 2022. Molecular convolutional neural networks with DNA regulatory circuits. Nat. Mach. Intell. 4 (7), 625–635.

Yeo, N.C., Chavez, A., Lance-Byrne, A., Chan, Y., Menn, D., Milanova, D., Kuo, C.-C., Guo, X., Sharma, S., Tung, A., et al., 2018. An enhanced CRISPR repressor for targeted mammalian gene regulation. Nature Methods 15 (8), 611.

Young, T., Hazarika, D., Poria, S., Cambria, E., 2018. Recent trends in deep learning based natural language processing. IEEE Comput. Intell. Mag. 13 (3), 55–75.

Zhang, Z., Geiger, J., Pohjalainen, J., Mousa, A.E.-D., Jin, W., Schuller, B., 2018. Deep learning for environmentally robust speech recognition: An overview of recent developments. ACM Trans. Intell. Syst. Technol. 9 (5), 1–28.

Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al., 2023. A survey of large language models. arXiv preprint arXiv: 2303.18223.