

# Synthetic neural-like computing in microbial consortia for pattern recognition

Ximing Li

Technion-Israel Institute of Technology

Luna Rizik

Technion-Israel Institute of Technology

Ramez Daniel (✉ [ramizda@bm.technion.ac.il](mailto:ramizda@bm.technion.ac.il))

Technion-Israel Institute of Technology

---

## Article

**Keywords:** artificial neural networks, ANN-like computing, microbial consortia, pattern recognition

**Posted Date:** September 28th, 2020

**DOI:** <https://doi.org/10.21203/rs.3.rs-82365/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

**Version of Record:** A version of this preprint was published at Nature Communications on May 25th, 2021. See the published version at <https://doi.org/10.1038/s41467-021-23336-0>.

# Synthetic neural-like computing in microbial consortia for pattern recognition

Ximing Li, Luna Rizik, Ramez Daniel\*

*Department of Biomedical Engineering  
Technion—Israel Institute of Technology  
Technion City, Haifa 32000, Israel*

## Abstract

Complex biological systems in nature comprise of cells that act collectively to solve sophisticated tasks. Synthetic biological systems, in contrast, are designed for specific tasks, largely following computational principles including logic gates, analog design, and control theory. Yet such approaches cannot be easily adapted for multiple tasks in biological contexts. Alternatively, artificial neural networks (ANN), comprised of flexible interactions for processing and decision-making, are widely adopted for numerous applications and support adaptive designs. Motivated by the structural similarity between ANNs and cellular networks, here we implemented ANN-like computing in bacteria consortia for recognizing patterns. In cellular ANNs, receiver bacteria collectively interact through quorum sensing (QS) with sender bacteria for decision-making processes. Input patterns formed by chemical inducers, activate sender circuits to produce QS signaling molecules with varying levels. These levels are programmed by tuning the promoter strength acting as weights. We also developed an algorithm based on gradient descent, which is well-accepted in artificial intelligence, to optimize weights and experimentally examined them using  $3 \times 3$ -bit patterns.

## 18 Introduction

Living systems are constantly engaged in computational processes such as detection, signal processing, and decision making to perform sophisticated tasks [1]. For example, in the vertebrate adaptive immune system, the invasion of pathogens triggers a series of actions from multiple cell types to protect the organisms [1]. The computational properties of biological systems can emerge from coordinative and collective interactions among basic components [2, 3]. These components can be neurons interacting with

25 other cells in the brain, bacteria communicating with other members in a community, or  
26 receptors participating in signaling pathways [4].

27 In contrast to natural living systems, synthetic biocircuits excel at only a narrow range  
28 of human-defined computations [2, 3]. One reason is that they are designed following  
29 principles from computer engineering, represented by implementations such as toggle  
30 switches [5], oscillators [6], memory devices [7] and state machines [8]. A few major  
31 computational paradigms have been widely adopted for circuit design, namely digital  
32 design, analog design, and control design [9–12]. Digital design takes inputs of binary-  
33 coded levels, highlighting concepts such as logic gates and Boolean functions. Analog  
34 design and control design handle a range of continuous input levels, focusing on system  
35 stability and design dynamics. Nevertheless, depending on design paradigms, synthetic  
36 gene circuits face challenges such as host limitations, random fluctuations, and unwanted  
37 interactions with host cells [13, 14]. Some studies have exploited the dynamic structure of  
38 cell communities and obtained more sophisticated functions than in individual cells [15,  
39 16]. Multi-cellular systems naturally allow distributed and parallel computing. Using  
40 these features, studies have successfully implemented edge detection [17] and spatial  
41 pattern formation [18, 19]. Furthermore, cells in communities can be organized with  
42 flexibility, such as being layered for logic gates [20, 21] or arranged to form various  
43 ecosystems [3].

44 So far, synthetic biocircuits are often designed for specific tasks and cannot easily be  
45 adapted for solving a wide range of problems. However, as the synthetic biology commu-  
46 nity attempts to create 'smart cells' for a variety of applications [22], it is important to  
47 build circuits that can be adapted and optimized without explicit programming. Multi-  
48 cellular systems provide a solution to this issue. In these systems, computations can nat-  
49 urally emerge from flexibly interconnected cells that act concertedly. The flexibility and  
50 interconnection, similar to the structure in neural networks, offer a novel design that can  
51 be adapted to solve a range of problems, overcoming the lack of generality in mainstream  
52 paradigms. Inspired by biological neural networks, artificial neural networks (ANNs) are  
53 adaptive computing models that are commonly adopted to solve a wide range of tasks [23].  
54 In this study, we draw an analogy from inter-cellular relations to ANNs and demonstrate  
55 that ANN provides a powerful design to engineer multi-cellular systems. ANNs model  
56 the network structure with layers of connected units. The connecting strengths between  
57 units, namely the weights, can be trained to achieve specific tasks. This trainable feature  
58 allows ANNs to 'learn' the weight values, so that tasks involving decision-making, such as  
59 pattern recognition, can be learned. A simple unit of ANN is perceptron (Fig. 1a), which  
60 performs a weighted summation of inputs, and becomes activated for decisions. Despite  
61 the simplicity, perceptrons can classify patterns that are linearly separable, which means  
62 input points on a plane belonging to distinct categories can be geometrically separated  
63 by a line (or a hyper-plane for patterns in high dimensions). By modifying weights, a

64 perceptron unit can be used for pattern recognition.

65 Specifically, here we explore the collective behaviors in *Escherichia coli* (*E. coli*) bac-  
66 teria cell consortia, demonstrating that the interactions between cell groups can be used  
67 to implement a perceptron network for pattern classification (Fig. 1b). The patterns  
68 are represented by the amount of inducer OC6 (acyl-homoserine lactone 3OC6-HSL)  
69 described in binary levels, either with or without inducer. The inducer can activate a  
70 group of sender bacteria to produce QS signaling molecules OHC14 (acyl-homoserine lac-  
71 tone 3OHC14:1-HSL), which diffuse into and activate the receivers. The receivers provide  
72 the activation function to convert the weighted sum of OHC14 collectively produced from  
73 senders into different activation states for classifying input patterns. We vary the strength  
74 of the  $P_{lux}$  promoter in senders to obtain different weights. We first experimentally exam-  
75 ined the QS system using simple 4-bit patterns. Then we developed an algorithm based  
76 on gradient descent, which is widely used in artificial intelligence, to optimize weights  
77 so that more sophisticated patterns could be categorized. Using our algorithm, we ob-  
78 tained the weights for 3×3-bit patterns and experimentally tested the weights using the  
79 QS system for these patterns. Our implementation demonstrates a framework to train  
80 genetic circuits *in silico* and satisfy the target functions *in vivo* for pattern recognition.  
81 This implementation provides a prototype to implement neural-network-like computing  
82 in living bacteria.

## 83 Results

### 84 Circuit engineering in senders and receivers

85 The basic parts for senders and receivers are shown in Fig. 1c. In senders, when OC6  
86 is given as an input, the constitutively expressed LuxR binds with OC6 to activate ,  
87 allowing *cinI* and *mCherry* to express. Expression of *cinI* catalyzes the synthesis of  
88 OHC14, which diffuses into receivers across the media. In receivers, OHC14 binds with  
89 the constitutively expressed CinR transcription factor, forming a complex that activates  
90 the  $P_{cin}$  promoter. The activity of the  $P_{cin}$  is measured by Enhanced Yellow Fluorescence  
91 Protein (EYFP) signals. However, when mixing senders exposed to different inputs, i.e.,  
92 with or without OC6, a cross-talk can occur between these senders. Specifically, senders  
93 that are initially inactivated ('0') may be affected by the residual OC6 from those that are  
94 activated ('1'). To minimize such a cross-talk, we added a double inversion circuit ( $P_{rod}$ -  
95 TetR and  $P_{tetO}$ -LacI) to the sender circuit (Fig. 2a). The additional circuit is designed  
96 to control the activity of  $P_{lacO}$  using anhydrotetracycline (aTc) to affect the expression  
97 of LuxR and the activity of  $P_{lux}$ . In the double inversion circuit, TetR is constitutively  
98 produced to repress  $P_{tetO}$  and block the LacI expression. Therefore, without aTc, LacI  
99 is not expressed and  $P_{lacO}$  remains constitutively on to express LuxR. In the presence of

100 aTc, TetR binds aTc and releases from  $P_{tetO}$ , allowing LacI to express. In this case, LacI  
 101 represses  $P_{lacO}$  and blocks the LuxR expression. Therefore, the presence of OC6 in the  
 102 cells cannot affect the expression levels of *cinI* and *mCherry*.

103 To improve the output dynamic range of the double inversion circuit, we added decoy  
 104 binding sites (Fig. 2a). In particular, we implemented six decoy lacO operators ( $P_{lacO}$   
 105 array) to sequester extra LacI produced from  $P_{tetO}$  [24]. Specifically, the  $P_{lacO}$  that  
 106 expresses LuxR is on a low-copy-number plasmid (LCP), and the decoy  $P_{lacO}$  array is on  
 107 a medium-copy-number plasmid (MCP). The sequestering effect can adjust the output  
 108 range of the double inversion circuit. As shown in Fig. 2b, with the  $P_{lacO}$  array on MCP,  
 109 aTc (200 ng ml<sup>-1</sup>) reduces Green Fluorescent Protein (GFP) expression (i.e., the activity  
 110 of  $P_{lacO}$  on LCP) by approximately 10 times (Fig. 2b, green bars). In contrast, without  
 111 the  $P_{lacO}$  array (Fig. 2b, orange bars), GFP expression is constantly low regardless of  
 112 aTc levels.

113 The purpose of the activation function in perceptron is to nonlinearly map the weighted  
 114 summation into separate states for decision making (Fig. 2c). Therefore, the steeper the  
 115 activation function, the sharper the decision boundary, which is easier for the network to  
 116 make decisions. In order to improve the steepness of the activation function, we imple-  
 117 mented a positive feedback regulation in receivers, which sharpens the transfer function  
 118 of receiver circuits (Fig. 2d). As shown in Table 1, the positive feedback circuit exhibits  
 119 a higher hill coefficient (n=1.97 in column R\_PF) than the open-loop circuit (n=0.809  
 120 in column R\_OL).

Table 1: Fittings parameters using Hill equations for transfer functions of receiver and sender circuits.

	R_PF	R_OL	S_mut40	S_mut7	S_mut8	S_mut15	S_plux_rep	R_mCh
$K_d$	0.28 $\mu$ M	3.41 $\mu$ M	15 $\mu$ M	50 $\mu$ M	50 $\mu$ M	50 $\mu$ M	0.252 $\mu$ M	1103
$\beta_0$	3.95E-02	7.23E-03	4.95E-03	9.1E-02	1.18E-01	4.45E-02	3.41E-01	1.95E-02
$\beta_m$	2.25E+04	2.57E+04	5521	1906	1119	2679	735	2.16E04
$n$	1.97	0.809	0.354	0.482	0.5	0.426	0.669	2.33

R\_PF: the transfer function of receiver circuits with positive feedback (Fig. 2d).

R\_OL: the transfer function of receiver circuits with open loop (Fig. 2d).

Column 3-7: transfer functions of senders with mutated  $P_{lux}$  (Fig. 2e).

Column 8: transfer function of receiver circuits with horizontal axis as mCherry levels (Fig.4c).

All fittings except for 'S\_plux\_rep' are based on the formula in the activation form  $f(x) = \beta_m \frac{(x/K_d)^n}{1+(x/K_d)^n} + \beta_m \beta_0$ , where  $x$  represents the inducer amount (for R\_mch,  $x$  is mCherry level),  $K_d$  is the dissociation constant,  $\beta_m$  is the max activity of promoter,  $\beta_0 * \beta_m$  is the basal activity of the promoter and  $n$  is the Hill coefficient, representing the cooperativity of TF binding with the promoter.

'S\_plux\_rep' is fitted using the formula in the repression form  $f(x) = \alpha_m \frac{1}{1+(x/K_d)^n} + \alpha_0 \alpha_m$ , where  $K_d$  and  $n$  are the same as as previously described. Similar to  $\beta_m$  and  $\beta_0 * \beta_m$ ,  $\alpha_m$  and  $\alpha_0 * \alpha_m$  describe the max activity and minimal activity of the  $P_{lux}$  repressor promoter.

121 The perceptron weights can be implemented by varying the promoter strength to affect  
 122 the transfer function of sender circuits in response to OC6. To obtain various perceptron  
 123 weights, we mutated the first four base pairs in  $P_{lux}$  and obtained promoters with different  
 124 strengths (Fig. 2e). These mutated senders comprise various positive weights in the  
 125 perceptron network. To build more sophisticated functions, we also implemented a weight

126 with a negative sign. For example, a XOR logic function can be implemented with two-  
127 perceptron layers consisting of negative and positive weights [25]. Here, we used the  
128  $P_{lux}$  repressor promoter to engineer the negative weight by placing the LuxR binding site  
129 downstream to a strong constitutive promoter (Fig. 2e). The arrangement allows binding  
130 of the OC6-LuxR complex to the promoter to block RNA Polymerase from initiating the  
131 transcription. The transfer functions of sender circuits containing  $P_{lux}$  activators and the  
132  $P_{lux}$  repressor were experimentally measured and fitted using the Hill equation (Table 1).

### 133 Classification of 4-bit patterns

134 With the engineered circuits in senders and receivers, we next examined whether they can  
135 be built into a perceptron to classify patterns of OC6. We started with an arbitrary weight  
136 vector, selected the patterns that can be classified by the weights and implemented using  
137 previously built genetic circuits. The weight vector [450, 3500, 900, 3500] was selected  
138 (Fig. 3a). The values correspond to estimates of mCherry levels measured from mutated  
139 senders after 210 min incubation when the OC6 concentration is 33  $\mu\text{M}$  (Fig. 2e). This  
140 OC6 level is in the high inducer analog range (33  $\mu\text{M}$  isn't high enough to saturate the  
141 mutated  $P_{lux}$  activity, where the max level used in Fig. 2e is 100  $\mu\text{M}$ ). There are in  
142 total twenty four 4-bit binary patterns, a subset of which can be well classified by the  
143 weight vector. For each pattern, we calculated the product of the weight vector and the  
144 pattern, and compared the product value with thresholds for decision making. Patterns  
145 with product values larger than an upper threshold are grouped as one class. Likewise,  
146 patterns with product values smaller than a lower threshold are grouped as another class.  
147 The selected patterns are shown in Fig. 3a. Further explanation of how a perceptron  
148 with the weight vector can classify 4-bit patterns is provided in the Supplementary Notes.

149 Senders grown overnight were diluted 100 times and incubated at 37 °C in separate  
150 wells for 210 min (Fig. 3b, further information is provided in Methods). For input bits '1',  
151 senders were incubated in LB containing OC6. For input bits '0', senders were incubated  
152 in LB containing aTc to reduce the unwanted cross-talk between senders after mixing.  
153 Next, all senders were diluted 100 times in LB, incubated for 30 min, and mixed together.  
154 The sender mixture was then transferred into receiver solutions (50 times diluted from  
155 overnight culture) in a 1-to-19 volume ratio and incubated at 37 °C for 90 min. Finally,  
156 EYFP and BFP signals from the incubated sample were measured using flow cytometry.  
157 In Fig. 3b, the input pattern  $\vec{p}_8$  ([1,0,1,1]) is demonstrated as an example. Note that  
158 pattern  $\vec{p}_{10}$  ([1,1,1,0]) also yields the same sender mixture, i.e., a mix of mut8 with OC6,  
159 mut40, mut7 with OC6, and mut40 with OC6.

160 Median EYFP levels from receivers were averaged from three trials for all patterns  
161 (Fig. 3c). As expected, receivers are activated to higher levels by patterns in set1 ( $\vec{p}_6$   
162 to  $\vec{p}_{11}$ ) than those in set0 ( $\vec{p}_0$  to  $\vec{p}_5$ ). The BFP expression levels are uniform across

163 patterns, indicating that the distinctions among observed EYFP signals are not due to  
164 the variability in receiver growth or a disproportionate mix of senders and receivers.  
165 These results suggest that OHC14 collectively produced by senders with various  $P_{lux}$   
166 mutations (i.e., weights) can be effectively summated in receivers. The activation function  
167 of receivers with a positive feedback loop was capable of classify the input patterns.

## 168 **Developing an algorithm to learn the weights**

169 Subsequently, we developed an algorithm to systematically search for  $P_{lux}$  promoter  
170 strength that can map patterns onto target activation levels in receivers. We modelled  
171 the system in a series of steps (Fig. 4a), following the procedures in experiments. First,  
172 we started with senders with random weights. Selected senders were induced by chemical  
173 inputs, then mixed and incubated together with receivers. Next, the receiver output lev-  
174 els were compared with predefined target levels for error calculation (i.e., loss function)  
175 to update weights. Newly updated weights were then fed into the model again to iterate  
176 the steps until the receiver outputs became sufficiently close to targets.

177 This iterative approach is a common algorithm used to train ANNs [26] based on the  
178 gradient descent method. Intuitively, consider the difference between the model output  
179 and target as a mountain ridge with changing slopes (Fig. 4b). The algorithm, like a  
180 hiker, aims to find the valley of the ridge, i.e., to minimize error. Starting at a random  
181 point, the hiker chooses the direction for the next step based on the slope at that point.  
182 Following the downward slope, step by step, the hiker can gradually reach the lowest point.  
183 The curve shown in Fig. 4b is simplified when optimizing one parameter. When multiple  
184 parameters are optimized, as in the case of a multi-element weight vector, we need to  
185 search the lowest point on a high dimensional 'mountain' surface. For a particular point  
186 on the surface, there can be multiple downward directions. The algorithm thus follows  
187 the steepest downward direction by calculating the gradient of error.

188 However, simply following the gradient, the solution can sometimes become trapped in  
189 a local minimal point, if the step size is not sufficiently large. In this case, the performance  
190 of the algorithm depends on the starting position and step size. When the error surface  
191 became too bumpy, we performed the search in two steps. First, using the gradient-  
192 based method, we reach a sub-optimal solution. Second, using a direct search in a  
193 nearby area we looked for the optimal solution. We present the details of the algorithm  
194 in the Supplementary Notes. In particular, the activation function  $\sigma$  of receivers was  
195 experimentally measured (Fig. 4c), using the sender mCherry levels as input.

## 196 **Classification of $3 \times 3$ -bit patterns**

197 Following the two-step method to learn weights, we performed classification of more  
198 complicated pattern sets (Fig. 5a) than the 4-bit patterns, using the genetic circuits

199 constructed previously. We choose three sets of  $3\times 3$ -bit patterns that have previously  
 200 been used for classification with state-of-the art electronic circuits [27]. The pattern  
 201 sets include three categories, 'z', 'v' and 'n'. Each set includes one ideal or noiseless  
 202 pattern and nine noisy ones with one bit flipped (Fig. 5a). We included three weight  
 203 vectors ( $\vec{w}_0, \vec{w}_1, \vec{w}_2$ ) to recognize the three pattern categories. In particular, the  $\vec{w}_0$  vector  
 204 generates an output vector approximate the target vector  $\vec{t}_0$  (Fig. 5a). Similarly, the  $\vec{w}_1$   
 205 and  $\vec{w}_2$  vectors can produce output vectors approximate  $\vec{t}_1$  and  $\vec{t}_2$ , respectively.

206 We obtained the three vectors using our two-step optimization method. The simu-  
 207 lated output is shown in Fig. 5b. Notably, step one alone can lead to relatively good  
 208 classification but the performance is highly sensitive to step size or learning rate and find-  
 209 ing an appropriate performance requires manual trials. In addition, many combinations  
 210 in the product of weights and inputs are repetitions (Fig. S6). Only ten combinations  
 211 are unique. Subsequently, we experimentally tested the ten non-repeating combinations  
 212 using selected weights. As shown in Fig. 5c, the first four products ( $\vec{w}_0^T \cdot \vec{p}_0, \vec{w}_0^T \cdot \vec{p}_1,$   
 213  $\vec{w}_0^T \cdot \vec{p}_3$  and  $\vec{w}_0^T \cdot \vec{p}_5$ ) result in high EYFP levels in receivers and the rest products result  
 214 in low output values. This is consistent with the simulation results in Fig. 5b, as the  
 215 products of  $\vec{w}_0$  and all patterns in 'z' ( $\vec{p}_0$  to  $\vec{p}_9$ ) are expected to be high. Meanwhile, the  
 216 BFP expression levels are uniform across all patterns.

## 217 Simulations of classifying more sophisticated patterns

218 Patterns in the real world are often larger than  $3\times 3$  pixels. To understand how the  
 219 algorithm performs on larger patterns, we further expanded the same pattern sets to  
 220  $5\times 5$ ,  $7\times 7$  and  $9\times 9$  bits. Similar to the  $3\times 3$ -bit patterns, each pattern set includes  
 221 one clean pattern and  $N^2$  noisy ones. Using only the step one algorithm, we obtained  
 222 good classifications for all patterns (Fig. 6a). Interestingly, in  $5\times 5$ -bit patterns, the  
 223 separations of high and low output values are better than in  $3\times 3$ -bit. The distinction is  
 224 further improved in  $7\times 7$ -bit and  $9\times 9$ -bit patterns. This is likely due to the refined image  
 225 resolution in large-scale patterns.

226 Circuits based on logic gates are often designed for patterns in binary values. One  
 227 advantage of ANN over logic gate design is that ANN also works on patterns with graded  
 228 values. To see whether our design can handle non-binary patterns, we modified the same  
 229  $5\times 5$ ,  $7\times 7$  and  $9\times 9$ -bit patterns by varying the '1' elements to random values from 0.5  
 230 to 1 (Fig. 6b). Simulated classification results using the step one algorithm indicated  
 231 that the algorithm can achieve good performance with non-binary patterns. The learning  
 232 rate needs manual adjustment for each pattern category to achieve satisfactory classifi-  
 233 cation. We list the relevant weight values that are obtained using the algorithm in the  
 234 Supplementary Tables.

## 235 Discussion

236 We demonstrated a genetic circuit implementation of perceptron networks based on bac-  
237 terial quorum sensing. The implementation allows *E. coli* to recognize chemical input  
238 patterns. Along with the genetic circuits, we also formulated an algorithm to obtain ap-  
239 propriate weights for pattern classification. We experimentally tested the implementation  
240 first using 4-bit patterns and then 3×3-bit patterns. In both experiments, the patterns  
241 were successfully classified. We further showed that the algorithm can solve more sophis-  
242 ticated patterns, which are larger in size and having non-binary inputs. The simulated  
243 results demonstrate the capability of the algorithm to tackle a range of pattern sets. The  
244 algorithm-guided approach provides an example that couples design *in silico* and imple-  
245 mentation *in vivo*. It is worthwhile pointing out that the genetic circuits are inherently  
246 noisy. In senders, the mCherry measurement can vary from trial to trial depending on  
247 the quality of growth media and antibiotics. Thus, it is not possible to match the exact  
248 weight values calculated using the algorithm. It is sufficient to select the mutated senders  
249 that are qualitatively different, that is, showing distinct fluorescence levels larger than  
250 noise levels. More importantly, through the experiments, we demonstrated that decision  
251 making in living cells does not rely on exact behaviors.

252 Previous works have implemented classifiers in living cells. Xie et al. [28] constructed  
253 logic circuits in single cells to identify cancer cells. Didovyk et al. [29] implemented a  
254 distributed classifier, in which the decision is made at a population scale. In particular,  
255 the latter design does not include inter-cellular communications and there is no particular  
256 cell group in charge of decision making. Different from these previous works, our design  
257 is based on communications between two groups of cells and demonstrates the power of  
258 multi-cellular computing. The two cell groups, i.e., senders and receivers, encapsulate  
259 distinct computational functions (e.g. weighting and summations) in separate compart-  
260 ments. In this way, cellular functions are separated in modules and metabolic burdens  
261 are distributed across species, allowing for increased system flexibility and scalability.  
262 With the design, computational behaviors arise from the interactions between senders  
263 and receivers, even though the tasks performed in individual cell groups are relatively  
264 primitive. Similar themes have also been presented in other studies for majority sensing  
265 [30] and tunable population dynamics [31].

266 ANNs are the enabling tool for today’s artificial intelligence technology. In computer  
267 engineering, its capability of learning offers a great advantage over traditional combina-  
268 torial logic circuits. Recently, ANNs have been implemented alternatively in memristors  
269 [27] (i.e., a new form of electronics), optics [32], DNA strands [33] and cell-free systems  
270 [34]. Likewise, as a computing model, ANNs could also provide a design architecture  
271 in synthetic biology to engineer biological systems with more adaptivity. For example,  
272 the framework and algorithm in our study can be potentially extended to engineer inter-

273 cellular communications in yeasts cells [35] and mammalian cells [19]. For the latter in  
274 particular, engineering how tissue cells contact each other would enable new applications  
275 for programming tissue development, growth and repair [19].

## 276 **Methods**

### 277 **Chemicals**

278 All chemicals used in the study are of the highest analytical grade. OC6 was obtained  
279 from Sigma Aldrich. OHC14 (N-(3-hydroxy-7-cis tetradecenoyl)-L-Homoserine lactone)  
280 was obtained from Cayman Chemical Company. Anhydrotetracycline (aTc) was from  
281 Takara Bio.

### 282 **Strains**

283 *E. coli* 10 $\beta$  was used for plasmid construction and all experiment assays. All liquid  
284 media used in the study was Luria-Bertani-Miller (LB). Two types of antibiotics were  
285 used, kanamycin (30  $\mu\text{g ml}^{-1}$ ) and cloramphenicol (25  $\mu\text{g ml}^{-1}$ ). The specifics *E. coli* 10 $\beta$   
286 is: araD139 D (ara-leu) 7697 fhuA lacX74 galK (W80 D (lacZ) M15) mcrA galU recA1  
287 endA1 nupG rpsL (StrR) D (mrr-hsdRMS-mcrBC).

### 288 **Plasmids construction**

289 The plasmid pAJM1642 containing the essential parts in receivers ( $P_{cin}$ -EYFP and  $P_{lacI}$ -  
290 CinR) was obtained from Christopher Voigt's Laboratory (Addgene plasmid #108535;  
291 <http://n2t.net/addgene:108535>; RRID:Addgene\_108535) [36]. All other plasmids were  
292 constructed using basic molecular cloning methods [37], including standard steps like  
293 PCR, restriction digestion, ligation and transformation. PCR was carried out in a Bio-  
294 Rad S1000 Thermal Cycler. Oligonucleotide primers were synthesized by Integrated  
295 DNA Technologies (Coralville, IA). Restriction digestion enzymes were purchased from  
296 New England Biolabs (Beverly, MA) and Thermo Scientific FastDigest. Ligations were  
297 performed using T4 DNA ligase with ligation buffer from New England Biolabs. For  
298 transformation, we used standard heat shock in *E. coli* 10 $\beta$ , followed by colony PCR  
299 screening on the next day. Selected colonies were grown overnight for miniprep (BioBa-  
300 sic) and sent for standard sequencing (Macrogen Europe, The Netherlands). Mutations  
301 in the  $P_{lux}$  promoter were performed using site-directed mutation (Agilent QuickChange  
302 lightning), following the manufacture's protocol. Mutations were first performed in a  
303 simple circuit,  $P_{lux}$ -GFP- $P_{lacO}$ -LuxR. Transfer functions of mutated colonies were char-  
304 acterized. The mutations with desired characteristics were then selected for sequencing  
305 and integrated with the other parts in senders.

## 306 Experiments using FACS

307 Raw data from FACS presented in the work contain 10000 events and the abort rate was  
308 kept less than 2%. Raw data were pre-processed using density-based gating by a Python  
309 library FlowCal [38] to obtain 8000 events. All circuit diagrams were drawn using a  
310 Python package DNAPlotlib [39].

### 311 Cross-talk experiment (Fig. 2b)

312 Relevant plasmids used in the experiments were XL267+XL208 (with  $P_{laco}$ -array), XL267+LR206  
313 (without  $P_{laco}$ -array) (see Table S1 for complete list). *E. coli* colonies from transforma-  
314 tion plates were inoculated in 4 ml LB solution with 4 ml kanamycin ( $30 \mu\text{g ml}^{-1}$ ) and 4  
315 ml chloramphenicol ( $25 \mu\text{g ml}^{-1}$ ), grown overnight at  $37^\circ\text{C}$  in a shaking incubator (Shel  
316 Labs SSI5) at 250 rpm. On the next day, overnight cultures were diluted 100 times in  
317 LB antibiotics with or without aTc (final concentration  $200 \text{ ng ml}^{-1}$ ). Aliquots ( $200 \mu\text{l}$ )  
318 of diluted culture were transferred into 96-well plates and incubated at  $37^\circ\text{C}$  for 210 min  
319 in a microplate shaker (Lumitron) at 500 rpm.

### 320 Receivers characterization experiment (Fig. 2d)

321 Relevant plasmids used in the experiments were XL340+XL291 (positive feedback) and  
322 pAJM1642+LR191 (open loop). *E. coli* colonies were grown overnight at  $37^\circ\text{C}$  in 4 ml  
323 LB solution with the appropriate antibiotic combinations. Three colonies were grown for  
324 each circuit. Overnight cultures were diluted 50 times and aliquoted into a 96-well plate.  
325 Inducer OHC14 (Cayman Chemical Company) was added (final max concentration  $33.3$   
326  $\mu\text{M}$ ) and then serial diluted three times across wells for each colony. The 96-well plate  
327 was taken to a microplate shaker (Lumitron) and incubated for 90 min at 500 rpm.

### 328 Senders characterization experiment (Fig. 2e)

329 Relevant plasmids used were XL140mut40+XL208, XL140mut15+XL208, XL140mut8+XL208,  
330 XL140mut7+XL208 and XL302+XL208. Procedures are similar to the circuit cross-talk  
331 experiment. Colonies were grown overnight in 4 ml LB with appropriate antibiotics and  
332 diluted 100 times the next day, aliquoted in a 96-well plate. Inducer OC6 was added to  
333 wells (final max concentration  $100 \mu\text{M}$ ) and serial diluted three times. The 96-well plates  
334 were then incubated in a microplate (Lumitron) shaker for 210 min at 500 rpm.

### 335 Pattern recognition experiments (Fig. 3c and Fig. 5c)

336 For 4-bit patterns, relevant plasmids were XL140mut40+XL208, XL140mut8+XL208,  
337 XL140mut7+XL208 and XL340+XL291. For  $3\times 3$ -bit pattern, sender plasmids were  
338 XL140mut15+XL208, XL140mut7+XL208, XL302+XL208 and XL340+XL291. In both

339 experiments, senders and receivers were grown overnight in 4 ml LB with the appropriate  
340 antibiotics. On the next day, senders were diluted 100 times in OC6 (final max concen-  
341 tration 33.3  $\mu\text{M}$ ) and aTc (final concentration 200  $\text{ng ml}^{-1}$ ) and incubated at 37 °C in  
342 a microplate shaker (Lumitron) at 500 rpm for 210 min. Next, the incubated solutions  
343 were diluted 100 times in LB containing antibiotics and incubated for another 30 min.  
344 The incubated solutions were mixed following the product of weights and patterns. The  
345 weight-pattern products are listed in Table S3 for 3×3-bit patterns. Sender mixtures  
346 were transferred with receivers (50 times dilution from overnight culture) in a 1-to-19  
347 volume ratio. The mixed senders and receivers were incubated in a microplate shaker  
348 (Lumitron) for 90 min at 500 rpm.

## 349 Data availability

The algorithm code and relevant sequences for plasmids are available on Github ([https://github.com/lxm1117/synthetic\\_perceptron\\_2020](https://github.com/lxm1117/synthetic_perceptron_2020)).

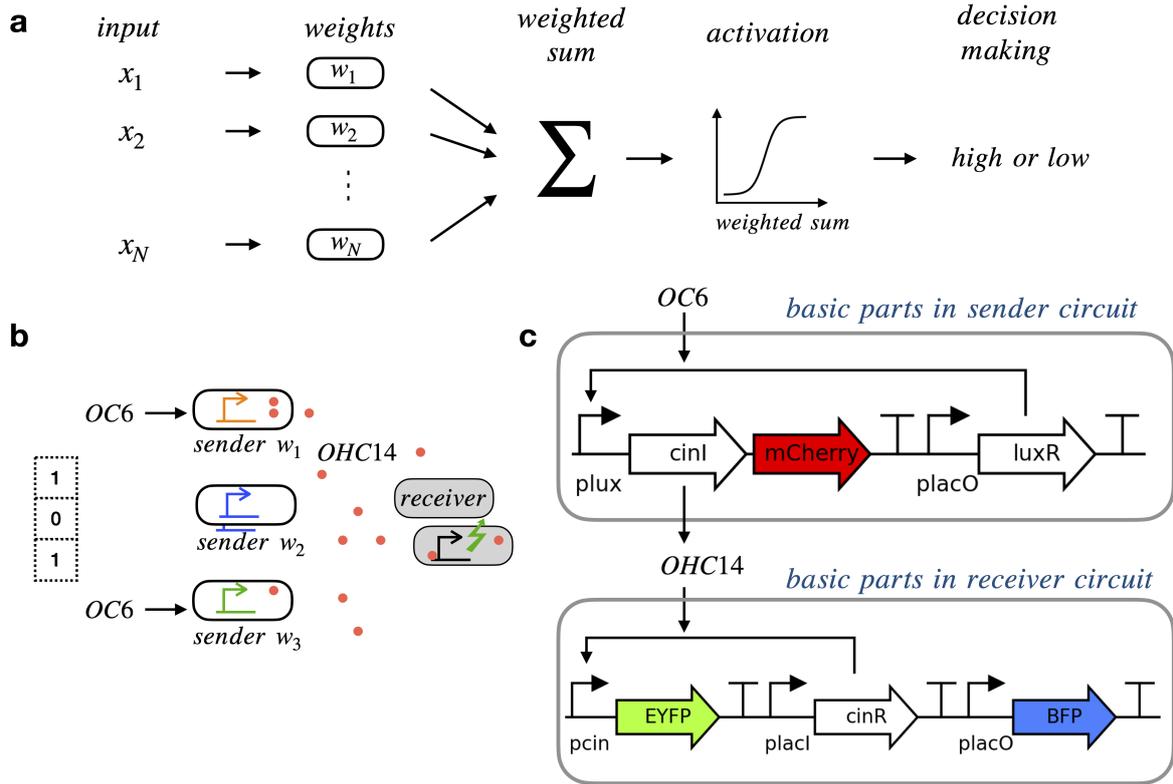
## References

1. Lim, W., Mayer, B. & Pawson, T. *Cell signaling: principles and mechanisms* ISBN: 9780815342441 (Garland Science, Taylor & Francis Group, 2015).
2. Grozinger, L. *et al.* Pathways to cellular supremacy in biocomputing. *Nature Communications* **10**, 5250. ISSN: 2041-1723 (Dec. 2019).
3. McCarty, N. S. & Ledesma-Amaro, R. Synthetic Biology Tools to Engineer Microbial Communities for Biotechnology. *Trends in Biotechnology* **37**, 181–197. ISSN: 01677799 (Feb. 2019).
4. Antebi, Y. E. *et al.* Combinatorial Signal Perception in the BMP Pathway. *Cell* **170**, 1184–1196.e24. ISSN: 00928674 (Sept. 2017).
5. Gardner, T. S., Cantor, C. R. & Collins, J. J. Construction of a genetic toggle switch in *Escherichia coli*. *Nature* **403**, 339–342 (Jan. 2000).
6. Elowitz, M. B. & Leibler, S. A synthetic oscillatory network of transcriptional regulators. *Nature* **403**, 335–338. ISSN: 0028-0836, 1476-4687 (Jan. 2000).
7. Friedland, A. E. *et al.* Synthetic Gene Networks That Count. *Science* **324**, 1199–1202. ISSN: 0036-8075, 1095-9203 (May 2009).
8. Roquet, N., Soleimany, A. P., Ferris, A. C., Aaronson, S. & Lu, T. K. Synthetic recombinase-based state machines in living cells. *Science (New York, N.Y.)* **353**, aad8559–aad8559 (July 2016).

9. Daniel, R., Rubens, J. R., Sarpeshkar, R. & Lu, T. K. Synthetic analog computation in living cells. *Nature* **497**, 619–623. ISSN: 0028-0836, 1476-4687 (May 2013).
10. Purcell, O. & Lu, T. K. Synthetic analog and digital circuits for cellular computation and memory. *Current Opinion in Biotechnology* **29**, 146–155. ISSN: 09581669 (Oct. 2014).
11. Qian, Y., McBride, C. & Del Vecchio, D. Programming Cells to Work for Us. *Annual Review of Control, Robotics, and Autonomous Systems* **1**, 411–440. ISSN: 2573-5144 (May 2018).
12. Aoki, S. K. *et al.* A universal biomolecular integral feedback controller for robust perfect adaptation. *Nature* **570**, 533–537. ISSN: 0028-0836, 1476-4687 (June 2019).
13. Brophy, J. A. N. & Voigt, C. A. Principles of genetic circuit design. *Nature Methods* **11**, 508–520 (May 2014).
14. Cardinale, S. & Arkin, A. P. Contextualizing context for synthetic biology - identifying causes of failure of synthetic biological systems. *Biotechnology Journal* **7**, 856–866. ISSN: 18606768 (July 2012).
15. Brenner, K., You, L. & Arnold, F. H. Engineering microbial consortia: a new frontier in synthetic biology. *Trends in Biotechnology* **26**, 483–489. ISSN: 01677799 (Sept. 2008).
16. Barger, N., Litovco, P., Li, X., Habib, M. & Daniel, R. Synthetic metabolic computation in a bioluminescence-sensing system. *Nucleic Acids Research* **47**, 10464–10474. ISSN: 0305-1048, 1362-4962 (Nov. 2019).
17. Tabor, J. J. *et al.* A Synthetic Genetic Edge Detection Program. *Cell* **137**, 1272–1281. ISSN: 00928674 (June 2009).
18. Basu, S., Gerchman, Y., Collins, C. H., Arnold, F. H. & Weiss, R. A synthetic multicellular system for programmed pattern formation. *Nature* **434**, 1130–1134. ISSN: 0028-0836, 1476-4687 (Apr. 2005).
19. Toda, S., Blauch, L. R., Tang, S. K. Y., Morsut, L. & Lim, W. A. Programming self-organizing multicellular structures with synthetic cell-cell signaling. *Science*. ISSN: 0036-8075, 1095-9203.
20. Tamsir, A., Tabor, J. J. & Voigt, C. A. Robust multicellular computing using genetically encoded NOR gates and chemical ‘wires’. *Nature* **469**, 212–215 (Jan. 2011).
21. Guiziou, S., Mayonove, P. & Bonnet, J. Hierarchical composition of reliable recombinase logic devices. *Nature Communications*, 1–7 (Jan. 2019).
22. Savage, N. Computer logic meets cell biology: how cell science is getting an upgrade. *Nature* **564**, S1–S3. ISSN: 0028-0836, 1476-4687 (Dec. 2018).

23. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444. ISSN: 0028-0836, 1476-4687 (May 2015).
24. Lee, T.-H. & Maheshri, N. A regulatory role for repeated decoy transcription factor binding sites in target gene expression. *Molecular Systems Biology* **8**, 1–11 (Mar. 2012).
25. Hassoun, M. H. *Fundamentals of artificial neural networks* ISBN: 9780262082396 (MIT Press, 1995).
26. Bishop, C. *Pattern recognition and machine learning* ISBN: 9780387310732 (Springer, 2006).
27. Prezioso, M. *et al.* Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* **521**, 61–64 (May 2015).
28. Xie, Z., Wroblewska, L., Prochazka, L., Weiss, R. & 2011. Multi-Input RNAi-Based Logic Circuit for Identification of Specific Cancer Cells. *science.sciencemag.org*.
29. Didovyk, A. *et al.* Distributed Classifier Based on Genetically Engineered Bacterial Cell Cultures. *ACS Synthetic Biology* **4**, 72–82. ISSN: 2161-5063, 2161-5063 (Jan. 2015).
30. Alnahhas, R. N. *et al.* Majority sensing in synthetic microbial consortia. *Nature Communications* **11**, 3659. ISSN: 2041-1723 (Dec. 2020).
31. Miano, A., Liao, M. J. & Hasty, J. Inducible cell-to-cell signaling for tunable dynamics in microbial communities. *Nature Communications* **11**, 1193. ISSN: 2041-1723 (Dec. 2020).
32. Lin, X. *et al.* All-optical machine learning using diffractive deep neural networks. *Science* **361**, 1004–1008. ISSN: 0036-8075, 1095-9203 (Sept. 2018).
33. Cherry, K. M. & Qian, L. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature*, 1–19 (July 2018).
34. Pandi, A. *et al.* Metabolic perceptrons for neural computing in biological systems. *Nature Communications*, 1–13 (Aug. 2019).
35. Khakhar, A., Bolten, N. J., Nemhauser, J. & Klavins, E. Cell–Cell Communication in Yeast Using Auxin Biosynthesis and Auxin Responsive CRISPR Transcription Factors. *ACS Synthetic Biology* **5**, 279–286. ISSN: 2161-5063, 2161-5063 (Apr. 2016).
36. Meyer, A. J., Segall-Shapiro, T. H., Glassey, E., Zhang, J. & Voigt, C. A. Escherichia coli “Marionette” strains with 12 highly optimized small-molecule sensors. *Nature Chemical Biology* **15**, 196–204. ISSN: 1552-4450, 1552-4469 (Feb. 2019).
37. Sambrook, J. & Russell, D. W. *Molecular cloning: a laboratory manual* 3rd ed. ISBN: 9780879695774 (Cold Spring Harbor Laboratory Press, 2001).

38. Castillo-Hair, S. M. *et al.* FlowCal: A User-Friendly, Open Source Software Tool for Automatically Converting Flow Cytometry Data from Arbitrary to Calibrated Units. *ACS Synthetic Biology* **5**, 774–780. ISSN: 2161-5063, 2161-5063 (July 2016).
39. Der, B. S. *et al.* DNAplotlib: Programmable Visualization of Genetic Designs and Associated Data. *ACS Synthetic Biology* **6**, 1115–1119. ISSN: 2161-5063, 2161-5063 (July 2017).



**Figure 1: Perceptron diagram and genetic network implementation in experiments.** **a** Schematic representation of a perceptron with an  $N$ -elements input. The elements of the input vector  $\vec{x}_j$  are multiplied with the elements of the weight vector  $\vec{w}$ , giving rise to a weighted sum. The weighted sum acts as input to an activation function, the output of which is taken for decision-making. **b** Illustration of a perceptron network based on quorum sensing between sender and receiver bacteria. An example pattern ‘101’ is represented by high acyl-homoserine lactone 3OC6-HSL (OC6), no OC6 and high OC6. Senders containing promoters with different strengths are shown in different colors, corresponding to different weights. Red dots are signaling molecules (i.e., acyl-homoserine lactone 3OHC14:1-HSL, OHC14) produced by senders. The molecules diffuse across the cell membrane and activate receivers to generate fluorescence signals. **c** Basic genetic circuit parts in senders and receivers. In senders, inducer OC6 binds with constitutively expressed transcription factor LuxR and activates  $P_{lux}$  to express mCherry signal and Acyl-homoserine-lactone synthase. The latter protein is encoded by *cinI* and catalyzes the synthesis of OHC14, which is the diffusible signaling molecule. In receivers, the OHC14 entering from media binds with transcription factor CinR and together activates  $P_{cin}$  to express the fluorescence signal EYFP. Constitutively expressed BFP signal serves as a control.



Figure 2: **Circuit engineering in senders and receivers.** **a** Double-inversion parts added to sender circuits to minimize the cross-talk between senders at different states. The double-inversion circuit can be controlled by anhydrotetracycline (aTc) to repress the activity of  $P_{lacO}$  on a low-copy-number plasmid (LCP). **b** The decoy binding sites  $P_{lacO}$ -array on a medium-copy-number plasmid (MCP) are necessary to maintain a large output dynamic range in the double inversion circuit. With the  $P_{lacO}$ -array, the expression of GFP is turned off in the presence of aTc (200 ng ml<sup>-1</sup>) (left green bars, +aTc vs -aTc). Without the  $P_{lacO}$ -array, GFP expression stays low regardless of the aTc concentration (right orange bars, +aTc vs -aTc). Error bars represent the standard error from multiple replicates. **c** Schematic drawing to show how the slope of the activation function affects the decision making of perceptron. Both the black curve and the brown curve are activation functions ( $\sigma_1$  and  $\sigma_2$ ) mapping the weighted sum to different activation levels. The black curve is steeper than the brown curve. For a pair of inputs from two classes, the steeper the activation function, the larger difference between the outputs of the two inputs. **d** (top) Genetic circuit diagram for receivers with positive feedback on LCP. (lower left) Transfer functions of receivers, with positive feedback (PF) and in open loop (OL). For each curve, median EYFP values are averaged from three replicates. Error bars represent standard error. Dashed lines are fittings using Hill equations from experiment measures, with fitting parameters shown in Table 1. **e** (top left) Schematic representation of the  $P_{lux}$  activator promoter and  $P_{lux}$  repressor promoter (top right). Transfer functions of senders with various mutated  $P_{lux}$  (TCTC, TATC, TTAA, TCCC) activator promoters and  $P_{lux}$  repressor promoter. For each curve, median mCherry levels are averaged from multiple replicates (more than eight replicates). Error bars represent standard error. Dashed lines are fittings using Hill equations from experiment measures, with fitting parameters shown in Table 1.

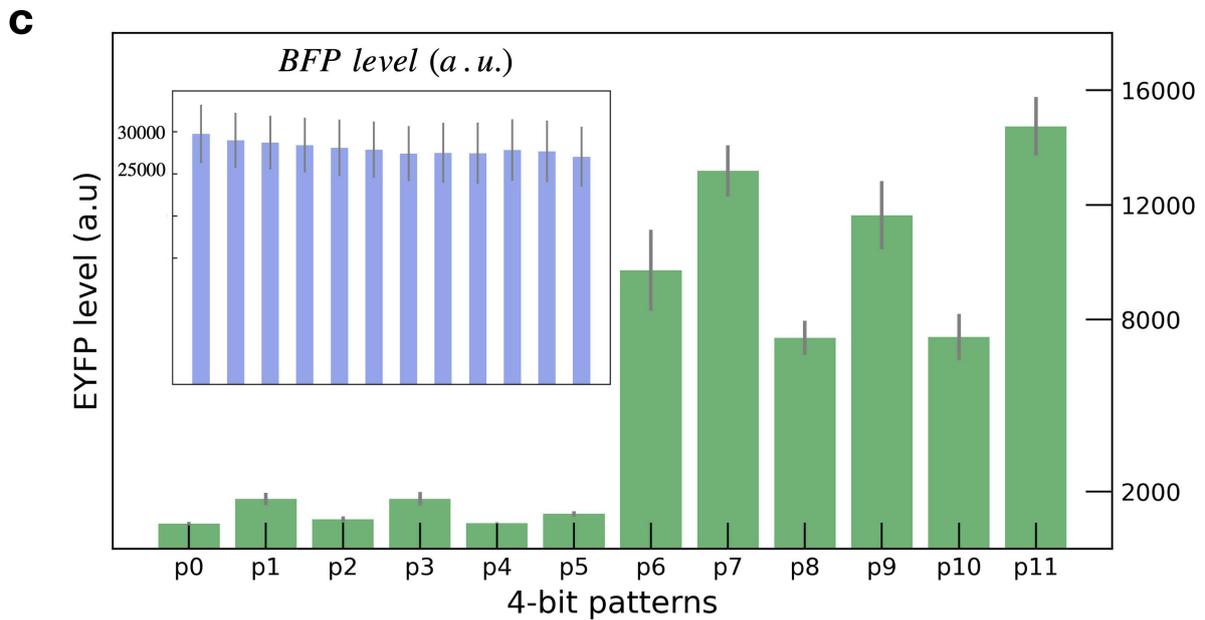
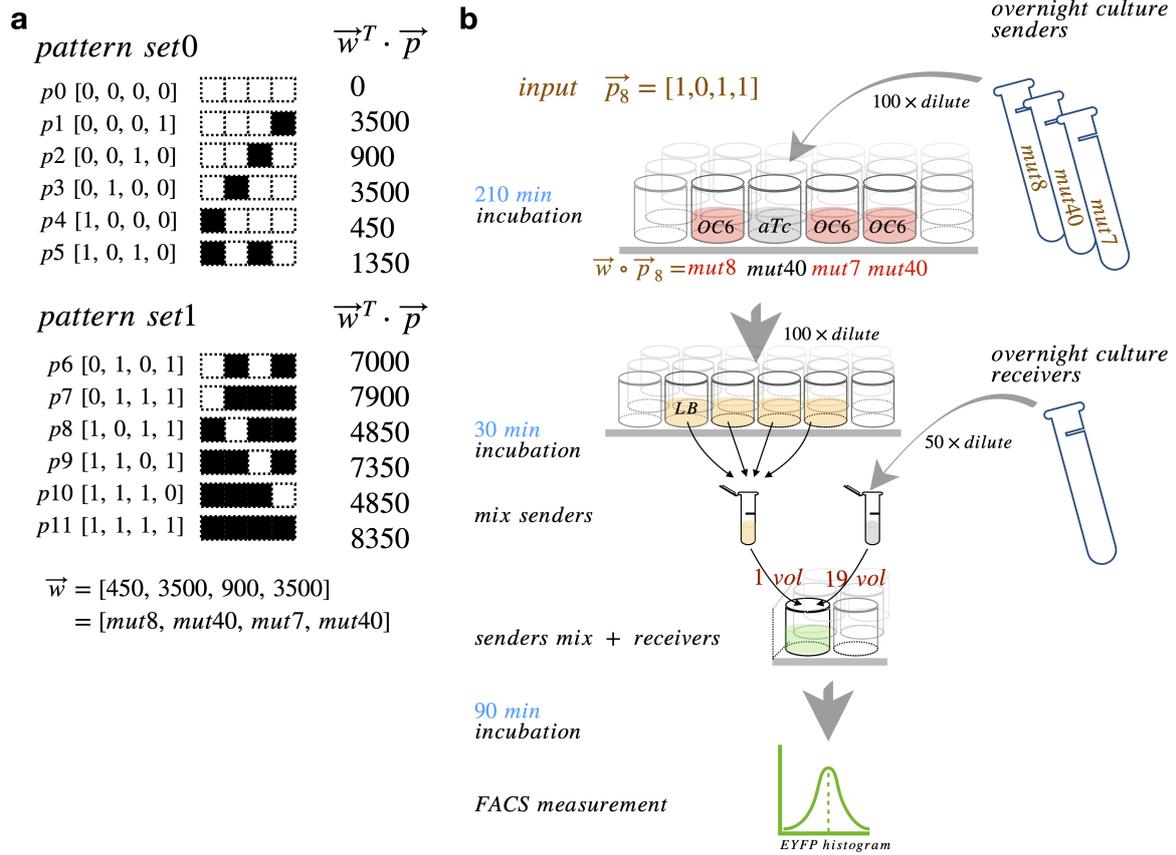


Figure 3: **Classification of 4-bit patterns.** **a** Two sets of 4-bit patterns with binary value entries. Weight vector [450, 3500, 900, 3500] can separate the patterns. **b** Workflow of experiment procedures using pattern [1,0,1,1] as an example. In brief, senders grown overnight were 100 times diluted and incubated at 37 °C in separate wells for 210 min. Depending on the input bits, senders were incubated in LB containing either OC6 ('1') to control the activity of  $P_{lux}$ , or aTc ('0') to prevent the cross-talk between senders after mixing. Next, all senders were diluted 100 times in LB, incubated for 30 min, mixed together and transferred into receiver culture in a 1-to-19 volume ratio. The sender-receiver mixture was incubated at 37 °C for 90 min and the fluorescence signals were then measured using flow cytometry. **c** Flow cytometry measurement of receiver output for 4-bit patterns. EYFP and BFP (upper left inset) are averaged median values from three replicates, displayed with standard error bars.

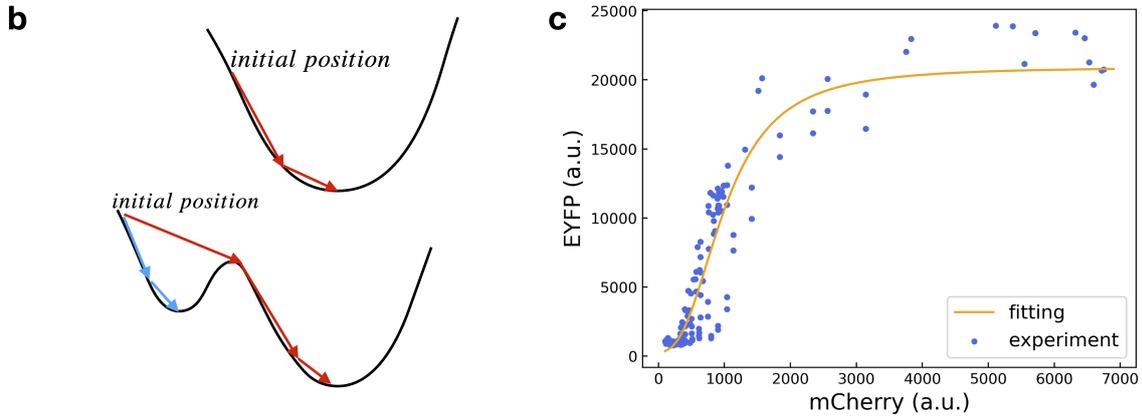
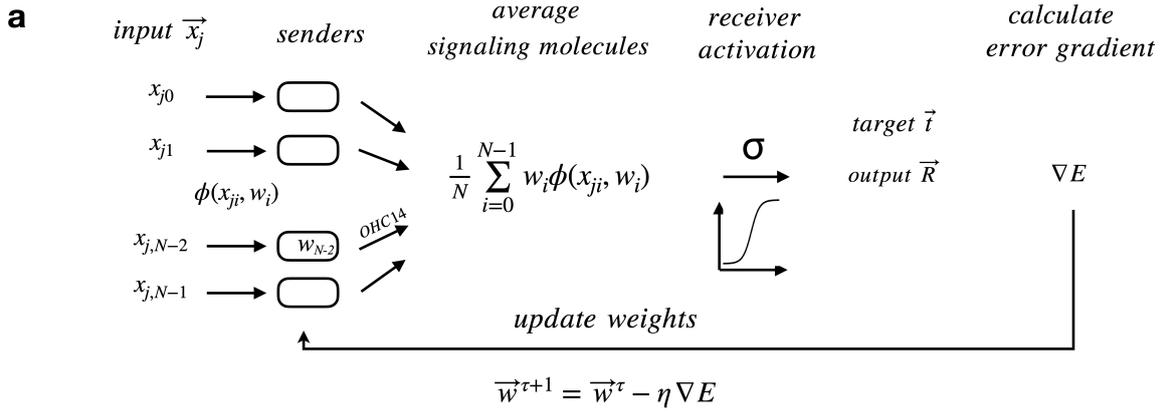


Figure 4: **Algorithm for learning weights.** **a** Schematic representation of steps involved in the algorithm. The input  $\vec{x}_j$  is one pattern from the complete pattern set. The weight vector  $\vec{w}$  represents the promoter strengths of the  $P_{lux}$  promoter in sender circuits. The product of the input and weight vectors is the amount of signaling molecules OHC14 collectively produced by senders. After sender mixture was transferred into receiver culture, the signaling molecules activate the receiver circuits, effectively as an input to the function  $\sigma$ . The output of  $\sigma$ ,  $\vec{R}$ , is compared with a target vector  $\vec{t}$  to calculate an error gradient, which is used to update weight values for the next iteration.  $\eta$  is the learning rate parameter, which is the same as the step size described in the text. **b** Cartoons of error function curves to illustrate the iterative procedure to search for the minimum point. In the lower graph, red arrows represent the large iterative step size, which leads to the global minimal point. Blue arrows represent the small iterative step size, which leads to the local minima. **c** Receiver EYFP levels as a function of mCherry levels in senders. Senders of different  $P_{lux}$  mutations are incubated with varying amounts of OC6 and mixed with receivers following the steps in Fig. 3b.

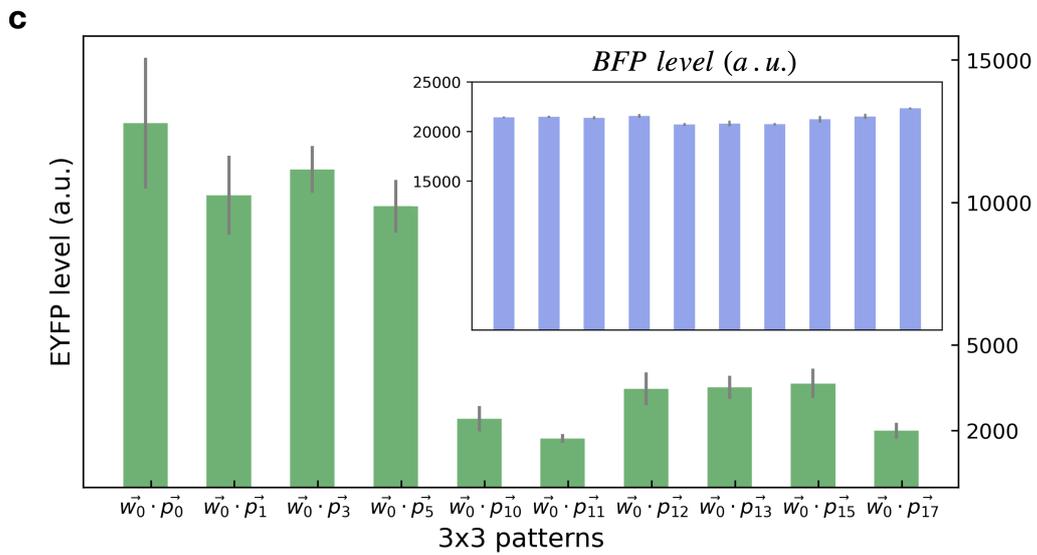
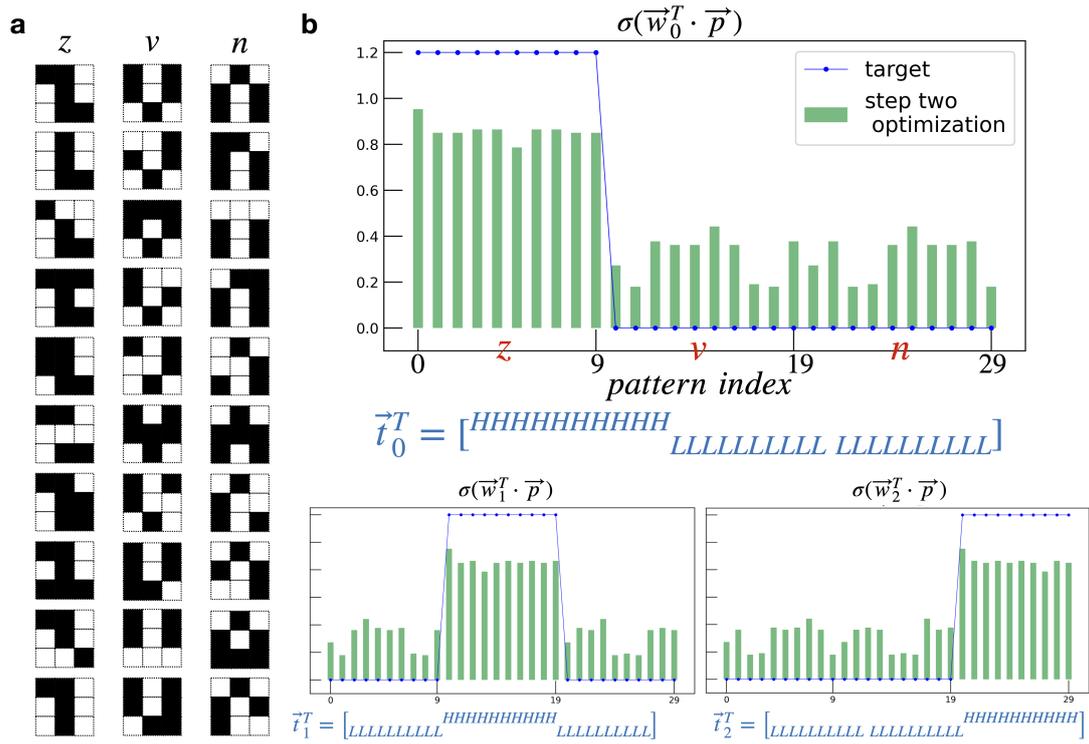


Figure 5: **Classification of  $3 \times 3$ -bit patterns.** **a**  $3 \times 3$ -bit patterns for 'z', 'v' and 'n'. Each pattern category consists of one ideal pattern and nine noisy patterns. Each noisy pattern has a one bit flip from the ideal pattern. **b** Simulated receiver output from the algorithm after the two-step optimization. Three graphs are shown separately for three weigh vectors. In each graph, there are 30 bars, corresponding to the 30 patterns (10 patterns  $\times$  3 categories), in the order of 'z', 'v' and 'n' patterns. Blue dotted lines mark the target vectors, which are also shown figuratively at the bottom of each graph. One target vector is assigned to one category of patterns, in which H represents a high value (set to 1.2 in the study) and L represents a low value (set to 0). In particular, the perceptron output of patterns in 'z' ( $\sigma(\vec{w}^T \cdot \vec{p}_0)$ , top graph) have output values approximate the target vector  $\vec{t}_0$ . Similarly, the outputs of patterns in 'v' and 'n' are close to  $\vec{t}_1$  (lower left graph) and  $\vec{t}_2$  (lower right graph), respectively. **c** Flow cytometry measurement of receiver output for  $3 \times 3$  patterns. In this case, nine groups of senders were selected, corresponding to the nine elements in weight vectors. Senders of different groups were incubated separately and mixed with receivers. The fluorecence signals from receivers were measured. EYFP and BFP (inset) are averaged median values from three replicates, displayed with standard error bars. The graph is in a similar format as the simulation results shown in **b**, but not all products of patterns and weights are presented. As discussed in the text, many pattern-weight products are repetitive (Fig. S6). We examiend only the unique and non-repeating products in experiments.

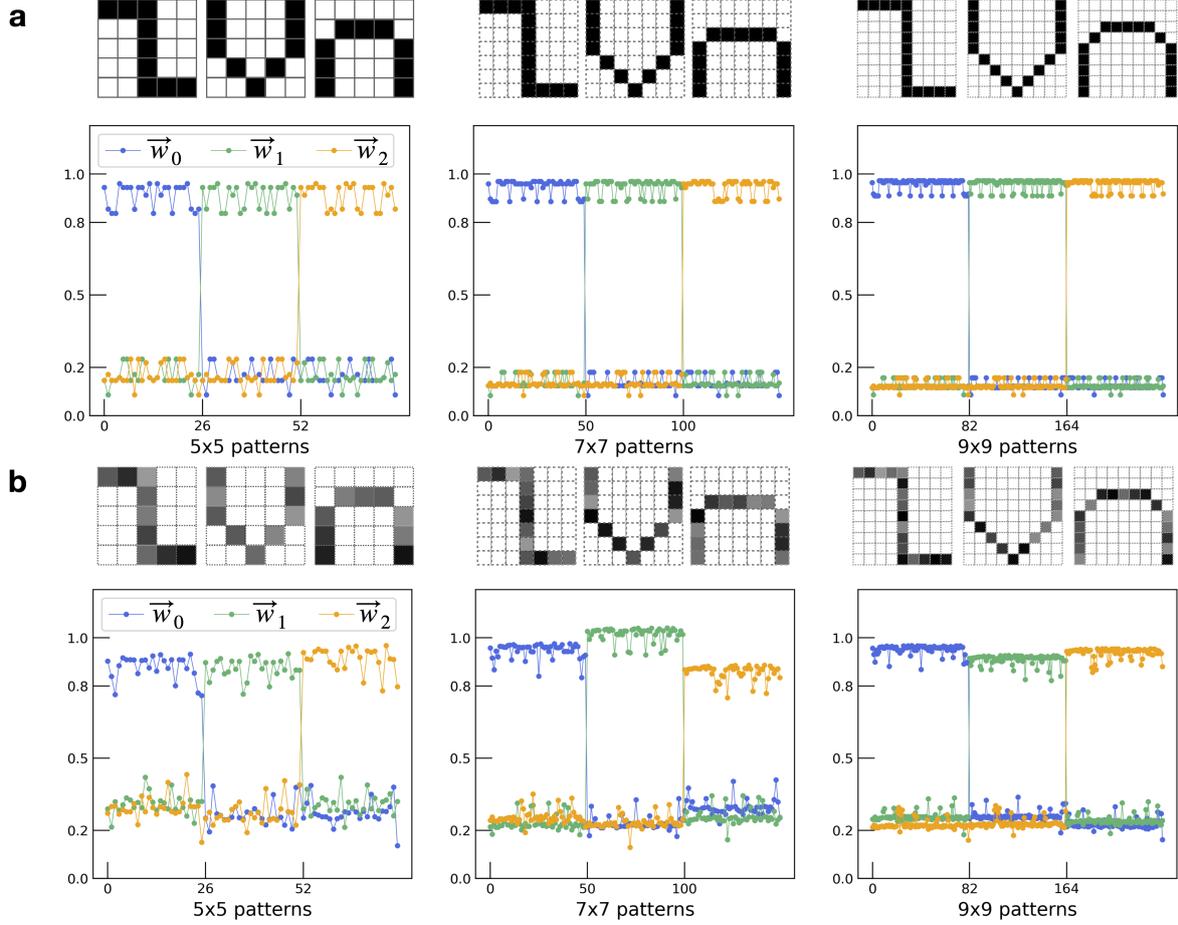
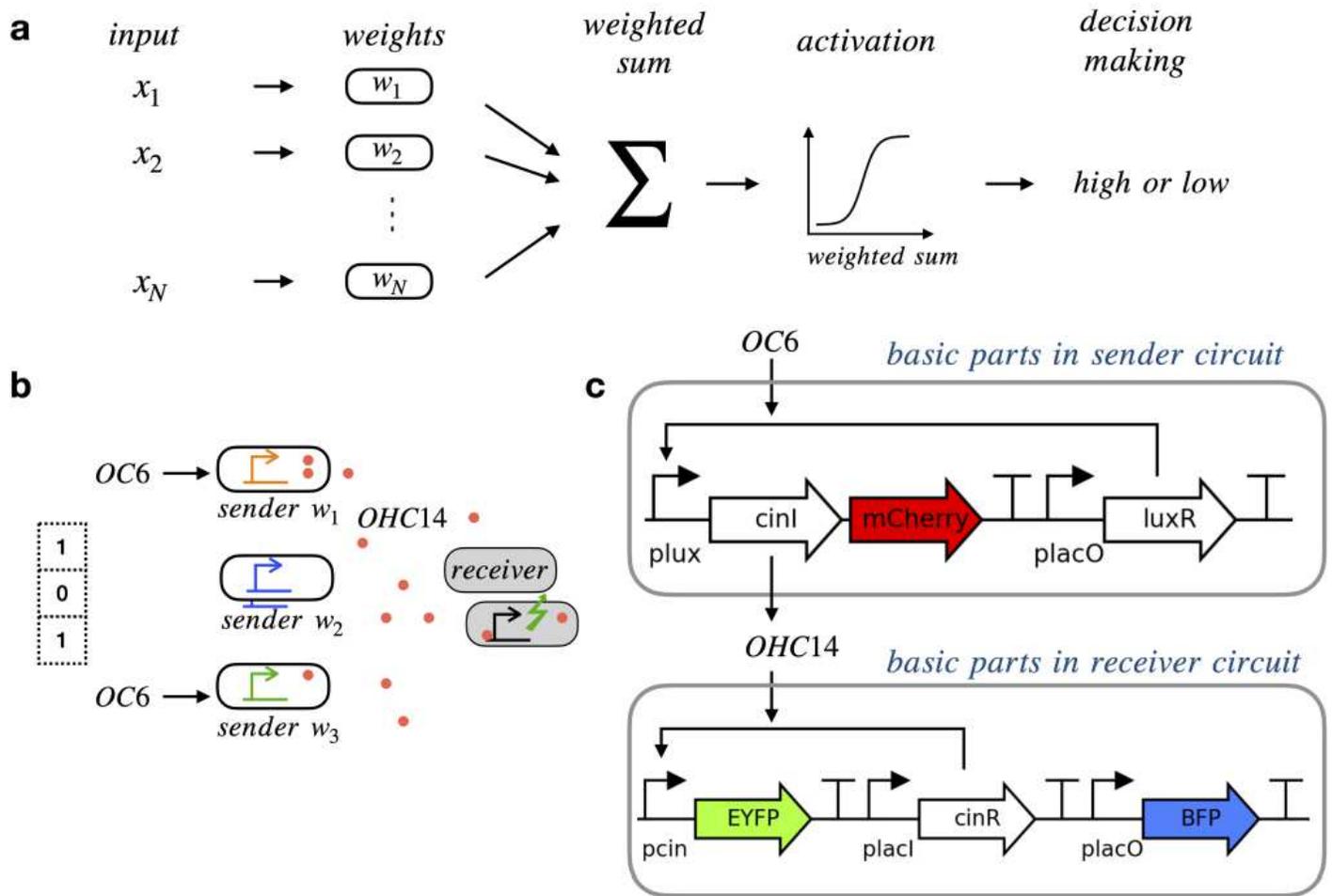


Figure 6: **Simulated results for classifying more sophisticated patterns.** **a** Simulated receiver output after step-one optimizations for  $5 \times 5$ ,  $7 \times 7$  and  $9 \times 9$  patterns. In the lower panel, receiver output for patterns in three categories are overlaid together. The output values are marked by lines in different colors and are presented the same way as in Fig. 5b. For the  $5 \times 5$  set, there are in total  $(1+25) \times 3 = 78$  patterns; for the  $7 \times 7$  set, the total number is  $(1+49) \times 3 = 150$ ; and for the  $9 \times 9$  set,  $(1+81) \times 3 = 246$ . Only the noiseless patterns are shown. **b** The same as in a but for patterns with non-binary inputs values. To generate these patterns, each element in a pattern is multiplied by a random number between 0.5 and 1 to emulate the randomized OC6 level. In particular, the same random number vector (of size  $1 \times 25$ ) is used for all  $5 \times 5$  patterns. Similarly, one random vector (of size  $1 \times 49$ ) for all  $7 \times 7$  patterns and one random vector (of size  $1 \times 81$ ) for all  $9 \times 9$  patterns.

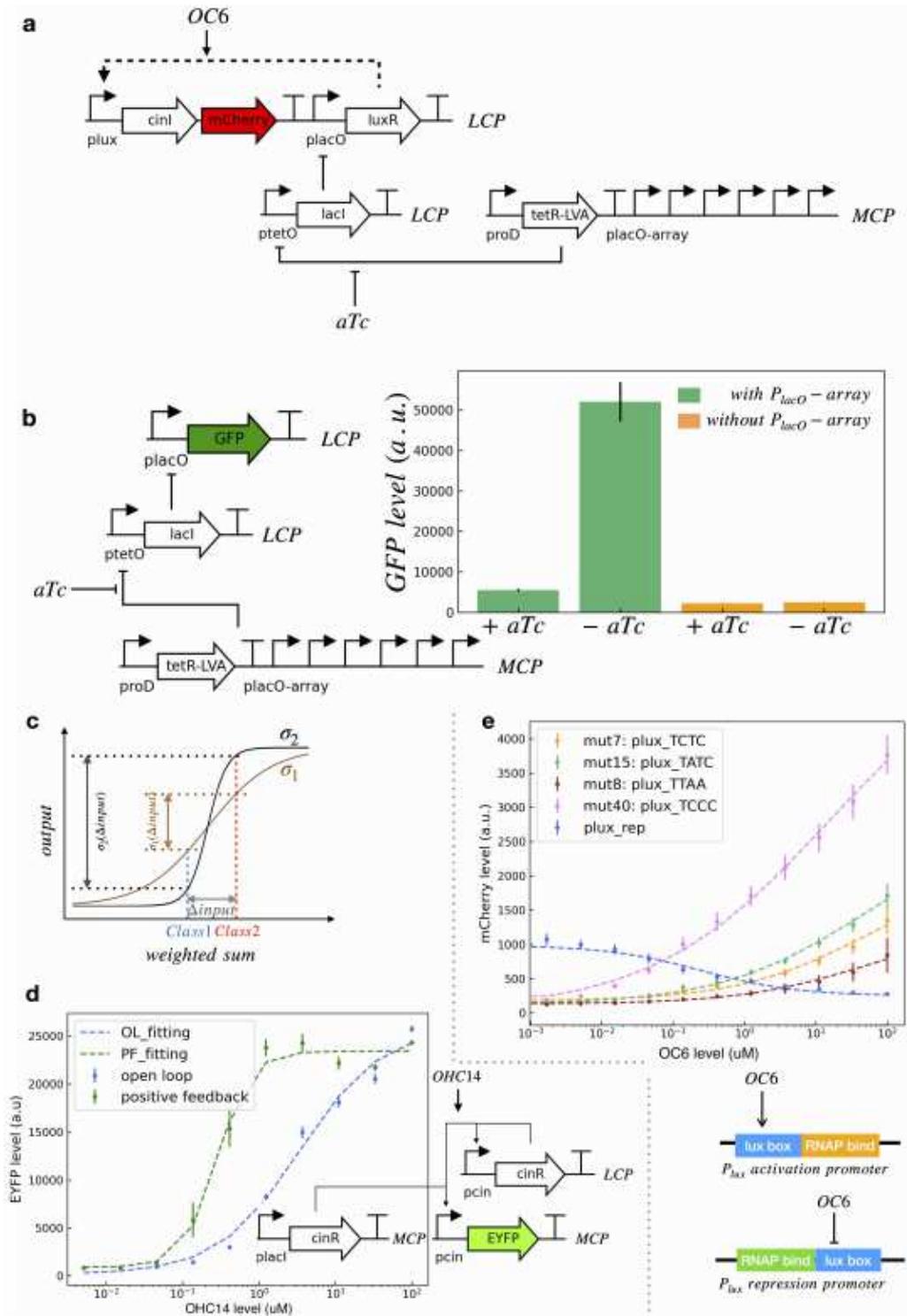
# Figures



**Figure 1**

Perceptron diagram and genetic network implementation in experiments. **a** Schematic representation of a perceptron with an  $N$ -elements input. The elements of the input vector  $\sim x_j$  are multiplied with the elements of the weight vector  $\sim w$ , giving rise to a weighted sum. The weighted sum acts as input to an activation function, the output of which is taken for decision-making. **b** Illustration of a perceptron network based on quorum sensing between sender and receiver bacteria. An example pattern '101' is represented by high acyl-homoserine lactone 3OC6-HSL (OC6), no OC6 and high OHC14. Senders containing promoters with different strengths are shown in different colors, corresponding to different weights. Red dots are signaling molecules (i.e., acyl-homoserine lactone 3OHC14:1-HSL, OHC14) produced by senders. The molecules diffuse across the cell membrane and activate receivers to generate fluorescence signals. **c** Basic genetic circuit parts in senders and receivers. In senders, inducer OC6 binds with constitutively expressed transcription factor LuxR and activates Plux to express mCherry signal and Acyl-homoserine-lactone synthase. The latter protein is encoded by cinI and catalyzes the synthesis of OHC14, which is the diffusible signaling molecule. In receivers, the OHC14 entering from media binds with transcription factor

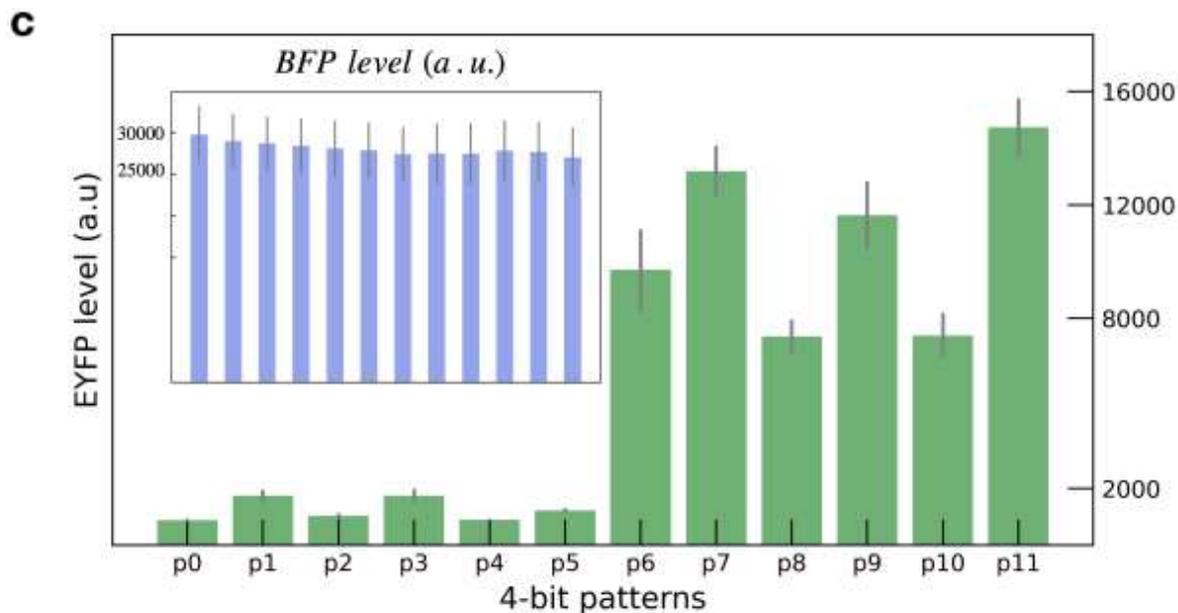
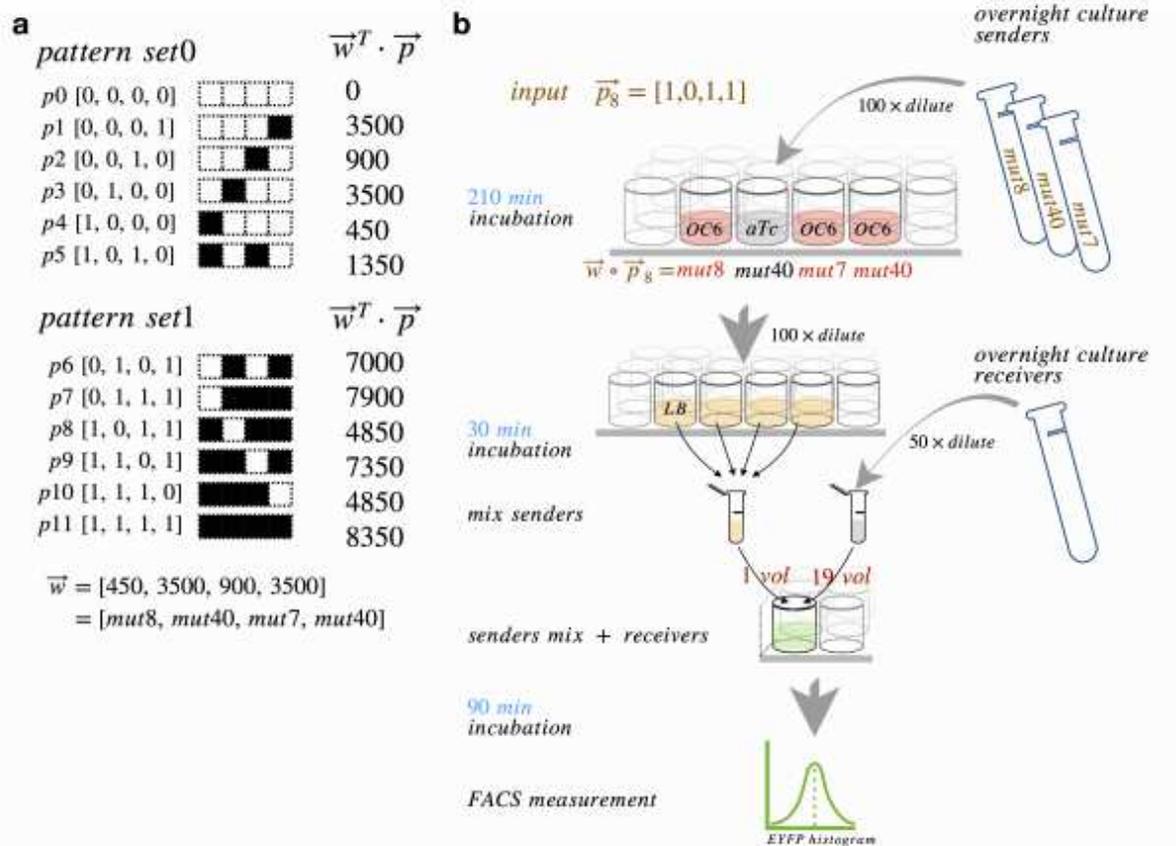
CinR and together activates Pcin to express the fluorescence signal EYFP. Constitutively expressed BFP signal serves as a control.



**Figure 2**

Circuit engineering in senders and receivers. a Double-inversion parts added to sender circuits to minimize the cross-talk between senders at different states. The double-inversion circuit can be controlled by anhydrotetracycline (aTc) to repress the activity of PlacO on a low-copy-number plasmid (LCP). b The

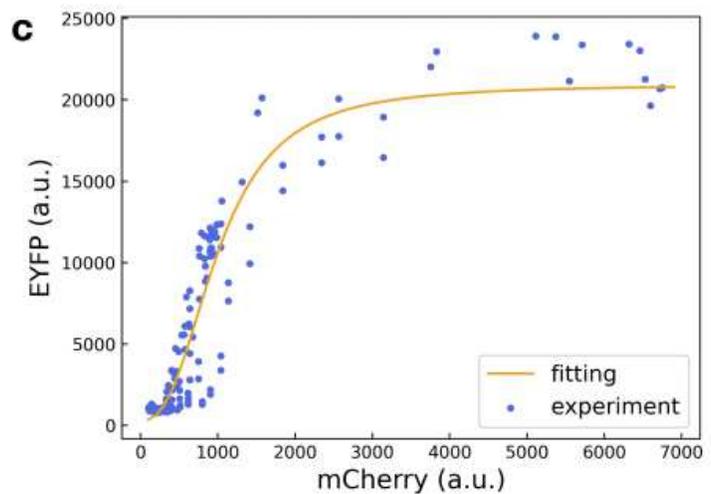
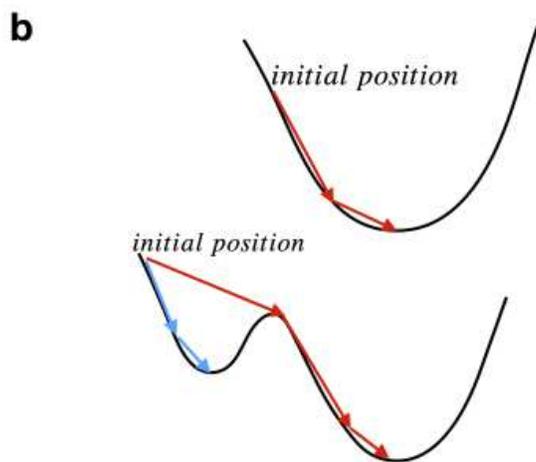
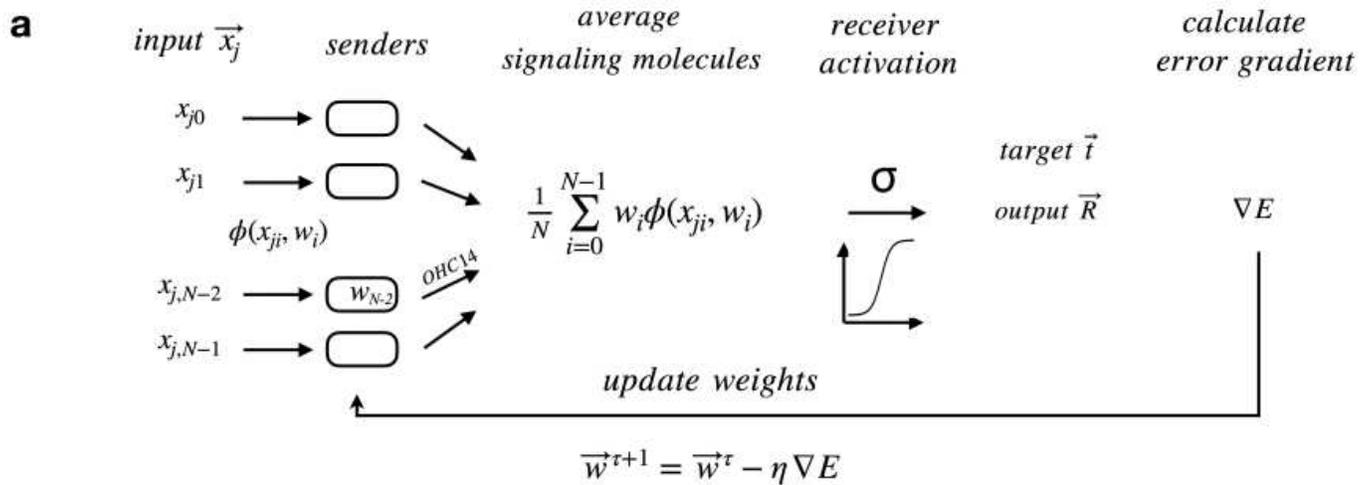
decoy binding sites PlacO-array on a medium-copy-number plasmid (MCP) are necessary to maintain a large output dynamic range in the double inversion circuit. With the PlacO-array, the expression of GFP is turned off in the presence of aTc (200 ng ml<sup>-1</sup>) (left green bars, +aTc vs -aTc). Without the PlacO-array, GFP expression stays low regardless of the aTc concentration (right orange bars, +aTc vs -aTc). Error bars represent the standard error from multiple replicates. c Schematic drawing to show how the slope of the activation function affects the decision making of perceptron. Both the black curve and the brown curve are activation functions ( $\sigma_1$  and  $\sigma_2$ ) mapping the weighted sum to different activation levels. The black curve is steeper than the brown curve. For a pair of inputs from two classes, the steeper the activation function, the larger difference between the outputs of the two inputs. d (top) Genetic circuit diagram for receivers with positive feedback on LCP. (lower left) Transfer functions of receivers, with positive feedback (PF) and in open loop (OL). For each curve, median EYFP values are averaged from three replicates. Error bars represent standard error. Dashed lines are fittings using Hill equations from experiment measures, with fitting parameters shown in Table 1. e (top left) Schematic representation of the Plux activator promoter and Plux repressor promoter (top right). Transfer functions of senders with various mutated Plux (TCTC, TATC, TTAA, TCCC) activator promoters and Plux repressor promoter. For each curve, median mCherry levels are averaged from multiple replicates (more than eight replicates). Error bars represent standard error. Dashed lines are fittings using Hill equations from experiment measures, with fitting parameters shown in Table 1.



**Figure 3**

Classification of 4-bit patterns. a Two sets of 4-bit patterns with binary value entries. Weight vector [450, 3500, 900, 3500] can separate the patterns. b Workflow of experiment procedures using pattern [1,0,1,1] as an example. In brief, senders grown overnight were 100 times diluted and incubated at 37 °C in separate wells for 210 min. Depending on the input bits, senders were incubated in LB containing either OC6 ('1') to control the activity of Plux, or aTc ('0') to prevent the cross-talk between senders after mixing. c

Next, all senders were diluted 100 times in LB, incubated for 30 min, mixed together and transferred into receiver culture in a 1-to-19 volume ratio. The senderreceiver mixture was incubated at 37 °C for 90 min and the fluorescence signals were then measured using flow cytometry. c Flow cytometry measurement of receiver output for 4-bit patterns. EYFP and BFP (upper left inset) are averaged median values from three replicates, displayed with standard error bars.



**Figure 4**

Algorithm for learning weights. a Schematic representation of steps involved in the algorithm. The input  $\vec{x}_j$  is one pattern from the complete pattern set. The weight vector  $\vec{w}$  represents the promoter strengths of the Plux promoter in sender circuits. The product of the input and weight vectors is the amount of signaling molecules OHC14 collectively produced by senders. After sender mixture was transferred into receiver culture, the signaling molecules activate the receiver circuits, effectively as an input to the function  $\sigma$ . The output of  $\sigma$ ,  $\vec{R}$ , is compared with a target vector  $\vec{t}$  to calculate an error gradient, which is used to update weight values for the next iteration.  $\eta$  is the learning rate parameter, which is the same as the step size described in the text. b Cartoons of error function curves to illustrate the iterative

procedure to search for the minimum point. In the lower graph, red arrows represent the large iterative step size, which leads to the global minimal point. Blue arrows represent the small iterative step size, which leads to the local minima. c Receiver EYFP levels as a function of mCherry levels in senders. Senders of different Plux mutations are incubated with varying amounts of OC6 and mixed with receivers following the steps in Fig. 3b.

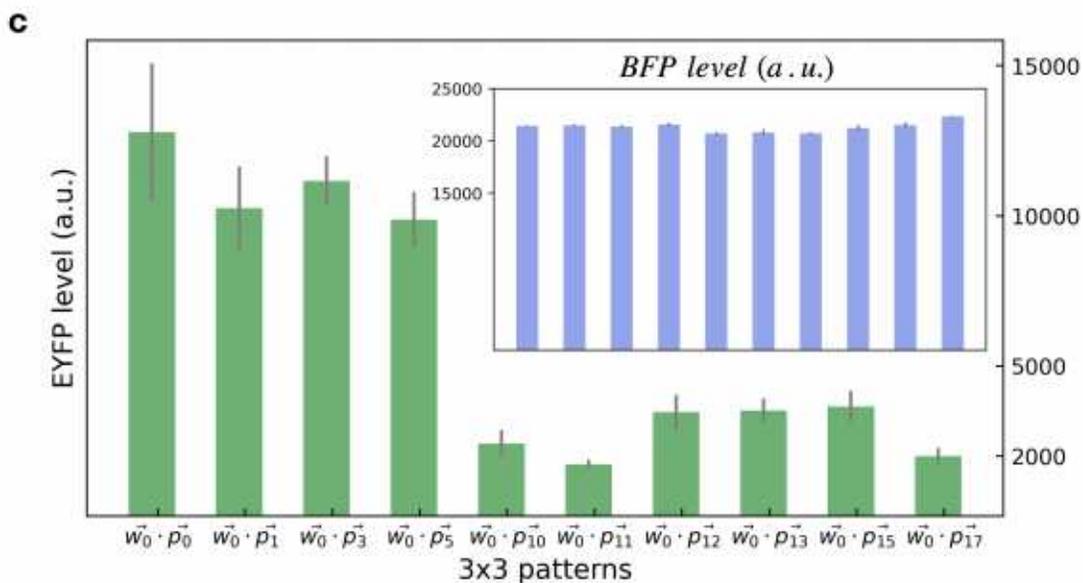
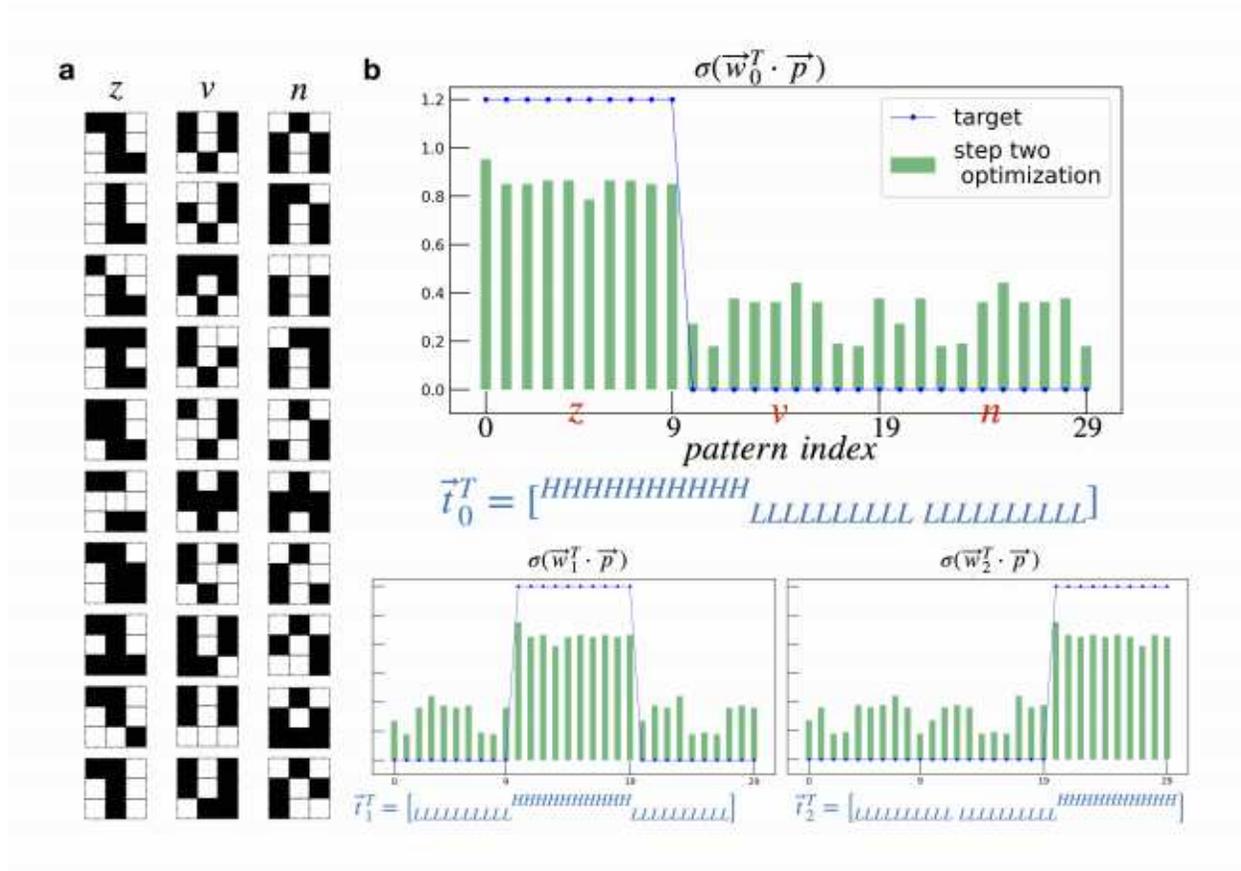
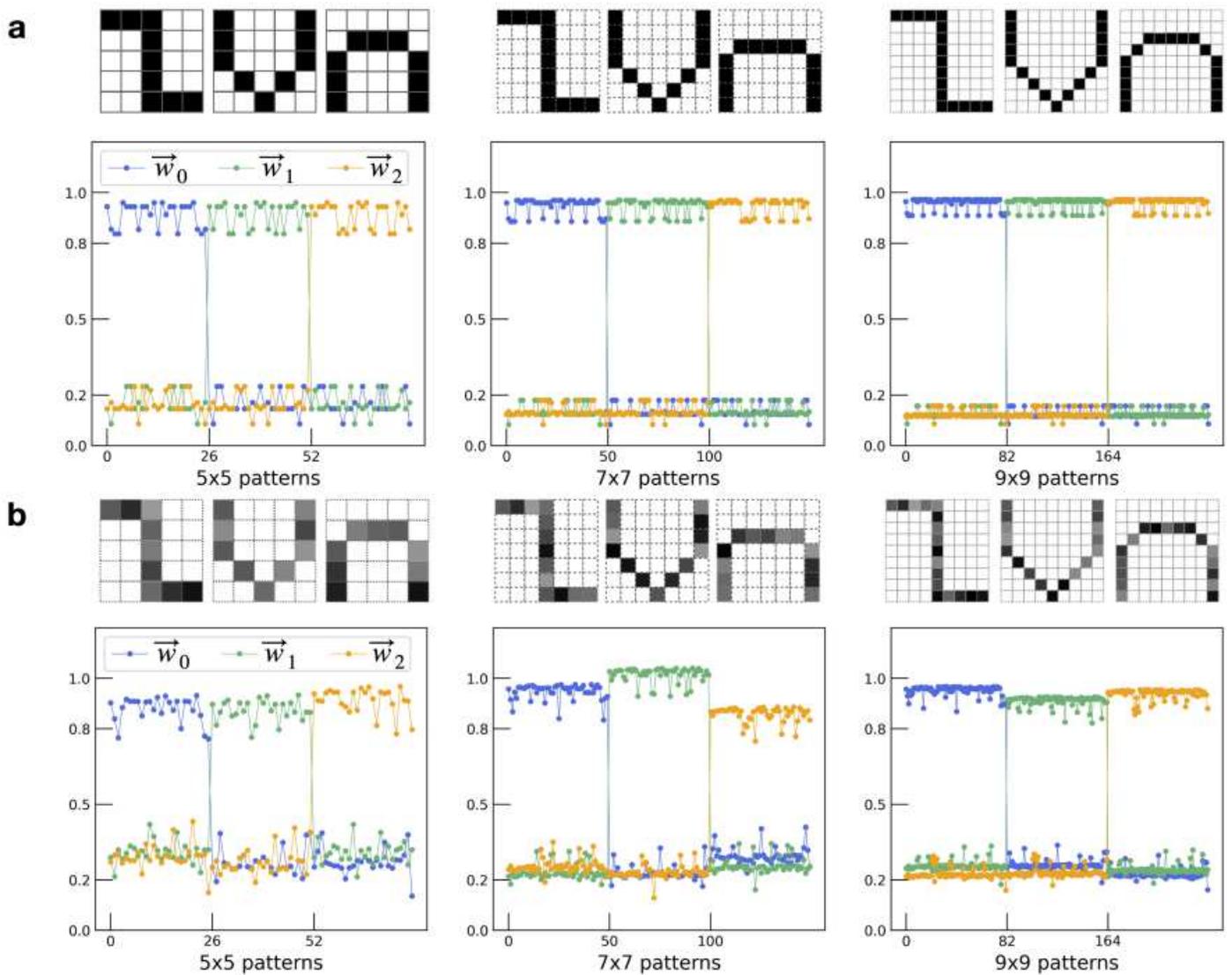


Figure 5

Classification of 3×3-bit patterns. a 3×3-bit patterns for 'z', 'v' and 'n'. Each pattern category consists of one ideal pattern and nine noisy patterns. Each noisy pattern has a one bit flip from the ideal pattern. b Simulated receiver output from the algorithm after the two-step optimization. Three graphs are shown separately for three weigh vectors. In each graph, there are 30 bars, corresponding to the 30 patterns (10 patterns×3 categories), in the order of 'z', 'v' and 'n' patterns. Blue dotted lines mark the target vectors, which are also shown figuratively at the bottom of each graph. One target vector is assigned to one category of patterns, in which H represents a high value (set to 1.2 in the study) and L represents a low value (set to 0). In particular, the perceptron output of patterns in 'z' ( $\sigma(\sim w T \cdot \sim p_0)$ , top graph) have output values approximate the target vector  $\sim t_0$ . Similarly, the outputs of patterns in 'v' and 'n' are close to  $\sim t_1$  (lower left graph) and  $\sim t_2$  (lower right graph), respectively. c Flow cytometry measurement of receiver output for 3×3 patterns. In this case, nine groups of senders were selected, corresponding to the nine elements in weight vectors. Senders of different groups were incubated separately and mixed with receivers. The fluorecence signals from receivers were measured. EYFP and BFP (inset) are averaged median values from three replicates, displayed with standard error bars. The graph is in a similar format as the simuation results shown in b, but not all products of patterns and weights are presented. As discussed in the text, many pattern-weight products are repetitive (Fig. S6). We examiend only the unique and non-repeating products in experiments.



**Figure 6**

Simulated results for classifying more sophisticated patterns. a Simulated receiver output after step-one optimizations for 5x5, 7x7 and 9x9 patterns. In the lower panel, receiver output for patterns in three categories are overlaid together. The output values are marked by lines in different colors and are presented the same way as in Fig. 5b. For the 5x5 set, there are in total  $(1+25)\times 3=78$  patterns; for the 7x7 set, the total number is  $(1+49)\times 3=150$ ; and for the 9x9 set,  $(1+81)\times 3=246$ . Only the noiseless patterns are shown. b The same as in a but for patterns with non-binary inputs values. To generate these patterns, each element in a pattern is multiplied by a random number between 0.5 and 1 to emulate the randomized OC6 level. In particular, the same random number vector (of size  $1\times 25$ ) is used for all 5x5 patterns. Similarly, one random vector (of size  $1\times 49$ ) for all 7x7 patterns and one random vector (of size  $1\times 81$ ) for all 9x9 patterns.

## Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryFile.pdf](#)