12578–12595 *Nucleic Acids Research, 2022, Vol. 50, No. 21* *Published online 1 December 2022*
https://doi.org/10.1093/nar/gkac1120

Implementing re-configurable biological computation with distributed multicellular consortia

David Canadell [1,2], Nicolás Ortiz-Vaquerizas[1,2], Sira Mogas-Diez [2,3], Eulàlia de Nadal [1,2,*], Javier Macia[2,3,*] and Francesc Posas [1,2,*]

[1]Institute for Research in Biomedicine (IRB Barcelona), The Barcelona Institute of Science and Technology, Barcelona 08028, Spain, [2]Department of Medicine and Life Sciences (MELIS), Universitat Pompeu Fabra (UPF), 08003 Barcelona, Spain and [3]Synthetic Biology for Biomedical Applications Group, Department of Medicine and Life Sciences (MELIS), Universitat Pompeu Fabra (UPF), 08003 Barcelona, Spain

Received August 10, 2022; Revised October 30, 2022; Editorial Decision November 02, 2022; Accepted November 17, 2022

ABSTRACT

The use of synthetic biological circuits to deal with numerous biological challenges has been proposed in several studies, but its implementation is still remote. A major problem encountered is the complexity of the cellular engineering needed to achieve complex biological circuits and the lack of general-purpose biological systems. The generation of reprogrammable circuits can increase circuit flexibility and the scalability of complex cell-based computing devices. Here we present a new architecture to produce reprogrammable biological circuits that allow the development of a variety of different functions with minimal cell engineering. We demonstrate the feasibility of creating several circuits using only a small set of engineered cells, which can be externally reprogrammed to implement simple logics in response to specific inputs. In this regard, depending on the computation needs, a device composed of a number of defined cells can generate a variety of circuits without the need of further cell engineering or rearrangements. In addition, the inclusion of a memory module in the circuits strongly improved the digital response of the devices. The reprogrammability of biological circuits is an intrinsic capacity that is not provided in electronics and it may be used as a tool to solve complex biological problems.

INTRODUCTION

The response of a biological system to stimuli, whether natural or synthetic, can ultimately be understood in terms of computation. Information processing and adaptation are two basic aspects of living systems, and they are two faces of the computational nature of cellular circuits (1). Building synthetic cellular devices able to perform complex computations is at the root of synthetic biology (2,3). In this context, the establishment of standard technology for designing and building programmable cellular computers for general purposes would provide new solutions for complex problems in many fields, such as biomedicine -with new treatments for complex diseases (4–6), bioremediation (7) and bio-industry (8), among many others. However, progress in this respect is hampered by lack of technological maturity to overcome problems such as issues relating to metabolic burden (9), unexpected interactions with the host cell (9) and the so-called wiring problem. These factors are closely related to the particular nature of biological systems (10) and are not present when computation is performed in other substrates such as electronic ones. Consequently, to date, most cellular computational devices have been developed and optimized for a single purpose, i.e. one problem one solution, and have usually been obtained after a costly trial and error process (11).

To overcome these limitations, the use of multicellular consortia to build biological circuits has been explored (12–14). Previous studies demonstrated that distributed computation in such consortia allows complex computations (15) in an easier and reusable manner. In these multicellular embodiments, each engineered cell corresponds to a module of the computation, which provides the flexibility to create new circuits by exchanging cells without additional cellular engineering. Furthermore, the use of space as an additional computational element also helps to reduce the need for cell engineering (15,16). Despite these improvements, scalability in computational complexity continues to be constrained by the increasing number of cell-cell connections required, each one of which must be implemented by different molecules (17). To address this limitation, a new approach was implemented based on the use of inverted logic

*To whom correspondence should be addressed. Tel: +34 93 40 37110; Email: francesc.posas@irbbarcelona.org
Correspondence may also be addressed to Javier Macia. Tel: +34 93 316 05 39; Email: javier.macia@upf.edu
Correspondence may also be addressed to Eulàlia de Nadal. Tel: +34 93 40 39895; Email: eulalia.nadal@irbbarcelona.org

based in distinct spatially segregated multicellular consortia (18). This strategy allowed for circuits of high complexity (e.g. a multiplexer 4-to-1 that senses six different inputs) with simpler wiring requirements and with the possibility to act, for instance, as memory devices (19) or with time-dependent dynamic responses (20).

Building general-purpose biological systems is still a limitation in the field. A reprogrammable circuit offers clear advantages in terms of flexibility and reusability with respect to most biological circuits developed to date, which can perform only one defined computation, with different degrees of complexity (11,21). Here we have used multicellular consortia to generate fully reprogrammable cellular devices. To implement any arbitrary computation, we have created a universal platform composed of engineered cell types that modify their behavior in response to external reprogramming signals, thereby allowing different computations to be achieved using the same computing substrate (biological device). In addition, we have coupled to the reprogrammable biological devices a memory module that can store computation improving the digital output response of the circuit. The development of reconfigurable biological circuits serves to demonstrate that biological computation can be highly flexible and dynamic in complex environments.

MATERIALS AND METHODS

Engineered yeast cell library and growth conditions

Yeast strain W303 (*ade2-1 his3-11,15 leu2-3,112 trp1-1 ura3-1 can1-100*) was used in this study. Reprogrammable Input Layer cells (RIL cells) were genetically modified to produce *Saccharomyces cerevisiae* α-factor (ScαF) following XOR logic. The output layer cells (OL1 cells) controlled output expression (fluorescent protein yEGFP) in response to ScαF. The memory module includes two cells; both implement a NOT logic and the co-culture produces a double-negative-feedback. NOT ScαF and NOT SjαF cells inhibiting each other by the production of two different pheromones, *Schizosaccharomyces japonicus* α-factor long version (SjαF_l) and ScαF, respectively. Each cell type used in this study is fully described in the Supplementary Data. The schematic genotypic of cells and plasmids are summarized in Figure S2 and S12, and Tables S1 and S2. Recombinant DNA techniques and transformation of bacterial and yeast cells were performed using standard methods. Some plasmids were generated using the MoClo Yeast Toolkit (YTK) modular cloning system (22). Building of plasmid constructs to use with standards of MoClo Yeast was achieved using Golden Gate assembly as described in (22). All new part sequences were either mutated or synthesized to avoid of the BsmBI, BsaI, and NotI recognition sequences. Plasmids were amplified using *Escherichia coli* DH5α or OmniMax-competent cells grown at 37°C in LB medium supplemented with appropriate antibiotics. Yeast cells were grown overnight in YPD without reaching stationary phase and then diluted to mid-exponential phase at 30°C. Cell proportions and computing times for each computing circuit are indicated in the following sections and in figure legends.

Hormones and synthetic pheromones

17-β-Estradiol (EST), progesterone (PRO) and dexamethasone (DEX) hormones (Sigma-Aldrich) were dissolved following the manufacturer's recommendations. The synthetic α-factor mating pheromones from *Gibberella zeae* (GzαF; WCTWKGQPCW), *Candida albicans* (CaαF; GFRLTNFGYFEPG), *Paracoccidioides brasiliensis* (PbαF; WCTRPGQGC), and *Schizosaccharomyces japonicus* long version (SjαF_l; PERRVSDRVKQMLSHWWNFRNPDTAN) were synthesized by the Peptide Synthesis Facility (UPF). The long version of *S. japonicus* alpha factor was selected because of its ability to highly induce the pheromone pathway in *S. cerevisiae* cells carrying *SjSTE2* gene and particularly for the narrow dynamic range of activation at nM dose-range (24) useful to generate a clear digital behavior. The peptides were diluted in H_2O to a final concentration of 3 mM for GzαF,PbαF, and SjαF_l and 500 μM for CaαF and then stored at −20°C. *S. cerevisiae* α-factor mating pheromone (ScαF; WHWLQLKPGQPMY) was purchased from ChinaPeptides Co., Ltd and dissolved in 50% methanol.

Output fluorescence measurements

Computational output was assessed by flow cytometry. After circuit computation, samples were diluted in PBS and analyzed using spectral flow cytometry Sony SA3800 Spectral Analyzer (4-laser prism array and 32-channel PMT as detection optics) or Cytek® Aurora (4-laser and 64 Fluorescence Emission Detection Channels). The full spectrum (420–800 nm SA3800 or 365–829 nm Aurora) of 10 000 yeast cells were recorded from each sample according to their FSC and SSC distributions and unmixed to identify the fluorescence signal for each fluorophore (mCherry, iRFP or GFP). The OL1 cells express constitutive mCherry, to distinguish them from other cells in the consortia, were selected by gate filtration using mCherry signal. Specific emission of GFP fluorescence by OL1 gated cells was measured *versus* side scatter (SSC) or versus autofluorescence channel (AF) when Cytek® Aurora was used (Supplementary Figure S5A). When OL1 cells were used, an output expression below 30% of GFP-positive cells corresponded to the 0 logic state (low threshold) and above 60% to the 1 logic state (high threshold) as in (18). To read the output coming from the memory module, reporter SjαF cells were diluted in supernatant coming from circuits. Samples were analyzed using spectral flow cytometry Cytek® Aurora. The full spectrum (365–829 nm) of 5000 yeast cells were recorded from each sample according to their FSC and SSC distributions and unmixed to identify the fluorescence signal for each fluorophore (iRFP or GFP). Reporter SjαF cells express constitutive iRFP were selected by gate filtration using iRFP signal. Then mean GFP fluorescence *versus* autofluorescence channel (AF) was measured, plotted as GFP intensity in arbitrary units (AU), or expressed as fold-change of each chamber in front of different inputs combinations (Supplementary Figure S5B). Cytometry data were analyzed using FlowJo™ Software (BD Life Sciences) and expressed as mean percentage of GFP fluorescence of OL1 or fold-change in multi-chamber circuits. The mean GFP

intensity and chamber fold-change were calculated using the same software as for reporter SjαF cells.

For microscopic analyses, cells were harvested and resuspended in PBS (Supplementary Figure S9). Images were collected with a Nikon Eclipse Ti Microscope using NIS elements Software (Nikon) and analyzed using ImageJ.

Characterization of engineered cells

The growth rates and transfer functions of engineered cells of a given consortia were measured individually against all inputs for each circuit. For growth rate measurements, cells were diluted to $OD_{600\,nm}$ 0.05 in YPD for 18 h and their growth curve was measured every 30 min in a Synergy H1 (BioTeK). Growth rate (μ) of different strains was measured using Growth Curves Analysis app based on Opm R-package (https://github.com/mdphan/GrowthCurvesAnalysis_shinyApp). The different growth rates were used to balance the proportion of each cell within the consortia.

For transfer function calculation, RIL cells were grown in YPD to mid-exponential phase and then diluted and mixed with OL1 cells at a ratio of 7.5:1 (10:1 ratio for RIL2). The mixture was subjected to different concentrations of input, reprogrammers or combination of both at a range of concentrations. Results were expressed as the percentage of GFP-positive cells assessed by flow cytometry using Sony SA3800 Spectral Analyzer and experimental data were fitted to a Hill equation using the sigmoidal fitting described in (23) All tested cells exhibited proper behavior that allowed the definition of clear thresholds between the 0 and 1 logic states. Based on these results, we established the concentration of inputs used in each of the circuits.

The transfer function of the reporter SjαF cell was assessed by growing cells $OD_{600\,nm} = 0.2$ in YPD in response to different concentrations of synthetic SjαF_1. Samples were incubated for 4 h at 30°C and analyzed by flow cytometry using Cytek® Aurora and fitted to Hill equation (Supplementary Figure S13A). For the transfer function of the cells of the memory module, cells were grown individually or together at a ratio 3:1 (NOT-ScαF:NOT-SjαF) in YPD to mid-exponential phase. The mixture was subjected to different concentrations of SjαF_1, for NOT-SjαF, or ScαF for NOT-ScαF or both NOT cells. Supernatant was collected after 4 h and then reporter ScαF was added for 4 h to the quantification of ScaF coming from NOT- SjαF. The supernatant from NOT-SjαF or memory module was added to SjαF as before to measure SjαF_1 accumulation. Results obtained by flow cytometry using Cytek® Aurora were expressed as the percentage of reporter SjαF GFP-positive cells fitted to a Hill equation as described above. The percentage of activation was based on the maximum stimulation of reporter ScαF with synthetic ScαF or the reporter SjαF_1 with synthetic SjαF_1 (Supplementary Figure S13 B and C).

Crosstalk analyses

The orthogonality of the yeast hormones used in this study (EST, PRO, and DEX) was assessed previously (18). The orthogonality of pheromones was tested in (24). In addition, we tested the crosstalk of various α-factors using two distinct approaches. Cells expressing GFP under the control of *FUS1* promoter and the different *STE2* receptors (*ScSTE2*, *CaSTE2*, *GzSTE2*, *PbSTE2* and *SjSTE2*) were grown in YPD to mid-exponential phase with 12.5 μM of each α-factor (ScαF, CaαF, GzαF, PbαF and SjαF_1) (Supplementary Figure S7). Moreover, perturbation of OL1 cells caused by the presence of the α-factors was analyzed by measuring their transfer function (Supplementary Figure S3). Cells were treated with different concentrations of *S. cerevisiae* α-factor in the presence of 12.5 μM of GzαF, CaαF, PbαF and SjαF_1. In both experiments, cells at $OD_{600\,nm}$ 0.2 were incubated in YPD for 4 h at 30°C and analyzed by flow cytometry. Results are expressed as the percentage of GFP-positive cells, as described before, and experimental data were fitted to a Hill equation.

Implementation of reprogrammable circuits with OL1

The circuits shown in Figures 4–6, and S10 were built by mixing RIL cells (2 cells for 2-input circuit or 3 cells for 3-input circuit) with OL1 cells. The following cell ratios based on the growth rate and performance of each strain were used for 2-input circuits: RIL1:RIL2:OL1 (3:7:1), RIL1:RIL3:OL1 (4:6:1), and RIL2:RIL3:OL1 (2.8:1.8:1). In the 3-input circuit the ratio used were RIL1:RIL2:RIL3:OL1 (2.4:4:3.2:1). Cell consortia in mid-exponential phase were treated with different combinations of inputs (12.5 μM GzαF, 12.5 μM CaαF, and 12.5 μM PbαF) and, as is indicated, with combinations of reprogrammer molecules (0.5 μM EST, 1 μM PRO and 350 μM DEX) in 96-well plates. After 6 h of computation at 30°C, to allow the consortia to reach a stationary state and produce enough wiring molecules, the percentage of GFP-positive cells of the output layer for each combination of inputs in each chamber was analyzed by flow cytometry. A positive signal (more than 60%) in any consortium accounts for 1 as the final output of the circuit. The same was done for negative (<30%) output (0).

Several chambers were used in 3-input circuits (Figure 6) each one producing a different level of GFP expression from OL1. In order to unify the output of the circuit, GFP levels from the different chambers were mathematically calculated as follows. The basal levels of GFP for each chamber was then determined by calculating the mean of at least four conditions with the lowest % GFP. This 'chamber background' was subtracted from all values and the corrected values were then used to calculate the fold-change induction in each chamber considering the highest value obtained as 1. The final output of the circuit for each combination of inputs was determined by selecting the highest normalized value of all chambers. Finally, the mean and standard deviation of circuit outputs of different independent experiments were calculated to generate the figure plot.

Implementation of reprogrammable circuits with a memory module

The circuits shown in Figure 7 and Supplementary Figures S14–S16 were built by mixing the three different RIL cells with the memory module consisting in a mixture of a NOT-ScαF and a NOT-SjαF as the output

layer. In every chamber, RIL cells were mixed at initial $OD_{600\,nm}$ = 0.25 at proportion RIL1:RIL2 (2.5:3), RIL1:RIL3 (2.5:3) and RIL1:RIL2:RIL3 (1:1.35:1.25). After washing twice in YPD by centrifugation, cells NOT-ScαF and NOT-SjαF were diluted at $OD_{600\,nm} = 1$ with ratio 3:1 (NOT-ScαF:NOT-SjαF). As in circuits containing the OL1 cell as the output layer, cell consortia were treated with different combinations of inputs (12.5 μM GzαF, 12.5 μM CaαF and 12.5 μM PbαF) and, as is indicated, with combinations of reprogrammer molecules (0.5 μM EST, 1 μM PRO and 350 μM DEX) in 96-well plates. After 6 h of computation at 30°C, plates were centrifuged to eliminate the supernatant that contained inputs and reprogrammer molecules, cells were washed in YPD, diluted and incubated for an additional 16 h to fix the memory in the memory module. After the memory has been set, the supernatant of the different chambers was collected and the output was assessed using the reporter SjαF cells for 4 h (Supplementary Figure S5B). The output of the circuit was expressed in absolute numbers of GFP intensity (AU) (Figure 7 and Supplementary Figures S14 and S15) or fold-change (Supplementary Figure S16).

RESULTS

Design of reprogrammable multicellular circuits

To implement multiuse reprogrammable biological devices, we designed a new general architecture based on the spatial segregation of multicellular consortia with inverted logic (18–20) (Figure 1A). Several independent chambers organized in two different layers compose a general computational device. The first layer in each chamber is the input layer (IL), composed by different cell types able to produce an internal wiring molecule (α) in response to a single external input (x_i). The relationship between input detection and wiring molecule production can follow ID logic, i.e. α is produced when x_i is present, or NOT logic, i.e. α is produced when x_i is absent. The second layer is the output layer (OL), implementing the so-called inverted logic, i.e. the output is produced in the absence of α. Finally, outputs produced in each chamber are combined to give the final output of the device. Of note, in this architecture, the computation is encoded in the different combinations of ID or NOT logic present in the input layer.

To develop a general reprogrammable device, we made a new generation of sensor cells conforming a reprogrammable input layer (RIL). Each cell of the RIL has the ability to produce a wiring molecule α according to ID or NOT logic in response to external inputs x_i in a regulated manner (Figure 1B). Briefly, the logic, ID or NOT, performed by each cell type can be selected using an external reprogramming signal p_i, specific to each cell type. In the absence of the reprogramming signal p_i, a cell responding to input x_i behaves as an ID logic gate, i.e. $\alpha = Id(x_i)$, whereas in the presence of the reprogrammer the cell shifts its behavior, responding as a NOT logic gate, i.e. $\alpha = NOT(x_i)$. The overall behavior of a RIL cell can be defined as an XOR gate (Figure 1B). Of note, in each consortium, RIL cells produced the same wiring molecule α. The wiring molecules produced by these RIL XOR cells were combined and determine the response of them linked to the NOT gate of the OL1 cells. The combination of them yields the final outcome of the circuit. Interestingly, only one wire α was needed and the output of the whole circuit was the OR combination of each consortium output (Figure 1A).

This design allowed a single biological device to implement different logic circuits simply by externally adding a particular combination of reprogramming molecules. The scalability of this architecture in terms of number of different circuits, the maximum number of different cell types and the maximum number of chambers required to implement these circuits were computationally analyzed for circuits responding to a different number of inputs. The computational results revealed that the proposed approach is scalable and that a large number of computational circuits (e.g. 65 536 number of circuits for four inputs) could be implemented with a reduced number of cells and 8 chambers (Supplementary Figure S1A). Also, most of the 256 circuits for 3 inputs could be implemented with only 2–4 chambers (Figure S1B). Therefore, while the number of cells showed a linear increase depending on the number of inputs N, the number of chambers increases according to 2^N and the number of circuits that could be implemented scales super-exponentially, i.e. 2^{2^N}, thus, the circuits that could be implemented showed an exponential increase.

Design and genetic engineering of a cell library to implement reprogrammable biocomputing circuits

The implementation of the general platform for reprogrammable cellular circuits required the construction of a library of engineered cells involved in the different layers. For RIL cells, a general genetic architecture implementing XOR logic was designed (Figure 2A). Briefly, the expression of a gene of interest (GOI) was triggered by a transcription factor (TF) induced by two independent promoters that respond to either the presence of an input (x_i) or a reprogrammer (p_i), respectively. The regulation of TF activity had two layers of control (transcriptional and post-translational regulation). At the transcriptional level, the TF was induced by the presence of x_i or p_i or was inhibited by two different types of repressor (TR-B or TR-A), whose expression was regulated by x_i or p_i, respectively. TR-A (induced by p_i) inhibited the expression of the TF by x_i, and TR-B (induced by x_i) repressed the induction of the TF by p_i. At the post-translational level, the TF was degraded by the action of a post-translational repressor complex P-TR C-D. This complex involved two components, P-TR C, expressed in presence of x_i, and P-TR D expressed in presence of p_i. The design of this complex network of repressors and inducible promoters allowed us to obtain digital behavior of the cells in the consortia. Some of the inducible promoters that received the signal from x_i and p_i were synthetic hybrid promoters to be induced by the specific signal, but they also contained operator regions for specific transcriptional repressors (TRs). Therefore, these synthetic hybrid promoters were induced by desired input and could be inhibited using orthogonal and specific repressors for each promoter (TR A or TR B). The inhibition exerted by the transcriptional repressors TR A and TR B may not be sufficient to fully shut down promoter activity. To obtain a digital behavior, we im-

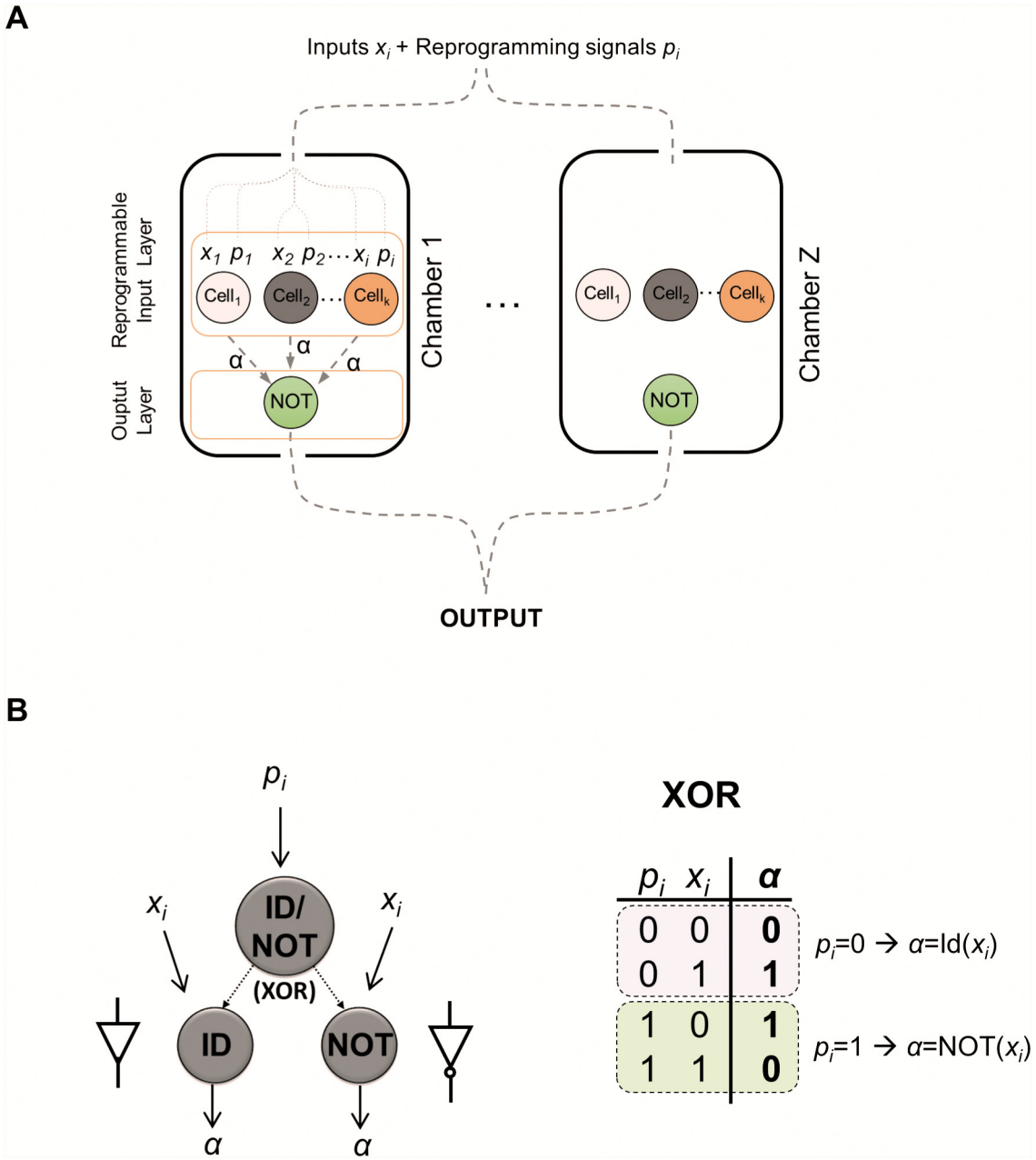
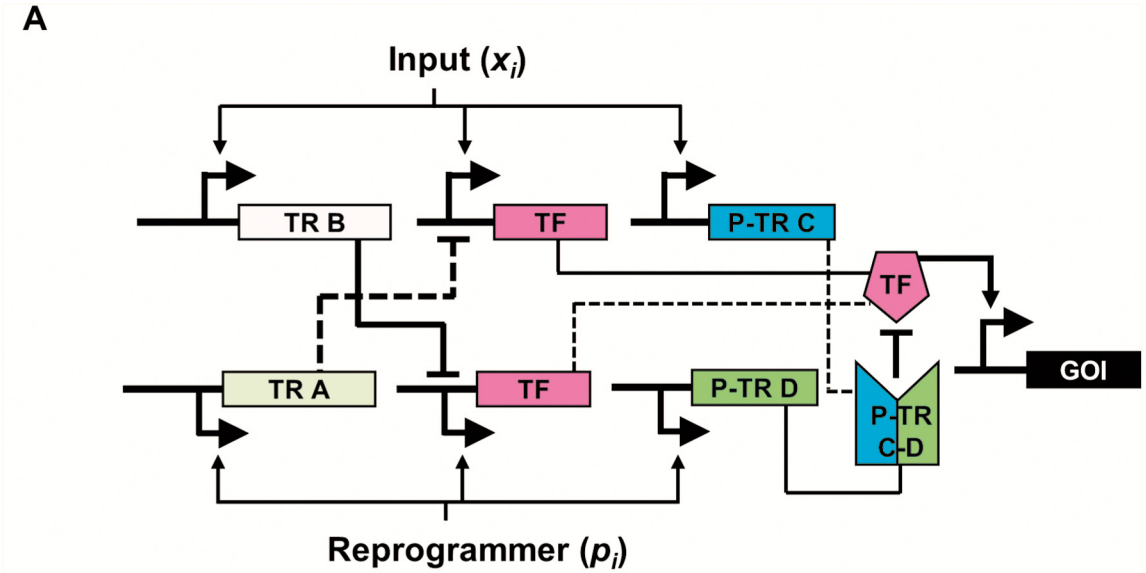


Figure 1. General architecture of multicellular reprogrammable circuits. (**A**) Any arbitrary reprogrammable cellular circuit can be implemented by combining Z different chambers. In each chamber there is a multicellular consortium composed by cells that detect external inputs x_i and reprogrammers p_i (the Reprogrammable Input layer (RIL)), and cells that produce the final output. i.e. the output layer. RIL cells produce a wiring molecule α in response to an external input x_i according to Identity (ID) logic, i.e. α is produced in the presence of x_i, or NOT logic, i.e. α is produced in the absence of x_i. The logic of the response can be re-programmed in RIL cells by means of external signals p_i. In the absence of p_i, RIL cells respond with ID logic, whereas in the presence of p_i the same cell behaves following NOT logic. (**B**) Schematic representation of the different behavior of RIL cells in response to the combination of the input x_i and the reprogramming molecule p_i. According to the truth table, the overall behavior of RIL cells corresponds to XOR logic.

plemented an additional layer of repression in the system by controlling TF degradation. Of note, post-translational repression must be done only when the two signals, x_i and p_i, are present in the media, so an active degradation complex is created only in this condition. This was accomplished using a degradation complex of two subunits that was active only when both P-TR C and P-TR D were present. Finally, the GOI was a secretable molecule that has to be excreted outside the cell in order to be used as a wiring molecule.

The logic of a RIL cell is then as follows (truth table in Figure 2A). In the absence of x_i and p_i, (i.e. $x_i = 0$ and $p_i = 0$), the TF was not produced and there was no expression of GOI (GOI = 0). In the presence only of input (x_i), i.e. $x_i = 1$ and $p_i = 0$, the TF, TR B and P-TR C were expressed. Under this configuration, the GOI was expressed (GOI = 1) because the full repressor complex P-TR was not formed, hence the promoter that responds to x_i was activated. The expression of TR B had no effect on the system. In contrast,



p_i	x_i	TR A	TR B	P-TR C	P-TR D	P-TR	TF	GOI
0	0	0	0	0	0	0	0	**0**
0	1	0	1	1	0	0	1	**1**
1	0	1	0	0	1	0	1	**1**
1	1	1	1	1	1	1	0	**0**

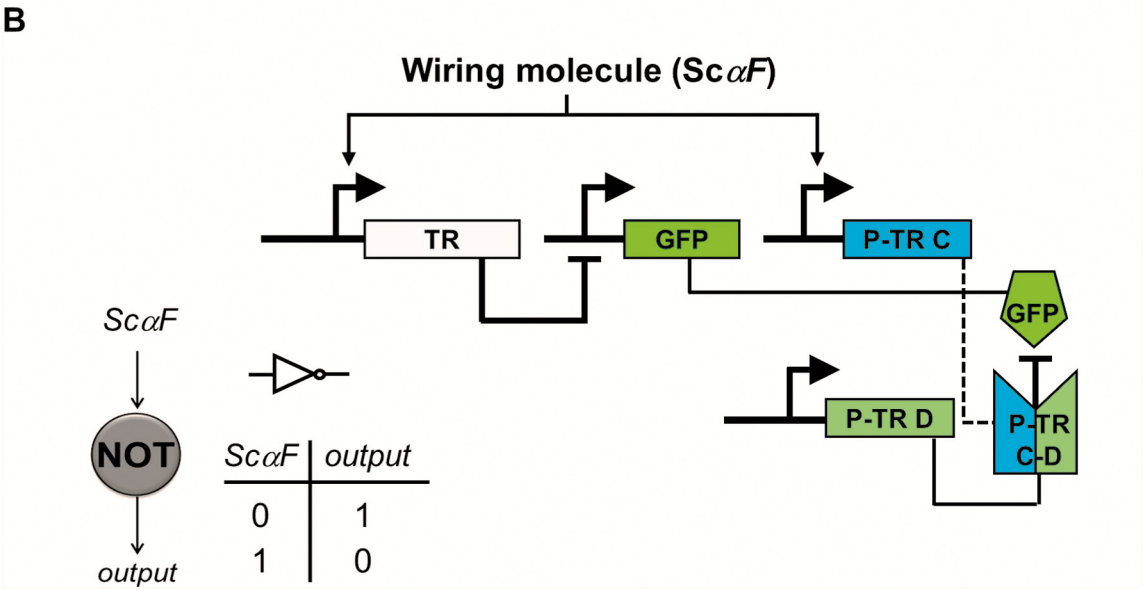


Figure 2. Schematic representation and internal architecture of input and output layer cells. (**A**) General architecture of RIL cells. Different promoters regulated by input x_i or reprogrammer signals p_i control the expression of Transcriptional Repressors (TR A and TR B), the Transcription Factor (TF), or the elements that regulate the degradation of the transcriptional factor [Post-translational Repressor (P-TR C-D)] that modulates the expression of the gene of interest (GOI). Truth table describing the expression of intermediate elements that regulate the expression of the GOI upon different combinations of signals. (**B**) General architecture of output layer cell. As in panel (A) the scheme shows the internal architecture to generate NOT logic using the wiring molecule (ScαF) coming from IL cells as an input. The promoters regulating the expression of a transcription factor (TF) and one of the elements of the repressing complex (P-TR A) are induced by the wiring molecule whilst the rest of promoters are constitutively active. Genetic circuits modulate the expression of GFP as output of the circuit.

in the absence of external input but in presence of a reprogrammable signal, i.e. $x_i = 0$ and $p_i = 1$, the TF, TR A and P-TR D were produced. Again, in this case, the GOI was expressed (GOI = 1) because induction of the TF was stimulated by p_i and the absence of the full repressor complex P-TR. The expression of TR A had no effect on the system. Finally, in the presence of an input and a reprogrammer (x_i and p_i), i.e. $x_i = 1$ and $p_i = 1$, the induction and activity of the TF was prevented by the two regulatory mechanisms. The main regulation was exerted by the repression of TF transcription though the binding of TR A and TR B at their respective promoters, which regulate the induction of the TF. To complement the transcriptional repression, the combination of P-TR C and P-TR D formed a functional P-TR C-D repressor complex that led to TF degradation. As a result of the action of the two mechanisms, the GOI was not expressed (GOI = 0).

To experimentally validate this general architecture, a library of engineered cells was built using *S. cerevisiae* as a model organism. We used pheromone α-factors from *G. zeae* (GzαF), *C. albicans* (CaαF) and *P. brasiliensis* (PbαF) as input molecules (x_i). The pheromone pathway is a MAPK signaling cascade that is well characterized in *S. cerevisiae* (25) and other yeast (26). The conservation of this pathway among yeast has allowed the use of highly orthogonal peptide pheromones from different species by simply replacing the corresponding upstream pheromone receptor (27). *S. cerevisiae* cells can be modified to express pheromone receptors from other fungi species that activate the pheromone signaling pathway in the presence of the α-factor of the same species (15,24). To create cells able to specifically sense the different pheromone inputs, we generated *S. cerevisiae* cells constitutively expressing GzSte2, CaSte2 or PbSte2 receptors. Three hormones, namely 17-β-estradiol (EST), progesterone (PRO) and dexamethasone (DEX), were used as reprogrammer molecules (reprogrammers) p_i. *S. cerevisiae* sense these hormones through the expression of specific receptors for each hormone (18,28) (see Supplementary Data for details).

To generate the library of RIL cells based on the use of these pheromone and hormone receptors, we integrated and modified several genes and promoters in the yeast genome (schematically depicted in Supplementary Figure S2; and fully described in Supplementary Data). The RIL cell library involved three cell types: RIL1 cells responding to GzαF that can be reprogrammed by EST; RIL2 cells responding to CaαF that can be reprogrammed by PRO; and RIL3 cells responding to PbαF that can be reprogrammed by DEX (see Supplementary Data for details). All these cells produced the same wiring molecule in response to input, following the logic defined by the reprogramming signal. The α-factor from *S. cerevisiae* (ScαF) serves as the wiring molecule to connect the RIL cells with the output layer.

The output layer was composed by a single cell type, namely OL1 (see Supplementary Data for details), which produced the final output of the circuit in response to the wiring molecule ScαF, following NOT logic. As proof-of-principle, the final output was the expression of a green fluorescent protein (GFP) that could be easily monitored by microscopy and flow cytometry (Figure 2B depicts the general architecture of this cell). The wiring molecule (ScαF) triggered the expression of a TF that inhibited the expression of constitutive promoters driving GFP expression. Like RIL cells, a second layer of control was introduced into the system, which regulated the levels of GFP by protein degradation. Thus, ScαF also activated the expression of P-TR C, which, combined with the constitutively expressed P-TR D, formed an active complex. The resulting P-TR C-D complex degraded GFP. The combination of the two mechanisms allowed for the implementation of NOT logic, in which GFP was not expressed in the presence of the wiring molecule ScαF. In contrast, in the absence of ScαF, TF and P-TR A were not produced, thus allowing the expression of GFP, as described in the corresponding truth table (Figure 2B). The response of OL1 cells to ScαF was well characterized and described in (18). Here we tested whether the transfer function of these cells in response to ScαF was affected by the presence of other α-factors used in the circuit (GzαF, CaαF, and PbαF). No significant differences in the transfer function of OL1 cells were observed in any case (Supplementary Figure S3), thus indicating the total orthogonality of the α-factors selected.

The RIL cell library implemented was able to respond to three distinct pheromone inputs and three reprogrammable molecules (steroid hormones), and it was fully characterized before combining them in complex circuits. Initially, we assessed the growth rate of the RIL and OL1 cells. Tested cells had very similar growth profiles (Supplementary Figure S4A). The similar fitness of all the cells in the consortia is essential to avoid disproportions of the circuit components during the computation and to ensure a robust response. For this reason, we measured the growth rate (μ) of each generated strain to calculate the initial amount of each cell to be included in the consortia to keep the proportion of cells during the computation that it is assessed in exponential phase (Supplementary Figure S4B).

Another parameter necessary to set up the cellular consortia is the performance of each cell in front of the inputs and reprogrammers. The transfer function of each cell in response to inputs and reprogrammers was evaluated. First, we quantified the activity of the RIL cells in combination with OL1 cells using flow cytometry (see Supplementary Figure S5 for experimental details). To this end, we measured GFP activity by flow cytometry in the presence of increasing concentrations of α-factors or hormones. For all conditions, the RIL cells exhibited proper double state behavior in response to inputs and reprogrammers (Supplementary Figure S6A), indicating high production of ScαF at low concentrations of inputs or reprogrammers. An additional transfer function was generated for each cell by fixing the α-factor inputs while the hormones were diluted to obtain their optimal range of concentrations for each type of cell used in the consortia (Supplementary Figure S6B).

In multicellular computation, the RIL and OL1 cells grew together in a chamber and the inputs and reprogrammers were presented to the whole cell consortium. It is therefore essential to assess the possible crosstalk between the different inputs and reprogrammers. The use of distinct chemical families for the pheromone inputs and reprogrammers blocked the possible crosstalk between the two differ-

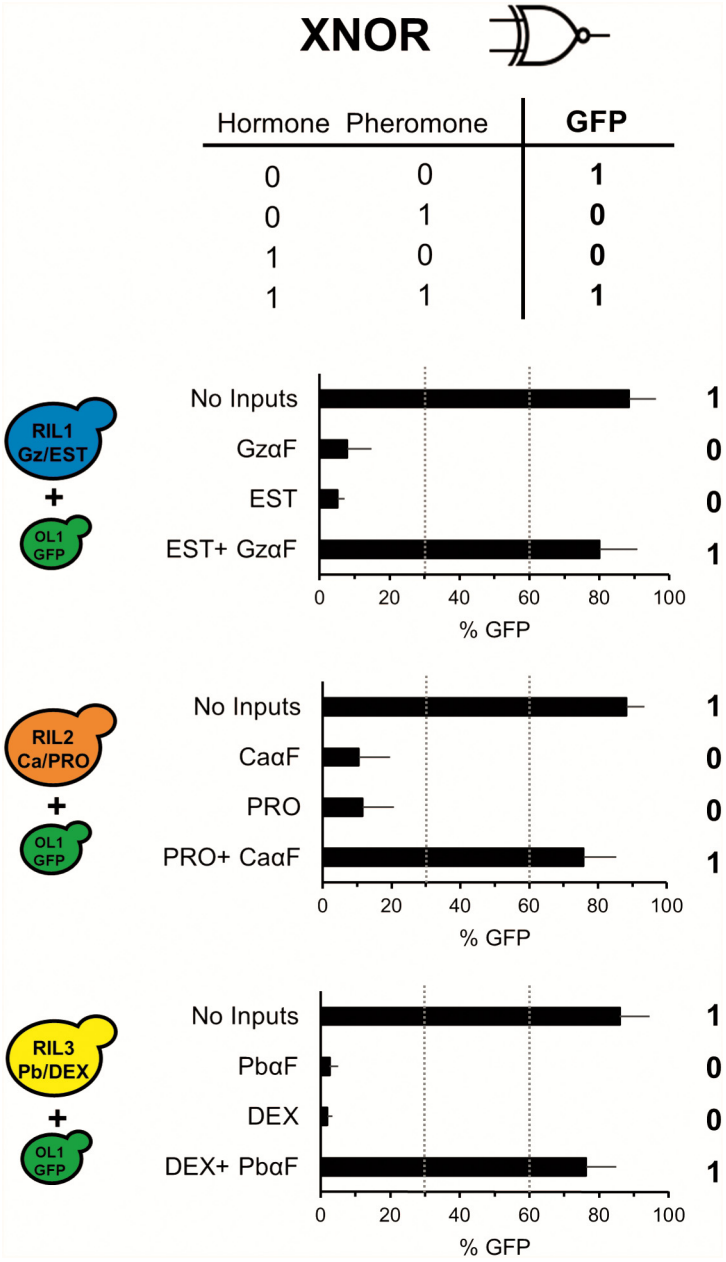


Figure 3. Assessment of cell computation of the library of engineered RIL cells mixed with OL1 cells. Truth table generated by mixing each type of input layer cell with the output cell (OL1). Cells stimulated with different combinations (see truth table) of input molecules (α-factor from *Gibberella zeae* (GzαF), *Candida albicans* (CaαF) or *Paracoccidioides brasiliensis* (PbαF)) at 12.5 μM and reprogrammer molecules (0.5 μM 17-β-estradiol (EST), 1 μM progesterone (PRO) or 350 μM dexamethasone (DEX)) were mixed with output OL1 cells. After 4 h of computation, the percentage of GFP-positive cells was analyzed using flow cytometry. Data represent the mean and standard error of five independent experiments.

ent signals. The orthogonality of yeast hormones used in this study (EST, PRO and DEX) has already been assessed (18). The orthogonality of pheromones was examined systematically (this work and (18,24,29)). Cells carrying *STE2* from the corresponding species of the α-factors used were built in a background were a pheromone-inducible *FUS1* promoter controlled the expression of GFP (see Supplementary Data for details). Cells induced the production of GFP only in the presence of their specific α-factor and no crosstalk with other α-factors was observed (Supplementary Figure S7).

To complete library characterization, a set of minimal circuits involving one RIL cell and the OL1 cell were built and characterized. Each circuit was composed by a RIL cell that senses a different input molecule, as well as a different reprogrammer and communicated with the output OL1 cell. The truth table of each cell consortium and the experimental results of GFP expression obtained by flow cytometry upon the different combinations of inputs and reprogramming signals are shown in Figure 3 and Supplementary Figure S8. Of note, each minimal circuit properly responded according to XNOR logic, resulting from the combination of

XOR logic in RIL cells with NOT logic in OL1 cells. Moreover, there was a clear separation between 0 and 1 logic states, indicating proper digital performance (Supplementary Figure S8). Similar results were obtained when quantifying cells by microscopy (Supplementary Figure S9). As in electronics, we defined a threshold. Cells producing an output above this threshold were considered to be in the 1 logic state, whereas when below the threshold cells were in the 0 logic state (see Material and methods). Taken together, the new engineered cell library behaved as expected in response to the different inputs and reprogrammers.

Implementation of reprogrammable single-chamber devices responding to two inputs

Based on the general architecture described above, we sought to implement different logic circuits using the same cellular device by only exposing it to external reprogrammers. To assess reprogrammability, first we considered biological devices composed by a single chamber containing RIL cells responding to 2 distinct inputs. Thus, we generated a general device responding to GzαF and CaαF as external inputs and containing two RIL cells (RIL1 and RIL2) in the input layer and OL1 in the output layer. Communication between the input and the output layers was mediated by ScαF as a wiring molecule. EST and PRO were used as reprogrammers in the circuit to perform different computations, such as simple 2-input logic gates (Figure 4A). As in the absence of EST and PRO (EST = 0 and PRO = 0), RIL cells responded as ID logic, which, combined with the OL1 cells, implemented a NOR circuit. The same cellular device behaved as an N-IMPLY circuit in the presence of EST (EST = 1 and PRO = 0). Finally, in presence of both reprogramming signals (EST = 1 and PRO = 1), the device implemented an AND logic gate (Figure 4A). Our results thus indicate that reprogrammability using the same biological device can be achieved to obtain different 2-input logic circuits.

A second device was implemented, including one of the previous cells (RIL1) with the RIL3 cell of the library. This single-chamber circuit responded to GzαF and PbαF as external inputs and was reprogrammed using EST and DEX. As in the previous circuit, the input layer was mixed with OL1 cells, and NOR, N-IMPLY and AND logic circuits were implemented depending on the combination of the reprogrammer signals (Figure 4B). A third device containing RIL2 and RIL3 cells was also implemented. In this device CaαF and PbαF where used as external inputs and was reprogrammed using PRO and DEX to achieve NOR, N-IMPLY and AND logic circuits (Supplementary Figure S10). Taken together, reprogrammable circuits can be implemented with the engineered RIL cells in the multicellular consortia.

Implementation of a multi-input reprogrammable single-chamber circuit responding to three inputs

To assess the feasibility of increasing the complexity of reprogrammable cellular circuits, a device responding to three external inputs in a single chamber was implemented. This device, which contained a consortium comprising the three RIL cells and the OL1 cells, responded to three inputs, namely GzαF, CaαF and PbαF, and could be reconfigured by three reprogrammers (EST, PRO and DEX). Figure 5 provides a schematic representation of this cellular device and the truth tables of the different logic circuits for all possible combinations of reprogrammers and external inputs. The experimental results showed good performance for every combination and, as in circuits with two inputs, a clear differentiation between 0 and 1 logic states was observed. Thus, chambers with three inputs and three reprogrammers can be used to obtain complex responses, thereby indicating that potentially more complex circuits could be created by combining several units of this single-chamber system.

Implementation of fully reprogrammable complex circuits responding to 3 inputs

The combination of three inputs and three reprogrammers permitted the implementation of highly complex reprogrammable devices. To demonstrate the feasibility of our approach, we built a device with four identical chambers responding to three input molecules (GzαF, CaαF and PbαF). Each chamber contained the same multicellular consortium composed by the three RIL and the OL1 cells (Figure 6A; pluripotential device). According to Supplementary Figure S1B, any arbitrary logic circuit (up to 256 different circuits) responding to three inputs can be reprogrammed in this device. Of note, although four is the maximum number of chambers for circuits responding up to three inputs, most of the circuits can be implemented with only three chambers (234 different circuits). In order to automate the design of complex reprogrammable circuits involving several inputs, chambers and reprogrammers, we developed a new open source software application called CDesign, available at https://zenodo.org/record/7278286 (see Supplementary Data for details). This application automatically transforms the truth table that defines the desired computation into a Boolean function expressed as a combination of XOR gates, responding to a combination of external inputs and reprogrammer signals, and NOT gates, responding to α molecules, (see Supplementary Data for a formal definition of this function) and maps this function of into a set of chambers. For each chamber, the software indicates the number of required RIL cells and the specific set of reprogrammers that need to be introduced in each chamber to configure a given circuit (see Supplementary Figure S11 with an example). After selection of the number of inputs and the type of response expected, this software defined the number of chambers required to implement the desired circuit and the cell configuration of the input layer that would be defined with the reprogrammers.

As proof-of-principle, we used the single pluripotential device to programme two different complex circuits: a three-bit adder and 2-to-1 multiplexer circuits. To set up the specific circuit in the pluripotential device, we used different combinations of reprogrammer signals (EST, PRO and DEX) in each of the four chambers, according to the CDesign software. Regarding the three-bit adder circuit (Figure 6B), in the first chamber, the wiring molecule ScαF was produced in response to GzαF following NOT logic (NOT

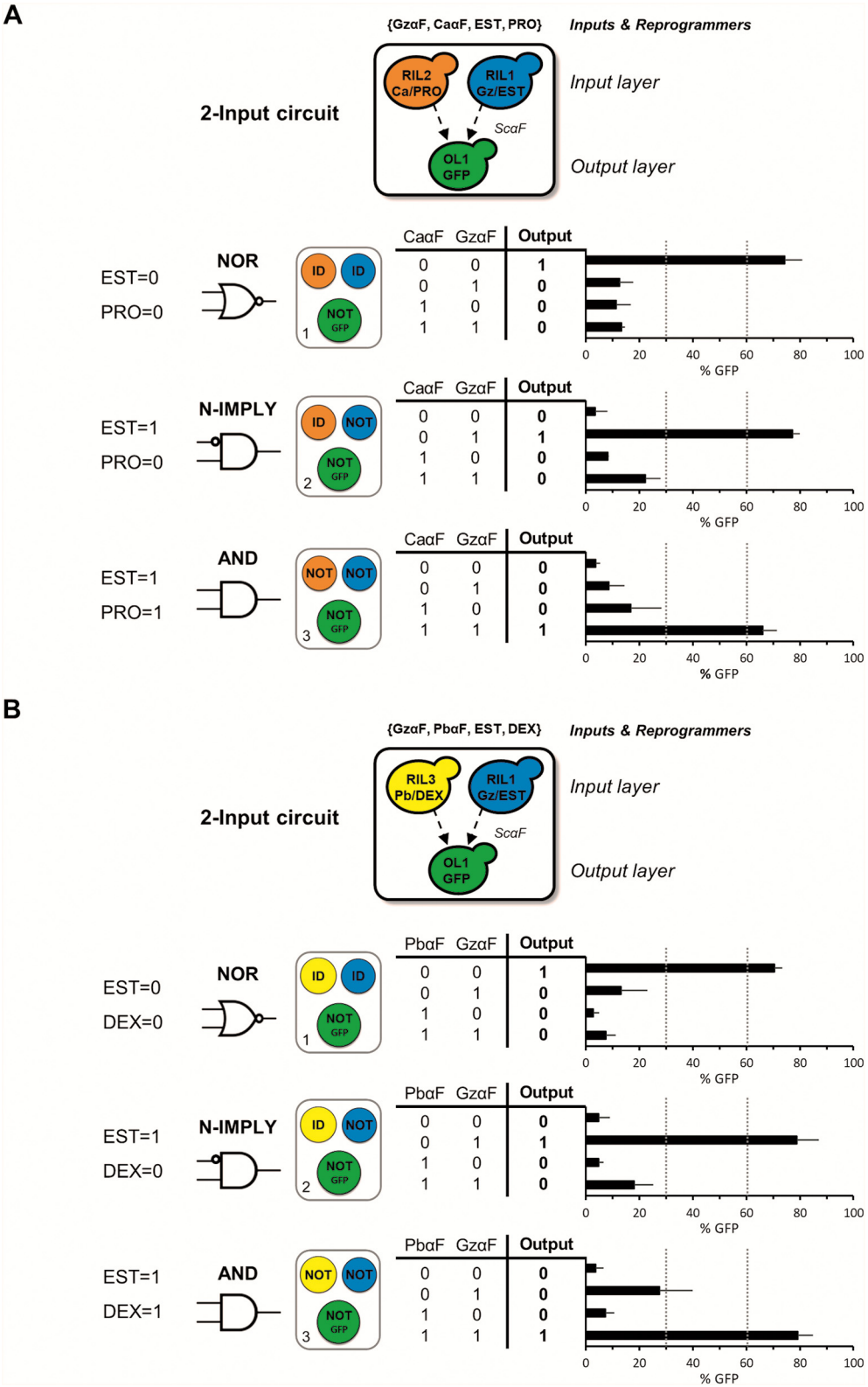


Figure 4. 2-input reprogrammable circuits. Upper panel: scheme of the chamber with 2-input circuit. The input layer is composed by RIL cells described in Figure 3 and the output layer by OL1 cells. Three different logic gates (NOR, N-IMPLY and AND gates) can be implemented using the same cell device. Schematic representation of the logic implemented in each circuit (left), truth table (middle) and percentage of GFP-positive cells (right). (**A**) RIL1, RIL2 and OL1 cells were mixed and treated with different combinations of inputs (12.5 μM GzαF, 12.5 μM CaαF) and reprogrammer molecules (0.5 μM EST, 1 μM PRO), as indicated. (**B**) RIL1, RIL3 and OL1 cells were mixed and treated with different combinations of inputs (12.5 μM GzαF, 12.5 μM PbαF) and reprogrammer molecules (0.5 μM EST, 350 μM DEX), as indicated. After 6 h of computation, the percentage of GFP-positive cells was analyzed using flow cytometry. Data represent the mean and standard error of five independent experiments.

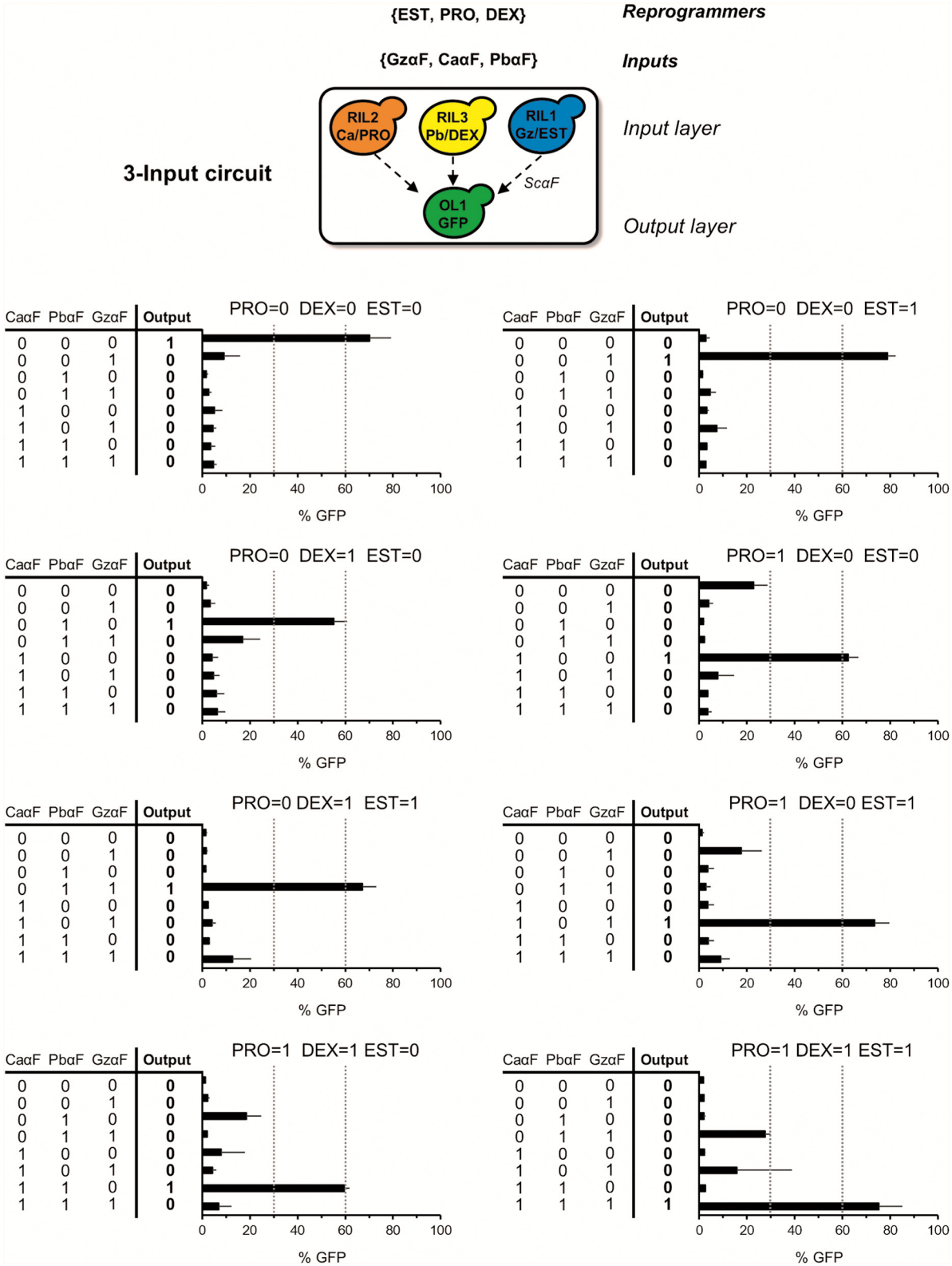


Figure 5. 3-input reprogrammable circuit. Top panel: Scheme of the chamber for 3-input circuit. The input layer is composed by the three RIL cells described in Figure 3. The output layer comprises OL1 cells. Truth table (left) and percentage of GFP-positive cells (right) for the different combinations of inputs and reprogrammer molecules. Cells were mixed and treated with different combinations of inputs (12.5 μM GzαF, 12.5 μM CaαF and 12.5 μM PbαF) and, as indicated, with combinations of reprogrammer molecules (0.5 μM EST, 1 μM PRO and 350 μM DEX). After 6 h of computation, the percentage of GFP-positive cells was analyzed using flow cytometry. Data represent the mean and standard error of three independent experiments.

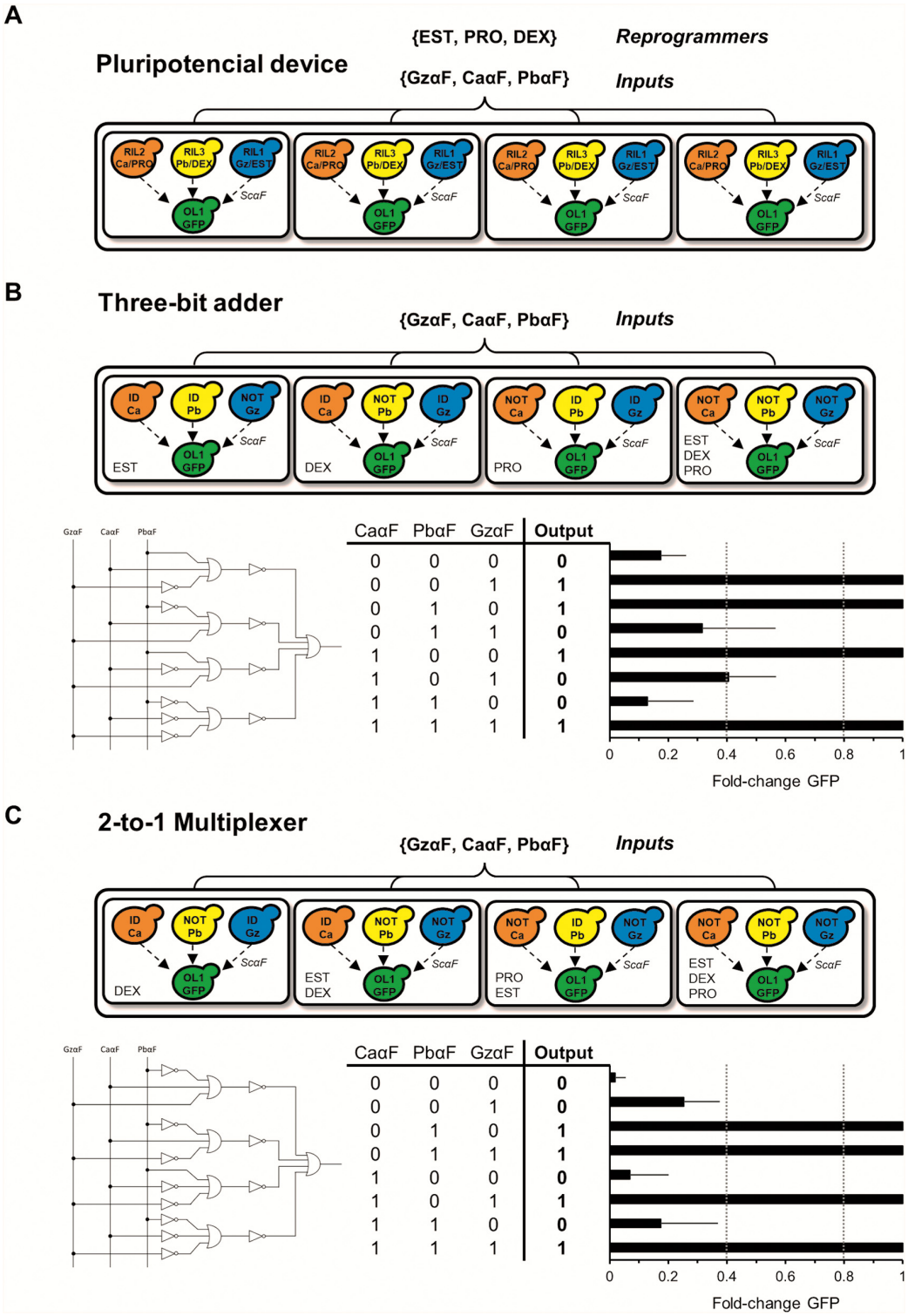


Figure 6. A pluripotential device can be reprogrammed to implement specific circuits such as a 3-bit adder or a multiplexor 2 to 1. (**A**) Scheme of the pluripotential device comprising 4 chambers. As in Figure 5, the input layer consists of the library of RIL cells, and the output layer is formed by OL1 cells. All 4 chambers have the same set of cells, which can be reprogrammed differently depending on the reprogrammer molecules present. (**B**, **C**) Top panel: Scheme of reprogrammed logic of the different cells in each chamber to implement the indicated circuit. All 4 chambers have the same set of cells, which are reprogrammed differently depending on the indicated reprogrammer molecules. Logic representation of the implemented circuit (left), truth table (middle) and the maximum fold-change of GFP-positive cells of the 4 chambers (right) for the different combinations of inputs is shown. Cells were treated with different combinations of inputs (12.5 μM GzαF, 12.5 μM CaαF and 12.5 μM PbαF) and, as indicated, reprogrammer molecules (0.5 μM EST, 1 μM PRO and 350 μM DEX). After 6 h of computation, the percentage of GFP-positive cells was analyzed using flow cytometry and quantified and normalized by the fold-change for each chamber. The highest value obtained from the four chambers was used to calculate the mean. Data represent the mean of maximum fold-changes and standard error of four independent experiments.

GzαF), whereas ScαF was produced in response to CaαF and PbαF following ID logic (ID CaαF and ID PbαF). To configure the RIL cells to achieve this behavior, only EST was supplied to this chamber (i.e. EST = 1, DEX = 0 and PRO = 0). Similarly, the proper configuration of the RIL cells in the second chamber was ID GzαF, ID CaαF and NOT PbαF, which was achieved by introducing only DEX into the chamber (i.e. EST = 0, DEX = 1 and PRO = 0). In the third chamber, the wiring molecule is produced following ID GzαF, NOT CaαF and ID PbαF, which was obtained by introducing PRO in the chamber (i.e. EST = 0, DEX = 0 and PRO = 1). Finally, the RIL cells in the last chamber responded following NOT GzαF, NOT CaαF and NOT PbαF logics when the three reprogrammers were added (i.e. EST = 1, DEX = 1 and PRO = 1). Once the four chambers had been reprogrammed using the defined molecules, all possible combinations of inputs were introduced, and the output, i.e. GFP, was experimentally measured for each chamber.

The design of this biological device is based on the use of multicellular consortia that are physically separated in order to use only one wiring molecule to allow communication between the cells in the input layer with those in the output layer. The spatial segregation also enables to use the same inputs and reprogrammer molecules in any chamber. For this reason, the production of the output is distributed, meaning that the final output of the circuit is the result of the combination of the outputs originated in the different chambers (18). Consequently, the final output of the circuit is obtained by deciphering the maximum expression of GFP from the 4 chambers for each combination of inputs. With the aim to standardize and facilitate the reading of the circuit, the GFP values obtained in each chamber were normalized by the fold-change of the maximum fluorescence of the chamber (details in Materials and Methods section). The experimental results shown in Figure 6B indicated that the device was properly reprogrammed implementing a three-bit adder circuit, consistent with the expected truth table.

In parallel, the same pluripotential device was reprogrammed to a 2-to-1 multiplexer (Figure 6C). As in the previous circuit, 4 chambers were also required. The configuration of the first chamber was ID GzαF, ID CaαF and NOT PbαF, obtained by adding EST = 0, DEX = 1 and PRO = 0. The configuration in the second chamber was NOT GzαF, ID CaαF and NOT PbαF, obtained by adding EST = 1, DEX = 1 and PRO = 0. The third chamber had the configuration of NOT GzαF, NOT CaαF and ID PbαF, obtained by adding EST = 1, DEX = 0 and PRO = 1. And the last chamber was NOT GzαF, NOT CaαF and NOT PbαF, obtained by adding EST = 1, DEX = 1 and PRO = 1. Like the previous circuit, once the device had been reprogrammed, all possible combinations of external inputs were assessed. The experimental results indicated that the computational device was properly reprogramed and implemented the truth table corresponding to a 2-to-1 multiplexer. Taken together, the data revealed that a single pluripotential device containing the same type of cells can be reconfigured to respond with different complex logics by only adding reprogrammers.

Reprogrammable circuits with a memory module to improve digital responses

The implementation of complex circuits is associated with an increase of the cellular complexity in the consortia, which affects the digital behavior of the circuits. To improve the output response of complex circuits, we implemented a memory module in the output layer. The use of a memory allows storing the output of the different internal states of input layer in a predictable manner and sustained over the time. In order to follow the architecture of multicellular reprogrammable circuits described in Figure 1A, the memory module implements the inverted logic. This can be achieved with a multicellular memory implementing a double-negative feedback loop as the one previously described in (19). This memory is based on the use of two NOT cells inhibiting each other by the production of two extracellular molecules. In addition, the output of the memory module is secreted which can be collected from the different chambers and detected with a reporter cell as the final output of the circuit. To this end, we designed and implemented two new NOT cells, NOT-ScαF and NOT-SjαF (see Supplementary figure S12 for details on the internal architecture) that constitutively expresses *S. japonicus* or *S. cerevisiae* alpha factors respectively. No crosstalk of SjαF_1 with other alpha factors or *STE2* genes used in the circuits was observed (Supplementary Figures S3 and S7). The production of the SjαF_1 by NOT-ScαF is inhibited by the presence of ScαF in the media, whereas the production of ScαF by NOT-SjαF is inhibited by SjαF_1 in the media. This double negative feedback loop permits to transit from one stable state to the other by the presence of the pheromones. In the absence of ScαF, that is no wiring molecule from RIL cells, the SjαF_1 is generated by the NOT-ScaF, inhibiting the secretion of ScαF from the NOT-SjαF and thus, it generates the stable state *on* of the memory, which can be read by the reporter SjαF. The negative-feedback loop generated by SjαF_1 blocks any production of ScαF and keeps the memory *on* (1 in Boolean notation). If ScαF is produced by the RIL cells of the consortia, it inhibits the production of SjαF_1 by the NOT-ScαF and the memory switches to an *off* state (0 in Boolean notation). Transfer functions of NOT-SjαF and NOT-ScαF showed a clear digital behavior that is even clearer when both cells are grown together (Supplementary Figure S13).

The memory module was first tested in combination with the three different RIL cells to assess the *on* and *off* states set by reprogrammable cells. Each RIL cell was mixed with the memory module and interrogated with the combination of inputs and reprogramming molecules. The use of the memory module combined with RIL cells clearly improved their digital behavior, especially in the *off* state (Supplementary Figure S14), in comparison with the use of OL1 cell in the output layer (Figure 3 and S8).

The use of the memory module was also tested by implementing circuits with two inputs such as NOR, N-IMPLY and AND circuits obtained by the reconfiguration of a 2-input device. As shown in Figures 4 and S10, the 2-input device can implement the different functions just by properly reprogramming the RIL cells. Here, the output layer cell OL1 was substituted by the memory module in a single

chamber. After incubation with the inputs and reprogrammers for 6 h, the media from the cellular consortia was removed to eliminate pheromones (inputs, wire and memory signals) and hormones (reprogrammers). Then the consortia was let grown in fresh YPD for 16 h to fix the state in the memory module and finally the supernatant was collected and assessed with the reporter SjαF. The experimental results using the combination of RIL1 and RIL2 cells (Supplementary Figure S15A) or RIL1 and RIL3 cells (Supplementary Figure S15B) showed good performance for the three logic circuits analyzed, and clear improvement between 0 and 1 states. Almost no signal is detected in *off* state and an abundant and consistent production of SjαF_1 is measured in the *on* state.

The memory module was also added in a single chamber responding to 3 inputs. The different combinations of reprogramming molecules were assessed in a chamber containing the 3 RIL cells of the input layer and the memory module. The output of the circuit was detected by a reporter SjαF cell in an extra output layer. The circuit was implemented under the same conditions as Figure 5, but after incubation with inputs and reprogrammers for 6 h, the media from the cellular consortia was removed. As before, the consortia was let grown for 16 h to fix the memory and the supernatant was collected. The fold-change of the reporter SjαF in the chamber was assessed. The experimental results showed good performance for every combination and a clear differentiation between 0 and 1 logic states was observed (Supplementary Figure S16).

Thanks to the improvement of the digital behavior obtained by the use of a memory module in circuits involving a single chamber, we designed and create a pluripotential device, as in Figure 6, responding to three inputs but now using the memory module instead of the OL1 in the consortia. Here, as a proof of concept, we implemented two complex circuits: a three-bit adder and 2-to-1 multiplexer circuits. To set up the specific circuit in the pluripotential device we performed the programming and incubation of cell consortia as the circuits in Figure 6, with the only difference that now the supernatant of the different chambers were collected and assessed using the reporter SjαF to quantify the output of the whole circuit. The experimental results indicated that the computational devices were properly reprogramed and implemented the truth table corresponding to both a three-bit adder and 2-to-1 multiplexer circuits (Figure 7). Of note, there was a clear digital response of the circuit, much better than when using OL1 cells in the output layer. Taken together, the data revealed that a single pluripotential device containing the same type of cells can be reconfigured to respond with a great digital response with different complex logics by only adding reprogrammers.

DISCUSSION

Cellular circuits with computational capabilities have promising applications for cellular computing in domains that are not easily reached by electronic computers. Successful outcomes in fields such as bioremediation (29), bioproduction (30) and targeted therapeutics (31) indicate that cellular computers can be extremely useful. However, because of the broad range of potential applications of cellular computers in complex environments, as happens with electronic computers, these devices should be flexible and able to modify their own functionalities for best performance. Reprogrammable cellular computers can provide this desired flexibility. The possibility of reconfiguring a cellular consortium is a characteristic of biological computers, offering plasticity that is not conferred by electronic computers (21). In electronic devices, a computer architecture describes the fix structural relationship between different components of a system and constrains how algorithms run on it, while in cellular computation, the composition, interactions or even logics can dynamically change to tackle a range of specific problems. However, despite the area of cellular computing has experienced significant development in recent years, in terms of different implementations of cellular devices and multiple applications, the creation of reprogrammable cellular devices has been much less developed. The main objective of this study was to develop and validate a general architecture for the creation of reprogrammable multicellular devices in which reprogramming is performed by means of external reprogramming signals.

In synthetic biology field, most standardization efforts have focused on genetic parts, *in silico* design tools or cloning practices (32). Several software tools has been developed to computer aided design of biocircuits (33,34). Some of the software enables building circuits building from different genetic parts, cell designs and after all, the implementation of specified logic functions. To date, no general methodology for building multipurpose cellular computers has been developed. To tackle this knowledge gap, this study presents a new approach to design and build reprogrammable cellular devices. The proposed architecture takes advantage of a multicellular embodiment. Distributed computation in multicellular consortia has been widely demonstrated to be a powerful methodology to develop complex logic devices with minimal genetic engineering (15,16,18–20,35). Cellular consortia have been used to solve challenging problems such as 2 × 2 maze problem using multi-input AND gates (36), to generate a digital displays (37) or for the production of synthetic products with metabolic pathways split in different cell populations (38). Even though complex circuits can be created using one cell (39), multicellular strategy overcomes most of the limiting constraints present in single-cell implementations, lowers the burden on each individual cell, and offers a way to more easily scale up the complexity of the computations performed.

When distributed computation is combined with spatial segregation (17), each cell of the multicellular consortium must perform only the minimal computation, i.e. responding to external inputs according to ID or NOT logic. Any arbitrary circuit can be constructed by integrating multiple NOR gates into a single strain (40), but in contrast with our ID/NOT approach, many genetic engineering is required to generate each circuit without option to reuse engineered cells for other circuits. As a given computation in a spatially segregated multicellular embodiment is encoded in the specific combinations of ID and NOT logic responses performed by each cell, these devices can be easily reprogrammed by simply modifying the logic of these cells, changing from ID to NOT logic or vice versa. To this end,

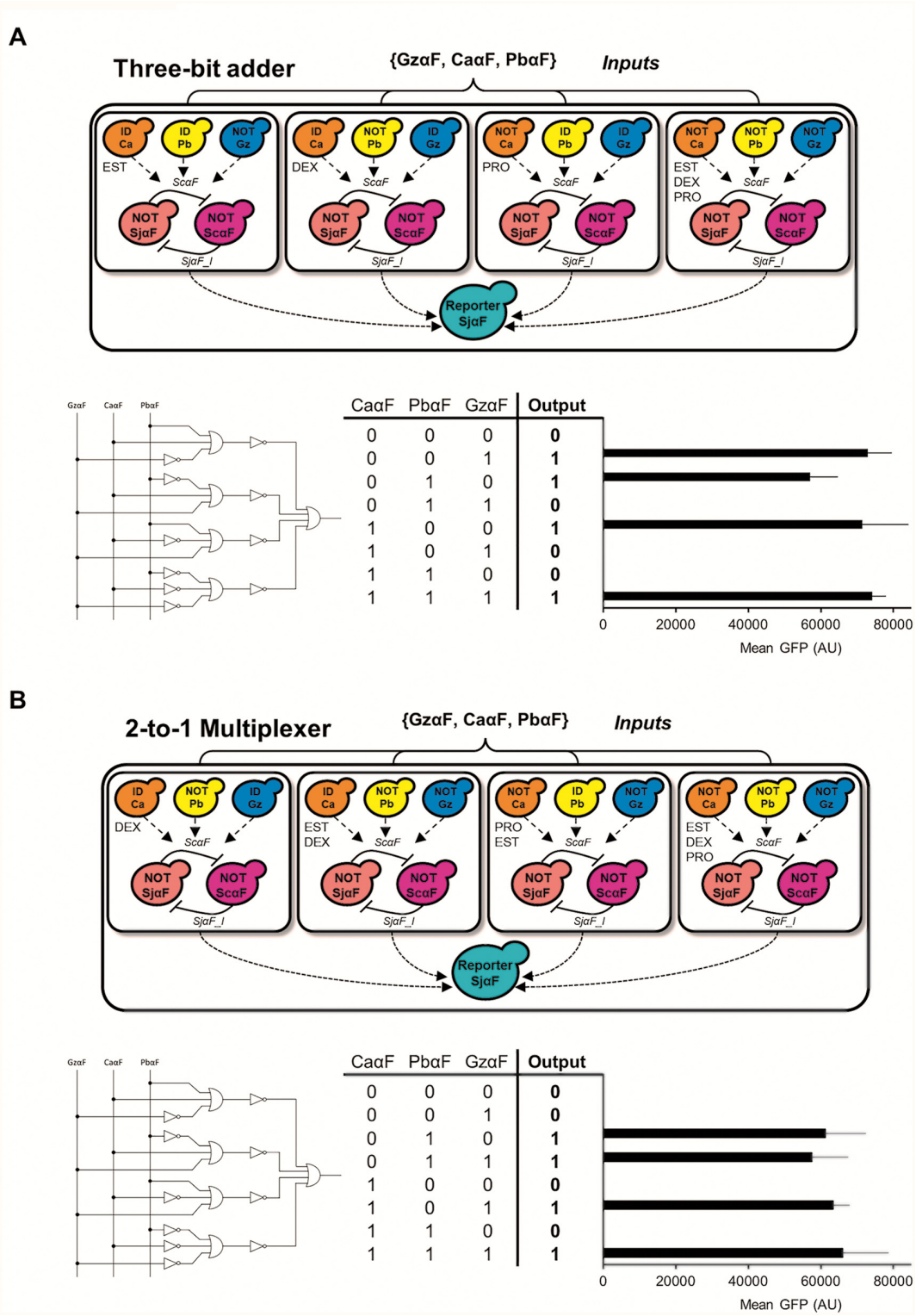


Figure 7. Implementation of the multicellular 3-input circuit with a memory module. (**A**) A 3-bit adder circuit and in (**B**) a 2-to-1 Multiplexer. In both, in the top panel represents a schematic diagram of the logic of the different cells in each chamber after reprogramming to implement the indicated circuit and the memory module. Cells in the memory module perform a double-negative-feedback loop. Both cell types respond with NOT logic inhibiting each other by the production of two different pheromones. NOT-ScαF cells produce and secrete SjαF_l α-factor mating pheromone. Reporter SjαF cells sense SjαF_l present in the medium and produce GFP. As in Figure 6, all four chambers have the same set of cells, which are reprogrammed according to the indicated reprogrammer molecules. Logic representation of the implemented circuit (left), truth table (middle) and mean GFP fluorescence of reporter SjαF measured by spectral flow cytometry is shown. After 6 h of computation, cells were washed and the memory was fixed. Supernatant from the different chambers was collected and the presence of SjαF assessed by the reporter SjαF as described before (for more experimental details see Supplementary Figure S5B). Data represent the mean GFP (AU) and standard error of 4 independent experiments.

here we developed a new library of engineered cells, based on a general genetic architecture able to implement ID or NOT logic. External reprogrammers allowed the selection of the logic to be followed by each cell. Experimental results revealed that these cells work efficiently and that reprogrammer signals can modulate the cellular behavior accordingly. In distributed computation, cell to cell communication is necessary, although some simple Boolean functions were created with distributed design without a wire (41). Cellular communication in bacteria circuits has been done using quorum sensing (42,43) conjugation (44) or specific communication systems to communicate cellular circuits using diffusible small molecules (41,45). In non-yeast eukaryotic systems, the communication between cells is more difficult, and it works better using 3D structures (46). Our model of distributed computation and spatial segregation limits the number of wire entities to only one yeast pheromone, facilitating the engineering and the reuse of cells. The use of different chamber also allows the reuse of the same reprogrammer and RIL cells, without needing numerous orthologue reprogrammer molecules and extensive engineered cell library. Theoretical works demonstrate that increasing the cell-to-cell communication could expand the number of implementable Boolean functions significantly (46) but with a payload of the need of multi-inputs cells with lots of genetic engineering, and several orthologues wiring molecules. In our reprogramming framework, complexity is distributed across the number of cells and chambers, rather than tough and time-consuming cell modifications. The limitation of our design is that in cases where the use of consortia isolation could be not applied, then a different approach should be implemented and the number of wires will increase. Also, we are aware that reprogramming requires the individual access of each chamber. Since the cellular composition is identical in all of the chambers within a device (pluripotencial devices) the only way to change the function of a given circuit is by selectively reprogram each chamber.

Based on described architecture, a universal reprogrammable computational device was designed. This device comprised the combination of identical chambers, which held the same multicellular consortium, involving the RIL and output inverted cells or a memory module. Our results demonstrate that this pluripotential device can be reprogrammed to perform completely different computations using the same cells. The codification of the specific computation to be performed was achieved by the introduction of different combinations of reprogrammers into each chamber, allowing the creation of cellular computers that can be remotely reconfigured.

The development of circuits with memory is essential step towards biological computation and decision-making systems. Long term memories can be obtained by DNA reorganization using recombinases (47–49), however biological memories where information can be stored and deleted or modified are essential for computation. Here, we implemented a memory module to improve remarkably the digital behavior of the circuits. The memory module is set with extracellular signals, consequently the memory state can be modified without need of cell engineering by addition of external modifiers (19). The use of a memory module allows for stable recording of the computation of each chamber with a clear differentiation of *on* and *off* states. The use memory also allows for buffered reading of the circuit output, an essential enhancement for multi-chamber embodiments.

Reprogrammable cellular devices offer enormous potential, and the approach presented in this study is especially appealing because of its simplicity. To date, exploration of reprogrammability of biological systems has been based on the control of bacterial community composition (50,51). Self-assembly DNA tiles can be reprogrammed to implement a limited number of 6-bit algorithms (52) but without universal reprogrammability. Computational simulations have demonstrated that a population of cells containing a reconfigurable logic gate can be switched between NOR and NAND logics (53) or multiple computations can be performed by a cellular population (54). A reconfiguration from OR logic to N-IMPLY by the use of dual-input hybrid promoters repressible upon blue light illumination is the only experimental example of reprogrammable logic using cells (55). Our work demonstrates that the reconfiguration of the logic of specific cells in multicellular consortia can be used to generate unlimited number of distinct complex circuits using the same biological 'hardware'. Such reprogrammable systems could be used in a wide variety of fields where direct access to the device is difficult. For example, in the field of biomedicine, a growing number of therapeutic applications rely on the use of cellular devices implanted in the patient. The possibility of having cell implants that can be reprogrammed externally to accommodate the patient's varying needs may be of great interest. Future work should be devoted to exploring other reprogrammable embodiments for cellular computing and their potential applications.

DATA AVAILABILITY

The selector circuit software code is available at https://zenodo.org/record/7278286.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We thank Romilde Manzoni, Arturo Urrios and Nuria Conde for their comments and discussion during the initial phase of this study. Thanks to Aida Fernández for technical assistance and members of our laboratories for critical feedback.

Authors contributions: D.C. designed and conducted the experiments and wrote the manuscript; N.O.V. participated in some experiments; S.M.D. participated in the generation of the circuit software code; F.P., E.dN. and J.M. designed and analyzed the experiment results and wrote the manuscript.

FUNDING

The laboratories of F.P. and E.dN. are supported by a coordinated grant from the Ministry of Science, Innovation, and Universities [PID2021-124723NB-C21/C22 and

FEDER, TED2021-131562B-100]; Government of Catalonia [2017 SGR 799, AEI-PID2020-119538RB-I00 to J.M.]; Fundació La Marató de TV3 [332/C/2016]; Office of Naval Research ONR and AFOSR [N62909-18-1-2065 NCE]; F.P. and E.dN. are recipients of an ICREA Acadèmia (Government of Catalonia); Ministry of Science, Innovation and Universities through the Centres of Excellence Severo Ochoa Award, and from the CERCA Programme of the Government of Catalonia and the Unidad de Excelencia María de Maeztu, funded by the AEI [CEX2018-000792-M]. Funding for open access charge: IRB core funding of the group.
Conflict of interest statement. None declared.

REFERENCES

1. Gerhart,J. and Kirschner,M. (1997) In: *Cells, Embryos, and Evolution: Toward a Cellular and Developmental Understanding of Phenotypic Variation and Evolutionary Adaptability*. Blackwell Science Malden.
2. Amos,M. (2004) In: *Cellular Computing*. Oxford University Press, USA.
3. Deplazes,A. and Huppenbauer,M. (2009) Synthetic organisms and living machines: positioning the products of synthetic biology at the borderline between living and non-living matter. *Syst. Synth. Biol.*, **3**, 55–63.
4. Brophy,J.A. and Voigt,C.A. (2014) Principles of genetic circuit design. *Nat. Methods*, **11**, 508–520.
5. Sedlmayer,F., Aubel,D. and Fussenegger,M. (2018) Synthetic gene circuits for the detection, elimination and prevention of disease. *Nat. Biomed. Eng.*, **2**, 399–415.
6. Ruder,W.C., Lu,T. and Collins,J.J. (2011) Synthetic biology moving into the clinic. *Science*, **333**, 1248–1252.
7. Jaiswal,S. and Shukla,P. (2020) Alternative strategies for microbial remediation of pollutants via synthetic biology. *Front. Microbiol.*, **11**, 808.
8. De Lorenzo,V., Prather,K.L., Chen,G., O'Day,E., von Kameke,C., Oyarzún,D.A., Hosta-Rigau,L., Alsafar,H., Cao,C. and Ji,W. (2018) The power of synthetic biology for bioproduction, remediation and pollution control. *EMBO Rep.*, **19**, e45658.
9. Ceroni,F., Boo,A., Furini,S., Gorochowski,T.E., Borkowski,O., Ladak,Y.N., Awan,A.R., Gilbert,C., Stan,G.B. and Ellis,T. (2018) Burden-driven feedback control of gene expression. *Nat. Methods*, **15**, 387–393.
10. Kwok,R. (2010) Five hard truths for synthetic biology. *Nature*, **463**, 288–290.
11. Moon,T.S., Lou,C., Tamsir,A., Stanton,B.C. and Voigt,C.A. (2012) Genetic programs constructed from layered logic gates in single cells. *Nature*, **491**, 249–253.
12. Brenner,K., You,L. and Arnold,F.H. (2008) Engineering microbial consortia: a new frontier in synthetic biology. *Trends Biotechnol.*, **26**, 483–489.
13. Chuang,J.S. (2012) Engineering multicellular traits in synthetic microbial populations. *Curr. Opin. Chem. Biol.*, **16**, 370–378.
14. Amos,M. (2014) Population-based microbial computing: a third wave of synthetic biology?*Int. J. Gen. Syst.*, **43**, 770–782.
15. Regot,S., Macia,J., Conde,N., Furukawa,K., Kjellen,J., Peeters,T., Hohmann,S., de Nadal,E., Posas,F. and Sole,R. (2011) Distributed biological computation with multicellular engineered networks. *Nature*, **469**, 207–211.
16. Tamsir,A., Tabor,J.J. and Voigt,C.A. (2011) Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'. *Nature*, **469**, 212–215.
17. Macia,J., Posas,F. and Sole,R.V. (2012) Distributed computation: the new wave of synthetic biology devices. *Trends Biotechnol.*, **30**, 342–349.
18. Macia,J., Manzoni,R., Conde,N., Urrios,A., de Nadal,E., Sole,R. and Posas,F. (2016) Implementation of complex biological logic circuits using spatially distributed multicellular consortia. *PLoS Comput. Biol.*, **12**, e1004685.
19. Urrios,A., Macia,J., Manzoni,R., Conde,N., Bonforti,A., de Nadal,E., Posas,F. and Sole,R. (2016) A synthetic multicellular memory device. *ACS Synth. Biol.*, **5**, 862–873.
20. Urrios,A., Gonzalez-Flo,E., Canadell,D., de Nadal,E., Macia,J. and Posas,F. (2018) Plug-and-play multicellular circuits with time-dependent dynamic responses. *ACS Synth. Biol.*, **7**, 1095–1104.
21. Grozinger,L., Amos,M., Gorochowski,T.E., Carbonell,P., Oyarzún,D.A., Stoof,R., Fellermann,H., Zuliani,P., Tas,H. and Goñi-Moreno,A. (2019) Pathways to cellular supremacy in biocomputing. *Nat. Commun.*, **10**, 5250.
22. Lee,M.E., DeLoache,W.C., Cervantes,B. and Dueber,J.E. (2015) A highly characterized yeast toolkit for modular, multipart assembly. *ACS Synth. Biol.*, **4**, 975–986.
23. Kemmer,G. and Keller,S. (2010) Nonlinear least-squares data fitting in excel spreadsheets. *Nat. Protoc.*, **5**, 267–281.
24. Billerbeck,S., Brisbois,J., Agmon,N., Jimenez,M., Temple,J., Shen,M., Boeke,J.D. and Cornish,V.W. (2018) A scalable peptide-GPCR language for engineering multicellular communication. *Nat. Commun.*, **9**, 5057.
25. Bardwell,L. (2004) A walk-through of the yeast mating pheromone response pathway. *Peptides*, **25**, 1465–1476.
26. Frawley,D. and Bayram,Ö. (2020) The pheromone response module, a mitogen-activated protein kinase pathway implicated in the regulation of fungal development, secondary metabolism and pathogenicity. *Fungal Genet. Biol.*, **144**, 103469.
27. Jones Jr,S.K. and Bennett,R.J. (2011) Fungal mating pheromones: choreographing the dating game. *Fungal Genet. Biol.*, **48**, 668–676.
28. Miller,C.A. III, Tan,X., Wilson,M., Bhattacharyya,S. and Ludwig,S. (2010) Single plasmids expressing human steroid hormone receptors and a reporter gene for use in yeast signaling assays. *Plasmid*, **63**, 73–78.
29. Dvorák,P., Nikel,P.I., Damborský,J. and de Lorenzo,V. (2017) Bioremediation 3.0: engineering pollutant-removing bacteria in the times of systemic biology. *Biotechnol. Adv.*, **35**, 845–866.
30. TerAvest,M.A., Li,Z. and Angenent,L.T. (2011) Bacteria-based biocomputing with cellular computing circuits to sense, decide, signal, and act. *Energy Environ. Sci.*, **4**, 4907–4916.
31. Chen,Y.Y. and Smolke,C.D. (2011) From DNA to targeted therapeutics: bringing synthetic biology to the clinic. *Sci. Transl. Med.*, **3**, 106ps42.
32. Tas,H., Amara,A., Cueva,M.E., Bongaerts,N., Calvo-Villamanan,A., Hamadache,S. and Vavitsas,K. (2020) Are synthetic biology standards applicable in everyday research practice?*Microb. Biotechnol.*, **13**, 1304–1308.
33. Nielsen,A.A., Der,B.S., Shin,J., Vaidyanathan,P., Paralanov,V., Strychalski,E.A., Ross,D., Densmore,D. and Voigt,C.A. (2016) Genetic circuit design automation. *Science*, **352**, aac7341.
34. Jones,T.S., Oliveira,S.M.D., Myers,C.J., Voigt,C.A. and Densmore,D. (2022) Genetic circuit design automation with cello 2.0. *Nat. Protoc.*, **17**, 1097–1113.
35. Guiziou,S., Mayonove,P. and Bonnet,J. (2019) Hierarchical composition of reliable recombinase logic devices. *Nat. Commun.*, **10**, 456.
36. Sarkar,K., Chakraborty,S., Bonnerjee,D. and Bagh,S. (2021) Distributed computing with engineered bacteria and its application in solving chemically generated 2 x 2 maze problems. *ACS Synth. Biol.*, **10**, 2456–2464.
37. Shin,J., Zhang,S., Der,B.S., Nielsen,A.A. and Voigt,C.A. (2020) Programming escherichia coli to function as a digital display. *Mol. Syst. Biol.*, **16**, e9401.
38. Tsoi,R., Wu,F., Zhang,C., Bewick,S., Karig,D. and You,L. (2018) Metabolic division of labor in microbial systems. *Proc. Natl. Acad. Sci. U.S.A*, **115**, 2526–2531.
39. Green,A.A., Kim,J., Ma,D., Silver,P.A., Collins,J.J. and Yin,P. (2017) Complex cellular logic computation using ribocomputing devices. *Nature*, **548**, 117–121.
40. Gander,M.W., Vrana,J.D., Voje,W.E., Carothers,J.M. and Klavins,E. (2017) Digital logic circuits in yeast with CRISPR-dCas9 NOR gates. *Nat. Commun.*, **8**, 15459.
41. Ji,W., Shi,H., Zhang,H., Sun,R., Xi,J., Wen,D., Feng,J., Chen,Y., Qin,X., Ma,Y. *et al.* (2013) A formalized design process for bacterial consortia that perform logic computing. *PLoS. One.*, **8**, e57482.
42. Scott,S.R. and Hasty,J. (2016) Quorum sensing communication modules for microbial consortia. *ACS Synth. Biol.*, **5**, 969–977.

43. Wang,S., Payne,G.F. and Bentley,W.E. (2020) Quorum sensing communication: molecularly connecting cells, their neighbors, and even devices. *Annu. Rev. Chem. Biomol. Eng*, **11**, 447–468.
44. Goñi-Moreno,A., Amos,M. and de la Cruz,F. (2013) Multicellular computing using conjugation for wiring. *PLoS. One.*, **8**, e65986.
45. Sexton,J.T. and Tabor,J.J. (2020) Multiplexing cell-cell communication. *Mol. Syst. Biol.*, **16**, e9618.
46. Al-Radhawi,M.A., Tran,A.P., Ernst,E.A., Chen,T., Voigt,C.A. and Sontag,E.D. (2020) Distributed implementation of boolean functions by transcriptional synthetic circuits. *ACS Synth. Biol.*, **9**, 2172–2187.
47. Siuti,P., Yazbek,J. and Lu,T.K. (2014) Engineering genetic circuits that compute and remember. *Nat. Protoc.*, **9**, 1292–1300.
48. Lloyd,J.P.B., Ly,F., Gong,P., Pflueger,J., Swain,T., Pflueger,C., Fourie,E., Khan,M.A., Kidd,B.N. and Lister,R. (2022) Synthetic memory circuits for stable cell reprogramming in plants. *Nat. Biotechnol.*, https://doi.org/10.1038/s41587-022-01383-2.
49. Siuti,P., Yazbek,J. and Lu,T.K. (2013) Synthetic circuits integrating logic and memory in living cells. *Nat. Biotechnol.*, **31**, 448–452.
50. Dinh,C.V., Chen,X. and Prather,K.L. (2020) Development of a quorum-sensing based circuit for control of coculture population composition in a naringenin production system. *ACS Synth. Biol.*, **9**, 590–597.
51. Liao,M.J., Din,M.O., Tsimring,L. and Hasty,J. (2019) Rock-paper-scissors: engineered population dynamics increase genetic stability. *Science*, **365**, 1045–1049.
52. Woods,D., Doty,D., Myhrvold,C., Hui,J., Zhou,F., Yin,P. and Winfree,E. (2019) Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. *Nature*, **567**, 366–372.
53. Goñi-Moreno,A. and Amos,M. (2012) A reconfigurable NAND/NOR genetic logic gate. *BMC Syst. Biol.*, **6**, 126.
54. Goñi-Moreno,A., de la Cruz,F., Rodríguez-Patón,A. and Amos,M. (2019) Dynamical task switching in cellular computers. *Life*, **9**, 14.
55. Jayaraman,P., Yeoh,J.W., Zhang,J. and Poh,C.L. (2018) Programming the dynamic control of bacterial gene expression with a chimeric Ligand- and Light-Based promoter system. *ACS Synth. Biol.*, **7**, 2627–2639.