Biological Computation in Yeast

Núria Conde Pueyo

TESI DOCTORAL UPF / ANY 2014

DIRECTORS DE LA TESI
Francesc Posas i Ricard Solé

Departament CEXS

Al meu Pare i la meva Mare,
culpables de predicar amb l’exemple i ser-hi sempre

Un dia el ruc al despertar es va dir:
"Vull aprendre de llegir;
si els savis per ser savis fan això,
també vull ser savi jo."
I alçant-se sense més raons se n'anà
a veure al mestre de minyons:
"Bon dia senyor mestre, li va dir;
vull aprendre a llegir."
El mestre li va dir: "Molt ben pensat;
n'aprendràs si ets aplicat."
"Aquesta lletra que aquí veus
que és com la teulada d'un convent,
és la primera lletra i es diu A.
Digues A per començar."
El ruc obrí una boca de tres pams
i engegà un parell de brams,
el mestre, satisfet, contestà: "molt bé vas!
saps perfectament la A."
El ruc inflat d'orgull, i pegant bots,
exclamava: "ja ho sé tot! ja ho sé tot!"
i ruc com va venir, ruc se'n va tornar,
i mai més de ruc passà.

Apel·les Mestres

Agraïments

Una tesi és un camí que es recorre amb moltes hores d'esforç, dedicació i preocupacions, moltes hores de laboratori i d'ordinador. Cada dia és un repte. No hi ha ruta marcada, apareixen dades que et fan ballar el cap, imprevistos que has d'anar resolent i noves complicacions que, més d'una vegada, et poden arribar a treure hores de son. Un camí que per a mi ja s'acaba. Abans, però, voldria agrair l'ajuda de tots els companys de viatge; tant de casa com del laboratori.

A la família: el Pare i la Mare pel seu recolzament, l'ajuda infinita i incondicional i els "tuppers". A la padrina que ja s'ho mereixia per tot el que va escoltar quan era a l'escola. A les meves Tates: l'Anna i la Gemma pels "cops de puny", els acudits i els ànims. I als que ja no hi són però van contribuir a que jo fos prou tossuda per arribar fins aquí: al Padrí i als Yayos.

A les nenes biòlogues, "els informàtics", la Pili, la Theta, el Pere, l'Esteban i a tots aquells amics que, amb totes les absències que comporta un compromís com l'elaboració d'una tesis, sempre m'han retrobat amb un somriure.

Als directors d'aquesta tesi, el Francesc i el Ricard, que han contribuït a formar-me com a professional i que m'han ensenyat que el més important és confiar en tu mateix i seguir lluitant, que això és el que val la pena, tot i que el resultat no sigui el que esperes.

A tota la llista llarga de companys de laboratori, tant del Grup de Senyalització com el de Complex.
En especial menció amb qui he compartit tots aquests anys de doctorat: El Salva, amb qui vaig sobreviure a SF; i la Mariona, que gaudint juntes de la ciència, s'ha convertit en una bona amiga amb la que "engordar".
A la Laia i el Javier per les hores de reunions. Als meus primers post-docs: el Tom, el Sergi (que ara sí que ja ho és) i el Carlos per les seves ensenyances científiques i, sobretot, de la vida. A la meva post-doc d'adopció: la Carme. Al Bernat, l'Andreea, l'Àlex, la Isabel, el Miqui i el Javi dels qui també vaig aprendre molt i van marxar. A l'Andrea, la

Silvia, l'Aïda, el Seisi, el Max, el Luís... que han arribat més tard però amb molta empenta! Al Josep, el Ben, el Sergi, l'Alba, l'Arnau, l'Albert, l'Alberto... bé, com ja he dit, a tots els companys que d'una manera o altra m'han ajudat, encara que el seu nom no figuri de forma explícita en aquestes línies. És impossible nombrar a tothom! M'agradaria pensar que he sabut transmetre l'agraïment dia a dia.

Per últim, agrair la col·laboració de tots aquelles institucions que -de forma directa o indirecta- han contribuït a que aquesta tesi s'iniciés i arribés a bon port.

Summary

Engineered synthetic biological devices have been designed to perform a variety of functions, from bioremediation to energy production and biomedicine. Ongoing efforts within synthetic biology have been directed towards the building of artificial computational devices using engineered biological units as basic building blocks. Such efforts, inspired in the standard design of electronic circuits, are limited by the difficulties arising from wiring the basic computational units (logic gates) through the appropriate connections, each one to be implemented by a different molecule. Thus, the actual main limitation of successful "in vivo" circuit implementation is the restrain of standard circuit design, which narrows the re–usability and scalability of the system.

Here, we show that there is a non–standard novel way of implementing complex Boolean logic computations that reduces wiring requirements thanks to a multicellular design with redundant distribution of the desired output among engineered cells. Practical implementations are presented using a library of engineered yeast cells, in which each genetic construct defines a logic function. This shows the great potential for re–utilization of small parts of the circuit to build distinct cells. In turn, the cells and their connections can be combined in multiple ways to allow easy building of more complex synthetic devices.

In the first manuscript, we proposed a modular multi–layer design involving a non–standard functional complete logic gates set. The engineered cells can perform the IDENTITY, NOT, AND and N–IMPLIES logics and are able to communicate with two different wiring molecules. As a proof of principle, we have implemented many logic gates by using just a few engineered cells. Noteworthy, small modifications and combination of those cells are enough for implementing more complex circuits such as a multiplexer or a 1–bit adder with carry. Our results support the approach of using cellular consortia as an efficient and robust way of engineering complex tasks not easily solvable using single–cell implementations.

SUMMARY

In the second manuscript, a universal architecture to engineer cellular consortia that is independent of the circuit's complexity is proposed. This novel design involves simple cells, performing `IDENTITY` and `NOT` logics, organized in two layers. The key aspect of the work is the spatial insulation of the modular design, that overcomes the combinatorial potential of standard circuits, solving the wiring problem and permitting re–usability. The potential of the approach is presented by implementing complex logical functions up to six inputs, such as a 4–input comparator or and a 4to1–multiplexer. Thus, this design allows easy implementation of multicellular computing chips in a flexible, robust and scalable manner.

Resum

Els dispositius biològics s'han dissenyat i sintetitzat per realitzar gran varietat d'aplicacions que van des de la bioremediació a la producció d'energia i a la biomedicina. En el camp de la biologia sintètica els esforços s'han dirigit a construir unitats biològiques enginyeritzades per usar–les com blocs bàsics en la construcció de dispositius computacionals artificials. Aquests esforços, inspirats en el disseny estàndard dels circuits electrònics, estan limitats per la dificultat que suposa cablejar les unitats bàsiques computacionals (portes lògiques), ja que cada connexió s'ha d'implementar amb una molècula diferent. Per tant, la principal limitació actual per a la implementació amb èxit de circuits "in vivo" és la restricció al disseny electrònic estàndard, el qual redueix la re–utilització i l'escalabilitat del sistema.

En aquesta tesi es mostra que hi ha una manera nova no–estàndard d'implementar funcions Booleanes complexes que redueix el requeriment del nombre de cables gràcies a un disseny multicel·lular amb una distribució de la sortida entre diferents cèl·lules. Es presenta una implementació pràctica utilitzant una llibreria de cèl·lules de llevat enginyeritzades, on cada constructe genètic defineix una funció lògica. Això posa de manifest el gran potencial que suposa re–utilitzar les parts del circuit per construir les diferents cèl·lules. Al mateix temps, les cèl·lules i les seves connexions es poden combinar de múltiples maneres permetent la construcció fàcil de dispositius sintètics més complexes.

En el primer article, proposem un disseny modular en múltiples capes que involucra un conjunt funcional complet de portes lògiques no–estàndard. Les cèl·lules modificades genèticament poden realitzar les lògiques `IDENTITY`, `NOT`, `AND` i `N-IMPLIES` i són capaces de comunicar–se utilitzant dues connexions diferents. Com a demostració experimental, s'han implementat varies portes lògiques utilitzant només unes poques cèl·lules. Cal mencionar a més, que petites modificacions i combinacions d'aquestes cèl·lules són suficients per implementar circuits més complexos tals com un multiplexor o un sumador d'un bit. Els

resultats recolzen l'enfocament d'utilitzar un conjunt cèl·lular com una manera eficient i robusta d'implementar tasques complexes que no serien gens fàcils d'aconseguir utilitzant una sola cèl·lula.

En el segon article, es proposa una arquitectura universal, que defineix un consorci cèl·lular, capaç d'implementar qualsevol circuit independentment de la seva complexitat. Aquest nou disseny només implica cèl·lules simples, que realitzen les lògiques `IDENTITY` i `NOT`, organitzades en dues capes. L'aspecte clau d'aquest disseny modular és l'aïllament espaial, ja que aquest reforça el potencial combinatori dels circuits solucionant el problema del cablejat permetent una alta re-utilització. El potencial d'aquest enfocament es demostra implementant funcions lògiques complexes de fins a sis entrades, tals com un comparador de 4 entrades o un multiplexor–4a1. Així, aquest disseny permetrà la fàcil implementació de "xips" computacionals multicèl·lulars d'una manera flexible, robusta i escalable.

Preface

The term of synthetic biology was first coined by the chemist Leduc in 1912, but it has not become the general term to describe the interface between molecular biology and engineering until beginning of 21st century. Synthetic biology is indeed a young scientific field. The fundamental idea behind it is that any biological system can be regarded as a combination of individual functional elements; like those found in man–made devices, aspiring to become a veritable methodology with which to construct complex biological systems following these principles. Synthetic biology is a shift from “molecular biology” to “modular biology”.

Pioneer works in the field have been the construction and characterization of a transcriptional toggle switch by Gardner et al. and the transcriptional oscillator of Elowitz et al. From these relatively simple examples, synthetic biologists have created many synthetic modules that incorporate sensing and processing components, able to respond to specific inputs and produce outputs that act as inputs for other modules or trigger cellular responses. By building and characterizing these modules, researchers have demonstrated their ability to control transcription, translation, protein regulation and pathway signalling. Therefore, synthetic biology also helps us to gain a deeper understanding of how biological systems function and brings the traditional engineering principles such as standardization, abstraction, and de–coupling into the field of cellular engineering.

The modules are typically intended for further reuse, to create new devices by combining them. However, although these modules function as predicted, stochastic fluctuations in gene expression levels and molecular cross talk significantly affects the ability to connect multiple modules to build a larger system in a single cell. In fact, unlike in electronics where all wires are made of the same material, in biological systems where substances share the same medium, each connection need to be performed by a different molecular element. Limiting, in that way,

the scalability. This is known as the wiring problem and to solve it became the motivation for our project.

Segregation of the basic modules in different cells can be an easy way of avoiding molecular crosstalk. It is feasible to implement a multicellular circuit where the computation is distributed among cells, and their communication mechanisms allow the cellular consortia to respond at a population level.

In this PhD thesis we have explored non–standard frameworks to design new multicellular architectures to reduce dramatically the need of different wiring molecules.

CONTENTS

INTRODUCTION

How do biological systems compute?

The metaphor of biological systems as homeostatic robust computing machines is not new. Computing is not —in fact, never was— a science only of the artificial. Since the 1930's pioneers of mainstream computer sciences, such John von Neumann, had been inspired by biology [Neumann, 1958]. Nature has achieved levels of complexity that by far surpass any man–made computing system. In the trillions of cells that make up a human being, functionally relevant faults are rare, and in the majority of cases, successfully detected and repaired. In this context, adopting features unique to the living world of cellular organization, such as self–replication and repair, and transpose them to the silicon world will be particularly desirable. In it extreme view, the biomimetic approach argues that the knowledge of the biological systems should be used to design artificial systems. In many ways, the molecular, cellular and tissue levels of biological organization have had a considerable inspirational impact on the development of computational paradigms. Such innovations include neural computing, cellular automatas or genetic programming [Rosenblatt, 1958; Wolfram, 1984; Holland, 1975].

However, there is a range of biological systems, from immune networks to ant colonies, where information processing escape from computational models such Turing machines, standard circuit engineering principles or neural networks. Information is not, as in a traditional computer machine, precisely and statically located in a particular place in the

system. Complex systems in biology have a fine, often noisy and fluid, grained architecture in that they consist of large numbers of relatively simple elements that work together in a highly parallel fashion. No individual component of the system can perceive the global state and have a limited information about their neighbours but alltogether constitute stable large–scale structures which allow complex collective decisions to take place (Fig 1). Here, information is through dynamic patterns over the system components. System actions are subject to noise and stochasticity but nevertheless exhibit unexpected levels of robustness. Remarkably, unlike artificial systems, this robustness is not a consequence of redundancy (two identical components of the system making the same function can replace each other in case of failure) but degeneracy (the capacity of elements of a given system that are structurally different to perform the same function) [Tononi et al., 1999; Macia and Sole, 2009].

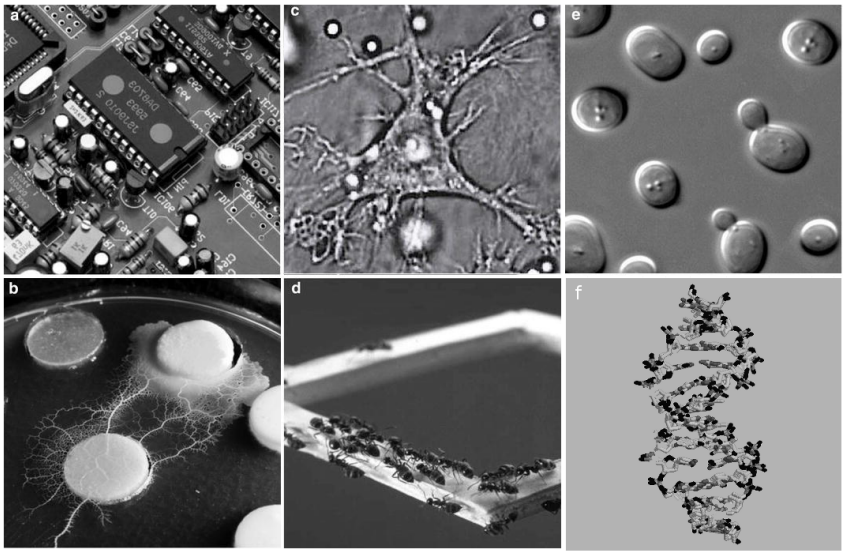


Figure 1: **Natural computing landscape examples.** **a** Electronic circuits, the iconic representation of standard man–made computing machines. **b** Physarum colonies able to decide among different food sources. **c** Dendritic immune system cells are a physiological example of distributed parallel computation. **d** Ant colonies able to find the shortest path among two alternatives. **e** Individual yeast cells respond to environmental fluctuations through complex intracellular decision making networks **f** Molecule of DNA act as a "information tape" to be read by the polymerase complex. Adapted from [Solé and Macía, 2013].

To fully appreciate the nature of signal processing in living systems, consider the response of a single cell to a single stimuli (for example, the response of a liver cell to glucagon). This is not a straight chain of cause and effect. It is a pattern of signals that spreads outward from the initial binding of ligand to its cognate receptor on the cell surface and passes down a number o parallel pathways before it triggers a specific response (for example, glucose production). Furthermore, the same second–messenger pathways can be influenced by other extracellular stimuli (such as insulin). There are also differences in the spatial and temporal unfolding of the various reactions, so that the spatio–temporal state of the network might change following an environmental input [Scott and Pawson, 2009]. Above all, the network of reactions is not entirely known yet. It is an inescapable conclusion that the molecular system of a living cell is an extremely complex and extensively interconnected network (Fig 2a).

Many aspects of cell behavior involve `computations`. For example, an enzyme in a biochemical pathway "reads" the concentration of its substrate and produces a corresponding level of product or a receptor on a cell surface `senses` the concentration of its ligand and `produces` a certain level of receptor–ligand complex to allow `signal transduction`. Molecular networks within cells can `amplify` external stimuli or even `adapt` to them. Another common feature of biological systems is their ability to `integrate multiple inputs`. Integration, in a complex molecular network context, rests ultimately on the convergence of signals into a single pathway and the generation of a specific response. For example, the regulation of DNA transcription in eukaryote cells that typically requires the assembly of complexes composed of enzymes, transcription factors and gene regulatory proteins, each with a different "blend" of proteins. Bacterial chemotaxis perfectly illustrates the integrative nature of biological systems [Bassler, 1999]. Molecular circuits also have capacity for `timing`, for example guidance of the cell through cell cycle by cascades of protein phosphorylation. Moreover, the permanent information encoded in DNA molecule or the imprint of the environment on the

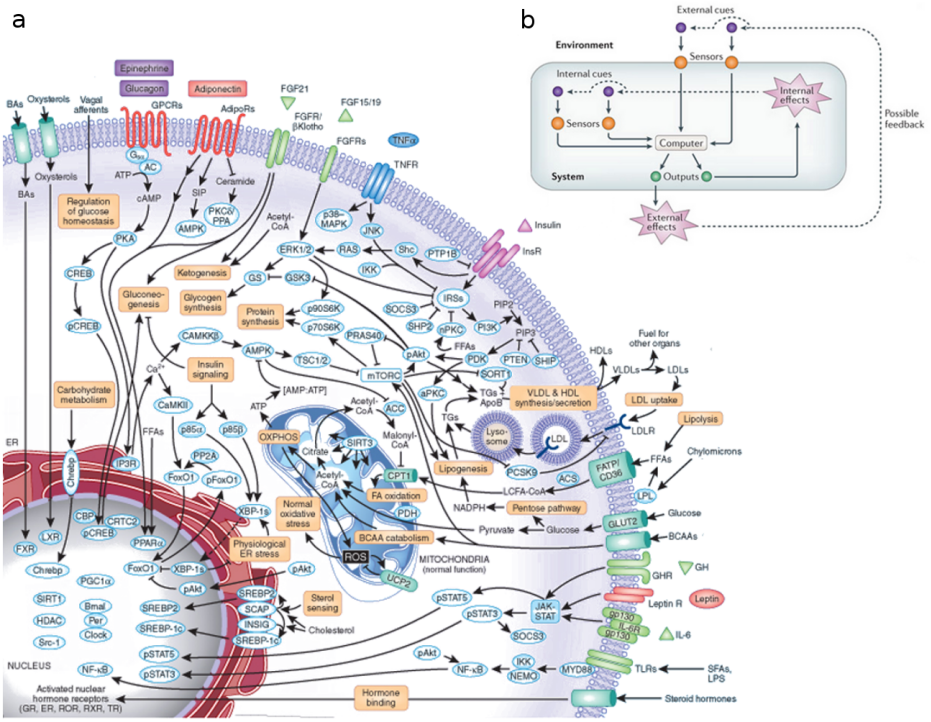


Figure 2: **Descriptive models of biological computation.** **a** Graphical representation of hepatocyte cell signalling. **b** General schema of a biological system. Adapted from [Levinson et al., 2012]; [Benenson, Yaakov, 2012]

concentration and activity of many thousands of molecules of the cell are a `memory` trace [Bray, 1995].

Using a simplified picture, living cells are dynamic systems that use complex molecular networks that sense environmental and internal signals, the `inputs`, transmit the information and process the proper physiological response, the `output` (Fig 2b). Computation is evident even in one–input one–output systems such as an inducible expression vector, because the specific way in which the inducer and the gene expression is linked is the result of a fixed relationship between them. In fact, any computation procedure (`program`/`function`)

can be formally described as a collection of relationships between any combination of inputs and its specific output; the so called `truth table` (Fig 3c). Biological computer "programs" generate a certain physiological effect in a cell depending on specific intra–and–extracellular cues; in the simplest case reflecting "all–or–none" cues of state at a given time. For example, "produce the effect when gene A is active and gene B is inactive and gene C is mutated" as depicted in Fig3d. Because any mapping of "all–or–none" cues to an "all–or–none" outcome is a logic function, diverse monitoring and control applications in cells can be enabled by systematic approaches to construct molecular logic circuits. As explained below, this circuits known as digital or boolean circuits, data is transmitted through `wires` between small computational units (`logic gates`) that perform simple operations (Fig 3b). These circuits resemble coupled molecular reactions in which the concentration of individual substances is interpreted as their values and interactions are compared to wires.

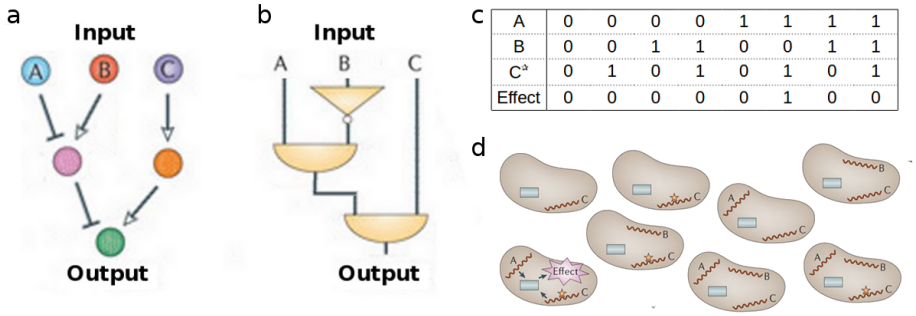


Figure 3: **Abstract models of biological computation.** **a** Network diagram abstraction to describe a decision function (`f`) with multiple inputs. **b** Scheme of the `f` logic function as a circuit diagram with the standard electronic symbols. **c** Truth table describing `f` complex function. **d** Pictorial exemplification of possible states of the same function (`f`) within the cell. Adapted from [Benenson, Yaakov, 2012].

Additional mathematical formalisms of computation also provides an accurate approximation of the molecular computation systems. The artificial neural networks (NN) consists of a layered architecture of units with an input layer, one or more "hidden" layers at intermediate levels

and an output layer [Rumelhart et al., 1986]. Connections are defined between units in one layer and units in lower layers. The state of each unit is determined by all its input connections and the output is a function of its state. Thus, a cell signalling network shares with NN the fundamental characteristic that it generates an output from an input through multiple interacting channels. However, cell signalling units are connected in a a network that does not obviously fall into distinct layers and moreover the signals of a cell signalling network do not always move in forward direction [Bray, 1990; Seoane and Solé, 2013]. Of note, "all–or–none" logic functions can also be represented as a network diagrams where nodes represent the molecular entities (Fig 3a).

Another example of mathematical formalisms of computation is the Turing's classical model of computation that, in principle, can describe any function or algorithm. A Turing machine is composed of many possible interconnected states, of which, one will be the final elected. This one is determined by the transition rules between states and the symbols of the string read. This formalization seems to be biologically supported by the fact that ribosomes act reading a "tape" defined by the messenger RNA, creating an output chain of aminoacids and starting and ending the process by means of detecting given sequences [Turing, 1936; Gurevich, 2000]. Pretty much as Turing–like machines the protein defines the outcome of the computation process (Fig 4).

The previous examples and paradigms define major threads in the universe of potential forms of computation. However, we are far from a completed picture of this universe. Defining a qualitative `theoretical morphospace` is a way of geometrically classify the spectrum of all possible computational structures. A tentative (and by no means exhaustive) picture of this space is provided in Fig 5.

The three axes are intended to capture the key features of computational systems. These are: (1) spatial embedding, (2) diversity of units and (3) degree of parallelism in computation. Space provides a well–defined dimension. It allows multiple segregate components to interact locally with others, provides a natural source of modularization and it also

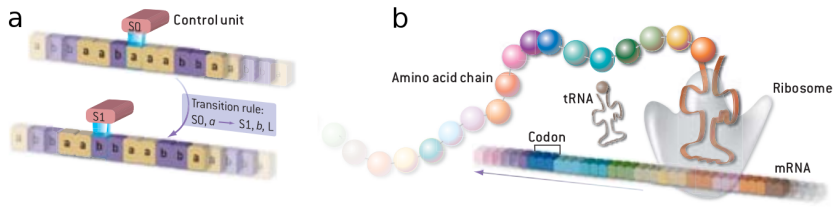


Figure 4: **Turing's classical model of computation. a** Hypothetical Turing machine device that operates on an information–encoding tape bearing symbols such as "a" and "b". The control unit reads and writes the tape, one symbol position at time, according to instructions provided by transition rules. The transition rule in this example dictates that if it the control unit's state is S0 and the symbol read is a, then the unit should change its state to S1, change the symbol to b and move left one position in the tape. **b** A ribosome reads the information encoded in a mRNA strand, one codon at time (Codons are the symbolic alphabet), and the tRNAs deliver the correct aminoacid. Adapted from [Shapiro and Benenson, 2006].

defines a deep connection between pattern formation. The role played by space is particularly important when dealing with communication taking place among different individuals, whether cells or ants. Phenotypic diversity among involved elements provides the second axis for our computational morphospace it could refer to genotype different cells for engineered systems or can be ant castes or electronic components in other contexts. A diverse range of agents able to perform different operations is another way of exploiting modularity and division of labour. Cell–cell communication in a spatial context allows to explore the formation of multicellular systems [Kinkhabwala and Bastiaens, 2010]. The third axis would be the relevance of parallel distribution of the computational task under consideration. Parallelism is a powerful solution to assure reliable computation against failure of some components. Just a glance, allows us to see what part of the space is occupied and what part is empty. Indeed, there is a largely unexplored space, where spatial degrees of freedom can help to further simplify our implementation and simultaneously increase our combinatorial power. The distributed computation approach is defined here as the right corner of this diagram (thick edges).

Nowadays the potential of synthetic biology provides the best scenario to construct de novo engineered living matter (molecules and cells) whereas non standard forms of computation are used as alternatives to the classical

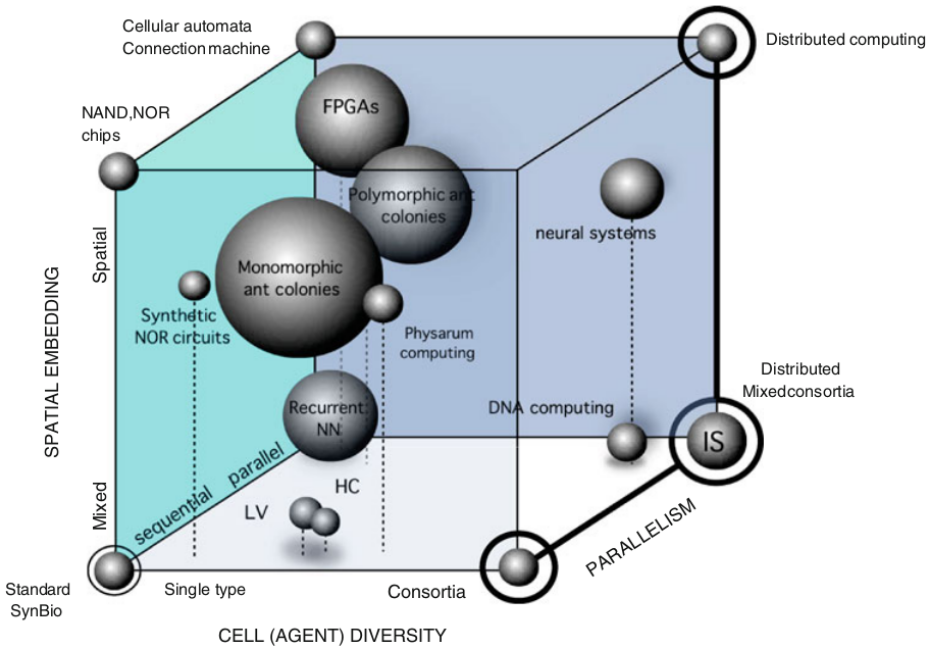


Figure 5: **Three–dimensional space representing an idealized landscape of biological computation**. Three axes to capture the key features: (1) to what extent is spatially segregated, (2) the degree of multicellularity (agent diversity) (3) to what extend the computation is sequential or parallel. Several well–known examples are located at roughly representative locations. The size of spheres is intended to reflect the abundance of known systems in each class. Most synthetic designs occupy the left wall of the cube. Distributed computation, that we explored in this work, occupies the domain highlighted on the right face of the cube using thick edges. Here: `LV` (Lotka–Volterra) and `HL` (mutualistic) synthetic ecosystems, `NAND, NOR Chips` are small chips mede up only by mentioned gates, `FPGAs` field programmalbe arrays, `NN` neural networks and `IS` Immune system. Adapted from [Solé and Macía, 2013]

engineering–inspired. Not surprisingly, computation has become a key component within the emergent field of synthetic biology [Bashor et al., 2010; Denning, 2007].

Basics of computer science

Computation was present long before computers were invented, although the remarkable shift to this realization occurred only in the last decade. The most essential foundations of Computer science are grounded in mathematics, it was the development in mathematical logic together with the computability theory what lead to the Computer Science as a whole. In order to perform a rigorous study of computation, computer scientists work with mathematical abstractions named models of computation. There are several models in use such as the already mentioned Turing machines or Boolean Networks. Interestingly, it was a mathematical model based on the brain learning supported with biological data what lead to the establishment of neural networks and parallel distributed processing [Rumelhart et al., 1986; Rosenblatt, 1958].

In parallel computation, in which many calculations are carried out simultaneously, operates with the principle that large problems can often be divided into smaller ones which can be solved concurrently. Communication and synchronization between the different subtasks are typically some of the greatest obstacles to good parallel performance. Distributed computation defines the computation with many autonomous computational entities, each of which has its own local memory, communicate their actions by passing messages and interact with each other in order to achieve a common goal. Three significant characteristics of distributed systems are: concurrency of components, lack of a global clock, and robustness of independent failure of components. An important challenge of distributed systems is the location of their elements. Many natural systems computations exhibit these features [Abelson et al., 1995; Gordon, DeborahM., 1999; Ben-Jacob and Levine, 2006].

Biological computation is a highly integrative discipline. It incorporates elements of systems biology, information theory, genetic engineering and molecular networks in order to deliver solutions to implement any function in a systematic way. Computer science is instrumental in this

process because recognizing that a specific function naturally falls under a particular computational model helps to generate a successful biological design. The derivation is done at different resolutions: an abstract network diagram, a detailed functional description for individual network nodes and ultimately an experimental setup comprising the species, their required concentrations and the laboratory protocols.

From analogical to digital

In the 1820s Charles Babbage began his Analytical Engine, a big steam–age machine based on the automatic loom. It was never fully completed but it was supposed to use metal punch cards and big columns of gears to do calculations. Ever since then it have used not only gears but pneumatics, hydraulics, transistors to make the fundamental units of any such machine: the switches. Later in 1847, George Boole published his book "The mathematical analysis of logic" where successfully demonstrated that logic functions could be represented by algebraic equations. It was not since 1938 when Claude Shannon conclusively proved that Boole's symbolic logic, as it applies to the representation of true/false, can be used to represent the functions of switches in electronic circuits. Being this the foundation for digital electronic design and computer engineering.

In electronic digital systems, a signal is "one" for certain voltage (which depends on the type of electronic logic in use) and for lower voltage is "zero". The assumption that gene network dynamics can be represented in terms of boolean discrete states is of course a strong one. But the truth is that the "switch–like" behaviour, the one of computational devices, is known to be the rule of gene expression rather than the exception. This can be illustrated with a simple system confirmed to be able to behave as a switch: a gene that is expressed and translated into a protein which forms dimers able to bind to the DNA and trigger its own synthesis until is degraded (Fig 6). Non linear kinetic description of this system shows that the ratio transcription/degradation (α/δ) can act as the bifurcation parameter between two stable states of the system. The switch effectively

define binary states with low and high levels of gene expression. The chosen state depend on how other cellular signals tune the transcription or the degradation.

$$\frac{dp}{dt} = \frac{\alpha p^2}{1+p^2} - \delta p$$

The stable states p^* obtained from, $\frac{dp}{dt} = 0$, are:

$$p^* = \frac{1}{2}\left(\frac{\alpha}{\delta} \pm \sqrt{\left(\frac{\alpha}{\delta}\right)^2 - 4}\right)$$

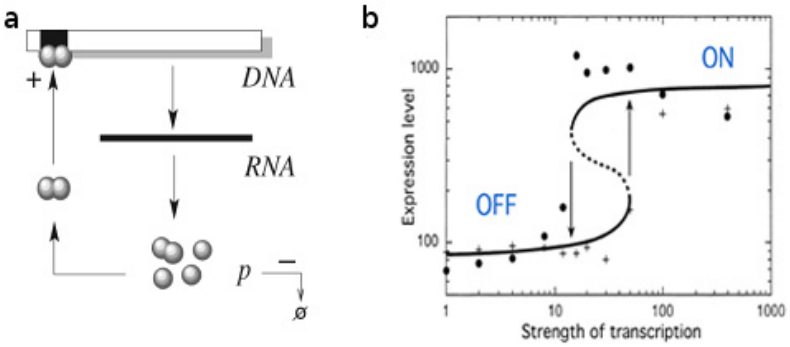


Figure 6: Simple system confirmed to be able to behave as a switch. **a** Scheme of the system: a gene coding for a protein able to dimerize and trigger its own expression. The degradation rate of the protein is also represented. **b** Graphic showing the the relationship between the strength of transcription and the expression levels of the protein. The resulting sharp sigmoidal curve illustrates how analog ranges can represent the on/off digital states.

This simple model illustrates how, although the underlying system is continuous, the nature of the final decision can be understood in terms of digital logic.

Thus, analog ranges represent the digital signals of zeros and ones. In synthetic biology, a transfer function is the relation between the input signal and the output signal of a system's module in steady state. An ideal transfer curve for a synthetic module will exhibit a sharp sigmoidal shape.

Basic Boolean logic

Boolean logic algebra is a mathematical foundation describing rules for input–output functions (truth tables). In the Boolean theoretical framework, state space is described by a set $\Sigma = \{0, 1\}$. A given input string I made of zeros and ones can be written as an element of

$$\Sigma^n = \{0, 1\} \times \overset{n}{\cdots} \times \{0, 1\}$$

Such string, in biology, would correspond to a set of present or absent input molecules. A given Boolean function Φ involving N inputs and one output, is formally indicated as a mapping $\Phi : \Sigma^N \longrightarrow \Sigma^1$

This function defines a input–output mapping between any given binary string of n bits $I \in \{0, 1\}^n$ and the two possible output values $\{0, 1\}$. We can then account that for n inputs 2^{2^n} functions can be performed. There are two particularly relevant subsets of Boolean functions. The set of one–input to one–output functions. And the set of two–input to one–output functions. They contain the minimal operation units, called Logic Gates. These 18 functions are named together with their truth table and icon in Fig 7.

As illustrated in the tables of Fig 7, each gate returns one for different combinations of inputs: `AND` when both inputs are present, `OR` in presence of any or both inputs, `XOR` when one of the inputs is present, `N-IMPLIES` only when one specific input is present. The `NAND`, `NOR`, `NXOR` and `IMPLIES` correspond to the inversion of previous ones. The `NOT` logic correspond to negation of the input. We call `IDN` the identical correspondence between input and output.

Assembling circuits

Complex circuits are build by connecting the output of one logic gate to one or more other logic gates. Multiple architectures and combinations of gates can perform the same complex function. A set of logic operators is considered functionally complete (FCSs) if all possible truth tables

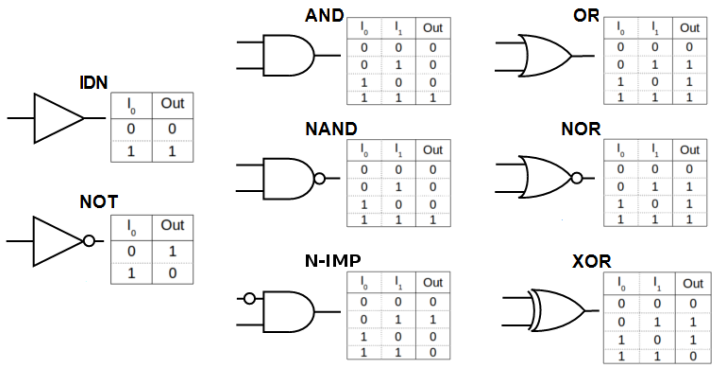


Figure 7: **Truth table, name and icon of Logic gate operators**. Unary and binary boolean function are the minimal unit of boolean logic. Truth table describes the correlation between each possible input combination and its corresponding output. Logic gates are represented by specific icons in electronic circuit diagrams.

can be obtained by combining elements (gates) within the set. Typical examples of FCSs are the pairs (`AND`, `NOT`) and (`OR`, `NOT`) but since the `NAND` and `NOR` gates are obtained from the combination of these previous pairs, it actually occurs that the single–function sets (`NAND`) and (`NOR`) are themselves functionally complete sets [Enderton, 2001].

Boolean canonical forms.

The Sum–of–products canonical algebraic expression, also known as minterm expansion, of any Boolean function can be obtained from its truth table by using an `OR` operator to combine all minterms for which the function is equal to 1. A minterm of a n inputs function denoted as m_i, where $0 \leq i < 2^n$, is a product (`AND`) in which each input (X_i) appears exactly once, non–negated (if $X_i = 1$) or negated (if $\overline{X}_i = 0$) but not both. Therefore, any Boolean function F can be expressed as a sum (`OR`) of its 1_minterms (minterms for which the function $F = 1$). A shorthand notation: $F(x) = \sum(1\_\text{minterms})$ for example (as in Fig 9):

$$F = \overline{a}\,\overline{b}\,c + \overline{a}\,b\,\overline{c} + a\,\overline{b}\,\overline{c} + a\,\overline{b}\,c + a\,b\,c$$

Analogously, the inverse approach consists in: the product–of–sums canonical expression, also known as maxterm expansion, of a n inputs Boolean function can be obtained from its truth table by using an AND operator to combine all maxterms for which the function is equal to 0. A maxterm denoted as M_i, where $0 \leq i < 2^n$, is a sum (OR) in which each input appears negated (if $\overline{X}_i = 1$) or no–negated (if $X_i = 0$). Thus, any Boolean function F can be expressed as a product (AND) of its 0_maxterms (maxterms for which the function $F = 0$). A shorthand notation: $F = \prod(0\_\text{maxterms})$ for example

$$F = (a + b + c) \cdot (a + \overline{b} + \overline{c}) \cdot (\overline{a} + \overline{b} + c)$$

The sum–of–products is implemented as a circuit using AND gates feeding into an OR gate, and the product–of–sums expressions leads to OR gates feeding an AND gate circuits.

Simplification of circuits

Canonical expressions allow for further analysis to simplify the functions, which is of great importance to minimize the digital circuits. These concepts are dual because of their complementary–symmetry relationship as expressed by Morgan's laws:

$$\overline{A + B} = \overline{A} \cdot \overline{B}, \qquad \overline{A \cdot B} = \overline{A} + \overline{B}$$

a

A	B	A·B	A·B	A	B	A+B
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

b

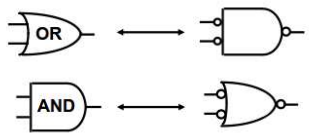


Figure 8: **Proof of Morgan laws**. **a** The table shows the complementary relationship between AND and OR operators. Remember also that $A = \overline{\overline{A}}$. **b** Graphical representation of equivalence between logic gates. The NOT operator is indispensable

Morgan's laws are used to simplify the boolean expressions to build equations only involving one type of gate, generally only using NAND or NOR gates (Fig 8).

Karnaugh maps are a common method to reduce the boolean expressions. Maximal simplification allows to use the lowest number of gates with the lowest number of inputs per gate when implementing a circuit. The boolean equation, the truth table and the Karnaugh map of a function are equivalent. A Karnaugh Map is just another way of presenting a truth table using a grid–like representation. Each position in the grid corresponds to a truth table entry. The map is designed so that any two adjacent cells in the map only differ by a change in one variable.

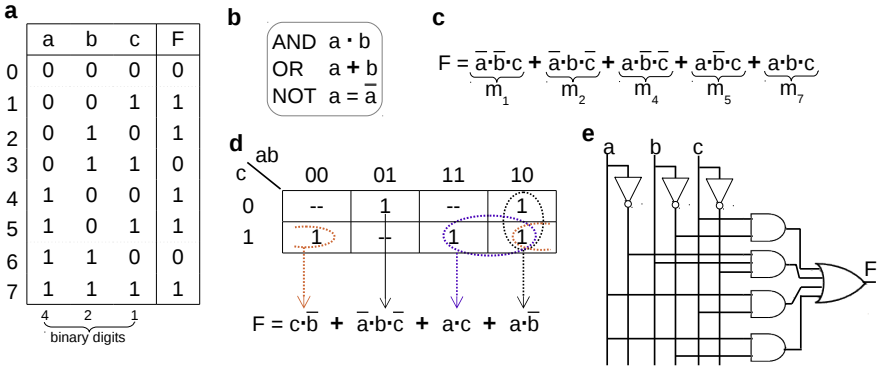


Figure 9: **Exemplification of different representations for a logic function a** Truth table for arbitrary function (F) with three inputs (a,b,c). In the left margin, the encoded decimal numbers, at the botom, the numeration of the binary digits (which increase at power two). **b** Correspondence between the logical operators and their algebraic nomenclature. **c** The Sum–of–products canonical algebraic expression of the F function (used minterms are denoted just below). **d** Karnaugh map of F function. All possible sets of minterms are circled, and the simplified form resulting of each is specified in the final algebraic expression for F. **e** Circuit diagram deduced from the simplified formulation.

Each single cell that contains a 1 represents a 1_minterm in the function. To simplify the function, the adjacent 1_minterms are grouped in sets of size of power of two $(2, 4, 8, \ldots)$. A group of two 1_minterms of n–inputs becomes one single term of the function with $n - 1$ variables, a group of 4 becomes a term with $n - 2$ variables and so on. Thus, larger groups lead to greater simplification. The simplified function is expressed as the sum of this reduced terms (Fig 9d). There may be more than one solution of equal complexity. The Karnaugh map approach can be applied

to boolean functions up to six inputs, but greater functions we will need more sophisticated simplification algorithms such Quine–McCluskey.

Synthetic Biology

Synthetic biology is a research field that combines the investigative nature of biology with the constructive nature of engineering. In practice, synthetic biology consists of selecting molecular elements from natural systems and using them to construct new networks that fulfill specific desired goals. Understanding cells as true programmable entities, allows biologists to develop novel strategies for assembling more and more modular, customizable, scalable and complex systems. The effective designs can be inspired by, but need not mimic, the classic electronic ones. Such new synthetic systems are a promising alternative to create applications in bioremediation, industry and biomedical therapies. While the promise of synthetic biology applications is vast, as well as our access to whole sequenced genomes and proteomic data, we are far from understanding how these components work as a functional complex system. In this sense, synthetic biology is a perfect tool to interrogate the organizational principles of living systems. Quoting the famous physicist Richard P. Feynman "What I cannot create, I do not understand".

Components of synthetic biological systems

Synthetic biologists construct circuits capable of realize functions, using natural biomolecular elements. A important property of synthetic constructs is that, often, is possible to use different types of components to achieve the same behaviour. Different synthetic modules allow control over different levels of the cellular information processing: Transcription, Translation and Post–translation.

Transcriptional control

A gene constitute a module with the input defined by the interaction between promoter and the transcription factors, and the output defined by the identity of the transcribed coding region. Creation of new promoters or modification of pre–existing ones, i.e changing their regulatory sites, provide transcriptional modes to control gene expression to engineer a circuit. Use of chimeric transcriptional modules consisting of eukaryotic coding regions and bacterial regulatory elements [Brent and Ptashne, 1984, 1985], or the other way around as for the yeast–two–hybrid screen [Fields and Song, 1989] precedes the synthetic biology. These early investigations already validated the ability to tightly control gene transcription. Later studies have explored the effects of modular reshuffling of promoter architecture. A combinatorial approach was used to re–arrange various promoter and coding elements into a library of three–node networks, this research constituted an experimental evidence that only few elements are needed to generate a surprising assortment of regulatory diversity [Guet et al., 2002]. Other analysis based on variations in the placement, number and affinity of operator sites of a promoter regulated by two repressors, showed that this simple changes can be used for tuning the genetic network strength and to perform a variety of logic gates for the two inputs [Cox et al., 2007]. Although transcriptional control can be intuitively considered slow, some genetic implementations determine that transcription, actually, is fast and tunable [Stricker et al., 2008] (Fig 10).

Translational control

Translation of RNA has an important regulatory role in the cell. Classical translational control consists of mutating ribosome–binding sites to increase or decrease expression levels of a protein. Molecular engineers are characterizing features from RNA's that, serve to build synthetic systems. An example is a RNA that responds to a signal nucleic acid molecule by base–pairing, is also known as riboregulator. Riboregulators consist of a sensor and an effector domains tailored to differentiate and

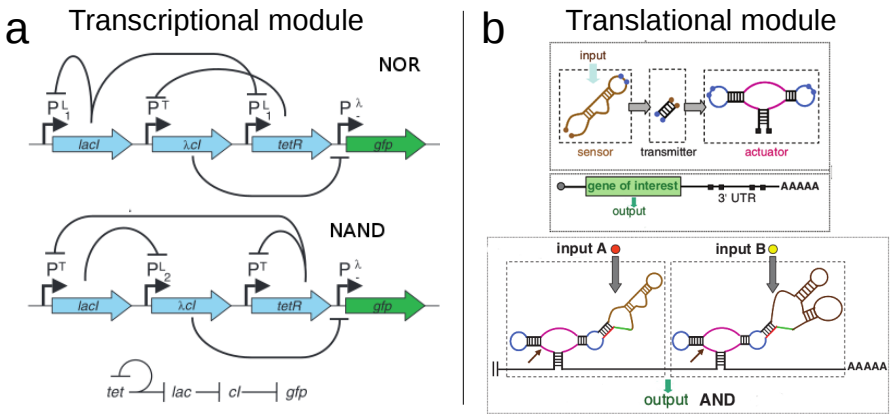


Figure 10: **Modules based on Transcriptional and Translational control. a** Genetic circuits created by combinatorial approach of three well–characterized transcriptional regulators: LacI, TetR, and lambda cI. The circuit’s inputs are IPTG and anhydrotetracycline (aTc).Despite having the same topology, their logical behavior is different. **b** Engineered RNA switch system. The non–coding RNA comprises three domains. The molecular input is received by the sensor domain and transmitted by the transmitter to a regulated activity of the actuator, which in turn controls the translation of a target transcript as an output. Upon binding of the ligand, a conformational change shifts the activity of the actuator to the “ribozyme-active” state that results in self-cleavage. Since the RNA device is coupled to the 3’ untranslated region of the output gene, the ribozyme self-cleavage inactivates the output. Adapted from [Guet et al., 2002] and [Win and Smolke, 2008]

respond to specific nucleotide sequences. Signal–responsive structures are usually introduced into the 5’region of the RNA molecules. The riboregulator activity is highly dependent of its secondary structure. When the sensor domain is a structured nucleotide pocket, or aptamer domain site, that binds to small molecules instead of complementary RNA/DNA strand, the riboregulator is renamed as riboswitch. All sets of riboregulators as microRNAs, interfering (si)RNAs, ribozymes, etc. . . rely on sequence specific binding to control, degrade or titrate pre–existing mRNA and thus, post–transcriptional gene expression [Uhlmann and Peyman, 1990; Erdmann et al., 2001].

The engineered RNA–switches and Ribozymes in *Saccharomyces cerevisiae* done by Smolke and co–workers are references in the field. They designed two opposite RNA–switches. The “on antiswitch” close off its antisense domain in presence of ligand, allowing the translation of the target mRNA. The “off antiswitch” conformation sequesters its

antisense domain, and when ligand is added, the antisense sequence becomes available avoiding the translation of the transcript. As expected, they found that variants with less complementarity within the antisense domain exhibited lower switching thresholds as a function of ligand concentration [Bayer and Smolke, 2005]. The authors extended the RNA–switches competences adding a hammerhead controllable ribozyme catalytic ability to cleave a target nucleic sequence. The ribozyme RNA molecule is activated or inactivated depending on the change of the secondary structure induced by hybridizing a signal ligand and its RNA cognate sequence. These novel molecules can implement several two input logic functions including `AND`, `NAND`, `NOR`, and `OR` gates [Win and Smolke, 2008]. The increasing number of RNA synthetic systems demonstrate its suitability for being customized to either repress or activate the expression of any target gene in response to environment molecules [Isaacs et al., 2004; Isaacs FJ, 2005] (Fig 10).

Post–translational protein control

Proteins are key elements in the signalling networks that mediate the processing of external signals. Different protein interaction motifs determine the topology of a given network and the dynamics of the response.

In turn, proteins are made up of multiple, independently–folding domains. A part of the domain which carries out the catalytic function (e.g a kinase domain which transfers a phosphate), also contain one or more protein–protein interaction domains. These regulatory domains lead the connectivity of signalling pathways, linking upstream with downstream targets (e.g a binding domain that localize a signalling protein to a specific subcellular region). Thus, in a signalling module the inputs are defined by the interaction of the regulatory domains with the external signals or their interaction partners, while output is defined by the activity of the catalytic domains. Manipulation by domain shuffling or pathway rewiring, can be used to dramatically change the input/output relationship of a signalling response.

Receptors are the link to the outside information, the starting point of signal transmission, and therefore a key element to redesign signalling networks. One way to engineer a new pathway is `deviate the native input to a novel output`. A simple example of this strategy are the G protein–coupled receptors (GPCrs) fused to a desired transcription factor by a tEv cleavage site. When the engineered GPCr is activated by its endogenous ligand it recruits the synthetic hybrid β–arresin–tEv protease liberating the transcription factor [Barnea et al., 2008].

An alternative building approach is `redirect the native output to novel input` such as small molecules, antigens or physical signals. One example of each approach are the rAssLs receptors, the CARs and the light–controlled receptors. The rAssLs, receptors activated solely by synthetic ligands, are mutated opioid GPCrs receptors that can not bind their proper ligand. Noteworthy, expression of rAssLs allowed control of heart rate by the administration of spiradoline [Conklin et al., 2008; Redfern et al., 1999]. The CARs (chimeric antigen receptors) consist in an engineered single–chain antibody fused to the intracellular region of the T cell receptor (CD3ζ chain) through its transmembrane domain [Gross et al., 1989; Sadelain et al., 2009]. The light–controlled receptors for mammalian cells had been adapted from ion channels of microbes or plant proteins (such the dimerization domain of phytochrome [Levskaya et al., 2009] or the plant LOV motif [Wu et al., 2009]. This increasing collection of light sensors represent an outbreak of the optogenetics field. This type of control is essential when studying processes that proceed at the rate of diffusion, and should be especially useful for interrogating localized events (Fig 11a).

Not surprisingly many of the efforts to engineer new signaling behaviors exploit strategies of `recombining modular functional protein domains` in novel ways, in effect harnessing an evolutionary strategy to engineer new functions [Peisajovich et al., 2010]. Actually this is the basis of the well known two–hybrid technique to detect protein–protein interactions. As a reminder, two fusions are constructed. One

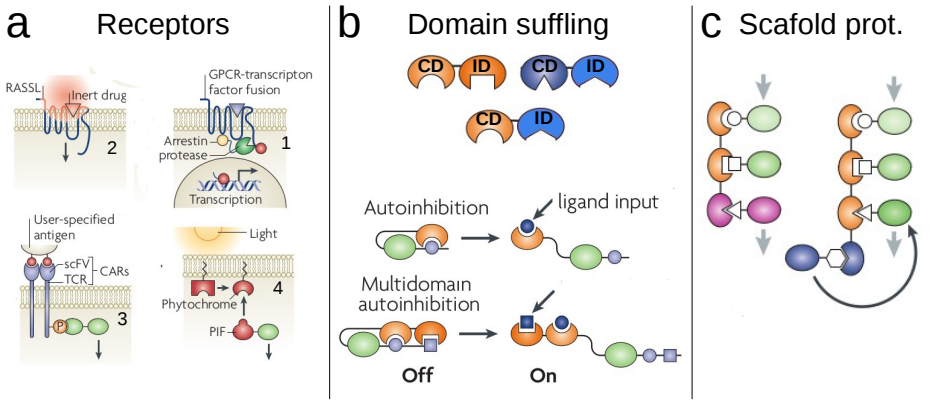


Figure 11: **Module designs for post–translational control. a** Receptor modifications. Redirect native inputs to novel outputs (GPCR fused to transcription factor domain by TEV protease site, 1). Engineering novel inputs as small molecule agonists (RASSL, 2), specific antigens (CARs, 3) or even light (using phytochrome proteins from plants, 4). **b** Modular eukaryotic signalling proteins are generally composed of catalytic (CD) and interaction domains (ID). The catalytic activity can be re–directed to specific substrates or locations by fusion with different ID. Alternatively, ID can allosterically regulate CD by engaging intramolecular autoinhibitory interactions. Switch proteins can be activated by competing ligands that relieve autoinhibition. **c** Pathways can be redirected through engineered chimeric scaffolds to assemble a novel combination of kinases. New interaction sites can also be added to scaffolds to recruit additional modulatory factors. These additional factors can build synthetic feedback loops to display diverse signalling dynamics. Adapted from [Lim, 2010].

protein of interest with the DNA Binding Domain (i.e. Gal4–DBD or LexA–DBD) and the other protein with the Activation Domain (i.e. Gal4–AD or VP16–AD). The first hybrid is localized at its specific DNA loci (i.e P*GAL* or LexAoperons). When the two considered proteins interact, the AD trigger the reporter gene transcription. The same principle brings to bear when tagging proteins. Protein tags are peptide sequences genetically grafted onto a recombinant protein, to re–localize, degrade or provide a visual readout on a protein.

Back to synthetic biology itself, there has been successful attempts to create more sophisticated structures. For example the chimaerical allosteric switches inspired in the N–WASP. Its output domain is constitutively repressed by an autoinhibitory interaction involving other two domains (GBD/B). Input ligands active the protein by disrupting

this auto–interaction. New input switch hybrid proteins were designed replacing the inhibitory domains by heterologous domain and ligand pair (i.e. PDZ). Following this principle, a small library of synthetic switches was created. The novel protein set was able to recapitulate the native AND and OR logic behavior [Dueber et al., 2003].

Scaffold proteins determine the wiring linkages of signaling proteins and control when or where signaling happens. Hence, attaching binding pockets of two different scaffold proteins results in crosstalk of its pathways [Park et al., 2003]. Furthermore, this signal–processing hubs are ideal targets for synthetic feedback loops to reshape the response's amplitude and timing. Which was shown using the Ste5, the scaffold protein of the yeast mating pathway, act as a platform to dynamic synthetic feedback loops. A basic zipper was fused to Ste5, and the complementary acidic zipper was fused to positive, Ste50, or negative modulators, Msg5. These engineered circuits reshaped the otherwise graded, linear mating pathway response into various non–linear behaviors like acceleration, pulse generation, delayed response times, and tunable adaptation [Bashor et al., 2008] (Fig 11c).

Engineered biological modules

Pheromone pathways

Saccharomyces cerevisiae exists in two haploid cell types: MATa and Matα. MATa cells secrete a–factor and MATα cells secrete α–factor. When a yeast cell is stimulated by pheromone secreted by a nearby cell of the opposite mating type, it undergoes a vast physiological changes in preparation for mating. These include variation in gene expression of about 200 genes, arrest in G1 phase of the cell–cycle, oriented growth toward the mating partner (shmoo) and the fusion of the plasma membranes of the mating partners followed shortly thereafter by the fusion of their nuclei. The final result is a diploid a/α yeast type. α–factor is an unmodified 13 residue peptide encoded by *MF(ALPHA)1* and *MF(ALPHA)2* genes, whereas a–factor is a C–terminally farnesylated

12 residue peptide encoded by *MFA1* and *MFA2* genes. But both yeast pheromones are initially synthesized as larger polipeptide precursors composed by an N–terminal secretory signal and four tandem repeats of the pheromone peptides. Several amino and carboxy peptidases (i.e Kex2 for α–factor) process the precursor along the secretory pathway, and the mature small peptides are released from the cell [Michaelis and Herskowitz, 1988; Achstetter and Wolf, 1985].

The mating response is controlled by the MAP kinase signalling cascade of the pheromone pathway. Each mating type expresses a different seven transmembrane G protein coupled receptor (GPCR), Ste3 of Matα cells senses a–factor and Ste2 of MATa cells senses α–factor. The extracellular loops 1 and 3 of Ste2 are responsible for alpha factor binding, the cytoplasmatic loop 3 and the C–terminal domain are important for signalling through G protein [Chirstopher and Kendall, 1994; Byung Kwon et al., 2001] (Fig 12a).

Both pheromone receptors are coupled to a common heterotrimeric G protein, Gpa1–Ste4–Ste18, where Gpa1 is the Gα subunit and Ste4–Ste18 form the G protein $\beta\gamma$ subunit. Pheromone stimulation induces GPD to GTP exchange in Gpa1, and thus, G protein is dissociated. Then, Ste4–Ste18 complex activates the Cdc42 GTP exchange factor (GEF), that recruits Ste5 and Ste20. Ste20 transmits the signal to the first protein of the cascade: MAPKKK Ste11. Ste5 is a scaffold protein, that tethers together the whole MAPK cascade. Ste11 phosphorylates the MAPKK Ste7 which in turn phosphorylates and activates the MAPK Fus3. Once activated, Fus3 orchestrates a wide variety of cellular processes. It phosphorylates and therefore activates the G1 cyclin CDK inhibitor Far1 that results in a cell cycle arrest at G1 [Peter et al., 1993]. Fus3 also phosphorylates and inhibits Dig1 and Dig2 and thus, liberates Ste12 transcription factor from Dig1–2 repression. Ste12 promotes transcription of pheromone dependent genes like Fus1 [van Drogen et al., 2001]. Pheromone stimulation also causes a transient phosphorylation of the filamentous growth pathway MAPK Kss1, but to ensure signalling specificity, Fus3 inhibits induced gene expression of Kss1. In fact, Fus1

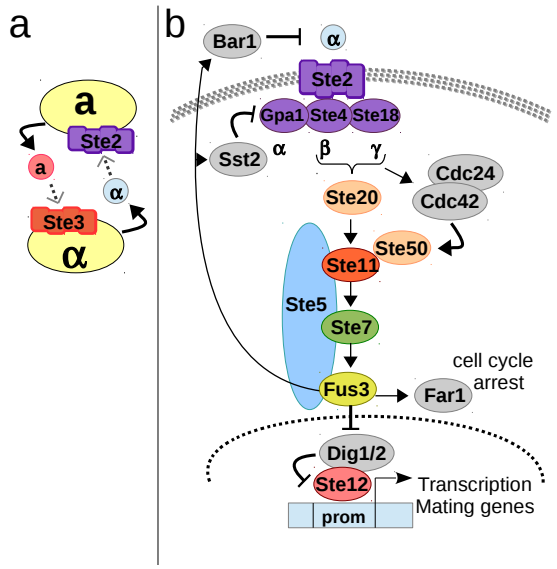


Figure 12: **Schematic representation of mating types and the pheromone pathway. a** MATa cells express a–factor, and Ste2 receptor to sense α–factor. MATα cells produce α–factor and Ste3 receptor to sense a–factor. **b** Alpha pheromone activate the G–protein coupled receptor Ste2 which lead to the activation of the MAPKKK cascade (Ste11–Ste7–Fus3). Active Fus3 Kinase causes cell cycle arrest and changes gene expression. Fus3 dependent desensitisation of the receptor, downregulates the pathway.

and Kss1 normally act separately, nonetheless, either of them can replace the other [Elion et al., 1991].

In the pheromone pathway, as in many others, negative feedback loops operate to promote adaptation and recovery. A extracellular pepsin–like protease that degrades alpha–factor, Bar1, is secreted by MATa cells in response to pheromone. Moreover, the pheromone–bound receptor is phosphorylated by Fus3, mono–ubiquitinated and then endocytosed [Feng and Davis, 2000]. Finally Sst2 , a regulator of G protein singaling (RGS) accelerates the GTP hydrolysis of Gα subunit, by at least 20–fold. Sst2 expression is induced by pheromone and its stability is enhanced by Fus3 phosphorylation [Garrison et al., 1999; Yu et al., 2008]. Both, Bar1 and Sst2 deletions had been used to increase pheromone sensitivity [Janiak et al., 2005] (Fig 12b).

The pheromone pathway is well conserved among yeast species, however pheromones and receptors have evolved to ensure specificity and avoid mating with heterologous species [Fitzpatrick et al., 2006; Scannell et al., 2007]. This conservation has been shown by expression of the *Candida albicans* alpha factor receptor, CaSte2, in *Saccharomyces cerevisiae*. Upon addition of *C.albicans* alpha factor, the foreign receptor is able to trigger the transcription of Fus1. The functionality of CaSte2 in *S. cerevisiae* suggests that conformational changes by hormone stimulation and signalling capacity to heterotrimeric G–protein are well conserved [Janiak et al., 2005].

The galactose regulon

Saccharomyces cerevisiae galactose regulon genes, named *GAL* genes, are regulated at the level of transcription in a carbon source–dependent manner [Lohr et al., 1995]. Three different transcriptional states are possible for *GAL* genes: not expressed (when are growing in a non–fermentable carbon source as ethanol or raffinose), highly induced (upon addition of galactose) and strongly repressed (in the presence of glucose) [Johnston, 1999]. The mechanism of glucose repression and galactose induction of *GAL* genes is one of the best characterized eukaryote systems of transcriptional regulation. The set of *GAL* genes encode: a galactose permease (*GAL2*), metabolic enzymes to transform galactose to glucose 1–phosphate (*GAL1, GAL7* and *GAL10*) and transcriptional regulatory proteins (*GAL3*, *GAL4* and *GAL80*). In essence, the functioning of the system is: The DNA–binding transcriptional activator, Gal4p, remains bound to upstream activating sequences (UAS) in the GAL promoters region. However, Gal80p interacts with Gal4p avoiting the transcription of GAL genes [Lohr et al., 1995]. In response to galactose, Gal3p inhibits the Gal80p repression, allowing Gal4p to activate the transcripion of GAL genes. Glucose mediated repression of Gal genes depends on two different mechanisms: strong glucose dependent P*GAL4* downregulation and binding of the glucose activated repressor: Mig1 to the *GAL* promoters (Fig 13a).

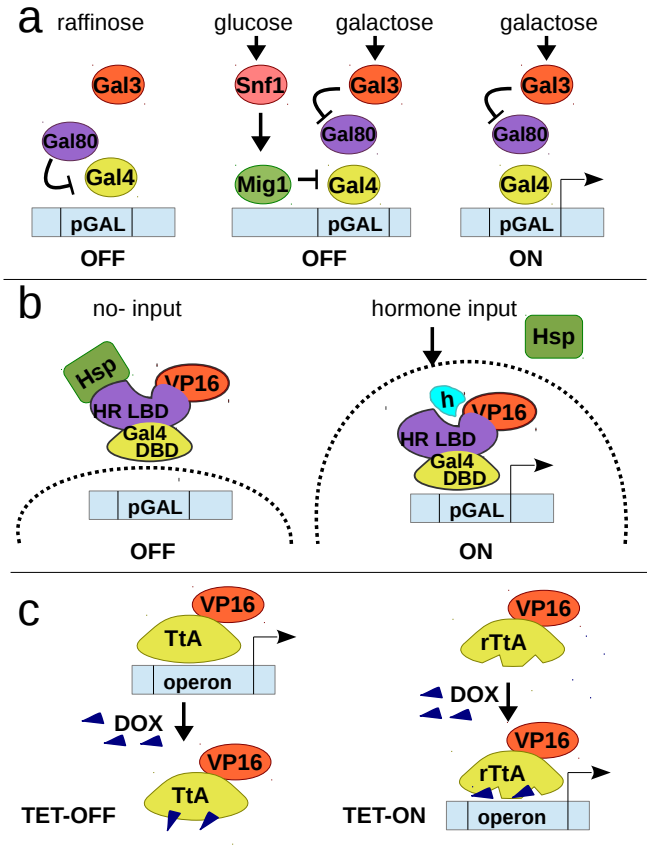


Figure 13: **Schematic representations of inducible systems. a** Regulation of *GAL* genes. In presence of raffinose Gal80 represses transcription by regulation of Gal4. Addition of galactose promotes inhibition of Gal80 by Gal3 and thus, the *GAL* genes are expressed. Glucose causes repression of *GAL* genes by Mig1. **b** Transcription induction of *GAL* genes in yeast cells by human hormone. The hybrid protein composed by the Gal4 DNA binding domain (Gal4DBD), the ligand binding domain of hormone receptor(HR LBD) and the VP16 transactivator domain is sequestered at cytoplasm by Hsp chaperones. Addition of hormone releases the chaperone and induces a the nuclear translocation of the hybrid protein, the Gal4DBD binds to of P*GAL* promotor and VP16 triggers the transcription of *GAL* genes. **c** Eukaryote Tet system. The TtA protein changes its DNA binding affinity in response to Doxyciclyne. TtA–VP16 protein is inhibited by DOX whereas rTtA–VP16 version is activated with DOX.

Human hormones used in yeast

A novel hybrid synthetic protein was developed to regulate *GAL* genes with 17–b–Estradiol [Louvion et al., 1993]. The chimaeric protein

composed of the Gal4 DNA binding domain, the ligand binding domain of estradiol human receptor (hER) and the activatting domain of VP16 virion protein is constitutively expressed (i.e under *ADH1* promoter). In basal conditions, this protein is sequestered in the cytosol by chaperones through the estradiol receptor domain. Addition of 17–b–estradiol into the medium, releases chaperons from the ER domain and the Gal4DBD–ER–VP16 protein translocates in to the nucleus (Fig 13b) [Takahashi and Pryciak, 2008].

In this thesis we developed other systems, based on this, that permitted to control gene expression with other human hormones. New protein fusions consist of the ligand binding domain of Progesterone receptor (hPR, that comprises from aminoacid 655 to 933) and the ligand binding domain of Aldosterone (hAR, from aminoacid 705 to 984). The structural information of these receptors are available from the Uniprot database (`http://www.uniprot.org`).

Interestingly, to create a yeast strain that responds to dexamethasone no chimaeric protein was needed. The whole version of the human Glucocorticoid receptor (hGR) was expressed under a constitutive promoter (*PGPD1*). The hGR, remains in the cytosol in absence of dexamethasone, but when the hormone is added, it translocates in to the nucleus, and binds to HERE sequences. The receptor was able to fire the transcription of a minimal promoter. To prevent leakines, a reduced version of *GAL1* promoter containing the Mig2 UAS was used as minimal promoter (minp) [Miller et al., 2010].

Use of *E. coli* proteins in yeast

The prokaryote tetra–cycline resistance operon of bacteria, also called **Tet system**, consist of two elements: the tetracycline–controlled transactivator (tTA) protein and the responsive Tet operators (tetO) [Gossen and Bujard, 1992]. Tetracycline and its derivatives (i.e doxycycline) modify the tTA domain to prevent the DNA binding activity of it. The Tet system for eukaryote organisms was derived from bacteria.

When fused to the virion protein VP16, tTA domain is able to trigger transcription of *tetO* promoters, and addition of doxycycline results in rapid transcription repression (TetOFF). To achieve a system with dual behaviour ON–OFF, a reverse transactivator (rtTA) was developed by [Gossen et al., 1995]. The rtTA is able to bind the *tetO* only in presence of doxycyline. Thus, the rtTa–VP16 fusion promotes expression of genes controlled by tetO containing promoters in response to doxycylcine (TetON), (Fig 13c).

An alternative "OFF" system can be implemented in yeast using the *E.coli* **LacI** transcriptional repressor protein (the mammalian–enhanced version of it is also functional in yeast cells) and a constitutive promoter but with addition of the *lacI* operons at 3–prime (i.e P*ADH1*i), that becomes a LacI repressible promoter [Cronin and Mansfield, 2001; Wyborski et al., 1996; Grilly et al., 2007].

Another useful system transferred from bacteria to yeast, is the cytoplasmic **ClpXP proteasome** to control protein degradation. The *E. coli* genes *clpP* and *clpX* (modified with 10 silent mutations) are properly expressed and become functional proteins in yeast [Gottesman et al., 1998; Karzai et al., 2000]. Proteins tagged with ssrA sequence (AANDENYALAA) to their C–terminus, are specifically recognized by the ClpXP proteasome and degraded efficiently. For example in yeast cells expressing yeGFP–ssrA, the induction of ClpXP protease reduces the half–life of yeGFP–ssrA about 6 times.

Relevant Biological circuits in Synthetic Biology

Synthetic biology approaches have been used to explore the construction of systems performing a broad range of functions, including among them: oscillators, bistable memory switches, logic gates operations, population control or multicellular patterning. This impressive array of efforts has shown that it is possible to build minimal systems that recapitulate a considerable catalogue of dynamic and spatial computations.

Switches and Oscillators: The One Bit Memory Core

Time–dependant or sequential Boolean logic devices have been implemented in living cells, starting with a toggle switch by Gardner et al. [Gardner et al., 2000] and a synthetic oscillator by Elowitz and Leibler [Elowitz and Leibler, 2000], both in *E. coli*. In fact, in one decade have been synthesized or proposed numerous biological memory devices.

Represilator: Elowitz and Leibler [2000] synthesized the first genetic oscillator, dubbed repressilator. It was composed by three transcriptional repressor genes (lacI, tetR and λcl) and an additional reporter gene (GFP). Each of the three genes represses the next gene in a loop, with the last gene repressing the first one. The repressilator is not a bi–stable switch but rather a self–maintaining oscillator that proceeds from one state to the next, autonomously [Elowitz and Leibler, 2000]. This original oscillator, though represented a major milestone for synthetic biology, showed relatively poor performance with damped oscillations persisting for no more than three periods (Fig 14a). Since then, many researchers have improved this circuit yielding to robust waves with either tunable amplitude or frequency, and yet being minimal designs as three dual positive–negative feedbacks [Stricker et al., 2008]. Interestingly, even though the former architectures were built from a purely transcriptional network, different molecular implementations have been used as antisense RNA to mediate feedback in mammalian cells [Tigges et al., 2009] or a combination of transcriptional feedback with the enzymatic interconversion of a metabolite pool [Fung et al., 2005].

Toggle Switch: The construction of a genetic toggle switch, a synthetic bistable transcriptionally controlled network, provided the first practical proof for a simple theory that predicts the conditions necessary for bistability. The toggle consists in two repressible promoters arranged in a mutually inhibitory network. The LacI repressor is expressed under the control of the *tetR* repressible promoter, PLtetO–1, meanwhile the TetR repressor is controlled by the *lacI* repressible promoter, Ptrc–2. It flips between stable states using transient chemical induction of IPTG

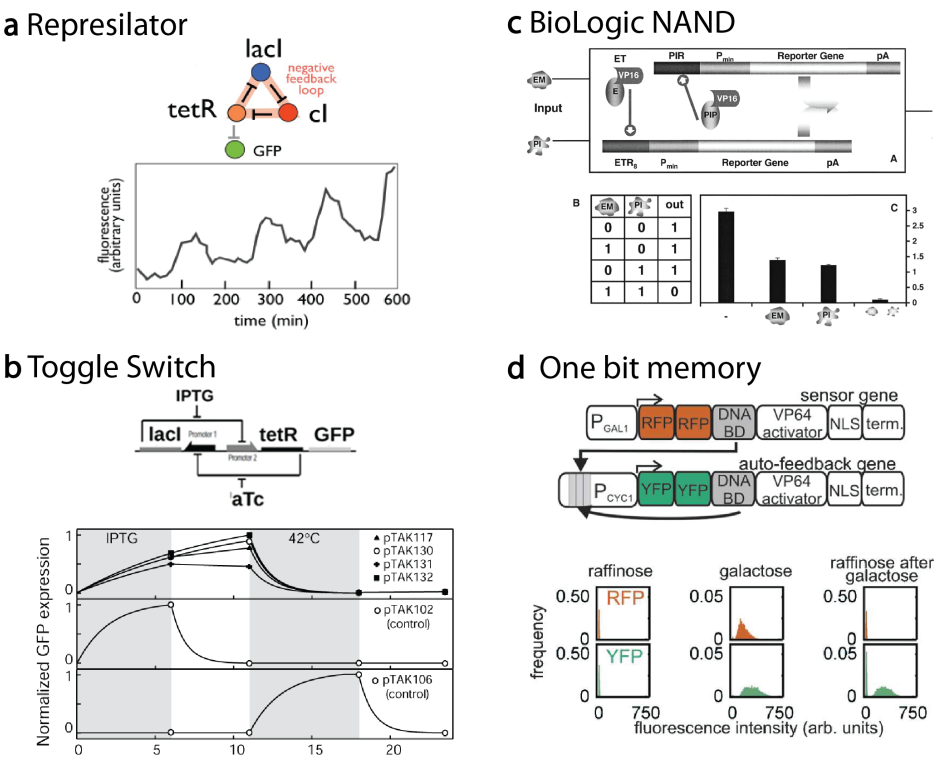


Figure 14: **Schematic architecture and graphical output of: Represilator,Toggle switch and One bit memory circuits.** **a** Circular inhibitory structure: TetR represses LacI production, that represses λCl production which closes the cycle repressing TetR production and derrepresing GFP output response. Graphic of florescence oscillations over time. **b** Genetic switch topology. In no–GFP–state: lacI is transcribed and tetR is repressed. In GFP–state *tetR* and GFP are transcribed and *lacI* is repressed. The graphics's top row represent the experimental results. The bottom rows are controls. Cells are set to GFP–state by exposure to IPTG and remain "on" after removal of inducer. When exposed to aTc, they toggle into the NO–GFP–state and remain "off" even washing the input. **c** Schematic representation of the logic trascriptional circuit. This device perform a NAND logic. In presence of both inputs (pristinamycin, erythromycin); both transactivators are dissociated from the DNA and there is no transcription of florescence. **d** Schematic representation of the one bit memory trascriptional auto–feedback. This device stores three steady states of florescence: never exposed to stimulus (no emission), stimulus present (red and yellow), and previously exposed to stimulus (only yellow). Adapted from Elowitz and Leibler [2000]; Gardner et al. [2000]; Kramer et al. [2004]; Ajo-Franklin et al. [2007].

or anhydrotetracycline (aTc), which inhibit the LacI or TetR repressor respectively [Gardner et al., 2000]. This pioneer design of a toggle switch exhibits a nearly ideal switching threshold, however, it requires two inputs

to operate (Fig 14b). Modified versions of Gardner's toggle switch, that can sense one input alone had been proposed. These modules are, by default, in a particular stable state and flip upon the introduction of the DNA damage agent, so does not need two inputs [Kobayashi et al., 2004]. The simplicity of the design by itself does not allow two truly stable states if not joined up with another module as a `NOR` gate [Lou et al., 2010]. Moreover despite the many works on genetic switches (also called flip–flops), all published designs until today work asynchronously. Probably, due to the great difficulty entailed in creating a sustained oscillating culture, a synchronous clock, that additionally coordinates the switch, and thus there had only been early crude attempts to do it [Atkinson et al., 2003].

As a practical device, toggle switch forms a synthetic, addressable cellular memory unit which is able to maintain the state even in the absence of stimulus. This challenge was approached in yeast cells [Ajo-Franklin et al., 2007].

In this memory device, a signal (galactose) induces the synthesis of a synthetic transcription factor (RFP–DBD$_{PCYC1}$–VP16) which triggers the expression of other synthetic transcription factor (YFP–DBD$_{PCYC1}$–VP16), that in turn, triggers its own expression in an "autofeedback". This synthetic network can exist in three different steady states: never exposed to stimulus (`off`), stimulus present (`on/RFP and YFP`), and previously exposed to stimulus (`memory/YFP`). The production rate of the transcription factor, and its autofeedback, must be balanced by its decay rate.

In the off stable state, the low basal levels of RFP–DBD$_{PCYC1}$–VP16 never produce de YFP output due to the comparatively high rate of degradation. In the memory state, a higher enough concentration of YFP–DBD$_{PCYC1}$–VP16 balances the decay rate and perpetuates its own production. Intermediate concentrations, lead to an unstable switch between both stable states. Therefore the growth rate of the yeast cells becomes important, since dilution of the transcription activator dictates its decay rate. (Fig 14d).

Several switch behavior devices can be build with distinct molecular substrates, which can involve the use of the already mentioned RNA riboswitches [Bayer and Smolke, 2005], switchable proteins [Dueber et al., 2003] or even DNA invertases. Placing the promoter of the reporter gene between two specific elements, the flipase (e.g FimE) is able to invert the DNA region between these two elements. It is worth noting that, as changes to the DNA are permanent, this method of defining state is inherited by the offspring [Ham et al., 2006, 2008]. This demonstrates the generalizability of design principles.

Logic Gates

In vivo cellular computing started in 1998 with *E.coli* by implementing 1–input and 2–input combinatorial Boolean logic gates: `NOT`, `AND` and `IMPLIES` [Knight Jr. and Sussman, 1998; Weiss et al., 1998]. This was latter implemented in eukaryotic cells.

The new synthetic regulatory networks were based in binary chimeric transcription modulators assembled by fusing mammalian transactivator (VP16) or transrepressor (KRAB) domains to procaryotic response regulators (PIP, E) which bind to a specific operator–containing promoters (PIR_3Op, ETR_8Op) in a input adjustable manner (pristinamycin, erythromycin). This modular building blocks were arranged in distinct combinations to perform a variety of logic responses. Cells that contain the repressors E–KRAB and PIP–KRAB and their cognate promoters P–ETR_8Op and P–PIR_3Op as two separate reporter expression units perform a `OR` gate. Identical architecture but replacing the repressor domain KREB by the activator domain VP16 compute a `NAND` logic. This same transactivator elements but with the promoters placed as a tandem results in a `NOR` function. Use of tTA / $tetO_7$ regulator unit allowed to compute three input functions [Kramer et al., 2004] (Fig 14c).

RNA based logic circuits appeared later, but potentially, either the ribozyme switches [Win and Smolke, 2008] or the i–RNA [Rinaudo et al., 2007] available designs, are able to perform any logic statement.

Coordinated behaviour of cellular consortia

Predictable and robust performance must deal with variation in phenotypes between cells. Synthetic biology studies, have demonstrated the feasibility of building gene circuits that act with a digital logic but also revealed major hurdles to achieving reliable permanent circuit performance due to noise in cellular processes and cell–to–cell divergence. To addresses these issues a coordinate population behaviour is needed.

Population Control: A milestone work of [You et al., 2004] demonstrated that coupling gene expression to cell death using cell–cell communication can control and stabilize cell density in a population despite variability of individual cells. To do that, the authors engineered *E. coli* cells able to express LuxI and LuxR upon induction. The LuxI, a protein from the well–characterized quorum sensing system from the marine bacterium *Vibrio fischeri*, synthesizes a diffusible molecule: acyl–homoserine lactone. The AHL accumulates in the experimental medium and inside the cells as the cell density increases. At sufficiently high concentrations, it binds and activates the LuxR transcriptional regulator, which in turn binds to *luxI* promoter and induces the expression of a killer gene (*ccdB*) under its control. The induced circuit culture would reach a stable cell density, although it might go through damped oscillations while approaching the steady state. A level of the killer protein lower than one for the steady state allows the population to grow; conversely, its excessive production decreases cell density. After some delay, the decline in cell density leads to a decrease in AHL concentration, which in turn leads to reduced levels of the killer protein, allowing the population to recover. The steady–state cell density would increase nearly proportionally with the AHL degradation rate. Thus, AHL serves as an external 'dial' to operate the circuit. Curiously, for the proposed model, cell heterogeneity in terms of size, age, gene expression (accordingly the sensibility to the killer protein) is an advantage. If all cells had the same phenotype, the circuit would fail to achieve a stable cell density, since all the population would become extinct once the killer

protein concentration reached a critical threshold (Fig 15a). To note, this concept of communication–regulated growth and death can be extended to engineering synthetic ecosystems [Balagadde et al., 2008; Weber et al., 2007; Kobayashi et al., 2004; Basu et al., 2004; Brenner et al., 2007; Waite and Shou, 2012].

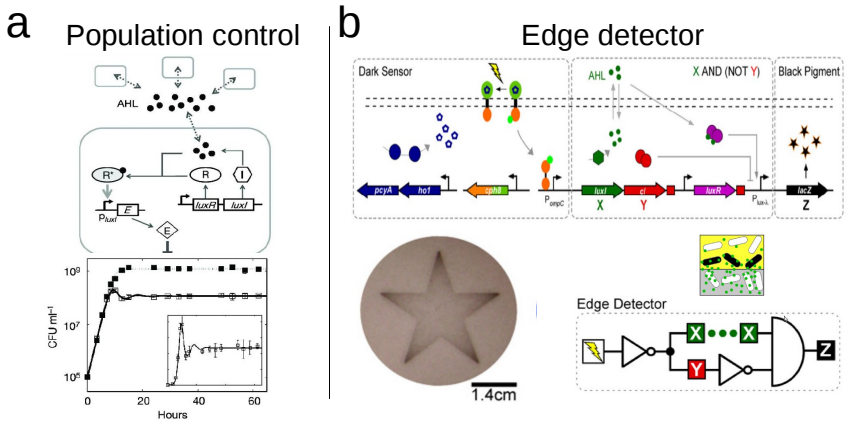


Figure 15: **Schematic diagrams and graphical output of population circuits. a** Circuit to control culture density. The cells transcribe luxR and luxI. luxI produces the AHL lactone. High concentrations of AHL actives luxR, that tanscribes the killer gene ccdB. Experimentally measured growth curves with the population–control circuit OFF (filled squares) and ON (open squares). **b** Edge detector circuit. After the light is applied cells identify light–dark edges and respond with the synthesis of a dark pigment. The circuit essentially executes two instructions: "`IF NOT` light, produce signal" and "`IF` signal `AND NOT` (`NOT` light) produce pigment". The circuit constitutively express genes ho1, pcyA, Cph8–EnvZ and luxR. Ho1 and pcyA are chromophore producers. In the dark, the photoreceptor Cph8, is un–inhibited and the fused EnvZ kinase domain, triggers PompC transcription through OmpR. Then, cI and luxI (thus, AHL lactone) are expressed polycistronically. The produced AHL, spreads locally and affects neighboring cells. It binds to the transcription factor LuxR activating the expression of lacZ under the promoter Plux–l whereas CI dominantly represses this promoter. β–galactosidase, the product of lacZ, cleaves a substrate in the media to produce black pigment. Adapted from [You et al., 2004; Tabor et al., 2009].

Multicellular Patterning Formation: Once individual genetic circuits that mimic basic electronic functions have been constructed, the current challenge is to assemble multiple circuits to engineer more sophisticated responses, such multicellular pattern formation. The regulatory networks that drive multicellular pattern formation combine logic and cell–cell communication allowing each cell in a population to respond appropriately to local signals without information regarding its position

or the global state of the system [Basu et al., 2005]. The image edge detector problem is an example of highly complex computation. In silico algorithms calculate the edges addressing each pixel in serie with a computation time that increases linearly with the number of pixels. In a bacterial culture, calculation is massively parallel and the computation time become independent of the image size. Trabor and Voight group programmed a isogenic layer of *E. coli* in a petri–dish able to sense the image light, communicate and compute to identify the light–dark edges, and visually present the result of the computation. This engineered *E. coli* strain constitutively expresses the heterogenic LuxR receptor and a light–sensor fused to the EnvZ kinase domain. The LuxI lactone (AHL) and CI expression are under the control of *ompC* promoter, regulated by the Ompr (phosphorylated by EnvZ). The promoter pLux–λ, that is activated by the LuxR–AHL complex and dominantly repressed by CI, controls the expression of LacZ (to produce black pigment). The engineered light sensor follows a `NOT` logic. In the dark, cells trigger the production of a diffusible signal (AHL) and a intracellular repressor (CI) that avoids pigment expression. Cells in the light area, do not produce the signal but can respond to it (with LuxR) producing the pigment due to absence of repressor (`N-IMPLY` logic). Thus only those cells that are in the light but proximal to dark areas, where the signal has been spread, activate the output which results in the enzymatic production of a black pigment [Tabor et al., 2009]. This elegant construct illustrates how a logic design can be translated into a set of linked genetic gates.

Modularity

One of the most fundamental issues in biology is how complex organization is established. A system is considered modular if the parts that comprise it (modules) can be rearranged and retain its function in a context–independent manner. Modularity of biological systems is undeniable [Simon, 1962], and opportunely, viewing a biological being in terms of a hierarchy of interlinked, functional modules is one useful way to parse their complex behaviours into parts that are more easily

understandable. At the beginning, synthetic biology was focused on creating and perfecting small genetic structures and minimal modules but today efforts are directed to combine these small parts and modules to create more intricate and sophisticated biological devices. For this reason, there is a general attempt in the field to characterize and standardize the small modules by measuring, manipulating and matching their input–output thresholds. If a component is a functional module, then it should retain its native functionality when placed in a modular synthetic system. Features required from an effective molecular module include: the existence of a robust digital regime, scalability (the capacity to receive an increasing number of inputs without dramatic design alterations) and composability (the capacity to operate together with other modules in parallel and or in cascades). Then, the ultimate goal is to create a catalogue of interchangeable parts that can be easily mixed and matched for circuit construction.

In contrast to engineering practices, nature of biology makes standardiza–tion in synthetic biology much more difficult due to cell death, crosstalk, mutations, intracellular and extracellular conditions, noise and so forth. As the number of device's modules grows with the complexity of a desired behaviour, it becomes increasingly difficult to coordinate. Unfortunately, the demand for a scalable implementation of complex devices is difficult to meet [Purnick and Weiss, 2009]. The main difficulty emerges with the fact that: in electronics, every wire is defined in terms of a conducting piece of material, which is always the same. In the cytosol or in culture media, where molecules share the same medium, the spatial insulation of electronic wires is no longer satisfied. In a biological device, each wire needs to be implemented by using a different molecular particle to avoid misleading connections. An enlarged complexity of computation involves an increase in the number of inputs and outputs, that inevitably correlates with a growing need for different specific diffusible molecule to communicate these modules without crosstalk. Ergo, the wiring problem is the major limit for combinatorial system scalability.

The spatial segregation of the basic modules is an easy way of avoiding

molecular mixing. Although the explicit use of aside locations in a single cell [Tamsir et al., 2011] is possible, a simpler scenario involves the use of multiple cell types. Separation of the computing elements among different cell types is known as "distributed computation". The design must take into account the need for communication: some cell types sense signals and send an output (called wiring molecule) that is received and properly processed by a receiver cell. Feedbacks are also introduced in most proposals [Alon, U., 2007; Çağatay et al., 2009; Seoane and Solé, 2013]. In such strategy, the new physical insulation introduced, allows to reuse the same genetic components from a general collection in different combinations for each cell type without additional engineering. At the same time, it facilitates circuit implementation and reduces interference with the host cell by minimizing the number of components introduced into each cell. The multicellular framework represent the emergence of a new layer of modularity since every cell type is a basic module that performs a very simple computation. The new cellular layer gives the ability to easily reuse and recombine its elements. Mixtures of different combinations of cells perform distinct computations (Fig 16).

Moreover, the result of the output–cells, comes from the wiring molecule secreted by several cells, thus buffering spurious or noisy responses. Cellular consortia with distributed computation, have already been attempted [Weber et al., 2007; Brenner et al., 2007; You et al., 2004; Basu et al., 2004, 2005; Kramer et al., 2004]. Paradoxically, although communication between isolated organisms in nature is an ordinary action and cooperation of different coexisting cell types is the essence of multicellular organisms, designing multicellular systems to exhibit permanent coordinated behavior is one of the biggest challenges for synthetic biologists [Danino et al., 2010]. The struggle lies in a key feature: the maintenance of ecosystem homeostasis. Several published works address this issue. In prokaryotes, as already mentioned, coupling the quorum sensing system to a lethal protein in high doses, results in a stable cell population over a long period of time [You et al., 2004]. Similarly, a predator–prey oscillatory ecosystem dynamics has also been implemented [Balagadde et al., 2008]. In yeast cells, two strains able

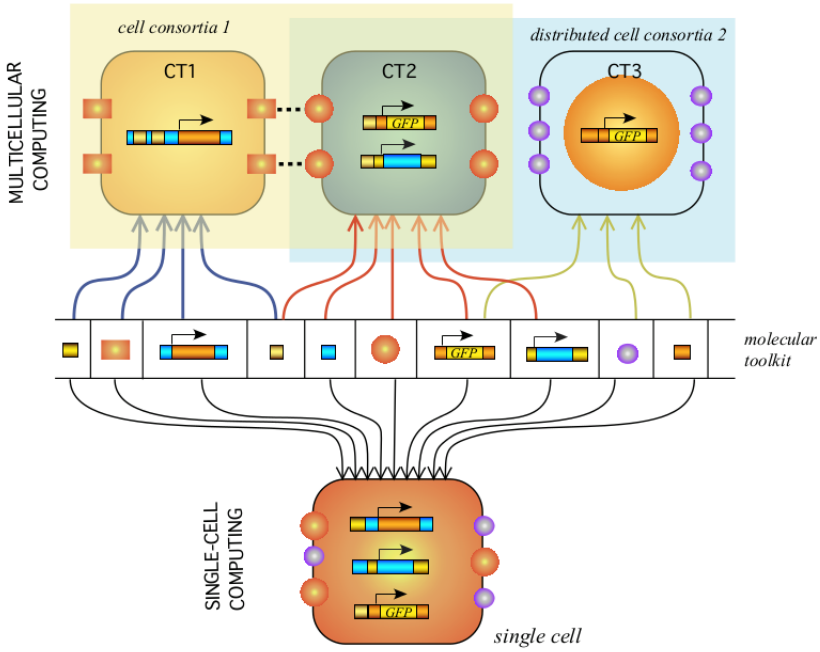


Figure 16: A genetic "toolkit" elements (central line) might include all sorts of sensors, regulatory elements and reporters. The genetic elements can be used to engineer cellular computations. In the single cell standard approach, all genetic elements needed to implement the desired function are used together within a cell. Thus, the same molecules can not be reused (bottom diagram). Alternatively, a library of different engineered cells can be created, thus defining a cellular consortia (top diagrams), the same genetic element can be used to engineer different cells. Moreover, different combination of cells allows implementation of different functions. Adapted from [Solé and Macia, 2011].

to secrete a metabolite that is vital for the survival of the other one have been engineered to perform an artificial symbiotic behaviour [Shou et al., 2007]. Recently, a new theoretical approach based on self–maintenance and differentiation of stem cells had been proposed to create a stable population of pancreatic beta–cells for diabetic patients [Miller et al., 2012]. Despite all advantages of multicellular design for modularity, researchers will always need to be mindful of intercellular and extracellular environments during the design process, and to keep in mind that the components will be most useful if they assist to the dynamic equilibrium of the eco–system.

Applications of biological computation

Despite the increasing complexity and highly innovative circuit design, synthetic biology's current state is still that of a 'proof of concept' discipline. Nevertheless, the progress toward future relevant applications to biomedicine and biotechnology is imminent. Synthetic biology engineers envision within the current designs a near horizon of unlimited real functional applications: Engineered cell populations programmed to form spatial patterns could serve as templates for building user–defined physical structures, with implications for programmed tissue regeneration and the formation of complex biomaterials. Bacteria could be engineered to treat biological wastes or to clean–up of toxic spills. The future engineered biosensors could enable the creation of organisms with exquisite detection capabilities that can identify hazards such as pathogens or explosives. Cells with synthetic networks could be utilized to supplement or supplant current methods for the industrial production or as medical implants to heal metabolic disorders, cancer or autoinmune diseases [Bacchus et al., 2013]. It is clear that the power to freely manipulate the set of instructions governing the behaviour of organisms will have a tremendous impact on our quality of life and our ability to control the physical world surrounding us. While the main problems are being solved: noise and the intrinsic fluctuations of natural systems, unpredictability of compatibility between endogenous host circuits and synthetic ones, unattainable expandable complexity and the long–term homeostatic system maintenance to ensure a proper dynamic response at all times; the developed prototypes are more and more ready to be finally used [Kwok, 2010].

Prototypes for Biomedical devices

To cure and regulate the type 2 **diabetes** a design of subcutaneous implants containing light–inducible transgenic cells was proposed. Controlled expression of the glucagon–like peptide 1 (GLP–1) is able to attenuate glycemic excursions in type II diabetic mice. The Melanopsin

of retinal ganglion cells, when stimulated with blue–light, activates transient receptor potential channels (TRP channels) via a G–protein signalling cascade, resulting in calcium influx. The nuclear factor of activated T cells (NFAT) is responsive to elevated calcium levels. By linking both systems, expression of the GLP–1 under the control of the NFAT–responsive promoter resulted in blue light–controlled blood–glucose homeostasis in type 2 diabetic mice [Ye et al., 2011]. To paliate the **metabolic syndrome**, a collection of interdependent pathologies including hypertension, hyperglycemia, obesity and dyslipidemia, mouse mammalian cells were engineered. This cells expressed a chimeric trace amine–associated receptor 1 (cTAAR1), which produced a stronger cAMP response compared with its native counterpart in response to the antihypertensive drug Guanabenz. Increased cAMP concentration triggers expression from a synthetic promoter (PCRE) via CREB1 protein. Thus, the dose of Guanabenz simultaneously controls hypertension as well as expression of a bifunctional hormone: GLP–1–Leptin. This dual protein combines the glucagon–like peptide 1 (GLP–1) anorexic and insulin secretion effect with the Leptin lipid level and food intake control capacity. Implanting the circuit in mice that were developing symptoms of the metabolic syndrome enabled simultaneous correction of all associated pathologies [Ye et al., 2013].

A highly sophisticated multi–input design circuit, which allowed for specific **cancer cell recognition**n and destruction, has been reported by [Xie et al., 2011]. They constructed a cell–type classifier that scored high and low levels of cancer cell–specific microRNAs and when matching the predetermined profile, programs the target cancer cells for apoptosis. The high–level of all three microRNA markers, miR–21, miR–17 and miR30a, targeted mRNA of the transactivator rtTA and the transrepressor LacI. The rtTA was designed to activate expression of LacI, while LacI in turn was designed to repress the expression of the apoptosis–inducing hBax, by binding to the CAGop promoter [Xie et al., 2011]. Chimeric antigen receptors (CArs, receptors designed with single–chain antibodies as part of their detection mechanism) have been developed, and ideally could achieve the same diversity and selectivity of recognition as antibodies

sensing disease associated antigens [Irving and Weiss, 1991]. This technique had been proposed to modify T lymphocytes or natural killer cells to identify and kill tumor cells to block autoimmune disease or the rejection of cell transfer therapy [Chen et al., 2010].

Prototypes for biotechnology industry

To relieve some pressure on agriculture, next–generation **biofuels** should be produced from non–food sources such as algae, cyanobacteria, and switchgrass. Synthetic biology offers the opportunity to engineer cells to secrete fuel intermediates (such as lipids and fatty acids) that could be refined later. Synthetic Genomics Co. is trying to create a genetically modified photosynthetic algae as a "biocrude" that can be refined into gasoline, diesel, and jet fuel. Apart from new fuels, better hazardous waste cleanup is also cited as one of synthetic biology's environmental promises. **Bioremediation** is already common in oil spill cleanups, Rhodococcus and Pseudomonas bacteria among others, naturally consume and degrade many petroleum components into less toxic products. Replacing non–essential genes, of this robust microbes that survive in harsh conditions, by engineered metabolic and regulatory circuits that degrade target compounds such such as dioxins, pesticides, or even radioactive elements could save millions of dollars otherwise spent on excavating and trucking polluted soils to hazardous waste landfills [Schmidt and de Lorenzo, 2012]. **Biosensors** have been long used, but new proposals following the modular architecture approach of synthetic biology are emerging [van der Meer and Belkin, 2010]. For example, a set of scalable *E.coli* biosensors constructed to detect and integrate multiple environmental contamination associated signals consist in: exchangeable genetic signal sensor, integration circuit, and actuation modules. The sensors, either native or from other bacteria, were wired to a genetic logic `AND` circuit to distinguish *As3+*, *Hg2+*, *Cu2+*, *Zn2+* and the 3OC6HSL quorum sensing molecule (and combinations of them) with quantitative fluorescent as output. Importantly, such engineered populations can function as a filter and an amplifier to enhance the selectivity and

sensitivity of actual cell–based biosensors and are also proved to work as a population sensor detecting and integrating the environmental signals via intercellular communication [Wang et al., 2013].

Can synthetic biology deliver what it promises outside the laboratory as well? To achieve this ambitious goal, it is crucially important to consolidate the advances that have been made to create still more robust and complex circuits, as these in turn will ensure safe and reliable usage. Time has now come to implement and validate these designed devices in industry, environmental tests or clinical assays. Therefore, the synthetic cells will have to traverse the same clinical phases and likely meet with similar technical challenges as gene therapies.

OBJECTIVES

The general objective of this phD thesis was the design and implementation of novel **synthetic cellular systems to perform highly complex functions**. The main problem to build scalable and reusable modular biological devices is that the number of different proteins required to implement a circuit is proportional to its complexity. Our efforts have focused on **solving the so called wiring problem**. The specific objectives of the PhD project were:

1.- Design and engineer a **Library** of yeast cells to sense several **inputs** without crosstalk and that respond with a specific logic.

2.- Implement different **wiring** molecules to connect independent cells.

3.- **Characterize** the cells of the yeast libraries: transfer function, stability and robustness.

4.- Design and implement complex **circuits**.

5.- Introduce **spatial constrains** on distributed computation circuits.

RESULTS

Distributed biological computation with multicellular engineered networks

Regot S, Macia J, Conde N, Furukawa K, Kjellén J, Peeters T, Hohmann S, de Nadal E, Posas F, Solé R. Distributed biological computation with multicellular engineered networks. Nature. 2011 Jan 13;469(7329):207-11. doi: 10.1038/nature09679

Results

Complex electronic devices run on digital circuits assembled from logic gates. All logic gates use unambiguous rules to convert inputs (0 or 1) into outputs of (0 or 1). To perform logic operation in biological systems researchers engineer synthetic circuits involving DNA, RNA or proteins. Compared with the assembly of electronic devices, construction of complex biological circuits by layering of elementary gates is tremendously challenging. The proportional growth between complexity and needed of different wires converts the implementation of complex functions in a difficult task due to crosstalk between molecular elements. Moreover, single–cell molecular computing devices are arduous to implement reliably, especially by overloads of its metabolism and, cannot be optimally reused. To minimize the need of wires we designed a multicellular strategy that permits to perform complex computation tasks flexibly.

Distributed computation leads to a reduction of wiring requirements

Distributing the computation among several cells, instead of introducing several wired logic gates into a single cell, if properly designed, could be a way to reduce the wires in a biological system. Standard method for circuitry design focus on the minimization of the number of logic gates but do not pay attention on the number of wires or the complexity of the connection pattern because in electronics this is not a limiting factor. Hence, we proposed a novel architecture following this rules:

1. Cells that respond to one input stimulus activate or inhibit the production of the wiring molecule or the output protein. Cells that respond to two inputs (input stimuli and a wiring molecule) can activate or inhibit the output production. Therefore, the output can be produced by different cell types simultaneously, the so called: distributed output production.

2. Feedback connections are not allowed.

3. Select the circuit design that involves the minimum number of wires and less different cell types.

Altogether, the final computation implemented by the network of cells will be determined by the number of cells involved, the specific function that each cell implements and the location of each cell within the network (Fig S1). Theoretical analysis to explore the possible minimal sets of gates to be used according with the above rules is performed through an evolutionary algorithm [Macia and Sole, 2009]. Interestingly, these different constraints converge into the same basic space of possible solutions. Here we demonstrated that it is possible to minimize the number of required cells and wires using distributed output and a small library of cells that implements only the AND and the N-IMPLIES gates. Furthermore, these gates, can be simplified and replaced by IDENTITY and NOT gates respectively allowing for circuit simplification. Importantly, this combination of gates define a functional complete set discrepant from the standard solutions (the NOR and NAND gates traditionally used as the single logic elements to implement any logic circuit). Of note, N-IMPLIES gate is not commonly used in electronic designs, but it has a clear biological role in many regulatory genetic networks in which one of the inputs triggers the expression of an output gene whereas other blocks gene expression.

Allowing more than one cell type to produce the output element, the requirement of different wirings is strongly reduced, facilitating the building of complex designs by combining a limited number of engineered cell types. The potential power of output distributed computation is illustrated by the fact that even a small number of engineered cell types makes possible to implement hundreds of synthetic circuits. Moreover, only with few wires the combinatorial power of the system rapidly increase in orders of magnitude, and with it, the number of potential functions. Almost all possible two and three input functions could be implemented using 2–5 different cells (Fig. S2A) and 3–4 different wires (Fig. S3A). In contrast a NAND based approach would require 13–14 different wires to achieve the same complexity (Fig S3C).

Moreover, several advantages arise from a multicellular design: The simplicity of each cell (only respond to one input and/or single wire molecule, with a NOT, AND or N−IMPLIES logic) reduces the interference between the host cell and the external components, the distribution of the logic blocks in different cells allows to reuse the same molecular elements to perform different tasks, and finally, noise diminishes because the output generated corresponds to the sum of a population of wiring producer cells buffering spurious responses.

All together, this approach significantly improves fundamental constrains of biological computation such as re–usage of biochemical modules, reduction of interferences with the host cell and reduction of wiring requirements.

The library of engineered yeast cells perform a set of logic gates functions

Yeast offers different advantages for biological computation such as easy genetic modification or well known molecular biology. There is an important number of available synthetic tools to modulate gene expression.

The yeast pheromone pathway was used as wiring system taking advantage of its secretable nature and knowledge of the signaling pathway. Several transcriptional circuits were used to regulate either pheromone production or pheromone signaling by conditionally regulating Fus3 MAPK expression. Another advantage of the pheromone system is that is conserved among yeast species but specialized enough to prevent crosstalk which allow the implementation of an increasing number of wiring molecules by functionally expressing receptors and pheromones of other yeast species in *S.cerevisiae* (i.e *C.albicans*).

Typically, the one input/one output cells expressed the pheromone following IDENTITY or NOT logics upon input addition. For the two input/one output cells, taking into account that one of the inputs was the pheromone, cells induced or repressed Fus3 MAPK in response to an

input (resulting in AND or N-IMPLIES logics respectively). Therefore, the same module components are reused, to trigger alpha factor or Fus3 kinase production. The final output of the computation consisted in the expression of GFP under the pheromone responsive promoter *FUS1*. Thus, we engineered a small library of cells implementing IDENTITY, NOT, AND and N-IMPLIES functions able to respond to distinct inputs and communicate by a wiring molecule. Any function can be performed only using that logic gates, thus, the cell library successfully implemented a functional complete set (Fig S4).

Each cell of the library performs the desired logic function. However biological systems also have to exhibit a proper dose-response (input–output) relationship. The question is whether the analogical behaviour of a engineered yeast cell can respond as a digital device. In digital computation clear separation between the logic state 0 and 1 is essential. This separation of the states is defined by a threshold. Some topologies of signaling networks have evolved to perform a sharp transition between two states [Macia et al., 2009b]. Actually, this feature is very desirable to survive to sudden environmental changes. The digital like response is imperative in multicellular computation to avoid noise amplification or signal wastage. Thus, we calculated the transfer curve for each cell in the library which consists in the quantification of the amount of output produced in a range of input concentrations. As shown in Fig 2B and S5, every cell showed an appropriate transfer curve in terms of high gain ranges (to ensure that signal will not be degraded from input to output) and step like shape without overlap between the high and the low state. GFP positive cells in the circuit cultures exhibit responses of nearly 100% (calculated from the total of cells that could respond in that specific input combination) demonstrates that the wiring is highly efficient (Fig. 2A). CMOS electronic technology defines a 5–fold increase from the low to the high state as a significant gap region to distinguish between them properly. As shown in Figures 2–4, S6, 7 and 9, in our biological circuits, the 0 logic state represent less than 10% of the maximal value, indicating that these circuits are comparable with electronics in terms of resolution. We also validated that each cell responded specifically to its input and thus, there

was no signal crosstalk within the circuit.

Therefore, the engineered yeast library, which individually performs specific digital response of AND, NOT, IDENTITY and N−IMPLIES logics and communicates by highly efficient wiring system, constitutes a perfect system for implementing distributed computation.

Different cell combinations serve to perform distinct computations

To implement different logic circuits, we combined cells from the library. All standard 2–input logic functions (AND, OR, NAND, NOR, IMPLIES and N−IMPLIES) were implemented by appropriate combinations of engineered cells. Cultures of two cells allowed to perform AND and NOR logics (Fig3a–b; Fig S6a–b). Three cell networks, combining cells previously used, compute OR, NAND and IMPLIES gates (Fig 3c–d; Fig S6c). Noteworthy only six different cells from the library were used. This illustrates the combinatorial advantage of modularity at cellular level. Although AND and N−IMPLIES gates were already implemented as single cells, they could also be build by combining two cells with different logics, responding to two input stimuli distinct from alpha factor. Using an alternative set of cells we created AND, NOR, OR, NAND, XOR and XNOR logic gates which responded to the same inputs (Glucose and Doxycycline; Fig S7). All together, these results strongly support that this multicellular approach allows programming multiple functions by combining reusable cells from a small library of cells.

To expand the complexity of the circuits, additional wiring molecules were needed. As mentioned above we chose the pheromone system as wiring molecule not only because it is a secreted peptide but also because its conservation among yeast species makes easier to implement new wirings. Following this principle, we expressed heterologous Ste2 receptor and pheromone of *Candida albicans* in *S.cerevisiae* yeast cells. Thereby, a second wiring was enabled without engineering new genetic modules. As an example of higher complex functions, we implemented a three input multiplexer (MUX2to1) using four cells and both wiring

pheromones (Fig 4a). In electronics, the selector input "chooses" one of the two inputs. Similarly, in our circuit, the presence or absence of Doxycycline (the selector) determines between *S.cerevisiae* and *C.albicans* wiring pheromone, thanks to `IDENTITY` and `NOT` cells.As mentioned above, `AND` and `N-IMPLIES` logic gates can be implemented by two cells or by a single cell if pheromone is considered one of the inputs. Therefore, the implementation of a three cell MUX2to1 is possible (Fig S10B). Although a circuit such as MUX2to1 can be constructed in a single cell [Moon et al., 2011] it is difficult to engineer and with limited hope of being re–used as part of a larger system.

The complexity of a function not only increases with the amount of inputs but outputs. To demonstrate that distributed computation is a suitable approach to perform a two input two output function, we implemented: a 1–bit adder with carry. Basically, the system computes an `XOR` logic for the adder output (GFP) and an `AND` logic for the carry (mCherry) responding to the same inputs (Doxycycline and Glucose). The circuit consist of five cells connected by two wirings, different fluorescent proteins allows to detect both outputs from the same culture. Interestingly `XOR` and `AND` logics shared one cell, which upon glucose secretes a wiring molecule to the media connecting with two distinct cells (Fig 4b). Therefore we demonstrated that our system allows scalability of complexity with small modifications in some components.

Cellular circuits are stable and reprogrammable

As specified in the Introduction, the main problem of cellular consortia are homoeostasis and stability. In multicellular distributed computation, several strains must coexist in one culture, thus, different growing rates between them might eventually impair circuit competence. In fact, phosphorylation of Far1 by the Fus3 kinase, in cells that respond to α–factor, cause cell cycle arrest. However, dynamical analysis of our distributed multicellular circuits proved that the response of a circuit such as an `AND` gate is stable for up 9 hours in presence of input signals (Fig S8a). Furthermore, when cells of the circuit were mixed and the

culture was kept at growing log phase for more than four generations, the circuit still maintained the competence (Fig S8b). Additionally, using a microfluidic device, cellular consortia was exposed to a dynamic input stimulation. In each chamber, the circuit was exposed to different combinations of inputs, and after correct computation, the cells were photo–bleached and the input combination was changed. The AND circuit was able to compute the correct output more than once (Fig S8c). Still, a single cell circuit would be more stable, thus, further genetic engineering is needed to make circuits with stable populations levels.

An additional layer of control above multicellular network enlarges the combinatorial possibilities of a multicellular network by selective change of the logic of a particular cell. The analog sensitive mutants of synthetic engineered kinases, designed to be inhibited by a specific molecules that are innocuous toward native cellular kinases, serve for this purpose. For example, when a selective inhibitor is added to cells with a engineered pheromone pathway (MAPK Fus3as) the logic response is altered (Fig S9a). Similarly, when glucose was added to the (NaCl, Galactose) `OR` gate, this responds as an `IDENTITY` for NaCl (Fig S9B). Therefore, the same multicellular circuit, by addition of reprogramming substances, was able to perform different computation.

Discussion

The main problem to build scalable and reusable modular biological devices is that the number of different proteins required to implement a circuit is proportional to its complexity. Our aim was to design and implement novel strategy to surpass the wiring problem and create a modular synthetic cellular system to perform highly complex functions. We addressed this objective by engineering a small library of yeast cells implementing `IDENTITY`, `NOT`, `AND` and `N-IMPLES` functions and combining them to perform multiple complex circuits. Each cell constitute a reliable digital response module, which can successfully communicate with the other ones by two different wirings. The cellular consortia were quite stable and able to respond to sequential stimulus as

well as to perform two output computations.

It should not be forgotten, that although implementation of biological circuits is fairly straightforward: the cells were mixed proportionally, the inputs were set at the beginning of the experiment and the output was measured at its end. This superficially simple method has many hidden degrees of freedom. Definition of the "off" and "on" states, the input working concentrations, the duration of the computation and the proportions of the circuit cell populations within the "space" that maintains the circuit competence, is an important experimental design feature, that needs to be studied in every cellular consortia.

The use of a novel non–standard functionally complete set to split the circuit design into simpler units in conjunction with the contravention of common assumption of specifically located output were the keys to make complex computations readily achievable and to drop the wiring requirements. Even so, the scalability of complexity is still limited by wire needs, and other distributed methodologies should be explored.

Recently, [Tamsir et al., 2011] had addressed an alternative multicellular distributed computation framework. They constructed simple NOR logic gates in *E.coli* by arranging two tandem promoters to induce transcription of a repressor. Hence, only in the absence of both inputs the output was produced. All NOR *E.coli* strains were engineered following this genetic architecture, but they sensed and produced different quorum sensing molecules. Different spatial arrangements of colonies on agar were designed to perform the 16 two–input elementary logic gates. Comparison between strategies in terms of numbers of cells or wires versus the possible different implementable functions showed that in both cases, our distributed method allows building a larger number of distinct computations [Macia et al., 2012]. For example with four cells; almost 200 three–input one–input functions are possible with distributed computation, but less than 25 using a NOR–based design. With four wires, (further research might involve developing new wirings), our distributed computation could compute around 200 functions but NOR–based design can perform only half. Of note, spatial segregation offers one way of

combining engineered cells. Even though the new approaches could mimic electronic circuit designs, we must take into account that biological computation has different purposes and therefore the optimal principles might be different. The reported results showed that our distributed output computation architecture is useful to create reusable, robust and reprogrammable complex circuits.

Personal contribution to this work: except for the development of the theoretical framework, which was performed by Dr.Macia, I have been fully involved in the design of cells and circuits their *in vivo* implementation and the discussion of the results described in this article.

A Universal Architecture for Engineered Complex Cellular Circuits

A Universal Architecture for Engineered Complex Cellular Circuits.

Javier Macia[1*,2], Romilde Manzoni[1*,3], **Núria Conde**[1*,2,3], Arturo Urrios[3], Eulàlia de Nadal[3], Ricard Solé[2,4,5]& Francesc Posas[3,5]

[1] ***These authors contributed equally to this work***
[2] *ICREA-Complex Systems Laboratory CEXS department UPF*
[3] *ICREA-Cell Signaling Research Group CEXS department UPF*
[4] *Santa Fe Institute, New Mexico, NM 87501,USA.*
[5] *Corresponding authors: ricard.sole@upf.edu; francesc.posas@upf.edu*

Engineered synthetic biological devices have been designed to perform a variety of functions, from sensing and bioremediation to energy production and biomedicine. Actual limitation of *in vivo* circuit implementation is the restrain of standard circuit design. Thus, future success of these constructs will depend on their robustness and scalability. Here, we show a universal architecture to engineer cellular consortia that is independent of the circuits complexity. We show how a modular biocomputer can be constructed in a flexible, robust and scalable manner. The potential of the approach is presented by implementing complex logical functions up to six inputs. Our spatial modular design overcomes the combinatorial potential of standard circuits, minimizing wiring requirements and permitting reusability. Thus, this approach allows easy implementation of multicellular computing chips.

Synthetic biological devices have been built that perform a variety of functions[1–3]. However,their extended applicability will depend on how we can build complex logic circuits. Nowadays,the creation of complex logic circuits capable to integrate a high number of different inputs is one of the major challenges of synthetic biology[4–6]. Usually, complex circuits are obtained by connecting small parts, the so-called logic gates, following standard combinatorial logic. Despite the important efforts devoted to develop sophisticated logic gates[4,6–19], these achievements are not translated into more complex circuits. Different constraints limit circuit complexity[5], particularly the so-called wiring problem, i. e. how to connect high number of living logic gates where each connection (wire) needs to be implemented by a different biochemical element. According with the standard architecture of logic circuits, circuit and functional complexity will scale with both the number of different logic gates and the number of wires. Albeit, the enormous effort for the standardization of genetic parts in synthetic biology over the last decade[6], the problems derived from the restrain of standard circuit design

and the limitation of reusable complex genetic engineered cellular components call for novel strategies[4]. Here, we show a universal architecture to engineer cellular consortia that is independent of the circuits complexity and created logic circuits in eukaryotic cells that respond to up to 6 inputs (such as a multiplexer 4to1), Thus showing how a modular biocomputer can be constructed in a flexible, robust and scalable manner.

Results

Inverted logic formulation (IFL) in multicellular consortia reduces wiring requirements. Here we present a novel alternative to standard architecture that overcomes these limitations and does not depend on the specific circuit to be implemented. The main feature is its universality, i.e. for a given number of inputs N any arbitrary circuit can be build by using the same architecture, independently on its complexity. The maximum number of different cells and modules required is determined by the number of inputs N instead of circuit's complexity. Furthermore, the circuits are modular that allows for reutilization of the same modules in different circuits. A multicellular consortia form each module[16–20], where logic gates are located in different cell types, and only one wire molecule is required to connect these cells. Of note, only Identity and NOT logic gates are necessary, minimizing the genetic engineering required to implement *in vivo* circuits. If modules are physically isolated the same wiring molecule can be used in each module. In consequence, this novel architecture provides, in several embodiments, a definitive solution of the wiring problem. Our method is summarized in Fig.1a-c. Circuits are composed of a library $L = \{C1, ..., CZ\}$ of Z engineered yeast cells. Each cell reacts to only one input from a set $X = \{X1, ..., XN\}$ of N input signals. These cells are organized in different subsets confined within separated spatial chambers (Fig. 1b) defining a set of M modules $\psi = \{\psi 1, ..., \psi M\}$.

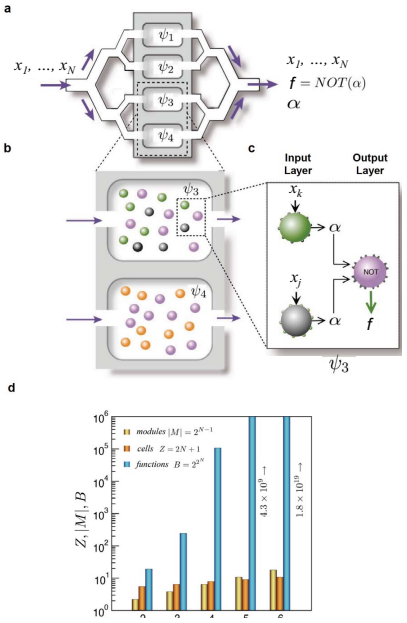


Figure 1: **Modular spatial computation. (a)** A spatially organized set of chambers defining a set of modules provides a source of modularity (test tubes or chambers in a microfluidics device) by separating different subsets of cells in different groups. **(b)** Each module contains one or more cells from the IL library, all sensing only one signal from a given repertoire of N inputs and responding by sending a communication molecule . **(c)** All modules also include an inverter cell (OL) implementing the NOT logic in response to molecule α. **(d)** The combinatorial potential of our system where we plot the required number of cells, the number of modules and the number of potential N-input logic functions against the number of inputs. Both the cost in terms of spatial modules and the library of cells scale slowly with N, whereas the repertoire of functions explodes even for small N.

In this setup, all inputs reach all chambers, whose a first layer of cells (Input Layer) will produce or not a wiring molecule according to their internal logic, either the Identity (where the state of the node (cell) is 1 only if the input is 1) or the NOT (where the state of the node (cell) is 1 only if the input is 0). This logic scheme

is completed with a key element, common to all modules, namely the inverter cell or Output Layer cell OL (Fig. 1c) sensing α and producing an output signal f following a NOT logic. It is this Inverted Logic Formulation (ILF) what actually allows for a drastic simplification of logic circuits. It can be shown (SI and Fig. S1 and S2) that ILF makes possible to avoid the standard three-layers architecture of logic circuits, which would include input, hidden and output layers and, consequently such design would require a large number of communication wires. Therefore, ILF allows creating complex circuits with a minimal configuration involving Input Layers, formed by cells that sense the external inputs that are directly connected with Output Layers that produce the final output, i.e. hidden layers are not required since Input and Output Layers are directly linked. When spatial modules are feasible, the same wiring molecule can be used to link Input and Output Layers. Increasing the complexity of these circuits implies an increase in the number of modules M but not in the number of wires. The enormous potential of these circuits is shown in Fig. 1d. A simple calculation reveals that, while the size of the cell library scales linearly as $Z = 2N + 1$ ($2N$ input cells and the inverter cell, OL) and the maximum number of spatial modules increases according to $M = 2^{N-1}$ (see SI), the number B of functions grows super-exponentially as $B = 2^{2^n}$. For instance, with $N = 5$ different inputs only 11 different cell types are involved to implement $B = 4.294.967.296$ different functions, which in the more complex case must be distributed into 16 different modules.

A universal architecture of cells permits minimal genetic manipulation to create *in vivo* multicellular consortia that respond specifically to different inputs. To implement modular biocomputing in vivo, we created a library of engineered yeast cells for the Input and Output Layers (IL and OL) of the circuits (Fig. 2a). The Input Layer library consist in a set of pairs of cells that responds to different extracellular stimulus (e.g. doxycycline, progesterone, 17-β-estradiol) (Fig. 2a); each pair consists in two different type of cells that respond to the same stimuli but with a different logic, either Identity (ID) or NOT (NOT) and secretes a single wiring molecule accordingly (i.e. yeast pheromone) (SI and Fig. S1). Briefly, ID cells secretes pheromone upon stimuli by expressing *MF(α)1* gene from a specific promoter that respond to a defined stimuli. To systematically create the corresponding pair wise cell with the NOT logic, the LacI repressor was expressed from the same stimuli-specific promoter used in the ID logic. These cells expressed the *MF(α)1* gene under a modified *TEF1* promoter containing LacI binding sites (P_{TEF1i}) and thus, in the presence of stimuli, pheromone expression is inhibited (Fig. 2a, S3 and S4a). The genotype and the graphical notation of the logic function performed by each cell of the library is depicted in Fig. S3. Cells in the Output Layer (OL) respond to pheromone, from the input Layer, with a NOT logic. Basically, those cells expressed constitutively under the *TEF1i* promoter a modified version of yEGFP or mCherry tagged for degradation (ssrA)[21]. The presence of pheromone induces LacI expression which leads to down-regulation of fluorescent protein expression. Pheromone also stimulates degradation of the fluorescent reporter by the induction of the ClpXP protease complex (Fig. S3 and S4b). The ability of cells to respond to external stimuli (inputs) was monitored by fluorescence in single cell by flow cytometry and normalized to the maximal number of cells able to produce output signal (SI and Fig. 2b, S5). Each cell type has been characterized by its ability to respond to the corresponding stimuli (transfer function) (Fig. S6). Furthermore, the crosstalk

of the Input Layer cells to the different inputs, measured using the NOT Output Layer, was not significant (Fig. 2c).

Combinatorial modular organization of a minimal library of modified cells permit to create complex logic circuits. We initially tested our library of cells to engineer a complex circuit, known as majority rule. This is a decision-making system used in electronics as security device against failure in redundant systems. In a biological context, these decision-making circuits might serve to determine the existence of the minimal conditions to trigger a cellular process. Here, we designed a majority rule circuit that responds to three different inputs (i.e., dexamethasone (DEX), 17-β-estradiol (EST) and progesterone (PRO)). The equivalent, single-cell design would be very difficult to implement *in vivo*. However, following ILF, the implementation just requires three cells from the IL library together with an inverter cell (OL1) and three chambers (Fig. 2d). The *in vivo* results clearly showed that only when at least two of the inputs were present there was a positive output (Fig. 2e).

Next, we increased circuit complexity by creating a circuit that responds to four different inputs producing two different outputs. We chose a 2-bit magnitude comparator, which permits to compare two numbers each having two bits (a_1, a_0 and b_1, b_0). Comparators are at the heart of most central processing units (CPUs) in computers performing a large portion of the logical operations. The circuit is able to respond to 4 inputs, upon 16 entries and yields three different outcomes from the computation ($A > B$, $A < B$ and $A = B$). The implementation involves four pairs of cells of the Input Layer library (ID and NOT) that respond to four stimuli (i.e., doxycycline (DOX), EST, PRO and DEX) combining them within six chambers. Half of the chambers contained either green (GFP) or

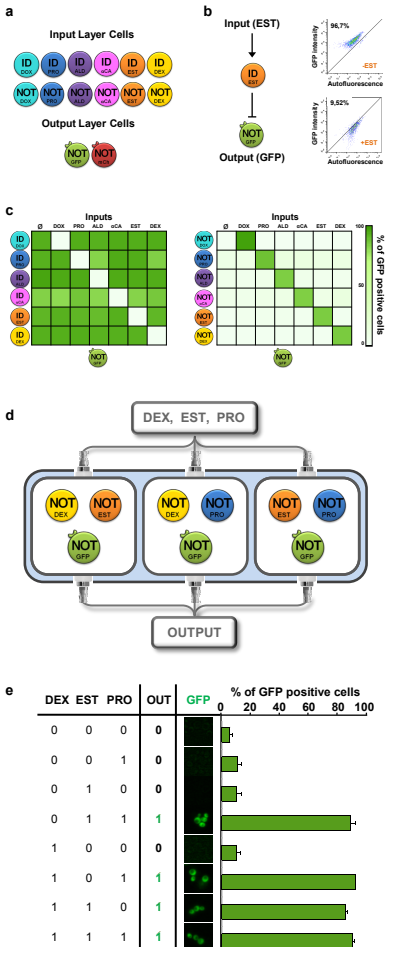
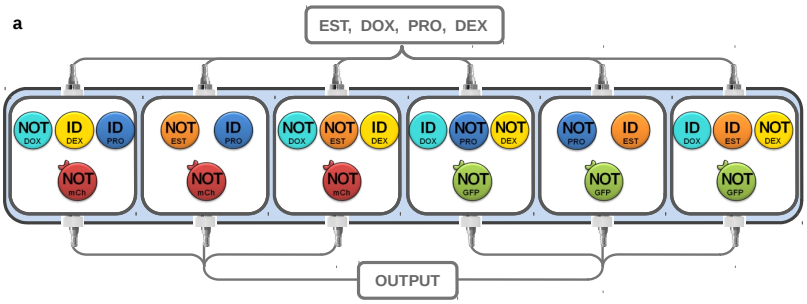


Figure 2: **Library of engineered cells to implement complex biological circuits.** **(a)** Schematic representation of the Input (IL) and Output (OL) Layer cell library. Each colour indicates cells that respond to a different input with Identity (ID) or NOT logic (see SI for complete cell library description). **(b)** Quantification of single cell computational output. An IL cell stimulated with 17-β-estradiol (EST) mixed with OL1 cell in the absence (-EST) or presence (+EST) of the input. Percentage of GFP-positive cells analyzed by FACS (right). **(c)** IL cells (ID, left; NOT, right) were mixed with OL cells (GFP) in presence of different inputs to assay for response and crosstalk. Results are expressed as percentage of GFP-positive cells. **(d)** Schematic representation and spatial organization of the cells used in the Majority Rule circuit. **(e)** Truth table (left), microscope images (middle) and percentage of GFP-positive cells (right). Cells were mixed proportionally and different combinations of inputs (dexamethasone, 17-β-estradiol and progesterone) were added at the same time. Data represent the mean and standard error of three independent experiments.

red (mCherry) Output Layer cells (Fig. 3a). The 17-β-estradiol and doxycycline serves to encode A, whereas progesterone and dexamethasone encodes B. The output yields green when $A < B$, red when $A > B$ and no signal when both were equal (Fig. 3b). All 16 combinations yielded the expected outcome supporting that multiple functions can be constructed from a small library of reusable cells. To demonstrate the scalability of this modular approach,



b

a1 EST	a0 DOX	A	b1 PRO	b0 DEX	B	A ≷ B
0	0	0	0	0	0	0 = 0
0	0	0	0	1	1	0 < 1
0	0	0	1	0	2	0 < 2
0	0	0	1	1	3	0 < 3
0	1	1	0	0	0	1 > 0
0	1	1	0	1	1	1 = 1
0	1	1	1	0	2	1 < 2
0	1	1	1	1	3	1 < 3
1	0	2	0	0	0	2 > 0
1	0	2	0	1	1	2 > 1
1	0	2	1	0	2	2 = 2
1	0	2	1	1	3	2 < 3
1	1	3	0	0	0	3 > 0
1	1	3	0	1	1	3 > 1
1	1	3	1	0	2	3 > 2
1	1	3	1	1	3	3 = 3

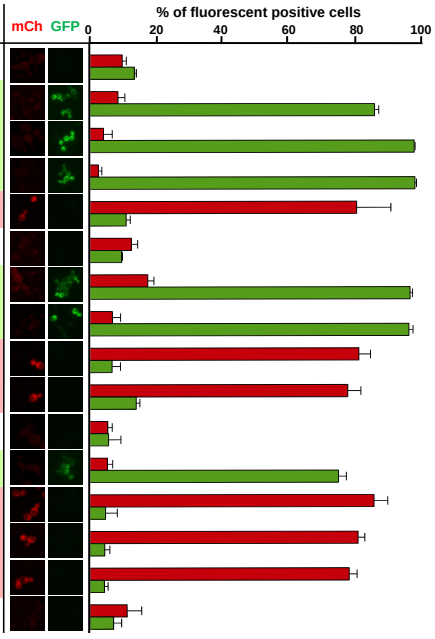
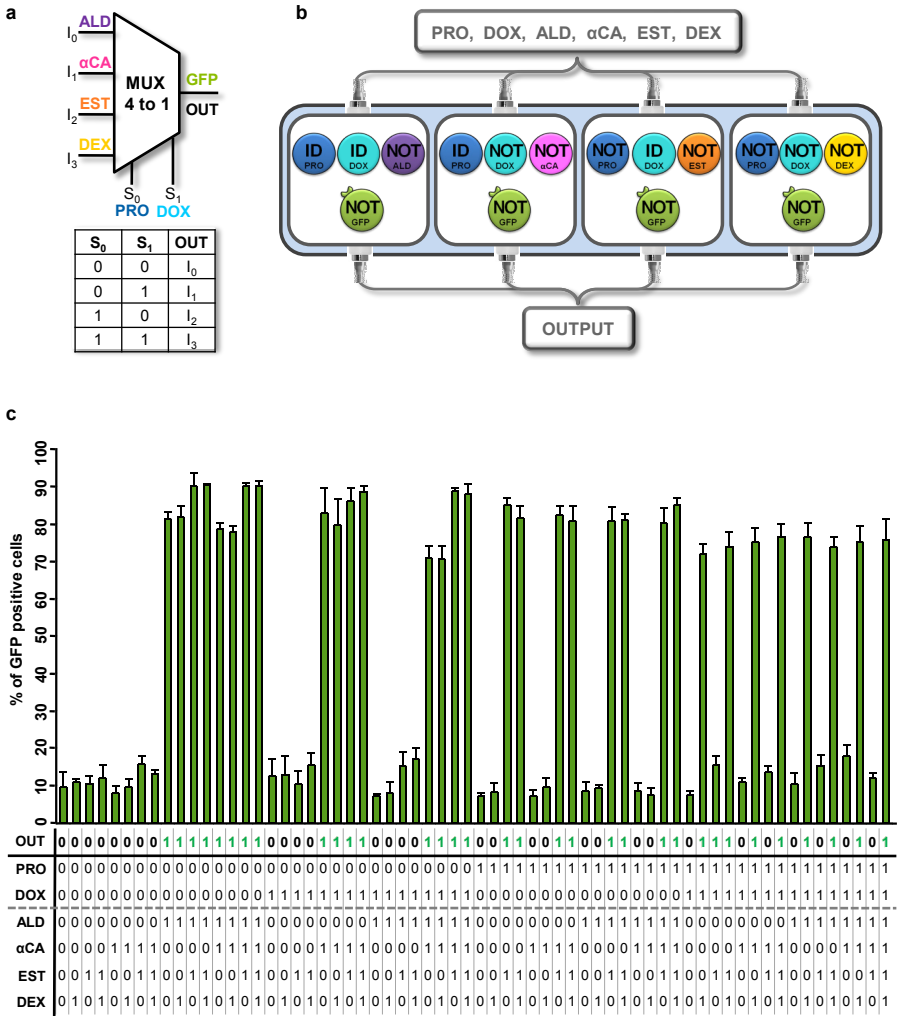


Figure 3: **Design and in vivo implementation of a 4-inputs COMPARATOR. (a)** Schematic representation and spatial distribution of the cells used in the COMPARATOR. **(b)** Truth table (left), microscope images (middle) and percentage of FACS fluorescent-positive cells (right). Cells were mixed proportionally and combinations of four inputs (17-β-estradiol, a_1; doxycycline, a_0; progesterone, b_1; and dexamethasone, b_0) were added simultaneously. Green (GFP) and red (mCherry) bars represent output quantification ($A > B$ red; $A < B$ green; $A = B$ no signal). Data represent the mean and standard error of three independent experiments.

we implemented a highly complex multiplexer involving 6 inputs. A multiplexer permits that several signals share one device instead of having one device per input signal. The MUX4to1 is a circuit that selects one of four different input signals (I_0-I_3) and forwards the selected input (S_0-S_1) into a single output. Here, four input signals (I_0-I_3: aldosterone (ALD); *C. albicans* pheromone (αCa), EST and DEX) are selected by two additional inputs (S0-S1: PRO and



S_0	S_1	OUT
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Figure 4: **Design and in vivo implementation of a 6-inputs multiplexer (MUX4to1). (a)** Schematic representation of the input stimuli (I_0-I_3) and selectors (S_0-S_1) in the MUX4to1 circuit. **(b)** Schematic representation and spatial distribution of the cells used in the MUX4to1 circuit. **(c)** Truth table (bottom) and percentage of GFP-positive cells analyzed by FACS (top). Cells were mixed proportionally and treated with combination of progesterone (selector 0, S_0), doxycycline (selector 1, S_1), aldosterone (input 0, I_0), *C. albicans* α-factor (input 1, I_1), 17-β-estradiol (input 2, I_2) and dexamethasone (input 3, I_3). Data represent the mean and standard error of three independent experiments.

DOX) (Fig. 4a). Thus, a total of 64 combinations of inputs are possible. This circuit, which would represent an enormous effort to be implemented in a single cell, can be assembled involving just eight cells from the IL library and the reporter cell (OL) combined in four independent chambers (Fig. 4b). Here, although the complexity of the circuit required a differential output to 64 different input combinations, the *in vivo* results clearly showed the expected response (Fig. 4c).

Discussion

A major challenge in the field of synthetic biology is the construction of flexible, scalable and complex functionalities using engineered cells. Many different strategies have been implemented to create logic circuits in biological systems over the last decade[6], however, the problems derived from the restrain of standard circuit design and limited reusability pose a big limitation towards scalability[4]. The results presented here indicate that using modular biocomputing, a small library of cells allows a virtually unlimited scalability. For instance, a library that responds to six inputs as reported here should be sufficient to create up to $1.8\text{x}10^{19}$ different circuits, with a (most complex case) maximum of 32 modules. Our modular biocomputing design is especially suitable for reaching the scalability required to build living chips based on the combinatorial logic of cellular consortia. Spatial confinement of cells has been achieved in a broad range of conditions using microfabricated environments[22,23]. This universal architecture is not limited to modular designs. In scenarios where modular systems can not be implemented, this design permits that by using a minimal set of wires, which for instance a MUX4to1 that respond to six different inputs would require only four wires, thus allowing that the system still permits maximal complexity with minimal wiring requirements. Given the potential for large-scale integration already developed in cell-free microfluidic chips[24] our design can provide a universal and robust way of exploring the landscape of living computational devices, comparable with the combinatorial richness of standard microelectronics.

Methods

Engineered yeast cell library: Yeast W303 (ade2-1 his3-11,15 leu2-3,112 trp1-1 ura3-1 can1-100) cells were genetically modified to be able to produce S. cerevisiae alpha factor from an inducible promoter (Input Layer cells), or to control fluorescent protein expression in response to the pheromone (Output Layer cells). Schematic genotypic characteristics of each cell and plasmid used are summarized in Fig.S3 and in Tables S1 and S2.

Growth conditions: Overnight cultures were diluted to $OD_{660nm} \approx 0.2$ and grown at 30°C in YPD or selective medium.

Characterization of the cellular properties of the engineered cells of the library: As in Regot et al. (2011)16, we followed standard electronics to define a positive signal from a circuit. As shown in Fig. 2-4, in our biological devices the resolution of the 1 logic is more than 70% and 0 logic is less than 20% of the maximal value, indicating that these circuits are comparable with electronics in terms of resolution. However, this separation between logic states is necessary but not sufficient condition to guarantee that multicellular circuits can be implemented connecting different cells acting as logic blocks. A proper characterization of the library of engineered cells is necessary to analyze the so-called Transfer Function, i.e. the cellular response with respect to different input levels. An adequate Transfer Function should be characterized by several key features25, 26, namely i) a step-like shape, ii) linear or higher gain ranges in order to ensure that the signal will not be degraded from input to output in a single cell, iii) the noise margins must be adequate, without overlap between the high and the low state, and iv) each cell must only respond properly to the specific inputs and must ignore the rest of inputs of the circuit. All these aspects have been experimentally addressed in the set of engineered cells of the library. Fig. S6 shows the full set of transfer functions for each cell. All these curves exhibit the proper shape to be logic blocks for a multicellular implementation. This procedure allows characterizing not only the cellular behaviour but the wire efficiency. Cells were grown in selective media or YPD to mid exponential

phase and then diluted to to $OD_{660nm} \approx 0.2$. Input Layer (NOT) cells were washed to remove the o/n production of alpha factor and resuspended in YPD. Each Input Layer cell was mixed with the GFP Output Layer cell (OL1) at 4:1 ratio and subjected to a different input concentration (Fig. S6a). Various concentrations of wiring molecule were added to OL cells (Fig.S6b).Samples were incubated for 4h at 30°C and analyzed by flow cytometry (Fig. S5). Data is expressed as the percentage of GFP positive cells. The transfer function represents the mean and standard deviation of three independent experiments.

Computational output detection by flow cytometry and microscopy in single cell: Output of the different chambers of the circuits, transfer function and crosstalk were analyzed after 4h incubation at 30°C with combination of inputs. Then samples were diluted in PBS and analyzed by flow cytometry (BD LSRFortessaTM). A total of 10.000 cells were collected from each sample. Constitutive fluorescence in Output Layer cells was used to assess them from Input Layer cells (mCherry for OL1 and YFP for OL2). Fluorescence intensity of Output Layer cells subset was measured versus autofluorescence and data are expressed as percentage of fluorescent positive cells (GFP for OL1 and mCherry for OL2). Data were analyzed with FlowJo or BD FACSDiva software. A representative FACS plot of our quantification method is presented in Fig. 2b and S5. For microscopy analyses, cells were harvested and resuspended in Low Fluorescent Media. Images were collected with Nikon Eclipse Ti Microscope using NIS elements Software (Nikon)and handled with ImageJ.

Crosstalk analyses: The individual cellular response of each cell in response to the different inputs they encounter within a circuit. Cells were grown in selective media or YPD to mid exponential phase ($OD_{660nm} \approx 0.2$). Input Layer (ID) cells were mixed with the GFP Output Layer cells at 2:1 ratio. Input Layer (NOT) cells were washed to remove the o/n production of alpha factor, resuspended in YPD and mixed with the GFP Output Layer cell at 3:1 ratio. Each mixture was subjected to all 6 inputs individually. Samples were incubated for 4h at 30°C and analyzed by flow cytometry. Data is expressed as the percentage of GFP positive cells. The experimental data shows that there is no undesired crosstalk and each cell responds only to the expected input.

References

1. Weber,W. & Fussenegger,M. Emerging biomedical applications of synthetic biology. Nature Reviews Genetics **13**, 21-35 (2011).
2. Bashor,C.J., Horwitz,A.A., Peisajovich,S.G., & Lim,W.A. Rewiring cells: synthetic biology as a tool to interrogate the organizational principles of living systems. Annu. Rev. Biophys. **39**, 515-537 (2010).
3. Bacchus,W., Aubel,D., & Fussenegger,M. Biomedically relevant circuit-design strategies in mammalian synthetic biology. Mol. Syst. Biol. **9**, 691 (2013).
4. Purnick,P.E. & Weiss,R. The second wave of synthetic biology: from modules to systems. Nat. Rev. Mol. Cell Biol. **10**, 410-422 (2009).
5. Kwok,R. Five hard truths for synthetic biology. Nature **463**, 288-290 (2010).
6. Benenson,Y. Biomolecular computing systems: principles, progress and potential. Nat. Rev. Genet. **13**, 455-468 (2012).
7. Elowitz,M.B. & Leibler,S. A synthetic oscillatory network of transcriptional regulators. Nature **403**, 335-338 (2000).
8. Gardner,T.S. & Collins,J.J. Neutralizing noise in gene networks. Nature. **405**, 520-521 (2000).
9. Voigt,C.A. Genetic parts to program bacteria. Curr. Opin. Biotechnol. **17**, 548-557 (2006).
10. Kramer,B.P., Fischer,C., & Fussenegger,M. BioLogic gates enable logical transcription control in mammalian cells. Biotechnol. Bioeng. **87**, 478-484 (2004).
11. Win,M.N. & Smolke,C.D. Higher-order cellular information processing with synthetic RNA devices. Science **322**, 456-460 (2008).
12. Greber,D. & Fussenegger,M. An engineered mammalian band-pass network. Nucleic Acids Res. **38**, 174 (2010).
13. Rinaudo,K. et al. A universal RNAi-based logic evaluator that operates in mammalian cells. Nat. Biotechnol. **25**, 795-801 (2007).
14. Brenner,K., You,L., & Arnold,F.H. Engineering microbial consortia: a new frontier in synthetic biology. Trends Biotechnol. **26**, 483-489 (2008).
15. Macia,J., Posas,F., & Sole,R.V. Distributed computation: the new wave of synthetic biology devices. Trends Biotechnol. **30**, 342-349 (2012).
16. Regot,S. et al. Distributed biological computation with multicellular engineered networks. Nature **469**, 207-211 (2011).
17. Daniel,R., Rubens,J.R., Sarpeshkar,R., & Lu,T.K. Synthetic analog computation in living cells. Nature **497**, 619-623 (2013).
18. Slusarczyk,A.L., Lin,A., & Weiss,R. Foundations for the design and implementation of synthetic genetic circuits. Nat. Rev. Genet. **13**, 406-420

(2012).

19. Moon,T.S., Lou,C., Tamsir,A., Stanton,B.C., & Voigt,C.A. Genetic programs constructed from layered logic gates in single cells. Nature **491**, 249-253 (2012).
20. Sol,R.V. & Macia,J. Expanding the landscape of biological computation with synthetic multicellular consortia. Natural Computing 1-13 (2013).
21. Grilly,C., Stricker,J., Pang,W.L., Bennett,M.R., & Hasty,J. A synthetic gene network for tuning protein degradation in Saccharomyces cerevisiae. Mol. Syst. Biol. **3**, 127 (2007).
22. El-Ali,J., Sorger,P.K., & Jensen,K.F. Cells on chips. Nature **442**, 403-411 (2006).
23. Goubko,C.A. & Cao,X. Patterning multiple cell types in co-cultures: a review. Materials Science and Engineering: C **29**, 1855-1868 (2009).
24. Thorsen,T., Maerkl,S.J., & Quake,S.R. Microfluidic large-scale integration. Science **298**, 580-584 (2002).
25. Weiss,R. et al. Genetic circuit building blocks for cellular computation, communications, and signal processing. Natural Computing **2**, 47-84 (2003).
26. Feng,X.J. et al. Optimizing genetic circuits by global sensitivity analysis. Biophys. J. **87**, 2195-2202 (2004).

Acknowledgements

We thank to L. Subirana and A. Fernandez for technical support. Dr. J Hasty (UCSD) for plasmids. RM was recipient of a long term EMBO fellowship. AU is recipient of a La Caixa fellowship. This work was supported by an ERC Advanced Grant Number 294294 from the EU seventh framework program (SYNCOM) to RS and FP, and the Santa Fe Institute to RS. FP and RS laboratories are also supported by the Fundacin Marcelino Botn (FMB). EN and FP are recipient of an ICREA Acadmia (Generalitat de Catalunya).

Author contributions

All authors shared all the phases and topics of the work. JM and RS developed the theoretical framework. Circuits were designed and implemented by JM, NC, and FP. EN, RS and FP wrote the paper.

Competing financial interests

The authors declare no competing financial interests.

Supplementary Information

Design of minimal logic circuits based on inverted logic formulation (ILF)
Inverted Logic Formulation (ILF)
Multicellular implementation of ILF with spatial segregation
Spatial segregation as an additional computational element
Example: the Majority Rule Circuit
Circuits' scalability

Full description of engineered yeast cell library

Tables S1 and S2

Supplementary Figures (S1 to S7)

Design of minimal logic circuits based on inverted logic formulation (ILF):

Design of minimal logic circuits based on inverted logic formulation (ILF) Despite the enormous efforts devoted to developing complex logic circuits able to perform non-trivial decision-making processes, the results obtained are far from the expected ones. Current limitations do not allow the degree of complexity necessary for industrial, environmental, or biomedical applications[1]. Logic circuits are made up from basic logic gates that, connected together, constitute the building blocks to produce more complex circuits. Several aspects determine the complexity, and hence the limitations, of a circuit, namely i) the complexity of the building blocks, e.g. logic gates, ii) the pattern of connections and, iii) the number of different wires involved.
Any set of logic gates which can be used to implement all possible logic functions only combining members of a given set, is called `functional complete set`. Not all logic gates are equally complex. For instance, biological implementation of the OR logic is easier than other logics, such as AND or XOR logics[2,3]. Additionally, logic gates that respond to a single input (Identity and NOT gates) are easier to implement than two-inputs (or higher) logic gates. Based on these arguments, the easier biological implementation of complex logic circuits can be achieved by combining several one-input logic gates with an OR logic, and by using the minimal number of wires with the simplest pattern of connections. Of note, the set OR, NOT defines a functional complete set[4].

Inverted Logic Formulation (ILF):
To analyze how a logic function can result in the optimal circuit implementation, we started with the formal definition of a logic function involving N inputs and one output. A logic function can be described by the so-called truth table, where all possible combinations of inputs and the associated outputs are defined. Systematically applying Boolean algebra rules[5], a given function can be expressed as a minimal combination of NOT, OR, and AND binary operators in the canonical form. Although there are different but equivalent expressions of a Boolean function, here we focused on minterms[6]. By this formalism, any arbitrary Boolean function with N inputs has the following general expression:

$$f = [\phi_{11}(x_1)\ AND\ \phi_{12}(x_2) \dots\ AND\ \phi_{1N}(x_N)]\ OR\ [\phi_{21}(x_1)\ AND\ \phi_{22}(x_2) \dots\ AND\ \phi_{2N}(x_N)]\ OR\ \dots [\phi_{M1}(x_1)\ AND\ \phi_{M2}(x_2) \dots\ AND\ \phi_{MN}(x_N)]$$

or in a compact form:

$$f = \sum_{i=1}^{M} \left[\prod_{j=1}^{N} \phi_{ij} x(j) \right]$$

In which Σ represents the OR operator and Π the AND operator. The function ϕ_{ij} is either a logic representation of the presence of a molecular input xj (Identity function) or of its absence (NOT function). Finally, M is the maximum number of terms present in the Boolean function, which depends on the complexity of the function, but the condition $M \leq 2^{N-1}$ is always satisfied[5]. The expression of a Boolean function f can be reduced by the systematic application of standard rules of simplification, such as the so-called Karnaugh maps[7] or the Quine-McCluskey algorithm[8].
A simplified expression of the canonical form is not the minimal implementation possible in a cellular context. Easier implementations can be systematically achieved modifying the canonical expression of the Boolean function to obtain an expression involving only OR logic (the simpler logic in a cellular implementation). This goal can be achieved applying a double negation.

$$f = \overline{\overline{f}} = \overline{\sum_{i=1}^{M} \left[\prod_{j=1}^{N} \phi_{ij} x(j) \right]}$$

According to Morgan's Laws[5],

$$\begin{cases} \overline{a \; OR \; b} = \overline{a} \; AND \; \overline{b} \\ \overline{a \; AND \; b} = \overline{a} \; OR \; \overline{b} \end{cases}$$

the Boolean function can be expressed as:

$$\overline{\overline{f}} = \overline{\sum_{i=1}^{M} \left[\prod_{j=1}^{N} \phi_{ij} x(j) \right]} = \overline{\prod_{i=1}^{M} \left[\prod_{j=1}^{N} \phi_{ij} x(j) \right]} = \overline{\prod_{i=1}^{M} \left[\sum_{j=1}^{N} \overline{\phi_{ij} x(j)} \right]} = \sum_{i=1}^{M} \left[\overline{\sum_{j=1}^{N} \overline{\phi_{ij} x(j)}} \right]$$

Where $\theta_{ij}(X_j) = \overline{\phi_{ij}(X_j)}$; then: $\sum_{i=1}^{M} \left[\overline{\sum_{j=1}^{N} \overline{\phi_{ij} x(j)}} \right] = \sum_{i=1}^{M} \left[\overline{\sum_{j=1}^{N} \theta_{ij} x(j)} \right]$

Hence, the Boolean function results in the OR combination of several computational modules ψ_i.

$$f = \sum_{i=1}^{M} \psi_i$$

These modules ψ_i are the inversion of OR combinations (symbol Σ) of inverted terms θ_{ij}, what we call Inverted Logic Formulation (**ILF**).

$$\psi_i = \overline{\sum_{j=1}^{N} \theta_{ij} x(j)}$$

Functions $\theta_{ij}(x_j)$ can be chosen among NOT or Identity functions:

$$\theta_{ij}(x_j) = \begin{cases} \overline{x_i} \\ OR \\ x_i \end{cases}$$

depending on the specific function to be implemented by the circuit.

Multicellular implementation of ILF with spatial segregation:
Using ILF, the Boolean function now is expressed as an OR combination of different computational modules ψ_i, and can be systematically translated into cellular circuits. The biological implementation is simplified by two means. First, the circuit is distributed into several cellular types (distributed computation$_{9,10}$). Second, each ψ_i modules can produce the final output, the so-called distributed output$_{9,10}$. An example could be a secretable molecule, e.g. hormones, which will give a final output of 1 no matter by which cell type is produced.

In this embodiment, each module ψ_i can be organized in two cell layers. The first one is the Input Layer formed by different cell types sensing external inputs x_j. Each Input Layer cell implements a one-input one-output logic function $\theta_{ij}(x_j)$ (Identity or NOT). Every cell, in response to the external input x_j, secretes or not a wiring molecule α. The wiring molecule is the same for each cell; hence, once it is secreted into the medium and mixed, the OR logic between the different functions $\theta_{ij}(x_j)$ into the same ψ_i module is performed.

The second layer of cells (Output Layer) is composed by a single cell type that implements a NOT function: in presence of the secreted wiring molecule, the output is not expressed whereas in its absence, the final output is expressed. Thus, the final response of ψ_i is inverted. Fig. S1 shows a schematic diagram of the architecture of a computational module ψ_i.

Spatial segregation as an additional computational element:
The global function f is the OR combination of different modules ψ_i. Here, the distributed output leads to the direct implementation of the OR function by default$_{10}$ and no additional elements are required. Still, for the correct computation, no cross-talks between different ψ_i are allowed. The number M of required computational modules ψ_i depends on the complexity of the Boolean function f, as well as the number of different wiring molecules needed. We introduced spatial segregation of the cells as a new computational element. If each ψ_i remains physically isolated, the same wiring molecule can be used in all modules, and the so-called wiring problem, in which every wire needs to be a different chemical entity, is now reduced since only one wire is needed for any given circuit independently on its complexity. The computational complexity

of the function is encoded by the number of different chambers involved. For instance, we can consider a microfluidic device in which each module ψ_i is placed in a different chamber (Fig. S1). Alternatively, other systems allowing physical separation of cells such as microencapsulation are suitable to be considered for the implementation of these type of circuits.

Example: the Majority Rule Circuit:
For illustrative purposes, we applied the ILF to design a Majority Rule circuit. Majority Rule circuits are a decision-making systems based on the presence of more than the half of all possible inputs. These type of circuits are typically used in electronics as security devices. In a biological context, these types of decision-making circuits can be used, for instance, to determine the existence of the minimal conditions to trigger a cellular process. The truth table and the corresponding canonical form describing the behaviour of a Majority Rule circuit are the following:

X_1	X_2	X_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$f = [\phi_{11}(x_1)\ AND\ \phi_{12}(x_2)\ AND\ \phi_{13}(x_3)]$$
$$OR[\phi_{21}(x_1)\ AND\ \phi_{22}(x_2)\ AND\ \phi_{23}(x_3)]$$
$$OR[\phi_{31}(x_1)\ AND\ \phi_{32}(x_2)\ AND\ \phi_{33}(x_3)]$$
$$OR\ [\phi_{41}(x_1)\ AND\ \phi_{42}(x_2)\ AND\ \phi_{43}(x_3)]$$

Here functions ϕ_{11}, ϕ_{22}, ϕ_{33} are NOT functions, whereas the rest are Identity functions:

$$f = [\overline{x_1}\ AND\ x_2\ AND\ x_3]\ OR\ [x_1\ AND\ \overline{x_2}\ AND\ x_3]$$
$$OR[x_1\ AND\ x_2\ AND\ \overline{x_3}]\ OR\ [x_1\ AND\ x_2\ AND\ x_3]$$

After applying a simplification method, e.g. Karnaugh maps, this function can be reduced to:

$$f = [x_1\ AND\ x_2]\ OR\ [x_1\ AND\ x_3]\ OR\ [x_2\ AND\ x_3]$$

Figure S2a shows the standard implementation of this circuit by combining OR and AND logic gates. Starting from this standard implementation, it is possible to reduce the complexity of the circuit by i) simplifying the logic gates involved, and ii) reducing the number of wires by applying the method presented above:

$$f = \overline{\overline{f}} = \overline{\overline{[x_1\ AND\ x_2]\ OR\ [x_1\ AND\ x_3]\ OR\ [x_2\ AND\ x_3]}}$$

$$f = \overline{\overline{[x_1\ AND\ x_2]}\ AND\ \overline{[x_1\ AND\ x_3]}\ AND\ \overline{[x_2\ AND\ x_3]}}$$

$$f = \overline{[\overline{x_1}\ OR\ \overline{x_2}]\ AND\ [\overline{x_1}\ OR\ \overline{x_3}]\ AND\ [\overline{x_2}\ OR\ \overline{x_3}]}$$

$$f = \overline{[\overline{x_1}\ OR\ \overline{x_2}]}\ OR\ \overline{[\overline{x_1}\ OR\ \overline{x_3}]}\ OR\ \overline{[\overline{x_2}\ OR\ \overline{x_3}]}$$

Therefore, in this example the circuit can be implanted by using the OR combination of three computational modules:

$$f = \psi_1\ OR\ \psi_2\ OR\ \psi_3$$

with:

$$\psi_1 = \overline{\theta_{11}(x_1)\ OR\ \theta_{12}(x_2)} = \overline{\overline{x_1}\ OR\ \overline{x_2}}$$

$$\psi_2 = \overline{\theta_{21}(x_1)\ OR\ \theta_{23}(x_3)} = \overline{\overline{x_1}\ OR\ \overline{x_3}}$$

$$\psi_3 = \overline{\theta_{32}(x_2)\ OR\ \theta_{33}(x_3)} = \overline{\overline{x_2}\ OR\ \overline{x_3}}$$

In a spatially segregated embodiment, this circuit requires three different chambers (or microcapsules), one for each ψ_i. The first chamber will contain two cells in the Input Layer. The first cell will respond to input x_1 producing the wiring molecule α according to the NOT logic. The second cell will sense the input x_2 and produce the same α molecule, following the same NOT logic. Finally, a single cell type that will produce the final output in absence of the α molecule forms the Output Layer. The rest of the chambers have the same architecture, differing in the Input Layer cell types but using the same α molecule and the same cell type in the Output Layer. Of note, the same cell type can be used in different chambers. For instance, the same Input Layer cell responding to x_1 is present simultaneously in chambers ψ_i and ψ_2.

Circuits' scalability:
According to the previous results, the requirements to implement any arbitrary complex Boolean function f involving N different inputs are:

i) A library of engineered cell types that in response to a single input x_j (Input Layer) secrete a molecule α according to either the Identity or the NOT logic (α is expressed in presence of x_j (Identity) or α is expressed in absence of x_j (NOT)). For a N-inputs function, $2N$ different cell types must be engineered (N Identities and N NOTs).

ii) A single cell type that in absence of α expresses the final output (inverted logic of the Output Layer).

iii) Several physically isolated chambers or capsules to allocate the different computational modules (Input and Output Layers). The number of chambers depends on the circuit complexity. Still, this number will be lower than 2^{N-1} which is the upper bound for the number of terms of the simplest canonical form of a Boolean function[5].

Taking all those elements into consideration, the number of different Boolean functions that can be implemented increases as 2^{2^n}, by using a library of $2 \cdot N+1$ cells. Fig. 1d compares, for different number of inputs N, the scalability of the requirements with the number of implementable functions.

Full description of engineered yeast cell library:

Yeast W303 (*ade2-1 his3-11,15 leu2-3,112 trp1-1 ura3-1 can1- 100*) cells were genetically modified to be able to produce *S. cerevisiae* alpha factor from an inducible promoter (Input Layer cells), or to control fluorescent protein expression in response to the pheromone (Output Layer cells). Schematic genotypic characteristics of each cell and plasmid used are summarized in Fig.S3 and in Tables S1 and S2.

Input Layer (IL) cells: Input Layer cells are engineered in pairs; each pair of cells contain a cell that secretes alpha factor in the presence of a given input (Identity logic; ID) and a second cell that secretes alpha factor in the absence of the same input (NOT logic; NOT). IL cells are *MAT*α yeast cells that contain *MF*α*1* and *MF*α*2* genes deleted to avoid endogenous alpha factor expression. *STE3* receptor has also been deleted to prevent mating with *MATa* cells within the circuit. Except for the TetON-TetOFF system, that performs the Identity-NOT logic in response to doxycycline, all the other pairs of cells share the same internal genetic architecture. As for the Identity cells, the wiring molecule (alpha factor) expression is controlled by a specific input inducible promoter (e.g. *GAL1* promoter that responds to progesterone to induce the *MF*α*1* gene).

In NOT cells, alpha factor is constitutively expressed under the control of the engineered promoter *TEF1*i that contains a LacI binding site (*P*$_{TEF1}$*-OplacI-MF*α*1*), and the LacI repressor is transcribed from the same specific input inducible promoter (i.e *GAL1* promoter that responds to progesterone to induce

the repressor of *MFα1* gene expression) (Fig. S4). Cells in the library respond to six different inputs: doxycycline (0.5μg/mL), progesterone (130nM), aldosterone (20μM), *C. albicans* alpha factor (1μM), dexamethasone (128μM) and 17-β-estradiol (20nM).

Cell#IL1 (ID DOX) is a cell that produces alpha factor when doxycycline (DOX) is present in the media. IL1 cells express *MFα1* gene under the control of two TetOperators in the centromeric plasmid *YCpTetO$_2$-MFα1* that also constitutively expresses the reverse Tet Transactivator (rtTA).

Cell#IL2 (NOT DOX) is a cell that produces alpha factor when doxycycline (DOX) is absent in the media. IL2 cells carry the plasmid *pCM183-MFα1* that expresses *MFα1* under the control of two TetOperators, and in addition, expresses the Tet Transactivator (tTA) constitutively.

Cell#IL3 (ID PRO) is a cell that produces alpha factor in the presence of progesterone (PRO). This cell contains the episomal plasmid that expresses the MFα1 gene under the control of the *GAL1* promoter (*pRS424-P_{GAL1}-MFα1*). It also contains the ADGPV integrative plasmid (*pIU-ADGPV*) that expresses the hybrid protein "GPV" under the control of the *ADH1* promoter. The "GPV" construct consists of three domains: the Gal4 DNA binding domain, the human progesterone receptor ligand binding domain and the VP16 activating domain. In the presence of progesterone, cells induce expression of alpha factor.

Cell#IL4 (NOT PRO) is a cell that produces alpha factor in the absence of progesterone (PRO). *MFα1* gene is constitutively expressed under the control of the engineered *TEF1i* promoter (*pRS404-P_{TEFi}-MFα1*). This cell expresses the LacI repressor from the *GAL1* promoter (*pRS403-P_{GAL1}-lacI*) and contains the ADGPV integrative vector (*pIU-ADGPV*) to regulate the expression of the *lacI* from the *GAL1* promoter in response to progesterone. In the presence of progesterone, the LacI repressor is produced and represses the expression of alpha factor.

Cell#IL5 (ID ALD) is a cell that produces alpha factor in the presence of aldosterone (ALD). This cell contains an episomal plasmid that expresses the *MFα1* gene under the control of the *GAL1* promoter (*pRS424-P_{GAL1}-MFα1*). It also contains the ADGMV integrative vector (*pIU-ADGMV*) that expresses the hybrid protein "GMV" under the control of the *ADH1* promoter. The "GMV" construct consists of three domains: the Gal4 DNA binding domain, the human mineralocorticoid receptor ligand binding domain and the VP16 activating domain. In the presence of aldosterone, cells induce expression of alpha-factor.

Cell#IL6 (NOT ALD) is a cell that produces alpha factor in the absence of aldosterone (ALD). The *MFα1* gene is constitutively expressed under the control of the engineered *TEF1i* promoter (*pRS404-P_{TEFi}-MFα1*). This cell expresses the LacI repressor from the *GAL1* promoter (*pRS403-P_{GAL1}-lacI*) and contains the ADGMV integrative vector (*pIU-ADGMV*) to regulate the expression of the lacI from the *GAL1* promoter in response to aldosterone. In the presence of aldosterone, the LacI repressor is produced and represses the expression of alpha factor.

Cell#IL7 (ID αCa) is a cell that produces *S. cerevisiae* alpha factor in the presence of *C. abicans* alpha factor (αCa). This cell contains the integrative plasmid (*pRS404-P_{FUS1}-MFα1*) that expresses the *MFα1* under the control of the *FUS1* promoter. This cell expresses the *C. albicans* pheromone receptor (*CaSTE2*) that has been inserted in the *HIS3* gene locus. In the presence of *C. albicans* alpha factor, this cell triggers *FUS1* promoter transcription to express *S. cerevisiae* pheromone.

Cell#IL8 (NOT αCa) is a cell that produces *S. cerevisiae* alpha factor in the absence of *C. abicans* alpha factor (αCa). *MFα1* gene is constitutively expressed under the control of the engineered *TEF1i* promoter (*pRS404-P_{TEFi}-MFα1*). This cell expresses the LacI repressor from the *FUS1* promoter (*pRS406-P_{FUS1}-lacI*) and contains the *C. albicans* pheromone receptor (*CaSTE2*) inserted in the *HIS3* gene locus. In the presence of *C. albicans* pheromone, this cell triggers *FUS1* promoter transcription to express the LacI repressor and repress the expression of *S. cerevisiae* alpha factor.

Cell#IL9 (ID DEX) is a cell that produces alpha factor in the presence of dexamethasone (DEX). This cell contains the centromeric *pRS416-P_{GPD1}-hGR* vector that constitutively expresses the human glucocorticoid receptor. It also contains the centromeric plasmid *pRS413-HEREminp-MFα1*. In the presence of dexamethasone, the receptor induces the transcription of MFα1 gene under the control of HERE sequences. To prevent leakiness, a reduced version of *GAL1* promoter containing the *Mig2 UAS* was used as minimal promoter (*minp*). In the presence of dexamethasone cells induce expression of alpha factor.

Cell#IL10 (NOT DEX) is a cell that produces alpha factor in the absence of dexamethasone (DEX). *MFα1* gene is constitutively expressed under the control of the engineered *TEF1i* promoter (*pRS404-P_{TEFi}-MFα1*). This cell expresses the LacI repressor under the control of HEREminp construct (*pRS413-HEREminp-lacI*). It also contains the integrative *pRS406-P_{GPD1}-hGR* vector. In the presence of dexamethasone, the human glucocorticoid receptor (hGR) induces the transcription of the LacI repressor to block expression of alpha factor.

Cell#IL11 (ID EST) is a cell that produces alpha factor in the presence of 17-β-estradiol (EST). This cell contains the episomal plasmid *pRS424-P_{GAL1}-MFα1* that expresses the *MFα1* gene under the control of the *GAL1* promoter. It also contains the ADGEV integrative vector (*pIU-ADGEV*) that expresses the hybrid protein "GEV" under the control of the *ADH1* promoter. The "GEV" construct consists of three domains: the Gal4 DNA binding domain, the human estradiol receptor ligand binding domain and the VP16 activating domain. In the presence of 17-β-estradiol cells induce expression of alpha factor.

Cell#IL12 (NOT EST) is a cell that produces alpha factor in the absence of 17-β-estradiol (EST). *MFα1* gene is constitutively expressed under the control of the engineered *TEF1i* promoter (*pRS404-P_{TEFi}-MFα1*). This cell expresses the LacI repressor from the *GAL1* promoter (*pRS405-P_{GAL1}-lacI*) and contains the ADGEV integrated vector to regulate the expression of the lacI from the *GAL1* promoter in response to 17-β-estradiol. In the presence of 17-β-estradiol the LacI repressor is produced and represses the expression of alpha factor.

Output Layer (OL) cells: Cells in the Output Layer (OL) are designed to perform a NOT logic: in the absence of the wiring molecule (alpha factor) they express a fluorescent ssrA tagged reporter protein[11] (yEGFP or mCherry). In the presence of alpha factor they down regulate the expression of the reporter protein and induce their degradation using the ClpX/ClpP protease system that targets the ssrA tag[11] (Fig. S3 and S4). Cells are mating type MATa and *BAR1* was deleted to increase the sensitivity to pheromone.

Cell#OL1 (NOT GFP) is a cell that produces yEGFP$^{\mathrm{ssrA}}$ in the absence of alpha factor. yEGFP$^{\mathrm{ssrA}}$ gene is constitutively expressed under the control of the *TEF1i* promoter (*pRS404-P_{TEF1i}-yEGFP$^{\mathrm{ssrA}}$*) and the LacI repressor is transcribed from *FUS1* promoter (*pRS405-P_{FUS1}-lacI*). In the presence of alpha factor, the LacI repressor is produced and represses the expression of yEGFP$^{\mathrm{ssrA}}$. This cell constitutively expresses the protease subunit ClpP under the *ADH1* promoter (*met1::P_{ADH1}-ClpP-KanMX*). The ClpX subunit is transcribed from the *FUS1* promoter (*pRS406-P_{FUS1}-ClpX*), which is induced only in the presence of *S. cerevisiae* alpha factor. When both subunits assemble the protease complex, yEGFP$^{\mathrm{ssrA}}$ is degraded. This cell contains the mCherry inserted in the *ENO1* locus (*ENO1::mCHERRY-HphNT*). The mCherry expression serves to follow OL1 cells (mCherry-positive) from Input Layer cells (mCherry-negative) when mixed.

Cell#OL2 (NOT mCherry) is a cell that produces mCherry$^{\text{ssrA}}$ in the absence of alpha factor. mCherry$^{\text{ssrA}}$ is constitutively expressed under the control of the *TEF1i* promoter (*pRS404/pRS405-*P_{TEF1i}*-mCHERRY*$^{\text{ssrA}}$; two copies of mCherry are needed to be able to assess fluorescence with a higher degree of confidence). The LacI repressor is transcribed from the *FUS1* promoter (*pRS403-*P_{FUS1}*-lacI*). In the presence of alpha factor the LacI repressor is produced and represses the expression of *mCHERRY*$^{\text{ssrA}}$. This cell constitutively expresses the protease subunit ClpP under the *ADH1* promoter (*met1::*P_{ADH1}*-ClpP-KanMX*). The ClpX subunit is transcribed from the *FUS1* promoter (*pRS406-*P_{FUS1}*-ClpX*), which is induced only in the presence of alpha factor. mCherry$^{\text{ssrA}}$ is degraded when both subunits assemble the protease complex. This cell expresses the fluorescent protein YFP inserted in the *ENO1* locus (*ENO1::YFP-HphNT*) to distinguish OL2 cells (YFP-positive) from the Input Layer cells (YFP-negative) when mixed.

Supplementary Tables

Table S1. Yeast strains used in this study

Name	Genotype[#]	Source
IL1	*MATα ste3::HIS3 mfα1::LEU2 mfα2::KanMX YCpTetO$_2$MFα1*	[10]
IL2	*MATα ste3::HIS3 mfα1::LEU2 mfα2::KanMX pCM183-MFα1*	[10]
IL3	*MATα ste3::NatNT mfα1::LEU2 mfα2::KanMX pRS424-P$_{GAL1}$-MFα1 pIU-ADGPV*[*]	this study
IL4	*MATα ste3::NatNT mfα1::LEU2 mfα2::KanMX pRS404-P$_{TEF1i}$-MFα1 pRS403-P$_{GAL1}$-lacI pIU-ADGPV*[*]	this study
IL5	*MATα ste3::NatNT mfα1::LEU2 mfα2::KanMX pRS424-P$_{GAL1}$-MFα1 pIU-ADGMV*[**]	this study
IL6	*MATα ste3::NatNT mfα1::LEU2 mfα2::KanMX pRS404-P$_{TEF1i}$-MFα1 pRS403-P$_{GAL1}$-lacI pIU-ADGMV*[**]	this study
IL7	*MATα ste3::HIS3 mfα1::LEU2 mfα2::KanMX pRS404-P$_{FUS1}$-MFα1 his3::P$_{GPD1}$-CaSTE2-HphNT*	this study
IL8	*MATα ste3::HIS3 mfα1::LEU2 mfα2::KanMX pRS404-P$_{TEF1i}$-MFα1 pRS406-P$_{FUS1}$-lacI his3::P$_{GPD1}$-CaSTE2-HphNT*	this study
IL9	*MATα ste3::NatNT mfα1::LEU2 mfα2::KanMX pRS413-HEREminp-MFα1 pRS416-P$_{GPD1}$-hGR*	this study
IL10	*MATα ste3::NatNT mfα1::LEU2 mfα2::KanMX pRS404-P$_{TEF1i}$-MFα1 pRS413-HEREminp-lacI pRS406-P$_{GPD1}$-hGR*	this study
IL11	*MATα ste3::NatNT mfα1::LEU mfα2::KanMX pRS424-P$_{GAL1}$-MFα1 pIU-ADGEV*	this study
IL12	*MATα ste3::HIS3 mfα1::LEU2 mfα2::KanMX pRS404-P$_{TEF1i}$-MFα1 pRS406-P$_{GAL1}$-lacI pIU-ADGEV*	this study
OL1	*MATa bar1::NatNT met1::P$_{ADH1}$-clpP-KanMX pRS406-P$_{FUS1}$-clpX pRS405-P$_{FUS1}$-lacI pRS404-P$_{TEF1i}$-yEGFP^{ssrA} ENO1-mCHERRY-HphNT*	this study
OL2	*MATa bar1::NatNT met1::P$_{ADH1}$-clpP-KanMX pRS406-P$_{FUS1}$-clpX pRS403-P$_{FUS1}$-lacI pRS404-P$_{TEF1i}$-mCHERRY^{ssrA} pRS405-P$_{TEF1i}$-mCHERRY^{ssrA} ENO1-YFP-HphNT*	this study

Strain background W303 (*ade2-1 his3-11,15 leu2-3,112 trp1-1 ura3-1 can1-100*)

Table S2. Plasmids used in this study

Name	Description	Source
pRS406-P_{FUS1}-clpX	Integrating *URA3 p $_{FUS1}$-clpX*	this study[11]
pRS405-P_{FUS1}-lacI	Integrating *LEU2 P_{FUS1}-lacI*	this study
pRS403-P_{FUS1}-lacI	Integrating *HIS3 P_{FUS1}-lacI*	this study
pYM-N6-P_{ADH1}-clpP-KanMX	Integrated at *MET1locus P_{ADH1}-clpP* Geneticin resistance	this study[11, 12]
pRS404-P_{TEF1i}-yEGFP^{ssrA}	Integrating *TRP1 P_{TEF1}-OplacI-yEGFP^{ssrA}*	this study[11]
pRS404-P_{TEF1i}-mCHERRY^{ssrA}	Integrating *TRP1 P_{TEF1}-OplacI-mCHERRY^{ssrA}*	this study[11]
pRS405-P_{TEF1i}-mCHERRY^{ssrA}	Integrating *LEU2 P_{TEF1}-OplacI-mCHERRY^{ssrA}*	this study[11]
pBS35-mCHERRY-HphNT	Integrated at *ENO1* terminator *mCHERRY* HygromycinB resistance	this study[12]
pYM40-YFP-HphNT	Integrated at *ENO1* terminator *YFP* HygromycinB resistance	this study[12]
YCpTetO$_2$MFα1	CEN *TRP1* reverse tTA transactivator *TetO$_2$MFα1*	[10]
pRS424-P_{GAL1}-MFα1	2 micron *TRP1 P_{GAL1}-MFα1*	this study
pRS404-P_{FUS1}-MFα1	Integrating *TRP1 P_{FUS1}-MFα1*	this study
pRS413-HEREminp-MFα1	CEN *HIS3 HEREminp-MFα1*	this study[13]
pCM183-MFα1	CEN *TRP1* tTA transactivator *TetO$_2$MFα1*	[10]
pRS404-P_{TEF1i}-MFα1	Integrating *TRP1 P_{TEF1}-OplacI-MFα1*	this study
pRS403-P_{GAL1}-lacI	Integrating *HIS3 P_{GAL1}-lacI*	this study
pRS406-P_{GAL1}-lacI	Integrating *URA3 P_{GAL1}-lacI*	this study
pRS406-P_{FUS1}-lacI	Integrating *URA3 P_{FUS1}-lacI*	this study
pRS413-HEREminp-lacI	CEN *HIS3 HEREminp-lacI*	this study[13]
pFA6a-hphNT1-P_{GPD1}-CaSTE2	Integrated at *HIS3 locus P_{GPD1}-CaSTE2* HygromycinB resistance	this study[12, 14]
pIU-ADGPV	Integrating *URA3 P_{ADH1}-[GAL4DBD-hPR LBD-VP16]***	this study
pIU-ADGMV	Integrating *URA3 P_{ADH1}-[GAL4DBD-hMR LBD-VP16]****	this study
pIU-ADGEV	Integrating *URA3 P_{ADH1}-[GAL4DBD-hER LBD-VP16]*	[15]
pRS416-P_{GPD1}-hGR	CEN *URA3 P_{GPD1}-hGR*	this study[13]
pRS406-P_{GPD1}-hGR	Integrating *URA3 P_{GPD1}-hGR*	this study[13]

*The hPR ligand binding domain stands from aminoacid 655 to 933.

**The hMR ligand binding domain stands from aminoacid 705 to 984.

Supplementary Figures

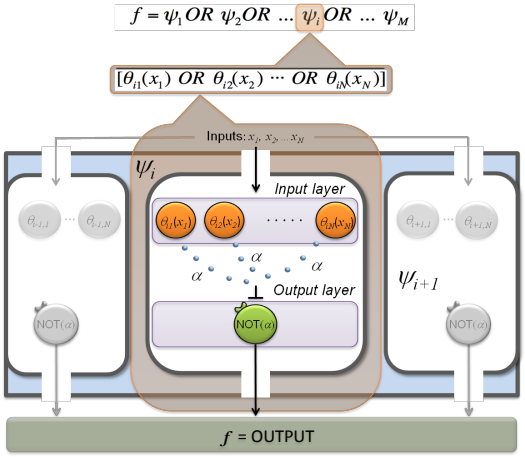


Figure S1. Schematic representation of a multicellular implementation of an arbitrary Boolean function by using the ILF. The circuit is composed of several modules ψi. Each module can produce the final output. The Input Layer is composed of several cells that sense the external inputs and respond according to Identity or NOT logic. These cells secrete a wiring molecule α that prevents the expression of the final output in the Output Layer. Here each logic module ψi is located in a different chamber of an ideal device, i.e. a microfluidics device. The total circuit is composed of several chambers without cross-talk, hence, the same wiring molecule α can be used in each chamber.

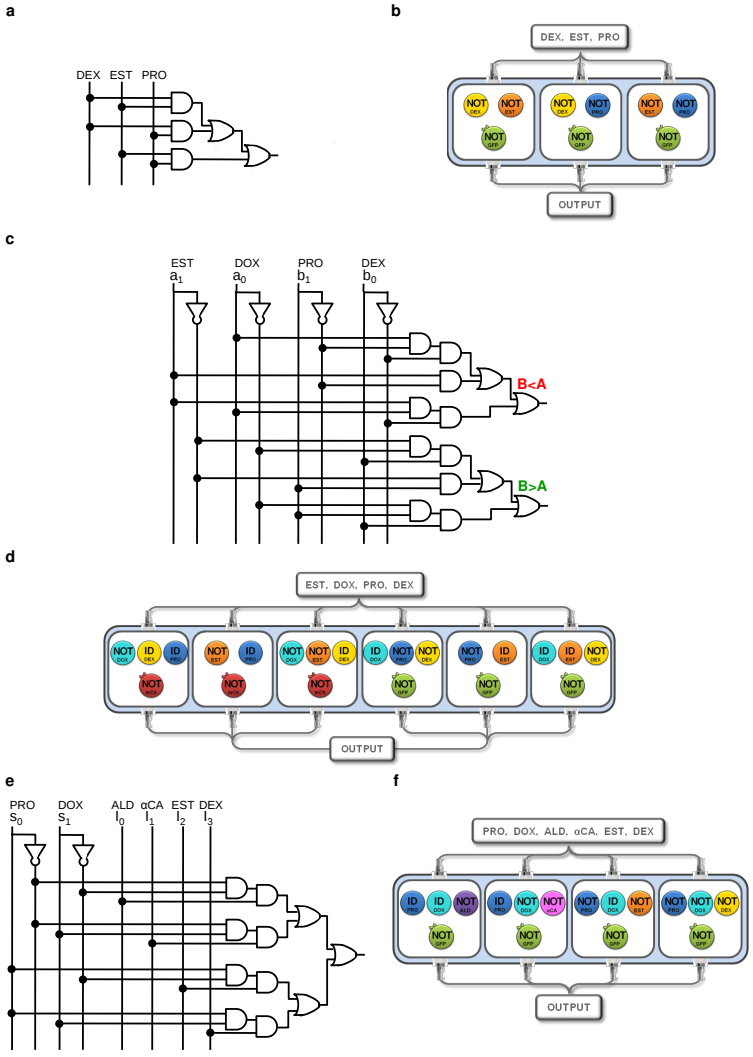


Figure S2. Comparison between biological and electronic implementation of logic circuits. a,Electronic design of a Majority Rule. This circuit involves 3 AND gates, 2 OR gates and 4 different wires. **b**, Biological implementation of a Majority Rule circuit involves 4 different cell types distributed in 3 chambers. **c**, Electronic design of a 4-inputs comparator. This circuit involves 10 AND gates, 4 OR gates, 4 NOT gates and 24 different wires. **d**, Biological implementation of a 4-inputs comparator circuit involves 10 different cell types distributed in 6 chambers. **e**, Electronic design of a MUX4to1. This circuit involves 8 AND gates, 3 OR gates, 2 NOT gates and 18 different wires. **f**, Biological implementation of a MUX4to1 circuit involves 9 different cell types distributed in 4 chambers. All computations were performed by cells implementing a one-input, one-output logic function and using only one wiring molecule. Of note, the ILF design allows a vast reduction for the implementation of biological devices.

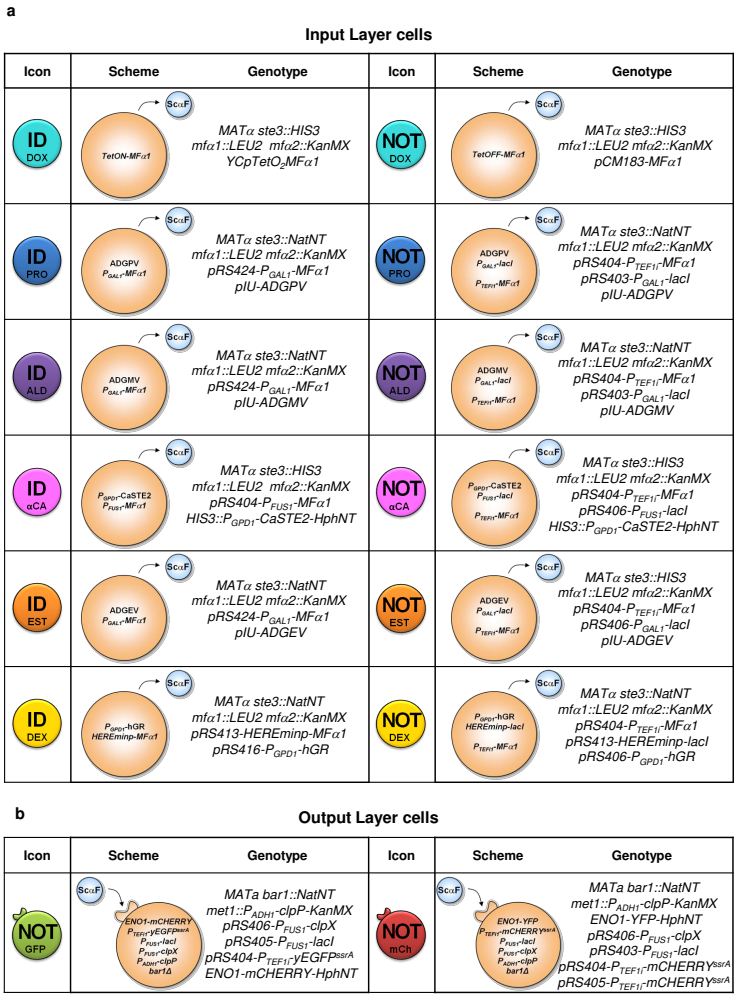


Figure S3. Complete description of the engineered yeast cell library. (a) Schematic representation and basic genetic information of the Input Layer cells. Cells in the library respond to six different inputs (DOX; doxycycline, PRO; progesterone, ALD; aldosterone, αCa; *C. albicans* alpha factor, EST; 17-β-estradiol, DEX; dexamethasone) with two different logics. In the presence of the input, Identity cells (ID, left) express *S. cerevisiae* α factor, whereas NOT cells (NOT,right) repress pheromone production in response to stimuli. (b) Schematic representation and basic genetic information of the two Output Layer cells used. The cells sense *S. cerevisiae* alpha factor and shut down the expression of a fluorescent protein (GFP, left; mCherry, right). All cells are W303 derivatives (see Supplementary Information for complete cells genotype).

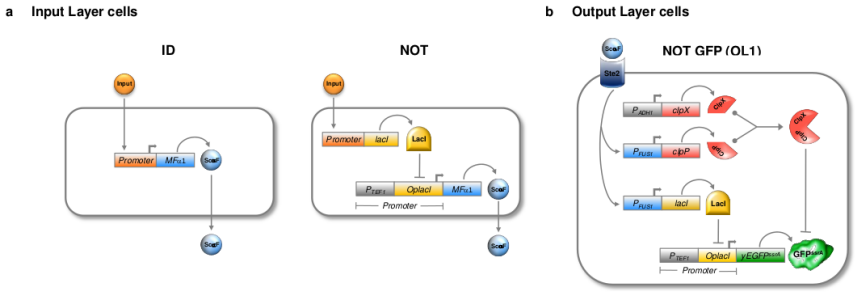


Figure S4. Graphic representation of the architecture of the engineered cells. **(a)** Generic description of the Input Layer cells. Identity cells (ID; left) express alpha factor under the control of an input-inducible promoter. NOT cells (NOT; right) constitutively express the pheromone under the control of a modified TEF1 promoter (TEF1-OplacI). In the presence of the input, alpha factor expression is repressed by LacI. **(b)** Output Layer cell description. An ssrA tagged version of yEGFP was expressed under the control of the TEF1i promoter. The ssrA tag allows the Clp protease complex to recognize the yEGFPsrrA and induce its degradation. The protease subunit ClpP was constitutively transcribed under the ADH1 promoter, whereas the ClpX subunit and the LacI repressor were expressed under the control of the FUS1 promoter, which is induced by alpha factor.

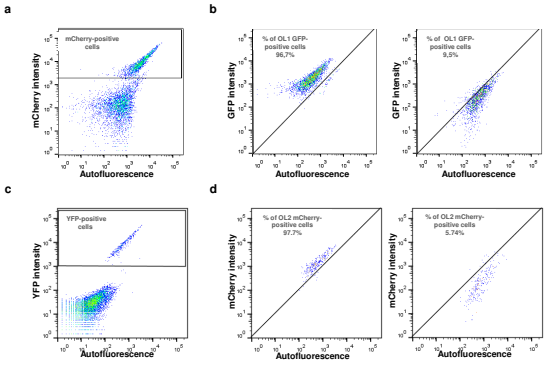


Figure S5. Representative FACS analysis using quantitative single cell output. Fluorescence from Output Layer cells was assessed by flow cytometry. A total of 10.000 cells were analyzed.**(a)** Panel shows mCherry intensity (Y axis) versus autofluorescence (X axis) and allow to select the OL1 cells (mCherry positive) from the Input Layer cells (mCherry negative). **(b)** Selected OL1 mCherry cells were analyzed by their GFP expression (Y axis) versus autofluorescence (X axis). Two examples are given: a GFP positive sample (left) and GFP negative one (right). **(c)** OL2 cells are analyzed as in **a**, using the YFP channel to select them from the Input Layer cells. **(d)** Selected YFP cells were assessed by their mCherry expression.

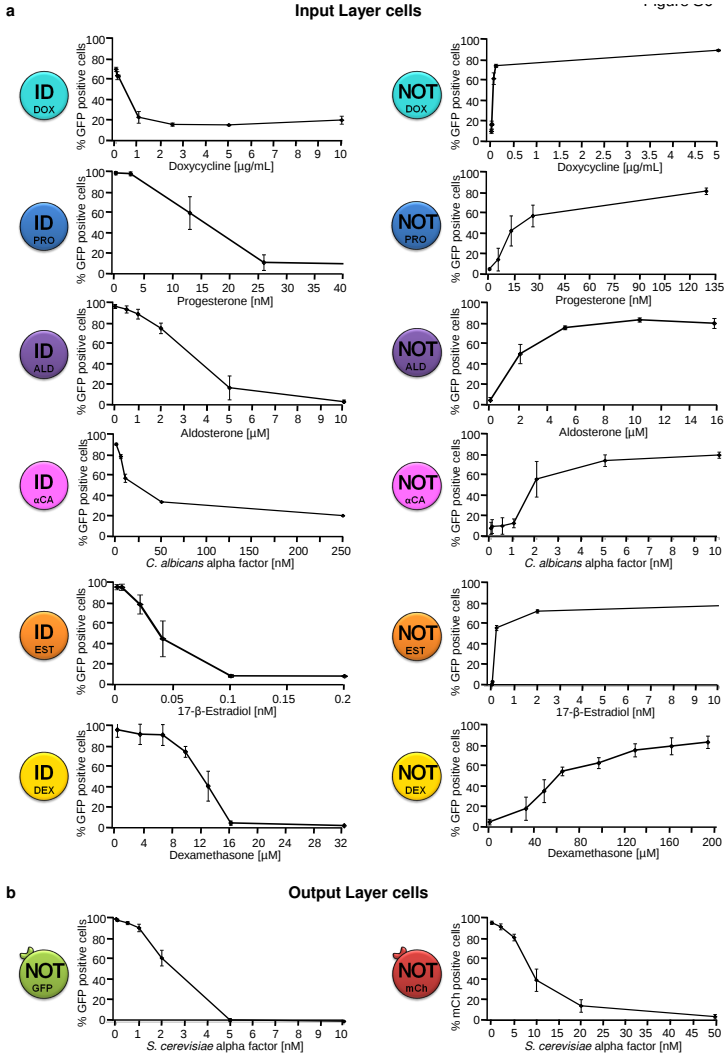


Figure S6. Transfer Function analyses of the engineered cell library. (a) Input Layer cells were mixed with the Output Layer GFP cells and treated with different inputs concentrations. Samples were incubated for 4h at 30oC and analyzed by FACS. Data are expressed as the percentage of GFP positive cells and represent the mean and standard deviation of three independent experiments. **(b)** Output Layer cells were incubated with different concentrations of S. cerevisiae alpha factor and analyzed as in **a**.

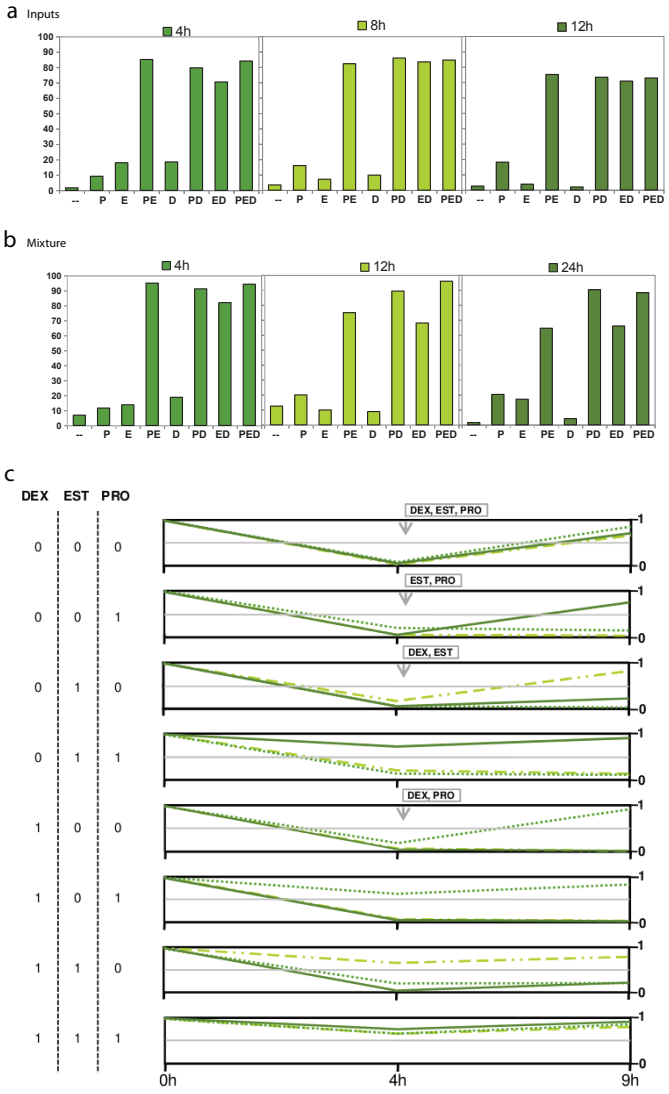


Figure S7. Stability and dynamics of MR circuit. **a** Cells were mixed proportionally with each input combination. Samples were quantified 4, 8 or 12 hours later. **b** IL cells were mixed proportionally and cultured for 0, 8 or 20 hours. Inputs and OL cells were added 4 hours before quantification. (E=EST; P=PRO; D=DEX) **c** MR circuit was quantified initially (0h) and at its final computation time (4h). Different inputs were then, added to the turned off samples. The circuit was quantified five hours later. Percentage of GFP positive cells is expressed in 1-0 scale.

References to SI

1. Kwok,R. Five hard truths for synthetic biology. Nature 463, 288-290 (2010).

2. Siuti,P., Yazbek,J., & Lu,T.K. Synthetic circuits integrating logic and memory in living cells. Nat. Biotechnol. 31, 448-452 (2013).

3. Bonnet,J., Yin,P., Ortiz,M.E., Subsoontorn,P., & Endy,D. Amplifying genetic logic gates. Science 340, 599-603 (2013).

4. Enderton,H. & Enderton,H.B. A mathematical introduction to logic(Access Online via Elsevier,2001).

5. Bender,E.A. & Williamson,S.G. A short course in discrete mathematics (DoverPublications. com,2012).

6. Hill,F.J. & Peterson,G.R. Introduction to switching theory and logical design(John Wiley & Sons, Inc.,1981).

7. Karnaugh,M. The map method for synthesis of combinational logic circuits. American Institute of Electrical Engineers, Part I: Communication and Electronics, Transactions of the 72, 593-599 (1953).

8. McCluskey,E.J. Introduction to the theory of switching circuits(McGraw-Hill NewYork,1965).

9. Macia,J., Posas,F., & Sole,R.V. Distributed computation: the new wave of synthetic biology devices. Trends Biotechnol. 30, 342-349 (2012).

10. Regot,S. et al. Distributed biological computation with multicellular engineered networks. Nature 469, 207-211 (2011).

11. Grilly,C., Stricker,J., Pang,W.L., Bennett,M.R., & Hasty,J. A synthetic gene network for tuning protein degradation in Saccharomyces cerevisiae. Mol. Syst. Biol. 3, 127 (2007).

12. Janke,C. et al. A versatile toolbox for PCR-based tagging of yeast genes: new fluorescent proteins, more markers and promoter substitution cassettes. Yeast 21, 947-962 (2004).

13. Miller,C.A., III, Tan,X., Wilson,M., Bhattacharyya,S., & Ludwig,S. Single plasmids expressing human steroid hormone receptors and a reporter gene for use in yeast signaling assays. Plasmid 63, 73-78 (2010).

14. Janiak,A.M. et al. Functional expression of the Candida albicans alpha-factor receptor in Saccharomyces cerevisiae. Fungal. Genet. Biol. 42, 328-338 (2005).

15. Louvion,J.F., Havaux-Copf,B., & Picard,D. Fusion of GAL4-VP16 to a steroid-binding domain provides a tool for gratuitous induction of galactose-responsive genes in yeast. Gene 131, 129-134 (1993).

Results

Implementation of a modular spatial computation with inverted logic to construct complex circuits with only one wire

The main problem to build scalable and reusable modular biological devices is that the number of different proteins required to implement a circuit is proportional to its complexity. Our efforts have focused on solving the so called wiring problem. In our previous study [Regot et al., 2011], we presented a method to design circuits, which reduced the wiring requirements by distributing the computation and the output production into several cellular types. In this second work, we have gone one step further and propose a new architecture that, with simpler logic gates, allows for higher scalability with bounded requirements of wiring and cell types. This method, also introduces spatial segregation of the cells as a new computational feature.

The new method is based on translating the minimal canonical form of a function f (already simplified using standard practices as Karnaugh maps and Morgan's laws) to a boolean expression involving only `OR` and `NOT` operations. That is, the `IDENTITY`, `NOT` and `OR` logic gates constitute a functionally complete set. The final function is a sum (`OR`) of different negated terms (`NOT`) which are the computational modules of the circuit. Each of those modules can be systematically translated into two–layered cellular circuits. The Input Layer consist in a combination of cell types sensing external inputs and producing the same wiring molecule following an `IDENTITY` or `NOT` logic. The Output Layer is composed by a single cell type that implements a `NOT` logic: in presence of the secreted wiring molecule, the output is repressed whereas in its absence, it is expressed. Of note, logic gates that respond to a single input (`IDENTITY` and `NOT`) need less engineering than two–input logic gates. Additionally, `OR` logic in a cellular consortia can be implemented taking into account the media, because when two different cell types secrete the wiring molecule into the media, the receptor cell responds to it no matter which cell was the source of it. We used again distributed output

computation as before [Regot et al., 2011] i.e, the final output of the function is the result of all modules according to an OR logic. That is, when any module equals to 1, the final output is one. Or the other way around, the final output is zero only when all modules equal to zero. We named this methodology as Inverted logic Formulation (ILF; Fig1 a–c; Fig S1).

Since each module is isolated (space is considered by implementing each module in a restricted space, i.e in different chambers), all modules can use the same wire. Therefore, space separation is the key to solve the wiring problem. The number of different modules required for a given circuit, depends on the complexity of the Boolean function, but for n–inputs function will be, at most, $M = 2^n/2$. The number of different cells required is also bounded but to $2n + 1$. Importantly, the number of modules (chambers or wiring molecules) or cell types needed grow at slower rate than the number of implementable n–input functions (2^{2^n}; Fig1 d). This fixed boundaries are, indeed, the most important property of this new architecture, ensuring the scalability towards complex circuits (Fig S2).

Cellular segregation of logic gates enable reusing the same genetic elements to engineer a cell library

The *in vivo* implementation of the ILF requires to engineer cells able to produce a wiring molecule in response of an external input (Input Layer cells; IL), or to control a fluorescent protein expression in response to the wiring molecule (Output Layer cells; OL).

We created a Input layer (IL) cell library consisting of six pairs of cells that respond to six inputs but performing IDENTITY or NOT logic functions to produce a wiring molecule. We chose *S.cerevisiae* pheromone to implement the wire as in [Regot et al., 2011] (Fig2 a). As the spatial segregation of the independent modules requires only one wire, we have implemented the *C.albicans* pheromone (αCa) as an input. Additional inputs were hormones whose receptors have already been

described (i.e. 17–β–estradiol, progesterone and aldosterone). Briefly, IDENTITY cells express alpha factor under the control of an inducible promoter: the TetON for doxycycline, the *FUS1* promoter for αCa, HERE elements for dexamethasone and the *GAL1* promoter for 17–β–estradiol, progesterone and aldosterone. The corresponding NOT cell expresses the lacI repressor under the same inducible promoter. All Input layer NOT cells constitutively express the wiring protein under an engineered *TEF1i* promoter that is repressed in the presence of lacI (Fig S4a).

The Output layer (OL) cell consist in a NOT cell able to respond to the wiring molecule and repress expression of a fluorescent reporter (the output) in its presence. To prevent transcription we used the same genetic elements as in the NOT Input layer cells. The *TEF1i* promoter constitutively produces the output and the inducible *FUS1* promoter, activated by alpha factor, controls lacI expression. The yEGFP used as output of the system is a very stable protein with a half–life that may exceed 24h [Clontech, 2001]. Therefore, to obtain a dynamic response, we engineered a strategy to reduce the half life of the protein. The yEGFP was tagged with the ssrA sequence and the Clp *E.coli* protease complex was cloned under the control of the *FUS1* promoter (Fig S4b). Thus, when alpha factor is present, Clp proteases are expressed and yEGFPssrA protein is degraded (Fig S4b). Using this system the half–life of the protein is reduced to approximately 2h [Grilly et al., 2007].

Therefore, the essential genetic elements to construct the whole library are only: a set of inducible promoters, a constitutive repressible promoter, its repressor protein, one wiring molecule and the desired output protein. (Fig S3 and Supplementary materials contain a complete description of the engineered yeast cell library).

The use of two fluorescent proteins enables accurate output quantification of circuits

The final computation in each circuit's module (chamber) depends on the response of Output Layer (OL) cells. Thus, we needed to distinguish OL

cells from Input Layer (IL) cells. Therefore, we tagged these cells with a constitutive different florescent protein. This allowed to quantify the percentage of positive cells in every chamber (i. e. the proportion of OL cells population that expresses the output fluorescence; Fig 2b and S5). From the percentages of positive cells of all chambers, the maximum, is considered the final computation of the circuit.

Engineered yeast cells library responds to 6 inputs without crosstalk

To engineer a library of yeast cells able to operate with several inputs, each IL cell has to be sensitive only to an specific input, without crosstalk with other inputs, otherwise, the misleading connections would produce an incorrect final computation. To demonstrate that the six different inputs of the library were suitable for computing together, we exposed each single IL cell to all different inputs. All IL cells, responded with the appropriate logic to its specific input (Fig 2c).

As already described, a digital like response is essential for biological computation to avoid noise propagation or signal diffusion. To confirm that the library exhibits a digital response, we calculated the magnitude of output produced in a range of input concentrations. As can be observed in Figure S6, every cell performed a step like shape transfer curve that prevented overlap between the "on" and "off" state. The input concentration in which the `IDENTITY` cells produce enough wiring molecule (to totally repress the production of output from OL1 cells) is always lower than the input concentration in which the `NOT` cells represses enough the wiring molecule secretion A higher input concentration triggers the repressor production strongly, reaching quickly the lacI concentration to repress the alpha factor transcription. Thus, the amount of alpha factor released into the media is lower enough to prevent OL1 cells to turn off the production of output.

Circuits can be implemented by combination of different cells from the library

For illustrative purposes, we implemented a three input circuit known as Majority Rule (MR; the output equals to 1 in the presence of two or more inputs). The implementation just required three cells from the IL library together with an inverter cell (OL1) and three chambers (Fig 2d). Each chamber contain a different pair of IL `NOT` cells plus a OL1 cell. Thus, any combination of two inputs results in inhibition of alpha factor production from both IL cells contained in a chamber and, hence, the OL of this chamber maintains the production of the output (Fig 2d). The *in vivo* results clearly showed the correct circuit response (Fig. 2e) and, moreover, the 0 logic state account for less than 20% of output positive cells whereas for 1 logic the percentage is higher than 60%. This resolution is comparable to the TTL electronic technology (where acceptable output signal voltages range from 0 volts to 0.5 volts for a low state, and 2.7 volts to 5 volts for the high state).

Of note, although we initially implemented a Majority Rule for dexamethasone, 17–β–estradiol and progesterone (as inputs), by combining other cells from the library alternative Majority Rule circuits responding to other inputs were implemented. By contrast, if the MR had been implemented in a single cell [Gaber et al., 2014], computing different inputs would have required much more genetic manipulation.

In multicellular distributed computation several strains must coexist in one culture, thus, different growing rates among them might eventually impair circuit response. We assessed the dynamics of the MR circuit and we found that its response was stable for up to 12 hours in the presence of input signals (Fig S7a). Furthermore, when IL cells of the circuit were mixed and the culture was kept at log phase for ten generations, the circuit still maintained the competence to respond properly (Fig S7b).Additionally, MR circuit was exposed to dynamic input stimulation; and found that the circuit was able to dynamically respond to input variation (Fig S7c).

RESULTS 2

Combinatorial modular organization permits to scale the complexity of biologic circuits.

To demonstrate that the modular spatial inverted logic design permits scaling up circuit complexity, we designed and implemented two complex circuits: the 4–input comparator and a multiplexor–4to1.

A **comparator** circuit, serves to compare two numbers (A and B) coded in binary, and respond if $A > B$ or $A < B$ being a two–output function. Additionally when both outputs are 0 it can be concluded that $A = B$. Our comparator device received two digit numbers, namely a_1 and a_0 for A (implemented by 17–β–estradiol and doxycycline respectively) and b_1 and b_0 for B (implemented by progesterone and dexamethasone). That is why it is called 4–input comparator. Such comparator is, indeed, a complex circuit. In fact, even when simplified at maximum, using the standard electronic design the circuit involves 10 `AND` gates, 4 `OR` gates, 4 `NOT` gates and 24 different wires (Fig S2c). Here, the biological implementation only required 8 different IL cells, 2 distinct OL cells and 6 spatial modules (Figure 3a), demonstrating that ILF lead to a huge reduction of circuit requirements.

As shown in Fig 3b, this circuit responded as expected and preserved the same resolution than the small Majority Rule.

A **multiplexor** (MUX4to1) is a circuit that receives four input signals (I0–I3) and forwards a single response based in the selection of one of them (S0–S1). In total, it is a six–input one–output circuit. The implementation of a circuit with such complexity has never been reported to date (Fig 4a).

The high complexity of the MUX4to1 circuit can be illustrated by the electronic design, which involves 8 `AND` gates, 3 `OR` gates, 2 `NOT` gates and 18 different wires (Fig S2e). Again, the ILF permitted vast simplification making implementation of this biological device possible. The MUX4to1 was performed using only 9 different cell types distributed in 4 chambers (Fig 4b). Each chamber, contained two different IL cells to sense two selector inputs (progesterone and doxycycline). Thus, for each

combination of selector inputs (00, 01, 10, 11), only the selected chamber prevents the wiring molecule production from both: progesterone and doxycycline IL cells. The Input layer of each chamber was completed with a NOT cell that sensed one of the inputs (aldosterone for I_0, candida α–factor for I_1, 17–β–estradiol for I_2 and dexamethasone for I_3). Thus, in the selected chamber, the presence of the input I_n inhibited the remaining production of alpha factor, and the Output Layer cell of that chamber maintained the GFP expression.

The *in vivo* implementation of the circuit resulted in the correct output for all 64 possible combinations of inputs (Fig 4c).

Therefore, the results presented here indicate that using spatial modular biocomputing confers a virtually unlimited scalability. In fact, the library presented in this work, that responds to six inputs, should be enough to create up to $1.8\,10^{19}$ different circuits using, even for the most complex scenairo, no more than 32 spatial modules.

Discussion

The main challenge in the field of synthetic biology is to attain flexible scalable designs to assemble complex functionalities. Although the question has been addressed in many different ways, the design of circuits constrained to the electronic standards has always lead to designs with poor reusability and big limitations towards scalability. Here, we proposed a modular design, that based on inverted logic and spatial segregation, is specially suitable for reaching the scalability required to build living "chips". We demonstrated that only with a library of simple cells (IDENTITY and NOT logic gates) sensing n inputs and communicated by one wiring molecule are sufficient to construct all 2^{2^n} possible functions. Again, this architecture permits maximal complexity with minimal wiring requirements.

Here, the spatial confinement of cells is the key to permit the use of a single wire, becoming a solution to the wiring problem. Fortunately, in recent times, many possible micro–environments have been improved,

such as microfluidic devices [Melin and Quake, 2007], cell microcapsules [Ausländer et al., 2012] or cell cultures patterns [Ricoult et al., 2012]. The microfluidic devices integrate isolated chambers, hence, the crosstalk between modules by diffusion of secreted wiring molecule is prevented. On the other hand, we are still far of being able to use microfuidic devices in applications to the environment or in a living body. Conversely, the micocapsules have already been used successfully as synthetic implants [Rőssger et al., 2013]. Unfortunately, they have not been prepared, yet, to be isolated spaces (to prevent wiring diffusion). Though, one might speculate that this could be solved in the future, for example, engineering a double layer capsule, with a specific protease anchored in the inter–layer space to degrade the wiring molecule before it could escape from the microcapsule. In this scenario, the capsules will serve as isolated modules.

Nevertheless, if spatial restriction were impossible, the cells of different modules could be re–engineered to communicate with different wiring molecules. Then, although all cells would share the same space, the individual computations of the modules would remain independent. The MUX4to1 that respond to six inputs, would only need four wires. Not only that, with eight wires all 4–input functions ($\approx 6.5\,10^4$) could be implemented, which could be hard to engineer, but still within the range of feasible connectors using molecules such as lactones or yeast pheromones [Wynendaele et al., 2013; Coelho et al., 2011].

Another important issue to be discussed is the variation in dose response, due to output distribution. Distinct combination of inputs can provide an output production from different number of chambers. However the nature of the application, might require a constant final output concentration. To solve that, all computational chambers could be connected to a new final collecting chamber. Here, the OL cells should be re–engineered to not produce the final output but a second wiring molecule. A new `IDENTITY` cell, to be placed at the collecting camber, should be engineered to respond to this second wiring and produce the final output. This cell must reach its maximum production (the desired

final output concentration), with the amount of second wiring produced by only one chamber. Thus, although the second wiring was produced in more than one chamber, the final output concentration will not change.

Additional layers of control can expand the repertoire of possible computations of a circuit by selective inhibition of some cells calling for reprogrammable cell circuits.
On the hand, an exhaustive use of spatial segregation: in a "chip" with 2^n chambers, each one of which containing an OL cell and one of the possible combinations of n IL cells, would have the potential to perform all possible n–input functions. Then, the ILF design of a particular n–input function could be performed in this "chip" by inhibiting the OL cell in not required chambers. On the other hand, the number of chambers of the "chip" could be reduced to $2^n/2$ by re–engineering each pair of `IDENTITY-NOT` cells of the IL in just one cell. These cells will perform a `XOR` logic with the external stimuli and a new reprogramming signal as an input. Of note, in theory, all ILF circuits could be also reprogrammed by using receptor antagonists or kinase inhibitors. Therefore, the ILF architecture is both, scalable and easy to reprogram.

Personal contribution to this work: except for the development of the theoretical framework, which was performed by Dr. Macia, I have been fully involved in the design of cells and circuits, their *in vivo* implementation as well as the discussion of the results.

GLOBAL DISCUSSION

The nature of biology makes reusability and assembly of logic circuits more difficult than in electronics. The main obstacle to real scalability is the connectivity among biological entities, the so called "wiring problem". Two different circuit designs are presented in this thesis. Both, showed that a multicellular distributed computation together with distributed output reduced the number of cell types and wiring requirements that are required for circuit assembly, when compared to standard methodologies.

The first design [Regot et al., 2011] is based on a **multi–layer** architecture. A boolean function is expressed as a sum of terms, each of them, implemented with a set of cells. With the restriction that the connection among sets is not allowed, each set contains: (i) a first cell that responds to a external input with an `IDENTITY` or `NOT` logic (ii) cells of intermediate layers that respond to two inputs: a wiring molecule and a external input with `AND` or `N-IMPLIES` logic; and (iii) a final cell able to produce the output. Each upper layer cell produces a specific wiring molecule to communicate with its immediate subsequent layer. With this design, only three cells are enough to perform 128 function of 3–inputs. Furthermore, using two wires (*S.cerevisiae* and *C.albicans*) half of all possible 3–input functions can be implemented.

Despite that the circuit design was introduced as a multi–layer architecture, none of the circuits were implemented with more than two cellular layers. Nonetheless, we demonstrated that when a first cell communicates with an intermediate one, that communicates with

an output producer cell by a different wiring molecule, the response is maintained [Regot et al., 2011, Fig S12]. The intermediate layer cells could be of any mat type, since it is known that both mat types are able to sense and secrete the same pheromone [Huberman and Murray, 2013]. Based on this architecture, any multi–layer circuit can be implemented if additional wiring molecules were engineered. These new wiring molecules could be implemented from other yeast species not suspicious of crosstalk such as *Sordaria macrospora, Debaryomyces hansenii, Gibberella zeae,* etc... [Marsh and Herskowitz, 1988; Mayrhofer and Poggeler, 2005; Chen and Weiss, 2005; Gonçalves-Sá, 2010]

Future studies should be performed to assess the advantages of an architecture with feedbacks or with a more compact design. For example, if the first layer of cells can compute as an `AND` and `N-IMPLIES` logic responding to two external inputs [Solé and Macía, 2014]. The intermediate cells could also respond to two external inputs only in presence of the wiring molecule. This could be implemented by altering different components of the same pathway, such as as Ste7 or Ste11 in addition to Fus3 [Jenness et al., 1987]. Thus, the number of layers would be reduced and, consequently, the number of required wirings, but at the expense of additional genetic engineering [Ji et al., 2013].

The architecture of the second design, **ILF** [manuscript in preparation], is composed by simpler cells organized in only two layers (IL and OL). This was accomplished by using the functional complete set {`NOT`, `OR`}. The cells implement the `IDENTITY` or `NOT` logics. The `OR` logic is performed when a OL cell senses a wiring molecule secreted by any IL cell. Spatial segregation of each set of cells in different isolated modules, reduces the number of required wires to one.

Again, the separation of logic gates in different cells allows to reuse the same genetic elements. Due to the simplicity of these cells, only three elements are needed: a set of inducible promoters, a repressible constitutive promoter and its repressor. In fact, the ILF library of cells, operates with six inputs but only uses four different inducible promoters.

This is because the hybrid receptors were built with a common DNA binding domain (Gal4) but different specific hormone ligand binding domains (17–β–estradiol, Progesterone and Aldosterone). The use of hybrid receptors is a successful method to increase the number of inputs without increasing the genetic elements required. The pheromone receptors of different yeast species also activate the same pathway (as demonstrated by *C.albicans*). This modularity at the genetic level allows to easily expand our library to other human hormones (such as testosterone) or other yeast pheromones.

In order to be implemented in biomedical devices, this design should be implemented in human cells [Nevozhay et al., 2013]. Of note, our proof–of–principle cell library can be easily transferred to mammalian cells. Hybrid hormone receptors similar to the ones we used have already been commercialized to perform screening assays in mammalian cells [Technologies, 2012]. Additionally, the use of hybrid receptors would avoid possible undesired transcription because they can only bind to the heterologous yeast GAL1 promoter sequence. Moreover only one wiring implementation is required [Bacchus et al., 2012; Miller et al., 2012] and the `NOT` logic can be implemented straightforward with interfering RNAs.

In summary, both designs reported in this thesis have demonstrated to be successful strategies to implement complex biological circuits;

The first strategy in multiple layers, was designed to be used in a single chamber, requires less number of cells but more wiring molecules, and yet, with only 4 wires almost all 3–input functions and around 2800 4–input functions could be implemented [Ji et al., 2013]. Although, if cellular spatial segregation was used this requirements would be reduced.

The second strategy (ILF; two cell layer implementation only with `OR`, `NOT` and `IDENTITY` operators), was intended for scenarios with available isolated spaces. In some cases, it requires more cells than the first strategy but only requires one wire. It allows for easily implementation of much more complex circuits, and is also more suitable for scenarios that demand the same response to different inputs.

There is no a better design than the other but a more appropriate design depending on each particular circuit and which resources are at our disposal. Overall and undoubtedly, three main advantages arise from both circuit designs: (i) distribution of the output to significantly reduce the wiring requirements (ii) distribution of logic gates in distinct cells to reuse the same molecular elements to build the gates and (iii) combination of different cells to easily implement diverse functions. Both designs presented here, demonstrated that multicellular distributed computation is a powerful strategy to build synthetic circuits performing robust results with a resolution comparable to electronics. These two designs opens a door for reusable, reprogrammable and truly scalable complex devices.

From "proof of principle" models to applications.

It is important to mention that forthcoming synthetic biology devices should overcome several current limitations in order to be really usable in natural environments. The future biological circuits will require strategies to: (a) maintain the stability, equilibrium and synchronization of the different cell populations, (b) modulate the dynamics of the response, (c) adapt and synchronize the response to variations of input and (d) store the response.

An appropriate proportion of distinct cell populations within a culture must result in similar amount of wiring molecule production from each cell type population (for a given input concentration). Besides, variations in the proportion of the different cell types populations, within certain margins, allows to modulate the dynamics of the response [Jahn et al., 2013]. Our results showed that yeast circuits can grow for eight generations and, maintained circuit competence and resolution. However, the ratios among different cell types were affected due to changes in cell cycle progression. Deletion of Far1 could be a possible solution to overcome this particular undesired feature, since *far1*Δ cells can not arrest cell cycle upon pheromone induction [Chang and Herskowitz, 1992]. Probably, future biomedical devices would rather be implemented with quiescent or growth controlled cell lines [Fursov et al., 2005].

The modulation of the dynamic response is essential. Some particular physiological applications must achieve an appropriate concentration of output quickly. To do that, a possible approach would be engineer cells to that efficiently control output production at the secretion. For example, in yeast the control of alpha factor release can be achieved by modulating the expression of some proteases (i.e. kex2). The speed of response can also be fine–tuned using scaffold proteins, that with additional factors, allows building of different synthetic feedback loops to generate diverse output dynamics [Park et al., 2003; Bashor et al., 2008].

To allow for reusable computing in changing environments, the genetic engineering of the cells must be expanded to not only respond once, but to interpret continuous stimuli changes and switch to the new correct output production. These designs should control the desensitization due to repeated exposures and the effect of receptor internalization, as happens with alpha factor [Moore, 1984; Schandel and Jenness, 1994]. To this end, we need to actively control the "ON" and "OFF" signals, ensuring that the switching times from 0 to 1 and from 1 to 0 are similar. This, together with a fine synchronization of cell populations will enable a periodic response [Danino et al., 2010]. This synchronization becomes essential in those sequential designs where the response not only depends on the inputs but also on the state and previous cell responses [Miller et al., 2012; Hoteit et al., 2012; Chuang and Lin, 2014; Zhang et al., 2014].

Stable gates will be useful for applications requiring tracking or processing memory [Bonnet et al., 2013]. As already mentioned, design and implementation of a biologic memory devices is a long sought goal [Ajo-Franklin et al., 2007; Hayat et al., 2006; Siuti et al., 2013]. Interestingly, heterochromatin remodelling and methylation can be seen as an opportunity to regulate heritable gene expression patterns and could be an effective technique to modulate epigenetic memory on demand.

There is now, a good foundational understanding of the logic of cell signalling machinery and the sources of functional plasticity. This vast knowledge will allow Synthetic biologists to successfully address the mentioned issues. Forthcoming synthetic devices are envisioned as stable

systems able to operate correctly even in highly complex environments, resistant to noise and to independent failure of components [Mee and Wang, 2012]. This century is likely to represent synthetic biology's advance from a "proof of concept" discipline to a tool commonly used for biological, environmental and medical applications.

CONCLUSIONS

- Distributed computation among different cells allows reusability of genetic elements and cells to construct different *in vivo* logic circuits.
- Distributed computation allows easy programmation of new circuits by combination of cells from small libraries.
- Distribute the final output in different cells allows reduction of wiring requirements for circuit implementation.
- The output of multicellular circuits is robust and exhibits a resolution comparable to electronics.
- Multicellular circuits of yeast cell libraries, are stable at least for four generations.
- The non-standard FCS (functionally complete set): `IDENTITY`, `NOT`, `AND`, `N-IMPLIES` reduces wiring requirements.
- Multicellular circuits can be reprogrammed upon addition of specific inhibitors.
- ILF (simplified function expression only with `OR` and `NOT` operators) allows a systematic implementation of any function in a two–layer architecture of simple `IDENTITY` and `NOT` cells with minimal number of wires.
- Spatial segregation of cells in isolated modules together with ILF serves to implement a virtually unlimited circuits with only 1 wire.

SUPPLEMENTARY ARTICLES

Dynamic Signalling in the Hog1 MAPK Pathway Relies on High Basal Signal Transduction

Javier Macia, Sergi Regot, Tom Peeters, Núria Conde, Ricard Solé and Francesc Posas

Science Signalling 2, 63 (2009)

Macia J, Regot S, Peeters T, Conde N, Solé R, Posas F. Dynamic signaling in the Hog1 MAPK pathway relies on high basal signal transduction. Sci Signal. 2009 Mar 24;2(63):ra13. doi: 10.1126/scisignal.2000056

SUPPLEMENTARY ARTICLES

Hog1 bypasses stress-mediated down-regulation of transcription by RNA polymerase II redistribution and chromatin remodeling

Mariona Nadal-Ribelles, Núria Conde, Oscar Flores, Juan González-Vallinas, Eduardo Eyras, Modesto Orozco, Eulália de Nadal and Francesc Posas.

Genome Biology 13:R106 (2012)

Nadal-Ribelles M, Conde N, Flores O, González-Vallinas J, Eyras E, Orozco M, et al. Hog1 bypasses stress-mediated down-regulation of transcription by RNA polymerase II redistribution and chromatin remodeling. Genome Biol. 2012; 13(11): R106. doi:10.1186/gb-2012-13-11-r106

SUPPLEMENTARY ARTICLES

In this article we addressed the mechanism by which osmostress gene induction is carried out in parallel with major gene repression in yeast. It is known that there is a massive dissociation of proteins from chromatin in response to stress. Accordingly, there is a genome-wide tendency to decrease association of RNA Pol II which leads the entire genome into a repressive state. On the other hand, while the entire genome is undergoing RNA Pol II dissociation, Hog1 selectively targets stress-responsive genes for transcription by recruiting RNA Pol II and chromatin remodelers that lead to profound changes in nucleosome eviction. Using Chip-Seq and MNase-seq assays we characterized the location of Hog1, RNA Pol II and nucleosomes in wild type and *hog1* strains with and without osmostress. This experimental set up allowed us to uncover several unknown properties of Hog1-dependent transcriptome. These results represent a more accurate genome-wide role of Hog1 than previously estimated.

Personal contribution to this work: Except for the initial design and execution of the experiments, which were performed by MN, I have been fully involved in the bioinformatic analyses of this study.

REFERENCES

Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Jr., T. F. K., F, T., Nagpal, R., Rauch, E., Sussman, G. J., and Weiss, R. (1995). Amorphous computing. *Communications of the ACM*, 43:74–82.

Achstetter, T. and Wolf, D. H. (1985). Hormone processing and membrane-bound proteinases in yeast. *EMBO J.*, 4(1):173–177.

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

Ajo-Franklin, C. M., Drubin, D. A., Eskin, J. A., Gee, E. P., Landgraf, D., Phillips, I., and Silver, P. A. (2007). Rational design of memory in eukaryotic cells. *Genes Dev.*, 21(18):2271–2276.

Alon, U. (2007). Network motifs: theory and experimental approaches. *Nat. Rev. Genet.*, 8(6):450–461.

Anderson, J. C., Clarke, E. J., Arkin, A. P., and Voigt, C. A. (2006). Environmentally controlled invasion of cancer cells by engineered bacteria. *J. Mol. Biol.*, 355(4):619–627.

Atkinson, M. R., Savageau, M. A., Myers, J. T., and Ninfa, A. J. (2003). Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in Escherichia coli. *Cell*, 113(5):597–607.

Ausländer, S., Wieland, M., and Fussenegger, M. (2012). Smart medication through combination of synthetic biology and cell microencapsulation. *Metab. Eng.*, 14(3):252–260.

REFERENCES

Bacchus, W., Aubel, D., and Fussenegger, M. (2013). Biomedically relevant circuit-design strategies in mammalian synthetic biology. *Mol. Syst. Biol.*, 9:691.

Bacchus, W., Lang, M., El-Baba, M. D., Weber, W., Stelling, J., and Fussenegger, M. (2012). Synthetic two-way communication between mammalian cells. *Nat. Biotechnol.*, 30(10):991–996.

Bader, O., Krauke, Y., and Hube, B. (2008). Processing of predicted substrates of fungal Kex2 proteinases from Candida albicans, C. glabrata, Saccharomyces cerevisiae and Pichia pastoris. *BMC Microbiol.*, 8:116.

Balagadde, F. K., Song, H., Ozaki, J., Collins, C. H., Barnet, M., Arnold, F. H., Quake, S. R., and You, L. (2008). A synthetic Escherichia coli predator-prey ecosystem. *Mol. Syst. Biol.*, 4:187.

Barnea, G., Strapps, W., Herrada, G., Berman, Y., Ong, J., Kloss, B., Axel, R., and Lee, K. J. (2008). The genetic design of signaling cascades to record receptor activation. *Proc. Natl. Acad. Sci. U.S.A.*, 105(1):64–69.

Bashor, C. J., Helman, N. C., Yan, S., and Lim, W. A. (2008). Using engineered scaffold interactions to reshape MAP kinase pathway signaling dynamics. *Science*, 319(5869):1539–1543.

Bashor, C. J., Horwitz, A. A., Peisajovich, S. G., and Lim, W. A. (2010). Rewiring cells: synthetic biology as a tool to interrogate the organizational principles of living systems. *Annu Rev Biophys*, 39:515–537.

Bassler, B. L. (1999). How bacteria talk to each other: regulation of gene expression by quorum sensing. *Current Opinion in Microbiology*, 2(6):582 – 587.

Basu, S., Gerchman, Y., Collins, C. H., Arnold, F. H., and Weiss, R. (2005). A synthetic multicellular system for programmed pattern formation. *Nature*, 434(7037):1130–1134.

Basu, S., Mehreja, R., Thiberge, S., Chen, M. T., and Weiss, R. (2004). Spatiotemporal control of gene expression with pulse-generating networks. *Proc. Natl. Acad. Sci. U.S.A.*, 101(17):6355–6360.

Bayer, T. S. and Smolke, C. D. (2005). Programmable ligand-controlled riboregulators of eukaryotic gene expression. *Nat. Biotechnol.*, 23(3):337–343.

Ben-Jacob, E. and Levine, H. (2006). Self-engineering capabilities of bacteria. *J. R. Soc. Interface.*, 3(6):197–214.

Benenson, Yaakov (2012). Biomolecular computing systems principles, progress and potential. *Nat Rev Genet*, 13(7):455–468.

Black, C. (2001). Compositions and methods for activating genes of interest. US Patent 6,323,003.

Black, C. (2003). Switching on gene therapy: Using gene profiles to design drugs. *Pharmagenomics*, 2:48–53.

Bonnet, J., Yin, P., Ortiz, M. E., Subsoontorn, P., and Endy, D. (2013). Amplifying genetic logic gates. *Science*, 340(6132):599–603.

Bray, D. (1990). Intracellular Signalling as a Parallel Distributed Process. *J. theor. Biol.*, 143:215–231.

Bray, D. (1995). Protein molecules as a computational elements in living cells. *Nature*, 376:307–312.

Brenner, K., Karig, D. K., Weiss, R., and Arnold, F. H. (2007). Engineered bidirectional communication mediates a consensus in a microbial biofilm consortium. *Proc. Natl. Acad. Sci. U.S.A.*, 104(44):17300–17304.

Brent, R. and Ptashne, M. (1984). A bacterial repressor protein or a yeast transcriptional terminator can block upstream activation of a yeast gene. *Nature*, 312(5995):612–615.

REFERENCES

Brent, R. and Ptashne, M. (1985). A bacterial repressor protein or a yeast transcriptional terminator can block upstream activation of a yeast gene. *Nature*, 314(6007):198.

Byung Kwon, L., Sanjay, K., Naider, F., and Jeffrey, M. B. (2001). Identification of Residues of the Saccharomyces cerevisiae G Protein-coupled Receptor Contributing to alpha-Factor Pheromone Binding. *JBC*, 276(41):37950–37961.

Chang, D. E., Leung, S., Atkinson, M. R., Reifler, A., Forger, D., and Ninfa, A. J. (2010). Building biological memory by linking positive feedback loops. *Proc. Natl. Acad. Sci. U.S.A.*, 107(1):175–180.

Chang, F. and Herskowitz, I. (1992). Phosphorylation of FAR1 in response to alpha-factor: a possible requirement for cell-cycle arrest. *Mol. Biol. Cell*, 3(4):445–450.

Chen, M. T. and Weiss, R. (2005). Artificial cell-cell communication in yeast Saccharomyces cerevisiae using signaling elements from Arabidopsis thaliana. *Nat. Biotechnol.*, 23(12):1551–1555.

Chen, Y. Y., Jensen, M. C., and Smolke, C. D. (2010). Genetic control of mammalian T-cell proliferation with synthetic RNA regulatory systems. *Proc. Natl. Acad. Sci. U.S.A.*, 107(19):8531–8536.

Chirstopher, J. S. and Kendall, J. B. (1994). The Third Cytoplasmic Loop of a Yeast G-Protein-Coupled Receptor Controls Pathway Activation, Ligand Discrimination, and Receptor Internalization. *Mol. Cell. Biol*, 14(5):3339–3349.

Chuang, C.-H. and Lin, C.-L. (2014). Synthesizing genetic sequential logic circuit with clock pulse generator. *BMC Systems Biology*, 8(1):63.

Clontech, L. (2001). *Living Colors User Manual. Protocol PT2040-1. Version No. PR1Y691*. Clontech Laboratories, Inc, http://www.clontech.com.

Coelho, M., Gonçalves, P., and Sampaio, J. (2011). Evidence for maintenance of sex determinants but not of sexual stages in red yeasts, a group of early diverged basidiomycetes. *BMC Evolutionary Biology*, 11(1).

Conklin, B. R., Hsiao, E. C., Claeysen, S., Dumuis, A., Srinivasan, S., Forsayeth, J. R., Guettier, J. M., Chang, W. C., Pei, Y., McCarthy, K. D., Nissenson, R. A., Wess, J., Bockaert, J., and Roth, B. L. (2008). Engineering GPCR signaling pathways with RASSLs. *Nat. Methods*, 5(8):673–678.

Cox, R. S., Surette, M. G., and Elowitz, M. B. (2007). Programming gene expression with combinatorial promoters. *Mol. Syst. Biol.*, 3:145.

Cronin, M. T. and Mansfield, E. S. (2001). Microfluidic arrays in genetic analysis. *Expert Rev. Mol. Diagn.*, 1(1):109–118.

Danino, T., Mondragon-Palomino, O., Tsimring, L., and Hasty, J. (2010). A synchronized quorum of genetic clocks. *Nature*, 463(7279):326–330.

Denning, P. J. (2007). Computing is a natural science. *Commun. ACM*, 50(7):13–18.

Drubin, D. A., Way, J. C., and Silver, P. A. (2007). Designing biological systems. *Genes Dev.*, 21(3):242–254.

Dueber, J. E., Yeh, B. J., Chak, K., and Lim, W. A. (2003). Reprogramming control of an allosteric signaling switch through modular recombination. *Science*, 301(5641):1904–1908.

Elion, E. A., Brill, J. A., and Fink, G. R. (1991). Functional redundancy in the yeast cell cycle: FUS3 and KSS1 have both overlapping and unique functions. *Cold Spring Harb. Symp. Quant. Biol.*, 56:41–49.

Elloumi-Hannachi, I., Maeda, M., Yamato, M., and Okano, T. (2010). Portable microcontact printing device for cell culture. *Biomaterials*, 31(34):8974–8979.

REFERENCES

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

Enderton, H. (2001). *A Mathematical Introduction to Logic*. Harcourt/Academic Press.

Erdmann, V. A., Barciszewska, M. Z., Szymanski, M., Hochberg, A., de Groot, N., and Barciszewski, J. (2001). The non-coding RNAs as riboregulators. *Nucleic Acids Res.*, 29(1):189–193.

Feng, Y. and Davis, N. G. (2000). Feedback phosphorylation of the yeast a-factor receptor requires activation of the downstream signaling pathway from G protein through mitogen-activated protein kinase. *Mol. Cell. Biol.*, 20(2):563–574.

Fields, S. and Song, O. (1989). A novel genetic system to detect protein–protein interactions. *Nature*, 340(6230):245–246.

Fitzpatrick, D. A., Logue, M. E., Stajich, J. E., and Butler, G. (2006). A fungal phylogeny based on 42 complete genomes derived from supertree and combined gene analysis. *BMC Evol. Biol.*, 6:99.

Friedland, A. E., Lu, T. K., Wang, X., Shi, D., Church, G., and Collins, J. J. (2009). Synthetic gene networks that count. *Science*, 324(5931):1199–1202.

Fung, E., Wong, W. W., Suen, J. K., Bulter, T., Lee, S. G., and Liao, J. C. (2005). A synthetic gene-metabolic oscillator. *Nature*, 435(7038):118–122.

Fursov, N., Cong, M., Federici, M., Platchek, M., Haytko, P., Tacke, R., Livelli, T., and Zhong, Z. (2005). Improving consistency of cell-based assays by using division-arrested cells. *Assay Drug Dev Technol*, 3(1):7–15.

Gaber, R., Lebar, T., Majerle, A., Ster, B., Dobnikar, A., Ben?ina, M., and Jerala, R. (2014). Designable DNA-binding domains enable

construction of logic circuits in mammalian cells. *Nat. Chem. Biol.*, 10(3):203–208.

Gardner, T. S., Cantor, C. R., and Collins, J. J. (2000). Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403(6767):339–342.

Garrison, T. R., Zhang, Y., Pausch, M., Apanovitch, D., Aebersold, R., and Dohlman, H. G. (1999). Feedback phosphorylation of an RGS protein by MAP kinase in yeast. *J. Biol. Chem.*, 274(51):36387–36391.

Gonçalves-Sá, J. (2010). *"Asymmetry in Sexual Pheromones Is Not Required for Ascomycete Mating "*. PhD thesis, Instituto de Tecnologia Quimica e Biologica da Universidade Nova de Lisboa, Oeiras.

Gordon, DeborahM. (1999). *Interaction patterns and task allocation in ant colonies*. Birkhäuser Basel.

Gossen, M. and Bujard, H. (1992). Tight control of gene expression in mammalian cells by tetracycline-responsive promoters. *Proc. Natl. Acad. Sci. U.S.A.*, 89(12):5547–5551.

Gossen, M., Freundlieb, S., Bender, G., Muller, G., Hillen, W., and Bujard, H. (1995). Transcriptional activation by tetracyclines in mammalian cells. *Science*, 268(5218):1766–1769.

Gottesman, S., Roche, E., Zhou, Y., and Sauer, R. T. (1998). The ClpXP and ClpAP proteases degrade proteins with carboxy-terminal peptide tails added by the SsrA-tagging system. *Genes Dev.*, 12(9):1338–1347.

Grilly, C., Stricker, J., Pang, W. L., Bennett, M. R., and Hasty, J. (2007). A synthetic gene network for tuning protein degradation in Saccharomyces cerevisiae. *Mol. Syst. Biol.*, 3:127.

Gross, G., Waks, T., and Eshhar, Z. (1989). Expression of immunoglobulin–T–cell receptor chimeric molecules as functional receptors with antibody–type specificity. *Proc. Natl. Acad. Sci. U.S.A.*, 86(24):10024–10028.

REFERENCES

Guet, C. C., Elowitz, M. B., Hsing, W., and Leibler, S. (2002). Combinatorial synthesis of genetic networks. *Science*, 296(5572):1466–1470.

Gurevich, Y. (2000). Sequential abstract-state machines capture sequential algorithms. *ACM Trans. Comput. Logic*, 1(1):77–111.

Ham, T. S., Lee, S. K., Keasling, J. D., and Arkin, A. P. (2006). A tightly regulated inducible expression system utilizing the fim inversion recombination switch. *Biotechnol. Bioeng.*, 94(1):1–4.

Ham, T. S., Lee, S. K., Keasling, J. D., and Arkin, A. P. (2008). Design and construction of a double inversion recombination switch for heritable sequential genetic memory. *PLoS ONE*, 3(7):e2815.

Hayat, S., Ostermann, K., Brusch, L., Pompe, W., and Rödel, G. (2006). Towards in vivo computing: Quantitative analysis of an artificial gene regulatory network behaving as a rs flip-flop and simulating the system in silico. In *Proceedings of the 1st International Conference on Bio Inspired Models of Network, Information and Computing Systems*, BIONETICS '06, New York, NY, USA. ACM.

Haynes, K. A. and Silver, P. A. (2009). Eukaryotic systems broaden the scope of synthetic biology. *J. Cell Biol.*, 187(5):589–596.

Holland, J. (1975). *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. Ann Arbor: University of Michigan Press.

Hoteit, I., Kharma, N., and Varin, L. (2012). Computational simulation of a gene regulatory network implementing an extendable synchronous single-input delay flip-flop. *Bio Systems*, 109(1):57—71.

Huberman, L. B. and Murray, A. W. (2013). Genetically engineered transvestites reveal novel mating genes in budding yeast. *Genetics*.

Irving, B. A. and Weiss, A. (1991). The cytoplasmic domain of the T cell receptor ζ chain is sufficient to couple to receptor-associated signal transduction pathways. *Cell*, 64(5):891–901.

Isaacs, F. J., Dwyer, D. J., Ding, C., Pervouchine, D. D., Cantor, C. R., and Collins, J. J. (2004). Engineered riboregulators enable post–transcriptional control of gene expression. *Nat. Biotechnol.*, 22(7):841–847.

Isaacs FJ, C. J. (2005). Plug-and-play with RNA. *Nat Biotechnol.*, 23(3):306–307.

Jahn, M., Molle, A., Rodel, G., and Ostermann, K. (2013). Temporal and spatial properties of a yeast multi-cellular amplification system based on signal molecule diffusion. *Sensors (Basel)*, 13(11):14511–14522.

Janiak, A. M., Sargsyan, H., Russo, J., Naider, F., Hauser, M., and Becker, J. M. (2005). Functional expression of the Candida albicans alpha-factor receptor in Saccharomyces cerevisiae. *Fungal Genet. Biol.*, 42(4):328–338.

Jenness, D. D., Goldman, B. S., and Hartwell, L. H. (1987). Saccharomyces cerevisiae mutants unresponsive to alpha-factor pheromone: alpha-factor binding and extragenic suppression. *Mol. Cell. Biol.*, 7(4):1311–1319.

Jenness, D. D. and Spatrick, P. (1986). Down regulation of the alpha-factor pheromone receptor in S. cerevisiae. *Cell*, 46(3):345–353.

Ji, W., Shi, H., Zhang, H., Sun, R., Xi, J., Wen, D., Feng, J., Chen, Y., Qin, X., Ma, Y., Luo, W., Deng, L., Lin, H., Yu, R., and Ouyang, Q. (2013). A formalized design process for bacterial consortia that perform logic computing. *PLoS ONE*, 8(2):e57482.

Joana Gonçalves-Sá and Andrew Murray (2011). Asymmetry in Sexual Pheromones Is Not Required for Ascomycete Mating. *Current Biology*, 21(16):1337–1346.

REFERENCES

Johnston, M. (1999). Feasting, fasting and fermenting. Glucose sensing in yeast and other cells. *Trends Genet.*, 15(1):29–33.

Karzai, A. W., Roche, E. D., and Sauer, R. T. (2000). The SsrA-SmpB system for protein tagging, directed degradation and ribosome rescue. *Nat. Struct. Biol.*, 7(6):449–455.

Kemmer, C., Fluri, D. A., Witschi, U., Passeraub, A., Gutzwiller, A., and Fussenegger, M. (2011). A designer network coordinating bovine artificial insemination by ovulation-triggered release of implanted sperms. *J Control Release*, 150(1):23–29.

Kinkhabwala, A. and Bastiaens, P. (2010). Spatial aspects of intracellular information processing. *Curr. Opin. Genet. Dev.*, 187(20):31–40.

Knight Jr., T. F. and Sussman, G. J. (1998). Cellular gate technology. In *UMC98: First International Conference On Unconventional Models Of Computation*, pages 257–272.

Kobayashi, H., Kaern, M., Araki, M., Chung, K., Gardner, T. S., Cantor, C. R., and Collins, J. J. (2004). Programmable cells: interfacing natural and engineered gene networks. *Proc. Natl. Acad. Sci. U.S.A.*, 101(22):8414–8419.

Kramer, B. P., Fischer, C., and Fussenegger, M. (2004). BioLogic gates enable logical transcription control in mammalian cells. *Biotechnol. Bioeng.*, 87(4):478–484.

Kwok, R. (2010). Five hard truths for synthetic biology. *Nature*, 463:288–290.

Lechner, R. (1992). Regulation of nucleic acid translation. CA Patent App. CA 2,100,251.

Lechner, R. L., Engler, M. J., and Richardson, C. C. (1983). Characterization of strand displacement synthesis catalyzed by bacteriophage T7 DNA polymerase. *J. Biol. Chem.*, 258(18):11174–11184.

Levinson, R., Kahn, R., Accili, D., Berrie, N., Fenwick, S., and Vicari, K. (2012). Hepatocyte Metabolic Homeostasis, Metabolic Syndrome ePoster. *Nat. Med.*

Levskaya, A., Chevalier, A. A., Tabor, J. J., Simpson, Z. B., Lavery, L. A., Levy, M., Davidson, E. A., Scouras, A., Ellington, A. D., Marcotte, E. M., and Voigt, C. A. (2005). Synthetic biology: engineering Escherichia coli to see light. *Nature*, 438(7067):441–442.

Levskaya, A., Weiner, O. D., Lim, W. A., and Voigt, C. A. (2009). Spatiotemporal control of cell signalling using a light-switchable protein interaction. *Nature*, 461(7266):997–1001.

Lim, W. A. (2010). Designing customized cell signalling circuits. *Nat Rev Mol Cell Biol.*, 11(6):393–403.

Lohr, D., Venkov, P., and Zlatanova, J. (1995). Transcriptional regulation in the yeast GAL gene family: a complex genetic network. *FASEB J.*, 9(9):777–787.

Lou, C., Liu, X., Ni, M., Huang, Y., Huang, Q., Huang, L., Jiang, L., Lu, D., Wang, M., Liu, C., Chen, D., Chen, C., Chen, X., Yang, L., Ma, H., Chen, J., and Ouyang, Q. (2010). Synthesizing a novel genetic sequential logic circuit: a push-on push-off switch. *Mol. Syst. Biol.*, 6:350.

Louvion, J. F., Havaux-Copf, B., and Picard, D. (1993). Fusion of GAL4-VP16 to a steroid-binding domain provides a tool for gratuitous induction of galactose-responsive genes in yeast. *Gene*, 131(1):129–134.

Macia, J., Posas, F., and Sole, R. V. (2012). Distributed computation: the new wave of synthetic biology devices. *Trends Biotechnol.*, 30(6):342–349.

Macia, J., Regot, S., Peeters, T., Conde, N., Sole, R., and Posas, F. (2009a). Dynamic signaling in the Hog1 MAPK pathway relies on high basal signal transduction. *Sci Signal*, 2(63):ra13.

REFERENCES

Macia, J. and Sole, R. V. (2009). Distributed robustness in cellular networks: insights from synthetic evolved circuits. *J R Soc Interface*, 6(33):393–400.

Macia, J., Widder, S., and Sole, R. (2009b). Why are cellular switches Boolean? General conditions for multistable genetic circuits. *J. Theor. Biol.*, 261(1):126–135.

Macia, J., Widder, S., and Sole, R. (2009c). Why are cellular switches Boolean? General conditions for multistable genetic circuits. *J. Theor. Biol.*, 261(1):126–135.

Marsh, L. and Herskowitz, I. (1988). STE2 protein of Saccharomyces kluyveri is a member of the rhodopsin/beta-adrenergic receptor family and is responsible for recognition of the peptide ligand alpha factor. *Proc. Natl. Acad. Sci. U.S.A.*, 85(11):3855–3859.

Mayrhofer, S. and Poggeler, S. (2005). Functional characterization of an alpha-factor-like Sordaria macrospora peptide pheromone and analysis of its interaction with its cognate receptor in Saccharomyces cerevisiae. *Eukaryotic Cell*, 4(4):661–672.

Mee, M. T. and Wang, H. H. (2012). Engineering ecosystems and synthetic ecologies. *Mol Biosyst*, 8(10):2470–2483.

Melin, J. and Quake, S. R. (2007). Microfluidic large-scale integration: the evolution of design rules for biological automation. *Annu Rev Biophys Biomol Struct*, 36:213–231.

Michaelis, S. and Herskowitz, I. (1988). The a-factor pheromone of Saccharomyces cerevisiae is essential for mating. *Mol. Cell. Biol.*, 8(3):1309–1318.

Miller, C. A., 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(2):73–78.

Miller, M., Hafner, M., Sontag, E., Davidsohn, N., Subramanian, S., Purnick, P. E., Lauffenburger, D., and Weiss, R. (2012). Modular design of artificial tissue homeostasis: robust control through synthetic cellular heterogeneity. *PLoS Comput. Biol.*, 8(7):e1002579.

Mitchell, M. (2009). *Complexity: A Guided Tour*. Oxford University Press, USA.

Moon, T. S., Clarke, E. J., Groban, E. S., Tamsir, A., Clark, R. M., Eames, M., Kortemme, T., and Voigt, C. A. (2011). Construction of a genetic multiplexer to toggle between chemosensory pathways in Escherichia coli. *J. Mol. Biol.*, 406(2):215–227.

Moore, S. A. (1984). Yeast cells recover from mating pheromone alpha factor-induced division arrest by desensitization in the absence of alpha factor destruction. *J. Biol. Chem.*, 259(2):1004–1010.

Neumann, J. v. (1958). *The Computer and the Brain*. Yale University Press, New Haven, CT, USA.

Nevozhay, D., Zal, T., and Balazsi, G. (2013). Transferring a synthetic gene circuit from yeast to mammalian cells. *Nat Commun*, 4:1451.

Park, S. H., Zarrinpar, A., and Lim, W. A. (2003). Rewiring MAP kinase pathways using alternative scaffold assembly mechanisms. *Science*, 299(5609):1061–1064.

Peisajovich, S. G., Garbarino, J. E., Wei, P., and Lim, W. A. (2010). Rapid diversification of cell signaling phenotypes by modular domain recombination. *Science*, 328(5976):368–372.

Peter, M., Gartner, A., Horecka, J., Ammerer, G., and Herskowitz, I. (1993). FAR1 links the signal transduction pathway to the cell cycle machinery in yeast. *Cell*, 73(4):747–760.

Purnick, P. E. and Weiss, R. (2009). The second wave of synthetic biology: from modules to systems. *Nat. Rev. Mol. Cell Biol.*, 10(6):410–422.

REFERENCES

Redfern, C. H., Coward, P., Degtyarev, M. Y., Lee, E. K., Kwa, A. T., Hennighausen, L., Bujard, H., Fishman, G. I., and Conklin, B. R. (1999). Conditional expression and signaling of a specifically designed Gi-coupled receptor in transgenic mice. *Nat. Biotechnol.*, 17(2):165–169.

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(7329):207–211.

Rőssger, K., Charpin-El-Hamri, G., and Fussenegger, M. (2013). A closed-loop synthetic gene circuit for the treatment of diet-induced obesity in mice. *Nat Commun*, 4:28–25.

Ricoult, S., Goldman, J., Stellwagen, D., Juncker, D., and Kennedy, T. (2012). Generation of microisland cultures using microcontact printing to pattern protein substrates. *Journal of Neuroscience Methods*, 208(1):10–17.

Rinaudo, K., Bleris, L., Maddamsetti, R., Subramanian, S., Weiss, R., and Benenson, Y. (2007). A universal RNAi-based logic evaluator that operates in mammalian cells. *Nat. Biotechnol.*, 25(7):795–801.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Rumelhart, D. E., McClelland, J. L., and PDP Research Group, C., editors (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press.

Sadelain, M., Brentjens, R., and Riviere, I. (2009). The promise and potential pitfalls of chimeric antigen receptors. *Curr. Opin. Immunol.*, 21(2):215–223.

Scannell, D. R., Butler, G., and Wolfe, K. H. (2007). Yeast genome evolution–the origin of the species. *Yeast*, 24(11):929–942.

Schandel, K. A. and Jenness, D. D. (1994). Direct evidence for ligand-induced internalization of the yeast alpha-factor pheromone receptor. *Mol. Cell. Biol.*, 14(11):7245–7255.

Schmidt, M. and de Lorenzo, V. (2012). Synthetic constructs in/for the environment: Managing the interplay between natural and engineered biology. *FEBS Letters*, 586(15):2199–2206.

Scott, J. D. and Pawson, T. (2009). Cell signaling in space and time: where proteins come together and when they're apart. *Science*, 326(5957):1220–1224.

Seoane, L. F. and Solé, R. V. (2013). Synthetic biocomputation design using supervised gene regulatory networks . *arXiv.org*.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656.

Shapiro, E. and Benenson, Y. (2006). Information technology: Bringing DNA computers to life. *j-SCI-AMER*, 294(5):44–51.

Shou, W., Ram, S., and Vilar, J. M. (2007). Synthetic cooperation in engineered yeast populations. *Proc. Natl. Acad. Sci. U.S.A.*, 104(6):1877–1882.

Simon, H. A. (1962). The architecture of complexity. In *Proceedings of the American Philosophical Society*, pages 467–482.

Siuti, P., Yazbek, J., and Lu, T. K. (2013). Synthetic circuits integrating logic and memory in living cells. *Nat. Biotechnol.*, 31(5):448–452.

Solé, R. (2011). *Phase Transitions*. Primers in Complex Systems. Princeton University Press.

REFERENCES

Solé, R. V. and Macia, J. (2011). Synthetic Biocomputation: The Possible and the Actual. *Euro. Conf. Artificial Life*, 7:29–36.

Solé, R. V. and Macía, J. (2013). Expanding the landscape of biological computation with synthetic multicellular consortia. *Natural Computing*, 12(4):485–497.

Solé, R. V. and Macía, J. (2014). How to make a mutlicellular computer. *PLOS computational*, 12(4):485–497.

Stricker, J., Cookson, S., Bennett, M. R., Mather, W. H., Tsimring, L. S., and Hasty, J. (2008). A fast, robust and tunable synthetic gene oscillator. *Nature*, 456(7221):516–519.

Szilárd, L. (1929). Über die Entropieverminderung in einem thermodynamischen System bei Eingriffen intelligenter Wesen. (German) [On entropy reduction in a thermodynamic system by interference by intelligent subjects]. *j-Z-PHYSIK*, 53(11–12):840–856. This paper contains Szilard's classic analysis of Maxwell's demon, and shows that the entropy of a unit of information is equal to $k \log 2$, foreshadowing Claude Shannon's famous 1948 work on "A Mathematical Theory of Communication".

Tabor, J. J., Salis, H. M., Simpson, Z. B., Chevalier, A. A., Levskaya, A., Marcotte, E. M., Voigt, C. A., and Ellington, A. D. (2009). A synthetic genetic edge detection program. *Cell*, 137(7):1272–1281.

Takahashi, S. and Pryciak, P. M. (2008). Membrane localization of scaffold proteins promotes graded signaling in the yeast MAP kinase cascade. *Curr. Biol.*, 18(16):1184–1191.

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

Technologies, L. (2012). Selectscreen TM. Cell-based nuclear receptor profiling service. Screening protocol and assay conditions.

Tecon, R., Wells, M., and van der Meer, J. R. (2006). A new green fluorescent protein-based bacterial biosensor for analysing phenanthrene fluxes. *Environ. Microbiol.*, 8(4):697–708.

Tigges, M., Marquez-Lago, T. T., Stelling, J., and Fussenegger, M. (2009). A tunable synthetic mammalian oscillator. *Nature*, 457(7227):309–312.

Tononi, G., Sporns, O., and Edelman, G. M. (1999). Measures of degeneracy and redundancy in biological networks. *Proc. Natl. Acad. Sci. U.S.A.*, 96(6):3257–3262.

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *j-PROC-LONDON-MATH-SOC-2*, 42:230–265. This is the paper that introduced what is now called the *Universal Turing Machine*.

Uhlmann, E. and Peyman, A. (1990). Antisense oligonucleotides: a new therapeutic principle. *Chemical Reviews*, 90(4):543–584.

van der Meer, J. R. and Belkin, S. (2010). Where microbiology meets microengineering: design and applications of reporter bacteria. *Nat. Rev. Microbiol.*, 8(7):511–522.

van Drogen, F., Stucke, V. M., Jorritsma, G., and Peter, M. (2001). MAP kinase dynamics in response to pheromones in budding yeast. *Nat. Cell Biol.*, 3(12):1051–1059.

Waite, A. J. and Shou, W. (2012). Adaptation to a new environment allows cooperators to purge cheaters stochastically. *Chemical Reviews*, 109(47):19079–19086.

Wang, B., Barahona, M., and Buck, M. (2013). A modular cell-based biosensor using engineered genetic logic circuits to detect and integrate multiple environmental signals. *Biosens Bioelectron*, 40(1):368–376.

REFERENCES

Weber, W., Daoud-El Baba, M., and Fussenegger, M. (2007). Synthetic ecosystems based on airborne inter- and intrakingdom communication. *Proc. Natl. Acad. Sci. U.S.A.*, 104(25):10435–10440.

Weiss, R., Homsy, G., and Nagpal, R. (1998). Programming biological cells. In *Eighth International Conference on Architectural Support for Programming Languages and Operating Systems*.

Win, M. N. and Smolke, C. D. (2008). Higher-order cellular information processing with synthetic RNA devices. *Science*, 322(5900):456–460.

Wolfram, S. (1984). Cellular automata as models of complexity. *Nature*, 311(5985):419–424.

Wu, Y. I., Frey, D., Lungu, O. I., Jaehrig, A., Schlichting, I., Kuhlman, B., and Hahn, K. M. (2009). A genetically encoded photoactivatable Rac controls the motility of living cells. *Nature*, 461(7260):104–108.

Wyborski, D. L., DuCoeur, L. C., and Short, J. M. (1996). Parameters affecting the use of the lac repressor system in eukaryotic cells and transgenic animals. *Environ. Mol. Mutagen.*, 28(4):447–458.

Wynendaele, E., Bronselaer, A., Nielandt, J., D'Hondt, M., Stalmans, S., Bracke, N., Verbeke, F., Wiele, C. V. D., Tré, G. D., and Spiegeleer, B. D. (2013). Quorumpeps database: chemical space, microbial origin and functionality of quorum sensing peptides. *Nucleic Acids Research*, 41(Database-Issue):655–659.

Xie, Z., Wroblewska, L., Prochazka, L., Weiss, R., and Benenson, Y. (2011). Multi-input RNAi-based logic circuit for identification of specific cancer cells. *Science*, 333(6047):1307–1311.

Ye, H., Charpin-El Hamri, G., Zwicky, K., Christen, M., Folcher, M., and Fussenegger, M. (2013). Pharmaceutically controlled designer circuit for the treatment of the metabolic syndrome. *Proc. Natl. Acad. Sci. U.S.A.*, 110(1):141–146.

Ye, H., Daoud-El Baba, M., Peng, R. W., and Fussenegger, M. (2011). A synthetic optogenetic transcription device enhances blood-glucose homeostasis in mice. *Science*, 332(6037):1565–1568.

You, L., Cox, R. S., Weiss, R., and Arnold, F. H. (2004). Programmed population control by cell-cell communication and regulated killing. *Nature*, 428(6985):868–871.

Yu, R. C., Pesce, C. G., Colman-Lerner, A., Lok, L., Pincus, D., Serra, E., Holl, M., Benjamin, K., Gordon, A., and Brent, R. (2008). Negative feedback that improves information transmission in yeast signalling. *Nature*, 456(7223):755–761.

Zhang, H., Lin, M., Shi, H., Ji, W., Huang, L., Zhang, X., Shen, S., Gao, R., Wu, S., Tian, C., Yang, Z., Zhang, G., He, S., Wang, H., Saw, T., Chen, Y., and Ouyang, Q. (2014). Programming a pavlovian-like conditioning circuit in escherichia coli. *Nature Communications*, 5.

Çağatay, T., Turcotte, M., Elowitz, M. B., Garcia-Ojalvo, J., and Süel, G. M. (2009). Architecture-dependent noise discriminates functionally analogous differentiation circuits. *Cell*, 139.