doi:10.1038/nature09679

1. Supplementary Information

Method for designing distributed biological computation

In logic terms, digital circuits can be described by a more or less complex Boolean function *f* with *N* input bits and *M* output bits. This function describes how to determine the Boolean outputs from any set of possible inputs, namely:

$$f: \{0,1\}^N \rightarrow \{0,1\}^M$$

Where $\{0,1\}^N$ and $\{0,1\}^M$ indicate the set of all possible strings of *N* inputs and *M* outputs, respectively.

In principle, every Boolean function, as defined here, can be implemented in multiple ways, by combining different sets of logic gates. Such sets of logic gates are known as *functionally complete sets*. One particular systematic implementation of an arbitrary logic circuit can be obtained by interconnecting a set of two particular binary gates (AND and OR) and the NOT gate[1,2]. Therefore, it is feasible to systematically implant complex circuits by using the following functionally complete set:,

$$\Omega=\{\text{AND, OR, NOT}\}$$

Boolean computation can be systematically implemented using circuits formed by multiple interconnected cells. In this framework, each cell responds to different input stimuli activating or inhibiting the production of a given output which can itself act as the input for another cell. To implement biological computation, we choose to allow that the final output of the circuit can be generated *in different cells* of the circuit. Two main advantages arise from this circuit design; the redundant distribution of the desired output significantly reduces wiring constrains, whereas the

distribution of the logic blocks in different cells allows to reuse the same molecular elements to perform different tasks. Thus, it is a key aspect to define a method that determines how to use and re-use engineered cells with a particular logical computation to implement different and complex Boolean functions.

For simplicity, to define a general method implementing any arbitrary Boolean function in a multicellular circuit, we consider systems with N inputs and one output. The generalization to $M>1$ outputs is straightforward using standard methods[1,2]. In general, it can be shown that any Boolean function with N inputs can be expressed in terms of the Ω *functionally complete set* as a propositional formula in N variables $\{x_1,\ldots,x_N\}$ according to[3]:

$$f = [\theta^{s_{11}}{}_{11}(x_1) \;\; AND \;\; \theta^{s_{12}}{}_{12}(x_2)\cdots \;\; AND \;\; \theta^{s_{1N}}{}_{1N}(x_N)] \;\; OR \;\; [\theta^{s_{21}}{}_{21}(x_1) \;\; AND \;\; \theta^{s_{22}}{}_{22}(x_2)\cdots \;\; AND \;\; \theta^{s_{2N}}{}_{2N}(x_N)] \;\cdots OR \;\; [\theta^{s_{\mu 1}}{}_{\mu 1}(x_1) \;\; AND \;\; \theta^{s_{\mu 2}}{}_{\mu 2}(x_2)\cdots \;\; AND \;\; \theta^{s_{\mu N}}{}_{\mu N}(x_N)] \quad (1)$$

or in a compact form:

$$f = \sum_{i=1}^{Q}\left[\Pi_{j=1}^{N}\theta^{s_{ij}}{}_{ij}(x_j)\right] \quad (2)$$

Here, Σ represents the OR operator and Π the AND operator. The function $\theta^{s_{ij}}{}_{ij}(x_j)$ is either a logic representation of the presence (S_{ij}=IDENTITY function) of a molecular input x_j or a logic representation of its absence (S_{ij}=NOT function), i.e. $\theta^{Id}{}_{ij}(x_j)=1$ if $x_j=1$ or $\theta^{NOT}{}_{ij}(x_j)=1$ if $x_j=0$, depending on the specific Boolean function implemented. Finally, Q is the maximum number of terms that depends on the complexity of the function, but always the condition $Q \leq 2^{N-1}$ is satisficed[3]. 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 or the Quine-McCluskey algorithm[2].

For a given set of input values $\{x_j\}$, f will be 1 if at least one of the terms $\Pi_{j=1}^{N}\theta^{s_{ij}}{}_{ij}(x_j)=1$ in the equation (1). Within the context of multicellular circuits f=1 can be interpreted in terms of production of a specific molecular entity. The i^{st} term $\Pi_{j=1}^{N}\theta^{s_{ij}}{}_{ij}(x_j)$ in (1) can be implemented by a sorted set of different connected cells $\rho_i = \{C_{i1}, C_{i2}, ...C_{iP_i}\}$. Here, the number of cells is $P_i \leq N$ depending of the specific function implemented. Hence, the whole circuit Φ is composed of Q different sets of cells, i.e. $\Phi = \{\rho_1, \rho_2, ..., \rho_k\}$, without connection among these sets (see Fig. 2 in the main text). In each set ρ_i there is a cell C_{iP_i} able to produce the desired output.

For a given set ρ_i, every cell C_{ij} produce an output α_j responding both to an external input x_j and to the output α_{j-1} of the previous cell C_{ij-1} according to the logic function $\alpha_{ij} = \left[\alpha_{ij-1} \; AND \; \theta^{s_{ij}}{}_{ij}(x_j)\right]$. Here, the wiring pattern is unidirectionally fixed, namely the behavior of cell C_{ij} is determined by the behavior of the cell C_{ij-1} but C_{ij} do not determine Cij_{-1}. The behavior of these cells can be described by the following truth tables:

If $S_{ij}=Id$					If $S_{ij}=NOT$			
x_j	$\theta^{d}{}_{ij}(x_j)$	α_{ij-1}		α_{ij}	x_j	$\theta^{NOT}{}_{ij}(x_j)$	α_{ij-1}		α_{ij}
0	0	0		0	0	1	0		0
0	0	1		0	0	1	1		1
1	1	0		0	1	0	0		0
1	1	1		1	1	0	1		0

In the case $S_{ij}=Id$ the cell C_{ij} implements an AND function, whereas if $S_{ij}=NOT$ then cell C_{ij} implements an inverted IMPLIES (named N-IMPLIES) gate[4], Particularly, for a given set of cells ρ_i the first cell C_{i1} only respond to an external input x_1 because there is not a previous cell connected to it. This particular case can be implemented in two different alternatives: i) with one input cell with the proper logic, i.e. IDENTITY or NOT respectively, or ii) with the addition of a cell C_{i2} that

responds to two external inputs x_1 and x_2 according to the behaviors described previously, i.e. AND or N-IMPLIES. Both possibilities have been implemented experimentally (see Fig. 3 and Fig. S6). Fig. S1a shows a schematic representation of multicellular circuits with a fixed wiring pattern. Circuits with this architecture are able to implement any combinational Boolean function. For a given function, the number of required cells and wires can be determined systematically.

As an example, a method for designing distributed biological computation is shown in Fig. S1b-g. The circuit implemented is a multiplexer with 3-inputs and 1-output named MUX2to1. Fig. S1b shows the truth table defining the relationship between the input strings $\{x_1,x_2, x_3\}$ and the output O. In this type of circuits, the output adopts the value of one of the inputs, namely x_2 or x_3 depending on the value of x_1, i.e. $O=x_2$ if $x_1=0$ and $O=x_3$ if $x_1=1$. Fig. S1c shows the Boolean function f implementing the truth table. This function can be simplified using the standard methods[2]. Fig. S1d shows the simplified function implementing the truth table. According to the method described, this function can be implemented using two sets of cells ρ_1 and ρ_2 with multiple possible configurations. Fig. S1e-g show several alternative options to implement the same logic function. Fig. S1e shows a circuit obtained by direct mapping of the function shown in Fig. S1d. In this example, each term of the function is implemented by a different cell. Fig. S1f shows an intermediate example, where the terms of ρ_1 are implemented in different cells, but the terms of ρ_2 are condensed in a single cell. Finally, Fig. S1g shows the opposite case, where the ρ_1 and ρ_2 have been condensed in a single cell each one. As a proof of principle this example has been implemented experimentally (see Fig. 4). Despite that the case shown in Fig. S1g involves less cells a direct mapping between the terms of the Boolean function and the engineered cells have the additional property that allows for a greater flexibility permitting to re-use parts (cells) of the circuit in other circuits or rebuilding the circuit just replacing some cells.

Computational capabilities of multicellular circuits

In the multicellular approach to cell-based computation presented in this work, computations are distributed among different cells to respond according to an AND or inverted IMPLIES logic. In this context, the question about how many cells are necessary to implement a given function arises. What is the combinatorial universe of Boolean functions that can be effectively built? We addressed this question by systematically generating all possible circuits compatible with using a given number of engineered cells and wires.

Number of cells required for implementing distributed computation. The number of different cells required is determined by two different constraints namely, the number of inputs *N* and the intrinsic complexity of the Boolean function. In general, we could determine the set S(N,M) of *N*-input, *M*-output Boolean expressions. Considering Boolean functions with a single output (*M*=1), it is possible to construct at the most

$$F(N,1)=2^{2^N}$$

different functions, i.e. functions which are different at least in one bit in the output. We have implemented all these functions for two and three inputs. Here, functions where the output does not have a dependence on the *N* inputs, e.g. trivial functions such as constant functions, have been no considered. More importantly, we have found how many functions *F(N,1,C)* can be implemented using different numbers *C* of engineered cells. Fig. S2a-b shows the total number of functions that can be implemented versus the number of cells required for their implementation. As shown by these statistical analyses, a significant increase in the number of functions that can be achieved with a small increase in the number of cells involved in the circuit. For instance, the number of possible Boolean functions with 3-input bits increases from 63 to 128 increasing from two to three cells (Fig. S2b). For just two inputs, the requirements in terms of multicellular complexity are rather small and there is no need for more than three cells (given that only two inputs are possible). Similarly, for *N*=3 the computational capabilities based on different combinations of *C* cells reaches its

maximum close to *C=4*. Increase the number of cells do not allow for a significant increase in the computational capabilities.

Number of wires required for distributed computation. The increasing number of different wires necessary to implement complex functions is one of the most limiting constraints to build cellular circuits. Increasing the circuitry's complexity requires an increase of the number of cells involved, as well as an increase in the number of different wires to connect them properly. In an intracellular environment each wire must be a different biochemical element, e.g. a different protein, in order to prevent undesired crosstalk.

This increasing biochemical diversity can limit the complexity of the constructs that can be implemented in a single cell. Our approach based on multicellular circuits allows for a significant reduction of the wiring requirements. This reduction is due to two key aspects, i) the distributed production of the output, and ii) the functionally complete set of basic functions used, i.e. AND and N-IMPLES functions. The distributed production of the output permits that all cells involved in a circuit respond to two different inputs at the most, one of them is an external input and the other can be the output of other cell of the circuit (wire) or an external input too. As a consequence, in the multicelullar approach there are not "hidden" cells responding only to internal wires (see Fig. 1f). Furthermore, the functional complete set used requires less number of logic gates, and hence lees number of wires, than other standard methods, for instance based on combinations of NAND or NOR gates. Figure S3 shows the number of different functions with 3-inputs that can be implemented versus the number of required wires for their implementation. As figure shows, in our approach (Fig, S3a) the wiring requirements are lower than other standard approaches (Fig S3b,c). As an example, half of the possible functions are already available in our approach using just two wires and that an additional wire already allows covering more than 80% of the possible spectrum. Instead, using either the NAND or NOR implementation, two and three wires allow building just 40 (18%) and 60 (27%) different functions, respectively.

Engineered yeast cell library

Yeast W303 (*ade2-1 his3-11,15 leu2-3, 112 trp1-1 ura3-1 can1*) cells were genetically modified. Schematic genotypic characteristics of each cell and plasmid used are summarized in Tables S1 and S2 (below).

Full description of each cell used in the biological circuits

Cell#1 is a MATα cell that contains *MFα1* and *MFα2* deletions to avoid expression of endogenous α-factor expression and *STE3* deletion to prevent mating with MATa cells within the circuit. It also contain *MFα1* gene under the control of the *STL1* osmo-responsive promoter in the episomal plasmid pRS424*STL1-MFα1* to express α-factor in the presence of NaCl. YCplac195-*fps1Δ1* plasmid encodes a constitutively open version of the Fps1 glycerol channel. *fps1Δ1* mutation is used to increase sensitivity to high osmo-stress, and thus induce higher α-factor expression. This cell implements an IDENTITY function.

Cell#2 is a MATa cell that contains *BAR1* deletion to increase α-factor sensitivity and *FUS3* and *KSS1* deletions to prevent activation of the mating pathway unless *fus3as* is expressed. *GFP* was introduced in the *FUS1* gene locus under its promoter. *GALS::fus3as* construct was integrated to regulate fus3as expression in galactose/glucose growing conditions. *GALS* version of *GAL* promoter was used to prevent leakiness in glucose. *ADGEV* construct encoding the hybrid transcription factor "GEV" (Gal4DBD-hER-VP16 fusion protein) under the control of the *ADH1* promoter was also integrated to regulate *GAL* genes with 17β-estradiol. This cell implements an AND function with 17β-estradiol and an N-IMPLIES function with glucose as input in galactose based circuits.

Cell3# is a MATα cell that contains *MFα1* and *MFα2* deletions to avoid expression of endogenous α-factor expression and *STE3* deletion to prevent mating with MATa cells within the circuit. It contains *MFα1* gene under the control of two TetOperators in the centromeric plasmid pCM183-*MFα1*

that also express the Tet Transactivator. This allows cells to repress α-factor expression in the presence of doxycycline. This cell implements a NOT function.

Cell#4 is a MATa cell that contains *BAR1* deletion to increase α-factor sensitivity and *FUS3* and *KSS1* deletions to prevent activation of the mating pathway unless fus3as is expressed. *GFP* was introduced in *FUS1* gene locus under its promoter. *fus3as* construct with its own promoter was integrated to regulate fus3as activity with 6a inhibitor. This cell implements an N-IMPLIES function.

Cell#5 is a MATα cell that contains *MFα1* and *MFα2* deletions to avoid expression of endogenous α-factor expression and *STE3* deletion to prevent mating with MATa cells within the circuit. *MFα1* gene is under the control of the *GAL1* promoter in the episomal plasmid pBEVY-GU-*MFα1* to express α-factor in galactose. This cell implements an IDENTITY function upon galactose addition or a NOT function in glucose in galactose based circuits.

Cell#6 (reporter cell) is a MATa cell that contains *BAR1* deletion to increase α-factor sensitivity. *GFP* was introduced in *FUS1* gene locus under its promoter. This cell implements an IDENTITY function.

Cell#7 is a MATa cell that contains *BAR1* deletion to increase α-factor sensitivity and *FUS3* and *KSS1* deletions to prevent activation of the mating pathway unless fus3as is expressed. *GFP* was introduced in *FUS1* gene locus under its promoter. *fus3as* gene under the control of 7 TetOperators in the episomal plasmid pRS413TetO7-*fus3as* that also express the reverse Tet Transactivator was introduced to regulate fus3as expression in doxycycline. *STE2* deletion is to prevent *S. cerevisiae* α-factor signaling. *CaSTE2* was expressed from the pAJ1CaSTE2 plasmid to make cells competent for *C. albicans* α-factor signaling. This cell implements an AND function with doxycycline but with *C. albicans* α-factor as a wire.

Cell8# is a MATα cell that contains *MFα1* and *MFα2* deletions to avoid expression of endogenous α-factor expression. *STE3* deletion to prevent mating with MATa cells within the circuit. *MFα1*

gene is under the control of the glucose responsive promoter *HXT1* in the episomal plasmid YEpHXT1-*MFα1*. This cell implements an IDENTITY function.

Cell#9 is a MATa cell that contains *BAR1* deletion to increase α-factor sensitivity and *FUS3* and *KSS1* deletions to prevent constitutive signaling ability. *GFP* was introduced in *FUS1* gene locus under its promoter. *fus3as* gene under the control of seven TetOperators in the integrative plasmid YIpTetO7-*fus3as* that also express the reverse Tet Transactivator was introduced to regulate *fus3as* expression in doxycycline. This cell implements an AND function.

Cell10# is a MATα cell that contains *MFα1* and *MFα2* deletions to avoid expression of endogenous α-factor expression. *STE3* deletion is to prevent mating with MATa cells within the circuit. *MFα1* gene under the control of 2 TetOperators in the centromeric plasmid YCpTetO2-*MFα1* that also express the reverse Tet Transactivator was introduced to regulate α-factor expression in doxycycline. This cell implements an IDENTITY function.

Cell#11 is a MATa cell that contains *BAR1* deletion to increase α-factor sensitivity and *FUS3* and *KSS1* deletions to prevent activation of the mating pathway unless *fus3as* is expressed. *GFP* was introduced in *FUS1* gene locus under its promoter. *fus3as* gene under the control of 7 TetOperators in the integrative plasmid YIpTetOff7-*fus3as* that also express the Tet Transactivator was introduced to repress *fus3as* expression in doxycycline. This cell implements an N-IMPLIES function.

Cell#12 is a MATa cell that contains *BAR1* deletion to increase α-factor sensitivity and *FUS3* and *KSS1* deletions to prevent activation of the mating pathway unless fus3as is expressed. *GFP* was introduced in *FUS1* gene locus under its promoter. *GALS::fus3as* construct was integrated and *ADGEV* construct encoding the hybrid transcription factor "GEV" (Gal4DBD-hER-VP16 fusion protein) under the control of the *ADH1* promoter was also integrated to regulate *GAL* genes with 17β-estradiol. *GAL4* was deleted to prevent activation of *GAL* genes in galactose. This cell implements an AND function.

Cell#13 is a MATα cell that contains *MFα1* and *MFα2* deletions to avoid expression of endogenous α-factor expression. *STE3* deletion is to prevent mating with MATa cells within the circuit. *CaMFα1* gene is under the control of 7 TetOperators in the episomal plasmid YEpTetOff7-*CaMFα1* that also express the Tet Transactivator. *CaMFα1* gene contains the *S. cerevisiae MFα1* signal peptide for secretion and proteolysis followed by just one copy of *C. albicans MFα1* peptide sequence. This allows cells to repress *C albicans* α-factor expression in presence of doxycycline. This cell implements a NOT function.

Cell#14 is a MATα cell that contains *MFα1* and *MFα2* deletions to avoid expression of endogenous α-factor expression and *STE3* deletion to prevent mating with MATa cells within the circuit. *CaMFα1* gene under the control of 7 TetOperators in the centromeric plasmid YCpTetO7-*CaMFα1* that also express the reverse Tet Transactivator was introduced to regulate α-factor expression in doxycycline. *CaMFα1* gene contains the *S. cerevisiae MFα1* signal peptide for secretion and proteolysis followed by just one copy of *C. albicans MFα1* peptide sequence. This cell implements an IDENTITY function.

Cell#15 is a MATa cell that contains *BAR1* and *SST2* deletions to increase α-factor sensitivity and *FUS3* and *KSS1* deletions to prevent activation of the mating pathway unless fus3 is expressed. *GFP* was introduced in *FUS1* gene locus under its promoter. *GALS::fus3* construct was integrated to regulate fus3 expression in galactose/glucose growing conditions. *GALS* version of *GAL* promoter was used to prevent leakiness in glucose. *STE2* deletion is to prevent *S Cerevisiae* α-factor signaling. *CaSTE2* is expressed in the YIpCaSTE2 plasmid to make cell competent for *C albicans* α-factor signaling. This cell implements an AND function.

Cell#16 is a MATa cell that contains *BAR1* deletion to increase α-factor sensitivity and *FUS3* and *KSS1* deletions to prevent activation of the mating pathway unless *fus3as* is expressed. *FUS1::mCherry* was integrated for different output production. *fus3as* gene under the control of 7 TetOperators in the integrative plasmid YIpTetO7-*fus3as* that also express the reverse Tet Transac-

tivator was introduced to regulate *fus3as* expression in doxycycline. This cell implements an AND function.

Table S1. Yeast Strains Used in This Study

name	genotype *	source
CELL#1	*MAT**α** ste3::HIS3 mfα1::LEU mfα2::Kan* pRS424 *STL1 MFα1* YEplac195-*fps1Δ1*	this study
CELL#2	*MAT**a** bar1::HIS3 fus3::LEU2 kss1::TRP1 Nat::PGALS-fus3as::Hph FUS1::GFP::Kan URA3::ADGEV* †	this study
CELL#3	*MAT**α** ste3::HIS3 mfα1::LEU mfα2::Kan* pCM183-*MFα1*	this study
CELL#4	*MAT**a** bar1::HIS3 fus3::LEU2 kss1::TRP1 FUS3as::URA FUS1::GFP::Kan*	this study
CELL#5	*MAT**α** ste3::HIS3 mfα1::LEU mfα2::Kan* pBEVY-GU-*MFα1*	this study
CELL#6	*MAT**a** bar1::HIS3 FUS1::GFP::Kan*	this study
CELL#7	*MAT**a** bar1::Nat fus3::LEU2 kss1::Hph ste2::URA3 FUS1::GFP::Kan* pAJ1-*CaSTE2* pRS413TetO7-*fus3as*	this study
CELL#8	*MAT**α** ste3::HIS3 mfα1::LEU mfα2::Kan* YEpHXT1-*MFα1*	this study
CELL#9	*MAT**a** bar1:: Nat fus3::LEU2 kss1::TRP1 FUS1::GFP::Kan* YIpTetO7-*fus3as*	this study
CELL#10	*MAT**α** ste3::HIS3 mfα1::LEU mfα2::Kan* YCpTetO2-*MFα1*	this study
CELL#11	*MAT**a** bar1::HIS3 fus3::LEU2 kss1::TRP1 FUS1::GFP::Kan YIpTetOff7-fus3as*	this study
CELL#12	*MAT**a** bar1::HIS3 fus3::LEU2 kss1::TRP1 Nat::PGALS-fus3as::Hph FUS1::GFP::Kan gal4::Phl URA3::ADGEV*	this study
CELL#13	*MAT**α** ste3::HIS3 mfα1::LEU2 mfα2::Kan YEpTetOff7-CaMFα1*	this study
CELL#14	*MAT**α** ste3::HIS3 mfα1::LEU2 mfα2::Kan YCpTetO7-CaMFα1*	this study
CELL#15	*MAT**a** bar1::HIS3 kss1::LEU2 ste2::URA3 FUS1::GFP::Kan sst2::Hph Yip-CaSTE2 Nat::PGALS-FUS3*	this study
CELL#16	*MAT**a** bar1:: Nat fus3::LEU2 kss1::Hph FUS1::URA3::Kan pRS304FUS1-mCherry YIpTetO7-fus3as*	this study

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

† "ADGEV" denotes a construct encoding the hybrid transcription factor "GEV" (Gal4DBD-hER-VP16 fusion protein) under the control of the *ADH1* promoter [5].

Table S2. Plasmids Used in This Study

name	description	source
pRS424STL1-*MFα1*	2 micron *TRP1 PSTL1- MFα1*	this study
pIU-ADGEV	integrating *URA3 PADH1*-[*GAL4DBD-hER-VP16*]	[5]
pCM183-*MFα1*	CEN *TRP1* tTA transactivator TetO2-*MFα1*	this study
pBEVY-GU-*MFα1*	2 micron *URA3 PGAL1- MFα1*	this study
pAJ1-*CaSTE2*	2 micron *TRP1 PGPD-CaSTE2*	[6]
YIp *CaSTE2*	integrating *TRP1 PTDH3- CaSTE2*	this study
YIpTetO7-*fus3as*	integrating *HIS3* reverse tTA transactivator TetO7-*fus3as*	this study
YEpHXT1-*MFα1*	2 micron *URA3 PHXT1-MFα1*	this study
YCpTetO2-*MFα1*	CEN *TRP1* reverse tTA transactivator TetO2-*MFα1*	this study
YEplac195-*fps1Δ1*	2 micron *URA3 fps1 Δ13-230*	[7]
pRS413TetO7-*fus3as*	CEN *HIS3* reverse tTA TetO2-*fus3as*	this study
YIpTetOff7-fus3as	integrating *URA3* tTA transactivator TetO7-*MFα1*	this study
pRS304FUS1-mCherry	integrating *TRP1 PFUS1-mCherry*	Donated by Dr. Kuchler
YEpTetOff7-CaMFα1	2micron *URA3* tTA transactivator TetO7-Ca*MFα1* (one copy)	this study
YCpTetO7-CaMFα1	CEN *TRP1* reverse tTA TetO7-Ca*MFα1* (one copy)	this study

Characterization of the cellular properties of the engineered cells of the library

In digital computation a clear separation between the logic state 0 and 1 is critical. This separation can be done defining a single threshold that separates states (as CMOS electronic devices do) or using two different thresholds, the low and the high (TTL electronic devices). In the second case there is a gap between both thresholds, which corresponds to an undefined state. Whereas in the first method to encode digital information a stochastic fluctuation can push the system from 0 to 1 (or vice versa), in the second case the logic sates are much more robust in terms of noise fluctuations because changing the state requires cross the gap region. In electronics it is enough a 6-fold increase from the low to the high state for a proper definition of the thresholds with a significant gap region. For instance, the family of TTL devices[8] works in a range from 0 to 5

volts. A voltage below 0.5 volts is considered 0 logic, whereas voltages above 3 volts (up to 5 volts) are 1 logic. Hence, in TTL devices 0 logic is less than 10% of the maximal voltage. As shown in Figures 2-4 and Fig. S6-7 and 9, in our biological devices the resolution of the 0 logic is less than 10% 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 Curve, i.e. the cellular response with respect to different input levels. An adequate transfer curve should be characterized by several key features[9-10], 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 for 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. Figure S5 shows the full set of transfer curves for each cell. All these curves exhibit the proper shape to be logic blocks for a multicellular implementation. In each 1-input 1-output cell the gain has been calculated[10]. All the functions have a gain above 7.6, which guarantee the maintenance of the signal in all the circuits. Cells that produce a diffusible molecule (alpha-factor) that acts as a wire. To detect the secretion of these molecules producer cells were mixed with a reporter cells expressing GFP upon alpha-factor. This procedure allows to characterizing not only the cellular behaviour but the wire efficiency.

The FACS data presented in the Supplemental data set 2 (a-m) displays the single cell measurements of output production upon stimuli. As data shows, the noise margins show an adequate separation between high and low logic levels.

Finally, Supplemental data set2 (n-z) shows the individual cellular response of each cell in response to the different inputs they encounter within a circuit. The experimental data clearly demonstrate that there are no undesired cross-talks and each cell responds only to the expected input. Similarly, circuits that contain two-wiring molecules (alpha factor from *S. cerevisiae* and from *C. albicans*) do not display crosstalk between cells responding to each one of them.

Experimental Methods

Engineered yeast strains and plasmids. W303 (*ade2-1 his3-11,15 leu2-3, 112 trp1-1 ura3-1 can1*) cells were genetically modified to be able to produce alpha factor from an inducible promoter. Strains were *mfα1* and *mfα2* to prevent endogenous production of alpha factor. *STE3* was deleted to avoid mating. Output cells were bar1 to increase sensitivity to pheromone. Some strains were *fus3* and *kss1* deletions to prevent mating signaling unless FUS3 was induced. GFPwas integrated in the *FUS1* ORF locus. Modified *C. albicans* alpha factor was expressed as a wiring molecule and detected by yeast expressing *CaSTE2*. A complete list of plasmids and strains is included above (Fig. S4; text and tables).

Computational output detection. For doxycycline and galactose induction/repression of promoters, cells were grown o/n separately with or without the input. After o/n growth in appropriate selective medium, cells were grown to mid exponential phase in YPD or YPGal, mixed at indicated different proportions and inputs were added at once (0.4M NaCl, 70μM 17β-estradiol (SIGMA), 10 μM 6a, 10μg/ml doxycycline (SIGMA) or 5 μg/ml *C. albicans* α-factor). For reference samples 2μg/ml *S. cerevisiae* α-factor was added. Cells were incubated at 30ºC for 4 hours, and analyzed by flow cytometry (FACScalibur Becton Dickinson). For 2D transfer functions contour plots were created with Sigma Plot.

Microscopy based microfluidic platform. Cells were loaded into Y04 plates (CellASIC). Cell loading into a chamber was performed at 8 psi and constant flow was maintained at 5 psi during position selection. During experiments, flow was reduced to 1 psi to reduce washing out of the produced alpha factor. Images were collected using NIS elements Software (Nikon) and a Nikon Eclipse Ti Microscope and, analyzed with Matlab.

References to SI

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

2. Quine, W. V. Methods of Logic, 4th edition. *Harvard University Press, Cambridge, MA* (1982).

3. Goodstein, R. L. Boolean Algebra. Dover *Publications, Inc. Minola, New York* (2007).

4. Yokobayashi, Y., Weiss, R. & Arnold, F.H. Directed evolution of a genetic circuit. *Proc. Natl. Acad. Sci. USA*. **99**, 16587-16591(2002).

5. Takahashi, S., Pryciak, P. M. Membrane localization of scaffold proteins promotes graded signaling in the yeast MAP kinase cascade. *Curr. Biol*. **18**, 1184-91 (2008).

6. Janiak, A. M., Sargsyan, H., Russo, J., Naider, F., Hauser, M. & Becker, J. M. Functional expression of the *Candida albicans* alpha-factor receptor in *Saccharomyces cerevisiae*. *Fungal Genet Biol.* **42**, 328-38 (2005).

7. Hedfalk, K., Bill, R. M., Mullins, J. G. L., Karlgren, S., Filipsson, C., Bergstrom, J., Tamás, M. J., Rydström, J. & Hohmann, S. A Regulatory Domain in the C-terminal Extension of the Yeast Glycerol Channel Fps1p. *J. Biol. Chem.* **279**, 14954-14960 (2004).

8. Millman, J. Microelectronics Digital and Analog Circuits and Systems. *New York. McGraw-Hill Book Company* (1979).

9. Weiss, R., Basu, S., Kalmbach, A., Karig, D., & Netravali, I. Genetic Circuit Building Blocks for Cellular Computation, Communications, and Signal Processing. *Natural Computing: an international journal* **2**(1), 47 – 84 (2003).

10. Feng, X., Hooshangi, S., Chen, D., Li, G., Weiss, R. & Rabitz, H. Optimizing Genetic Circuits by Global Sensitivity Analysis. *Biophysical Journal* **87**, 2195–2202 (2004).