

# Multicellular Computing Using Conjugation for Wiring

Angel Goñi-Moreno<sup>1\*</sup>, Martyn Amos<sup>2</sup>, Fernando de la Cruz<sup>3</sup>

**1** Systems Biology Program, Centro Nacional de Biotecnología CSIC, Cantoblanco-Madrid, Spain, **2** School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Manchester, United Kingdom, **3** Departamento de Biología Molecular e Instituto de Biomedicina y Biotecnología de Cantabria (IBBTec), Universidad de Cantabria-CSIC-SODERCAN, Santander, Spain

## Abstract

Recent efforts in synthetic biology have focussed on the implementation of logical functions within living cells. One aim is to facilitate both internal “re-programming” and external control of cells, with potential applications in a wide range of domains. However, fundamental limitations on the degree to which single cells may be re-engineered have led to a growth of interest in multicellular systems, in which a “computation” is distributed over a number of different cell types, in a manner analogous to modern computer networks. Within this model, individual cell type perform specific sub-tasks, the results of which are then communicated to other cell types for further processing. The manner in which outputs are communicated is therefore of great significance to the overall success of such a scheme. Previous experiments in distributed cellular computation have used global communication schemes, such as quorum sensing (QS), to implement the “wiring” between cell types. While useful, this method lacks specificity, and limits the amount of information that may be transferred at any one time. We propose an alternative scheme, based on specific cell-cell conjugation. This mechanism allows for the direct transfer of genetic information between bacteria, via circular DNA strands known as plasmids. We design a multicellular population that is able to compute, in a distributed fashion, a Boolean XOR function. Through this, we describe a general scheme for distributed logic that works by mixing different strains in a single population; this constitutes an important advantage of our novel approach. Importantly, the amount of genetic information exchanged through conjugation is significantly higher than the amount possible through QS-based communication. We provide full computational modelling and simulation results, using deterministic, stochastic and spatially-explicit methods. These simulations explore the behaviour of one possible conjugation-wired cellular computing system under different conditions, and provide baseline information for future laboratory implementations.

**Citation:** Goñi-Moreno A, Amos M, de la Cruz F (2013) Multicellular Computing Using Conjugation for Wiring. PLoS ONE 8(6): e65986. doi:10.1371/journal.pone.0065986

**Editor:** Mark Isalan, Center for Genomic Regulation, Spain

**Received:** March 5, 2013; **Accepted:** May 1, 2013; **Published:** June 20, 2013

**Copyright:** © 2013 Goñi-Moreno et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Funding:** This work was supported by the European Commission FP7 Future and Emerging Technologies Proactive initiative: Bio-chemistry-based Information Technology (CHEM-IT, ICT-2009.8.3), project reference 248919 (BACTOCOM). Work in FdIC lab was supported by Spanish Ministry of Education (BFU2011-26608), and European VII Framework Program grants num 248919/FP7-ICT-2009-4 and 282004/FP7-HEALTH.2001.2.3.1-2. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The authors have declared that no competing interests exist.

\* E-mail: agoni@cnb.csic.es

## Introduction

The growing field of synthetic biology [1–4] is concerned with the application of engineering principles, concepts and techniques to the modification and/or construction of biological systems. This (re-)engineering may be motivated by a desire to better understand the underlying biological substrate, or by novel applications of biological “devices”. Although the field traces its roots back to early work on genetic engineering, it emerged as a serious research area at the turn of the century, with the simultaneous publication of two significant papers. The first, by Elowitz and Leibler [5], described a fully-synthetic genetic oscillator engineered into the *E. coli* bacterium. Their “repressilator” relied on mutual repression of a “loop” of connected genes in order to achieve oscillation. The other paper, due to Gardner, Cantor and Collins [6], outlined the design and construction of a synthetic toggle switch (also in *E. coli*), the state of which could be “flipped” from outside by either chemical or thermal induction. Both of these constructions are now standard motifs in the design of synthetic biological systems, and provided inspiration for the construction of a number of genetic devices [7–9]. However, just as the pioneers of computer technology quickly incorporated the early transistor into larger

circuits in order to build the first solid-state computers, researchers in synthetic biology rapidly sought to build ever larger devices using these gene-based components.

There exists, though, a fundamental limitation on the amount and type of novel genetic “circuitry” that may be introduced into a single living cell. As the authors of [10] argue, “...establishing the wiring of an electronic circuit just requires linking each pair of connected elements by a wire (e.g. a piece of copper). But inside a cell, the cables need to have a different implementation: different proteins must be used for each different pair. Additionally, because of the intrinsic difficulties of implementing them, the resulting constructs are usually specific for the given problem and cannot be reused afterwards”. Because of these related issues of cross-talk and lack of modularity, many researchers now seek an alternative approach. By expanding the scope of synthetic biology beyond *single-cell* solutions, and into the domain of *multicellular* systems, we seek to harness the inherent power of biological “nanotechnology”, but in a way that readily allows for scalability, noise tolerance and component reusability. With this in mind, attention is turning to the engineering of microbial *consortia* [11]; *multiple* populations of microbes that can interact to perform functions beyond those

achievable by *individual* populations. The power of such consortia is derived from both their *robustness* and their ability to perform complex tasks in a *distributed* fashion. These attributes are due to two main features; *communication* between consortium members (that is, the exchange of information between individuals), and *division of labour* (the overall behaviour of the system is the result of combining many sub-tasks, each performed by individuals or sub-populations) [11].

Recently, the notion of distributed, multicellular computation using engineered cells has gained increasing traction [10]. One of the first such systems was demonstrated by Basu *et al.* in 2005 [12], in which the authors demonstrated programmed pattern formation, using populations of engineered “sender” and “receiver” cells. More recently, Tamsir *et al.* [13] showed how simple logic gates may be constructed by programming communication between bacterial colonies, and Regot *et al.* [14] demonstrated a similar system in yeast. Central to the implementation of multicellular computation is controlled *communication* between cells and populations of cells. So far, this has *generally* been implemented using the *global* communication capabilities offered by *quorum sensing* (QS) [13–15] (although the use of bacteriophage has also recently been proposed [16]). Within the QS system, one cell (sender) uses small signalling molecules that diffuse over distance and thus reach other cells (receivers) [17]. Apart from the implementation of logical functions, QS has been used for other purposes, such as the synchronization of engineered oscillators [18]. However, we believe that multicellular computation will greatly benefit from a more *varied* range of communication protocols. As Ortiz and Endy highlight [18], QS-based communication is limited by both the *type* and *content* of messages that are possible using chemical signals.

We therefore propose a scheme for multicellular computation based on a *local* communication protocol. The foundation for this is the process of genetic exchange between bacteria known as *bacterial conjugation* [19], which has often been likened to “bacterial sex” [20,21]. During conjugation, two cells establish a direct, bridge-like connection, called the *pilus*, which brings the cells together. A separate channel is then opened in the respective cell walls, through which a single DNA strand is transferred from the *donor* cell to the *recipient* cell [20]. The importance of this transfer process is that it facilitates the transmission of large, specific genetic messages, which can have arbitrary content. We therefore seek to harness its potential in order to facilitate communication within an engineered cellular population.

In order to achieve this goal, we use *site-specific recombination systems* [22], which allow individual cells to dynamically rewrite their DNA “message”. Recently, three bacterial site-specific recombination systems have been used to implement biocomputing devices. The Cre/*lox* system was used to engineer a genetic switch [23], and the *fim* system was used to engineer both a multiplexer [24] and a sequential switch-based memory [25,26]. The latter example made use of the Hin/*hix* system, which was also used to solve various small instances of mathematical problems in bacterial populations [27,28]. These implementations demonstrate the power and applicability of such site-specific recombination systems. Furthermore, as this manuscript was under preparation, Siuti *et al.* [29] published their recombinase-based approach to the implementation of logic and memory functions in *E. coli*, underlining the utility of this approach. We present the results of extensive simulation-based experiments, which support the in-principle feasibility of our approach. This work offers a firm foundation for experimental investigations into distributed multicellular computation using conjugation as a core “technology”.

## Results

We first describe the communication mechanism through conjugation by showing how it may be used, in principle, to implement a single Boolean NOR function. Secondly, we expand the concept by designing a distributed population to implement the exclusive OR (XOR) function, based on mixing three bacterial strains with individual NOR functionality. While using only one site-specific recombination system is sufficient for the NOR-based approach, a combination of two is necessary for the adequate functioning of the XOR computation. For clarity of description, we use abstract labels for the components (except those involved in the recombinase-based logic systems).

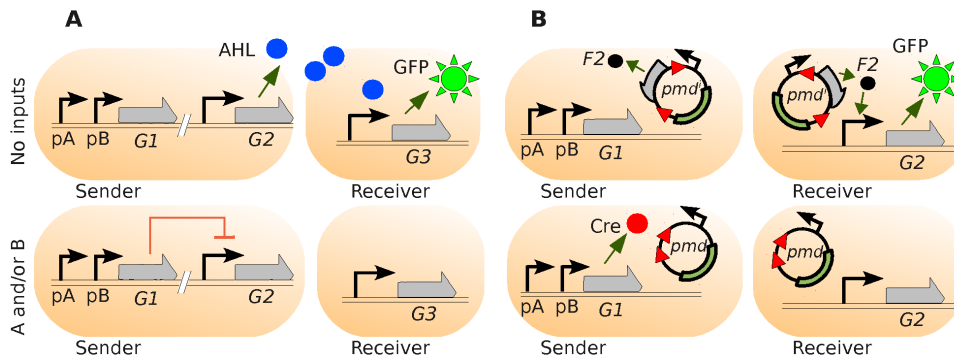
### Wiring with conjugative plasmids: 2-strain NOR population

Recall that the two-input NOR function is a negated OR, and thus returns the value 1 if and only if both inputs are zero, and 0 in all other cases. Figure 1A shows how a NOR gate may be constructed using two engineered bacterial strains that communicate via the QS molecule AHL (as in [13]). The evaluation of the NOR function is executed by a *sender* strain, and the output is sent to the *receiver* strain via QS. In this way, QS acts as a *wire* connecting the two components. When no inputs are present (top row of 1A), gene *G1* is *off* (not being transcribed), and the AHL signalling molecules (controlled by a repressible promoter) are expressed by *G2*. These molecules arrive at the *receiver* cell and, after binding to the corresponding transcription factor, induce expression of gene *G3* (the reporter gene). This latter product is read as the output of the NOR logic function (fluorescence detection of the amount of green fluorescent protein, GFP). On the other hand, when one or more inputs are present (bottom row, “A and/or B”), their corresponding promoter is activated (*pA* and/or *pB*), and gene *G1* expresses a repressor, which in turn inhibits the production of AHL. Thus, the *receiver* cells are *off*, and no fluorescence is observed.

Our *alternative* design for a distributed NOR gate using conjugation as wiring is shown in Figure 1B. As before, the sender cell computes the NOR function and communicates the result to the receiver cell. The key difference with the previous scheme lies in the communication mechanism; here, we use *conjugative plasmids* [30], rather than QS, to transmit a result. Conjugative plasmids are circular strands of DNA that may be transferred between bacterial cells during the process of conjugation [20,31]. Although, in the wild, these plasmids allow bacteria to exchange potentially useful genes, here we use them for the transmission of *logical values*.

When no inputs are present (Figure 1B(top row)), promoters *pA* and *pB* are not induced, and the expression product of *G1* (in this case, protein Cre) is not present in the cell. As a result, a constitutive promoter in the plasmid *pmd<sup>i</sup>* (a plasmid that produces inducers) expresses its downstream gene, which results in inducer *F2*.

When the sender cell comes into contact and conjugates with a recipient, thus forming a “wiring” connection, the plasmid is transferred into the receiver cell. Inducers *F2* are then produced inside the receiver by plasmid *pmd<sup>i</sup>*, which are in charge of inducing the expression of the reporter gene (*G2*). If the resulting product is the green fluorescent protein (GFP), the fluorescence (output) is *turned on*. Otherwise (one or more inputs, bottom row), the protein Cre is expressed, which *deletes* a specific DNA segment surrounded by *lox* sites in the plasmid. As a result, the plasmid *pmd<sup>i</sup>* is converted into *pmd* (a plasmid that is unable to produce inducers, due to the lack of the *lox*-flanked DNA segment). When

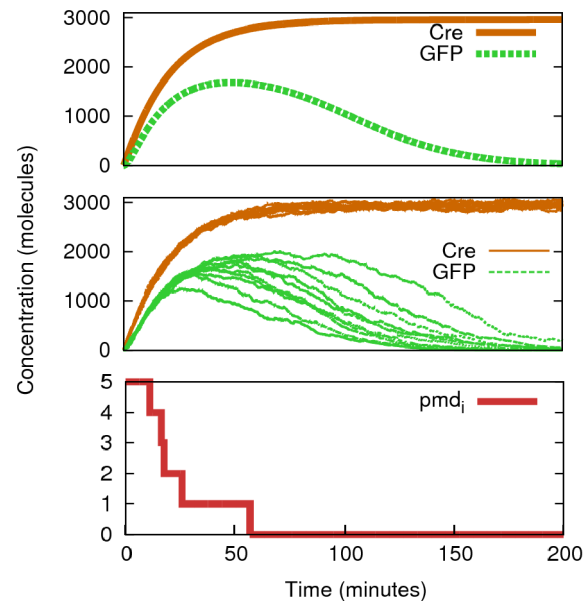


**Figure 1. Intercell connection of a NOR logic function.** **A.** Traditional wiring through AHL signalling. When no inputs (*A* and *B*) are present (upper row), the AHL quorum-sensing effectors (expressed by gene *G2* and controlled by a repressible promoter) are produced by the Sender cell and the light (green fluorescent protein, GFP, transcribed by an inducible promoter) is switched on in the Receiver cell. On the other hand, when one or more inputs are present (lower row), AHL production is repressed (via the expression product of gene *G1*) and the fluorescence switched off. **B.** Wiring by using conjugation. In the 0-0 case (no inputs, upper row) the plasmid  $pmd^i$  ( $i$  = inducer) travels from the Sender to the Receiver cell without modification and induces the expression of GFP. In any other case, the protein Cre is expressed (from gene *G1*) in the Sender, which irreversibly alters the plasmid by deleting the segment between *lox* sites (red triangles). The modified plasmid cannot produce inducers (*F2*), and the Receiver is switched off.  
doi:10.1371/journal.pone.0065986.g001

transferred into the receiver cell, this plasmid will *not* express inducer *F2*, and fluorescence emission will be switched off. Thus, the conjugation wire successfully transmits the signal via a plasmid.

It is important to note that when the modified plasmid  $pmd$  is produced - via deletion - in the sender and transferred to the receiver, it cannot coexist in the latter strain with the previously-introduced unaltered plasmids  $pmd^i$  (before the input signals were introduced to the population). Incompatibility is a property of plasmids that contain the same replication genes [32], and is used here to induce “competition”. In this scenario, the population of plasmids in the receiver cells gradually shifts towards the “new” (transmitted) plasmid, as the latter gradually replaces the original. Since both plasmids are viewed as identical from the point of view of replication control, the copies of  $pmd^i$  and  $pmd$  compete for an effective resource - the number of plasmids that are allowed in the cell, referred to as the *copy number*. Since copies of  $pmd$  are continuously “pumped” to receiver cells, in practical terms they eliminate  $pmd^i$ . As we see later, this fact has been incorporated into the model, and results in a progressive change in the plasmid population of the receiver, from  $pmd^i$  to  $pmd$ .

**Single cell behaviour.** The simulated single-cell behaviour of this design is observed, over time, in Figure 2, where the logic case 1-1 (both inputs present) is applied to a population initially in the 0-0 state. In this idealised set of simulations, we consider the existence of only one cell of each kind (sender and receiver). We also assume a constant conjugation process (as if the wire was permanently *connected*), so that both cells share the plasmid populations (a theoretical ideal state induced in order to test the individual components of the system). As we observe in both deterministic (top) and stochastic (middle) plots, we see a short *flash* from the receivers (GFP initially being expressed), while the expected output should be “0” (corresponding to the NOR 1-1 case). This is due to the fact that from time 0, the senders transfer plasmid  $pmd^i$  (in its initial configuration) into the receivers. Thus, inducer *F2* is initially expressed. As Cre is produced, it transforms the plasmid  $pmd^i$  into  $pmd$  (the transformation of  $pmd^i$  being seen in the bottom graph). When no  $pmd^i$  plasmids remain in the ideal system studied in Figure 2, the amount of GFP is controlled only by degradation, and the desired output is reached. Other molecular relations corresponding to the deterministic simulation are shown in Figure S1.

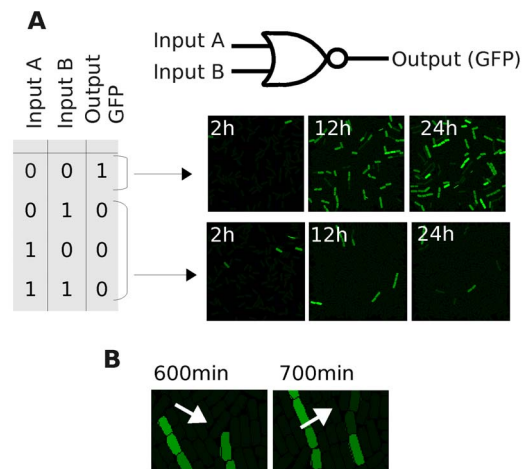


**Figure 2. Deterministic and stochastic time evolution of the 1-1 logic case.** These results show the simulation of Figure 1B with input molecules (*A* and *B*) set up to 500 (each) ( $k_{11} = k_{12} = 500$  molecules  $\text{min}^{-1}$ ) and the copy number of the plasmid to 5. By monitoring (deterministically, upper graph, one run; stochastically, middle graph, ten runs) proteins Cre (in the Sender) and GFP (in the Receiver) we see how GFP production is initially triggered by those plasmids that have not been modified yet ( $pmd^i$ ). When Cre has been functioning long enough ( $t \approx 57$  min in this simulation) the remaining GFP is only controlled by degradation rates, as no more fluorescent proteins are being expressed. Lower graph shows the stochastic evolution of  $pmd^i$  over time, which determines the delay in displaying the correct output according to the NOR logic function (0 output for the 1-1 case).  
doi:10.1371/journal.pone.0065986.g002

**Multicellular behaviour.** A more realistic simulation, which considers physical interactions between cells of a population, is shown in Figure 3. While the cell’s logic is simulated via deterministic equations, plasmid numbers ( $pmd^i$  and  $pmd$ ) are

discrete (see Section Methods for details on discretization). Importantly, deterministic behaviours are not appropriate for low molecular concentrations, and can result in unrealistic behaviours (Figure S2 shows a further study on this issue). Two populations are grown into surfaces with a different distribution of inputs over 24 h. The first (Figure 3A top) has no inputs present and, as a result, recipient cells (receivers) display green light (GFP) corresponding to a logic “1” in output (images from Movie S1). The second population is grown on a simulated surface containing *both* inputs (A and B, Figure 3A) (images from Movie S2). In the latter simulation, we clearly see the effects that the initial *flash* (explained in Figure 2) has on the output of the circuit. While the plasmids  $pmd^i$  are transformed in the donors at an early stage of the process, those that were initially copied into the receivers are constantly being replicated (until they reach their respective copy number) in the population. This *competition* between  $pmd$  acquisition and  $pmd^i$  replication (in receivers) results in fluorescent cells after 24 h. However, this fluorescence eventually decays (plasmids  $pmd^i$  are eventually lost), and we clearly observe the output “0”. Figure 3B shows the process of *losing* plasmid  $pmd^i$  (due to the proliferation of cells without any copy of  $pmd^i$ ). At time 930 min, a fluorescent cell has only one copy of  $pmd^i$ , due to the lack of further replication of the existing plasmids in the cell (the rest of the copies are  $pmd$ ). After division (at 950 min) the  $pmd^i$  plasmid will go to one of the daughters (all the plasmids are shared randomly). This cell cannot maintain the level of fluorescence after a few min (GFP decay; 990 min). Almost identical behaviour to that seen in input case 1-1 is observed in cases 0-1 and 1-0 (data not shown), as Cre is also expressed in sender cells.

By varying the conjugation frequency we observe changes in the quantitative information of the system, while qualitative behav-



**Figure 3. Spatial behaviour of the NOR “wire” over 24 h. A.** Truth table of the NOR “wire” represented in Figure 1B where inputs A and B can be “0” (0 molecules) or “1” (1000 molecules). Two simulated cell populations are monitored at 2, 12 and 24 h, where the intensity of green colour is directly proportional to the concentration level of GFP (green fluorescent protein). The first one (upper row) corresponds to the logic case 0-0, resulting in an output “1” (lights on). The second one (lower row) corresponds to the logic case 1-1 (similar functioning for 0-1 and 1-0 cases) resulting in lights off. **B.** Detail of the 1-1 case simulation where the plasmid with inducer ( $pmd^i$ ) is being replaced by the modified version due to inflow (conjugation) and replication after division. Numbers in cells represent the amount of  $pmd^i$  plasmids. After division of a cell with only 1  $pmd^i$ , one of the daughter cells will have no copies of this set. Thus, GFP will decay. doi:10.1371/journal.pone.0065986.g003

ours are still the same. Increasing this frequency (Figure S3) causes cells to be overloaded by plasmids (above their copy number), and the replication of plasmids becomes less frequent. For details about cell features, including conjugation frequencies, please see the Methods section.

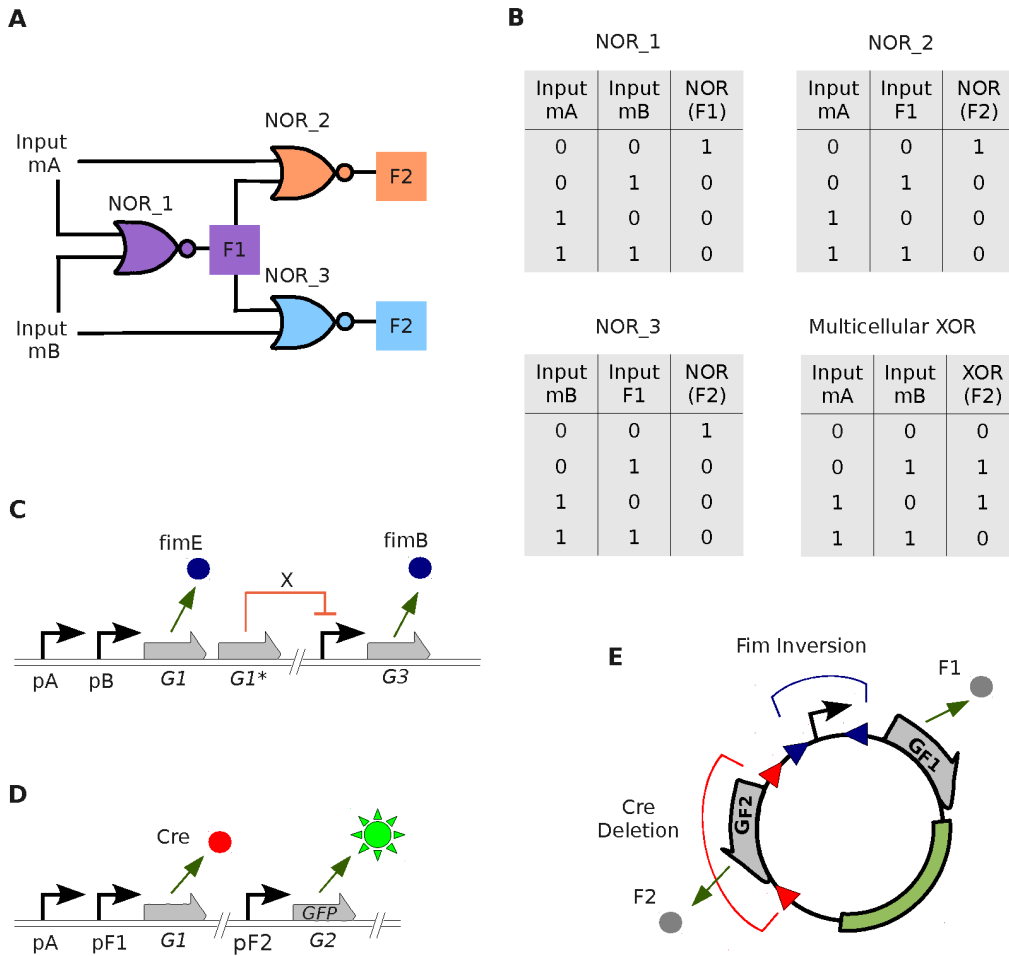
### Extending the approach: 3-strain XOR population

We now show how collections of NOR gates may be connected together in order to compute the exclusive OR (XOR) function, as in [13]. Figure 4 shows the design of a three-strain distributed XOR population, as a proof of principle of the extensibility of the initial approach. The XOR (exclusive OR) function outputs “1” when the inputs are different and “0” when the inputs are the same. Importantly, the inputs are found as *clear* digital values, which are either abundant (“1”) or non-existent (“0”). This avoids potential problems with *half-values* [33]. This logic function may be simulated by connecting three NOR gates, as shown in Figure 4A, which is the configuration of our engineered three-strain population. Exactly the same scheme was used in [13], only with QS molecules connecting the gates. The cells named NOR\_1 are the “donors” of the community, and their inputs correspond to the inputs of the whole XOR function (molecules A and B). Their output, F1 (Figure 4A) encodes the information that travels through the “wire” towards the next logic gates NOR\_2 and NOR\_3. These latter gates take F1 as one of their inputs, and either A or B (respectively) as the other. The output of NOR\_2 and NOR\_3 cells, which we call F2 (the same signal), is the final output of the XOR function. In Figure 4B we show the truth tables of individual strains as well as the emergent XOR logic.

The configuration of NOR\_1 cells is depicted in Figure 4C. If no inputs are present, gene G3 - controlled by a repressible promoter - produces FimB. This causes random inversions of the promoter in the plasmids (Figure 4E). As a result, about half of the plasmids will have the promoter pointing towards the gene named  $G_{F1}$ , while the rest will have the promoter pointing towards gene  $G_{F2}$ . Thus, there will be a positive amount of molecule F1 in the environment. This output is taken as a logical “1”. Otherwise, inducible promoters  $pA$  and/or  $pB$  switch *on* genes G1 and G1\*, which produce FimE and repressor X respectively. Repressor X inhibits the expression of G3. Thus, FimB will disappear from the sender cells, and the presence of FimE causes directional inversion in the promoter, which will then point towards gene  $G_{F2}$ . There will thus be a lack of expression products F1 in the environment, which represents a logical “0”. Importantly for our design, as noted in [24], the FimE protein causes directional inversions with nearly 100% fidelity, while FimB causes inversions on both directions equally well [34,35].

Figure 4D shows the inside *program* of NOR\_2 cells. Importantly, this configuration is identical to the combination of the senders and receivers of Figure 1B. If no inputs are present, which means that input molecule A is not in the environment and molecule F1 is not being produced by the plasmid (all promoters are pointing at gene  $G_{F2}$ ), the protein Cre is not expressed. Thus, the segment surrounded by *lox* sites in the plasmid is not deleted. Molecules F2 are then effectively expressed and, in turn, induce the production of GFP. If either one or both inputs are present (because A is present and/or F1 is being expressed by one of the plasmids), protein Cre is expressed. This causes a modification in the plasmid, which loses the  $G_{F2}$  gene region by deletion. No inducers of GFP are present and the fluorescence is *switched off*.

**Simple behaviour.** A simplified simulation (not considering spatial factors or physical dynamics) of the donor (NOR\_1) cells, according to a specific input profile over time, is shown in Figure 5. Input case 1-1 (both inputs present) is induced at 0 min by



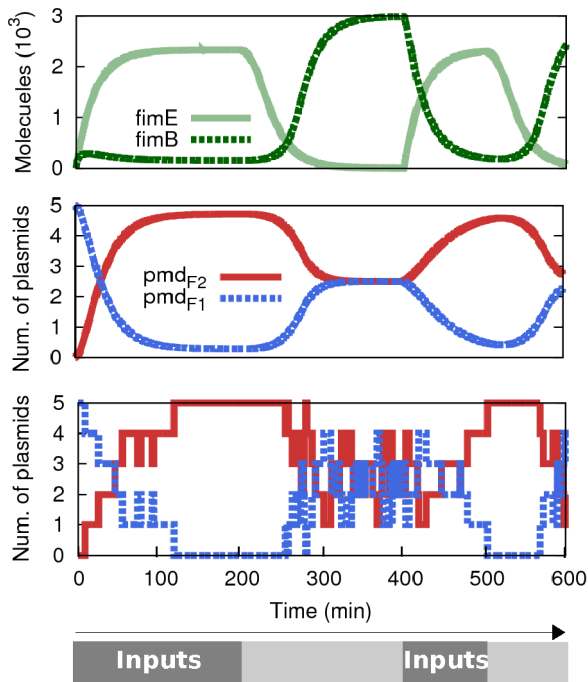
**Figure 4. Multicellular design of a distributed XOR circuit.** **A.** Schema of the three-strain population and its connections. Cell NOR<sub>1</sub> (donor) takes its inputs from the concentration of molecules A and B (inputs of the whole circuit) and its output is named “F1” (which is a specific plasmid configuration). The inputs for cell NOR<sub>2</sub> (recipient) are molecule A and F1. Similarly, the inputs for cell NOR<sub>3</sub> (recipient) are B and F1. The output for the XOR function, “F2” is the combination of outputs from NORs 2 and 3 (green fluorescent protein, GFP, in this example). **B.** Truth tables for each single gate and for the full XOR circuit. **C.** Inside logic of NOR<sub>1</sub>. In the case 0-0 (no inputs present), gene *G3* expresses *FimB* which, in turns, alters the plasmid present in that strain by inverting the promoter in random direction. In any other case, gene *G1* expresses *FimE* which inverts the promoter region of the plasmid pointing towards gene “F2”; While gene *G1\** expresses repressor *X* which stops the production of *FimB*. **D.** Inside logic of cell NOR<sub>2</sub>. Once the plasmid is in the cell (wire connected), if no inputs present, GFP is expressed. Otherwise, *Cre* is produced (inducers *A* bound to inducible promoter *pA* and/or inducer *F1* bound to *pF1*) which deletes gene *F2* from the plasmid. (Cell NOR<sub>3</sub> has the same circuitry but it is sensitive to input *B* instead of *A*). **E.** Wiring plasmid. Due to *Fim* inversion, the promoter can point towards *G<sub>F1</sub>* or *G<sub>F2</sub>*. Due to *Cre* deletion, gene *G<sub>F2</sub>* can be removed.  
doi:10.1371/journal.pone.0065986.g004

introducing the inputs to the population and maintaining their concentration. At time 200 min we stop introducing inputs, and their concentration is only controlled by degradation rates until case 0-0 is reached. From time 400 min to 500 min we repeat the process. According to that input profile, we observe deterministic oscillation of *FimE*/*FimB* (top graph), as well as oscillation between the two possible plasmids in NOR<sub>1</sub> (middle graph) (*pmd<sub>F2</sub>*, plasmid where the promoter points towards *G<sub>F2</sub>* and *pmd<sub>F1</sub>*, plasmid where the promoter points towards *G<sub>F1</sub>*). The latter relation is shown stochastically (bottom graph), with a copy number of 5 plasmids, in order to show realistic transitions within discrete numbers. In all graphs we clearly observe the delay produced between the time we stop introducing inputs and the time they effectively disappear from the system. It is important to note that the initial distribution of plasmids (*t*=0) is no longer repeated in the simulations. That pristine condition is only valid at the very beginning, and is not representative of the working

system. From this simulation we can clearly see how the logic is fully reversible, requiring only changes to inputs, with no modification to the cells. This feature is possible precisely because of the reversible nature of the *fimE*/*B* system (i.e., it does not delete the segment).

**Spatial simulation.** A full spatial simulation of the XOR population is shown in Figure 6. The three different strains are grown over a surface that contains the corresponding combination of inputs (the four logic cases shown). The visual output is represented by the intensity of green colour (high = “1”; low = “0”) proportional to the concentration of GFP in each cell. All snapshots are taken after 32 h cultivation (cells have a doubling time of approximately 100 min). As expected, cases 0-1 and 1-0 are the ones that display a positive output following the XOR function. Below each snapshot, we show a graph that pictures the number of plasmids of each four possibilities (*pmd<sub>F2</sub>*, promoter pointing towards *G<sub>F2</sub>*; *pmd*, promoter pointing towards *G<sub>F2</sub>* after its





**Figure 5. Time evolution of NOR\_1 according to a specific input profile.** Logic case 1-1 is induced during the intervals [0...200] min and [400...500] min (A and B=500 molecules -constant entry- during the interval). The case 0-0 domains during the rest of the 600 min. According to that profile we observe the deterministic oscillation of FimE/FimB (top graph) as well as the oscillation between the two possible plasmids in NOR\_1 ( $pmd_{F2}$  and  $pmd_{F1}$ ). The latter relation is shown deterministically (middle graph) and stochastically (bottom graph) (copy number=5). Delays in response are due to input degradation times.

doi:10.1371/journal.pone.0065986.g005

deletion;  $pmd_{F1,2}$ , promoter pointing towards  $G_{F1}$ ;  $pmd_{F1}$ , promoter pointing towards  $G_{F1}$  after gene  $G_{F2}$  deletion). The number of plasmids is averaged per cell within the receiver population (cells NOR\_2 and NOR\_3) at the same time as the snapshot is taken (32 h).

The phenomenon of unwanted light *flashes* still occurs at this point in NOR\_2 and NOR\_3. As described previously, the presence of this phenomenon does not affect the overall functioning of the circuit. Also, conjugation frequency plays an important role in the quantitative behaviour. However, qualitative functioning is still that of an XOR function (Figure S4).

## Discussion

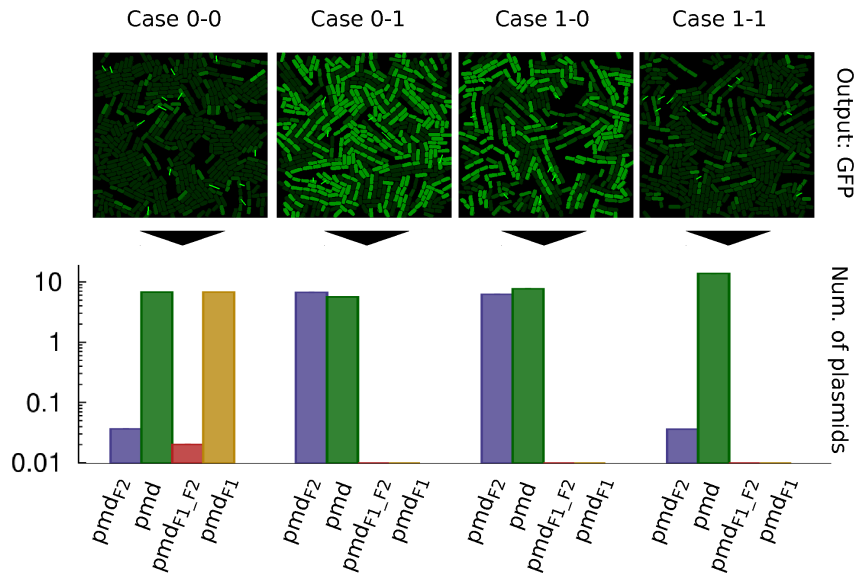
It seems increasingly clear that a significant amount of future research in the field of synthetic biology will be concerned with the construction of engineered microbial consortia [10,11,14]. The growing complexity of proposed applications (e.g., in energy, biomedical engineering, or environmental monitoring and bioremediation [36]) will require approaches that allow a task to be broken down into separate sub-tasks, each of which may be addressed by a separately-optimised population of cells. From an historical perspective, this is entirely consistent with the development of computer software, which has moved from a *monolithic* model (in which an application is self-contained and isolated from other programs) to an object-oriented, *distributed* model, in which a system is viewed as a collection of *interacting entities* [37]. Apart from the benefits accrued from a division of labour, mixed populations

are also robust to environmental perturbations, which will be particularly significant in “real world” applications [11]. In order to achieve this distributed, engineered cellular computation, we require a reliable *communication* protocol that will allow cells and populations to exchange information, exert control and influence the overall system behaviour. As has already been demonstrated, several such protocols are possible, based on (for example) quorum sensing (QS) [13,14] or phage infection [16]. In this paper, we propose a third alternative communication scheme, based on bacterial *conjugation*.

A clear advantage of the proposed system is the *amount* of information that may be communicated. Quorum-sensing systems are based on the exchange of small signalling molecules, which are ideal for global synchronization of a system, where the controlling signal may be relatively simple. However, such systems lack the richness and complexity of cell-cell DNA-based methods (such as conjugation or phage), using which we may exchange relatively large “packets” of genetic material [16]. An added advantage of using conjugation as a wiring system is the requirement for *physical contact* between donor and recipient cells. Thus, if cells are constrained in their movement (e.g., because they are growing on a solid surface), conjugation allows fine-grained *spatial resolution* of computational processes (as opposed to the global “broadcast” model of QS). This will allow, for example, the precise targeting of the *location* of a specific input signal within a monolayer of communicating cells.

Another important benefit of engineered microbial consortia is *modularity* and *reusability* of components [10]. The ability to combine basic components together in order to build larger structures is a fundamental engineering principle, and facilitating the reuse of cellular systems is of paramount importance to practitioners in synthetic biology. We illustrate this with reference to our proposed XOR circuit. Our initial two-strain population implementing the NOR function is effectively non-reusable, as it cannot be “reset”. This problem is solved in the XOR population, where the system may compute indefinitely due to the reversible nature of the fimE/B system.

Computational simulations allow us to “test” proposed cellular systems in many different scenarios. In this paper, we provide not only deterministic/stochastic single-cell simulations (which provide us with an idealised perspective of the system, impossible to achieve *in vivo*), but also multicellular spatial simulations, which allow us to better understand the *population-level* dynamics of the system. Taken together, these computational studies offer a valuable insight into the proposed system, in order to prepare future wet-lab implementations of our design. Such a framework will allow for relatively easy investigation of implementation-specific issues, which may be expensive or time-consuming to test in the laboratory. For example, in future, we may use our *in silico* approach to explore a wider combination of site-specific recombination tools (only two were combined in this work), and how we may harness their different functionalities (such as insertions, cointegrations, deletions and inversions). Implementation of conjugation-based wiring requires the consideration of various practical issues. Most importantly, there are two types of conjugation channels; those that work with bacteria moving freely in a *liquid* medium (liquid maters, e.g., plasmids F and R64), and those that only work on the surface of a *solid* medium (surface maters, e.g., plasmids RP4, R388 and R46) [38]. The latter allow better control of the experiments, since conjugation only starts when the bacterial population is spread on a solid surface (e.g., an agar plate), and conjugation stops when the cells are taken out of the surface [39]. As a result, the experimenter could divide the computing process in steps, if required. Further control over the



**Figure 6. Spatial simulation of the XOR population using the four logic cases.** For each input case (0-0, 0-1, 1-0 and 1-1) a snapshot of the population after 32 hours is shown along a profile (bar chart) of the average number of plasmids of each kind (the four possibilities) in recipients (NOR\_2 and NOR\_3) (copy number = 10). The output, visually identified as green colour cells (from green fluorescent protein, GFP) corresponds to the XOR function. The average number of plasmids  $pmd_{F2}$  in a population is directly proportional to the fluorescence of it. The “y” axis in the bar graph is displayed in logarithmic scale. Short bright green lines in spatial figures of the population represent a conjugation process currently happening (those “springs” link donor and recipient).  
doi:10.1371/journal.pone.0065986.g006

experimental course may be achieved by selection for any of the computing strains, for instance by applying antibiotic selection. Periodic selection might be necessary since the various strains used in the computation might have different growth rates, particularly if they are loaded with plasmids (cells containing plasmids typically grow more slowly than plasmid-free ones). Therefore, even if the initial concentrations are 50% donors D and 50% recipients R, this ratio can shift rapidly. Growth rates can be adjusted by applying sub-inhibitory concentrations of antibiotics, and the effects of such changes may be checked using our simulations. As stated in the Methods Section, a conjugation experiment can be prolonged by mixing the cells of a mating and plating out again on fresh plates. Additional donors (or recipients) may be added at this stage to equilibrate the D/R ratio.

Another advantage of our system, which provides an obvious line of enquiry for future work, is the potential *scalability* of the system. There are many plasmid conjugation systems that work in an essentially orthogonal manner (meaning each plasmid uses its own DNA transport system, and the different systems do not interfere, even if acting on the same cell). This is easily understood, since each conjugative system promotes only the transfer of the plasmid it recognizes as its own [20]. In the *E. coli* bacterium there are at least 12 different conjugative systems, coded by compatible plasmids, that can act *together* in the *same* cell [40]. Many strains of *E. coli* isolated from clinical settings carry three or more plasmids, and these assemblies are stable and evolutionarily successful [41]. Moreover, a given conjugative plasmid can bring about the mobilization of several independent mobilizable plasmids [42]. Thus, the numbers of potential orthogonal systems quickly escalate.

We fully expect future work in synthetic biology to adhere closely to this model, in which computational simulations and laboratory investigations are inextricably linked in a tight cycle of feedback. Our proposed system offers one possible addition to the ever-growing “toolbox” available to biological engineers, and we

hope that experimental validation of its basic principles will be quickly forthcoming.

## Methods

### Modelling genetic logic

The model used for all simulations is supplied in this Section. We make the assumption that modelling the translation processes and the slow transcriptions (basal rates) make no qualitative difference. Thus, we adopt the following two simplifications in our system: 1) transcription and translation are joined into one single process called “transcription”, and 2) basal transcription rates are considered to be null in the system.

Ordinary differential equations (ODEs) from 1 to 13 explain the deterministic dynamics of the first approach (2-strain NOR population) using an *idealized* two-cell environment where physical contact is constant:

$$dG_1/dt = -k_1AG_1 + K_{-1}G_1^a - k_2BG_1 + k_{-2}G_1^b \quad (1)$$

$$dG_1^a/dt = k_1AG_1 - k_{-1}G_1^a - k_3BG_1^a + k_{-3}G_1^{ab} \quad (2)$$

$$dG_1^b/dt = k_2BG_1 - k_{-2}G_1^b - k_4AG_1^b + k_{-4}G_1^{ab} \quad (3)$$

$$dG_1^{ab}/dt = k_3BG_1^a - k_{-3}G_1^{ab} + k_4AG_1^b - k_{-4}G_1^{ab} \quad (4)$$

$$dCre/dt = k_5G_1^a + k_6G_1^b + k_7G_1^{ab} - k_{13}Cre - k_8Cre \quad (5)$$

$$dA/dt = k_{11} - k_1AG_1 - k_4AG_1^b - k_9A \quad (6)$$

$$dB/dt = k_{12} - k_2BG_1 - k_3BG_1^a - k_{10}B \quad (7)$$

$$dpmd_i/dt = -k_{13}Cre \quad pmd_i \quad (8)$$

$$dpmd/dt = k_{13}Cre \quad pmd_i \quad (9)$$

$$dG_2/dt = -k_{15}G_2F_2 + k_{-15}G_2^{F_2} \quad (10)$$

$$dG_2^{F_2} = k_{15}G_2F_2 - k_{-15}G_2^{F_2} \quad (11)$$

$$dF_2/dt = k_{14}pmd_i - k_{15}G_2F_2 + k_{-15}G_2^{F_2} - k_{17}F_2 \quad (12)$$

$$dGFP/dt = k_{16}G_2^{F_2} - k_{19}GFP \quad (13)$$

where  $G_1$  denotes the gene of the sender cell without inputs bound;  $G_1^a$ ,  $G_1^b$  and  $G_1^{ab}$  denotes the same gene with inputs A, B and both bound (respectively) to its promoters;  $Cre$  is the Cre protein;  $A$  and  $B$  are the inputs;  $pmd_i$  and  $pmd$  the plasmid with the segment that contains the gene in charge of expressing inducers or without it (respectively);  $G_2$  denotes the gene in the receiver cell without inducer bound;  $G_2^{F_2}$  is the same gene with inducers bound to its promoter;  $F_2$  are the inducers expressed by the *wiring* plasmid; and  $GFP$  are the green fluorescent proteins.

Biochemical reactions and kinetic rate values ( $k_i$ ) are detailed in Text S1. All rates are specified within standard ranges found in the literature [43–48]. Kinetic rate  $k_{13}$  has been fixed according to the experimental results of [23]. Importantly, similar parameters have the similar values, following the objective of defining as general a system as possible (free from parameter constraints that could compromise future validation). A perturbation analysis is performed (Figure S5) by adding Gaussian noise to every rate (being the mean the original value and the standard deviation 20% of the mean) and running the simulations several times. This analysis reinforces our conclusion about the qualitative robustness of our model. The quantitative behaviour may change without altering the overall goal of the computational process.

Stochastic simulations, that take into account the randomness of the chemical reactions, used the Gillespie algorithm [49] (using reactions described in Text S1). In this approach, time is discrete and the rates ( $k_i$ ) indicate the propensity of each reaction to happen.

The deterministic behaviour of cells NOR\_1 (component of the 3-strain XOR population) are described by equations 14–20:

$$dfimE/dt = k_5G_1^a + k_6G_1^b + k_7G_1^{ab} - k_8fimE - k_{25}fimE \quad (14)$$

$$pmd_{F1} - k_{26}fimE \quad pmd_{F2}$$

$$dX/dt = k_5G_1^a + k_6G_1^b + k_7G_1^{ab} - k_{22}XG_3 + k_{-22}G_3^X - k_{20}X \quad (15)$$

$$dG_3/dt = -k_{22}XG_3 + k_{-22}G_3^X \quad (16)$$

$$dG_3^X/dt = k_{22}XG_3 - k_{-22}G_3^X \quad (17)$$

$$dfimB/dt = k_{21}G_3 - k_8fimB - k_{23}fimB \quad (18)$$

$$pmd_{F1} - k_{24}fimB \quad pmd_{F2}$$

$$dpmd_{F1}/dt = -k_{23}fimB \quad pmd_{F1} + k_{24}fimB \quad (19)$$

$$pmd_{F2} - k_{25}fimE \quad pmd_{F1}$$

$$dpmd_{F2}/dt = k_{23}fimB \quad pmd_{F1} - k_{24}fimB \quad (20)$$

$$pmd_{F2} + k_{25}fimE \quad pmd_{F1}$$

where  $fimE$  and  $fimB$  denote the inversion proteins;  $X$  represents the repressor;  $G_3$  is the gene that expresses  $fimB$ ;  $G_3^X$  denotes the same gene when the repressor  $X$  is bound to its promoter; and  $pmd_{F1}$  and  $pmd_{F2}$  are the possible plasmid configurations inside NOR\_1 cells. Equations 1, 2, 3, 4, 6 and 7 are also used in this example as they define dynamics for existing components in NOR\_1 cells. Reactions and rates are, as before, detailed in Text S1. Importantly,  $k_{-22}$  is a low value in order to simulate a strong repression done by  $X$ .

Separately, NOR\_2 cells are simulated by ODEs 21 to 24. As stated in Section Results, the inside *program* of these cells are similar to the combination of both senders and receivers of the 2-strain NOR design (Figure S6 shows the transition to logic case 1-0 inside a NOR\_2 cell). The only difference relies on the four plasmid configurations (instead of two) that can be present in the cell. Thus, their equations are the same as in the NOR example but with some modifications:

$$dCre/dt = k_5G_1^a + k_6G_1^b + k_7G_1^{ab} - k_{13}Cre \quad (21)$$

$$pmd_{F2} - k_{27}pmd_{F2}^{F1} - k_8Cre$$

$$dF1/dt = k_{28}pmd^{F1} + k_{29}pmd_{F2}^{F1} - k_2F1G_1 - k_3F1G_1^a - k_{10}F1 \quad (22)$$

$$dpmd_{F2}^{F1}/dt = -k_{27}Cre \quad pmd_{F2}^{F1} \quad (23)$$

$$dpmd^{F1}/dt = k_{27}Cre \quad pmd_{F2}^{F1} \quad (24)$$

where  $F1$  is the expression product that induces the production of  $Cre$ ;  $pmd_{F2}^{F1}$  and  $pmd^{F1}$  are the plasmid with or without (respectively) gene  $F2$  when the promoter is pointing towards gene  $F1$ . The plasmids  $pmd_i$  and  $pmd$  (equations 8 and 9 also used in cells NOR\_2) make reference to plasmids  $pmd_{F2}$  and  $pmd$ . The gene  $G_1$  inside the NOR\_1 cell is not the same than the gene  $G_1$  inside the NOR\_2 cell, although they share the label. Stochastic simulations are done (via Gillespie algorithm) by using the biochemical reactions with the kinetic rates shown in Text S1.



The code of the previous model is included in Code S1 (all code in Python). This code is enough to reproduce Figures 2, 5, S1, S2 and S6. The spatial figures (Figures 3, 6, S3 and S4) make use of the package DiSCUS as explained in the following section.

### Population dynamics

A simulation tool for population dynamics is needed in order to get realistic spatial results (Figures 3, 6, S3 and S4) for the whole system behaviour. We present a simulation framework for bacterial growth, movement and horizontal gene transfer called DiSCUS (DiScrete Simulation of Conjugation Using Springs - <http://code.google.com/p/discus/>). Using an agent-based model (AbM) approach, our software combines the management of intercellular interactions with the simulation of intracellular genetic networks. AbMs are widely used to study microbial growth and biofilm formation [50–53] and conjugation has also been included in several simulations [31,54,55], but these (unrealistically) consider cells as abstract circular objects. Having rod-shaped cells (as in [56–59]) is essential to obtain accurate conjugation dynamics. Thus, DiSCUS is the first platform, to our knowledge, to offer the possibility of simulating conjugation dynamics in rod-shape bacteria.

ODEs are introduced *inside* every cell independently, so each bacterium runs its own copy of the circuit according to its design. While the circuit is simulated deterministically, the plasmid number is always discretized using several build-in functions. The discrete number corresponding to the plasmid concentration is used when: a) the cell divides, when a random selection of the plasmids will be part of each daughter (without losing plasmids); b) the plasmids have lower copy number than the limit, in which case they increase their copies; and c) the cell transfers (through conjugation) a small number of plasmids (1–3) to the recipient. The discretized values are always used to update deterministic concentrations. This process is essential to assure the numbers are kept consistent and realistic.

Conjugation events take place while the cells carry out their normal growing activity. DiSCUS handles conjugation processes through a probability distribution that can be easily tuned to correspond to the behaviour of different cell types. The *low* and *high* conjugation frequencies used in this paper are obtained via visual validation against real data taken from a recent study [60] (Figure S7A–C) where a difference is established between original donors (*low* conjugation frequency) and new donors or *transconjugants* (*high* conjugation probability). Biomechanical validation of DiSCUS (Figure S7D–I) against real data [59] has also been performed. In our design there is only one conjugation frequency, as transconjugant cells are unable to conjugate between them or back to donors (due to the fact that they do not carry a complete genetic transfer system). However, we run duplicate simulations considering both alternative frequencies (*low*: Figures 3 and S4; *high*: Figures 6 and S3) in order to explore the range of correct functioning of the system. We observe quantitative changes, but not significant alterations in the qualitative logic behaviour of the circuits. In order to allow conjugation to achieve complete infection of the recipient population during the experiment, the cells were mixed (shuffled) every 400 min so that new pairs donor-recipient can arise in the population (this phenomenon has been tested and validated experimentally in [39]; we provide simulated proof in Figure S8).

### Supporting Information

**Code S1 Python code for the intra-cell circuits.** Equations 1 to 13, that explain the deterministic behaviour of the first 2-

strain NOR population are coded in the file *NORdet.py* while the rates are simulated stochastically (Gillespie) in the file *NOTsto.py*. Regarding the XOR example, the NOR\_1 strain is simulated deterministically in *XOR(NOR1)det.py* (ODEs 14–20) and stochastically in *XOR(NOR1)sto.py*; The ODEs for the NOR\_2 strain are coded in *XOR(NOR2)det.py*.  
(ZIP)

**Figure S1 Molecular deterministic relations in the simulation of Figure 1B.** Using the same simulation of Figure 2 (case 1-1), these graphs show the behavioural changes in the relation between different molecules when the rate  $k_{13}$  (Cre deletion rate) is changed. All graphs show information at time = 200 min. **A.** Cre proteins and altered plasmids *pmd* (copy number = 1). **B.** Cre proteins and GFP molecules. **C.** F2 inducers and GFP molecules.  
(EPS)

**Figure S2 Comparison of stochastic and deterministic behaviour in plasmid concentration.** Twelve different simulations of Figure 1B (case 1-1) over 150 min. Only *pmd<sup>i</sup>* and *pmd* are monitored. Copy number equals 1 (six graphs on the left) and 20 (six graphs on the right). Rate  $k_{13}$  equals  $2E-5$  (top row),  $2E-6$  (middle row) and  $2E-3$  (bottom row). The unrealistic transitions of deterministic simulations (there is no such a thing as “0.243” plasmids in a cell) match better the stochastic transitions with bigger copy number. Also, big values of  $k_{13}$  will give better deterministic approximations (due to small delay in conversion).  
(EPS)

**Figure S3 Alternative spatial simulations to Figure 3 with greater conjugation frequency.** Red outline cells: donors (senders). Blue outline cells: recipients (receivers). Green colour intensity proportional to GFP concentration level (output). **A.** Logic 0-0 case. **B.** Logic 1-1 case. Bars in bar graphs proportional to plasmid numbers in the whole population (excluding sender cells).  
(EPS)

**Figure S4 Alternative spatial simulation for case 0-0 in Figure 6 with lower conjugation frequency.** Snapshot of an XOR population after 32 h when no inputs are present (right) and bar graph showing the average number of plasmids (per configuration) in receivers (left) at the same instant ( $y$  axis in logarithmic scale). By setting up the conjugation frequency to a low value, the number of plasmids *pmd<sub>F2</sub>* (which control the output) increases. As a result, some cells display a positive output (GFP) when they should not according to the XOR function. However, this positive output is residual.  
(EPS)

**Figure S5 Perturbation analysis of model reaction rates.** In each graph (A–D), the results of 50 simulations are shown, 45 of which use a perturbed set of reaction rates, and 5 of which use the original rates (control set, above dashed horizontal lines in graphs). In each experiment, we perturb every rate by adding Gaussian noise to the original value (standard deviation is 20%). **A,B:** Perturbation analysis of the model (Figure 2) for the 2-strain NOR population (23 rates), where the graphs show the effect of the output (GFP) when the inputs are 0-0 and 1-1 respectively. **C,D:** perturbation analysis for the XOR model (Figure 5), NOR\_1-strain (22 rates; inputs considered constraint in the profile), where the graphs measure the level of the *fimE* protein and the *pmd<sub>F1</sub>* plasmids, respectively.  
(EPS)

**Figure S6 Simulation results of NOR\_2 during its logic 1-0 case.** This case means that molecules  $A$  are present (logic “1” = 1000 molecules at the start;  $k_{11} = 500$  molecules  $\text{min}^{-1}$ ) and the plasmids coming from donors (NOR\_1) are all in configuration  $pmd_{F2}$ . In this scenario, we would expect a logic “0” in the output (no GFP). However, we can observe an initial GFP expression (bottom graph) which corresponds to the time delay needed to change the configuration of the plasmid from  $pmd_{F2}$  to  $pmd$  (top graph). After GFP is no longer expressed, its degradation controls the concentration. Copy number = 5. (EPS)

**Figure S7 Validation of DiSCUS.** Left (figures A–C): validation of conjugation dynamics using real data. **A.** Figure extracted from [60] where a colony of *Pseudomonas putida* is divided into dark red donor cells (DsRed), yellow recipient cells (YFP) and transconjugants, expressing both yellow and green light (YFP and GFP). The upper row shows the transconjugant signal, and the bottom row shows the whole community. **B and C.** Simulation results. Two simulations of similar colonies are recorded over exactly the same time intervals (min). The colours of the cells match the colours observed in **A.** Right (graphs D–I): validation of cell movement using real data. Graphs **D, F and H** are extracted from [59], and show experimental results of *Escherichia coli* growth regarding density, velocity gradient and ordering (respectively). Graphs **E, G and I** correspond to simulations in similar conditions to [59], for the same parameters (density, velocity gradient and ordering respectively). Test 1, 2 and 3 in graphs correspond to different spatial distribution of cells inside the microfluidic channel. (EPS)

**Figure S8 Effects of manual mixing on conjugation frequency.** **A.** Recipient-trapping behaviour of a population

## References

- Oldham P, Hall S, Burton G (2012) Synthetic biology: mapping the scientific landscape. *PLoS One* 7: e34368.
- Heinemann M, Panke S (2006) Synthetic biology: putting engineering into biology. *Bioinformatics* 22: 2790–9.
- Andrianantoandro E, Basu S, Karig DK, Weiss R (2006) Synthetic biology: new engineering rules for an emerging discipline. *Mol Syst Biol* 2: 2006.0028.
- De Lorenzo V, Danchin A (2008) Synthetic biology: discovering new worlds and new words. *EMBO reports* 9: 822–827.
- Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. *Nature* 403: 335–338.
- Gardner TS, Cantor CR, Collins JJ (2000) Construction of a genetic toggle switch in *Escherichia coli*. *Nature* 403: 339–342.
- Lou C, Liu X, Ni M, Huang Y, Huang Q, et al. (2010) Synthesizing a novel genetic sequential logic circuit: a push-on push-off switch. *Molecular systems biology* 6.
- Purcell O, Savery NJ, Grierson CS, di Bernardo M (2010) A comparative analysis of synthetic genetic oscillators. *J R Soc Interface* 7: 1503–24.
- Purcell O, di Bernardo M, Grierson CS, Savery NJ (2011) A multi-functional synthetic gene network: a frequency multiplier, oscillator and switch. *PLoS One* 6: e16140.
- Macia J, Posas F, Solé RV (2012) Distributed computation: the new wave of synthetic biology devices. *Trends Biotechnol* 30: 342–9.
- Brenner K, You L, Arnold FH (2008) Engineering microbial consortia: a new frontier in synthetic biology. *Trends Biotechnol* 26: 483–9.
- Basu S, Gerchman Y, Collins CH, Arnold FH, Weiss R (2005) A synthetic multicellular system for programmed pattern formation. *Nature* 434: 1130–4.
- Tamsir A, Tabor JJ, Voigt CA (2011) Robust multicellular computing using genetically encoded NOR gates and chemical ‘wires’. *Nature* 469: 212–5.
- Regot S, Macia J, Conde N, Furukawa K, Kjellen J, et al. (2011) Distributed biological computation with multicellular engineered networks. *Nature* 469: 207–211.
- Goñi-Moreno A, Redondo-Nieto M, Arroyo F, Castellanos J (2011) Biocircuit design through engineering bacterial logic gates. *Natural Computing* 10: 119–127.
- Ortiz M, Endy D (2012) Engineered cell-cell communication via dna messaging. *Journal of Biological Engineering* 6: 16.
- Atkinson S, Williams P (2009) Quorum sensing and social networking in the microbial world. *J R Soc Interface* 6: 959–78.
- Danino T, Mondragón-Palmino O, Tsimring L, Hasty J (2011) A synchronized quorum of genetic clocks. *Nature* 463: 326–330.
- Tatum EL, Lederberg J (1947) Gene recombination in the bacterium *Escherichia coli*. *Journal of Bacteriology* 53: 673–684.
- De La Cruz F, Frost LS, Meyer RJ, Zechner EL (2010) Conjugative dna metabolism in gram-negative bacteria. *FEMS Microbiology Reviews* 34: 18–40.
- Llosa M, Gomis-Roth FX, Coll M, Cruz Fdl (2002) Bacterial conjugation: a two-step mechanism for dna transport. *Molecular Microbiology* 45: 1–8.
- Grindley ND, Whiteson KL, Rice PA (2006) Mechanisms of site-specific recombination\*. *Annual Review of Biochemistry* 75: 567–605.
- Yamanishi M, Matsuyama T (2012) A modified cre-lox genetic switch to dynamically control metabolic flow in *Saccharomyces cerevisiae*. *ACS Synthetic Biology* 1: 172–180.
- Moon TS, Clarke EJ, Groban ES, Tamsir A, Clark RM, et al. (2011) Construction of a genetic multiplexer to toggle between chemosensory pathways in *Escherichia coli*. *Journal of Molecular Biology* 406: 215–227.
- Ham TS, Lee SK, Keasling JD, Arkin AP (2008) Design and construction of a double inversion recombination switch for heritable sequential genetic memory. *PLoS ONE* 3: e2815.
- Ham TS, Lee SK, Keasling JD, Arkin AP (2006) A tightly regulated inducible expression system utilizing the fim inversion recombination switch. *Biotechnology and Bioengineering* 94: 1–4.
- Haynes K, Broderick M, Brown A, Butner T, Dickson J, et al. (2008) Engineering bacteria to solve the burnt pancake problem. *Journal of Biological Engineering* 2: 8.
- Baumgardner J, Acker K, Adefuye O, Crowley S, DeLoache W, et al. (2009) Solving a Hamiltonian path problem with a bacterial computer. *Journal of Biological Engineering* 3: 11.
- Siuti P, Yazbeck J, Lu TK (2013) Synthetic circuits integrating logic and memory in living cells. *Nature Biotechnology*.
- Garcillán-Barcia MP, de la Cruz F (2008) Why is entry exclusion an essential feature of conjugative plasmids? *Plasmid* 60: 1–18.
- Gregory R, Saunders JR, Saunders VA (2008) Rule-based modelling of conjugative plasmid transfer and incompatibility. *Biosystems* 91: 201–15.
- Novick RP (1987) Plasmid incompatibility. *Microbiological Reviews* 51: 381–395.
- Goñi-Moreno A, Amos M (2012) A reconfigurable nand/nor genetic logic gate. *BMC Systems Biology* 6: 126.

34. Gally DL, Bogan JA, Eisenstein BI, Blomfield IC (1993) Environmental regulation of the fim switch controlling type 1 fimbrial phase variation in *Escherichia coli* K-12: effects of temperature and media. *Journal of Bacteriology* 175: 6186–6193.
35. McClain MS, Blomfield IC, Eisenstein BI (1991) Roles of fimb and fime in site-specific DNA inversion associated with phase variation of type 1 fimbriae in *Escherichia coli*. *Journal of Bacteriology* 173: 5308–5314.
36. Khalil AS, Collins JJ (2010) Synthetic biology: applications come of age. *Nature Reviews Genetics* 11: 367–379.
37. Meyer B (1988) *Object-Oriented Software Construction*, volume 2. Prentice Hall, New York.
38. Smillie C, Garcilln-Barcia MP, Francia MV, Rocha EPC, de la Cruz F (2010) Mobility of plasmids. *Microbiology and Molecular Biology Reviews* 74: 434–452.
39. del Campo I, Ruiz R, Cuevas A, Revilla C, Vielva L, et al. (2012) Determination of conjugation rates on solid surfaces. *Plasmid* 67: 174–182.
40. Garcilln-Barcia MP, Francia MV, De La Cruz F (2009) The diversity of conjugative relaxases and its application in plasmid classification. *FEMS Microbiology Reviews* 33: 657–687.
41. Grad YH, Godfrey P, Cerquiera GC, Mariani-Kurkdjian P, Gouali M, et al. (2013) Comparative genomics of recent shiga toxin-producing *Escherichia coli* O104:H4: Short-term evolution of an emerging pathogen. *mBio* 4.
42. Cabezn E, Sastre J, de la Cruz F (1997) Genetic evidence of a coupling role for the traG protein family in bacterial conjugation. *Mol Gen Genet* 254: 400–6.
43. Dublanche Y, Michalodimitrakis K, Kummerer N, Foglierini M, Serrano L (2006) Noise in transcription negative feedback loops: simulation and experimental analysis. *Molecular systems biology* 2.
44. Goñi-Moreno A, Amos M (2012) Continuous computation in engineered gene circuits. *Biosystems* 109: 52–56.
45. Balagaddé FK, Song H, Ozaki J, Collins CH, Barnet M, et al. (2008) A synthetic *Escherichia coli* predator-prey ecosystem. *Mol Syst Biol* 4: 187.
46. Andersen JB, Sternberg C, Poulsen LK, Bjorn SP, Givskov M, et al. (1998) New unstable variants of green fluorescent protein for studies of transient gene expression in bacteria. *Appl Environ Microbiol* 6: 2240–2246.
47. Bolouri H, Davidson EH (2003) Transcriptional regulatory cascades in development: Initial rates, not steady state, determine network kinetics. *Proceedings of the National Academy of Sciences* 100: 9371–9376.
48. de Leon SBT, Davidson EH (2009) Modeling the dynamics of transcriptional gene regulatory networks for animal development. *Developmental Biology* 325: 317–328.
49. Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry* 81: 2340–2361.
50. Goroehowski TE, Matyjaszkiewicz A, Todd T, Oak N, Kowalska K, et al. (2012) Bsim: an agent-based tool for modeling bacterial populations in systems and synthetic biology. *PLoS One* 7: e42790.
51. Kreft JU, Booth G, Wimpenny JWT (1998) Bacsim, a simulator for individual-based modelling of bacterial colony growth. *Microbiology* 144: 3275–3287.
52. Xavier JB, Picioreanu C, van Loosdrecht MCM (2005) A framework for multidimensional modeling of activity and structure of multispecies biofilms. *Environ Microbiol* 7: 1085–103.
53. Lardon LA, Merkey BV, Martins S, Dotsch A, Picioreanu C, et al. (2011) Dynamics: next-generation individual-based modelling of biofilms. *Environ Microbiol* 13: 2416–34.
54. Krone SM, Lu R, Fox R, Suzuki H, Top EM (2007) Modelling the spatial dynamics of plasmid transfer and persistence. *Microbiology* 153: 2803–16.
55. Merkey BV, Lardon LA, Seoane JM, Kreft JU, Smets BF (2011) Growth dependence of conjugation explains limited plasmid invasion in biofilms: an individual-based modelling study. *Environ Microbiol* 13: 2435–52.
56. Melke P, Sahlin P, Levchenko A, Jönsson H (2010) A cell-based model for quorum sensing in heterogeneous bacterial colonies. *PLoS Comput Biol* 6: e1000819.
57. Rudge TJ, Steiner PJ, Phillips A, Haseloff J (2012) Computational modeling of synthetic microbial biofilms. *ACS Synthetic Biology* 1: 345–352.
58. Cho H, Jönsson H, Campbell K, Melke P, Williams JW, et al. (2007) Self-organization in high-density bacterial colonies: efficient crowd control. *PLoS Biol* 5: e302.
59. Volfson D, Cookson S, Hasty J, Tsimring LS (2008) Biomechanical ordering of dense cell populations. *Proceedings of the National Academy of Sciences* 105: 15346–15351.
60. Seoane J, Yankelevich T, Dechesne A, Merkey B, Sternberg C, et al. (2011) An individual-based approach to explain plasmid invasion in bacterial populations. *FEMS Microbiology Ecology* 75: 17–27.