# Bacterial computing with engineered populations

Martyn Amos[1], Ilka Maria Axmann[2], Nils Blüthgen[3], Fernando de la Cruz[4], Alfonso Jaramillo[5], Alfonso Rodriguez-Paton[6] and Friedrich Simmel[7]

[1]*School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, UK.*
[2]*Institut für Synthetische Mikrobiologie, Heinrich-Heine-Universität Düsseldorf, Germany.*
[3]*Institut für Pathologie, Charite - Universitätsmedizin Berlin, Germany.*
[4]*Intergenomics Group, Instituto de Biomedicina y Biotecnología de Cantabria, Universidad de Cantabria, Spain.*
[5]*School of Life Sciences, University of Warwick, UL.*
[6]*Laboratorio de Inteligencia Artificial, Universidad Politécnica de Madrid, Spain.*
[6]*Systems Biophysics and Bionanotechnology, Technische Universität München, Germany.*

We describe strategies for the construction of bacterial computing platforms, by describing a number of results from the recently completed BACTOCOM project. In general, the implementation of such systems requires a framework containing various components such as intra-cellular circuits, single cell input/output and cell-cell interfacing, as well as extensive analysis. In this overview paper we describe our approach to each of these, and suggest possible areas for future research.

*Keywords: Synthetic biology, unconventional computing, conjugation, simulation, mathematical modelling.*

## 1. Introduction

Various *natural computing* paradigms exist that are inspired by biological processes (e.g., artificial neural networks, immune systems, genetic algorithms, ant colony algorithms (see [13] for a review)). However, we can now go further than mere inspiration: instead of developing computing systems that are loosely *modelled* on natural phenomena, we can now directly *use* biological substrates and processes to encode, store and manipulate information.

Since the work of Adleman [1] and others, the feasibility of using biological substrates for computing has been well-established (see [2] for a review). More recent work on synthetic biology [4, 7, 14, 36] has shown that the living cell may now be considered as a programmable computational device [5, 16, 19], capable of sophisticated, human-controlled individual and collective behaviour [6, 10, 51]. However, such biological engineering is inherently difficult, due to the nature of the biological substrate. Attempts at rational design are often thwarted by factors such as crosstalk, cell death, mutation, noise and other external conditions [39],

*Proc. R. Soc. A* 1–18; doi: 10.1098/rspa.00000000
Monday 15th June, 2015

and bio-engineers sometimes adopt what has been called a "design then mutate" approach [25].

The BACTOCOM (Bacterial Computing with Engineered Populations) project[1] outlined a framework for engineering biological computation, by *harnessing* the inherent stochasticity of the underlying biological system. Here, we discuss the proposed system, describe its implementation, and draw lessons from our experiences.

The rest of the paper is organized as follows: in Section 2 we give an overview of the BACTOCOM conceptual framework. In the Sections that follow, we describe a selection of system components (specifically, internal cell logic, and cell-cell interfacing), as well as over-arching issues such as analysis. We conclude with a discussion of how this work might be developed in the future.

## 2. BACTOCOM framework

The overall aim of BACTOCOM was to develop a relatively self-contained, semi-autonomous biological computing platform, which should be capable of constructing simple logic circuits in *E. coli* bacteria. Early related work in this area [24, 56] used a combinatorial approach, in which genetic circuit components were "shuffled" or "evolved" in order to achieve the desired behaviour. Although this approach is capable of generating functioning circuits, it is limited by the need for repeated intervention, in the form of the evaluation through screening of intermediate circuits. The ambitious goal of BACTOCOM was to limit the need for such human intervention, by embedding *within the system itself* a method for evaluating the "quality" of assembled circuits. We now briefly describe the BACTOCOM scheme.

In Figure 1 we give a schematic overviewof proposed system functionality. Fundamentally, we sought to allow a population of bacteria to "evolve" (in the sense of performing an adaptive search on a landscape) by facilitating the construction and sharing of functional genetic circuit components. This process both informed and was *informed by* computational modelling and simulation of system components, in a tight feedback loop.

Implementation begins with a population of *E. coli* bacteria, which forms the core of our "computer". We engineer a set of computational plasmids – circular strands of DNA representing "components" – which may be combined together within the cell to form a simple logical circuit [50]. These components may be exchanged between individual bacteria via the process of *conjugation*; the transfer of genetic material via direct cell-cell contact [33, 54]. By introducing large numbers of these computational plasmids, we initialise the system. Over time, the bacteria integrate the components into their genomes, thus "building" logical circuits. The "output" of these circuits is measurable, and by defining "success" in terms of correlation with a desired signal profile, we allow successful components to flourish via a process of selection. Our intention was that this selection would not be performed manually (as before), but by an in-built *comparator device*,

[1] Three-year project, supported by the European Commission, FP7 FET Proactive, Biochemistry-based Information Technology (CHEM-IT) initiative, project number 248919. Website at http://www.bactocom.eu

which increases a cell's output of "good" combinations of computational plasmids (which may then be taken up by neighbouring cells). By controlling the desired signal profile, we *direct* the random search of the population towards novel, robust computational structures.

## 3. Cells implementing logic

In previous work, Leibler's group proposed a combinatorial procedure to screen genetic networks showing digital behaviour under selected inducers [24]. For this, they generated a library of five promoters containing different operator sites, and joined them to the coding parts of three repressors. In this way, they generated 125 different circuits, some of them implementing logic gates such as NAND, NOR or NOT. Interestingly, they found that the same circuit topology, but with different proteins (i.e., kinetic parameters) would sometimes yield two different logic gates.

The construction of synthetic gene networks in our system relies on the assembly of characterized regulatory elements, using rational design principles. It is important, therefore, to analyze the scalability and limits of such a design workflow. To study the design capabilities of our proposed libraries of regulatory elements, we earlier developed *Genetdes* [41], the first automated design approach that combines such elements to search the genotype space associated with a given phenotypic behavior (this corresponds to the "Computational plasmids" component of Figure 1). We used an updated version of this software to calculate the designability of dynamical functions obtained from circuits assembled with a given genetic library [42]. We dissected the functional diversity of a constrained library, and found that even such relatively simple libraries can provide a rich variety of potential behaviours.

As we have established, given a library of regulatory elements, it is possible to construct many circuits with various dynamical behaviours. However, some behaviours occur more often than others. To quantitatively analyze this, we computed the designability of a set of useful behaviours [42]. We constructed a more reduced library of regulatory elements, in order to assemble all possible circuits up to three genes and process their dynamics. Remarkably, the library involved promoters that had been previously characterized and even used for engineering various synthetic circuits in the bacterium *E. coli.* Interestingly, we found that a limited registry could encode a large number of behaviours. After this initial work was performed, it was further developed into the *AutoBioCAD* system for the automated design of gene regulatory circuits [44][2].

The most significant challenge of the project involved the construction of a comparator device, which would allow us to fully "automate" the search for useful circuits (as opposed to performing it *in silico*), by comparing the output of circuits assembled *in vivo* with some external reference signal (for example, the presence or absence of a heavy metal pollutant). This would then allow for the repression (or otherwise) of the replication of the computational plasmids, and required the design and construction of an exclusive-OR (XOR) gate in a single cell. With hindsight, this was an over-ambitious goal at the time, and single-cell XOR implementations have been described only relatively recently [5, 48].

[2] Available for download at http://jaramillolab.issb.genopole.fr/display/sbsite/Download

Bonnet, *et al.*, citing [5], describe the inherent difficulty of such a task, due to the problems of “reusing regulatory molecules within the self-mixing environments of individual cells ... a single-cell two-input XOR gate ... required controlled expression of four gate-specific regulatory molecules from four plasmids” [8]. Put simply, our proposed implementation would have required the development, testing and integration of too many new “parts" with the time and resources available to the project, and this only became apparent once we were already underway.

In summary, the construction of this device proved to be extremely challenging, and we decided to broaden out the scope of the project to investigate (a) alternative methodologies for circuit construction, (b) new analytical methods, and (c) new forms of cell-cell communication (including consideration of multi-cellular computation, which would address at least some of the issues inherent to single-cell computation). We now describe aspects of each of these new areas.

### (*a*) *Alternative circuit construction methods*

Alongside work on gene networks, we investigated the use of RNA for the purposes of constructing intra-cellular circuits. A grand challenge in synthetic biology is to use our current knowledge of RNA science to perform the automatic engineering of completely synthetic sequences encoding functional RNAs in living cells. In [44], we report on a fully automated design methodology and experimental validation of synthetic RNA interaction circuits working in a cellular environment. The computational algorithm, based on a physicochemical model, produces novel RNA sequences by exploring the space of possible sequences compatible with predefined structures. We tested our methodology in *E. coli* by designing several positive riboregulators with diverse structures and interaction models, suggesting that only the energy of formation and the activation energy (free energy barrier to overcome for initiating the hybridization reaction) are sufficient criteria to engineer RNA interaction and regulation in bacteria. The designed sequences exhibit non-significant similarity to any known noncoding RNA sequence. Our riboregulatory devices work independently, and in combination with transcription regulation, to create complex logic circuits.

Developing this work further, we also investigated small RNAs (sRNAs), which can operate as regulatory agents to control protein expression [45]. We developed a physicochemical framework, relying on base pair interaction energies, to design multi-state sRNA devices by solving an optimization problem with an objective function accounting for the stability of the transition and final intermolecular states. Contrary to the analysis of the reaction kinetics of an ensemble of sRNAs, we solved the inverse problem of finding sequences satisfying targeted reactions. We showed that our objective function correlates well with measured riboregulatory activity of a set of mutants. This has enabled the application of the methodology for an extended design of RNA devices with specified behaviour, assuming different molecular interaction models based on Watson-Crick interaction. We designed several YES, NOT, AND, and OR logic gates, including the design of combinatorial riboregulators. For an overview of work on the automatic design of synthetic biology regulatory systems, see [43].

### (*b*) *New analytical methods*

In parallel with this work, we also investigated, using mathematical modeling, the *quantitative* aspects of gene regulation via sRNAs that are essential for engineering synthetic circuits [47] (see also [32]). We analysed the temporal characteristics of stress responses, not only in theory, but also by quantitative experimental analyses, and found that sRNAs can be responsible for a pronounced delay in the accumulation of their target gene transcript, which ensures a rapid decline in mRNA levels once external stress triggers are removed. These kinetic properties allow the system to selectively respond to sustained stimuli and thus establish a temporal threshold. Biological information is frequently encoded in the quantitative aspects of intracellular signals, a mechanism that becomes exquisitely relevant for reprogramming cellular networks. Starting from a minimal model of one sRNA interacting negatively with one mRNA, more complex scenarios can be implemented to exploit the regulatory potential of synthetic sRNA networks. Coordinated regulation of functionally related proteins optimizes cellular responses, and is typically thought to be established by a shared regulator simultaneously controlling the expression of multiple genes. We explored the possibility for post-transcriptional co-regulation of multiple mRNAs by a common sRNA which inhibits translation and/or triggers mRNA degradation [47]. By deriving an analytical approximation for the steady state, we showed that sRNAs establish gene expression thresholds that lie in the same concentration range for all target mRNAs, even if their sRNA affinities differ significantly.

We also studied several different types of *oscillator*, as these form well-established functional units that have been used extensively in synthetic biology [12, 16, 20, 38]. We performed several mathematical and experimental studies of the microbial circadian clock [11, 26], established that reliable single gene oscillation is possible without a requirement for negative feedback [35], described a synthetic transcriptional clock that works *in vitro* [18], and studied experimentally the behaviour of a synthetic oscillator under temporal perturbations [45]. All of these studies contribute to our understanding of fundamental underlying processes, serve as models for how to design more holistic networks of gene regulation and growth dynamics, and further develop principles of effective “insulation” between biochemical subsystems, which will be critical for the synthesis of larger and more complex systems.

## 4. Cells exchange components

Cells are (and can be used as) programmable computational devices. Basic programming instructions can be delivered to cells by means of (for example) small diffusible molecules (inducers or co-repressors), in order to switch on or off single inputs. However, delivering more complex instructions becomes progressively harder. Intracellular transmission – when the receptors of the chemical signals are transcription factors or regulatory RNAs – also suffers from a limited number of independent signals that can be used for transmission. Moreover, circuit generation by the use of coupled I/O devices suffer usually from the problem of *impedance matching* [56]; that is, the differences between the range of concentrations of a modulator produced by the output of a device and the required input concentration range of the next. Chemical signaling (for example, by quorum-sensing (QS) messengers) is limited by the number of orthogonal (mutually non-interfering) signals. All these problems result in severe difficulties when trying to build complex circuits within a *single* cell. Thus, scientists and engineers are progressively turning to engineering *microbial consortia* in order to be able to build more complex circuits [9, 10, 40, 51].

There are two main types of channel for cell-cell communication, which we can call *non-programmable* (small molecules or peptides that can be sensed in just one way), or *programmable* (if they contain DNA and, therefore, incorporate a new "program" when they enter a cell). As we have just discussed, chemical signals have been used (e.g., quorum sensing signals, pheromones), but these have relatively low informational content, lack of diversity, and are inherently *global* (relying, as they do, on diffusion). They will not be discussed any further here. Programmable channels are doubtless superior, in that the input signal can be more information-rich *and* specific, and thus capable of triggering different or even opposite effects in target cells [37]. Programmable channels are specific types of nano-machines that bacterial cells use naturally for cell-cell communication. The most important processes that bacteria use for cell-cell communication are called *transformation*, *transduction* and *conjugation*, which correspond to three types of channel (three distinct physical layers): naked DNA, bacteriophages and plasmids. Here, we focus on the last of these.

The genetic information in bacterial cells is usually encoded in a single chromosome, that is, a circular DNA molecule of roughly 5 million base-pairs (b.p.), enough to contain five thousand protein-coding genes. Besides the main chromosome, most bacteria contain additional autonomous, small, dispensable DNA molecules called plasmids [49]. Plasmid sizes range from about 1,500 to 250,000 b.p. (1.5-250 Kb). Plasmids usually carry genetic information for adaptive genes, such as antibiotic resistance, virulence, utilization of rare food sources, degradation of xenobiotics, etc. Besides that, of course, they carry the genetic information necessary for their own survival. This includes genes for DNA replication, stability, propagation and establishment in new bacterial cells [17].

Plasmid conjugation is therefore a communication protocol between bacteria. It is naturally used as a powerful mechanism by which bacteria acquire new, preformed genetic information that can be used as an adaptation tool for facing changing environmental conditions. For instance, bacteria rapidly become resistant to antibiotics by acquiring plasmids with pre-formed genes or operons that result in antibiotic resistance. Being a natural process, conjugation has been continually optimized by evolution (for thousands of millions of years). Evolution resulted in sophisticated conjugation systems, displaying several properties that are of significant interest for the engineering of biological circuits.

We require precise control over the conditional replication of plasmids inside cells, the idea being that cells containing "good" circuits would produce more copies of their plasmids (which would then be taken up by other cells). We constructed several plasmids that were able to coexist together in the same cell. Importantly, their replication could be repressed by TetR (a repressor protein), which allowed for very fine control over their numbers, and they could be independently selected. We also developed an innovative new method for estimating conjugation rates that is faster and less noisy than the classical method

[15]. Following this work, we focussed on exploring further the capabilities of conjugation *per se* as a useful communication scheme for more general-purpose, *multi-cellular* computing.

## 5. Multi-cellular computing

As the field of synthetic biology matures, it is increasingly moving away from "single-cell" solutions, and towards the use of collections (or "consortia") of engineered microbes [3]. These multi-cellular solutions allow us to transcend the fundamental limitations of single cells, and facilitate a "division of labour" across mixed populations. One of the key challenges, therefore, is the engineering of communication between cells. In this Section, we describe our computational and experimental contributions to this area.

### (*a*) *Simulation platforms*

Programming intercellular communication between bacteria using conjugative plasmids (as described in the previous Section) was a central aspect of the BACTOCOM project. A central question of interest concerned the rate of propagation of specific conjugative plasmid in a microbial population. Such questions are often approached via mathematical modelling or computational simulation, but the only simulator available to predict plasmid transfer dynamics in bacterial populations on solid surfaces [30] was not considered to be adequate for our purposes. We therefore developed the computational tools capable of simulating these conjugation-based multicellular systems. In order to minimise risk, and to ensure that our results were not influenced by a particular simulation paradigm, we developed platforms that used both *continuous* space (DiSCUS: Discrete Simulation of Conjugation Using Springs) and *discrete* space (BactoSIM I and II). Both platforms used an individual-based modelling (IBM) approach.

The earliest platform for individual-based modelling of microbial populations was BACSIM [28]. This was further developed by members of Kreft's group [29] and by researchers in TU Delft [55]. The simulator's most recent version, iDynoMICS [31], specialises in biofilm growth and development. Relatively recent IBM approaches to microbes allow for the simulation of rod-shaped bacteria [27, 46]. With respect to the specific simulation of conjugative plasmid propagation, there exist relatively few studies [30, 34]. The first paper, in particular, modelled conjugation using an interacting particle system. This system ignores the physics of microcolony growth, and uses a grid with simple cellular automata like rules that prevent growth once a cell has no free space. At a macro level the model fits biological data and provides a predictive framework to understand plasmid-bacteria interactions. However, we sought a more realistic handling of physical properties of the system, which led us to our first package.

#### (a.1) *DiSCUS*

DiSCUS (Discrete Simulation of Conjugation Using Springs)[3] is a simulation framework designed to represent bacterial growth, movement and horizontal gene

[3] Available from http://code.google.com/p/discus/)

transfer [22, 23]. This simulator is implemented using an agent-based model (ABM) approach. This allows the software to handle each cell individually. The simulator combines the management of intercellular interactions with the operation of intracellular genetic networks. The goal is to better understand and predict the behaviour of bacterial colonies in the biology lab. In this Section we describe how our simulation platform has been used to study single-cell and multicellular behaviour in the context of engineered bacterial logic gates.

DiSCUS represents bacterial cells as rod-shaped objects and implements conjugation. Physical forces between cells (occurring when they are joined during conjugation) are modelled as springs. Ordinary differential equations (ODEs) are introduced *inside* every cell independently, so that each bacterium runs its own copy of a genetic circuit. Conjugation events occur during the normal growing activity of the cells. The frequency of these events is governed by probability distributions that are tuneable for each cell type. These frequencies can be validated with experimental results and tuned by the user. Another feature that was included in the simulations is the possibility of reshuffling cells in order to obtain new donor-recipient pairs.

DiSCUS is written in Python and uses the physics engine Pymunk to model the physical interactions. DiSCUS is the first platform, to our knowledge, to offer the possibility of simulating conjugation dynamics in rod-shape bacteria. So far, it has been used to study a population-based, reconfigurable logic gates (without conjugation, Figure 2) [22] and a multicellular comparator (using conjugation, Figure 4) [23]. We now briefly discuss the result in Figure 2; this work concerns the possibility of engineering *multi-functional* genetic devices, which may switch between *modes* of operation depending on some external signal (although this example does not illustrate the full potential of the system, as it does not consider conjugation). In [22] we describe a device which may be switched between NAND/NOR behaviour, depending on the concentration of input signals (that is, the underlying circuitry remains constant, but its behaviour switches depending on the concentration range of the inputs it receives). This builds on our previous work on *continuous computation* in genetic devices [21], and allows us to address the common lack of *constant* input signals in such systems (that is, input signals may only "exist" for a brief period, as they are often based on "bursts" of molecular signals, which rapidly decay). We propose a single circuit, which is engineered into a host bacterium, and which comprises a combination of the NOR gate described in [51] and the toggle switch from [19] (Figure 3). The circuit takes in two different molecules as inputs, the presence of either corresponding to a 1. These are then fed into the NOR circuit, which (depending on the value of the inputs) induces a change in the state of the toggle switch. The entire circuit produces a "high" (1) signal *by default*, in the absence of any input signals, and is kept high by the state of the toggle switch. As soon as *either* input to the NOR gate equals 1, the output signal is "pulled low" to 0; once both high-valued inputs are removed, the circuit defaults back to the "high" (1) state, and the toggle switch is flipped again. In this way, only a brief "spike" in the output of our inverter is sufficient to flip the toggle switch - there is no need for constant input.

We simulate a population of cells containing this engineered circuit, using an agent-based approach that considers a mono-layer of cells. The simulation considers physical factors such as cell-cell pressure, collisions, cell movement, walls and so on. The output of each cell is signalled by the production of green fluorescent protein (GFP), which causes cells to "glow" green (light corresponding to an output value of 1). The simulated surface is divided into quadrants (Figure 2, lower), and each quadrant contains a different combination of input signal molecules. As we first consider the NOR case, it is clear that we should see an output of 1 only in the quadrant where both inputs are equal to 0; the simulated population is innoculated in this quadrant, and gradually grows (through cell division) to occupy the entire surface (time proceeds on the x-axis). We see that the cells occupying quadrants where either (or both) inputs are equal to 1 do not give off light (equivalent to an output value of 0); the few cells in the top-right corner that are doing so have actually been pushed into that quadrant by other cells.

Importantly, we show how a simple change in the definition of a "high" input value (1) can switch the circuit's functionality from NOR to NAND. The key to this is the fact that the NOR circuit uses a particular molecular concentration to represent 1; by significantly reducing the input concentration, the circuit "flips" (without modification into NAND mode (the specifics of how this works are beyond the scope of this review paper, but full details are given in [22]). In Figure 2 (top), we see the results of another simulation, this time for the NAND case, in which we can clearly see that the only quadrant in which the cells are *not* "on" is (correctly) the one in which both inputs are equal to 1.

Figure 4 (from [23]) shows a similar multi-cellular simulation of a distributed XOR gate (comparator). This implementation *does* use conjugation, which facilitates direct cell-cell transfer of genetic information. We distribute the comparator over three connected NOR gates, which are engineered into separate bacterial strains (one of which contains the GFP reporter gene). As shown in Figure 4, we observe light only in the cases where the inputs have different values, which is the correct behaviour.

### (a.2) *BactoSIM*

We now consider the BactoSIM I and II simulators, which operate in discrete space. These two simulators are spatially explicit, individual-based models of bacterial conjugation using a discrete representation of time and space. Both models share many features. Space is defined as a discrete grid where agents are placed and evolve through their local interactions. Each agent is described as a state vector which is updated according to the model rules that take into account the local agent state and the states of neighbour agents located at adjacent cells. The rules for updating the states of each individual represent metabolic processes such as nutrient uptake, growth, reproduction and conjugation of each bacterial cell in the colony. It also takes into account non-metabolic processes such as nutrient diffusion. Conjugation rules and the final purpose of both simulators are the main differences between both models. Both simulators are written in Java. BactoSIM I is being developed using the REPAST open source framework for asynchronous execution of the simulations[4].

[4] The code for BactoSIM I can be accessed from https://code.google.com/p/haldane/ and BactoSIM II from the LIA web page at http://www.lia.upm.es/index.php/simulators

### (*b*) *Droplet computation*

In addition to these computational studies, we undertook experimental work on the encapsulation of *E. coli* bacteria within microemulsion droplets. Specifically, we engineered a system based on *quorum sensing* (QS) to implement a spatially extended communication system [53]. QS has been widely used in synthetic biology applications [6, 51, 52], and our experiments augment this work by showing how the slow diffusion of chemical signals from one "compartment" to another might be used to implement directed pattern formation and/or distributed cellular computing at scales much smaller than those possible in purely acqueuous media (thus making such constructions both "faster" and "smaller"). In addition, our approach demonstrates the potential for partitioning sub-populations of bacteria within the same media, allowing for engineered consortia containing cells with very different environmental requirements.

## 6. Summary and Conclusions

In this review article we have presented a (necessarily incomplete) snapshot of key results from the BACTOCOM project. We performed foundational work on artificial oscillators and RNA-based gene regulation, shed extra light on the specific dynamics of bacterial conjugation, provided core infrastructure for the future simulation of multi-cellular computing systems, and performed important early work on droplet computation. In terms of sustainability, results from BACTOCOM form the basis of three new FP7 projects; Plaswires[5] (which investigates the use of plasmids as "wires" connecting computational cells), Evoprog[6] (which uses phage and bacteria, combined with microfluidics, to implement *in vivo* directed evolution) and RiboNets[7] (which looks at RNA-based devices).

### *Acknowledgments*

We thank the European Commission for supporting the BACTOCOM project under FET Proactive 3: Bio-chemistry-based Information Technology (CHEM-IT) (ICT-2009.8.3), project reference 248919, and the editors for inviting us to submit this article.

## References

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

[2] Martyn Amos. *Theoretical and Experimental DNA Computation.* Springer, 2005.

[5] http://www.plaswires.eu/

[6] http://evoprog.eu/

[7] http://www.ribonets.eu/

[3] Martyn Amos. Population-based microbial computing: a third wave of synthetic biology? *International Journal of General Systems*, 43(7):770–782, 2014.

[4] Ernesto Andrianantoandro, Subhayu Basu, David K. Karig, and Ron Weiss. Synthetic biology: new engineering rules for an emerging discipline. *Molecular Systems Biology*, 2:2006.0028, 2006.

[5] Simon Ausländer, David Ausländer, Marius Müller, Markus Wieland, and Martin Fussenegger. Programmable single-cell mammalian biocomputers. *Nature*, 487(7405):123–7, 7 2012.

[6] Subhayu Basu, Yoram Gerchman, Cynthia H. Collins, Frances H. Arnold, and Ron Weiss. A synthetic multicellular system for programmed pattern formation. *Nature*, 434(7037):1130–4, 4 2005.

[7] Steven A. Benner and A. Michael Sismour. Synthetic biology. *Nature Reviews Geneticst*, 6(7):533–43, 7 2005.

[8] Jerome Bonnet, Peter Yin, Monica E Ortiz, Pakpoom Subsoontorn, and Drew Endy. Amplifying genetic logic gates. *Science*, 340(6132):599–603, 2013.

[9] Katie Brenner, David K Karig, Ron Weiss, and Frances H Arnold. Engineered bidirectional communication mediates a consensus in a microbial biofilm consortium. *Proceedings of the National Academy of Sciences*, 104(44):17300–17304, 2007.

[10] Katie Brenner, Lingchong You, and Frances H. Arnold. Engineering microbial consortia: a new frontier in synthetic biology. *Trends in Biotechnology*, 26(9):483–9, 9 2008.

[11] Christian Brettschneider, Rebecca J Rose, Stefanie Hertel, Ilka M Axmann, Albert JR Heck, and Markus Kollmann. A sequestration feedback determines dynamics and temperature entrainment of the KaiABC circadian clock. *Molecular Systems Biology*, 6(1), 2010.

[12] Tal Danino, Octavio Mondragón-Palomino, Lev Tsimring, and Jeff Hasty. A synchronized quorum of genetic clocks. *Nature*, 463(7279):326–30, 1 2010.

[13] L. de Castro. Fundamentals of natural computing: an overview. *Physics of Life Reviews*, 4(1):1–36, 3 2007.

[14] Víctor De Lorenzo and Antoine Danchin. Synthetic biology: discovering new worlds and new words. *EMBO Reports*, 9(9):822–827, 2008.

[15] Irene del Campo, Raúl Ruiz, Ana Cuevas, Carlos Revilla, Luis Vielva, and Fernando de la Cruz. Determination of conjugation rates on solid surfaces. *Plasmid*, 67(2):174–182, 2012.

[16] Michael B. Elowitz and Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, 2000.

[17] Raúl Fernández-López, M Pilar Garcillán-Barcia, Carlos Revilla, Miguel Lázaro, Luis Vielva, and Fernando De La Cruz. Dynamics of the IncW genetic backbone imply general trends in conjugative plasmid evolution. *FEMS Microbiology Reviews*, 30(6):942–966, 2006.

[18] Elisa Franco, Eike Friedrichs, Jongmin Kim, Ralf Jungmann, Richard Murray, Erik Winfree, and Friedrich C Simmel. Timing molecular motion and production with a synthetic transcriptional clock. *Proceedings of the National Academy of Sciences*, 108(40):E784–E793, 2011.

[19] T. S. Gardner, C. R. Cantor, and J. J. Collins. Construction of a genetic toggle switch in *escherichia coli*. *Nature*, 403:339–342, 2000.

[20] Angel Goñi Moreno and Martyn Amos. Model for a population-based microbial oscillator. *BioSystems*, 105(3):286–94, 9 2011.

[21] Angel Goñi-Moreno and Martyn Amos. Continuous computation in engineered gene circuits. *BioSystems*, 109(1):52–56, 2012.

[22] Angel Goñi-Moreno and Martyn Amos. A reconfigurable NAND/NOR genetic logic gate. *BMC Systems Biology*, 6(1):126, 2012.

[23] Angel Goñi-Moreno, Martyn Amos, and Fernando de la Cruz. Multicellular computing using conjugation for wiring. *PLOS ONE*, 8(6):e65986, 2013.

[24] Călin C. Guet, Michael B. Elowitz, Weihong Hsing, and Stanislas Leibler. Combinatorial synthesis of genetic networks. *Science*, 296(5572):1466–70, 5 2002.

[25] Jeff Hasty. Design then mutate. *Proceedings of the National Academy of Sciences*, 99(26):16516–16518, 2002.

[26] Stefanie Hertel, Christian Brettschneider, and Ilka M Axmann. Revealing a two-loop transcriptional feedback mechanism in the cyanobacterial circadian clock. *PLoS Computational Biology*, 9(3):e1002966, 2013.

[27] Seunghee S. Jang, Kevin T. Oishi, Robert G. Egbert, and Eric Klavins. Specification and simulation of synthetic multicelled behaviors. *ACS Synthetic Biology*, 1(8):365–374, 2012.

[28] J. U. Kreft, G. Booth, and J. W. T. Wimpenny. Bacsim, a simulator for individual-based modelling of bacterial colony growth. *Microbiology*, 144(12):3275–3287, 1998.

[29] J. U. Kreft, C. Picioreanu, J. W. T. Wimpenny, and M. C. M. van Loosdrecht. Individual-based modelling of biofilms. *Microbiology*, 147(11):2897–2912, 2001.

[30] Stephen M. Krone, Ruinan Lu, Randal Fox, Haruo Suzuki, and Eva M. Top. Modelling the spatial dynamics of plasmid transfer and persistence. *Microbiology*, 153(Pt 8):2803–16, 8 2007.

[31] Laurent A. Lardon, Brian V. Merkey, Sónia Martins, Andreas Dötsch, Cristian Picioreanu, Jan-Ulrich U. Kreft, and Barth F. Smets. iDynoMiCS: next-generation individual-based modelling of biofilms. *Environmental Microbiology*, 13(9):2416–34, 9 2011.

[32] Stefan Legewie, Dennis Dienst, Annegret Wilde, Hanspeter Herzel, and Ilka M Axmann. Small rnas establish delays and temporal thresholds in gene expression. *Biophysical Journal*, 95(7):3232–3238, 2008.

[33] Matxalen Llosa, F. Xavier Gomis-Rüth, Miquel Coll, and Fernando la de Cruz. Bacterial conjugation: a two-step mechanism for DNA transport. *Molecular Microbiology*, 45(1):1–8, 2002.

[34] Brian V. Merkey, Laurent A. Lardon, Jose M. Seoane, Jan-Ulrich U. Kreft, and Barth F. Smets. Growth dependence of conjugation explains limited plasmid invasion in biofilms: an individual-based modelling study. *Environ Microbiol*, 13(9):2435–52, 9 2011.

[35] Jesús M Miró-Bueno and Alfonso Rodríguez-Patón. A simple negative interaction in the positive transcriptional feedback of a single gene is sufficient to produce reliable oscillations. *PLOS ONE*, 6(11):e27414, 2011.

[36] Paul Oldham, Stephen Hall, and Geoff Burton. Synthetic biology: mapping the scientific landscape. *PLOS ONE*, 7(4):e34368, 2012.

[37] Monica E. Ortiz and Drew Endy. Engineered cell-cell communication via DNA messaging. *Journal of Biological Engineering*, 6(1):16, 2012.

[38] Oliver Purcell, Nigel J. Savery, Claire S. Grierson, and Mario di Bernardo. A comparative analysis of synthetic genetic oscillators. *Journal of the Royal Society: Interface*, 7(52):1503–24, 11 2010.

[39] Priscilla E. M. Purnick and Ron Weiss. The second wave of synthetic biology: from modules to systems. *Nature Reviews Molecular Cell Biology*, 10(6):410–22, 6 2009.

[40] Sergi Regot, Javier Macia, Núria Conde, Kentaro Furukawa, Jimmy Kjellén, Tom Peeters, Stefan Hohmann, Eulàlia de Nadal, Francesc Posas, and Ricard Solé. Distributed biological computation with multicellular engineered networks. *Nature*, 469(7329):207–11, 1 2011.

[41] Guillermo Rodrigo, Javier Carrera, and Alfonso Jaramillo. Genetdes: automatic design of transcriptional networks. *Bioinformatics*, 23(14):1857–1858, 2007.

[42] Guillermo Rodrigo, Javier Carrera, and Alfonso Jaramillo. Computational design of synthetic regulatory networks from a genetic library to characterize the designability of dynamical behaviors. *Nucleic Acids Research*, 39(20):e138–e138, 2011.

[43] Guillermo Rodrigo, Javier Carrera, Thomas E Landrain, and Alfonso Jaramillo. Perspectives on the automatic design of regulatory systems for synthetic biology. *FEBS Letters*, 586(15):2037–2042, 2012.

[44] Guillermo Rodrigo and Alfonso Jaramillo. Autobiocad: full biodesign automation of genetic circuits. *ACS Synthetic Biology*, 2(5):230–236, 2012.

[45] Guillermo Rodrigo, Thomas E Landrain, Eszter Majer, José-Antonio Daròs, and Alfonso Jaramillo. Full design automation of multi-state RNA devices to program gene expression using energy-based optimization. *PLoS Computational Biology*, 9(8):e1003172, 2013.

[46] T. J. Rudge, P. J. Steiner, A. Phillips, and J. Haseloff. Computational modeling of synthetic microbial biofilms. *ACS Synthetic Biology*, 1:345–352, 2012.

[47] Jörn Matthias Schmiedel, Ilka Maria Axmann, and Stefan Legewie. Multi-target regulation by small RNAs synchronizes gene expression thresholds and may enhance ultrasensitive behavior. *PLOS ONE*, 7(8):e42296, 2012.

[48] Piro Siuti, John Yazbek, and Timothy K Lu. Synthetic circuits integrating logic and memory in living cells. *Nature Biotechnology*, 31(5):448–452, 2013.

[49] Chris Smillie, M Pilar Garcillán-Barcia, M Victoria Francia, Eduardo PC Rocha, and Fernando de la Cruz. Mobility of plasmids. *Microbiology and Molecular Biology Reviews*, 74(3):434–452, 2010.

[50] Jesse Stricker, Scott Cookson, Matthew R. Bennett, William H. Mather, Lev S. Tsimring, and Jeff Hasty. A fast, robust and tunable synthetic gene oscillator. *Nature*, 456(7221):516–9, 11 2008.

[51] Alvin Tamsir, Jeffrey J. Tabor, and Christopher A. Voigt. Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'. *Nature*, 469(7329):212–5, 1 2011.

[52] Ron Weiss, T Knight, and Gerald Sussman. Cellular computation and communication using engineered genetic regulatory networks. In Martyn Amos, editor, *Cellular Computing*, pages 120–147. Oxford University Press USA, 2001.

[53] Maximilian Weitz, Andrea M§ckl, Korbinian Kapsner, Ronja Berg, Andrea Meyer, and Friedrich C Simmel. Communication and computation by bacteria compartmentalized within microemulsion droplets. *Journal of the American Chemical Society*, 136(1):72–75, 2013.

[54] N. Willetts and B. Wilkins. Processing of plasmid DNA during bacterial conjugation. *Microbiological Reviews*, 48(1):24–41, 1984.

[55] Joao B. Xavier, Cristian Picioreanu, and Mark C. M. van Loosdrecht. A framework for multidimensional modelling of activity and structure of multispecies biofilms. *Environmental Microbiology*, 7(8):1085–103, 8 2005.

[56] Yohei Yokobayashi, Ron Weiss, and Frances H. Arnold. Directed evolution of a genetic circuit. *Proceedings of the National Academy of Sciences*, 99(26):16587–16591, 2002.

Figure 1: BACTOCOM schematic.

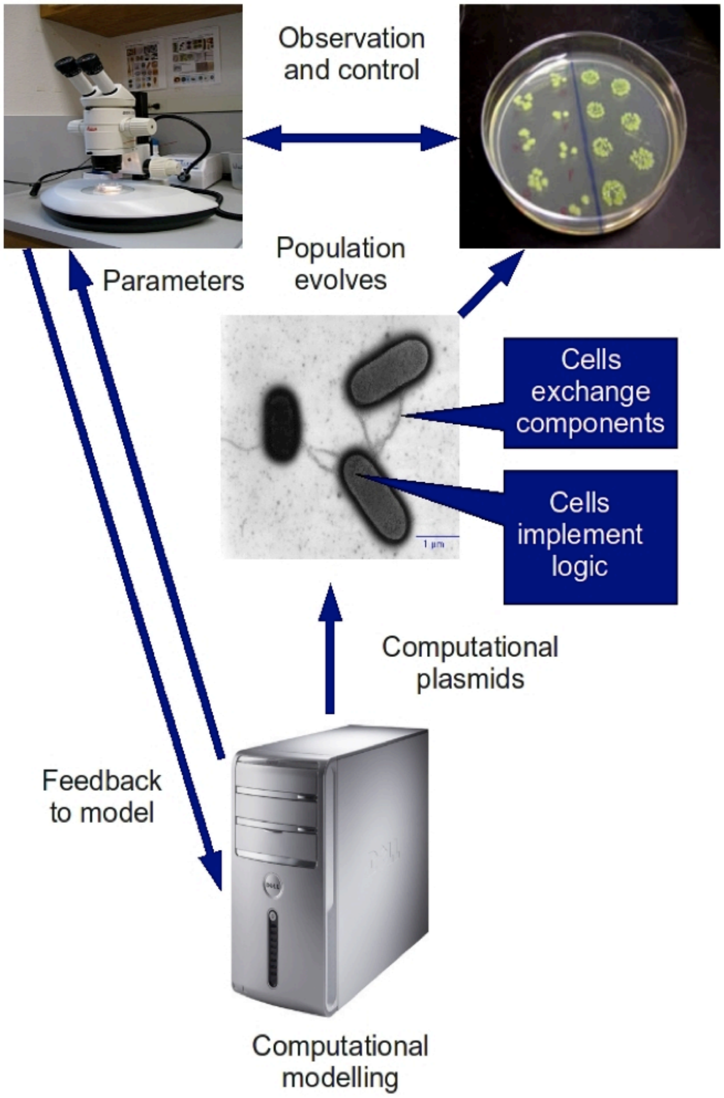


Figure 2: DiSCUS being used to simulate population-based NAND and NOR gates (from [22]).

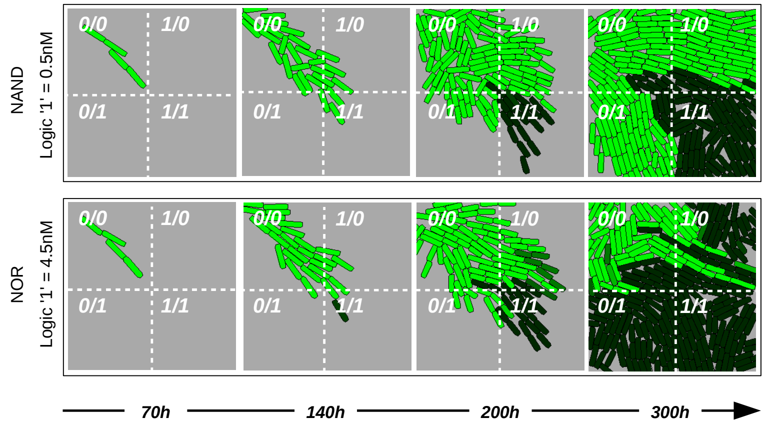


Figure 3: Schematic representation of single-cell NOR gate with toggle switch and GFP reporter.

| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

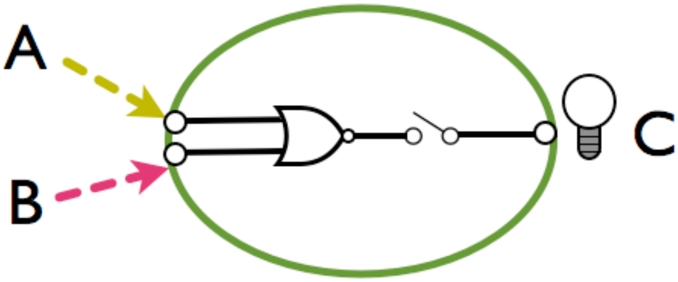
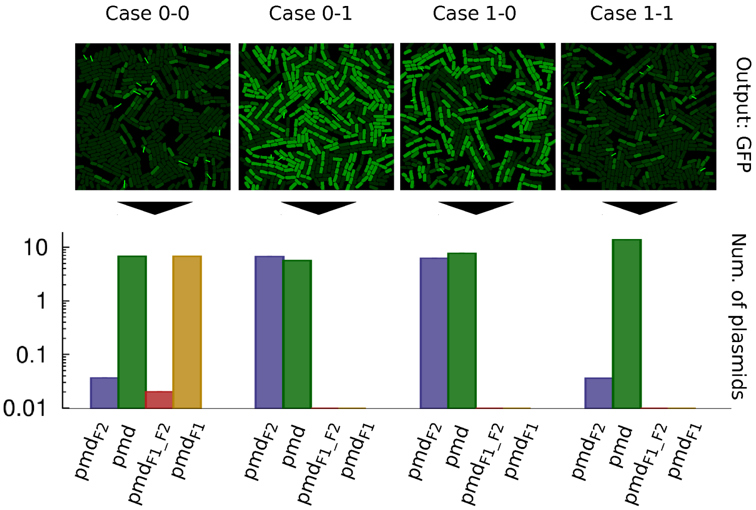


Figure 4: DiSCUS being used to simulate the operation of a multicellular XOR (comparator) gate (from [23]).


Case 0-0
Case 0-1
Case 1-0
Case 1-1
Output: GFP
Num. of plasmids
$pmd_{F2}$
$pmd$
$pmd_{F1\_F2}$
$pmd_{F1}$