DiSCUS: A Simulation Platform for Conjugation Computing

Angel Goñi-Moreno[1] and Martyn Amos[2]

[1] Systems Biology Program, Centro Nacional de Biotecnología, Madrid, Spain
[2] Informatics Research Centre, School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Manchester, UK

Abstract. In bacterial populations, cells are able to cooperate in order to yield complex *collective* functionalities. Interest in population-level cellular behaviour is increasing, due to both our expanding knowledge of the underlying biological *principles*, and the growing range of possible *applications* for engineered microbial consortia. The ability of cells to interact through small signalling molecules (a mechanism known as *quorum sensing*) is the basis for the majority of existing engineered systems. However, horizontal gene transfer (or *conjugation*) offers the possibility of cells exchanging messages (using DNA) that are much more information-rich. The potential of engineering this conjugation mechanism to suit specific goals will guide future developments in this area. Motivated by a lack of computational models for examining the specific dynamics of conjugation, we present a simulation framework for its further study.

(This paper was first presented at the Spatial Computing Workshop of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Paris, France, May 5-9 2014. There were no published proceedings).

1 Introduction

"Imagine a discipline of cellular engineering that tailor-makes biological cells to function as sensors and actuators, as programmable delivery vehicles for pharmeceuticals, or as chemical factories for the assembly of nanoscale structures" (Abelson, *et al.*, talking about *amorphous computing*, in the year 2000 [1]).

This growing discipline is now known as *synthetic biology* [3, 12, 22], and researchers in the field have successfully demonstrated the construction of several types of device based on populations of engineered microbes [29]. Recent work has focussed attention on the combination of *single-cell* intracellular devices [5, 17] with *intercellular* engineering, in order to build increasingly complex systems [6, 11]. As Beal argues, "Biological systems can often be viewed as spatial computers: space-filling collections of computational devices with strongly localized communication." [9] This is *precisely* the view of living cells that we take here; that is, microbes may be engineered to both implement some "program", *and*

share information with other cells in order to implement distributed computations. This concept has already been successfully demonstrated in the laboratory (see [2] for a review), with applications including programmed pattern formation [8], edge detection [35], distributed evaluation of Boolean logic [32, 36], and a synthetic "predator-prey" ecosystem [7]. These papers (and many others) clearly demonstrate how engineered living cells extend, beyond traditional silicon-based machines, the definition of what it means to "compute" .

To date, most work on engineered cell-cell communication has focussed on quorum-sensing (QS) [4], which may be thought of as a communication protocol to facilitate inter-bacterial communication via the generation and receiving of small *signal molecules*. However, recent studies on DNA messaging [31] highlight the importance and utility of transferring whole sets of DNA molecules from one cell (the so-called donor) to another (the recipient). Bacterial *conjugation* is a cell-to-cell communication mechanism [13, 14] that enables such transfers to occur. We have recently proposed the notion of *conjugation computing*: multicellular computation that uses conjugation as its fundamental mode of information transfer [19]. In this paper, we expand on this result, and present full implementation details of our simulation platform for conjugation computing. DiSCUS (Discrete Simulation of Conjugation Using Springs) realistically simulates (in a modular fashion) both intracellular genetic networks and intercellular communication via conjugation. To our knowledge, this is the *first* such platform to offer both of these facilities. We first review previous work on cell simulation, before presenting the details of our model. We validate it against previous experimental work, and then discuss possible applications of our method.

2 Previous work

The rapid development of bacterial-based devices is accompanied by a need for computational simulations and mathematical modelling to facilitate the characterisation and design of such systems. A number of platforms and methods are available for this purpose. Agent-based models (AbMs) are widely used [20], and were first used to study microbial growth in *BacSim* [26]. Continuous models have also been proposed [30], and recent developments make use of hardware optimisation, by using GPUs (Graphics Processing Units) in order to scale up the number of cells simulated [33].

Because of the complexity of the system under study, several computational platforms focus on either specific cellular behaviours (e.g., bacterial chemotaxis [15], morphogenesis of dense tissue like systems [24]) , or on specific organisms (e.g., *Myxococcus xanthus* [23]). Platforms that incorporate cell-cell communication generally focus their attention on quorum-sensing. Simulations of conjugation do exist, but these consider cells as abstracted *circular objects* [27, 34]. We demonstrate in this paper how a consideration of the *shape* of cells is an essential feature for understanding the conjugation behaviour of the population. We now describe our model for bacterial growth, in which conjugation is handled explicitly.

3 Methods

We apply an *individual-based modelling* approach [28] to the study of conjugation dynamics. This models each cell as an individual, mobile entity, each of which is subject to physical forces arising from contact with other cells and the environment (e.g., surfaces). Each cell has a number of different *attributes*, listed in Table 1, which correspond to various physiological states and characteristics.

Table 1. Cell attributes.

Attribute	type	Definition
`shape`	*pymunk*.Shape	Shape of the cell
`program`	[m_0 ... m_i]	List of the i regulatory network molecules (m)
`elongation`	[int,int]	Elongation values (one per cell pole)
`position`	[x,y]	Coordinates of centre point, x and y
`speed`	float	Velocity
`conjugating`	Boolean	Conjugation state
`plasmid`	Boolean	Program state (present/not present)
`role`	int	Donor (0), recipient (1) or transconjugant (2)
`partner`	int	Role of plasmid transfer cell

Bacteria are modelled as rod-shaped cells with a constant radius (parameter `width` in Table 2). Elongation processes occur along the longitudinal axis, which has a minimum dimension of `length`, and division takes place whenever the cell measures 2*`length`. The division of a cell into two new daughter cells is also controlled by `max_overlap`, which monitors the physical *pressure* affecting each cell; if the pressure exceeds this parameter value, the cell delays its growth and division. Thus, a cell with pressure grows slower than without it. The global parameter `growth_speed` (Table 2) also helps us simulate cell flexibility in a realistic fashion. This parameter defines a "cut off" value for the number of iterations in which the physics engine must resolve *all* the current forces and collisions. Thus, smaller values will cause the solver to be effectively "overloaded", and some collisions may, as a result, be partially undetected. This means that cells behave as flexible shapes, which gives the simulation a more realistic performance.

Horizontal genetic transfer (or conjugation) is modelled using an *elastic spring* to connect donor and recipient cells [25]. Parameter `c_time` defines the duration of that linkage, which determines the time in which the DNA is transferred. The springs are constantly monitored to ensure that they physically connect both cells during conjugation. Importantly, during conjugation, the resolution of collisions involving relevant cells considers the forces produced by the spring connection, in order to calculate the final movement of the bacteria. By coupling cells in this way, we obtain realistic population-level physical patterns that emerge as a result of large numbers of conjugation events. This agent-based

algorithm has an iteration-driven structure, where - after initialisation of the main global parameters - it repeatedly performs the following steps for each cell: (1) Update springs (position and timing); (2) Perform cell division (if cell is ready); (3) Elongate cell (every `growth_speed` steps); (4) Handle conjugation; (5) Update physical position.

Conjugation decisions (step 4) made by cells are driven by three sequential steps: (1) *Decide*, following a probability distribution, whether or not to conjugate (one trial per iteration); (2) If conjugating, randomly select a mate from surrounding bacteria (if present); (3) If valid mate is found, effect conjugation transfer.

The discrete probability distribution used for the conjugation process is $C(N, p, \texttt{c\_time})$, where N is the number of trials in a cell lifetime (`width` * `length`), p is the success probability in each trial (with p $\in [0...1]$) and `c_time` is the time interval during $p = 0.0$ (i.e., when the cell is already conjugating). As stated in [34], p can vary, depending on whether the cell is a donor (`p_d`), a transconjugant that received the DNA message from a donor (`p_t1`), or a transconjugant that received the DNA from another transconjugant (`p_t2`).

Table 2. Global simulation parameters.

Parameter	Definition
`screenview`	Size of the simulated *world*
`width`	Width of each cell (lattice squares)
`network_steps`	Number of steps of the ODEs per `Gt`
`number_donors`	Initial number of donor cells
`real_Gt`	Real doubling time of the studied cells (minutes)
`Gt`	Doubling time of the simulated cells (iterations)
`number_recipients`	Initial number of recipient cells
`length`	Length of each cell (lattice squares)
`max_overlap`	Pressure tolerance of cells
`bac_friction`	Friction coefficient (Coulomb friction model)
`spring_damping`	The amount of viscous damping to apply
`bac_mass`	Mass of the cell (for calculating the moment)
`c_time`	Duration of the conjugation process
`p_d`	Probability of conjugation event (donors)
`p_t1`	Probability of conjugation event (transconj.1)
`p_t2`	Probability of conjugation event (transconj.2)
`spring_rest_length`	Natural sprint expansion/contraction
`growth_speed`	Iterations between elongation processes
`spring_stiffness`	The tensile modulus of the spring
`cell_infancy`	Time lag (percentage)
`pymunk_steps`	Update the space for the given time step
`pymunk_clock_ticks`	Frame frequency (FPS - frames per second)

Intracellular circuits (that is, any new genetic components that are introduced into the cells in order to implement a computation) are modelled separately, and then held in each cell by storing the state (i.e., protein concentrations, etc.) of the circuit in an attribute of the cell (`program`). Thus, there are effectively as many copies of the circuit as cells in the simulation (the number of cells we currently handle can range from single digits to around two thousand before we hit significant performance issues). This circuit simulation is implemented in a modular fashion, so that the internal cellular "program" may be easily replaced with any other. In this paper we demonstrate the principle using a two-component genetic oscillator as the DNA message that is exchanged through conjugation. The ordinary differential equations (ODEs) for this circuit are:

$$\frac{dx}{dt} = \Delta \left(\beta \frac{1 + \alpha x^2}{1 + x^2 + \sigma y^2} - x \right) \tag{1}$$

$$\frac{dy}{dt} = \Delta \gamma \frac{1 + \alpha x^2}{1 + x^2} - y \tag{2}$$

which are detailed in [21], as well as the meaning and value of the parameters (we use the same values in the code provided). We recently used our software platform to investigate the spatial behaviour of a *reconfigurable* genetic logic circuit (without conjugation) [18], which demonstrates (1) how it may easily be modified to accommodate different sets of equations, and (2) how it may be used as a "general purpose" cell simulation platform, with conjugation "turned off". The actions controlling the growth rates of cells occur on a longer time scale than the integration steps that control molecular reactions (as equations 1 and 2). In order to ensure synchronisation, the parameter `network_steps` defines the number of integration steps of the ODEs that run per `Gt`. Thus, a number of `network_steps/Gt` integration steps will update the attribute `network` of each cell every iteration. Other important physical parameters listed in Table 2 are `spring_rest_length`, `spring_stiffness` and `spring_damping`; these are three parameters to model the material and behaviour of the bacterial pilus (i.e. the spring) during conjugation. Parameter `cell_infancy` is a delay period, during which a cell is considered to be too young to conjugate (as observed experimentally [34]). Parameters `pymunk_steps` and `pymunk_clock_ticks` are used by the physics engine to update the world, and may be adjusted by the user in order to alter the performance of the simulation (machine dependent). Parameters `bac_mass` and `bac_friction` play a role in collision handling. Our platform is written in *Python*, and makes use of the physics engine *pymunk* (www.pymunk.org) as a wrapper for the 2D physics library Chipmunk, which is written in C (www.chipmunk-physics.net/). As cells are represented as semi-rigid bodies in a 2D lattice, pymunk handles the physical environment on our behalf. For monitoring purposes, parameters `Gt` and `real_Gt` allow us to stablise the relation between iterations and clock minutes: $minute$ = `Gt/real_Gt` (units: iterations).

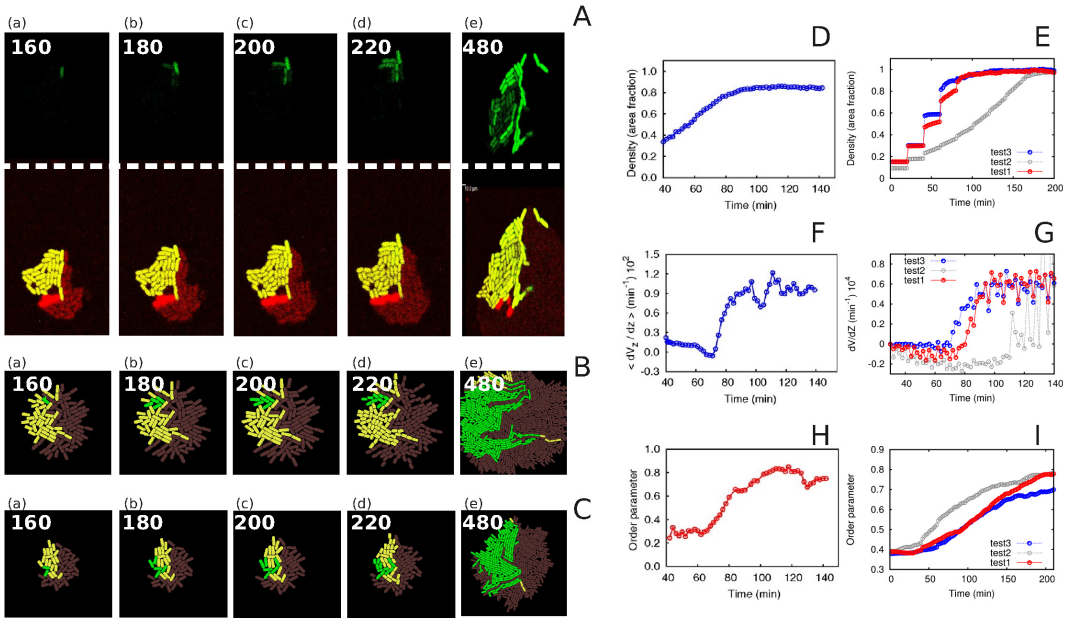

Fig. 1. Validation of cell movement and conjugation dynamics using real data. (A): Figure extracted from [34] where a colony of *Pseudomonas putida* is divided into dark red donor cells (DsRed), yellow recipient cells (YFP) and transconjugants, expressing both yellow and green light (YFP and GFP). The 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). Graphs (D), (F) and(H) are extracted from [37], and show **experimental results** of *Escherichia Coli* growth regarding density, velocity and ordering (respectively). Graphs (E), (G) and (I) correspond to our **simulation results**, using similar conditions to [37], for the same parameters (density, velocity and ordering respectively). Tests 1, 2 and 3 in graphs correspond to different spatial distribution of cells inside the microfluidic chanel (details in text). This figure first appeared as Supporting Information Figure S7 in [19] (i.e., not as part of the main paper).

4 Results

We now describe the results of experiments to validate our conjugation model, using three sets of simulations. We first validate individual conjugation dynamics; then we validate the biomechanical properties of the simulation; the final set of experiments studies the effects of mixing on conjugation dynamics.

4.1 Conjugation dynamics

The objective of the first set of experiments is to validate the software in terms of *conjugation dynamics*. For that purpose, we first focus on conjugation, using images of a *Pseudomonas putida* population (Figure 1A) extracted, with permission, from [34]. These show donor cells (dark red) growing in contact with recipients (yellow). The DNA information they share after conjugation makes the transconjugant cells display GFP (green fluorescent protein). We adjusted the parameters of our simulations until the behaviour matched the images of real cells (two simulations shown: Figures 1B and 1C), in terms of both time-series behaviour and the type of physical pattern displayed. The algorithm for this adjustment used information on the number of transconjugants, donors and recipients at a particular time (taken from images of actual colonies), and then explored (in the simulation) the space of conjugation probabilities until values were found that gave rise to the observed numbers). It is important to note that the differential probabilities of conjugation of donors and transconjugants (higher in the latter) causes directional spreading of the DNA information. After the first transconjugant appears (160 minutes), the newly-formed transconjugants appear -most probably - in the immediate neighbourhood. The final parameter values used to reproduce this experiment are: `width`=5, `length`=15, `growth_speed`=30, `p_d`=0.001, `p_t1`=0.02, `p_t2`=0.05 and `c_time`=450 (the rest of the parameters are as defined in the DiSCUS distribution). Movie *DemoConjugation1* (found in the project repository) shows a simulation of a similar experiment where the transconjugants do not act as new donors.

4.2 Biomechanical properties

The second set of validation experiments focuses on *biomechanical movement*. We use experimental data from [37], which describe an *Escherichia coli* colony growing in a microfluidic channel (30 * 50 * 1 μm^3) (Figures 1D, 1F and 1H). Using exactly the same setup (`width`=5, `length`=24, `growth_speed`=30) we highlight how different initial positioning of cells inside the channel can affect the final result (*test1*, with more cells observed in the centre than at the edges; *test2* with all cells initially in the centre; *test3* with all cells homogeneously spread along the channel). Density graphs (Figures 1D and 1E) show the increasing curve as the channel becomes more populated (results vary depending on which area is considered for monitoring). Velocity gradients (Figures 1F and 1G) depict the differential velocity across the longitudinal axis of the channel with respect to the centre (we see negative values when the cells in the centre move faster

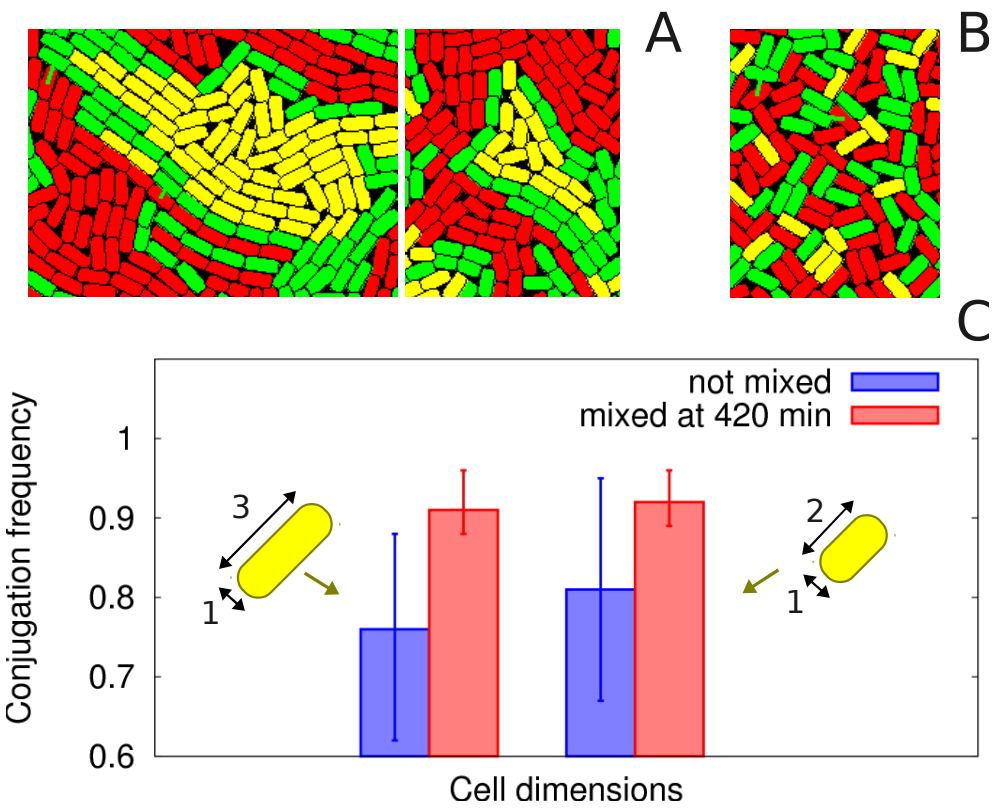

Fig. 2. Effects of manual mixing on conjugation frequency. (A): Recipient-trapping behaviour of a population with donors (red), transconjugants (green) and recipients (yellow). Two snapshots depict clearly-observed clusters. (B): Population after random mixing, where the clusters are automatically dissolved. (C): Graph showing conjugation frequencies ($Y = T/(R + T)$) of 560-minute experiments (ratio $D/R = 50\%$). Blue bars represent Y on an untouched population, while red bars represent Y when the population is mixed at 420 minutes. The two sets of bars correspond to experiments with different cell dimensions (1x3 -left- and 1x2 -right). Error bars show variation across 15 experiments of each class. This figure first appeared as Supporting Information Figure S8 in [19] (i.e., not as part of the main paper).

than the rest). The difference in the y axis is due to our considering different spacial intervals in the velocity gradient calculation. Ordering graphs (Figures 1H and 1I) are based on calculating the cosine of a cell's angle with respect to the longitudinal axis of the channel (e.g. angle 0, cos(0)=1, completely aligned). As time increases, we see that the cells tend to align themselves.

4.3 Effects of mixing

Conjugation behaviour within a population may be altered in different ways to achieve different behaviours, depending on the desired application. For example, in the previous experiments described in this paper, transconjugants are unable to act as recipients (simulating a *radical* entry exclusion [16]). That is to say, they will not receive more plasmids (genetic circuits) from either donors or transconjugants. Furthermore, we may also engineer the transconjugants to stop acting as *new donors* [14], so that only the original donors have the ability to transfer the DNA message. *Mixing* of the cell population becomes essential in this last scenario, in order to ensure maximal contact between donors and recipients.

Investigations of how manual mixing can affect conjugation frequencies are described in in [14], using an *Escherichia coli* population. We now reproduce those results using our software, and give valuable insight into the reasons for that behaviour: the *isolation* of the recipients. For that purpose (Figure 2) we grow a population of donors (D, red) and recipients (R, yellow) in which the ratio D/R is 50% and the transconjungants (T, green) are unable to act as new donors. The frequency of conjugation, Y, is measured as $Y = T/(R + T)$. The graph in Figure 2C shows the frequency after 560 minutes of *untouched* populations (not mixed, blue bars) and populations that have been *manually mixed* at 420 minutes (red bars). The difference that the mixing produces is based on the isolation of the recipients in untouched populations. Figure 2A shows two different occasions in which clusters of recipients are formed, where the transconjugants do not allow donors to reach new possible mates. After the population is completely "shuffled" (2B), the clusters are dissolved, and new pairs of donor-recipient can arise in the new topology.

An interesting result from Figure 2C is the fact that the smaller the size of the cell, the higher results we observe for conjugation frequencies. This may be due to the fact that smaller cells are able to slip through physical gaps, and the biomechanical ordering of the population becomes more "fuzzy". This underlines the importance of considering the physical shape of cells, since circle-shaped cells would not give valid results. Importantly, all of these results are *entirely consistent* with the behaviour observed in the laboratory study [14].

5 Discussion

The conjugation model presented here is the first agent-based model to explicitly simulate conjugation processes with growing rod-shaped cells. Full validation against real data is performed, which shows the capacity of the software to reproduce observed behaviour. In addition, the mixing study offers valuable insights into the design of multi-strain populations. The software also allows for genetic *programs* to be *installed* inside cells; the potential for horizontal gene transfer to recreate distributed information processing within a microbial consortium is of significant interest in synthetic biology/spatial computing [10], and the software presented will aid the design and testing of systems before their *wet-lab* implementation. Possible future work may focus on further validation of the model through (1) studying the frequency of conjugation in different bacterial strains, and under different conditions, (2) studying the effect of the cell's shape and/or doubling (reproduction) time, and (3) investigating mixing effects caused by the topology of the region(s) bounding the cell colony. The computational cost of the simulations may also prove to be a limiting factor, so it may be useful to investigate parallelisation of the code (either on GPUs, or by using a platform such as MPI). This may, in the future, open up the possibility of using the code for three-dimensional biofilm studies.

Simulation code and movies are available at http://www.bactocom.eu.

Acknowledgments

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).

References

1. Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas F. Knight Jr, Radhika Nagpal, Erik Rauch, Gerald Jay Sussman, and Ron Weiss. Amorphous computing. *Communications of the ACM*, 43(5):74–82, 2000.
2. Martyn Amos. Population-based microbial computing: a third wave of synthetic biology? *International Journal of General Systems*, 43(7):770–782, 2014.
3. Ernesto Andrianantoandro, Subhayu Basu, David K. Karig, and Ron Weiss. Synthetic biology: new engineering rules for an emerging discipline. *Mol Syst Biol*, 2:2006.0028, 2006.
4. Steve Atkinson and Paul Williams. Quorum sensing and social networking in the microbial world. *J R Soc Interface*, 6(40):959–78, 11 2009.
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. William Bacchus and Martin Fussenegger. Engineering of synthetic intercellular communication systems. *Metab Eng*, 16:33–41, 3 2013.
7. Frederick K. Balagaddé, Hao Song, Jun Ozaki, Cynthia H. Collins, Matthew Barnet, Frances H. Arnold, Stephen R. Quake, and Lingchong You. A synthetic Escherichia coli predator-prey ecosystem. *Mol Syst Biol*, 4:187, 2008.
8. 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.
9. Jacob Beal. Bridging biology and engineering together with spatial computing. In Marian Gheorge, Gheorge Paun, Grzegorz Rozenberg, Arto Salomaa, and Sergey Verlan, editors, *Membrane Computing*, volume LNCS 7184, pages 14–18. Springer, 2012.
10. Jacob Beal and Jonathan Bachrach. Cells are plausible targets for high-level spatial languages. In *Self-Adaptive and Self-Organizing Systems Workshops, 2008. SASOW 2008. Second IEEE International Conference on*, pages 284–291. IEEE, 2008.
11. Katie Brenner, Lingchong You, and Frances H. Arnold. Engineering microbial consortia: a new frontier in synthetic biology. *Trends Biotechnol*, 26(9):483–9, 9 2008.
12. Allen A. Cheng and Timothy K. Lu. Synthetic biology: an emerging engineering discipline. *Annu Rev Biomed Eng*, 14:155–78, 2012.
13. Fernando de la Cruz, Laura S. Frost, Richard J. Meyer, and Ellen L. Zechner. Conjugative DNA metabolism in Gram-negative bacteria. *FEMS Microbiol Rev*, 34(1):18–40, 1 2010.
14. 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):174–82, 3 2012.
15. Thierry Emonet, Charles M. Macal, Michael J. North, Charles E. Wickersham, and Philippe Cluzel. Agentcell: a digital single-cell assay for bacterial chemotaxis. *Bioinformatics*, 21(11):2714–21, 6 2005.
16. M. Pilar Garcillán-Barcia and Fernando de la Cruz. Why is entry exclusion an essential feature of conjugative plasmids? *Plasmid*, 60(1):1–18, 7 2008.
17. Timothy S. Gardner, Charles R. Cantor, and Jim J. Collins. Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403:339–342, 2000.
18. Angel Goñi Moreno and Martyn Amos. A reconfigurable NAND/NOR genetic logic gate. *BMC Syst Biol*, 6(1):126, 9 2012.
19. Angel Goñi Moreno, Martyn Amos, and Fernando de la Cruz. Multicellular computing using conjugation for wiring. *PLoS ONE*, 8(6):e65986, 2013.
20. Thomas E. Gorochowski, Antoni Matyjaszkiewicz, Thomas Todd, Neeraj Oak, Kira Kowalska, Stephen Reid, Krasimira T. Tsaneva-Atanasova, Nigel J. Savery, Claire S. Grierson, and Mario di Bernardo. BSim: an agent-based tool for modeling bacterial populations in systems and synthetic biology. *PLoS ONE*, 7(8):e42790, 2012.
21. Raul Guantes and Juan F. Poyatos. Dynamical principles of two-component genetic oscillators. *PLoS Computational Biology*, preprint(2006):e30, 2005.
22. Matthias Heinemann and Sven Panke. Synthetic biology–putting engineering into biology. *Bioinformatics*, 22(22):2790–9, 11 2006.
23. Antony B. Holmes, Sara Kalvala, and David E. Whitworth. Spatial simulations of myxobacterial development. *PLoS Comput Biol*, 6(2):e1000686, 2 2010.

24. J. A. Izaguirre, R. Chaturvedi, C. Huang, T. Cickovski, J. Coffland, G. Thomas, G. Forgacs, M. Alber, G. Hentschel, S. A. Newman, and J. A. Glazier. Compucell, a multi-model framework for simulation of morphogenesis. *Bioinformatics*, 20(7):1129–37, 5 2004.
25. Jana Jass, Staffan Schedin, Erik Fällman, Jörgen Ohlsson, Ulf J. Nilsson, Bernt Eric Uhlin, and Ove Axner. Physical properties of escherichia coli p pili measured by optical tweezers. *Biophys J*, 87(6):4271–83, 12 2004.
26. 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.
27. 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.
28. 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. *Environ Microbiol*, 13(9):2416–34, 9 2011.
29. Javier Macía, Francesc Posas, and Ricard V. Solé. Distributed computation: the new wave of synthetic biology devices. *Trends Biotechnol*, 30(6):342–9, 6 2012.
30. Pontus Melke, Patrik Sahlin, Andre Levchenko, and Henrik Jönsson. A cell-based model for quorum sensing in heterogeneous bacterial colonies. *PLoS Comput Biol*, 6(6):e1000819, 6 2010.
31. Monica E. Ortiz and Drew Endy. Engineered cell-cell communication via DNA messaging. *J Biol Eng*, 6(1):16, 2012.
32. 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.
33. Timothy J. Rudge, Paul J. Steiner, Andrew Phillips, and Jim Haseloff. Computational modeling of synthetic microbial biofilms. *ACS Synthetic Biology*, 1:345–352, 2012.
34. Jose Seoane, Tatiana Yankelevich, Arnaud Dechesne, Brian Merkey, Claus Sternberg, and Barth F. Smets. An individual-based approach to explain plasmid invasion in bacterial populations. *FEMS Microbiol Ecol*, 75(1):17–27, 1 2011.
35. Jeffrey J. Tabor, Howard M. Salis, Zachary Booth Simpson, Aaron A. Chevalier, Anselm Levskaya, Edward M. Marcotte, Christopher A. Voigt, and Andrew D. Ellington. A synthetic genetic edge detection program. *Cell*, 137(7):1272–81, 6 2009.
36. 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.
37. Dmitri Volfson, Scott Cookson, Jeff Hasty, and Lev S. Tsimring. Biomechanical ordering of dense cell populations. *Proceedings of the National Academy of Sciences*, 105(40):15346–15351, 2008.