Simulating multicell populations with an accelerated gro simulator

Martín Gutiérrez[1,2] and Alfonso Rodríguez-Patón[1]

[1]Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Madrid, Spain
[2]Escuela de Informática y Telecomunicaciones, Universidad Diego Portales, Santiago, Chile
mgutierrez@fi.upm.es, arpaton@fi.upm.es

Abstract

Synthetic Biology follows a Design-Build-Test-Learn cycle for the construction of novel biocircuits with a predefined behavior. Recently, a few simulation and bioCAD tools to assist synthetic biologists in the biocircuit engineering process have started to appear. One of these simulators is gro, a 2D Individual based Model (IbM). The objectives of our work on gro was to: Implement an IbM platform that can quickly simulate a large amount of E. Coli cells and to extend the array of functions that a cell executes. Our goal is to provide a fast and easy to use IbM simulator for synthetic circuits. We developed multiple extensions for gro as external modules programmed in C/C++. We improved the simulator speed through a new physics engine called CellEngine. It implements a ring-based algorithm for cell shoving. A "binary protein" module that directs intracellular regulation along with a new biocircuit language specification layer were added to the simulator as well. We also improved the existing signal capabilities in the simulator. Finally, bacterial conjugation (transmission of plasmids between neighbor bacteria) was implemented into gro as a new intercellular communication mechanism. Our new gro version enables the simulation of growing colonies having up to 10^5 bacteria in minutes; while the previous version took over five days to reach the same number. The new specification layer and the underlying regulatory module offer a new structure-oriented paradigm where cell behavior emerges from the specification. All of the new features extend gro to be a fast and versatile tool for prototyping multicellular biocircuits in microbial populations.

Introduction

Prototyping tools are becoming a requirement in the process of implementing multicellular synthetic circuits. IbM simulators are well-suited tools for this purpose, as each individual entity represents a cell. Population-level properties emerge from the interaction among cells and the environment. IbM cell-colony simulators include Jang et al. (2012); Rudge et al. (2012). A recent review of these and other simulators was published in Gorochowski (2016). Realistic simulations involve a large amount of bacteria, a versatile, extensible and up-to-date function toolkit, and should also be easily translated from and to wet lab experiments. We have chosen gro, presented in Jang et al. (2012), as a base system for improving as the source code is open and allows for prompt extension. gro can be useful tool for complementing evolutionary game theory due to its strong spatial component. It can also improve results obtained through cellular automata simulations, as growth and colony physics are modeled realistically. Another recent application of gro was shown in Pascalie et al. (2016), where the authors study developmental biology within synthetic bacterial architectures. Our group has developed external modules that couple with gro to improve it: a new physics engine, CellEngine, improves simulator performance. CellPro, a module for intracellular protein expression dynamics and its accompanying specification language, ProSpec. Intercellular signalling was improved and packed into a new module, CellSignals. Finally, a new form of intercellular communication, bacterial conjugation, was implemented directly into gro. These extensions are fully reported in Gutiérrez et al. (2017), and most of them being independent and external modules, can be reused by other systems. All sources can be found at https://github.com/liaupm/GRO-LIA, and more information at http://www.lia.upm.es.

CellEngine

CellEngine is a new physics engine that handles bacterial shoving in gro. It implements an algorithm based on rings. Rings are groups of bacteria exhibiting similar pressure. The expansion algorithm assumes that position and orientation of each cell is dependent on its closest neighbors and that bacterial colonies grow outwards. Unlike in the traditional collision detection - response cycle, rings are first arranged by using few iterations of this cycle (local overlap is resolved), and then relocated outwards (accounting for global pressure). This algorithm is linear in the number of cells, as each cell position is computed at most twice. It is more efficient than the quadratic complexity of the collision detection - response cycle. The speedup achieved by CellEngine results in gro now being able to simulate colonies of 10^5 bacteria in 8 minutes, compared to the previous version which could reach about 5000 simulated bacteria in the same time. Globally, an improvement of between one and two orders of

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

magnitude on the simulated bacteria was achieved.

CellPro and ProSpec

In the vast majority of cases, proteins are responsible for cell function. Following this assumption, we designed and implemented CellPro: a simulator for protein expression. We simplified and aggregated the process of gene expression, yielding the following directives: 1) Proteins are assumed to be binary, they are present (1) or absent (0). 2) The whole protein expression process is achieved in an activation time (t_{act}). Similarly, protein degradation occurs in a degradation time (t_{deg}). Both of these parameters are simple time delays driving the whole mechanism. 3) Protein expression and degradation are controlled by promoters, which are modeled in CellPro as logic gates whose inputs are proteins. With these directives it is possible to construct genetic circuits by making each of these elements interact. Also, a system in which cell function (such as glowing, growing, death, signal emission, etc.) is triggered based on protein conditions was put in place. To complement CellPro, we built a language specification layer (ProSpec) for quick and simple simulation implementation. Unlike the guarded command based language of the original version of `gro`, ProSpec is based on a structural form of specification. This is, proteins are organized into operons, operons are then placed in plasmids which in turn are included in cells. This form of specification seeks to resemble how microbiologists describe a genetic circuit. ProSpec is compatible with the original form of specification.

CellSignals and Bacterial conjugation

Cell-cell communication is a basic requirement for designing and simulating multicellular synthetic circuits. Cells can communicate locally (bacterial conjugation) or with a longer range (environmental signals or phages). `gro` already provided environmental signal functionalities. However, they were designed for a system in which only a few thousand bacteria would be simulated. This would limit the physical area in which signals were computed for diffusion. CellSignals implements a signal diffusion simulator offering a dynamic resizable area for signal allocation and provides further configurability in the emission/absorption of signals: cells can now emit signals over the span of their whole surface, the diffusion coefficients of the finite element method can be set by the user and different kinds of grids can be created. Bacterial conjugation, a local communication method in which a plasmid is transferred from a donor bacterium to a recipient one in its immediate neighborhood was added to `gro`. The occurrence of conjugation is based on a configurable conjugation rate. Both of these tools provide the basis for simulating communication protocols (such as Quorum Sensing). An example of population interaction is shown and briefly explained in Figure 1.

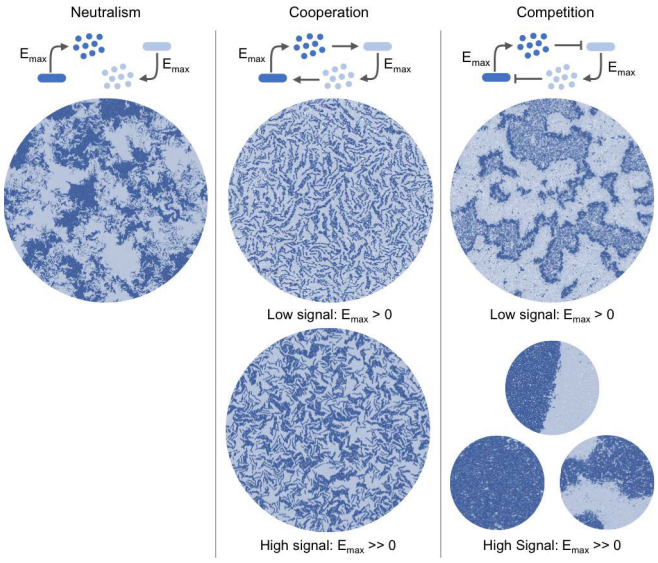

Figure 1: Two simulated bacterial populations interact through environmental signals generated by each population that cause a specific effect (null, detrimental or beneficial) on the opposite population. Maximum signal emission rate is denoted by E_{max}. Three scenarios are shown: Neutralism - both signals have a null effect. Cooperation - both signals have a beneficial effect. Competition - both signals are detrimental.

Acknowledgements

The authors thank Fernando de la Cruz lab, Haseloff lab and Eric Klavins. This work was supported by the European Union project PLASWIRES (612146/FP7- FET-Proactive) and by Spanish MINECO projects TIN2012-36992 and TIN2016-81079-R. Universidad Diego Portales and CONICYT through Becas Chile also supported this work.

References

Gorochowski, T. E. (2016). Agent-based modelling in synthetic biology. *Essays Biochem.*, 60(4):325–336.

Gutiérrez, M., Gregorio-Godoy, P., Pérez del Pulgar, G., Muñoz, L., Sáez, S., and Rodríguez-Patón, A. (2017). A New Improved and Extended Version of the Multicell Bacterial Simulator `gro`. *ACS Synth. Biol.* doi: `10.1021/acssynbio.7b00003`.

Jang, S. S., Oishi, K. T., Egbert, R. G., and Klavins, E. (2012). Specification and simulation of synthetic multicelled behaviors. *ACS Synth. Biol.*, 1(8):365–374.

Pascalie, J., Potier, M., Kowaliw, T., Giavitto, J.-L., Michel, O., Spicher, A., and Doursat, R. (2016). Developmental design of synthetic bacterial architectures by morphogenetic engineering. *ACS Synth. Biol.*, 5(8):842–861.

Rudge, T. J., Steiner, P. J., Phillips, A., and Haseloff, J. (2012). Computational modeling of synthetic microbial biofilms. *ACS Synth. Biol.*, 1(8):345–352.

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).