

Implementing *in vivo* Cellular Automata using Toggle Switch and Inter-Bacteria Communication Mechanism

Yasubumi Sakakibara
Biosciences and Informatics,
Keio University
3-14-1 Hiyoshi, Kohoku-ku
Yokohama, 223-8522, Japan
yasu@bio.keio.ac.jp

Hiroataka Nakagawa
The Kansai Electric Power
Co., Inc.
6-16 Nakanoshima
3-chome, Kita-ku
Osaka 530-8270, Japan

Yusaku Nakashima
Biosciences and Informatics,
Keio University
3-14-1 Hiyoshi, Kohoku-ku
Yokohama, 223-8522, Japan
nakashima@dna.bio.keio.ac.jp

Katsuyuki Yugi
Biosciences and Informatics,
Keio University
3-14-1 Hiyoshi, Kohoku-ku
Yokohama, 223-8522, Japan
yugi@bio.keio.ac.jp

ABSTRACT

Artificial genetic circuits have been recently proposed to realize unprecedented biological systems which do not exist in nature by combining several genes. For example, recombinant *Escherichia coli* provides novel artificial functions such as blinking periodically or invading cancer cells. In protein engineering, new functional proteins have been created by site-directed mutagenesis methods. The main purpose of our research is to implement the Cellular Automata (CA) *in vivo* employing these methods. Computational capacity of CA is equivalent to the universal Turing Machine. CA are multi-cellular systems in which uniform cells are allocated on lattice grid. Each cell has a finite number of states. The state transition is determined by the current state and the states of neighbor cells. Three mechanisms are required to implement CA in bacteria: (1) Sending and receiving signals between cells, (2) Sustaining the state, (3) Sensing input signals and changing the state following the state transition rules. To encode signals for the mechanism (1), we use small molecules inducing transcription. For the mechanism (2), the toggle switch circuit was employed to represent a finite number of states using gene expression. If one gene does express and the other does not, the state is 1, and vice versa, 0. For the mechanism (3), the state transition functions were implemented as logic gates using transcriptional regulatory proteins which bind to specific signal molecules (chemicals). In this paper, we applied this method to *in vivo* implementation of one-dimensional CA and corroborated whether it behaves correctly. In principle, this method can be applied to any dimensional CA by employing more signal molecules

to construct more complex logic gate. Furthermore, we succeeded to execute one step calculation of one-dimensional CA *in vivo* in a laboratory experiment.

1. INTRODUCTION

One of the most significant goals of synthetic biology is rewiring of genetic regulatory networks to realize controllable systems in a cell. Recently, this technology has been applied to development of biomolecular computations *in vivo*. Nakagawa et al. [5] proposed a novel framework to develop a programmable and autonomous *in vivo* computer using *E. coli*, and implemented *in vivo* finite-state automata by employing the protein-synthesis mechanism of *E. coli*. However, previous works could not exhibit computational systems higher than finite state machine in terms of Chomsky hierarchy. Here we show the construction of a genetic circuit in *E. coli* which provides a one-dimensional cellular automaton, i.e., a Turing-complete computer. For the first step toward this *in vivo* parallel computing, we attempt to develop a bacterial cellular automata by applying the technique of artificial genetic circuit and protein engineering. The designed genetic network can sustain binary states represented by expression of certain genes and can change one state to another in the presence of signalling molecules informing the adjacent cells' states.

We designed and implemented the mechanism (1) using quorum-sensing mechanism with *luxI* and *luxR* genes and *lux box* promoter, the mechanism (2) using *lacI* gene and *cI857* gene, and the mechanism (3) using *luxR* gene and its mutant. For the mechanism (3), the output was observed via GFP fluorescence. We experimentally corroborated the function of mechanism (1), the implementation of finite states using mechanism (2), and executed one step of the state transition *in vivo*. The signals were transmitted manually to a neighbor colony of bacteria which was assumed to represent a cell of CA. Constructed cellular automata provides similar behavior to Rule 250 automata. We believe that this is the first Turing equivalent biomolecular computation performed in living cells.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Bionetics '07, December 10-13, 2007, Budapest, Hungary
Copyright 2007 ICST 978-963-9799-11-0

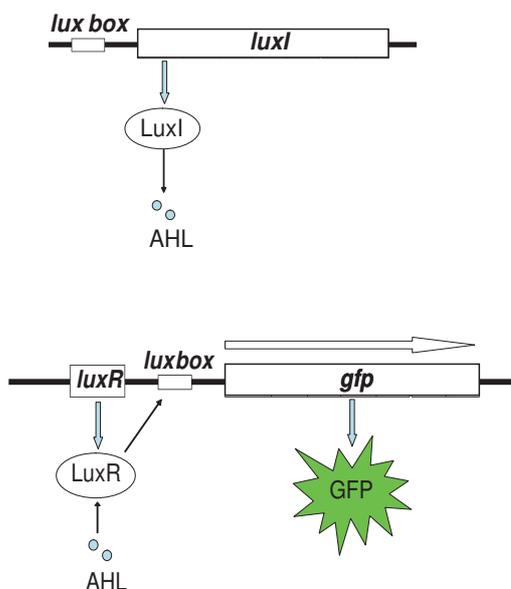


Figure 1: (above) Sender cell consisting of *luxI* to synthesize AHL, and (below) receiver cell consisting of *luxR* and *lux box* promoter to detect AHL.

2. MOLECULAR CELLULAR AUTOMATA

2.1 Molecular communication by quorum sensing

One necessary and fundamental mechanism to implement the cellular automata is the communication between cells for sending and receiving the signals. Such inter-cellular communication was realized by means of the quorum-sensing mechanism, which in nature allows the living bacteria to detect cell density [6]. The quorum sensing mechanism relies on the synthesis and detection of a species unique chemical, called autoinducer, which mediates inter-cellular communications. In *Vibrio fischeri*, one of autoinducer chemical, AHL (Acyl Homoserine Lacton), has been identified so far. The gene *luxI* to synthesize this chemical, and the response mechanism to AHL with the gene *luxR* have also been identified. The LuxR protein first detects AHL, and after interaction with AHL, LuxR binds to the *lux box* that is located on the promoter of the LuxI operon. The binding of LuxR to *lux box* activates the transcription of downstream genes.

We have implemented the *in vivo* inter-cellular communication by using quorum sensing mechanism. First, we have developed a plasmid consisting of *luxI* gene to synthesize AHL and a plasmid consisting of *luxR* gene and *lux box* promoter followed by a reporter gene *gfp* to detect AHL compound. The *E.coli* cell, called sender cell, was transformed with plasmid of *luxI* by electroporation, and the cell, called receiver cell, was transformed with plasmid of *luxR*, *lux box* and *gfp*. (See Figure 1).

A laboratory experiment was performed to confirm the functions of sender cell and receiver cell. We have made two plates one of which only contains receiver cell and the other contains both sender and receiver cells. The sender cells are placed at the center of the plate and the receiver cells are placed around the sender cells. The result is shown in Figure 2. The strong fluorescence was only detected in

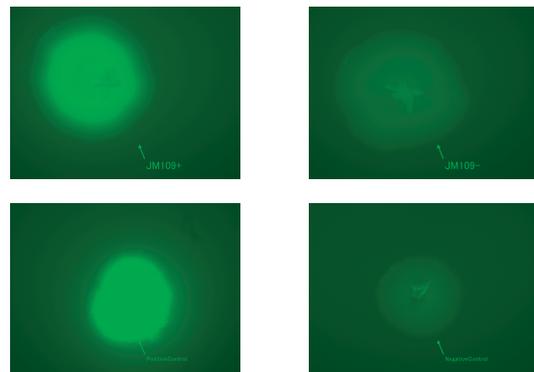


Figure 2: (above left) Strong fluorescence observed in sender-receiver plate, (above right) a low fluorescence observed in receiver-only plate, (below left) positive control, and (below right) negative control.

the sender-receiver plate and a low fluorescence which was comparable with the negative control plate was detected in the receiver-only cell. This result clearly shows that the receiver cell expressed the reporter gene *gfp* only at receiving the AHL signals and the sender cell succeeded to send the AHL signals. Therefore, we succeeded implementing the inter-cellular communication by using the sender and receiver cells made from quorum sensing mechanism.

2.2 Implementing finite states by toggle switch

In order to realize a finite number of states *in vivo*, we employ the toggle switch mechanism proposed by Gardner et al. [2]. The toggle switch is composed of two repressors and two constitutive promoters (See Figure 3 (above)). Each promoter is inhibited by the repressor that is transcribed by the opposing promoter.

The behavior of the toggle switch and the conditions for bistability to represent two states can be formalized using the following equations:

$$\begin{aligned} \frac{dV_{cI857}}{dt} &= \frac{\alpha_1}{1 + V_{lacI}^\beta} - V_{cI857} \\ \frac{dV_{lacI}}{dt} &= \frac{\alpha_2}{1 + V_{cI857}^\gamma} - V_{lacI} \end{aligned}$$

where V_{cI857} is the concentration of repressor gene *cI857*, V_{lacI} is the concentration of repressor gene *lacI*, α_1 is the effective rate of synthesis of repressor *cI857*, α_2 is the effective rate of synthesis of repressor *lacI*, β is the cooperativity of repression of promoter P_{trc} , and γ is the cooperativity of repression of promoter P_{L^*} . The bistability of the toggle switch depends on these parameters.

We term the state where *cI857* is expressed and *lacI* is repressed the state '1' and the state where *cI857* is repressed and *lacI* is expressed the state '0', as shown in Figure 3 (below). The state 1 is switched to state 0 by a thermal pulse of 42°C induction and the state 0 is switched to state 1 by a pulse of IPTG induction, which is a molecular compound that triggers transcription of the lac operon.

Finally, the toggle switches were implemented on *E. coli* plasmid which had originally been made in Kobayashi et al. [4] experiments.

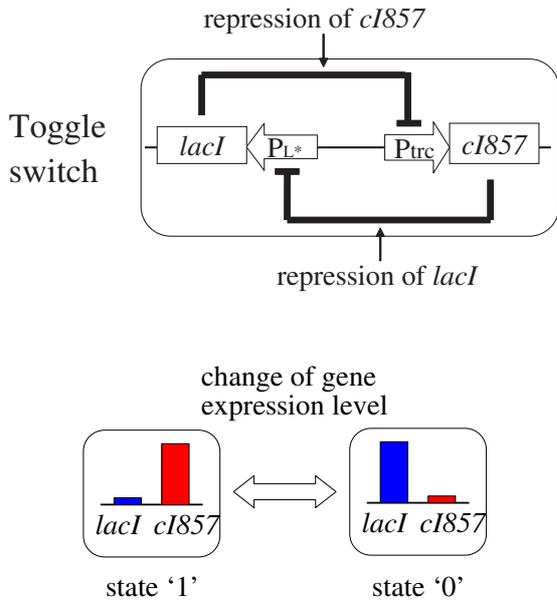


Figure 3: (above) Toggle switch circuit and (below) two states using gene expression levels.

2.3 implementation of in vivo CA

By combining those two mechanisms of quorum-sensing and toggle switch, we have implemented a simple cellular automaton of one dimensional topology.

A cellular automaton is a formal computation model with two specific components [3]. First, a regular, discrete and infinite network, which represents the underlying space structure of the cellular automaton. Second, a finite-state automaton, which consists of a finite set of states and a state-transition function. Each node of the network is called a cell and communicates with a finite number of other cells, which determine its *neighborhood*. This communication, which is local, deterministic, uniform, and synchronous determines a global behavior of the CA along discrete time steps. The following is the formal definition of one-dimensional CA.

(Definition of Cellular Automaton:) One-dimensional cellular automaton (CA, for short) is defined by a five tuple $CA = (\mathbf{Z}, N, Q, \delta, q_0)$, where \mathbf{Z} is the set of integers, N is the neighborhood index $N = \{r_0, r_1, \dots, r_{k-1}\}$ that is a subset of \mathbf{Z} and represents the relative locations of the neighboring cells, Q is a finite set of states, δ is a state-transition function such that $\delta : Q^k \rightarrow Q$ and $s_i(t+1) = \delta(s_{i+r_0}, s_{i+r_1}, \dots, s_{i+r_{k-1}})$ for the state s_i of location i and at time $t+1$, and q_0 is the initial state.

(Example: Rule 250) One-dimensional CA with $\{-1, 1\}$ neighborhood, $\{0, 1\}$ set of states, with the state-transition function of rule 250 is given as follows: $CA = (\mathbf{Z}, \{0, 1, -1\}, \{0, 1\}, \delta, 0)$ and the state-transition function is defined as $\delta(0, 0, 0) = 0$, $\delta(0, 0, 1) = 1$, $\delta(0, 1, 0) = 0$, $\delta(0, 1, 1) = 1$, $\delta(1, 0, 0) = 1$, $\delta(1, 0, 1) = 1$, $\delta(1, 1, 0) = 1$, $\delta(1, 1, 1) = 1$. This state-transition function represents the OR logical function such that if either adjacent cell holds the state 1 at the current step, the own cell changes the state to 1 at the next step.

(Behavior of CA with rule 250) If we consider a CA as a machine which evolves with time, the behavior of the CA

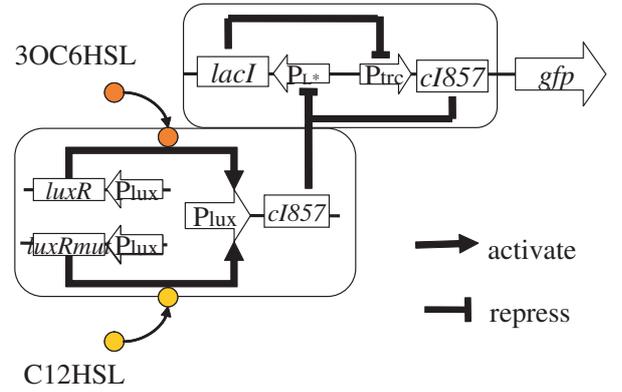


Figure 4: Genetic circuit to represent the state-transition function equivalent to Rule 250 of CA.

is a sequence of configurations, that is, instantaneous description, determined by the given initial configuration and the state-transition function. The following is a behavior of Rule 250 CA with initial configuration “00000100000”:

0 0 0 0 0 1 0 0 0 0 0	time=0
0 0 0 0 1 0 1 0 0 0 0	time=1
0 0 0 1 0 1 0 1 0 0 0	time=2
0 0 1 0 1 0 1 0 1 0 0	time=3
0 1 0 1 0 1 0 1 0 1 0	time=4
1 0 1 0 1 0 1 0 1 0 1	time=5

The state-transition function is implemented by using the genetic circuit shown in Figure 4. This genetic circuit combines the toggle switch and the “OR” logic gate that corresponds to the rule 250 of CA. This OR logic gate is implemented by *luxR* gene and its mutant *luxR-G2E-R67M* (*luxRmut*, for short). The protein LuxR binds to a signal chemical 3OC6HSL (*N*-(β -ketocaproyl)-L-homoserine lactone) and LuxRmut binds to a signal C12HSL (*N*-Dodecanoyl-DL-homoserine lactone) [1], and both activate the *lux* promoter of the gene *cI857* (denoted “Plux” in Figure 4). Therefore, this OR logic gate receives either 3OC6HSL signal or C12HSL signal, and activates the gene *cI857* expression if either signal or both are received, that is, true “OR logical function”. We virtually assume that 3OC6HSL signal is sent from the right-hand adjacent cell and C12HSL is sent from the left-hand cell in the topology of one dimensional CA.

3. EXPERIMENTAL RESULT OF ONE-STEP STATE TRANSITION OF IN VIVO CA

We have done a laboratory experiment to confirm one-step state-transition of using this genetic circuit. First, we set the state of toggle switch to 0 by a thermal pulse of 42°C. Second, we input either 3OC6HSL signal or C12HSL signal to the genetic circuit. If the state-transition function can correctly work, the expression of *cI857* represses *lacI* and successively the repression of the promoter P_{trc} is released and finally the downstream reporter gene *gfp* is expressed.

The result is shown in Figure 5. In the first 12 hours, since the toggle switch is set to 0 by a thermal pulse of 42°C, the expression level of GFP fluorescence is low. In the second 12 hours, since 3OC6HSL or C12HSL is input to the genetic circuit, the state is changed from 0 to 1 and hence the

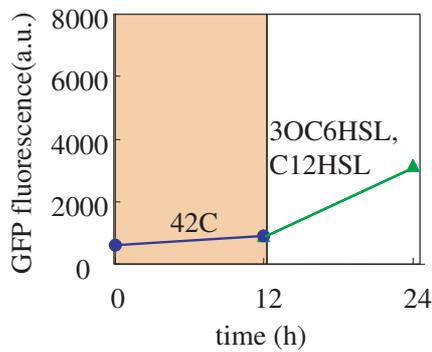


Figure 5: Experiment of one-step state-transition from state ‘0’ to state ‘1’ by input signal. The change of the state is observed by expression level of GFP fluorescence.

fluorescence level becomes higher. Therefore, we conclude that our implementation succeeded to correctly implement *in vivo* CA of Rule 250 on *E.coli*.

4. ACKNOWLEDGMENTS

This work is supported in part by Grant program for bioinformatics research and development of Japan Science and Technology Agency, Grant-in-Aid for Scientific Research on Priority Area No. 17018029. We would like to thank Dr. Frances Arnold (Caltech) and Dr. Hideki Kobayashi (Jamstec) for their kind supplies of their plasmids.

5. REFERENCES

- [1] C. H. Collins, J. R. Leadbetter, and F. H. Arnold. Dual selection enhances the signaling specificity of a variant of the quorum-sensing transcriptional activator LuxR. *Nat. Biotechnol.*, 24(6):708–712, 2006.
- [2] T. S. Gardner, C. R. Cantor, and J. J. Collins. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 403:339–342, January 2000.
- [3] A. Ilachinski. *Cellular Automata: A Discrete Universe*. World Scientific Publishing, 2001.
- [4] H. Kobayashi, M. Kaern, M. Araki, K. Chung, T. S. Gardner, C. R. Cantor, and J. J. Collins. Programmable cells: interfacing natural and engineered gene networks. *Proc. Natl. Acad. Sci.*, 101(22):8414–8419, 2004.
- [5] H. Nakagawa, K. Sakamoto, and Y. Sakakibara. Development of an *in vivo* computer based on *Escherichia coli*. In *Proceedings of 11th International Meeting on DNA Based Computers*, pages 68–77. London, Ontario, June 2005.
- [6] N. A. Whitehead, A. M. Barnard, H. Slater, N. J. Simpson, and G. P. Salmond. Quorum-sensing in gram-negative bacteria. *FEMS Microbiology Reviews*, 25:365–404, 2001.