Bio-Molecular Computing of Finite-State Machine

[Extended Abstract]

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

ABSTRACT

We overview a series of our research on implementing finite automata *in vitro* and *in vivo* in the framework of DNA-based computing [2, 3]. First, we employ the length-encoding technique proposed and presented in [5, 4] to implement finite automata in test tube. In the length-encoding method, the states and state transition functions of a target finite automaton are effectively encoded into DNA sequences, a computation (accepting) process of finite automata is accomplished by self-assembly of encoded complementary DNA strands, and the acceptance of an input string is determined by the detection of a completely hybridized double-strand DNA. Second, We report our intensive *in vitro* experiments in which we have implemented and executed several finite-state automata in test tube. We have designed and developed practical laboratory protocols which combine several *in vitro* operations such as annealing, ligation, PCR, and streptavidin-biotin bonding to execute *in vitro* finite automata based on the length-encoding technique. We have carried laboratory experiments on various finite automata of from 2 states to 6 states for several input strings. Third, we present a novel framework to develop a programmable and autonomous *in vivo* computer using *Escherichia coli (E. coli)*, and implement *in vivo* finite-state automata based on the framework by employing the protein-synthesis mechanism of *E. coli*. Our fundamental idea to develop a programmable and autonomous finite-state automata on *E. coli* is that we first encode an input string into one plasmid, encode state-transition functions into the other plasmid, and introduce those two plasmids into an *E. coli* cell by electroporation. Fourth, we execute a protein-synthesis process in *E. coli* combined with four-base codon techniques to simulate a computation (accepting) process of finite automata, which has been proposed for *in vitro* translation-based computations in [4]. This approach enables us to develop a programmable *in vivo* computer by simply replacing a plasmid encoding a state-transition function with others. Further, our *in vivo* finite automata are autonomous because the protein-synthesis process is autonomously executed in the living *E. coli* cell. We show some successful experiments to run an *in vivo* finite-state automaton on *E. coli*.

Bionetics '08, November 25-28, 2008, Hyogo, Japan

Keywords

DNA computing, molecular computing, finite-state automata.

1. INTRODUCTION

Biological molecules such as DNA, RNA and proteins are natural devices to store information, activate (chemical) functions and communicate between systems (such as cells). DNA computer study utilizes these biological devices to make a computer. One of the ultimate goals is to make an autonomous cell-based turing machine and apply to genetic and life engineering. Our attempts to make a bacteria-based computer make progress toward this goal.

The finite-state automata (machines) are the most basic computational model in Chomsky hierarchy and are the start point to build universal DNA computers. Several works have attempted to develop finite automata *in vitro*. However, there have been no experimental research works which attempt to build a finite automaton *in vivo*. Benenson et al. [1] have successfully implemented the two state finite automata by the sophisticated use of the restriction enzyme (actually, *Fok*I) which cut outside of its recognition site in a double-stranded DNA. However, their method has some limitations for extending to more than 2 states. Yokomori et al. [5] have proposed a theoretical framework using length-encoding technique to implement finite automata on DNA molecules. Theoretically, the length-encoding technique has no limitations to implement finite automata of any larger states.

In our first research work [2], we have attempted to implement and execute finite automata of a larger number of states *in vitro*, and carry intensive laboratory experiments on various finite automata of from 2 states to 6 states for several input strings.

On the other hand, in our next research work [3], we have previously proposed a method using the protein-synthesis mechanism combined with four-base codon techniques to simulate a computation (accepting) process of finite automata *in vitro* [4] (a codon is normally a triplet of base, and different base triplets encode different amino acids in protein). The proposed method is quite promising and has several advanced features such as the protein-synthesis process is very accurate and overcomes mis-hybridization problem in the self-assembly computation and further offers an autonomous

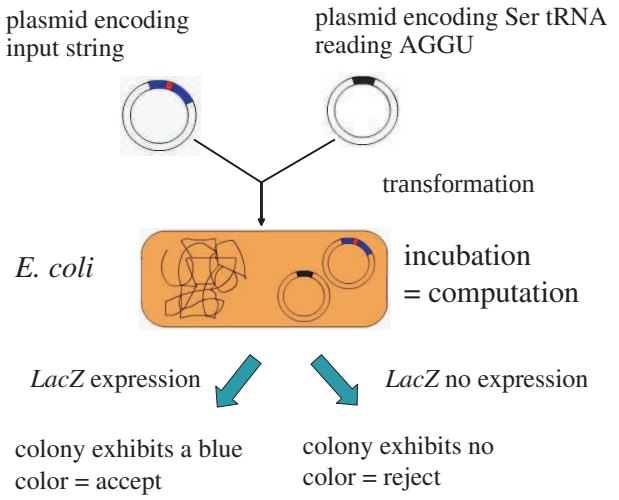

Figure 1: The framework of our *in vivo* computer system based on *E. coli*.

computation. Our aim was to extend this novel principle into a living system, by employing the *in vivo* protein-synthesis mechanism of *Escherichia coli (E. coli)*. (Escherichia coli is a typical bacteria living inside our body, large intestine.) This *in vivo* computation possesses the following two novel features, not found in any previous biomolecular computer. First, an *in vivo* finite automaton is implemented in a living *E. coli* cell; it does not mean that it is executed simply by an incubation at a certain temperature. Second, this automaton increases in number very rapidly according to the bacterial growth; one bacterial cell can multiply to over a million cells overnight. The present study explores the feasibility of *in vivo* computation.

The main feature of our *in vivo* computer based on *E. coli* is that we first encode an input string into one plasmid, encode state-transition functions into the other plasmid, and transform *E. coli* cells with these two plasmids by electroporation. Second, we execute a protein-synthesis process in *E. coli* combined with four-base codon techniques to simulate a computation (accepting) process of finite automata, which has been proposed for *in vitro* translation-based computations in [4]. The successful computations are detected by observing the expressions of a reporter gene linked to mRNA encoding an input data. Therefore, when an encoded finite automaton accepts an encoded input string, the reporter gene, *lacZ*, is expressed and hence we observe a blue color. When the automaton rejects the input string, the reporter gene is not expressed and hence we observe no blue color. Our *in vivo* computer system based on *E. coli* is illustrated in Fig. 1.

Thus, our *E. coli*-based computer enables us to develop a programmable and autonomous computer. To our knowledge, this is the first experimental development of *in vivo* computer and has succeeded to execute an finite-state automaton on *E. coli*.

2. REFERENCES

[1] Y. Benenson, T. Paz-Ellzur, R. Adar, E. Keinan, Z. Livneh, and E. Shapiro. Programmable and autonomous computing machine made of biomolecules. *Nature*, 414:430–434, 2001.

[2] J. Kuramochi and Y. Sakakibara. Intensive in vitro experiments of implementing and executing finite automata in test tube. In *Proceedings of 11th International Meeting on DNA Based Computers*, pages 59–67. London, Ontario, 2005.

[3] 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, 2005.

[4] Y. Sakakibara and T. Hohsaka. In vitro translation-based computations. In *Proceedings of 9th International Meeting on DNA Based Computers*, pages 175–179. Madison, Wisconsin, 2003.

[5] T. Yokomori, Y. Sakakibara, and S. Kobayashi. A magic pot : Self-assembly computation revisited. In *Formal and Natural Computing (LNCS 2300)*, pages 418–429. Springer-Verlag, 2002.