Vol.2011-MPS-84 No.3
2011/7/18

DNA Logic Circuits with a DNA Polymerase and a Nicking Enzyme

Ryo Hirose,†1 Satoshi Kobayashi†1 and Ken Komiya†2

The current most popular and established approach to DNA logic circuits is the implementation by DNA strands reaction networks where toehold-mediated strand displacement invokes and performs the evaluation of each DNA logic gate. Strand displacement approach requires, however, a large amount of time, for instance, approximately 30 minutes, to execute a logic operation[18]. Furthermore, the concentration of an output molecule released from a gate can not exceed that of the gate molecule. Therefore, it is often the case that large quantities of input and gate molecules are required when the gate is of large out-degree. In order to overcome these problems, it is indispensable to devise a DNA logic gate which runs quickly and can amplify the quantity of the output molecule. We will propose a DNA implementation of logic gates with such good properties using DNA polymerase and nicking enzyme.

1. Introduction

In 1994, Adleman initiated the paradigm of DNA computing by devising and demonstrating a biological experimental protocol to solve Directed Hamiltonian Path Problem[1]. It has achieved important progress of the technologies in making DNAs work as computing devices[4][3][2]. They also produce some new and important key technologies[14][10] in the field of DNA nanotechnology, where it is aimed to construct intended nano-scale shapes or structures by self-assembly of DNA molecules. In these fields, it is currently emerging the movement for establishing the methodology to construct automatic molecular robots performing intended tasks in some specified environment[8][5]. A molecular robot should, however, contain at least three important components, a sensor, a circuit, and an actuator, in itself. The current technology[8][5] does not satisfy these three requirements, and it is still at a premature stage. So, it is very challenging to explore a possible framework of the methodology to construct a molecular robot with a sensor, a circuit, and an actuator.

Molecular circuits equipped in molecular robots should serve as logic circuits, memory devices, and control devices, etc[12]. In this paper, we will focus on a molecular circuit as a computing device performing logic operations. There are many works which have proposed DNA implementations of logic circuits. One of the most popular and established approach to DNA logic circuits is the implementation by DNA strands reaction networks where toehold-mediated strand displacement[16], strand displacement for short, invokes and performs the evaluation of each DNA logic gate[17][15][18]. Strand displacement approach requires, however, a large amount of time, for instance, approximately 30 minutes, to execute a logic operation[18]. Furthermore, the concentration of an output molecule released from a gate can not exceed that of the gate molecule. Therefore, it is often the case that large quantities of input and gate molecules are required when the gate is of large out-degree. In order to overcome these problems, it is indispensable to devise a DNA logic gate which runs quickly and can amplify the quantity of the output molecule. We will propose a DNA implementation of logic gates with such good properties using a DNA polymerase and a nicking enzyme.

In the rest of this paper, for a DNA strand X, by X^*, we denote the complementary strand of X.

2. Related Work

The current most promising approach to the construction of DNA logic gates is the DNA implementation of logic operation by the use of strand displacement. Its basic mechanism is illustrated in Figure 1. Consider a complex consisting of strands $5'-A-B-3'$ and $5'-C^*-B^*-3'$ with B and B^* hybridized, where A, B, C are sequences of bases. If another strand $5'-B-C-3'$ exists in a solution, then $5'-B-C-3'$ hybridizes to the complex with C and C^* hybridized. The strand $5'-B-C-3'$ gradually extends its pairing with $5'-C^*-B^*-3'$ by replacing the strand $5'-A-B-3'$ in a random walk fashion. This process is called "branch migration".

There have been many research works to propose bio-lab methods to imple-

†1 University of Electro-Communications
†2 Tokyo Institute of Technology

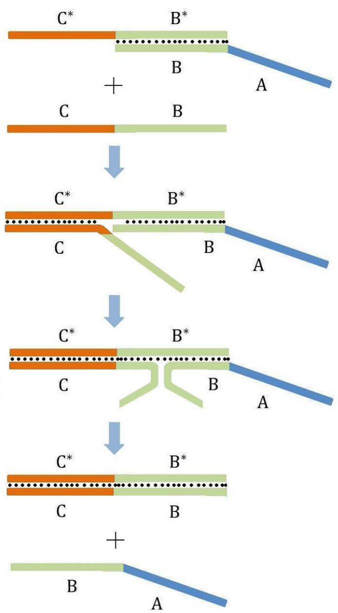

Fig. 1 Toehold-mediated Strand Displacement

ment some computational tasks by the use of strand displacement[11)17)15)18)], and it is considered as one of the most promising bio-lab techniques in DNA computing. The computational capability of strand displacement is also an interesting research topic from the viewpoint of computation theory[12)9)].

Strand displacement is, however, a slow reaction if we want to use it as an essential bio-lab operation to implement DNA logic circuits of molecular robots. Furthermore, in these approaches, the concentration of an output molecule released from a gate can not exceed that of the gate molecule. In this sense, the construction of large circuits based only on strand displacement is not feasible.

In the next section, we will propose a new bio-lab method for constructing logic circuits with DNA molecules, where we use the amplification system based on nicking enzyme and DNA polymerase. Waker first applied this reaction to the construction of isothermal amplification system of DNA[13)]. The idea of using nicking enzyme to computational molecular devices was also accomplished by Matsuda and Yamamura[6)], where they used a DNA polymerase and a nicking enzyme in order to cascade molecular state transition systems based on Whiplash PCR[3)]. Very recently, Montagne, et al., proposed to use a DNA polymerase and a nicking enzyme to construct a programmed DNA oscillator[7)]. In this paper, we will apply a similar reaction system to the construction of DNA logic circuits, and we will show by computational simulations that the proposed system is time-efficient. Furthermore, in the proposed system, the output of each DNA logic gate can be amplified by the use of a DNA polymerase and a nicking enzyme.

3. Construction of Combinatorial Circuit

In this section, we will propose a method to construct combinatorial circuits by using DNA polymerase and nicking enzyme. As in other related works about DNA logic gates, we will prepare for each boolean variable x, a DNA strand X. The existence of X in the solution implies the evaluation of the boolean variable x as 1. On the other hand, how can we encode the evaluation $x = 0$ into the solution? For this purpose, we will also prepare another DNA strand NX, and we regard the existence of NX in the solution as the evaluation of the variable x as 0. Thus, the strand X and NX can not exist at the same time in the solution. Although most of the other works encode the evaluation $x = 0$ as the nonexistence of X in the solution, we will use the negation strand NX in order to implement NOT gate simply.

In order to implement DNA logic circuits, we will use the following three basic *abstract-level* chemical reactions:

1. AND reaction: $\mathrm{A} \wedge \mathrm{B} \rightarrow \mathrm{C}$ — the strand C is produced if and only if both of the strands A and B exist in the solution.
2. OR reaction — $\mathrm{A} \mid \mathrm{B} \rightarrow \mathrm{C}$ — the strand C is produced if and only if either of the strands A or B exists in the solution.
3. PROPAGATE (PROP) reaction : $\mathrm{A} \rightarrow \mathrm{B}$ — the strand B is produced if and only if the strand A exists in the solution.

When we want to construct a chemical reaction system which utilizes any given boolean function, it suffices to devise chemical implementation of logic gates, AND, OR, and NOT. Then, it is easy to construct AND, OR, and NOT gates using the three basic reactions above.

1. AND gate construction — Consider an AND gate with input variables a and b and with an output variable c. Then, we will prepare the strands A, NA for the variable a, B, NB for the variable b, C, NC for the variable c. It is easy to see that the AND and OR reactions, $\mathrm{A} \wedge \mathrm{B} \rightarrow \mathrm{C}$ and $\mathrm{NA} \mid \mathrm{NB} \rightarrow \mathrm{NC}$, implement the AND gate.
2. OR gate construction — Consider an OR gate with input variables a and b and with an output variable c. Then, we will prepare the strands A, NA for the variable a, B, NB for the variable b, C, NC for the variable c. It is easy to see that the AND and OR reactions, $\mathrm{A} \mid \mathrm{B} \rightarrow \mathrm{C}$ and $\mathrm{NA} \wedge \mathrm{NB} \rightarrow \mathrm{NC}$, implement the OR gate.
3. NOT gate construction — Consider a NOT gate with an input variable a and with an output variable b. Then, we will prepare the strands A, NA for the variable a, B, NB for the variable b. It is easy to see that the PROP reactions, $\mathrm{A} \rightarrow \mathrm{NB}$ and $\mathrm{NA} \rightarrow \mathrm{B}$, implement the NOT gate.

In the rest of this section, we will propose a method to implement AND, OR and PROP reactions with the use of DNA polymerase and nicking enzyme. We assume that we will use a nicking enzyme which recognizes a double stranded DNA sequences $\begin{matrix} 5'-R-3' \\ 3'-R^*-5' \end{matrix}$, and cleaves the 3' end of the lower strand R* only.

Because of space constraint, we will only explain the construction of AND reaction which, with the existence of sufficient amounts of input DNA single strands A and B in a solution, outputs a single stranded DNA sequence C. Principal molecule of this reaction, called *AND complex*, consists of two DNA strands $5'-\mathrm{C}^*-\mathrm{R}-\mathrm{B}^*-3'$ and $5'-\mathrm{B}-\mathrm{R}^*-\mathrm{A}^*-3'$ with B and B* hybridized at its initial state (Fig.2(1)). Let us consider the situation where sufficient amounts of DNA strands A and B exist in a solution. At first, A hybridizes to A* of the AND complex (Fig.2(2)). Then, DNA polymerase bind to this double stranded part (consisting of A and A*) and elongates the sequence A until it reaches to the 5'-end of the sequence $5'-\mathrm{B}-\mathrm{R}^*-\mathrm{A}^*-3'$ (Fig.2(3)). This makes the double stranded part (consisting of B and B*) detached from the AND complex and we will have DNA strand $5'-\mathrm{C}^*-\mathrm{R}-\mathrm{B}^*-3'$ in the solution (Fig.2(4)). Then, the strand B hybridizes to B* of $5'-\mathrm{C}^*-\mathrm{R}-\mathrm{B}^*-3'$ (Fig.2(5)), and DNA polymerase binds to this double stranded part (consisting of B and B*) and elongates the sequence B until it reaches to the 5'-end of the sequence $5'-\mathrm{C}^*-\mathrm{R}-\mathrm{B}^*-3'$ (Fig.2(6)). This complex is recognized by a nicking enzyme at the site R and R*, and it cleaves the strand R* (Fig.2(8)) . DNA polymerase, while pushing the strand C away, elongates the strand $5'-\mathrm{B}-\mathrm{R}^*-3'$ again until it reaches to the 5'-end of the sequence $5'-\mathrm{C}^*-\mathrm{R}-\mathrm{B}^*-3'$ and the output strand C is released. Repeating the process (8), (9) and (10), the output strand C is amplified and released.

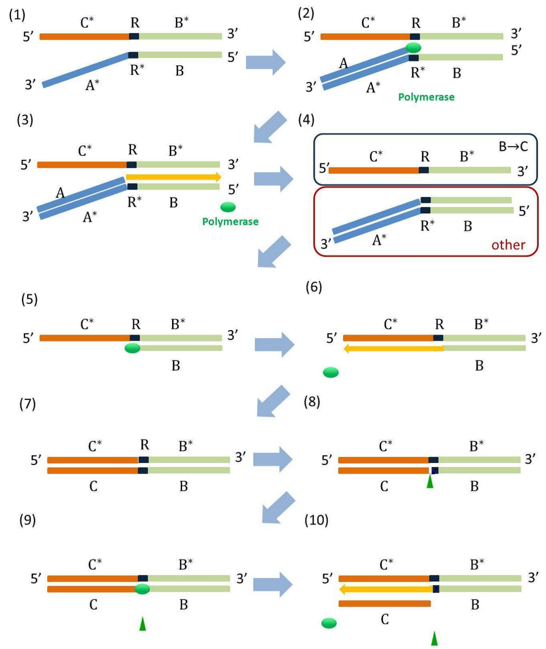

Fig. 2 AND Reaction

4. Mathematical Model of Chemical Reaction Networks

In this section, we will explain the mathematical model of chemical reaction networks for the proposed combinatorial DNA circuits. We only give the explanation for the case of AND reaction, since the models of other basic operations, OR and PROP, are almost similar to that of AND reaction and can be obtained easily.

4.1 Chemical Reaction Network of AND Reaction

For any rate constant k, by k' we denote the rate constant of the reverse reaction of the reaction corresponding to k. Parameters k^A_{poly}, k^B_{poly} and k^C_{poly} are rate constants of polymerase reaction elongating the strands, A, B, and C,respectively. The parameter k_{nick} is the rate constant for nicking enzyme to cleave strands.

$$
\begin{array}{lcl@{\qquad}lcl}
\mathrm{G} + \mathrm{A} & \underset{k'_A}{\overset{k_A}{\rightleftarrows}} & \mathrm{G}\cdot\mathrm{A} & \mathrm{G}\cdot\mathrm{A} & \overset{k^A_{poly}}{\rightarrow} & \mathrm{G_1} + \mathrm{G_2}\cdot\mathrm{A} \\
\mathrm{G_1} + \mathrm{B} & \underset{k'_B}{\overset{k_B}{\rightleftarrows}} & \mathrm{G_1}\cdot\mathrm{B} & \mathrm{G_1}\cdot\mathrm{B} & \overset{k^B_{poly}}{\rightarrow} & \mathrm{G_1}\cdot\mathrm{BC} \\
\mathrm{G_1}\cdot\mathrm{BC} & \overset{k_{nick}}{\rightarrow} & \mathrm{G_1}\cdot\mathrm{B}\cdot\mathrm{C} & \mathrm{G_1}\cdot\mathrm{B}\cdot\mathrm{C} & \overset{k^C_{poly}}{\rightarrow} & \mathrm{G_1}\cdot\mathrm{BC} + \mathrm{C}
\end{array}
$$

Fig. 3 Full Reaction Network of AND Gate

Chemical reaction network of an *abstract-level* AND reaction is described in Figure 3, where AND complex, its upper strand, and its lower strand are represented as G, $\mathrm{G_1}$, and $\mathrm{G_2}$, respectively.

4.2 Differential Equations of AND Reaction

Differential equation system to mathematically model and simulate the AND reaction is given in Figure 4.

5. Simulation Results

We used the following kinetic parameters for the simulation, which are determined by referring to the data reported in Mongagne's paper[7].

$$
\begin{array}{ll}
k_A = k_B = 2.4 \times 10^7 & (\mathrm{M^{-1}min^{-1}}) \\
k'_A = k'_B = 1.2 \times 10^{-3} & (\mathrm{M^{-1}min^{-1}}) \\
k^B_{Poly} = 2.4 \times 10^1 & (\mathrm{min^{-1}}) \\
k^A_{Poly} = k^C_{Poly} = 2.4 \times 10^1 \times 0.8 & (\mathrm{min^{-1}}) \\
k_{nick} = 6.0 \times 10^0 & (\mathrm{min^{-1}})
\end{array}
$$

We applied the proposed method to the construction of a majority vote circuit.

$$
\begin{aligned}
\frac{d[\mathrm{A}]}{dt} &= -k_A[\mathrm{G}][\mathrm{A}] + k'_A[\mathrm{G}\cdot\mathrm{A}] \\
\frac{d[\mathrm{B}]}{dt} &= -k_B[\mathrm{G_1}][\mathrm{B}] + k'_B[\mathrm{G_1}\cdot\mathrm{B}] \\
\frac{d[\mathrm{C}]}{dt} &= k^C_{poly}[\mathrm{G_1}\cdot\mathrm{B}\cdot\mathrm{C}] \\
\frac{d[\mathrm{G}]}{dt} &= -k_A[\mathrm{G}][\mathrm{A}] + k'_A[\mathrm{G}\cdot\mathrm{A}] \\
\frac{d[\mathrm{G}\cdot\mathrm{A}]}{dt} &= k_A[\mathrm{G}][\mathrm{A}] - k'_A[\mathrm{G}\cdot\mathrm{A}] - k^A_{poly}[\mathrm{G}\cdot\mathrm{A}] \\
\frac{d[\mathrm{G_1}]}{dt} &= k^A_{poly}[\mathrm{G}\cdot\mathrm{A}] - k_B[\mathrm{G_1}][\mathrm{B}] + k'_B[\mathrm{G_1}\cdot\mathrm{B}] \\
\frac{d[\mathrm{G_2}\cdot\mathrm{A}]}{dt} &= k^A_{poly}[\mathrm{G}\cdot\mathrm{A}] \\
\frac{d[\mathrm{G_1}\cdot\mathrm{B}]}{dt} &= k_B[\mathrm{G_1}][\mathrm{B}] - k'_B[\mathrm{G_1}\cdot\mathrm{B}] - k^B_{poly}[\mathrm{G_1}\cdot\mathrm{B}] \\
\frac{d[\mathrm{G_1}\cdot\mathrm{BC}]}{dt} &= k^B_{poly}[\mathrm{G_1}\cdot\mathrm{B}] - k_{nick}[\mathrm{G_1}\cdot\mathrm{BC}] + \\
&\quad + k^C_{poly}[\mathrm{G_1}\cdot\mathrm{B}\cdot\mathrm{C}] - k'^C_{poly}[\mathrm{G_1}\cdot\mathrm{BC}][\mathrm{C}] \\
\frac{d[\mathrm{G_1}\cdot\mathrm{B}\cdot\mathrm{C}]}{dt} &= k_{nick}[\mathrm{G_1}\cdot\mathrm{BC}] - k^C_{poly}[\mathrm{G_1}\cdot\mathrm{B}\cdot\mathrm{C}]
\end{aligned}
$$

Fig. 4 Full system of differential equations for AND reaction

The circuit outputs 1 if the number of input variables assigned to 1 is greater than that of input variables assigned to 0. We applied the method to 3-variable case. The circuit is given in Figure 5. For each variable, x, y, z, a, b, c, d, e, we will prepare the DNA strands X, Y, Z, A, B, C, D, E, for representing the evaluation of each variable to 1. Furthermore, the DNA strands NX, NY, NZ, NA, NB, NC, ND, NE are used for representing the evaluation of each variable to 0.

We set the concentration of each input strand X, Y, Z, NX, NY, NZ, to 1.0×10^{-10} (M) if it should exist in the solution, to 0 (M) otherwise. Furthermore, the concentration of the AND-complex and OR-complex at the 1st, 2nd, and 3rd layers are set to 1.0×10^{-9} (M), 1.0×10^{-7} (M), and $5.0^{\times}10^{-8}$ (M), respectively.

Figure 6 gives the simulation result for the case of inputs $x = y = z = 1$. That is, we put 1.0×10^{-10} (M) of X, Y, Z, and 0 (M) of NX, NY, NZ, in the solution. As expected, the concentration of the output strand E grows rapidly, and NE stays at 0 for all the time of the simulation, which correctly simulates the behavior of the majority vote circuit.

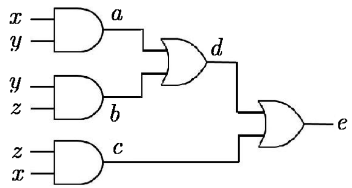

Fig. 5 Majority Vote Circuit

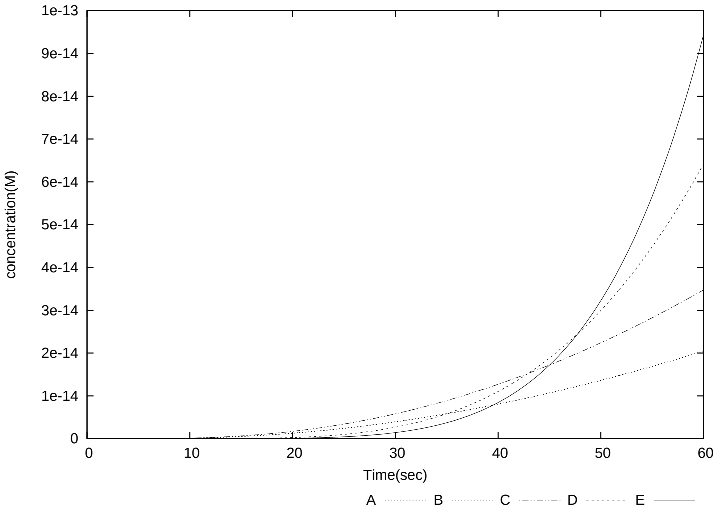

Fig. 6 Simulation of Majority Vote Circuit (1)

For the case of inputs $x = y = z = 0$. That is, we put 1.0×10^{-10} (M) of NX, NY, NZ, and 0 (M) of X, Y, Z, in the solution. As expected, the concentration of the output strand NE grows rapidly, and E stays at 0 for all the time of the simulation, which correctly simulates the behavior of the majority vote circuit (Fig.7). Note that in Fig. 7, the concentrations of NA and NC are very small, so they almost stay at 0 in the graph.

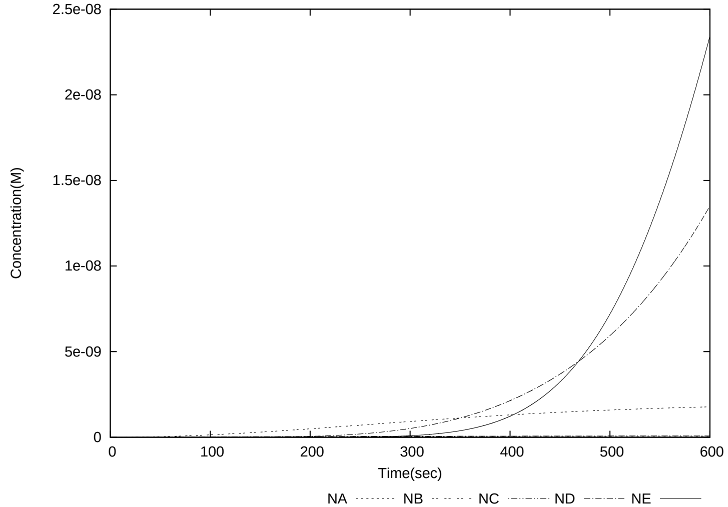

Fig. 7 Simulation of Majority Vote Circuit (2)

Figure 8 gives the global view of the simulation result for the case of inputs $x = y = z = 1$. We can verify that the growth of the 3rd layer output is very rapid (at most 10 minutes are enough for 3 steps of logic operation).

6. Conclusions

We proposed a new bio-lab method for constructing logic circuits with DNA molecules, where we use the amplification system based on a nicking enzyme and a DNA polymerase similar to the work by Walker[13], Matuda[6], and Montagne[7]. The proposed method has some good properties as computational devices that it is time-efficient and that the output of each DNA logic gate can be amplified, and so, the scalability and the feasibility of the proposed system is better than the previous methods.

However, in order to increase the amount of the output molecule, the con-

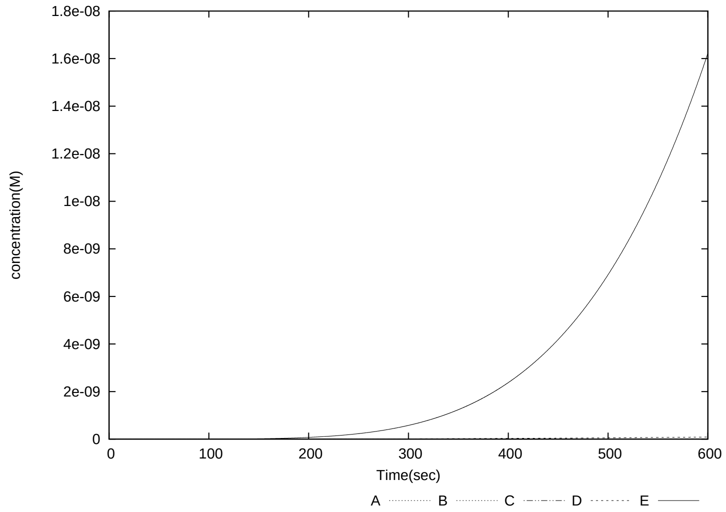

Fig. 8 Simulation of Majority Vote Circuit (3)

centration of each gate complex should be carefully designed. Furthermore, the proposed method has a problem that the amplification of the output strands continues until DNA polymerase uses up substrates. So, we need a bio-lab method for stopping or inhibiting the amplification process in this framework. An idea is to use inhibitor strands and exonuclease as in the work by Montagne[7]. All of these issues are our future research topics.

Acknowledgments The second author is supported in part by Grant-in-Aid for Scientific Research (C) No.22500010, Japan Society for the Promotion of Science.

References

[1] L. Adleman, Molecular Computation of Solutions to Combinatorial Problems. *Science* **266**, pp.1021-1024, 1994.

[2] Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, E. Shapiro, Programmable and autonomous computingmachine made of biomolecules, *Nature* **414**, pp.430-434, 2001.

[3] M.Hagiya, M.Arita, D.Kiga, K.Sakamoto, S.Yokoyama, Towards parallel evaluation of Boolean μ-formulas with molecules, In *DNA Based Computers III*(American Mathematical Society), pp.57-72, 1999.

[4] R.Lipton, DNA solution of hard computational problem, *Science* **268**, pp.542-545, 1995.

[5] D. Zhang, E. Winfree, K. Lund, A. Manzo, N. Dabby, N. Michelotti, A. Johnson-Buck, J.Nangreave, S.Taylor, R.Pei, M.Stojanovic, N.Walter, E.Winfree, H.Yan, Molecular robots guided by prescriptive landscapes, *Nature*, **465**, pp.206-210, 2010.

[6] D. Matsuda, M. Yamamura, Cascading Whiplash PCR with a Nicking Enzyme, DNA Computing — Proc. of International Workshop on DNA-Based Computers, DNA8 (2002), Lecture Notes in Computer Science, pp.38-46, 2003.

[7] K.Montagne, R.Plasson, Y.Sakai, T.Fujii, Y.Rondelez, Programming an *in vitro* RNA oscillator using a molecular networking strategy, *Molecular Systems Biology*, **7**, Article Number 466, 2011.

[8] T.Omabegho, R.Sha, N.Seeman, A Bipedal DNA Brownian Motor with Coordinated Legs, *Science* **324**, pp.67-71, 2009.

[9] L. Qian, D. Soloveichik, E. Winfree, Efficient Turing-universal computation with DNA polymers, *Lecture Notes in Computer Science* **6518**, pp.123-140, 2011.

[10] P.Rothemund, Folding DNA to create nanoscale shapes and patterns, *Nature* **440**, pp.297-302, 2006.

[11] G.Seelig, D.Soloveichik, D.Zhang, E.Winfree, Enzyme-free nucleic acid logic circuits, *Science*, **314**, pp.1585-1588, 2006.

[12] D.Soloveichik, G.Seelig, E.Winfree, DNA as a Universal Substrate for Chemical Kinetics, *Proc. Natl. Acad. Sci. USA*, **107**, pp.5393-5398, 2010.

[13] G.Walker, M.Little, J.Nadeau, D.Shank, Isothermal *in vitro* amplification of DNA by a restriction enzyme/DNA polymerase system, *Proc. Natl. Acad. Sci. USA*, **89**, pp.392-396, 1992.

[14] E. Winfree, F. Liu, L. Wenzler, N. Seeman, Design and self-assembly of two-dimensional DNA crystals, *Nature* **394**, pp.539-544, 1998.

[15] P. Yin, H. Choi, C. Calvert, N. Pierce, Programming biomolecular self-assembly pathways, *Nature*, **451**, pp.318-322, 2008.

[16] B. Yurke, A.J. Turberfield, A.P. Mills, F.C. Simmel, J. Neumann, A DNA-fuelled molecular machine made of DNA. Nature **406**, pp.605-608, 2000.

[17] D.Zhang, A.Turberfield, B.Yurke, E.Winfree, Engineering Entropy-Driven Reactions and Networks Catalyzed by DNA, *Science*, **318**, pp.1121-1125, 2007.

[18] D.Zhang, E.Winfree, Control of DNA Strand Displacement Kinetics Using Toehold Exchange, *J. of American Chemical Society*, **131**, pp.17303-17314, 2009.