Automatic Design of DNA Logic Gates Based on Kinetic Simulation

Ibuki Kawamata, Fumiaki Tanaka, and Masami Hagiya

Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
ibuki@is.s.u-tokyo.ac.jp, fumiaki@dna-comp.org, hagiya@is.s.u-tokyo.ac.jp

Abstract. Recently, DNA logic gates and DNA machines have been developed using only a simple complementary base pairing of DNA, that is, hybridization and branch migration. Because such reaction systems have been designed by trial and error, it has been difficult to design a complex system and to correctly verify the reaction. The purpose of this research is to develop a method for automatically searching and designing DNA logic gates based on a kinetic simulation. Since the solution space that should be searched is quite large, a simulated-annealing method is used to search for a highly evaluated system from many candidates and find a semi-optimal one. A simulator based on a kinetic model is developed, which calculates the time change of concentrations of abstracted DNA molecules. An evaluation function, in which the evaluation value rises when the logic gate works correctly, is also designed. The effectiveness of the proposed method is evaluated experimentally with an AND gate, which is designed automatically.

1 Introduction

In recent years, molecular computing has become an important aspect of nanotechnology. DNA devices, such as enzyme-free DNA logic gates [1] and entropy-driven reactions [2] have been developed. A reaction graph [3] assists in the designing of DNA-oriented systems by representation of assembly pathways. To synthesize a larger-scale logic gate, a simple DNA gate motif, called a seesaw gate, was designed [4]. Since these devices were often designed by trial and error, a great deal of effort had been required to design such systems.

The main purpose of this work is to develop a method for designing a DNA logic gate system using computer simulation. To give one solution to this problem, we developed a searching method for automatically designing a DNA logic gate using a kinetic simulation. The searching method is an algorithm that increases the evaluation value, which indicates how correctly the gate works. This evaluation value is calculated using a simulator that estimates the time change of the concentration based on kinetics. Various logic gates limited to two inputs and one output were designed using our method. We conducted an experiment with an AND gate, and showed that our method works in that case.

R. Deaton and A. Suyama (Eds.): DNA 15, LNCS 5877, pp. 88–96, 2009.

2 Methods

Figure 1 shows the flow chart of the automatic design of a DNA logic gate system. We regard this design of a DNA logic gate as a problem of maximizing the evaluation value. Starting from a random initial state, the system increases the evaluation value gradually by exchanging the state with a neighbor state, which is a local change. Though the last state after the iteration may not be the global optimum, it is possible to design a semi-optimal system, which has the highest evaluation value in the iteration, using this algorithm. In this section, a DNA reaction simulator is discussed first. Second, our developed evaluation function using the simulator is explained. The final section describes our method that is used to search for a system with higher evaluation, which is calculated with this evaluation function.

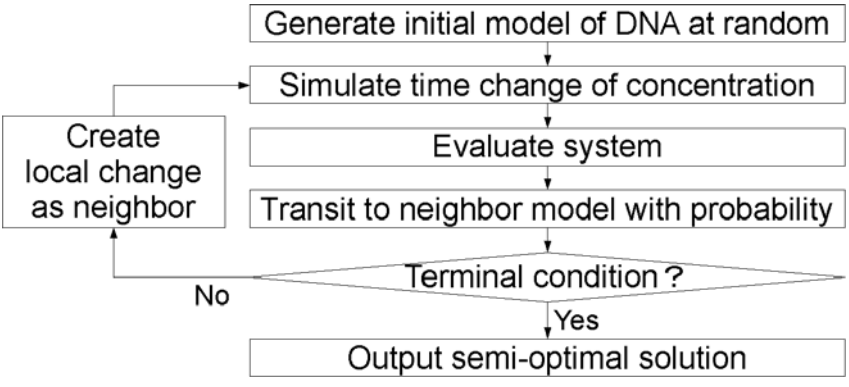

Fig. 1. Flow chart of automatic design of a DNA logic gate system

2.1 Kinetic Simulation

Simulation Model. A simulator is used to calculate the reactions of DNA molecules and the time change of the virtual concentration of each structure with an abstraction and approximation.

Model of DNA. To investigate DNA reactions on the simulator, we developed a computational DNA model. Although a single-stranded DNA molecule is a sequence of bases, the simulator does not recognize each base, and the DNA is abstracted by designating a sequence of bases as a segment. By this abstraction, a single DNA strand can be expressed by a sequence of segments. One letter of the alphabet was allocated to one segment, and uppercase and lowercase letters were used to express complementary base pairs. We treated a DNA structure as a graph, in which the nodes consist of segments and the edges consist of bonds between segments. The upper part of Figure 2 shows the model of an enzyme-free AND gate [1] and the lower part shows our abstraction.

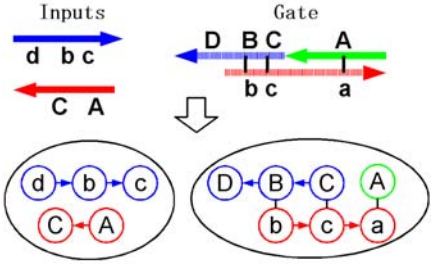

Fig. 2. Example of DNA model (enzyme-free logic gate [1])

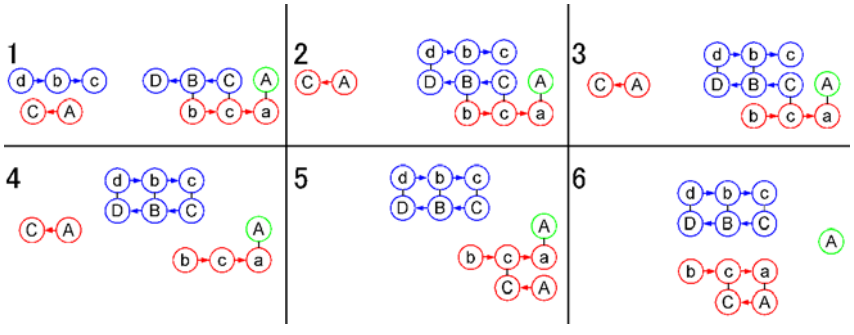

Fig. 3. Example of DNA reactions (enzyme-free logic gate [1])

To design simple DNA systems, only three reactions are simulated as interactions or internal actions of DNA strands, that is to say, hybridization, branch migration, and denaturation. In the simulator, hybridization corresponds to the addition of bond information between unconnected complementary segments, and denaturation corresponds to the opposite, which is the removal of the bond. Branch migration corresponds to exchanging base pairing information of the shorter connection to a new longer connection. Hybridization can be divided into three groups, hybridization of two different structures into one structure (type 1), intramolecular hybridization that extends the double helix by not forming a loop (type 2), and intramolecular hybridization by forming a loop (type 3). Figure 3 shows the series of graph transformations, which correspond to the reactions of the enzyme-free AND gate [1] with both inputs.

Kinetics. Chemical kinetics were used to simulate the time change of the virtual concentration of each DNA structure. For instance, the rate of change for a reaction $A + B \rightarrow C$ is calculated by the equation;

$$\frac{d[C]}{dt} = k[A][B]$$

where k is the rate constant. In the simulator, the rate constant, which is a fixed value that determines the speed of each reaction rate, was not calculated exactly and a virtual value was used. It is considerably difficult to estimate the true value of the three reactions because the rate constant depends on the base sequence. For this reason, we fixed the speeds of hybridization and branch migration at an approximate constant, and only that of denaturation was changeable as a parameter. The rate constant of hybridization type 1 is set to 0.1, type 2 to 0.2, and type 3 to 0.001 and branch migration to 0.005. The simulator distinguishes the hybridization of a segment in a loop structure because it has a special behavior. Part of a strand in a loop does not hybridize simply to another strand as a free single strand does [5]. For this reason, the speed of hybridization is modified to be fifty times slower than the speed of branch migration if either of the connected segments is in a loop. Although we need to adjust these values carefully in the future, these values are enough to design a simple logic gate.

Differential Equation. With the assumptions mentioned above, the simulation becomes a problem to solve a differential equation with an initial condition. The initial condition is given as pairs of structures and concentrations. To solve the differential equation, we used Heun's method.

A concentration threshold was introduced because the number of predicted structures would cause an explosion of combinations. In other words, most of the structures simulated were not the main product of the reaction. Structures that do not have a higher concentration than the threshold concentration 10^{-5} were disregarded. This limitation prevents this explosion of combinations, and stabilizes the simulation.

2.2 System Evaluation

A kinetic simulation is used to evaluate a DNA logic gate system, and the evaluation value indicates how correct the system works as a logic gate. The evaluation value is a weighted average of v_1 which is a gap value, and v_2, which is an average gap;

$$\text{evaluation value} = \frac{9v_1 + v_2}{10}.$$

v_1 is calculated as

$$v_1 = min(T) - max(F),$$

where T is a set of concentrations of the output with inputs that return a true logical value, and F is a set of concentrations of the output with inputs that return a false logical value. The gap value is the main point of the estimation. This calculation subtracts the worst value of the false input from the worst value of the true input. If this gap is big, the system is recognized as working correctly from the viewpoint that it is possible to distinguish the true state from the false state.

Let n be the number of total single strands in the system, S be the set of single stranded structures, and C_i be the concentrations of four states of structure i calculated with the simulation,

$$v_2 = \frac{1}{n} \sum_{s \in S} (max(C_s) - min(C_s)).$$

This function calculates the average gap of each concentration of the structure after simulation. If v_2 is high, it is possible to understand that a more complex reaction is taking place in the system being evaluated.

2.3 Design of System

Our method can be used to design a DNA logic gate that consists of three collections of strands, that is, inputs, a gate, and output. The output contains one strand that is chosen automatically by selecting from the gate, which returns the highest evaluation value.

Search Algorithm. To design a highly evaluated system that satisfies the given condition, a heuristic method was used. We used a simulated-annealing algorithm to search for a semi-optimal DNA logic gate system. The simulated-annealing algorithm gradually increases the evaluation value, and it is possible to escape from a local solution while the temperature parameter is high enough.

The outline of the algorithm is as follows, first, find a random answer for an initial state, and initialize the temperature parameter. Second, make another state with some changes as a neighbor. If the exchange condition, which is to avoid falling into a local maximum, is satisfied, then change the state to the neighbor. If the condition is not satisfied, the state remains. Next, cool the temperature parameter as scheduled. Repeat the procedure from the second step until the terminal condition. Finally output the present state as an answer.

Settings of Initial Solution and Neighbors. A random initial state is constructed from a random gate and output, and random inputs. By assuming a random strand as a sequence of random segments with a random length, random inputs consist of two random strands. A random gate is a collection of four or less strands. The random length of a strand is a number between 1 and 4, and random segments are chosen from 'a' to 'f' or from 'A' to 'F'. Each number is modifiable by changing the condition of the method. One strand is chosen from

Table 1. Probabilities of each change

change of system	change inputs	0.3
	change gate	0.7
change of strand	insert one segment	0.3
	remove one segment	0.3
	flip one segment	0.4

the gate as an output because of the assumption that an output is automatically generated from the gate.

A neighbor state is a local change of the present state, which is calculated by changing one of the inputs or gate strands, adding a random strand to the gate or removing a random strand from the gate. Table 1 shows the probabilities used to generate a random neighbor.

3 Experimental Results

By using our method, the design of six kinds of logic gates, which were AND, OR, NAND, NOR, NOT and XOR gate, was experimented. The difference among the experiments was only the T and F sets throughout the evaluation. As a result of the search, several kinds of logic gates were provided automatically, and some examples are introduced below.

Figure 4 shows an example of the topology of an AND gate system found using our search method. Input 1 "AB" displaces the output strand "A" by branch migration and wastes a structure in which "AB" and "ba" are together. Although the output strand becomes single stranded, the output strand hybridizes to another strand making a bond between 'A' and 'a'. Input 2 "AC" similarly displaces the output if the reaction of input 1 is completed. If either input is insufficient, the output strand will not be emitted. This is how this DNA logic gate system acts as an AND gate. The calculated evaluation value was about 0.92. There was a segment attached to one of the gate strands that had no function in the reaction.

A simulation was carried out to investigate the time change of the output concentration. The results of simulating the AND gate is shown in Figure 5. The concentration of the output increased if and only if both inputs were added.

To prove the correctness of our method, we experimented with the AND gate system designed by our method using the strands shown in Table 2. The AND gate was checked by the change of fluorescent intensity (Figure 6) at 37 °C using a F-2500 Fluorescence Spectrophotometer (HITACHI). The excitation and emission wavelengths for FAM were respectively 494 and 518 nm. And all oligonucleotides were first dissolved in a 10 mM Tris Buffer with a pH of 7.4

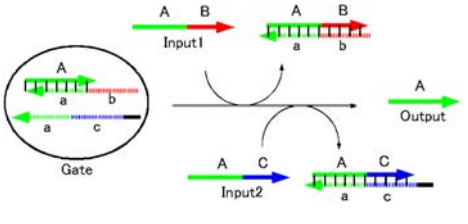

Fig. 4. Designed AND Gate

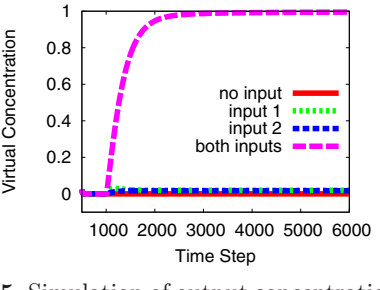

Fig. 5. Simulation of output concentration of AND gate

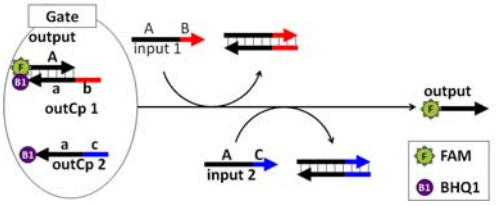

Fig. 6. Experimental AND Gate

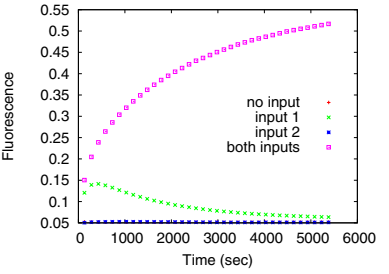

Fig. 7. Fluorescence experiment

Table 2. Sequence of oligonucleotides (5' end to 3' end)

Input 1	CCAAACTACTTACGTTGAACATACACCGAGGTTTAGTGAACACTTCTAAGCAACTAA
Input 2	TGAACATACACCGAGGTTTAGTCCAAA
Output	TGAACATACACCGAGGTTTAG
OutComp 1	TGTTCACTAAACCTCGGTGTATGTTCA
OutComp 2	TTTGGACTAAACCTCGGTGTATGTTCA

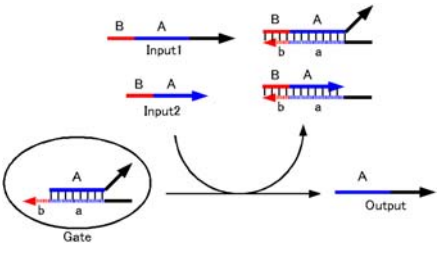

Fig. 8. Designed OR Gate

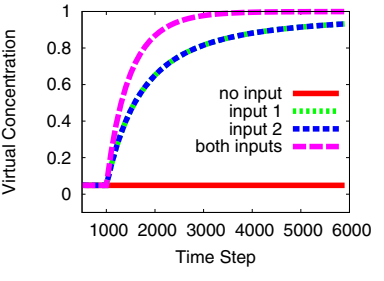

Fig. 9. Simulation of output concentration of OR gate

and then diluted with a 1 SSC buffer containing 0.15 M of NaCl and 15 mM of sodium citrate with a pH of 7.0. Figure 7 shows the change in fluorescence. It is clear that the system works correctly as an AND gate because the fluorescence is measured highly only if both inputs are added.

Although we were able to find an OR, NAND, NOR and NOT gate, we have not tested the gates in vitro. OR and NAND gates are explained below. Figure 8 is an OR gate designed by our method, where both inputs can displace the output by branch migration using segment 'b' as a toehold. This OR gate is essentially the same as the Seelig's enzyme-free OR gate [1]. Though three segments are unnecessary for the reaction, it was found that this DNA logic system functions as an OR gate, from the results of simulating the output concentration (Figure 9). Figures 10 and 11 are the same as those for the NAND gate. Input 1 is a hairpin structure with segment 'a' in a loop. Although the output is a single strand with one segment 'A', the speed of hybridization between these segments is regulated on the assumption that a segment in a loop is treated as an exception. Input 2 opens the hairpin structure of input 1 by branch migration, and segment 'a'

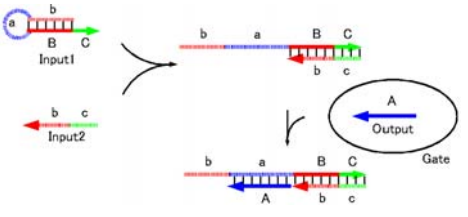

Fig. 10. Designed NAND Gate

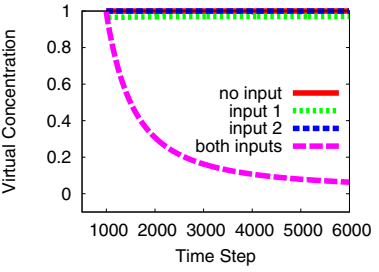

Fig. 11. Simulation of output concentration of NAND gate

becomes accessible. As a result, the concentration of the output strand decreases by hybridizing to the segment. The time change of the output concentration is shown in Figure 11, which indicates that this DNA logic system behaves as a NAND gate.

4 Discussion

By listing the examples that our method found and testing in vitro, it appears a simple DNA logic gate can be automatically designed using this method. Among these DNA logic gate systems, branch migration and hybridization were the center of the reaction. The reason for segments or strands that have no effect on a DNA logic gate system may be that those segments do not affect the evaluation value.

The results of the fluorescence experiment and simulation are basically the same, although details of the two graphs are different. The difference was caused by not adjusting the rate constant between simulation and the fluorescence experiment. Although it was possible to design an XOR gate with a high evaluation value using our method, the gate worked well on the simulation and did not seem to be adaptable to a chemical experiment. This is because of the rough abstraction and approximation of the simulator. These results suggest that our method has potential for designing a more complex DNA logic gate system by improving the simulator.

By assuming the third type of hybridization, there is a possibility to compose a pseudoknot structure, even if it is not a natural structure. It should be noted that the simulation of DNA concentration does not correspond to the actual concentration. Although it is a virtual concentration, it seems that the simulator can calculate the behavior of the DNA strand as a whole. Two inputs in the gate found using our method are not completely different, for the inputs can have the same segment in common. Because our method was developed to design the topology of a system constructed only with DNA, it does not recognize reactions with enzyme and does not support sequence design.

Because our method searches for only logic gates, further tasks should include searching for other devices such as a DNA walker, DNA comparator, or exponential growth system. We also plan to improve the accuracy and speed of the kinetic simulation.

5 Conclusion

Since it has been a task to design a nano-device based on DNA, we developed a program for automatically designing a DNA logic gate system using a kinetic simulation. Although there were several limitations, it was possible to design several DNA logic gates using the method, and an AND gate worked correctly with the fluorescence experiment. By conquering the limitations, this system will support more applications in the future.

References

1. Seelig, G., Soloveichik, D., Zhang, D.Y., Winfree, E.: Enzyme-free nucleic acid logic circuits. Science 314(5805), 1585–1588 (2006)
2. Zhang, D.Y., Turberfield, A.J., Yurke, B., Winfree, E.: Engineering entropy-driven reactions and networks catalyzed by DNA. Science 318(5853), 1121–1125 (2007)
3. Yin, P., Choi, H.M.T., Calvert, C.R., Pierce, N.A.: Programming biomolecular self-assembly pathways. Nature 451(7176), 318–322 (2008)
4. Qian, L., Winfree, E.: A simple dna gate motif for synthesizing large-scale circuits. In: Proceedings of 14th International Meeting on DNA Computing (DNA14), pp. 139–151 (2008)
5. Seelig, G., Yurke, B., Winfree, E.: Catalyzed relaxation of a metastable DNA fuel. Journal of the American Chemical Society 128(37), 12211–12220 (2006)