Vol.2011-MPS-84 No.12
2011/7/18

Abstraction of DNA Graph Structures for Efficient Enumeration and Simulation

Ibuki Kawamata ,[†1] Fumiaki Tanaka [†1]
and Masami Hagiya[†1]

We propose a graph model of DNA molecules and an abstraction of that model for efficient simulation of molecular systems powered by DNA hybridization. In this paper, we first explain our DNA molecule model composed of graph data structures and highlight the problem of the large number of DNA structures that results. We then define an abstraction of the model, which focuses on local structures of DNA strands, and introduce reactions among the local structures. To verify the effectiveness of the abstraction, we develop simulators for the original and abstract models, and compare the number of structures generated by those simulators. Based on this research, computer-aided design of reaction systems that consist of biological molecules may become easier than conventional designs that rely on human trial and error.

1. Introduction

Molecular systems using DNA and its simple hybridization mechanism have been recently developed, including nano-scale DNA structures[1),2)], DNA logic gates[3),4)], and DNA amplification machines[5),6)]. The design of such systems, however, is extremely difficult for humans because the combination of molecules in the system increases rapidly as the number of molecular species increases. This combinatorial explosion prevents humans from predicting system behavior and limits the total number of molecular species that can be used.

A variety of approaches for overcoming this difficulty in combining molecules have been proposed, most of which are based on simplified molecules and restricted reactions. Good examples of such approaches include simple hairpin strands of DNA that allow a cascade of reactions [7)], the programming language for DNA circuits[8)], and the computer-aided tool to produce three-dimensional DNA structures[9)]. Even these methods, however, still require human trial and error to synthesize systems of interest.

We previously proposed a method for the automatic design of DNA logic gates to synthesize small systems based on a DNA model without human trial and error[10)]. In that method, we defined a graph data structure to represent DNA molecules and developed a simulator based on chemical kinetics. Although we restricted the model of structures and introduced threshold to ignore unimportant structures, the simulator still led to a combinatorial explosion of structures.

In this study, we abstract the model by focusing on the local structure of DNA strands to overcome the explosion problem. The approach based on the local structures is similar to the equilibrium computation for hybridization reaction systems[11)] and the rule-based language for cellular signaling pathways[12)].

Fig. 1 DNA modeling

2. Graph Structure Modeling

In this section, we briefly explain how to model DNA by simple graph data structures in our previous work[10)]. Remaining parts after this section are based on this graph model.

2.1 Structure

We modeled DNA molecule as a graph data structure to provide a computational model for systems composed of DNA[10)]. For example, **Fig. 1** shows the application of the model to a DNA logic gate[3)] in a step-by-step manner. Chemically, DNA is a sequence of nucleotides that can be specified by a string of the four elemental bases 'A', 'T', 'G', and 'C'. The duplex structure is formed by hydrogen bonds between complementary base pairs in antiparallel directions (leftmost in Fig. 1). Because the target system was a logic gate, we ignored information about the duplex and saved the directions of phosphate backbones by representing a single DNA strand as an arrow and hydrogen bonds as connected lines (second left in Fig. 1).

†1 Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo

© 2011 Information Processing Society of Japan

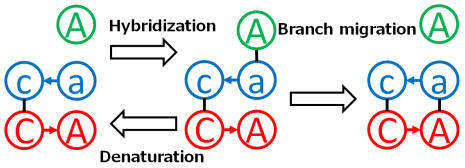

Fig. 2 Reaction rules

We treated a reaction unit of bases as a segment, and a single DNA strand was abstracted into a sequence of segments by allocating a letter to each segment (second right in Fig. 1). We used lowercase and uppercase letters to represent information about complementary relationships between segments. For example, 'a' is complementary to 'A'.

Although many kinds of systems are designed using a similar modeling technique, we further abstracted this model as a graph data structure to simplify the reaction rules. We regarded segments as nodes, hydrogen bonds as undirected edges, and phosphate backbones as directed edges (rightmost in Fig. 1). We assumed that one DNA structure corresponds to a connected graph and regarded a disconnected graph as a set of DNA structures.

2.2 Reaction

After the DNA graph data structure is thus obtained, we defined three reaction rules, namely, hybridization, denaturation, and branch migration (**Fig. 2**), because many artificial DNA systems can be developed using only these three simple mechanisms (such as[3)–6)]). Hybridization represents a reaction in which antiparallel complementary base pairs bind together with hydrogen bonds. This corresponds to adding a new undirected edge between nodes of uppercase and lowercase letters (transition from the left to center in Fig. 2). Denaturation is the inverse reaction in which hybridized complementary base pairs separate from each other. This corresponds to erasing the undirected edge (transition from the center to left in Fig. 2). Branch migration is a reaction in which an exchange of hydrogen bonds occurs in a single molecule at the branching position of three hybridized strands. This corresponds to transferring an undirected edge (transition from the center to right in Fig. 2).

This data structure and the reaction model are sufficient to represent artificial systems powered by DNA hybridization reactions.

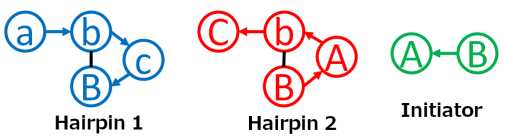

Fig. 3 HCR components in the graph model

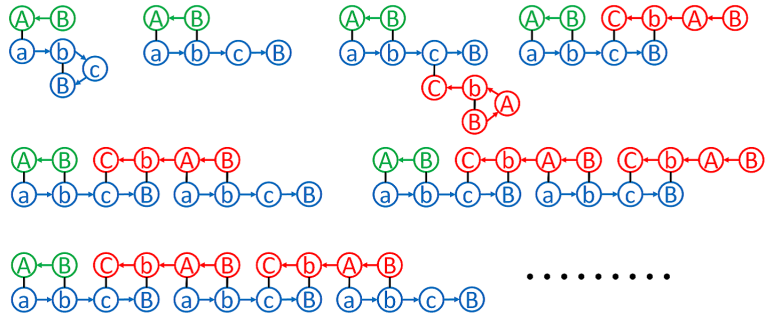

Fig. 4 List of HCR structures

3. Explosion Problem

The combinatorial explosion of molecules is a fundamental problem, especially in simulations of molecular reaction systems including those inside a cell. For example, an unbounded number of structures are produced by a hybridization chain reaction (HCR) that causes a cascade of hybridization reactions triggered by an initiator[6)]. In an HCR, there are two hairpin DNA strands at the initial condition of the system and one initiator strand that serves as input (**Fig. 3**).

By adding the initiator to the system, hybridization and branch migration reactions occur alternately and the length of the structure grows unboundedly because of the very large number of copies of hairpin strands (**Fig. 4**). The figure lists the possible structures of the early stage of HCR using the DNA graph model to illustrate the concept of unbounded growth.

Simulating this kind of system is impossible because of the requirement to allocate an unbounded number of variables to each structure.

© 2011 Information Processing Society of Japan

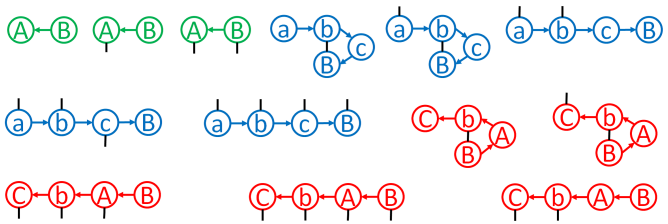

Fig. 5 Local structures example

4. Abstraction by Local Structure

To avoid such unbounded numbers of structures, we introduce an abstraction of the graph model by focusing on the local structure. Although the information about the global structure is lost by the abstraction, using the simulator to design DNA circuits is possible when the outputs are assumed to be single-stranded. The abstraction is done by enumerating possible connecting states of single strands; this is possible because the number of strands is limited even if the number of structures is unbounded. At least 13 local structures of single strands exist for the HCR reaction, as shown in **Fig. 5**. Note that each undirected line corresponding to a hydrogen bond contains information about the segment to which it connects but the information is omitted in the figure. By a calculation explained later, the total number of local structures is 126, which means that a finite number is obtained by enumerating local structures.

The concept of local structure is defined formally as follows. Assume that an alphabet Σ and a set of single strands $S \subseteq \Sigma^*$ are given in advance, and the binary relation $X \subseteq \Sigma \otimes \Sigma$ is also defined to represent the complementary relationships of segments, where $\otimes$ represents a direct product of sets. By distinguishing all segments of strands, we define the set of local segments $G \subseteq S \otimes \mathbb{N}$ as

$$G = \{(s, i) \mid s \in S, i \in \mathbb{N}, i \leq |s|\},$$

where $\mathbb{N}$ denotes the set of all positive integers, and $|s|$ denotes the length of s. We define a function $LETTER$, which is a map from G to Σ such that for any $g = (s, i) \in G$ and $s = a_1\, a_2\, a_3\, \cdots$, $LETTER(g) = a_i$ holds. This function gives the corresponding letter of a given local segment. As a consequence, the set of local structures $L \subseteq S \otimes (G \cup \epsilon)^*$ is defined as

$$L = \{(s, g_1\, g_2\, \cdots\, g_n) \mid s = a_1\, a_2\, \cdots\, a_n \in S,$$
$$\text{either } (a_i, LETTER(g_i)) \in X \text{ or } g_i = \epsilon \text{ holds for all } 1 \leq i \leq n\}.$$

Note that we use ϵ as a symbol to represent unconnected segments, and sequence of ϵ is allowed in $g_1\, g_2\, \cdots$. Thus, $(s, g_1\, g_2\, \cdots\, g_n)$ corresponds to single-stranded DNA if $g_i = \epsilon$ holds for all $1 \leq i \leq n$.

For example, modeling an HCR by the graph data structure gives sets

$\Sigma = \{$ ‘a’, ‘A’, ‘b’, ‘B’, ‘c’, ‘C’$\}$

$S = \{$“abcB”, “BAbC”, “BA”$\}$

and the relation

(‘a’, ‘A’) $\in X$, (‘b’, ‘B’) $\in X, \cdots$.

Local segments and local structures are defined as

$G = \{(\text{“abcB”}, 1), (\text{“abcB”}, 2), (\text{“abcB”}, 3), (\text{“abcB”}, 4), (\text{“BAbC”}, 1), \cdots\}$

$L = \{(\text{“abcB”}, \epsilon\,\epsilon\,\epsilon\,\epsilon), (\text{“abcB”}, \epsilon(\text{“BA”}, 1)\,\epsilon\,\epsilon), (\text{“abcB”}, (\text{“BA”}, 2)\,\epsilon\,\epsilon\,\epsilon), \cdots\}$.

Enumeration of local structures is performed by finding all possible $l \in L$. To enumerate the total number of local structures from a given alphabet and strands, we define functions $SEGMENTS$, $COMPLEMENTS$, and $CONNECTIONS$. First, $SEGMENTS$ is a map from S to 2^G defined as

$$SEGMENTS(s) = \{(s, 1), (s, 2), \cdots, (s, |s|)\},$$

which expresses all local segments in a given strand. Next, $COMPLEMENTS$ is a map from G to 2^G defined as

$$COMPLEMENTS(g) = \{g' \mid (LETTER(g), LETTER(g')) \in X\}.$$

This finds all segments that are complementary to the given segment. Then, $CONNECTIONS$ is a map from S to $\mathbb{N}$ defined as

$$CONNECTIONS(s) = \prod_{g \in SEGMENTS(s)} (|COMPLEMENTS(g)| + 1),$$

where $|COMPLEMENTS(g)|$ denotes the cardinality of set $COMPLEMENTS(g)$. This calculates the number of all combinations of connections from a given strand. Finally, the total number of local structures is calculated by the following expression

$$\sum_{s \in S} CONNECTIONS(s).$$

© 2011 Information Processing Society of Japan

5. Simulation

This modeling process makes simulation of the concentration changes possible by solving the differential equation using numerical analysis. We defined two kinds of deterministic simulations using either the original or abstracted models of DNA structures. We refer to the simulator based on the original graph and abstracted local model as the original simulator and local simulator, respectively. If a user defines the initial configuration as a set of structures and their concentrations, the simulators return the calculation results and the user can trace the concentration changes. These simulators perform the calculations in two stages: enumerating structures that can be constructed from initial structures, and analyzing the concentration changes numerically.

At the beginning of a simulation, the simulators enumerate whole structures in a system to determine the number of variables and their relationships, where each variable represents the concentration of the corresponding structure. The total number of structures is determined by applying the three reaction rules to the initial set of structures as shown in the HCR reaction example in the previous section.

The original simulator enumerates possible graph structures with two restrictions. First, a structure cannot contain two or more identical single strands; this prevents the combinatorial explosion of structures that contain a repeated sub-structure. Second, structures that have a concentration less than 10^{-5} are disregarded to ignore unimportant structures that may not be the main products of a simulation. To implement this feature, the original simulator generates structures dynamically and checks whether the concentration of each structure exceeds the threshold given in advance. More concretely, the period of a simulation is divided into intervals, and the simulator checks the concentration at the beginning of each interval. The simulator then continues the rest of the simulation with the remaining structures whose concentration does not exceed the threshold.

On the other hand, the local simulator enumerates all of the possible local structures without restriction as explained in section 4.

After enumerating structures and reactions among them, the simulators assign variables to each structure and define differential equations using chemical kinetics. The simulators formalize all three reactions. **Fig. 6** and **Fig. 7** show schematic examples of the original and local simulators, respectively.

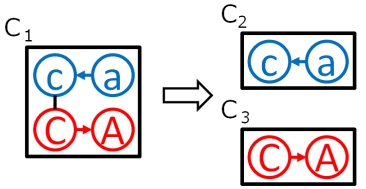

Fig. 6 Formalization for denaturation in original simulation

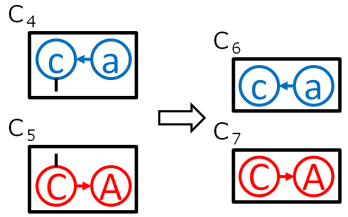

Fig. 7 Formalization for denaturation in local simulation

According to the reactions shown in the figures, differential equations for the original simulation are

$$\frac{d}{dt}C_1 = -k_d C_1, \frac{d}{dt}C_2 = k_d C_1, \frac{d}{dt}C_3 = k_d C_1,$$

and differential equations for the local simulation are

$$\frac{d}{dt}C_4 = -k_d R(C_4C_5), \frac{d}{dt}C_5 = -k_d R(C_4C_5),$$
$$\frac{d}{dt}C_6 = k_d R(C_4C_5), \frac{d}{dt}C_7 = k_d R(C_4C_5),$$

where k_d is the reaction rates for denaturation and $C_1, \cdots, C_7$ are the variables assigned to each structure as a concentration. Because many reactions occur in a single simulation, each of $\frac{d}{dt}C_1, \cdots$ is defined by summing up all of the reactions on which the corresponding structure depends. In the local simulation, we introduce an arrangement (represented by the symbol R) in the differential equations compared with ordinary chemical kinetics. This R is introduced to emulate multi-molecular reactions as unimolecular reactions because denaturation (especially denaturation reactions that separate multiple segments in a row) and branch migration must be unimolecular reactions. This calculates the ratio of concentration among all possible connections from reacting segments. Suppose that C_l denotes the concentration of the local structure l, and function $CONNECTED$ is a map from G to 2^L as

© 2011 Information Processing Society of Japan

$CONNECTED(g) = \{l \mid s \in S, \vec{g} \in (G \cup \epsilon)^*, l = (s, \vec{g}) \in L, g \text{ appears in } \vec{g}\}$, where "$g$ appears in $\vec{g}$ " means that $\vec{g} = g_1\, g_2 \cdots g_n$ and $g = g_i$ holds for some i. $CONNECTED$ finds all local structures that are connected to the given local segment. $R(C_{l_1}C_{l_2})$ for denaturation between the segments g_1 of local structure l_1 and g_2 of l_2 is defined as

$$\frac{C_{l_1}C_{l_2}}{\sum_{l \in CONNECTED(g_1)} C_l} \text{ or } \frac{C_{l_1}C_{l_2}}{\sum_{l \in CONNECTED(g_2)} C_l},$$

where the expressions are equivalent to each other.

The rate of each reaction is defined by rule of thumb, and the kinetics of hybridization and branch migration are fixed. Only the kinetic velocity of denaturation is calculated according to the information of segments that are separating.

6. Enumeration Efficiency

We have described two types of simulations for DNA reaction systems. The efficiency of the two types of simulations was tested in terms of the number of structures. As a benchmark, we generated a random system as a random sequence of letters, which determines the set S. Note that the size of set Σ was fixed to 14 for all simulations. We first fixed a maximum size of S, and then we generated and simulated 200 random systems to obtain the average and maximum numbers of structures. After that, we took another maximum size in turn and repeated the calculation for each maximum size. We tried 21 different sizes (whose values were suitable for simulations), which range over the x-axis of **Fig. 8** and **Fig. 9**.

The figures show the average and maximum number of structures produced by four types of simulation: original simulation without threshold, local simulation, original simulation with threshold, and stochastic simulation. Note that original simulations with or without threshold impose the restriction on DNA structures mentioned above. The x-axis of the figure corresponds to the maximum number of local segments, which is the number of letters in S to generate a random system. The y-axis of the figure corresponds to the number of structures for each simulation.

The stochastic simulation was not explained above because it is not an integral part of this research. Stochastic simulation is an algorithm for simulating discrete

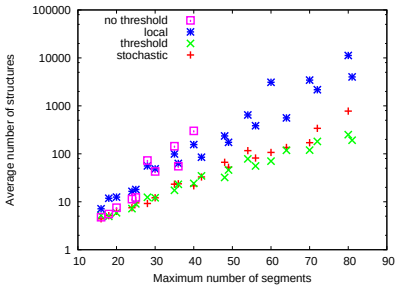

Fig. 8 Average number of structures

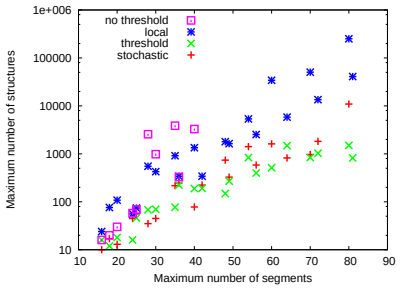

Fig. 9 Maximum number of structures

chemical reaction systems using a statistical simulation method. This statistical method simulates chemical reactions stochastically one by one according to the distribution of possibility of each reaction. We actually implemented Gillespie's algorithm[13] for this method.

Note that the results of original simulation with threshold and stochastic simulation are only shown as references in the figures. Direct comparison of the results is not fair because the values for the original simulation with threshold and the stochastic simulation depend on parameters such as threshold and copy number.

As expected, the original simulation without threshold exhibited faster combinatorial explosion than the others because entire combinations of structures were tested by the execution. Completion of the original simulation with a size greater than 40 was impossible due to an out-of-memory error. The increase in the number of structures in the local simulation seemed to be slower than that of the original simulation without threshold because of the limit on the number of local structures.

These results indicate that an appropriate model and simulation are necessary for the efficient enumeration and simulation of DNA hybridization systems.

7. Discussion

A limit to designing very complex DNA systems lies in the combinatorial explosion problem of the DNA structures. This is critical because preventing the combinatorial explosion by enumerating all of the possible structures is difficult

© 2011 Information Processing Society of Japan

where an unbounded number of structures can occur. Imposing a threshold or artificial limitations on the model of structures did not eliminate the problem and introduced the possibility of incorrect simulation.

A new approach to avoiding the combinatorial explosion was proposed that focused on the local structure, and the efficiency of this approach was better than the original model. The rapid increase in the number of structures was reduced in the local simulation. Considering all of the possible local structures in a simulation was possible because none of the structures were ignored as the result of imposing an artificial threshold. The strongest aspect of the model was the ability to express any kind of structure at the expense of losing some part of the information, even in the case where unbounded structures were involved.

While the purpose of the related work[11] was to theoretically compute equilibrium state of hybridization reaction system based on locality, we gave a concrete simulator in this work that can trace the time change of concentration. Though this work shares the basic idea with the related work[12], we defined the local structure for our original purpose, which is to simulate DNA hybridization systems. We actually showed that the local simulation was effective in enumeration of structures and more precise than the original simulation with threshold. Because the targets of our automatic design were gates that output single-stranded DNA, the modeling using the local structure can be regarded as a novel abstraction that serves our purpose.

8. Conclusion

DNA hybridization systems have been applied to a wide range of applications including molecular robotics, nano-scale structures, and medication control. Because selecting combinations of molecules to achieve some desired functionality is difficult for humans, we previously proposed an automatic design method using an evolutionary computation. Modeling for molecules and reactions was defined by regarding molecules as a graph data structure. Because the automatic design method required efficient enumeration and simulation to avoid combinatorial explosion, we abstracted the model to limit the number of structures by enumeration, even if an unbounded number of structures can be constructed. On the basis of this modeling technique, we developed a simulator and investigated the efficiency in the enumeration of structures. Synthesis of large systems that are more complex than human beings can design will be possible using this new abstracted model.

References

1) Rothemund,P.W.K.: Folding DNA to create nanoscale shapes and patterns, *Nature*, Vol. 440, No. 7082, pp. 297–302 (2006).
2) Andersen,E.S., Dong,M., Nielsen,M.M., Jahn,K., Subramani,R., Mamdouh,W., Golas,M.M., Sander,B., Stark,H., Oliveira,C.L.P., Pedersen,J.S., Birkedal,V., Besenbacher,F., Gothelf,K.V. and Kjems,J.: Self-assembly of a nanoscale DNA box with a controllable lid, *Nature*, Vol. 459, No. 7243, pp. 73–76 (2009).
3) Seelig,G., Soloveichik,D., Zhang,D.Y. and Winfree,E.: Enzyme-Free Nucleic Acid Logic Circuits, *Science*, Vol. 314, No. 5805, pp. 1585–1588 (2006).
4) Qian,L. and Winfree,E.: A simple DNA gate motif for synthesizing large-scale circuits, *DNA Computing*, Vol. 5347 of LNCS, pp. 70–89 (2008).
5) Zhang,D.Y., Turberfield,A.J., Yurke,B. and Winfree,E.: Engineering Entropy-Driven Reactions and Networks Catalyzed by DNA, *Science*, Vol. 318, No. 5853, pp. 1121–1125 (2007).
6) Dirks,R.M. and Pierce,N.A.: Triggered amplification by hybridization chain reaction, *Proc. Natl. Acad. Sci. U. S. A.*, Vol. 101, No.43, pp. 15275–15278 (2004).
7) Yin,P., Choi,H.M.T., Calvert,C.R. and Pierce,N.A.: Programming biomolecular self-assembly pathways, *Nature*, Vol. 451, No. 7176, pp. 318–322 (2008).
8) Phillips,A. and Cardelli,L.: A programming language for composable DNA circuits, *J. R. Soc. Interface*, Vol.6, pp. 419–436 (2009).
9) Douglas,S.M., Marblestone,A.H., Teerapittayanon,S., Vazquez,A., Church,G.M. and Shih,W.M.: Rapid prototyping of 3D DNA-origami shapes with caDNAno, *Nucleic Acids Res.*, Vol.37, No.15, pp. 5001–5006 (2009).
10) Kawamata,I., Tanaka,F. and Hagiya.,M.: Automatic Design of DNA Logic Gates Based on Kinetic Simulation, *DNA Computing and Molecular Programming*, Vol. 5877 of LNCS, pp. 88–96 (2009).
11) Kobayashi,S.: A New Approach to Computing Equilibrium State of Combinatorial Hybridization Reaction Systems, in *Bio-Inspired Models of Network, Information and Computing Systems*, pp. 330–335 IEEE (2008).
12) Danos,V., Feret,J., Fontana,W. and Krivine,J.: Abstract Interpretation of Cellular Signalling Networks, in *Verification, Model Checking, and Abstract Interpretation*, pp. 83–97 Springer (2008).
13) Gillespie,D.: Exact stochastic simulation of coupled chemical reactions, *The journal of physical chemistry*, Vol.81, No.25, pp. 2340–2361 (1977).

© 2011 Information Processing Society of Japan