Sequence Design Support System for 4 × 4 DNA Tiles

Naoki Iimura[1], Masahito Yamamoto[2], Fumiaki Tanaka[3], and Azuma Ohuchi[2]

[1] NTT DoCoMo Hokkaido, Inc.
iimura@complex.eng.hokudai.ac.jp
[2] Graduate School of Information Science and Technology, Hokkaido University
{masahito,ohuchi}@complex.eng.hokudai.ac.jp
[3] Graduate School of Information Science and Technology, Univercity of Tokyo
fumi95@is.s.u-tokyo.ac.jp

Abstract. A DNA computation model by DNA tiles needs sequence design in order to correctly form tile structure and self-assembly. We design sequence, demonstrate biochemical experiments by a trial and error approach, and, repeatedly analyze tiles. Because no integrated sequence design system computes data that indicates properties of sequences, we must analyze designed sequences by hand and many types of software. In this paper, we develop a sequence design support system for 4× 4 DNA tiles that analyzes and optimizes tile sequences to support sequence design. The most remarkable feature of this system is optimization based on free energy. The optimization strategy is developed so that the energy of perfect tile is the stablest.

1 Introduction

New computation models and DNA nanotechnology by DNA tiles based on Watson-Crick complementarity pairing [1, 2] have been proposed. We have to design a stable tile sequence that minimizes mis-hybridization because these computation models by DNA tiles presupposes correct hybridization. However, sequence design is never easy due to the repetition of the trial and error approach. In DNA computing, sequence design that forms a wanted structure or does not form an unwanted structure has been researched in terms of various indexes, including melting temperature, GC content, free energy, and so on. Conventional work of DNA tile sequence design minimizes the reuse of the fragment of sequence without these indexes [3, 4]. This method allows sequence design that does not cause unnecessary hybridization regardless of the simple algorithm. However, it has problems. It does not consider the distinction of stability by base pair or loop structure and does not quantify the evaluation of tiles. Thus, if we design and analyze sequences by other evaluation indexes to solve these problems, we need to use many types of software[5–8].

In this paper, we develop a sequence design support system for a 4× 4 DNA tile. The 4×4DNA tile developed by Yan *et al*, consists of nine sequences: one CORE, four SHELL, and four ARM sequences (Fig. 1). This tile has four-way

M.H. Garzon and H. Yan (Eds.): DNA 13, LNCS 4848, pp. 140–145, 2008.

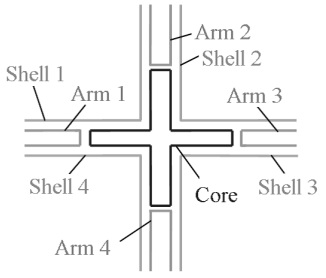

Fig. 1. Definition of sequence name

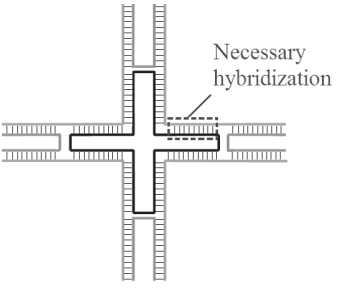

Fig. 2. Necessary hybridizations for 4×4 DNA tile

arms (north, south, east, and west) made of a single strand DNA molecule called "sticky-end" and forms a DNA self-assembly by connecting with other tiles. Applications of the 4×4 DNA tile have been proposed and demonstrated by several research groups[9, 10].Our system has two features for design support. First, it can analyze existing sequence data that are often used when designing sequences. Second, it can design them using optimization based on the stability evaluation of DNA tiles by free energy. Free energy is introduced because it can reduce mis-hybridization by making necessary hybridization stable and unnecessary hybridization unstable. Our proposed system are designed so that the evaluation function of tile structures can be easily replaced.

2 Design Strategy

The stability of tile structure allows sequence design that minimizes mis-hybridization. From the standpoint of tile stability, correct tile structure without mis-hybridization is considered the most stable; that is, the free energy of the correct tile must be the lowest. We apply free energy, which is an index that can evaluate the stability of loop structure and base pairs in DNA computing, to the stability of a 4×4 DNA tile. There are two kinds of free energy: of the secondary structure within a single strand DNA molecule, and of the hybridization between two single strand DNA molecules, however, there is no effective prediction method of free energy of a DNA tile. Here we are trying to quantify the stability of tile structure by these free energies.

We suppose that the whole tile structure becomes more stable as each necessary hybridization portion stabilizes, therefore, the summation of necessary hybridization calculated by the free energy between two sequences can indicate tile stability. Necessary hybridization is a base pair to form tile structure, as shown in Fig. 2. Because these necessary hybridization portions may incorrectly form loop structure, bulge loop, and so on, evaluation of the bond strength of the base pair is used to judge correct hybridization. Furthermore, it is desirable that the stablest structure is only the one structure of any and all structure with the potential to be formed by tile sequences. The reason for this is that

sequences decrease the possibility of forming correct tiles if the structure that does not form tile is as stable as the tile structure.

Desirable sequences have lower energy if the tile is formed correctly and higher energy if the tile is not formed. In other words, desirable sequences have a big difference between the lowest and second lowest energy. A device is needed to design these sequences as well as to stabilize necessary hybridization. Our method incorporates inhibitory factors to avoid forming the secondary structure of a single strand DNA molecule and unnecessary hybridization. The inhibitory factor calculated by the free energy within the sequence and between two sequences reduces tile stability. Sequence forming correct tile are designed by optimization based on tile stability by free energy.

3 Support System

We developed a sequence design system with the previous strategy based on free energy. The system also has an analysis function of existing tiles besides sequence design by optimization because sequence design comprehensively uses not only free energy but also various indexes. These functions should reduce computational costs.

3.1 Analysis Module

We often comprehensively judge tile sequences by amount of data. Before analyzing a tile, the system requests sequence length and SHELL sequences from users, who input sequence data following the input forms on the screen. Figs. 3 and 4 are examples of the screen. Fig. 3 shows the input form of the sequence length of a CORE fragment that is hybridized to the SHELL sequence. This system deals with any tile size and any bulge loop in the corner of CORE. Fig. 4 shows the input form to enter the SHELL sequences. We adopt nucleic code, which has A, G, T, C, S(G or C), H(A,T or C), N(A,T,C or C), and so on by IUPAC, because input forms need to accept existing sequences and constraints for allocating bases. The input of sequence by nucleic code can directly, enter existing sequences, and randomly allocate bases if GC pairs or AT pairs are fixed. Additionally, a user can avoid including specific sequence fragments in a sequence when the system randomly allocates bases.

Fig. 5 shows a screen of the analysis result. The system analyzes the following data after a user inputs the essential data:

(1) GC content of each sequence
(2) Melting temperature of each sequence
    The value indicates the melting temperature between each sequence and its complementary sequence.
(3) Free energy of each sequence
    This value indicates the stability of the secondary structure within a single strand DNA molecule and that is calculated with MFOLD [5, 6].

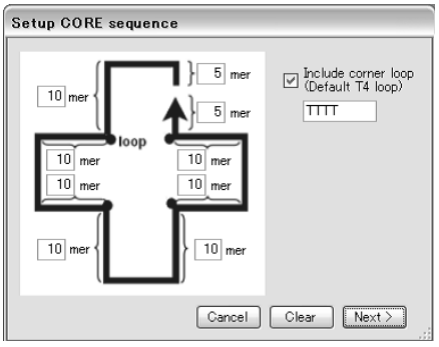

Fig. 3. Input form of CORE sequence data

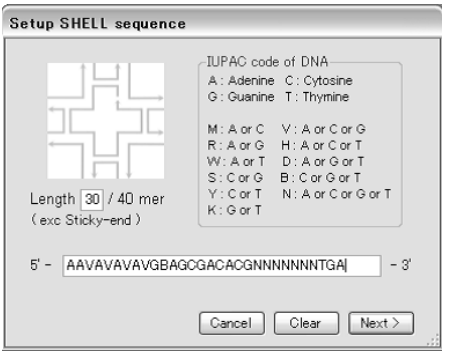

Fig. 4. Input form of SHELL sequences

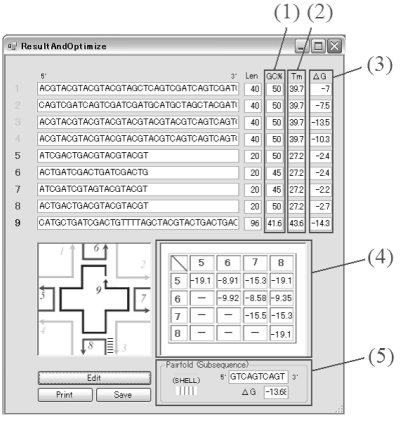

Fig. 5. Sytem interface

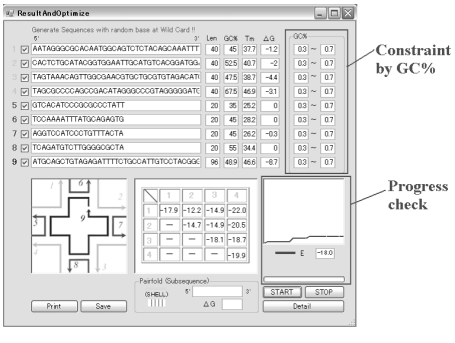

Fig. 6. Optimization interface

(4) Free energy between sequences that do not require hybridization
This value indicates the stability of hybridization between sequences that should not be hybridized to form tiles. The system calculates the free energy between CORE–CORE sequences, between SHELL[i]–SHELL[j] sequences ($i \neq j$), and between ARM[i]–ARM[j] sequences ($i \neq j$) with PAIRFOLD [7] (Fig. 1). The combination of two sequences in all sequences is not calculated in terms of computation cost.

(5) Free energy between sequences that require hybridization to form tiles
This value indicates the stability of hybridization between sequences that should be hybridized and stabilized to form tiles. A 4×4 DNA tile has 16 necessary hybridization parts (Fig. 2). The system uses a fragment of sequence along with a motif of the tile.

The values of (4) and (5) are displayed by click on the motif.

3.2 Optimization Module

This system can optimize an existing or a brand-new tile by free energy known as a more precise index of hybridization stability than other indexes in DNA computing [11]. Our system introduces free energy into the evaluation function that uses the weighting addition of the free energy in the previous section; that is, the sum of free energy (3)–(5). The evaluation function for any tile x is as follows.

$$\begin{aligned} E(x) &= I_1 + \alpha I_2 + \beta I_3 \\ I_1 &= \Sigma \text{ (free energy values of (3))} \\ I_2 &= \Sigma \text{ (free energy values of (4))} \\ I_3 &= -\Sigma \text{ (free energy values of (5))} \end{aligned}$$

Terms $I_1, I_2, and I_3$ indicate the bond strength of the secondary structure within a single strand, the bond strength of mis-hybridization between non-objective sequences, and the bond strength of hybridization between fragments of objective sequences, respectively. A sequence qualifies as a stable tile as each value increases.

The optimization algorithm adopts a hill-climbing algorithm. The system retains the nucleic code and the avoidance fragment in principle while optimizing sequences; furthermore, it can set constraints of GC content in each sequence. Optimization steps are initially 1,000, which a user can increase to 2,000. Running time is about nine minutes on a PC with 3.0 GHz Pentium4 processor and 512 MB RAM running Windows XP, if optimization steps are 2,000. Fig. 6 shows the interface of optimization. We can confirm the optimization progress and stop it if required. The system displays the sequence and analysis results at that time.

3.3 I/O Module

The system has the following convenient additional functions. I/O module inputs and outputs sequence data. The system can save and read sequences, their length, and the result of analysis or optimization by an XML document. This function not only save data but also alleviates input. The preparation of XML documents as templates of tile size facilitates various optimizations. Furthermore, the system can print these data.

4 Discussions and Concluding Remarks

We have designed sequences by minimizing the reuse of fragment of sequence and have analyzed their melting temperature, free energy, and so on by many types of software. Our system can analyze these data of existing sequences, optimize sequences by free energy, and analyze and design sequences from scratch. Optimization results have verified that optimized sequences can form tile correctly in terms of free energy. However, the actual verification of sequence needs

biochemical experiments *in vitro*D Our system provides sequences that users want to design by inputting by nucleic code, setting specific fragments to avoid, setting the constraints of GC content, and changing the parameters of the evaluation function. We consider that this reflects knowledge gained by biochemical experiment into the system. Additionally, the system can design sequences that are not designed by the conventional algorithms.

We suggest that sequences by this system form a tile and self-assembly more precisely. However, we may need to consider the concentration of each sequences, curvature of tiles, other optimization methods and constraints in the future.

References

1. Winfree, E., Liu, F., Wenzler, L.A., Seeman, N.C.: Design and self-assembly of two-dimensional DNA crystals. Nature 394, 539–544 (1998)
2. Yan, H., Feng, L., LaBean, T.H., Reif, J.H.: Parallel Molecular Computations of Pairwise Exclusive-Or (XOR) Using DNA "String Tile" Self-Assembly. J. Am. Chem. Soc. 125(47), 14246–14247 (2003)
3. Seeman, N.C.: De Nove Design of Sequence for Nucleic Acid Structual Engineering. Jornal of Biomolecular Structure & Dynamics 8, 739–1102 (1990)
4. Wei, B., Wang, Z., Mi, Y.: Uniquimer: Software of De Novo DNA Sequence Generation for DNA Self-Assembly -An Introduction and the Related Applications in DNA Self-Assembly. Journal of Computational and Theoretical Nanoscience 4(1), 133–141 (2007)
5. Zuker, A.M., Mathews, B.D.H., Turner, C.D.H.: Algorithms and Thermodynamics for RNA Secondary Structure Prediction: A Practical Guide. In: Barciszewski, J., Clark, B.F.C. (eds.) RNA Biochemistryand Biotechnology. NATO ASI Series, Kluwer Academic Publisers, Dordrecht (1999)
6. Zuker, M.: Mfold web server for nulceic acid folding and hybridization prediction. Nucleic Acids Reserch 31(13), 3406–3415 (2003)
7. Andronescu, M., Aguirre-Hernandez, R., Condon, A., Hoos, H.H.: RNA soft: A suite of RNA secondary structure prediction and design software tools. Nucleic Acids Research 31(13), 3416–3422 (2003)
8. Hofacker, I.L.: Vienna RNA secondary structure server. Nucleic Acids Research 31(13), 3429–3431 (2003)
9. Yan, H., Park, S.H., Finkelstein, G., Reif, J.H., LaBean, T.H.: DNA-Templated Self-Assembly of Protein Arrays and Highly Conductive Nanowires. Science 301, 1882–1884 (2003)
10. Park, S.H., Yan, H., Reif, J.H., LaBean, T.H., Finkelstein, G.: Electronic nanostructures templated on self-assembled DNA scaffolds. Nanotechnology 15, S525–S527 (2004)
11. Tanaka, F., Kameda, A., Yamamoto, M., Ohuchi, A.: Design of nucleic acid sequences for DNA computing based on a thermodynamic approach. Nucleic Acids Research 33(3), 903–911 (2005)