nature 28 September 2000

letters to nature

Nature **407**, 493 - 496 (2000)

Logical computation using algorithmic self-assembly of DNA triple-crossover molecules

CHENGDE MAO*, THOMAS H. LABEAN†, JOHN H. REIF† & NADRIAN C. SEEMAN*

* Department of Chemistry, New York University, New York, 10003, USA &
† Department of Computer Science, Duke University, Durham, North Carolina 27707, USA

Correspondence and requests for materials should be addressed to N.C.S. (e-mail: ned.seeman@nyu.edu).

Recent work[1-3] has demonstrated the self-assembly of designed periodic two-dimensional arrays composed of DNA tiles, in which the intermolecular contacts are directed by 'sticky' ends. In a mathematical context, aperiodic mosaics may be formed by the self-assembly of 'Wang' tiles[4], a process that emulates the operation of a Turing machine. Macroscopic self-assembly has been used to perform computations[5]; there is also a logical equivalence between DNA sticky ends and Wang tile edges[6,7]. This suggests that the self-assembly of DNA-based tiles could be used to perform DNA-based computation[8]. Algorithmic aperiodic self-assembly requires greater fidelity than periodic self-assembly, because correct tiles must compete with partially correct tiles. Here we report a one-dimensional algorithmic self-assembly of DNA triple-crossover molecules[9] that can be used to execute four steps of a logical (cumulative XOR) operation on a string of binary bits.

A variety of different DNA tile types have been used in previous assemblies, including double-crossover molecules[1], triple-crossover molecules[9], and parallelograms produced from Holliday junction analogues[3]. Here we have used triple-crossover molecules; their molecular structure is illustrated in Fig. 1a. The molecule contains four strands (shown in red, green, blue and purple) that self-assemble through Watson–Crick base pairing to produce three double helices in a roughly planar arrangement; each double helix is connected to adjacent double helical domains at two points where their strands cross over between them. The ends of the central double helix are closed by hairpin loops, but the other helices can terminate in sticky ends containing information that directs the assembly of the tiles.

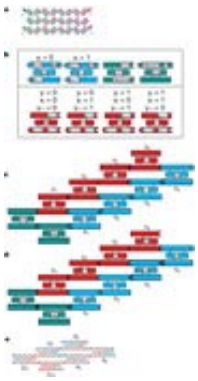

Figure 1 Calculation of cumulative XOR by self-assembly of DNA tiles. Full legend

High resolution image and legend (129k)

In periodic assemblies, the sticky ends contain positional information that directs the associations of one or more tile types to produce a periodic lattice. Here, the sticky ends serve the same role, but the self-assembly of the tiles is used to perform a computation, and the arrangement of the tiles does not display simple periodicity. Figure 1b shows the tiles used to perform the cumulative XOR calculation. The tiles are represented schematically; the three helices are depicted as connected rectangular forms terminating in sticky ends, which are represented as geometrical shapes, or non-cohesive blunt ends and hairpins, which are drawn flush.

The result of the XOR operation is a 0 if two input numbers are the same (two zeros or two ones), but it is 1 if one of the two numbers is 0 and the other is 1. The cumulative XOR consists of a series of Boolean inputs x_1, $x_2, x_3, \ldots x_n$, and the output is also a series of Booleans, $y_1, y_2, y_3, \ldots y_n$, where $y_1 = x_1$, and for $i > 1$, $y_i = y_{i-1}$ XOR x_i. The value of any y_i in these calculations also reports the even or odd parity of the first i values of x. Thus, two different kinds of input x tiles are needed, one whose value is 0 and a second whose value is 1. Chemically, the value of a tile, 0 or 1, is denoted by the presence of a restriction site: Pvu II (CAGCTG) represents 0 and EcoR V (GATATC) represents 1.

The x tiles are shown in blue in Fig. 1b: Their value (0 or 1) is shown in their central rectangle, the upper-left sticky end reports this value, and the upper-right and lower-left sticky ends provide the means of connecting successive x tiles. These sticky ends are shown as geometrically complementary, as they would be for a general set of 16 parallel calculations. However, to demonstrate the efficacy of the procedure, the calculations performed here are definite four-bit calculations for which the order of the input x tiles was specified exactly by a series of different sticky ends.

Like x tiles, two values of y tiles are necessary, again representing 0 and 1. However, there are two ways to get each of these results: the value of a y tile can be 0 either because both inputs are 0 or because both are 1; likewise, the value of a y tile can be 1 because the value of one input is 0 and the other is 1, or vice versa. Thus, four different y tiles are needed.

The y tiles are shown in red in Fig. 1b. The tile values again are displayed on the central domain, and this value is reported by the sticky end on the right of the upper domain. The two inputs derive from the sticky ends on the left (y_{i-1}) and right (x_i) of the bottom domain. We note that the same sticky end in the input domain represents a given input, independent of the other end. For example, the right-side sticky end $x_i = 1$ has the same shape (sticky end) regardless of whether the left-side sticky end represents $y_{i-1} = 0$ (leading to a tile value of 1) or $y_{i-1} = 1$ (leading to a tile value of 0). There are only two different left sticky ends in the input (bottom) domain, and likewise only two different right sticky ends. Consequently, both sticky ends on each tile must pair correctly for the proper y_i tile to be inserted in the assembly. In contrast to periodic assembly, where correct tiles compete with incorrect tiles for each site in the lattice, here correct tiles are competing with partially correct tiles.

We have performed two different XOR-related self-assemblies, illustrated in Fig. 1c and d. In addition to the x and y tiles, two corner tiles, C1 and C2 (green in Fig. 1b) are used to initialize the two values of x_1 and y_1, and to connect the input to the output. The self-assembly in Fig. 1c has the inputs $x_1 = x_2 = x_3 = 1$ and $x_4 = 0$. These correspond to output values of $y_1 = 1, y_2 = 0$, and $y_3 = y_4 = 1$. In a second self-assembly (Fig. 1d), $x_1 = 1$, $x_2 = 0$, $x_3 = 1$ and $x_4 = 0$, corresponding to $y_1 = y_2 = 1$ and $y_3 = y_4 = 0$. Note that for $i > 1$, the sticky ends on the bottom domain of each y_i tile complement those on the y_{i-1} tile on its left and the x_i tile on its right.

Once the self-assembly has occurred, it is necessary to extract the answer. For this purpose, each molecular tile contains a 'reporter strand'[10], which traverses the tile in a diagonal pathway[9]; the reporter strand is illustrated as a thick red strand in the tile shown in Fig. 1a, which is an x-type tile. Following self-assembly, the reporter strands are ligated to each other to produce a long reporter strand that contains the inputs and outputs

of the calculation. The ligated long reporter strand in the vicinity of the corner of the assembly is shown as a thick red strand on the molecular diagram in Fig. 1e.

The sticky ends used in the assembly of the C_1– C_2–x_1–x_2– x_3–x_4 unit contain seven nucleotides, and the sticky ends used to include the y_i tiles in the assembly contain five nucleotides. The tiles were first assembled individually from their component strands by cooling slowly from 90 °C to room temperature, as done previously[1-3]. 20-μl aliquots of stock solutions (in USB ligation buffer) of both C tiles (100 nM), the four x tiles (100 nM), and the four y tiles (400 nM) were then combined and incubated for 30 min each at temperatures of 37, then 22, and finally 4 °C. During incubation and subsequent steps, 20-μl aliquots of three double helices, each with a sticky end (one helix to pair with the free sticky end on x_4 and two helices to pair with the two possible free sticky ends on y_4), that contained radioactively labelled PCR primers (800 nM) were also present in solution.

Ligation was initiated by adding 20 units of T4 DNA ligase and, over a 3-hour period, the solution was brought slowly to 16 °C as ligation proceeded. The strand was amplified by polymerase chain reaction (PCR), using the primers that were ligated to each end of the long reporter strand. A strand of the proper length was eluted from a denaturing gel, was re-annealed, and was subjected to restriction by either of the restriction enzymes. The results are displayed in Fig. 2. The answer produces a barcode display, much like that used in ref. 11 to visualize the answers to RNA computation of chess problems. The correct answers are evident as dark bands in the gels shown in Fig. 2.

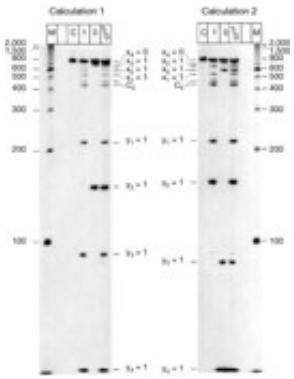

Figure 2 Denaturing gels illustrating cumulative XOR calculations. Full legend

High resolution image and legend (47k)

Small proportions of incorrect bands are visible on the original gels for both calculations. Thus, in lane 1 (EcoR V) of calculation 1, there is an incorrect band at the 0 position of y_2, and a similar incorrect band in the 0 lane at y_3 is also present. However, if the array fills from the corner, some of the incorrect y_3 intensity could result from 'correct' molecules propagating the previous error. Similarly, calculation 2 has one detectable error band resulting from cleavage of EcoR V at y_3.

We estimate the error level to be about 2–5%, but quantitative error analysis is complicated by differential cleavage activities between the two enzymes, combined with the possibility of star-activity (sequence infidelity) and probable multiple cleavage of the same strand. Also, the individual enzymes cleave with different activities at different sites, as seen in the differential cleavage of the two EcoR V sites in the C2 tile. We may have reduced our observation of self-assembly errors by selecting only those tiles that ligated correctly, because the enzyme specificity for exact pairing, although imperfect[12], may have performed some discrimination for the system. A previous two-molecule, single-step competition experiment estimated error rates below 1.6% (ref. 13).

The algorithmic molecular assembly described here demonstrates a non-trivial DNA computation done by

self-assembly. Examples of SAT (satisfaction) problems solved in a DNA context[11,14,15] entailed laboratory operations for each clause in a logical statement, whereas a single self-assembly step is used here. This suggests that computation by self-assembly may be scalable. Another recent work[16] also uses only a single assembly step, but its scalability relies on proper hairpin formation in very long single-stranded molecules.

XOR computation on pairs of bits (as done here) can be used for executing a one-time pad cryptosystem that provides theoretically unbreakable security[17]. Other applications could involve the algorithmically directed self-assembly of intricate patterns and smart materials. We used y tiles repetitively in both assemblies, and would need no more species of y tiles, regardless of the length of the calculation. Thus, if the assembly principles applied here can be extended to two and three dimensions, it will be possible to prepare nanoscale patterns and smart materials by laying out components algorithmically, without the need to specify and prepare a unique element for every position of the array.

By using more nucleotides in the sticky ends of the input tiles than the output tiles, we have used the principle of 'frames'[6,7,18] This feature performs the computation in the presence of a well-defined border. Such borders are likely to be useful, because they set limits on the extent of the calculation or patterning; combining framed arrays will facilitate a modular approach to the process.

Supplementary information is available on *Nature*'s World-Wide Web site (http://www.nature.com) or as paper copy from the London editorial office of *Nature*.

Received 27 April 2000; accepted 3 August 2000

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). Links
2. Liu, F., Sha, R. & Seeman, N. C. Modifying the surface features of two-dimensional DNA crystals. *J. Am. Chem. Soc.* **121**, 917-922 (1999).
3. Mao, C., Sun, W. & Seeman N. C. Designed two-dimensional DNA Holliday junction arrays visualized by atomic force microscopy. *J. Am. Chem. Soc.* **121**, 5437-5443 (1999).
4. Wang, H. in Proceedings of a Symposium in the Mathematical Theory of Automata 23-26 (Polytechnic Press, New York, 1963).
5. Rothemund, P. W. K. Using lateral capillary forces to compute by self-assembly. *Proc. Nat. Acad. Sci. USA* **97**, 984-989 (2000). Links
6. Winfree, E. in DNA Based Computers: Proceedings of a DIMACS Workshop, April 4, 1995, Princeton University (eds Lipton, R. J. & Baum, E. B.) 199-221 (American Mathematical Society, Providence, RI, 1996).
7. Winfree, E. Algorithmic Self-Assembly of DNA. PhD Thesis, Caltech (1998).
8. Adleman, L. Molecular computation of solutions to combinatorial problems. *Science* **266**, 1021-1024 (1994). Links
9. LaBean, T. et al. The construction, analysis, ligation and self-assembly of DNA triple crossover complexes. *J. Am. Chem. Soc.* **122**, 1848-1860 (2000).
10. Seeman, N. C. Nucleic acid nanostructures and topology. *Angew. Chem. Int. Edn Engl.* **37**, 3220-3238 (1998).
11. Faulhammer, D., Cukras, A. R., Lipton, R. J. & Landweber, L. F. Molecular computation: RNA solutions to chess problems. *Proc. Natl Acad. Sci. USA* **97**, 1385-1389 (2000).
12. Harada, K. & Orgel, L. E. Unexpected substrate specificity of T4 DNA ligase revealed by in vitro selection. *Nucleic Acids Res.* **21**, 2287-2291 (1993).
13. Winfree, E., Yang, X. & Seeman, N. C. in DNA Based Computers: II Proceedings of a DIMACS Workshop, June 10-12, 1996, Princeton University (eds Landweber, L. F. & Baum, E. B.) 217-254 (American Mathematical Society, Providence, RI, 1999).
14. Liu, Q. et al. DNA computing on surfaces. *Nature* **403**, 175-179 (2000). Links
15. Pirrung, M. C. et al. The arrayed primer extension method for DNA microchip analysis. Molecular computation of satisfaction problems. *J. Am. Chem. Soc.* **122**, 1873-1882 (2000).
16. Sakamoto, K. et al. Molecular computation by DNA hairpin formation. *Science* **288**, 1223-1226 (2000). Links
17. Gehani, A., LaBean, T. H. & Reif, J. H. in DNA Based Computers: Proceedings of a DIMACS Workshop, June 1999, MIT (ed. E. Winfree) (DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Providence, RI, in the press).
18. Reif, J. H. in DNA Based Computers: III Proceedings of a DIMACS Workshop, June 23-25, 1997, University of Pennsylvania (eds Rubin, H. & Wood D. H.) 217-254 (American Mathematical Society, Providence, RI, 1999).

Acknowledgements. We thank E. Winfree and A. Carbone for valuable discussions. This work has been supported by grants from DARPA and the National Science Foundation to J.H.R. and N.C.S.; ONR, USAF, NSF and NIH grants to N.C.S.; and NSF and ARO grants to J.H.R.