nature
biotechnology

A deoxyribozyme-based molecular automaton

Milan N Stojanovic[1] & Darko Stefanovic[2]

We describe a molecular automaton, called MAYA, which encodes a version of the game of tic-tac-toe and interactively competes against a human opponent. The automaton is a Boolean network of deoxyribozymes that incorporates 23 molecular-scale logic gates and one constitutively active deoxyribozyme arrayed in nine wells (3×3) corresponding to the game board. To make a move, MAYA carries out an analysis of the input oligonucleotide keyed to a particular move by the human opponent and indicates a move by fluorescence signaling in a response well. The cycle of human player input and automaton response continues until there is a draw or a victory for the automaton. The automaton cannot be defeated because it implements a perfect strategy.

Various approaches to molecular computation[1–4] culminated recently in the construction of the first programmable molecular automata[5,6]. These successes represent initial steps in the development of molecular devices capable of interactive analysis and response to serial inputs. On the cellular level, synthetic circuits of interacting biomolecules[7–11] may lead to the engineering of new and practical cellular behaviors, as recently demonstrated through the construction of phenotypic characteristics resembling binary digital (that is, Boolean) circuits[10]. We have recently described[12,13] nucleic acid catalysts[14] modulated by up to two oligonucleotides as allosteric regulators[15]. These constructs are capable of carrying out only the simplest Boolean calculations, but our construction of a network capable of simple arithmetic operations[16] and our ability to arrange several gates around a common substrate[17] suggested the possibility that circuits of highly regulated deoxyribozymes could support complex molecular decision trees. Here we report three-input deoxyribozyme-based logic gates i_AANDi$_B$ANDNOTi$_C$ and use them in a bottom-up approach as building blocks to construct a solution-phase Boolean molecular network, called MAYA, which plays a dynamic game of tic-tac-toe against a human opponent and never loses.

Its straightforward game tree (the diagram of decisions that can be made by the players) has made tic-tac-toe a classic choice for the development of new computational paradigms. Charles Babbage conceptualized a tic-tac-toe automaton while working on his 'analytical engine' in the 1840s, and the first reported game coded on an electronic computer was tic-tac-toe on the EDSAC at Cambridge in 1949. This game is an example of two-player, three-in-a-row games, which are played in almost all civilizations and are among the oldest of recorded games. Tic-tac-toe is a game of 'perfect information'; if it is played optimally, the player moving first is guaranteed either a win or a draw. We considered that the construction of an automaton playing this game would be an appropriate challenge for deoxyribozyme circuits and turned our attention to encoding it in its simplest form: the automaton plays first and the first response of the human opponent is limited to a specific corner or a specific side move.

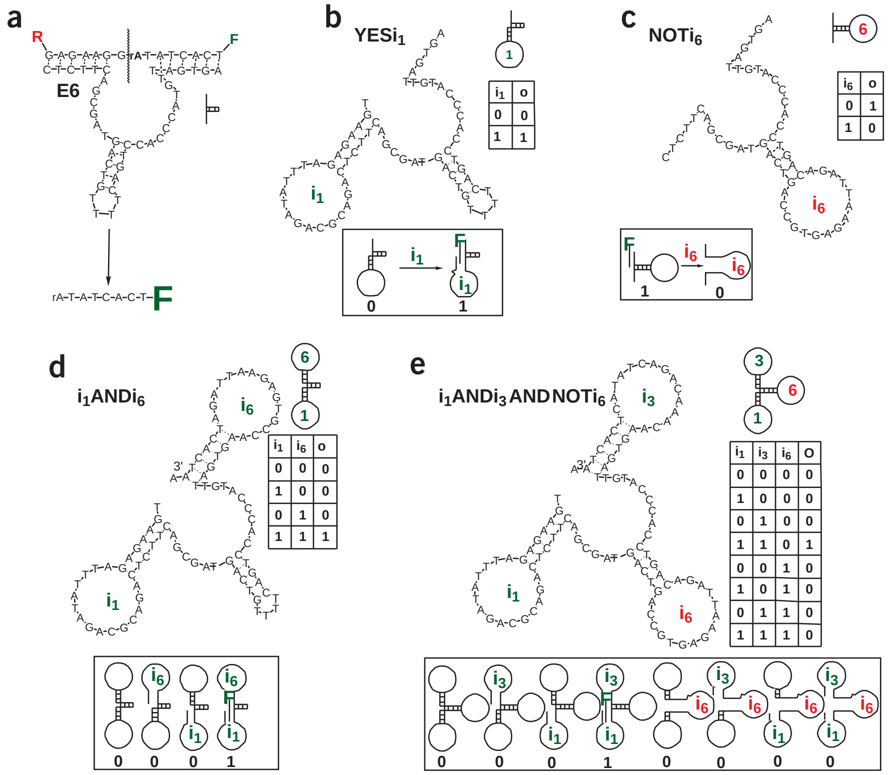
RESULTS

Constituent gates and inputs of the automaton

We constructed deoxyribozyme-based logic gates by modular design[15]. One module, consisting of the deoxyribozyme E6[18] (a DNA-based RNAse that cleaves a fluorogenic substrate)[19,20] (**Fig. 1a**) is combined with a stem loop module, which undergoes a conformational change (stem opening) upon binding of the oligonucleotide complementary to its loop. Modifying the core of E6 with the stem loop at either end of the substrate recognition domain yields detector or YES gates (that is, catalytic molecular beacons[12]) (**Fig. 1b**). The stem loop effectively blocks the access of the substrate to the deoxyribozyme module. Therefore, YES gates can exist in two states, one of which is active; they are activated by the addition of a complementary oligonucleotide that relieves the inhibition conferred by the stem. Similarly, addition of a stem-loop module to the catalytic core results in two-state inverter or NOT[13] gates (NOTi$_6$ in **Fig. 1c**); opening of the stem in the presence of an oligonucleotide complementary to the loop distorts the catalytic core and effectively inhibits the fluorogenic reaction. In addition we devised AND[13] gates by adding stem loops at both ends of the substrate recognition domains. These gates exist in four states, one of which is active; they are preferentially activated in the presence of two input oligonucleotides (i_1ANDi$_6$ in **Fig. 1d**). Because individual games of tic-tac-toe can have as many as four moves by the human player (and thus four inputs to the automaton), we needed a logic gate with the capacity to interpret a higher number of oligonucleotides, ideally one substrate and as many as four regulators. We proceeded to construct three-input deoxyribozyme-based i_AANDi$_B$ANDNOTi$_C$ gates (sometimes called INHIBIT[21] gates) with eight possible states, one of which is active. A NOT module was combined with an AND module to yield this gate, which is actuated if and only if two inputs are provided, but not the third, exemplified by i_1ANDi$_3$ANDNOTi$_6$ in **Figure 1e**. We describe below how the logic representation of a tic-tac-toe strategy was transformed so that such three-input gates sufficed.

[1]Division of Clinical Pharmacology and Experimental Therapeutics, Department of Medicine, Columbia University, Box 84, 650W 168th Street, New York, New York 10032, USA. [2]Department of Computer Science, Farris Engineering Center #157, University of New Mexico, Albuquerque, New Mexico 87131, USA. Correspondence should be addressed to M.N.S. (mns18@columbia.edu).

Published online 17 August 2003; doi:10.1038/nbt862



i1	o
0	0
1	1

i6	o
0	1
1	0

i1	i6	o
0	0	0
1	0	0
0	1	0
1	1	1

i1	i3	i6	O
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	0

Figure 1 Basic gate structures, derived from allosterically regulated deoxyribozyme E6. (**a**) Fluorogenic substrate cleavage by active deoxyribozyme core E6 produces an increase in fluorescence by fluorescein (F) as tetramethylrhodamine (R) quencher is separated (cf. refs. 2 and 3). (**b**) Molecular-scale YES (for example, i_1 or sensor i_1) gate together with a schematic representation of two states of this gate. (**c**) NOT gate (for example, NOTi_6) together with a schematic representation of two states of this gate; this gate is used in combination with AND gate to construct ANDANDNOT. (**d**) AND (for example, i_1ANDi_6) together with a schematic representation of four potential states of this gate. (**e**) ANDANDNOT gate (for example, i_1ANDi_3ANDNOTi_6, the truth table as shown) gates with one, two and three stem loops complementary to each of the input oligonucleotides, together with eight possible states of this gate. Binding of input oligonucleotides to loops in YES gates and AND gates causes an increase in enzymatic activity (green). Note that an ANDANDNOT gate is equivalent to an AND gate with an additional NOT loop; binding of oligonucleotides to this loop causes inhibition of enzymatic activity (red).

We attempted to optimize all gates used in MAYA to achieve a ratio >10:1 between the initial rates of cleavage in the active vs. inactive states. Furthermore, the achieved rates were sufficient to detect fluorogenic activity within 15 min from the time the activating inputs were added, at a gate concentration of 250 nM and a tenfold excess of substrates. To fulfill these criteria, we had to shorten the substrate recognition region of the deoxyribozyme module and extend the stems of the inhibitory YES loops that we used previously[13], eventually arriving at the structures shown in **Figure 1**. We note, however, that quantitative characteristics of gates may vary for each individual gate depending on the structures of input oligonucleotides and the concentrations of individual components of the reaction.

Next, we arbitrarily chose eight input oligonucleotides from a computer-generated library of >10,000 15-mers without strong secondary structures and with intermediate GC content (**Table 1**). Sequences complementary to each of the selected oligonucleotides were individually inserted into the loops of deoxyribozyme-based logic gates. Using mfold Server[22], we eliminated those inserts that favored alternative secondary structures (that is, structures different from those shown in **Figure 1**). This procedure was repeated until we had a sufficient number of gates to implement the game.

Technology mapping

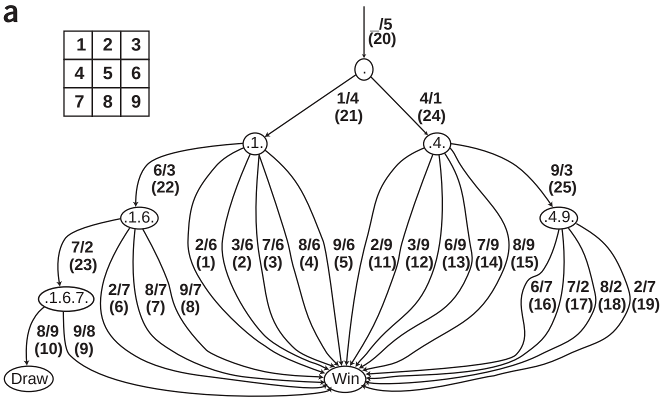
The game of tic-tac-toe is played on a 3×3 grid of nine squares and is broken down into turns. Each player is assigned a mark, either an X or an O. Player X goes first. During a turn, a player writes the assigned mark in an empty square. The game ends in victory for the first player to achieve three marks in a straight line, horizontally, vertically or diagonally; or in a draw if neither player achieves a straight line before the board is full. The chosen strategy, or game tree, for tic-tac-toe, diagrammed as a Mealy automaton[23], is shown in **Figure 2a**. We chose to simplify this first demonstration by having the automaton move first and by symmetry-pruning the response to that move. Thus, the first automaton move is into the center (square 5), and the first move of the human opponent is either into square 1 (corner move) or into square 4 (side move). From this point forward, we make no further restrictions, other than that each move of the human must be legal. Therefore, the game tree has two major branches, with ten and nine games each, corresponding to the human's choice of the corner or the side as the first move. Of the 19 games, 10 end in victory for the automaton after two moves of the human, 7 after three moves, 1 after four moves, and 1 game is a draw.

From the edges of the game tree, the desired mapping of human inputs to automaton outputs can be read off as Boolean formulas,

Table 1 Input oligonucleotides i_1 to i_4 and i_6 to i_9, corresponding to the moves by the human opponent

Move	Input oligonucleotides
i_1	TCT GCG TCT ATA AAT
i_2	ATC GTA TGT TGT TCA
i_3	GTA TAG TCT GTT TGT
i_4	G TAA GTG CTC AAA TGT C
i_6	G TCT AAT TCT CAC GGT C
i_7	TAG TCT GTG TGT TGT
i_8	TCT ATA TGA GCG TAA
i_9	TGT CCA TCT AAA TCC

A move by the human to a particular square entails addition of the keyed input oligonucleotide to all wells of the matrix. The input has no effect in some wells, actively inhibits the gates of other wells, but in every case will fully activate the gate in a single well. The gate fully activated corresponds to the automaton response. Inputs i_4 and i_6 were empirically increased in size to improve their inhibitory properties.



b

$$o_1 = \underbrace{i_4}_{\text{edge (24)}}$$

$$o_2 = \underbrace{(i_6 \wedge i_7 \wedge \neg i_2)}_{\text{edge (23)}} \vee \underbrace{(i_7 \wedge i_9 \wedge \neg i_1)}_{\text{edge (17)}} \vee \underbrace{(i_8 \wedge i_9 \wedge \neg i_1)}_{\text{edge (18)}}$$

$$o_3 = \underbrace{(i_1 \wedge i_6)}_{\text{edge (22)}} \vee \underbrace{(i_4 \wedge i_9)}_{\text{edge (25)}}$$

$$o_4 = \underbrace{i_1}_{\text{edge (21)}}$$

$$o_5 = \underbrace{1}_{\text{edge (20)}}$$

$$o_6 = \underbrace{(i_1 \wedge i_2 \wedge \neg i_6)}_{\text{edge (1)}} \vee \underbrace{(i_1 \wedge i_3 \wedge \neg i_6)}_{\text{edge (2)}} \vee \underbrace{(i_1 \wedge i_7 \wedge \neg i_6)}_{\text{edge (3)}} \vee \underbrace{(i_1 \wedge i_8 \wedge \neg i_6)}_{\text{edge (4)}} \vee \underbrace{(i_1 \wedge i_9 \wedge \neg i_6)}_{\text{edge (5)}}$$

$$o_7 = \underbrace{(i_2 \wedge i_6 \wedge \neg i_7)}_{\text{edge (6)}} \vee \underbrace{(i_6 \wedge i_8 \wedge \neg i_7)}_{\text{edge (7)}} \vee \underbrace{(i_6 \wedge i_9 \wedge \neg i_7)}_{\text{edges (8) and (16)}} \vee \underbrace{(i_9 \wedge i_2 \wedge \neg i_1)}_{\text{edge (19)}}$$

$$o_8 = \underbrace{i_9 \wedge i_7 \wedge \neg i_4}_{\text{edge (9)}}$$

$$o_9 = \underbrace{(i_7 \wedge i_8 \wedge \neg i_4)}_{\text{edge (10)}} \vee \underbrace{(i_4 \wedge i_2 \wedge \neg i_9)}_{\text{edge (11)}} \vee \underbrace{(i_4 \wedge i_3 \wedge \neg i_9)}_{\text{edge (12)}} \vee \underbrace{(i_4 \wedge i_6 \wedge \neg i_9)}_{\text{edge (13)}} \vee \underbrace{(i_4 \wedge i_7 \wedge \neg i_9)}_{\text{edge (14)}} \vee \underbrace{(i_4 \wedge i_8 \wedge \neg i_9)}_{\text{edge (15)}}$$

Figure 2 Game tree for the symmetry-pruned tic-tac-toe and the corresponding Boolean formulas. (**a**) Game tree for the automaton-plays-first, symmetry-pruned game of tic-tac-toe. Each node is labeled according to the inputs seen on the path to it. Each edge is labeled i/o(n), where o is the output that is activated on input i, and n is the edge (gate) identifier. The numbering of gates is arbitrary. (**b**) The final Boolean formulas computing automaton outputs; individual gates are identified underneath each conjunction.

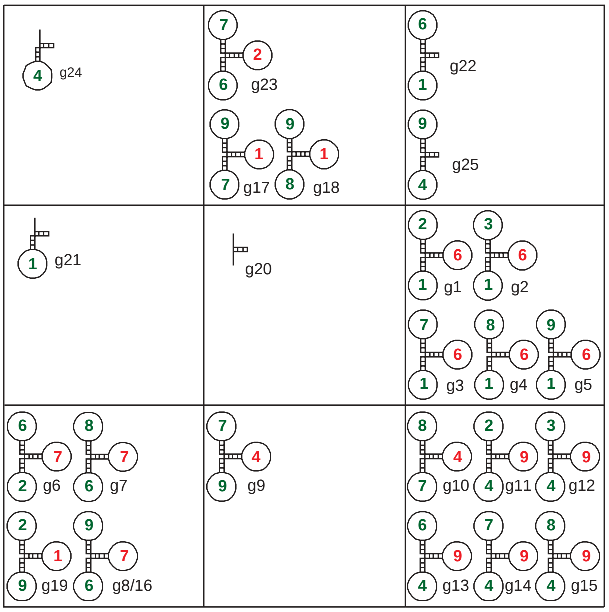


Figure 3 Schematic representation of the distribution of gates in the individual wells of the MAYA automaton. Note that all gates in a given well carry out an implicit OR function because they cleave the same substrate. A constitutively active deoxyribozyme signaling the first move by the automaton is in well 5, a YES gate is in each of well 1 and well 4, two AND gates are in well 3, whereas in the remaining wells we have various ANDANDNOT gates. Inhibitory inputs are marked red, whereas activating inputs are marked green.

collecting together all the edges that give rise to a particular output. Each edge is represented as a conjunction over the set of inputs seen from the root to that edge. Conflicts arising from overlapping sets of inputs that trigger two different outputs were resolved (disambiguated) by introducing negated inputs into the formulas as appropriate. We completed technology mapping by transforming these formulas into a logically equivalent (with respect to the game tree) set of formulas subject to the restriction that each conjunction can have at most three inputs. This allowed us to implement the formulas, by making use of the ANDANDNOT three-input gates but without the need for four-input gates. As a result, all outputs were computed with a single layer of logic (the final OR is implicit in the simultaneous presence in solution of all its AND gate predecessors), and the automaton's 'thinking time' is shortened to a quarter of an hour.

As an example, according to the outlined procedure, output 7, which is activated on edges 6, 7, 8, 16 and 19, was transformed in three steps from $o_7 \leftarrow (i_1 \wedge i_6 \wedge i_2) \mid\mid (i_1 \wedge i_6 \wedge i_8) \mid\mid (i_1 \wedge i_6 \wedge i_9) \mid\mid (i_4 \wedge i_9 \wedge i_6) \mid\mid (i_4 \wedge i_9 \wedge i_2)$ through $o_7 = (i_1 \wedge i_6 \wedge i_2) \vee (i_1 \wedge i_6 \wedge i_8 \wedge \neg i_7) \vee (i_1 \wedge i_6 \wedge i_9 \wedge \neg i_7) \vee (i_4 \wedge i_9 \wedge i_6) \vee (i_4 \wedge i_9 \wedge i_2)$ and $o_7 = (i_2 \wedge i_6 \wedge \neg i_4) \vee (i_6 \wedge i_8 \wedge \neg i_7) \vee (i_6 \wedge i_9 \wedge \neg i_7) \vee (i_9 \wedge i_2 \wedge \neg i_1)$ to the final form $o_7 = (i_2 \wedge i_6 \wedge \neg i_7) \vee (i_6 \wedge i_8 \wedge \neg i_7) \vee (i_6 \wedge i_9 \wedge \neg i_7) \vee (i_9 \wedge i_2 \wedge \neg i_1)$. This procedure is described in full detail in **Supplementary Methods** online.

Our initial attempts to play the game led us to further empirical improvements in structures and distribution of gates to minimize responses in the wrong wells (as a result of incompletely digital behavior) until we finally arrived at the Boolean formulas shown in **Figure 2b**. Each final formula specified a set of gates that act upon a common substrate in a single well corresponding to a board square. The additional empirical improvements included the following. First, inputs i_4 and i_6 were extended to 17 nucleotides, because we noticed that the original 15-mers were not able fully to open the stem loop and to inhibit appropriate NOT gates. This result indicates that the set of criteria used to compile our original oligonucleotide library was insufficient to eliminate 15-mer inputs with unsatisfactory digital behavior. Second, i_9ANDi$_7$ANDNOTi$_4$ in well 8 had to be extended by one additional C at the 5′ end to suppress gate activity in the presence of inputs i_1, i_7 and i_8. Third, concentrations of YES gates in wells 1 and 4 were increased to 250 nM from the initially tested 200 nM, whereas concentrations of all other gates were decreased to 150 nM to decrease cumulative activities of multiple semiactivated gates in some wells in the presence of i_1 or i_4. Fourth, for a nonobvious reason, one of the initial gates in well 2 (i_6ANDi$_7$ANDNOTi$_4$) was active in the presence of inputs i_1, i_6 and i_2 in initial tests, and we replaced it with i_6ANDi$_7$ANDNOTi$_2$. The need for so many empirical improvements indicates that, despite the apparent success of modular design, we are not yet able to directly translate computer-predicted structures into functioning automata. Other improvements, such as shortening the stem in original deoxyribozymes and extending inhibitory overlaps at 5′ and 3′ ends, were tested at the level of individual gates but did not offer substantial advantages.

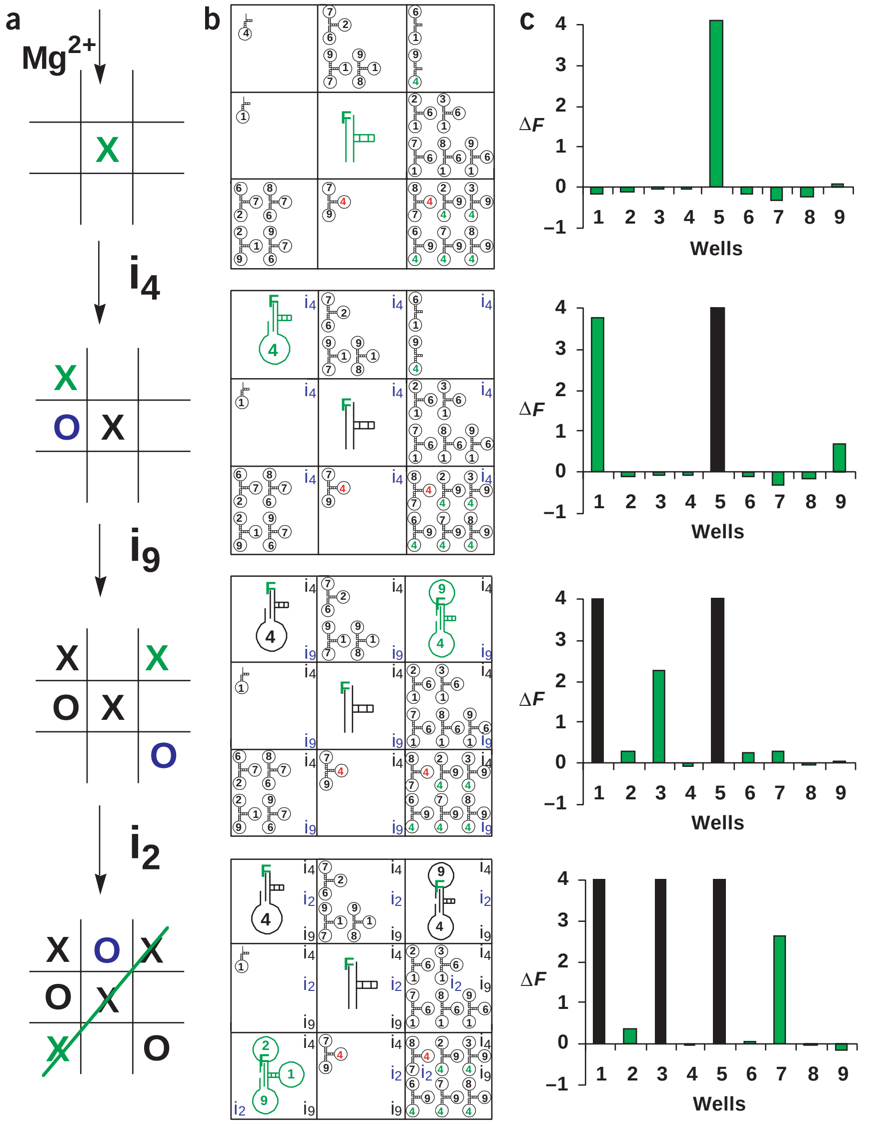


Figure 4 A representative game. Human's first move is into the side field; the human is left without a good strategy and loses after three moves. Each move is represented in three ways from left to right. (**a**) Tic-tac-toe board representation, in which the automaton's new moves are represented by green x marks, previous moves by black x marks, new human moves by blue o marks and previous human moves by black o marks. Input oligonucleotides triggering these moves are also indicated on the vertical arrows connecting successive board situations. (**b**) Representation of the contents of the wells, in which we show automaton moves as large green gates complexed with substrate and human moves as blue-colored inputs in all wells. New automaton moves are colored green, previous moves black; new human moves are colored blue and previous moves black. (**c**) Bar graph representation of actual values of fluorescence in all wells 15 min after the addition of input oligonucleotides to all wells. Δ*F* is in tens of thousands. New moves are marked by green bars, whereas previous moves are indicated by maximized bars of black color.

Implementation and testing of MAYA

The final formulas were implemented in our laboratory as a Molecular Array of YES and ANDANDNOT gates (MAYA) in the 3×3 wells of a 386-well plate, with 24 deoxyribozymes in nine wells (**Fig. 3**). The automaton consists of 2 YES gates, 2 AND gates and 19 ANDANDNOT gates, distributed to carry out eight Boolean calculations in eight wells, and one permanently active deoxyribozyme in the center square.

The game is initiated by adding the required cofactor, Mg^{2+} ions, to all wells. With MAYA playing first in the center by design, there are exactly eight possible inputs by the human opponent, represented by oligonucleotides 1–4 and 6–9 (**Table 1**), keyed to each of the remaining eight squares. To make a move, the human adds the appropriate input oligonucleotide to all wells, and this addition elicits an output response (greatest increase in fluorescence) by the automaton in exactly one well according to the Boolean formulas in **Figure 2b**. The output interface is a fluorescence plate reader, and we followed the changes in fluorescence upon addition of input oligonucleotides. MAYA took 15 min to report each move unambiguously.

The game tree has 19 possible games, and we have played all 19 against MAYA at least four times each. In every case, MAYA performed to specification and was never defeated—as expected, because it implements a perfect strategy. We also varied the concentrations of gates and the lengths of inhibiting stems in gates; we varied the sizes of inputs, either following real-time fluorescence changes or taking aliquots and changing the structures of quenchers in the substrates. Although these variations changed the duration and the quality of the responses, we could always detect the automaton's move unambiguously.

We present here the details of the two longest games in each of the main branches of the game tree (**Figs. 4** and **5**), and in our **Supplementary Figures 1–5** we give examples of the remaining 17 games. In all these games we defined moves as an increase in fluorescence (Δ*F*, λex = 480 nm, λem = 520 nm) by >15,000 relative units in a new well within 15 min upon input addition. To observe the kinetics of reactions and to confirm that the move is being reported by the catalytic reaction and not a conformational change, we followed the fluorescence changes for as long as 45 min following input addition, and a typical move was characterized by an increase of >70,000 relative units. Nonresponse wells were typically characterized by at least five-fold less increase in fluorescence, if any.

In the two games described here, after addition of Mg^{2+} to all wells, the human player (M.N.S) takes a fluorescent readout and finds out that the automaton played its first move in the center through the rapid increase of fluorescence intensity in well 5 (**Fig. 4**, first row). In the first game (**Fig. 4**) the human makes a strategic error and plays the side move. Consequently, the human is left without a draw strategy and the automaton perfectly executes a series of moves that culminate with a three-in-a-row regardless of the human's moves. Individual moves are as follows: (i) the human adds i_4 to all wells as his first move, resulting in activation of the YESi_4 gate in well 1 (**Fig. 4**, second row). (ii) To block diagonal three-in-a-row by the automaton, the human adds i_9 to all wells, thus activating only the i_4ANDi_9 gate in well

3 (**Fig. 4**, third row). We note that any other input oligonucleotide would immediately activate one of the gates in well 9, leading to MAYA's victory. As a result of the human's move, the automaton opens a double opportunity to close three-in-a-row, and the human is left without good choices. Namely, any input oligonucleotide will activate one of the gates in either well 2 or well 7. (iii) The human chooses to add the i_2 oligonucleotide to all wells, blocking diagonal three-in-a-row, but this activates the i_2ANDi$_9$ANDNOTi$_1$ gate in well 7, resulting in MAYA's victory (**Fig. 4**, fourth row). Another three games with the identical first two moves and five games with the identical first move are given in **Supplementary Figures 1** and **4** online. They attest to the reproducibility of MAYA's response to identical inputs. In the second representative game, the human started with a corner move (**Fig. 5**). Because neither player was able to establish three-in-a-row, the game ended in a draw. Once again, another game with the identical first three moves, as well as three games with the identical first two moves and five games with the identical first move, are given in **Supplementary Figures 2**, 3 and 5.

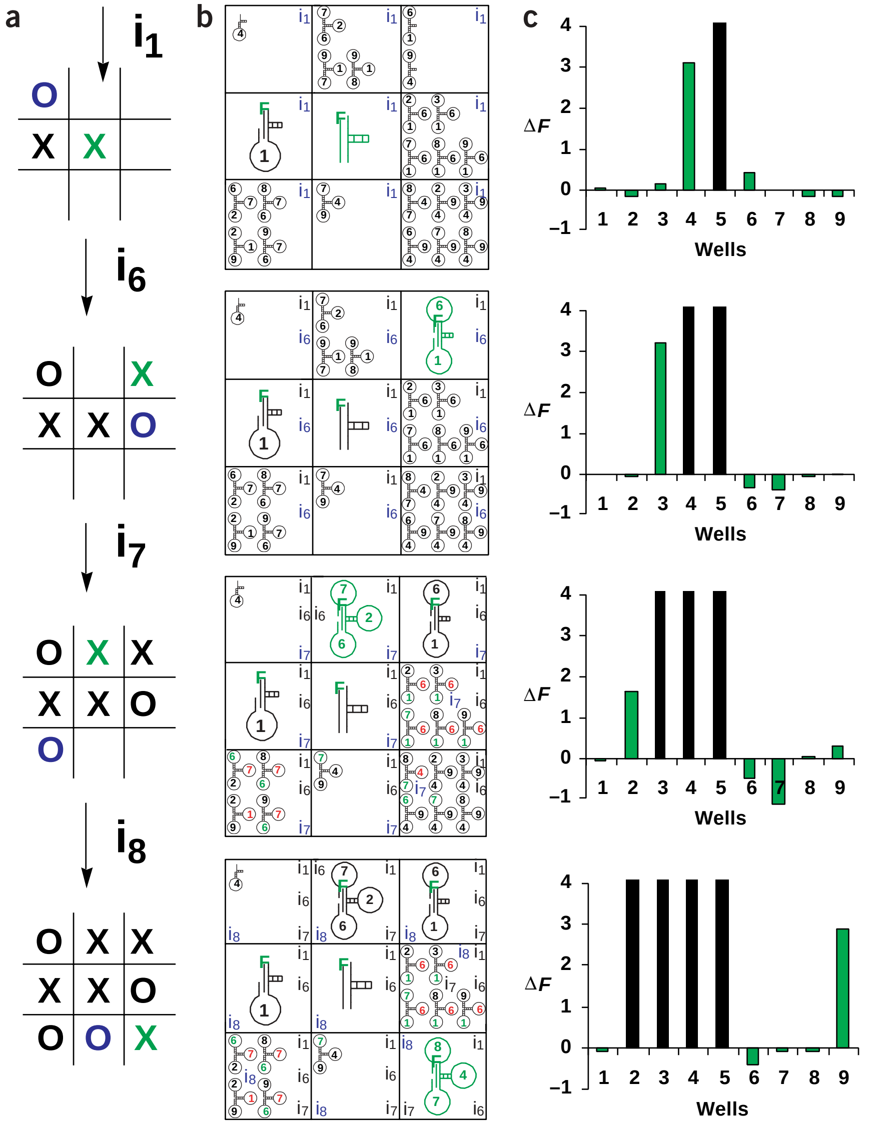


Figure 5 A second representative game. Human's first move is into the corner field; both the human and the automaton play perfectly and the game ends in a draw. Each move is represented as described in **Figure 4**.

DISCUSSION

The deoxyribozyme-based automaton MAYA precisely executes a linear program encoded through a spatial distribution of nucleic acid catalysts behaving as logic gates. In a total of >100 games played, we detected no erroneous moves by MAYA; in other words, under all tested conditions the fully activated gate was always more fluorogenically active than an assembly of several partially active gates. However, we detected some increase in nonresponse wells; for instance, upon addition of the i_1 input we noticed a strong and expected increase in well 4 and a detectable increase in well 9 from the five partially activated ANDANDNOT gates. This observation indicates that error propagation in complex networks may in the future lead to nondigital behavior. The digital behavior of individual gates can be perfected one gate at a time by increasing the lengths of inhibitory stems in the presence of specific inputs. This, however, suppresses the reaction rate in the presence of activating oligonucleotide, apparently as a result of the incomplete stem opening, and it leads to an increase in MAYA's 'thinking time'. We have recently successfully decreased the response time by adding Zn^{2+} ions to the solution of gates[16]; however, our eventual practical applications are going to be *in vivo* and intracellular, and we therefore prefer to use only biocompatible components in our circuits. If required for future applications, new gates with improved reaction rates and digital behavior at physiological conditions could be generated through selection procedures.

It is instructive to compare MAYA with the other published molecular approaches[2–5,24–28] to computation in solution using DNA. MAYA does not take advantage of massively parallel computation, used in the Adleman–Lipton paradigm[4], nor does MAYA use the power of DNA to form well-defined supramolecular complexes, as in the Seeman–Winfree paradigm[24,28]. MAYA is unique in that it consists of individual biomolecular building blocks that behave analogously to digital logic circuits used in electronic computers or to enzymes in metabolic circuits. In analogy to approaches in electrical engineering, individual components are arrayed in feed-forward circuits carrying out Boolean calculations without human interface–operated steps.

Deoxyribozyme-based computation approaches are unlikely ever to compete with silicon-based computing in solving computationally complex problems. However, these circuits have the potential to be directly interfaced with molecular sensors and therapeutic agents[3], and Boolean networks of nucleic acid enzymes may be useful for controlling autonomous therapeutic and diagnostic molecular-scale devices.

No assembly of biomolecules that can autonomously play a dynamic game has been described before the present report.

Oligonucleotide hybridization has been used to solve the 'knight problem' from chess on a 3×3 grid[29], but this approach was based on a static combinatorial search, and individual logic operations were controlled through human-operated interfaces. In MAYA, the logic gates are organized in parallel. However, the substrate (or the output) of one gate can also be the input of another, paving the way to automata that can carry out even more complex tasks, including those encoded in nonlinear programs. For example, such complex networks could be used in synthetic biology[30] to learn more about genetic, regulatory and metabolic systems networks that culminate in complex decisions such as division, differentiation and movement. Another application, using stem loops that control ribozymes expressed in cells, might be *in vivo* computation networks and cells with engineered properties.

METHODS

Materials and procedures. All oligonucleotides were custom-made and purified by either IDT DNA (Coralville, IA, USA) or TriLink Biotechnologies, Inc. (San Diego, CA, USA) and used as received. All structures corresponded to generic formulas defined in **Figure 1**, except for o_8, in which we had to increase the size of the corresponding overlapping stems by one nucleotide, so as to suppress the background cleavage reaction in the presence of i_9. In the final practical implementation, each well contained a solution of gates producing the output keyed to that well at individual concentrations of 150 nM, except YES gates with concentrations of 250 nM, fluorogenic substrate at 2 μM concentrations and 20 mM Mg^{2+} ions in HEPES buffer (pH 7.4, 1 M NaCl). Inputs were added at a concentration of 2.5 μM to each well, and fluorescence was followed for 45 min.

Available online are (i) an analysis of the tic-tac-toe game, derivation of the logic gate implementation and mapping to deoxyribozyme-based logic gate technology (**Supplementary Methods**) and (ii) the remaining games played between MAYA and M.N.S (**Supplementary Figures 1–5**).

Note: Supplementary information is available on the Nature Biotechnology website.

ACKNOWLEDGMENTS

We acknowledge support by the National Aeronautics Space Agency (NAS2-02039 to M.N.S), the National Institute of Health (NIBIB, RO1 EB000675-1 to M.N.S), the National Science Foundation (EIA-0218262 to M.N.S and D.S, and CCR-0219587 and CCR-0085792 to D.S.), Microsoft Research (D.S.) and Hewlett-Packard (gift 88425.1 to D.S). M.N.S. acknowledges the support by Searle Scholars Program (03-C-103). We particularly thank Donald W. Landry for continued support of this project. We are also grateful to Deepak Kapur, Stephanie Forrest, Robert Veroff, Cristopher Moore, Lance Williams, Benjamin Andrews, and Clint Morgan for helpful discussions, suggestions and support.

COMPETING INTERESTS STATEMENT

The authors declare that they have no competing financial interests.

Received 2 January; accepted 16 June 2003
Published online at http://www.nature.com/naturebiotechnology/

1. Ball, P. Chemistry meets computing. *Nature* **406**, 118–120 (2000).
2. Ruben, A.J. & Landweber, L.F. The past, present and future of molecular computing. *Nature Rev. Mol. Cell Biol.* **1**, 69–72 (2000).
3. Cox, J.C. & Ellington, A.D. DNA computation function. *Curr. Biol.* **11**, R336 (2001).
4. Braich, R.S., Chelyapov, N., Johnson, C., Rothemund, P.W.K. & Adleman, L. Solution of a 20-variable 3-SAT on a DNA computer science. *Science* **296**, 499–502 (2002).
5. Benenson, Y. *et al.* Programmable and autonomous computing machine made of biomolecules. *Nature* **414**, 430–434 (2001).
6. Benenson, Y., Adar, R., Paz-Elizur, T., Livneh, Z. & Shapiro, E. DNA molecule provides a computing machine with both data and fuel. *Proc. Natl. Acad. Sci. USA* **100**, 2191–2196 (2003).
7. Elowitz, M.B. & Leibler, S. A synthetic oscillatory network of transcriptional regulators. *Nature* **403**, 335–338 (2000).
8. Gardner, T.S., Cantor, C.R. & Collins, J.J. Construction of a genetic toggle switch in *Escherichia coli. Nature* **403**, 339–342 (2000).
9. Guet, C.C., Elowitz, M.B., Hsing, W. & Leibler, S. Combinatorial synthesis of genetic networks. *Science* **296**, 1466–1470 (2002).
10. Atkinson, M.R., Savageau, M.A., Myers, J.T. & Ninfa, A.J. Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli. Cell* **113**, 597–607 (2003).
11. Yokobayashi, Y., Weiss, R. & Arnold, F.H. Directed evolution of a genetic circuit. *Proc. Natl. Acad. Sci. USA* **99**, 16587–16591 (2002).
12. Stojanovic, M.N., de Prada, P. & Landry, D.W. Catalytic molecular beacons. *ChemBioChem.* **2**, 411–415 (2001).
13. Stojanovic, M.N., Mitchell, T.E. & Stefanovic, D. Deoxyribozyme-based logic gates. *J. Am. Chem. Soc.* **124**, 123–125 (2002).
14. Wilson, D. & Szostack, J.W. *In vitro* selection of functional nucleic acids. *Annu. Rev. Biochem.* **68**, 611–647 (1999).
15. Soukup, G.E. & Breaker R.R. Engineering precision RNA molecular switches. *Proc. Natl. Acad. Sci. USA* **96**, 3584–3589 (1999).
16. Stojanovic, M.N. & Stefanovic, D. Deoxyribozyme-based half adder. *J. Am. Chem. Soc.* **125**, 6673–6676 (2003).
17. Stojanovic, M.N., Nikic, D.B. & Stefanovic, D. Implicit-OR tiling of deoxyribozymes: construction of molecular scale OR, NAND and four-input logic gates. *J. Serb. Chem. Soc.* **68**, 321–326 (2003).
18. Breaker, R.R. & Joyce, G.F. A DNA enzyme with Mg^{2+}-dependent RNA phosphoesterase activity. *Chem. Biol.* **2**, 655–660 (1995).
19. Jenne, A., Gmelin, W., Raffler, N. & Famulok, M. Real-time characterization of ribozymes by fluorescence resonance energy transfer (FRET), *Angew. Chem. Int. Edn.* **38** (9), 1300–1303 (1999).
20. Stojanovic, M.N., de Prada, P. & Landry, D.W. Homogeneous assays based on deoxyribozyme catalysis. *Nucleic Acids Res.* **28**, 2915–2918 (2000).
21. de Silva, A.P. *et al.* Integration of logic functions and sequential operation of gates at the molecular scale. *J. Am. Chem. Soc.* **121**, 1393–1394 (1999).
22. Mathews, D.H. *et al.* An updated recursive algorithm for RNA secondary structure prediction with improved free energy parameters *Am. Chem. Soc. Symp. Series* **682**, 246–257 (1998).
23. Mealy, G.H. A method for synthesizing sequential circuits. *Bell Systems Tech. J.* **34**(5), 1045–1079 (1955).
24. Winfree, E., Liu, F.R., Wenzler, L.A. & Seeman, N.C. Design and self-assembly of two-dimensional DNA crystals. *Nature* **394**, 539–544 (1998).
25. Sakamoto, K. *et al.* State transitions by molecules. *BioSystems* **52**, 81–91 (1999).
26. Hartemink, A.J., Gifford, D.K. & Khodor, J. Automated constraint-based nucleotide sequence selection for DNA computation. *BioSystems* **52**, 227–235 (1999).
27. Khodor, J. & Gifford, D.K. Design and implementation of computational systems based on programmed mutagenesis. *BioSystems* **52**, 93–97 (1999).
28. Mao, C., LeBean, T.H., Reif, J.H. & Seeman, N.C. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature* **407**, 493–496 (2000).
29. Faulhammer, D., Cukras, A.R., Lipton, R.J. & Landweber, L.F. Molecular computation: RNA solutions to chess problems. *Proc. Nat. Acad. Sci. USA* **97**, 1385–1389 (2000).
30. Benner, S.A. Synthetic biology: act natural. *Nature* **421**, 118 (2003).