An autonomous molecular computer for logical control of gene expression

Yaakov Benenson[1,2], Binyamin Gil[2], Uri Ben-Dor[1], Rivka Adar[2] & Ehud Shapiro[1,2]

[1]*Department of Computer Science and Applied Mathematics and* [2]*Department of Biological Chemistry, Weizmann Institute of Science, Rehovot 76100, Israel*

Early biomolecular computer research focused on laboratory-scale, human-operated computers for complex computational problems[1–7]. Recently, simple molecular-scale autonomous programmable computers were demonstrated[8–15] allowing both input and output information to be in molecular form. Such computers, using biological molecules as input data and biologically active molecules as outputs, could produce a system for 'logical' control of biological processes. Here we describe an autonomous biomolecular computer that, at least *in vitro*, logically analyses the levels of messenger RNA species, and in response produces a molecule capable of affecting levels of gene expression. The computer operates at a concentration of close to a trillion computers per microlitre and consists of three programmable modules: a computation module, that is, a stochastic molecular automaton[12–17]; an input module, by which specific mRNA levels or point mutations regulate software molecule concentrations, and hence automaton transition probabilities; and an output module, capable of controlled release of a short single-stranded DNA molecule. This approach might be applied *in vivo* to biochemical sensing, genetic engineering and even medical diagnosis and treatment. As a proof of principle we programmed the computer to identify and analyse mRNA of disease-related genes[18–22] associated with models of small-cell lung cancer and prostate cancer, and to produce a single-stranded DNA molecule modelled after an anticancer drug.

Taking our cue from the terminology of medical treatment, we consider that our molecular computer performs *in vitro* a computational version[23,24] of 'diagnosis'—the identification of a combination of mRNA molecules at specific levels, which in our example is a highly simplified model of cancer—and 'therapy'—production of a biologically active molecule, which in our case is a drug-like single-stranded (ss)DNA with known anticancer activity (Fig. 1a). The computer operation is governed by a 'diagnostic rule' that encodes medical knowledge in simplified form (Fig. 1b). The left-hand side of the rule consists of a list of molecular indicators for a specific disease, and its right-hand side indicates a molecule to be released, which could be a drug for that disease. For example, the diagnostic rule for prostate cancer states[20] that if the genes *PPAP2B* and *GSTP1* are underexpressed and the genes *PIM1* and hepsin (*HPN*) are overexpressed then administer the ssDNA molecule GTTGGTATTGGACATG, which inhibits[25] the synthesis of the protein MDM2 by binding to its mRNA. The computer design is flexible in that any sufficiently long RNA molecule can function as a molecular indicator and any short ssDNA molecule, up to at least 21 nucleotides, can be administered.

The computation module is a molecular automaton[12–15] (Supplementary Fig. S1) that processes such a rule as depicted in Fig. 1c. The automaton has two states: positive (Yes) and negative (No). The computation starts in the positive state and if it ends in that state we call the result 'positive diagnosis', otherwise it is called 'negative diagnosis'. To facilitate rule processing by the automaton, the left-hand side of the diagnostic rule is represented as a string of symbolic indicators, or symbols for short, one for each molecular

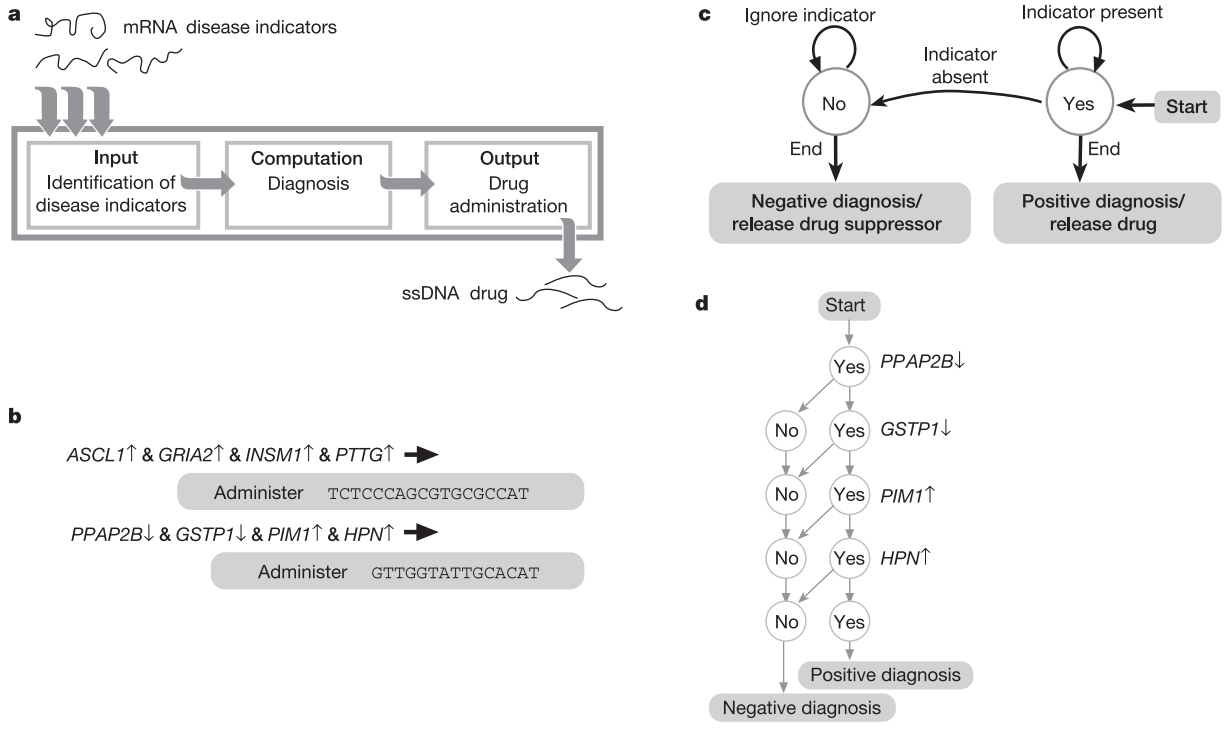

Figure 1 Logical design and logical operation of the molecular computer. **a**, Function and modular organization of the molecular computer. **b**, Example diagnostic rules for simplified models of SCLC[19] and prostate cancer[20], indicating overexpression (↑) or underexpression (↓) of a disease-related gene. The first rule states that if genes *ASCL1*, *GRIA2*, *INSM1* and *PTTG1* are overexpressed then administer the ssDNA molecule TCTCCCAGCGTGCGCCAT (oblimersen), purported to be an antisense therapy drug for SCLC[27]. The second rule states that if the genes *PPAP2B* and *GSTP1* are underexpressed and the genes *PIM1* and *HPN* are overexpressed then administer the ssDNA molecule GTTGGTATTGCACAT, purported to be a drug for prostate cancer[25]. **c**, Transition diagram of the diagnostic automaton. **d**, The computation that diagnoses prostate cancer.

indicator. For example, the string for the prostate cancer rule is *PPAP2B* ↓ *GSTP1* ↓ *PIM1* ↑ *HPN* ↑ . For each symbol the automaton has three types of transitions: positive (Yes → Yes), negative (Yes → No) and neutral (No → No). The automaton processes the string from left to right, one symbol at a time. When processing a symbol in the positive state, the computer takes the positive transition if it determines that the molecular indicator is present and the negative transition, changing to a negative state, otherwise. As the No → Yes transition is not allowed, once the automaton enters the negative state it can use only the neutral transition and thus remains in the negative state for the duration of the computation. The possible computation paths of the automaton processing the prostate cancer diagnostic rule are shown in Fig. 1d.

The molecular automaton is stochastic[14,16] in that it has two competing transitions, positive and negative, for each symbol while in the positive state. A novel molecular mechanism, explained below, regulates the probability of each positive transition by the corresponding molecular indicator, so that the presence of the indicator increases the probability of a positive transition and decreases the probability of its competing negative transition, and vice versa if the indicator is absent. Because the confidence with which the presence or absence of an indicator can be determined is a continuous rather than a discrete parameter, so is the regulation of transition probabilities, the level of which is correlated with this confidence. The resulting stochastic behaviour of the automaton is governed by the confidence in the presence of each indicator, so that the probability of a positive diagnosis is the product of the probabilities of the positive transitions for each of the indicators processed (see Supplementary Note). By changing the ratio between positive and negative transitions for a particular indicator we can have fine control over the sensitivity of diagnosis to the presence of that indicator. We note our unusual use of automaton components: its formal input (the diagnostic rule to be processed) functions in our application like a program, and its formal program (the software molecules) functions in our application as part of the input module, detecting the presence of molecular indicators.

Instead of releasing an output molecule on positive diagnosis and doing nothing on negative diagnosis, we opted to release a biologically active molecule (for example, a drug) on positive diagnosis and its suppressor molecule on negative diagnosis. This allows fine control over the diagnosis confidence threshold beyond which an active drug is administered. Rather than using a single automaton for both tasks (which cannot be done with our current design), we implement this by using two types of automata: one that releases a drug molecule on positive diagnosis, and another that releases a drug-suppressor molecule on negative diagnosis. The ratio between the drug and drug-suppressor molecules released by a population of automata of these two types determines the final active drug concentration.

The molecular computer realizing this logical design consists of diagnostic molecules that encode diagnostic rules (Fig. 2a; see also Supplementary Fig. S2); transition molecules that realize automaton transitions and are regulated by molecular indicators (Fig. 2b; see also Supplementary Fig. S2); and hardware molecules (the restriction enzyme *Fok*I) (Supplementary Fig. S1).

A diagnostic molecule has a diagnosis moiety and a drug-administration moiety (Fig. 2a; see also Supplementary Fig. S2). The diagnosis moiety realizes each symbol by a unique double-stranded (ds)DNA sequence 7 base pairs (bp) in length. For all symbol-encoding sequences, the first four nucleotides represent the symbol combined with the positive state and nucleotides three to six represent the symbol combined with the negative state. The automaton processes the diagnosis moiety one symbol at a time, determining whether the corresponding indicator is present (Fig. 1c). After all symbols are processed, the computation proceeds to processing the drug-administration moiety. There are two kinds of drug-administration moieties: drug release and drug-suppressor release. Both consist of a double-stranded stem that protects a single-stranded loop containing the drug or the drug suppressor. The stem prevents interactions between the drug, drug suppressor and target mRNA. Its length (three symbols) was determined empirically (data not shown). After positive diagnosis, the stem of a drug-suppressor moiety is cleaved by special Yes-verification transitions and releases the drug; in the case of negative diagnosis it remains intact, protecting the drug (Fig. 2d). After negative diagnosis, the stem of the drug-suppressor release moiety is cleaved by special No-verification transitions and releases the drug suppressor; in the case of positive diagnosis it remains intact, protecting the drug suppressor (Fig. 2d).

The ratio between diagnostic molecules with a drug release moiety and those with a drug-suppressor release moiety need not necessarily be 1:1. For example, if the measured probability of positive diagnosis does not exceed 50% owing to computer design limitations, combining the two types of molecules at a ratio of 2:1 will render an active drug for 1/6 of the computations. Inverting this ratio can compensate for false-positive diagnosis by suppressing drug release below any desired threshold.

The molecules released in our experiments are the antisense ssDNA for MDM2 (GTTGGTATTGCACAT)[25] and the ssDNA molecule having the same sequence as the drug Vitravene[26] (GCGTTTGCTCTTCTTCTTGCG) (Supplementary Data). Similar molecules, such as oblimersen[27] (TCTCCCAGCGTGCGCCAT), could similarly be released. The same method can be used to release short RNA molecules.

For each symbolic indicator, a pair of competing transition molecules (Fig. 2b; see also Supplementary Fig. S2) perform the corresponding molecular indicator verification. Presence of a molecular indicator entails high concentration of the positive transition molecule and low concentration of its competing negative transition molecule and vice versa. This regulation is accomplished by sequence-specific interaction between the indicator and a partially single-stranded fragment of a transition molecule, as follows. A positive transition checking for overexpression is activated by a high concentration of its corresponding mRNA, whereas a positive transition checking for underexpression is inactivated by a high concentration of its corresponding mRNA. The corresponding negative transitions are oppositely regulated by a similar mechanism (Fig. 2b; see also Supplementary Fig. S2). A similar mechanism allows for transition regulation by point mutation (Supplementary Fig. S2). The logical switch between configurations of the transition molecules is similar to the state that switching a DNA fuel molecule causes in a DNA nanoactuator[28]. We also note an alternative approach to sensing biochemical signals, known as 'chemical logic gates'[29,30].

The computer consists of three molecular modules—input (Fig. 2b), computation (Fig. 2a) and output (Fig. 2d)—that interact with the disease-related molecules and with each other via a complex network (Fig. 2). Each molecular computer autonomously performs one diagnosis and drug administration task; multiple tasks can be performed by multiple computers that operate in parallel within the same environment without mutual interference, while sharing the hardware molecules and potentially sharing some or all of the software molecules. All pairs of transition molecules are regulated simultaneously by their respective indicators (Fig. 2b) and perform a stochastic computation over diagnostic molecules to administer a drug upon diagnosis (Fig. 2d).

To provide a successful implementation, the computer must be robust both to imprecision of molecular components and to variations in external parameters. This is achieved by three mechanisms. First, imprecision in transition regulation may be compensated by variation in the relative amounts of the active and inactive transition molecules and by addition of excess ssDNA oligonucleotides that form these transitions. Second, changes in the absolute level at which a molecular indicator should be positively detected

are compensated for by a similar change in absolute concentration of the transition molecules (Supplementary Fig. S3). Third, false-positive or false-negative diagnoses may be compensated for as explained above.

The operation of the computer modules was verified separately and jointly: transition regulation by the input module (Fig. 2b; see also Supplementary Fig. S2) was verified independently (Fig. 3a; see also Supplementary Fig. S3); the input and computation modules

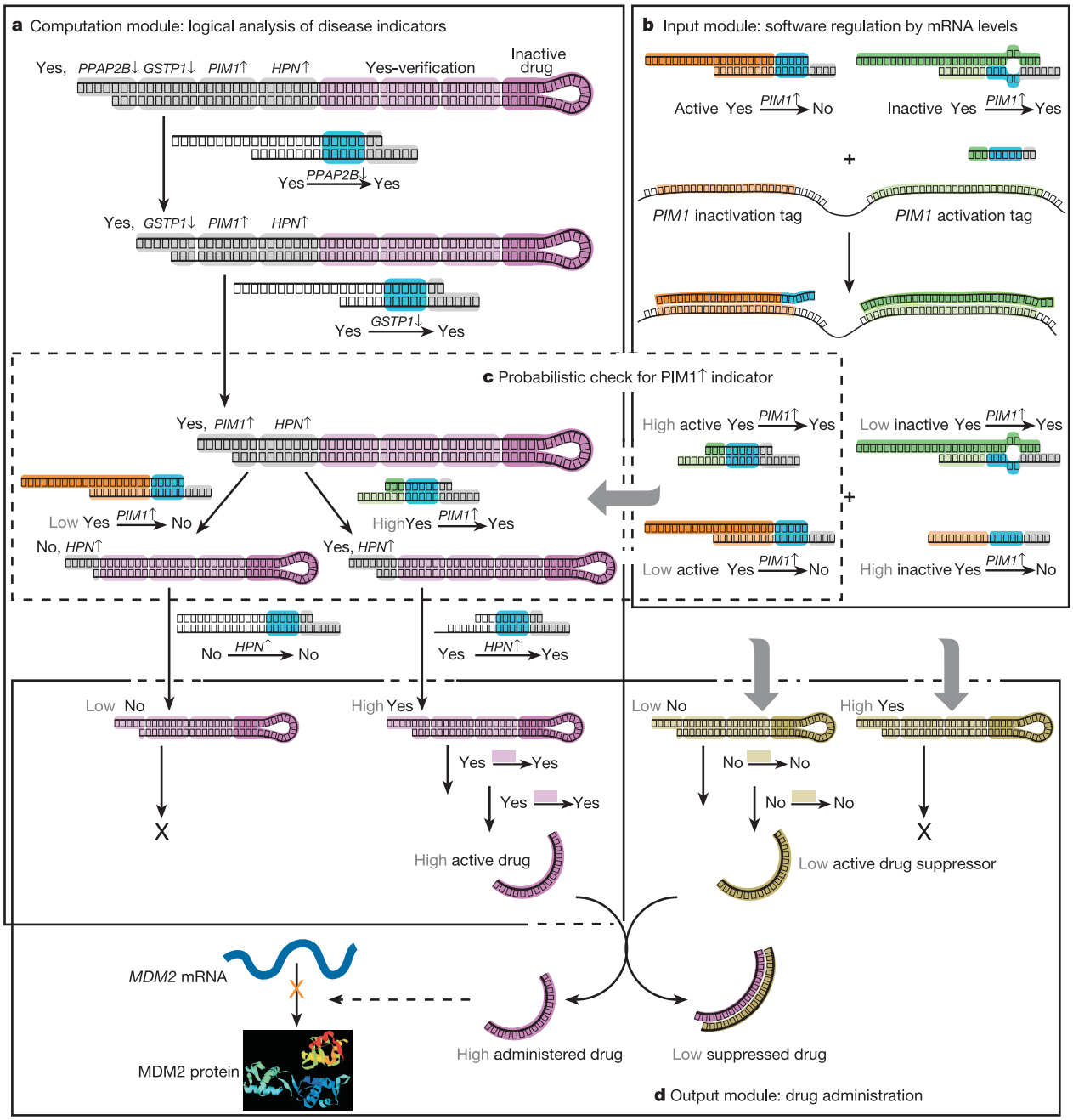

Figure 2 Operation of the molecular computer. The complete sequences for all molecules shown are given in the Supplementary Methods. **a**, Part of the computation path for the diagnostic molecule for prostate cancer with all molecular indicators present, ending in drug release. The initial diagnostic molecule consists of a diagnosis moiety (grey) that encodes the left-hand side of the diagnostic rule (Fig. 1b) and a drug-administration moiety (light purple) incorporating an inactive drug loop (dark purple). At each computation step, the prevailing transition is shown, except for the processing of the symbol *PIM1*↑, for which details of the stochastic choice, accomplished by a regulated pair of competing transition molecules, are shown (dashed box, see **c**). **b**, Regulation of the two transitions for *PIM1*↑ by sub-sequences (tags) of overexpressed *PIM1* mRNA, resulting in a relatively high level of the Yes$\overset{PIM1\uparrow}{\longrightarrow}$Yes transition molecules and low level of the Yes$\overset{PIM1\uparrow}{\longrightarrow}$No molecules. Each transition molecule contains regulation (green, orange) and computation (blue, grey) fragments. The 'inactivation tag' of *PIM1* mRNA (light orange) displaces the 5′ → 3′ strand of the transition molecule Yes$\overset{PIM1\uparrow}{\longrightarrow}$No and destroys its computation fragment. The 'activation tag' of *PIM1* mRNA (light green) activates the transition molecule Yes$\overset{PIM1\uparrow}{\longrightarrow}$Yes. Initially, a protecting oligonucleotide (green) partially hybridizes to the 3′ → 5′ strand of the transition molecule and blocks the correct annealing of its 5′ → 3′ strand. The 'activation tag' displaces the protecting strand, allowing such annealing and rendering an active Yes$\overset{PIM1\uparrow}{\longrightarrow}$Yes transition. Ideally, one *PIM1* mRNA molecule inactivates one Yes$\overset{PIM1\uparrow}{\longrightarrow}$No and activates one Yes$\overset{PIM1\uparrow}{\longrightarrow}$Yes transition molecule. **c**, Stochastic processing of the symbol *PIM1*↑ by a regulated pair of competing transition molecules. The probability of a Yes → Yes transition is high, resulting in a high level of diagnostic molecules in the state Yes and a low level in state No. **d**, Combining computation results for both types of diagnostic molecules, both with high Yes and low No final states, results in high release of drug and low release of drug suppressor, and hence in the administration of the drug.

performing transition regulation and diagnostic computation (Fig. 2a) were verified together (Fig. 3b, c); finally, the input, computation and output modules were combined to perform transition regulation, diagnostic computation and drug administration (Figs 2d and 4a).

The regulation of transition molecules by mRNA (Fig. 2b) was demonstrated (Fig. 3a) with an mRNA transcript of about 1,900 nucleotides as an example indicator. A correlation between the mRNA level and the probability of positive diagnosis, effected by a pair of transitions regulated to check for underexpression, is shown in Supplementary Fig. S3. Regulation by mutation was confirmed with a ssDNA model simulating a point substitution representing a small-cell lung cancer (SCLC)-related mutation in the gene encoding p53 (ref. 21) (Supplementary Fig. S3). We expect to extend this approach to detect insertion and deletion mutations.

In diagnosing a disease model with multiple indicators we used ssDNA oligonucleotides to represent disease-related mRNA and two concentrations to represent mRNA levels: 0 μM for low level and 3 μM for high level. Transition regulation can be adjusted by changing the absolute concentration of competing transitions to distinguish between no mRNA and mRNA at concentrations as low as 100 nM, which represent about 50 mRNA copies per mammalian cell. In our experiments we used up to four molecular indicators, although the specific symbol encoding used can provide up to eight indicators.

We tested the input and computation modules of the computer

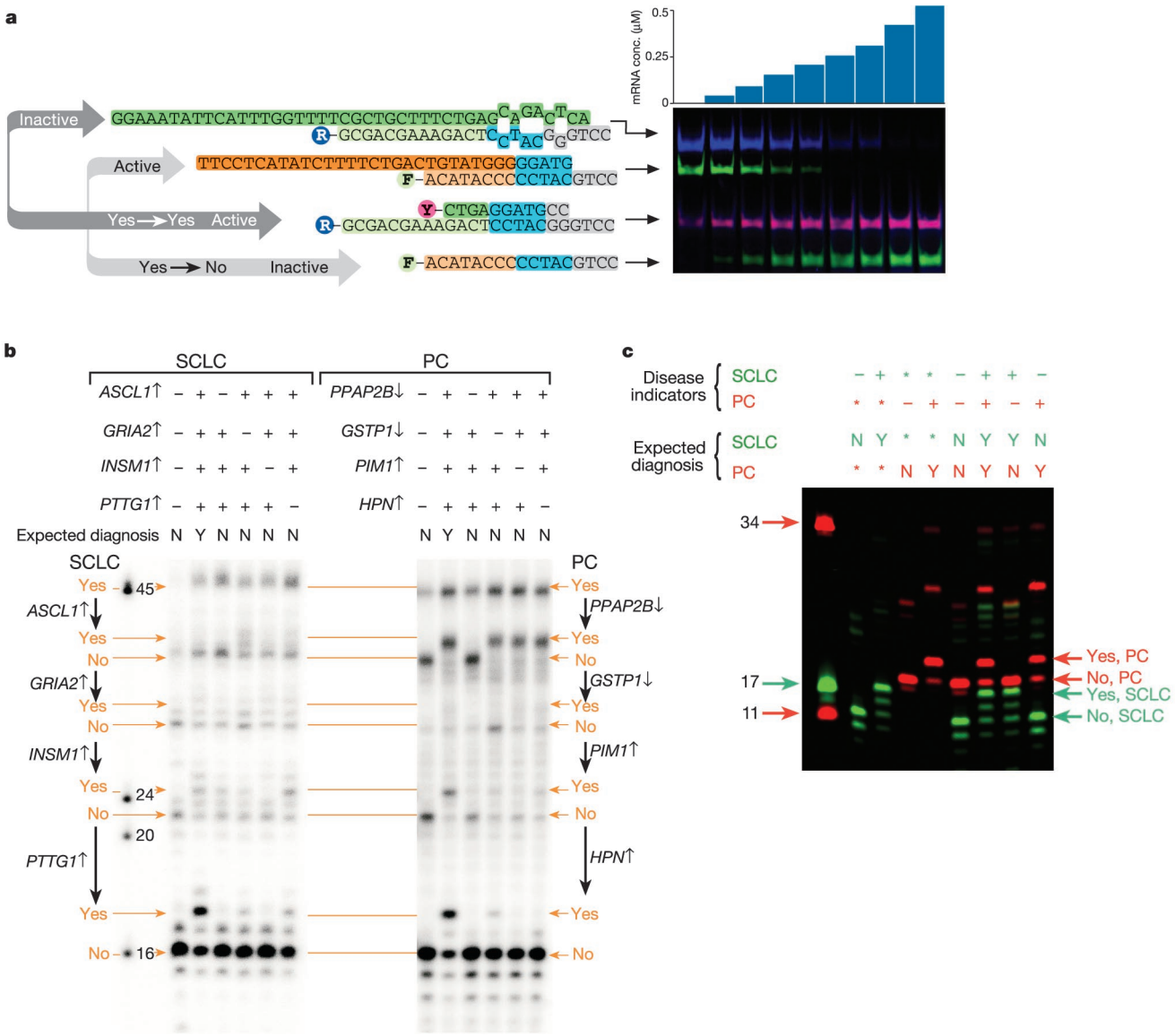

Figure 3 Experimental demonstration of diagnosis. **a**, Regulation of competing transitions by pTRI-Xef mRNA (bar chart) representing an example molecular indicator. The gel shows positive (blue and pink) and negative (green) transition, each in both active and inactive states, the relative concentration of which is regulated by mRNA concentration. F, carboxyfluorescein; R, tetramethyl rhodamine; Y, Cy5. **b**, Validation of the diagnostic automata with the diagnostic rules for SCLC and prostate cancer (PC). Each lane shows the result of a diagnostic computation for the indicated combination of molecular indicators. The location of the initial, intermediate and output molecules is indicated on the left and a predicted trace of the diagnostic computations is shown on both sides. Some, but not all, intermediates are visible owing to their incomplete processing by the automaton. **c**, Parallel diagnosis of two disease models by two diagnostic automata. The diagnosed disease model contains the two indicators of SCLC checked by the diagnostic string *PTTG1*↑ *CDKN2A*↑ and the two indicators of prostate cancer checked by the string *PIM1*↑ *HPN*↑. The gene *CDKNA2* is another indicator for SCLC when overexpressed, and it is used here for technical reasons. The presence of indicators for each disease model as well as the expected diagnostic output by each automaton are indicated above the lanes.

NATURE | doi:10.1038/nature02551 | www.nature.com/nature

on molecular models of SCLC and prostate cancer with diagnostic automata for the diagnostic rules shown in Fig. 1b. The diagnostic output was identified by the size of an inert fragment released on completion of the computation. Each automaton reaches positive diagnosis with significant probability only when all four molecular indicators are present (Fig. 3b). The false-negative diagnosis is due to limitations in the design of the transition molecules, but can be compensated for during drug administration (Fig. 4c) as discussed above. We also confirmed the selectivity of the diagnostic process by testing the two diagnostic automata in mixed conditions, in which molecular indicators for none, one, or both disease models are present (Supplementary Fig. S4). In all cases a positive diagnosis was made with significant probability by a diagnostic automaton only when all molecular indicators were present. We confirmed the possibility of parallel diagnosis of multiple disease models by running together two diagnostic automata under various compositions of molecular indicators (Fig. 3c). Indeed, each automaton performed its diagnosis correctly, irrespective of the computation performed by the other automaton.

Drug administration is demonstrated for the prostate cancer disease model (Fig. 4). We constructed the drug-release diagnostic molecule for *PPAP2B* ↓ *GSTP1* ↓ *PIM1* ↑ *HPN* ↑ (Fig. 2a) and tested the extent of active drug release for different diagnostic outcomes, effected by varying amounts of ssDNA representing *HPN* mRNA and, in a separate experiment, an example mRNA that substitutes for *GSTP1* mRNA. The presence of other indicators was modelled by appropriately formed positive transitions. We show that the amount of active drug increases with the confidence in a positive diagnosis (Fig. 4a). We demonstrate the concept of drug regulation by a drug suppressor using diagnostic molecules for *PPAP2B* ↓ *GSTP1* ↓ with drug release and drug-suppressor release moieties. We show that the prevailing species is the active drug for high probability values, a drug/drug-suppressor hybrid for intermediate probability values, and an active drug suppressor for low probability values (Fig. 4b). We also demonstrate the ability to control the relative amounts of drug and drug suppressor for the 1:1 ratio of positive and negative diagnosis (Fig. 4c). The result demonstrates the robustness of our proposed compensation mechanism and illustrates how multiple degrees of freedom of the system allow it to overcome imperfections in its components.

Our work demonstrates a robust and flexible molecular computer capable of logical analysis of mRNA disease indicators *in vitro* and the controlled administration of biologically active ssDNA molecules, including drugs. The modularity of the design facilitates improving each computer component independently. For example, computer regulation by other biological molecules such as proteins,

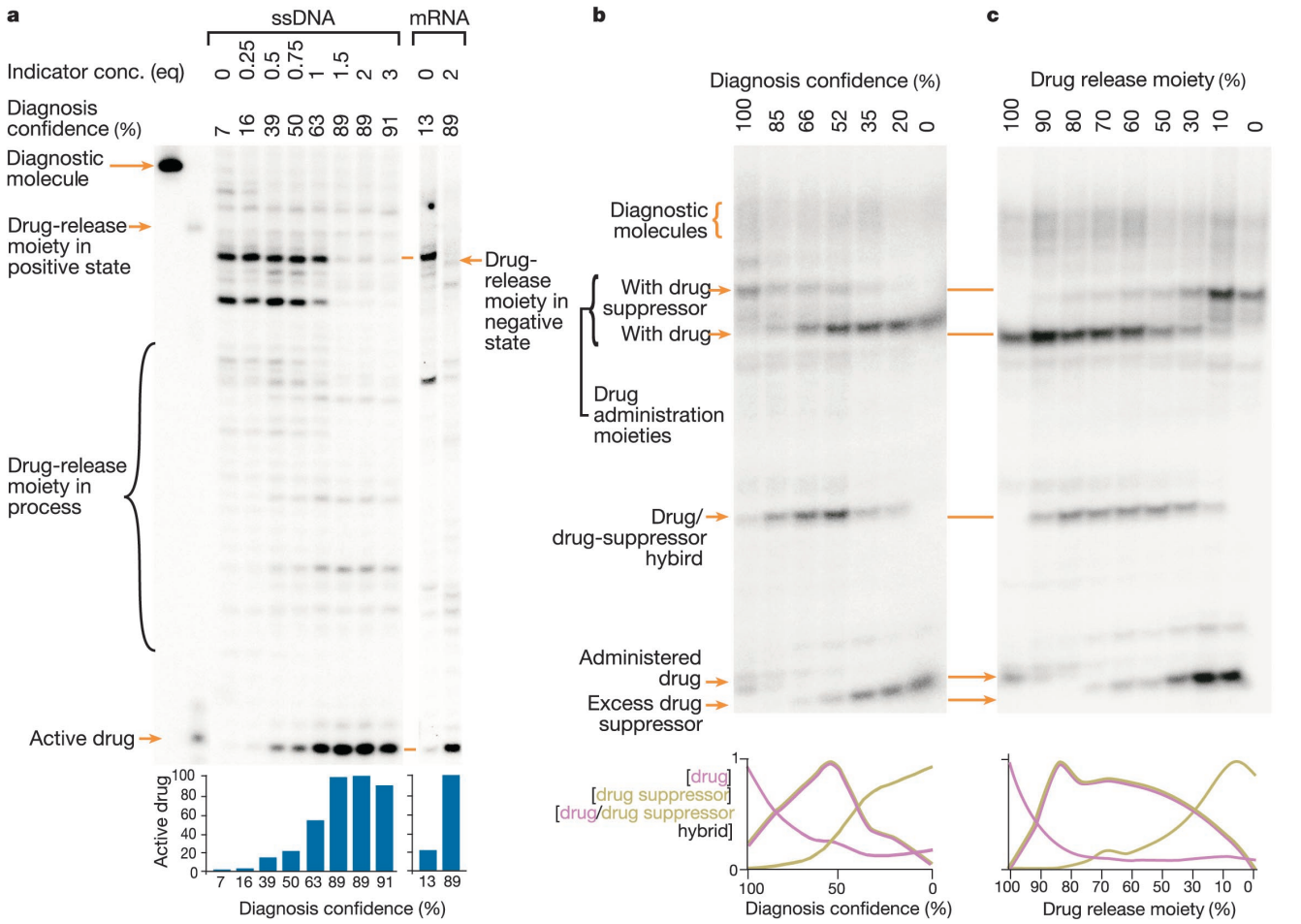

Figure 4 Experimental demonstration of drug administration. **a**, Release of an active drug by a drug-release *PPAP2B* ↓ *GSTP1* ↓ *PIM1* ↑ *HPN* ↑ diagnostic molecule, showing absolute amount of the active drug versus positive diagnosis probability. **b**, Different diagnostic outcomes are modelled using active transition molecules with a mixture of equal amounts of the drug-release and drug-suppressor release moieties for the diagnostic string *PPAP2B* ↓ *GSTP5* ↓ . Each lane shows the distribution of drug-administration moieties, active drug, excess drug suppressor and drug/drug-suppressor hybrid, as indicated. **c**, Variation in the distribution of active drug, excess drug suppressor and drug/drug-suppressor hybrid for a given diagnostic outcome and for varying relative amounts of drug release and drug-suppressor release diagnostic moiety. *y*-axis units in **b** and **c** are arbitrary units.

the output of other biologically active molecules such as RNA interference, and *in vivo* operation can all be explored simultaneously and independently. □

Received 4 February; accepted 6 April 2004; doi:10.1038/nature02551.
Published online 28 April 2004.

1. Adelman, L. M. Molecular computation of solutions to combinatorial problems. *Science* **266,** 1021–1024 (1994).
2. Lipton, R. J. DNA solution of hard computational problem. *Science* **268,** 542–545 (1995).
3. Ouyang, Q., Kaplan, P. D., Liu, S. & Libchaber, A. DNA solution of the maximal clique problem. *Science* **278,** 446–449 (1997).
4. Khodor, J. & Gifford, D. K. Design and implementation of computational systems based on programmed mutagenesis. *Biosystems* **52,** 93–97 (1999).
5. 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).
6. Liu, Q. *et al.* DNA computing on surfaces. *Nature* **403,** 175–179 (2000).
7. Ruben, A. J. & Landweber, L. F. The past, present and future of molecular computing. *Nature Rev. Mol. Cell Biol.* **1,** 69–72 (2000).
8. 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).
9. Mao, C., LaBean, T. H., Reif, J. H. & Seeman, N. C. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature* **407,** 493–496 (2000).
10. Sakamoto, K. *et al.* State transitions by molecules. *Biosystems* **52,** 81–91 (1999).
11. Sakamoto, K. *et al.* Molecular computation by DNA hairpin formation. *Science* **288,** 1223–1226 (2000).
12. Benenson, Y. *et al.* Programmable and autonomous computing machine made of biomolecules. *Nature* **414,** 430–434 (2001).
13. 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).
14. Adar, R. *et al.* Stochastic computing with biomolecular automata. *Proc. Natl Acad. Sci. USA* (submitted).
15. Benenson, Y. & Shapiro, E. in *Dekker Encyclopedia of Nanoscience and Nanotechnology* (eds Schwarz, J. A., Contescu, C. I. & Putyera, K.) 2043–2056 (Dekker, New York, 2004).
16. Rabin, M. O. Probabilistic automata. *Inform. Contr.* **6,** 230–245 (1963).
17. Bar-Ziv, R., Tlusty, T. & Libchaber, A. Protein-DNA computation by stochastic assembly cascade. *Proc. Natl Acad. Sci. USA* **99,** 11589–11592 (2002).
18. Sidransky, D. Emerging molecular markers of cancer. *Nature Rev. Cancer* **2,** 210–219 (2002).
19. Pedersen, N. *et al.* Transcriptional gene expression profiling of small cell lung cancer cells. *Cancer Res.* **63,** 1943–1953 (2003).
20. Dhanasekaran, S. M. *et al.* Delineation of prognostic biomarkers in prostate cancer. *Nature* **412,** 822–826 (2001).
21. Takahashi, T. *et al.* The p53 gene is very frequently mutated in small-cell lung cancer with a distinct nucleotide substitution patter. *Oncogene* **6,** 1775–1778 (1991).
22. Thorns, C., Gaiser, T., Lange, K., Metz, H. & Feller, A. C. cDNA arrays: Gene expression profiles of Hodgkin's disease and anaplastic large cell lymphoma cell lines. *Pathol. Int.* **52,** 578–585 (2002).
23. Ledley, R. S. & Lusted, L. B. Reasoning foundation of medical diagnosis. *Science* **130,** 9–21 (1959).
24. Holzer, S., Fremgen, A. M., Hundahl, S. A. & Dudeck, J. Analysis of medical-decision making and the use of standards of care in oncology. *J. Am. Med. Inf. Assoc.* (Suppl. S) 364–368 (2000).
25. Capoulade, C. *et al.* Apoptosis of tumoral and nontumoral lymphoid cells is induced by both mdm2 and p53 antisense oligodeoxynucleotides. *Blood* **97,** 1043–1049 (2001).
26. Holmlund, J. T. Applying antisense technology. *Ann. NY Acad. Sci.* **1002,** 244–251 (2003).
27. Klasa, R. J., Gillum, A. M., Klem, R. E. & Frankel, S. R. Oblimersen Bcl-2 antisense: Facilitating apoptosis in anticancer treatment. *Antisense Nucleic Acid Drug Dev.* **12,** 193–213 (2002).
28. Yurke, B., Turberfield, A. J., Mills, A. P., Simmel, F. C. & Neumann, J. L. A DNA-fuelled molecular machine made of DNA. *Nature* **406,** 605–608 (2000).
29. Stojanovic, M. N. & Stefanovic, D. A deoxyribozyme-based molecular automaton. *Nature Biotechnol.* **21,** 1069–1074 (2003).
30. Balzani, V., Credi, A. & Venturi, M. Molecular logic circuits. *Chemphyschem* **4,** 49–59 (2003).

Supplementary Information accompanies the paper on **www.nature.com/nature**.

Acknowledgements We thank K. Katzav for the design and preparation of the figures; G. Linshiz for discussion and help in oligonucleotide purification; Z. Livneh for encouraging us to pursue this research direction; A. Regev for critical review and suggestions; and M. Vardi for discussion and references. This work was supported by the Moross Institute for Cancer Research, Israeli Science Foundation and the Minerva Foundation.

Competing interests statement The authors declare competing financial interests: details accompany the paper on **www.nature.com/nature**.

Correspondence and requests for materials should be addressed to E.S. (Ehud.Shapiro@weizmann.ac.il).