

Rivka Adar  
 Yaakov Benenson  
 Gregory Linshiz  
 Adam Wasserstrom  
 Dan Frumkin  
 Sivan Tuvi  
 Ilan Gronau  
 Binyamin Gil  
 Uri Ben-Dor

# Molecular computer: Doctor in a test tube

## Department of Computer Science & Applied Mathematics and Biological Chemistry

Tel. 972 8 934 4506 Fax. 972 8 947 1746

E-mail: [ehud.shapiro@weizmann.ac.il](mailto:ehud.shapiro@weizmann.ac.il)

Web page: [www.weizmann.ac.il/udi](http://www.weizmann.ac.il/udi)

Electronic computers and living organisms are similar in their ability to carry out complex physical processes under the control of digital information -- electronic gate switching controlled by computer programs and organism biochemistry controlled by the genome. Yet they are worlds apart in their basic building blocks --- wires and logic gates on the one hand, and biological molecules on the other hand. While electronic computers are the only “computer species” we are accustomed to, the abstract notion of a universal programmable computer, conceived by Alan Turing, has nothing to do with wires and logic gates. In fact, Turing’s design of the so-called Turing machine has striking similarities to information-processing biomolecular machines such as the ribosome and polymerases. This similarity holds the promise that biological molecules can be used to create a new “computer species”. Such computers can have direct access to a patient’s biochemistry, a major advantage over electronic computers used for medical applications. Eventually, progress in the development of the molecular computers may lead to a ‘Doctor in a cell’: A biomolecular computer that operates inside the living body, programmed with medical knowledge to diagnose diseases and produce the requisite drugs. Our laboratory embarked on the attempt to design and build these molecular computing devices and lay the foundation for their future biomedical applications.

‘DNA computing’ was pioneered by L. Adleman and originally aimed at competing heads-on with electronic computers by solving compute-intensive problems using human-assisted, laboratory-scale manipulation of DNA. With a ‘Doctor in a cell’ vision in mind, we pursue a different paradigm: autonomous, programmable simple computers called finite automata. It turns out that the features of autonomous operation and programmability are much more important than sheer computational

power when it comes to potential applications. At the first stage we designed, implemented and extensively studied the molecular realizations of two-state finite automata. In these automata the input is encoded as a double-stranded (ds) DNA molecule, software, called transition rules, is encoded by another set of dsDNA molecules, and the hardware consists of DNA manipulating enzymes. A computation commences when all molecular components are present in solution, and proceeds by stepwise processing of the input molecule, resulting in a DNA molecule that encodes the output of the computation. An automaton can be stochastic, namely have two or more competing transitions for each state-symbol combination, each with a prescribed probability. A stochastic automaton is useful for processing uncertain information, like most biological and biomedical information. Recently, we have demonstrated a stochastic molecular automaton.

The ‘Doctor in a cell’ concept calls for an autonomous molecular-scale computer that can be programmed to check for disease indicators; to diagnose these indicators according to medical knowledge; and to administer, upon diagnosis, the requisite drug (Fig. 1). Recently, we have developed a molecular computer that performs these operations *in vitro* on simplified molecular disease models (Fig. 2). These models consist of a combination of several disease indicators,

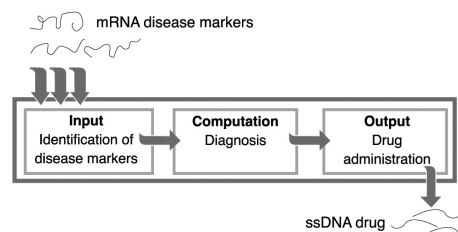


Fig. 1 Logical design of the medical molecular computer.

including over-expressed, under-expressed and mutated genes that indicate cancer. The computer operation is governed by a diagnostic rule; e.g., the rule for prostate cancer states that if the genes PPAP2B and GSTP1 are under-expressed and the genes PIM1 and HEPSIN are over-expressed then administer an anti-cancer antisense drug.

The diagnostic module of the computer is a stochastic molecular automaton with positive (Yes) and negative (No) states. The computation starts in the positive state and if it ends in that state the result is positive diagnosis, otherwise negative diagnosis. The set of indicators is represented as a string of symbols, one for each molecular indicator. For each indicator symbol the automaton has three types of transitions: Yes→Yes (positive); Yes→No (negative); and No→No (neutral). A novel molecular mechanism regulates the probability of each positive transition by the corresponding indicator. 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. High ratio of Yes to No final output means that all sampled disease indicators exist with high probability.

Instead of releasing a drug on positive diagnosis and doing nothing on negative diagnosis, we opted to release a drug suppressor 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, we implement this by using two types of automata, one that releases a drug molecule upon positive diagnosis; and another that releases a drug-suppressor molecule upon negative diagnosis. The ratio between the drug and drug-suppressor molecules released by the two types of automata determines the active drug level.

Our results encourage the speculation that 'smart drugs' --- molecular computers that roam the human body and diagnose a diseased cell, tissue or organ and administer the appropriate therapy *in situ* might be a 'killer application' for molecular computers. However, radical changes will probably be required for our computer to operate *in vivo*. Adapting the medical computer to the *in vivo* environment is a major challenge that lies ahead.

### Selected Publications

- Benenson, Y., Paz-Elizur, T., Adar, R., Livneh, Z., Keinan, E., Shapiro, E. (2001) Programmable and autonomous computing machine made of biomolecules. *Nature* 414, 430-434.
- Benenson, Y., Adar, R., Paz-Elizur, T., Livneh, Z., Shapiro, E. (2003) DNA molecule provides a computing machine with both data and fuel. *Proc. Natl. Acad. Sci. USA* 100, 2191-2196.
- Benenson, Y., Shapiro, E. (2004) Molecular computing machines, in *Dekker Encyclopedia of Nanoscience and Nanotechnology*, 2043-2056
- Benenson, Y., Gil, B., Ben-Dor, U., Adar, R., Shapiro, E. (2004) An autonomous molecular computer for logical control of gene expression. *Nature*, doi:10.1038/nature02551
- Adar, R., Benenson, Y., Linshiz, G., Rosner, A., Tishbi, N., Shapiro, E. Stochastic computing with biomolecular automata, *PNAS*, in press.

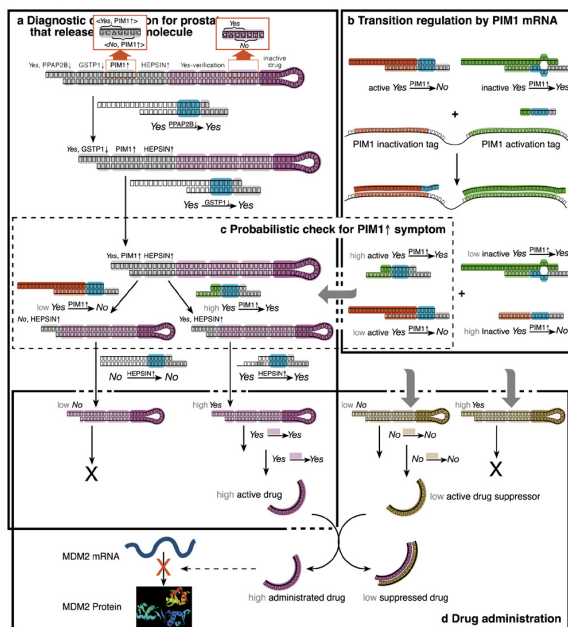


Fig. 2 Operation of the molecular computer.

### Acknowledgements:

Israeli Science Foundation; Ministry of Science; Minerva Foundation; Horowitz Foundation-Yeda; Dolfi and Lola Ebner center for Biomedical Research; Samuel R. Dweck Foundation; Center for New Scientists; Moross MD Cancer Research institute; Equipment Committee; Arthur and Rochelle Belfer Institute of Mathematics and Computer Science; President's Contingency Fund