This experiment elegantly demonstrates the effect of surface corrugation on wheel rotation. When the nanowheel is laterally manipulated along a direction parallel to the surface troughs or on an atomically smooth surface, such as Cu (100), no rolling motion is achieved. Instead, the entire molecule hops onto the next adsorption site. This results in characteristic 'pushing' or 'pulling' signals[7] being recorded during the STM manipulation. The authors also show that the wheel can be rolled only at larger tip–molecule distances (above 4 Å) and is pushed or pulled (Fig. 2b,c) at smaller separations. Therefore, to roll a nanowheel we need an appropriate surface (an 'atomic road') and a means of pushing the wheel in the correct direction.

A detailed grasp of this rolling mechanism of a nanowheel at the atomic limit may allow scientists to design and build better and smarter nanovehicles using individual molecules, with implications for the transport of materials on the nanoscale. It also impacts the development of nanomachines in general as wheels form the basis of many parts of machines, not just the means by which transport occurs. For instance, rotation of a nano-pinion against a molecular rack was recently demonstrated by Chiaravalloti and co-workers[8]. The rotation of the pinion is similar to the rolling of a nanowheel, but the pinion lies flat on the surface and moves along a serrated edge of an island-like structure.

The invention of the wheel revolutionized our civilization. It can be expected that nanowheels will follow suit in revolutionizing the nanoscopic world.

References

1. Grill, L. *et al. Nature Nanotech.* **2,** 95–98 (2007).
2. Gimzewski, J. K. *et al. Science* **281,** 531–533 (1998).
3. Jimenez-Bueno, G. & Rapenne, G. *Tetrahedron Lett.* **44,** 6261–6263 (2003).
4. Shirai, Y., Osgood, A. J., Zhao, Y., Kelly, K. F. & Tour, J. M. *Nano Lett.* **5,** 2330–2334 (2005).
5. Eigler, D. M. & Schweizer, E. K. *Nature* **344,** 524–526 (1990).
6. Hla, S.-W. *J. Vac. Sci. Tech. B* **23,** 1351–1360 (2005).
7. Bartels, L., Meyer, G., & Rieder, K.-H. *Phys. Rev. Lett.* **79,** 697–700 (1997).
8. Chiaravalloti, F. *et al. Nature Mater.* **6,** 30–33 (2007).

BIOTECHNOLOGY

Logic goes *in vitro*

Since its earliest inception, the computer has evolved with the development of faster and smaller electronics. Now, DNA logic circuits tread in water — where no electronic circuit can.

Ehud Shapiro and Binyamin Gil

are in the Departments of Computer Science and Applied Math and Biological Chemistry at the Weizmann Institute of Science, Rehovot, 76100 Israel.

e-mail: ehud.shapiro@weizmann.ac.il; binyamin.gil@weizmann.ac.il

The first universal computer, described by Alan Turing in 1936, was a rather abstract mathematical device. It consisted of symbols written on a potentially unlimited length of tape and a simple device that could move along the tape and process and rewrite each symbol, one at a time, according to a finite set of rules. A decade later, John von Neumann and colleagues conceived and implemented the first practical programmable computer, representing Boolean logic '1's and '0's through the presence or absence of electrical signals. Since then practically all computers have been built according to the von Neumann design.

In retrospect, the similarity of Turing's conceptual machine to the enzymes and ribosomes that process DNA and RNA according to well defined chemical rules is quite astonishing, considering that his ideas predated our current understanding of the structure and function of these basic biological molecules. The idea driving research in DNA computing is to use DNA and enzymes, rather than electronic circuits, to implement mathematical models of computation. Early attempts at DNA computing included conceptual and experimental implementations of Turing machines[1–2], combinatorial algorithms[3–6] and finite automata[7].

Writing in *Science*, Eric Winfree and colleagues[8] at the California Institute of Technology come full circle by demonstrating DNA-based logic circuits that follow the same design principles as modern electronic computing. Why implement logic circuits from DNA if electronic computers have been doing so well for several decades now? We believe the answer is that logical circuits made of biological molecules will have an important advantage compared with electronic circuits in their ability to interact with other biological molecules. DNA computing will likely occur initially *in vitro*, in biotechnology applications; eventually *ex vivo*, facilitating the analysis and manipulation of living cells for biological and biomedical research; and ultimately *in vivo*, as so-called 'smart drugs'[7] that can activate a medical treatment *in situ* by releasing a drug molecule based on a positive logical diagnosis of a disease.

Winfree and colleagues report the implementation of logic circuits using only DNA and demonstrate AND, OR and NOT gates, which are sufficient to effectively compute any Boolean function. The authors show that their application can handle noisy signals and incorporate feedback and cascading, in which the output of one gate is the input of another gate. The latter is trivial in this system thanks to the fact that inputs and outputs are designed to be of the same form. In addition, they show that complex circuits can be made from simpler ones in a modular and scalable way.

The core of the DNA logical operations is strand-displacement. Figure 1a shows the starting components of a basic process: an 'input' strand, A, and a duplex of bound strands, B and C. Strand A, binds by a base pair complementary to the unbound 'toe-hold' of strand B, ultimately displacing strand C. The process ends with the free C strand forming the 'output' and a new, more stable, AB duplex. This process occurs spontaneously because the duplex formed between strands A and B is longer and contains more hydrogen bonds. The end product is therefore thermodynamically more stable. As the single-stranded toe-hold initiates the process, changing its length may change the process speed[9].

The '1's and '0's of Boolean logic are represented by the presence or

absence of a single stranded DNA molecule, respectively, and for each gate, a different set of input strands satisfies the 'true value' that results in output release. The computation reactions are performed in a water-based solution: the concentration of strands determines the input for a logic gate, and the output strands may be read out optically — if they are fluorescently labelled — or may determine the input for another gate. In addition to the output strand (OUT in Fig. 1a) associated with each gate, there is another strand (OUT^P in Fig. 1a), which binds to the output strand and protects it from being 'released' to interact with downstream gates or produce a signal.

Simple logical operations are built on combinations of displacement processes. In the AND gate (Fig. 1b) one output strand and two protecting strands are required for the two consecutive displacement steps. First IN_1 must displace the OUT^{P1} strand. Once the OUT^{P2} toe-hold is revealed, it can initiate a displacement process, driven by IN_2. Thus, only in the presence of both input strands ($IN_1 = 1$ AND $IN_2 = 1$) can the output be released ($OUT_1 = 1$).

The OR operation (Fig. 1c) is achieved by using two AND gates that produce the same output. Each of those AND gates determines the presence of a particular input strand. In other words, these gates are acting as 'translators' that convert one sequence to another. As both gates have the same output, one of the inputs is enough ($IN_3 = 1$ OR $IN_4 = 1$) to release the desired output strand ($OUT_2 = 1$).

Similar ideas can be applied to forming a NOT gate, although this gate requires a 'helping strand' which is complementary to the input strand. Increasingly complex operations can be performed using the same basic design principles. To address false positive output (an output is produced although a gate was not satisfied) and false negative output (no output is produced although a gate was satisfied) release, threshold gate and signal restoration mechanisms were designed and implemented, respectively. Cascading was verified by letting the output of one gate serve as the input of another gate. Amplification of the signal was used when the output of one gate was needed in larger quantities for a consecutive gate. Modularity and scalability were also shown by implementing complex circuit combinations of up to 12 basic gates.

These properties of the system mimic the characteristics of electronic circuits and demonstrate the robustness and applicability of the system. The Caltech group has also shown that their

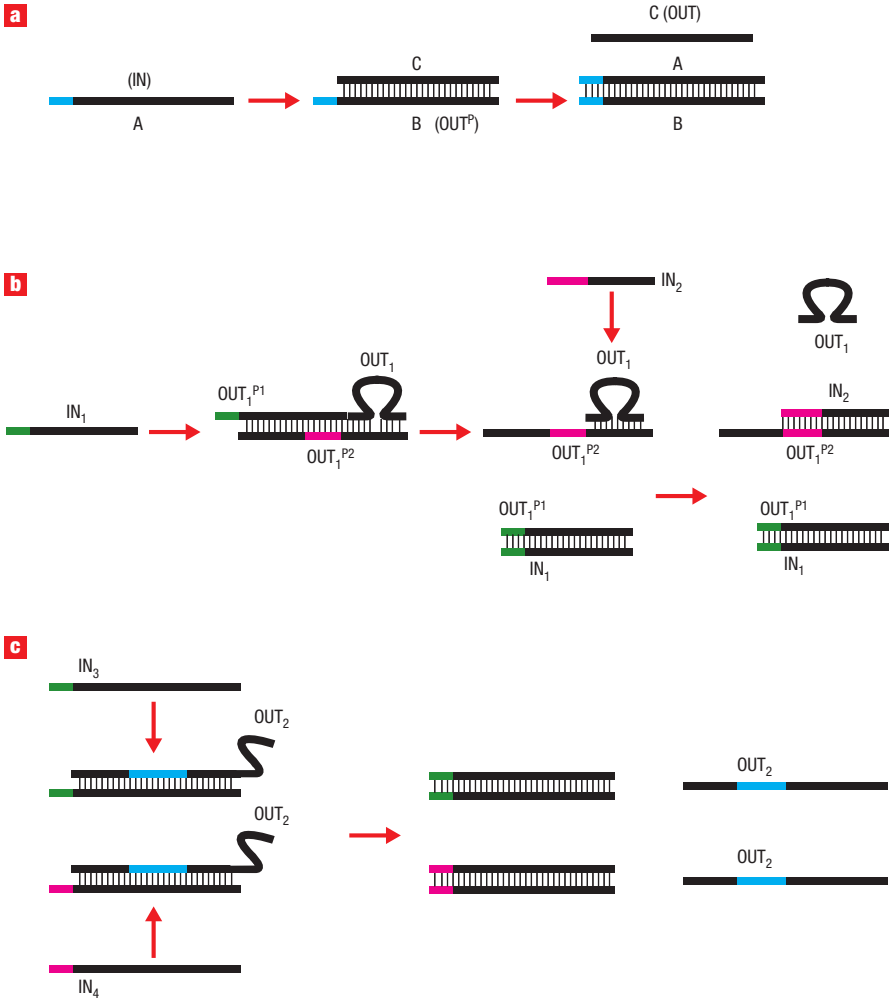

Figure 1 DNA based logical gates. **a**, Strand displacement is the basis of logical operations. An input strand A, binds to the toe-hold of strand B (unbound portion of the BC duplex) and displaces strand C, which then forms the output strand. Logical AND (**b**) and OR (**c**) gates can be formed by various combinations of the process in (**a**).

circuits can operate without interference in the presence of mouse brain total RNA extract, providing a preliminary demonstration that the system can work in living environments.

The work describes a novel way of implementing logic circuits using only DNA. The simplicity of the basic steps and its modularity and scalability renders it a promising foundation for future applications. For example, DNA logic circuits could process native molecules such as messenger RNA and microRNA as inputs to detect complex expression patterns in biological samples. To realize such an application, the system will need to overcome several challenges: to operate in physiological conditions; to interact with molecules in physiological concentrations; and to do so within a biologically relevant timescale. In addition, the probabilistic nature of biological systems and of biomedical data might require probabilistic logic, rather than the stricter Boolean logic, for their analysis. We are looking forward to future incarnations of this system that address these challenges.

References

1. Bennett C. H. *Int. J. Theor. Phys.* **21,** 905–940 (1982)
2. Rothemund, P. W. K. in *Proc. DIMACS Workshop 1995* 75–119 (American Mathematical Society, Providence, Rhode Island, 1996).
3. Adleman, L. M. *Science* **266,** 1021–1024 (1994).
4. Lipton, R. J. *Science* **268,** 542–545 (1995).
5. Ruben, A. J. & Landweber, L. F. *Nature Rev. Mol. Cell Biol.* **1,** 69–72 (2000).
6. Braich, R. S., Chelyapov, N., Johnson, C., Rothemund, P. W. K. & Adleman, L. *Science* **296,** 499–502 (2002).
7. Benenson Y. Gil, B., Ben-Dor, U., Adar, R. & Shapiro, E. *Nature* **429,** 423–429 (2004).
8. Seelig, G., Soloveichik, D., Zhang, D. Y. & Winfree, E. *Science* **314,** 1585–1588 (2006).
9. Turberfield A. J. *et al. Phys. Rev. Lett.* **90,** 118102 (2003).