Published in final edited form as:

Curr Opin Biotechnol. 2010 August ; 21(4): 392–400. doi:10.1016/j.copbio.2010.05.003.

Shaping up nucleic acid computation

Xi Chen¹ and **Andrew D. Ellington^{1,2}**

Xi Chen: xichen@mail.utexas.edu; Andrew D. Ellington: andy.ellington@mail.utexas.edu

¹ Department of Chemistry and Biochemistry, University of Texas at Austin, Austin TX, 78712

² Institute for Cell and Molecular Biology, University of Texas at Austin, Austin TX, 78712

Summary of recent advances (abstract)

Nucleic acid-based nanotechnology has always been perceived as novel, but has begun to move from theoretical demonstrations to practical applications. In particular, the large address spaces available to nucleic acids can be exploited to encode algorithms and/or act as circuits, and thereby process molecular information. In this review we revisit several milestones in the field of nucleic acid-based computation, but also highlight how the prospects for nucleic acid computation go beyond just a large address space. Functional nucleic acid elements (aptamers, ribozymes, and deoxyribozymes) can serve as inputs and outputs to the environment, and can act as logical elements. Into the future, the chemical dynamics of nucleic acids may prove as useful as hybridization for computation.

Introduction

Soon after the elucidation of the DNA double-helix in 1953, the almost unique ability of this macromolecule to carry information was noticed not only by biologists but also by physicists and engineers. As early as 1959, in his famous talk *There's plenty of room at the bottom*, physicist Richard Feynman had already pointed out that only “approximately 50 atoms are used for one bit of information” in DNA and challenged his fellow physicists to devise other systems that were as efficient at recording information. While the explosion of electronic computation could have theoretically led to parallel efforts in using DNA for molecular computation, the maturation of biotechnologies (in the form, for example of recombinant DNA, chemical synthesis, enzymatic amplification, and sequencing) has only recently led to exploitation of DNA as an artificial information carrier. During this fallow period, Ned Seeman led a long (and lonely) battle to convince a larger community that nucleic acid complementarity provided the perfect basis for building nanoscale architectures [1]. Then, in 1994, Leonard Adleman published a seminal work in which an algorithm was encoded in DNA and executed by self-assembly [2**].

Although the massive parallelism of DNA hybridization was alluring, it has so far proven unobtainable for any practical computational applications. Indeed, it seems highly likely that DNA will never rival silicon computers on the execution of any algorithm that could be implemented on both. However, this does not mean that DNA computation is not useful. Rather, it is useful because of the way in which it interfaces with both biology and materials. While the output of an electronic computation is in general some electronic state of a machine,

Correspondence to: Andrew D. Ellington, andy.ellington@mail.utexas.edu.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

the output of a DNA computation is matter itself. Therefore, the real question is how to adapt DNA computation to the novel production of new forms of matter.

DNA computation has so far been implemented in a variety of ways, with the variability being reflected by the way input, output and program are conceived. Notable examples of quite different schema include algorithmic DNA self assembly [3], hybridization-based combinatorial search [2,4], restriction enzyme-based DNA automata [5–6], deoxyribozyme-based DNA automata [7**] and toehold exchange-based dynamic hybridization circuits [8**–9**]. We will review some of the key implementations, and then suggest where computation with matter is going.

The prospects and limitations of nucleic acid computation

Any treatment must start with Adleman's initial instantiation. The computational problem attempted by Adleman was a classic NP-complete problem – the Hamiltonian path problem: given a set of n cities and a set of m directed paths between them (Figure 1a), how can one find a touring path that starts with a given starting city, ends with a given destination city and trespasses all cities only once? To solve this problem, Adleman encoded each city by a 20-nt ssDNA with an arbitrary sequence (referred to as “city strand” hereafter) and also encoded each directed path that connects two cities by a 20-nt ssDNA (referred to as “path strand” hereafter) with the following rule: The first 10 nt of a path strand is complementary to the last 10 nt of the DNA representing the first city that the path connects; the last 10 nt of a path strand is complementary to the first 10 nt of the DNA representing a second city that the path connects (Figure 1b). When city strands and path strands are mixed in the presence of DNA ligase, a city strand can serve as a splint to facilitate the ligation of two path strands if the two paths traverse the city. Ligated path strands can be further ligated with additional path strand(s) in the same manner. When the ligation product is PCR-amplified using the strand corresponding to the starting city as the forward primer and the complementary DNA of the strand corresponding to the destination city as the reverse primer, the PCR product would represent a mixture of random touring paths that start from the given starting city and end with the given destination city. Since the Hamiltonian path goes through each city once, the PCR product corresponding to a Hamiltonian path would have a length of $n \times 20$ nt, which can be gel-isolated from other PCR products. The PCR products of the correct size can be subjected to $n - 2$ cycles of magnetic bead-based affinity purification to select those assemblies that contain all n city strands. This final product can then be sequenced to give the identity of the correct Hamiltonian path.

It was initially thought that this brilliant implementation might provide the basis for extraordinarily parallel systems, up to and perhaps including those that could break the Data Encryption System (DES) [10]. However, as Adleman and his group attempted to expand these systems, they quickly discovered that there were limitations in the kinetics and fidelity of hybridization, and that these limitations also inherently limited the size and complexity of the computation. In the end, they were only able to achieve a 20-variable 3-SAT problem [4], a problem that can be trivially solved by electronic computers.

This was more or less where the field stood until Erik Winfree and co-workers picked up the intellectual strands initiated by Adleman but added a new twist. In Winfree's implementation of DNA computation, algorithms were not so much encoded in DNA as DNA reaction networks or circuits became the algorithm. This DNA on DNA computation, where the outcome was meant to be a piece of matter (a strand) rather than a piece of information (a sequence) was subtly but profoundly different from what had come before.

Central to the designed DNA reaction networks was the concept of toehold-mediated strand displacement [11]. Imagine a DNA hemiduplex formed by a 20-nt top strand and a 26-nt bottom

strand with 6-nt 3' overhang, as shown in Figure 2a, top panel. The displacement of the top strand by an incoming 20-nt strand that has exactly the same sequence as the top strand is a very slow process, due to the stability of DNA duplex. However, if the incoming strand is 26-nt long and is complementary to the bottom strand (Figure 2a, middle panel), the 6-nt segment at the 5' end of the incoming strand can first bind to the 6-nt segment of the bottom strand at its 3' end, and use this initial hybridization as a toehold to initiate a branch migration that finally displaces the original top strand (Figure 2b). Although the branch migration is reversible with no preference to either direction, once the initial top strand has been fully displaced it would not have a toehold on the bottom strand, resulting in an irreversible strand displacement. If the incoming strand is 20-nt long but has a shifted sequence that is complementary to the 3' 20-nt segment of the bottom strand (Figure 2a, bottom panel), toehold-mediated strand displacement will still occur, but the displaced strand will also have a 6-nt toehold on the bottom strand that can initiate the backward strand displacement. This will result in a reversible toehold-mediated strand displacement, a process also called toehold exchange.

Although this scheme at first seems simple and passive, it can in fact support interesting reaction pathways and circuitry. The possibilities for programmed reaction networks become clear as soon as one considers that the top, bottom, and incoming strands in Figure 2 can all be extended at either or both ends. Therefore, the toehold that is exposed by one strand-displacement reaction can potentially initiate a second reaction. For example, given two ~20-nt input strands with known sequence designated as I_1 and I_2 , an AND gate that generates a fluorescent signal only in the presence of both strands can be readily designed (Figure 3a) [8**]. In this scheme, the program consists of three information-processing structures, also called gate structures, named P_1 , P_2 , and P_3 . In gate P_3 , strand F_f has a fluorophore labeled at its 3' end and strand E_q has a quencher labeled at its 5' end, resulting in quenched fluorescence. The separation of fluorophore from quencher, which yields an increased fluorescent signal, will serve as the final readout of the logic gate. When both strands I_1 and I_2 are present, I_1 can bind the toehold of strand K in gate P_1 , displace strand J_{out} and in doing so expose the toehold of J_{out} which was initially occluded (Figure 3a, reaction 1). Similarly, input I_2 can displace strand M_{out} and exposes its toehold (Figure 3a, reaction 2). Once J_{out} has an open toehold, it can hybridize with strand G of gate P_3 through toehold-mediated branch migration (Figure 3a, reaction 3), and as a result expose the toehold of strand F_f . This newly exposed toehold can associate with the open toehold on M_{out} and lead to displacement of strand E_q from F_f and an increase in fluorescent signal (Figure 3a, reaction 4). The serial connection of such gates is also possible, and in the same manuscript Seelig et al. demonstrated that up to 6 input signals can be processed by 11 gate structures in a 4-layer serial architecture.

Although the computational power of toehold exchange reactions is evident, one obvious limitation is that one input molecule can generate, on average, no more than one output molecule. Therefore, a damping of signal is inevitable since each reaction requires, in theory, an infinite amount of time to reach completion. Synthesis errors and side reactions will also contribute to signal loss. To implement signal amplification in DNA circuits the signal-carrying strand must somehow produce more strands. Winfree and colleagues have in fact devised a strand displacement scheme in which one input strand leads to the production of more than one output strands [9**]. A simplified version of this catalytic system has been published [12] in which there are three types of molecules (Figure 3b): the input strand, the gate complex, and the fuel strand. The output strand, which can potentially act as input for another gate, is part of the gate complex. In the first step, the input strand first binds the gate complex through toehold hybridization and then initiates the branch migration to displace the output strand (step 1, similar to Figure 2a, bottom panel). Since this strand displacement is reversible, it is possible that the output strand will displace the input strand in the reverse reaction. However, since the concentration of fuel strand is much higher than that of the displaced output strand, the fuel will instead displace the input strand, resulting in a waste duplex and a recycled input strand

that can proceed to liberate the output strand of another gate duplex with the concomitant production of another waste duplex.

To demonstrate the mechanism of amplification, consider a reaction where the concentration of input strand, gate complex and fuel strand are 1 nM, 10 nM and 90 nM, respectively. The net effect of the amplification reaction is that the input strand triggers the exchange of output strand and the fuel strand on the dock strand. The end point of the reaction is an equilibrium in which the concentration of free output strand is 9 nM. In this case, the signal intensity is amplified by a factor of 9. In theory as few as one molecule of input strand can trigger the production of almost any number of output strands, given sufficient time. The fold-amplification is only limited by practical factors such as the rates of side reactions.

At first glance the driving force of the amplification is not obvious, since there is no change in total number of base pairing and no irreversible reaction. In fact, the reaction is driven by an increase in the entropy of the system. Before the reaction, the distribution of output strand and fuel strand is imbalanced: output strands are exclusively bound to the dock strand and the fuel strands are exclusively unbound. With the completion of the reaction, output and fuel strands will share the dock strand according to their respective concentration: 1/10 of the dock strand (1 nM) is bound to output strand and 9/10 of the dock (9 nM) is bound to the fuel strand. Consequently, 9 nM of output strand is unbound, and the overall entropy is increased.

In summary, the toehold-mediated strand displacement system has made two significant advancement molecular programming. First, the input and output signals are of the same nature, enabling serial connection of gates. Second, signal amplification can guard against signal damping and therefore makes possible the connection of many layers of gates. Building on these features, large DNA reaction pathways that involve ~100 DNA strands and emulate neural networks are being investigated [13]. These pathways will represent the largest molecular systems (at least in terms of interacting molecular species) that have been rationally designed and synthesized. While there are almost certainly many practical issues with implementation, it has been credibly suggested that DNA can be used as a universal substrate for designed chemical reaction pathways [14*].

Adding nucleic acid elements to computations

While the development of DNA circuitry has given new life to the field of DNA computation, the application of this work is still somewhat academic. This is largely because the outputs of the computation are still just DNA strands. To further expand the applicability of DNA computation we must go beyond mere hybridization and understand that DNA can also function almost uniquely as a programmable nanomachine. Nucleic acids can assume shapes that have a variety of functionalities, including forming modular nanostructures (building blocks like TectoRNA [15]), binding ligands (aptamers), catalyzing reactions (ribozymes and deoxyribozymes), and undergoing ligand-induced conformational changes (variously called signaling aptamers, aptazymes, and riboswitches).

One of the first ways that structured nucleic acids can play a role in DNA computation is as computational elements in their own right. A striking feature of DNA and RNA is that they can form both predictable double-helical structures that are governed by Watson-Crick base-pairing and intricate but irregular 3D structures that utilize non-canonical base-pairs and tertiary interactions. These combined features are exemplified in the design of allosteric ribozymes that are controlled by oligonucleotides [16].

As an elegant example, Stojanovic and coworkers systemically engineered two small deoxyribozyme cleavases, E6 [17] (Figure 4a) and 8–17 [18], to create deoxyribozymes that are responsive to up to 3 different oligonucleotide inputs. To make these DNAzymes responsive

to input oligonucleotides, either or both substrate binding domains were extended to form hairpin loops, such that the substrate-binding domain was blocked (Figure 4b). When an input oligonucleotide that was complementary to the loop sequence was present, it could hybridize to the loop and spring the hairpin open, as in molecular beacons, making the substrate-binding domain accessible (Figure 4b). Since the loop sequence could be arbitrarily designed, the deoxyribozyme could be rendered responsive to any input sequence. If both substrate-binding domains were extended to become hairpins with different loop sequences the deoxyribozyme could only become active when both input oligonucleotides were present (Figure 4c). If the cleavage or non-cleavage of the substrate is defined as a logical output (with values 1 and 0, respectively), and the presence or absence of the oligonucleotide input is defined as logical inputs, then the designed deoxyribozyme can be viewed as a molecular AND gate. An OR gate can similarly be implemented by mixing two deoxyribozymes that are activated by two different input oligonucleotides. To construct a NOT gate, Stojanovic and coworkers exploited the fact that the size and sequence of the loop in the catalytic core of E6 deoxyribozyme (Figure 4a) are not critical, whereas the stem must be formed for the deoxyribozyme to be active. In their design, the original 4-nt loop of the catalytic core was replaced by a longer loop with arbitrary sequence (Figure 4d). Therefore the deoxyribozyme is only functional in the absence of input oligonucleotides complementary to this loop. To demonstrate the scalability of their designed gate structures, Stojanovic and coworkers implemented a set of logic circuits that, in combination, are capable of playing tic-tac-toe with a human player [7**]. The readout of this device was ultimately the cleavage of an oligonucleotide, resulting in the separation of a fluorophore and quencher, and hence in the increase in fluorescent signal in any given well.

While this silicomimetic implementation was extremely impressive, a crucial benchmark for DNA computation remains the ability to rationally compose the connectivity of molecular logic gates. As computational devices, logic gates are most powerful only when the output of one gate can act as the input of another gate. In this sense, although many molecular tricks have been touted as logic gates, few can be viewed as being the intellectual (much less practical) equivalent of electronic logic gates. In this regard, there have been few implementations of nucleic acid enzymes in arrangements that can be connected and rationally composed (although the toehold exchange-based DNA logic circuit discussed above [8**] is a notable example). Such connectivity can in principle be implemented in deoxyribozyme-based logic gates since the activation of one deoxyribozyme can lead to the cleavage of the precursor of a second deoxyribozyme, resulting in its activation (Levy and Ellington, 2003). In another example, a deoxyribozyme recoder was generated in which the presence of an input strand led to the ligation and production of a different output strand [19].

Beyond including structured nucleic acids as part of computations, matter computers can produce structured nucleic acids as outputs, and potentially use them as inputs. We saw how deoxyribozymes could cleave nucleic acids to produce fluorescent signals. Similarly, deoxyribozymes have been used to catalyze the turnover of organic substrates. Nearly all DNA G-quadruplexes exhibit peroxidase activity when hemin is used as a cofactor [20] and in the presence of well-developed substrates such as ABTS (2,2'-Azinobis [3-ethylbenzothiazoline-6-sulfonic acid]-diammonium salt) can produce colorimetric signals (or with luminol luminescent signals). Considering the long history of protein peroxidases as reporters in immunoassays, G-quadruplexes may prove to be ideal, simple reporters for nucleic acid-based computations. Quadruplexes have already been used extensively as biosensors (e.g. Ref [21–22]).

By using structured inputs, nucleic acid computers can potentially directly access information embedded in the environment, a key advantage relative to electronic computers. For example, structured nucleic acid aptamers can bind a variety of ligands, and can also be programmed to change their conformation upon ligand-binding [23]. This suggests that it should be possible

to use aptamers as transducers that can convert chemical information into nucleic acid sequence or activity, and hence into signals that can be apprehended and processed by nucleic acid computers.

The ability of aptamers to transduce signals was first demonstrated during the development of aptazymes, aptamer-ribozyme fusions whose catalytic activities can be regulated by ligands. Using rational design [24] and/or directed evolution [25] numerous aptazymes that can be either activated or inhibited by analytes have now been engineered. Some aptazymes already have primitive computation capacity. For example, Breaker and coworkers designed a two-input aptazyme that can function as a logical AND gate [26*]. In this design, one stem of the hammerhead ribozyme was tandemly grafted with two different aptamers that bind theophylline and flavin mononucleotide (FMN), respectively. The aptazyme by itself is in a catalytically inactive conformation, and binding of FMN to the anti-FMN aptamer domain can shift the aptazyme towards an active conformation. However, the anti-FMN aptamer itself is apparently in a disordered conformation unless the anti-theophylline aptamer is bound by theophylline. Thus, the aptazyme is only active when both theophylline and FMN are present.

Aptamers that undergo ligand-induced conformational changes can be adapted not only to ribozyme transducers but also to cellular machines that participate in or regulate gene expression. Natural and engineered riboswitches can be used to transduce ligand concentrations into impacts on the stability or processing of mRNAs [27–28] or shRNAs [29]. Two-input riboswitches that function like AND or NAND gates have also been engineered through directed evolution [30]. These riboswitches only allow (for the AND gate) or suppress (for the NAND) the expression of an adjacent gene only when two ligands, theophylline and thiamine pyrophosphate (TPP), are simultaneously present in the cytoplasm of *E. coli*.

While each of these aptazyme or cellular gates can potentially be serially connected to make more complex and interesting computational devices this will likely be easiest to do if they are engineered to serve as inputs for the computationally versatile DNA reaction pathways that have already been demonstrated, like the toehold exchange system. Fortunately this is a relatively easy task. Given an adequate understanding of the secondary (and preferably tertiary) structure of an aptamer, structure-switching versions of the aptamer that occlude or expose a toehold in response to a ligand can be readily designed and tested [28]. For example, Dirks and Pierce [31] engineered an anti-ATP DNA aptamer which exposes a single-stranded region in the presence of ATP and thereby triggers the hybridization chain reaction, a growing chain of cross-hybridized DNA hairpins. Similar designs could easily be employed in toehold-exchange based logic and/or amplification circuits.

The prospects for amorphous computation

In all of the computations and circuits described above the output is typically Boolean: a path or a signal is generated, or not. Even as a matter computer, DNA computation has been largely designed to function similar to current electronic computers, executing algorithms that lead to a pre-determined answer. However, in some ways such algorithms are profoundly a biological. For example, organisms can be viewed as matter computers in their own right, and the execution of a developmental program does not lead to a pre-determined, Boolean outcome, but rather to a general form whose substance may vary based on stochastic and environmental factors. This is a very different type of computation, termed as amorphous computation by computer scientists at the MIT Media Lab who have developed electronic implementations of what biology does continuously [32]. Amorphous automata are created by programming cells or processors that have a position in space, can communicate with one another, and have a limited rule set. The outcomes produced by these automata are extremely interesting, in that they often

mimic biological processes. There is even an amorphous Origami algorithm that can fold sheets into patterns of any desired complexity, based on a few simple rules.

The implementation of these or similar algorithms requires: (i) identifying cells or processors that have the qualities described above; (ii) figuring out how to program these cells or processors; and (iii) massively producing the cells or processors at low cost. Considering these requirements it seems that the closest route to developed amorphous molecular computation is to program cells or other replicable chemical systems. In fact, it can be argued that synthetic biologists have already begun to march towards this goal, although modularity and composability are far from the same things as programmability [33]. One formidable problem for synthetic biology approaches is the lack of well-characterized parts. This problem becomes especially damning when one considers that the entire notion of modular, composable parts may be antithetical to biological systems, in part because of the inevitable crosstalk that will take place between different parts and with the host. As pointed out by [33], although more and more synthetic biological systems have been successfully implemented, the maximum complexity (defined by the number of promoters used in the system) has so far plateaued at 6. When systems akin to amorphous computers are considered, the inputs and outputs have so far been limited to only a few quorum-sensing molecules, and the possibilities for greatly expanding the number of quorum sensors (to, say, more than 10) are vanishingly small. It can be argued that this is one reason we tend to use neurons for computation rather than hormones.

In contrast, the ample information space available to molecules such as DNA can provide enough complexity to implement amorphous computing. Until cells learn to import and export such information rich molecules, this will likely limit the implementation of complex amorphous computations to non-cellular systems. However, the molecular computing systems we have so far discussed have their own limitations. Most notably, almost all of the complex DNA reaction pathways discussed above [7**–9**] are one-time, energetically downhill reactions. Thus, while different reaction routes can play out according to differing initial conditions, these systems cannot currently respond to changes in their environments in real time. One origin of this limitation is lack of a continuous energy supply (and molecular devices that can use this supply). Although it is not impossible to imagine enzyme-free, time-responsive molecular systems in which energy is continuously introduced, using enzymes to incorporate energy into the system is likely an easier solution. For example, Kim et al. [34*] developed a transcriptional toggle circuit that contained several short template DNAs, each containing a regulatable T7 promoter. The RNA product of one template could regulate the transcription rate of another template. The RNAs were also constantly degraded by RNaseH. The continuous production and degradation of RNA consumed energy stored in NTPs, and rendered the system reversible and time-responsive. Systems like this are more akin to dynamic gene regulation networks and are likely more suitable for implementing amorphous algorithms. Indeed, we have hypothesized that a transcriptional NAND gate implemented as an amorphous computation could potentially be used to create Turing patterns [35].

Conclusions

Although DNA computation was originally touted because of its massive parallelism, and there were fantasies that it might one day challenge the silicon computers, the practical implementation of theoretical parallelism has been severely hindered by the kinetics, error rates and scaling of DNA-based reactions. Despite these limitations, one unique and important feature of DNA computation is its ability to interface directly with chemical and biological systems. To this end, research into DNA automata (where DNA constructs can process information without human intervention) has flourished and, in combination with achievements in other fields, might one day lead to the development of smart drugs that can sense multiple cues in physiology, carry out computations, and then take one or more paths to

outputs such as the release of drugs or the regulation of genes. Another promising direction is the use of DNA as a tool for bottom-up manufacture. This includes not only self-assembly of DNA into defined 3D geometries, but also self-organization of DNA into dynamic and amorphous patterns, through dynamic DNA-based reactions and computations.

Irrespective of what application rises to the fore, it will likely remain true that the unique features of nucleic acids that have allowed these intriguing proofs-of-principle to be carried out will remain unique: nucleic acids embody the largest available address space for molecules (while other biopolymers, such as proteins and sugars, can claim similar informational complexity, there is no good way to actually address them); nucleic acids are the only replicable molecule of any complexity (crystals are their closest competitors); and until protein computational design reaches a new zenith, nucleic acids will remain the only molecule of any complexity where molecular features can be programmed into extensible architectures. Some of the achievements in this field, such as enzyme-free amplifiers and DNA nanostructures, already stand on the verge of translation into real-world applications. With the development of new algorithms and an increasingly modular engineering approach, nucleic acid computation is expected to provide approach to many challenging nano-scale problems and may ultimately fill in the room at the bottom.

Acknowledgments

This work was supported by the Welch Foundation (Grant F-1654) and National Institutes of Health (Grant R01 GM077040). The project described was supported in part by Award Number R01GM077040 from the National Institute of General Medical Sciences. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of General Medical Sciences or the National Institutes of Health.

References

1. Seeman NC. Nucleic acid junctions and lattices. *Journal of Theoretical Biology* 1982;99:237–247. [PubMed: 6188926]
- 2**. Adleman LM. Molecular computation of solutions to combinatorial problems. *Science* (New York, NY) 1994;266:1021–4. In this work Adleman demonstrated the solution of a Hamiltonian path problem via DNA hybridization. This paper is widely considered the inaugural article of the field of DNA computing.
3. Rothmund PWK, Papadakis N, Winfree E. Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS biology* 2004;2:e424. [PubMed: 15583715]
4. Braich RS, et al. Solution of a 20-variable 3-SAT problem on a DNA computer. *Science* (New York, NY) 2002;296:499–502.
5. Benenson Y, et al. Programmable and autonomous computing machine made of biomolecules. *Nature* 2001;414:430–4. [PubMed: 11719800]
6. Benenson Y, et al. An autonomous molecular computer for logical control of gene expression. *Nature* 2004;429:423–9. [PubMed: 15116117]
- 7**. Stojanovic MN, Stefanovic D. A deoxyribozyme-based molecular automaton. *Nature biotechnology* 2003;21:1069–74. This work remains one of the most advanced examples of deoxyribozyme-based circuitry, and was a landmark in the degree of algorithmic complexity that could be achieved.
- 8**. Seelig G, et al. Enzyme-free nucleic acid logic circuits. *Science* (New York, NY) 2006;314:1585–8. This work was the debut of enzyme-free logic circuits based on toehold exchange, which for the first time allowed serial connection of gates.
- 9**. Zhang DY, et al. Engineering entropy-driven reactions and networks catalyzed by DNA. *Science* (New York, NY) 2007;318:1121–5. The authors of this paper took toehold exchange one step further: by designing a reversible toehold exchange and introducing the so-called fuel strands, signal-carrying ssDNA strands were converted into catalysts that could amplify the original signal. For the first time DNA computation began to move from being a toy to a tool.

10. Boneh, D.; Dunworth, C.; Lipton, RJ. Breaking DES using a molecular computer. Proceedings of DIMACS workshop on DNA computing; AMS; 1995.
11. Yurke B, et al. A DNA-fuelled molecular machine made of DNA. *Nature* 2000;406:605–608. [PubMed: 10949296]
12. Zhang DY, Winfree E. Control of DNA strand displacement kinetics using toehold exchange. *Journal of the American Chemical Society* 2009;131:17303–14. [PubMed: 19894722]
13. Qian, L.; Winfree, E. *Lecture Notes in Computer Science*. Springer Berlin Heidelberg; Berlin, Heidelberg: 2009. A Simple DNA Gate Motif for Synthesizing Large-Scale Circuits; p. 70-89.
- 14*. Soloveichik, D.; Seelig, G.; Winfree, E. *Lecture Notes in Computer Science*. Springer Berlin Heidelberg; Berlin, Heidelberg: 2009. DNA as a Universal Substrate for Chemical Kinetics; p. 57-69. In this theoretical work the authors proposed a scheme for using DNA as a universal substrate to emulate arbitrary chemical reaction pathways
15. Chworos A, et al. Building programmable jigsaw puzzles with RNA. *Science (New York, NY)* 2004;306:2068–72.
16. Penchovsky R, Breaker RR. Computational design and experimental validation of oligonucleotide-sensing allosteric ribozymes. *Nature biotechnology* 2005;23:1424–33.
17. Breaker RR, Joyce GF. A DNA enzyme with Mg(2+)-dependent RNA phosphoesterase activity. *Chemistry & biology* 1995;2:655–60. [PubMed: 9383471]
18. Santoro SW, Joyce GF. A general purpose RNA-cleaving DNA enzyme. *Proceedings of the National Academy of Sciences of the United States of America* 1997;94:4262–6. [PubMed: 9113977]
19. Tabor JJ, Levy M, Ellington AD. Deoxyribozymes that recode sequence information. *Nucleic acids research* 2006;34:2166–72. [PubMed: 16648360]
20. Cheng X, et al. General peroxidase activity of G-quadruplex-hemin complexes and its application in ligand screening. *Biochemistry* 2009;48:7817–23. [PubMed: 19618960]
21. Shlyahovsky B, et al. Logic Gates and Antisense DNA Devices Operating on a Translator Nucleic Acid Scaffold. *ACS nano* 2009;3:1831–1843.
22. Elbaz J, et al. Cooperative multicomponent self-assembly of nucleic acid structures for the activation of DNzyme cascades: a paradigm for DNA sensors and aptasensors. *Chemistry (Weinheim an der Bergstrasse, Germany)* 2009;15:3411–8.
23. Liu J, Cao Z, Lu Y. Functional nucleic acid sensors. *Chemical reviews* 2009;109:1948–98. [PubMed: 19301873]
24. Hall B, Hesselberth JR, Ellington AD. Computational selection of nucleic acid biosensors via a slip structure model. *Biosensors & bioelectronics* 2007;22:1939–47. [PubMed: 16996258]
25. Soukup, Ga; Breaker, RR. Engineering precision RNA molecular switches. *Proceedings of the National Academy of Sciences of the United States of America* 1999;96:3584–9. [PubMed: 10097080]
- 26*. Jose AM, Soukup GA, Breaker RR. Cooperative binding of effectors by an allosteric ribozyme. *Nucleic acids research* 2001;29:1631–7. This work demonstrated that in vitro selection could be used to engineer allosteric ribozymes that can be regulated by multiple ligands, and thereby provided an alternative formulation for how logic gates might be structured. [PubMed: 11266567]
27. Desai SK, Gallivan JP. Genetic screens and selections for small molecules based on a synthetic riboswitch that activates protein translation. *Journal of the American Chemical Society* 2004;126:13247–54. [PubMed: 15479078]
28. Bayer TS, Smolke CD. Programmable ligand-controlled riboregulators of eukaryotic gene expression. *Nature biotechnology* 2005;23:337–43.
29. An, C-i; Trinh, VB.; Yokobayashi, Y. Artificial control of gene expression in mammalian cells by modulating RNA interference through aptamer-small molecule interaction. *RNA (New York, NY)* 2006;12:710–6.
30. Sharma V, Nomura Y, Yokobayashi Y. Engineering complex riboswitch regulation by dual genetic selection. *Journal of the American Chemical Society* 2008;130:16310–5. [PubMed: 18998646]
31. Dirks RM, Pierce Na. Triggered amplification by hybridization chain reaction. *Proceedings of the National Academy of Sciences of the United States of America* 2004;101:15275–8. [PubMed: 15492210]

32. Abelson H, et al. Amorphous computing. *Communications of the ACM* 2000;43:74–82.
33. Purnick PEM, Weiss R. The second wave of synthetic biology: from modules to systems. *Nature reviews. Molecular cell biology* 2009;10:410–22.
- 34*. Kim J, White KS, Winfree E. Construction of an in vitro bistable circuit from synthetic transcriptional switches. *Molecular systems biology* 2006;2:68. In this paper the authors developed a transcription circuit that functioned as a toggle switch. Rather than using oligonucleotides as fuel, this circuit was powered by the hydrolysis of NTPs, a more powerful and versatile reaction. [PubMed: 17170763]
35. Simpson ZB, et al. Modelling amorphous computations with transcription networks. *Journal of the Royal Society, Interface/the Royal Society* 2009;64(Suppl):S523–33.

Figure 1.

Adleman's solution of a Hamiltonian path problem (adapted from [2]). **(a)** The schematic figure of a 7-city Hamiltonian path problem. Each city is named by a number and color coded. **(b)** Examples of city strands and path strands, with color-coding as in **(a)**. **(c)** Ligation of two path strands is facilitated by a city strand as a splint.

Figure 2.

Toehold-mediated strand displacement. **(a)** Schematic figure of toehold-mediated strand displacement. **Top panel:** slow displacement without toehold. **Middle panel:** largely irreversible strand displacement. **Bottom panel:** reversible strand displacement, also called toehold exchange. **(b)** Reaction pathway for toehold-mediated irreversible strand displacement. Round and triangular ends of the lines represent 5' and 3' termini, respectively. Dashed line with arrow indicates toehold binding.

Figure 3.

Examples of DNA circuits based on toehold exchange. **(a)** An AND gate, adapted from [8**]. **(b)** An entropy driven amplifier, adapted from [12]. Names of strands are shown in blue and names of gate complexes are shown in black. Solid gray lines with arrows indicate the reactants and products of individual reactions. Dashed gray lines with arrows indicate toehold binding.

Figure 4.

Deoxyribozyme-based logic gates. **(a)** Sequence and secondary structure of deoxyribozyme E6. The substrate is shown in red. The substrate-binding and catalytic domains of E6 are shown in orange and black, respectively. The bolded “A” in the substrate denotes the ribonucleotide adjacent to the cleavage site (dashed red line). A schematic representation of this structure is shown in the inset. These schematics represents **(b)** a single-input YES gate, **(c)** an AND gate, and **(d)** a NOT gate. In **(b)** to **(d)** the input strands are shown in blue and dark green. The corresponding receptors (input-binding loops) are shown in cyan and light green, respectively.