DNA- and RNA-Based Computing Systems

DNA- and RNA-Based Computing Systems

Edited by

Evgeny Katz

WILEY-VCH

Editor

Dr. Evgeny Katz
Clarkson University
Department of Chemistry
Clarkson Avenue 8
NY
United States

Contents

Preface

The use of biomolecular systems for processing information, performing logic operations, computational operations, and even automata performance is a rapidly developing research area. The entire field was named with the general buzzwords, "biomolecular computing" or "biocomputing." Exciting advances in the area include the use of various biomolecular systems including proteins/enzymes, DNA, RNA, DNAzymes, antigens/antibodies, and even whole biological (usually microbial) cells operating as "hardware" for unconventional computing. The present book concentrates on DNA and RNA molecules utilized for information processing (biocomputing). Extensive ongoing research in the DNA- and RNA-based biocomputing has been motivated by speeding up computation, at least for solving some special problems, due to massive parallel operation of numerous biomolecules. The advantages of the DNA and RNA computing systems are also in their ability to operate in a biological environment for solving biomedical problems in terms of diagnostics and possibly therapeutic action, operating as nanorobots in living organisms. DNA molecules are also applicable as memory material with extremely high data density storage.

The present book summarizes research efforts of many groups in different universities and countries. The book reviews and exemplifies these developments, as well as offering an outlook for possible future research foci. The various topics covered highlight key aspects and the future perspectives of the DNA- and RNA-based computing. The different topics addressed in this book will be of high interest to the interdisciplinary community active in the area of unconventional biocomputing. The readers can find additional complementary material on molecular [1], biomolecular [2], and enzyme-based [3] computing published recently by Wiley-VCH (see book cover pages below). It is hoped that the present book will be important and beneficial for researchers and students working in various areas related to biochemical computing, including biochemistry, materials science, computer science, and so on. Furthermore, the book is aimed to attract young scientists and introduce them to the field while providing newcomers with an enormous collection of literature references. I, indeed, hope that the book will spark the imagination of scientists to further develop the topic.

I would like to conclude this preface by thanking my wife Nina for her support in every respect in the past 49 years. Without her help it would not have been

possible to complete this work. Also, cooperation and hard work of all authors working together with me on this edited volume are highly appreciated.

Potsdam, NY, USA   *Evgeny Katz*
January 2020

References

1 Katz, E. (ed.) (2012). *Molecular and Supramolecular Information Processing: From Molecular Switches to Logic Systems*. Weinheim: Wiley-VCH.
2 Katz, E. (ed.) (2012). *Biomolecular Information Processing – From Logic Systems to Smart Sensors and Actuators*. Weinheim: Wiley-VCH.
3 Katz, E. (2019). *Enzyme-Based Computing Systems*. Weinheim: Wiley-VCH.
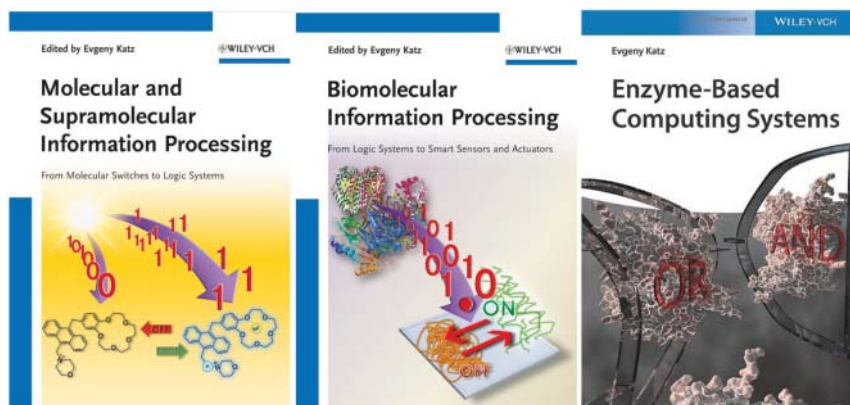
1

DNA Computing: Origination, Motivation, and Goals – Illustrated Introduction

Evgeny Katz

Clarkson University, Department of Chemistry and Biomolecular Science, Potsdam, NY 13699, USA

1.1 Motivation and Applications

Exponential development of computing systems based on silicon materials and binary algorithms formulated as "Moore's law" [1] (Figure 1.1) is coming to the end being limited by further component miniaturization and by the speed of operation. Conceptually novel ideas are needed to break through these limitations. The quest for novel ideas in the information processing has resulted in several exciting directions in the general area of unconventional computing [2–4], including research in quantum computing [5] and biologically inspired molecular computing [6–9]. Molecular computing systems, generally motivated by mimicking natural biological information processing [10, 11], are not necessarily based on biomolecules and could be represented by synthetic molecules with signal-controlled switchable properties. Synthetic molecular systems and nano-species have been designed to mimic operation of Boolean logic gates and demonstrate basic arithmetic functions and memory units. However, despite progress achieved in assembling synthetic molecular systems performing basic Boolean operations and simple computations [6–9], these systems have limited complexity, and further increase of their complexity is very challenging. A new advance in the development of molecular information systems has been achieved with use of biomolecular species [12] (Figure 1.2) such as deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) [13–16], oligopeptides [17], proteins [18], enzymes [2, 19, 20], antigens/antibodies [21], and even whole biological cells/organisms [22–24] capable of operating in a biological environment [25], borrowing some ideas from systems biology [26]. The advantage of the biomolecular computing systems is their ability to be integrated in artificially designed complex reacting processes mimicking multistep information processing networks. These systems are still far away from the natural information processing in cells but are already much more complex than pure synthetic molecular systems. In fact, biochemical reactions are at the core of the mechanism of life itself, and therefore one could set rather ambitious expectations for how far can (bio)chemical reaction systems be scaled up in complexity, if not speed, for information processing. While in a long perspective
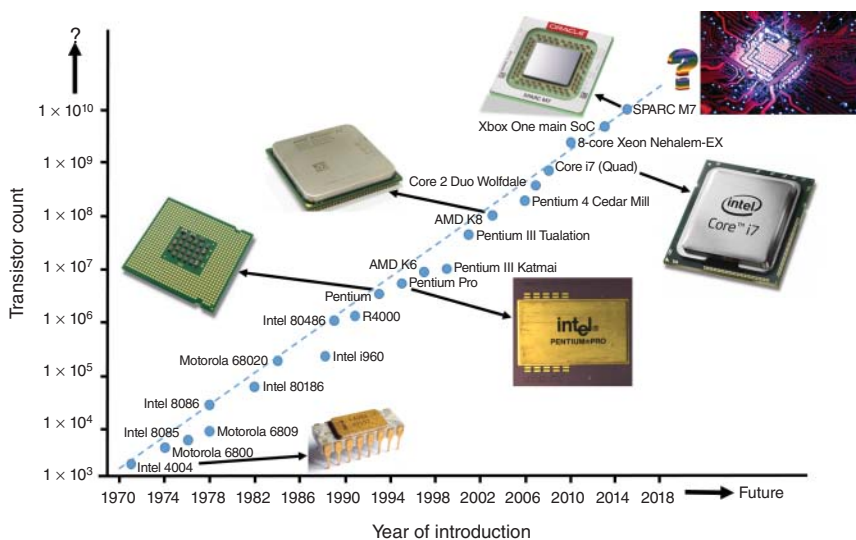
Figure 1.1 Moore's law – exponential increase of transistors on integrated circuit chips. (The plot shown in the figure is based on the data provided by Wikipedia: https://en.wikipedia.org/wiki/Moore%27s_law.) Source: From Katz [2]. Reprinted with the permission of John Wiley and Sons.
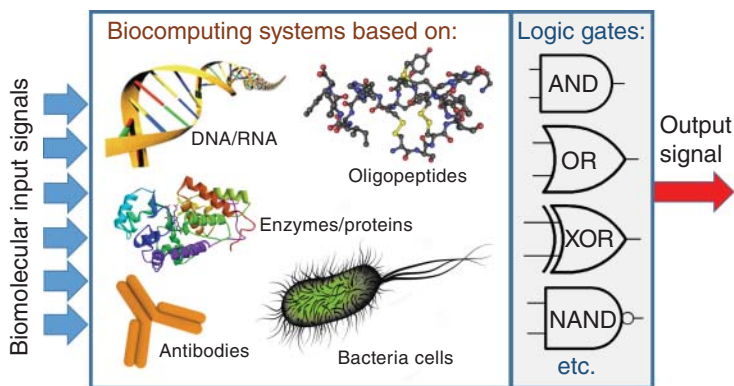
Figure 1.2 Biomolecular computing systems mimicking operation of different Boolean logic gates and circuitries can be based on various species including oligopeptides, enzymes/proteins, DNA/RNA, antibodies, and even whole biological (e.g., microbial) cells. Source: From Katz 2019 [2], Boolean Logic Gates Realized with Enzyme-Catalyzed Reactions – Unusual Look at Usual Chemical Reactions. ChemPhysChem © 2018. Reproduced with the permission of John Wiley & Sons.

a "biocomputer" might become a reality [27], particularly for some special applications, e.g., for solving complex combinatorial problems [28], potentially promising to have an advantage over silicon-based electronic computers due to parallel computing performed by numerous biomolecular units, the present level of technology does not allow any practical use of biomolecular systems for real computational applications. For achieving any practical result soon, some other

applications, different from making a biocomputer, should be considered using the (bio)molecular systems with a limited complexity. One of the immediate possible applications for molecular logic systems is a special kind of biosensing [29–31] where the multiple input signals are logically processed through chemical reactions resulting in YES/NO decisions in the binary (**0,1**) format. In this subarea of biomolecular logic systems, practical results are already possible at the present level of the system complexity, particularly for biomedical applications [32–35]. Overall, the research in molecular/biomolecular information processing, which has been motivated originally to progress unconventional computing applications, is broadly developing to areas not directly related to computing in its narrow definition. This research is bringing us to novel areas in sensing/biosensing [29–31], switchable "smart" materials controlled by logically processed signals [32–36], bioelectronic devices (e.g., biofuel cells) controlled by external signals [37, 38], signal-controlled release processes [39–43], etc.

1.2 DNA- and RNA-Based Biocomputing Systems in Progress

While the general topics of the biomolecular computing [12] and specifically the enzyme-based computing [44] have been covered with recently published books, the present book is concentrated on the use of DNA and RNA molecules in computing systems, broadly defined as information processing systems. From the time (1953) when James D. Watson and Francis H.C. Crick (Figure 1.3) discovered chemical structure of DNA (Figure 1.4) [45], the progress in the DNA study resulted in many novel fundamental scientific concepts [46–48] and highly important practical applications [49]. Among many other, mostly biomedical applications, DNA molecules have been extensively studied over last two decades for unconventional biomolecular computing [13, 15, 50–55], following the pioneer work (1994) by Leonard M. Adleman [28, 56] (Figure 1.5). In his seminal work Adleman demonstrated for the first time computational use of DNA molecules for solving a "traveling salesman problem," Hamiltonian path problem. Actually, this work initiated (bio)molecular computing research not necessary using DNA molecules.

The "traveling salesman problem" asks the following question [57–59]: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?" It is a problem in combinatorial optimization, important in theoretical computer science. It is frequently used to test computational algorithms and computer hardware. In general, the traveling salesman problem is hard to solve, particularly when the number of the visited cities is increasing. Adleman solved the problem for seven cities only (Figure 1.6), which was rather a trivial task, but importantly it was solved using computational power of DNA reactions [28, 56]. The DNA molecules hybridized in a special way to solve the problem, and the computation was performed by numerous DNA sequences (actually rather short oligonucleotides) operating in parallel. This was important conceptual difference from
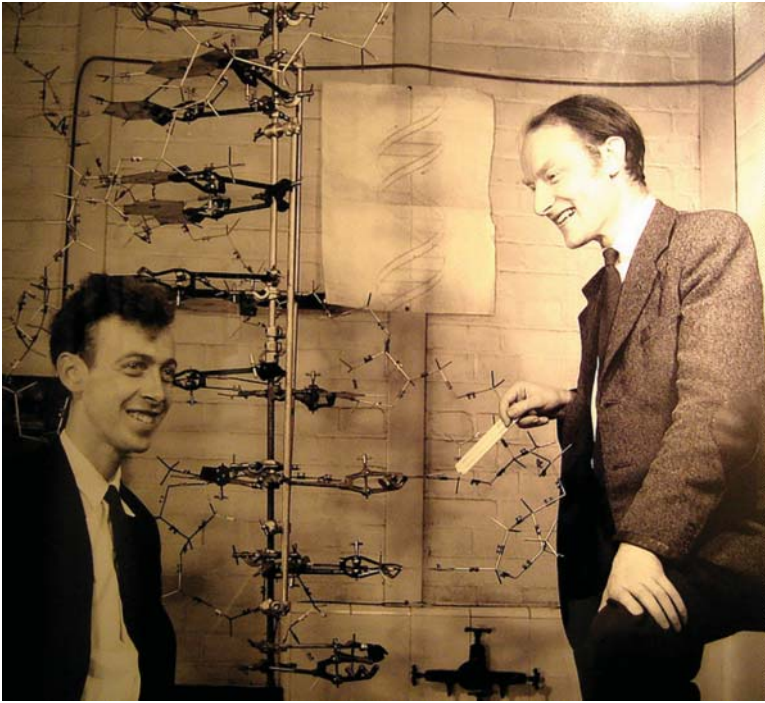
Figure 1.3 The discoverers of the structure of DNA. James Watson (b.1928) at left and Francis Crick (1916–2004), with their model of part of a DNA molecule in 1953. Photographed in the Cavendish Laboratory, University of Cambridge, UK, in May 1953. Source: From Watson and Crick [45]. https://cnx.org/contents/8M7b3dzJ@2/DNA-Structure. Licensed Under CC BY 4.0.

Si-based electronic computers that perform all operations in a sequence. The computation in Adleman's experiment was performed at the speed of 1014 operations per second, a rate of 100 teraflops or 100 trillion floating point operations per second (comparable to the fastest presently available quantum computer) – all because of massively parallel processing capabilities of the DNA computing operation [54, 60]. The promise for extremely fast computation ignited the interest to the DNA computing concept, then being extended to a broader area of molecular [8] and biomolecular [12] computing. Despite the fact that the practical results have not been obtained after almost 25 years of the active research, optimistic expectations for building DNA computers are still present [27, 55, 61, 62]. The advantages of the DNA and RNA computing systems are not only in their potentially high speed of operation due to the parallel information processing but also in their ability to operate in a biological environment for solving biomedical problems in terms of diagnostics and possibly therapeutic action (theranostics) [16, 63], for example, for logic control of gene expression [64]. RNA-based computing systems are particularly promising for *in vivo* operation, thus being excellent candidates for nanomedicine with implemented Boolean logic [65]. DNA computers can operate as a Turing machine [51] and can be sophisticated enough to mimic neural network computations similar

- ○ Hydrogen
- ● Oxygen
- ● Nitrogen
- ● Carbon
- ● Phosphorus

Minor groove

Major groove

T
A

C
G

Pyrimidines   Purines
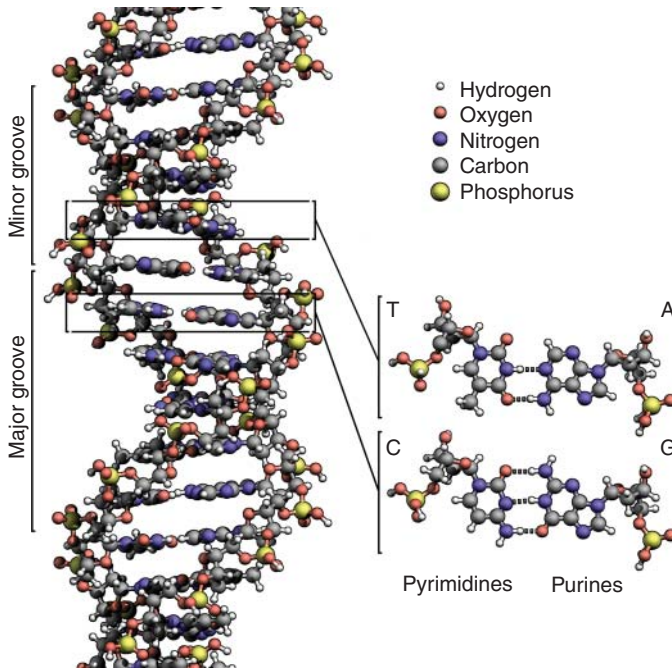
Figure 1.4 The structure of the DNA double helix. The atoms in the structure are color-coded by element, and the detailed structures of two base pairs are shown in the bottom right. Source: From Watson and Crick [45]. Also adapted from Zephyris, DNA Structure, Wikimedia commons, 2011. Public Domain.

Figure 1.5 Leonard Adleman – a pioneer of the biomolecular computing; the photo of 1993 when the first experiments on DNA computing were running. Source: Courtesy of Prof. Leonard Adleman.
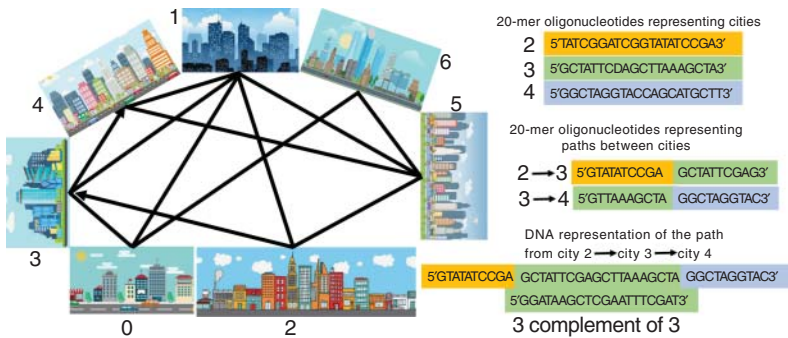
Figure 1.6 The principle of Leonard Adleman's DNA computer to solve the "traveling salesman problem" (see detailed explanation in Ref. [54]). Source: Based on Parker [54].
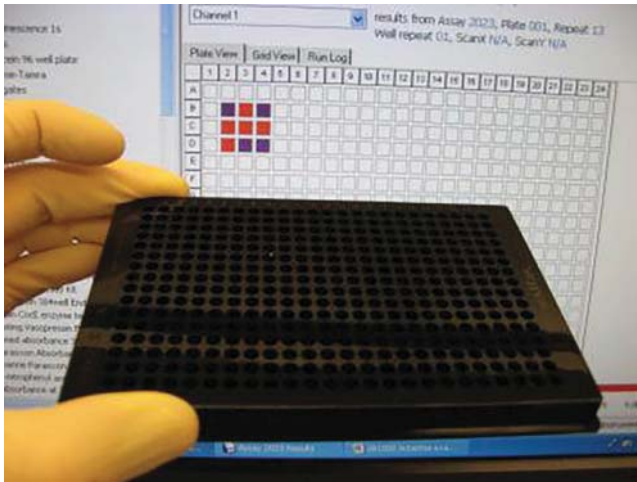
Figure 1.7 The DNA computer playing the tic-tac-toe game. Shown in the foreground is a cell culture plate containing pieces of DNA that code for possible "moves." A display screen (background) shows that the computer (red squares) has won the game against a human opponent (blue). Source: Courtesy of Prof. Milan Stojanovic, Columbia University.

to human brain, obviously in a very simplified way [66]. The DNA computing systems playing a tic-tac-toe game against human have been "smart" enough to win [13, 67–69] (Figure 1.7).

The progress in the DNA computing has been based on three major developments: (i) the use of sophisticated DNA structures (e.g., origami), (ii) the use of more powerful instrumentation for automatic operation of DNA computing steps (DNA chips), and (iii) specialized programming languages specifically developed for the DNA computing. The invention of the DNA origami structures [70, 71] – nanoscale folding of DNA resulting in nonarbitrary two- and three-dimensional shapes [72, 73] (Figure 1.8) – resulted in further sophistication of the DNA computing systems [74], capable of operating as nanorobots in living organisms [75–77]. The use of DNA microarrays (DNA chips [78]) allowed
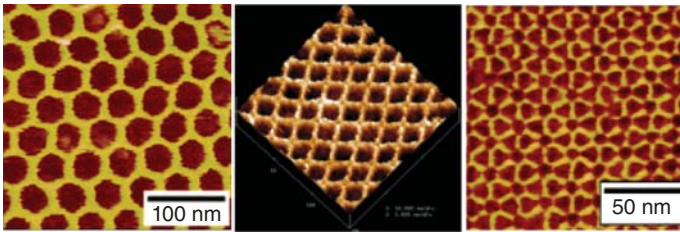
Figure 1.8 Atomic force microscopy (AFM) images of DNA origami with different shapes. Source: From Hong et al. [73]. Reprinted with the permission of American Chemical Society.
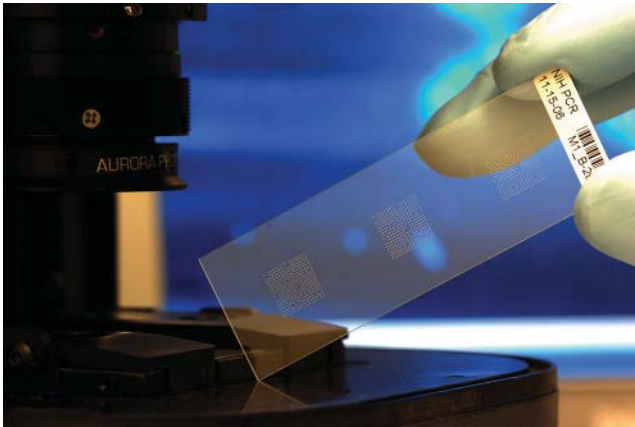
Figure 1.9 An example of a DNA chip used in the DNA sensing and computing. The chip represents a DNA microarray as a collection of microscopic DNA spots attached to a solid surface. Each DNA spot contains picomoles of a specific DNA sequence. The chip allows simultaneous analysis of many DNA probes. The analysis of the probes can be performed optically (as it is in the present example) or electrochemically (then the chip should be based on a microelectrode array). Source: Courtesy of Argonne National Laboratory and Mr. Calvin Chimes.

simultaneous analysis of large numbers of DNA probes [53], thus introducing a powerful hardware for the DNA computing (Figure 1.9). A special computational language, DNA strand displacement (DSD) tool, similar to programming languages used in electronic computers, has been developed by scientists at Microsoft Research for programing DNA computing [79, 80] (Figure 1.10). The language uses DSD as the main computational mechanism, which allows devices to be designed solely in terms of nucleic acids. DSD is a first step toward the development of design and analysis tools for DSD and complements the emergence of novel implementation strategies for DNA computing. The DNA computation can be performed in living cells by DNA-encoded circuits that process sensory information and control biological functions. A special computing language, "Cello," has been developed for programing DNA logic operations *in vivo* [81]. Overall, the use of computing languages simplified the design of DNA computing systems of high complexity.

```
directive rules {
bind(P1,P2,Q,D!i) :-
  P1 = C1 [D], P2 = C2 [D'], compl(D, D'),
  Q = C1 [D!i] I C2 [D'!i], freshBond(D!i, P1|P2).

displace(P,Q,E!j,D!i) :-
  P = C [E!j D] [D!i] [D'!i E'!j],
  Q = C [E!j D!i] [D] [D'!i E'!j].

displaceL(P,Q,E!j,D!i) :-
  P = C [D!i] [D E!j] [E'!j D'!i],
  Q = C [D] [D!i E!j] [E'!j D'!i].

unbind(P,Q,D!i) :-
  P = C [D!i] [D'!i], toehold(D),
  Q = C [D] [D'], not adjacent(D!i,_,P).

adjacent(D!i,E!j,P) :- P = C [D!i E!j] [E'!j D'!i].
adjacent(D!i,E!j,P) :- P = C [E!j D!i] [D'!i E'!j].

reaction([P1;P2], "bind",Q) :- bind(P1,P2,Q,_).
reaction([P],"displace",Q) :- displace(P,Q,_,_).
reaction([P],"displace",Q) :- displaceL(P,Q,_,_).
reaction([P],"unbind",Q) :- unbind(P,Q,_).
}
directive parameters [
  bind = 0.003; displace = 1; unbind = 0.1
]
( 10 [<tb^ b>]
| 10 [<tx^ x>]
| 100 [<to^*!1 x*!2 tx^*!3 b*!4 tb^*>
     | <x!2 to^!1> | <b!4 tx^!3>] )
```

(a)

(b)

Figure 1.10 Logic program (a) and automatically generated chemical reaction network (b) for a DNA strand displacement example. Source: Adapted from Spaccasassi et al. 2019 [80] with permission; open access paper.

1.3 DNA-Based Information Storage Systems

Human civilization generates huge amount of information increasing exponentially and required to be stored. The total digital information today amounts to 3.52×10^{22} bits globally and at its consistent exponential rate of growth is expected to reach 3×10^{24} bits by 2040 [82]. Data storage density of silicon chips is limited, and magnetic tapes used to maintain large-scale permanent archives begin to deteriorate within 20 years. Alternative methods/materials for storing high density/large amount of information with reliable preservation over long period of time are urgently needed. DNA has been recognized as a promising natural medium for information storage [83]. Indeed, the DNA molecules were created by nature to keep the genetic code, which can be easily "written" and "read" by biomolecular systems. With information retention times that range from thousands to millions of years, volumetric density 10^3 times greater than flash memory, and energy of operation 10^8 times less, DNA is a memory storage material viable and compelling alternative to electronic memory. Recent research in the area of information storage with DNA molecules resulted in the proof-of-the-concept systems [82, 84–87], while the practical use of the DNA memory systems is only limited by technological problems. Both processes in the information storage with DNA, "writing" and "reading" information, are available, but they are not as simple as needed to be implemented with the present computer technology. In other words, the DNA memory is technically possible, but it is not convenient enough to be integrated with standard Si-based computers operated by end users.
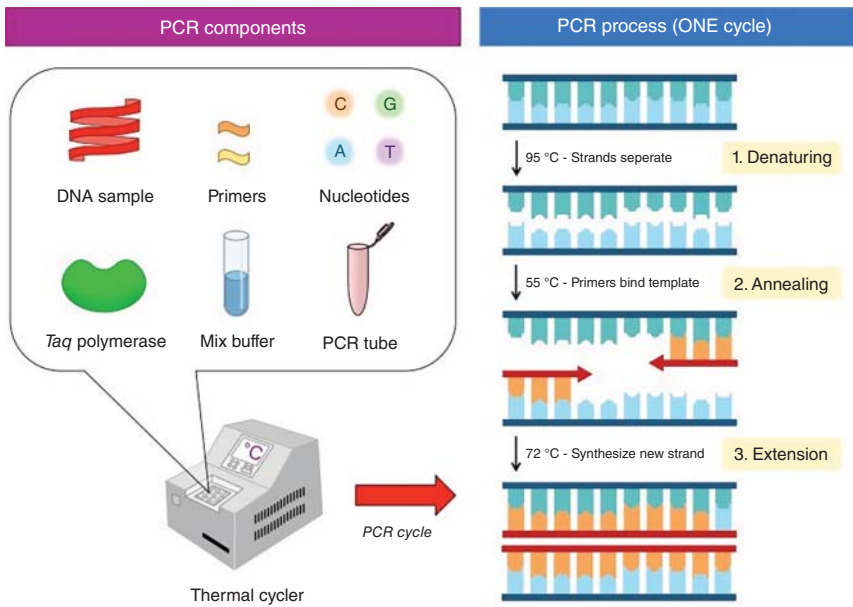
Figure 1.11 PCR method for copying DNA molecules: a thermal cycler, components of the reaction mixture, and reaction steps. For detailed description of the method and instrument see Refs. [89–92]. Source: From Keagile Bati, Polymerase Chain Reaction: Innovation that Revolutionized Molecular Biology, Nov 2018. Public Domain.

Synthetic procedure for production of DNA molecules with specific nucleotide sequences is well known in organic chemistry [88] and can be used to "write" information in the DNA. Once the DNA molecules with the encoded information are prepared, they can be multiplied using a polymerase chain reaction (PCR) [89–92] (Figure 1.11), which is a technique to make many copies of a specific DNA *in vitro* (in a test tube rather than an organism). This technique is rather advanced in the instrumental realization but still requires special apparatus that cannot be connected easily to an electronic computer, at least at the end-user convenience. "Reading" the DNA-encoded information (DNA sequencing [93]) was advanced during the Human Genome Project [94] and presently is very technologically effective. Further improvements in sequencing throughput ($>10^4$) and parallelization ($>10^7$) are expected in the next five years [84]. Emerging technologies such as nanopore sequencing [95] will further reduce errors, cost, time, and energetics during reading the DNA-encoded information. While future advances can result in novel technological approaches, already available techniques based on the DNA memo-chips have been tested [96]. A simple chemical, rather than electronic, apparatus operating as the end-to-end automatic DNA data storage was designed and demonstrated the automatic "writing"–"reading" DNA processes [97] (Figure 1.12). The recent research efforts opened the way toward practical, high-capacity, low-maintenance information storage in synthesized DNA [98–100]. As an example, a 5.27-megabit book was stored using DNA microchips and then read the book by using the
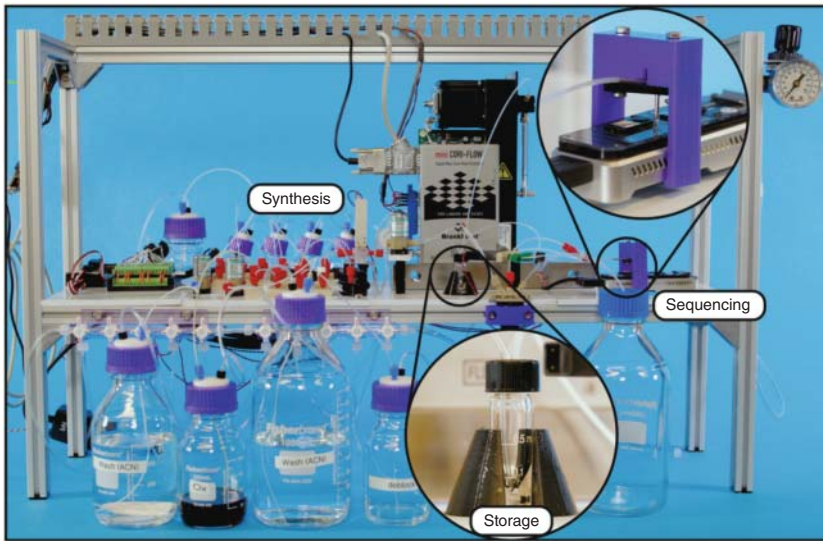
Figure 1.12 Apparatus operating as the end-to-end automatic DNA data storage allowing automatic "writing"–"reading" DNA processes. Source: From Takahashi et al. [97]. https://www.nature.com/articles/s41598-019-41228-8. Licensed Under CC BY 4.0.

DNA sequencing [101]. Other, even more impressive, examples demonstrated encoding the pixel values of black-and-white images and a short movie into the genomes of a population of living bacteria and then retrieving them back by the DNA sequencing [102].

1.4 Short Conclusions and Comments on the Book

Overall, the DNA computing is a multidisciplinary research area with major contributions from synthetic biology, nanotechnology, computer science, chemical engineering, biosensing and biotechnology, biology and medicine, etc. Some of the research areas are already reaching the mature states, while others are still in the infancy. It is still not easy to predict in what direction the research will go and what applications will be more benefiting from the DNA computing. In the most probability, practical applications will be in two major subareas: medicine with the DNA information processing nanorobotic systems operating *in vivo* [103, 104] and large data storage systems providing extremely high density of the information storage [84, 105]. Many other applications of the DNA computing are in the research and discussion [106, 107]. However, it is quite unexpected that the DNA computing will come to the end users instead of standard electronic computers, at least in the short perspective.

The present book, composed of the chapters written by the best experts in the field, covers all subtopics of the DNA computing, including the design of Boolean logic gates and circuitries, programming the DNA information processing systems, their biomedical applications and operation *in vivo*, DNA data storage and

nanopore DNA decoding, and interfacing of the DNA computing with enzyme logic systems, and many more detailed explanations on the DNA and RNA computing with many references and illustrations.

References

1 Moore, G.E. (1998). *Proc. IEEE* 86: 82–85.

2 Katz, E. (2019). *ChemPhysChem* 20: 9–22.

3 Calude, C.S., Costa, J.F., Dershowitz, N. et al. (eds.) (2009). *Unconventional Computation*, Lecture Notes in Computer Science, vol. 5715. Berlin: Springer.

4 Adamatzky, A. (ed.) (2017). *Advances in Unconventional Computing*, Emergence, Complexity and Computation, 2 volumes. Switzerland: Springer.

5 Mermin, N.D. (2007). *Quantum Computer Science: An Introduction*. Cambridge: Cambridge University Press.

6 Szacilowski, K. (2012). *Infochemistry – Information Processing at the Nanoscale*. Chichester: Wiley.

7 de Silva, A.P. (2013). *Molecular Logic-Based Computation*. Cambridge: Royal Society of Chemistry.

8 Katz, E. (ed.) (2012). *Molecular and Supramolecular Information Processing – From Molecular Switches to Logic Systems*. Weinheim: Willey-VCH.

9 Sienko, T. (ed.) (2003). *Molecular Computing* (Series Eds.: Adamatzky, A., Conrad, M., and Rambidi, N.G.). Cambridge, MA: MIT Press.

10 Spitzer, N.C. and Sejnowski, T.J. (1997). *Science* 277: 1060–1061.

11 Kampfner, R.R. (1989). *BioSystems* 22: 223–230.

12 Katz, E. (ed.) (2012). *Biomolecular Computing – From Logic Systems to Smart Sensors and Actuators*. Weinheim: Wiley-VCH.

13 Stojanovic, M.N., Stefanovic, D., and Rudchenko, S. (2014). *Acc. Chem. Res.* 47: 1845–1852.

14 Stojanovic, M.N. and Stefanovic, D. (2011). *J. Comput. Theor. Nanosci.* 8: 434–440.

15 Ezziane, Z. (2006). *Nanotechnology* 17: R27–R39.

16 Xie, Z., Wroblewska, L., Prochazka, L. et al. (2011). *Science* 333: 1307–1311.

17 Ashkenasy, G., Dadon, Z., Alesebi, S. et al. (2011). *Isr. J. Chem.* 51: 106–117.

18 Unger, R. and Moult, J. (2006). *Proteins* 63: 53–64.

19 Katz, E. and Privman, V. (2010). *Chem. Soc. Rev.* 39: 1835–1857.

20 Katz, E. (2015). *Curr. Opin. Biotechnol.* 34: 202–208.

21 Halámek, J., Tam, T.K., Chinnapareddy, S. et al. (2010). *J. Phys. Chem. Lett.* 1: 973–977.

22 Rinaudo, K., Bleris, L., Maddamsetti, R. et al. (2007). *Nat. Biotechnol.* 25: 795–801.

23 Arugula, M.A., Shroff, N., Katz, E., and He, Z. (2012). *Chem. Commun.* 48: 10174–10176.

24 Adamatzky, A. (2010). *Physarum Machines – Computers from Slime Mould*. London: World Scientific.

25 Kahan, M., Gil, B., Adar, R., and Shapiro, E. (2008). *Physica D* 237: 1165–1172.

26 Alon, U. (2006). *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Boca Raton, FL: Chapman & Hall/CRC.

27 Benenson, Y. (2009). *Mol. Biosyst.* 5: 675–685.

28 Adleman, L.M. (1994). *Science* 266: 1021–1024.

29 Katz, E., Wang, J., Privman, M., and Halámek, J. (2012). *Anal. Chem.* 84: 5463–5469.

30 Wang, J. and Katz, E. (2011). *Isr. J. Chem.* 51: 141–150.

31 Wang, J. and Katz, E. (2010). *Anal. Bioanal. Chem.* 398: 1591–1603.

32 Halámková, L., Halámek, J., Bocharova, V. et al. (2012). *Analyst* 137: 1768–1770.

33 Halámek, J., Windmiller, J.R., Zhou, J. et al. (2010). *Analyst* 135: 2249–2259.

34 Minko, S., Katz, E., Motornov, M. et al. (2011). *J. Comput. Theor. Nanosci.* 8: 356–364.

35 Tokarev, I., Gopishetty, V., Zhou, J. et al. (2009). *ACS Appl. Mater. Interfaces* 1: 532–536.

36 Pita, M., Minko, S., and Katz, E. (2009). *J. Mater. Sci. - Mater. Med.* 20: 457–462.

37 Katz, E. and Minko, S. (2015). *Chem. Commun.* 51: 3493–3500.

38 Katz, E. (2010). *Electroanalysis* 22: 744–756.

39 Okhokhonin, A.V., Domanskyi, S., Filipov, Y. et al. (2018). *Electroanalysis* 30: 426–435.

40 Filipov, Y., Gamella, M., and Katz, E. (2018). *Electroanalysis* 30: 1281–1286.

41 Gamella, M., Privman, M., Bakshi, S. et al. (2017). *ChemPhysChem* 18: 1811–1821.

42 Gamella, M., Zakharchenko, A., Guz, N. et al. (2017). *Electroanalysis* 29: 398–408.

43 Katz, E., Pingarrón, J.M., Mailloux, S. et al. (2015). *J. Phys. Chem. Lett.* 6: 1340–1347.

44 Katz, E. (2019). *Enzyme-Based Computing Systems*. Wiley-VCH.

45 Watson, J.D. and Crick, F.H.C. (1953). *Nature* 171: 737–738.

46 Micklos, D. and Freyer, G. (2003). *DNA Science: A First Course*, 2e. New York: Cold Spring Harbor Laboratory Press.

47 Calladine, C.R., Drew, H., Luisi, B., and Travers, A. (2004). *Understanding DNA: The Molecule and How It Works*, 3e. San Diego, CA: Elsevier Academic Press.

48 Douglas, K. (2017). *DNA Nanoscience*. Boca Raton, FL: CRC Press.

49 Fitzgerald-Hayes, M. and Reichsman, F. (2009). *DNA and Biotechnology*, 3e. Amsterdam, Imprint: Academic Press: Elsevier.

50 Boneh, D., Dunworth, C., Lipton, R.J., and Sgall, J. (1996). *Discrete Appl. Math.* 71: 79–94.

51 Rozen, D.E., McGrew, S., and Ellington, A.D. (1996). *Curr. Biol.* 6: 254–257.

52 Eghdami, H. and Darehmiraki, M. (2012). *Artif. Intell. Rev.* 38: 223–235.

53 Ogihara, M. and Ray, A. (2000). *Nature* 403: 143–144.

54 Parker, J. (2003). *EMBO Rep.* 4: 7–10.

55 Mehrotra, A. (2015). *Int. J. Latest Technol. Eng. Manage. Appl. Sci. (IJL-TEMAS)* 4 (12): 62–67.

56 Adleman, L.M. (1998). *Sci. Am.* 279 (2): 54–61.

57 Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., and Shmoys, D.B. (eds.) (1995). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Chichester: Wiley-Interscience.

58 Laporte, G. (1992). *Eur. J. Oper. Res.* 59: 231–247.

59 Applegate, D.L., Bixby, R.E., Chvátal, V., and Cook, W.J. (2006). *The Traveling Salesman Problem – A Computational Study*. Princeton, NJ: Princeton University Press.

60 Reif, J.H. (1999). *Algorithmica* 25: 142–175.

61 Kumar, S.N. (2015). *Am. J. Nanomater.* 3 (1): 1–14.

62 Moe-Behrens, G.H.-G. (2013). *Comput. Struct. Biotechnol. J.* 7 (Art. No.: e201304003).

63 Bonnet, J., Yin, P., Ortiz, M.E. et al. (2013). *Science* 340: 599–603.

64 Benenson, Y., Gil, B., Ben-Dor, U. et al. (2004). *Nature* 429: 423–429.

65 Green, A.A., Kim, J., Ma, D. et al. (2017). *Nature* 548: 117–121.

66 Qian, L., Winfree, E., and Bruck, J. (2011). *Nature* 475: 368–372.

67 Macdonald, J., Stefanovic, D., and Stojanovic, M.N. (2008). *Sci. Am.* 299 (5): 84–91.

68 Stojanovic, M.N. and Stefanovic, D. (2003). *Nat. Biotechnol.* 21: 1069–1074.

69 Elstner, M. and Schiller, A. (2015). *J. Chem. Inf. Model.* 55: 1547–1551.

70 Seeman, N.C. (1982). *J. Theor. Biol.* 99: 237–247.

71 Rothemund, P.W.K. (2006). *Nature* 440: 297–302.

72 Jun, H., Zhang, F., Shepherd, T. et al. (2019). *Sci. Adv.* 5 (Art. No.: eaav0655).

73 Hong, F., Zhang, F., Liu, Y., and Yan, H. (2017). *Chem. Rev.* 117: 12584–12640.

74 Wang, D.F., Fu, Y.M., Yan, J. et al. (2014). *Anal. Chem.* 86: 1932–1936.

75 Amir, Y., Ben-Ishay, E., Levner, D. et al. (2014). *Nat. Nanotechnol.* 9: 353–357.

76 Kogikoski, S. Jr., Paschoalino, W.J., and Kubota, L.T. (2018). *TrAC, Trends Anal. Chem.* 108 (Art. No.: 88e97).

77 Endo, M. and Sugiyama, H. (2018). *Molecules* 23 (Art. No.: 1766).

78 Ramsay, G. (1998). *Nat. Biotechnol.* 16: 40–44.

79 Phillips, A. and Cardelli, L. (2009). *J. R. Soc. Interface* 6: S419–S436.

80 Spaccasassi, C., Lakin, M.R., and Phillips, A. (2019). *ACS Synth. Biol.* 8: 1530–1547.

81 Nielsen, A.A.K., Der, B.S., Shin, J. et al. (2016). *Science* 352 (6281): 53.

82 Panda, D., Molla, K.A., Baig, M.J. et al. (2018). *3 Biotech* 8 (Art. No.: 239).

83 Baum, E.B. (1995). *Science* 268: 584–585.

84 Zhirnov, V., Zadegan, R.M., Sandhu, G.S. et al. (2016). *Nat. Mater.* 15: 366–370.

85 Ceze, L., Nivala, J., and Strauss, K. (2019). *Nat. Rev. Genet.* 20: 456–466.

86 Zhang, S.F., Huang, B.B., Song, X.M. et al. (2019). *3 Biotech* 9 (Art. No.: 342).

87 Tomek, K.J., Volkel, K., Simpson, A. et al. (2019). *ACS Synth. Biol.* 8: 1241–1248.

88 Ma, S., Tang, N., and Tian, J. (2012). *Curr. Opin. Chem. Biol.* 16: 260–267.

89 Behlke, M.A., Berghof-Jäger, K., Brown, T. et al. (2019). *Polymerase Chain Reaction: Theory and Technology*. Poole: Caister Academic Press.

90 Garibyan, L. and Avashia, N. (2013). *J. Invest. Dermatol.* 133 (3), art No. e6.

91 Alves Valones, M.A., Lima Guimarães, R., Cavalcanti Brandão, L.A. et al. (2009). *Braz. J. Microbiol.* 40 (1): 1–11.

92 Powledge, T.M. (2004). *Adv. Physiol. Educ.* 28: 44–50.

93 Shendure, J., Balasubramanian, S., Church, G.M. et al. (2017). *Nature* 550: 345–353.

94 Dolgin, E. (2009). *Nature* 462: 843–845.

95 Agah, S., Zheng, M., Pasquali, M., and Kolomeisky, A.B. (2016). *J. Phys. D: Appl. Phys.* 49 (Art. No.: 413001).

96 Stefano, G.B., Wang, F.Z., and Kream, R.M. (2018). *Med. Sci. Monit.* 24: 1185–1187.

97 Takahashi, C.N., Nguyen, B.H., Strauss, K., and Ceze, L. (2019). *Sci. Rep.* 9 (Art. No.: 4998).

98 Goldman, N., Bertone, P., Chen, S. et al. (2013). *Nature* 494: 77–80.

99 Anavy, L., Vaknin, I., Atar, O. et al. (2019). *Nat. Biotechnol.* 37: 1229–1236.

100 Organick, L., Ang, S.D., Chen, Y.-J. et al. (2018). *Nat. Biotechnol.* 36: 242–248.

101 Church, G.M., Gao, Y., and Kosuri, S. (2012). *Science* 337: 1628.

102 Shipman, S.L., Nivala, J., Macklis, J.D., and Church, G.M. (2017). *Nature* 547: 345–349.

103 Li, S., Jiang, Q., Liu, S. et al. (2018). *Nat. Biotechnol.* 36: 258–264.

104 Ma, W., Zhan, Y., Zhang, Y. et al. (2019). *Nano Lett.* 19: 4505–4517.

105 De Silva, P.Y. and Ganegoda, G.U. (2016). *BioMed Res. Int.* 2016 (Art. No.: 8072463).

106 Tagore, S., Bhattacharya, S., Islam, M.A., and Islam, M.L. (2010). *J. Proteomics Bioinform.* 3: 234–243.

107 Currin, A., Korovin, K., Ababi, M. et al. (2017). *J. R. Soc. Interface* 14 (Art. No.: 20160990).

2

DNA Computing: Methodologies and Challenges

Deepak Sharma and Manojkumar Ramteke

Department of Chemical Engineering, Indian Institute of Technology Delhi, New Delhi, Hauz Khas, New Delhi 110016, India

2.1 Introduction to DNA Computing Methodologies

Humans are looking for new approaches to computing since the starting of civilization. Over the years, researchers have invented many systems for computation, from "counting with abacus" to "complex computing by using modern-day computers." According to Moore's observation [1], the number of transistors on a silicon chip is found to be doubling in every 18–24 months, which results in the development of faster computing devices. However, in the coming decades, producing such faster computing devices will be more challenging as the size of the transistor is already approaching to a molecular level. Moreover, engineering such silicon chips is gradually becoming more complex and less cost effective. This compelled the researchers to look for alternative computing devices and methodologies. Biomolecular computing is one such excellent alternative to traditional silicon-based computing methods.

Biomolecular computing is illustrated for the first time by Adleman in 1994 [2] to solve the Hamiltonian path problem (HPP) using deoxyribonucleic acid (DNA). In such DNA-based computing (*referred to* as DNA computing), enzymatic reactions and manipulations such as addition, amplification, and cutting of the DNA are used for performing the computing. Subsequently, DNA computing is used by several researchers [3–8] to solve a variety of combinatorial problems. These studies have exploited high parallelism of DNA reactions over sequential operations occurring in silicon-based computers to solve the computationally intractable problems. Such parallel processing in DNA computer builds the confidence for solving the problems that are presently not solvable with silicon-based computers. Moreover, DNA became an effective and efficient material for faster computation, storage, and information processing owing to the significant advancements in biomolecular techniques such as gel electrophoresis, polymerase chain reaction (PCR), affinity separation, restriction enzyme digestion, etc. [9, 10].

Despite initial success, the increase in problem size leads to a significant bottleneck in scaling the existing DNA computing procedures for large size problem

formulations. This is primarily because the amount of DNA required increases exponentially with size even though the number of biochemical steps required increases with a polynomial function. Further, there is a significant compounding of experimental errors involved, which makes these procedures redundant for solving the bigger size formulations. Also, the real-life problems often have continuous search spaces and multiple optimal solutions, whereas the existing DNA computing procedures are mostly developed for discrete search spaces involving a single optimal solution.

2.2 Key Developments in DNA Computing

In 1994, Adleman [2] used the DNA for computation in the first-ever molecular experiment performed to solve the HPP. Molecular biology experiments were performed to address the instance graph having seven vertices and fourteen edges. A year later, Lipton [3] presented a new model for solving another NP-complete (nondeterministic polynomial time complete) problem known as satisfiability (SAT) problem using a DNA computer. SAT problem is also solved by Smith et al. [4] and Liu et al. [6] with new surface-based DNA computing models.

Along with these models, other researchers [7, 11, 12] also solved NP-complete problems with a different approach. Sakamoto et al. [5] used DNA hairpin formation to solve the SAT problem. Chao et al. [13] developed a single-molecule "DNA navigator" to solve a maze. These models of DNA computing are discussed in the next section.

2.2.1 Adleman Model

Adleman's DNA computer [2] solved a small instance of a hard computational problem, the HPP. For a given graph, this problem asks for a path with a fixed start and end vertices that visits every vertex exactly once (Figure 2.1a). In his representation, each vertex is encoded using 20 bp nucleotides. Edges are encoded as 20-mers, with the first 10 nucleotides complementary to the last 10 nucleotides of the start vertex and the last 10 complementary to the first 10 of the end vertex (Figure 2.1b). By mixing and ligating oligonucleotides corresponding to vertices and edges, concatenates are formed that represent paths through the network (Figure 2.1c). A path through the graph involves all vertices such that each vertex represented only once is a correct solution to the problem and is referred to as the Hamiltonian path. However, the random nature of ligations leads to the formation of other incorrect paths that do not meet the required condition of the Hamiltonian path. Therefore, several biochemical steps are required (Figure 2.1d,e) to extract the correct path from the set of all paths generated in the ligation process. First, such biochemical step is the PCR, which amplifies only those paths that start and end with right vertices. In the second step, only the paths of correct length are extracted using gel electrophoresis. Finally, the presence of every vertex sequence is confirmed using affinity separation. If any DNA remains after the last separation, this must correspond to a Hamiltonian path.
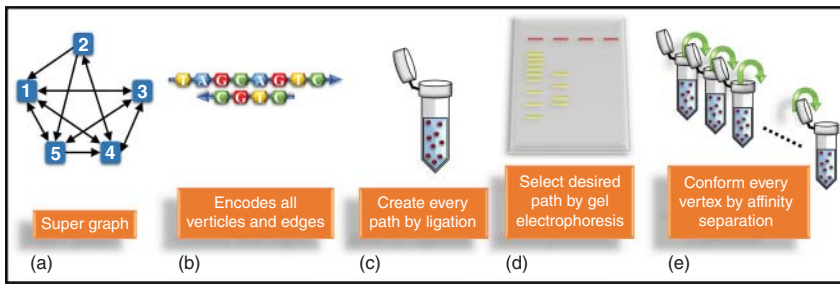
Figure 2.1 Adleman's DNA computing procedure [2]. For (a) given super graph, vertices and edges are encoded using the (b) encoding strategy, and (c) mixed to generate all possible paths through the graph using ligation. The correct length path is selected using (d) gel electrophoresis. All correct length paths are further screened to confirm the presence of sequence corresponding to each vertex one by one starting from first to the last vertex using (e) affinity chromatography.

Consider a supergraph shown in Figure 2.1a involving five vertices. The objective is to obtain a path that starts from vertex 1 and ends at vertex 5 such that each vertex is represented only once. Initially, the DNA sequences of 20 bp are designed for each vertex and edge such that these should lead to linear DNA. For starting and ending edge, the complementary sequence of all 20 bp of starting and ending vertex is used instead of just 10 complementary base pairs. All encoded single-stranded DNA (ssDNA) for each vertex and edge are mixed together to form double-stranded DNA (dsDNA) representing all possible paths in the given supergraph in the ligation step. Since each vertex must be visited exactly once in the Hamiltonian path, a graph of N ($=5$) vertices having each vertex encoded with L ($=20$ bp) nucleotides must comprise total $N \times L$ ($=100$ bp) nucleotides. Therefore, all dsDNA molecules of different lengths obtained after the ligation step are separated using gel electrophoresis. In this step, the gel slice corresponding to the band of the desired length ($=100$ bp) is then separated from the gel by a cutter as the correct DNA sequence corresponding to an optimal solution is expected to be present only in this slice. Further, this band may comprise some undesired solutions with the correct length of 100 bp. The DNA solutions are extracted from the gel slice and are amplified using PCR to generate enough number of desired sequences of DNA representing the solution to the given HPP beginning with one and ending with five. Subsequently, the DNA solution undergoes affinity separation process using streptavidin–biotin magnetic beads to check the presence of vertices 1–5 sequentially one after another in tubes 1–5. The PCR products of these tubes are then analyzed by gel electrophoresis where the bands of respective lengths are obtained, signifying the location of these vertices in the entire sequence. If these tubes give the bands of 20, 40, 60, 80, and 100 bp, then it confirms that all N ($=5$) vertex are visited, and depending on the location of the primer, the location of the vertex in the entire sequence is also determined. For a given example, the Hamiltonian path is 1–3–4–2–5, which corresponds to the bands of 20, 40, 60, 80, and 100 bp, respectively, on the gel image.
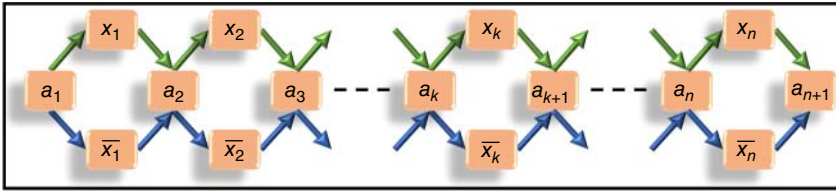
Figure 2.2 Lipton's graph [3] for constructing a binary number for a general variable string $(x_1\,x_2\,x_3\,\ldots\,x_n)$.

2.2.2 Lipton's Model

Lipton [3] solved the SAT problem with a linear complexity of biomolecular operations. In SAT problem the operators logical OR ($\vee$), logical AND ($\wedge$), and logical NOT ($\neg$) along with the variables (x_1, x_2, x_3, …, x_n) constitute a Boolean formula, e.g. $(x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)$. In this formula, a literal can be either a variable or the negation of the variable, e.g. x_1 or $\neg\,x_1$. The clause (C_i) is a disjunction of literals, e.g. $(x_1 \vee x_2)$. Finally, a Boolean formula is a conjunction of clauses [e.g. $(x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)$] such as $C_1 \wedge C_2 \wedge C_3 \wedge \ldots.\ C_m$ for m clauses. In a SAT problem, the objective is to confirm whether there exists a set of input conditions (i.e. the input variables, x_1, x_2, x_3, …, $x_n \in [0, 1]$) for which a given Boolean formula is satisfiable, i.e. the output is 1.

To solve the SAT problem, first, the variables are arranged in a string as $x_1\,x_2\,x_3\,\ldots\,x_n$. In Lipton's model, this variable string, $x_1\,x_2\,x_3\,\ldots\,x_n$, is represented in a graph form. Since the variables can have 0 or 1 values, the vertices with no bars (e.g. x_1, x_2, etc.) represent the 1 value, whereas those with bars (e.g. $\overline{x}_1$, $\overline{x}_2$, etc.) represent 0 value. Further to aid the graphical representation, vertices $a_1 - a_{n+1}$ are added in the sequence as a_1 (x_1 or $\overline{x}_1$) a_2 (x_2 or $\overline{x}_2$) a_3 (x_3 or $\overline{x}_3$) … a_n (x_n or $\overline{x}_n$) a_{n+1} as shown in Figure 2.2a. It is to be noted that $a_1 - a_{n+1}$ are the additional vertices included in the graph to commonly connect x_1 and $\overline{x}_1 - x_n$ and $\overline{x}_n$, respectively, and facilitate in representing all possible combinations for the string $x_1\,x_2\,x_3\,\ldots\,x_n$. The graphical form a_1 (x_1 or $\overline{x}_1$) a_2 (x_2 or $\overline{x}_2$) a_3 (x_3 or $\overline{x}_3$) … a_n (x_n or $\overline{x}_n$) a_{n+1} is represented in the DNA world by using the sequences for each vertex and edge in the same manner as that explained in Adleman's model. These DNA solutions can be amplified and separated easily based on the specific sequence represented for each variable using the biochemical steps of PCR, gel electrophoresis, and affinity separation described earlier.

The objective is to extract a binary sequence for $x_1\,x_2\,x_3\,\ldots\,x_n$ from all possible combinations that satisfy all the clauses C_1, C_2, C_3, …, C_m. To solve this problem, the ssDNA sequences of $x_1, \overline{x}_1, x_2, \overline{x}_2, x_3, \overline{x}_3, \ldots, x_n$ and $\overline{x}_n$ along with sequences of $a_1 - a_{n+1}$ are added in the first test tube. Here all the sequences are selected such that the ligation represents the path between the vertices like Adleman's model. All feasible paths are represented in the first test tube. These paths are constituted by the edges from a_k to both x_k and $\overline{x}_k$ and from both x_k and $\overline{x}_k$ to a_{k+1} for any kth variable. If the vertex takes the x_k label, it encodes the value "1," and if it takes the $\overline{x}_k$ label, it encodes the value "0." For example, the path $a_1\overline{x}_1a_2x_2a_3x_3a_4$ encodes binary number 011. The next operation is the extraction in which the

solution that satisfies the Boolean formula has to be extracted from all feasible solutions present in the first test tube. For this, the DNA solutions are operated by an extraction operator E (t, i, v) in several sequential steps in which t represents the sequences of the tube where an ith bit of variable string $x_1\,x_2\,x_3 \ldots x_n$ is equal to v {0, 1}. Here the extraction operator E (t, i, v) is selected sequentially such that clauses $C_1, C_2, C_3, \ldots, C_m$ are satisfied one after another. Thus, the first extraction operation is selected to make C_1 satisfiable. All DNA solutions satisfying C_1 are then subjected to the next extraction operation, which makes C_2 satisfiable and so on. If any solution is left in the tube after sequentially satisfying $C_1, C_2, C_3, \ldots, C_m$ clauses, then it implies that the given Boolean formula is satisfiable for the sequence remained in the tube. If no solution is left, then it implies that the given Boolean formula is not satisfiable.

Consider a simple example of $(x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)$ for illustration. In this formula, variables x_1, x_2, and x_3 are present. The objective is to find a binary string for $x_1\,x_2\,x_3$, which satisfies the clauses $C_1 = (x_1 \vee x_2)$ and $C_2 = (\neg x_2 \vee x_3)$. Boolean formula $(x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)$ will be solved by making the test tubes for all possibilities, as given in Table 2.1. There are three variables, x_1, x_2, and x_3, represented by "0" or "1," which leads to total eight possibilities ($2^3 = 8$). These eight possible ways are encoded in the form of DNA molecules just by pouring all individual sequences of vertices and edges into the test tube t_0 and performing the ligation. Next, the extraction operation is performed on this tube t_0. The first extraction operator is E $(t_0, 1, 1)$, which extract values having the first bit as "1" since this makes first clause $C_1 = (x_1 \vee x_2)$ true. Also, the second bit corresponding to x_2 can make $C_1 = (x_1 \vee x_2)$ true. Therefore, all remaining solutions that do not satisfy the E $(t_0, 1, 1)$ are extracted in the tube t_1'. These solutions from t_1' are then subjected to the second condition where x_2 becomes 1. Thus, the next extraction operation, t_2, is $E(t_1', 2, 1)$. The solutions extracted by t_1 and t_2 represent the solutions that satisfy $C_1 = (x_1 \vee x_2)$. These solutions are stored in tube t_3 and subjected to the next extraction operation, which makes $C_2 = (\neg x_2 \vee x_3)$ satisfiable. In the next operation, $E(t_3, 2, 0)$ is performed to extract the solution having $\neg x_2$ equal to 1, which satisfies $C_2 = (\neg x_2 \vee x_3)$. Further, the SAT of $C_2 = (\neg x_2 \vee x_3)$ depends on x_3. Therefore, all the remaining solutions stored in t_4' are subjected to $E(t_4', 3, 1)$. This extracts the solutions having x_3 equal to 1. These extracted solutions are stored in t_5. Finally, the solutions left in t_4 and t_5 are the solutions that make the given Boolean formula satisfiable. Similarly, the sequence of extraction steps for evaluating the SAT of any Boolean formula can be designed efficiently, as illustrated in the above example.

2.2.3 Smith's Model

The surface-based DNA computing model was introduced by Smith et al. [4]. In this model, DNA molecules are attached to a solid surface, instead of DNA molecules floating in a solution. Solid-phase nucleotide synthesis is used for the immobilization of the nucleotides on the solid surface. The procedure given by Smith et al. [4] involves following six steps: (i) make, (ii) attach, (iii) mark, (iv) destroy, (v) unmark, and (vi) read out as shown in Figure 2.3.

Table 2.1 The sequence of extraction operations for a given illustrative SAT problem $[(x_1 \lor x_2) \land (\neg x_2 \lor x_3)]$.

Test tube	Operator	Values present
t_0		000, 001, 010, 011, 100, 101, 110, 111
t_1	$E(t_0, 1, 1)$	100, 101, 110, 111
t'_1	$t_0 - t_1$	000, 001, 010, 011
t_2	$E(t'_1, 2, 1)$	010, 011
t_3	$t_1 + t_2$	010, 011, 100, 101, 110, 111
t_4	$E(t_3, 2, 0)$	100, 101
t'_4	$t_3 - t_4$	010, 011, 110, 111
t_5	$E(t'_4, 3, 1)$	011, 111
t_6	$t_4 + t_5$	100, 011, 101, 111
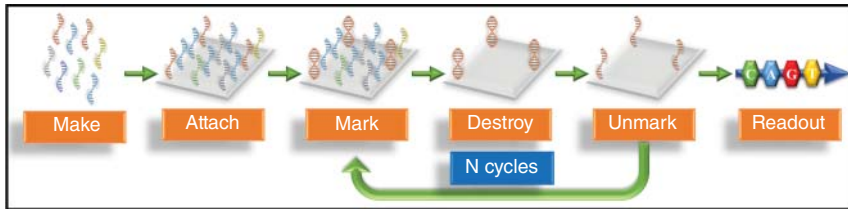
Figure 2.3 Representation of surface-based DNA computing method [4].
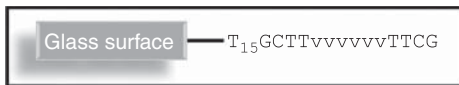
Figure 2.4 Representation of surface-bound DNA sequence.

In this procedure, first, the sequence corresponding all *possible combinations* of variables is designed in step (i). To represent each combination of variables in a SAT problem, Smith et al. [4] used the DNA sequences consisting of unique DNA sequences at each end and variable DNA sequences in the middle for hybridization such as T_{15}GCTTvvvvvvTTCG. In this, the variable DNA sequences are represented by "vvvvvv." The one end of these sequences has a spacer [15 "T" nucleotides (T_{15})] that attaches to the surface. In step (ii), the sequences generated for all combinations are attached to the solid surface, as shown in Figure 2.4.

In step (iii), the sequences corresponding to the satisfaction of each clause are marked by hybridizing these with the complementary sequences corresponding to "vvvvvv." In step (iv), all single-stranded sequences remaining after the hybridization are destroyed by treating with *Escherichia coli* Exonuclease I. In step (v), all hybridized sequences are unmarked to get the single-stranded molecules for all the remaining surface-bound sequences. Steps (iii)–(v) are repeated for all the clauses one after another. The unmarked sequences remaining at the end are analyzed in a readout operation using PCR in step (vi).

The procedure is explained in the context of the same illustrative SAT problem $(x_1 \lor x_2) \land (\neg x_2 \lor x_3)$ of three variables (x_1, x_2, x_3) described earlier. This problem has a solution space of size 8 (each variable can be either "1," or "0," total solution space $= 2^3$). All combinations for this problem are shown in the second row (corresponding to the test tube t_0) of Table 2.1. The problem involves two clauses: $C_1 = (x_1 \lor x_2)$ and $C_2 = (\neg x_2 \lor x_3)$. The SAT is checked for these clauses one by one. $C_1 = (x_1 \lor x_2)$ is not satisfied only if $x_1 x_2 x_3$ is represented by {000} and {001}. Therefore, the complementary sequences for all "vvvvvv" except for the above two clauses are hybridized. This eliminates the above two combinations. The remaining hybridized combinations are unmarked. Next, the SAT of $C_2 = (\neg x_2 \lor x_3)$ is checked. $C_2 = (\neg x_2 \lor x_3)$ is not satisfiable for {010} and {110}. Except these, all the sequences are allowed to hybridize. This leads to hybridization of {011}, {100}, {101}, and {111}. These are unmarked and identified in a readout step using PCR.

2.2.4   Sakamoto's Model

Sakamoto et al. [5] introduced a hairpin formation model for solving an SAT problem using molecular biology techniques. For a given illustrative SAT problem $(x_1 \lor x_2) \land (\neg x_2 \lor x_3)$, literal strings $(x_1, \neg x_2)$, (x_1, x_3), $(x_2, \neg x_2)$, and (x_2, x_3) are formed. A literal string is a string used to encode the given formula with conjunctions of the literals selected from each clause. The literal strings are obtained by concatenating of DNA sequences corresponding to each literal in a ligation step. In these literal strings, if a variable is represented in both original and negation form, then it violates the SAT condition of the given SAT problem. The literal strings without such violation in which a variable is represented only and at least in one form (either actual or negation) constitute a satisfiable solution to the given SAT problem. In Sakamoto's model, possible literal strings are first obtained by ligation. A length of the literal string equals to the number of clauses × nucleotides used for each literal. Subsequently, the obtained literal strings are subjected to temperature variation, which leads to a hairpin formation if a variable is represented in its original and negation form. The restriction enzyme destroys all such hairpins. These solutions are readily eliminated in the subsequent gel electrophoresis operation where only the literal strings with the desired length are separated. All the literal strings separated are analyzed using the sequencer, and the solution of the given SAT problem is obtained. It is to be noted that the given procedure eliminates a large number of unsatisfying literal strings, which makes it easier to deduce the correct solution from the analysis of the remaining satisfying literal strings. The procedure is useful for large size problems. However, it also has a risk of missing some literal strings due to experimental errors that may lead to an erroneous solution to the given SAT problem.

For the given illustrative SAT problem $[(x_1 \lor x_2) \land (\neg x_2 \lor x_3)]$, a single-stranded sequence for all literals is ligated to form the literal strings $(x_1, \neg x_2)$, (x_1, x_3), $(x_2, \neg x_2)$, and (x_2, x_3). The ligated strings are shown in Figure 2.5. These ligated strings will be subjected to restriction enzyme digestion, which eliminates the string $(x_2, \neg x_2)$, and finally, the remaining literal strings are $(x_1, \neg x_2)$, (x_1, x_3), and (x_2, x_3). From these remaining three literal strings, a solution to the given SAT
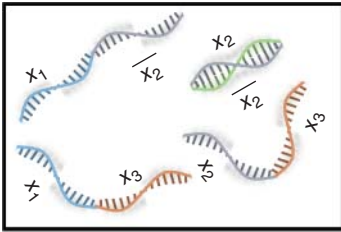
Figure 2.5 Illustration of four literal strings for the DNA hairpin formation-based computation.

problem is deduced by *mathematical analysis*. It is to be mentioned that only one literal string is eliminated for the given problem as it has only $(2)^2$ [= (number of literals in each clause)$^{\text{number of clauses}}$] equal to four literal strings. Though for the given illustrative example the search space is reduced only by 25% (as it removes only one literal string), for bigger problems the reduction in search space is significant. Sakamoto et al. [5] illustrated the benefit of the method by solving 6-variable, 10-clause problem [= $(x_1 \lor x_2 \lor \neg x_3) \land (x_1 \lor x_3 \lor x_4) \land (x_1 \lor \neg x_3 \lor \neg x_4) \land (\neg x_1 \lor \neg x_3 \lor x_4) \land (x_1 \lor \neg x_3 \lor x_5) \land (x_1 \lor x_4 \lor \neg x_6) \land (\neg x_1 \lor x_3 \lor x_4) \land (x_1 \lor x_3 \lor \neg x_4) \land (\neg x_1 \lor \neg x_3 \lor \neg x_4) \land (\neg x_1 \lor x_3 \lor \neg x_4)$] having three literals in each clause. This example has $3^{10} = 59\,049$ literal strings out of which only 24 literal strings were found to be satisfying the condition. Thus, ~99.95% of the search space is eliminated using the above methodology to finally obtain the correct solution just by analyzing 24 literal strings. One of the advantages of this methodology is the use of just one step, unlike sequential elimination steps used in earlier models.

2.2.5 Ouyang's Model

Ouyang et al. [11] solved a maximal clique problem using DNA computing method. A maximal clique is the largest subset in the graph in which each vertex is connected to every other vertex in the subset. In maximal clique problem, the maximum size of the clique in terms of the vertices has to be evaluated. For example (Figure 2.6a), the graph has five vertices, and eight edges where the vertices (5, 4, 2, 1) is the largest clique; thus the maximum size of the clique is four. Ouyang et al. [11] solved the maximal clique problem using DNA computing as follows.

(a) (b)
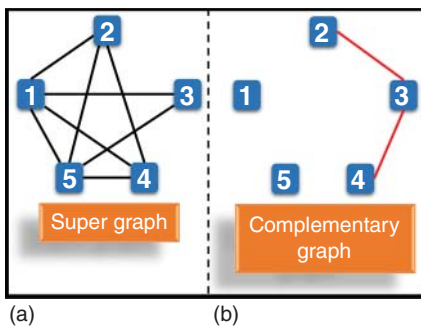
Figure 2.6 (a) The five-node graph and (b) its complementary graph used to solve the maximal clique problem.

First, all possible cliques in the graph of N vertices are represented by an N-digit binary string. In the clique, if the vertex is present, then it is represented by "1," and if the vertex is absent, then it is represented by "0." For the case of 5-vertex graph (shown in Figure 2.6a), a clique involving {5, 4, 2} vertices is represented by a binary string as {11010}. Initially, all possible combinations of this N-digit binary number are generated. Some of the vertices in the graph are not connected by the edges. A graph of such unconnected vertices is referred to as the complementary graph (see Figure 2.6b). In the next step, the combinations comprising the edges present in the complementary graph are removed. For the given illustrative example (Figure 2.6b), the combinations with {cc11c} and {c11cc} are removed from the data pool (c ε {0, 1}). Lastly, find out the binary number having the largest number of "1," which represents the size of the maximal clique. This procedure is performed using the DNA sequences as follows.

Each bit in the above binary string corresponds to a vertex and is represented as a DNA sequence having three parts. In this, the second part corresponds to the vertex number, whereas the first and the third part correspond to the position number. Further, these vertices have to be connected in sequential order (e.g. $V_1 - V_N$) as represented in the binary string. In order to have such connection, the value "0" for the first vertex is represented as $P_1 V_1^0 P_2$, whereas the value "1" is represented by $P_1 V_1^1 P_2$. Since the next vertex V_2 has to be connected to V_1, its "0" value is represented by the complementary sequence $\overline{P_3 V_2^0 P_2}$, whereas its value "1" is represented by the sequence $\overline{P_3 V_2^1 P_2}$. Similarly, the next vertex V_3 will be connected to V_2 by $P_3 V_3^0 P_4$ and $P_3 V_3^1 P_4$ for "0" and "1" values. Thus, all odd-numbered vertices are represented by $P_i V_i^0 P_{i+1}$ and $P_i V_i^1 P_{i+1}$ for "0" and "1" values, whereas those for even-numbered vertices are represented by $\overline{P_{i+1} V_i^0 P_i}$ and $\overline{P_{i+1} V_i^1 P_i}$, respectively. Further, all sequences with V_i^0 have 10 nucleotides representing the second part, whereas these are absent in V_i^1. The ssDNA sequences corresponding to each bit combine in a sequential manner to form a dsDNA representing all combination of 0 or 1 for each vertex. Initially, all the sequences starting with P_1 and ending with P_N are amplified using PCR. The complementary graph is removed from the data pool by treating the DNA solutions using the restriction enzyme repeatedly for each edge present in the corresponding graph. For each edge, the DNA solution is divided into two equal parts in two tubes, and the restriction enzymes corresponding to the constituting vertices are added in the respective tubes. After restriction enzyme digestion, solutions of both tubes are added, and the process is repeated for all the edges of the corresponding graph. After this, a PCR is performed with starting P_1 and ending P_N. The amplified solutions are then analyzed using gel electrophoresis to obtain the maximal clique for the given supergraph. On the gel electrophoresis, the largest clique is corresponding to the shortest DNA strands. This is primarily because all the sequences comprising maximum V_i^1 will lead to the maximum "1"s in the binary string and will have the shortest length in base pairs. Such strands are analyzed by cloning and sequencing to obtain the maximal clique. For the given illustrative example (Figure 1.12b), {cc11c} and {c11cc} sequences will be removed. Therefore, the sequences that will lead to the maximal clique will be {11011}. From these, the maximal size of

the clique for the given illustrative example is four, which corresponds to the clique $V_1–V_2–V_4–V_5$.

2.2.6 Chao's Model

Chao et al. [13] developed a single-molecule "DNA navigator" to solve a maze (tree graph) of 10 vertices with three junctions. In this, the desired path is explored out of all possible paths of the maze present on an origami that is used as a substrate. On this origami, some sites are specifically used for the binding of the vertices of the tree graph. This helps in the propagation of the path on the origami.

The DNA used for the process is very specific in size and design. The designing of DNA is described next. Hairpin DNA Y is attached to the origami and has a typical sequence layout of the structure $5' - toehold - stem - loop - \overline{stem} - 3'(5' - t_h - s_t - l - \overline{s_t} - 3')$, as shown in Figure 2.7. Another type of DNA used is hairpin DNA Z, which is present freely in the solution. Its structure is like $5' - \overline{s_t} - \overline{t_h} - s_t - \overline{l} - 3'$. Additionally, an initiator DNA that has a sequence $s_t \, t_h'$ is used to start the process. This initiator DNA binds to the entry vertex. Similarly, an exit DNA is present, which does not have a loop to form a hairpin.

After the binding of the initiator, a hairpin loop of the DNA Y opens to make it free to bind to DNA Z and vice versa as both have complementary sites for free form of each other. This type of hybridization continues until it reaches the exit DNA. This hybridization chain also produces those paths that are not the solution to the maze. The exit DNA corresponding to an end vertex of the maze is biotin labeled. If the path is correct, then this biotin-labeled DNA is free from the exit vertex; otherwise, biotin remains attached to the DNA corresponding to the exit vertex. All the biotin-labeled sequences are then removed from the solution by streptavidin magnetic beads. The correct path sequence remained in the solution as it is not attached to the biotin. This solution is then analyzed by AFM for identifying the final path.

2.2.7 DNA Origami

Self-assembly of DNA is one of the most popular techniques used to create a variety of two-dimensional and three-dimensional structures. The DNA origami is an excellent example of self-assembly of DNA, which was introduced by Rothemund [14]. The procedure of DNA origami is shown in Figure 2.8. DNA origami is generated from a thousand base pair long ssDNA, known as a scaffold. Some shorter and linear ssDNA called staples are mixed with the scaffold to create DNA origami. For this purpose, the solution mixture is first heated to 95 °C. The solution is then cooled down to room temperature from 95 °C; throughout the cooling, the staples bind to the scaffold through Watson–Crick complementary base pairing. The scaffold then becomes a static structure or pattern of DNA. The sequence of DNA staples results in the formation of various shapes, for example, squares, smiley faces, cube, hexagon, star, and many more.
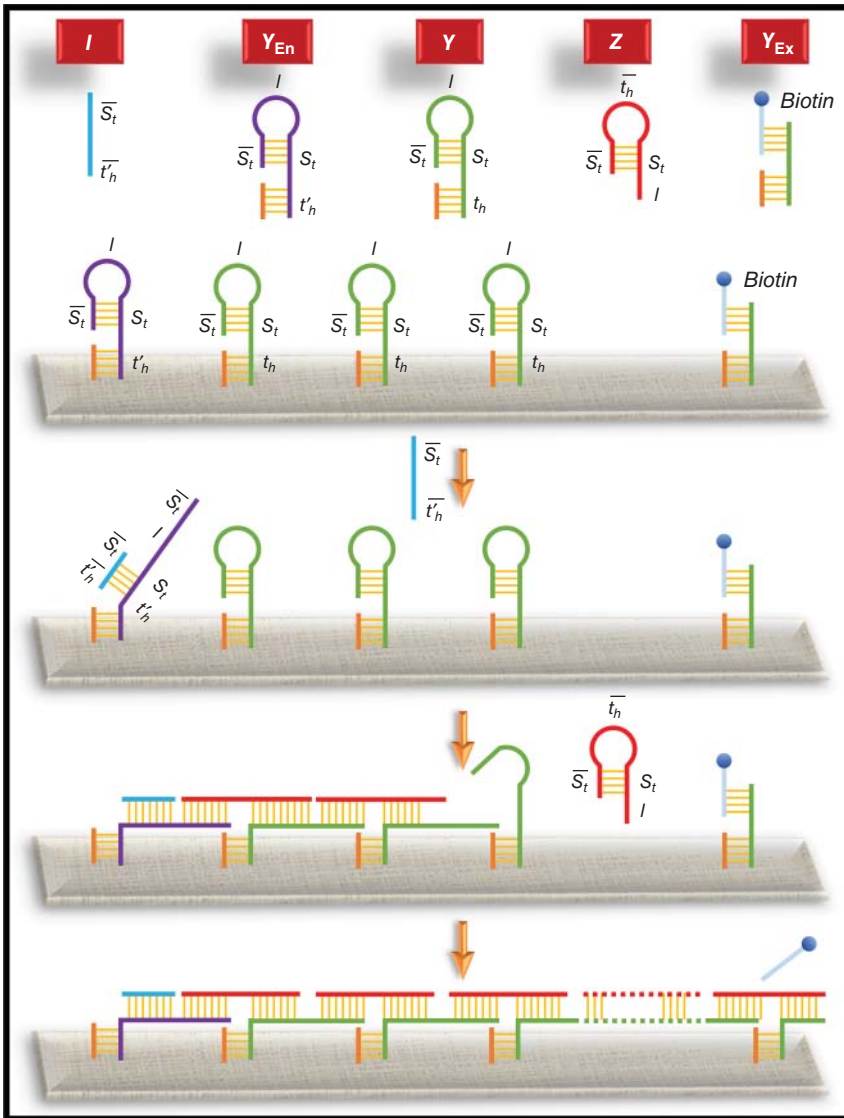
Figure 2.7 Chao's single-molecule DNA navigator [13] for solving the maze.

Based on the structure or pattern of the DNA origami, it has multiple applications in different fields of biology, chemistry, physics, computer science, and materials science. Broad application of DNA origami can be seen as in molecular robotics [15, 16], DNA walkers [17], protein function [18, 19] and structure determination [20, 21], single-molecule force spectroscopy [22, 23], nanopore construction [24–26], drug delivery [27, 28], nucleic acid analysis [29–31], enzymatic nanostructure formations [32, 33], and many more.
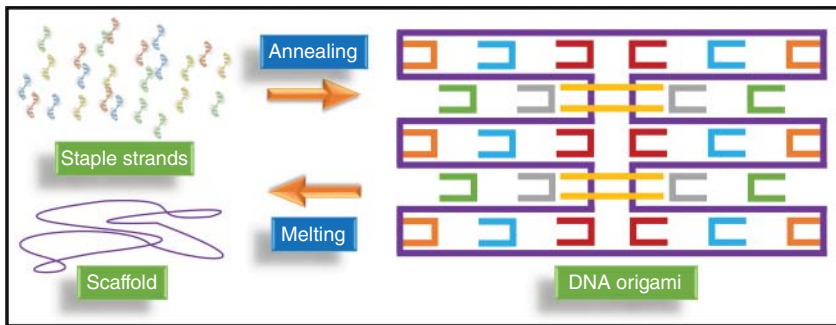
Figure 2.8 DNA origami is a group of linked dsDNA. The structure consists of one scaffold and many staple strands that are complementary to one or two domains on the scaffold.

2.2.8 DNA-Based Data Storage

DNA carries all the genetic information such as the color of eyes and hairs in the form of A, T, G, and C, which is transferred from one generation to another. Inspired by this, the researchers are now looking at the capabilities of the DNA to store the data. Today we are dealing with a large amount of data that is increasing day by day. DNA appears to be the right choice as an alternative for storing the information. In DNA-based data storage, first, the digital information present in the form of bits is encoded in DNA. This encoded information is then processed and decoded back to original binary data.

Church et al. [34] performed an experiment for storing 5.27 MB data generated from his book (contains 53 426 words, 11 images, and 1 JavaScript program) on the DNA. A simple encoding of one bit to one base is used to represent the data on the DNA. The result of this experiment clearly showed that DNA is a good material for storing digital information in addition to other storage media. Goldman et al. [35] stored over five million bits of digital information of the DNA that is later retrieved and reproduced the information with an accuracy between 99.99% and 100%.

Researchers are now looking for high data storage with high accuracy in recovery. Blawat et al. [36] stored and recovered 22 MB of a MPEG compressed movie from DNA with zero errors. Erlich and Zielinski [37] reported another encoding method that can store the 215 petabytes digital data in one gram of DNA. Recently, Organick et al. [38] demonstrated approximately 220 MB digital data storage with random access on the DNA with successful retrieval from it. In 2019, researchers from Microsoft and the University of Washington have demonstrated a fully automated system to encode and decode data in DNA [39].

2.3 Challenges

Though the use of DNA for computing can have the benefit of performing millions of operations simultaneously with very high energy efficiency, it also has

several challenges. The amount of DNA required is substantial even for a simple formulation. Therefore, solving the large size problems becomes impractical owing to the requirement of a large amount of DNA. Unlike the traditional silicon-based computers in which memory reallocation is performed readily, reuse of DNA material is challenging in DNA computing as specific designs of DNA are required.

The success of DNA computing procedures is based on error-free operations of biochemical steps involved. However, the practical operations do involve experimental errors that increase with the increase in the number of steps. Further, the operations involve human interventions during the process. Therefore, solving the bigger size formulations also involves the higher probability of missing the correct answers. Further, increase in formulation size requires extracting the correct solution from the pool involving large number of incorrect solutions. Therefore, the extraction efficiency decreases with increase in formulation size. Also, the large amount of DNA representing the incorrect solutions is discarded as waste. Complete automation of all the biochemical steps is required for building a reliable DNA computer.

Another challenge for DNA computing is its application to real-world problems. Since the data of every problem has to be represented in the form of DNA, the design of DNA is specific to each problem and cannot be used for other problems. Further, for error-free biochemical operations, the DNA designs should have specific GC content with unique (noncomplementary) nucleotide sequence and should lead to a specific structure (i.e. hairpin or linear formation). These requirements reduce the designing flexibility and therefore restrict the application to bigger size formulations. Moreover, real-world problems often involve continuous search spaces with multiple optimal solutions. For such problems, the existing DNA computing procedures that are originally developed for solving the combinatorial problems involving the discrete search space need to be modified.

In conclusion, DNA computing shows great potential and has many advantages in the field of computing and data storage over conventional computing, primarily due to its ability to perform millions of calculations simultaneously using molecules. Despite this, the DNA computer is far from matching the reliability of conventional silicon-based computer owing to several challenges such as poor scaling and limited ability to handle real-world problems. The comparative analysis of existing DNA computing and data storage models illustrated their pros and cons, which is opening up new directions in materials science and biomedical applications.

Acknowledgment

References

1 Moore, G.E. (1975) Progress in digital integrated circuits. Electron Devices Meeting, 11–13.

2 Adleman, L.M. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266 (5187): 1021–1024.

3 Lipton, R.J. (1995). DNA solution for hard computational problems. *Science* 268 (5210): 542–545.

4 Smith, L.M., Corn, R.M., Condon, A.E. et al. (1998). A surface-based approach to DNA computation. *J Comput Biol* 5 (2): 255–267.

5 Sakamoto, K., Gouzu, H., Komiya, K. et al. (2000). Molecular computation by DNA hairpin formation. *Science* 288 (5469): 1223–1226.

6 Liu, Q., Wang, L., Frutos, A.G. et al. (2000). DNA computing on surfaces. *Nature* 403 (6766): 175–179.

7 Braich, R.S., Chelyapov, N., Johnson, C. et al. (2002). Solution of a 20-variable 3-SAT problem on a DNA computer. *Science* 296 (5567): 499–502.

8 Kahan, M., Gil, B., Adar, R., and Shapiro, E. (2008). Towards molecular computers that operate in a biological environment. *Physica D* 237 (9): 1165–1172.

9 Bell, S.A., McLean, M.E., Oh, S.K. et al. (2003). Synthesis and characterization of covalently linked single-stranded DNA oligonucleotide-dendron conjugates. *Bioconjug Chem* 14 (2): 488–493.

10 Shagin, D.A., Shagina, I.A., Zaretsky, A.R. et al. (2017). A high-throughput assay for quantitative measurement of PCR errors. *Sci Rep* 7 (1): 1–11.

11 Ouyang, Q., Kaplan, P.D., Liu, S., and Libchaber, A. (1997). DNA solution of the maximal clique problem. *Science* 278 (5337): 446–449.

12 Faulhammer, D., Cukras, A.R., Lipton, R.J., and Landweber, L.F. (2000). Molecular computation: RNA solutions to chess problems. *Proc Natl Acad Sci* 97 (4): 1385–1389.

13 Chao, J., Wang, J., Wang, F. et al. (2019). Solving mazes with single-molecule DNA navigators. *Nat Mater* 18 (3): 273–279.

14 Rothemund, P.W. (2006). Folding DNA to create nanoscale shapes and patterns. *Nature* 440 (7082): 297–302.

15 Krishnan, Y. and Simmel, F.C. (2011). Nucleic acid based molecular devices. *Angew Chem Int Ed* 50 (14): 3124–3156.

16 Yurke, B., Turber, A.J., Mills, A.P. Jr. et al. (2000). A DNA-fuelled molecular machine made of DNA. *Nature* 406: 605–608.

17 He, Y. and Liu, D.R. (2010). Autonomous multistep organic synthesis in a single isothermal solution mediated by a DNA walker. *Nat Nanotechnol* 5 (11): 778–782.

18 Funke, J.J. and Dietz, H. (2016). Placing molecules with Bohr radius resolution using DNA origami. *Nat Nanotechnol* 11 (1): 47–52.

19 Hariadi, R.F., Appukutty, A.J., and Sivaramakrishnan, S. (2016). Engineering circular gliding of actin filaments along myosin-patterned DNA nanotube rings to study long-term actin-myosin behaviors. *ACS Nano* 10 (9): 8281–8288.

20 Douglas, S.M., Chou, J.J., and Shih, W.M. (2007). DNA-nanotube-induced alignment of membrane proteins for NMR structure determination. *Proc Natl Acad Sci U S A* 104 (16): 6644–6648.

21 Martin, T.G., Bharat, T.A.M., Joerger, A.C. et al. (2016). Design of a molecular support for cryo-EM structure determination. *Proc Natl Acad Sci U S A* 113 (47): E7456–E7463.

22 Shrestha, P., Jonchhe, S., Emura, T. et al. (2017). Confined space facilitates G-quadruplex formation. *Nat Nanotechnol* 12 (6): 582–588.

23 Kilchherr, F., Wachauf, C., Pelz, B. et al. (2016). Single-molecule dissection of stacking forces in DNA. *Science* 353 (6304): aaf5508.

24 Venkatesan, B.M. and Bashir, R. (2011). Nanopore sensors for nucleic acid analysis. *Nat Nanotechnol* 6 (10): 615–624.

25 Bell, N.A.W., Keyser, U.F., Puchner, E.M. et al. (2014). Nanopores formed by DNA origami: a review. *Fed Eur Biochem Soc* 588: 3564–3570.

26 Krishnan, S., Ziegler, D., Arnaut, V. et al. (2016). Molecular transport through large-diameter DNA nanopores. *Nat Commun* 7: 1–7.

27 Ouyang, X., Li, J., Liu, H. et al. (2013). Rolling circle amplification-based DNA origami nanostructrures for intracellular delivery of immunostimulatory drugs. *Small* 9 (18): 3082–3087.

28 Ora, A., Järvihaavisto, E., Zhang, H. et al. (2016). Cellular delivery of enzyme-loaded DNA origami. *Chem Commun* 52 (98): 14161–14164.

29 Endo, M. and Sugiyama, H. (2014). Single-molecule imaging of dynamic motions of biomolecules in DNA origami nanostructures using high-speed atomic force microscopy. *Acc Chem Res* 47 (6): 1645–1653.

30 Rajendran, A., Endo, M., and Sugiyama, H. (2012). Single-molecule analysis using DNA origami. *Angew Chem Int Ed* 51 (4): 874–890.

31 Wang, D., Fu, Y., Yan, J. et al. (2014). Molecular logic gates on DNA origami nanostructures for microRNA diagnostics. *Anal Chem* 86 (4): 1932–1936.

32 Linko, V., Nummelin, S., Aarnos, L. et al. (2016). DNA-based enzyme reactors and systems. *Nanomaterials* 6 (8): 139.

33 Fu, J., Liu, M., Liu, Y., and Yan, H. (2012). Spatially-interactive biomolecular networks organized by nucleic acid nanostructures. *Acc Chem Res* 45 (8): 1215–1226.

34 Church, G.M., Gao, Y., and Kosuri, S. (2012). Next-generation digital information storage in DNA. *Science* 337 (6102): 1628–1628.

35 Goldman, N., Bertone, P., Chen, S. et al. (2013). Toward practical high-capacity low-maintenance storage of digital information in synthesised DNA. *Nature* 494 (7435): 77–80.

36 Blawat, M., Gaedke, K., Hütter, I. et al. (2016). Forward error correction for DNA data storage. *Procedia Comput Sci* 80: 1011–1022.

37 Erlich, Y. and Zielinski, D. (2017). DNA fountain enables a robust and efficient storage architecture. *Science* 355 (6328): 950–954.

38 Organick, L., Ang, S.D., Chen, Y.-J. et al., and others (2018). Random access in large-scale DNA data storage. *Nat Biotechnol* 36 (3): 242.

39 Takahashi, C.N., Nguyen, B.H., Strauss, K., and Ceze, L. (2019). Demonstration of end-to-end automation of DNA data storage. *Sci Rep* 9 (1): 1–5.

3

DNA Computing and Circuits

Chuan Zhang [1,2]

[1] *National Mobile Communications Research Laboratory, Southeast University, Nanjing 211189, China*
[2] *Purple Mountain Laboratories, Nanjing 211189, China*

3.1 From Theory to DNA Implementations

One of the critical challenges of Moore's law is the physical limits of transistor scaling. To this end, alternative non-silicon substitutes have been researched, among which are quantum computing, spintronic computing, and DNA computing.

Computing using DNA materials has been studied in the last few decades. Different from traditional silicon-based computing, DNA computing is inherently massively parallel, molecular scale, and well suited for complex computing. The theoretical analysis of such computing usually builds on an abstraction model of DNA reactions, chemical reaction networks (CRNs) [1–5]. Based on such a model, there are mapping methods that can directly translate programmed CRNs to experimentally implementable DNA reactions [6, 7]. Also, to enable the construction of more complex systems, compilers have also been developed [8, 9]. Apart from that, some researchers also tried to establish instruction sets based on DNA reactions [10]. Overall, researchers are constructing DNA computing systems in a manner similar to constructing early computers, and more progress in this area can be expected in the near future.

CRNs have been proven a type of effective computation tool for DNA computing [1–5]. Stochastic chemical reaction networks (SCRNs) can well model the interactions among a small number of molecules [11] and are Turing-universal [1]. The relation between the SCRN model and the conventional concepts in computer science has already been established by [2]. For both time/space bounded and unbounded computations, SCRNs are capable of simulating Turing machines with time complexity under an asymptotic upper bound and error probability under an upper bound hence are Turing-universal. Figure 3.1 shows the simulation of a register machine as an example [1]. SCRNs are also able to stably compute functions under an asymptotic upper bound of time if and only if the functions have semilinear graphs [3]. Chemical reaction computer, defined as a CRN with initial settings, is thus a powerful computation tool. Regarding the computation correctness issues, an error correction

Reactions	Logical Function
Inc: $C + S_i \rightarrow S_j + M_r + C$	*Inc(i,r,j)*
Dec: $S_i + M_r \rightarrow S_j$ *Dec: $C_i + S_i \rightarrow S_k + C_l$*	*dec(i,r,j,k)*

(a)

Reactions	Catalysts
$C_l \rightarrow C_{l-1}$ $C_{l-1} \rightarrow C_l$	A^* A
...	
$C_3 \rightarrow C_2$ $C_2 \rightarrow C_3$	A^* A
$C_2 \rightarrow C_1$ $C_1 \rightarrow C_2$	A^* A

(b)

Figure 3.1 The simulation of a register machine. (a) Simulation of bounded register machine. (b) Clock module used to simulate the Turing machine. Source: From Soloveichik et al. [1]. Reproduced with the permission of Springer Nature.

method in the construction of CRN simulation of register machines has been proposed [4]. This validates that Turing-universal probability 1 computation (the probability of producing a correct answer is 1 as time approaches infinity) can be implemented in CRNs. A more recent work [5] addresses the composability problem in continuous CRNs. A method to construct composable CRNs is proposed to replace the error-prone direct cascade of different CRNs.

With such powerful computing abilities, CRNs have the potential to build complex functional computing systems. However, it is difficult to rely on manual design when building large biocomputing systems; hence there are attempts to build corresponding compilers [8, 9]. Like compilers of modern programming languages, these tools can synthesize CRNs based on a higher-level abstraction of algorithms. Syntax of hardware description language like Verilog HDL may be used [12], where users can utilize the previously defined modules to construct their systems in a manner similar to digital hardware design. There are also attempts to synthesize CRNs from more software-like descriptions. One critical technique is to map control flows like linear flow, branch statement, and loop statement to CRNs [9]. Another recently proposed tool [8] integrates basic arithmetic operations like additions and subtractions and is able to compile codes written in the high-level description language called CRN++. There are some simulation results of such synthesized CRNs, showing that CRNs can well perform computation as intended. Besides synthesizing CRNs from manually written codes, by formulating the problem of CRN design (including design of reactions and related parameters) as a satisfiability modulo theory (SMT) problem and solving this problem by existing mathematical toolkits, CRNs that satisfy user specifications can be directly synthesized [13] (Figure 3.2). Apart from CRN compilers, Thubagere et al. [14] targets the compilation of lower-level DNA reactions. Using DNA sequences generated by the compiler, DNA circuits can be conveniently built. Since the construction of the
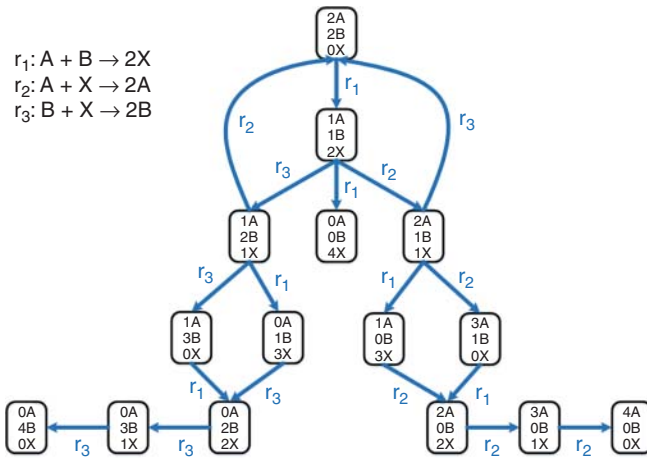
Figure 3.2 State graph with state transition implemented by CRNs. Source: Adapted from Murphy et al. [13].

compiler takes synthesis error into account, the experimental process can also be simplified.

To show the capability of DNA materials to establish computers, many researchers begin with building Turing machines. Apart from the aforementioned simulations of Turing machines in Figure 3.1 [1], there is a universal molecular Turing machine based on site-directed mutagenesis [15]. Qian et al. employ a "history-free" method to implement CRNs, based on which the authors implement irreversible and reversible stack machines [16]. Besides the attempts to build Turing machines, some researchers start from the design of DNA strand displacement reactions and propose instruction sets for DNA computers. The work called SIMD||DNA (Single Instruction Multiple Data DNA) [10] leverages the cascade of DNA strand displacement reactions. By introducing a set of binary encoding rules that utilize different arrangements of upper DNA strands of DNA complexes to represent "1" and "0," data can be stored in DNA "registers" (multistranded complexes). When adding auxiliary single DNA strands that equal to executing instructions, the upper single DNA strands can be attached/displaced/detached from the DNA complexes, hence changing the encoded data stored in DNA "registers." An example of Rule 110 Automata is shown in Figure 3.3.

With the computing power of CRNs, one critical issue is how to implement such formal reactions in the real world using DNA reactions. Soloveichik et al. propose a DNA reaction substrate in [6], based on which theoretically designed reactions with less than three reactants can be readily mapped to DNA strand displacement reactions. The kinetic features of original formal reactions are approximately retained, as proved by mass action kinetic analysis. For bimolecular reactions, Qian et al. propose a design of DNA strand displacement reactions [16] to map such formal reactions. The model can address both reversible and irreversible formal reactions; unimolecular reactions and reactions with higher orders can be similarly constructed. Experimental work
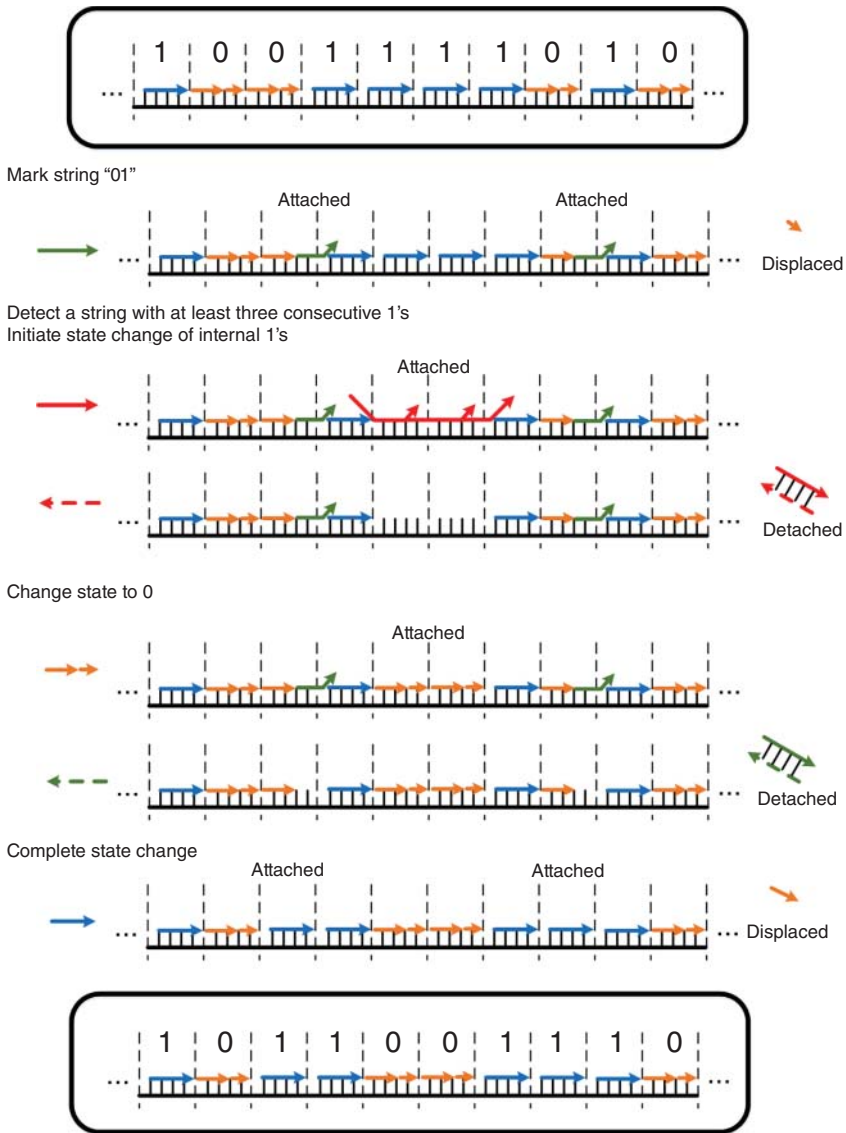
Figure 3.3 Implementation of Rule 110 Automata. Two different arrangements of upper single DNA strands within each cell (slices of DNA complex separated by dashed lines) represent "1" and "0," respectively. When executing an instruction, some specific DNA strands are added to the system to initiate the attachment, detachment, and displacement of upper DNA strands. The combination of such instructions changes the state of related cells, resulting in state transitions of the automata after the execution of all instructions. Source: Modified from Wang et al. [10].

[7], which shows that bimolecular reactions can be mapped to DNA strand displacement reaction cascade, is based on a similar mechanism. Fluorophore experiments prove that the cascaded DNA reactions overall simulate the initial formal reactions. The diagrams of such mapping models are provided in Figure 3.4.

In conclusion, from the perspective of theoretical computer science, chemical materials are powerful computing tools. It is capable of performing universal computing, and its programmability can be utilized by designers to create computing systems with desired functions. To fully exploit the computing power of chemical materials, there are researches focusing on the features and organization methods of CRNs and the wet experimental implementation of such systems. More applications of molecular computing are expected in future works.

3.2 Application-Specific DNA Circuits

In order to take advantage of the merits of DNA computing, researchers do not only use DNA computing to implement Turing machine but also try to employ it for specific applications, especially those involving complex problems. There are two basic questions arising when applying DNA computing for specific applications. One is encoding the real-world signals to the input variables for CRNs and then decoding them back into real-world signals after computation (Figure 3.5). The other one is how to design chemical reactions for specific functions.

In order to use the concentrations of molecules to represent variables' values, researchers have considered three types of encoding – the direct representation, the dual-rail representation [18], and the fractional representation [17]. In the direct representation, values of all variables are indicated by concentrations of molecular types. In the dual-rail representation, the difference between concentrations of two species represents the value of a variable. In the fractional representation, values of variables are determined by ratios of two molecular species in the reaction system. To be specific, e.g. if (X_0, X_1) is the fractional representation for a variable x, its value is $x = [X_1]/([X_0] + [X_1])$, where $[\cdot]$ denotes concentrations of molecular types.

After defining the various input signals in a biochemical system with one of the three types of encoding representations mentioned above, the system can be solved through ordinary differential equations (ODEs). For CRN analysis, mass action kinetics is considered as a proper kinetic scheme [19]. For mass action kinetics, the rate of a chemical reaction is proportional to the product of concentrations of reactants. For instance, consider a reaction given by

$$X_1 + X_2 \rightarrow X_3$$
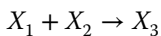
Since the reaction fires at a rate proportional to $[X_1][X_2]$, or [rate of reaction] $\propto k[X_1][X_2]$, where k is rate constant associated with the reaction, we can

(a)

(b)

(c)
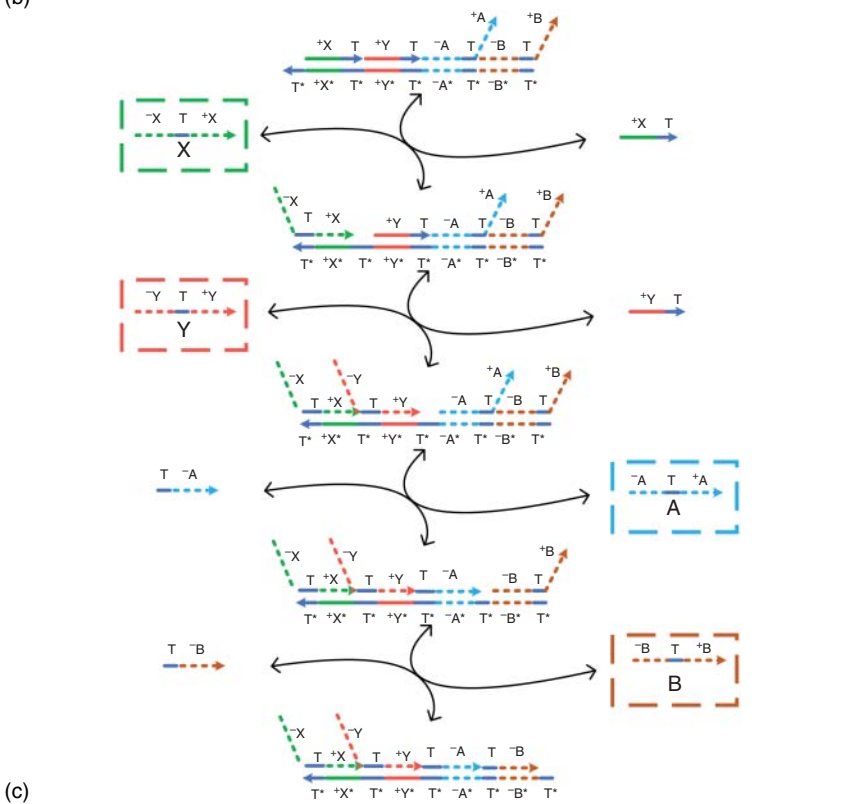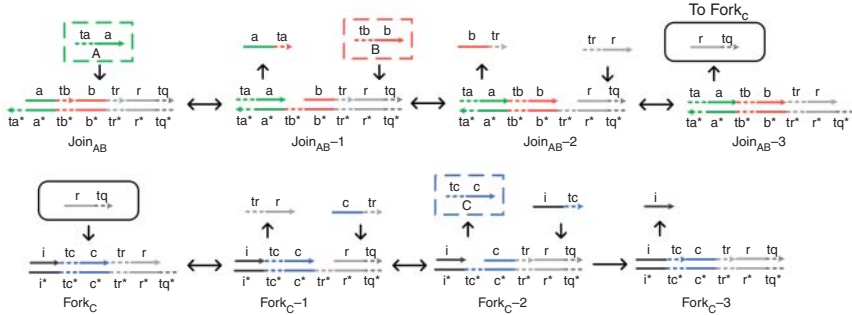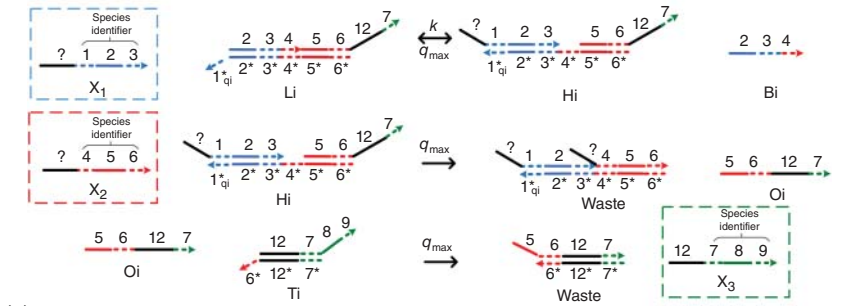
Figure 3.4 Diagrams of the DNA implementation models. (a) Mapping model of bimolecular reactions in [6]. The two reactants are taken into the system using the first two reactions: one reversible and one irreversible, and the third irreversible reaction displaces the products of the formal reaction. Source: Modified from Soloveichik et al. [6]. (b) Reaction design of A + B → C. By cascading several displacement reactions, the output is eventually displaced, and kinetic features are well reserved. Source: Adapted from Chen et al. [7]. (c) Implementation of bimolecular reversible formal reaction. Source: Adapted from Qian et al. [16].

Figure 3.5 The system performing encoding, computing, and decoding signals in CRNs. Source: From Salehi et al. [17]. Reproduced with the permission of American Chemical Society.

model the reaction by ODEs as follows:

$$\frac{d[X_1]}{dt} = -k[X_1][X_2]$$

$$\frac{d[X_2]}{dt} = -k[X_1][X_2]$$

$$\frac{d[X_3]}{dt} = k[X_1][X_2]$$

ODE simulation is a continuous deterministic model of chemical kinetics.

An alternative approach to achieving mass action kinetics modeling is referred to as stochastic simulation [20]. Compared with deterministic modeling, stochastic simulation is discrete and stochastic, and the computation is based on probabilities.

Many researchers have investigated methods to implement digital logic with molecular reactions, including combinational components and sequential components. For combinational components, the inverter is the simplest but very important logic gate since other more complicated structures such as NAND gates, adders, and multipliers will make use of it. In a biochemical system, it can be implemented by implementing the transfers between the molecular types representing 0 and 1, respectively [21]. The simplification methods for digital combinational logic have been studied in [22].

Take the AND gate as an example for two-input logic gates implemented by molecular reactions [21]. Suppose the inputs of the gate are X and Y and the output is Z, respectively. The inputs and output signals are represented by the concentration of X_0/X_1, Y_0/Y_1, and Z_0/Z_1. If the value of X is 0, then all X_1 will be transferred to X_0. According to the target logic function, the chemical reactions are designed as

$$X_0 + Z_1 \rightarrow X_0 + Z_0$$

$$Y_0 + Z_1 \rightarrow Y_0 + Z_0$$

$$X_1 + Y_1 \rightarrow X_1 + Y_1 + Z_1'$$

$$2Z_1' \rightarrow \emptyset$$

$$Z_1' + Z_0 \rightarrow Z_1$$
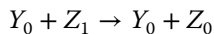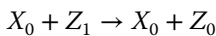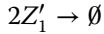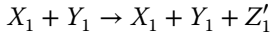
where Z_1' represents an indicator to generate Z_1 and can be transferred to an external sink indicated by the fourth reaction. The analysis of inverter and AND gate implementation with molecular reactions can be applied to explain other complex gates such as NOR, XOR, and NAND [21], which form the basic blocks for modules like multipliers and digital signal processors.

For sequential components, all modules are under control of the clock signals. Sequential digital logic circuits can be classified into two categories, synchronous circuits and asynchronous circuits, depending on whether the circuits are governed by a global clock or not. A sustained-chemical-oscillator-based synchronization mechanism is introduced to implement synchronous circuits with molecular reactions, which have been widely studied by the synthetic biology community. An example is "red-green-blue" (RGB) oscillator [23–25], which is first proposed by [23] and can be used to establish an order for the transformation of molecular quantities in the counter implemented by molecular reactions. The RGB oscillator is also useful for generating a global clock as the designers wish. Reactions in an RGB oscillator are assigned to one of the three categories – red, green, and blue. Quantities are transformed between color categories according to the absence of molecules in the third category as (Figure 3.6a)

$$b + R \rightarrow G + b$$

$$r + G \rightarrow B + r$$

$$g + B \rightarrow R + g$$
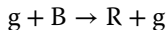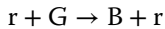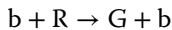
Here, R, G, and B are introduced molecular types. And r, g, and b are the "absence indicators" corresponding to R, G, and B, respectively, and are continually generated as

$$\emptyset \rightarrow r$$

$$R + r \rightarrow R$$
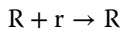
The feature of indicators quickly consumed by corresponding signal molecules assures that the succeeding phase cannot begin unless all reactions in a given phase have completed (Figure 3.6b). With the aid of such clock signals, analog circuits for basic arithmetic, like addition, subtraction, multiplication, and division, can be implemented with molecular reactions [27].

Asynchronization circuits are implemented by locking the computation of biochemical modules. In asynchronous circuit designs, it is analogous to hand-shaking mechanisms. By introducing a specific molecular type, the module's key, to each module, the sequence of reactions is prevented from firing without the key, thus under proper control [28].
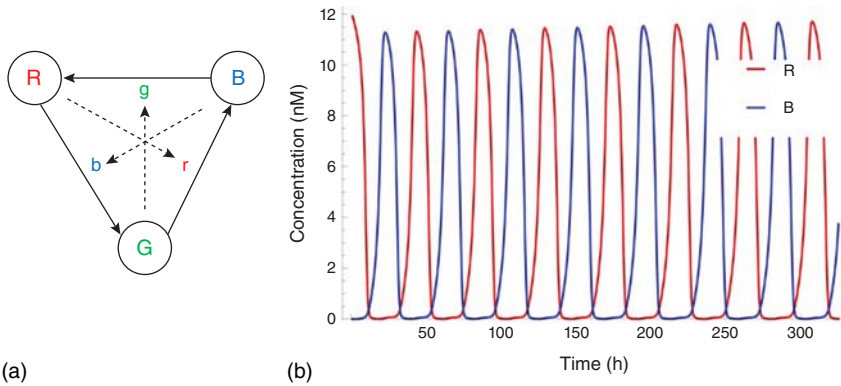
Figure 3.6 (a) Sequence of reactions for the three-phase clock based on the RGB oscillator. Source: Adapted from Kharam et al. [26]. (b) ODE-based simulation of the chemical kinetics of the proposed *N*-phase clock (here *N* = 2), where the amplitude and frequency of oscillation waves can be adjusted. Source: From Jiang et al. [25]. Reproduced with the permission of American Chemical Society.

Several researchers have turned their attention to the implementation of more complicated functions in specific systems [29–32]. Salehi et al. [29] first points out that stochastic logic could be converted to molecular designs that can be readily utilized in the design of molecular filters and channel decoders. Using fractional coding [17], DNA computing-based vector machine and artificial neural networks have been proposed in [30–32]. To resolve several complex design issues raised by nonlinear functions, Taylor series are applied to approximate a function with a polynomial [17]. A polynomial is a mathematical expression involving variables and coefficients; its operations are limited to multiplication, subtraction, addition, and nonnegative integer exponents of variables. The nucleus of designing functions with chemical reactions is to implement addition, subtraction, and multiplication with molecular reactions.

Take the addition $a + b = c$ as an example. The corresponding chemical reactions are designed as

$$A \rightarrow C$$

$$B \rightarrow C$$

where the concentrations of molecular types A, B, and C represent the values of *a*, *b*, and *c*, respectively. As both the inputs A and B are transferred to C, the concentration of C is the sum of the initial concentrations of A and B, namely, the values of *a* and *b*.

Digital signal processing (DSP) modules such as filters and fast Fourier transform (FFT) processors are of good importance and perform a wide variety of functions. CRN is a novel alternative to traditional application-specific integrated circuits (ASICs) to implement DSP algorithms since they are also applicable to the field of molecular computing. Methods for implementing DSP algorithms using synchronous, RGB, and asynchronous schemes have been demonstrated in detail by [23, 24, 33, 34] (Figures 3.7 and 3.8).
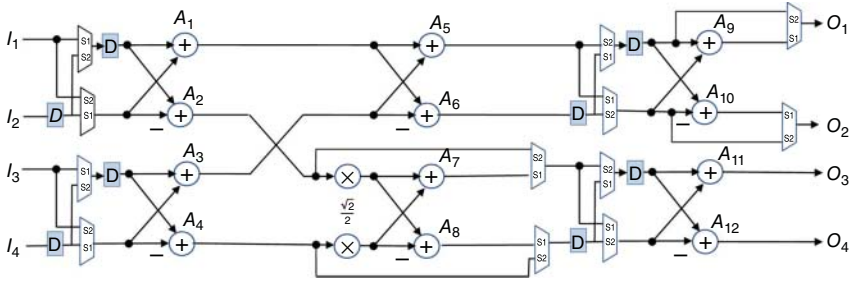
Figure 3.7 A folded eight-point four-parallel real-valued FFT processor. Source: From Jiang et al. [25]. Reproduced with the permission of American Chemical Society.
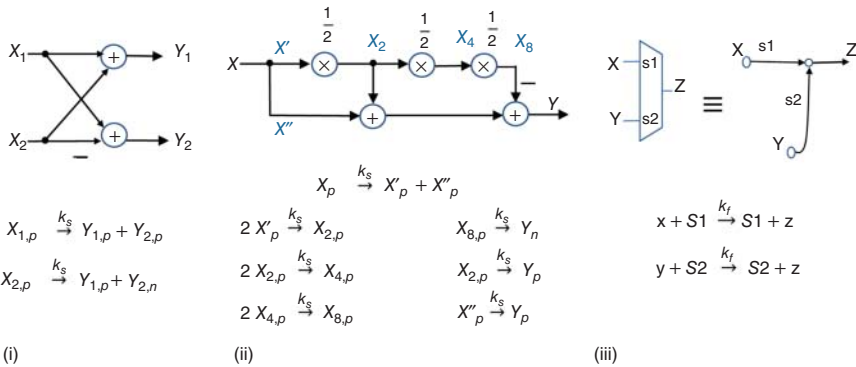
Figure 3.8 Molecular reactions for each element of the FFT processor in Figure 3.7. Source: From Jiang et al. [25]. Reproduced with the permission of American Chemical Society.

In the field of synthetic chemical circuits, DNA-based logic gates played a crucial role. One interesting idea is to develop DNA circuit construction techniques and scale it up, which can help us to build larger molecular circuits systematically. In [35], starting from simple building blocks called DNA gate motif, the authors developed an abstract model for the design of large-scale DNA circuits. The authors show that following the proposed method, circuits such as feed forward digital circuits can be effectively constructed. Stemming from this construction method, Qian and Winfree [36] presents a design of relatively complex digital logic circuits in DNA computing systems. The design is based on dual-rail representation [18] and "seesaw" gate motif [35, 36], which enables logic operations such as AND, OR, and NOT to be implemented by DNA strand displacement reactions. Such logic gates are proved cascadable, and a four-bit square root DNA circuit is constructed as a design case to validate the scalability and computing capability of DNA logic circuits.

In 2011, an experimental method to implement neural network computation was proposed [37]. With DNA displacement reactions that implement linear threshold function, traditional computation in neural networks can be performed in molecular computing systems. A simple but interesting experiment called Hopfield associative memory validates the functionality of the system.

In 2018, a more powerful DNA reaction implementation of neural network computing was provided [38]. Based on winner-take-all mechanism [39], computation in neural networks such as weighted summation and thresholding of binary input data can be performed in DNA reactions with high accuracy. An example of recognizing handwritten numbers from Modified National Institute of Standards and Technology (MNIST) database [40] shows that such seemingly complex computations are able to be handled by DNA strand displacement reactions. Though training in these two works is performed in silicon-based computers, the applications show the potential of DNA materials in building scaled functional computing systems. In 2018, implementation of probabilistic switching circuits based on DNA strand displacement reactions was proposed [41]. In the fabricated system, input signals can be converted to output signals with predefined probabilities, and experiments proved the functionality of such DNA circuits.

To conclude, the designs of molecular computing systems present a design hierarchy. Starting from employing data representation methods such as fractional encoding, researchers build basic molecular circuit modules such as logic gates and clock generator. Based on that, more complex functions, e.g. DSP and neural network computation, can be realized. There are various applications in silicon-based hardware that can be implemented by the interaction of chemical materials, which will be the main focus of future research.

Acknowledgments

References

1 Soloveichik, D., Cook, M., Winfree, E., and Bruck, J. (2008). *Nat. Comput.* 7: 615–633.

2 Cook, M., Soloveichik, D., Winfree, E., and Bruck, J. (2009). Programmability ofchemical reaction networks. In: *Algorithmic Bioprocesses*, Natural Computing Series (eds. A. Condon, D. Harel, J.N. Kok, et al.), 543–584. Berlin Heidelberg: Springer.

3 Chen, H., Doty, D., and Soloveichik, D. (2014). *Nat. Comput.* 13: 517–534.

4 Cummings, R., Doty, D., and Soloveichik, D. (2016). *Nat. Comput.* 15: 245–261.

5 Chalk, C., Kornerup, N., Reeves, W., and Soloveichik, D. (2018). Composable rate-independent computation in continuous chemical reaction networks.

In: *International Conference on Computational Methods in Systems Biology*, 256–273. Springer.

6 Soloveichik, D., Seelig, G., and Winfree, E. (2010). *Proc. Natl. Acad. Sci. U.S.A.* 107: 5393–5398.

7 Chen, Y., Dalchau, N., Srinivas, N. et al. (2013). *Nat. Nanotechnol.* 8: 755–762.

8 Vasic, M., Soloveichik, D., and Khurshid, S. (2020). CRN++: molecular programming language. *Nat. Comput.* 19. Springer: 1–17.

9 Huang, D., Jiang, J.R., Huang, R., and Cheng, C. (2012). Compiling program control flows into biochemical reactions. In: *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 361–368.

10 Wang, B., Chalk, C., and Soloveichik, D. (2019). DNA: single instruction, multiple data computation with DNA strand displacement cascades. In: *International Conference on DNA Computing and Molecular Programming*, 219–235. Springer.

11 McQuarrie, D.A. (1967). *J. Appl. Probab.* 4: 413–478.

12 Shea, A., Fett, B., Riedel, M.D., and Parhi, K. (2010). Writing and compiling code into biochemistry. In: *Biocomputing 2010*, 456–464. World Scientific.

13 Murphy, N., Petersen, R., Phillips, A. et al. (2018). *J. R. Soc. Interface* 15: 20180283.

14 Thubagere, A.J., Thachuk, C., Berleant, J. et al. (2017). *Nat. Commun.* 8: 14373.

15 Beaver, D. (1995). *DNA Based Comput.* 27: 29–36.

16 Qian, L., Soloveichik, D., and Winfree, E. (2011). Efficient Turing-universal computation with DNA polymers. In: *DNA Computing and Molecular Programming (Lecture Notes in Computer Science)*, vol. 6518, 123–140.

17 Salehi, S.A., Parhi, K.K., and Riedel, M.D. (2017). Chemical reaction networks for computing polynomials. *ACS Synth. Biol.* 6 (1): 76–83, ACS Publications.

18 Chen, H.L., Doty, D., and Soloveichik, D. (2014). *ACM Conference on Innovations in Theoretical Computer Science*, 313–326.

19 Horn, F. and Jackson, R. (1972). *Arch. Ration. Mech. Anal.* 47: 81–116.

20 Cheng, B. and Riedel, M. (2009). Stochastic transient analysis of biochemical systems and its application to the design of biochemical logic gates. In: *Biocomputing 2009*, 4–14.

21 Jiang, H., Riedel, M.D., and Parhi, K.K. (2013). Digital logic with molecular reactions. In: *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 721–727. IEEE.

22 Ge, L., Zhong, Z., Wen, D. et al. (2016). *IEEE Trans. Mol. Biol. Multi-Scale Commun.* 3: 33–47.

23 Jiang, H., Kharam, A.P., Riedel, M.D., and Parhi, K.K. (2010). A synthesis flow for digital signal processing with biomolecular reactions. In: *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 417–424. IEEE.

24 Jiang, H., Riedel, M.D., and Parhi, K.K. (2011). Synchronous sequential computation with molecular reactions. In: *Proceedings of the 48th Design Automation Conference*, 836–841.

25 Jiang, H., Salehi, S.A., Riedel, M., and Parhi, K.K. (2013). *ACS Synth. Biol.* 2: 245–254.

26 Kharam, A., Jiang, H., Riedel, M., and Parhi, K.K. (2011). Binary counting with chemical reactions. In: *Biocomputing 2011*, 302–313.

27 Li, M., Ge, L., You, X., and Zhang, C. (2018). Basic arithmetics based on analog signal with molecular reactions. In: *2018 IEEE International Conference on Communications (ICC)*, 1–6. IEEE.

28 Fett, B. and Riedel, M.D. (2008). Module locking in biochemical synthesis. In: *2008 IEEE/ACM International Conference on Computer-Aided Design*, 758–764. IEEE.

29 Salehi, S.A., Liu, X., Riedel, M.D., and Parhi, K.K. (2018). *Sci. Rep.* 8: 8312.

30 Liu, X. and Parhi, K.K. (2020). Molecular and DNA artificial neural networks via fractional coding. *IEEE Trans. Biomed. Circuits Syst.* 14 (3): 490–503.

31 Liu, X. and Parhi, K.K. (2019). Training DNA perceptrons via fractional coding. In: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE.

32 Liu, X. and Parhi, K.K. (2019). Computing radial basis function support vector machine using DNA via fractional coding. In: *Proceedings of the 56th Annual Design Automation Conference 2019*, 1–6.

33 Salehi, S.A., Jiang, H., Riedel, M.D., and Parhi, K.K. (2015). *IEEE Trans. Mol. Biol. Multi-Scale Commun.* 1: 249–264.

34 Salehi, S.A., Riedel, M.D., and Parhi, K.K. (2014). Asynchronous discrete-time signal processing with molecular reactions. In: *2014 48th Asilomar conference on signals, systems and computers*, 1767–1772. IEEE.

35 Qian, L. and Winfree, E. (2011). *J. R. Soc. Interface* 8: 1281–1297.

36 Qian, L. and Winfree, E. (2011). *Science* 332: 1196–1201.

37 Qian, L., Winfree, E., and Bruck, J. (2011). *Nature* 475: 368–372.

38 Cherry, K.M. and Qian, L. (2018). *Nature* 559: 370–376.

39 Maass, W. (2000). *Neural Comput.* 12: 2519–2535.

40 Li, D. (2012). *IEEE Signal Process Mag.* 29: 141–142.

41 Wilhelm, D., Bruck, J., and Qian, L. (2018). *Proc. Natl. Acad. Sci. U.S.A.* 115: 903–908.

4

Connecting DNA Logic Gates in Computational Circuits

Dmitry M. Kolpashchikov[1,2] and Aresenij J. Kalnin[3]*

[1] *University of Central Florida, Chemistry Department, 4111 Libra Drive, Orlando, FL 32816-2366, USA*
[2] *University of Central Florida, Burnett School of Biomedical Sciences, 6900 Lake Nona Blvd, Orlando, FL 32816, USA*
[3] *SCAMT Institute, Laboratory of Molecular Robotics and Biosensor Materials, 9 Lomonosova Street, St. Petersburg, 191002, Russian Federation*

4.1   DNA Logic Gates in the Context of Molecular Computation

Electronic microprocessor systems are based on semiconductor logic gates, which employ electronic input and output signals and power supplies [1]. A critical feature, which contributes to the undoubted success of electronic circuits, is input–output signal homogeneity: the same electron voltage value emerging as an output of one gate can be admitted as an input of another gate. Such connections of logic gates can achieve selected functions of varying complexity. This very large-scale integration is a crucial component of modern silicon processors [2, 3]. The development of more powerful microprocessors depends on continued progress in miniaturizing their components. However, if current trends continue, conventional silicon chips will soon reach their physical limits [4]. Several research groups have created molecular ensembles that perform logic operations [2, 3, 5–9]. Even though small-scale integration of logic elements has been achieved, there is still a lack of examples of universal large-scale integration. Therefore, the challenges of component integration must be further addressed to advance the molecular computation field, as well as for its practical implementations [2, 3].

DNA has been considered as an excellent candidate both for *in vitro* computation [10] and as a convenient building block for molecular switches and other devices [11–13]. Pioneered by Stojanovic [14], a great number of nucleic acid-based logic gates of various designs have been proposed in the last 17 years [15–26]. Despite significant progress in the design of individual molecular logic gates, there is still a great challenge in solving the following technical problems: achieving high scale integration of molecular logic units, precise localization of the molecular gates in nano-environment for efficient inter-gate communication,

---

* Email address: Dmitry.Kolpashchikov@ucf.edu

and achieving reusability of DNA hardware [27–30]. Moreover, connecting the gates to an appropriate interface for convenient communication with human users is needed [31, 32].

This chapter describes approaches for connecting DNA logic gates in circuits with the emphasis on (i) deoxyribozyme (Dz) logic gates, (ii) strand displacement (seesaw) logic gates, and (iii) DNA logic gates connected via four-way junctions (4WJs). Most common problems on the way toward creating long chains of communicating DNA logic gates are discussed.

4.2 Connecting Deoxyribozyme Logic Gates

The logic gate designed by Stojanovic et al. [14] took advantage of deoxyribozymes – nonnaturally occurring DNA sequences that catalyze chemical reactions [33, 34]. One of most commonly used deoxyribozyme classes in DNA-based molecular computation is a class of RNA-cleaving deoxyribozymes (RCDZ). The design principles for RCDZ-based two-input AND gate (2iAND) are shown in Figure 4.1. A DNA strand containing an RCDZ catalytic core and two substrate-binding arms is rendered inactive by the two inactivating stems. Binding the two oligonucleotide inputs I1 and I2 to the input-recognition loop modules unwind the inactivating stems, thus enabling binding of a fluorogenic F substrate (F sub). The cleavage of F sub results in fluorescent increase due to the separation of the fluorophore from quencher. Importantly, when added separately, I1 and I2 are unable to restore the catalytic activity of RCDZ, which ensures the appropriate logic behaviours of the 2iAND construct. NOT, OR, ANDNOT, and even three-input ANDNOTANDNOT gates can be designed by following similar principles of RCDZ core activation and inactivation [35, 36]. Splitting the RCDZ core in two halves enables designing the gates that can process up to five inputs (e.g. 5iAND gate shown in Figure 4.1b) [37]. RCDZ logic gates have been used to build complex systems including automatons that can play Tic-Tac-Toe game with human, which utilize over hundreds of logic gates [38, 39] (reviewed in [11]).

The RCDZ logic gates can be connected via cascades of deoxyribozyme-catalyzed reactions. For example, Stojanovic et al. connected YES, NOT, AND, and ANDNOT logic gates to downstream YES RCDZ gate [40]. An example of 2iAND gate is shown in Figure 4.2. Deoxyribozyme ligase-based 2iAND gate, when activated by the two DNA inputs I_A and I_B, can bind the two short strands OUTa and OUTb and covalently link them into a longer oligonucleotide OUT (Figure 4.2c). The latter can be recognized by a downstream RCDZ gate, as shown in Figure 4.2d. Such cascade resulted in a two-layer logic gate integration. Incubation period for up to 60 minutes was required for this system to achieve fluorescent response above the background. The disadvantage of the system is slow release of the output oligonucleotide from the complex with the ligase gate, since the product of ligation has higher affinity to the DNA ligase than the substrates (OUTa and OUTb).

Yashin et al. immobilized RCDZ constructs on microsphere beads together with their substrates (Figure 4.2b) [41]. The substrates were inactivated by complementary strands, which could be removed by inputs I1 and I2,
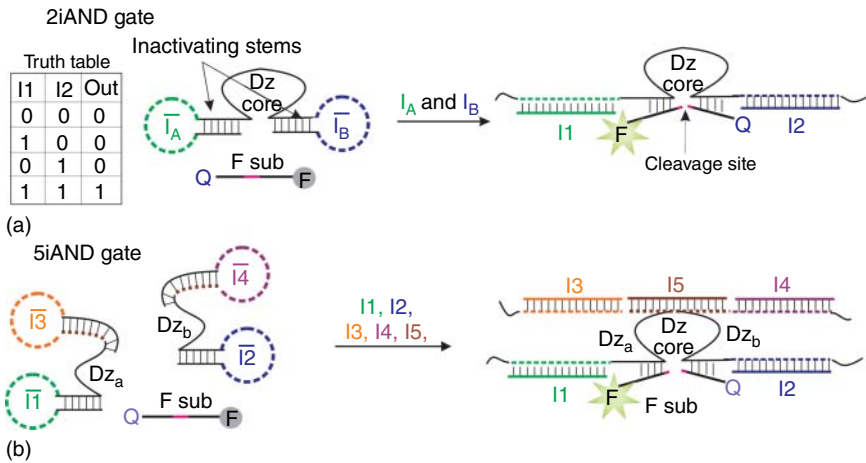
Figure 4.1 Examples of deoxyribozyme-based logic gates. (a) One of the first DNA logic gates: deoxyribozyme (Dz)-based two-input AND gate (2iAND) [14]. Input-recognition modules are in green and blue. Upon hybridization of inputs I_A and I_B, the substrate-binding arms are unblocked, which restores RCDZ activity to cleave a fluorophore- and quencher-labeled reporter substrate (F sub). FAM is fluorescein; Q is a dark quencher of fluorescence. (b) Dz-based five-input AND gate (5iAND). Dz catalytic core regains activity only when all five oligonucleotide inputs (**I1–I5**) are present. **I1**, **I2**, **I3**, and **I4** open the inactivating stems, while **I5** bridges strands Dz_a and Dz_b together to form a catalytic core [37]. Source: Based on Stojanovic et al. [14].

respectively (not shown in Figure 4.2b). The YES beads, activated by certain input combination, could release their oligonucleotide outputs in solution (OUT1 and OUT2), which then could be recognized by downstream logic gates (AND2,3 gate in Figure 4.2b). Three-layer integration was achieved. The advantage of the approach is in the ability to monitor the fluorescent signal from individual beads by flow cytometry. However, hybridization of the bead-immobilized oligonucleotides might be significantly slower than that in solution [42]. Indeed, time needed to observe a strong signal on the last bead in this case was 16 hours [41].

Bone et al. used split cascades based on the most catalytically active 10–23 Dz that enable realizing inactivated RCDZ [43]. This approach can reduce the amount of input required for cascade activation from 20–1000 nM to 20–100 pM [37, 43, 44].

4.3 Connecting Gates Based on DNA Strand Displacement

The design of seesaw gates [53] takes advantage of DNA strand displacement, e.g. the ability of a partial DNA duplex to release one strand upon hybridization of the second strand with an interfering strand that can form more Watson–Crick base pairs than presented in the original duplex (Figure 4.3a). The phenomenon has been used as a probe for the detection of specific nucleic acids in several variations [45–52]. For example, a DNA duplex composed of a 5′-fluorophore-labeled
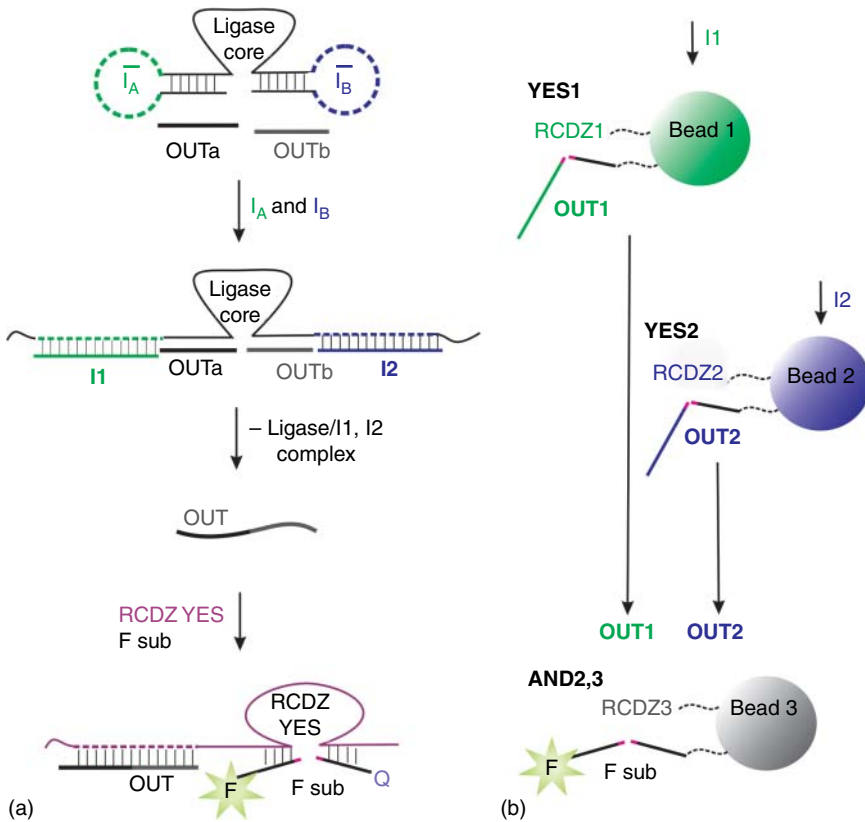
Figure 4.2 AND deoxyribozyme ligase gate connected to YES RCDZ gates. (a) DNA ligase-based two-input AND gate: the ligase Dz is inhibited by two stem-loop structures. Inputs I_A and I_B unwind the inactivating stem-loops and enable the DNA ligase to ligate strands OUTa and OUTb to produce the OUT strand. The OUT strand activates RCDZ for cleavage of F sub followed by fluorescence increase. (b) Beads 1 and 2 contain immobilized RCDZ constructs and their corresponding substrates. They act as YES logic gates: in the presence of the oligonucleotide inputs I1 and I2, they activate RCDZ and release OUT1 and OUT2 oligonucleotides. The released OUT1 and OUT2 can activate AND2,3 gate on the third bead, which can be monitored via fluorescence.

and a 3′-quencher-labeled oligonucleotide strands can be interrogated by a complementary analyte that "pushes" the fluorophore-containing strand in solution (Figure 4.3a), which can be reported as a fluorescent signal in a quantitative real-time polymerase chain reaction (PCR) format [48].

The advantages of this system for DNA logic gate design are the following:

(i) Design simplicity.
(ii) The double-stranded constructs with only short single-stranded overhangs (toeholds) reduce the nonspecific associations between oligonucleotides, which enables usage of many different strands in the same solution.
(iii) Both input and output signals are specific DNA sequences, which enables building cascades of communicated logic gates.
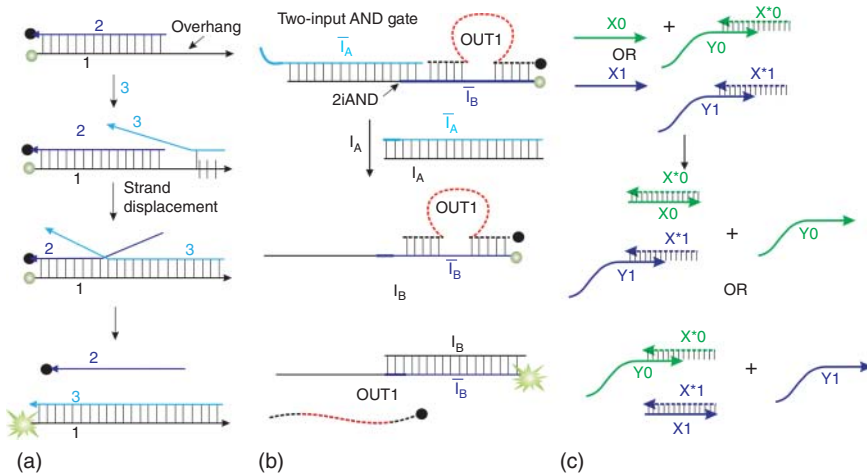
Figure 4.3 Design of strand displacement (seesaw) DNA logic gates [53]. (a) Strand displacement-based sensor for nucleic acid detection. (b) Two-input AND logic gate, which consist of the complex $\bar{\imath}_A$ and $\bar{\imath}_B$ with OUT1, releases OUT1 as an output upon binding to inputs $\bar{\imath}_A$ and $\bar{\imath}_B$. Red dashed line indicates unique sequence independent on the sequence of $\bar{\imath}_A$ and $\bar{\imath}_B$ strands. (c) Dual-rail logic. YES gate made with rail logic. X0 and X1 represent negative or positive input, respectively. Y0 and Y1 represent negative or positive output signal, respectively.

Seelig et al. designed a series of logic gates including AND, NOT, OR, thresholder, and amplifier [53]. A general idea for the design of a 2iAND gate is illustrated by Figure 4.3b. Strand $\bar{\imath}_B$ forms a complex with strands $\bar{\imath}_A$ and OUT1. Strand OUT1 is released free in solution only in the presence of the two oligonucleotide inputs I_A and I_B, which bind strands $\bar{\imath}_A$ and $\bar{\imath}_B$, respectively, forming the "waste" duplex products. Strand OUT1 can then serve as a unique input for the downstream gates due to the presence of a unique sequence (shown as a red fragment in Figure 4.3b). Using this strategy, Seelig et al. demonstrated a five-layer DNA gate integration, which consisted of 11 gates and accepted six inputs [53]. A systematic construction of strand displacement gates can lead to up to 78 gates performing simultaneously [54]. The gates can be used for solving as complex tasks as mimicking natural neural networks [55]. The disadvantages of strand displacement gates include the following: (i) The intensity of signal is proportional to strand concentration, unlike that of RCDZ, in which signal is enhanced by the catalytic action of RCDZ. (ii) DNA interactions are thermodynamically driven, which leads to irreversibility of the circuits or continuous accumulation of the DNA waste products in each operational round. (iii) Strand displacement is slower than hybridization of two single-stranded oligonucleotides.

Instead of interpreting presence of a particular strand as a positive signal and absence of the strand as a negative signal, one can use two sequences as positive and negative signals, respectively. Such method is called dual-rail logic (Figure 4.3c) [56]. Using this method, any gate that consists of AND/OR/NOT gates can be redesigned by using AND and NOT gates [56]. This approach was used for making logic gates suitable for both cascading and multiple operations
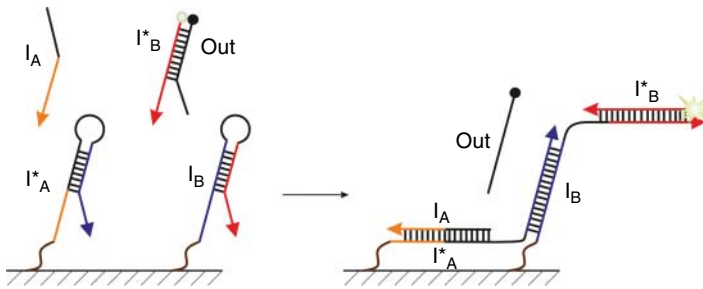
Figure 4.4 DNA strand displacement logic gates attached to a DNA origami tile. Input strand I_A binds to hairpin I^*_A and unwinds it. The unwound end interacts with a closely placed hairpin I_B, which consequently becomes unwound and encounters a fluorophore-labeled strand I^*_B, with fluorescence being attenuated by the quencher-carrying strand Out. I_B reassociates with I^*_B, which leads to increase in fluorescence.

[56, 57]. Specific sequence complementary to one of the signal strands can be introduced in the solution, bind the input sequence, decompose the gate–input complex, and reactivate the gates. It leads to reassociation of the gate with its outputs. After that, a new input can be introduced into the computational system, with the ability to generate the proper output being saved.

The response rate of a DNA computational unit is limited by the time required for an input sequence and a gate to encounter each other in solution (Figure 4.4). To mitigate this limitation, the logic gates can be localized at a short distance from each other, which allows for faster inter-gate communication [27]. A scaffold/substrate, suitable for gate immobilization, can be a DNA tile [27]. Chatterjee et al. localized strand displacement DNA logic gates on a DNA origami tile (Figure 4.4) [58]. Up to eight hairpins could communicate with each other [58], suggesting a possibility to integrate eight layers of DNA logic gates. Co-localization of the circuit elements decreased the computation time from hours to minutes, as compared with solution-based seesaw circuits.

The decrease in signal intensity associated with the increase in the number of integrated gates was observed for the propagation of information through the chain of the conjugated gates. This is an expected limitation of the approach, since the presence of only one oligonucleotide input provides only limited contribution to the stabilization energy for the DNA strand association (e.g. shown in Figure 4.4): additional energy input is needed to ensure robust communication of the conjugated gates. A combination of both strand displacement and deoxyribozyme logic gates [59, 60] can address this issue, since an active RCDZ can be produced **and** used for **fueling** the cascades by cleaving RNA phosphodiester bonds.

4.4 Logic Gates Connected Via DNA Four-Way Junction (4WJ)

Arguably, any DNA-based sensor can be converted into a set of DNA logic gates. One of the most elegant nucleic acid sensors is the molecular beacon (MB) probe
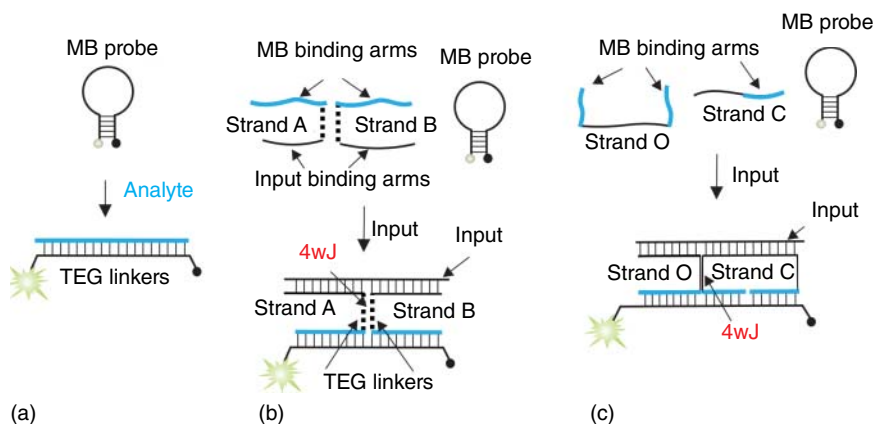
Figure 4.5 Design of molecular beacon (MB)-based DNA logic gates. (a) MB probe fluoresces upon bind complementary analyte. (b) The 4WJ-1 design for a YES gate. Oligoethylene glycol linkers triethylene glycol (TEG) are shown as dashed lines. Source: Based on Lake et al. [63]. (c) The 4JW-2 design of YES gate. Source: Based on Cornett et al. [64].

[61, 62], a fluorophore- and a quencher-conjugated DNA hairpin (Figure 4.5a). MB probes have been used in DNA analysis in biomedical practice [62, 65]. We designed a series of logic gates using an MB probe as a convenient fluorescent reporter [27, 63, 64, 66, 67]. Figure 4.5b,c demonstrates two designs for DNA YES gates. In both designs, DNA strands associate to form a structure stabilized through the formation of 4WJ, which gave the name to the DNA gates: "4WJ-1" and "4WJ-2." 4WJ-1 YES gate consists of two oligoethylene glycol-modified DNA strands A and B and an MB probe. These three oligonucleotides coexist in solution in a dissociated state when a nucleic acid analyte is absent; MB is in the stem-loop conformation, and the fluorescent signal is low. Addition of a DNA input leads to the cooperative hybridization of strands A and B both to the analyte and to the MB probe, which results in the formation of a 4WJ-containing complex (Figure 4.5b, bottom). The fluorophore and the quencher are remote from each other in this complex, which results in high fluorescence. Addition of oligoethylene glycol linkers to strands A and B was required to ensure the elongated conformation of the MB probe in the 4WJ complex [63]. YES gate in 4WJ-2 design responds to the input presence according to a similar scheme. However, strands O and C constituting the gate do not require oligoethylene glycol modifications, since the two MB-binding arms of strand C ensured the required elongated MB conformation in the input-bound complex [64].

   We have designed and optimized individual 4WJ logic gates in solution with the main purpose of integrating them into computational circuits [63, 64, 66, 67]. The designs of NOT and AND gates are presented in Figure 4.6a,b. NOT gate forms a fluorescent complex with an MB probe; addition of an oligonucleotide input disrupts the 4WJ structure, which results in the release of the MB probe from the NOT complex yielding low fluorescent signal. AND gate consists of three DNA strands folded in hairpin structures (Figure 4.6b). Two oligonucleotide inputs open their correspondent AND hairpins and trigger association of the two strands with the MB probe. Again, the complex is stabilized by the 4WJ
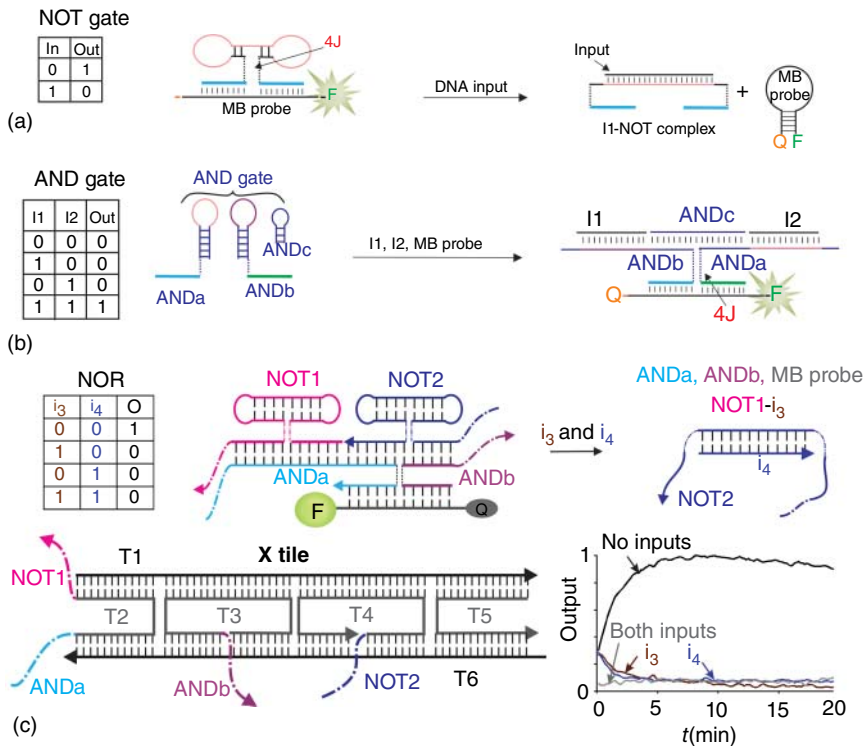
Figure 4.6 4WJ DNA logic gates and tile-integrated DNA circuits. (a) 4WJ NOT gate: a DNA strand (NOT) holds the opened MB probe in the absence of an input. Addition of the DNA input decomposes the complex, thus releasing the MB probe. (b) Two-input 4WJ AND gate. The gate consists of ANDa, ANDb, ANDc, and an MB probe. The five-stranded 4WJ association is formed only in the presence of both inputs I1 and I2. (c) Two-input 4WJ NOR gate integrated into a DNA tile. NOT1, NOT2, ANDa, and ANDb strands are attached to a DNA crossover (X) tile at the indicated points. In the absence of inputs, the 5′-terminal NOT1 and 3′-terminal NOT2 output fragments are bound to the input-recognition fragments of the ANDa and ANDb strands, respectively, which enables formation of the high-signal (fluorescent) 4WJ NOR association. Addition of inputs i3 and/or i4, which are complementary to the input-recognition fragments of NOT1 and NOT2, respectively, results in dissociation of the 4WJ NOR complex. The gate performs as predicted according to the observed changes in fluorescent output (lower right corner).

structure. The output of the gates is a DNA fragment formed by the portions of the two AND strands (e.g. blue and green in Figure 4.6b). Therefore, the 4WJ gate design preserves the input–output homogeneity, which is important for the integration of the gates into circuits, since an output of one 4WJ gate can be recognized as an input by the downstream gate. We achieved three layers of gate integration in solution by building an XOR logic from one OR, two AND, and two NOT gates [66]. Expectedly, the signal intensity would decrease upon adding each new level of integrated gates [66].

To facilitate inter-gate communication and reduce the undesirable crosstalk, we attempted to confine the gates in a nano-environment by attaching them to

a DNA scaffold in proper orientation and near their direct communication partners [27]. For example, Figure 4.6c demonstrates integration of two NOT with an AND gate to produce a NOR logic: the two NOT gates feed their outputs to a 2iAND. The NOT and AND strands form a fluorescent complex with an MB probe, while addition of at least one input decomposes one of the NOT gates followed by the collapse of the entire complex into the separate strands. The obtained circuits were shown to be reusable multiple times if RNA inputs are used in the presence of RNase H as a buffer component. As the case for strand displacement-based logic gates, a universal powering method is needed to build long chains of communicating logic gates.

4.5 Conclusion

Majority of the DNA logic gates, however, explore only two to five layers of integration, which faces significant signal reduction as the signal propagates along the chain of communicating gates. At least partially, this problem can be mitigated by localizing logic gates in a specific order and at precise positions on a DNA tile for efficient communication as it is used in electronic processors. An energy input is required to "push" the signal through the DNA association, an approach that has not been realized yet. Alternatively, parallel computation using multiple small-scale integrated circuits can be explored. While all the technical problems can be eventually addressed given the appropriate time and effort, the future of molecular computation depends on the practical usability of DNA computers. Indeed, it becomes clear that computers based on hybridization of DNA strands cannot compete with electronic devices in terms of the processing speed due to much slower rates of DNA hybridization than electron transfer in semiconductor materials. Instead, biocompatible and biodegradable DNA-based logic constructs can be used for manipulating biological molecules and objects (cells), which can eventually find applications in addressing biological and biomedical problems.

References

1 Malvino, A.P. and Brown, J.A. (1993). *Digital Computer Electronics*, 3e. Lake Forest: Glencoe.

2 de Silva, A.P. and Uchiyama, S. (2007). *Nat. Nanotechnol.* 2: 399–410.

3 de Silva, A.P., Leydet, Y., Lincheneau, C., and McClenaghan, N.D. (2006). *J. Phys. Condens. Matter* 18: S1847–S1872.

4 Ball, P. (2000). *Nature* 406: 118–120.

5 Katz, E. (2017). *Anal. Bioanal. Chem.* 409: 81–94.

6 Katz, E. (2015). *Curr. Opin. Biotechnol.* 34: 202–208.

7 Schneider, H.J. (2017). *ChemPhysChem* 18: 2306–2313.

8 Benenson, Y. (2016). *Chimia (Aarau)* 70: 392–394.

9 Erbas-Cakmak, S., Kolemen, S., Sedgwick, A.C. et al. (2018). *Chem. Soc. Rev.* 47: 2228–2248.

10 Adleman, L.M. (1994). *Science* 266: 1021–1024.

11 Stojanovic, M.N., Stefanovic, D., and Rudchenko, S. (2014). *Acc. Chem. Res.* 47: 1845–1852.

12 Fu, T., Lyu, Y., Liu, H. et al. (2018). *Trends Biochem. Sci.* 43: 547–560.

13 Ariga, K., Nishikawa, M., Mori, T. et al. (2019). *Sci. Technol. Adv. Mater.* 20: 51–95.

14 Stojanovic, M.N., Mitchell, T.E., and Stefanovic, D. (2002). *J. Am. Chem. Soc.* 124: 3555–3561.

15 Saghatelian, A., Völcker, N.H., Guckian, K.M. et al. (2003). *J. Am. Chem. Soc.* 125: 346–347.

16 Okamoto, A., Tanaka, K., and Saito, I. (2004). *J. Am. Chem. Soc.* 126: 9458–9463.

17 Yoshida, W. and Yokobayashi, Y. (2007). *Chem. Commun.* 14: 195–197.

18 Penchovsky, R. and Breaker, R.R. (2005). *Nat. Biotechnol.* 23: 1424–1433.

19 He, H.Z., Chan, D.S., Leung, C.H., and Ma, D.L. (2013). *Nucleic Acids Res.* 41: 4345–4359.

20 Kahan-Hanum, M., Douek, Y., Adar, R., and Shapiro, E. (2013). *Sci. Rep.* 3: 1535.

21 Li, T., Lohmann, F., and Famulok, M. (2014). *Nat. Commun.* 5: 4940.

22 Guo, Y., Zhou, L., Xu, L. et al. (2014). *Sci. Rep.* 4: 7315.

23 Fan, D., Wang, K., Zhu, J. et al. (2015). *Chem. Sci.* 6: 1973–1978.

24 Green, A.A., Kim, J., Ma, D. et al. (2017). *Nature* 548: 117–121.

25 Gao, J., Liu, Y., Lin, X. et al. (2017). *Sci. Rep.* 7: 14014.

26 He, K., Li, Y., Xiang, B. et al. (2015). *Chem. Sci.* 6: 3556–3564.

27 Gerasimova, Y.V. and Kolpashchikov, D.M. (2016). *Angew. Chem. Int. Ed.* 55: 10244–10247.

28 Harding, B., Pollak, N.M., Stefanovic, D., and Macdonald, J. (2019). *Nano Lett.* https://doi.org/10.1021/acs.nanolett.9b02326.

29 Tam, D.Y., Dai, Z., Chan, M.S. et al. (2016). *Angew. Chem. Int. Ed.* 55: 164–168.

30 O'Steen, M.R., Cornett, E.M., and Kolpashchikov, D.M. (2015). *Chem. Commun. (Camb)* 51: 1429–1431.

31 Mailloux, S., Gerasimova, Y.V., Guz, N. et al. (2015). *Angew. Chem. Int. Ed.* 54: 6562–6566.

32 Fedotova, T.A. and Kolpashchikov, D.M. (2017). *Chem. Commun. (Camb)* 53: 12622–12625.

33 Liu, H., Yu, X., Chen, Y. et al. (2017). *Nat. Commun.* 8: 2006.

34 Zhou, W., Ding, J., and Liu, J. (2017). *Theranostics* 7: 1010–1025.

35 Stojanović, M.N. and Stefanović, D. (2003). *J. Am. Chem. Soc.* 125: 6673–6676.

36 Lederman, H., Macdonald, J., Stefanovic, D., and Stojanovic, M.N. (2006). *Biochemistry* 45: 1194–1199.

37 Gerasimova, Y.V. and Kolpashchikov, D.M. (2015). *Chem. Commun. (Camb)* 51: 870–872.

38 Macdonald, J., Li, Y., Sutovic, M. et al. (2006). *Nano Lett.* 6: 2598–2603.

39 Stojanovic, M.N. and Stefanovic, D. (2003). *Nat. Biotechnol.* 21: 1069–1074.

40 Stojanovic, M.N., Semova, S., Kolpashchikov, D. et al. (2005). *J. Am. Chem. Soc.* 127: 6914–6915.

41 Yashin, R., Rudchenko, S., and Stojanovic, M.N. (2007). *J. Am. Chem. Soc.* 129: 15581–15584.

42 Han, C.M., Katilius, E., and Santiago, J.G. (2014). *Lab Chip* 14: 2958–2967.

43 Bone, S.M., Lima, N.E., and Todd, A.V. (2015). *Biosens. Bioelectron.* 70: 330–337.

44 Guz, N., Fedotova, T.A., Fratto, B.E. et al. (2016). *ChemPhysChem* 17: 2247–2255.

45 Ellwood, M.S., Collins, M., Fritsch, E.F. et al. (1986). *Clin. Chem.* 32: 1631–1636.

46 Srinivas, N., Ouldridge, T.E., Sulc, P. et al. (2013). *Nucleic Acids Res.* 41: 10641–10658.

47 Morrison, L.E., Halder, T.C., and Stols, L.M. (1989). *Anal. Biochem.* 183: 231–244.

48 Li, Q., Luan, G., Guo, Q., and Liang, J. (2002). *Nucleic Acids Res.* 30: E5.

49 Luk, K.C., Devare, S.G. Jr., and Hackett, J.R. (2007). *J. Virol. Methods* 144: 1–11.

50 Huang, Q., Zheng, L., Zhu, Y. et al. (2011). *PLoS One* 6: e16033.

51 Huang, S., Salituro, J., Tang, N. et al. (2007). *Nucleic Acids Res.* 35: e101.

52 Zhang, D.Y., Chen, S.X., and Yin, P. (2012). *Nat. Chem.* 4: 208–214.

53 Seelig, G., Soloveichik, D., Zhang, D.Y., and Winfree, E. (2006). *Science* 314: 1585–1558.

54 Thubagere, A.J., Thachuk, C., Berleant, J. et al. (2017). *Nat. Commun.* 8: 14373.

55 Cherry, K.M. and Qian, L. (2018). *Nature* 559: 370–376.

56 Qian, L. and Winfree, E.J.R. (2011). *J. R. Soc. Interface* 8: 1281–1297.

57 Qian, L. and Winfree, E. (2011). *Science* 332: 1196–1200.

58 Chatterjee, G., Dalchau, N., Muscat, R.A. et al. (2017). *Nat. Nanotechnol.* 12: 920.

59 Brown, C.W., Lakin, M.R., Stefanovic, D., and Graves, S.W. (2014). *ChemBioChem* 15: 950–954.

60 Yang, J., Wu, R., Li, Y. et al. (2018). *Nucleic Acids Res.* 46: 8532–8541.

61 Tyagi, S. and Kramer, F.R. (1996). *Nat. Biotechnol.* 14: 303–308.

62 Kolpashchikov, D.M. (2012). *Scientifica (Cairo)* 2012: 928783.

63 Lake, A., Shang, S., and Kolpashchikov, D.M. (2010). *Angew. Chem. Int. Ed.* 49: 4459–4462.

64 Cornett, E.M., Campbell, E.A., Gulenay, G. et al. (2012). *Angew. Chem. Int. Ed.* 51: 9075–9077.

65 Navarro, E., Serrano-Heras, G., Castaño, M.J., and Solera, J. (2015). *Clin. Chim. Acta* 439: 231–250.

66 Gerasimova, Y.V. and Kolpashchikov, D.M. (2012). *Chem. Asian J.* 7: 534–540.

67 Campbell, E.A., Peterson, E., and Kolpashchikov, D.M. (2017). *ChemPhysChem* 18: 1730–1734.

5

Development of Logic Gate Nanodevices from Fluorogenic RNA Aptamers

Trinity Jackson, Rachel Fitzgerald, Daniel K. Miller, and Emil F. Khisamutdinov

Ball State University, Department of Chemistry, 2000 W. University Ave., Muncie, IN 47306, USA

5.1 Nucleic Acid: The Material of Choice for Nanotechnology

Manipulation of matter at a nanometer scale is very challenging yet one of the central goals of the twenty-first century. Particles at nanometer scale can exhibit unique phenomena. For example, they emit variable ultraviolet and visible light frequencies [1, 2] and can exist in numerous intricate 3D forms [3]. These nanoparticles can be made from diverse spectra of materials including metal atoms (e.g. gold, silver, iron oxide nanoparticles) [4], lipids (e.g. micelles, liposome) [5], amino acids (e.g. antibodies) [6, 7], and nucleic acids (nanostructures made of DNA, RNA, or hybrid oligonucleotides) [8]. While all previously mentioned materials could be used to synthesize or assemble particles in a controlled and preprogrammed way, nucleic acids have some particular advantages.

DNA and RNA are biopolymers with four distinct types of monomeric units or nucleotides (nts) (AGCT and AGCU) where A pairs with T (U in RNA) and G pairs with C. However, the order at which these nts are positioned within the sequence dictates stability and folding of the overall nucleic acid conformation. These specific variations are particularly important for structural RNA applications. The folding of RNA into a secondary structure can be predicted with high level of accuracy with user-friendly and online available tools such as *mfold* [9] and *NUPACK* [10]. The folding algorithm is the same for RNA and DNA and utilizes empirically defined nearest-neighbor thermodynamic parameters for each base pair step [11, 12]. However, the RNA folding process often occurs through long-range intramolecular interactions, as RNA is single-stranded product in nature. For DNA, the folding process is dictated by intermolecular interaction. In addition, the computational prediction of DNA secondary structure is more accurate because only the G–C and A–T base pairs (Watson–Crick pairs) and the 10 unique base pair steps contribute to the stacking interaction of a double-stranded helix. In RNA molecules, this process is more sophisticated due to the potential formation not only canonical (Watson–Crick pairs) but also non-Watson–Crick base pairs as well as base triples and base quadrupoles as summarized by Leontis and Westhof [13–15].

The study of DNA nanoparticles has been one of significant interest since the early 1980s due to the progressive developmental foundation proposed by Nadrian Seeman [16, 17]. Since the conceptual layout of DNA nanotechnology was proposed, widespread interest toward DNA nanotechnology [18–20] and RNA nanotechnology [21–24] has occurred in the scientific community. The design of nucleic acid nanoparticles into well-defined two- or three-dimensional shapes can be accomplished by using the DNA origami technique [25, 26]. This approach utilizes designing multiple short DNA fragments ("staple" strands) that force folding of a long single-strand DNA (DNA template) into a preprogrammed shape. Computational tools are used to calculate the placement of individual staple strands within a specific region of the DNA template, and due to Watson–Crick base pairing, the necessary sequences of all staple strands can be executed. Figure 5.1 demonstrates an example of the DNA dolphin-shaped structure obtained by DNA origami [27]. Inspired by the DNA origami technique, researchers developed RNA origami. In this approach, RNA polymerase is implemented to transcribe a long RNA strand (RNA template) that can fold into a pre-rendered shape at isothermal conditions without a need for staple strands [29]. More often, RNA nanoparticles are designed using known crystal structures of complex RNA molecules such as ribosomal RNA containing multiple, well-defined, and often recurrent RNA structural motifs [30]. These motifs are then manually extracted from larger RNA complexes using 3D modeling software such as Swiss-PDBViewer [31]. RNA motifs serve as modular building blocks that can be further interconnected to obtain a desired shape [22, 28, 32, 33]. As a result, infinite numbers of nucleic acid nanoparticles with intricate shapes and dimensions can be modeled and assembled utilizing above techniques as exemplified in Figure 5.1. Researchers have used nucleic acids, both DNA and RNA, to fabricate artificial nucleic acid complexes for a variety of applications [34–40]. This has led to the development of therapeutic nucleic acid nanotechnology [41, 42], various devices for structure probing *in vitro* and *in vivo* [43, 44], and biomimetic systems [45], as well as development of nucleic acid "smart" devices capable of performing simple and complex molecular computations [43, 46].

5.2 RNA Aptamers are Modular and Programmable Biosensing Units

In parallel to the nucleic acid nanotechnology field, scientific attention has been drawn to nucleic acid aptamer development and their implementations. A nucleic acid aptamer consists of RNA and/or DNA oligonucleotides that bind to a specific target moiety (ligand) with high affinity and specificity. A wide range of various chemical and biological entities can serve as ligands ranging from a small molecule to a mammalian cell [47]. The development of aptamer oligonucleotides that bind to specific ligands was engineered through repeated rounds of *in vitro* selection also known as SELEX (systematic evolution of ligands by exponential enrichment) in the early 1990s in two independent groups of Larry Gold and Jack
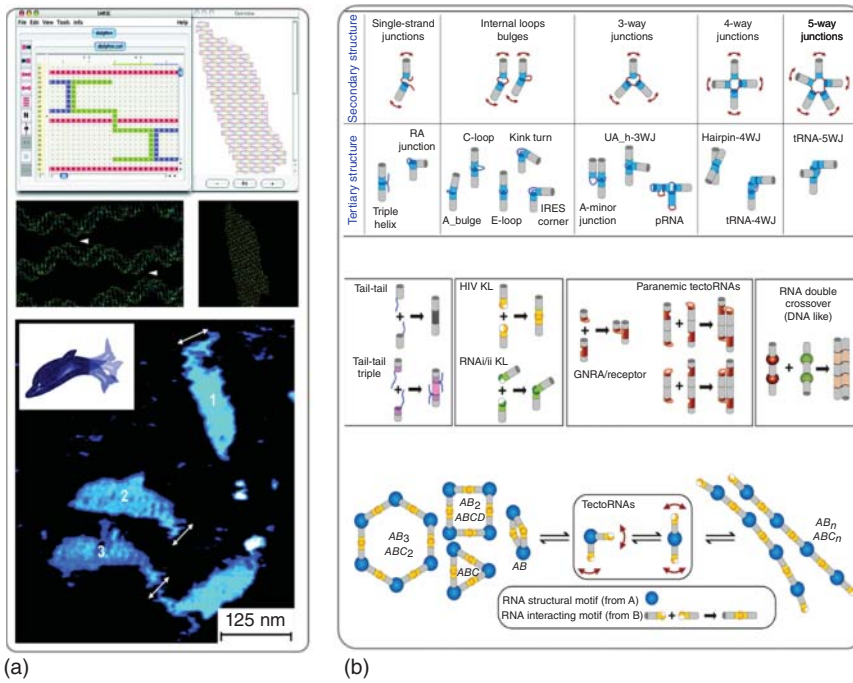
Figure 5.1 Nucleic acid nanostructure designing techniques. (a) DNA origami method was used to computationally design and assemble a dolphin-shaped DNA nanostructure. (b) Example of RNA nanoparticle design approach. Variety of RNA tertiary structures (tecto-RNAs) were combined to construct different nano-objects of 1D or linear shape, 2D or polygonal shapes, and 3D shapes. Source: (Panel a) From Andersen et al. [27]. Reprinted with the permission of American Chemical Society; (Panel b) From Grabow and Jaeger [28]. Reproduced with the permission of American Chemical Society.

Szostak [48, 49]. Since then, numerous different types of aptamers were engineered in academic research labs and found applications in various biotechnological fields (Figure 5.2a). Several recent reviews provide a critical evaluation of nucleic acid aptamer technology with their applications *in vitro* and *in vivo* [50–54].

In this chapter, we are primarily focusing on fluorogenic RNA aptamers, also known as light-up RNA aptamers. In particular, discussions will cover fluorogenic RNA aptamers as potential candidates to be used in biocomputing applications. Utilizing light-up aptamers in bioanalytical disciplines for an analyte sensing *in vivo* has several advantages. For instance, RNA aptamers can be genetically fused to genes coding for cellular RNAs of interest that in turn can be advantageous in endogenous synthesis of the targeting sequence [55–58]. Inserting light-up aptamers directly into the target endogenous RNA can be restricted by the labor-intensive genetics. However, these aptamers, even when used in multiple copies, do not significantly increase the size of the cellular RNA. Only the imaging dye needs to be introduced exogenously, and usually small nonpolar organic dyes can be passively diffused into cells. The major
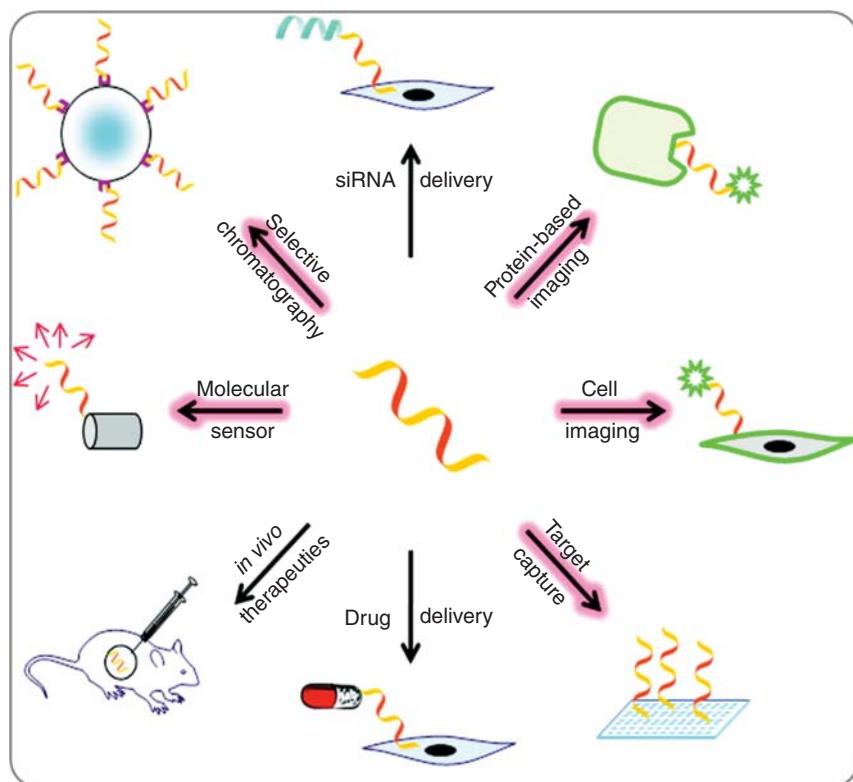
(a)
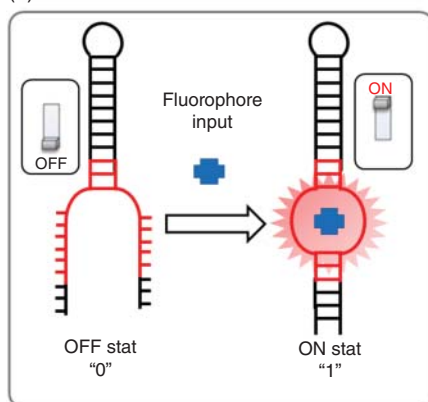
(b)
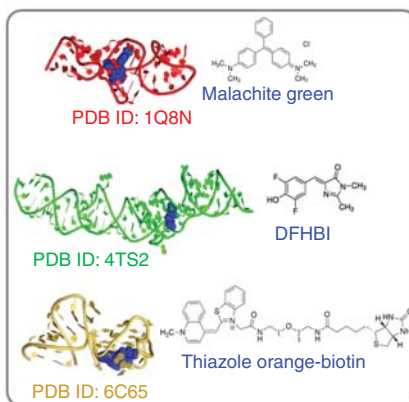
(c)

Figure 5.2 (a) Examples of the diverse application of nucleic acid aptamers. (b) Schematic 2D representation of fluorogenic RNA aptamer YES gated function. (c) Examples of known light-up RNA 3D structures of MG-binding aptamer (red), Spinach RNA aptamer (green), and Mango RNA aptamer (gold) with corresponding fluorophore ligands. Source: (Panel a) From Iliuk et al. [50]. Reproduced with the permission of American Chemical Society.
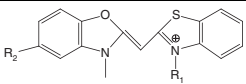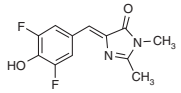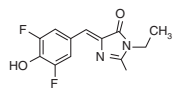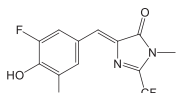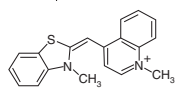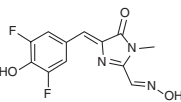
concerns of applying unmodified RNA aptamers *in vivo* are their susceptibility to degradation by nucleases, making it difficult to intracellularly express them. To tackle this problem, aptamer molecules are expressed by being inserted into a complex noncoding RNA strand, like tRNA, or into an RNA junction to protect the aptamer from degradation and induce folding. Such insertion of the RNA aptamer sequence into endogenous RNA not only allows transcription to be monitored *in vitro* but can also be used to regulate gene expression in a logical manner [59, 60].

In a fluorogenic RNA aptamer, the ligand is typically a small organic dye that can emit fluorescent light upon binding to its aptamer host molecule (otherwise, it is in a nonfluorescent state). This feature is very attractive to develop binary ON and OFF systems where a fluorescence state is ON and a nonfluorescence state is OFF as shown in Figure 5.2b. Over the past two decades, several RNA fluorogenic aptamers were developed, and all can be potentially implemented for fabrication of a nanocomplex with a simple YES logic operation with fluorescence outputs ON and OFF. The examples of such RNA aptamers and ligands as well as their properties are summarized in Table 5.1, and structures of the three most common RNA aptamers are illustrated in Figure 5.2c. The interaction between a ligand and a binding pocket of the host aptamer is usually non-covalent. Therefore, the interaction can be easily reversed by introducing denaturing agents or certain conditions that will disrupt the correct pocket conformation. Such disruption prevents ligand binding and thus serves as input signals. Once the ligands are in an unbound state (free in solution), their intrinsic fluorescence emission is diminished.

A highly effective RNA-based fluorogenic unit should possess specific features. The ideal dye needs to display a high absorption coefficient (ε) to ensure sensitive detection and to minimize fluorescence background. The fluorophore should show a low ratio of photons absorbed to photons emitted (quantum yield), meaning it should have a high fluorescence enhancement and brightness. The RNA–fluorophore interaction should be highly specific and occur with high affinity to make it possible to use low concentrations while still obtaining high contrast and keeping background fluorescence low. The aptamer–fluorophore complex also needs to be photostable to extend the ability for data acquisition. Often, these types of fluorogenic aptamers have superior characteristics over the electrochemical and colorimetric approach for sensing and imaging. For example, RNA strands do not require chemical conjugation, and the RNA aptamer provides high sensitivity and high speed of response while also exhibiting high spatial resolution [68, 69].
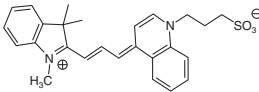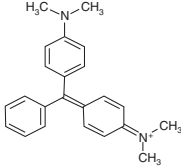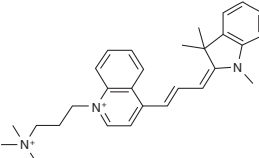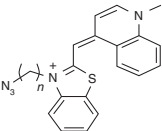
Malachite green (MG)-binding RNA aptamer was one of the earliest models of RNA light-up aptamer and was extensively studied in laboratory settings [38–40, 43, 46, 67, 70, 71]. Excitation of free triphenylmethane fluorophore (MG) in a solution results in low fluorescence due to easy vibrational de-excitation (i.e. excess energy from the MG excited state is dissipated in the form of vibrational movement). When MG is bound to its RNA aptamer, MG is stabilized in a planar form, and vibrations are restricted, which results in a 2000-fold increase of fluorescence [67]. In nucleic acid nanofabrication, fluorescent aptamers can potentially act as fluorescent reporter units that can be harbored into a larger

Table 5.1 List of some common RNA aptamers with corresponding fluorogenic ligands and properties.

Fluorogen dye	Light-up aptamer	K_D (nM)	Ex/Em (nm)	$\mathscr{E}$ (M^{-1}/cm)	Φ [a]	Fluorogen structure	PDB ID [b]	References
OTB	DiR2s-Apt	662	380/421	73 000	51		6DB9	[61]
DFHBI	Spinach	540	469/501	24 300	72		4TS0	[62]
DFHBI-1T	Spinach2	560	482/505	31 000	94		6B3K	[63]
DFHBI-2T	Spinach2	1300	500/523	29 000	12		6B3K	[63]
TO-1	Mango	3	510/535	77 500	14		5V3F	[64]
DFHO	Corn	70	505/545	29 000	25		6E80	[65]

(continued overleaf)

Table 5.1 (Continued)

Fluorogen dye	Light-up aptamer	K_D (nM)	Ex/Em (nm)	$\mathscr{E}$ (M^{-1}/cm)	Φ [a]	Fluorogen structure	PDB ID [b]	References
DIR	DIR apt	86	600/646	134 000	26		3T0W	[66]
Mal. Green	MG aptamer	117	630/650	150 000	19		1Q8N	[67]
DIR-pro	DIR2s-Apt	252	600/658	164 000	33		6D89	[61]
TO-3	Mango	6–8	637/658	9300	N/A		5V3F	[64]

a) Φ referred to quantum yield of the complex expressed in percentage.
b) Protein Data Bank ID number.

complex nanoparticle by simple extension of individual strands. Several previous reports have shown that MG-binding aptamers can be used for co-transcriptional assembly verification [43] as well as monitoring of the dynamic behavior of interdependent RNA–DNA hybrids [35].

More recently, a novel and much less intracellular toxic RNA aptamer, as compared with MG RNA aptamer, Spinach RNA aptamer, was developed [62]. The Spinach RNA aptamer binds the green fluorescent protein (GFP) fluorophore analog DFHBI ((Z)-4-(3,5-difluoro-4-hydroxybenzylidene)-1,2-dimethyl-1H-imidazol-5(4H)-one) [72]. The work on the Spinach aptamer has been extended to produce Spinach 2 that has much greater thermostability and brightness. However, Spinach 2 is more susceptible to degradation by nucleases [63, 73]. Further effort has been made to develop yet another "vegetable" aptamer called Broccoli that binds DFHBI-1T (derivative of the DFHBI dye) [74].

The interaction between the aptamer and its target often causes slight structural rearrangement in favor of stabilization of the RNA–ligand complex. This feature can be used to control RNA–ligand binding allosterically, where the allosteric site (sensing module) can be connected to the aptamer region (reporting module) through a communication module. This strategy was developed by Ronald Breaker using ribozymes as reporting modules [75, 76], and later a similar strategy was implemented using MG-binding RNA aptamer as a reporter unit [77, 78]. Allosteric biosensors can also be used for protein detection for specific applications [79]. With these biosensors, target metabolite molecules as well as enzymes participating in an intracellular pathway can be identified. For instance, RNA biosensors are now commonly used to sense the presence of the following metabolites: cyclic AMP [80], cyclic di-AMP [73], S-adenosylmethionine (SAM) [81], FMN [78], S-adenosyl-L-homocysteine (SAH) [82], and thiamine pyrophosphate (TPP) [83]. Also, the development of these aptamers has simplified RNA imaging in mammalian, yeast, and bacterial cells [44, 84–87].

5.3 Construction of RNA Nanoparticles with Integrated Logic Gate Operations Using Light-Up Aptamers

This section is focused exclusively on the fluorogenic RNA aptamer-based methods to fabricate responsive logic-gated nanodevices. Based on the aforementioned properties, RNA-based light-up aptamers are ideal for binary logic system development. These systems can be programmed to respond to an input signal, inducing conformational change within the RNA aptamer's binding pocket that will further dictate binding strength of a fluorescent dye molecule. Various factors can serve as inputs. Nonspecific factors include temperature, pH, and ionic strength. Specific factors can include predesigned competitive short oligonucleotide strands, nonfluorescent molecule mimicking structures of the ligand dye, and RNA-binding protein biomolecules, to name a few. The concentration of the inputs often needs to be fine-tuned to display desirable outcomes and define the threshold between ON and OFF values. Short DNA

oligonucleotides are commonly used as inputs because they are relatively inexpensive, are stable in aqueous solutions, and can hybridize with DNA as well as RNA strands to form RNA–DNA duplexes. When the inputs are present at a certain concentration, the overall fluorogenic aptamer structure can mimic a computer's function toggling between fluorescence (ON) and nonfluorescence (OFF) states. However, these can be achieved only when specific LOGICAL conditions are satisfied following Boolean algebra or function. The Boolean algebra is used to analyze and simplify the digital (logic) circuits and uses only the binary numbers 0 and 1. Other values include "YES and NO" or "ON and OFF," and this type of notation is often referred to as a binary algebra. Logic gate plays a role as an elementary building block of digital circuits. Depending on the complexity of the operations and tasks, some circuits may have only a few logic gates, while others, such as microprocessors, have combinatorial circuits embedding millions of logic gates.

5.3.1 Implementation of MG-Binding RNA Aptamer to Design Binary Logic Gates

The biochemical applications of logic gates include biosensing and data processing [88]. The simplest logic gates are based on one input and correspond with YES and NOT gates. All fluorogenic (light-up) RNA aptamers are macromolecules possessing YES logic functions since they turn ON only in the presence of input fluorophore molecules (Figure 5.2b). Examples of other common logic gates requiring two inputs implemented in binary algebra and their truth tables defining each function are summarized in Figure 5.3. The six most common binary logic gates include AND, OR, XOR, NAND, NOR, and XNOR. Among these gates, the OR logic gate yields 1 output when at least one of the inputs equals 1. The AND logic gate generates 1 output if and only if both inputs are 1. It is important to note that theoretically to form a complete set of all possible logic gates, only four basic YES, NOT, AND, and OR gates are required.

The RNA light-up aptamers are the perfect system to develop devices operating in a gated manner. However, currently there only a handful of reports assessing fluorogenic aptamers from this perspective [46, 89–92]. Recent work reported by Goldsworthy et al. [46] notably utilized MG-binding RNA aptamers to design complex systems performing AND, OR, NAND, and NOR logic gate functions. The design principle is illustrated in Figure 5.4a. In AND and OR specific examples, the MG-binding RNA aptamer sequence (shown in representative red color) was extended at each 5′- and 3′-ends to have 26 nts programmed to interfere with the structure of the MG-binding pocket. The resulting AND and OR gates with default setting 0-0 (no inputs present) produced 0 output. Two short DNA oligonucleotides served as inputs (A and B) that hybridize with the interfering ends (shown in Figure 5.4 in green and blue colors). The AND logic achieved output 1 (ON conformation) only when two inputs were present at the same time. The OR logic function is converted into the ON state when at least one of the inputs was present. Since the RNA sequences for the AND and OR operations were the same, inputs of varying length were chosen to selectively manipulate the response from the MG-binding RNA aptamer.

AND

Input		Output
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR

Input		Output
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

XOR

Input		Output
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

NAND

Input		Output
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

NOR

Input		Output
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

XNOR

Input		Output
A	B	
0	0	1
1	0	0
0	1	0
1	1	1
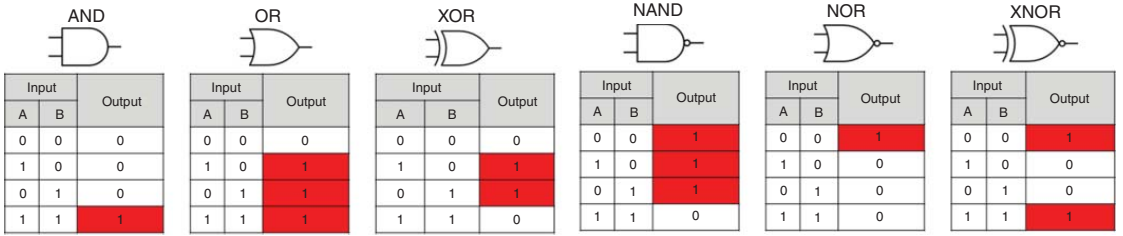
Figure 5.3 Common binary logic gate symbols and truth tables.
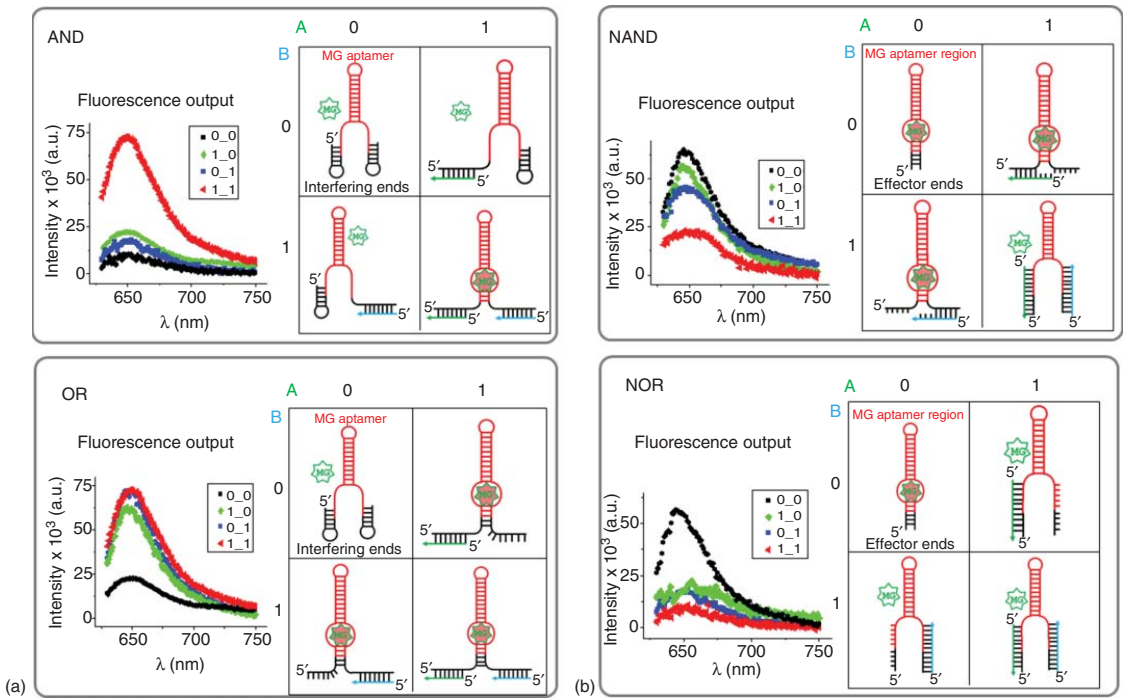
Figure 5.4 Logic gates design principle using MG-binding RNA aptamer. (a) Representative AND and OR gates showing fluorescence response and 2D RNA structures with no inputs, presence of one or both inputs. (b) Examples of NAND and NOR gates with fluorescence readouts and 2D structures of the RNA construct in response to presence of either input and both inputs. Source: (Panel b) Adapted from Goldsworthy et al. [46].

AND Boolean logic has been successfully applied by using split RNA aptamer systems [89–93]. The design does not utilize DNA as input to trigger conformational changes of the aptamer structure. Instead, the RNA aptamer itself is bisected into a two-component system. This allows the split aptamer to perform the AND logic function because two halves are required to bind the dye.

The NAND and NOR gates were designed slightly differently. The extensions at the 5′- and 3′-ends of the core MG-binding aptamer region were shorter (18 nt at the 5′-end and 17 nt at the 3′-end). These are non-interfering ends; thus they do not cause disruption of the MG-binding pocket. Therefore, in the absence of both inputs, 0-0, the output is 1 (Figure 5.4b). In NAND logic, the non-interfering ends must be able to bind inputs A or B without changing the structure of the MG-binding pocket. However, when both A and B are presented, the conformation of the aptamer needs to be sufficiently distorted to achieve OFF state. In NOR logic, the presence of either inputs significantly disrupts the conformation of the RNA molecule, rendering MG binding impossible. Hence, the output was "1" only in the absence of both potential DNA inputs.

Programs such as *NUPAC* and *mfold* are often used to design DNA–RNA aptamers with MG-binding potential. Unfortunately, these programs cannot analyze stability of hybrid RNA–DNA interactions. Since the overall design relies primarily on the strand displacement reaction between RNA–RNA and RNA–DNA interactions, the folding predictions might be inaccurate. To overcome this, various concentrations of RNA gates and DNA inputs should be explored to gain better ON and OFF separation threshold.

5.3.2 Implementation of MG-Binding RNA Aptamer and Broccoli RNA Aptamer to Design Half-Adder Circuit

Simplest level circuits like half-adders and full-adders include combinations of logic gate operations. The half-adder circuit can be constructed from combinations of logic functions AND and XOR with two inputs and two outputs: SUM (XOR gate) and CARRY (AND gate). The half-adder is used for adding together the two least significant digits in a binary sum (Figure 5.5). The four possible combinations of binary digits A and B are shown in Figure 5.5 (a) where the sum of the two digits is given for each of these combinations. For case A = 1 and B = 1, the sum is 10, where the 1 generated is the CARRY to the next stage of the addition.

An RNA tetragonal structure with infusion of two different light-up RNA aptamers acting as AND (MG-binding aptamer) and XOR (Broccoli aptamer) gates was fused into a tetragonal particle to fabricate the half-adder circuit. These gates utilize the same DNA inputs but produce two distinctly different fluorescence emissions as output signals, SUM ($\lambda_{em} = 510$ nm) and CARRY ($\lambda_{em} = 650$ nm). Both the MG RNA aptamer and Broccoli RNA aptamers were incorporated on alternating vertices of the tetragon. To satisfy the XOR and AND truth values (in absence of inputs 0-0 the output must be OFF), additional DNA inhibitor (XOR_DNA and AND_DNA inhibitors) strands were introduced to interfere the corresponding ON states of both aptamers (Figure 5.5c). The fluorescence readout corresponding to XOR and AND operations was achieved by relying on the displacement principle between inhibitor DNA strands, DNA
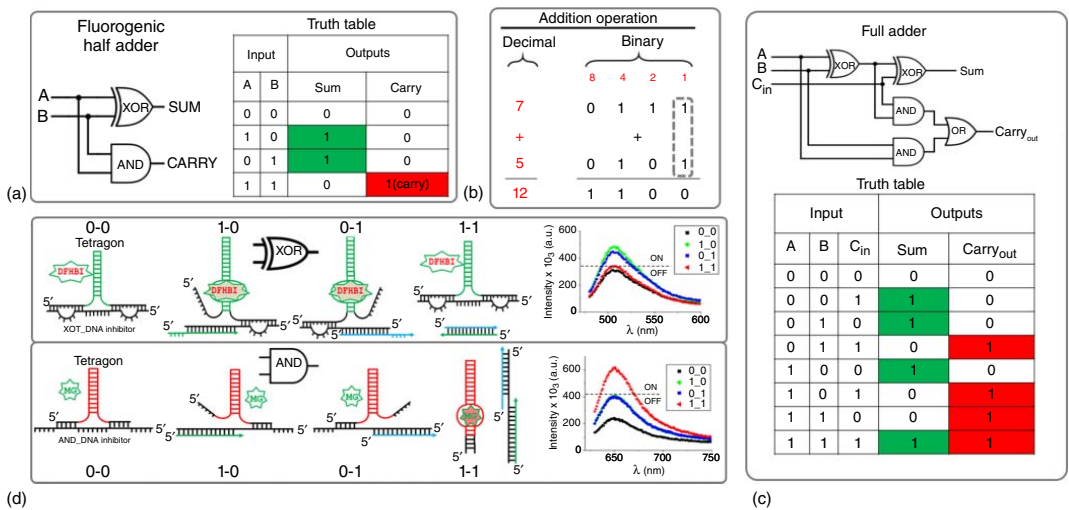
Figure 5.5 Examples of combinatorial logic gates using half-adder and full-adder. (a) Half-adder circuit diagram containing XOR and AND logic gates and its truth table. (b) Example of addition operation by the half-adder that adds the two least significant bits highlighted in puncture/dashed line. (c) Circuit schematic and truth table for full-adder. (d) Secondary structures demonstrating design principles for the Broccoli aptamer (XOR gate) and MG RNA aptamer (AND gate) and their fluorescence outputs in response to DNA inputs. Source: (Panel c) Adapted from Goldsworthy et al. [46]; (Panel d) From Goldsworthy et al. [46]. Licensed under CC by 4.0.

inputs, and RNA aptamers. Specific design approaches including RNA and DNA sequences chosen for combinatorial implementation of the AND and the XOR gates to satisfy the requirements for a half-adder have been published [46]. The next step is construction of a full-adder based on RNA light-up aptamer, which has yet to be achieved. The full-adder circuits extend the concept of the half-adder by providing an additional carry-in (Cin) input as demonstrated in the diagram in Figure 5.5c. This design has three inputs (A, B, and Cin) and two outputs (Sum and Carry-out).

5.4 Conclusion

The integration of advances in nucleic acid nanotechnology and in nucleic acid aptamer technologies makes it possible to build novel nanoparticles playing intermediate roles between electronic computers and biological systems. Programming with biological molecules, especially with nucleic acids (NA), is now becoming very attractive due to their potential of functions ranging from simple fluorescence emission to sophisticated gene regulation *in vivo*. The structural behavior encompassed within their sequences can be predicted and manipulated using 2D folding algorithms. The resulting nucleic acid biopolymers can then be used as logic-gated nano-agents for specific biomedical applications. Fluorogenic RNA aptamers can be designed to function as a simple circuit within individual binary logic gates. This demonstrates the great potential of nucleic acid nanotechnology and holds promise to develop cutting-edge technologies, especially if synergistically combined with other computing and nanorobotic systems.

Acknowledgments

References

1 Cinteza, L.O. (2010). Quantum dots in biomedical applications: advances and challenges. *J. Nanophotonics* 4 (Art. No.: 042503) https://doi.org/10.1117/1.3500388.

2 Wagner, A.M., Knipe, J.M., Orive, G., and Peppas, N.A. (2019). Quantum dots in biomedical applications. *Acta Biomater.* 94: 44–63. https://doi.org/10.1016/j.actbio.2019.05.022.

3 Jeevanandam, J., Barhoum, A., Chan, Y.S. et al. (2018). Review on nanoparticles and nanostructured materials: history, sources, toxicity and regulations. *Beilstein J. Nanotechnol.* 9: 1050–1074. https://doi.org/10.3762/bjnano.9.98.

4 Iriarte-Mesa, C., Lopez, Y.C., Matos-Peralta, Y. et al. (2020). Gold, silver and iron oxide nanoparticles: synthesis and bionanoconjugation strategies aimed at electrochemical applications. *Top. Curr. Chem. (Cham)* 378: 12. https://doi.org/10.1007/s41061-019-0275-y.

5 Mirahadi, M., Ghanbarzadeh, S., Ghorbani, M. et al. (2018). A review on the role of lipid-based nanoparticles in medical diagnosis and imaging. *Ther. Deliv.* 9: 557–569. https://doi.org/10.4155/tde-2018-0020.

6 Chiu, M.L., Goulet, D.R., Teplyakov, A., and Gilliland, G.L. (2019). Antibody structure and function: the basis for engineering therapeutics. *Antibodies (Basel)* 8 https://doi.org/10.3390/antib8040055.

7 Fernandez, L.A. and Muyldermans, S. (2011). Recent developments in engineering and delivery of protein and antibody therapeutics. *Curr. Opin. Biotechnol.* 22: 839–842. https://doi.org/10.1016/j.copbio.2011.08.001.

8 Michelotti, N., Johnson-Buck, A., Manzo, A.J., and Walter, N.G. (2012). Beyond DNA origami: the unfolding prospects of nucleic acid nanotechnology. *Wiley Interdiscip. Rev. Nanomed. Nanobiotechnol.* 4: 139–152. https://doi.org/10.1002/wnan.170.

9 Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* 31: 3406–3415. https://doi.org/10.1093/nar/gkg595.

10 Zadeh, J.N., Steenberg, C.D., Bois, J.S. et al. (2011). NUPACK: analysis and design of nucleic acid systems. *J. Comput. Chem.* 32: 170–173. https://doi.org/10.1002/jcc.21596.

11 Turner, D.H. (1996). Thermodynamics of base pairing. *Curr. Opin. Struct. Biol.* 6: 299–304. https://doi.org/10.1016/s0959-440x(96)80047-9.

12 Petersheim, M. and Turner, D.H. (1983). Base-stacking and base-pairing contributions to helix stability: thermodynamics of double-helix formation with CCGG, CCGGp, CCGGAp, ACCGGp, CCGGUp, and ACCGGUp. *Biochemistry* 22: 256–263. https://doi.org/10.1021/bi00271a004.

13 Leontis, N.B., Stombaugh, J., and Westhof, E. (2002). The non-Watson-Crick base pairs and their associated isostericity matrices. *Nucleic Acids Res.* 30: 3497–3531. https://doi.org/10.1093/nar/gkf481.

14 Sweeney, B.A., Roy, P., and Leontis, N.B. (2015). An introduction to recurrent nucleotide interactions in RNA. *Wiley Interdiscip. Rev.: RNA* 6: 17–45. https://doi.org/10.1002/wrna.1258.

15 Leontis, N.B. and Westhof, E. (2002). The annotation of RNA motifs. *Comp. Funct. Genomics* 3: 518–524. https://doi.org/10.1002/cfg.213.

16 Seeman, N.C. (1982). Nucleic acid junctions and lattices. *J. Theor. Biol.* 99: 237–247. https://doi.org/10.1016/0022-5193(82)90002-9.

17 Chen, J.H. and Seeman, N.C. (1991). Synthesis from DNA of a molecule with the connectivity of a cube. *Nature* 350: 631–633. https://doi.org/10.1038/350631a0.

18 Pinheiro, A.V., Han, D., Shih, W.M., and Yan, H. (2011). Challenges and opportunities for structural DNA nanotechnology. *Nat. Nanotechnol.* 6: 763–772. https://doi.org/10.1038/nnano.2011.187.

19 Chen, T., Ren, L., Liu, X. et al. (2018). DNA nanotechnology for cancer diagnosis and therapy. *Int. J. Mol. Sci.* 19 https://doi.org/10.3390/ijms19061671.

20 Cox, A.J., Bengtson, H.N., Rohde, K.H., and Kolpashchikov, D.M. (2016). DNA nanotechnology for nucleic acid analysis: multifunctional molecular DNA machine for RNA detection. *Chem. Commun. (Camb)* 52: 14318–14321. https://doi.org/10.1039/c6cc06889h.

21 Afonin, K.A., Schultz, D., Jaeger, L. et al. (2015). Silver nanoclusters for RNA nanotechnology: steps towards visualization and tracking of RNA nanoparticle assemblies. *Methods Mol. Biol.* 1297: 59–66. https://doi.org/10.1007/978-1-4939-2562-9_4.

22 Bui, M.N., Brittany Johnson, M., Viard, M. et al. (2017). Versatile RNA tetra-U helix linking motif as a toolkit for nucleic acid nanotechnology. *Nanomedicine* 13: 1137–1146. https://doi.org/10.1016/j.nano.2016.12.018.

23 Guo, P. (2010). The emerging field of RNA nanotechnology. *Nat. Nanotechnol.* 5: 833–842. https://doi.org/10.1038/nnano.2010.231.

24 Shukla, G.C., Haque, F., Tor, Y. et al. (2011). A boost for the emerging field of RNA nanotechnology. *ACS Nano* 5: 3405–3418. https://doi.org/10.1021/nn200989r.

25 Rothemund, P.W. (2006). Folding DNA to create nanoscale shapes and patterns. *Nature* 440: 297–302. https://doi.org/10.1038/nature04586.

26 Woo, S. and Rothemund, P.W. (2014). Self-assembly of two-dimensional DNA origami lattices using cation-controlled surface diffusion. *Nat. Commun.* 5: 4889. https://doi.org/10.1038/ncomms5889.

27 Andersen, E.S., Dong, M., Nielsen, M.M. et al. (2008). DNA origami design of dolphin-shaped structures with flexible tails. *ACS Nano* 2: 1213–1218. https://doi.org/10.1021/nn800215j.

28 Grabow, W.W. and Jaeger, L. (2014). RNA self-assembly and RNA nanotechnology. *Acc. Chem. Res.* 47: 1871–1880. https://doi.org/10.1021/ar500076k.

29 Sparvath, S.L., Geary, C.W., and Andersen, E.S. (2017). Computer-aided design of RNA origami structures. *Methods Mol. Biol.* 1500: 51–80. https://doi.org/10.1007/978-1-4939-6454-3_5.

30 Parlea, L.G., Sweeney, B.A., Hosseini-Asanjan, M. et al. (2016). The RNA 3D Motif Atlas: computational methods for extraction, organization and evaluation of RNA motifs. *Methods* 103: 99–119. https://doi.org/10.1016/j.ymeth.2016.04.025.

31 Kaplan, W. and Littlejohn, T.G. (2001). Swiss-PDB viewer (deep view). *Briefings Bioinf.* 2: 195–197. https://doi.org/10.1093/bib/2.2.195.

32 Jasinski, D., Haque, F., Binzel, D.W., and Guo, P. (2017). Advancement of the emerging field of RNA nanotechnology. *ACS Nano* 11: 1142–1164. https://doi.org/10.1021/acsnano.6b05737.

33 Shu, Y., Haque, F., Shu, D. et al. (2013). Fabrication of 14 different RNA nanoparticles for specific tumor targeting without accumulation in normal organs. *RNA* 19: 767–777. https://doi.org/10.1261/rna.037002.112.

34 Afonin, K.A., Bindewald, E., Kireeva, M., and Shapiro, B.A. (2015). Computational and experimental studies of reassociating RNA/DNA hybrids containing split functionalities. *Methods Enzymol.* 553: 313–334. https://doi.org/10.1016/bs.mie.2014.10.058.

35 Halman, J.R., Satterwhite, E., Roark, B. et al. (2017). Functionally-interdependent shape-switching nanoparticles with controllable properties. *Nucleic Acids Res.* 45: 2210–2220. https://doi.org/10.1093/nar/gkx008.

36 Hong, E., Halman, J.R., Shah, A.B. et al. (2018). Structure and composition define immunorecognition of nucleic acid nanoparticles. *Nano Lett.* 18: 4309–4321. https://doi.org/10.1021/acs.nanolett.8b01283.

37 Parlea, L., Bindewald, E., Sharan, R. et al. (2016). Ring catalog: a resource for designing self-assembling RNA nanostructures. *Methods* 103: 128–137. https://doi.org/10.1016/j.ymeth.2016.04.016.

38 Jasinski, D.L., Khisamutdinov, E.F., Lyubchenko, Y.L., and Guo, P. (2014). Physicochemically tunable polyfunctionalized RNA square architecture with fluorogenic and ribozymatic properties. *ACS Nano* 8: 7620–7629. https://doi.org/10.1021/nn502160s.

39 Khisamutdinov, E.F., Jasinski, D.L., and Guo, P. (2014). RNA as a boiling-resistant anionic polymer material to build robust structures with defined shape and stoichiometry. *ACS Nano* 8: 4771–4781. https://doi.org/10.1021/nn5006254.

40 Khisamutdinov, E.F., Jasinski, D.L., Li, H. et al. (2016). Fabrication of RNA 3D nanoprisms for loading and protection of small RNAs and model drugs. *Adv. Mater.* 28: 10079–10087. https://doi.org/10.1002/adma.201603180.

41 Guo, P., Haque, F., Hallahan, B. et al. (2012). Uniqueness, advantages, challenges, solutions, and perspectives in therapeutics applying RNA nanotechnology. *Nucleic Acid Ther.* 22: 226–245. https://doi.org/10.1089/nat.2012.0350.

42 Shu, D., Shu, Y., Haque, F. et al. (2011). Thermodynamically stable RNA three-way junction for constructing multifunctional nanoparticles for delivery of therapeutics. *Nat. Nanotechnol.* 6: 658–667. https://doi.org/10.1038/nnano.2011.105.

43 Shu, D., Khisamutdinov, E.F., Zhang, L., and Guo, P.X. (2014). Programmable folding of fusion RNA in vivo and in vitro driven by pRNA 3WJ motif of phi29 DNA packaging motor. *Nucleic Acids Res.* 42 (Art. No.: e10) https://doi.org/10.1093/nar/gkt885.

44 Guet, D., Burns, L.T., Maji, S. et al. (2015). Combining Spinach-tagged RNA and gene localization to image gene expression in live yeast. *Nat. Commun.* 6: 8882. https://doi.org/10.1038/ncomms9882.

45 Shen, H., Wang, Y., Wang, J. et al. (2019). Emerging biomimetic applications of DNA nanotechnology. *ACS Appl. Mater. Interfaces* 11: 13859–13873. https://doi.org/10.1021/acsami.8b06175.

46 Goldsworthy, V., LaForce, G., Abels, S., and Khisamutdinov, E.E. (2018). Fluorogenic RNA aptamers: a nano-platform for fabrication of simple and combinatorial logic gates. *Nanomaterials (Basel)* 8 (Art. No.: 984) https://doi.org/10.3390/nano8120984.

47 Kang, K.N. and Lee, Y.S. (2013). RNA aptamers: a review of recent trends and applications. *Adv. Biochem. Eng./Biotechnol.* 131: 153–169. https://doi.org/10.1007/10_2012_136.

48 Tuerk, C. and Gold, L. (1990). Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science* 249: 505–510. https://doi.org/10.1126/science.2200121.

49 Ellington, A.D. and Szostak, J.W. (1990). In vitro selection of RNA molecules that bind specific ligands. *Nature* 346: 818–822. https://doi.org/10.1038/346818a0.

50 Iliuk, A.B., Hu, L., and Tao, W.A. (2011). Aptamer in bioanalytical applications. *Anal. Chem.* 83: 4440–4452. https://doi.org/10.1021/ac201057w.

51 Panigaj, M., Johnson, M.B., Ke, W. et al. (2019). Aptamers as modular components of therapeutic nucleic acid nanotechnology. *ACS Nano* 13: 12301–12321. https://doi.org/10.1021/acsnano.9b06522.

52 Goud, K.Y., Reddy, K.K., Satyanarayana, M. et al. (2019). A review on recent developments in optical and electrochemical aptamer-based assays for mycotoxins using advanced nanomaterials. *Mikrochim. Acta* 187: 29. https://doi.org/10.1007/s00604-019-4034-0.

53 Li, F., Yu, Z., Han, X., and Lai, R.Y. (2019). Electrochemical aptamer-based sensors for food and water analysis: a review. *Anal. Chim. Acta* 1051: 1–23. https://doi.org/10.1016/j.aca.2018.10.058.

54 Pehlivan, Z.S., Torabfam, M., Kurt, H. et al. (2019). Aptamer and nanomaterial based FRET biosensors: a review on recent advances (2014–2019). *Mikrochim. Acta* 186: 563. https://doi.org/10.1007/s00604-019-3659-3.

55 Seelig, G., Soloveichik, D., Zhang, D.Y., and Winfree, E. (2006). Enzyme-free nucleic acid logic circuits. *Science* 314: 1585–1588. https://doi.org/10.1126/science.1132493.

56 Bao, G., Rhee, W.J., and Tsourkas, A. (2009). Fluorescent probes for live-cell RNA detection. *Annu. Rev. Biomed. Eng.* 11: 25–47. https://doi.org/10.1146/annurev-bioeng-061008-124920.

57 Benenson, Y., Gil, B., Ben-Dor, U. et al. (2004). An autonomous molecular computer for logical control of gene expression. *Nature* 429: 423–429. https://doi.org/10.1038/nature02551.

58 Zhang, X., Potty, A.S., Jackson, G.W. et al. (2009). Engineered 5S ribosomal RNAs displaying aptamers recognizing vascular endothelial growth factor and malachite green. *J. Mol. Recognit.* 22: 154–161. https://doi.org/10.1002/jmr.917.

59 Masuda, I., Igarashi, T., Sakaguchi, R. et al. (2017). A genetically encoded fluorescent tRNA is active in live-cell protein synthesis. *Nucleic Acids Res.* 45: 4081–4093. https://doi.org/10.1093/nar/gkw1229.

60 Culler, S.J., Hoff, K.G., and Smolke, C.D. (2010). Reprogramming cellular behavior with RNA controllers responsive to endogenous proteins. *Science* 330: 1251–1255. https://doi.org/10.1126/science.1192128.

61 Tan, X., Constantin, T.P., Sloane, K.L. et al. (2017). Fluoromodules consisting of a promiscuous RNA aptamer and red or blue fluorogenic cyanine dyes: selection, characterization, and bioimaging. *J. Am. Chem. Soc.* 139: 9001–9009. https://doi.org/10.1021/jacs.7b04211.

62 Paige, J.S., Wu, K.Y., and Jaffrey, S.R. (2011). RNA mimics of green fluorescent protein. *Science* 333: 642–646. https://doi.org/10.1126/science.1207339.

63 Song, W., Strack, R.L., Svensen, N., and Jaffrey, S.R. (2014). Plug-and-play fluorophores extend the spectral properties of Spinach. *J. Am. Chem. Soc.* 136: 1198–1201. https://doi.org/10.1021/ja410819x.

64 Dolgosheina, E.V., Jeng, S.C., Panchapakesan, S.S. et al. (2014). RNA mango aptamer-fluorophore: a bright, high-affinity complex for RNA labeling and tracking. *ACS Chem. Biol.* 9: 2412–2420. https://doi.org/10.1021/cb500499x.

65 Song, W., Filonov, G.S., Kim, H. et al. (2017). Imaging RNA polymerase III transcription using a photostable RNA-fluorophore complex. *Nat. Chem. Biol.* 13: 1187–1194. https://doi.org/10.1038/nchembio.2477.

66 Constantin, T.P., Silva, G.L., Robertson, K.L. et al. (2008). Synthesis of new fluorogenic cyanine dyes and incorporation into RNA fluoromodules. *Org. Lett.* 10: 1561–1564. https://doi.org/10.1021/ol702920e.

67 Babendure, J.R., Adams, S.R., and Tsien, R.Y. (2003). Aptamers switch on fluorescence of triphenylmethane dyes. *J. Am. Chem. Soc.* 125: 14716–14717. https://doi.org/10.1021/ja037994o.

68 Bouhedda, F., Autour, A., and Ryckelynck, M. (2017). Light-up RNA aptamers and their cognate fluorogens: from their development to their applications. *Int. J. Mol. Sci.* 19 https://doi.org/10.3390/ijms19010044.

69 Ouellet, J. (2016). RNA fluorescence with light-up aptamers. *Front. Chem.* 4: 29. https://doi.org/10.3389/fchem.2016.00029.

70 Grate, D. and Wilson, C. (1999). Laser-mediated, site-specific inactivation of RNA transcripts. *Proc. Natl. Acad. Sci. U.S.A.* 96: 6131–6136. https://doi.org/10.1073/pnas.96.11.6131.

71 Khisamutdinov, E.F., Li, H., Jasinski, D.L. et al. (2014). Enhancing immunomodulation on innate immunity by shape transition among RNA triangle, square and pentagon nanovehicles. *Nucleic Acids Res.* 42: 9996–10004. https://doi.org/10.1093/nar/gku516.

72 Warner, K.D., Chen, M.C., Song, W. et al. (2014). Structural basis for activity of highly efficient RNA mimics of green fluorescent protein. *Nat. Struct. Mol. Biol.* 21: 658–663. https://doi.org/10.1038/nsmb.2865.

73 Kellenberger, C.A., Chen, C., Whiteley, A.T. et al. (2015). RNA-based fluorescent biosensors for live cell imaging of second messenger cyclic di-AMP. *J. Am. Chem. Soc.* 137: 6432–6435. https://doi.org/10.1021/jacs.5b00275.

74 Filonov, G.S., Moon, J.D., Svensen, N., and Jaffrey, S.R. (2014). Broccoli: rapid selection of an RNA mimic of green fluorescent protein by fluorescence-based selection and directed evolution. *J. Am. Chem. Soc.* 136: 16299–16308. https://doi.org/10.1021/ja508478x.

75 Breaker, R.R. (2002). Engineered allosteric ribozymes as biosensor components. *Curr. Opin. Biotechnol.* 13: 31–39. https://doi.org/10.1016/s0958-1669(02)00281-1.

76 Zivarts, M., Liu, Y., and Breaker, R.R. (2005). Engineered allosteric ribozymes that respond to specific divalent metal ions. *Nucleic Acids Res.* 33: 622–631. https://doi.org/10.1093/nar/gki182.

77 Kolpashchikov, D.M. (2005). Binary malachite green aptamer for fluorescent detection of nucleic acids. *J. Am. Chem. Soc.* 127: 12442–12443. https://doi.org/10.1021/ja0529788.

78 Stojanovic, M.N. and Kolpashchikov, D.M. (2004). Modular aptameric sensors. *J. Am. Chem. Soc.* 126: 9266–9270. https://doi.org/10.1021/ja032013t.

79 Song, W., Strack, R.L., and Jaffrey, S.R. (2013). Imaging bacterial protein expression using genetically encoded RNA sensors. *Nat. Methods* 10: 873–875. https://doi.org/10.1038/nmeth.2568.

80 Sharma, S., Zaveri, A., Visweswariah, S.S., and Krishnan, Y. (2014). A fluorescent nucleic acid nanodevice quantitatively images elevated cyclic adenosine

monophosphate in membrane-bound compartments. *Small* 10: 4276–4280. https://doi.org/10.1002/smll.201400833.

81 Paige, J.S., Nguyen-Duc, T., Song, W., and Jaffrey, S.R. (2012). Fluorescence imaging of cellular metabolites with RNA. *Science* 335: 1194. https://doi.org/10.1126/science.1218298.

82 Su, Y., Hickey, S.F., Keyser, S.G., and Hammond, M.C. (2016). In vitro and in vivo enzyme activity screening via RNA-based fluorescent biosensors for *S*-adenosyl-ʟ-homocysteine (SAH). *J. Am. Chem. Soc.* 138: 7040–7047. https://doi.org/10.1021/jacs.6b01621.

83 You, M., Litke, J.L., and Jaffrey, S.R. (2015). Imaging metabolite dynamics in living cells using a Spinach-based riboswitch. *Proc. Natl. Acad. Sci. U.S.A.* 112: E2756–E2765. https://doi.org/10.1073/pnas.1504354112.

84 Ilgu, M., Ray, J., Bendickson, L. et al. (2016). Light-up and FRET aptamer reporters; evaluating their applications for imaging transcription in eukaryotic cells. *Methods* 98: 26–33. https://doi.org/10.1016/j.ymeth.2015.12.009.

85 Saurabh, S., Perez, A.M., Comerci, C.J. et al. (2016). Super-resolution imaging of live bacteria cells using a genetically directed, highly photostable fluoro-module. *J. Am. Chem. Soc.* 138: 10398–10401. https://doi.org/10.1021/jacs.6b05943.

86 Pothoulakis, G., Ceroni, F., Reeve, B., and Ellis, T. (2014). The spinach RNA aptamer as a characterization tool for synthetic biology. *ACS Synth. Biol.* 3: 182–187. https://doi.org/10.1021/sb400089c.

87 Strack, R.L., Disney, M.D., and Jaffrey, S.R. (2013). A superfolding Spinach2 reveals the dynamic nature of trinucleotide repeat-containing RNA. *Nat. Methods* 10: 1219–1224. https://doi.org/10.1038/nmeth.2701.

88 Tregubov, A.A., Nikitin, P.I., and Nikitin, M.P. (2018). Advanced smart nanomaterials with integrated logic-gating and biocomputing: dawn of theranostic nanorobots. *Chem. Rev.* 118: 10294–10348. https://doi.org/10.1021/acs.chemrev.8b00198.

89 Chandler, M., Lyalina, T., Halman, J. et al. (2018). Broccoli fluorets: split aptamers as a user-friendly fluorescent toolkit for dynamic RNA nanotechnology. *Molecules* 23 https://doi.org/10.3390/molecules23123178.

90 Kikuchi, N. and Kolpashchikov, D.M. (2016). Split spinach aptamer for highly selective recognition of DNA and RNA at ambient temperatures. *ChemBioChem* 17: 1589–1592. https://doi.org/10.1002/cbic.201600323.

91 Kikuchi, N. and Kolpashchikov, D.M. (2017). A universal split spinach aptamer (USSA) for nucleic acid analysis and DNA computation. *Chem. Commun. (Camb)* 53: 4977–4980. https://doi.org/10.1039/c7cc01540b.

92 Rogers, T.A., Andrews, G.E., Jaeger, L., and Grabow, W.W. (2015). Fluorescent monitoring of RNA assembly and processing using the split-spinach aptamer. *ACS Synth. Biol.* 4: 162–166. https://doi.org/10.1021/sb5000725.

93 Alam, K.K., Tawiah, K.D., Lichte, M.F. et al. (2017). A fluorescent split aptamer for visualizing RNA-RNA assembly in vivo. *ACS Synth. Biol.* 6: 1710–1721. https://doi.org/10.1021/acssynbio.7b00059.

6

Programming Molecular Circuitry and Intracellular Computing with Framework Nucleic Acids

Jiang Li[1] and Chunhai Fan[2]

[1] *Shanghai Synchrotron Radiation Facility, Zhangjiang Laboratory, Shanghai Advanced Research Institute, Chinese Academy of Sciences, 99 Haike Rd., Shanghai 201204, China*
[2] *Institute of Molecular Medicine, Renji Hospital, Shanghai Jiao Tong University, School of Chemistry and Chemical Engineering, Frontiers Science Center for Transformative Molecules, School of Medicine, 800 Dongchuan Rd., Shanghai 200240, China*

6.1 Framework Nucleic Acids

The intracellular environment is highly compartmentalized; biomolecules (nucleic acids, proteins, lipids, etc.) are spatiotemporally organized to form functional modules, signal pathways, and interacting networks. Inspired by nature, many researchers are interested in designing and/or repurposing artificial biomolecular assemblies to engineer living organisms with the goals of elucidating their molecular mechanisms and constructing novel biological computers or robots. These studies will have broad impacts in various areas ranging from fundamental biological studies to applications including synthetic biology, theranostics, and biocomputing [1].

Among the biomolecules, the double helical structure model of DNA forms the basis of genetics ranging from microorganisms and plants to animals. Besides this best-known structure, nucleic acids have shown great versatility in nature. For example, single- or double-stranded riboswitches [2], ribozymes [3], circular RNA [4], and more complex four-stranded G-quadruplex [5] and i-motif [6] have been found to play important regulatory roles in living cells. In synthetic world, nucleic acid nanotechnology pioneered by Ned Seeman quickly evolved into a booming field [7, 8]. By exploiting the unparalleled precision and programmability of nucleic acid hybridization, researchers now have been able to construct virtually any prescribed nucleic acid nanostructure in a bottom-up manner. So far, there have been two main pathways of building nucleic acid nanostructures. One pathway is using single-stranded nucleic acids or multistranded nucleic acid assemblies as building blocks (or so-called tiles) to assemble a higher-ordered structure (tile-based DNA nanostructure; Figure 6.1). The other pathway is to fold a long single-stranded nucleic acid (scaffold) into a compact structure so-called DNA/RNA origami (Figure 6.2). Especially, the development of shell or skeleton DNA/RNA frameworks, or more recently dubbed "framework nucleic acids" (FNAs), sheds new light
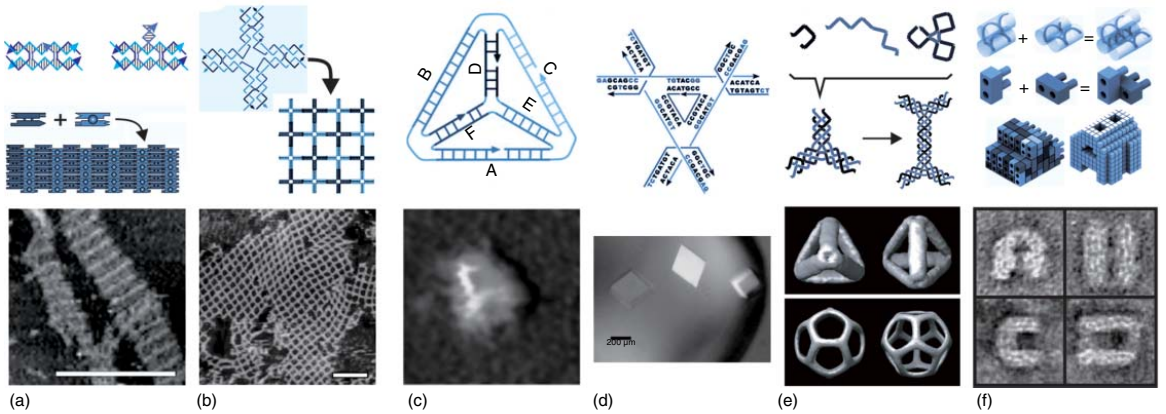
Figure 6.1 Representative tile-based DNA nanostructures. (a) 2D DNA crystalline arrays self-assembled from synthetic DNA double-crossover tiles [9]. (b) 2D square lattice assembled with 4 × 4 DNA tiles [10]. (c) DNA tetrahedral structure [11]. (d) Self-assembled 3D DNA crystal from a tensegrity DNA triangle motif [12]. (e) Hierarchical polyhedral DNA structures [13]. (f) 3D structures built with single-stranded tile DNA bricks [14].

DNA origami nanostructures
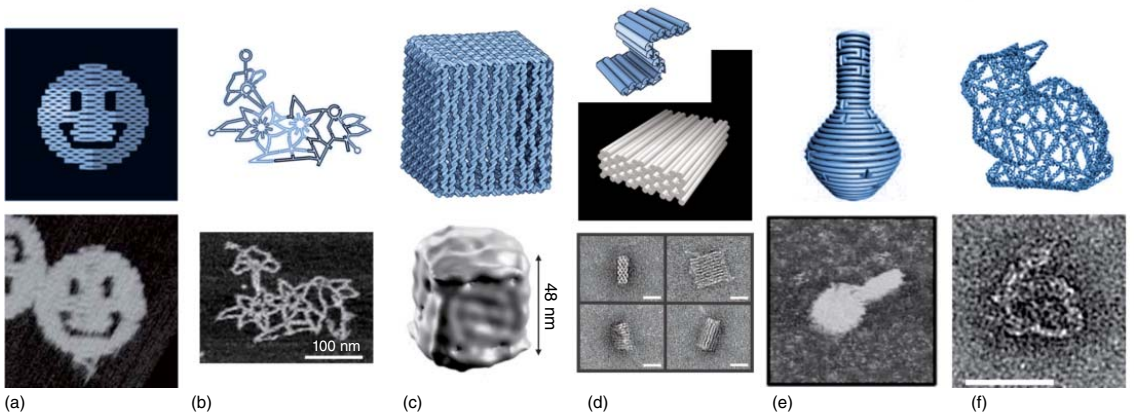
Figure 6.2 Representative DNA origami nanostructures. (a) DNA origami folded by a long single-stranded scaffold DNA and hundreds of short staple DNA oligos [15]. (b) Wireframe DNA origami nanostructures with multi-arm junction vertices [16]. (c) Hollow DNA box with a controllable lid [17]. (d) 3D DNA origami built with multiple pleated layers [18]. (e) 3D DNA origami with complex curvatures [19]. (f) Arbitrary 3D structure built with polyhedral meshes [20].

on spatial organization of molecules and materials *in vitro* and *in vivo* [21], which enables spatiotemporal control of biomolecular interactions resembling their intracellular counterparts. Despite the remarkable advances for studies in test tubes, the efficient and tailored fabrication and manipulation of circuit modules in cells and animals remains enormously challenging due to the extreme complexity of natural metabolic networks [22, 23].

In this chapter, we review the recent progress in this highly interdisciplinary field, with a focus on constructing molecular circuitry for synthetic biology by exploiting FNAs.

6.2 A Toolbox for Biomolecular Engineering of Living Systems

The precision of nucleic acid hybridization ensures the fidelity of DNA replication during the transmission of genetic information. DNA nanotechnology exploits the precise and predictable Watson–Crick base pairing rules to construct a variety of self-assembled DNA nanostructures ranging from one-dimensional (1D) to three-dimensional (3D), from periodic to discrete, and from static to dynamic architectures [7].

Recent advances in the field have revealed the unprecedented power in using engineered nucleic acid nanostructures for various applications, especially biomolecular engineering both *in vitro* and *in vivo*. To demonstrate the potential for constructing modularly designed synthetic circuits in living organisms, we summarize DNA/RNA nanotechnology-enabled tools and related emerging techniques for on-demand nucleic acid manipulation (Figure 6.3).

6.2.1 Biomolecular Scaffolds

Nature has harnessed this principle to evolve structural proteins to support and maintain physiological functions of cells, as exemplified by cytoskeleton proteins that form scaffolds to maintain the cell shape. In contrast, the scaffolding functions of the other two types of biomolecules involved in the central dogma, DNA and RNA, are less explored in biology, which have nevertheless been the focus of study in structural DNA nanotechnology. Self-assembled DNA nanostructures are fully addressable and can accommodate precise numbers of small molecules, DNA/RNA, proteins [32, 33], lipids [34], and even nanoparticles at nearly any prescribed position [24]. Especially, the development of well-defined FNAs sheds new light on spatial organization of molecules and materials *in vitro* and *in vivo*. This power is complemented by the commercial availability of a wide range of nucleotide modifications that offer great flexibility in choosing robust and nearly quantitative conjugation chemistry. Researchers have taken advantage of these attractive features to design various FNAs to make functional modules such as enzyme cascades [24, 35–37], photonic coupling [38], or electronic wiring [39]. The ability to spatially organize enzymes for enhanced substrate channeling has important implications for *in vivo* applications [24, 35–37]. Despite this progress,
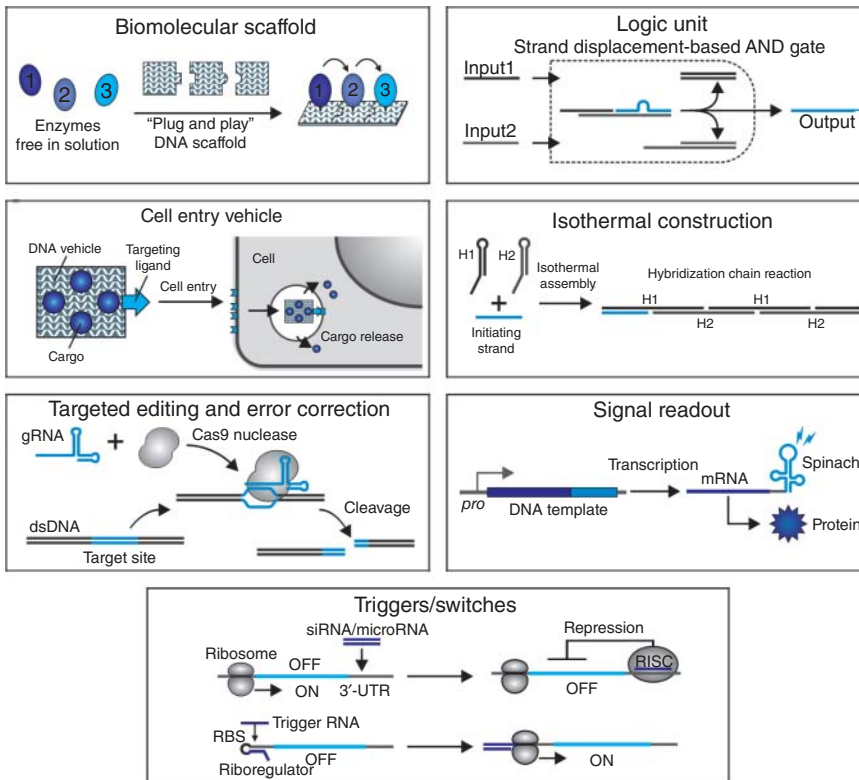
Figure 6.3 DNA/RNA nanotechnology-enabled toolbox for synthetic circuits. A diverse set of useful tools have been available, e.g. biomolecular scaffolds based on addressable DNA nanostructures [24], logic units based on DNA strand displacement reactions [25], DNA nanostructure cell entry vehicle [26], HCR-based isothermal construction of DNA structures [27], targeted editing and error correction based on CRISPR systems [28], signal readout based on fluorescent RNA motifs [29], and triggers/switches based on siRNAs/microRNAs [30] or riboregulators [31]. Source: From Yang et al. [24], Zhang and Seelig [25], Li et al. [26], Dirks and Pierce [27], Wright et al. [28], Paige et al. [29], Rinaudo et al. [30], and Green et al. [31].

scaling up these organized cascade reactions to large synthetic circuits mimicking intracellular signaling or metabolic pathways remains challenging. A very recent study demonstrating the assembly of FNA scaffolds in a "plug-and-play" manner represents a plausible solution to this problem [40].

6.2.2 Logic Units

Dynamic DNA structures incorporating functional nucleic acids (fNAs) (e.g. aptamers, DNAzymes, and RNAzymes) that are responsive to chemical/biochemical stimuli have been exploited to develop DNA logic gates that can be specifically triggered in physiological environments [25, 41, 42]. If integrated with strand exchange reactions [25], it is possible to transform input signals into the release of nucleic acids that can be relayed in cells to trigger subsequent

logic-driven gene expression. In addition, tensioned or compressed DNA motifs have been used as "entropic springs" that store stress energy that can be triggered to induce mechanical forces [17, 43].

Recent publications described the use of DNA as universal input/output (I/O) to construct cascadable logic gates (so-called seesaw gates) and form scalable prototype molecular circuits [44]. In addition to using linear DNA sequences, the introduction of DNA nanostructures allows expansion to 2D and 3D and offers additional operations (e.g. shape complementarity) [45]. Notably, new strategies including toehold exchange, strand displacement, hybridization chain reaction (HCR), and light-induced structural switching can dynamically and isothermally manipulate DNA hybridization, making biocomputing possible under physiological conditions [46–50]. In several recent reports, digital circuits regulating intracellular gene expression and mimicry of neural network computation have been realized using DNA- and RNA-based reactions [31, 51, 52].

fNAs provide new recognition and catalytic functions that nicely complement classic nucleic acid hybridization, which has been widely employed to process and execute biocomputation [53, 54]. Given the ubiquity of natural ribozymes in cells, delivery of designed DNAzymes/RNAzymes into cells to regulate biological circuitry is a highly promising approach [43, 55]. Nevertheless, the development of large-scale circuits *in vivo* remains challenging due to the need to avoid interference by the myriad biomolecules that are present [56, 57].

Reversibility is key to implementing continuous computation responsive to changing inputs, which is challenging for logic units based on kinetic-controlled nonequilibrium reactions. However, by using DNA structures that equilibrate between ON and OFF states [58], or DNA nanodevices possessing good mechanical reversibility [59], resettable DNA logic units have also been implemented.

6.2.3 Cell Entry Vehicles

Cytoplasmic membranes are largely negatively charged and form a natural electrostatic barrier for polyanionic nucleic acids. Interestingly, self-assembled FNAs are readily internalized by mammalian cells via energy-dependent endocytosis [60, 61]. This endocytosis efficiency is correlated to their structural properties including size, shape, and rigidity [62–64]. Several groups have employed FNAs as cell entry vehicles to deliver molecular payloads that can stimulate immunological responses and suppress tumor growth [65–67]. The use of biocompatible and degradable nucleic acids carriers largely circumvents the toxicological concerns of inorganic nanomaterial-based nanomedicine. Moreover, cargos can be covalently or non-covalently loaded on FNAs [26]. The addressability of FNAs allows precise control of the quantity and stoichiometric ratio of cargo molecules and cell-targeting ligands [61], which could enable the targeted intracellular delivery and controlled release of various small molecules, nucleic acids, and proteins [26, 67]. The tailorable nature of FNAs also opens new opportunities for developing dynamic, responsive vehicles to circumvent many barriers encountered at different stages in cell entry [68].

6.2.4   Isothermal Construction

DNA nanostructures are often self-assembled in test tubes by annealing at nonphysiological temperatures, limiting their potential for *in vivo* assembly under physiological conditions. A variety of isothermal strategies have recently been developed to fabricate DNA/RNA nanostructures, paving the way for intracellular DNA/RNA nanostructure replication. Several nanostructures with a few or even a single DNA/RNA strand have been successfully assembled using isothermal protocols at physiological temperatures using intramolecular self-folding [69] and HCR [27] assembly approaches. Structures composed of dozens of strands have been assembled using single-stranded DNA tiles [70]. Anderson and coworkers designed an elegant approach to co-transcriptionally fold single-stranded RNAs into well-defined nanostructures with T7 RNA polymerase [71]. Natural RNA motifs (e.g. transfer ribonucleic acids [tRNAs] and packing ribonucleic acids [pRNAs]) have also been utilized to construct nanoassemblies under mild conditions [72]. These nanostructures can be cloned in cells by delivering plasmids carrying DNA templates into host cells and rely on the intracellular transcriptional machinery to perform replication [73]. Transcription from plasmids has also been successfully applied to the assembly of periodic RNA nanostructures within *Escherichia coli* [74]. Very recently, synthesis of single-stranded DNA motifs in living bacteria has been realized by Elbaz et al. via reverse transcription [75]. Moreover, the Dietz group employed transcription activator-like (TAL) effector proteins as staples to isothermally fold dsDNAs into nanostructures at room temperature in near-physiological buffer conditions, which provides a potential path to fabrication of intracellular nanostructures using biosynthesized protein and nucleic acid components. Although the designability of isothermally folded nanostructures is still limited, the development of computer-assisted algorithms might provide a feasible solution to isothermal construction in cells and *in vivo* [71].

6.2.5   Targeting and Editing

Zinc finger nuclease (ZFN), transcription activator-like effector nuclease (TALEN), and clustered regularly interspaced short palindromic repeats (CRISPR) are three highly precise targeted gene editing tools [76]. The rapidly emerging RNA-guided CRISPR-Cas9 endonuclease system has revolutionized our ability to edit genomes in cells and even whole organisms. CRISPR is composed of two components: a Cas9 endonuclease and a guide RNA. Highly specific RNA/DNA hybridization guides Cas9 to targeted sites in the genome, where it precisely cuts gene fragments [77–79]. This approach opens a new door to efficiently rewrite naturally existing molecular circuitry and/or adapt artificial circuitry to molecular machinery in cells and animals [80]. We envision that the marriage of DNA nanotechnology with CRISPR might overcome some shortcomings of the latter (e.g. high off-target rates [81]), allow the former to better control biological circuits [80, 82, 83], and serve to augment CRISPR capabilities by coupling its activity with other enzymes on a DNA scaffold [28].

6.2.6   Signal Readout

Several conditionally fluorescent RNA motifs including "Spinach" and "Broccoli" are new tools for signal transduction and readout in cells. These live-cell fluorescent reporters mimic green fluorescent protein (GFP) to allow real-time monitoring of genetically associated processes and can be co-transcribed with other genetic modules in cells [29, 84, 85]. Hence, these fluorescent RNA motifs might be coupled with cloneable DNA/RNA nanostructures to signal the output of synthetic circuits. The fluorescent emission might be relayed to optogenetic control of cells and animals [86]. Nevertheless, *in vivo* signal reporting has long been a challenge due to strong background emission and limited imaging depth [87]. The development of fluorescent RNA motifs with near-infrared emission, and probably other advanced probes with two-photon excitation, magnetic resonance, or photoacoustic properties, should greatly expand the potential of *in vivo* biocomputing.

6.2.7   Triggers and Switches

Nucleic acids can perform various recognition and regulation functions aside from their genetic roles. Noncoding RNAs (e.g. microRNA and small interfering ribonucleic acid [siRNA]) are a class of naturally existing fNAs that dynamically regulate gene expression at the posttranscriptional level. Ribonucleic acid interference (RNAi) technology in particular has shown great potential in gene therapy and synthetic biology; it has been used to implement various synthetic logic gates functioning in live cells [30]. In addition, certain secondary structures at the 5′ untranslated region (UTR) of mRNAs (termed riboregulators) have been *de novo* designed as prokaryotic translational regulators that prevent binding of ribosomes to the ribosomal binding site (RBS), thereby blocking translation initiation. When a trans-activating RNA is introduced, the repressed secondary structure is unfolded by this trigger RNA via strand displacement, which leads to target gene expression [31, 88]. A wide assortment of other RNA-based systems has been developed to regulate at the transcriptional level [89, 90] and to repress gene expression in response to a trigger RNA [91, 92].

fNAs are readily obtained from genomic DNAs in natural organisms or systematic evolution of ligands by exponential enrichment (SELEX) to serve as gene regulation switches [93]. For example, endonuclease-like ribozymes are utilized to cleave mRNAs in a sequence-specific way to selectively repress their expression [94]. Notably, their catalytic activity often relies on certain ions or small molecules that are regarded as trans-factors for gene regulation [94]. Thus, a wide range of fNAs clearly enriches the toolbox for specific and tunable regulation and detection of intracellular events [95, 96].

6.2.8   Error Correction and Resilience

The precision of DNA base pairing does not guarantee error-free signal transduction in either *in vivo* replication or artificial molecular circuitry. In natural living systems, self-monitoring and repairing machineries (e.g. the immune system and

DNA damage repairing system) are indispensable for error correction and functional resilience of the system. These mechanisms provide inspiration for developing self-correction mechanisms within artificial biomolecular circuitry [97]. A straightforward approach to error correction is to make use of redundancy. In a typical DNA self-assembly reaction, over 10^{10} nanostructures can be fabricated in a 1 ml reaction system [11]. Hence, DNA nanotechnology enables massively parallel computing and the potential to design redundant units to reduce error rates [48, 98, 99]. In addition, multilayered DNA information storage systems (e.g. realized by selective modification of DNAs) [100] may also provide redundant space for error suppression tasks. These simple yet powerful mechanisms are expected to be adapted to increase the resilience of complicated synthetic systems. Alternatively, targeted editing may provide a more intelligent way to implement error correction in synthetic biology, especially with the rise of the CRISPR technology [101]. These gene editing tools provide unprecedented precision for cleaving and repairing error-containing segments via homologous recombination or homology-independent pathways *in vivo*.

6.3 Targeted Applications

Exploiting the nucleic acid nanotechnology-enabled toolbox makes it possible to develop novel synthetic circuitry that allows the rewiring of natural pathways for various applications. Here we summarize recent progress in drug delivery, cellular imaging, metabolic engineering, and cellular pathway investigation.

6.3.1 Drug Delivery

Targeted drug delivery holds the promise to improve drug efficacy and reduce side effects. In a typical protocol, drug molecules are conjugated with antibodies or aptamers that possess high affinity to the target. The Tan group demonstrated that DNA nanostructures appended with polyvalent aptamer motifs greatly improved targeted delivery [102, 103]. However, difficulty in finding highly specific ligand–receptor pairs largely restricts practical applications. Because DNA logic gates can process multiple inputs to make a pondered decision, it is possible to rely on DNA nanotechnology to integrate multiple factors subject to user-specified Boolean logic expressions to reliably distinguish targets. For example, the Church group developed a DNA origami-based logistic nanorobot triggered by two aptamers [43]. Using AND logic, the nanocontainer can only be opened when both aptamers bind their corresponding targets. A recent study demonstrated successful application of this strategy in living cockroaches [55]. Li et al. constructed a DNA nanorobot for intelligent release of thrombin to tumor-associated blood vessels, which can induce intravascular thrombosis, resulting in tumor necrosis and inhibition of tumor growth in mice [104].

6.3.2 Cellular Imaging

Given the flexibility and versatility in position-specific functionalization with various ligands and fluorophores, DNA nanostructures have demonstrated

great potential for targeted imaging in cells and *in vivo* [105, 106]. For example, Förster resonance energy transfer (FRET) pairs can be site-specifically anchored on well-defined dynamic DNA nanostructures to construct structural switching probes that undergo stimuli-responsive fluorescence changes in cells [41, 103]. The Krishnan group successfully performed spatiotemporal intracellular imaging of a variety of ions using responsive DNA nanostructures [107–110]. The incorporation of the CRISPR-Cas9 system may increase targeting ability in live cells, thus enabling real-time imaging of chromatin structure dynamics [111, 112]. However, this method proved suitable only for repetitive sequences or required tiling of guide RNAs along the target locus to yield detectable fluorescence signals. DNA nanostructures have been deployed in multiple previous reports to improve the sensitivity and resolution of cellular imaging. To image low-copy mRNAs in cells, the Pierce group exploited HCR to amplify fluorescent signals [113]. Jungmann et al. realized multiplexed 3D cellular super-resolution imaging by exploiting transient binding of short fluorescent-labeled oligonucleotides on synthetic DNA nanostructures, a method dubbed DNA-PAINT (point accumulation for imaging in nanoscale topography) [114, 115].

6.3.3 Metabolic Engineering and Cellular Pathway Investigation

Metabolic pathway efficiency largely depends on the spatiotemporal organization of the enzymes involved. Given the organizational capability of nucleic acid nanostructures, they are powerful tools for metabolic engineering in synthetic biology. Delebecque et al. used intracellularly assembled RNA structures as scaffolds to organize the interaction between hydrogenase and ferredoxin in bacteria [74]. The coupling efficiency of these enzymes was improved through control of their relative positions, which led to dramatic increases in hydrogen production. Using a similar strategy, several enzyme cascade systems have been engineered to realize high efficiency coupling [32, 35, 37]. The development of caged 3D DNA nanostructures enables further confinement of enzymes, reactants, and intermediates in 3D space, which could allow the construction of nanoscale reactors with high catalytic efficiency [36, 40].

FNA-mediated delivery of CRISPR-Cas9 and siRNAs also provide new ways to probe cellular pathways by allowing specific knock-in or knockout of target genes or regulatory elements. For example, with the use of CRISPR-targeted gene suppression, genetic logic circuits could be implemented to conditionally activate/deactivate genes in a pathway [80, 116]. Nucleic acid strand exchange was also employed to conditionally activate RNAi in live cells [57, 117]. Conditional gene knock-in and knockout tools are therefore very promising tools for future cellular pathway investigations.

6.4 Nucleic Acid Nanotechnology-Enabled Computing Kernel

A cell is in principle an elaborate natural biocomputer that collects physical/chemical information from the environment, performs calculations via signal

pathways or molecular circuitry, and uses this data to perform actions. Several types of live-cell logic gates and biocomputers have been developed by engineering gene regulatory networks in bacterial, yeast, and mammalian cells [30, 31, 80, 118–121]. To date, most designed biological circuits are highly case specific, making them difficult to adapt and transplant from one organism to another. Hence, it is highly desirable to develop a more general synthetic circuitry scheme. DNA and RNA are ideal tools for realizing such organism-independent circuitry given their ubiquity in all forms of life and their predictable base pairing interactions.

In an electronic computing kernel, rational compilation of basic elements (e.g. transistors and wires) can reach arbitrary complexity. Given the proven ability of DNA to encode and solve combinatorial/high-dimensional mathematical problems [122–125], it is envisioned that the computing kernel of a synthetic circuit can be similarly constructed by the rational assembly of a few basic DNA logic gates. Specifically, strand exchange of nucleic acids provides a route to wire DNA logic gates in cells and *in vivo*. The I/O in these reactions are nucleic acids that can seamlessly implement multilayer logic gates [48]. More importantly, strand exchange reaction kinetics can be finely tuned with rational sequence programming to fit the compilation of arbitrarily defined systems [25]. Hence, strand exchange appears to possess the universality and flexibility required for biocomputing.

Soloveichik et al. utilized strand exchange as a molecular primitive to implement the formalism of arbitrary chemical reaction networks [56]. They demonstrated several fascinating examples including a limit cycle oscillator, a chaotic system, and feedback logic systems, all of which provide inspiration for constructing generic computing modules. Qian and Winfree transformed logic unit form and function by simply introducing rationally designed auxiliary strands. Without changing other components, they could switch the gates between AND and OR functions by tuning the concentration of a "threshold" strand (Figure 6.4a) [44]. Green et al. employed multiple toehold switch riboregulators in *E. coli* to form a layered logic circuit for evaluating four-input AND expression using RNAs as inputs [31]. Toehold switches were also used to detect the expression of endogenous RNAs in the cell, demonstrating the capacity for FNAs to directly interface with native cellular RNAs. Very recently, the Seelig group demonstrated the successful use of strand exchange in biological environments for biocomputing [57]. By designing four-way strand exchange reactions, they activated functional siRNAs and realized subcellular co-localization of logic gate operation by exploiting native mRNA as a scaffold (Figure 6.4b). To further increase the sophistication of intracellular DNA/RNA circuitry, the use of FNAs as circuit boards to precisely integrate DNA calculation units (Figure 6.4c) has proved to substantially increase the computing power of strand exchange circuits through localization of reaction components, allowing implementations such as multi-input logic calculation, maze solving, and cargo sorting [43, 127–130]. Such localization approaches can improve circuit performance in much the same way that substrate channeling is known to increase metabolic reaction rates.

In addition to DNA/RNA, a range of enzymes including restriction endonucleases, ligases, recombinases, and polymerases can serve as biocomputational
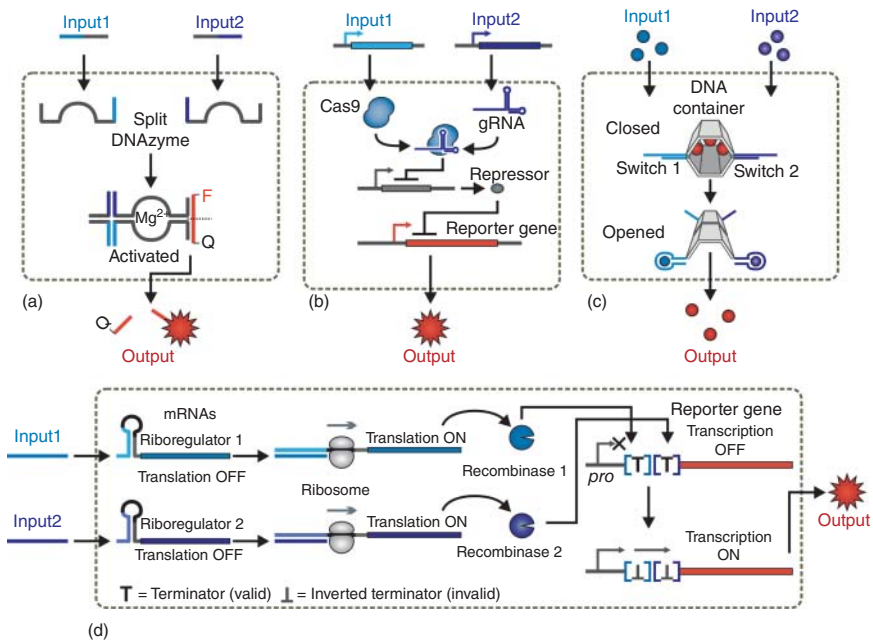
Figure 6.4 Typical AND gate circuits. (a) A DNAzyme-enabled AND gate. Source: Based on Green et al. [31]. (b) A CRISPR-Cas9 system. Source: Based on Liu et al. [116]. (c) A logic-gated DNA nanorobot that conditionally releases payload molecules. Source: From Douglas et al. [43] and Amir et al. [55]. (d) A combination of riboregulators and recombinases. Source: Based on Siuti et al. [126]. Based on these principles, other kinds of logic gates (OR, XOR, NOR, etc.) and logic-gated cascades can be readily implemented as well.

operators by specifically cutting, ligating, recombining, and replicating input nucleic acids under physiological conditions. These constitute programmable and autonomous computing machine components to manipulate gene expression in cells (Figure 6.4d) [126, 131, 132]. The recently developed CRISPR-Cas9 technology is in fact a combination of DNA/RNA hybridization and the restriction endonuclease activity, which offers highly flexible and precise genome targeting. These molecular tools should be adapted with DNA nanotechnology to regulate cellular networks and implement live-cell biocomputation [116]. Recently, a CRISPR-Cas9-based "core processor" prototype has been fabricated, which takes user-defined guide RNAs as inputs to program a transcriptional regulator, allowing a wide range of bitwise computations in live human cells, from simple Boolean logic gates to arithmetic operations such as the half-adder [133].

Natural biological systems often enable continuous responses to environmental stimuli due to the analog nature of biochemical reaction networks [118]. Hence, DNA-based analog computation represents an alternative yet flexible approach for dealing with wide-range graded signals (e.g. molecular concentrations) and producing output with rich information [118]. As an example, Qian and Winfree implemented analog time-domain circuits by exploiting DNA strand displacement reactions (SDRs) that are intrinsically analog [134]. More recently, by using a droplet-based microfluidic system, Genot et al. realized

massively parallel analog circuits for DNA-bistable switches and oscillators [135]. Given the remarkably high ability of analogy circuits for computing nonlinear functions [136], they are especially useful for high-throughput analysis in drug discovery, systematic evolution, and next-generation sequencing.

6.5 I/O and Human–Computer Interfacing

In modern electronic computers, a keyboard and mouse translate human actions to digital electronic signals, and the computed results are converted to visualizable output on the display. Recapitulating these capabilities using biomolecular components requires a diverse set of transducers for converting inputs into usable biomolecular signals and an array of outputs to modulate cell state and to return the results of the biocomputation. Regulatory proteins (e.g. receptors, kinases, and trans-regulatory factors) and nucleic acids (e.g. siRNA and microRNA) provide a rich library of molecular signal transducers to trigger intracellular events. They effectively put cells in a predefined state (e.g. protein phosphorylation and gene activation/deactivation), which is readily converted to optical output signals with the use of live-cell reporters (e.g. GFP). The readout device is often a fluorescence microscope that can image live cells and identify their specific states using coupled fluorescence. Although this approach has been widely employed to develop live-cell biocomputers, many of these systems involve highly specialized reactions under elaborate protocols.

DNA nanotechnology could provide a straightforward approach to modularly design user-friendly interfaces for signal I/O in which all signal transduction, information processing, and signal generation tasks are carried out by DNA/RNA circuit elements. First, since many DNA nanostructures are readily internalized by live cells [60, 105] and present numerous sites for binding, they can be employed as multifunctional delivery vehicles to carry regulatory proteins and DNA/RNA computing circuits into living organisms [55, 65]. Furthermore, these complexes can be guided to targeted organelles or organs with small ligands, aptamers, or peptide signals preloaded on DNA nanostructures [61, 67, 103]. Such systems would make it possible to design a universal system for bringing genetic instructions into the cell using DNA nanostructure-based modules. Second, dynamic DNA nanostructures incorporating FNAs offer great flexibility in designing signal transducers due to the availability of a wide range of FNAs identified by SELEX or mined from natural genomes. These transducers can specifically transform an input signal (often a chemical/biochemical stimulus) into digital, binary states (ON/OFF) via structural switching [137]. Regulatory biomolecules released during structural switching can be relayed and rewired to intracellular signal pathways or circuits [31, 88, 138]. Third, DNA/RNA circuitry can act directly on the nucleic acids produced during input signal transduction to process information and reach logical decisions. Fourth, once signals have been processed, DNA/RNA can be used to generate the output signals that are passed to an external observer or other cells. Such signals can take multiple forms, for instance, as a combination of new genes expressed by the

cell. Alternatively, conditionally fluorescent RNA motifs (e.g. Spinach, Broccoli) are GFP-like live-cell reporters that do not need to be translated [29, 84, 139], making them desirable as fluorescent signal outputs for cell-based biocomputation. We also envision that DNA nanotechnology may enable high-resolution "soft lithographic" patterns to present outputs with visualizable DNA origami that resemble display screens under atomic force microscopy [140, 141] or employ precise localization of fluorophores to provide highly multiplexed circuit readout [142].

Given these new opportunities offered by DNA nanotechnology, it is possible to standardize the control of signal I/O and achieve multi-signal integration and processing in live cells. The ability to develop fan-in and fan-out cell-based circuits should have valuable applications in multi-maker solution, pattern recognition, and neural network systems for intelligent disease diagnosis and therapy.

6.6 Information Storage

DNA molecules are arguably the most important and powerful carrier of information in nature. Several recent studies have shown that using DNA molecules to store digital information leads to extremely high capacity and durability, far exceeding those of currently available storage media [143, 144]. These studies exploited the nucleotide sequence information of DNA alone to store data. DNA nanostructures, however, provide a number of unique features that could be harnessed to substantially increase DNA information storage density and to provide read/write/erase access of stored information for biocomputing.

Similar to the chromatin structures found in eukaryotes, DNA nanostructures can condense DNA down to nearly the smallest possible volumes, yet without intervening proteins. 3D DNA origami, for instance, consists of tightly packed DNA double helices with inter-helix spaces down to only 0.1–0.4 nm [18]. Thus, storing information within a compact DNA nanostructure could yield immediate increases in the number of bits per unit volume. The structural control afforded by DNA nanostructures also provides additional means of encoding information beyond the basic DNA sequence specification. Just as the information contained in digital files is defined by their file format, information contained within DNA could be stored in 3D assemblies with a precisely defined structure in terms of the number of helices, turns, topology, and overall geometry. Knowing the "file format" of this genetic information, sequencing results could be threaded into the predefined target structure. Thus, each element of sequence information can also be tied to its location within the DNA nanostructure, increasing the overall information content of the DNA itself. This additional information content could be used to store more data per unit volume or mass or to establish additional error correction and redundancy measures.

Nucleic acid nanostructures formed within the cells could also serve as repositories for storing and presenting information concerning the past history

of the cell. Nanostructures can be constructed with a programmable number of addressable capture or release sites (e.g. DNA/RNA binding sites or aptamers) that can interact with different species produced by the cell, such as RNAs, proteins, and small molecules. The binding and unbinding of these species effectively become the write/erase elements of the information storage device. The resulting intracellular recorders can then be read by DNA-based logic circuitry that directly interfaces with nucleic acid nanostructure to process the stored information. Beyond serving as breadboards for information storage, nanostructures also provide the opportunity to dynamically change their geometry in response to intracellular events, such as the binding of RNA molecules or interactions with a protein. Structure switching can be used to control access to the stored information in much the same way that chromatin structures in eukaryotes exert spatiotemporal control over the availability of genetic information [145].

Application of DNA nanostructure-based storage media to biocomputing will also benefit from new mechanisms for reading, writing, and copying data. The Sleiman group demonstrated the transfer of encoded DNA strands on different 3D DNA nanostructures to the surface of gold nanoparticles in a manner similar to lithography [146]. This method provides a way to read multidimensional information from DNA storage media. To realize DNA writing in live cells, Farzadfard and Lu developed a genetically encoded approach to record arbitrary transcriptional signals in genomic DNA by generating targeted mutations [147]. The stored information could be read out with fluorescence imaging of reporter genes. Recombinases have been utilized to record status information on genomic DNA, enabling the implementation of complex cellular state machines [126]. Recently, the CRISPR-Cas system has also been exploited to addressably write information on genomic DNA in live cells, which have proven to be a robust method for storing practical amounts of real data [148–150]. Inspired by epigenetics, Mayer et al. recently demonstrated that multiple layers of information could be encoded in and retrieved from a single DNA template via selective chemical modifications of nucleobases. In addition, advances in autonomous self-replication and the cloneable fabrication of nucleic acid nanostructures are paving the way to realizing information duplication and signal amplification in cells and *in vivo* [113, 147, 151].

6.7 Perspectives

Given their programmable and self-assembling nature, artificial DNA nanostructures can provide a library of modularly designed scaffolds for spatially organizing DNA/RNA-based logic gates into synthetic circuits reminiscent of electronic breadboards but with DNA/RNA as the principal signal carrier. Use of DNA/RNA ensures that there is substantial signal bandwidth for information processing and diverse options for input and output signals. This vision for DNA/RNA nanotechnology-based live-cell circuitry is outlined schematically in Figure 6.5. With the assistance of cell-targeting aptamers and lipid-functionalized
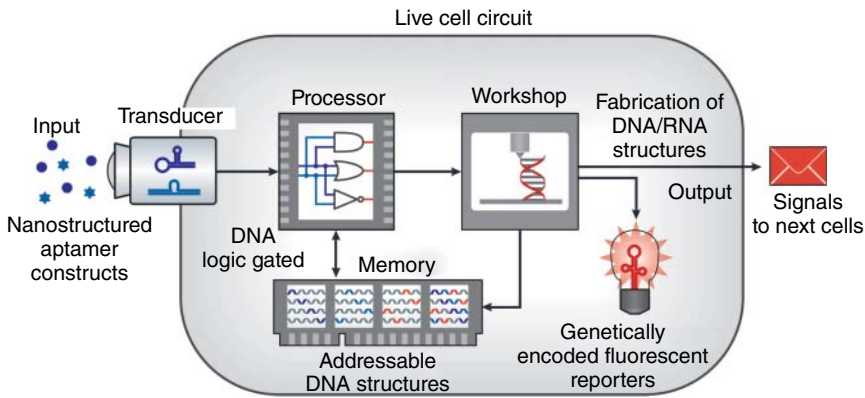
Figure 6.5 Scheme of an integrated live-cell circuit enabled by DNA/RNA nanotechnology. Inputs from outside the cell are converted via transducers into DNA/RNA signals that interface directly with intracellular DNA/RNA-based circuitry and memory storage elements. The biosynthetic capabilities of the cell itself are used to fabricate DNA/RNA nanostructures and to produce output signals for a human operator or the intended recipient cells. This live-cell circuit holds promise in applications like diagnosis, therapy, optogenetics, biomanufacturing, and robotics. Source: From Li et al. [1]. Reproduced with the permission of Springer.

nanostructures, programmable DNA-based input signal transducers can be positioned at the cell membrane to monitor the presence of circulating input molecules via FNAs. Upon binding, input signals are converted into nucleic acid signal molecules that are passed on to intracellular DNA/RNA-based processor and memory elements that perform biocomputation and memory storage operations, respectively. Both processor and memory units can communicate with one another via nucleic acid signals. Once a decision is reached or more circuit elements are needed, the DNA/RNA circuitry can harness the cell as a biomolecular assembly plant to fabricate the required components. DNA or RNA synthesized by the cell can be programmed to self-assemble into new processor and memory storage modules. Outputs in the form of proteins, genetically encoded fluorescent reporters (e.g. GFP, spinach), or DNA/RNA circuitry can be synthesized and passed as signals to other cells or used within the parent cell itself.

Such DNA/RNA nanotechnology-based circuits can also make use of non-nucleic acid functional elements such as endonucleases, recombinases, cell-targeting proteins, and the CRISPR complex to expand their function. These elements along with FNAs may also be precisely incorporated in fixed copy numbers to receive input from environmental or cellular signals, specifically target organelles in cells or organs *in vivo*, and responsively trigger signal pathways in a predefined time series. The outputs in the form of new biomolecules from such circuits can be used to correct a malignant cell state, induce apoptosis, or initiate production and release of daughter DNA/RNA circuits. Hence, we expect that DNA nanotechnology will provide great opportunities to rewire intracellular signal pathways in a plug-and-play manner. Such engineered

molecular circuitry could be repurposed for applications including diagnosis, cancer therapy, optogenetics, biomanufacturing, and robotics.

DNA nanotechnology has played important roles in many facets of cell-based biocomputation [55, 105, 152]. However, our fundamental understanding of how DNA nanostructures function in cells remains poor. Additional studies are needed to clarify their structural–functional relationship to optimize intracellular delivery, elucidate their endocytic pathways and intracellular traffic, and understand their circulation, distribution, metabolism, and excretion *in vivo*. Furthermore, it is imperative to refine DNA nanotechnology to allow its application in synthetic biology. Indeed, there have been several examples of DNA nanotechnology-based logical control *in vivo* [55, 153, 154]. For instance, DNA nanorobots implementing various logic calculations have been deployed in living cockroaches, which can release different drug molecules in response to protein cues [55]. Notably, *in vivo* implementation of a half-adder has also been realized, revealing the high potential of complex biocomputation in living systems. In other examples, by using DNA/RNA nanodevices, siRNAs have also been delivered and activated in living cells based on logic calculations, resulting in conditional knocking down of target genes [57, 155]. Yet, given the complex environment in living systems, grand challenges remain for *in vivo* biocomputation.

Concerning the speed of a single logic operation, DNA/RNA-based computation is intrinsically limited by the kinetics of chemical reactions; a typical DNA hybridization reaction takes seconds, which is 9–10 orders of magnitude slower than an electronic logic switch. Nevertheless, given the massively parallel operation of DNA/RNA-based biocomputation, it is probably wise to employ it for solving combinatorial problems. More importantly, the potential of seamlessly integrating biocomputation within living organisms makes DNA/RNA circuitry appropriate for biomedical applications, where operations that take place over minutes are adequate.

The cellular environment is generally complex and hostile to foreign structures due to the presence of various ions, enzymes, and endogenous nucleic acids. Although DNA nanostructures are more resistant to enzymatic degradation than single- or double-stranded DNA [156, 157], their stability remains a major concern when performing complicated computations in cells and *in vivo*. Previous studies have shown that DNA nanostructures remain intact in cell lysates and live cells for up to several hours, and these results form the basis for intracellular drug delivery and cellular imaging [60, 158, 159]. However, *in vivo* studies revealed that the blood circulation time of DNA nanostructures in mice was at the minute scale [160], which seemingly restricts their biomedical applications. Recent investigations also reported that DNA origami nanostructures tended to disintegrate in the presence of physiological concentrations of Mg^{2+}, which could compromise their intracellular and *in vivo* use [160]. However, a recent study showed that DNA origami structures preferentially retained in the kidneys of mice and possessed renal protective properties [161] thus have promise in *in vivo* therapeutic applications.

Several approaches have been developed to further stabilize DNA nanostructures in cells by ligating free ends that are vulnerable to nucleases or by

designing closed and interlocked structures [162, 163]. Niemeyer and coworkers reported that non-covalent modification with intercalators could rigidify DNA origami to prevent disintegration [164]. Also, it might be possible to enhance intracellular DNA nanostructure stability by combining them with metal nanoparticles [165, 166], inorganic minerals [167, 168], or lipids [34]. Seelig's group recently presented an elegant example showing operation of nucleic acid-based computing in cells by optimizing various factors including composition, chemical modifications, and transfection [57]. Despite the progress, there is no simple and generic approach to engineer DNA nanostructures for reliable intracellular applications. The goal remains to design DNA nanostructures with long-term intracellular and *in vivo* stability. At this stage, it might be wise to exploit their conditional stability to temporally trigger controlled intracellular release, as exemplified by several elegant designs and applications [169–171]. Alternatively, DNA/RNA nanostructures could be continually synthesized within the cell [74] and reach a stable steady state where the rate of nanostructure repair matches the rate of degradation.

The advantages of DNA nanotechnology are often compromised by the high cost of large-scale DNA nanostructure synthesis, which greatly hampers its practical application and the ability to screen large numbers of potential structures for optimal function. Chip-based *de novo* DNA synthesis provides a new avenue to produce large numbers (>20 000) [172] of different DNA strands [173]. Shih and coworkers developed a multi-round rolling circle amplification strategy to selectively amplify sequences from chip-synthesized libraries to produce DNA origami and DNA brick structures [174]. Use of these protocols could enable up to ~100 DNA origami designs to be tested from a single chip, facilitating the screening of nanostructure designs for desired properties, such as cellular uptake, endosomal release, and *in vivo* stability. Chip-based oligonucleotides can thus be used to identify nanostructures with a desirable combination of properties that can later be scaled. Although there have been several approaches to controllably assemble DNA nanostructures with nearly 100% yield [11, 15, 175], the requirement for hundreds of oligonucleotides to form a single structure makes production beyond the μg–mg scale challenging. Cloneable DNA nanostructures that exploit natural DNA biosynthesis systems in cells are a potential solution to the scale problem [69]. RNA is also an intriguing alternative for large-scale nucleic acid nanostructure synthesis. RNA nanostructures have the potential to be produced intracellularly through co-transcriptional folding using approaches like those described by Geary et al. [71]. In addition, previous studies have shown that nonstructured recombinant RNA molecules can be produced at levels as high as $4.5\,mg\,l^{-1}$ of bacterial culture [176], while structured RNAs can be obtained at up to $50\,mg\,l^{-1}$ of culture [177]. The bacterium *Rhodovulum sulfidophilum*, which naturally produces extracellular RNAs, has been harnessed to secrete recombinant RNA aptamers into its growth medium, suggesting potential uses for continuous RNA production in industrial bioreactors [178]. Hence, with continued improvements, a combination of the above RNA assembly and biosynthesis methods could enable expansion into gram-scale production levels.

Intracellular environments are highly heterogeneous in comparison with the homogeneous solutions of test tubes. Because biomolecule motion and distribution are generally restricted in time and space by vesicular membranes and cellular trafficking, DNA nanostructures do not naturally locate themselves in the right place or time. For these reasons, artificial circuits that work well in test tubes may not be effective in live cells. To address this problem, several groups developed modular DNA nanostructures coupled with specific peptide sequences that are widely recognized as signals for transcellular/transmembrane transportation and intracellular localization [61, 179, 180]. For example, cell-penetrating peptides (CPPs) improve the internalization efficiency of DNA origami nanostructures [166], and nuclear localization sequences (NLS) can guide the entry of tetrahedral DNA nanostructures into nuclei [61]. On a related note, it is challenging to couple peptides to DNA nanostructures with high efficiency, but several non-covalent coupling approaches were recently developed to overcome this problem. The researchers exploited naturally existing specific interactions between nucleic acids and proteins (e.g. transcription factors [181] and zinc fingers [182]) to facilitate quantitative coupling. This bioinspired wisdom is expected to endow DNA nanotechnology with desirable targeting abilities in cells and *in vivo*.

It is probably even more challenging to realize *in vivo* biocomputation by rewiring existing signaling networks in a prescribed way at the multicellular/organism level. Tamsir et al. developed an elegant strategy to implement multicell computation by rewiring intercellular communication among cell colonies [183]. Reengineering the maternal genes on *Drosophila* has also been realized, which could lead to animal-scale population control [153]. Given the utility of CRISPR-Cas9 in genetically modifying embryos and organisms with high precision, it has been possible to directly translate designed synthetic networks from the cellular level to the tissue, organ, or even organism level with unprecedented power [76, 184, 185]. Despite these advances, the off-target rate and delivery efficiency of CRISPR-Cas9 are still two major concerns [28]. The use of elaborately designed DNA nanostructures to overcome these problems may open new doors for *in vivo* computing in whole organisms [186].

6.8   Conclusion

In summary, the recent progress in DNA/RNA nanotechnology has provided exciting opportunities to precisely manipulate naturally existing signal pathways and networks. By discovering and adopting new rules of nucleic acid-based molecular design and programming, we are now in a position to rewire signal pathways in cells and even *in vivo*. Nevertheless, numerous challenges must be overcome before achieving the ultimate goal of reconstructing a modularized, transplantable, and versatile integrated circuitry systems for synthetic biology.

6.8.1 Terminology

Framework nucleic acids (FNAs)	Shell or skeleton DNA/RNA nanostructures serving as frameworks for spatial organization of molecules or materials [21]. Because FNAs are monodispersed nanostructures with near-atomistic precision, they can be used to precisely define the valence, spatial arrangements, and microenvironments of guest molecules or materials, which may lead to unique chemical/physical/biological properties
DNA origami	A method for fabricating finite DNA nanostructures [15]. It employs hundreds of short oligonucleotides (staples) to fold a long ssDNA strand (scaffold, often the genomic DNA of M13 bacteriophage), resulting in high yields of predefined shapes or patterns with nanometer precision and addressability
CRISPR-Cas	Clustered regularly interspaced short palindromic repeats (CRISPR) are prokaryotic DNA segments spaced by foreign sequences. Their transcripts, known as CRISPR RNAs (crRNAs), can guide CRISPR-associated (Cas) endonucleases to recognize and cut foreign DNAs. They together constitute the prokaryotic adaptive immune system known as the CRISPR-Cas system. CRISPR-Cas-based genome editing allows permanent modification of genes in eukaryotic organisms [77, 187]
Logic unit	A computational unit that implements a fixed logic function (e.g. AND, OR, NOT) with one, two, or more binary inputs and produces a binary output. Assembly of logic units yields digital circuits capable of evaluating complex calculations
DNA computing	A type of biocomputation using DNA and DNA-involved reactions. In the pioneering work reported by Adleman in 1994 [188], a seven-point Hamiltonian path problem was solved by applying operations (including enzymatic ligation, PCR amplification, and separation) on DNA strands that encoded input information. The output was visualized with gel electrophoresis
Digital computing and analog computing	Digital computing is based on logic units that deal with binary values (discrete states). Thus, for continuous signals commonly seen in nature, analog-to-digital conversion is required. In contrast, analog computing is based on gates that can directly compute using continuous values and is thus fast and cost-efficient in solving problems like differential equations. So far, both modes of computing have been implemented using DNA-based models.
Functional nucleic acids (fNAs)	A class of nucleic acids that either possess cofactor-dependent catalytic activities toward their substrate (known as DNAzymes/RNAzymes, also termed deoxyribozymes/ribozymes, or catalytic DNAs/RNAs) or show high affinity to certain targets (known as aptamers) from metal ions to small molecules, drugs, proteins, DNA/RNA, and even whole cells. Binding of fNAs to their targets often leads to changes in fNA structure, which can be used to generate output signals in DNA/RNA-based circuits. Stefanovic and coworkers and Willner et al. implemented a series of logic gates using fNAs [54, 189, 190].

Strand displacement reaction (SDR)	The process through which an oligonucleotide can initially bind to a partially double-stranded complex by a single-stranded domain called a toehold and then displace and release the originally bound strand after branch migration occurs. A series of enzyme-free logic gates and related circuits have been developed based on SDR [25, 48]. These gates also use oligonucleotides as inputs and outputs, enabling large-scale circuits with thresholding and catalysis within every logical operation to perform digital signal restoration [44]
Fluorescent RNAs	RNA motifs that fluorescence when bound to dye molecules, which enables facile fluorescent tagging of endogenous RNAs through genetic fusion in a way similar to fluorescent proteins. Jaffrey and coworkers reported the "Spinach" RNA aptamer that binds to the dye DFHBI to emit green fluorescence [84]
Riboregulator	An engineered RNA motif that can regulate the translation or transcription of a gene though a conformational change mediated by other nucleic acids. In a classic strategy developed by Collins and coworkers [88], a motif is inserted into the 5′ untranslated region (UTR) of an mRNA, forming a secondary structure that sequesters the ribosomal binding site (RBS) and prevents the ribosome from initiating translation. Binding of a complementary nucleic acid releases the RBS from the repressing secondary structure and allows the ribosome to begin translation.

References

1 Li, J., Green, A.A., Yan, H., and Fan, C. (2017). *Nat. Chem.* 9: 1056–1067.

2 Mandal, M. and Breaker, R.R. (2004). *Nat. Rev. Mol. Cell Biol.* 5: 451–463.

3 Uhlenbeck, O.C. (1987). *Nature* 328: 596–600.

4 Sanger, H.L., Klotz, G., Riesner, D. et al. (1976). *Proc. Natl. Acad. Sci. U.S.A.* 73: 3852–3856.

5 Sen, D. and Gilbert, W. (1988). *Nature* 334: 364–366.

6 Gehring, K., Leroy, J.-L., and Guéron, M. (1993). *Nature* 363: 561–565.

7 Jones, M.R., Seeman, N.C., and Mirkin, C.A. (2015). *Science* 347: 1260901.

8 Gerling, T., Wagenbauer, K.F., Neuner, A.M., and Dietz, H. (2015). *Science* 347: 1446–1452.

9 Winfree, E., Liu, F., Wenzler, L.A., and Seeman, N.C. (1998). *Nature* 394: 539–544.

10 Yan, H., Park, S.H., Finkelstein, G. et al. (2003). *Science* 301: 1882–1884.

11 Goodman, R.P., Schaap, I.A., Tardin, C.F. et al. (2005). *Science* 310: 1661–1665.

12 Zheng, J.P., Birktoft, J.J., Chen, Y. et al. (2009). *Nature* 461: 74–77.

13 He, Y., Ye, T., Su, M. et al. (2008). *Nature* 452: 198–U141.

14 Ke, Y., Ong, L.L., Shih, W.M., and Yin, P. (2012). *Science* 338: 1177–1183.

15 Rothemund, P.W.K. (2006). *Nature* 440: 297–302.

16 Zhang, F., Jiang, S., Wu, S. et al. (2015). *Nat. Nanotechnol.* 10: 779–784.

17 Andersen, E.S., Dong, M., Nielsen, M.M. et al. (2009). *Nature* 459: 73–76.

18 Douglas, S.M., Dietz, H., Liedl, T. et al. (2009). *Nature* 459: 414–418.

19 Han, D., Pal, S., Nangreave, J. et al. (2011). *Science* 332: 342–346.

20 Benson, E., Mohammed, A., Gardell, J. et al. (2015). *Nature* 523: 441–444.

21 Ge, Z., Gu, H., Li, Q., and Fan, C. (2018). *J. Am. Chem. Soc.* 140: 17808–17819.

22 Benner, S.A. and Sismour, A.M. (2005). *Nat. Rev. Genet.* 6: 533–543.

23 Church, G.M., Elowitz, M.B., Smolke, C.D. et al. (2014). *Nat. Rev. Mol. Cell Biol.* 15: 289–294.

24 Yang, Y.R., Liu, Y., and Yan, H. (2015). *Bioconjugate Chem.* 22: 22.

25 Zhang, D.Y. and Seelig, G. (2011). *Nat. Chem.* 3: 103–113.

26 Li, J., Fan, C., Pei, H. et al. (2013). *Adv. Mater.* 25: 4386–4396.

27 Dirks, R.M. and Pierce, N.A. (2004). *Proc. Natl. Acad. Sci. U.S.A.* 101: 15275–15278.

28 Wright, A.V., Nunez, J.K., and Doudna, J.A. (2016). *Cell* 164: 29–44.

29 Paige, J.S., Nguyen-Duc, T., Song, W., and Jaffrey, S.R. (2012). *Science* 335: 1194.

30 Rinaudo, K., Bleris, L., Maddamsetti, R. et al. (2007). *Nat. Biotechnol.* 25: 795–801.

31 Green, A.A., Silver, P.A., Collins, J.J., and Yin, P. (2014). *Cell* 159: 925–939.

32 Wilner, O.I., Weizmann, Y., Gill, R. et al. (2009). *Nat. Nanotechnol.* 4: 249–254.

33 Myhrvold, C. and Silver, P.A. (2015). *Nat. Struct. Mol. Biol.* 22: 8–10.

34 Perrault, S.D. and Shih, W.M. (2014). *ACS Nano* 8: 5132–5140.

35 Fu, J., Yang, Y.R., Johnson-Buck, A. et al. (2014). *Nat. Nanotechnol.* 9: 531–536.

36 Fu, Y.M., Zeng, D.D., Chao, J. et al. (2013). *J. Am. Chem. Soc.* 135: 696–702.

37 Fu, J.L., Liu, M.H., Liu, Y. et al. (2012). *J. Am. Chem. Soc.* 134: 5516–5519.

38 Pal, S., Deng, Z.T., Wang, H.N. et al. (2011). *J. Am. Chem. Soc.* 133: 17606–17609.

39 Maune, H.T., Han, S.P., Barish, R.D. et al. (2010). *Nat. Nanotechnol.* 5: 61–66.

40 Zhao, Z., Fu, J., Dhakal, S. et al. (2016). *Nat. Commun.* 7: 10619.

41 Pei, H., Liang, L., Yao, G. et al. (2012). *Angew. Chem. Int. Ed.* 51: 9020–9024.

42 Benenson, Y. (2012). *Nat. Rev. Genet.* 13: 455–468.

43 Douglas, S.M., Bachelet, I., and Church, G.M. (2012). *Science* 335: 831–834.

44 Qian, L. and Winfree, E. (2011). *Science* 332: 1196–1201.

45 Mao, C.D., LaBean, T.H., Reif, J.H., and Seeman, N.C. (2000). *Nature* 407: 493–496.

46 Zhang, D.Y. and Winfree, E. (2009). *J. Am. Chem. Soc.* 131: 17303–17314.

47 Bi, S., Chen, M., Jia, X. et al. (2015). *Angew. Chem. Int. Ed.* 54: 8144–8148.

48 Seelig, G., Soloveichik, D., Zhang, D.Y., and Winfree, E. (2006). *Science* 314: 1585–1588.

49 Zhou, M.G., Liang, X.G., Mochizuki, T., and Asanuma, H. (2010). *Angew. Chem. Int. Ed.* 49: 2167–2170.

50 Lohmann, F., Weigandt, J., Valero, J., and Famulok, M. (2014). *Angew. Chem. Int. Ed.* 53: 10372–10376.

51 Qian, L., Winfree, E., and Bruck, J. (2011). *Nature* 475: 368–372.

52 Cherry, K.M. and Qian, L.L. (2018). *Nature* 559: 370–376.

53 Elbaz, J., Lioubashevski, O., Wang, F.A. et al. (2010). *Nat. Nanotechnol.* 5: 417–422.

54 Willner, I., Shlyahovsky, B., Zayats, M., and Willner, B. (2008). *Chem. Soc. Rev.* 37: 1153–1165.

55 Amir, Y., Ben-Ishay, E., Levner, D. et al. (2014). *Nat. Nanotechnol.* 9: 353–357.

56 Soloveichik, D., Seelig, G., and Winfree, E. (2010). *Proc. Natl. Acad. Sci. U.S.A.* 107: 5393–5398.

57 Groves, B., Chen, Y.-J., Zurla, C. et al. (2015). *Nat. Nanotechnol.* 11: 287–294.

58 Genot, A.J., Bath, J., and Turberfield, A.J. (2011). *J. Am. Chem. Soc.* 133: 20080–20083.

59 Li, T., Lohmann, F., and Famulok, M. (2014). *Nat. Commun.* 5: 4940.

60 Walsh, A.S., Yin, H., Erben, C.M. et al. (2011). *ACS Nano* 5: 5427–5432.

61 Liang, L., Li, J., Li, Q. et al. (2014). *Angew. Chem. Int. Ed.* 53: 7745–7750.

62 Ding, H., Li, J., Chen, N. et al. (2018). *ACS Cent. Sci.* 4: 1344–1351.

63 Bastings, M.M.C., Anastassacos, F.M., Ponnuswamy, N. et al. (2018). *Nano Lett.* 18: 3557–3564.

64 Wang, P.F., Rahman, M.A., Zhao, Z.X. et al. (2018). *J. Am. Chem. Soc.* 140: 2478–2484.

65 Liu, X., Xu, Y., Yu, T. et al. (2012). *Nano Lett.* 12: 4254–4259.

66 Jiang, Q., Song, C., Nangreave, J. et al. (2012). *J. Am. Chem. Soc.* 134: 13396–13403.

67 Lee, H., Lytton-Jean, A.K.R., Chen, Y. et al. (2012). *Nat. Nanotechnol.* 7: 389–393.

68 Ohta, S., Glancy, D., and Chan, W.C. (2016). *Science* 351: 841–845.

69 Lin, C., Rinker, S., Wang, X. et al. (2008). *Proc. Natl. Acad. Sci. U.S.A.* 105: 17626–17631.

70 Myhrvold, C., Dai, M., Silver, P.A., and Yin, P. (2013). *Nano Lett.* 13: 4242–4248.

71 Geary, C., Rothemund, P.W.K., and Andersen, E.S. (2014). *Science* 345: 799–804.

72 Hao, C.H., Li, X., Tian, C. et al. (2014). *Nat. Commun.* 5: 3890.

73 Grabow, W.W. and Jaeger, L. (2014). *Acc. Chem. Res.* 47: 1871–1880.

74 Delebecque, C.J., Lindner, A.B., Silver, P.A., and Aldaye, F.A. (2011). *Science* 333: 470–474.

75 Elbaz, J., Yin, P., and Voigt, C.A. (2016). *Nat. Commun.* 7: 11179.

76 Gaj, T., Gersbach, C.A., and Barbas, C.F. 3rd, (2013). *Trends Biotechnol.* 31: 397–405.

77 Cong, L., Ran, F.A., Cox, D. et al. (2013). *Science* 339: 819–823.

78 Mali, P., Yang, L., Esvelt, K.M. et al. (2013). *Science* 339: 823–826.

79 Jinek, M., East, A., Cheng, A. et al. (2013). *eLife* 2: e00471.

80 Kiani, S., Beal, J., Ebrahimkhani, M.R. et al. (2014). *Nat. Methods* 11: 723–726.

81 Tsai, S.Q., Wyvekens, N., Khayter, C. et al. (2014). *Nat. Biotechnol.* 32: 569–576.

82 Gilbert, L.A., Larson, M.H., Morsut, L. et al. (2013). *Cell* 154: 442–451.

83 Qi, L.S., Larson, M.H., Gilbert, L.A. et al. (2013). *Cell* 152: 1173–1183.

84 Paige, J.S., Wu, K.Y., and Jaffrey, S.R. (2011). *Science* 333: 642–646.

85 Filonov, G.S., Moon, J.D., Svensen, N., and Jaffrey, S.R. (2014). *J. Am. Chem. Soc.* 136: 16299–16308.

86 Tye, K.M. and Deisseroth, K. (2012). *Nat. Rev. Neurosci.* 13: 251–266.

87 Antaris, A.L., Chen, H., Cheng, K. et al. (2016). *Nat. Mater.* 15: 235–242.

88 Isaacs, F.J., Dwyer, D.J., Ding, C. et al. (2004). *Nat. Biotechnol.* 22: 841–847.

89 Lucks, J.B., Qi, L., Mutalik, V.K. et al. (2011). *Proc. Natl. Acad. Sci. U.S.A.* 108: 8617–8622.

90 Chappell, J., Takahashi, M.K., and Lucks, J.B. (2015). *Nat. Chem. Biol.* 11: 214–220.

91 Mutalik, V.K., Qi, L., Guimaraes, J.C. et al. (2012). *Nat. Chem. Biol.* 8: 447–454.

92 Takahashi, M.K. and Lucks, J.B. (2013). *Nucleic Acids Res.* 41: 7577–7588.

93 Bayer, T.S. and Smolke, C.D. (2005). *Nat. Biotechnol.* 23: 337–343.

94 Winkler, W.C., Nahvi, A., Roth, A. et al. (2004). *Nature* 428: 281–286.

95 Isaacs, F.J., Dwyer, D.J., and Collins, J.J. (2006). *Nat. Biotechnol.* 24: 545–554.

96 Callura, J.M., Cantor, C.R., and Collins, J.J. (2012). *Proc. Natl. Acad. Sci. U.S.A.* 109: 5850–5855.

97 Heath, J.R., Kuekes, P.J., Snider, G.S., and Williams, R.S. (1998). *Science* 280: 1716–1721.

98 Fujibayashi, K., Zhang, D.Y., Winfree, E., and Murata, S. (2008). *Nat. Comput.* 8: 589–612.

99 Schulman, R., Wright, C., and Winfree, E. (2015). *ACS Nano* 9: 5760–5771.

100 Mayer, C., McInroy, G.R., Murat, P. et al. (2016). *Angew. Chem. Int. Ed.* 55: 11144–11148.

101 Brouns, S.J.J., Jore, M.M., Lundgren, M. et al. (2008). *Science* 321: 960–964.

102 Zhu, G.Z., Hu, R., Zhao, Z.L. et al. (2013). *J. Am. Chem. Soc.* 135: 16438–16445.

103 Hu, R., Zhang, X.B., Zhao, Z.L. et al. (2014). *Angew. Chem. Int. Ed.* 53: 5821–5826.

104 Li, S., Jiang, Q., Liu, S. et al. (2018). *Nat. Biotechnol.* 36: 258.

105 Modi, S., Swetha, M.G., Goswami, D. et al. (2009). *Nat. Nanotechnol.* 4: 325–330.

106 Bhatia, D., Surana, S., Chakraborty, S. et al. (2011). *Nat. Commun.* 2: 339.

107 Modi, S., Nizak, C., Surana, S. et al. (2013). *Nat. Nanotechnol.* 8: 459–467.

108 Thekkan, S., Jani, M.S., Cui, C. et al. (2019). *Nat. Chem. Biol.* 15: 1165–1172.

109 Narayanaswamy, N., Chakraborty, K., Saminathan, A. et al. (2019). *Nat. Methods* 16: 95–102.

110 Leung, K., Chakraborty, K., Saminathan, A., and Krishnan, Y. (2019). *Nat. Nanotechnol.* 14: 176–183.

111 Chen, B., Gilbert, L.A., Cimini, B.A. et al. (2013). *Cell* 155: 1479–1491.

112 Ma, H., Naseri, A., Reyes-Gutierrez, P. et al. (2015). *Proc. Natl. Acad. Sci. U.S.A.* 112: 3002–3007.

113 Choi, H.M., Chang, J.Y., Trinh le, A. et al. (2010). *Nat. Biotechnol.* 28: 1208–1212.

114 Jungmann, R., Steinhauer, C., Scheible, M. et al. (2010). *Nano Lett.* 10: 4756–4761.

115 Jungmann, R., Avendano, M.S., Woehrstein, J.B. et al. (2014). *Nat. Methods* 11: 313–318.

116 Liu, Y., Zeng, Y., Liu, L. et al. (2014). *Nat. Commun.* 5: 5393.

117 Bindewald, E., Afonin, K.A., Viard, M. et al. (2016). *Nano Lett.* 16: 1726–1735.

118 Daniel, R., Rubens, J.R., Sarpeshkar, R., and Lu, T.K. (2013). *Nature* 497: 619–623.

119 Damle, S.S. and Davidson, E.H. (2012). *Proc. Natl. Acad. Sci. U.S.A.* 109: 1548–1553.

120 Auslander, S., Auslander, D., Muller, M. et al. (2012). *Nature* 487: 123–127.

121 Nielsen, A.A., Der, B.S., Shin, J. et al. (2016). *Science* 352: aac7341.

122 Lipton, R.J. (1995). *Science* 268: 542–545.

123 Ouyang, Q., Kaplan, P.D., Liu, S., and Libchaber, A. (1997). *Science* 278: 446–449.

124 Liu, Q., Wang, L., Frutos, A.G. et al. (2000). *Nature* 403: 175–179.

125 Sakamoto, K., Gouzu, H., Komiya, K. et al. (2000). *Science* 288: 1223–1226.

126 Siuti, P., Yazbek, J., and Lu, T.K. (2013). *Nat. Biotechnol.* 31: 448–452.

127 Chatterjee, G., Dalchau, N., Muscat, R.A. et al. (2017). *Nat. Nanotechnol.* 12: 920–927.

128 Chao, J., Wang, J., Wang, F. et al. (2018). *Nat. Mater.* 18: 273–279.

129 Li, J., Johnson-Buck, A., Yang, Y.R. et al. (2018). *Nat. Nanotechnol.* 13: 723–729.

130 Thubagere, A.J., Li, W., Johnson, R.F. et al. (2017). *Science* 357: eaan6558.

131 Weizmann, Y., Elnathan, R., Lioubashevski, O., and Willner, I. (2005). *J. Am. Chem. Soc.* 127: 12666–12672.

132 Bonnet, J., Yin, P., Ortiz, M.E. et al. (2013). *Science* 340: 599–603.

133 Kim, H., Bojar, D., and Fussenegger, M. (2019). *Proc. Natl. Acad. Sci. U.S.A.* 116: 7214–7219.

134 Qian, L. and Winfree, E. (2011). *J. R. Soc. Interface* 8: 1281–1297.

135 Genot, A.J., Baccouche, A., Sieskind, R. et al. (2016). *Nat. Chem.* 8: 760–767.

136 Song, T.Q., Garg, S., Mokhtar, R. et al. (2016). *ACS Synth. Biol.* 5: 898–912.

137 Vinkenborg, J.L., Karnowski, N., and Famulok, M. (2011). *Nat. Chem. Biol.* 7: 519–527.

138 Pardee, K., Green, A.A., Ferrante, T. et al. (2014). *Cell* 159: 940–954.

139 Wang, Z., Luo, Y., Xie, X. et al. (2018). *Angew. Chem. Int. Ed.* 57: 972–976.

140 Zhang, Z., Zeng, D.D., Ma, H.W. et al. (2010). *Small* 6: 1854–1858.

141 Subramanian, H.K.K., Chakraborty, B., Sha, R., and Seeman, N.C. (2011). *Nano Lett.* 11: 910–913.

142 Lin, C., Jungmann, R., Leifer, A.M. et al. (2012). *Nat. Chem.* 4: 832–839.

143 Goldman, N., Bertone, P., Chen, S. et al. (2013). *Nature* 494: 77–80.

144 Church, G.M., Gao, Y., and Kosuri, S. (2012). *Science* 337: 1628.

145 Boettiger, A.N., Bintu, B., Moffitt, J.R. et al. (2016). *Nature* 529: 418–422.

146 Edwardson, T.G.W., Lau, K.L., Bousmail, D. et al. (2016). *Nat. Chem.* 8: 162–170.

147 Farzadfard, F. and Lu, T.K. (2014). *Science* 346: 1256272.

148 Shipman, S.L., Nivala, J., Macklis, J.D., and Church, G.M. (2016). *Science* 353: aaf1175.

149 Shipman, S.L., Nivala, J., Macklis, J.D., and Church, G.M. (2017). *Nature* 547: 345–349.

150 Sheth, R.U., Yim, S.S., Wu, F.L., and Wang, H.H. (2017). *Science* 358: 1457–1461.

151 Kim, J., Lee, J., Hamada, S. et al. (2015). *Nat. Nanotechnol.* 10: 528–533.

152 Tam, D.Y. and Lo, P.K. (2015). *J. Nanomater.* 2015: 765492.

153 Chen, C.H., Huang, H., Ward, C.M. et al. (2007). *Science* 316: 597–600.

154 Kemmer, C., Gitzinger, M., Daoud-El Baba, M. et al. (2010). *Nat. Biotechnol.* 28: 355–360.

155 Ren, K., Liu, Y., Wu, J. et al. (2016). *Nat. Commun.* 7: 13580.

156 Keum, J.W. and Bermudez, H. (2009). *Chem. Commun.*: 7036–7038.

157 Hamblin, G.D., Carneiro, K.M.M., Fakhoury, J.F. et al. (2012). *J. Am. Chem. Soc.* 134: 2888–2891.

158 Mei, Q., Wei, X., Su, F. et al. (2011). *Nano Lett.* 11: 1477–1482.

159 Li, J., Pei, H., Zhu, B. et al. (2011). *ACS Nano* 5: 8783–8789.

160 Hahn, J., Wickham, S.F., Shih, W.M., and Perrault, S.D. (2014). *ACS Nano* 8: 8765–8775.

161 Jiang, D., Ge, Z., Im, H.-J. et al. (2018). *Nat. Biomed. Eng.* 2: 865–877.

162 Hu, L., Lu, C.H., and Willner, I. (2015). *Nano Lett.* 15: 2099–2103.

163 Cassinelli, V., Oberleitner, B., Sobotta, J. et al. (2015). *Angew. Chem. Int. Ed.* 54: 7795–7798.

164 Brglez, J., Nikolov, P., Angelin, A., and Niemeyer, C.M. (2015). *Chem. Eur. J.* 21: 9440–9446.

165 Cutler, J.I., Auyeung, E., and Mirkin, C.A. (2012). *J. Am. Chem. Soc.* 134: 1376–1391.

166 Yan, J., Hu, C., Wang, P. et al. (2015). *Angew. Chem. Int. Ed.* 54: 2431–2435.

167 Liu, X., Zhang, F., Jing, X. et al. (2018). *Nature* 559: 593–598.

168 Nguyen, L., Doblinger, M., Liedl, T., and Heuer-Jungemann, A. (2019). *Angew. Chem. Int. Ed.* 58: 912–916.

169 Sun, W.J., Jiang, T.Y., Lu, Y. et al. (2014). *J. Am. Chem. Soc.* 136: 14722–14725.

170 Lu, C.H. and Willner, I. (2015). *Angew. Chem. Int. Ed.* 54: 12212–12235.

171 Banerjee, A., Bhatia, D., Saminathan, A. et al. (2013). *Angew. Chem. Int. Ed.* 52: 6854–6857.

172 LeProust, E.M., Peck, B.J., Spirin, K. et al. (2010). *Nucleic Acids Res.* 38: 2522–2540.

173 Kosuri, S. and Church, G.M. (2014). *Nat. Methods* 11: 499–507.

174 Schmidt, T.L., Beliveau, B.J., Uca, Y.O. et al. (2015). *Nat. Commun.* 6: 8634.

175 Zhao, Z., Liu, Y., and Yan, H. (2011). *Nano Lett.* 11: 2997–3002.

176 Nelissen, F.H., Leunissen, E.H., van de Laar, L. et al. (2012). *Nucleic Acids Res.* 40: e102.

177 Ponchon, L. and Dardel, F. (2007). *Nat. Methods* 4: 571–576.

178 Suzuki, H., Ando, T., Umekage, S. et al. (2010). *Appl. Environ. Microbiol.* 76: 786–793.

179 Chan, M.S. and Lo, P.K. (2014). *Small* 10: 1255–1260.

180 Patel, P.C., Giljohann, D.A., Seferos, D.S., and Mirkin, C.A. (2008). *Proc. Natl. Acad. Sci. U.S.A.* 105: 17222–17226.

181 Crawford, R., Erben, C.M., Periz, J. et al. (2013). *Angew. Chem. Int. Ed.* 52: 2284–2288.

182 Nakata, E., Liew, F.F., Uwatoko, C. et al. (2012). *Angew. Chem. Int. Ed.* 51: 2421–2424.

183 Tamsir, A., Tabor, J.J., and Voigt, C.A. (2011). *Nature* 469: 212–215.

184 Gantz, V.M. and Bier, E. (2015). *Science* 348: 442–444.

185 Niu, Y., Shen, B., Cui, Y. et al. (2014). *Cell* 156: 836–843.

186 Sun, W.J., Ji, W.Y., Hall, J.M. et al. (2015). *Angew. Chem. Int. Ed.* 54: 12029–12033.

187 Jinek, M., Chylinski, K., Fonfara, I. et al. (2012). *Science* 337: 816–821.

188 Adleman, L.M. (1994). *Science* 266: 1021–1024.

189 Stojanovic, M.N., Semova, S., Kolpashchikov, D. et al. (2005). *J. Am. Chem. Soc.* 127: 6914–6915.

190 Stojanovic, M.N., Mitchell, T.E., and Stefanovic, D. (2002). *J. Am. Chem. Soc.* 124: 3555–3561.

7

Engineering DNA Switches for DNA Computing Applications

Dominic Lauzon[1], Guichi Zhu[2], and Alexis Vallée-Bélisle[1,2]

[1]*Université de Montréal, Laboratory of Biosensors & Nanomachines, Département de Chimie, 1375, Ave Thérèse-Lavoie-Roux Montréal, Montréal, QC H2V 0B3, Canada*
[2]*Université de Montréal, Institut de Génie Biomédical, Département de Pharmacologie et Physiologie, 2900, boul Édouard-Montpetit, Montréal, QC H3T 1J4, Canada*

7.1 Introduction

Through billions of years of evolution, living organisms have developed a myriad of finely tuned nanomachines to monitor changes in their environment. In order to respond to these changes, or input, cells rely on biomolecules that undergo structural changes in the presence of specific chemical or physical inputs (e.g. temperature, pH, small molecules, proteins and other macromolecules, or even viruses and bacteria) [1]. Upon activation, these structure-switching molecules can then trigger one or multiple output mechanisms to efficiently react to the perturbation previously detected. It can be, for example, by changing the function of other biomolecules, by increasing/decreasing gene expression, by opening/closing transmembrane proteins, or by triggering the self-assembly of biomolecules [2, 3]. Examples of natural structure-switching molecules, also called biomolecular switches, include the regulating protein calmodulin that changes its structure and activity following calcium binding (four Ca^{2+} binding sites), which transduces a change in cell function by regulating downstream effectors [4, 5]. Another example of biomolecular switches are the G protein-coupled receptors, a membrane protein family containing over 800 identified members, that control cellular fate via binding-induced structural variation triggers by various chemical inputs (light, odorant molecules, hormones, etc.) [6, 7].

Inspired by the efficiency of natural switches, chemists and engineers have begun to synthesize molecular systems that take advantage of these nanoscale switching mechanisms. For example, some have created switches using light-, binding-, and current-induced structural changes [1, 8–10]. One superstar molecule to create structure-switching nanosystems is DNA. This is due to its high programmability (i.e. folding and binding energies) combined with its
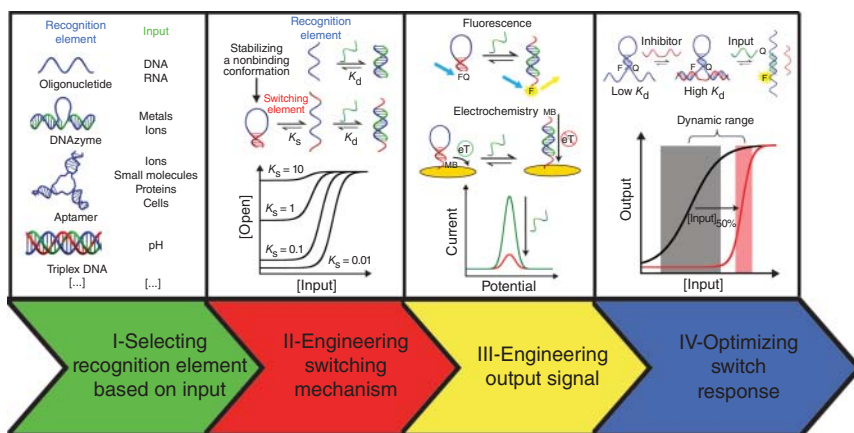
Figure 7.1 Engineering steps of DNA switches. (I) DNA can adopt a wide range of nanostructures acting as recognition elements for specific chemical input. (II) The selected recognition element must, therefore, be converted into a structure-switching nanosystem by stabilizing a nonbinding conformation. (III) Analytical readouts, such as fluorescence or current, can be introduced to record the input-induced structure switching. (IV) Different strategies exist to optimize the dose–response profile of a switch to better suit the desired application. Source: From Harroun et al. [1]. Reproduced with the permission of Royal Society of Chemistry.

ease of synthesis and relative ease of chemical conjugation to a wide range of molecules and nanomaterials [1]. DNA, for example, can specifically bind to its complementary sequence and can also fold into a wide range of nanostructures (e.g. i-motif, G-quadruplex, aptamers, DNA triplex, hairpin, etc.) that can selectively interact with other class of chemical species (e.g. ions, small molecules, proteins, etc.) [8, 9]. Given these features, DNA switches have thus found many applications in DNA computing enabling, for example, the creation of molecular automaton that can play games [11], perform square root calculations [12], function as a security system [13], and perform cancer theranostics [13, 14] as well as molecular diagnostics [15].

To rationally develop DNA switches into logic circuits, one must first think about the inputs that will trigger structure switching and then choose recognition elements accordingly (Figure 7.1I). Once chosen, the recognition elements must be converted into structure-switching molecules by stabilizing a nonbinding conformation (Figure 7.1II). In this design phase, one should also consider how different recognition elements can be combined to obtain a switch that responds to more than one input molecule. Then, an output function must be introduced to enable an analytical readout of the switch (Figure 7.1III). Finally, and only in some cases, the switching behavior is not always optimal for the desired application and thus needs to be optimized to better suit it (Figure 7.1IV). In this chapter, we summarize the major steps and considerations required to create DNA switches, from scratch, and we further discuss the rationale behind the design and creation of DNA computing systems based on DNA switches [1, 16].

7.2   Selecting Recognition Element Based on Input

The initial step of designing DNA switches for DNA computing applications is to select an appropriate recognition element for the desired input species (Figure 7.1I). Inputs are typically classified into three categories, including physical phenomena (e.g. temperature and light), chemical stimuli (e.g. protons, metal and nonmetal ions, small molecules, nucleic acids, and proteins), and biological units (e.g. viruses, cells, and bacteria). Through programming the stability of a DNA fold by its length or GC/AT base pair composition, for example, one can create a variety of temperature-induced DNA switches that can be activated at various specific temperatures [17–19]. Light-sensitive DNA switches that employ chemically modified DNA strands have also been explored [20–22]. As is well known, DNA can selectively bind its complementary sequence, which has led to the development of fluorescence-producing structure-switching molecular beacons by Kramer and Tyagi in 1996 [23]. Some DNA recognition elements can also selectively bind non-nucleic acid molecules. For example, a triplex DNA strand has been designed through both Watson–Crick and Hoogsteen base pairing interactions to determine a solution's pH value [24]. The i-motif is another noncanonical DNA structure that is stabilized under acidic pH conditions due to the protonation of cytosine and can thus serve as a pH sensor [25]. DNA G-quadruplex structures can be formed or stabilized in the presence of potassium ions (K^+) [26], while mercury ions (Hg^{2+}) [27] and silver ions (Ag^+) [28] stabilize DNA conformations containing thymine–thymine and cytosine–cytosine base pair mismatches, respectively. DNAzymes are another widely used recognition element for metal ions. These latter can often act as specific cofactors to catalyze the cleavage of nucleic acid substrate strands (e.g. Mg^{2+} [29], Pb^{2+} [30], and UO_2^{2+} [31]). Short single-stranded DNA (ssDNA) or RNA sequences can also be selected *in vitro* by the systematic evolution of ligands by exponential enrichment (SELEX) [32]. These sequences enable the binding of small molecules and proteins with a typical high affinity and specificity (see aptamers) [33]. Finally, the biological units of viruses, cells, and bacteria also can be recognized by their respective aptamer sequences that specifically bind to the viral proteins and cell membrane proteins [34–37].

For DNA computing applications, the recognition element often needs to process multiple inputs. To do so, the DNA recognition elements must interact with two (or more) inputs that are often chemically distinct. Sometimes, such DNA strands already exist. This is the case, for example, of one thrombin aptamer that requires the presence of K^+ to fold into a G-quadruplex in order to bind the thrombin protein [26]. Therefore, this DNA recognition element can be used to sense both K^+ and thrombin. However, this exception is somewhat peculiar, as most of the time the design of multiple input recognition elements must be engineered from scratch by combining two (or more) elements together. For example, an aptamer sequence was introduced into the loop section of a clamp-like triplex DNA, thereby rendering the binding of the aptamer with its input pH dependent [38]. Similarly, an aptamer sequence was introduced into the arm section of a DNAzyme, thus making the switch sensitive toward both the aptamer's input and the metallic cofactor [39]. Another strategy is to fuse together two relevant DNA

recognition elements into a stem-loop, which renders its opening sensitive to the presence of both inputs simultaneously [40]. Overall, selection of the right recognition element for detection of multiple inputs becomes only limited by one's creativity to merge various DNA recognition elements into a broader nanosystem.

7.3   Engineering Switching Mechanisms

Efficient signaling of artificial nanoswitches is often related to their capacity to undergo large conformational changes upon binding of the desired input (Figure 7.1II). For example, fluorescence- or electrochemical-producing switches generally require large conformational change in order to generate high output signaling (e.g. Figure 7.1III). Some DNA nanostructures already spontaneously undergo large conformational changes upon binding. For example, the i-motif undergoes large conformational changes upon protonation of cytosine, going from a nonstructured random coil conformation to a well-defined intercalated nanostructure [25, 41]. DNA input molecules will also drastically affect the structure of their host DNA receptors upon binding by triggering a structure change from a flexible unfolded ssDNA conformation to a more rigid double helix conformation. On the other hand, many DNA structures offer limited structure-switching behavior upon binding to their input molecules. This is often the case with aptamers, where the screening effort via SELEX does not consider structural motifs in the selection process and thus typically leads to DNA sequences that are more stable in their binding-competent state [42, 43]. To overcome this limitation, strategies have been developed to introduce (or enhance) conformational changes upon binding of the input [44]. These strategies mostly rely on the population-shift mechanism, which involves the stabilization of a nonbinding conformation to improve the magnitude of conformational changes [5]. This mechanism is typically thought to proceed through a three-state equilibrium that involves a first switching equilibrium between the nonbinding and the binding-competent states along with the binding equilibrium of the input that can only interact with the binding-competent state (Figure 7.2a). The presence of the input will thus trigger the switching of the DNA by shifting its equilibrium toward the bound state through the gain of new favorable interactions between the input molecule and the DNA.

   To ensure good switching, the nonbinding state must remain the most favorable conformation of the DNA sequence in the absence of the input (i.e. lowest energy state). Designing such switching systems thus requires a good understanding of the folding free energy (ΔG) of both the nonbinding and binding-competent states. In order to do so, websites like NUPACK [45], Mfold [46], and IDT SciTools [47] enable user-friendly estimation of the folding free energy of DNA secondary structures based on Watson–Crick interactions. Unfortunately, such websites or software do not yet exist for more complex DNA tertiary structures such as G-quadruplexes, i-motifs, and aptamers. Therefore, the design of these systems mostly relies on experimental characterization of their free energy. Validation and characterization of switching (or binding)
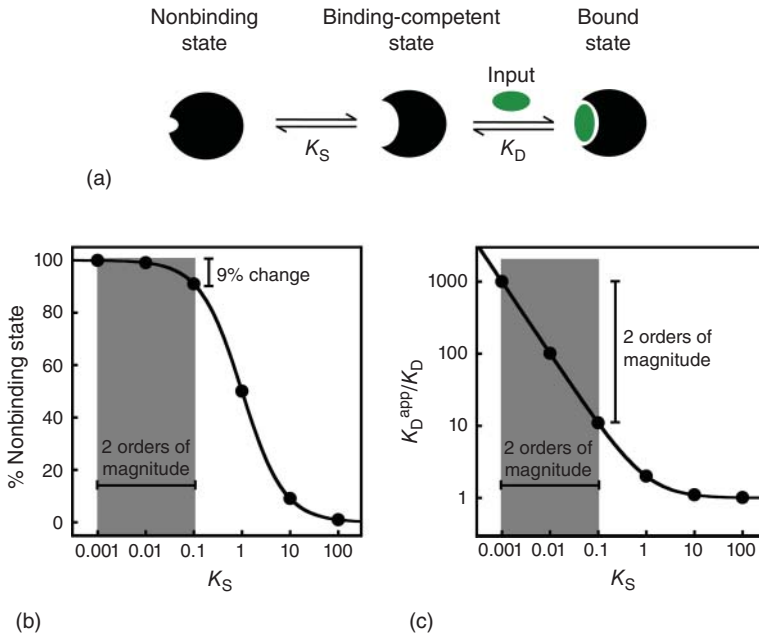
Figure 7.2 (a) Cartoon representation of the population-shift mechanism. The switch is in equilibrium between the nonbinding and binding-competent states (K_S), and this equilibrium is shifted toward the bound state (K_D) upon addition of the input. To ensure a large population of DNA in the nonbinding state without drastically altering the apparent binding affinity of our input (K_D^{app}), K_S must remain lower than 0.1 because (b) high K_S does not provide enough population of DNA in the nonbinding state, thus ultimately leading to an insufficient population shift to be accurately monitored, while (c) a lower K_S will result in a drastic energetic penalty for the binding of the input (K_D^{app}).

free energy can be easily measured either through urea [48] or temperature [49] denaturation curves. To achieve good switching behavior, the equilibrium switching constant (K_S = [binding-competent state]/[nonbinding state]), which is related to the free energy (ΔG_S) through Eq. (7.1), must remain below 0.1. This ensures a low background with a large signal change because at least 90% of the DNA will, in the presence of its input, switch from the nonbinding state to the binding-competent state (Figure 7.1II). One must keep in mind that employing a $K_S \gg 0.1$ leads to a system wherein the population of DNA in the nonbinding state is too low, thus ultimately leading to not enough switches remaining to generate a population shift large enough to be accurately monitored. In contrast, over-stabilizing the nonbinding conformation via a $K_S \ll 0.1$ will increase the concentration of input needed to trigger the switching (K_D^{app}) relative to the intrinsic affinity between the input and the DNA recognition element (K_D) (see Eq. (7.2)) [5]. To illustrate this relationship more quantitatively, a switch with a K_S of 0.1 provides maximal change in population of 90.9% with only a 10-fold penalty in observed affinity, whereas a K_S of 0.001 only improves the population shift by 9% (90.9% vs. 99.9%) while drastically increasing the energetic penalty of binding by 2 orders of magnitude (10-fold vs. 1000-fold) (Figure 7.2b,c). As we

will see in step IV (optimizing switch response), optimizing of the switching constant K_S can also be used to optimize the switch response within a specific input concentration range.

$$\Delta G_S = -RT \ln(K_S) \tag{7.1}$$

$$K_D{}^{\text{app}} = K_D \frac{(1 + K_S)}{K_S} \tag{7.2}$$

Many strategies exist to stabilize a DNA recognition motif into a nonbinding state (Figure 7.3). A widely used strategy is to modify the DNA recognition element into a stem-loop (Figure 7.3a) [50]. This can be achieved by introducing two short complementary sequences at the 3′ and 5′ extremities that constrain the entire DNA sequence to adopt a stem-loop conformation different from the one it adopts when bound to the input molecule (i.e. the linear conformation adopted in a duplex DNA vs. the specific tertiary structure of an aptamer sequence). The new interactions between the input and the recognition element
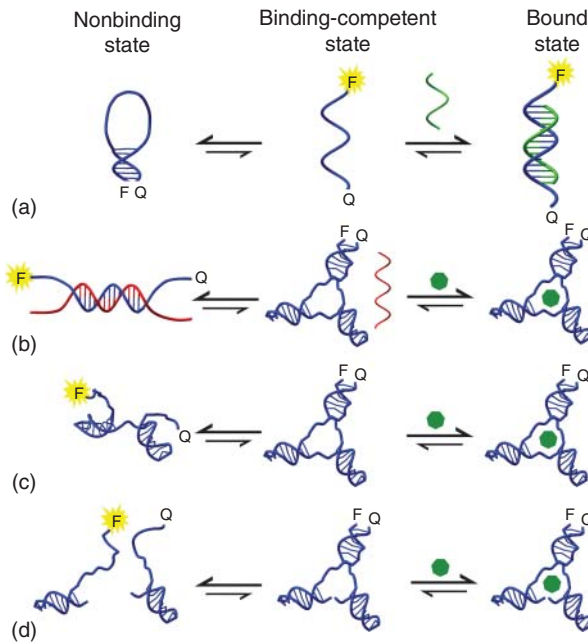
Figure 7.3 Based on the population-shift model, different strategies exist to stabilize a DNA recognition element into a nonbinding conformation. (a) Short complementary sequence can be added at the 5′ and 3′ extremities of the recognition element, which will bend it into a molecular beacon. (b) A complementary sequence of our recognition element (in red) can be added to promote the formation of a DNA duplex. (c) Nucleotides can be mutated or deleted to disrupt interactions present in the binding-competent state, thus rendering the nonbinding state more favorable. (d) Splitting the recognition element in half also destabilizes the binding-competent state. Source: From Harroun et al. [1]. Reproduced with the permission of Royal Society of Chemistry.

will therefore act as the driving force to disrupt the Watson–Crick base pairs formed in the stem. A second convenient strategy to create a structure-switching mechanism consists of inserting a DNA strand (red) that is complementary to the DNA recognition element (Figure 7.3b). This duplex can still sample the binding-competent state and can therefore be displaced toward the binding state by the input. A third strategy involves a mutational (or deletion) method, where some nucleotides are changed (or removed) in order to destabilize the binding-competent state, thus rendering the nonbinding state more favorable (Figure 7.3c). Of course, mutation (or deletion) should not be performed with nucleotides that are known to be important for the selectivity and specificity of the recognition element. A good understanding of the secondary structure of the DNA recognition element is thus required to avoid any perturbation of the binding surface. A fourth strategy requires one to split the DNA recognition element into two DNA sequences that will be brought together upon addition of the input (Figure 7.3d). As with the mutational/deletion strategy, splitting should be avoided in regions that are relevant for the binding of the input. Breaking the phosphodiester bond at such relevant positions may disrupt interactions that cannot be retrieved when dimerization is triggered by the input.

As mentioned previously, DNA computing typically requires the processing of information obtained from multiple inputs. To access information based on two molecular inputs, the population-shift model can be readily adapted to consider the effect of an allosteric effector on the affinity between the switch and the initial input (Figure 7.4a). We have previously described [51] how DNA allosteric effectors can be readily designed to stabilize/destabilize the binding-competent state (or nonbinding state) of a switch. Allosteric activation happens when an effector molecule binds and stabilizes the switch into the binding-competent state, thus increasing K_S (Figure 7.4a, top). This makes it easier for the input to bind the switch and reduces the midpoint (K_D^{input}) toward a lower concentration of input (Figure 7.4b). This behavior is well modeled by Eq. (7.3), where K_S is the switching constant, K_D is the dissociation constant between the input and the switch, K_A is the dissociation constant between the activator and the switch, $[A]$ is the concentration of activator, and α is the ratio of dissociation constants in the presence and absence of activator [51]. It is important to note that the midpoint can only be shifted until a certain threshold defined by α. In other words, further addition of an activator will not push the midpoint toward a lower concentration, but rather, it will limit it to a specific threshold as defined by the affinity of the input for a switch fully bound by the activator. Therefore, one must optimize the activator to enable a change in input affinity that is large enough to create a measurable change in the output signal. Likewise, allosteric inhibition happens when the effector molecule binds and stabilizes the nonbinding state, thus reducing K_S (Figure 7.4a, bottom). This makes it harder for the input to bind, as it increases the energetic penalty of binding, thus increasing the midpoint (K_D^{input}) toward a higher concentration of input (Figure 7.4c). This behavior is well modeled by Eq. (7.4), where K_I is the dissociation constant between the inhibitor and the switch and $[I]$ is the concentration of inhibitor [51]. In this case, no threshold is observed, as a higher concentration of inhibitor always leads to a higher midpoint.
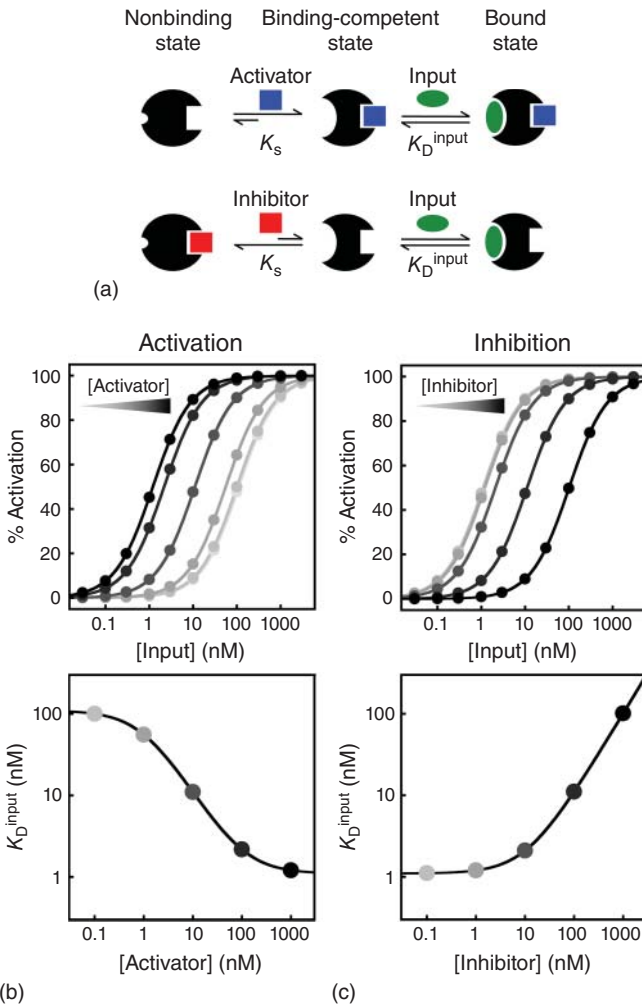
Figure 7.4 Creating switches based on the population-shift model for double input detection. (a) The population-shift model can readily be adapted to consider the effect of a new input called an effector (shown as a square). We call the effector an activator when the molecule (blue square) stabilizes the binding-competent state, and conversely, we call it an inhibitor when the molecule (red square) stabilizes the nonbinding state. (b) The presence of an activator increases the affinity of the switch for our initial input until a certain threshold defined by the affinity of that input for a switch fully bound by the activator. (c) On the other hand, the presence of an inhibitor reduces the affinity of the switch for the initial input by increasing the energetic penalty of binding related to lowering K_S.

$$K_D{}^{input} = K_D \left(\frac{1 + K_S}{K_S} \right) \left(\frac{1 + \frac{[A]}{K_A}}{1 + \frac{[A]}{\alpha K_A}} \right) \tag{7.3}$$

$$K_{\mathrm{D}}{}^{\mathrm{input}} = K_{\mathrm{D}} \left(\frac{1 + K_{\mathrm{S}} \left(\frac{K_{\mathrm{I}}}{K_{\mathrm{I}}+[I]} \right)}{K_{\mathrm{S}} \left(\frac{K_{\mathrm{I}}}{K_{\mathrm{I}}+[I]} \right)} \right) \qquad (7.4)$$

Allosteric inhibition and activation mechanisms enable the introduction of a second variable (i.e. the effector) that renders simple two-variable Boolean operations possible. Using allosteric activation, for example, one can easily produce an OR gate, where the presence of an activator and/or the input can stabilize the switch into the binding-competent state (Figure 7.5a). In this scenario, if one wants to detect a specific concentration of at least one molecule (input and/or activator), it is imperative that each molecule have their concentration higher than their respective dissociation constant (K_{A} < [activator], $K_{\mathrm{D}}{}^{\mathrm{input}}$ < [Input]). This ensures that a good shift in population happens when at least one of the two molecules is present. This strategy has recently been employed to activate the catalytic activity of a DNAzyme (Figure 7.5b) [52]. Here, the DNAzyme has been transformed into a switch by incorporating a DNA strand that sequesters the catalytic loop, thus preventing its activity. Two input DNA strands were also designed to be complementary to the sequestering DNA strand. Therefore, the activity of the DNAzyme can be recovered by adding either one of the inputs, both of which cause the displacement of the sequestering DNA strand and the correct folding of the DNAzyme. This highlights how the switch design can easily be combined with allosteric effectors to enable the creation of well-controlled logic gate.

Another logic gate that can be created using an allosteric activator is the AND gate (Figure 7.5c). This gate provides an output signal only when all inputs are present. In contrast with the OR gate, each molecule must have their concentration lower than their respective dissociation constant (K_{A} > [activator], $K_{\mathrm{D}}{}^{\mathrm{input}}$ > [Input]). This ensures that both molecules cannot individually activate the switch and that the presence of each is required to stabilize the switch into the binding-competent state. For that to happen, one must program the concentration of activator such that it does not, alone, significantly shift the population of switches toward the binding-competent state ([activator] < K_{A}), but still enhances the binding of the input significantly. This strategy has been incorporated into a DNAzyme by intentionally mutating one arm of the DNAzyme switch (blue arm) such that the substrate cannot efficiently bind (Figure 7.5d) [53]. In this case, a DNA effector strand (blue strand) was also rationally introduced to bind to the mutated section and allow the recovery of the missing Watson–Crick base pairs for the substrate, thus enhancing the affinity for the substrate. The DNAzyme is, therefore, only active when both inputs (effector and substrate) are present. Here, we can appreciate how the mutation strategy was exploited to create a logic gate by simply using one input to enable the recovery of the native interactions, therefore favoring the binding of the second input.

Finally, using an allosteric inhibiter, INHIBIT gates become readily achievable (Figure 7.5e). Such gates only provide an output signal when one input of interest is present alone. Therefore, the affinity of the input must be lower than the concentration that it is intended to detect ($K_{\mathrm{D}}{}^{\mathrm{input}}$ < [Input]) in order to provide
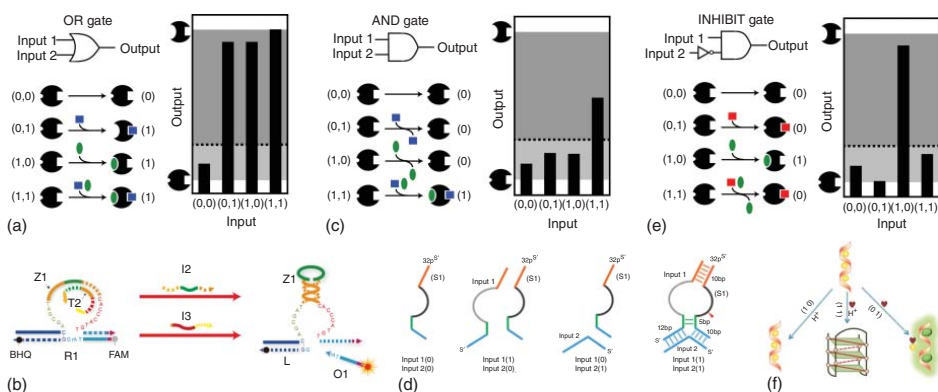
Figure 7.5 Exploiting allosteric effectors to create logic gates. (a) Using allosteric activation, one can produce an OR gate where the presence of an activator (blue square) and/or the input (green circle) can stabilize the switch into the binding-competent state. (b) This strategy has been exploited using a DNAzyme by adding a sequestering strand (T2) into the catalytic loop. The activity can be retrieved by the addition of either one of the inputs (I2 and I3), which are both complementary to the sequestering strand. (c) Allosteric activation can also be used to create an AND gate if the activator (blue square), when bound, does not significantly increase the affinity for the input (green circle). (d) This strategy has been adapted with a DNAzyme by intentionally mutating one arm of the DNAzyme switch (blue arm). Therefore, the presence of both DNA strands is required to reform the DNAzyme and retrieve its activity. (e) Allosteric inhibition can easily be adapted to create an INHIBIT gate. In this case, an allosteric inhibitor (red square) stabilizes the switch into the nonbinding state. Therefore, only the input alone (green circle) can activate the switch and produce a signal. (f) This strategy was exploited using a silver-deposited DNA that can detect the presence of cysteine (heart shape) by removing the silver deposition (yellow circle) from the DNA scaffold through the formation of Ag—S bonds. However, the DNA recognition element was also built to respond to a change of H^+, the allosteric inhibitor, when the silver depositions are removed by the cysteine. It does so by stabilizing the switch into a nonbinding state (i.e. a i-motif), thus rendering the switch only active in the presence of cysteine alone. Data from panel a, c, and e are simulated using the population-shift model with Eqs. (7.3) and (7.4). Source: (Panel b) From Zheng et al. [52]. Reproduced with the permission of Oxford University Press; (Panel d) From Furukawa and Minakawa [53]. Reproduced with the permission of Royal Society of Chemistry; (Panel f) From Gao et al. [54]. Reproduced with the permission of Royal Society of Chemistry.

a good change in population for that input. Also, the inhibitor must be programmed such that its presence prevents the binding between the input and the switch by favoring the nonbinding state. This can be done by using a concentration of inhibitor higher than its K_I and that is also large enough to significantly decrease the affinity of the input. This strategy has been used many times by simply introducing an inhibitor molecule (e.g. DNA strand or small molecule) that either favors the formation of the nonbinding state of the switch or prevents the analytical readout when bound to the switch (Figure 7.5f) [53–55]. For example, a switch made using the RET proto-oncogene and silver deposition was used to detect the presence of cysteine by removing the silver deposition from the DNA scaffold through the formation of Ag—S bonds [54]. The DNA is then free to bind thioflavin T, which results in a fluorescent signal. However, the addition of protons, the inhibitor, into the mixture favors the folding of the DNA into an i-motif, thus preventing its detection by thioflavin T.

Many examples of these different logic gates have been developed in recent years. Pei et al., for example, have developed a DNA tetrahedron with logic response by incorporating dynamic sequences into the edges of the nanostructure (Figure 7.6a) [56]. Here, a small hairpin containing a 5′-CCGC-3′/5′-GCGG-3′ stem ($\Delta G = -2.1\,\text{kcal}\,\text{mol}^{-1}$ at 37 °C using Mfold, $K_S = 0.033$) was introduced into one edge, providing an expected 97% population shift with only a 30-fold penalty in observed affinity. Each input was engineered to bind half of the loop and invade a section of the stem. In order to correctly fold the edge of the tetrahedron, the presence of both inputs (AND gate) is required to compensate the lost of stability caused by the disruption of the stem. This versatile switching DNA tetrahedron was further adapted for the detection of intracellular ATP in living cells by incorporating an ATP aptamer in one of the edges. Our second example illustrates how switching thermodynamics can have a huge impact on the activity of a switch. Here, Zhang et al. have created a nano-assembly containing a multi-hairpin motif engineered to produce logic response in the presence of microRNAs (Figure 7.6b) [57]. The microRNAs cannot bind with one of their DNA construct (called L0) because it contains hairpins that were too stable (ΔG of $-6.2\,\text{kcal}\,\text{mol}^{-1}$, $K_S = 4 \times 10^{-5}$ and $-8.4\,\text{kcal}\,\text{mol}^{-1}$, $K_S = 1 \times 10^{-6}$) and therefore drastically increases the energetic penalty of binding by at least 5 orders of magnitude. A second construct (called L3) with less stable hairpins (ΔG of $-3.4\,\text{kcal}\,\text{mol}^{-1}$, $K_S = 4 \times 10^{-3}$ and $-5.2\,\text{kcal}\,\text{mol}^{-1}$, $K_S = 2 \times 10^{-4}$) has shown promising results by producing a logic response in the presence of three different microRNAs. This study highlights the importance of engineering switches with optimal thermodynamics. In our last example, Chen and Zeng used the sequestration and splitting strategy to adapt an ATP aptamer and a thrombin aptamer into many different logic switches [58]. The first design introduces a signaling DNA strand that is complementary to both aptamers' sequences (Figure 7.6c). Therefore, the presence of any of the inputs triggers the displacement of that signaling DNA duplex, thus leading to a NOR gate. This same strategy can be used to create a NAND gate by using two signaling DNA strands that will individually sequester each aptamer (Figure 7.6d). Also, the splitting strategy was also used to create a recognition element and a signaling strand that each contain both sections of the ATP and thrombin aptamers. This
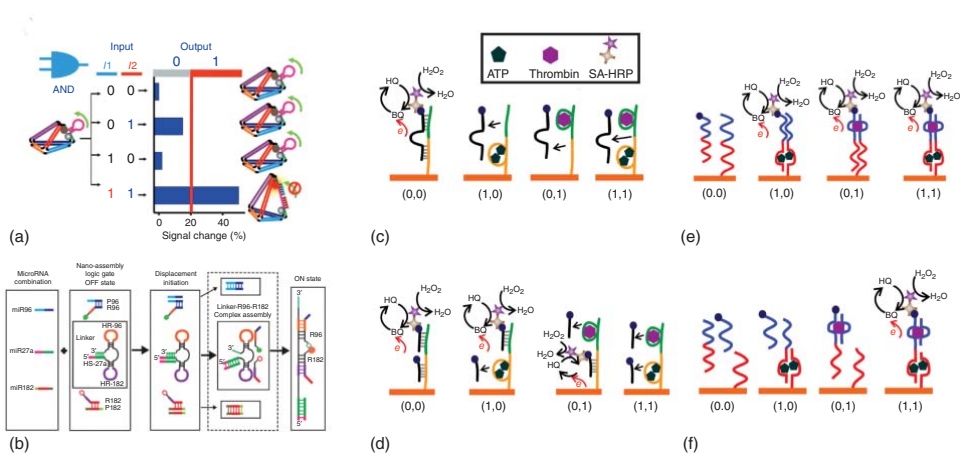
Figure 7.6 Examples of logic gate created using the strategies discussed in this chapter. (a) Here, a small stem-loop containing 4GC base pairs was introduced at one edge of a tetrahedron to make its opening dependent on the binding of two inputs that bind each half of the loop. (b) A microRNA logic gate was built using a multi-hairpin motif. (c) A new recognition element was built by fusing an ATP aptamer and a thrombin aptamer together. A signaling strand was then designed to be complementary to that new recognition element. Therefore, the presence of any input will displace the signaling strand, thus resulting in a NOR gate. (d) Cutting that same signaling strand in two halves creates two smaller signaling strands that can individually bind to each aptamer's section. Therefore, each signaling strand can be displaced by their respective inputs, thus creating a NAND gate. (e) The splitting strategy was also used to create a recognition element and a signaling strand that each contain both sections of the ATP and thrombin aptamers. This leads to an OR gate, where the presence of at least one input is required to bring both strands together. (f) This strategy was rendered more complex by splitting the system into three different strands, where the presence of both inputs is required to bring all strands together, thus creating an AND gate. Source: (Panel a) From Pei et al. [56]. Reproduced with the permission of John Wiley & Sons; (Panel b) From Zhang et al. [57]. Reproduced with the permission of Royal Society of Chemistry; (Panels c–f) From Chen and Zeng [58]. Reproduced with the permission of Elsevier.

leads to the creation of an OR gate, where the presence of at least one input is required to pay the energetic price of bringing back together the two split sections of the aptamers (Figure 7.6e). This splitting strategy has also been used to create an AND gate by splitting the DNA sequence into three components, where the presence of both inputs is required to pay the energetic price (Figure 7.6f).

7.4 Engineering Logic Output Function Response

Selection of effective logic output mechanism plays a critical role in DNA switch design (Figure 7.1III). Fortunately, it is convenient to engineer the output signal for DNA switches due to the general ease and simplicity of chemically labeling DNA strands with reporter molecules. Among the various types of output signals, fluorescence spectroscopy and electrochemical analysis are two most widely employed methods used for DNA switches in DNA computing applications. Fluorescence possesses the advantages of high sensitivity, homogeneous assays, excellent reproducibility, and easy operation [59]. For example, by virtue of distance change-induced Förster resonance energy transfer (FRET), a variety of fluorescent methods have been developed to record the conformational changes of DNA switches [60]. Typically, a fluorophore and a quencher are added at the extremity of stem-loop DNA switches (5′ and 3′ terminals – see, for example, the molecular beacon [23]), and a very low fluorescence signal is obtained in the absence of a DNA input due to the significant energy transfer between the fluorophore and quencher. Upon addition of a DNA sequence complementary to the loop sequence, the stem structure is opened, and a strong fluorescence signal is generated due to the separation of the fluorophore and quencher. Alternatively, nanomaterials including gold nanoparticles [61] and graphene oxides [62] can also be employed as a quencher in DNA switches. Moreover, nucleotide analogs with specific fluorescent properties, such as 2-aminopurine [63], can also be added into the DNA strand as a fluorescent reporter for the study of DNA structure and dynamics [64]. In comparison to the classic fluorophore/quencher pairs, 2-aminopurine is less susceptible to photobleaching because its excitation wavelength is outside of the visible light range [65]. In order to decrease the synthesis cost of chemically modified fluorescent reporters, label-free methods (e.g. fluorescent intercalators) have also been explored for DNA switches [66].

Electrochemical analysis is the other commonly used output signal to monitor the conformational change of DNA switches. Typical electrochemical techniques include cyclic voltammetry (CV), alternative current voltammetry (ACV), differential pulse voltammetry (DPV), and square wave voltammetry (SWV) [67]. These electrochemical methods have attracted increasing attention due to their high sensitivity, good specificity, low cost, and especially insusceptibility to the matrix effects of biological samples [68]. The electrochemical output signal depends on the specific binding between the DNA switches and the specific input molecules in order to conduct electron transfer at the electrode surface, thereby generating an electrochemical output signal. Since 2003, Plaxco's lab

has described many strategies based on the binding-induced conformational changes of DNA switches in an electrochemical format [69, 70]. As a proof of principle, a redox-labeled DNA switch is first immobilized on a gold electrode surface through a Au—S bond to form a stem-loop structure [69], which is analogous to the fluorescent molecular beacon reported by Kramer [23]. In the absence of target DNA, the stem-loop structure of the DNA switch brings the redox element into close proximity with the electrode surface and generates a high electrochemical current. However, in the presence of target DNA, the stem-loop structure is opened via the complementary hybridization between the DNA switch and input DNA, which leads to a significant decrease of the electrochemical current since the redox element is pushed away from the surface. Small molecules and proteins are also used as input molecules to trigger the conformational change of DNA switches on the electrode surface [71–74]. Various logic gates have also been designed using similar electrochemical strategies [75]. Of note, label-free electrochemical methods have attracted attention due to their inherent simplicity and low cost. For example, electrochemical impedance spectroscopy (EIS) is a promising label-free strategy that measures electron transfer resistance change between ssDNA and dsDNA (or aptamer and target–aptamer complex) in the presence of a redox reporter such as ferricyanide [76]. Typically, an increase in electron transfer resistance is observed in dsDNA [77] (or target–aptamer complex [78–80]) due to the stronger negative charge repulsion between ferricyanide and dsDNA (or target–aptamer complex). Besides fluorescence spectroscopy and electrochemical analysis, colorimetry [81], chemiluminescence [82], and surface-enhanced Raman scattering (SERS) [83] have also been explored as output signals in DNA switches.

7.5 Optimizing Switch Response

Introducing switching behavior to the chosen DNA recognition element does not always lead to switches that are functional within the desired input concentration range. For this reason, the dose–response profile of the switch must be optimized to obtain a relevant dynamic range (Figure 7.1IV) [16]. Two simple parameters can be used to describe the response profile of a switch, namely, (i) the midpoint and (ii) the dynamic range. The midpoint characterizes the concentration of input needed to produce a 50% change in output signal ($[\text{Input}]_{50\%}$, Eq. (7.5)). This parameter is generally referred as the apparent dissociation constant (K_D^{app}, or simply K_D) because it represents the inflection point of the sigmoidal dose–response curve. The simplest method to tune the midpoint of a switch has already been discussed previously (Section 7.3) and involves the stabilization of the nonbinding conformation consistent with the population-shift mechanism (see Eq. (7.2)).

$$\text{Midpoint} = K_D^{\text{app}} = [\text{Input}]_{50\%} \tag{7.5}$$

$$\text{DR} = \frac{[\text{Input}]_{90\%}}{[\text{Input}]_{10\%}} = 81^{(1/n_H)} \tag{7.6}$$

The dynamic range (DR), also referred as the sensitivity, defines the range of input concentrations needed to produce structure switching. It can be numerically assessed by the ratio of concentrations that give 90% and 10% output signal (DR = [Input]$_{90\%}$/[Input]$_{10\%}$). It can also be assessed through the empirical Hill factor (n_{H}), which is linked with the DR by Eq. (7.6) [84]. Typical dose–response curves have a dynamic range of 81-fold ($n_{\mathrm{H}} = 1$). A simple strategy to extend the dynamic range is to use multiple switches that possess differing midpoints (Figure 7.7a, top). To achieve that, a good understanding of the input/output response of each individual switch is required to determine the optimal ratio of switches to build a switch system providing an optimal "linear" extended dynamic range. For example, by mixing in an equimolar ratio of three switches with midpoints at 1, 10, and 100 nM, respectively, the dynamic range can be extended by almost an order of magnitude from 81-fold to 791-fold without affecting the linearity of the response (Figure 7.7a, bottom). An easy method to obtain multiple variants of the same switch is to introduce mutations or by deleting some of the nucleotides. This creates switches that will have lower affinity compared with the native switch. Using that strategy, the dynamic range of a cocaine switch system was extended by 330 000-fold by combining four different variants of the cocaine aptamer [85]. However, because this strategy uses switch variants of reduced affinity, the midpoint will therefore always be higher than the midpoint of the native switch. To overcome that limitation, it is possible to use an allosteric approach to create new switches that will have higher affinity (activation) or lower affinity (inhibition) [51]. This strategy has been used to create a mixture of Hg^{2+} molecular beacons that are activated over a 333-fold dynamic range and that remain centered around its natural K_{D} of 16 μM [86].

We have previously discussed strategies that enable one to extend the dynamic range of switches. However, in DNA computing, it is often more relevant to engineer all-or-none switches (i.e. switches that are activated over a narrow dynamic range). Such systems can be engineered by exploiting mechanisms used in regulatory networks [87]. One of these strategies involves the introduction of a molecule (called a depletant) that will sequester the input (Figure 7.7b, top). It is important that this new molecule, which can be a small molecule, DNA, protein, or others, has a higher affinity toward the input ($K_{\mathrm{D}}^{\mathrm{dep}} < K_{\mathrm{D}}^{\mathrm{input}}$) to prevent the accumulation of free input. As long as there is depletant molecule available, the input will not be able to activate the switch until reaching a certain concentration threshold defined by the total concentration of depletant in the system (Figure 7.7b, bottom). Above that concentration, further addition of input leads to a large increase in free input that can immediately activate the switch, thus producing a "pseudo-cooperative" dose–response curve (i.e. a small dynamic range). It has been demonstrated that a key parameter to achieve a small dynamic range is the stoichiometric binding parameter, which corresponds to the ratio of depletant concentration over the affinity of the switch for the input ([depletant]/$K_{\mathrm{D}}^{\mathrm{input}}$) [88]. This parameter dictates whether mass action will favor the depletant-induced sequestration (>1) or the formation of free input (<1). It has been shown that the dynamic range starts increasing monotonically when the stoichiometric binding parameter becomes higher than unity. Simulation of
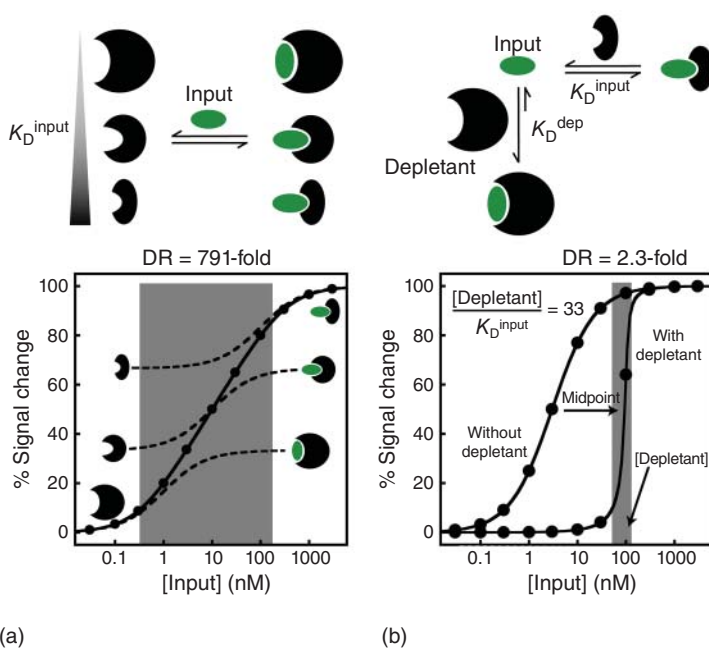
Figure 7.7 (a) Two or more switches with different affinities for the input can be combined to enlarge the dynamic range. For example, using three different switches with midpoints of 1, 10, and 100 nM enables one to enlarge the dynamic range to 791-fold without affecting the linearity of the response. (b) The sequestration mechanism can be used to achieve an ultrasensitive response at a threshold concentration corresponding to the depletant concentration. For example, using a stoichiometric binding parameter of 33 enables one to decrease the dynamic range up to 2.3-fold.

that mechanism predicts that it is theoretically possible to achieve a Hill factor as high as 21.6 (DR = 1.22-fold) when using a stoichiometric binding parameter of 10 000 [88]. However, narrowing the dynamic range using that strategy comes with the trade-off of increasing the midpoint of the curve (Figure 7.7b, bottom). Nonetheless, by using this strategy, many researchers were able to engineer all-or-none switches that can, for example, activate by a 4-fold change in input concentration when using TATA-binding protein, 3-fold when using the cocaine aptamer, and even 1.5-fold when using a molecular beacon [85, 89].

7.6 Perspective

In this chapter, we have summarized various strategies for engineering DNA switches and their applications for DNA computing. Such switches are likely to drive many innovations in the fields of medicine, green chemistry, and nanotechnology, but several challenges lie ahead before realizing this promise beyond laboratory-scale prototypes [90]. One such challenge is developing switches that achieve sufficient specificity and selectivity (i.e. only triggered

by a specific molecular input) even in complex conditions or environments (such as whole blood, soil, etc.). To that end, we believe that expending the DNA code with other artificial nucleotides should greatly contribute to creating more specific recognition elements [91, 92]. Another challenge consists of better characterizing and optimizing DNA switch systems to obtain their innate structural and dynamic profiles. Such information would provide the rational basis to better optimize the switch's function and response behavior. With this in mind, we believe that novel tools to characterize the switch's thermodynamic signature (see Ref. [48]) and tune it, using simple and inexpensive strategies (such as employing inhibitors or activators [51]), should greatly contribute to making design strategies more rational and quantitative. Concerning switch kinetics, it is also important to note that activation and deactivation of DNA switches using a DNA trigger remains relatively slow, which limits applications in DNA computing. More specific challenges to move beyond laboratory-scale prototypes include optimizing the accuracy (through calibration against a standard), stability, repeatability, and reproducibility of DNA switch systems [90]. *In vivo* applications will also require a better understanding of the mechanisms underlying intracellular uptake, trafficking, and nanotoxicology [93], in addition to the characterization of their pharmacokinetic properties. Finally, a better understanding of large-scale production of DNA-based systems for better yields and lower costs [94] is mandatory to scale-up laboratory or pilot technologies to reach the production and commercialization stages.

Acknowledgments

References

1 Harroun, S.G., Prévost-Tremblay, C., Lauzon, D. et al. (2018). *Nanoscale* 10: 4607–4641.

2 Ma, C.-W., Zhou, L.-B., and Zeng, A.-P. (2018). Engineering biomolecular switches for dynamic metabolic control. In: *Synthetic Biology – Metabolic Engineering* (eds. H. Zhao and A.-P. Zeng), 45–76. Cham: Springer International Publishing.

3 Wieland, M. and Fussenegger, M. (2012). *Annu. Rev. Chem. Biomol. Eng.* 3: 209–234.

4 Chin, D. and Means, A.R. (2000). *Trends Cell Biol.* 10: 322–328.

5 Vallée-Bélisle, A., Ricci, F., and Plaxco, K.W. (2009). *Proc. Natl. Acad. Sci. U.S.A.* 106: 13802.

6 Lagerström, M.C. and Schiöth, H.B. (2008). *Nat. Rev. Drug Discovery* 7: 339–357.

7 Rosenbaum, D.M., Rasmussen, S.G.F., and Kobilka, B.K. (2009). *Nature* 459: 356–363.

8 Krishnan, Y. and Simmel, F.C. (2011). *Angew. Chem. Int. Ed.* 50: 3124–3156.

9 Wang, F., Liu, X., and Willner, I. (2015). *Angew. Chem. Int. Ed.* 54: 1098–1129.

10 Harris, J.D., Moran, M.J., and Aprahamian, I. (2018). *Proc. Natl. Acad. Sci. U.S.A.* 115: 9414.

11 Pei, R., Matamoros, E., Liu, M. et al. (2010). *Nat. Nanotechnol.* 5: 773–777.

12 Qian, L. and Winfree, E. (2011). *Science* 332: 1196.

13 Chen, J., Zhou, S., and Wen, J. (2015). *Angew. Chem. Int. Ed.* 54: 446–450.

14 Pei, H., Zuo, X., Zhu, D. et al. (2014). *Acc. Chem. Res.* 47: 550–559.

15 Engelen, W., Meijer, L.H.H., Somers, B. et al. (2017). *Nat. Commun.* 8: 14473.

16 Vallée-Bélisle, A., Ricci, F., and Plaxco, K.W. (2012). *J. Am. Chem. Soc.* 134: 2876–2879.

17 Ebrahimi, S., Akhlaghi, Y., Kompany-Zareh, M., and Rinnan, A. (2014). *ACS Nano* 8: 10372–10382.

18 Gareau, D., Desrosiers, A., and Vallée-Bélisle, A. (2016). *Nano Lett.* 16: 3976–3981.

19 Ke, G., Wang, C., Ge, Y. et al. (2012). *J. Am. Chem. Soc.* 134: 18908–18911.

20 Asanuma, H., Ito, T., Yoshida, T. et al. (1999). *Angew. Chem. Int. Ed.* 38: 2393–2395.

21 Lubbe, A.S., Liu, Q., Smith, S.J. et al. (2018). *J. Am. Chem. Soc.* 140: 5069–5076.

22 Zhou, M., Liang, X., Mochizuki, T., and Asanuma, H. (2010). *Angew. Chem. Int. Ed.* 49: 2167–2170.

23 Tyagi, S. and Kramer, F.R. (1996). *Nat. Biotechnol.* 14: 303–308.

24 Idili, A., Vallée-Bélisle, A., and Ricci, F. (2014). *J. Am. Chem. Soc.* 136: 5836–5839.

25 Benabou, S., Aviñó, A., Eritja, R. et al. (2014). *RSC Adv.* 4: 26956–26980.

26 Kwok, C.K. and Merrick, C.J. (2017). *Trends Biotechnol.* 35: 997–1013.

27 Ono, A. and Togashi, H. (2004). *Angew. Chem. Int. Ed.* 43: 4300–4302.

28 Ono, A., Cao, S., Togashi, H. et al. (2008). *Chem. Commun.*: 4825–4827.

29 Breaker, R.R. and Joyce, G.F. (1995). *Chem. Biol.* 2: 655–660.

30 Liu, J. and Lu, Y. (2003). *J. Am. Chem. Soc.* 125: 6642–6643.

31 Liu, J., Brown, A.K., Meng, X. et al. (2007). *Proc. Natl. Acad. Sci. U.S.A.* 104: 2056–2061.

32 Tuerk, C. and Gold, L. (1990). *Science* 249: 505–510.

33 Jayasena, S.D. (1999). *Clin. Chem.* 45: 1628–1650.

34 Fang, X. and Tan, W. (2010). *Acc. Chem. Res.* 43: 48–57.

35 Gopinath, S.C., Hayashi, K., and Kumar, P.K. (2012). *J. Virol.* 86: 6732–6744.

36 Hamula, C.L., Zhang, H., Guan, L.L. et al. (2008). *Anal. Chem.* 80: 7812–7819.

37 Shangguan, D., Li, Y., Tang, Z. et al. (2006). *Proc. Natl. Acad. Sci. U.S.A.* 103: 11838–11843.

38 Zheng, J., Li, J., Jiang, Y. et al. (2011). *Anal. Chem.* 83: 6586–6592.

39 Achenbach, J.C., Nutiu, R., and Li, Y. (2005). *Anal. Chim. Acta* 534: 41–51.

40 Park, K.S., Seo, M.W., Jung, C. et al. (2012). *Small* 8: 2203–2212.

41 Abou Assi, H., Garavís, M., González, C., and Damha, M.J. (2018). *Nucleic Acids Res.* 46: 8038–8056.

42 Sullivan, R., Adams, M.C., Naik, R.R., and Milam, V.T. (2019). *Molecules* 24: 1572.

43 Cai, S., Yan, J., Xiong, H. et al. (2018). *Analyst* 143: 5317–5338.

44 Zhou, W., Jimmy Huang, P.-J., Ding, J., and Liu, J. (2014). *Analyst* 139: 2627–2640.

45 Zadeh, J.N., Steenberg, C.D., Bois, J.S. et al. (2011). *J. Comput. Chem.* 32: 170–173.

46 Zuker, M. (2003). *Nucleic Acids Res.* 31: 3406–3415.

47 Owczarzy, R., Tataurov, A.V., Wu, Y. et al. (2008). *Nucleic Acids Res.* 36: W163–W169.

48 Idili, A., Ricci, F., and Vallée-Bélisle, A. (2017). *Nucleic Acids Res.* 45: 7571–7580.

49 You, Y., Tataurov, A.V., and Owczarzy, R. (2011). *Biopolymers* 95: 472–486.

50 Wang, K., Tang, Z., Yang, C.J. et al. (2009). *Angew. Chem. Int. Ed.* 48: 856–870.

51 Ricci, F., Vallée-Bélisle, A., Porchetta, A., and Plaxco, K.W. (2012). *J. Am. Chem. Soc.* 134: 15177–15180.

52 Zheng, X., Yang, J., Zhou, C. et al. (2018). *Nucleic Acids Res.* 47: 1097–1109.

53 Furukawa, K. and Minakawa, N. (2014). *Org. Biomol. Chem.* 12: 3344–3348.

54 Gao, R.-R., Shi, S., Zhu, Y. et al. (2016). *Chem. Sci.* 7: 1853–1861.

55 Stojanovic, M.N., Mitchell, T.E., and Stefanovic, D. (2002). *J. Am. Chem. Soc.* 124: 3555–3561.

56 Pei, H., Liang, L., Yao, G. et al. (2012). *Angew. Chem. Int. Ed.* 51: 9020–9024.

57 Zhang, L., Bluhm, A.M., Chen, K.-J. et al. (2017). *Nanoscale* 9: 1709–1720.

58 Chen, J. and Zeng, L. (2013). *Biosens. Bioelectron.* 42: 93–99.

59 Schlichthaerle, T., Strauss, M.T., Schueder, F. et al. (2016). *Curr. Opin. Biotechnol.* 39: 41–47.

60 Huang, J., Yang, X., He, X. et al. (2014). *TrAC, Trends Anal. Chem.* 53: 11–20.

61 Jayagopal, A., Halfpenny, K.C., Perez, J.W., and Wright, D.W. (2010). *J. Am. Chem. Soc.* 132: 9789–9796.

62 Li, F., Huang, Y., Yang, Q. et al. (2010). *Nanoscale* 2: 1021–1026.

63 Jean, J.M. and Hall, K.B. (2001). *Proc. Natl. Acad. Sci. U.S.A.* 98: 37–41.

64 Millar, D.P. (1996). *Curr. Opin. Struct. Biol.* 6: 322–326.

65 Zhu, G., Liang, L., and Zhang, C.Y. (2014). *Anal. Chem.* 86: 11410–11416.

66 Feng, C., Dai, S., and Wang, L. (2014). *Biosens. Bioelectron.* 59: 64–74.

67 Tang, Y., Ge, B., Sen, D., and Yu, H.-Z. (2014). *Chem. Soc. Rev.* 43: 518–529.

68 Ronkainen, N.J., Halsall, H.B., and Heineman, W.R. (2010). *Chem. Soc. Rev.* 39: 1747–1763.

69 Fan, C., Plaxco, K.W., and Heeger, A.J. (2003). *Proc. Natl. Acad. Sci. U.S.A.* 100: 9134–9137.

70 Lubin, A.A. and Plaxco, K.W. (2010). *Acc. Chem. Res.* 43: 496–505.

71 Xiao, Y., Lubin, A.A., Heeger, A.J., and Plaxco, K.W. (2005). *Angew. Chem. Int. Ed.* 44: 5456–5459.

72 Baker, B.R., Lai, R.Y., Wood, M.S. et al. (2006). *J. Am. Chem. Soc.* 128: 3138–3139.

73 Vallée-Bélisle, A., Ricci, F., Uzawa, T. et al. (2012). *J. Am. Chem. Soc.* 134: 15197–15200.

74 Mahshid, S.S., Camiré, S., Ricci, F., and Vallée-Bélisle, A. (2015). *J. Am. Chem. Soc.* 137: 15596–15599.

75 Kang, D., White, R.J., Xia, F. et al. (2012). *NPG Asia Mater.* 4: e1.

76 Katz, E. and Willner, I. (2003). *Electroanalysis* 15: 913–947.

77 Bardea, A., Patolsky, F., Dagan, A., and Willner, I. (1999). *Chem. Commun.*: 21–22.

78 Fan, D., Fan, Y., Wang, E., and Dong, S. (2018). *Chem. Sci.* 9: 6981–6987.

79 Li, W., Nie, Z., Xu, X. et al. (2009). *Talanta* 78: 954–958.

80 Zayats, M., Huang, Y., Gill, R. et al. (2006). *J. Am. Chem. Soc.* 128: 13666–13667.

81 Gao, W., Zhang, L., Zhang, Y.-M. et al. (2014). *J. Phys. Chem. C* 118: 14410–14417.

82 Freeman, R., Liu, X., and Willner, I. (2011). *J. Am. Chem. Soc.* 133: 11597–11604.

83 Kim, N.H., Lee, S.J., and Moskovits, M. (2010). *Nano Lett.* 10: 4181–4185.

84 Simon, A.J., Vallée-Bélisle, A., Ricci, F. et al. (2014). *Angew. Chem. Int. Ed.* 53: 9471–9475.

85 Porchetta, A., Vallée-Bélisle, A., Plaxco, K.W., and Ricci, F. (2012). *J. Am. Chem. Soc.* 134: 20601–20604.

86 Porchetta, A., Vallée-Bélisle, A., Plaxco, K.W., and Ricci, F. (2013). *J. Am. Chem. Soc.* 135: 13238–13241.

87 Buchler, N.E. and Cross, F.R. (2009). *Mol. Syst. Biol.* 5: 272.

88 Buchler, N.E. and Louis, M. (2008). *J. Mol. Biol.* 384: 1106–1119.

89 Ricci, F., Vallée-Bélisle, A., and Plaxco, K.W. (2011). *PLoS Comput. Biol.* 7: e1002171.

90 Fadel, T.R., Farrell, D.F., Friedersdorf, L.E. et al. (2016). *ACS Sens.* 1: 207–216.

91 Gupta, S., Hirota, M., Waugh, S.M. et al. (2014). *J. Biol. Chem.* 289: 8706–8719.

92 Drolet, D.W., Green, L.S., Gold, L., and Janjic, N. (2016). *Nucleic Acid Ther.* 26: 127–146.

93 Bamrungsap, S., Zhao, Z., Chen, T. et al. (2012). *Nanomedicine* 7: 1253–1271.

94 Tørring, T. and Gothelf, K.V. (2013). *F1000Prime Rep.* 5: 14.

8

Fluorescent Signal Design in DNA Logic Circuits

Dan Huang[1], Shu Yang[2], and Qianfan Yang[1]

[1] *Sichuan University, College of Chemistry, No.24 South Section 1, Yihuan Road, Chengdu 610065, China*
[2] *Sichuan University, West China School of Pharmacy, No. 17, Section 3, Southern Renmin Road, Chengdu 610041, China*

Information processing, or computation, is a central theme of computer science. This task can be performed not only by our acquainted computers made from silicon chips but also by the natural "computers" composed of cells and molecules. The concept of the molecular computer dates decades back [1]. First experiment was implemented by Adleman in 1994, solving the Hamiltonian path problem via using DNA hybridization, demonstrating that an algorithm can be encoded in DNA and used to perform computation operations [2]. Since then, molecules were biochemically manipulated to generate DNA/RNA-based systems that attempt to solve computationally problems [3–5].

Though there are still many challenges in designing DNA computation devices that would rival silicon-based computation, DNA-based logic circuits are developing with great rapidity, owing to their stability, predictable structure, and easy synthesis [6, 7]. Besides, the capability of interacting with various biochemical factors endows DNA-based logics good biocompatibility, which is of importance in developing cellular computation [8, 9] and medical application [10, 11].

For any logic device, either man-made objects as artificial intelligence or natural ones as brains, the mechanism works with the same input/output pattern: **sensor** that collects information (*input*) from inside and outside the system, **processor** (or computer) that interprets the information to decide on the response, and **actuator** (*output*) that carries the response out [6]. For DNA-based logic circuits, the **sensors** and **processors** are integrated in most cases. DNA strands with certain secondary structures that can be regulated by input stimuli (well-encoded oligonucleotides or selected ligands) always play the both roles. To realize the process, current designs are generally based on DNA hybridization, molecular beacon (MB) probes [12, 13], DNAzyme function [14], aptamer ligand binding [15, 16], toehold-mediated strand displacement [17], and so on (Figure 8.1). In the case of **actuator**, however, DNA can hardly provide detectable signal by itself. In most reported DNA-based logic systems, the **actuators** are independent from nucleic acid sequences and designed to provide easy-to-detect signals, like optical [19–22], electrochemical [23–26], and even biological ones [27–29], among
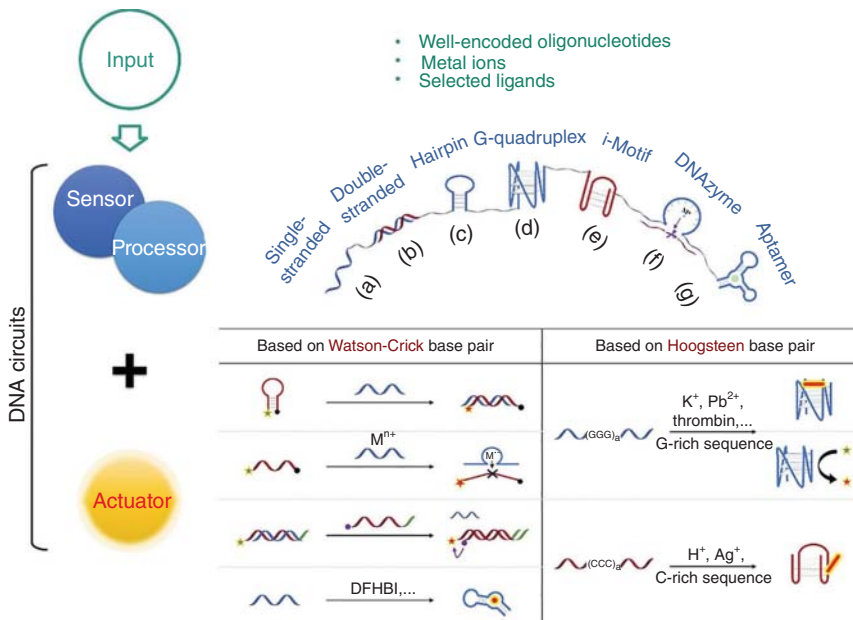
Figure 8.1 The universal input/output mechanism of DNA-based logic circuit. The sensor and processor are DNA strands with certain secondary structures: (a) single stranded, (b) double stranded, (c) hairpin, (d) G-quadruplex, (e) i-motif, (f) DNAzyme, and (g) aptamer. The fluorescence actuator is extrinsic in most cases, and some classic signal strategies based on the two kinds of DNA structures are stabilized by Watson–Crick and Hoogsteen hydrogen bonds. Source: Based on Wilner and Willner [18].

which fluorescent signals are studied and applied most extensively due to their high sensitivity, multichannel scalability, rapid response, and less dependence on equipment.

In this chapter, we focused on the latest design strategies of fluorescent output for DNA logic circuits. We began this chapter with an overview of basic strategies on generating single-channel fluorescence signal based on certain structure transformations of DNA. Next, we discussed some designs on constructing multi-output systems, including combining parallel signal transducers, designing probes that can participate in the logic process, and introducing probes with multichannel. Finally, we concluded by summarizing some significant questions in the field.

8.1 Basic Signal Generation Strategies Based on DNA Structures

The DNA secondary structure is of great diversity. Besides the well-known double helix structure of complementary nucleotide base pairs that uncovered by Watson and Crick in 1953, other structures such as G-quadruplex self-assembled by guanosine bases [30–32], i-motif [33–35], or triplex [36–38] stimulated by

acidic condition, and stem-loop structure bridged by metal ions [39] or induced by aptamer ligands [40, 41], have been established successively. And the versatile structures lead to numerous strategies on signal design in DNA logic circuits.

The hydrogen bonds involved in DNA secondary structures, in spite of their identity in nature, are often classified into two categories: classic Watson–Crick hydrogen bonds and Hoogsteen ones that disobey the base pairing principle. In this section, we reviewed some classic signal strategies separately based on these two kinds of hydrogen bonds (Figure 8.1).

8.1.1 Strategies Based on Watson–Crick Hydrogen Bond

8.1.1.1 Signal Derived from Hairpin Structure/Molecular Beacon

Hairpin structure, firstly reported by Tyagi and Kramer in 1996 [42], has been used in logic device construction most successfully so far [43, 44]. The typical hairpin structure is a single-stranded (ss) DNA chain containing two complementary base regions, which could fold into a double-stranded (ds) stem by hybridizing its two ends with an unpaired loop. So, it is also termed as stem-loop structure. The classic MB probe is a hairpin labeled with a fluorophore and a corresponding quencher at two ends, which has been widely used in biochemical research and practical application, such as gene detection, miRNA analysis, and medical diagnosis [45]. When an MB probe is OFF (in the form of hairpin structure), it would not emit fluorescence signal due to the fluorescence resonance energy transfer (FRET) effect between the fluorophore and quencher that are close spatially. While it is ON (the stem is degenerated), the fluorophore is apart from the quencher and presents strong fluorescence. The ON/OFF switch based on the structural conversion is perfect for binary readout, i.e. 1/0, in logic operation, making MB or MB-like structure one of the most popular strategies in DNA logic constructing [12, 13, 46].

Based on this strategy, Kolpashchikov and coworkers [46] developed a versatile DNA system for implementing basic YES and AND logic functions. They designed a system consisting of MB probe **MB1**, ssDNA strands **YES1** and **YES2**, and **I1m,** which is complementary with the input strand **I1**. Without the input **I1**, **MB1** was in the OFF state with low fluorescence, corresponding to a binary "0." Addition of excess amount of **I1** results in concurrent formation of both the signaling YES complex (**MB1**-**YES1**-**YES2**-**I1**) and **I1-11m** duplex, leading to a high fluorescence signal "1" (Figure 8.2).
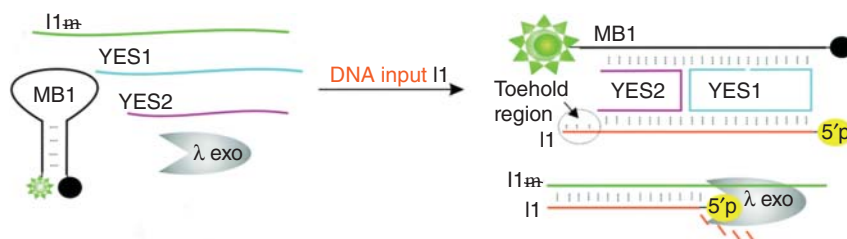
Figure 8.2 Schematic representation of the YES gate based on the hairpin strategy. Source: From O'Steen et al. [46]. Reproduced with the permission of Royal Society of Chemistry.

Following this line of thinking, Park et al. [47] succeeded in developing a simple and universal platform for more logic gate operations. They designed an MB probe as a universal component. ssDNAs were used as the input to induce conformational change of the MB probe, resulting in the fluorescence variation accompanied by the opening of the MB probe. On this platform, a complete set of two-input logic gates, including OR, AND, XOR, INHIBIT, NOR, NAND, XNOR, and IMPLICATION, were implemented at the molecular level. On a similar strategy, Yang et al. [48, 49] constructed a set of advanced arithmetic logic circuits, including half-adder (HA), half-subtractor (HS), and full-subtractor (FS). And Wang's group made the best of the MB strategy and developed a universal platform to expand the realizable logic functions, including both arithmetic [50] ones, like HA, HS, full-adder (FA), and FS, and non-arithmetic [51] ones, like multiplexer (MUX), demultiplexer (DEMUX), encoder (EC), and decoder (DC).

In addition to systems cantered on hairpin-structured strand, the MB probe can also be involved in high-order DNA assembly systems as an individual signal transducer. Kolpashchikov's group designed a series of tile-integrated DNA modules, which could implement various basic logic functions, like NOT, AND, INHIBIT, and NOR gates [52]. In addition, they positioned DNA logic gates in a precise order on a two-dimensional platform by hairpin–duplex structural interconversion and realized approximately 10^{14} DNA circuits in 1 ml solution and achieved the expected digital responses and circuit resetting within minutes [53], which leads to a possible way to manufacture DNA-based integrated circuit with large scale and affordable cost.

8.1.1.2 Signal Derived from DNAzyme Activity

Aside from base pairing hybridization, DNAzyme that is capable of performing catalytic activity [14, 45] (usually cleaving targeted DNA/RNA) also provides a set of unique tools for manipulating DNA/RNA substrates specifically and producing consequent signal [54, 55].

Willner and coworkers [56] constructed several computing modules, which consist of a library of DNAzyme subunits and fluorophore/quencher-labeled ssDNA substrates. As shown in Figure 8.3, they designed 14 DNAzyme subunits and 3 hairpin inputs (I_1, I_2, and I_3), which could self-assemble to 7 Mg^{2+}-dependent DNAzymes (P_1 to P_7) targeting 2 labeled substrates, S_1 and S_2. P_1 to P_3 have the ability to cleave S_1, P_4 to P_6 cleave S_2, and P_7 cleaves both. In different inputting situations, a certain DNAzyme is formed, and the corresponding substrate(s) is cleaved to emit fluorescence signals. In this way, the system can deal with multiple independent inputs simultaneously and implement some complicated functions, including the FA, MUX, and DEMUX. In similar way, Kolpashchikov and coworker [57] realized multi-input AND gates, which might be applied in analyzing multiple biomarkers.

Moreover, these kinds of modularized DNA units can be combined at various scales and implement more complex functions, even intelligent activities [58]. Stojanovic and Stefanovic built three generations of game-playing automata, MAYA I–III based on catalytic MB units. MAYA I [59] presents the ability to play a symmetry-pruned game of tic-tac-toe, MAYAII [60] could play the unrestricted
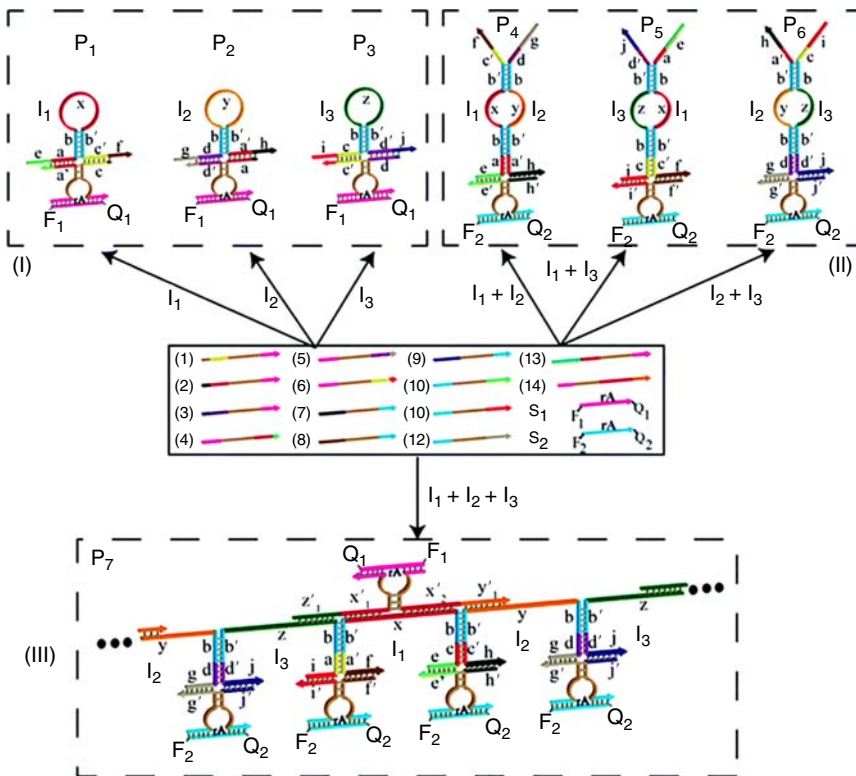
Figure 8.3 Schematic representation of the DNA computing assembly based on the DNAzyme strategy. The system contains a library of Mg^{2+}-dependent DNAzyme subunits (P_1 to P_7), two different fluorophore/quencher-modified substrates (S_1 and S_2), and three different hairpin inputs (I_1, I_2, and I_3). Panel (I) – computing modules generated by inputs I_1, I_2, or I_3, leading to the fluorescence output F_1. Panel (II) – computing modules formed by the reconfiguration of the hairpin inputs $I_1 + I_2$, $I_1 + I_3$, or $I_2 + I_3$, leading to the output F_2. Panel (III) – computing module consisting of polymeric DNAzyme wires, generated by the sequential inter-input hybridization, leading to the outputs F_1 and F_2. Source: Modified from Orbach et al. [56].

game using a richer encoding of inputs, and MAYA III [61] could be trained to play specific strategies in a specially designed simple game. All these works provide a brilliant prospect of DNA-based artificial intelligence.

8.1.1.3 Signal Derived from Strand Displacement Reaction

Although DNA structures show advantages in constructing computing circuits, like designable scaffold, reconfigurable activities, and good scalability, the self-assembly process requires a heating and time-consuming anneal treatment, which hinders their further application. In 1992, Walker [62] firstly reported an isothermal DNA dynamic transformation termed strand displacement reaction that a ssDNA replaces one chain hybridized in the template duplex spontaneously and forms a new duplex with enhanced stability induced by a sticky end (termed toehold) [17, 63]. Since the strand displacement reaction can be easily cascaded and mimic the wire function in digital electronics, some
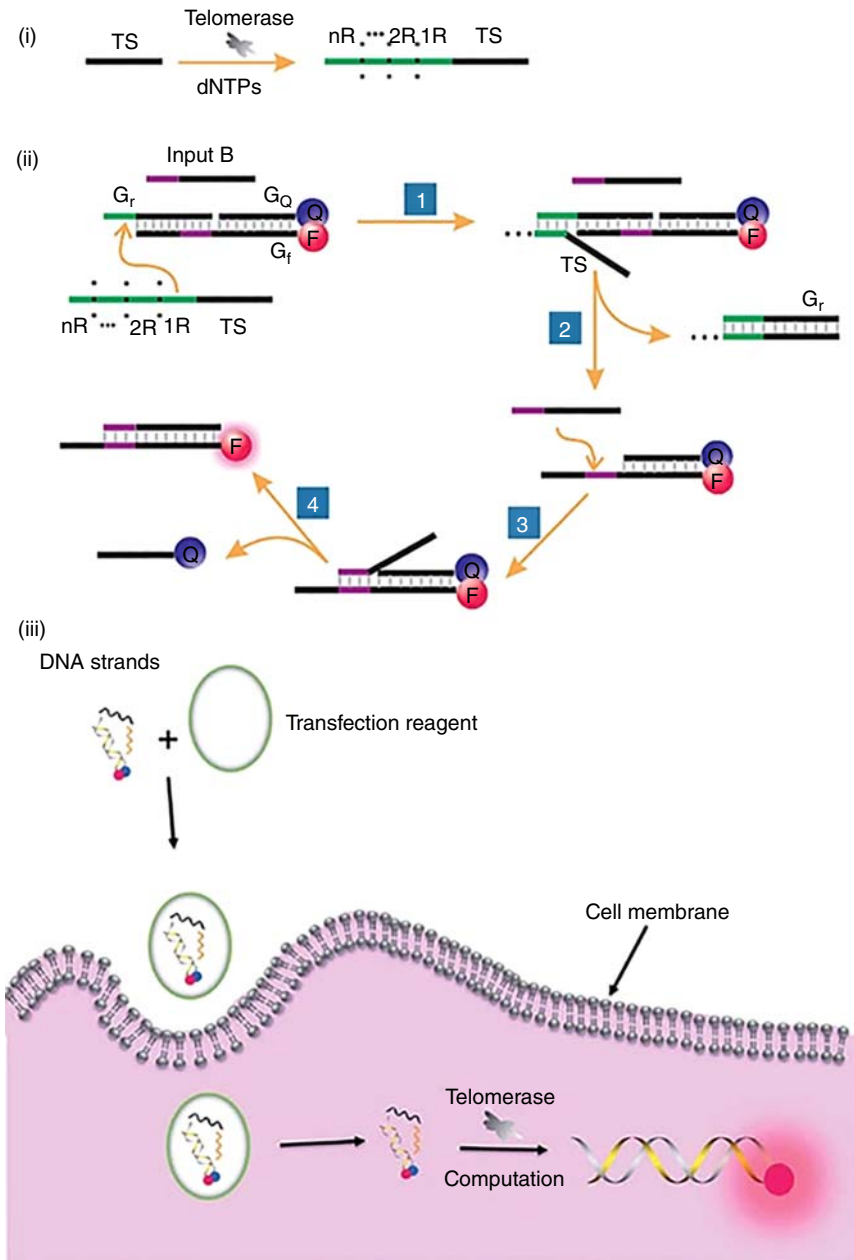
Figure 8.4 Schematic representation of cascaded DNA computation based on the strand displacement reaction. (i) Upon extension by telomerase in cells as a YES gate. (ii) The extended **TS** probe (**TS** + **nR**) and **Input B** strand worked as inputs to initiate toehold-mediated strand displacement reaction and implement AND gate function. (iii) Schematic representation of the evaluation of intracellular telomerase using the cascade DNA logic gate. Source: From Wang et al. [67]. Licensed under CC by 3.0.

logic devices were also developed based on the spontaneous DNA reaction [64–66].

Zhang, Zhu, and coworkers [67] constructed a cascade nucleic acid logic gate that responded to intracellular telomerase and applied it in imaging intracellular telomerase activity. They designed a DNA computation system includes the **TS** probe, the telomerase substrate primer that could be extended by intracellular telomerase, a toehold-bearing DNA duplex that consists of a G_T strand, a fluorophore-modified G_F strand and a quencher-modified G_Q strand, and **Input B** that is the partial-complementary strand of G_F. Firstly, the **TS** probe could be extended by telomerase with repetitive sequences of TTAGGG, forming **TS + nR** strand (Figure 8.4i). Then, the **TS + nR** strand could recognize and hybridize with the toehold domain of the duplex (G_T–G_F–G_Q) to initiate spontaneous strand migration and displacement, thereby separating the fluorophore and the quencher, producing a fluorescence signal indicating the presence of telomerase (Figure 8.4ii). In this system, intracellular telomerase worked as the specific input to activate the first YES gate and the output of this YES gate (**TS + nR**) and **Input B** strand together performed the following AND gate computation. Besides the computation prototype, they also transfected the system in cells to verify its feasibility *in vitro* and studied its imaging capability in cancer cells (Figure 8.4iii). Finally, they succeeded in mapping the telomerase activity in various cell lines and measuring it quantitatively in HeLa cells.

Based on strand displacement reaction, the thermodynamics have been extensively studied, and several isothermal amplification technologies, like loop-mediated isothermal amplification (LAMP) [68], rolling circle amplification (RCA) [69, 70], hybridization chain reaction (HCR) [71–73], and catalytic hairpin assembly (CHA) [74, 75], have been developed. Especially, the protein enzyme-free amplification technologies (CHA and HCR) have been widely used in DNA-based logic systems [76, 77].

Although most logic systems based on strand displacement reaction employ FRET signal strategy, there are still some works attempt to develop other signal producing mechanisms. Deiters and coworkers [78] reported a logic system containing oligonucleotides modified with 7-amino-4-methylcoumarin (AMC) and phosphine separately, the hybridization of which triggered the Staudinger reaction partners into close proximity and yielded small molecular output signals. The mechanism is quite like FRET, but the system offers turned-on fluorescence signal by hybridization. By step-by-step strand displacement reactions, several basic logic functions were realized via inputting certain miRNAs. Taking the AND gate as an example (Figure 8.5), miR-122 and miR-21 were selected as inputs, and two strands, **TG122** labeled with phosphine and **AND-G21** with AMC, were designed as the logic processor. By sequence design, miR-122 and miR-21 could remove the complement strands of **TG122** and **AND-G21**, respectively. Then, the two labeled strands exposed new toeholds and hybridized together, resulting in the Staudinger reaction between phosphine and AMC, and consequently high fluorescence signal. While inputting either miR-21 or miR-122 could not trigger the downstream strand displacement and gives low fluorescence.
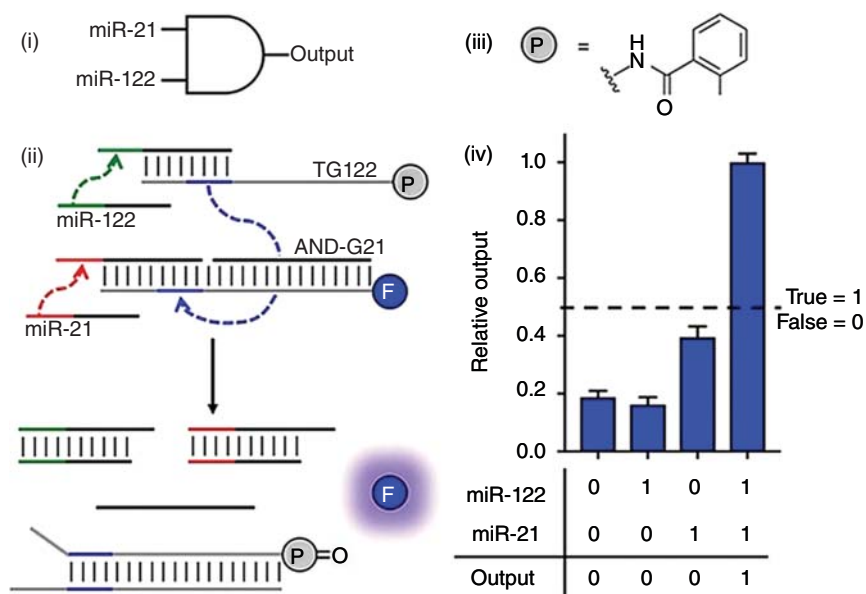
Figure 8.5 Schematic representation of the AND gate based on the strand displacement reaction. (i) Electronic symbol for the AND gate. (ii) Simplified schematic of the cascaded strand displacement reactions and the release of the signal molecule in the AND gate. Toehold regions are shown in green, red, and blue. (iii) The structure of the phosphine, 2(diphenylphosphino)benzamide (2DPBM). (iv) The normalized AMC fluorescence and the truth table of the AND gate. Source: From Morihiro et al. [78]. Reproduced with the permission of American Chemical Society.

8.1.2 Strategies Based on Hoogsteen Hydrogen Bond

Inspired by MB probe, various FRET/FRET-like mechanisms [79] have been designed to control the ON/OFF switch of fluorescence signal based on traditional base pairing hybridization, as well as DNAzyme catalysis, strand displacement reaction, aptamer–ligand interaction, and metal-bridged base pair mismatch [80]. However, all these systems require pre-labeled fluorophore, leading to sophisticated synthesis, high cost, strict preservation condition, and limited lifespan. The introduction of DNA motifs based on Hoogsteen hydrogen bond, such as G-quadruplex, i-motif, and triplex DNA, brings more types of signal modules, enriching the diversity of logic design.

8.1.2.1 Signal Derived from G-Quadruplex

G-quadruplex is a kind of DNA secondary structure that forms by stacked planar G-quartets composed of four guanines (G) by Hoogsteen hydrogen bonding and is usually stabled by metal cations [81, 82]. It adopts a variety of topologies and can be approximately classified into three subtypes based on the strand orientation, i.e. parallel, antiparallel, and hybrid [31]. Due to the unique topologies, some small molecular ligands, such as porphyrin [83, 84], alkaloid [85, 86], and cyanine dye [87–89], can specifically bind on certain G-quadruplexes via non-covalent interaction and present distinguish signals, which could be used

to tell G-quadruplexes from massive ssDNA/dsDNA, even to recognize certain subtypes [90]. For example, *N*-methyl mesoporphyrin IX (NMM) [91] and cyanine dye 3,3′-di(3-sulfopropyl)-4,5,4′,5′-dibenzo-9-methylthiacarbocyanine triethylammonium salt (MTC) [92] are reported as excellent probes for parallel/hybrid G-quadruplex by strong emission around 600 nm, while thioflavin T (ThT) [93, 94] can indicate antiparallel G-quadruplex via fluorescence near 500 nm. The specific interactions to probes make G-quadruplex an excellent signal module, and the conversion between ssDNA and certain G-quadruplex is also a universal strategy for constructing label-free logic systems.

It has been investigated that the topology and stability of G-quadruplex depend on many factors, including the length and sequence composition, the size of the loops between the guanines, strand stoichiometry, and alignment [95–99]. Besides, various metal cations [100], such as K^+, Na^+, Pb^{2+}, etc., and ligands, such as thrombin [101], hemin [102], berberine [86], sanguinarine [103], etc., have the ability to facilitate the formation of certain G-quadruplex or to regulate it from one subtype to another, providing abundant input candidates and sufficient space for DNA sequence design [104].

Pei and coworkers [105] designed a GT-rich ssDNA, **GT24**, which could be induced to form various structures by different inputting stimuli, such as parallel G-quadruplex by K^+ and thrombin, antiparallel G-quadruplex by Pb^{2+}, and T-Hg^{2+}-T dsDNA by Hg^{2+}. Employing NMM as the probe to indicate the formation of parallel G-quadruplex, they succeeded in implementing a series of Boolean logic functions, including NOT, NOR, OR, and AND gate, in an easy way.

Another interesting logic system with G-quadruplex was constructed recently by Bader and Cockroft [106], in which three G-quadruplex-based Boolean logic gates were operated and monitored simultaneously in a single reaction vessel. Their design was inspirited from **Plas24**, *Plasmodium falciparum* telomeric G-quadruplex sequence, and ThT was employed to indicate the formation of antiparallel G-quadruplex. In the initial state, the G-rich strand was complemented, while inputting oligonucleotides could trigger strand displacement reactions according to the appropriate Boolean rules, i.e. YES, OR, and AND release the G-rich strand to form G-quadruplex and turn on the ThT signal. To differentiate the response of each gate in a mixed solution, three different G-rich strands were designed to have either zero, one, or two flanking sequences, which could be distinguished by gel electrophoresis. Such a versatile, specific, and cost-efficient strategy could be employed for monitoring and debugging dynamic multitasking DNA-based devices or exploited in biosensing and theranostic applications where high levels of background noise might otherwise be encountered.

Besides NMM and ThT, other G-quadruplex ligands also show similar fluorescence feature and have the potential to be involved in DNA-based circuit construction. For example, the fluorescence of protoporphyrin IX (PPIX) can be sharply increased by certain G-quadruplex [107] and be quenched by Cu^{2+}, which provides a feasible way to construct basic logic gates [108]. MTC is a kind of cyanine dye that can self-assemble into aggregates. Certain G-quadruplex can bind specifically with MTC and disassemble it into monomer and enhance

its fluorescence up to 2000 times [87]. By utilizing the excellent specificity and sensitivity of MTC as the G-quadruplex probe, our group recently achieved to construct several advanced logic circuits and applied them as intelligent sensors of lead in complicated environments [109].

In addition to non-covalent fluorescent probes, G-quadruplex-based circuits could also rely on G-quadruplex/hemin DNAzyme system (termed as G4zyme), which presents peroxidase-like catalyze activity. For instance, a few K^+-stabilized G4zymes have the ability to catalyze a bunch of substrates mediated by H_2O_2 and present either colorimetric or fluorescence results. Common used colorimetric substrates include 2,2′-azino-bis(3-ethylbenzothiazoline-6-sulfonic acid) diammonium salt (ABTS) [110, 111] and 3,3′,5,5′-tetramethylbenzidine (TMB) [112, 113], and fluorogenic ones include tyramine hydrochloride (T-HCl) [114], Scopoletin (Sc) [115], and Amplex Red (AR) [116, 117]. The optical signal switch of oxidizable substrate makes G4zyme a popular label-free signal actuator for biosensing and DNA logic computing.

Wang and Dong's group [118] found a monomolecular G-rich strand, **PW17**, as a K^+-dependent G4zyme. It exhibits high hemin-binding affinity and peroxidase activity in the presence of K^+ while no activity with Pb^{2+}, since K^+ can induce **PW17** into parallel G-quadruplex while Pb^{2+} to antiparallel one. Based on it, they employed K^+ and Pb^{2+} as two inputs to modulate G4zyme activity and vary the color of ABTS to operate INHIBIT and IMPLICATION functions. Afterward, their group undertook a series of works in this field. They respectively applied fluorogenic substrates T-HCl, Sc, and AR to construct a series of label-free logic gates [36], as well as a couple of advanced non-arithmetic circuits, like EC and DC [115]. Based on these works, they realized the cascade of functional logic devices for the first time that would be applied to logic-programmed label-free ratiometric detection and length measurement of target DNA.

Then, they further constructed a series of logic systems with novel advanced functions, such as voter [119], parity generator/checker [120–122], contrary logic pairs [117], and so on. Take the parity generator/checker as an example (Figure 8.6). By inputting predesigned DNA strands, the initial DNA strand (**TP**) could either be well complemented into duplex, presenting normal state, or form intermolecular G-quadruplex, presenting error state. By employing G4zyme system (hemin, TMB, and H_2O_2) or NMM, the error-state G-quadruplex structure could generate visual or fluorescence signal, respectively. Besides the verification function, the error state of the system could also be corrected by adding glutathione (GSH) that can reduce blue colored TMB^+ to colorless TMB or Cu^{2+} that can dissociate G-quadruplex and quench the fluorescence of NMM. Furthermore, this system could execute multi-input triggered concatenated logic computations with dual output modes, which largely fulfilled the requirements of complicated computing.

Chen's group [123] developed a sensing platform by combining DNAzyme with endonuclease activity and G4zyme with peroxidase activity, which could realize a complete set of Boolean logic gates and cascaded circuits. As shown in Figure 8.7, they designed several strands that could assemble into Mg^{2+}-dependent DNAzyme units synergistically stabilized by certain input(s) and a hairpin containing G-rich sequence. The activated DNAzyme could cut
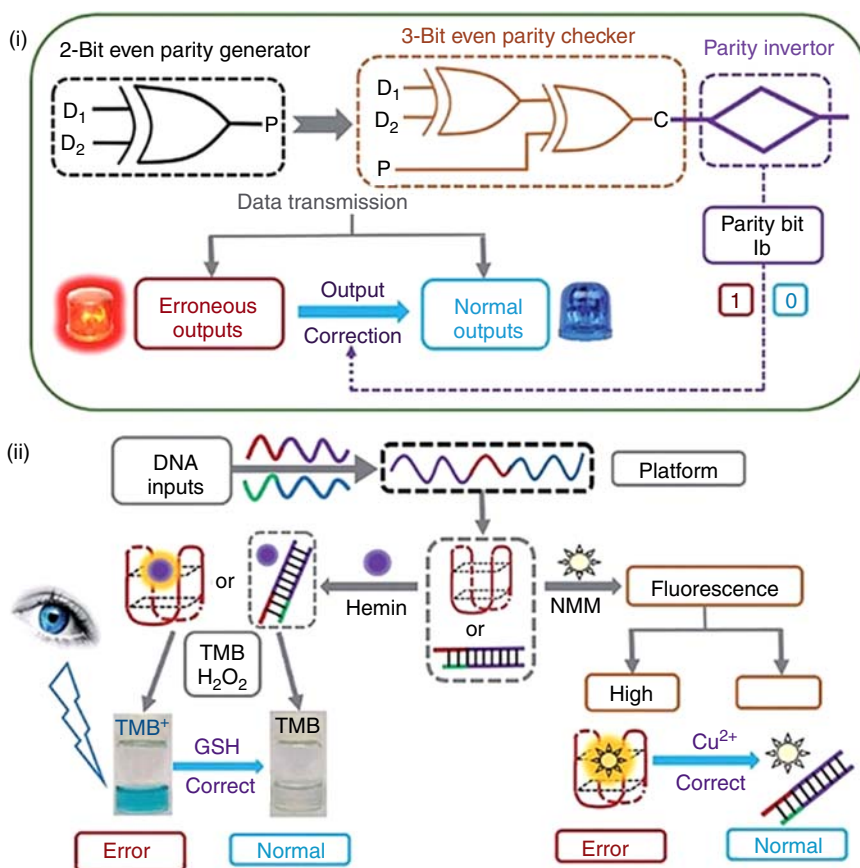
Figure 8.6 Schematic representation of the parity generator/checker based on either G-quadruplex ligand NMM or G4zyme system. (i) The 2-bit even parity generator and 3-bit even parity checker for error detection through data transmission with the "output-correction" function. (ii) The visual outputs (color changes of TMB) and fluorescence signals (fluorescence of NMM) of the parity checking system using different DNA inputs modulated the formation of G-quadruplex (red structure) and the correction of visual/fluorescence erroneous outputs by GSH and Cu^{2+}, respectively. Source: From Fan et al. [120]. Reproduced with the permission of Royal Society of Chemistry.

off the hairpin, release G-rich sequence to form G4zyme with hemin, and consequently catalyzed the oxidation of TMB by H_2O_2 to generate visible readout signals. Besides the basic logic gates, they subsequently connected the XOR and AND gates in series and the XOR and NOR gates in parallel. This strategy is simple in design and economic in operation, showing good potential in constructing versatile logic circuits.

8.1.2.2 Signal with the Help of i-Motif

Another tetra-stranded structure besides G-quadruplex is known as i-motif, which is composed of two parallel stranded cytosine (C)-rich duplexes hydrogen bonded together in an antiparallel orientation by C^+–C base pair intercalation
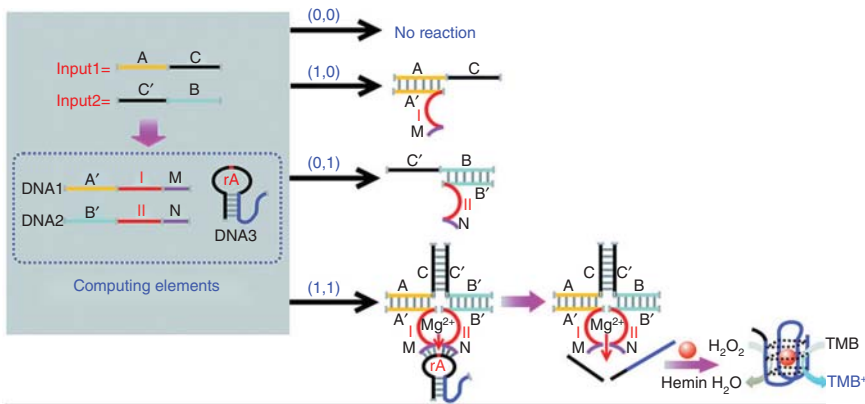
Figure 8.7 Schematic representation of the AND logic gate that consists of the DNAzyme subunits (**DNA1** and **DNA2**), the substrate (hairpin **DNA3**, the caged G-quadruplex sequence in the stem structure of the hairpin is indicated in blue), and the input DNA. Source: From Chen et al. [123]. Licensed under CC by 3.0.

[124]. i-Motif used to be considered inexistence under physiological environment until Daniel's and Marcel's groups recently verified its formation in the nuclei of human cells and the regulatory regions of the human genome, including promoters and telomeric regions [34]. Though the applications of i-motif is much less than G-quadruplex, more and more researches are adopting i-motif DNA in logic device development [125–127], owing to its physical properties, structural mechanism, and especially its potential function as an anticancer drug target.

i-Motif structure can be induced and consolidated by a slightly acidic pH (protonation of the N3 in cytosine) or Ag^+ at neutral pH (formation of $C–Ag^+–C$) so that basic logic gates could be easily operated with inputting pH modulated factor or Ag^+ affinity agents, such as cysteine or halide ions. Similar to G-quadruplex, many fluorescent dyes have been reported that have the capability of binding i-motif with enhanced luminescent signals, such as $[Ru(phen)_2(dppz)]^{2+}$ [128], crystal violet [129], ThT [130], cyanine dye 2,2'-diethyl-9-methylselenacarbocyanine bromide (DMSB) [131], thiazole orange (TO) [132, 133], and berberine [134], which provide i-motif-based circuits detectable outputs [135–137]. For instance, Tang and coworkers [138] designed a C-rich strand to realize OR and INHIBIT functions by employing Ag^+/H^+ and Ag^+/I^- as the inputs, respectively, and DMSB as the signal actuator. Luo and coworkers [133] took TO as i-motif monitor and Ag^+/cysteine as inputs to operate an INHIBIT gate.

In addition to i-motif solely, some researches also attempt to associate C-rich strands with G-rich ones to take full advantage of various DNA secondary structures, including G–C paired duplex, G-quadruplex, i-motif, and sometimes triplex or other motifs under certain conditions, which broadens the designability and enhances the practicability of diverse logic units [16]. However, such logic systems with association of various structures were mostly reported using G-quadruplex as signal modules rather than i-motif. For instance, Famulok and
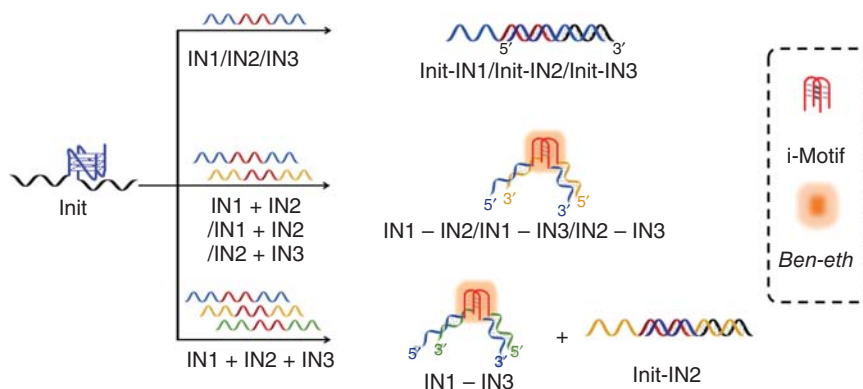
Figure 8.8 Schematic representation of the three-input majority logic gate. Source: From Huang et al. [139]. Reproduced with the permission of John Wiley & Sons.

coworkers [36] designed hybridizing DNA, RNA, and DNA/RNA duplexes with different stabilities that could be unwound to G-quadruplex and i-motif/triplex by K^+ and H^+, respectively. Based on the variation among DNA/RNA structures and their structural conversion under K^+/H^+ conditions, a series of logic functions have been implemented.

Recently, our group designed a totally label-free DNA tetraplex platform by integrating G-quadruplex and i-motif, on which a series of advanced functions were realized, including arithmetic (adders and subtractors) and non-arithmetic ones (majority and dual-transfer gates) [139]. Take the three-input majority logic gate as the example, which employs i-motif structure as the signal module. As shown in Figure 8.8, an initial strand **Init** and three voting strands, **IN1**, **IN2**, and **IN3**, were designed. Each voting strand is composed of a C-rich region (red wavy line) and two flanking regions (blue, yellow, or green wavy lines). The C-rich regions can either complement **Init** or form intermolecular i-motifs with each other, and the flanking ones can complement other inputs in pairs. The flanking regions are designed to be longer than the C-rich sequence so that **IN1**, **IN2**, and **IN3** prefer hybridizing with each other instead of complementing **Init**. In the one voter approved (inputting only one voting strand) case, **Init** would complement with **IN1**, **IN2**, or **IN3** to form duplexes. When two or more voters passed (inputting more than one voting strand simultaneously), the voting strands tended to complement with each other and to form heterogeneous i-motifs. By employing an i-motif-specific probe Ben-eth, the voting results could be indicated by the enhanced fluorescence signal.

It can be observed that the signal strategies on the conversion among various DNA structures can be realized in two ways roughly, labeled and label free. Since the most structures based on Watson–Crick hydrogen bond are linear repeated that lack of specific binding sites, the signal actuators have to be modified on DNA strands covalently to achieve high specificity. On the contrary, in the case of Hoogsteen hydrogen bond-based systems, the unique tetraplex structures can

provide specific binding or active sites, like conjugated stacking plane and limited space enclosed by loops, which could be recognized by small molecular ligands against normal ssDNA/dsDNA. Therefore, label-free probes are commonly employed in these systems.

8.1.3 Signal Derived from Aptamer–Ligand Interaction

Aptamers are oligonucleotides that bind to a specific target molecule, along with unique structural conversion in most cases. One of the most famous aptamer in signal designing is termed Spinach, which can specifically bind to a fluorescent dye, 3,5-difluoro-4-hydroxybenzylidene imidazolinone (DFHBI), and enhance its emission more than 2000 times [41]. Then, an aptamer termed Broccoli with similar function but simpler structure was also reported; it can bind to 3,5-difluoro-4-hydroxybenzylidene-1-trifluoroethyl-imidazolinone (DFHBI-1T) specifically [140]. Since the aptamer–dye systems are nontoxic, are cell membrane permeable, does not interact with cell components, and has a low background [141], they have been widely applied in biochemical analysis, both *in vitro* and *in vivo* [142–145].

Due to the non-covalent interaction between aptamer strand and DFHBI, the system provides another way to generate label-free signal in DNA-based logic circuit construction [15, 146]. However, the related work is quite rare till now.

8.2 Designs for Constructing Multi-output Signals

The abovementioned strategies work well in single-output logic gates. However, one of the huge potential advantages of molecular computer against silicon chip is that the former is functional integrated, which can cut down the intermediate, modularized units and make the whole system performing the expected computing spontaneously. It is of importance for minimizing the size of computer and circumventing the limitation of miniaturization. Therefore, it is trend to design integrated DNA-based circuits that can implement more and more complicated functions in one step, rather than units with fractional functions in series. The advanced circuits with multi-output channels put forward serious challenges not only for the sequence design of DNA processors but also for novel signal actuators and strategies.

To achieve the goal, there are two common ways in signal design: (i) selecting individual signal transducers for each channel – in another word, for each DNA topological structure – and (ii) designing multifunctional probes that can either involve in logic process or produce multiple, distinguished signals simultaneously. In this section, we introduced some multi-output logic systems based in both ways.

8.2.1 Selecting Individual Signal Transducers

In some cases, it is quite easy to construct multi-output circuits by designing individual signals for each channel. Especially in transducer-labeled systems, fluorophores with various emission bands can be selected to construct

non-interfering outputs. For example, Yang et al. [49] selected two fluorophores, i.e. 2-fluoro-N-methyl-N-naphthalen-1-ylacetamide (FAM) ($\lambda_{em} = 520$ nm) and CAL Fluor Orange 560 ($\lambda_{em} = 560$ nm), as the Sum-bit and Carry-bit outputs for the HA. Moreover, some other multi-output circuits were also realized on similar MB-based strategies, such as FA/FS, HA/HS [147, 148], EC/DC [149], and MUX/DEMUX [150].

To enrich the design and simplify the system, nanomaterials like graphene oxide (GO) [151] and gold (Au) surface were also widely used as quenchers in advanced circuits. Cui and coworkers [149] reported a series of multi-output logic devices, including 4-to-2 EC, 8-to-3 EC, and 1-to-2 DC. Especially, GO that can selectively adsorb ssDNA was used to quench three fluorophores, i.e. FAM, ROX, and Cy5, and generate an 8-to-3 EC. Likewise, Au surface that can quench fluorophore was also applied in constructing circuits like FS [48, 49]. However, covalent label would add extra effort and cost to the DNA-based circuits, especially for the multi-labeled systems. And label-free signal actuator is becoming a fashion feature and tendency.

Some works tried to introduce Hoogsteen hydrogen bond-based structures, mainly G-quadruplex, into MB-based systems and achieved partially label-free circuits. A representative work is the universal platform developed by Wang's group, which can implement various advanced functions, including HA, FA, HS, FS, DC, EC, MUX, DEMUX, and so on [50, 152]. Take the 2-to-4 DC as an example, which needs four distinctive output channels. In their system, an MB probe labeled with FAM and Dabcyl, strands labeled by 6-Carboxy-2′,4,4′,5′,7,7′-hexachlorofluorescein, succinimidyl ester (HEX) and pyrene, and G-quadruplex probe NMM were employed as the four outputs, respectively. As shown in Figure 8.9, to implement 2-to-4 DC function, a pyrene-labeled **DC-IN0** and a HEX and pyrene-labeled **DC-IN1** were designed as the inputs, which can complement with each other. In the initial state, an assistant DNA strand (**MBC**) is introduced to hybridize with and open the **MB** probe, accompanied by strong FAM fluorescence. Inputting G-rich **DC-IN0** would hybridize with **MBC** and consequently close **MB** probe, leading to the
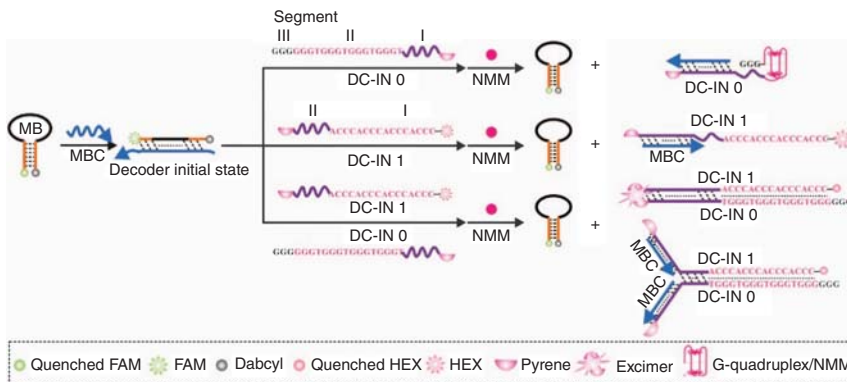
Figure 8.9 Schematic representation of the 2-to-4 DC. Source: From Li et al. [51]. Licensed under CC by 4.0.

decrease of the FAM signal and the production of the NMM signal due to the formation of G-quadruplex. Inputting HEX-labeled **DC-IN1** would play the similar function, to close the FAM signal and to produce HEX signal. Inputting **DC-IN0** and **DC-IN1** together would result in the formation of the **DC-IN0-DC-IN1** complexes, which can produce an obvious pyrene excimer fluorescence signal.

By employing three labeled different dyes and a non-covalent probe, the partial label-free 2-to-4 DC was constructed in a very complicated way. It could be seen that G-quadruplex can only provide one label-free channel. i-Motif can also provide label-free signal by itself; however, most of its probes lack specificity and cannot tell i-motif from G-quadruplex or duplex [153], making that i-motif structure was indeed involved in some advanced circuits but usually did not produce output signal directly. Our group screened a specific probe for i-motif, Ben-eth, and realized several advanced logic functions with simple design [139]. This work provides a possible strategy of designing total label-free logic systems via integrating G-quadruplex and i-motif.

To achieve totally label-free system, fluorescence nanoparticles, like AgNCs [151, 154] and CuNPs, are introduced in some works.

By integrating CuNPs and SYBR Green I (SG), Dong's group [155, 156] succeeded in realizing some DNA-based logic circuits under fully label-free condition for the first time, including a series of Boolean logic gates, 4-to-2 EC, 1-to-2 DC, and 1-to-2 DEMUX. As shown in Figure 8.10, CuNPs could be encapsulated by polyT ssDNA and produce strong fluorescence, while SG is a commercial fluorescence dye that prefer binding dsDNA to ssDNA. And two complementary
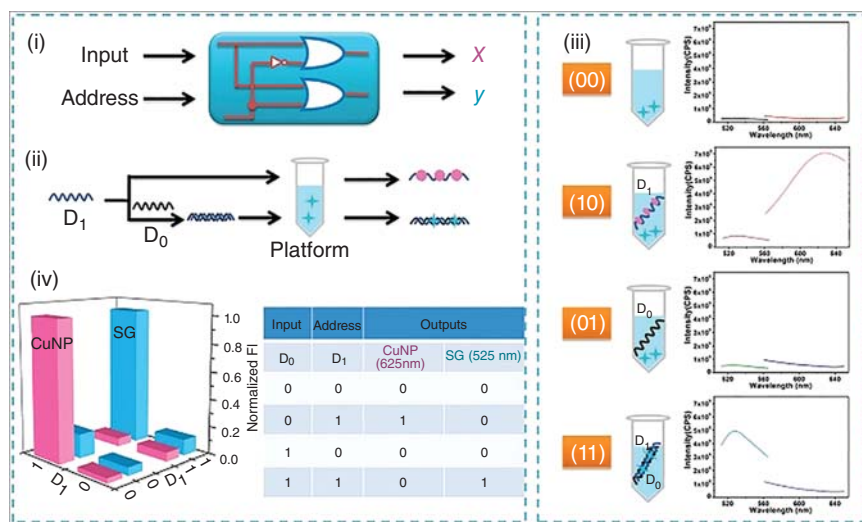
Figure 8.10 The electronic diagram (i) and the schematic representation (ii) of the fully label-free 1-to-2 DEMUX. (iii) Fluorescence spectra of CuNPs and SG under various logic operations. (iv) The normalized fluorescence intensities of fluorescent CuNPs at 625 nm and SG at 525 nm as functions of various inputs. (v) The truth table of the 1-to-2 DEMUX. Source: From Wu et al. [155]. Reproduced with the permission of Royal Society of Chemistry.
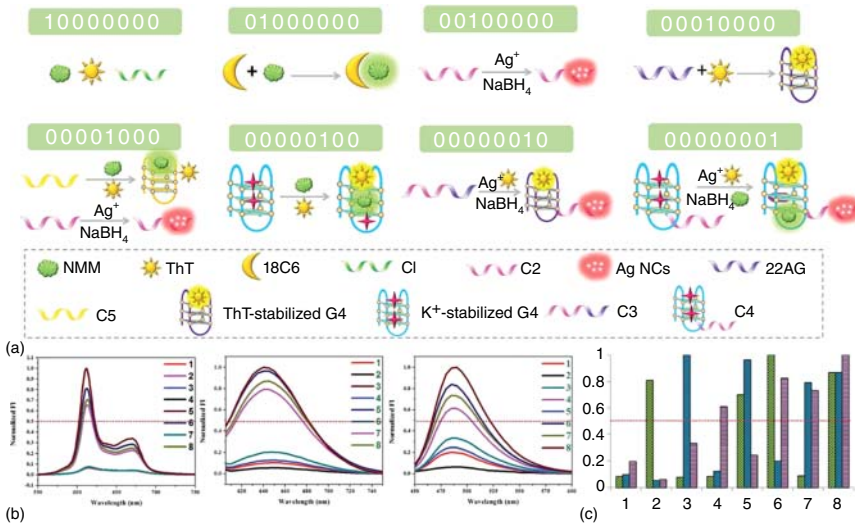
Figure 8.11 (a) Schematic representation of the label-free 8-to-3 encoder. (b) Normalized fluorescence spectra of NMM, AgNCs, and ThT with different combinations of inputs. The red dashed line shows the threshold (0.5). (c) Column diagram of the normalized fluorescence intensities of NMM, AgNCs, and ThT. Source: From Gao et al. [157]. Reproduced with the permission of Royal Society of Chemistry.

DNA strands, D_0 and D_1, were designed as the inputs. Without D_0, D_1 with polyT sequence could induce the formation of CuNPs and enhanced fluorescence signal around 630 nm, while in the presence of D_0, D_1 was hybridized to dsDNA and the SG signal (around 520 nm) would be lightened. In this case, D_0 functions like a single-pole double throw switch. The output signals of the system can be switched between the two fluorescent channels by the presence/absence of D_0, which can mimic the 1-to-2 DEMUX function.

Yao and coworkers [157] then reported a series of multi-output logic circuits by employing three label-free fluorescent probes, among which NMM and ThT can bind on certain G-quadruplexes formed by different sequences and AgNCs can be induced by C-rich domains. By inputting various DNAs or crown ether, an 8-to-3 EC was firstly realized on a completely label-free system (Figure 8.11). Moreover, a parity checker identifying decimal odd and even numbers was also constructed, presenting great scalability of the system. The work opened a new era for the construction of totally label-free multi-output circuits and showed excellent potential in medical diagnosis.

8.2.2 Designing Multifunctional Probes

Most of the reported DNA logic systems follow the strategy of designing DNA processors and signal actuators individually. However, it is and will be a challenge to select parallel transducers with increasing complexity of circuits. For example, the DNA-based platform designed by Wang's group [51] employed four different fluorescent transducers (FAM, NMM, HEX, and pyrene-based excimer) with
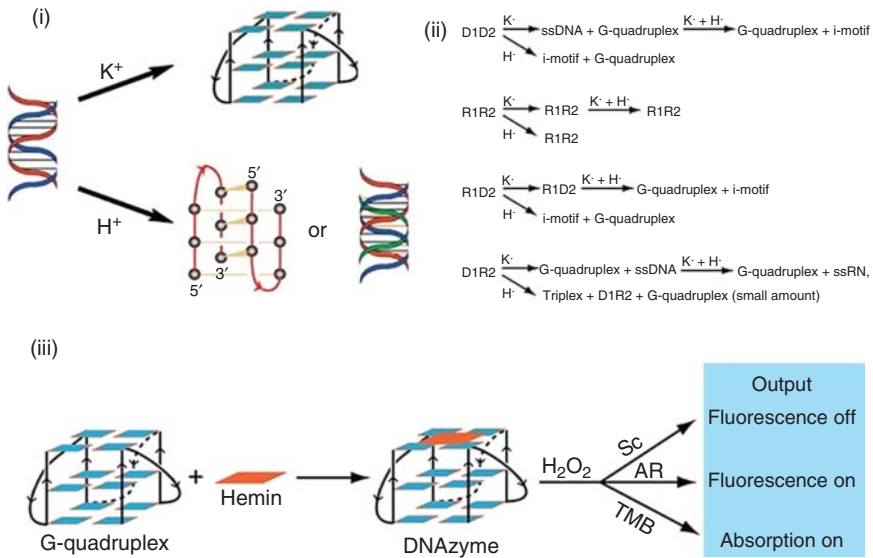
Figure 8.12 (i) Schematic representation of DNA structural conversions induced by K^+ and H^+. (ii) Overview of the structural conversion of nucleic acid helices triggered by K^+ and H^+. (iii) The signal output of the logic system built on G4zyme. DNA or RNA G-quadruplex with hemin exhibits peroxidase activity in the presence of substrates Sc, AR, and TMB, resulting in a change in the readout signal. Source: Based on Xie et al. [36].

partially overlapped signals to construct a 2-to-4 DC, leading to an underperforming on-to-off ratio. Predictably, the higher complexity of the logic functions, the more difficult is to find enough compatible fluorophores.

The idea about designing multifunctional probes that can not only transduce fluorescent signals but also serve as the logic processor combined with DNA may offer a possible way out for designing advanced logic circuits. Some works introduced above already indicated the feasibility of this idea. For example, Famulok and coworkers [36] constructed a logic system with three G4zyme substrates that have different signal patterns (Figure 8.12). Sc is a signal turn-off probe whose fluorescence can be quenched by either G4zyme or acids, AR is a signal turn-on probe whose fluorescence can be induced by G4zyme but quenched by acids, and TMB is a colorimetric probe that can indicate G4zyme without the disturbances of K^+ and H^+. Based on the structural convention of DNA/RNA dsDNAs and the optical properties of the three substrates under K^+/H^+ conditions, they realized several logic gates simultaneously. In this case, the three substrates not only indicated the changes of DNA/RNA structure but also reflected the input stimuli directly, acting as both signal actuator and logic processor.

Besides multi-role molecules, probes that can produce versatile signal channels simultaneously are another kind of ideal candidates for multi-output circuits. DNA-templated AgNCs, which exhibit robust and size-dependent fluorescence emission, have been widely employed as a new type of fluorophore in the fields of biosensing [158] and logic operating [159, 160]. Their ability to emit tunable
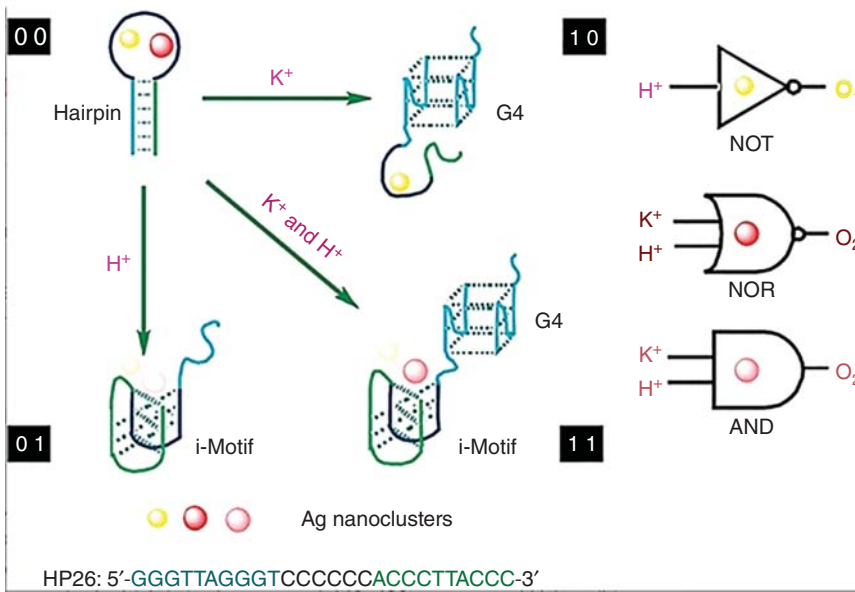
Figure 8.13 Schematic representation of the logic operations based on **HP26**-tuned fluorescent AgNCs. K^+ and H^+ serve as two inputs to trigger the allosterism of **HP26** and modulate the fluorescence output. Source: From Li et al. [161]. Reproduced with the permission of American Chemical Society.

fluorescence can contribute to the development of DNA-based circuits with simple content. Wang and coworkers [161] designed a C/G-rich hairpin **HP26** that could be unwound by K^+/H^+ and form G-quadruplex/i-motif, respectively. As shown in Figure 8.13, **HP26** itself could induce several kinds of AgNCs, emitting fluorescence at 570, 601, and 646 nm, respectively. Inputting K^+ facilitates the formation of G-quadruplex, and the flanking segment could only induce 570 nm signal. However the i-motif with H^+ could hardly induce any fluorescent signal. In the addition of K^+ and H^+ together, **HP26** could form G-quadruplex and i-motif simultaneously, and the fluorescence of AgNCs was tuned to 601 nm. In this system, one kind of material, Ag, was induced to three kinds of AgNCs with various sizes and distinguished signals by different DNA structures, which enables two or more logic operations to be performed together via multichannel fluorescence output.

Recently, our group introduced supramolecular assembly in DNA-based circuit construction as the probe with both multi-role and multichannel properties. We designed a cyanine dye MTC, which could present several assembly behaviors under different conditions. In alkaline buffer solutions (pH 8.5), MTC has a dominating absorption peak at 528 nm assigned to dimer (termed D-band), while in acidic conditions (pH 4) it self-assembles to J-aggregates (whose typical absorption peak is around 655 nm, termed J-band). Some cations can also facilitate the self-assembly of MTC. For example, MTC can assemble to either J- or H-aggregates (whose absorption peak is in the range of 440–480 nm,
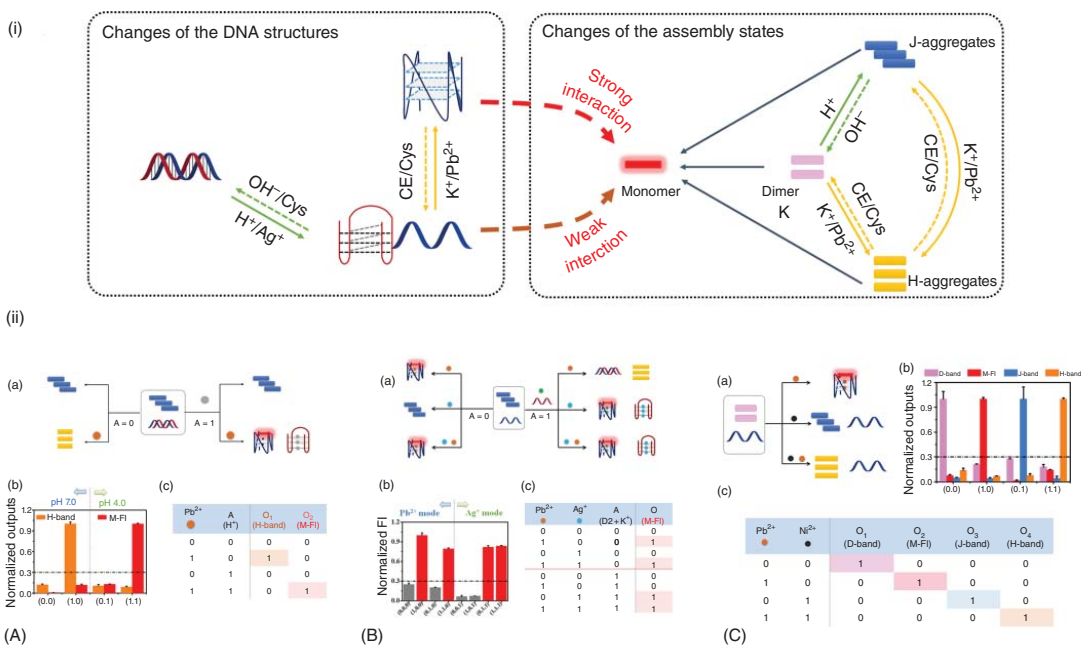
Figure 8.14 (i) Schematic representation of the DNA-MTC supramolecular logic platform. The changes of the structures of DNA and the assembly states of MTC can be triggered by various inputs simultaneously. Source: Based on Yang et al. [92]. (ii) Schematic illustration (a), the normalized output signals (b), and the truth table (c) of the implementations of the logic operations: (A) the 1-to-2 DEMUX, (B) the 2-to-1 MUX, and (C) the 2-to-4 DC. Source: From Yang et al. [109]. Reproduced with the permission of American Chemical Society.

termed H-band) in the presence of potassium. Besides, we previously reported that certain DNA G-quadruplex structure could specifically disassemble MTC aggregates into monomer, accompanying with a dramatic enhancement of the fluorescence (whose maximum emission wavelength is around 600 nm, termed M-Fl) [160]. The unique assembly behavior of MTC that can provide at least four distinguishable output signals (the D-band, J-band, H-band, and M-Fl) makes it convenient to construct multi-output circuits.

Firstly, by combining the DNA structure changes with supramolecular assembly, we developed a versatile DNA-MTC supramolecular logic platform (Figure 8.14i). Owing to the multiple assembly states of MTC, the platform contains only one signal block but can provide multiple parallel outputs and easily implement several types of information processing functions, including data filtration (the binary and ternary INHIBIT gates), selection (the MUX/DEMUX), and verification (the parity checker and the comparator). In addition to the combinational circuits, a fundamental sequential logic circuit, namely, the counter, also was fabricated at the molecular level [92]. The logic platform not only performs various intelligent logic functions but also shows potential applications in multiplex chemical analysis and clinical diagnosis. Based on the platform, we designed three prototypes of advanced circuits and developed related intelligent sensors of Pb in different contexts (Figure 8.14ii) [109]. The DEMUX that can split signal flow was used to determine Pb^{2+} in different pH conditions, the MUX that can alternate signal channels was applied to detect Pb^{2+} or Ag^+ selectively, and the DC that can extract information was utilized to test Pb^{2+} and the coexisted Ni^{2+} simultaneously. All the three sensors present practicable sensitivities and specificities, indicating their excellent potential in environmental monitoring, biochemical analysis, and medical diagnosis.

Then, we constructed a reconfigurable logic unit by employing cyanine probes as part of the logic processor [162]. As shown in Figure 8.15, we combined DNA and cyanine dye 3,3′-di(3-sulfopropyl)-4,5,4′,5′-dibenzo-9-methylthiacarbocyanine triethylammonium salt (DMT) together as the logic processor. For the DNA processor, two complementary DNA strands were designed: G-rich **D1** that can form a parallel G-quadruplex in the presence of K^+ and C-rich **D2** that can be folded into i-motif structure in low pH condition. Via base sequence design, the stabilities of the structures were adjusted as follows: **D2** i-motif > **D1D2** complex > **D1** G-quadruplex. Therefore, inputting H^+, K^+, or both could induce the strands into **D2** i-motif, **D1D2** complex, or both **D1** G-quadruplex and **D2** i-motif, respectively, and consequently, implement certain logic function. On the other hand, K^+ also can induce DMT self-assembling to three kinds of aggregates (pure J- or H-ones, or a mixture of both) in different concentrations. Changing the concentration of inputted K^+ would further influence the output signals and reconfigure the unit to other modes. In the presence of low K^+ concentration (10 mM), DMT prefers forming J-aggregates, and an HA could be realized by choosing J-band and Fl-M as the output. Increasing K^+ concentration (to 30 mM), DMT would be induced to the mixture of J- and H-aggregates, and the unit is switched to perform HS function. Further increase K^+ concentration (to 60 mM), DMT would be induced to pure H-aggregate. In this case, the
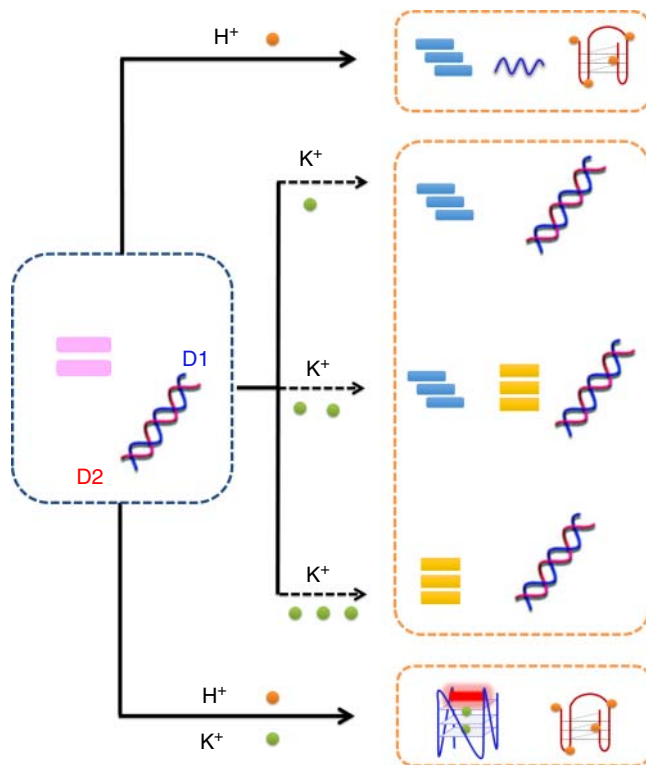
Figure 8.15 Schematic representation of the reconfigurable DNA-supramolecular logic unit. Source: From Yang et al. [162]. Reproduced with the permission of John Wiley & Sons.

four kinds of DMT assembly states – the monomer, dimer, J-aggregates, and H-aggregates – can be totally utilized, and a 2-to-4 DC is constructed. Therefore, the DNA-supramolecular logic unit has three functional modes, which can be reconfigured simply by adjusting the inputted K^+ concentration. These kinds of logic units with multiple functions show great potential in developing devices with faster operating and wider scope, owing to their versatile and reconfigurable ability.

On the similar strategy, we also realized a serial of scalable ECs under the input stimuli of various metal ions [163]. Though study the regulating effect of metal ions to the assembly behavior of cyanine dye MTC adequately, we selected nine cations that could induce MTC into various aggregations. Specifically, these cations can regulate MTC to three kinds of aggregates with distinct absorption features: J_a-aggregates (absorption peak at 660 nm, termed J_a-band) induced by Co^{2+}, J_b-aggregates (absorption peak around 620 nm, termed J_b-band) induced by Cu^{2+} or Mg^{2+}, and H-aggregates (absorption band around 440–480 nm, termed H-band) induced by Pb^{2+}. Moreover, MTC can also be induced to the mixtures of various aggregates: J_a/J_b-aggregates induced by Zn^{2+} or Ni^{2+}, J_b/H-aggregates induced by Mn^{2+}, and J_a/J_b/H-aggregates induced by Ca^{2+}. By utilizing the effects of these cations on both MTC assembly behavior and DNA

structural transformation, a serial of multiple input/output EC prototypes were realized, including 4-to-2, 7-to-3, and even 14-to-4 ones.

All these works prove that DNA-supramolecular platform possess high reconfigurability, flexibility, scalability, and enormous parallelism, setting up a promising future in the field of molecular computing and multiplex biochemical analysis.

8.3 Summary and Outlook

DNA computing has the potential to execute orders of magnitude more powerful functions than traditional silicon circuitry and has a wide range of applications in medical diagnosis [164], *in situ* analysis [165], and artificial intelligence. However, since nucleic acids cannot present detectable signals by themselves, designing suitable signal modules is becoming a crucial issue, as important as the design of the DNA sequence. Till now, there are numerous signal strategies, especially fluorescent ones, that have been proposed to realize both basic DNA gates and advanced DNA circuits.

In addition to pre-labeled fluorophores that only indicate the state of DNA processor, non-covalent probes show more significant advantages in DNA-based logic system. Some multifunctional probes have been designed to either participate in logic process or produce multiple signal channels simultaneously, which enriches the realizable logic operation and practical application.

So far, DNA computing is still in its infancy. Most of the reported works are conceptional, with isolated logic functions and limited applications. Aside from core DNA scaffolds or reactions, the design of signal transducers and modules is equally important in an ideal logic circuit. In our opinion, the breakthrough of novel fluorescence signal system in the following ways might promote the development of DNA computing:

Homogeneous and label-free signal actuators. Considering time and economic costs, label-free fluorescent probes may have huge advantage over the pre-labeled ones. Following the track of development, label-free is becoming a more and more important consideration in DNA logic construction. Although total label-free logic circuits have been reported recently [151, 154], they all introduced heterogeneous components somehow, such as GO and AgNCs. These insoluble materials would not only increase the complexity of DNA-based logic system but also give rise to some practical problems, such as preservation issue, potential biotoxicity, and so on. As mentioned above, DNA logic circuits not only are the foundation of molecular calculating but also can be applied to biochemical intelligent analysis directly due to its excellent biocompatibility [166]. In this sense, homogeneous system may have better application prospect.

Probes that can recognize G-quadruplexes specifically are ideal candidates for constructing homogeneous and label-free logic circuits. But it is hard to design G-quadruplexes with different subtypes in a multi-output system, and only few works have attempted in this way [157]. As mentioned, i-motif has similar spatial property as G-quadruplex and has been involved in some logic systems. But

few multi-output circuits designed signals directly on the structural conversion of i-motif till now, owing to the lack of high-specific i-motif probe [153]. Therefore, designing probes targeting i-motif specifically can make the most of C-rich and G-rich DNA and might be a possible solution for constructing label-free multi-output circuit.

Another possible way is to utilize the specific interaction between certain aptamers and their fluorescent ligands, such as Spinach [41] and Broccoli [140] systems, which can provide more flexibility in DNA/RNA sequence design. However, the usable fluorescent dyes in this strategy are very limited, mostly DFHBI and its analogs, and their emission ranges center around 510 nm. The nonoptional emission property limits the selection of parallel fluorophores in multi-output circuits.

Reversible signal modules. To adapt practical application better, DNA circuits should contain a reset mechanism, by which they can be shut down and reoperated easily as required [167]. Besides, the recycle of the circuits can save resources and shrink the size of complicated device. Correspondingly, the signal modules should also be reversible. The strategies based on the cleavage of DNA strands and redox reaction of fluorophore groups may be not suitable, and further design thoughts are required in this area.

Probes with multi-role and multichannel. As mentioned, some extrinsic components in DNA logic circuits are playing roles in both indicating the change of DNA structure and taking part in judging and processing information [36, 92]. And probes with multiple parallel signal channels can assist in integrating basic logics and increase the efficiency of the system [161]. These probes can definitely enrich the realizable function and improve the computing ability.

Another point is that both the combinatorial and sequential logics are of importance in modern electronics; however, the most reported DNA circuits focus mainly on the combinatorial one. Sequential logic units like accumulator, register, and timer are widely used in modern computer and play fundamental roles in frequency division, timing, and arithmetic operations, but the only reported sequential logic units are keypad lock [168–170] and flip-flop [157]. Taking the simplest counter, which is used to count the stimulus pulses, as an example, a logic system must fulfill the following requirements: (i) multichannel compatibility, (ii) transformations among different output channels triggered by the same stimulus with different amounts, and (iii) availability of chemical or biochemical pairs with opposite actions to the system. These requirements are so challenging to be satisfied simultaneously that the sequential functions are hard to realize only depending on DNA structural conversions and reactions. Novel probes with multichannel signals and pulse/time-dependent properties can satisfy partial requirements of the sequential logic and possibly endow new signal patterns on DNA system [92]. After addressing these issues, it is expected that logic circuits based on DNA, as well as versatile signal modules, will facilitate the development of next-generation computer and find more application in intelligent analysis.

References

1 Sugita, M. (1961). *J. Theor. Biol.* 1: 415–430.

2 Adleman, L.M. (1994). *Science* 266: 1021–1024.

3 Lipton, R.J. (1995). *Science* 268: 542–545.

4 Faulhammer, D., Cukras, A.R., Lipton, R.J., and Landweber, L.F. (2000). *Proc. Natl. Acad. Sci. U.S.A.* 97: 1385–1389.

5 Han, D., Kang, H., Zhang, T. et al. (2014). *Chem. Eur. J.* 20: 5866–5873.

6 Benenson, Y. (2012). *Nat. Rev. Genet.* 13: 455–468.

7 Harroun, S.G., Prevost-Tremblay, C., Lauzon, D. et al. (2018). *Nanoscale* 10: 4607–4641.

8 Hemphill, J. and Deiters, A. (2013). *J. Am. Chem. Soc.* 135: 10512–10518.

9 Lienert, F., Torella, J.P., Chen, J.H. et al. (2013). *Nucleic Acids Res.* 41: 9967–9975.

10 Konry, T. and Walt, D.R. (2013). *J. Am. Chem. Soc.* 131: 13232–13233.

11 Jung, C. and Ellington, A.D. (2014). *Acc. Chem. Res.* 47: 1825–1835.

12 Genot, A.J., Bath, J., and Turberfield, A.J. (2011). *J. Am. Chem. Soc.* 133: 20080–20083.

13 Gerasimova, Y.V. and Kolpashchikov, D.M. (2012). *Chem. Asian J.* 7: 534–540.

14 Willner, I., Shlyahovsky, B., Zayats, M., and Willner, B. (2008). *Chem. Soc. Rev.* 37: 1153–1165.

15 Kikuchi, N. and Kolpashchikov, D.M. (2017). *Chem. Commun.* 53: 4977–4980.

16 Du, Y., Peng, P., and Li, T. (2018). *Chem. Commun.* 54: 6132–6135.

17 Zhang, D.Y. and Seelig, G. (2011). *Nat. Chem.* 3: 103–113.

18 Wilner, O.I. and Willner, I. (2012). *Chem. Rev.* 112: 2528–2556.

19 Ma, D.L., He, H.Z., Leung, K.H. et al. (2013). *Chem. Soc. Rev.* 42: 3427–3440.

20 Liu, X.Q., Aizen, R., Freeman, R. et al. (2012). *ACS Nano* 6: 3553–3563.

21 Miyoshi, D., Inoue, M., and Sugimoto, N. (2006). *Angew. Chem. Int. Ed.* 45: 7716–7719.

22 Lin, Y.H., Tao, Y., Pu, F. et al. (2011). *Adv. Funct. Mater.* 21: 4565–4572.

23 Kang, D., White, R.J., Xia, F. et al. (2012). *NPG Asia Mater.* 4: e1.

24 Xia, F., Zuo, X.L., Yang, R.Q. et al. (2010). *J. Am. Chem. Soc.* 132: 8557–8559.

25 Xiang, Y., Qian, X.Q., Chen, Y. et al. (2011). *Chem. Commun.* 47: 2080–2082.

26 Zhu, X., Xu, H.F., Gao, X.Y. et al. (2011). *Chem. Commun.* 47: 9080–9082.

27 Siuti, P., Yazbek, J., and Lu, T.K. (2013). *Nat. Biotechnol.* 31: 448–453.

28 Bonnet, J., Subsoontorn, P., and Endy, D. (2012). *Proc. Natl. Acad. Sci. U.S.A.* 109: 8884–8889.

29 Daniel, R., Rubens, J.R., Sarpeshkar, R., and Lu, T.K. (2013). *Nature* 497: 619–623.

30 Zhang, S., Sun, H., Wang, L. et al. (2018). *Nucleic Acids Res.* 64: 7522–7532.

31 Burge, S., Parkinson, G.N., Hazel, P. et al. (2006). *Nucleic Acids Res.* 34: 5402–5415.

32 Wang, Y. and Patel, D.J. (1993). *J. Mol. Biol.* 234: 1171–1183.

33 Leroy, J.L. (2009). *Nucleic Acids Res.* 37: 4127–4134.

34 Zeraati, M., Langley, D.B., Schofield, P. et al. (2018). *Nat. Chem.* 10: 631–637.

35 Cheng, E., Xing, Y., Chen, P. et al. (2009). *Angew. Chem. Int. Ed.* 48: 7660–7663.

36 Li, T., Ackermann, D., Hall, A.M., Famulok, M. (2012). *J. Am. Chem. Soc.* 134: 3508–3516.

37 Liu, S.N., Peng, P., Wang, H.H. et al. (2017). *Nucleic Acids Res.* 45: 12080–12089.

38 Zheng, J., Jiao, A.L., Yang, R.H. et al. (2012). *J. Am. Chem. Soc.* 134: 19957–19960.

39 Torigoe, H., Ono, A., and Kozasa, T. (2010). *Chem. Eur. J.* 16: 13218–13225.

40 Huizenga, D.E. and Szostak, J.W. (1995). *Biochemistry* 34: 656–665.

41 Paige, J.S., Wu, K.Y., and Jaffrey, S.R. (2011). *Science* 333: 642–646.

42 Tyagi, S. and Kramer, F.R. (1996). *Nat. Biotechnol.* 14: 303–308.

43 Wang, K., Tang, Z., Yang, C.J. et al. (2009). *Angew. Chem. Int. Ed.* 48: 856–870.

44 Zheng, J., Yang, R., Shi, M. et al. (2015). *Chem. Soc. Rev.* 44: 3036–3055.

45 Li, D., Song, S.P., and Fan, C.H. (2010). *Acc. Chem. Res.* 43: 631–641.

46 O'Steen, M.R., Cornett, E.M., and Kolpashchikov, D.M. (2015). *Chem. Commun.* 51: 1429–1431.

47 Park, K.S., Seo, M.W., Jung, C. et al. (2012). *Small* 8: 2203–2212, 2129.

48 Lin, H.Y., Chen, J.Z., Li, H.Y., and Yang, C.N. (2015). *Sci. Rep.* 5: 10686.

49 Yang, C.N., Hsu, C.Y., and Chuang, Y.C. (2012). *Chem. Commun.* 48: 112–114.

50 Li, H., Guo, S., Liu, Q. et al. (2015). *Adv. Sci.* 2: 1500054.

51 Li, H.L., Liu, Y.Q., Dong, S.J., and Wang, E.K. (2015). *NPG Asia Mater.* 7: e166.

52 Campbell, E.A., Peterson, E., and Kolpashchikov, D.M. (2017). *ChemPhysChem* 18: 1730–1734.

53 Gerasimova, Y.V. and Kolpashchikov, D.M. (2016). *Angew. Chem. Int. Ed.* 55: 10244–10247.

54 Katz, E. and Privman, V. (2010). *Chem. Soc. Rev.* 39: 1835–1857.

55 Lilienthal, S., Klein, M., Orbach, R. et al. (2017). *Chem. Sci.* 8: 2161–2168.

56 Orbach, R., Remacle, F., Levine, R.D., and Willner, I. (2014). *Chem. Sci.* 5: 1074–1081.

57 Gerasimova, Y.V. and Kolpashchikov, D.M. (2015). *Chem. Commun.* 51: 870–872.

58 Stojanovic, M.N., Stefanovic, D., and Rudchenko, S. (2014). *Acc. Chem. Res.* 47: 1845–1852.

59 Stojanovic, M.N. and Stefanovic, D. (2003). *Nat. Biotechnol.* 21: 1069–1074.

60 Macdonald, J., Li, Y., Sutovic, M. et al. (2006). *Nano Lett.* 6: 2598–2603.

61 Pei, R., Matamoros, E., Liu, M. et al. (2010). *Nat. Nanotechnol.* 5: 773–777.

62 Walker, G.T., Fraiser, M.S., Schram, J.L. et al. (1992). *Nucleic Acids Res.* 20: 1691–1696.

63 Li, Q.Q., Luan, G.Y., Guo, Q.P., and Liang, J.X. (2002). *Nucleic Acids Res.* 30: e5.

64 Prokup, A., Hemphill, J., and Deiters, A. (2012). *J. Am. Chem. Soc.* 134: 3810–3815.

65 Seelig, G., Soloveichik, D., Zhang, D.Y., and Winfree, E. (2006). *Science* 314: 1585–1588.

66 Brown, C.W. 3rd, Lakin, M.R., Stefanovic, D., and Graves, S.W. (2014). *ChemBioChem* 15: 950–954.

67 Wang, W., Huang, S., Li, J. et al. (2017). *Chem. Sci.* 8: 174–180.

68 Li, C.P., Li, Z.P., Jia, H.X., and Yan, J.L. (2011). *Chem. Commun.* 47: 2595–2597.

69 Deng, R., Tang, L., Tian, Q. et al. (2014). *Angew. Chem. Int. Ed.* 53: 2389–2393.

70 Hourcade, D., Dressler, D., and Wolfson, J. (1973). *Proc. Natl. Acad. Sci. U.S.A.* 70: 2926–2930.

71 Choi, H.M.T., Chang, J.Y., Trinh, L.A. et al. (2010). *Nat. Biotechnol.* 28: 1208–1212.

72 Ge, Z., Lin, M., Wang, P. et al. (2014). *Anal. Chem.* 86: 2124–2130.

73 Dirks, R.M. and Pierce, N.A. (2004). *Proc. Natl. Acad. Sci. U.S.A.* 101: 15275–15278.

74 Wang, H.M., Li, C.X., Liu, X.Q. et al. (2018). *Chem. Sci.* 9: 5842–5849.

75 Jiang, Y., Li, B.L., Milligan, J.N. et al. (2013). *J. Am. Chem. Soc.* 135: 7430–7433.

76 Li, D.D., Cheng, W., Li, Y.J. et al. (2016). *Anal. Chem.* 88: 7500–7506.

77 Bi, S., Yue, S.Z., Wu, Q., and Ye, J.Y. (2016). *Chem. Commun.* 52: 5455–5458.

78 Morihiro, K., Ankenbruck, N., Lukasak, B., and Deiters, A. (2017). *J. Am. Chem. Soc.* 139: 13909–13915.

79 Wang, S., Sun, J., Zhao, J. et al. (2018). *Anal. Chem.* 90: 3437–3442.

80 Liu, C.W., Huang, C.C., and Chang, H.T. (2009). *Anal. Chem.* 81: 2383–2387.

81 Biffi, G., Tannahill, D., McCafferty, J., and Balasubramanian, S. (2013). *Nat. Chem.* 5: 182–186.

82 Hazel, P., Parkinson, G.N., and Neidle, S. (2006). *Nucleic Acids Res.* 34: 2117–2127.

83 Phan, A.T., Kuryavyi, V., Gaw, H.Y., and Patel, D.J. (2005). *Nat. Chem. Biol.* 1: 167–173.

84 Seenisamy, J., Bashyam, S., Gokhale, V. et al. (2005). *J. Am. Chem. Soc.* 127: 2944–2959.

85 Yang, S., Xiang, J.F., Yang, Q.F. et al. (2010). *Fitoterapia* 81: 1026–1032.

86 Gornall, K.C., Samosorn, S., Tanwirat, B. et al. (2010). *Chem. Commun.* 46: 6602–6604.

87 Yang, Q.F., Xiang, J.F., Yang, S. et al. (2010). *Anal. Chem.* 82: 9135–9137.

88 Gai, W., Yang, Q.F., Xiang, J.F. et al. (2013). *Nucleic Acids Res.* 41: 2709–2722.

89 Yang, Q.F., Xiang, J.F., Yang, S. et al. (2009). *Chem. Commun.*: 1103–1105.

90 He, H.Z., Chan, D.S.H., Leung, C.H., and Ma, D.L. (2013). *Nucleic Acids Res.* 41: 4345–4359.

91 Hu, D., Huang, Z.Z., Pu, F. et al. (2011). *Chem. Eur. J.* 17: 1635–1641.

92 Yang, C., Song, L., Chen, J. et al. (2018). *NPG Asia Mater.* 10: 497–508.

93 Liu, Z.L., Luo, X.Y., Li, Z. et al. (2017). *Anal. Chem.* 89: 1892–1899.

94 Wang, H.X., Li, Y., Zhao, K.L. et al. (2017). *Biosens. Bioelectron.* 91: 400–407.

95 Hardin, C.C., Perry, A.G., and White, K. (2000). *Biopolymers* 56: 147–194.

96 Guedin, A., Gros, J., Alberti, P., and Mergny, J.L. (2010). *Nucleic Acids Res.* 38: 7858–7868.

97 Bugaut, A. and Balasubramanian, S. (2008). *Biochemistry* 47: 689–697.

98 Patel, D.J., Phan, A.T., and Kuryavyi, V. (2007). *Nucleic Acids Res.* 35: 7429–7455.

99 Bochman, M.L., Paeschke, K., and Zakian, V.A. (2012). *Nat. Rev. Genet.* 13: 770–780.

100 Yu, Z., Zhou, W., Han, J. et al. (2016). *Anal. Chem.* 88: 9375–9380.

101 Shen, G., Zhang, H., Yang, C. et al. (2017). *Anal. Chem.* 89: 548–551.

102 Tang, L., Liu, Y., Ali, M.M. et al. (2012). *Anal. Chem.* 84: 4711–4717.

103 Yang, S., Xiang, J.F., Yang, Q.F. et al. (2010). *Chin. J. Chem.* 28: 771–780.

104 Guo, Y.H., Yao, W.R., Xie, Y.F. et al. (2016). *Microchim. Acta* 183: 21–34.

105 Guo, Y., Zhou, L., Xu, L. et al. (2014). *Sci. Rep.* 4: 7315.

106 Bader, A. and Cockroft, S.L. (2018). *Chem. Eur. J.* 24: 4820–4824.

107 Li, T., Wang, E.K., and Dong, S.J. (2010). *Anal. Chem.* 82: 7576–7580.

108 Zhang, L., Zhu, J., Ai, J. et al. (2013). *Biosens. Bioelectron.* 39: 268–273.

109 Yang, C., Yang, S., Li, J. et al. (2018). *Anal. Chem.* 90: 10585–10590.

110 Li, T., Dong, S.J., and Wang, E. (2009). *Anal. Chem.* 81: 2144–2149.

111 Li, T., Wang, E., and Dong, S. (2010). *Anal. Chem.* 82: 1515–1520.

112 Li, T., Li, B.L., Wang, E.K., and Dong, S.J. (2009). *Chem. Commun.*: 3551–3553.

113 Du, Y., Li, B.L., Guo, S.J. et al. (2011). *Analyst* 136: 493–497.

114 Cai, Y., Li, N., Kong, D.M., and Shen, H.X. (2013). *Biosens. Bioelectron.* 49: 312–317.

115 Fan, D.Q., Zhu, J.B., Zhai, Q.F. et al. (2016). *Chem. Commun.* 52: 3766–3769.

116 Wang, S.R., Fu, B.S., Wang, J.Q. et al. (2014). *Anal. Chem.* 86: 2925–2930.

117 Fan, D.Q., Wang, E.K., and Dong, S.J. (2017). *Mater. Horiz.* 4: 924–931.

118 Li, T., Wang, E., and Dong, S. (2009). *J. Am. Chem. Soc.* 131: 15082–15083.

119 Fan, D., Wang, K., Zhu, J. et al. (2015). *Chem. Sci.* 6: 1973–1978.

120 Fan, D., Wang, E., and Dong, S. (2017). *Chem. Sci.* 8: 1888–1895.

121 Fan, D.Q., Wang, E.K., and Dong, S.J. (2017). *ACS Appl. Mater. Interfaces* 9: 1322–1330.

122 Fan, D., Fan, Y., Wang, E., and Dong, S. (2018). *Chem. Sci.* 9: 6981–6987.

123 Chen, J., Pan, J., and Chen, S. (2018). *Chem. Sci.* 9: 300–306.

124 Kumar, N., Nielsen, J.T., Maiti, S., and Petersen, M. (2007). *Angew. Chem. Int. Ed.* 46: 9220–9222.

125 Hurley, L.H. (2001). *Biochem. Soc. Trans.* 29: 692–696.

126 Kang, H.J., Kendrick, S., Hecht, S.M., and Hurley, L.H. (2014). *J. Am. Chem. Soc.* 136: 4172–4185.

127 Dong, Y., Yang, Z., and Liu, D. (2014). *Acc. Chem. Res.* 47: 1853–1860.

128 Poynton, F.E., Hall, J.P., Keane, P.M. et al. (2016). *Chem. Sci.* 7: 3075–3084.

129 Zhou, J., Amrane, S., Korkut, D.N. et al. (2013). *Angew. Chem. Int. Ed.* 52: 7742–7746.

130 Lee, I.J., Patil, S.P., Fhayli, K. et al. (2015). *Chem. Commun.* 51: 3747–3749.

131 Shi, Y., Sun, H., Xiang, J. et al. (2015). *Anal. Chim. Acta* 857: 79–84.

132 Xu, B.C., Wu, X.Y., Yeow, E.K.L., and Shao, F.W. (2014). *Chem. Commun.* 50: 6402–6405.

133 Kang, B.H., Gao, Z.F., Li, N. et al. (2016). *Talanta* 156: 141–146.

134 Xu, L., Hong, S., Sun, N. et al. (2016). *Chem. Commun.* 52: 179–182.

135 Shi, S., Zhao, J., Geng, X. et al. (2010). *Dalton Trans.* 39: 2490–2493.

136 Ma, D.L., Kwan, M.H., Chan, D.S. et al. (2011). *Analyst* 136: 2692–2696.

137 Gao, R.R., Shi, S., Zhu, Y. et al. (2016). *Chem. Sci.* 7: 1853–1861.

138 Shi, Y., Sun, H., Xiang, J. et al. (2014). *Chem. Commun.* 50: 15385–15388.

139 Huang, D., Yang, C., Yao, Y. et al. (2019). *Chem. Eur. J.* 25: 6996–7003.

140 Filonov, G.S., Moon, J.D., Svensen, N., and Jaffrey, S.R. (2014). *J. Am. Chem. Soc.* 136: 16299–16308.

141 You, M.X. and Jaffrey, S.R. (2015). *Annu. Rev. Biophys.* 44: 187–206.

142 Autour, A., Westhof, E., and Ryckelynck, M. (2016). *Nucleic Acids Res.* 44: 2491–2500.

143 DasGupta, S., Shelke, S.A., Li, N.S., and Piccirilli, J.A. (2015). *Chem. Commun.* 51: 9034–9037.

144 Mudiyanselage, A.P.K.K.K., Yu, Q.K., Leon-Duque, M.A. et al. (2018). *J. Am. Chem. Soc.* 140: 8739–8745.

145 Filonov, G.S., Kam, C.W., Song, W.J., and Jaffrey, S.R. (2015). *Chem. Biol.* 22: 649–660.

146 Alam, K.K., Tawiah, K.D., Lichte, M.F. et al. (2017). *ACS Synth. Biol.* 6: 1710–1721.

147 Voelcker, N.H., Guckian, K.M., Saghatelian, A., and Ghadiri, M.R. (2008). *Small* 4: 427–431.

148 Li, W., Zhang, F., Yan, H., and Liu, Y. (2016). *Nanoscale* 8: 3775–3784.

149 He, Y., Chen, Y., Li, C., and Cui, H. (2014). *Chem. Commun.* 50: 7994–7997.

150 Orbach, R., Willner, B., and Willner, I. (2015). *Chem. Commun.* 51: 4144–4160.

151 Fan, D., Zhu, J., Liu, Y. et al. (2016). *Nanoscale* 8: 3834–3840.

152 Xu, S.L., Li, H.L., Miao, Y.Q. et al. (2013). *NPG Asia Mater.* 5: e76.

153 Pagano, A., Iaccarino, N., Abdelhamid, M.A.S. et al. (2018). *Front. Chem.* 6: 1–13.

154 Zhou, C.Y., Wu, C.T., Liu, Y.Q., and Wang, E.K. (2016). *RSC Adv.* 6: 106641–106647.

155 Wu, C., Zhou, C., Wang, E., and Dong, S. (2016). *Nanoscale* 8: 14243–14249.

156 Fan, D., Zhu, X., Dong, S., and Wang, E. (2017). *ChemPhysChem* 18: 1767–1772.

157 Gao, R.R., Yao, T.M., Lv, X.Y. et al. (2017). *Chem. Sci.* 8: 4211–4222.

158 Liu, X.Q., Wang, F., Aizen, R. et al. (2013). *J. Am. Chem. Soc.* 135: 11832–11839.

159 Huang, Z.Z., Tao, Y., Pu, F. et al. (2012). *Chem. Eur. J.* 18: 6663–6669.

160 Sun, H., Xiang, J., Gai, W. et al. (2013). *Chem. Commun.* 49: 4510–4512.

161 Li, T., Zhang, L., Ai, J. et al. (2011). *ACS Nano* 5: 6334–6338.

162 Yang, C., Zou, D., Chen, J. et al. (2018). *Chem. Eur. J.* 24: 4019–4025.

163 Yang, C., Yang, S., Song, L. et al. (2019). *Chem. Commun.* https://doi.org/10.1039/C9CC00577C.

164 Liu, Y.L., Wu, H.P., Zhou, Q. et al. (2018). *Chem. Sci.* 9: 1666–1673.

165 Guo, S., Huang, H.Y., Deng, X.J. et al. (2018). *Nano Res.* 11: 2592–2604.

166 Green, A.A., Kim, J.M., Ma, D. et al. (2017). *Nature* 548: 117–121.

167 Elbaz, J., Moshe, M., and Willner, I. (2009). *Angew. Chem. Int. Ed.* 48: 3834–3837.

168 Pu, F., Liu, Z., Yang, X.J. et al. (2011). *Chem. Commun.* 47: 6024–6026.

169 Chen, J.H., Zhou, S.G., and Wen, J.L. (2015). *Angew. Chem. Int. Ed.* 54: 446–450.

170 Xu, X.Y. and Yan, B. (2017). *Adv. Funct. Mater.* 27: 1700247.

9

Nontraditional Luminescent and Quenching Materials for Nucleic Acid-Based Molecular Photonic Logic

Rehan Higgins, Melissa Massey, and W. Russ Algar

University of British Columbia, Department of Chemistry, 2036 Main Mall, Vancouver, BC V6T 1Z1, Canada

9.1 Introduction

Molecular logic devices (MLDs) are chemical systems that take multiple independent inputs and perform a Boolean operation to produce a single output [1]. In recent years, nucleic acids have emerged as a promising material for the design of MLDs because of their facile synthesis and the highly specific and tunable nature of Watson–Crick base pairing [2]. The simplest units of MLDs are Boolean logic gates, which take two inputs and perform a Boolean operation to produce a single binary output. The design of nucleic acid logic gates is motivated, in large part, by the idea of DNA computing, with the eventual goal of producing computing devices constructed entirely of DNA. Much progress has been made in this field since the first study on DNA computation [3], but significant hurdles remain before DNA will be competitive with the incredibly efficient silicon-based technology that is prevalent today. Arguably, more practical applications have been found for DNA logic in the areas of bioanalysis, targeted drug delivery, and theranostics [2, 4]. In the context of bioanalysis, for example, MLDs have the potential to respond to multiple molecular biomarkers of disease and autonomously produce a single "healthy" or "sick" diagnosis. Such capability may be highly advantageous for rapid screening of disease or infection.

One of the major challenges in the design of molecular logic gates is generation of easily measurable, Boolean-like output signals. Ideally, output signals are directly linked to the input event, providing a rapid response with no additional steps (e.g. nucleotide sequencing) prior to detection. It is here that MLDs typically differ from their silicon-based electronic analogs: the inputs and output are not of the same nature. Fluorescence, or "photoluminescence" (PL) more generally, is widely used as a photonic output signal for logic gates that have nucleic acid sequences and other molecules as inputs [2, 5]. In principle, PL may encode a signal in the form of intensity, color (i.e. wavelength), lifetime, and anisotropy, albeit that intensity and color are much more common in practice. PL signals are also readable at any time after the molecular logic event, and readout is nondestructive and possible without direct contact with the sample.

Among photoluminescent materials, fluorescent organic dyes are extensively utilized and have well-established methods for conjugation to DNA (i.e. labeling). Their small size tends to be minimally perturbative, and dyes that fluoresce in a wide range of colors from the ultraviolet (UV) to the near-infrared (NIR) are widely available. Despite these advantages, fluorescent dyes have properties that can be significant drawbacks for output signaling from a logic gate. These drawbacks include, but are not necessarily limited to, a propensity for photobleaching, sensitivity to environmental conditions, relatively broad emission spectra, small spectral separation between excitation and emission bands, short luminescence lifetimes, and low overall brightness [6]. Dyes also function very well as labels, but are not useful as scaffolds for the construction of multivalent systems. Given these potential drawbacks, the exclusive use of fluorescent dyes may impose limitations on the development of photonic logic gates. These limitations have motivated recent efforts to produce logic gates employing nontraditional luminescent materials such as photoluminescent nanoparticles and luminescent lanthanide complexes (LLCs). These materials provide a wider range of physical and optical properties, enabling increased control over the photonic outputs of the logic gates.

This chapter will summarize some recent advancements in the use of nontraditional luminescent materials for the design of photonic logic gates. First, we briefly introduce some of the fundamental concepts of Boolean logic and its implementation in nucleic acid MLDs. We then introduce the relevant luminescent materials, as well as their advantageous properties, and describe selected examples of their application in the context of nucleic acid Boolean logic gates. The luminescent materials covered are primarily photoluminescent or quenching materials. The applications covered are, for the most part, limited to fundamental studies for the development of logic devices as opposed to logic-based sensing or imaging applications.

9.2 DNA Molecular Photonic Logic Gates

To evaluate the effectiveness of different luminescent materials in photonic molecular logic gates, it is necessary to first define the characteristics of an ideal molecular logic gate. The defining feature of an ideal Boolean logic gate is a binary output in response to binary inputs. The binary states are TRUE and FALSE, or, alternatively, 1 and 0, respectively. The simplest logic functions are YES and NOT, which return a TRUE and FALSE output, respectively, from a single TRUE input. The six elementary two-input logic gates are summarized in Figure 9.1. Of these, the most intuitive and most common are the AND gate (TRUE output in response to TRUE states for both inputs) and the OR gate (TRUE output in response to a TRUE state for either or both inputs). Other simple two-input gates can be expressed as combinations of these elementary operations. For example, the INH gate, which returns a TRUE output in response to *only* one TRUE input, can be thought of as a combination of a NOT gate and an AND gate. The NAND and NOR gates are universal logic gates in the sense

Input		Output					
		AND	OR	NAND	NOR	XOR	XNOR
A	B						
0	0	0	0	1	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	1	0	0	0	1

Figure 9.1 Truth tables and symbols for the elementary two-input Boolean logic gates.

that these can be connected to themselves to execute any logical operation. Real logic gates, including those based on nucleic acids, do not have infinite signal-to-background ratios, but rather defined thresholds that define FALSE and TRUE states.

The inputs for DNA-based logic gates are generally nucleic acid sequences but are sometimes other types of DNA-interacting molecules. In contrast to a classic digital electronic logic gate based on silicon, DNA logic gates generally have analog-like inputs. The reasons are that large populations of molecules are usually worked with and measured and that the binding behaviors that actuate Boolean signaling are concentration dependent. Nevertheless, a digital Boolean output remains desirable. This single digital output, in combination with multiple inputs, distinguishes a DNA-based logic gate from a DNA-based sensor or assay. As noted earlier, the output from a DNA-based logic gate is often not a nucleic acid sequence but rather a photonic signal. Nucleic acid outputs are potentially useful for initial or intermediate logic gates in a circuit, but the final logic gates in a circuit need to be quickly and easily readable, which is not the case for molecular outputs. Photonic output does meet this requirement.

Photonic signaling from DNA logic gates with organic fluorescent dyes is often achieved via energy transfer (ET) networks. Förster resonance energy transfer (FRET) is the most common ET mechanism between fluorescent dyes [7]. Briefly, FRET occurs when a donor is excited through absorption of a photon and subsequently transfers this excitation energy non-radiatively and through space to an acceptor. The process is strongly distance dependent, with its characteristic length scale on the order of several nanometers, and requires overlap between the emission spectrum of the donor and absorption spectrum of the acceptor. In most logic gates based on FRET, the signaling fluorescent dye is either sensitized by a donor or quenched by an acceptor. In cases where the fluorescent dyes are covalently linked to DNA, the distance between the dye and its FRET partner is modulated by the DNA-based logic gate to produce changes in the distance-dependent ET efficiency and fluorescence output signal [8]. Multiple dye-labeled oligonucleotide sequences are a feature of many DNA logic gate designs.

Another signaling strategy with DNA-based logic gates is the use of intercalating dyes. These dyes exhibit a significant increase in fluorescence intensity upon association with nucleic acids and are often selective to certain DNA secondary structures (e.g. specific intercalation of N-methyl mesoporphyrin IX

[NMM] into G-quadruplexes). Intercalating dyes may be used in combination with dye-labeled oligonucleotides and ET strategies.

The proximity between donors and acceptors in ET-based logic gates, as well as the binding of certain dyes to DNA structural motifs, is frequently actuated through preferential binding. That is, the logic gate inputs cause the overall logic gate system to convert to a more energetically favorable molecular state. These conversions tend be predictable and tunable because of highly specific hybridization of complementary oligonucleotides [9, 10]. A workhorse mechanism in DNA logic gate systems is toehold-mediated strand displacement, which is the displacement of a partial-length complementary strand from a duplex by a longer complementary strand. This process occurs in three steps: binding to the toehold (an unhybridized section at the end of the duplex), branch migration, and dissociation of the original oligonucleotide. With the reversible nature of DNA hybridization, the kinetics of this process are highly variable and can be tuned by altering the length of the toehold, the overall sequence length, the G/C content of the sequence, and the salt concentration [11, 12]. Besides the standard Watson–Crick base pairing, several noncanonical DNA conformations are also prevalent in the design of logic gates. These include the self-assembly of guanine-rich sequences into G-quadruplexes [13], the formation of triplex DNA [14], and metal-mediated base pairing (e.g. T–Hg–T and C–Ag–C) [15]. Though more applicable to sensing applications, DNA aptamers that specifically bind protein or small molecule analytes, as well as DNAzymes that catalyze strand-cleavage reactions, are also important functional motifs for DNA nanostructures [16, 17].

Given all of the foregoing, a frequent challenge with DNA-based logic gates is something akin to analog-to-digital conversion. Each individual logic gate may behave digitally, responding photonically to single-input molecules, but these individual logic gates are only observable with sophisticated spectroscopy and imaging systems. The much more common ensemble-level measurements of millions or billions of logic gates require very high contrast between FALSE and TRUE photonic outputs for Boolean-like signaling to be observed across wide ranges of input concentrations. Overcoming this challenge is one of several possible ways that the physical and optical properties of nontraditional luminescent materials may be advantageous in the design of DNA-based molecular photonic logic gates.

9.3 Nontraditional Luminescent Materials

Here, we define *nontraditional luminescent materials* as luminescent materials that are not fluorescent proteins or organic fluorescent dyes. Likewise, *nontraditional quenching materials* are strong quenchers of PL that are not organic molecular chromophores. Some of these materials are physically similar to traditional fluorescent dyes but have significantly different PL properties (e.g. LLCs), whereas other materials are physically very different (e.g. nanoparticles) with PL and other optical properties that range from similar to dyes to very distinct. When used strategically, nontraditional luminescent materials may

greatly improve the performance of logic gates. The materials discussed in this chapter include semiconductor quantum dots (QDs), LLCs, upconversion nanoparticles (UCNPs), gold nanoparticles (AuNPs), metal nanoclusters (NCs), graphene-based materials, carbon dots (CDs), and conjugated polymers (CPs).

With the exception of LLCs and CPs, the materials listed in the previous paragraph are nanoparticles. These materials have varied architectures, surface areas, and surface chemistries that enable a range of interactions with nucleic acids. In tandem with their optical properties, these interactions can be exploited for the design of photonic logic gates. Many nanoparticles also non-covalently bind with DNA, either through electrostatic interactions with its negatively charged phosphate backbone or through π–π stacking interactions with its nucleobases [18]. These interactions are typically dependent on the DNA secondary structure (e.g. single-stranded DNA [ssDNA], double-stranded DNA [dsDNA], G-quadruplex), and induced changes in structure are useful for producing changes in PL signals. Nucleic acids can also be covalently linked to most nanoparticle surfaces through established bioconjugation chemistries [19]. The multivalency enabled by nanoparticles is one manner in which nontraditional luminescent materials can enable new DNA logic gate designs. In addition, the PL properties of these materials are also well suited for use in ET networks and may provide new or improved capabilities.

9.4 Semiconductor "Quantum Dot" Nanocrystals

9.4.1 Quantum Dots

QDs are colloidal semiconductor nanocrystals with very bright PL. Cadmium chalcogenide semiconductors are the most widely used QD materials and generally provide QDs with the best PL properties in the visible region of the spectrum, although a variety of other types of QDs have been developed. The small size of these nanocrystals, typically between 1 and 10 nm in diameter, gives rise to quantum confinement effects that are responsible for their optical and PL properties. Briefly, the relatively low number of atoms in the nanocrystals (compared with a bulk material) causes the valence and conduction bands to have discrete energy levels at the band edges. The results are a band gap energy that increases with decreasing size, strong light absorption, and efficient PL emission [20]. As the PL emission wavelength is directly proportional to the band gap energy, the PL of QDs is continuously tunable with nanocrystal size and composition, such that QDs with PL emission ranging from violet to the infrared (400–1350 nm) are available [21].

In comparison with traditional molecular fluorophores, QDs have PL emission peaks that are more symmetric and narrow spectrally (full-width-at-half-maximum [FWHM] 25–40 nm in the visible spectrum) and show a much greater resistance to photobleaching [22]. These properties, paired with the exceptional brightness provided by high quantum yields (0.1–0.9) and large molar extinction coefficients (10^5–10^7 M^{-1} cm^{-1}), have made QDs an attractive material for many applications, particularly in the fields of bioanalysis and

imaging [23, 24]. Advances in ligand exchange and encapsulation methods have aided the development of these applications by enabling stable colloidal dispersions of QDs in aqueous media and conjugation with a wide variety of biomolecules [25, 26]. DNA-coated QDs were first developed in 1999, where thiol-modified oligonucleotides were used to displace other reversibly bound surface ligands [27]. Several alternative methods for QD–DNA conjugation have since been developed [19, 28, 29].

With respect to ET-based signaling, the narrow and tunable emission of QDs is an ideal property for FRET donors, and their size allows for conjugation of multiple organic dye acceptors per QD [28]. Though they are typically FRET donors, QDs can also act as FRET acceptors in configurations with other nontraditional fluorescent materials (e.g. LLCs or other QDs) [30]. In systems with multiple QDs conjugated or aggregated together, QDs can act as both FRET acceptors and donors, with smaller QDs sensitizing the emission of larger QDs. In aggregates of nominally monodisperse QDs, slightly smaller QDs act as donors to the slightly larger QDs, resulting in narrowing and redshifting of the overall PL emission [31]. These strategies have been employed for the use of QD–DNA conjugates in photonic logic gate applications.

9.4.2 Logic Gates with QDs

He et al. created a QD-based set of all six elementary logic gates (OR, AND, NOR, NAND, XOR, and XNOR) plus an INH gate using QD–DNA conjugates and unlabeled nucleotides [32]. In this study, different QD–DNA conjugates (red, blue, and green) with well-separated PL emission spectra were hybridized to a single DNA template to produce the logic gates (Figure 9.2I). In some of these gates (AND, OR, and XOR), FRET quenching between the QD pairs was regulated by displacement of individual QDs from the template by input ("fuel") strands, with the resulting increase in PL intensity constituting the logic gate output signals. For example, in the OR gate, the initial state had the DNA template hybridized with two QDs (blue and green) via partially complementary QD-conjugated oligonucleotides, with the blue QD acting as the FRET donor to the green QD acceptor (Figure 9.2II). The input strands were fully complementary to the QD-conjugated DNA, displacing the QD–DNA conjugates from the template strand via toehold-mediated strand displacement. Addition of either or both inputs led to release of the QDs, and therefore loss of the QD–QD FRET, and restored the PL emission of the quenched blue QD (the logical output signal). The inverse of this process was also used, with inputs causing hybridization of the QDs to the template strand, in the INH, NAND, NOR, and XNOR gates. In the INH gate, the initial configuration had two QDs (blue and green) hybridized to the template strand, with a third QD (red) free in solution with a blocking strand hybridized to its surface nucleotides (Figure 9.2III). One input ("fuel") functioned as in the OR gate example, releasing the green QD when added, while the other input ("anti-fuel") instead displaced the blocking strand on the red QD, allowing it to hybridize to the template strand. Addition of the "anti-fuel" always resulted in quenching of the blue QD by the newly hybridized red QD, meaning only addition of the "fuel" input alone resulted in PL signal from the blue QD, giving the
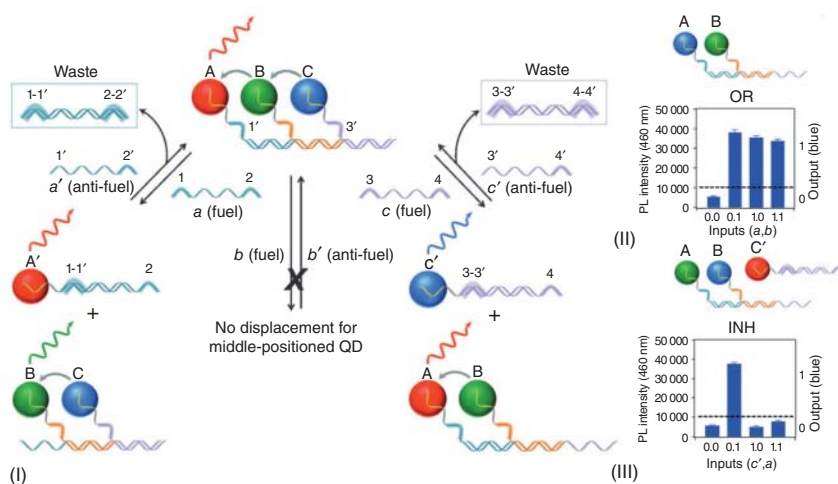
Figure 9.2 Three-color QD logic gates. (I) Mechanism for control of hybridization between QD–DNA conjugates and a template strand via the addition of fuel and anti-fuel oligonucleotides. Also shown are the initial configuration and PL outputs for the (II) OR and (III) INH gates. Source: From He et al. [32]. Reproduced with the permission of John Wiley & Sons.

system INH gate behavior. The other five elementary logic gates were similarly created by varying the initial configuration, number of QDs (two or three), and the oligonucleotide inputs, all with good TRUE/FALSE signal contrast.

Other QD–DNA logic systems have been designed to provide selective detection of multiple biologically important analytes through a single output via a logic operation. One class of target for these assays is microRNAs (miRNAs), which are short RNA fragments that play a key role in gene expression [33] and are promising targets for treatment of a range of diseases [34]. A study by Miao et al. employed various miRNAs as inputs for two- and three-input AND/OR logic gates with Ag_2S QDs [35]. The miRNA inputs displaced QD–DNA conjugates from a template strand fixed to a solid gold support, releasing the QDs into bulk solution where the PL intensity was measured. Another class of target analyte is metal cations that form sequence-specific complexes with DNA (e.g. T–Hg–T, C–Ag–C) [36]. A study by Freeman et al. used the sequence-specific binding of Hg^{2+} and Ag^+ ions to oligonucleotides on QD–DNA conjugates to induce OR gate quenching of the QD PL via electron transfer to the cations [37]. In contrast to many other logic gates, the TRUE signal was a decrease, rather than increase, in PL intensity.

9.5 Lanthanide-Based Materials

9.5.1 Luminescent Lanthanide Complexes

Trivalent lanthanide ions (Ln^{3+}) are stable in aqueous solution and are photoluminescent. The ground-state $[Xe]4f^n$ ($n = 0–14$) electron configuration [38],

in combination with the shielding of the 4f electrons by the filled $5s^2 5p^6$ sub-shells, gives rise to unique PL properties. Emission occurs from f–f transitions, which are inner shell transitions. The result is multiple sharp, line-like emission bands at well-defined wavelengths with minimal sensitivity to the surrounding environment, as the 4f orbitals have little interaction with the ligand field [39, 40]. Emission from 4f–4f inner shell electrons is formally parity forbidden via electric-dipole (ED) transitions, so the emission from lanthanides in solution comes from 4f–4f metastable states [41], resulting in long excited-state lifetimes (*vide infra*) and very low molar absorption coefficients ($0.1–10\,M^{-1}\,cm^{-1}$) [38, 42, 43].

To overcome the limitation of low absorption coefficients, brightly luminescent materials based on lanthanide ions are achieved through indirect excitation via sensitization or antenna effects. In the case of lanthanide complexes, bright luminescence results from initial excitation of a chromophoric organic chelator or cryptand that surrounds and binds the metal ion [39, 41]. The energy absorbed by a chromophore is efficiently transferred to the trivalent lanthanide, which then undergoes metal-centered emission [39, 42]. This antenna effect can increase the brightness of lanthanides from $<1–5\,M^{-1}\,cm^{-1}$ to values as high as $10^4–10^5\,M^{-1}\,cm^{-1}$ [42]. In addition to binding the ion and acting as an antenna, the chelators or cryptands serve as a handle for conjugation to a (bio)molecule of interest [44]. Typically, the complexes feature octadentate binding that fills the coordination sphere of the trivalent lanthanide so that water does not occupy coordination sites and quench luminescence (water molecules vibrationally deactivate lanthanide excited states) [39].

The partially forbidden nature of 4f–4f transitions also gives rise to excited-state lifetimes that are on the order of microseconds to milliseconds for lanthanide complexes [45, 46]. The excited-state lifetime, sometimes referred to as the fluorescence or PL lifetime, is a measure of the period of time over which PL emission is observed following initial excitation. (It is formally the inverse of the sum of relaxation rates for the excited state.) Most other luminescent materials have excited-state lifetimes on the order of nanoseconds. Despite their long-lived excited states, lanthanide ions are relatively resistant to photobleaching [43].

For applications in bioanalysis and imaging, Tb^{3+} and Eu^{3+} are the most commonly used lanthanides by virtue of their emission in the visible region of the spectrum and favorable PL quantum yields [39]. The multiple narrow emission lines in the visible region are spectrally well separated from the absorption of the chelator/cryptand in the UV region of the spectrum. Time-gated measurements are frequently used with LLCs of Tb^{3+} and Eu^{3+}, leveraging typical excited-state lifetimes between hundreds of microseconds and a few milliseconds. Time-gated measurements consist of an excitation pulse, a delay time, and an integration time over which PL signal is collected. In this manner, short-lived PL (i.e. nanosecond timescale) is rejected, and long-lived PL is selectively measured. Some commercially available fluorescence instruments have time-gated measurement capability built in.

9.5.2 Coupling Lanthanide Complexes with Energy Transfer

LLCs engage in FRET, albeit sometimes called luminescence resonance energy transfer (LRET), where lanthanide ions are the energy donors [44, 47]. It is worth highlighting some particularly useful features of FRET with LLCs and why these materials are excellent choices for energy donors. Spectrally, the multiple, narrow, and well-spaced emission lines make Tb-LLCs a suitable donor for many colors of fluorescent dye acceptor while still permitting spectral resolution of the acceptor dye's PL emission. The large spectral separation between excitation and emission is also convenient. With these properties, a Tb-LLC can be simultaneously used as a donor for multiple acceptor dyes in multiplexed FRET. In addition, because the acceptor is sensitized by the excited state of the LLC donor, the PL from the energy acceptor takes on the excited-state lifetime of the donor. Given the typical mismatch between the long excited-state lifetime of an LLC donor and the comparatively short excited-state lifetime of most acceptor materials (e.g. nanoseconds for dyes), any unwanted, directly excited PL from the acceptor is avoided by time-gated measurements. Background autofluorescence from a sample or interference from scattering of excitation light is also suppressed.

9.5.3 Logic Gates with LLCs and Lanthanide Ions

A luminescent terbium cryptate (an LLC) was used as a FRET donor to create time-gated DNA-based photonic logic gates for AND, OR, NAND, and NOR operations [48]. The logic gates were actuated by toehold-mediated strand displacement by unlabeled oligonucleotide targets, and the logic gate designs comprised an LLC-labeled oligonucleotide hybridized with acceptor dye-labeled oligonucleotides in various configurations. The output signals were the time-gated, FRET-sensitized dye emission. Multiple design elements were leveraged to achieve some of the highest contrast ratios (up to 10-fold) between TRUE and FALSE states to date: time-gated measurements rejected background PL from direct excitation of acceptors, and FRET between the LLC and fluorescent dyes competed with FRET between the LLC and nonluminescent quenchers that were strategically placed in the logic gate construct (Figure 9.3). The multiplexing advantage of LLCs also enabled two-color multiplexed logic with two inputs (AND and OR) and three inputs (AND and OR), and operation in serum was demonstrated.

Tb cryptates have also been used to create FRET-based bionanophotonic logic devices by attaching different numbers of LLCs (input) and molecular dyes (input) through peptide linkers to a central QD scaffold to achieve both simple and complex logic function computation [49, 50]. For the simple logic functions of OR, AND, INH, XOR, NOR, and NAND, a PL signal was the output, where an improved threshold between TRUE and FALSE states was attainable through the use of time gating. Although this example is not DNA based, the peptidyl linkers, in theory, could be replaced with oligonucleotide sequences.

In addition to cryptands and chelates, uncomplexed lanthanide ions can utilize guanosine 5′-monophosphate (GMP) nucleotides as a coordinating
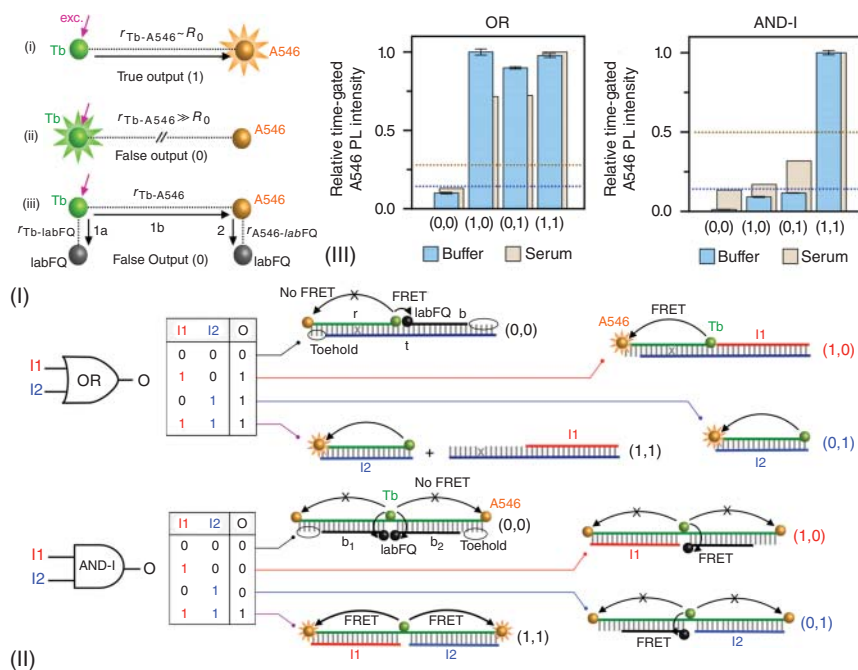
Figure 9.3 Time-gated photonic logic gates with LLCs as FRET donors. (I) ET-based strategies for generating TRUE and FALSE output signals. (II) Truth tables and cartoons showing the design and operation of OR and AND logic gates. (III) Time-gated PL intensity outputs (from the A546 FRET acceptor) for the OR and AND logic gates. Source: From Massey et al. [48]. Copyright 2017, Reproduced with the permission of the American Chemical Society.

ligand for modulation of PL intensity through ET [51]. Coordination polymer nanoparticles (CPNs) generated through self-assembly of GMP and lanthanide ions (Tb^{3+}, Eu^{3+}, and Ce^{3+}) have been developed for color-tunable PL and Boolean logic gates [52]. GMP-Tb CPNs were used to generate combinatorial gates of INH–INH (Tb^{3+}, dipicolinic acid [DPA], ethylenediaminetetraacetic acid [EDTA] inputs) and NOR–OR (Cu^{2+}, Hg^{2+}, Cys inputs), and GMP-Eu CPNs were used to generate an AND–INH (DPA, EDTA, Cu^{2+} inputs) logic system. The outputs of the first gate were used as inputs for the next gate, with a PL output acting as the final signal. In a different example, GMP-Eu CPNs were used along with NMM, fluorescein, and Hoechst 33342 (an intercalating dye) to design AND, INH, and implication (IMP) gates, respectively, in aqueous solution [53]. The inputs were GMP and Eu^{3+}, with fluorescence from NMM, fluorescein, or Hoechst 33342 as the output. Although it is debatable whether reagents that comprise the CPNs can be representative of true logic gate inputs, the outputs nonetheless gave the correct truth table outputs to the corresponding logic operator. Cysteine and histidine have also been used as inputs along with Ni^{2+} and *N*-ethylmaleimide (NEM) for a Tb^{3+}/GMP-Cu^{2+} ensemble-based logic device where time-gated luminescence of the Tb^{3+} was used as the signal for a true logic state in an integrated INH–INH–OR logic system [54]. The output of the two INH gates served as the inputs for the OR gate.

9.5.4   Upconversion Nanoparticles

Upconversion luminescence (UCL) is a type of PL with the characteristic feature that the emitted photons have higher energy than the absorbed photons. Incorporation of trivalent lanthanide ions in carefully designed nanoparticles enables the exploitation of their long excited-state lifetimes to produce UCL. These UCNPs vary in composition but most commonly consist of a $NaYF_4$ host matrix with relatively high levels of doping of a sensitizer ion (Yb^{3+} at ~20%) and much lower levels of doping of an emitter ion (Tm^{3+} or Er^{3+} at <3%). The nanoparticles range in size from tens to hundreds of nanometers [55, 56]. The emitter ions alone can produce UCL by an excited-state absorption (ESA) mechanism, but co-doping of a sensitizer enables an energy transfer upconversion (ETU) mechanism that greatly increases quantum yields. In ETU, NIR light excites the sensitizer, populating an excited state from which energy can be transferred to excite the emitter ion to a metastable intermediate excited state. Upconversion occurs when energy is transferred from the excited state of the sensitizer to the now populated metastable state of the emitter, resulting in population of a higher excited state. Emissive relaxation from this state produces UCL.

Like LLCs, UCNPs display several sharp emission peaks and are often used as ET donors [55, 57]. The upconversion effectively eliminates the problem of background signal from matrix scattering and autofluorescence. These properties also enable effective use of an inner filter effect (IFE) as a means of UCL modulation. IFE results from the absorption of emission (or excitation) light by an "absorber" in the detection system [58]. The narrow emission bands of UCNPs fully overlap with absorbers easily, while a large spectral separation removes the problem of absorption of the excitation light. IFE avoids the requirement of direct attachment to the nanoparticles, as it occurs over larger distances, but is typically much less efficient than ET quenching. IFE quenching can be induced by chemical modification of the absorber to change its optical properties, creating overlap between its absorbance and the UCL emission.

9.5.5   Logic Gates with UCNPs

Fan et al. developed a set of logic gates utilizing the DNA-assisted catalytic oxidation of an IFE absorber to quench UCNP emission [59]. For these logic gates, UCNPs were initially added to a solution containing 3,3′,5,5′-tetramethylbenzidine (TMB), H_2O_2, and hemin. In this state, UCL was observable upon excitation with a NIR laser, as the solution was colorless under visible light. Oxidation of the TMB (to create oxTMB) caused a significant shift in the absorbance spectrum, introducing a strong band overlapping with the UCNP PL emission. The oxidation of TMB was catalyzed by G-quadruplex-bound hemin and H_2O_2, such that the addition of G-quadruplex-forming oligonucleotides triggered quenching of the UCNP PL. In the INH gate, for example, the inputs were a G-quadruplex-forming oligonucleotide and its complement. The G-quadruplex strand catalyzed the oxidation of TMB, resulting in quenched UCL and a blue solution, both of which were suitable as TRUE output signals. Addition of the complementary strand resulted in formation of a duplex, preventing oxidation

of the TMB, and the output remained in the FALSE or colorless and unquenched state. By changing the oligonucleotide inputs, two-input AND, OR, and XOR gates were also created. Interestingly, oxTMB was effectively reduced to TMB through the addition of glutathione (GSH). This property was used in a series of combinatorial logic circuits with as many as five inputs, with the GSH acting as the "inhibiting" input in the INH gates.

9.6 Gold Nanoparticles

9.6.1 Gold Nanoparticles

AuNPs have fascinating optical properties that have been noted for nearly two centuries [60]. Due to their small size (c. 5–100 nm), the electric field of incident light can penetrate the AuNP, causing a coherent oscillation of the conduction band electrons of the particle – a phenomenon called localized surface plasmon resonance (LSPR) [61, 62]. The intense color of AuNPs comes from the wavelength-dependent absorption and scattering of visible light as it interacts with the free conduction band electrons [63]. The LSPR of an AuNP is primarily determined by its size and shape, giving strong absorbance bands at the resonant frequencies [64]. These frequencies also depend on a range of other factors including surface coating and aggregation state, which is a feature that has been utilized in a wide range of bioanalysis methods [65–70]. AuNPs have good biocompatibility and well-established chemistry for conjugation with biomolecules, especially nucleic acids [71]. In particular, thiol-modified oligonucleotides bind strongly to the surface of AuNPs, and this conjugation strategy is very widely utilized [72]. With respect to optical signaling, AuNPs are typically used as colorimetric indicators or as efficient quenchers of fluorescence.

9.6.2 Logic Gates with AuNPs and Colorimetric Output

The interactions between the LSPRs of AuNPs in very close proximity (at distances approximately less than the average particle diameter) cause a significant shift in color (Figure 9.4I) [73, 75, 76]. The aggregation of AuNPs can be selectively induced in a variety of ways that are largely dependent on their surface chemistry and bioconjugation [77]. This phenomenon has been employed in DNA-based detection schemes with both modified [78–81] and unmodified [82–84] AuNPs. The colorimetric signal is easily measured with simple optical equipment or even by the naked eye, making it an attractive reporter system for a variety of applications.

Jiang et al. developed two logic gates with visually detectable signal changes based on the aggregation of AuNPs [74]. Surface adsorption of ssDNA on the AuNPs was mediated by hybridization of the ssDNA with a complementary oligonucleotide and by encouraging the strand to form a G-quadruplex by increasing the H^+ concentration (i.e. lowering the pH). Both methods disrupted the DNA–AuNP interaction, resulting in aggregation of the AuNPs. To create an XOR gate, the AuNPs were initially stabilized by a G-rich oligonucleotide,
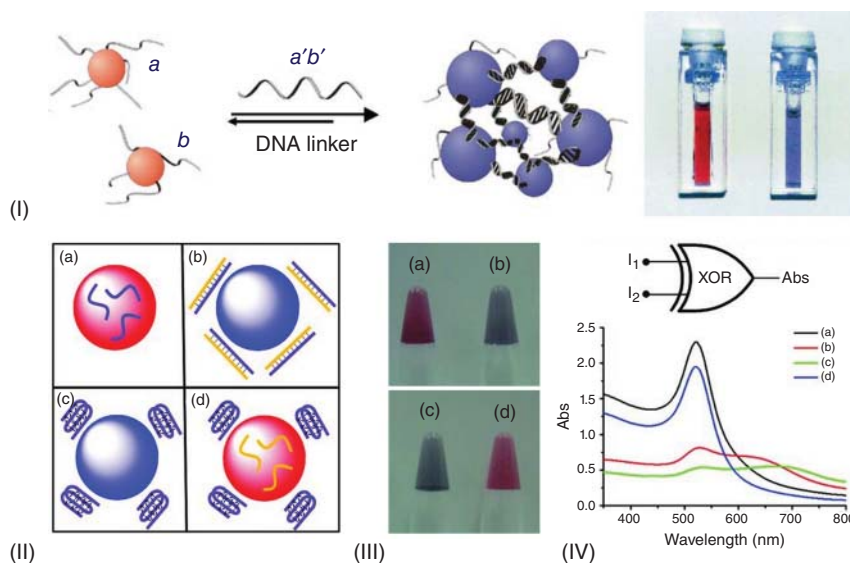
Figure 9.4 DNA-mediated AuNP aggregation as a colorimetric indicator. (I) Visible difference in color between colloidally dispersed (red) and aggregated (purple) AuNPs. The cartoon shows aggregation mediated by hybridization. (II) Colorimetric XOR gate with the aggregation state mediated by adsorption of ssDNA on AuNP surface. Color change detectable by (III) visible eye or (IV) absorbance spectroscopy. Source: (Part I) From Thaxton et al. [73]. Reproduced with the permission of Elsevier; (Parts II–IV) From Jiang et al. [74]. Reproduced with the permission of John Wiley & Sons.

with a partially complementary oligonucleotide and H^+ acting as the two inputs (Figure 9.4II). The formation of the G-quadruplex was energetically favorable over the hybridization of the two strands in the presence of H^+ ions (pH of 5.5), such that the addition of both inputs caused no significant signal change because of the stabilization of the AuNPs by the input oligonucleotide. Addition of either input alone, however, led to aggregation of the AuNPs, with the stabilizing nucleotide forming a duplex (oligonucleotide input) or a G-quadruplex (H^+ input). Loss of the absorbance peak at ~520 nm, or naked eye detection of the color change from red to purple, was taken as the output for the resulting XOR gate. To produce an AND gate, a second oligonucleotide with stronger hybridization to the input oligonucleotide and no ability to form a G-quadruplex was added to the starting system. Addition of either input released only one strand from the AuNP surface, and aggregation was only induced after addition of both inputs. Although the signal contrast was lower in this case, naked eye detection of the color change was still sufficient to detect the AND gate response of the system (Figure 9.4III). Another study described a similar system, using DNA aptamers to stabilize the AuNPs, with aggregation occurring upon addition of the aptamer targets (cocaine and adenosine) to provide AND and OR sensing [85].

Aggregation between DNA-functionalized AuNPs may occur through cross-linking or non-cross-linking mechanisms [86]. Cross-linking-induced

aggregation usually involves hybridization-mediated linkages between AuNPs, with multiple copies of each strand on the AuNPs enabling cross-linking between many particles to form large aggregates. As an example, the controlled hybridization between AuNPs based on metal-ion-mediated base pairing was used to create a set of logic gates [87]. In each case, one of the inputs was Hg^{2+} or Ag^+, which induced T–Hg–T or C–Ag–C bonding to hybridize AuNPs functionalized with partially complementary oligonucleotides, while the second input varied between designs for AND, INH, and XOR logic gates.

Although the functionalization of AuNPs with dsDNA typically prevents salt-induced aggregation, a high surface density of dsDNA may induce aggregation under similar conditions via end-to-end stacking interactions between duplexes [86]. These interactions can be easily disrupted, such that even a single-nucleotide mismatch at the distal end of the duplex can prevent aggregation [88]. As an example of this concept, Kanayama et al. used metal-ion-mediated base pairing to regulate the non-cross-linking aggregation of AuNPs for a set of logic gates [89]. The AuNPs were functionalized with thiolated nucleotides, and a mostly complementary nucleotide was added to form duplexes with several T–T and/or C–C mismatches at the distal ends. The disruption of the duplex due to the mismatched base pairs prevented end-to-end stacking, and the particles remained dispersed in solution. For the AND gate, the mismatched end of the duplex comprised two T–T pairs and one C–C pair at the terminus. Addition of Hg^{2+} or Ag^+ alone left mismatches at the end of the duplex, resulting in no aggregation and no signal change. Addition of both inputs formed all three metal-mediated base pairs and a fully closed duplex, leading to aggregation. The visible color change was taken as the output signal. An OR gate was made by modifying the DNA duplexes to have less mismatches at the distal end, allowing a single metal ion to complete the hybridization.

9.6.3 Logic Gates with AuNPs and PL Quenching

AuNPs can also act as extremely efficient quenchers for luminescent materials [90, 91]. The mechanisms responsible for ET from the fluorophores to AuNPs have a weaker distance dependence than FRET, allowing AuNPs to quench fluorescence with greater efficiency over a longer distance compared with traditional organic quenchers [90, 92]. More efficient quenching is often ideal for logic gate applications, where a fully dark (FALSE) state provides higher signal contrast. Besides distance, this quenching is highly dependent on factors including particle size and morphology [92].

Two related studies by Yang et al. employed quenching of an organic dye by AuNPs to create AND logic gates and more advanced logic circuits [93, 94]. In the first study, AuNPs were conjugated with two distinct sequences of thiol-terminated DNA, each complementary to separate segments of a longer Cy3-labeled DNA strand [93]. The dye emission was initially quenched because of its proximity to the AuNP. Input DNA strands preferentially hybridized with the AuNP-conjugated strands via toehold-mediated strand displacement, releasing the Cy3-labeled strand to restore its fluorescence intensity. As the hybridization of two sequences was necessary to release the Cy3-labeled strand, the system
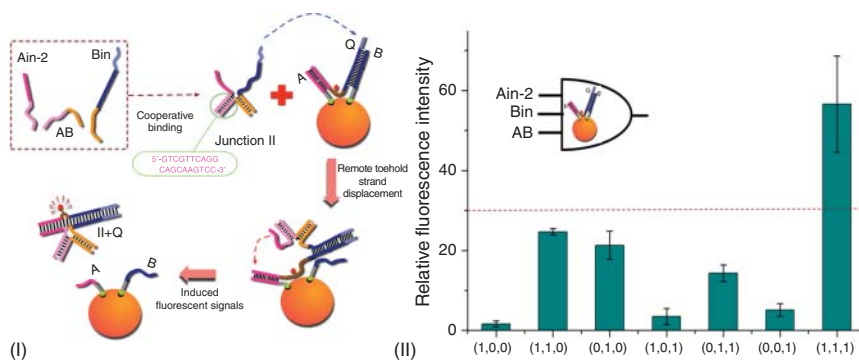
Figure 9.5 Three-input logic gate with an AuNP quencher. (I) Cooperative binding of inputs leading to remote toehold-mediated strand displacement and fluorescence restoration. (II) Fluorescent response of the three-input AND gate. Source: From Yang et al. [94]. Reproduced with the permission of American Chemical Society.

was an AND gate. The second study expanded on this idea by introducing a cooperative "binding-induced" mechanism, where DNA inputs combined, via hybridization, to create a merged longer input before displacing the Cy3 strand via a single toehold (Figure 9.5) [94]. This strategy enabled the construction of more complex three-input AND logic gates, although the signal contrast was significantly reduced. A similar study by Liu et al. used AuNP quenching of Cy5-labeled oligonucleotides to produce an XOR gate, with detection of the logic output signal through single-particle imaging and photon counting [95].

9.7 Metal Nanoclusters

9.7.1 Metal Nanoclusters

Fluorescent metal NCs are an emerging class of photoluminescent materials that have properties that are intermediate between metal nanoparticles and traditional organic fluorescent dyes. Metal NCs are stabilized clusters of a few to roughly one hundred atoms (≤ 3 nm) with markedly different properties than those of larger nanoparticles with hundreds to thousands of atoms. With this limited number of atoms, NCs have a molecule-like electronic structure with well-separated energy levels and are often considered nonmetallic [96]. The change from the mostly continuous energy bands of metal NPs to the discrete energy levels of NCs results in a loss of conductive properties but can lead to PL from electronic transitions between these levels [97]. As the gaps between energy levels are directly related to the number of atoms, this emission can be tuned by changing the size of the NC.

Recent advances in overcoming the challenges of low quantum yield and poor colloidal stability in water have made metal NCs increasingly popular luminescent materials [98]. The typical synthetic method, which is the reduction of a metal salt in the presence of a stabilizing ligand, is generally same as for the synthesis of NPs. Careful choice of ligands and reaction conditions is required

to obtain NCs rather than NPs [99] and to provide favorable conditions for high fluorescence quantum yields [100]. Among the variety of ligands that have been utilized, DNA has emerged as a particularly useful ligand for templating NCs. The four nucleobases have different abilities as stabilizing ligands, and the size of the DNA-templated NCs is tuned by changing the sequence of the oligonucleotide [101].

Silver nanoclusters (AgNCs) are the most well-studied and well-established DNA-templated NCs, due in part to the strong and specific coordination of Ag^+ by cytosine [102]. More recently, copper [103, 104] and gold [105, 106] NCs have also been successfully synthesized with DNA as a stabilizing ligand. Oligonucleotide-templated metal NCs are a particularly attractive fluorescent material for DNA-based applications because they eliminate the need for subsequent bioconjugation methods [107]. The PL from metal NCs is sensitive to their local microenvironment, and a variety of signaling strategies take advantage of this sensitivity; for example, alteration of the DNA strands hybridized to the NC-bearing oligonucleotide [108]. ET-based signaling strategies are also used with metal NCs.

9.7.2 Logic Gates with Metal Nanoclusters

Fan et al. developed a range of label-free logic gates and devices using ET quenching of fluorescent DNA-templated AgNCs by graphene oxide (GO) [109]. In the general design, a single-stranded AgNC-containing oligonucleotide was initially adsorbed to GO, which quenched the AgNC fluorescence. Addition of input oligonucleotides modulated the fluorescence by forming a duplex with the NC-containing strand, removing it from the GO surface. For example, an XOR gate had two oligonucleotide inputs that were both partially complementary to the NC-containing strand and more complementary to each other. Addition of either input released the AgNC from the GO surface through duplex formation, whereas the addition of both inputs did not affect the NC-containing strand as the inputs preferentially hybridized with each other. To allow for the implementation of more advanced logic functions, some of these duplexes were designed to form G-quadruplexes, which were detected through the intercalation and subsequent fluorescent enhancement of NMM. The fluorescent emissions of the AgNC and NMM were used as dual outputs in logic devices including a 4-to-2 encoder and a comparator. Another study used photoinduced electron transfer (PET) quenching of DNA aptamer-templated AgNCs by hemin as the output mechanism for AND gate detection of hemin and a protein biomarker for cancer (platelet-derived growth factor-BB) [110].

Other logic gate designs have leveraged the sensitivity of AgNC PL to its microenvironment for signaling. Huang et al. created an impressive set of logic gates based on the fluorescent enhancement caused by introducing a guanine (G)-rich strand to the local microenvironment of an AgNC [111]. In these systems, "dark" AgNCs were first synthesized on a polyacrylic acid (PAA) template. The AgNCs were readily transferred to oligonucleotides because of their higher affinity for cytosine (C). The ssDNA-AgNC system remained in the "dark" state, but the addition of a partially complementary strand, which placed a G-rich
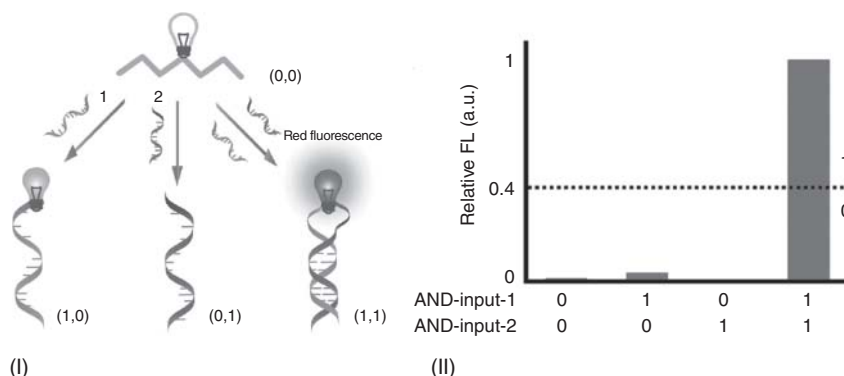
Figure 9.6 PAA-templated fluorescent AgNC logic gates. (I) Transfer of AgNC from PAA to oligonucleotides for (II) AND gate fluorescence enhancement. Source: From Huang et al. [111]. Reproduced with the permission of John Wiley & Sons.

section in proximity to the AgNCs, caused a sharp increase in PL intensity, which was taken as the TRUE output signal. In the AND gate (Figure 9.6), for example, the PAA-AgNCs were transferred to a C-rich nucleotide (input 1), but only produced significant fluorescence upon hybridization of the G-rich strand (input 2). Using the same signal enhancement strategy with different oligonucleotide inputs and starting conditions (e.g. AgNC stabilized either by PAA or different oligonucleotides), the authors were able to create a set of seven logic gates (AND, OR, XOR, XNOR, NOR, NAND, and INH). Another study from Lin et al. used the same AgNC signal enhancement strategy in combination with fluorescent signal from NMM to produce a series of logic gates that were implemented for the detection of pathogenic bacteria genes [112]. Other studies have used metal cation-induced changes in DNA secondary structure [113–115] and proximity of a second AgNC [116] to produce changes in AgNC PL as the output signal of DNA-based logic gates.

9.8 Carbon Nanomaterials

Carbon-based nanomaterials have attracted a great deal of research interest. These materials are based on the honeycomb lattice structure of graphite and include zero-dimensional (CDs, graphene QDs), one-dimensional (carbon nanotubes), and two-dimensional (graphene/GO) nanomaterials. Carbon-based nanomaterials exhibit a range of interesting optical properties and are useful as both fluorophores and quenchers.

9.8.1 Graphene and Graphene Oxide

Graphene is an atomically thin, two-dimensional material made up of a single carbon crystal arranged in a honeycomb lattice [117]. Pristine graphene (containing only sp^2-hybridized carbon) has been shown to be a highly effective quencher of PL. The quenching is a result of efficient non-radiative ET [118]

and has been demonstrated with fluorescent materials including organic dyes [119, 120], QDs [121, 122], and CPs [123, 124]. Graphene is also very efficient at adsorbing organic compounds, notably nucleotides [125] and ssDNA [126], as a result of its extended conjugated π-system and huge specific surface area (2600 m^2 g^{-1}) [127]. Nevertheless, with its low-yield synthesis and aqueous insolubility, pristine graphene is rarely used in fluorescence applications.

GO, typically made through chemical exfoliation of graphite, is a more accessible and more soluble alternative to pristine graphene [128]. The GO structure is similar to that of graphene but is partially oxidized, resulting in a mixture of sp^2- and sp^3-hybridized carbon atoms with a variety of oxygen-containing functional groups attached. The PL quenching properties of GO and the tendency for organic molecules to adsorb to GO arise from sp^2-hybridized regions within the lattice, which retain properties similar to graphene [127]. Chemical reduction of GO alters the ratio of sp^2- and sp^3-hybridized carbons, producing reduced graphene oxide (rGO), which has properties more similar to graphene while retaining much of the aqueous solubility of GO [118]. GO and rGO also exhibit PL that originates from the sp^2 regions or electronic transitions involving the oxygen-containing functional groups [129, 130]. The PL emission can be tuned through the relative numbers of different functional groups, which are determined by the reduction method [131].

DNA adsorption on GO and rGO occurs through π–π stacking between the nucleobases and the hydrophobic sp^2 regions, with binding affinity varying between the DNA bases [127]. Electrostatic repulsion between the negatively charged DNA backbone and the negative charge of some oxygen-containing functional groups on GO and rGO also plays a role in the adsorption interaction, potentially overcoming the attractive forces and preventing adsorption at low ionic strength [132]. As a consequence of the shielding of the aromatic bases in duplex DNA, adsorption is highly selective for ssDNA. The quenching of dye-labeled oligonucleotides by adsorption to GO or rGO can thus be regulated by the presence or absence of a complementary strand. This strategy is commonly used to induce changes in PL.

9.8.2 Logic Gates with Graphene and GO

Wang et al. developed a series of arithmetic logic units using GO as a fluorescence quencher for a fluorescein (FAM)-labeled, single-stranded oligonucleotide (Figure 9.7) [133]. In these systems, the initial state had the FAM-labeled strand adsorbed to a GO sheet, with the fluorescence efficiently quenched by the GO, providing a low background signal. This solution also contained NMM in its low fluorescence state. In the case of the half-adder, the inputs were partially complementary strands capable of either weakly hybridizing with the dye-labeled strand or capable of forming a G-quadruplex-containing duplex. Either of the inputs alone formed a duplex with the FAM-labeled strand, disrupting its attraction to the GO and thereby restoring its fluorescence signal. If both inputs were present, the stronger hybridization between them led to the formation of the G-quadruplex-containing duplex, leaving the FAM-labeled strand undisturbed. In this case, the NMM fluorescence signal was greatly
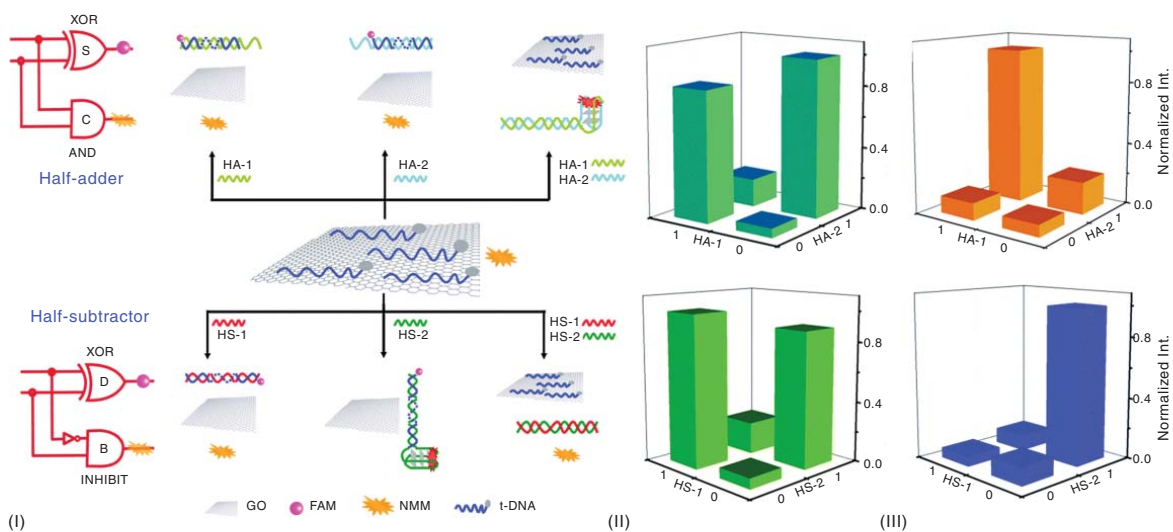
Figure 9.7 Fluorescence quenching by graphene oxide for two-output logic circuits. (I) Cartoon schematics for half-adder and half-subtractor circuits alongside the (II) fluorescein and (III) NMM output signals. Source: From Wang et al. [133]. Reproduced with the permission of Royal Society of Chemistry.

enhanced upon intercalation into the newly formed G-quadruplex. Looking at the inputs individually, this system acted as an AND gate with respect to the NMM signal and as an XNOR gate with respect to the FAM signal, consistent with the "half-adder" arithmetic function. The authors described a similar system with different inputs but the same two output signals to perform the "half-subtractor" function. A subsequent study by the same group used a very similar system to produce three-input logic circuits (full-adder, full-subtract, and majority logic gate), albeit with significantly lower signal contrasts [134].

Several other logic gates have been developed using GO to quench dye-labeled nucleotides. A 2011 study by Lin et al. used a FAM-labeled nucleotide, GO, and silver-metallized DNA to create a series of logic gates [135]. These systems used cysteine and an oligonucleotide as inputs and achieved signal control by either releasing or binding the complement of the FAM-labeled strand. Other studies have used GO quenching in combination with DNA binding of silver ions to create a variety of logic sensors [136, 137]. Another study took advantage of the broad absorption of GO to simultaneously quench two spectrally resolved dyes (fluorescein and rhodamine X) in several sets of paired logic gates [138]. Other examples have included INH gates with ATP/thrombin and oligonucleotide inputs [139] and an AND gate with an oligonucleotide–protein complex and adenosine-5′-O-(3-thiotriphosphate) (ATPγS) as the inputs [140].

9.8.3 Carbon Dots

CDs are photoluminescent nanoparticles composed mainly of carbon [141]. The structure and composition of these materials are varied, but, for the purpose of this chapter, CDs are generalized as roughly spherical nanoparticles with oxygen- and nitrogen-based surface functional groups and various degrees of amorphous or graphite-like carbon within their core. Although CDs have much broader PL emission than QDs, they have good resistance to photobleaching, good quantum yields (up to ∼0.8), and are water soluble [142]. CDs are produced by a variety of top-down and bottom-up methods, the most notable of which are perhaps the "green chemistry" bottom-up synthesis approaches using biomass as a starting material [143, 144]. Oxygen- and nitrogen-based functional groups on the surface of the CD are useful for the conjugation of DNA and other molecules.

The source of PL in CDs is typically attributed to one or more of three mechanisms, though the degree to which each contributes remains a subject of debate [142]. Surface defects in CDs cause trap states, where recombination of trapped excitons can produce emission. Studies have shown that the degree of surface oxidation, and therefore surface defects, and the identity of surface functional groups have strong effects on CD emission [145, 146]. PL has also been shown to arise from internal graphite-like crystalline structure via a quantum confinement effect, with some CDs showing size-dependent PL properties that are unrelated to the surface chemistry [147]. These mechanisms are not mutually exclusive, and it has recently been suggested that CD PL is likely a result of both effects and their interactions with each other [148, 149]. Finally, fluorescent impurities are sometimes responsible for a significant amount of the PL in CD samples produced via bottom-up synthesis from organic precursors, though highly dependent on the synthesis and purification methods [150].
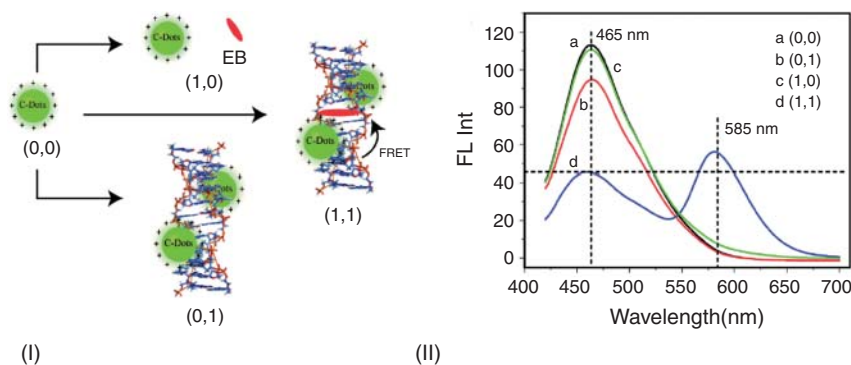
Figure 9.8 Logic gate with CDs (labeled as "C-dots") and EB. (I) Adsorption of the C-dots to dsDNA, enabling FRET to intercalated EB. (II) AND gate response; spectra a, b, and c are FALSE responses, whereas spectrum d is a TRUE response. Source: From Feng et al. [151]. Reproduced with the permission of Oxford University Press.

9.8.4   Logic Gates with CDs

The use of CDs in DNA-based logic gates is limited. Two studies have taken two markedly different approaches to their use. Feng et al. reported the use of positively charged spermine-functionalized CDs as ET donors to ethidium bromide (EB), an intercalating dye, in a DNA duplex [151]. This system exploited the affinity of the positively charged CDs to the negatively charged DNA to bring the CDs into close proximity with the intercalated EB, allowing for FRET sensitization of the EB signal. Starting with the CDs alone, the DNA and EB were taken as inputs, resulting in an AND gate configuration with EB fluorescence as the output signal (Figure 9.8). A NAND gate was also achieved, albeit with lower signal contrast, by taking the quenching of the CD fluorescence as the output signal. In contrast, Gui et al. used CDs directly conjugated with DNA to produce a PL output signal for two logic gates (OR and INH) [152]. The single-stranded oligonucleotides that were conjugated to the CDs adsorbed to GO, resulting in quenching of the CD PL. In the OR gate configuration, addition of a complementary strand and/or mercury ions (causing self-hybridization through T—Hg—T bonds) [153] resulted in dissociation of the CD from the GO surface and therefore restoration of its PL. Though limited in scope, these studies show that CDs have diverse properties that hold promise for future applications in the development of fluorescent DNA logic gates.

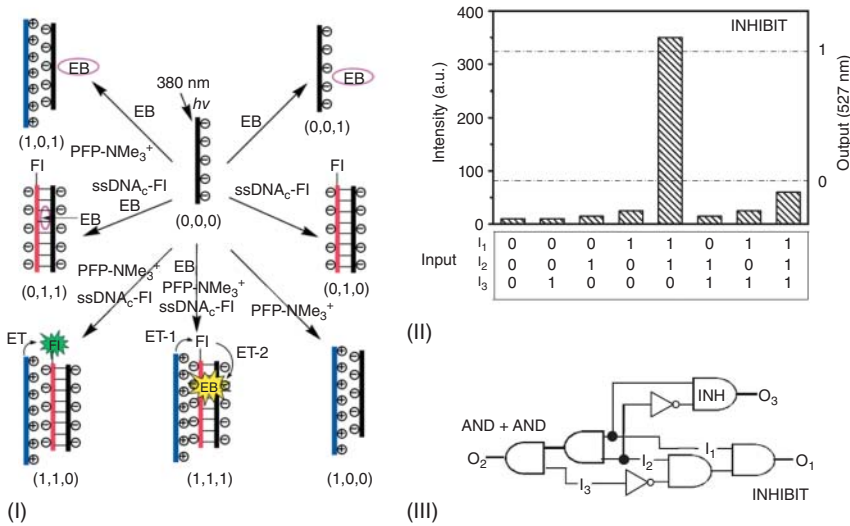## 9.9   Conjugated Polymers

9.9.1   Conjugated Polymers

CPs are a class of luminescent materials that have attracted a lot of interest for use in optical sensing methods because of their ability to amplify fluorescent signal responses. CPs are polymers in which the backbone carbons are sp or sp^2 hybridized, resulting in an overall molecular semiconductor band structure [154].

In certain systems, this band structure results in PL, the strength of which is often related to the delocalization and polarization of the electronic structure [155]. Due to their conjugated structure, CPs also act as very efficient transport mediums for energy. Excitation of the CP results in the generation of an exciton, which is able to easily move throughout a single polymer chain [156]. The mobility of these excitons means that a single molecule attached to the polymer chain can accept ET from excited states generated across a large area of the CP [157]. Using this principle, systems have been designed with CPs as "light-harvesting" energy donors to either quenchers or fluorophores associated with analytes [154]. Notably, this strategy has been used successfully for ultrasensitive detection of specific DNA sequences [158, 159].

Cationic conjugated polymers (CCPs) have been used extensively in DNA-related applications because of their high aqueous solubility and the electrostatic interactions with the negatively charged backbone of DNA [160]. The interaction between certain CCPs and DNA can result in color change and quenching of fluorescence, effects that differ between ssDNA and dsDNA, allowing for simple detection of unlabeled oligonucleotides [161]. To take advantage of their signal amplifying properties, many other systems employ CCPs as FRET donors to organic dye-labeled oligonucleotides or to intercalating dyes within the DNA duplex [162–164]. Intercalating dyes have the particular advantage of specific detection of DNA secondary structures (e.g. dsDNA, G-quadruplexes), allowing for more specific oligonucleotide detection. FRET efficiency between the CCP and intercalating dye is sometimes low due to the fixed orientation of the latter, but use of a dye-labeled oligonucleotide as an ET bridge can greatly enhance the signal [165].

9.9.2 Logic Gates with CPs

Tang et al. used a CCP as a FRET donor to two dyes associated with an electrostatically bound DNA strand to produce a multiply configurable logic gate system [166]. The blue fluorescence of the CCP was detectable in the absence of DNA, and FRET between the CCP and fluorescein on the bound DNA led to quenching of the CCP emission and sensitization of the fluorescein emission. To add an increased level of complexity, EB was used as an input for a third fluorescent signal. Direct FRET between the CCP and EB was not possible due to the rigid, unfavorable orientations of the transition dipoles. The presence of fluorescein, however, allowed for two-step FRET to occur, resulting in sensitization of the EB signal by ET from the CCP via fluorescein. This FRET network was used as the basis for several two- and three-output logic circuits, with the different outputs computing distinct logic operations. In a simple example, the system started with the fluorescein-labeled oligonucleotide bound to the CCP, with EB and a complementary oligonucleotide acting as inputs. With both inputs added, efficient FRET from the CCP to fluorescein to EB resulted in a sensitized EB signal, with the other fluorescent signals quenched. Therefore, the system acted as an AND gate with the EB signal at 600 nm as the output or as a NAND gate with the fluorescein signal at 527 nm as the output. This system was expanded to a three-input, three-output logic circuit, measuring all three fluorescent signals

Figure 9.9 Three-input logic circuit with a conjugated polymer FRET donor. (I) Illustration of the various input states for the logic circuit. (II) Fluorescence signal output for the fluorescein output (O_1), and (III) logic circuit diagram for the three-input, three-output system. Source: From Tang et al. [166]. Reproduced with the permission of John Wiley & Sons.

with EB, the CCP, and the fluorescein-labeled oligonucleotide as inputs to the starting system of an unlabeled oligonucleotide (Figure 9.9).

Liu et al. developed a similar system employing two-step FRET from a CCP to complementary fluorescein- and TAMRA-labeled oligonucleotides [167]. In this case, the FRET pathways were modulated by selective cleavage of the dye-labeled nucleotides by restriction endonucleases HaeIII and PvuII, which were taken as the inputs for the NOR logic gate. Interestingly, the addition of T4 DNA ligase could repair the cleaved DNA restoring the FRET pathway and effectively regenerating the logic gate. Though the signal contrasts were low (typically <2 : 1), the study showed the potential for the use of enzymes to achieve reversibility in DNA logic gates, a highly desirable outcome. Another study used three endonuclease inputs (HaeIII, PvuII, and EcoRV) to disrupt FRET from a CCP to a branched DNA structure with three oligonucleotides labeled with different dyes (fluorescein, Cy5, and Texas Red) [168]. This system was used to construct two- and three-input logic circuits, albeit with low signal contrast (<2 : 1). Finally, a study used pH, temperature, and DNA as inputs to regulate FRET between a CCP and an intercalating dye (Genefinder) to create a series of logic gates [169].

9.10 Conclusions and Perspective

Arguably, the prospective diagnostic and theranostic applications of molecular logic are more likely to have societal impact in the near future than DNA-based computing. Molecular logic-based screening for diseases and infections may help address inefficiencies in healthcare systems and may contribute to the realization

of personalized medicine as an approach to patient care. For the former, the fundamental idea is that complex biomarker information is reduced to a simple output, enabling broader deployment by healthcare workers with less specialized training and fewer resources. For the latter, the idea is that a molecular device would make an autonomous decision about delivery or activation of a molecular therapeutic based on local biomarker information. Collectively, such molecular logic represents a route to "smart" probes, sensors, and therapeutic delivery vectors. DNA-based molecular logic systems are particularly appealing for these applications because of the important roles of DNA and various forms of RNA in biological systems. The versatility and programmability of nucleic acids, as aptly demonstrated by the examples in this chapter and, indeed, throughout this book, are also powerfully enabling in the design of such systems. Nontraditional luminescent materials have excellent potential to augment the capabilities of molecular logic gates. For example, the surface area and chemistry of nanoparticles add new dimensions to the actuation of molecular logic gates, whether through adsorptive interactions, multivalent interactions, or other interactions with their tailorable surface chemistry. The optical properties of nontraditional luminescent materials also offer new capabilities for readout of photonic logic signals. Examples include brighter PL signals or more efficient PL quenching for better contrast between TRUE and FALSE states, mechanisms for the rejection of unwanted background signals, and the facilitation of multicolor systems and PL measurements for parallel logic operations. The optical and PL properties of some materials even enable readout of logic operations via the naked eye or, potentially, using a smartphone camera [170]. Given their properties and the progress to date, it is anticipated that nontraditional luminescent and quenching materials will have important roles in the ongoing development of nucleic acid-based molecular photonic logic.

References

1 Amelia, M., Zou, L., and Credi, A. (2010). *Coord. Chem. Rev.* 254: 2267–2280.
2 Wu, C., Wan, S., Hou, W. et al. (2015). *Chem. Commun.* 51: 3723–3734.
3 Adleman, L. (1994). *Science* 266: 1021–1024.
4 Tregubov, A.A., Nikitin, P.I., and Nikitin, M.P. (2018). *Chem. Rev.* 118: 10294–10348.
5 de Silva, A.P. and Uchiyama, S. (2007). *Nat. Nanotechnol.* 2: 399–410.
6 Resch-Genger, U., Grabolle, M., Cavaliere-Jaricot, S. et al. (2008). *Nat. Methods* 5: 763–775.
7 Medintz, I. and Hildebrandt, N. (2013). *FRET – Förster Resonance Energy Transfer* (eds. I. Medintz and N. Hildebrandt). Weinheim, Germany: Wiley-VCH.
8 Didenko, V.V. (2001). *Biotechniques* 31: 1106–1121.
9 Saghatelian, A., Völcker, N.H., Guckian, K.M. et al. (2003). *J. Am. Chem. Soc.* 125: 346–347.

10 Voelcker, N.H., Guckian, K.M., Saghatelian, A., and Ghadiri, M.R. (2008). *Small* 4: 427–431.

11 Zhang, D.Y. and Seelig, G. (2011). *Nat. Chem.* 3: 103–113.

12 Schaeffer, J.M., Srinivas, N., Ouldridge, T.E. et al. (2013). *Nucleic Acids Res.* 41: 10641–10658.

13 Guo, Y., Yao, W., Xie, Y. et al. (2016). *Microchim. Acta* 183: 21–34.

14 Kamenetskii, M.D.F. (1995). *Annu. Rev. Biochem.* 64: 65–95.

15 Scharf, P. and Müller, J. (2013). *ChemPlusChem* 78: 20–34.

16 Bunka, D.H.J. and Stockley, P.G. (2006). *Nat. Rev. Microbiol.* 4: 588–596.

17 Zhou, Y., Tang, L., Zeng, G. et al. (2016). *Sens. Actuators, B* 223: 280–294.

18 Samanta, A. and Medintz, I.L. (2016). *Nanoscale* 8: 9037–9095.

19 Sapsford, K.E., Algar, W.R., Berti, L. et al. (2013). *Chem. Rev.* 113: 1904–2074.

20 Bawendi, M.G., Steigerwald, M.L., and Brus, L.E. (1990). *Annu. Rev. Phys. Chem.* 41: 477–496.

21 Pinaud, F., Michalet, X., Bentolila, L.A. et al. (2006). *Biomaterials* 27: 1679–1687.

22 Algar, W.R., Susumu, K., Delehanty, J.B., and Medintz, I.L. (2011). *Anal. Chem.* 83: 8826–8837.

23 Petryayeva, E., Algar, W.R., and Medintz, I.L. (2013). *Appl. Spectrosc.* 67: 215–252.

24 Sutherland, A.J. (2002). *Curr. Opin. Solid State Mater. Sci.* 6: 365–370.

25 Mattoussi, H., Mauro, J.M., Goldman, E.R. et al. (2000). *J. Am. Chem. Soc.* 122: 12142–12150.

26 Chan, W.C. (1998). *Science* 281: 2016–2018.

27 Mitchell, G.P., Mirkin, C.A., and Letsinger, R.L. (1999). *J. Am. Chem. Soc.* 121: 8122–8123.

28 Medintz, I.L., Uyeda, H.T., Goldman, E.R., and Mattoussi, H. (2005). *Nat. Mater.* 4: 435–446.

29 Sun, D. and Gang, O. (2013). *Langmuir* 29: 7038–7046.

30 Algar, W.R., Kim, H., Medintz, I.L., and Hildebrandt, N. (2014). *Coord. Chem. Rev.* 263–264: 65–85.

31 Chou, K. and Dennis, A. (2015). *Sensors* 15: 13288–13325.

32 He, X., Li, Z., Chen, M., and Ma, N. (2014). *Angew. Chem. Int. Ed.* 53: 14447–14450.

33 Ambros, V. (2004). *Nature* 431: 350–355.

34 Krützfeldt, J., Rajewsky, N., Braich, R. et al. (2005). *Nature* 438: 685–689.

35 Miao, P., Tang, Y., Wang, B., and Meng, F. (2016). *Anal. Chem.* 88: 7567–7573.

36 Clever, G.H., Kaul, C., and Carell, T. (2007). *Angew. Chem. Int. Ed.* 46: 6226–6236.

37 Freeman, R., Finder, T., and Willner, I. (2009). *Angew. Chem. Int. Ed.* 48: 7818–7821.

38 Bünzli, J.-C.G. and Eliseeva, S.V. (2010). Basics of lanthanide photophysics. In: *Lanthanide Luminescence Photophysical, Analytical and Biological Aspects* (eds. P. Hänninen and H. Härmä), 1–45. Berlin, Heidelberg: Springer.

39 Moore, E.G., Samuel, A.P.S., and Raymond, K.N. (2009). *Acc. Chem. Res.* 42: 542–552.

40 Leif, R.C., Vallarino, L.M., Becker, M.C., and Yang, S. (2006). *Cytometry, Part A* 69: 767–778.

41 Bünzli, J.C.G. and Piguet, C. (2005). *Chem. Soc. Rev.* 34: 1048–1077.

42 Bünzli, J.-C.G. (2015). *Coord. Chem. Rev.* 293–294: 19–47.

43 Bünzli, J.C.G. (2009). *Chem. Lett.* 38: 104–109.

44 Selvin, P.R. (2002). *Annu. Rev. Biophys. Biomol. Struct.* 31: 275–302.

45 Bünzli, J.C.G. (2010). *Chem. Rev.* 110: 2729–2755.

46 Bünzli, J.C.G. and Eliseeva, S.V. (2013). *Chem. Sci.* 4: 1939–1949.

47 Selvin, P.R. (1996). *IEEE J. Sel. Top. Quantum Electron.* 2: 1077–1087.

48 Massey, M., Medintz, I.L., Ancona, M.G., and Algar, W.R. (2017). *ACS Sens.* 2: 1205–1214.

49 Claussen, J.C., Algar, W.R., Hildebrandt, N. et al. (2013). *Nanoscale* 5: 12156.

50 Claussen, J.C., Hildebrandt, N., Susumu, K. et al. (2014). *ACS Appl. Mater. Interfaces* 6: 3771–3778.

51 Fu, P.K.L. and Turro, C. (1999). *J. Am. Chem. Soc.* 121: 1–7.

52 Gao, R.R., Shi, S., Li, Y.J. et al. (2017). *Nanoscale* 9: 9589–9597.

53 Pu, F., Ren, J., and Qu, X. (2014). *ACS Appl. Mater. Interfaces* 6: 9557–9562.

54 Xue, S.F., Lu, L.F., Wang, Q.X. et al. (2016). *Talanta* 158: 208–213.

55 DaCosta, M.V., Doughan, S., Han, Y., and Krull, U.J. (2014). *Anal. Chim. Acta* 832: 1–33.

56 Haase, M. and Schäfer, H. (2011). *Angew. Chem. Int. Ed.* 50: 5808–5829.

57 Zhou, J., Liu, Q., Feng, W. et al. (2015). *Chem. Rev.* 115: 395–465.

58 Chen, S., Yu, Y.L., and Wang, J.H. (2018). *Anal. Chim. Acta* 999: 13–26.

59 Fan, D., Wang, E., and Dong, S. (2019). *Mater. Horiz.* 6: 375–384.

60 Faraday, M. (1857). *Philos. Trans. R. Soc. London* 147: 145–181.

61 Ghosh, S.K. and Pal, T. (2007). *Chem. Rev.* 107: 4797–4862.

62 Amendola, V., Pilot, R., Frasconi, M. et al. (2017). *J. Phys. Condens. Matter* 29: 203002.

63 Link, S. and El-Sayed, M.A. (2002). *J. Phys. Chem. B* 103: 4212–4217.

64 Kelly, K.L., Coronado, E., Zhao, L.L., and Schatz, G.C. (2003). *J. Phys. Chem. B* 107: 668–677.

65 Raschke, G., Kowarik, S., Franzl, T. et al. (2003). *Nano Lett.* 3: 935–938.

66 Himmelhaus, M. and Takei, H. (2000). *Sens. Actuators, B* 63: 24–30.

67 Lepinay, S., Staff, A., Ianoul, A., and Albert, J. (2014). *Biosens. Bioelectron.* 52: 337–344.

68 He, L., Musick, M.D., Nicewarner, S.R. et al. (2000). *J. Am. Chem. Soc.* 122: 9071–9077.

69 Xiao, Y., Patolsky, F., Katz, E. et al. (2003). *Science* 299: 1877–1881.

70 Li, Y., Schluesener, H.J., and Xu, S. (2010). *Gold Bull.* 43: 29–41.

71 Letsinger, R.L., Mirkin, C.A., Elghanian, R. et al. (1999). *Phosphorus, Sulfur Silicon Relat. Elem.* 144: 359–362.

72 Mirkin, C.A., Letsinger, R.L., Mucic, R.C., and Storhoff, J.J. (1996). *Nature* 382: 607–609.

73 Thaxton, C.S., Georganopoulou, D.G., and Mirkin, C.A. (2006). *Clin. Chim. Acta* 363: 120–126.

74 Jiang, Q., Wang, Z.G., and Ding, B. (2013). *Small* 9: 1016–1020.

75 Kreibig, U. and Genzel, L. (1985). *Surf. Sci.* 156: 678–700.

76 Dusemund, B., Hoffmann, A., Salzmann, T. et al. (1991). *Z. Phys. D: At. Mol. Clusters* 20: 305–308.

77 Chegel, V., Rachkov, O., Lopatynskyi, A. et al. (2012). *J. Phys. Chem. C* 116: 2683–2690.

78 Chen, Z., Huang, Y., Li, X. et al. (2013). *Anal. Chim. Acta* 787: 189–192.

79 Elghanian, R., Storhoff, J.J., Mucic, R.C. et al. (1997). *Science* 277: 1078–1081.

80 Liu, J. and Lu, Y. (2003). *J. Am. Chem. Soc.* 125: 6642–6643.

81 Lee, J.S., Han, M.S., and Mirkin, C.A. (2007). *Angew. Chem. Int. Ed.* 46: 4093–4096.

82 Yang, C., Wang, Y., Marty, J.L., and Yang, X. (2011). *Biosens. Bioelectron.* 26: 2724–2727.

83 Li, H. and Rothberg, L. (2004). *Proc. Natl. Acad. Sci. U.S.A.* 101: 14036–14039.

84 Wang, L., Liu, X., Hu, X. et al. (2006). *Chem. Commun.*: 3780–3782.

85 Xu, X., Zhang, J., Yang, F., and Yang, X. (2011). *Chem. Commun.* 47: 9435–9437.

86 Valentini, P. and Pompa, P.P. (2013). *RSC Adv.* 3: 19181–19190.

87 Zhang, L., Wang, Z.X., Liang, R.P., and Qiu, J.D. (2013). *Langmuir* 29: 8929–8935.

88 Sato, K., Hosokawa, K., and Maeda, M. (2003). *J. Am. Chem. Soc.* 125: 8102–8103.

89 Kanayama, N., Takarada, T., Fujita, M., and Maeda, M. (2013). *Chem. Eur. J.* 19: 10794–10798.

90 Yun, C.S., Javier, A., Jennings, T. et al. (2005). *J. Am. Chem. Soc.* 127: 3115–3119.

91 Powell, R.D., Halsey, C.M.R., Spector, D.L. et al. (1997). *J. Histochem. Cytochem.* 45: 947–956.

92 Dulkeith, E., Morteani, A.C., Niedereichholz, T. et al. (2002). *Phys. Rev. Lett.* 89: 12–15.

93 Yang, J., Shen, L., Ma, J. et al. (2013). *ACS Appl. Mater. Interfaces* 5: 5392–5396.

94 Yang, J., Dong, C., Dong, Y. et al. (2014). *ACS Appl. Mater. Interfaces* 6: 14486–14492.

95 Liu, J., Ji, H., Huang, J. et al. (2016). *ChemistrySelect* 1: 347–353.

96 Zheng, J., Nicovich, P.R., and Dickson, R.M. (2006). *Annu. Rev. Phys. Chem.* 58: 409–431.

97 Zhang, L. and Wang, E. (2014). *Nano Today* 9: 132–157.

98 Mooradian, A. (1969). *Phys. Rev. Lett.* 22: 185–187.

99 Zheng, J. and Dickson, R.M. (2002). *J. Am. Chem. Soc.* 124: 13982–13983.

100 Wu, Z. and Jin, R. (2010). *Nano Lett.* 10: 2568–2573.

101 Liu, J. (2014). *TrAC, Trends Anal. Chem.* 58: 99–111.

102 Petty, J.T., Zheng, J., Hud, N.V., and Dickson, R.M. (2004). *J. Am. Chem. Soc.* 126: 5207–5212.

103 Hu, R., Liu, Y.R., Kong, R.M. et al. (2013). *Biosens. Bioelectron.* 42: 31–35.

104 Jia, X., Li, J., Han, L. et al. (2012). *ACS Nano* 6: 3311–3317.

105 Liu, G., Shao, Y., Ma, K. et al. (2012). *Gold Bull.* 45: 69–74.

106 Kennedy, T.A.C., MacLean, J.L., and Liu, J. (2012). *Chem. Commun.* 48: 6845–6847.

107 Choi, S., Dickson, R.M., and Yu, J. (2012). *Chem. Soc. Rev.* 41: 1867–1891.

108 Yeh, H.-C., Sharma, J., Shih, I.-M. et al. (2012). *J. Am. Chem. Soc.* 134: 11550–11558.

109 Fan, D., Zhu, J., Liu, Y. et al. (2016). *Nanoscale* 8: 3834–3840.

110 Wang, G., Zhu, Y., Chen, L., and Zhang, X. (2015). *Biosens. Bioelectron.* 63: 552–557.

111 Huang, Z., Tao, Y., Pu, F. et al. (2012). *Chem. Eur. J.* 18: 6663–6669.

112 Lin, X., Liu, Y., Deng, J. et al. (2018). *Chem. Sci.* 9: 1774–1781.

113 Li, J., Jia, X., Li, D. et al. (2013). *Nanoscale* 5: 6131.

114 Li, T., Zhang, L., Ai, J. et al. (2011). *ACS Nano* 5: 6334–6338.

115 Zhang, L.-P., Zhang, X.-X., Hu, B. et al. (2012). *Analyst* 137: 4974–4980.

116 Gao, R.-R., Yao, T., Lv, X.-Y. et al. (2017). *Chem. Sci.* 8: 4211–4222.

117 Novoselov, K.S. (2004). *Science* 306: 666–669.

118 Loh, K.P., Bao, Q., Eda, G., and Chhowalla, M. (2010). *Nat. Chem.* 2: 1015–1024.

119 Liu, Y., Liu, C.Y., and Liu, Y. (2011). *Appl. Surf. Sci.* 257: 5513–5518.

120 Ramakrishna Matte, H.S.S., Subrahmanyam, K.S., Venkata Rao, K. et al. (2011). *Chem. Phys. Lett.* 506: 260–264.

121 Brus, L.E., Nuckolls, C., Chen, Z. et al. (2010). *ACS Nano* 4: 2964–2968.

122 Cao, A., Liu, Z., Chu, S. et al. (2010). *Adv. Mater.* 22: 103–106.

123 Wang, Y., Kurunthu, D., Scott, G.W., and Bardeen, C.J. (2010). *J. Phys. Chem. C* 114: 4153–4159.

124 Choi, B.G., Hong, W.H., Jung, Y.M., and Park, H. (2011). *Chem. Commun.* 47: 10293–10295.

125 Varghese, N., Mogera, U., Govindaraj, A. et al. (2009). *ChemPhysChem* 10: 206–210.

126 Husale, B.S., Sahoo, S., Radenovic, A. et al. (2010). *Langmuir* 26: 18078–18082.

127 Liu, B., Salgado, S., Maheshwari, V., and Liu, J. (2016). *Curr. Opin. Colloid Interface Sci.* 26: 41–49.

128 Park, S. and Ruoff, R.S. (2009). *Nat. Nanotechnol.* 4: 217–224.

129 Pal, S.K. (2015). *Carbon* 88: 86–112.

130 Mei, Q., Liu, B., Han, G. et al. (2019). *Adv. Sci.* 6: 1900855.

131 Li, M., Cushing, S.K., Zhou, X. et al. (2012). *J. Mater. Chem.* 22: 23374–23379.

132 Liu, J. (2012). *Phys. Chem. Chem. Phys.* 14: 10485–10496.

133 Wang, K., Ren, J., Fan, D. et al. (2014). *Chem. Commun.* 50: 14390–14393.

134 Zhou, C., Liu, D., Wu, C. et al. (2016). *Nanoscale* 8: 17524–17531.

135 Lin, Y., Tao, Y., Pu, F. et al. (2011). *Adv. Funct. Mater.* 21: 4565–4572.

136 Xie, W.Y., Huang, W.T., Li, N.B., and Luo, H.Q. (2012). *Chem. Commun.* 48: 82–84.

137 Lv, H., Li, S., Liu, Y. et al. (2015). *Microchim. Acta* 182: 2513–2520.

138 Liu, X., Aizen, R., Freeman, R. et al. (2012). *ACS Nano* 6: 3553–3563.

139 Wang, L., Zhu, J., Han, L. et al. (2012). *ACS Nano* 6: 6659–6666.

140 Tang, L., Li, D., and Li, J. (2013). *Chem. Commun.* 49: 9971–9973.

141 Kozák, O., Sudolská, M., Pramanik, G. et al. (2016). *Chem. Mater.* 28: 4085–4128.

142 Liu, M.L., Chen, B.B., Li, C.M., and Huang, C.Z. (2019). *Green Chem.* 21: 449–471.

143 Suryawanshi, A., Biswal, M., Mhamane, D. et al. (2014). *Nanoscale* 6: 11664–11670.

144 Briscoe, J., Marinovic, A., Sevilla, M. et al. (2015). *Angew. Chem. Int. Ed.* 54: 4463–4468.

145 Bao, L., Zhang, Z.L., Tian, Z.Q. et al. (2011). *Adv. Mater.* 23: 5801–5806.

146 Wang, L., Zhu, S.J., Wang, H.Y. et al. (2014). *ACS Nano* 8: 2541–2547.

147 Li, H., He, X., Kang, Z. et al. (2010). *Angew. Chem. Int. Ed.* 49: 4430–4434.

148 Zhu, S., Song, Y., Wang, J. et al. (2017). *Nano Today* 13: 10–14.

149 Yang, T., Wang, N., Li, N. et al. (2018). *Sci. China Chem.* 61: 490–496.

150 Essner, J.B., Kist, J.A., Polo-Parada, L., and Baker, G.A. (2018). *Chem. Mater.* 30: 1878–1887.

151 Feng, L., Zhao, A., Ren, J., and Qu, X. (2013). *Nucleic Acids Res.* 41: 7987–7996.

152 Gui, R., Jin, H., Wang, Z. et al. (2015). *Nanoscale* 7: 8289–8293.

153 Cui, X., Zhu, L., Wu, J. et al. (2015). *Biosens. Bioelectron.* 63: 506–512.

154 Alvarez, A., Costa-Fernández, J.M., Pereiro, R. et al. (2011). *TrAC, Trends Anal. Chem.* 30: 1513–1525.

155 McQuade, D.T., Pullen, A.E., and Swager, T.M. (2000). *Chem. Rev.* 100: 2537–2574.

156 Thomas, S.W., Joly, G.D., and Swager, T.M. (2007). *Chem. Rev.* 107: 1339–1386.

157 Fan, L.J., Zhang, Y., Murphy, C.B. et al. (2009). *Coord. Chem. Rev.* 253: 410–422.

158 Ren, X. and Xu, Q.H. (2009). *Langmuir* 25: 43–47.

159 Najari, A., Ho, H.A., Nobert, P. et al. (2006). *Anal. Chem.* 78: 7896–7899.

160 Liu, B. and Bazan, G.C. (2004). *Chem. Mater.* 16: 4467–4476.

161 Ho, H.-A., Najari, A., and Leclerc, M. (2008). *Acc. Chem. Res.* 41: 168–178.

162 Pu, K.Y. and Liu, B. (2009). *Adv. Funct. Mater.* 19: 1371–1378.

163 Feng, F., Tang, Y., He, F. et al. (2007). *Adv. Mater.* 19: 3490–3495.

164 Feng, F., Liu, L., and Wang, S. (2010). *Nat. Protoc.* 5: 1255–1264.

165 Wang, S., Gaylord, B.S., and Bazan, G.C. (2004). *J. Am. Chem. Soc.* 126: 5446–5451.

166 Tang, Y., He, F., Wang, S. et al. (2006). *Adv. Mater.* 18: 2105–2110.

167 Liu, Y., Tang, Y., and Cao, A. (2013). *Polym. Chem.* 4: 5206–5211.

168 Feng, X., Duan, X., Liu, L. et al. (2009). *Angew. Chem. Int. Ed.* 48: 5316–5321.

169 Pu, F., Wang, C., Hu, D. et al. (2010). *Mol. Biosyst.* 6: 1928.

170 Petryayeva, E. and Algar, W.R. (2015). *RSC Adv.* 5: 22256–22282.

10

Programming Spatiotemporal Patterns with DNA-Based Circuits

Marc Van Der Hofstadt[1], *Guillaume Gines*[2], *Jean-Christophe Galas*[1], *and André Estevez-Torres*[1]

[1] *Sorbonne Université and CNRS, Laboratoire Jean Perrin, 4 place Jussieu, 75005, Paris, France*
[2] *CNRS, ESPCI Paris, PSL Research University, Laboratoire Gulliver, 10 rue Vauquelin, 75005, Paris, France*

Common ways of computing do not use physical space to perform a single calculation [1]. However, in the physical world, and in particular in living systems, space has a major influence in the outcome of computations. In this chapter we will discuss DNA programs that take spatial inputs and compute spatial outputs. We will focus on systems that perform these calculations by reaction–diffusion (RD), an important mechanism to describe the spatial behavior of large groups of molecules. We will first introduce basic concepts such as spatial and analog computing and energy consumption in molecular computing. We will then briefly review the three current experimental implementations that allow to do so with DNA programs: DNA strand displacement (DSD), genelets, and PEN DNA reactions. We will then discuss time-dependent spatial patterns that have been demonstrated with these systems, such as edge detection and traveling patterns. We will next make a survey of recent methods to control the parameters that influence the computation, in particular reaction and diffusion rates and boundary conditions. We will end by describing the design of steady-state patterns such as band patterns, which are relevant in early embryo development, and providing some perspectives for the future.

10.1 Introduction

10.1.1 What is Spatial Computing?

The majority of computations in everyday life are performed by microprocessors made of transistors. Within microprocessors, a computation is decomposed in operations that are carried out sequentially in time, thanks to a central clock. The spatial position of the input information or of the computing transistors does not influence the result. The opposite is true in many natural systems. For instance, groups of individual living agents use algorithms where the spatial position plays a crucial role. This is the case of herds of animals – such as birds or bees – where

collective behaviors emerge from local interactions that are regulated by the behavior of nearby individuals [2]. It is also observed in developing embryos, where the final shape of the organism, but also the biochemical composition of each cell, depend on position. We thus define spatial computing as any form of computation that is influenced by spatial coordinates, in particular because the physical process that performs the computation depends on space.

10.1.2 Digital vs. Analog Computing

Computing can be digital or analog. Digital computing works with discrete signals, while analog computing operates with continuous ones. Digital and analog computing differ in two important points: the nature of the computing primitives and the propagation of noise [3]. In digital computing the primitives are based on the mathematics of Boolean logic (AND, OR, etc.), and the integration of a large number of these primitives into a complex program is a science that can be rationalized and automated. In contrast, in analog computing, the primitives are based on the physics of the computing system, such as the charge and discharge of a capacitor or the kinetics of a chemical reaction. Physical primitives have the advantage of being more efficient than Boolean ones to perform a given calculation. However, combining physical primitives to perform complex calculations is an art difficult to rationalize and automate. This is a problem for engineers, but not for natural systems that have spent their evolutionary time trying out the most efficient ways to implement analog computations that are useful for survival.

DNA computing can also be digital or analog. An example of digital implementation are logic gates based on DSD reactions [4, 5]. However, spatial computations have mainly involved analog implementations whose computing primitives are given by chemical kinetics, and we will discuss them in Section 10.2.

10.1.3 Computing Consumes Energy

Because any computation implies the transformation of a physical system, it must consume energy. Energy supply is relatively straightforward in electronics through the use of power supplies. In molecular systems, computations are performed by chemical reactions and thus need "chemical supplies" to run continuously. "Chemical supplies" are ubiquitous in living systems – that is why we eat and breath – but are difficult to engineer in synthetic systems. The reason is that in electronics we have spatial separation through cables and we can thus use a single power supply that provides high-voltage electrons for all the computing elements. In molecular systems, the computing reactions are all mixed in solution: they are thus connected through similar reactivities and isolated from each other using orthogonal reactivities.

Let's consider a series of orthogonal reactions $R_i \rightarrow P_i$ that are thermodynamically favorable (and thus their free energy change $\Delta_i G < 0$). A "chemical supply" is a process that provides enough free energy to drive the conversion $P_i \rightarrow R_i$. Ideally, this recycling process is a chemical reaction that turns fuel F into waste W, $F \rightarrow W$, with an associated free energy change $\Delta_{cs} G < \sum_i \Delta_i G$. We thus need a single reaction that shares reactivity with many reactions that are orthogonal

to each other, which is very hard to accomplish with the chemistry of small molecules. Nature solved this issue by evolving enzymes, large molecules that bear two (or more) orthogonal reactivities: one that is specific to a particular substrate and a second one that consumes a common fuel shared by a large set of enzymes, typically adenosine triphosphate (ATP).

In DNA computing several solutions to this problem exist:

1) *One-shot computations in a closed reactor*. In most implementations the reactants are mixed in a closed reactor without "chemical supply," and the solution can only perform a given computation once. This is the case of DSD logic gates [4, 5], for instance.
2) *Long transients in a closed reactor*. In some instances, the closed reactor contains a "chemical supply." In the case of strand displacement, you cannot choose at the same time orthogonal sequences for the reactants (called gates; see below) and shared sequences for the fuels. The solution is thus to use as many fuel molecules as there are gates in the reaction [6]. DNA/enzyme computing systems, such as genelets and PEN reactions, use DNA hybridization to make orthogonal reactions and DNA-dependent enzymatic reactions coupled to an ATP-like fuel to implement a single "chemical supply" that is orthogonal to the hybridization chemistry. If the fuel is consumed slowly compared with the timescale of the computing reactions, such implementation can maintain the system out of equilibrium in a closed reactor for long enough to perform complex computations.
3) *Long transients in an open reactor*. Another way to implement a "chemical supply" that recycles products back into reactants is to run the reactions in an open reactor that exchanges matter with the external world. An open reactor is physically connected to a source that flows in fresh reactants and to a sink that takes away the reacted mixture. This way, the reactor is constantly traversed by a free energy flow that keeps the system out of equilibrium.

10.1.4 Molecules Compute in Space Through Reaction–Diffusion Primitives

In this chapter we will review recent examples of spatial computations performed with DNA programs using RD primitives, because this mechanism is pervasive to reacting molecules in solution [7]. We will not discuss patterns created by DNA programs in the absence of diffusion, such as those driven by self-assembly processes in DNA nanostructures, or patterned materials created from them, which are reviewed elsewhere [8–10], nor RD patterns generated by protein [11, 12] or transcription–translation networks [13, 14]. We further refer the interested reader to a recent review on pattern generation with DNA programs [15].

In the absence of space (for instance, if the reactor is well mixed), the primitives of molecular computing are ruled by the kinetics of chemical reactions. For instance, the reaction of two single-stranded DNA (ssDNA) A and B to give the double-strand C is written:

$$A + B \underset{k_2}{\overset{k_1}{\rightleftharpoons}} C \tag{10.1}$$

where k_1 and k_2 are the hybridization and dehybridization kinetic rate constants, respectively. Supposing mass action law kinetics, the temporal evolution of the concentration of species C is given by

$$\frac{\mathrm{d}C}{\mathrm{d}t} = k_1 A \cdot B - k_2 C \tag{10.2}$$

where the concentration of a given species is noted in italics. Equation (10.2), together with similar equations for species A and B, is the computing primitive of a bimolecular reaction.

In the absence of mixing, the transport of each species by diffusion must be taken into account. For instance, in a one-dimensional (1D) reactor, the spatiotemporal evolution of C involved reaction (10.2) is given by

$$\frac{\partial C}{\partial t} = k_1 A \cdot B - k_2 C + D_C \frac{\partial^2 C}{\partial x^2} \tag{10.3}$$

where D_C is the diffusion coefficient of species C and x the spatial coordinate. Equation (10.3), together with the corresponding equations for A and B, is the primitive for the RD dynamics of a bimolecular reaction.

In the general case where n reactive species form a reaction network characterized by the reaction matrix F, RD dynamics are given by

$$\frac{\partial u_i}{\partial t} = F_i(u_1, \dots, u_n) + D_i \frac{\partial^2 u_i}{\partial x^2}, \qquad i = 1, \dots, n, \tag{10.4}$$

where u_i is the concentration of species i and D_i its diffusion coefficient. The term reaction–diffusion was coined by Alan Turing in his seminal work *The chemical basis of morphogenesis* [16]. The interest of RD systems is that they make emerge a spatial distance $\lambda = \sqrt{D/k}$, where k is a first-order rate constant characteristic of the reaction kinetics and D a diffusion coefficient [17]. RD is thus a convenient way of computing distances that depend on chemical inputs and that provide chemical outputs. Under some circumstances [17, 18], a system obeying Eq. (10.4) generates spatial structures of wavelength λ, namely,

- edge detectors;
- traveling fronts, waves, and spirals;
- turing patterns; and
- stationary fronts and band patterns.

All these patterns have been observed and investigated, principally between 1970 and 2000, with reactions based on the redox chemistry of small molecules, of which an archetypal example is the Belousov–Zhabotinsky (BZ) oscillator [19]. However, redox chemistry is not programmable, and its harsh acidic conditions make it incompatible with biological materials. Engineering RD patterns with DNA programs solves these two issues. In Sections 10.3 and 10.4, we will see how these patterns – except Turing ones – have been engineered with DNA programs.

10.2 Experimental Implementation of DNA Analog Circuits

In this section we discuss the three DNA-based experimental systems that can currently perform analog computations coupled to a "chemical supply" and that

are thus amenable to nontrivial RD computing: DSD, genelets, and PEN reactions. We review them by showing how to implement a cornerstone of nonequilibrium dynamics: a chemical oscillator.

10.2.1 DNA Strand Displacement Oscillators

Toehold-mediated DSD was developed by Yurke et al. as a way to use ssDNA as a catalyst to fuel DNA-based nanomachines [20, 21]. Their simple and powerful idea was to control the kinetics of dsDNA dehybridization through a hybridization event: if the partially double-stranded complex t:b carries a dangling end on the bottom strand b, then the dissociation of the top strand t is significantly faster in the presence of strand b*, fully complementary to b, because b* may hybridize to the dangling end of t:b, called toehold, and eject t by strand displacement. This strategy introduces two new features compared with standard DNA hybridization between fully complementary sequences: (i) one can quantitatively control the rate of production of species t by up to six orders of magnitude by changing the length or the position of the toehold [22, 23], and (ii) every DSD reaction can be used to generate a new toehold that may subsequently react, thus opening the path to using ssDNA as a building block of chemical reaction networks.

Although DSD reactions were first applied to digital computing by Winfree and collaborators [4, 5, 24], they can also be used in analog computations. This was first proposed theoretically in 2010 [6] and recently demonstrated experimentally by synthesizing a DSD oscillator [25].

To design DSD analog computations, one first chooses a suitable formal mechanism. As an example, let's consider the bimolecular reaction $A + C \rightarrow B$. Species A, C, and B are encoded with ssDNA strands composed of a species-specific and a reaction-specific domain (respectively, colored and black in Figure 10.1a). The reaction is implemented in two steps with two gates R_{ACI_B} and $R_{I_B B}$ (Figure 10.1b), which are DNA complexes composed of two or more partially hybridized DNA strands bearing reactive toeholds. First, the step $A + C \rightarrow I_B$ is implemented by R_{ACI_B} that is an AND gate that produces intermediate I_B in the presence of both A and C. A second gate $R_{I_B B}$ takes the released species I_B as an input and produces B.

Autocatalytic reactions of the type $A + B \rightarrow 2A$ can be experimentally implemented by using a gate that takes two different inputs and generates two identical outputs [25]. The leak inherent to any autocatalytic reaction may be efficiently suppressed by adding a thresholding module that suppresses the output that detaches from the gate in the absence of the input. By connecting three of these autocatalytic modules that repressed each other, Srinivas et al. [25] succeeded the tour de force of synthesizing a DSD oscillator in a closed reactor, thus proving for the first time that complex analog networks with feedbacks can be built and kept out of equilibrium with DSD reactions (Figure 10.1c, d).

10.2.2 DNA/Enzyme Oscillators

DSD networks have the advantage of being fully programmable; however up to seven DNA species are needed to implement a single autocatalytic node [25], four of them being fuel molecules, which increases the number of control
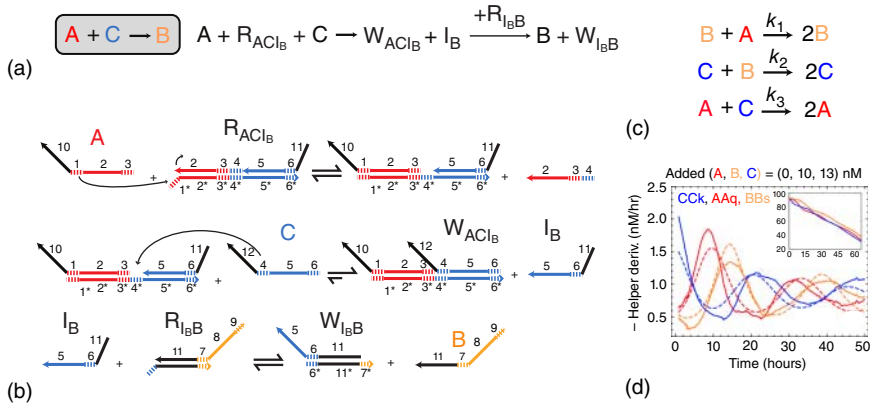
Figure 10.1 Principle of DNA strand displacement (DSD) reactions (a,b) and oscillations in a DSD circuit in a closed reactor (c,d). (a,b) DSD implementation of the reaction kinetics $A + C \rightarrow B$. The reaction needs the reaction gates R_{ACI_B} and R_{I_BB} that act as fuels and are converted into waste W_{ACI_B} and W_{I_BB} (a). (b) Detailed mechanism of the reaction in (a): ssDNA are noted as arrows, the $3'$ end being at the tip of the arrow. Sequence domains are indicated by numbers, and an asterisk * denotes their complementary sequence. Toeholds are represented as dashed lines. (c) Formal reaction mechanism of the DSD rock-paper-scissors oscillator. (d) Experimental results showing oscillations of the network in panel (c) in a closed reactor (solid lines: experimental data, dashed lines: model fits). Source: Panels (c,d) From Srinivas et al. [26]. ©2017. Reproduced with the permission from the authors [25].

parameters to be optimized to obtain the desired dynamics. A way to circumvent this problem is to use enzymes that catalyze the conversion of an input into an output strand and that use a common fuel that is orthogonal to DNA hybridization chemistry. In this regard, a powerful idea is to emulate what happens in gene regulatory networks, where genes produce protein transcription factors (TFs) that increase or reduce the rate of production of other TFs, all being degraded by a specific enzyme.

This idea was first implemented in 2006 by Kim et al. [27], who built a bistable network and later an oscillator [28]. To do so, they combined DNA and RNA strand displacement reactions with transcription and RNA degradation assisted by two enzymes, RNA polymerase (RNAP) and RNAse. This reaction framework receives the name of genelet. A similar idea was implemented in a different manner by Rondelez and coworkers who built a three-node relaxation oscillator [29] and later a predator–prey (PP) oscillator [30] and two bistables [31, 32]. For this they used short DNA strands and three enzymes, a polymerase, an exonuclease, and a nicking enzyme, in a reaction framework called PEN DNA toolbox (PEN stands for the first letters of the three enzymes involved).

10.2.2.1 Genelets
Genelets are constituted of three types of nucleic acid species (Figure 10.2a). Nodes, O_i, are RNA strands that are created and degraded and thus play the role of TFs. Switches, S_{ij}, are partially dsDNA species that may create nodes by transcription depending on their activity state and are thus equivalent
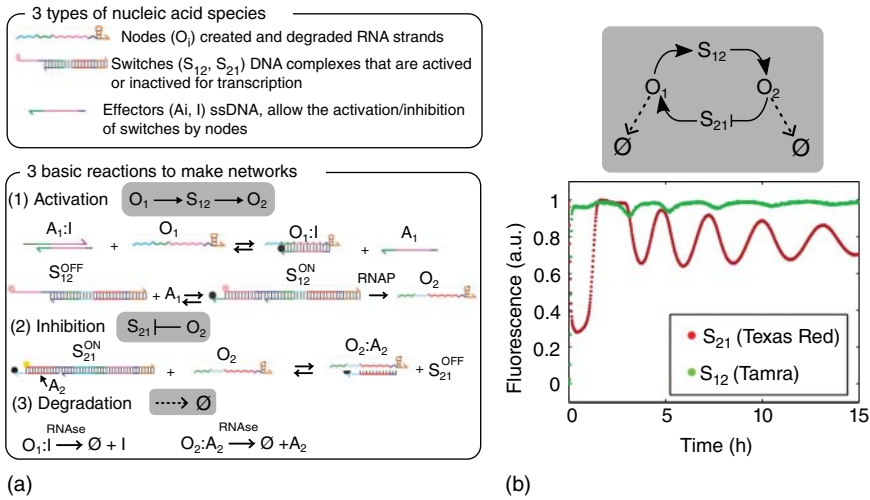
Figure 10.2 Mechanism of the genelet reaction system (a) and network oscillating in a closed reactor (b). Harpoon arrows denote ssDNA, curly arrows refer to RNA, and colors indicate sequence domains, similar colors indicating complementary sequences. RNAP stands for RNA polymerase. Source: Adapted from Kim et al. [28].

to genes. Effectors, A_i and I, are ssDNA species that make the link between nodes and switches. To construct networks with genelets, one needs to modulate the activity of switches S_{ij} with input nodes O_i, which is performed by a combination of strand displacement and transcription reactions.

In its OFF state, noted S_{ij}^{OFF}, a switch carries an incomplete RNAP promoter sequence and is thus inactive for transcription. S_{ij}^{OFF} is activated in the presence of A_i that binds to S_{ij}^{OFF} and completes the RNAP promoter sequence, generating species S_{ij}^{ON}. In Figure 10.2, inhibition of S_{21}^{ON} by O_2 is directly made by the hybridization of O_2 with strand A_2 on S_{21}^{ON}, generating S_{21}^{OFF} and making the partial DNA duplex $O_2:A_2$. Activation of S_{12}^{OFF} by O_1 is made through the intermediary of complex $A_1:I$. O_1 reacts with $A_1:I$ and liberates A_1 that activates S_{12}^{OFF}. In the presence of RNAseH, which specifically degrades RNA hybridized to DNA, species $O_1:I$ and $O_2:A_2$ continuously regenerate I and A_2 and destroy the nodes O_i. The production of RNA by consuming nucleotide triphosphates (NTPs) and its degradation through RNAse ensure that the network is kept out of equilibrium for 10–15 hours in a closed reactor. Genelets have been used to construct bistable [27, 33] and oscillatory networks [28, 34].

10.2.2.2 PEN Reactions

PEN reaction networks are assembled with two types of species (Figure 10.3a). Templates, T_{ij}, are ssDNA species that carry the information about the topology of the network, and their concentrations do not change over time; they are equivalent to genes. Nodes (A, B, I) are shorter ssDNA strands that are processed and created by the templates; they are both produced and degraded over time; they play the role to TFs. These two species, together with three enzymes, can
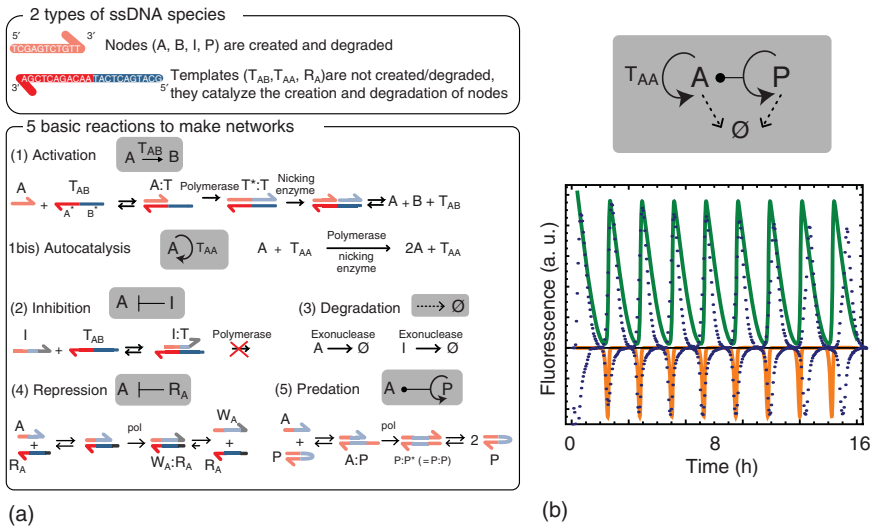
Figure 10.3 Mechanism of the PEN DNA toolbox reaction system (a) and network oscillating in a closed reactor (b). (a) Harpoon arrows denote ssDNA, and colors indicate sequence domains, similar colors indicating complementary sequences. (b) PEN toolbox predator–prey oscillator. Topology of the reaction network (top) and experimental data (dots) and fits to a two-variable model (lines). P appears in green and A in orange.

perform five types of reactions in the presence of dNTPs: activation, inhibition, degradation, repression, and predation.

Template T_{AB} catalyzes the activation reaction $A \rightarrow A + B$, where A and B are nodes. T_{AB} is typically 20–25 nt long, and it carries two sequence domains, the input domain, of sequence A^*, and the output domain, noted B^*, respectively, complementary to A and B. Nodes are typically 10–15 nt long. During activation, A binds to the input site of T_{AB} to form species $A{:}T_{AB}$, which is extended by a DNA polymerase, pol, into dsDNA species $T^*_{AB}{:}T_{AB}$ (T^*_{AB} has the sequence A–B). A nicking enzyme, nick, recognizes a 5 or 6 nt long sequence on $T^*_{AB}{:}T_{AB}$ and cuts the upper strand T^*_{AB} between domains A and B, which dehybridizes into species A, B, and T_{AB}. The temperature is chosen in the range 37–45° such that the complex $A{:}T_{AB}$ is close to the melting temperature but $T^*_{AB}{:}T_{AB}$ is stable. Autocatalysis can be encoded in an activation template T_{AA} whose input and output domains are identical. Note that, as it happens with any autocatalytic reaction, PEN autocatalysis "leaks," i.e. it starts in the absence of input A, because the polymerase is able to synthesize A in the absence of template [35]. In standard conditions this leak happens within 100 minutes, but it can be simply reduced in the presence of high concentrations of nicking enzyme to reach 10 hours, and even totally suppressed in the presence of repression (see below), which turns the monostable autocatalytic node into a bistable one [32]. A second side reaction of PEN autocatalysis is the generation of autocatalytic parasites, which result from untemplated autocatalysis [35]. This reaction generates mixtures of DNA strands spanning 10 to several 1000 nucleotides

that ultimately break the designed dynamics of PEN networks at long times. Depending on the conditions, parasites emerge after 5–50 hours. However, these parasites may be suppressed from functional PEN networks by adopting a three-letter encoding [36].

Inhibition of T_{AB} is performed by strand I, typically 15 nt long, that partially hybridizes to domains A^* and B^* on T_{AB}, forming I: T_{AB}. A single-stranded overhang on the $3'$ end of I bound to T_{AB} precludes the polymerase to extend it, and a careful choice of its sequence prevents nick from cutting it. Degradation of nodes is performed by a ssDNA exonuclease that does not degrade templates because they are chemically modified on their $5'$ end. Repression of an autocatalytic node can be implemented by adding a template R_A that takes A as an input and adds a short sequence to its $3'$ end, converting it into a waste product, W_A, unable to react with the template T_{AA}. In this configuration, when R_A is a degradable node with palindromic sequence, noted P, a predation reaction of the type A + P → 2P can be implemented. To construct a network, activating and inhibiting links are selected such that the sequence of output B is, respectively, the input or the inhibitor of a downstream template. The continuous production and degradation of node strands, which consumes deoxynucleotide triphosphates (dNTPs), keeps the network out of equilibrium in a closed reactor.

The PEN toolbox has produced so far oscillators that are significantly more robust than those reported using genelets (Figures 10.2b and 10.3b). The two- and three-node genelet oscillators [28, 34] oscillate for six periods during 20 hours and for three periods during 15 hours, respectively. In contrast, the two-node PP [30] and the three-node oligator [29] PEN oscillators oscillate for 26 periods for 32 hours and for 18 periods during 30 hours, respectively. Unpublished results with the PP oscillator demonstrate more than 100 periods for more than 130 hours. The reason of this greater robustness may be attributed to a cleaner degradation mechanism in the PEN system; while RNAse H only partially degrades RNA strands, the DNA exonuclease used in PEN reactions transforms the nodes into single nucleotides, reducing inhibition by degradation products. Table 10.1 recapitulates the properties of the three analog implementations reviewed here and summarizes, to the best of our knowledge, the temporal and spatial patterns obtained with them.

10.3 Time-Dependent Spatial Patterns

In this section we review experimental realizations of spatial calculations which output is a spatial concentration pattern that depends on time. We will first see how an incoherent feed forward loop network performs edge detection and how an autocatalytic program generates propagating concentration fronts. We will then describe go-fetch fronts that are able to interrogate the presence of a particular DNA sequence at a distance and traveling waves and spirals based on DNA oscillators. Finally, we will review methods to control the diffusion coefficient and the geometry of the environment where the spatial pattern evolves.

Table 10.1 Principal characteristics of the three experimental systems capable of RD computations with DNA. Programmable and non-programmable sizes refer to the number of bases of the elements that can be designed or not in the different implementations (non-programmable corresponds to enzymes), extracted from [25]. The timescale and the lifetime correspond to the typical period and duration of the oscillations in a closed reactor. Types of temporal and spatial patterns experimentally implemented with each system. Numbers in brackets point to references.

	DSD	Genelets	PEN toolbox
Programmable size (nt)	1386	469	71
Non-programmable size (bp)	0	∼ 4000	∼ 7700
Timescale (h)	20 [25]	3 [28]	1.5 [30]
Lifetime (h)	50 [25]	20 [28]	80 [37]
Temporal pattern	Autocatalyst [38], oscillator [25]	Bistable [27, 33], oscillator [28, 34]	Autocatalyst [29], bistable [31, 32], oscillator [29, 30], chaos [30], excitable [32]
Spatial pattern	Edge detection [39], linear gradient [40], band pattern [40]	Traveling front [41], pulse [42]	Traveling front [43, 44], wave [45] and spiral [45]; go-fetch front [46]; band pattern [47, 48]; french flag pattern [47], colony formation [46]

10.3.1 Edge Detection

RD programs can perform some image-processing algorithms where the input and output are images encoded as concentration patterns. This was first demonstrated using the Belousov–Zhabotinsky reaction in 1989 [49]. In 2013, Chen and coworkers engineered DNA reaction networks that detected the edge of an input image (Figure 10.4) [39]. The computation was performed by a reaction network where the input I comes in the form of light and A, B, and C are DNA strands. On the one side, I activates A and triggers the cascade A → B → C. On the other side, I inhibits B and thus inhibits de cascade B → C. Because I activates and inhibits C, such network is called an incoherent feed forward loop (IFFL). In addition, A needs to diffuse faster than B and C. Let's now consider that we illuminate the medium with a pattern with a border between light and no light (1/0, Figure 10.4a). The illuminated zone will produce A and destroy B, and thus no output C will be observed in this zone. The dark zone will not produce A nor B, but A will diffuse across the border from the illuminated zone and react with B, producing C at the border between the two zones. The low diffusion of B and C makes the generated concentration profile of C sharper.

The authors implemented this program by converting the input light pattern into DNA concentration through caged DNA strands containing a photocleavable nitrophenyl amide spacer. The network was based on the catalyzed hairpin
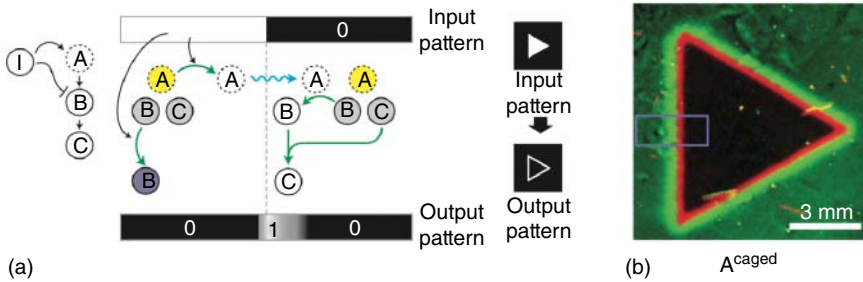
Figure 10.4 Reaction–diffusion edge detection pattern engineered with DNA strand displacement circuits. (a) Mechanism of the incoherent feed forward loop (IFFL) DNA network. Light is denoted as the input species I, and the only fast-diffusing species is A. A and B are photoresponsive DNA strands that, upon illumination, break into two strands, inactive B uncapable of producing C in the illuminated area, and active A that diffuses into the non-illuminated area and activates B and fluorescent product C, creating the highlighted edge. (b) Fluorescent pattern obtained when flashing light with a triangular shape in a gel filled with an IFFL DNA network. Source: From Chirieleison et al. [39]. ©2013. Reproduced with the permission of Springer Nature.

assembly (CHA) reaction [50], which is related to DSD but implements gates using hairpins, instead of duplexes. For instance, A was an unreactive hairpin that was cleaved by UV light and formed a reactive ssDNA. In contrast, B was a reactive hairpin whose toehold was cleaved by light, yielding an unreactive species. Figure 10.4b shows the experimental output pattern in red. Note that, because there is no "chemical supply" in this implementation, the output pattern is transient and will fade away by diffusion after some time. To our knowledge, this was the first experimental demonstration of RD patterns programmed with DNA.

Complementary to edge detection, Abe et al. demonstrated the computation of a line segment equidistant from two source points [51]. These experiments were also performed in a gel matrix. A DNA logic AND gate was anchored everywhere in the gel, and two holes made in the gel were filled with the gate inputs. Both inputs diffused through the matrix and activated the AND gate only at the equidistant region from the source points, thus producing a Voronoi pattern. More complex patterns were observed when multiple source points were involved.

10.3.2 Traveling Patterns

In the absence of reaction, the diffusion of a chemical species is quite boring: a concentration profile will fade away until reaching a spatially homogeneous final state, following Fick's diffusion law. In contrast, in the presence of an autocatalytic reaction, which is a transformation where the product catalyzes its own production, an inhomogeneous concentration profile will generate concentration patterns that propagate, often with constant velocity. Traveling patterns are an efficient way to convey chemical information across distances where diffusion is too slow.

10.3.2.1 Fronts

The simplest traveling pattern is the front (a structure with a single low-to-high concentration transition) that just needs a single autocatalytic loop, which we can simply write

$$A \xrightarrow{k} 2A \tag{10.5}$$

where k is the reaction rate. The first observation of a traveling front of concentration was reported by Luther [52] (translated in [53]) in a redox reaction. Luther proposed an expression for the velocity v of propagation of the front that is still valid:

$$v = a\sqrt{kD}, \tag{10.6}$$

where a is "a constant between 2 and 10," k the rate constant of the autocatalytic reaction, and D the diffusion coefficient of the autocatalyst. Luther's formula may be obtained by an order of magnitude argument. Let's consider an autocatalytic species A initially distributed along a one-dimensional reactor with a front concentration profile. Let's define the characteristic time of the autocatalytic reaction $\tau_{\text{chem}} = 1/k$. The distance traveled by diffusion during this time is $l_{\text{diff}} = 2\sqrt{D\tau_{\text{chem}}}$. For times shorter that τ_{chem}, A at the tip of the front diffuses and does not react. For times longer than τ_{chem}, the autocatalytic reaction amplifies A until saturating all the regions where A has diffused, regenerating an identical front ahead of the initial one. The velocity of such a front is thus $v \sim l_{\text{diff}}/\tau_{\text{chem}} = 2\sqrt{kD}$. The grounds for the theory of traveling fronts were independently developed by Fisher [54] and by Kolmogorov et al. [55] in 1937. In particular, $v = 2\sqrt{kD}$ is exact for a single-variable RD system following Eq. (10.4) with a reaction function $F(u)$ that is monostable and verifies $F(u)/u < F'(u)$ [56]. Such systems are called Fisher–KPP fronts. The interested reader may refer to Refs. [17, 56–58].

In 2015, Zadorin et al. demonstrated that programmable traveling fronts could be obtained with PEN autocatalyzers [43]. The autocatalytic behavior of node A growing on template T_{AA} is demonstrated in Figure 10.5a, where the concentration of A:T_{AA} is measured by fluorescence in the presence of a DNA intercalator and plotted *vs.* time. At short time (and thus low A), the system behaves as $A = A(0)e^{kt}$, where k is the rate constant of autocatalysis.

In a one-dimensional reactor such as the one depicted in Figure 10.5b – filled homogeneously with a solution containing template T_{AA}, the polymerase and nicking enzymes, and dNTPs, and containing an excess of autocatalytic node A on the left-hand side – a front propagating with uniform velocity, typically 60 μm min^{-1}, was observed through time-lapse fluorescence microscopy (Figure 10.5c). To check if the front followed Luther's equation (10.6), its velocity was measured for different reaction rates k and different effective diffusion coefficients D. The rates were controlled by changing the concentration T_0 of the template (in a certain concentration range, $k \sim T_0$ in PEN reactions), while D was reduced by attaching a hydrodynamic drag to the template strand (see below). In all these cases, the measured velocities followed Luther's scaling $v = a\sqrt{kD}$ with $a = 2.6$, both for k and D.
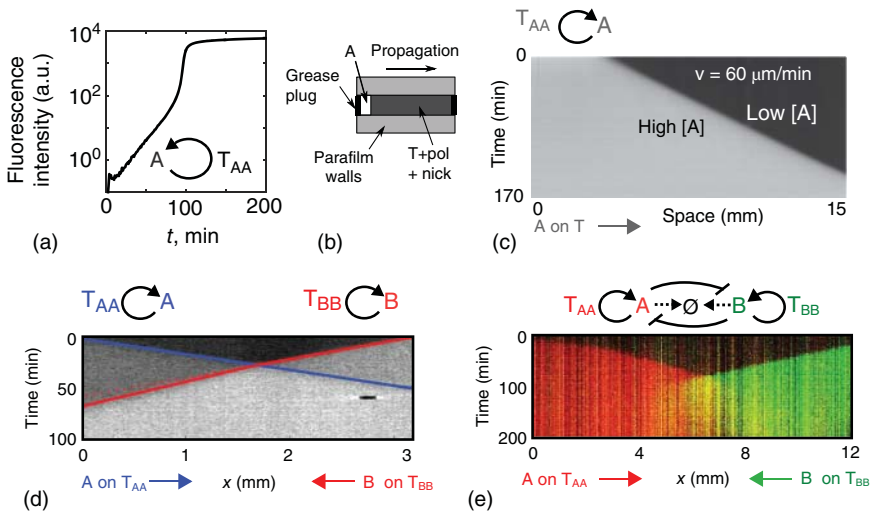
Figure 10.5 PEN autocatalyzers generate programmable concentration fronts that travel at constant velocity. (a) Temporal dynamics of a simple PEN autocatalyzer. The log-lin plot shows DNA intercalator fluorescence proportional to the concentration of A; the initial linear slope indicates exponential growth. (b) Sketch of the experimental setup to observe fronts. (c–e) Kymographs of the fluorescence signal for different autocatalytic networks show a single front (c) and two counter-propagating fronts that either do not interact (d) or that strongly inhibit each other (e). Source: Panels (b,d) From Zadorin et al. [43]. ©2015. Reproduced with the permission of American Physical Society.

The programmability of this approach was illustrated by designing different networks containing autocatalysis that resulted in controlled spatiotemporal dynamics. Two autocatalysts with orthogonal sequences generated fronts that cross-propagated with little interaction (Figure 10.5d). In contrast, two autocatalysts that cross-inhibited themselves created repelling cross-propagating fronts (Figure 10.5e).

Note that traveling fronts have not yet been demonstrated with DSD networks even if autocatalytic networks exist [38, 50]. A possible reason is that it is more difficult to control the leak of DSD autocatalysts compared with PEN ones. Indeed if the autocatalytic reaction leaks, the area ahead of the front will get triggered before the front arrives and homogeneous amplification will be observed. However, considering that CHA autocatalysts may remain untriggered for 3 hours [50] and recent DSD autocatalysts are stable for tens of hours [25], DNA-only traveling fronts may soon be observed. Finally, autocatalytic fronts have also been observed in genelet networks [41], although it remains to be tested that they verify Luther's formula.

10.3.2.2 Go-Fetch Fronts

The abovementioned spatiotemporal reaction networks operate with species that freely diffuse in solution. It may also be possible to localize the reaction by grafting the catalysts on different positions of a substrate. For instance, Gines et al. used particle-bound DNA strands to spatialize the chemical reactions in a fluidic

chamber [46]. Here, the DNA templates were grafted via a biotin–streptavidin linkage onto micrometric hydrogel particles, which results in the localized production of output strands on the particles, while the degradation happens in the whole reservoir.

Particles bearing a PEN autocatalyzer generated a traveling front of constant velocity. In addition, the authors reported a "go-fetch" chemical system that computes the distance between two specific DNA sequences through two orthogonal traveling fronts (Figure 10.6a). This experimental implementation uses 4 populations of DNA-programmed particles. The sender (S_A) initiates the production of strand A on the left side of the channel using a – leaky – monostable autocatalytic loop. A first relay population (R_A), grafted with a bistable switch that amplifies A, transmits the signal across the chamber. When the signal A comes across the receiver particle (C_{AB}), placed on the right side of the channel, it activates the production of strand B. This reaction is catalyzed by an activation template T_{AB} that converts A into B. B finally propagates through the second relay population (R_B) until reaching S_A, which in turn exhibits a fluorescent signal. This system is programmed to compute the distance between S_A and C_{AB}, which correlates with the time it takes to the S_A particles to fluoresce upon receiving the B strand. Interestingly, RD enables long-distance communication (~ 1 cm), which is at least 3 orders of magnitude larger than the particle size (~ 10 μm). A limitation of this protocol is that the particles are randomly distributed in a microchamber, with a poor control on the localization of the reactants. It would be interesting in the future to implement the precise disposition of particles programmed with different sensing networks and build on these organized arrays to create tissue-like systems.

10.3.2.3 Waves and Spirals

An autocatalytic loop coupled to a delayed inhibition makes an oscillator that, in the presence of diffusion, creates chemical waves and spirals that travel at constant velocity [18]. Such patterns have been observed in the BZ oscillator in the early 1970s [59, 60] but have only recently been observed in DNA systems with the PEN PP oscillator [45].

The mechanism of the PP oscillator is simple and produces robust oscillations [30] (Figure 10.3b). Species N, the prey, grows autocatalytically on template T_{NN}. The trick is that species P, the predator, is both palindromic, i.e. it is self-complementary and contains the sequence of N (Figure 10.3a). When P and N bind together, the polymerase extends N, yielding two P, and thus P grows autocatalytically consuming the prey ($N + P \rightarrow 2P$). In addition N and P are both degraded, mimicking the natural decay of preys and predators.

When a spatial reactor was filled with the PP mixture and the initial concentrations of N and P were homogeneous, traveling concentration waves, of prey, followed by predator waves were observed [45] (Figure 10.6c). The wave velocities were in the range 80–400 μm/min and were in fair agreement with a two-variable model related with Eq. (10.6). When the initial condition of prey was inhomogeneous, spirals were observed (Figure 10.6d). This was the first report of traveling waves in a chemical reaction network built from the bottom up. It was also the first observation of PP waves in the laboratory.
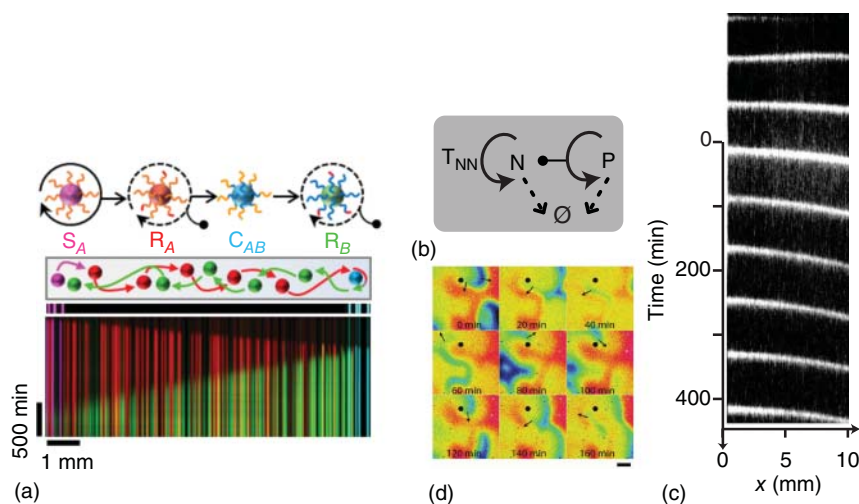
Figure 10.6 "Go-fetch" fronts (a) and waves and spirals with PEN reactions (b–d). a) A "go-fetch" program is implemented when PEN template strands are attached to microparticles distributed in 4 populations: S_A = sender particle, R_A and R_B = relay particles, C_{AB} = converter particles. When the signal A transported by R_A reaches C_{AB}, it is converted to B that travels back transported by R_B to the initial position (top). The bottom kymograph represents the go front in red and the reply front in green. Source: Panel (a) From Gines et al. [46], ©2017. Reprinted with the permission of Springer Nature, Nature Nanotechnology. (b) Topology of the PEN predator–prey (PP) network. (c) Kymograph of prey fluorescence in a one-dimensional reactor. Oblique white lines correspond to traveling waves. (d) Time-lapse fluorescent images of a prey (N) spiral turning around the black dot (false color). Source: Panels (c,d) From Padirac et al. [45]. ©2013. Reprinted with the permission of American Chemical Society.

10.3.3 Controlling Spatio-Temporal Patterns

We have seen in Section 10.1.4 that the output of an RD computation is given by the solution to the system of partial differential equations in Eq. (10.4). Such solution depends on four important features:

1) The topology of the reaction network, i.e. the function F in Eq. (10.4),
2) The reaction rates,
3) The diffusion coefficients, and
4) The initial and boundary conditions.

We have just discussed some network topologies that provide different outputs: an IFFL that makes an edge detector or an autocatalytic node that generates a traveling front, for instance. We have also seen that reaction rates can be changed in DSD reactions by changing the length of the toehold and in PEN reactions by tuning the concentration of the template strands. Here we further discuss strategies to control diffusion coefficients and initial and boundary conditions.

10.3.3.1 Controlling Diffusion Coefficients

The first strategy for reducing D is to increase the viscosity of the solution by adding a viscous solute such as glycerol or polyethylene glycol. The drawback is that this method is not specific and all DNA species will be slowed down by a

similar factor. The second strategy is to perform the reaction inside a sieving matrix, like a hydrogel, such that large DNA species will be trapped and small ones will diffuse freely. This method was successfully used in the edge detection network discussed above [39]. In this case the fast-diffusing strand A was shorter than the other reagents (8 vs. 64 nt long), and reactions were performed in 20% cross-linked polyacrylamide gel, resulting in a striking 10-fold difference in diffusion coefficient. The last strategy is to specifically change D for a given species, which can be performed by attaching a DNA strand to a hydrodynamic drag. This method was employed to modify the diffusion of PEN autocatalyzers, the drag being a triton micelle of ~ 5 nm radius to which a cholesteryl-modified template T_{AA} was attached (Figure 10.7a). Autocatalyzer A needed to bind to T_{AA} to grow, and, depending on the molar fraction of cholesteryl template, D was reduced up to 2.7-fold [43]. Importantly, not only the effective diffusion of a passive solution was controlled but also the one of A involved in an RD front, which was demonstrated by measuring the velocity of the front.

An improved hydrodynamic drag was demonstrated by Rodjanapanyakul et al. by using a linear copolymer of polyacrylamide and strand T [61], reaching a reduction of D of fivefold in DSD reactions, an approach also modeled by Allen et al. [62]. With this strategy the effective diffusion of freely diffusing strand A, complementary to T, could be modulated by changing the concentration of a competitor strand C that also bound to T (Figure 10.7a,b) [61]. A similar strategy was used to control the electrophoretic mobility of DNA involved in DSD programs [63]. Recently, the copolymerization of DNA with polyacrylamide was used to stabilize PEN static RD patterns for more than 60 hours [48].

10.3.3.2 Initial and Boundary Conditions

In the aforementioned examples, the initial condition, or input signal, was introduced either by adding a droplet of solution containing the input species, which results in poor spatial resolution, or by using a light pattern, with high spatial resolution. Other ways of controlling the initial condition include microfluidic injection using PDMS monolithic pumps [44] and electric switches [64]. In this last example, Kurylo et al. grafted an oligonucleotide on gold electrodes embedded in microfluidic channels. They took advantage of the electrochemical properties of the thiol–gold bond to release specific DNA localized in space and time through a voltage pulse. In particular, the DNA release was able to trigger PEN autocatalytic fronts. Without involving delicate liquid handling, this method offers the possibility to control and interact in real time with running DNA-based molecular systems.

The simplest way to control boundary conditions in an RD process is to change the geometry of the reactor where the process takes place. This can be performed with standard microfabrication methods, although one has to take care to choose a material that limits evaporation, which is particularly important for PEN reactions that occur in the temperature range 37–45 °C. To this end, NOA microfluidic devices [65] were used to investigate the effect of curvature in the propagation dynamics of PEN autocatalyzers and to demonstrate that DNA-based pulses can compute the optimal path within a maze [44].

(a)

(b)

(c)
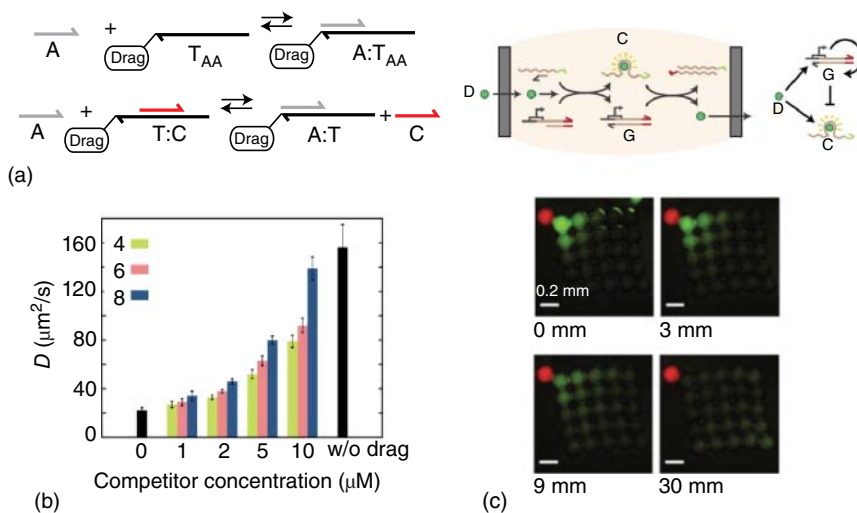
Figure 10.7 Strategies to control the diffusion and geometry of RD patterns. (a) Two methods to tune the diffusion of strand A proposed in [43] (top) and [61] (bottom). Source: (a) Modified from Zadorin et al. [43] (b) Diffusion coefficient of species A measured for the competitor strategy in panel (a), bottom, for different competitor concentrations. The colors refer to different lengths of the toehold where A binds on T: C. Source: (b) From Rodjanapanyakul et al. [61]. ©2018. Reproduced figure with permission of American Physical Society. (c) An incoherent feed forward loop genelet network (top) processes information from DFHBI green fluorophore, D, present in the red droplet (bottom) across a square array of droplets bearing protein nanopores and generates a diffusion pulse (from the red droplet on the top left corner to the bottom right corner droplet). Source: (c) From Dupin and Simmel 2019 [42]. ©2019. Reproduced figure with permission of Springer Nature, Nature Chemistry.

An interesting alternative for controlling the geometry of RD patterns was proposed by Dupin and Simmel [42] (Figure 10.7d). They distributed a genelet reaction network inside a two-dimensional array of microdroplets that were interconnected by protein nanopores. These nanopores allowed the droplets to exchange DFHBI, a small molecule that fluoresces when bound to the spinach RNA aptamer, but hindered the DNA program from leaving the DNA-impermeable droplets. By constructing an IFFL reaction network that takes DFHBI as an input and produces the spinach aptamer as output, a transient pulse of fluorescence across an array of droplets was observed in the presence of a seed droplet with high concentration of DFHBI, which was reported in both one- and two-dimensions. With this clever idea, the authors obtained compartments that displayed a certain degree of autonomy and at the same time the capability to exchange and process specific chemical information, mimicking cell–cell communication in living tissues. For the moment, the messaging molecule, DFHBI, cannot be amplified autocatalytically, and thus the propagation velocity v of the fluorescent pulse was not constant, as in Luther's formula, but is rather expected to follow a diffusive scaling $v \sim 1/\sqrt{t}$. We thus anticipate that developing strategies to transfer DNA-rich information across membrane droplets will be an important question in the near future.

10.4 Steady-State Spatial Patterns

In the above we have discussed RD computations with a time-dependent output. However, in many instances, a time-independent output is desirable. This is very challenging because diffusion will continuously dilute the output so that reaction needs to balance diffusion to create a steady-state pattern. Steady-state RD patterns are observed during early embryo development, where a robust output is needed. In addition, they can be useful to create synthetic morphogenetic materials with a chemically defined final shape.

10.4.1 Colony Formation

Intercellular communication is a critical parameter to precisely orchestrate tissue development and differentiation. In an attempt to emulate very primitively these communication channels between different objects, Gines et al. used DNA-programmed particles (cf. Section 10.3.2.2) that exhibit synergism and cooperativity (Figure 10.8). The system uses two types of particles, P_{AB} and P_{BA}, that produce the output strand B from A, and conversely, using PEN reactions. The particles are dispersed in a solution containing the PEN exonuclease that degrades A and B, such that, when the particles are far from each other, homogeneous degradation is stronger than local particle production and no strand production is detected. However, when both particle types are present in close proximity, their cross-activation outcompetes degradation, and both A and B are produced autocatalytically. The system evolves toward a steady state where sharp concentration profiles are drawn around clusters of synergic particles, creating colony-like patterns in a large population of thousands of particles, which last for over dozens of hours. Interestingly, these colonies became more active as their density increased, suggesting long-distance interactions between
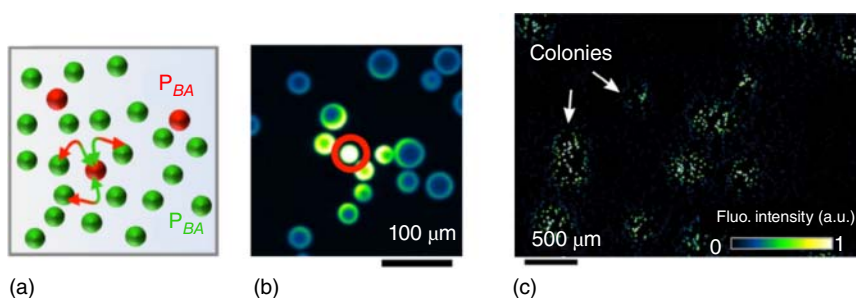
Figure 10.8 Steady-state "colony" formation in a population of synergic particles functionalized with DNA templates. (a) Red particles P_{BA} produce strand A in the presence of B, and green particles P_{AB} produce B in the presence of A through PEN reactions. (b) When a red particle (circled in red) is present in close proximity to green particles, the emergent autocatalytic network outcompetes homogeneous degradation and forms a colony of high concentration of A and B (high fluorescence) at steady state. (c) Large-scale view of a small population of red particles dispersed among a large population of green particles. Colonies appear as clusters of green dots. Source: From Gines et al. [46]. ©2017. Reprinted with the permission of Springer Nature, Nature Nanotechnology.

the colonies that reinforce each other, in addition to the local synergic particle activation mechanism. Particle-based systems coupled to RD thus enable to easily control the topography of the reactions to start building large networks that display steady-state patterns. Although the particles used in this study were too big to remain suspended, Brownian particles would enable the exploration of 3D pattern formation.

10.4.2 Patterns with Positional Information

The development of a living embryo is a fascinating process of spatial computation. Its early phase is called pattern formation, and it is characterized by the generation of spatially defined chemical concentration patterns. The principles underlying embryo pattern formation are still under debate [66], and they are largely dependent on the organism, the region, and the developmental stage [67]. Nevertheless, two major conceptual frameworks have dominated our understanding of this process for the last 50 years [68]: Turing instability and positional information. The first one, developed by Turing [16, 69], demonstrated that RD processes can form patterns. The second, introduced by Wolpert [70], proposed a simple way to explain how cells may compute their position within the embryo in the presence of a concentration gradient. The principal difference between the two is that the Turing instability is a symmetry-breaking mechanism that generates a heterogeneous concentration state from a homogeneous one, while positional information is a sharpening mechanism that amplifies a concentration heterogeneity from an initial state where the symmetry has already been broken. While patterns of positional information have recently been engineered with DNA networks [40, 47], DNA Turing patterns have yet to be demonstrated [71].

The idea of positional information was introduced by Wolpert to explain how an embryo with a single break of symmetry could be further split into several distinct regions [70]. He coined the term "French flag problem" to illustrate the challenge – fundamental because it is pervasive in the development of virtually all complex organisms – of creating three distinct regions of space with sharp borders from an amorphous mass and a shallow concentration gradient. The idea of Wolpert is simple. Let's consider a model embryo formed by a one-dimensional array of cells along the x axis submitted to an initial, monotonously decreasing, gradient of morphogen $M(x)$ (Figure 10.9). We suppose, for instance, that the cells on the left will become the head, those on the central part the thorax, and those on the right the abdomen. In this situation, to know its position, each cell "just" needs to read out the concentration of the morphogen along the gradient. Cells reading a concentration above a certain threshold $M(x) > M_1$ will express the blue protein, those below a second threshold $M(x) < M_2$ will express the red protein, and those in between will express the white protein (Figure 10.9). We say that the gradient provides *positional information* to the cells. Patterns generated by positional information were first observed in the *Drosophila* fly embryo [72–74]. They were engineered for the first time *in vitro* in a transcription–translation system by Isalan et al. [13]. In the following we will discuss recent DSD and PEN implementations of positional information patterns.
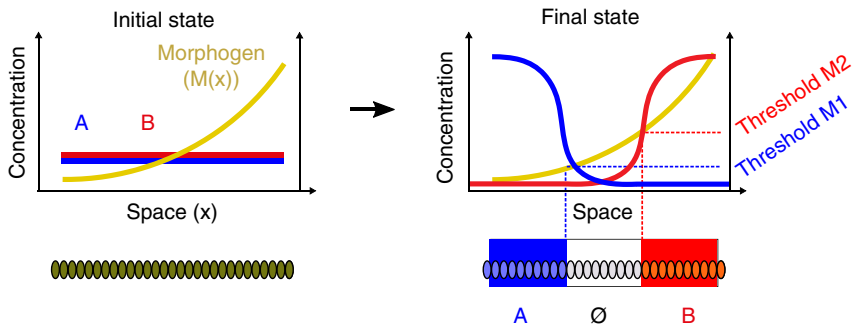
Figure 10.9 Illustration of Wolpert's concept of positional information as a solution of the "French flag problem" in the presence of a shallow gradient of morphogen concentration in a one-dimensional embryo.

Zenk et al. experimentally engineered static positional information patterns using DSD reactions [40]. In particular, they obtained linear and hill-shaped concentration patterns at steady state in an open gel reactor (Figure 10.10). To do so, a reaction network was designed to maintain the concentration of output strand O at steady state by a combination of rapid creation and slow degradation as follows:

$$S + I \xrightarrow{k_1} O + W_1 \tag{10.7}$$

$$Rec + O \xrightarrow{k_2} W_2 + W_3 \tag{10.8}$$

where S and O are ssDNA and I and Rec are DSD gates with fast and slow toeholds, respectively (i.e. $k_1 \gg k_2$). The RD pattern evolved in an open reactor composed of an agarose gel pad, about 1-cm long, connected on each side to a liquid reservoir in a linear geometry: reservoir/gel/reservoir. When the initial concentrations of both O and Rec where homogeneous in the three zones of the reactor
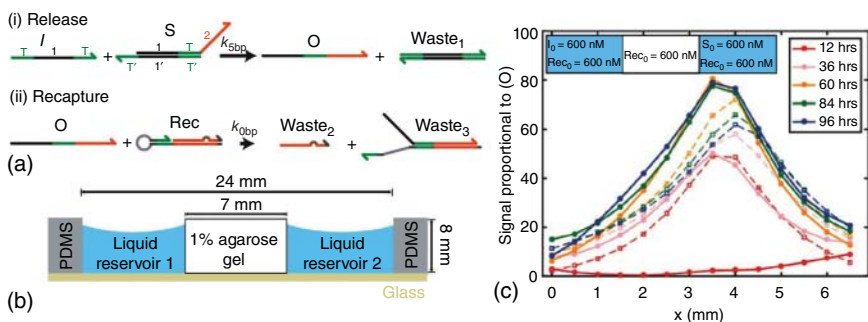
Figure 10.10 Static pattern of positional information engineered with DNA strand displacement circuits. (a) Reaction mechanism of the DSD network used to maintain a steady-state concentration of strand O. (b) Open gel reactor with liquid reservoirs on each side. The pattern is observed inside the gel pad. (c) Dynamics of the hill-shaped pattern of species O obtained inside the gel reactor when the left reservoir contained species I, the right reservoir contained species S, and Rec was distributed homogeneously. Source: From Zenk et al. 2017 [40]. ©2017. Reproduced with the permission of Royal Society of Chemistry.

but I and S were only present, respectively, in the left and right reservoirs, stable linear gradients of I and S appeared in the agarose pad, together with a stable hill-shaped profile of O. These patterns took about 30 hours to form and lasted for about 70 hours. Importantly, they could also be observed in 2D. Note that, due to the lack of a chemical sink, the liquid reservoirs needed to be replenished every 24 hours to keep the system out of equilibrium. Finally, the authors demonstrated the programmability of their approach by engineering a second network, orthogonal with the initial one, that generated a second hill-shaped pattern. A related strategy was recently demonstrated by Chen and Seelig, obtaining more complex band patterns [75]. However, in this last case, the patterns were generated in a closed reactor, and thus they were not static.

A complementary approach to engineer static positional information patterns was demonstrated by Zadorin et al. using PEN reactions [47]. In particular, they obtained immobile RD fronts that could be assembled into a French flag pattern that produced three bands with distinct compositions. Their implementation is based on spatial bistability [76]: a bistable reaction network in the presence of a gradient of a species acting as a bifurcation parameter generates an immobile RD front.

PEN reactions may be assembled in a bistable network either by using two autocatalysts that cross-inhibit each other [31] (see Figures 10.3a and 10.5e) or one autocatalyst that is repressed by a saturable pathway [32]. In the last case the network is composed of the autocatalytic production of A by its template T_{AA} (as in Figure 10.5a), the degradation of A by the exonuclease and its repression by R_A that converts it into waste (see Figures 10.3a and 10.11a). This system is bistable because at low A, autocatalysis is slower than repression but at high A (typically $A > R_A$), the repression reaction saturates and shuts off. As a result the concentration of R_A acts as a bifurcation parameter.

Because PEN networks can be kept out of equilibrium for tens of hours in a closed reactor, implementing positional information is relatively straightforward: one just needs to fill a channel with a bistable network and generate a gradient of R_A along the channel (Figure 10.11b). The authors used a 5 cm long glass capillary where a gradient of morphogen R_A was generated using Taylor dispersion [48, 77]. This way, the gradient was set up within minutes but remained stable for tens of hours, because diffusion over centimeter scales is very slow. With such an experimental setting, several multistable PEN networks were tested with different repressor gradients, and a variety of steady-state band patterns were observed: an immobile front, two immobile fronts that repel each other, and two immobile fronts that form a French flag pattern (Figure 10.11c). All the patterns self-organized in the same way: first a reaction-only phase generated a mobile front on the side of the channel where activation was stronger than repression, and then the front(s) traveled through an RD mechanism toward the repression side, slowing down until an RD steady-state was obtained. In addition, the borders of the bands were one order of magnitude sharper than the initial gradients (typically 1 mm compared with 1 cm). However, PEN autocatalytic parasites (see Section 10.2.2.2) broke down the steady state pattern after 10 hours. This problem has recently been solved by constructing parasite-robust PEN networks [36] that are able to generate steady-state patterns that last for several days [48].
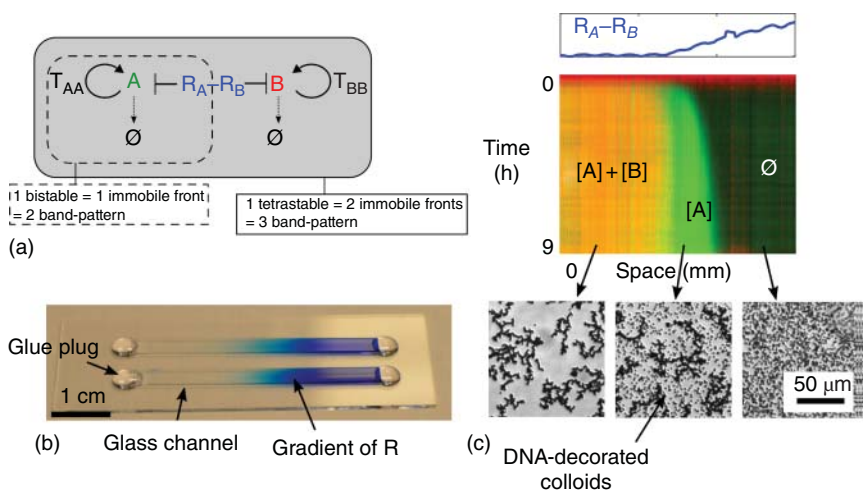
Figure 10.11 Static patterns of positional information can be engineered with PEN reactors and coupled to a simple material. (a) Tetrastable network with autocatalytic nodes A, B, and repressor R_A–R_B used to create a three-band French flag pattern. Enclosed in dashed lines the bistable network that creates a two-band pattern. (b) Photograph of the closed reactor where the patterns were obtained: a glass channel with a stable gradient of R_A–R_B (in blue). (c) A tetrastable network in panel a forms a French flag pattern with three zones of different composition in the presence of a gradient of the corresponding repressor. The top image represents the stable underlying gradient concentration profile (in blue) and a kymograph of DNA fluorescence inside the channel. When the channel is initially filled with a homogeneous dispersion of DNA-decorated colloids, the reaction network can be designed to specifically control the aggregation of the beads (bottom images). Source: Adapted from Zadorin et al. [47] ©2017. Reprinted with the permission of Springer Nature, Nature Chemistry.

Pattern formation in the embryo is used for spatially controlling subsequent developmental steps, such as cell differentiation. If we consider the embryo as a complex material and development as a self-fabrication process, the pattern would be the self-fabricated blueprint. Zadorin et al. emulated this idea in a simple artificial system by coupling their PEN French flag pattern with the conditional aggregation of DNA-decorated colloids [47]. As a result, the DNA RD patterns just evoked differentiated an initially homogeneous material – a suspension of 1 μm colloids – into different zones with different microscopic structures (Figure 10.11c). Recently, Urtel et al. were able to maintain these band patterns at steady state inside an autonomous hydrogel material [48], opening the way to building self-patterning autonomous materials.

10.5 Conclusion and Perspectives

We have seen in this chapter that DNA circuits are well suited to perform basic spatial computations. We have focused on computations that use reaction-diffusion primitives, which are the most natural operations that molecules perform in solution and, in particular, in illustrating their experimental implementations. The majority of DNA spatial computations involve

analog circuits, and we have briefly discussed three of them: DSD networks, genelets, and PEN DNA reactions, all capable of generating oscillations in a closed reactor. DSD circuits have the advantage of being fully programmable because there are tools to predict the thermodynamics and kinetics of DNA hybridization reactions [78–80]. However, engineering "chemical supplies" that maintain DSD networks out of equilibrium in a closed reactor is difficult. In this regard, DNA/enzyme networks, such as genelets and PEN reactions, are complementary to DSD: one needs to integrate the non-programmability of enzymes in the design process, but, in exchange, maintaining the system out of equilibrium in a closed reactor is greatly facilitated. Although both DSD and DNA/enzyme approaches are both expected to be biocompatible (they run in aqueous buffer at pH 7 and 37 $^\circ$ C), the absence of enzymes in DSD networks may make them more widely compatible with biological systems containing biomolecules or living cells.

Autocatalytic nodes are a basic element of RD pattern formation because the exponential growth from autocatalysis balances the dilution arising from diffusion, creating traveling patterns. Autocatalysis can be easily implemented in PEN circuits, and this has been used to demonstrate a series of nontrivial traveling patterns that are constructed around the principle of a traveling front [43–45, 47, 48]. Autocatalysis has been demonstrated in DSD circuits, but not for generating patterns, probably due to undesired leak reactions. However, recent strategies have succeeded in dramatically reducing leak in DSD reactions [25, 81], and thus autocatalytic DSD patterns may soon be observed. Instead, DSD patterns have explored other important network elements, such as IFFL [39] and steady-state generators [40]. In addition, the design of more complex DSD patterns [82] and of RD cellular automata [83] has been illustrated through simulations, and we may soon see these realizations in experiments.

Now that the engineering of RD patterns with DNA programs has been thoroughly demonstrated, we see three interesting directions for future work. Firstly, to push forward the complexity of the engineered patterns. In this regard, an important objective is the engineering of Turing patterns, for which diffusion control strategies [43, 61] evoked above are essential, and multiphasic approaches [42] may also be advantageous. Secondly, the investigation of the fundamental mechanisms of molecular self-organization with DNA patterning systems. In particular, engineered patterns of positional information could help to ask questions about how developing embryos form patterns [68]. To succeed, this challenging approach will need a strong collaboration between DNA molecular programmers and developmental biologists. Finally, coupled to responsive DNA materials [34, 84–86], DNA patterning systems could create a new generation of life-like materials capable of self-construction, communication, and healing.

Acknowledgments

References

1 DeHon, A., Giavitto, J.-L., and Gruau, F. (2007). 06361 executive report – computing media languages for space-oriented computation. In: *Computing Media and Languages for Space-Oriented Computation*, Number 06361 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany (ed. A. DeHon, J.-L. Giavitto, and F. Gruau). Schloss Dagstuhl: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI).

2 Cavagna, A., Cimarelli, A., Giardina, I. et al. (2010). Scale-free correlations in starling flocks. *Proc. Natl. Acad. Sci. U.S.A.* 107 (26): 11865–11870.

3 Sarpeshkar, R. (1998). Analog versus digital: extrapolating from electronics to neurobiology. *Neural Comput.* 10 (7): 1601–1638.

4 Seelig, G., Soloveichik, D., Zhang, D.Y., and Winfree, E. (2006). Enzyme-free nucleic acid logic circuits. *Science* 314 (5805): 1585–1588.

5 Qian, L. and Winfree, E. (2011). Scaling up digital circuit computation with DNA strand displacement cascades. *Science* 332 (6034): 1196–1201.

6 Soloveichik, D., Seelig, G., and Winfree, E. (2010). DNA as a universal substrate for chemical kinetics. *Proc. Natl. Acad. Sci. U.S.A.* 107 (12): 5393–5398.

7 Kondo, S. and Miura, T. (2010). Reaction-diffusion model as a framework for understanding biological pattern formation. *Science* 329 (5999): 1616–1620.

8 Xing, Y., Liu, B., Chao, J., and Wang, L. (2017). DNA-based nanoscale walking devices and their applications. *RSC Adv.* 7 (75): 47425–47434.

9 Hong, F., Zhang, F., Liu, Y., and Yan, H. (2017). DNA origami: scaffolds for creating higher order structures. *Chem. Rev.* 117 (20): 12584–12640.

10 Scalise, D. and Schulman, R. (2019). Controlling matter at the molecular scale with DNA circuits. *Annu. Rev. Biomed. Eng.* 21 (1): 469–493.

11 Loose, M., Fischer-Friedrich, E., Ries, J. et al. (2008). Spatial regulators for bacterial cell division self-organize into surface waves in vitro. *Science* 320 (5877): 789–792.

12 Zieske, K., Schwille, P., and Balasubramanian, M. (2014). Reconstitution of self-organizing protein gradients as spatial cues in cell-free systems. *eLife* 3: e03949.

13 Isalan, M., Lemerle, C., and Serrano, L. (2005). Engineering gene networks to emulate *Drosophila* embryonic pattern formation. *PLoS Biol.* 3 (3): 488–496.

14 Tayar, A.M., Karzbrun, E., Noireaux, V., and Bar-Ziv, R.H. (2015). Propagating gene expression fronts in a one-dimensional coupled system of artificial cells. *Nat. Phys.* 11: 1037–1041.

15 Wang, S.S. and Ellington, A.D. (2019). Pattern generation with nucleic acid chemical reaction networks. *Chem. Rev.* 119 (10): 6370–6383.

16 Turing, A.M. (1952). The chemical basis of morphogenesis. *Philos. Trans. R. Soc. London, Ser. B* 237 (641): 37–72.

17 Murray, J.D. (2003). *Mathematical Biology II: Spatial Models and Biomedical Applications*. New York: Springer.

18 Epstein, I. and Pojman, J.A. (1998). *An Introduction to Nonlinear Chemical Reactions*. New York: Oxford University Press.

19 Epstein, I.R. and Showalter, K. (1996). Nonlinear chemical dynamics: oscillations, patterns, and chaos. *J. Phys. Chem.* 100 (31): 13132–13147.

20 Yurke, B., Turberfield, A.J., Mills, A.P. et al. (2000). A DNA-fuelled molecular machine made of DNA. *Nature* 406 (6796): 605–608.

21 Turberfield, A.J., Mitchell, J.C., Yurke, B. et al. (2003). DNA fuel for free-running nanomachines. *Phys. Rev. Lett.* 90 (11): 118102.

22 Zhang, D.Y. and Winfree, E. (2009). Control of DNA strand displacement kinetics using toehold exchange. *J. Am. Chem. Soc.* 131 (47): 17303–17314.

23 Genot, A.J., Zhang, D.Y., Bath, J., and Turberfield, A.J. (2011). Remote toehold: a mechanism for flexible control of DNA hybridization kinetics. *J. Am. Chem. Soc.* 133 (7): 2177–2182.

24 Zhang, D.Y. and Seelig, G. (2011). Dynamic DNA nanotechnology using strand-displacement reactions. *Nat. Chem.* 3 (2): 103–113.

25 Srinivas, N., Parkin, J., Seelig, G. et al. (2017). Enzyme-free nucleic acid dynamical systems. *Science* 358 (6369): eaal2052.

26 Srinivas, N., Parkin, J., Seelig, G. et al. (2017). Enzyme-free nucleic acid dynamical systems. *bioRxiv*. https://doi.org/10.1101/138420.

27 Kim, J., White, K.S., and Winfree, E. (2006). Construction of an in vitro bistable circuit from synthetic transcriptional switches. *Mol. Syst. Biol.* 2: 68.

28 Kim, J. and Winfree, E. (2011). Synthetic in vitro transcriptional oscillators. *Mol. Syst. Biol.* 7: 465.

29 Montagne, K., Plasson, R., Sakai, Y. et al. (2011). Programming an in vitro DNA oscillator using a molecular networking strategy. *Mol. Syst. Biol.* 7: 466.

30 Fujii, T. and Rondelez, Y. (2013). Predator-prey molecular ecosystems. *ACS Nano* 7 (1): 27–34.

31 Padirac, A., Fujii, T., and Rondelez, Y. Bottom-up construction of in vitro switchable memories. (2012). *Proc. Natl. Acad. Sci. U.S.A.* https://doi.org/10.1073/pnas.1212069109.

32 Montagne, K., Gines, G., Fujii, T., and Rondelez, Y. (2016). Boosting functionality of synthetic DNA circuits with tailored deactivation. *Nat. Commun.* 7: 13474.

33 Schaffter, S.W. and Schulman, R. (2019). Building in vitro transcriptional regulatory networks by successively integrating multiple functional circuit modules. *Nat. Chem.* 11 (9): 829–838.

34 Franco, E., Friedrichs, E., Kim, J. et al. (2011). Timing molecular motion and production with a synthetic transcriptional clock. *Proc. Natl. Acad. Sci. U.S.A.* 108: E784–E793.

35 Zyrina, N.V., Antipova, V.N., and Zheleznaya, L.A. (2014). Ab initio synthesis by DNA polymerases. *FEMS Microbiol. Lett.* 351 (1): 1.

36 Urtel, G., Van Der Hofstadt, M., Galas, J.-C., and Estevez-Torres, A. (2019). rEXPAR: an isothermal amplification scheme that is robust to autocatalytic parasites. *Biochemistry* 58 (23): 2675–2681.

37 Genot, A.J., Baccouche, A., Sieskind, R. et al. (2016). High-resolution mapping of bifurcations in nonlinear biochemical circuits. *Nat. Chem.* 8: 760–767.

38 Zhang, D.Y., Turberfield, A.J., Yurke, B., and Winfree, E. (2007). Engineering entropy-driven reactions and networks catalyzed by DNA. *Science* 318 (5853): 1121–1125.

39 Chirieleison, S.M., Allen, P.B., Simpson, Z.B. et al. (2013). Pattern transformation with DNA circuits. *Nat. Chem.* 5 (12): 1000–1005.

40 Zenk, J., Scalise, D., Wang, K. et al. (2017). Stable DNA-based reaction-diffusion patterns. *RSC Adv.* 7: 18032–18040.

41 Simmel, F.C. and Schulman, R. (2017). Self-organizing materials built with DNA. *MRS Bull.* 42 (12): 913–919.

42 Dupin, A. and Simmel, F.C. (2019). Signalling and differentiation in emulsion-based multi-compartmentalized in vitro gene circuits. *Nat. Chem.* 11 (1): 32–39.

43 Zadorin, A.S., Rondelez, Y., Galas, J.-C., and Estevez-Torres, A. (2015). Synthesis of programmable reaction-diffusion fronts using DNA catalyzers. *Phys. Rev. Lett.* 114 (6): 068301.

44 Zambrano, A., Zadorin, A.S., Rondelez, Y. et al. (2015). Pursuit-and-evasion reaction-diffusion waves in microreactors with tailored geometry. *J. Phys. Chem. B* 119 (17): 5349–5355.

45 Padirac, A., Fujii, T., Estévez-Torres, A., and Rondelez, Y. (2013). Spatial waves in synthetic biochemical networks. *J. Am. Chem. Soc.* 135 (39): 14586–14592.

46 Gines, G., Zadorin, A.S., Galas, J.-C. et al. (2017). Microscopic agents programmed by DNA circuits. *Nat. Nano* 12: 351–359.

47 Zadorin, A.S., Rondelez, Y., Gines, G. et al. (2017). Synthesis and materialization of a reaction-diffusion french flag pattern. *Nat. Chem.* 9: 990.

48 Urtel, G., Estevez-Torres, A., and Galas, J.-C. (2019). DNA-based long-lived reaction-diffusion patterning in a host hydrogel. *Soft Matter* 15 (45): 9343–9351.

49 Kuhnert, L., Agladze, K.I., and Krinsky, V.I. (1989). Image-processing using light-sensitive chemical waves. *Nature* 337 (6204): 244–247.

50 Yin, P., Choi, H.M.T., Calvert, C.R., and Pierce, N.A. (2008). Programming biomolecular self-assembly pathways. *Nature* 451 (7176): 318–322.

51 Abe, K., Kawamata, I., Nomura, S.-i.M., and Murata, S. (2019). Programmable reactions and diffusion using DNA for pattern formation in hydrogel medium. *Mol. Syst. Des. Eng.* 4 (3): 639–643.

52 Luther, R. (1906). Space propagation of chemical reactions. *Z. Elektrochem. Angew. Phys. Chem.* 12: 596–600.

53 Luther, R. (1987). Propagation of chemical reactions in space. *J. Chem. Educ.* 64 (9): 740.

54 Fisher, R.A. (1937). The wave of advance of advantageous genes. *Ann. Eugen.* 7: 355–369.

55 Kolmogoroff, A., Petrovsky, I., and Piscounoff, N. (1937). Etude de l'equation de la diffusion avec croissance de la quantité de matiére et son application à un probleme biologique. *Bull. Univ. Moskou, Ser. Int., Sec. A* 6: 1–25.

56 Volpert, V. and Petrovskii, S. (2009). Reaction–diffusion waves in biology. *Phys. Life Rev.* 6 (4): 267–310.

57 van Saarloos, W. (2003). Front propagation into unstable states. *Phys. Rep.* 386: 29–222.

58 Roques, L. (2013). *Modèles de réaction-diffusion pour l'écologie spatiale.* Versailles: Editions Quae.

59 Zaikin, A.N. and Zhabotinsky, A.M. (1970). Concentration wave propagation in two-dimensional liquid-phase self-oscillating system. *Nature* 225 (5232): 535–537.

60 Winfree, A.T. (1972). Spiral waves of chemical activity. *Science* 175 (4022): 634–636.

61 Rodjanapanyakul, T., Takabatake, F., Abe, K. et al. (2018). Diffusion modulation of DNA by toehold exchange. *Phys. Rev. E* 97 (5): 052617.

62 Allen, P.B., Chen, X., Simpson, Z.B., and Ellington, A.D. (2014). Modeling scalable pattern generation in DNA reaction networks. *Nat. Comput.* 13 (4): 583–595.

63 Allen, P., Chen, X., and Ellington, A. (2012). Spatial control of DNA reaction networks by DNA sequence. *Molecules* 17 (11): 13390–13402.

64 Kurylo, I, Gines, G., Rondelez, Y. et al. (2018). Spatiotemporal control of DNA-based chemical reaction network via electrochemical activation in microfluidics. *Sci. Rep.* 8 (1): 6396.

65 Bartolo, D., Degré, G., Nghe, P., and Studer, V. (2008). Microfluidic stickers. *Lab. Chip* 8: 274–279.

66 Jaeger, J., Irons, D., and Monk, N. (2008). Regulative feedback in pattern formation: towards a general relativistic theory of positional information. *Development* 135 (19): 3175–3183.

67 Wolpert, C. and Tickle, L. (2011). *Principles of Development.* Oxford: Oxford University Press.

68 Green, J.B. and Sharpe, J. (2015). Positional information and reaction-diffusion: two big ideas in developmental biology combine. *Development* 142 (7): 1203–1211.

69 Turing, A. (1990). The chemical basis of morphogenesis. *Bull. Math. Biol.* 52 (1): 153–197.

70 Wolpert, L. (1969). Positional information and the spatial pattern of cellular differentiation. *J. Theor. Biol.* 25 (1): 1–47.

71 Zambrano, A. (2016). Synthesis of reaction-diffusion patterns with DNA: towards turing patterns. PhD thesis. Université Paris-Saclay.

72 Driever, W. and Nüsslein-Volhard, C. (1988). A gradient of bicoid protein in *Drosophila* embryos. *Cell* 54 (1): 83–93.

73 Driever, W. and Nüsslein-Volhard, C. (1988). The bicoid protein determines position in the *Drosophila* embryo in a concentration-dependent manner. *Cell* 54: 95–104.

74 Johnston, D.St. and Nüsslein-Volhard, C. (1992). The origin of pattern and polarity in the *Drosophila* embryo. *Cell* 68 (2): 201–219.

75 Chen, S. and Seelig, G. (2020). Programmable patterns in a DNA-based reaction-diffusion system. *Soft Matter.* 16 (14): 3555–3563.

76 Rulands, S., Klünder, B., and Frey, E. (2013). Stability of localized wave fronts in bistable systems. *Phys. Rev. Lett.* 110 (3): 038102.

77 Taylor, G.I. (1953). Dispersion of soluble matter in solvent flowing slowly through a tube. *Proc. R. Soc. London, Ser. A* 219 186–203

78 Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* 31 (13): 3406–3415.

79 Zadeh, J.N., Steenberg, C.D., Bois, J.S. et al. (2011). NUPACK: analysis and design of nucleic acid systems. *J. Comput. Chem.* 32 (1): 170–173.

80 Lorenz, R., Bernhart, S.H., Zu Siederdissen, C.H. et al. (2011). Viennarna package 2.0. *Algorithm. Mol. Biol.* 6 (1): 1–14.

81 Wang, B., Thachuk, C., Ellington, A.D. et al. (2018). Effective design principles for leakless strand displacement systems. *Proc. Natl. Acad. Sci. U.S.A.* 115 (52): E12182–E12191.

82 Scalise, D. and Schulman, R. (2014). Emulating cellular automata in chemical reaction-diffusion networks. In: *DNA Computing and Molecular Programming* (ed. M. Satoshi and K. Satoshi), 67–83. Cham: Springer International Publishing.

83 Scalise, D. and Schulman, R. (2014). Designing modular reaction-diffusion programs for complex pattern formation. *Technology* 02 (01): 55–66.

84 Rothemund, P.W.K. (2006). Folding DNA to create nanoscale shapes and patterns. *Nature* 440 (7082): 297–302.

85 Jones, M.R., Seeman, N.C., and Mirkin, C.A. (2015). Programmable materials and the nature of the DNA bond. *Science* 347 (6224): 1260901-1, 1260901-1 1.

86 Cangialosi, A., Yoon, C.K., Liu, J. et al. (2017). DNA sequence–directed shape change of photopatterned hydrogels via high-degree swelling. *Science* 357 (6356): 1126–1130.

11

Computing Without Computing: DNA Version

Vladik Kreinovich[1] and Julio C. Urenda[2]

[1] *The University of Texas at El Paso, Department of Computer Science, 500 W University Ave., El Paso, TX 79968, USA*
[2] *The University of Texas at El Paso, Department of Mathematical Sciences and Department of Computer Science, 500 W University Ave., El Paso, TX 79968, USA*

11.1 Introduction

In his famous 1994 paper [1], Leonard Adleman showed that, in principle, we can drastically speed up computations if we use the fact that DNA fragments combine together – in a process known as *ligation* – if the corresponding nucleotides match, i.e. if:

- A is matched with T,
- T is matched with A,
- C is matched with G, and
- G is matched with C.

For example, fragments ACTTG and TGAAC match perfectly.

Specifically, this paper showed that we can speed up the solution to the following *Hamiltonian path problem*:

- given a graph,
- find a path in this graph that visits every vertex exactly once.

This seminal paper started the field of *DNA computing*, which now includes both:

- using actual DNA fragments (as Adleman did) and
- using computer simulation of the corresponding processes.

One of the main advantages of computing via molecular interactions, when each molecule serves as a processor, is that in each mole, we have 10^{23} molecules – and thus, 10^{23} processors working in parallel. Such unbelievable parallelism – many orders of magnitude higher than the usual thousands of processors in a supercomputer – is a clear indication that this approach has a great potential.

Later, similar DNA-based algorithms were proposed for solving other complex problems, such as propositional satisfiability (this problem is explained, in detail, later in this chapter). For reasonably recent overviews, see, e.g. [2–6].

All these algorithms are based on actually using (or simulating) the ligation process. This is similar to how quantum computers use quantum activities to perform computations. Interestingly, in quantum computing, there is another phenomenon known as *computing without computing*, when, somewhat surprisingly, the result of the computation appears without actually invoking quantum processes. In this chapter, we show that similar phenomenon is possible for DNA computing:

- in addition to the more traditional way of using or simulating DNA *activity*,
- we can also use DNA *inactivity* to solve complex problems.

11.2   Computing Without Computing – Quantum Version: A Brief Reminder

DNA computing is one of several directions in the general quest for using novel physical phenomena in computing. Another – probably even more well known – direction is *quantum computing*, the use of quantum effects to speed up computations; see, e.g. [7, 8].

Most quantum algorithms actually use quantum effects to perform the corresponding computations, but there is an interesting version called *counterfactual quantum computing*, or, alternatively, *computing without computing*. The idea is that:

- we *set up* the corresponding quantum computations, but
- we *do not* actually *run* them,

and still, because of the quantum effects, we get the desired result with some probability.

This idea was first proposed in [9]. The main motivation behind this idea was not so much about *computing* but rather about *testing*: the same idea can be, in principle, used to test the complex equipment without actually running it. For example, in principle, we can test whether the atomic bomb (that has been in storage for a long time) will actually explode when triggered – without actually having to explode it to find this out.

At this moment, this quantum computing-without-computing phenomenon is far from practical use – just like most quantum computing algorithms and most DNA computing algorithms are still far from practical use. However, there has been a lot of progress in this direction. For example:

- Initially, there was a fear that the probability of getting the correct result in the computing-without-computing setting may be too low to be practically useful.
- However, in 2006, a seminal paper [10] showed that this probability can be increased to almost 1.

The fact that in quantum computing, it is possible to perform some computations without actually running these computations encouraged us to check whether a similar phenomenon is possible for DNA computing as well. We were even further encouraged by the fact that computing without computing is also theoretically possible in yet another direction of using novel physical phenomena in computing – namely, in the use of acausal effects. Let us briefly recall this idea.

11.3 Computing Without Computing – Version Involving Acausal Processes: A Reminder

How can we speed up computations? A natural science fiction idea is to use a *time machine* (also known as an *acaual* – i.e. causality violating – process):

- we let the computer spend as much time as needed, even it means several thousand years, and then
- we use the time machine to bring these results back to us.

For a long time, acausal processes remained mostly the subject of science fiction. Serious physicists mostly believed that time machines are not possible – due to well-known paradoxes. These paradoxes can be summarized by stating the probably well-known paradox of time travel – the grandfather paradox: what if a time traveler goes into the past and kills his own grandfather before the traveler's parents are conceived?

In spite of the paradoxes, acausal processes continued to naturally emerge in many areas of physics. This emergence is mostly related to the fact that:

- in contrast to pre-quantum physics, where everything is deterministic,
- in quantum physics, we can only make probabilistic predictions.

In other words, there are always fluctuations, deviations of the actual values from the expected values of the corresponding physical quantities.

In pre-quantum physics, at each moment of time, a particle is in a certain spatial location, with a certain velocity – and, in principle, we can measure both location and velocity with any desired accuracy. In quantum physics, such exact measurements are no longer possible. A particle's location and velocity are always probabilistic: e.g. even if we prepare several particles in the identical states and measure their velocities, we will get slightly different results. And the smaller the region we consider, the larger these fluctuations.

Similarly, the space-time tensor – which describes the geometry of space-time and the direction of causality – fluctuates. The smaller the region we consider, the larger these fluctuations. As a result, the maximal possible speed fluctuates from the usual macroscopic speed-of-light value c:

- In some microscopic locations, the maximal speed is slightly larger than c.
- In some other microscopic locations, the maximal speed is slightly smaller than c.

If a microparticle follows the locations when the local maximal speed is larger than c, then, from the macroscopic viewpoint, this perfectly physical particle goes faster than the speed of light – and, according to special relativity, this implies the possibility of going back in time.

Many other schemes naturally appeared in physics, thus leading to acausal effects. As a result, in the late 1980s, a group of physicists led by a future Nobelist Kip Thorne decided to overcome the previous taboo and to seriously analyze possible acausal processes; see, e.g. [11–14].

But what about the paradoxes? Here, the probabilistic nature of quantum physics also helps. As we have mentioned, in quantum world, nothing is guaranteed. If the time traveler attempted to kill his grandfather, then:

- since the grandfather was alive enough after that attempt to sire a son,
- this means that this attempt failed.

In other words, some event happened, which prevented the killing:

- Maybe a policeman walked by and prevented the murder.
- Maybe the gun got stuck.
- Maybe a meteorite fell on the gun at that exact moment.

We can try to prevent all such events, but no matter how much we try, no matter how many possibilities we take into account, there is always a possibility of some rare, low-probability event that would disrupt the process. So, the only real consequence of trying to implement a time-travel paradox is that some very low-probability event will happen.

And, interestingly, this can be used to computations – i.e. we can use the *possibility* of acausal effects to perform computations without actually invoking these effects. In other words, we have another case of computing without computing. Indeed, suppose that:

- we are given a graph and
- we need to find a Hamiltonian path in this graph.

What we can do is:

- use a random number generators to generate some (random) path through this graph and then
- check if the resulting path is Hamiltonian.

If the path is not Hamiltonian, we launch a time machine – which is set up in such a way that its launch leads to some low-probability event, with probability $p_0 \ll 1$.

On the other hand, e.g. in a binary graph, the probability that a random selection of a direction at each of n nodes will lead to a selected path is 2^{-n}. So, nature has a choice:

- It can set up random processes so as to select a Hamiltonian path.
- It will have to implement a low-probability event, with probability $p_0 \ll 1$.

According to the general idea of statistical physics, in most cases, nature selects the event with higher probability. So, if $p_0 \ll 2^{-n}$, nature will select a Hamiltonian

path – and thus, we will find this path fast without actually having to use the time machine.

Comments

- This idea is described, e.g. in [15–22].
- Now that we have learned how computing without computing is possible in quantum and acausal computing, let us show how (and why) this idea is possible in DNA computing as well.

11.4 Computing Without Computing: – DNA Version

11.4.1 Main Idea

Let us show that with DNA computing, it is also possible to solve complex problems by using or simulating DNA inactivity.

The possibility of inactivity makes perfect biological sense:

- When resources are plentiful, it makes sense for the living creatures to be active and to actively multiply, but
- in situations when resources become scarce, such an activity would exhaust these resources really fast.

In such situations, it is important to slow down all the biological processes as much as possible.

In nature, we observe such slowing down all the time:

- from hibernating bears
- to plants that stop practically all activities in winter and
- to bacteria and viruses that can slow down to such an extent that they can survive in this slowed-down condition for hundreds and even thousands of years.

The slowdown occurs on all the levels:

- from the macro level, when an animal (e.g. a hibernating bear) stops moving almost completely,
- to the cell level, where all the usual biochemical processes grind practically to a halt.

On the DNA level, this means that instead of enhancing the possible ligations, in such situations, the cell tries to prevent ligation as much as possible, so as to keep all the processes inactive. This phenomenon has indeed been traced on the gene expression level; see, e.g. [23]. The possibility for such prevention comes from the fact that:

- contrary to a somewhat simplified version of DNA processes used in the traditional DNA computing,
- the actual DNA-related biochemical processes do not simply involve matching of different parts of the RNA and DNA.

There is also a *control* that switches some genes (i.e. some parts of the RNA and DNA) on and off. This control is determined:

- partly by other genes, and
- partly by the signals that the cell gets from the environment.

From this viewpoint, in the case of scarce resources, the corresponding control processes are organized in a way to maximally prevent ligations.

We will describe this control process in precise terms, and let us show that the corresponding problem is NP-hard – which means that it can be used to solve complex computational problems. But before we do that, let us explain why we believe that such control can be used for computations.

11.4.2 It Is Not Easy to Stop Biological Processes

The great potential of DNA computing comes from the fact that the corresponding biological processes are very complicated. In spite of the original optimism, even though the genomes of many living creatures – including humans – have been decoded, we are almost as far from the full understanding of the corresponding processes as before – and even farther from artificially synthesizing even the simplest living creatures. The problems are complex, but within each of numerous cells of numerous living creatures, nature solves the corresponding complex problems all the time. Thus, it is natural to try to use these naturally occurring solutions to solve our complex problems.

DNA processes are complex, but nature knows how to solve them – and thus, they occur all the time. Stopping these processes is much more difficult, even for nature – indeed, very few living creatures can do it, and we are still far from understanding how this is done:

- A grain left outside eventually spoils and rots, but some grains got preserved for thousands of years – and, when planted, turned into plants.
- Freezing kills most living creatures, but some mysteriously survive and get revived when thawed out.
- Viruses and bacteria can survive for years in the cosmic cold – there is even a *panspermia* hypothesis that this is how life spreads between the planets and this is how it originated on Earth.
- The possibility to stop biological processes in a human being – known as *anabiosis* – is a common feature in science fiction, but in real life, it remains a far-from-possible dream.

Since stopping of biological processes is too difficult, even more difficult than running them, it is even more reasonable to use this stopping – in addition to the DNA processes themselves – to solve other complex problems.

11.4.3 Towards Describing Ligation Prevention in Precise Terms

In general, we have several fragments that, in principle, have matching parts. Each fragment consists of several sub-fragments, and we can decide which of these

sub-fragments is switched on to be active. We want to select the sub-fragments in such a way that no two active sub-fragments are matched.

Here is a precise formulation of the problem.

11.4.4   What Is Given

We have several (N) nucleotide sequences ("fragments") $s_1, \ldots, s_N$, i.e. sequences consisting of symbols C, G, A, and T. Each fragment s_i is a concatenation of several subsequences ("sub-fragments") $s_i = s_{i1} \ldots s_{ik_i}$.

The sub-fragments s and s' *match* (or are *complementary*) if s' can be obtained from s by replacing A with T, T with A, C with G, and G with C.

11.4.5   What We Want to Find

The problem is to find the integers $j_1, \ldots, j_N$ such that $1 \leq j_i \leq k_i$ and for every two fragments i and i', the corresponding sub-fragments s_{ij_i} and $s_{i'j_{i'}}$ do not match.

11.4.6   Let Us Prove that the Ligation Prevention Problem Is NP-Hard

In practice, we are usually interested in the problems in which, once someone provides us with a candidate for a solution, we can feasibly tell whether this is a solution or not. The class of all such problems is known as the class NP; see, e.g. [18, 24].

Some computational problems are NP-hard, meaning that every problem from the class NP can be reduced to this problem. In other words, if we have an efficient algorithm for solving an NP-hard problem, this means that by reducing to this problem, we can solve *any* practical problem in feasible time [18, 24].

If a problem is NP-hard *and* itself belongs to the class NP, then this general problem is known as *NP-complete.*

Let us show that the ligation prevention problem is NP-hard. Since it is easy to check that no two sub-fragments are complementary to each other, this means that this problem is also in the class NP and is, thus, actually NP-complete.

This would mean that if – as we believe – nature has a way to solve the ligation prevention problem (at least many instances of this problem), then by reducing to this problem, we will be able to solve many practical problems in reasonable time.

11.4.7   How NP-Hardness Is Usually Proved

To show that a given problem P_{given} is NP-hard, it is sufficient to show that a known NP-hard problem P_{known} can be reduced to this problem. Indeed, by definition of NP-hardness, every problem P from the class NP can be reduced to P_{known}, and since the problem P_{known} can be, in its turn, reduced to P_{given}, this would mean that a two-stage reduction $P \rightarrow P_{\text{known}} \rightarrow P_{\text{given}}$ reduces P to P_{given}. Since this is true for every problem P from the class NP, this means that the given problem P_{given} is indeed NP-hard.

11.4.8 How We Will Prove NP-Hardness

As the known problem P_{known}, we select the propositional satisfiability problem for 3-CNF formulas, historically the first problem proven to be NP-hard. In this general problem, we deal with *Boolean* (= *propositional*) variables, i.e. variables $x_1, \ldots, x_v$ that can take two possible values: 1 (meaning "true") and 0 (meaning "false"). A *literal a* is either a variable x_k or its negation $\neg x_k$.

A *clause* is an expression of the type $a \vee b$ or $a \vee b \vee c$ where a, b, and c are literals. Examples are $x_1 \vee \neg x_2$ or $\neg x_1 \vee \neg x_5 \vee x_9$.

Finally, a *formula F* is an expression of the type $C_1 \,\&\, C_2 \,\&\, \ldots \,\&\, C_m$, where C_i are clauses. An example of a formula is the expression

$$(x_1 \vee \neg x_2) \,\&\, (\neg x_1 \vee \neg x_5 \vee x_9).$$

The general propositional satisfiability problem is:

- given a formula,
- find the values of the variables that make it true (or, to be more precise, to check whether there exist values x_i that make it true).

11.4.9 The Actual Proof by Reduction

Let us assume that we are given an instance F of the propositional satisfiability problem, i.e. that we are given a propositional formula F of the type $C_1 \,\&\, \ldots \,\&\, C_m$ with v Boolean variables $x_1, \ldots, x_v$.

To reduce this instance to an appropriate instance of the ligation prevention problem, first, we assign, to each Boolean variable x_j, a fragment $f(x_j)$ consisting of letters C, G, A, and T, in such a way that fragments assigned to two different variables are not complementary.

There are many ways to do it. For example, we can assign v different fragments to v variables and then add a letter A in front of each of these fragments. This way, no two fragments will fully match, since for them to match, their first symbols must match as well, but A does not match with A – it only matches with T.

To each negation $\neg x_j$, we assign a fragment – which we will denote by $f(\neg x_j)$ – which is complementary to $f(x_j)$, i.e. which is obtained from $f(x_j)$ by replacing A with T, T with A, C with G, and G with C.

Finally, to each clause C_i, we assign a fragment s_i in the following way:

- If the clause has the form $a \vee b$, then we take a fragment $s_i = f(a)f(b)$ consisting of two sub-fragments $f(a)$ and $f(b)$.
- If the clause has the form $a \vee b \vee c$, then we take a fragment $s_i = f(a)f(b)f(c)$ consisting of three sub-fragments $f(a)$, $f(b)$, and $f(c)$.

Let us show that the original formula F is satisfiable if and only if it is possible to select a sub-fragment in each fragment s_i so that none of the selected sub-fragments are complementary to each other.

Indeed, if the formula F is satisfiable, this means that there exists an assignment of truth values to all the Boolean variables $x_1, \ldots, x_v$ that makes the formula F true – which means that each of the clauses C_i is true. The fact that a clause

C_i is true means that one of its literals is true. We thus select a sub-fragment corresponding to one of the true literals.

No two selected sub-fragments are complementary to each other – indeed, complementary would mean that they represent a variable x_j and its negation $\neg x_j$ and the variable and its negation cannot be both true.

Vice versa, let us assume that for each fragment s_i corresponding to a clause $C_i = a \vee \ldots$, we selected a sub-fragment – let us denote it by $f(a_i)$ – so that no two sub-fragments are complementary to each other. The fact that they are not complementary means that no two corresponding literals a_i and $a_{j'}$ are negations of each other. Thus, we can assign the truth value to each of the Boolean variables x_j as follows:

- If one of the selected sub-fragments has the form $f(x_j)$, then we make the Boolean variable x_j true.
- If one of the selected sub-fragments has the form $f(\neg x_j)$, then we make the Boolean variable x_j false.
- If none of the selected sub-fragments are of the form $f(x_j)$ or $f(\neg x_j)$, then we assign any truth value to x_j.

Since no two selected sub-fragments have the form $f(x_j)$ and $f(\neg x_j)$, this means that this assignment is consistent. In this assignment, for each clause C_i, the literal a_i corresponding to the selected sub-fragment $f(a_i)$ is true. Thus, under this assignment, each clause C_i is true, and hence, the whole formula $F = C_1 \, \& \, \cdots \, \& \, C_m$ is true.

The reduction is proven.

Comment   While we reduced propositional satisfiability to our problem, in fact, this proof can be viewed as reducing another NP-complete problem to our problem – namely, the problem of finding a *clique* of given size k in a given graph. A clique is defined as a subset of the graph's vertices in which every two vertices are connected to each other by an edge. Our proof is actually a modification of the standard proof that the clique problem is NP-complete; see, e.g. [24].

In this proof, we reduce the propositional satisfiability problem to the clique problem in the following way. Let an instance F of the propositional satisfiability problem be given. This instance has the form $C_1 \, \& \, \ldots \, \& \, C_m$, where C_i are clauses. For each literals a from each clause C_i, we add a vertex $V_i(a)$ to the resulting graph.

For example, for the formula $(x_1 \vee \neg x_2) \, \& \, (x_1 \vee x_2 \vee x_3)$, we have a graph with five vertices:

- Two vertices $V_1(x_1)$ and $V_1(\neg x_2)$ corresponding to the first clause.
- Three vertices $V_2(x_1)$, $V_2(x_2)$, and $V_2(x_3)$ corresponding to the second clause.

We then connect, by edges, vertices corresponding to different literals provided that they do not correspond to opposite literals x_i and $\neg x_i$. In the above example,

- the vertex $V_1(x_1)$ is connected to $V_2(x_1)$, $V_2(x_2)$, and $V_2(x_3)$; and
- the vertex $V_1(\neg x_2)$ is connected to $V_2(x_1)$ and $V_2(x_3)$ (but *not* to $V_2(x_2)$).

The fact that this is indeed a reduction can be easily proven.

Indeed, if the original formula F is satisfiable, then in each clause, (at least) one of the literals is true. We can select one true literal in each clause. These literals cannot be opposite: since we cannot have x_i and $\neg x_i$ both true. Thus, every two corresponding vertices are connected – i.e. the resulting subgraph indeed forms a clique of size m.

Vice versa, if we have a clique of size m, then, since literals corresponding to the same clause are not connected, this means that vertices from this clique correspond to different clauses. And since we have exactly m clauses, this means that the clique contains exactly one vertex corresponding to each clause. Now, we can select, for each variable x_i, the value "true" or "false" depending on whether the clique contains a vertex corresponding to x_i or a vertex corresponding to $\neg x_i$. The clique cannot contain both – since vertices corresponding to opposite literals are not connected. (For the variables not reflected in any of the vertices from the clique, we can select any truth value.)

Since each clause C_i contains at least one vertex $V_i(a)$ from the clique, the corresponding literal a is true in this assignment, and thus, the clause C_i is also true. So, under this assignment, all clauses are true – and hence, the original formula $F = C_1 \,\&\, \ldots \,\&\, C_m$ is also true.

The reduction is proven.

11.5   DNA Computing Without Computing Is Somewhat Less Powerful than Traditional DNA Computing: A Proof

11.5.1   Which of the Two DNA Computing Schemes is More Powerful?

In Section 11.4, we have shown that, in addition to the traditional DNA computing that utilizes the actual DNA-related chemical processes, we can also perform effective computations by using the ability of a body to stop these chemical processes. A natural question is: which of the two DNA computing schemes is more powerful, the active or the passive one?

Overall, they are both NP-complete; in this sense they are both equally powerful. However, we can still talk about which problems are more powerful and which problems are less powerful if we take into account a subtle subdivision of NP-complete problems.

11.5.2   W-hierarchy: A Brief Reminder

The subtle subdivision that we have in mind – called *W-hierarchy* – is based on the notions of *fixed parameter tractable* (fpt) problems and of *weft*. We will briefly explain these notions in this section; readers interested in detail can check, e.g. [25–27].

The main idea is that while a problem may be, in general, NP-hard – which means that unless it turns out that P = NP, we cannot have a feasible (polynomial-time) algorithm for solving this problem – there usually is a parameter k such that if we bound the value of this parameter, the problem can

be solved in polynomial time, i.e. for some computable functions $f(k)$ and $C(k)$, a problem with input x of size $n \overset{\text{def}}{=} \text{len}(x)$ can be solved in time $f(k) \cdot n^{C(k)}$. This way, if we fix some bound k_0 and only consider problems for which the value of k is bounded by k_0, then all thus limited problems can be solved in time $\leq f_0 \cdot n^{C_0}$, where $f_0 \overset{\text{def}}{=} \max(f(1), f(2), \ldots, f(k_0))$ and $C_0 \overset{\text{def}}{=} \max(C(1), C(2), \ldots, C(k_0))$.

For some problems, the corresponding exponent $C(k)$ does not grow with k. Such problems are called *fixed parameter tractable* (fpt). In precise terms, a problem is fpt if, for some computable function $f(k)$ and for some constant C, a problem with input x of size $n = \text{len}(x)$ can be solved in time $f(k) \cdot n^C$. This way, if we fix some bound k_0 and only consider problems for which the value of k is bounded by k_0, then all thus limited problems can be solved in time $\leq f_0 \cdot n^C$.

Similarly to the usual reduction, we can define *fpt-reduction* as a reduction that preserves both the size of the inputs (modulo a possible feasible – polynomial-size – increase) *and* preserves the bounds on the parameter, so that problems for which the value of the parameter is bounded by some value k_0 get transformed into problems for which the parameter is bounded by $g(k_0)$ for some feasible function $g(x)$.

The W-hierarchy is based on reduction to computational schemes of a certain *weft*. To describe the weft of a computation scheme, we first represent this scheme as a directed graph:

- whose vertices are elementary logical (bit) operations, and
- where an edge from a vertex a to a vertex b means that the output of a is one of the inputs of the operation b.

For commutative and associative logical operations,

- in addition to the usual binary operations,
- we also allow operations with more than two inputs.

Such operations are "and," "or," and addition modulo 2 (which is the same as "exclusive or").

The weft is defined as the largest number of logical units from an input to the output. For each natural number $i = 0, 1, 2, \ldots$, the ith class $W[i]$ is defined as the class of the problems that can be reduced to a computation scheme of weft $\leq i$ with several inputs $v_1, \ldots, v_m$ and one output v for which:

- the original problem x with parameter k has a solution if and only
- there is a combination of inputs $v_1, \ldots, v_m$ that produces the result $v =$"true" and in which at most k inputs v_j are 1s – the rest are 0s.

It can be shown that $W[0]$ is exactly the class FPT of all fpt problems, and one can easily see that $W[0] \subseteq W[1] \subseteq W[2] \subseteq \ldots$

It is not proven that classes $W[i]$ corresponding to different i are indeed different, but most computer scientists believe that they *are* different, i.e. that the containment is strict: $W[0] \subset W[1] \subset W[2] \subset \ldots$ Within each class $W[i]$, there are problems that are the hardest in this class – in the sense that every other problem from this class can be fpt-reduced to this problem. Such problems are called *W[i]-complete*.

In particular,

- *the Hamiltonian path problem* – historically the first problem for which a DNA-based solution has been proposed – has been proven to be $W[2]$-complete for k being the graph width (see, e.g. [28]), while, e.g.
- *the Clique problem* – the problem of finding, in a given graph a clique of a given size k – is known to be $W[1]$-complete; see, e.g. [25, 26].

Since

- the original DNA computing solves the Hamiltonian path problem while
- the DNA-based computing without computing is equivalent to the clique problem,

we thus arrive at the following conclusion.

11.5.3 Conclusion

The traditional DNA computing is more powerful than DNA computing without computing.

Specifically, while both traditional DNA computing and DNA-based computing without computing solve NP-complete problems,

- the traditional DNA computing is $W[2]$-complete, while
- the DNA-based computing without computing is only $W[1]$-complete, i.e. complete for the somewhat less complex class of the W-hierarchy.

11.6   First Related Result: Security Is More Difficult to Achieve than Privacy

11.6.1   What We Plan to do in this Section

The result from Section 11.5 can be applied to a topic that is not related to DNA computing, but that is very important: the need to maintain privacy and security when using computers.

The reason why such an application is possible is that the main problems of both privacy and security can be reformulated in graph terms.

11.6.2   How to Describe Privacy in Graph Terms

Privacy means that

- while we *should* have access to our own records,
- we *should not* get unauthorized access to any other records.

This means, in particular, that

- if we perform a simple modification of code words and other means to get access to our own records,
- we should not be able to gain access to records of anybody else (unless that person gave us a special permission).

To describe this in graph terms, let us form a graph in which individuals are vertices and two vertices *a* and *b* are connected if:

- it is not possible for the individual corresponding to vertex *a* to access *b*'s record by a simple modification of *a*'s access information, and
- vice versa, it is not possible for the individual corresponding to vertex *b* to access *a*'s record by a simple modification of *b*'s access information.

Each abstract access scheme can be represented as such a general graph. The question is: can we use this abstract scheme to provide full privacy for a given number *k* of users? In terms of the above graph, this is equivalent to finding a subset of *k* vertices in which every two vertices are connected to each other – i.e. to finding a clique of the given size *k*.

Thus, in graph terms, maintaining privacy is equivalent to solving the clique problem. We already know that this problem is NP-complete and $W[1]$-complete.

11.6.3   How to Describe Security in Graph Terms

In general, computer security (and security in general) means that we have resources so that:

- if we have trouble at some location (physical or virtual),
- one of these resources is available to resolve the corresponding problem.

In the ideal world, we should have such resources at each location. However, realistically, this is usually not possible, so only some locations have resources. In terms of the police example, this means that:

- while we cannot place a police officer at every house, but
- we need to make sure that if a crime is reported, the police from the nearby police station should arrive on time to stop this crime.

Similarly, in computer security, if a suspicious message appears on a computer, the corresponding server should be able to block the corresponding virus from infecting other computers.

This situation can also be described as a graph. Namely, its vertices are possible locations. We connect the two locations *a* and *b* if these two locations are "close" in the following sense:

- a resource located at location *a* can reach location *b* in time to resolve any possible problem, and
- a resource located at location *b* can reach location *a* in time to resolve any possible problem.

Based on the geography and/or on communication ability of the corresponding network, we can form a graph of possible locations, in which edges correspond to the above "closeness." Our overall resources are limited. So, the question is:

- given that we only have *k* resources,
- is it possible to place them in such a way that every location in the graph is covered – i.e. that each vertex is close to one of the *k* selected locations?

In graph terms, the corresponding set of k locations is called a *dominating set.* In these terms, the question is: given a graph, is there a dominating set of size k in this graph? It is known that this problem is NP-complete and $W[2]$-complete; see, e.g. [29].

11.6.4 Conclusion: Security Is More Difficult to Maintain than Privacy

Since

- security corresponds to a $W[2]$-complete problem and
- privacy corresponds to a $W[1]$-complete problem – which are, in general, somewhat less complex than $W[2]$-complete problems,

we can therefore conclude that security is somewhat more difficult to maintain than privacy.

11.7 Second Related Result: Data Storage Is More Difficult than Data Transmission

11.7.1 Application to Information Science

A similar result is related to information science, the science of *storing* and *transmitting* information; see, e.g. [30]. This result is very relevant for DNA computing, since this is exactly the main objective of DNA: to store and transmit the biological information.

11.7.2 Data Storage

The first type of problems relates to the first objection of information science: storing information. Let us consider situations in which we need to store information about different objects. Let X denote the set of the corresponding objects. In mathematical terms, these objects may be signals, 2D images, 3D bodies, etc.

In many practical cases, storing all possible information about each object requires too much memory space. For example,

- if we want to store the whole information about a human body cell by cell,
- we will need to store all the information about billions of cells, the relation between them, etc. – this is not easy to store.

In practice,

- we often do not need the exact information,
- it is usually sufficient to reconstruct it with some reasonable accuracy.

For example, if we want to store a photo, a minor change in intensity will not even be noticeable by a human eye.

To describe this in precise terms, we can form a graph in which

- vertices are elements of the set X and
- two objects x and y are connected by an edge if and only if they are practically indistinguishable, i.e. if, for practical purposes, it is OK to reconstruct x if the actual object is y and vice versa.

Usually, indistinguishability is described by a formula $d(x, y) \leq \varepsilon$ for an appropriate metric $d(x, y)$ on the set X and an appropriate positive real number $\varepsilon > 0$.

So, instead of storing the actual elements $x \in X$, we only store, for each element x, its approximation s – which should be practically indistinguishable from x. The set S of all such approximation must be such that each element $x \in S$ is practically indistinguishable from some element $s \in S$ – i.e. in graph terms, the set S must be a dominating set in the corresponding graph.

For example, if we want to store a single real number and we are OK with reconstructing it with accuracy 2^{-n}, then we can restrict ourselves to numbers $0, 2^{-n}$, $2 \cdot 2^{-n}, 3 \cdot 2^{-n}$, etc.

How many bits do we need to store such approximating elements? We need as many bits as are needed to distinguish between different elements of the set S:

- If we use 1 bit, which has 2 possible values 0 and 1 – which can represent 2 different elements.
- If we use 2 bits, with $2^2 = 4$ possible combinations, we can represent 4 different elements.
- With b bits, we can represent 2^b different elements.

So, to represent a set consisting of k elements, we need to have $2^b \geq k$. The smallest such number of bits is $\lceil \log_2(k) \rceil$.

Thus, to find out how many bits of memory we need to represent each element of the original set S, we need to know the binary logarithm of the smallest size of the dominating set. This binary logarithm is known as ε-*entropy*. This notion was first introduced by Kolmogorov and his research group [31–33]; they also provided asymptotic formulas for the ε-entropy of different function spaces X.

It is known that computing ε-entropy is NP-hard. The above result shows that this problem is $W[2]$-complete.

11.7.3 Data Transmission

The data needs to be transmitted. Let us denote by n the overall number of signals that we want to send. We need to assign, to each of these signals s, a physical signal $x(s)$. Examples are:

- the sequence of instantaneous pulses – as when the information is transmitted in a brain, or
- a sequence of shorter and longer pulses, as in the Morse code.

Transmission usually comes with noise. We therefore need to make sure that, even when the transmitted signals are corrupted by noise, we can still distinguish between them. Let us describe this problem in precise terms. Let X denote the set of all physical signals. We can then form a graph in which:

- possible physical signals are vertices, and
- two signals are connected if and only they can still be distinguishable after applying the noise.

For example, if we know the largest possible change δ caused by a noise – i.e. we know that the distance $d(x, \tilde{x})$ between the original signal x and the noised signal $\tilde{x}$ cannot be larger than δ – then the signals x and y can be separated if $d(x, y) > \varepsilon \overset{\text{def}}{=} 2\delta$. Indeed, in this case, from the triangle inequality, we can conclude that

$$d(\tilde{x}, \tilde{y}) \geq d(x, y) - d(x, \tilde{x}) - d(y, \tilde{y}) > 2\delta - \delta - \delta = 0,$$

so $d(\tilde{x}, \tilde{y}) > 0$ and thus $\tilde{x} \neq \tilde{y}$. So, corrupted versions of two different signals are always different.

Once we know the set X of possible physical signals, we want to know whether we can use these signals to correctly transmit a given number k of different signals in the presence of noise – and, if yes, what physical signal $x(s)$ we should use to transmit each symbol s from the original messages. In terms of the above described graph, this means that we need to find a clique of size k in the graph. As we have mentioned, this problem is $W[1]$-hard.

11.7.4 Conclusion: Data Storage Is More Difficult than Data Transmission

Since

- data storage corresponds to a $W[2]$-complete problem, and
- data transmission corresponds to a $W[1]$-complete problem – which are, in general, somewhat less complex than $W[2]$-complete problems,

we can therefore conclude that data storage is a somewhat more difficult problem than data transmission.

Acknowledgments

References

1 Adleman, L.M. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266 (5187): 1021–1024.
2 Amos, M. (2005). *Theoretical and Experimental DNA Computation*. Berlin, Heidelberg: Springer-Verlag.

3 Ignatova, Z., Martínez-Pérez, I., and Zimmermann, K.-H. (2008). *DNA Computing Models*. New York: Springer.

4 Namasudra, S. and Deka, G.C. (eds.) (2019). *Advances of DNA Computing in Cryptography*. Boca Raton, FL: CRC Press.

5 Paun, G., Rozenberg, G., and Salomaa, A. (2006). *DNA Computing: New Computing Paradigms*. Berlin, Heidelberg: Springer-Verlag.

6 Thachuk, C. and Liu, Y. (2019). *DNA Computing and Molecular Programming, Lecture Notes in Computer Science*, vol. 11648. Proceedings of the 25th International Conference DNA'25, Seattle, Washington, USA, 5–9 August 2019. Springer.

7 Nielsen, M.A. and Chuang, I.L. (2000). *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press.

8 Williams, C.P. and Clearwater, S.H. (2000). *Ultimate Zero and One: Computing at the Quantum Frontier*. New York: Springer.

9 Jozsa, R. (1999). Quantum effects in algorithms. In: *Quantum Computing and Quantum Communications, Springer Lecture Notes in Computer Science*, Chapter 7, vol. 1509, Selected Papers from the First NASA International Conference QCQC'98, Palm Springs, California, USA, February, 1998 (ed. C.P. Williams), 103–112. New York: Springer.

10 Hosten, O., Rakher, M.T., Barreiro, J.T. et al. (2006). Counterfactual quantum computation through quantum interrogation. *Nature* 439 (7079): 949–952.

11 Morris, M.S. and Thorne, K.S. (1988). Wormholes in spacetime and their use for interstellar travel: a tool for teaching general relativity. *American Journal of Physics* 56: 395–412.

12 Morris, M.S., Thorne, K.S., and Yurtzever, U. (1988). Wormholes, time machines, and the weak energy condition. *Phys. Rev. Lett.* 61: 1446–1149.

13 Thorne, K.S. (1991). Do the laws of physics permit closed timelike curves? *Ann. N.Y. Acad. Sci.* 631: 182–193.

14 Thorne, K.S. (1994). *From Black Holes to Time Warps: Einstein's Outrageous Legacy*. New York: W. W. Norton & Company.

15 Dimitrov, V., Koshelev, M., and Kreinovich, V. (1997). Acausal processes and astrophysics: case when uncertainty is non-statistical (fuzzy?). *Bulletin for Studies and Exchanges on Fuzziness and its Applications (BUSEFAL)* 69: 183–191.

16 Koshelev, M. and Kreinovich, V. (1997). Towards computers of generation omega – non-equilibrium thermodynamics, granularity, and acausal processes: a brief survey. In: *Proceedings of the International Conference on Intelligent Systems and Semiotics ISAS'97* (ed. A.M. Meystel), 383–388. Gaithersburg, MD: U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology.

17 Kosheleva, O.M. and Kreinovich, V. (1981). What can physics give to constructive mathematics. In: *Mathematical Logic and Mathematical Linguistics (in Russian)*. Kalinin, Russia: Kalinin State University. 117–128.

18 Kreinovich, V., Lakeyev, A., Rohn, J., and Kahl, P. (1998). *Computational Complexity and Feasibility of Data Processing and Interval Computations*. Dordrecht: Kluwer.

19 Kreinovich, V. and Mignani, R. (1977). Noncausal quantum processes and astrophysics. *Boll. Soc. Ital. Fis.* 112: 88.

20 Maslov, S.Yu. (1987). *Theory of Deductive Systems and Its Applications*. Cambridge, MA: MIT Press.

21 Moravec, H. (1991). *Time Travel and Computing*. Carnegie-Mellon University, Computer Science Department, Preprint.

22 Novikov, I.D. (1989). Analysis of the operation of a time machine. *Sov. Phys. JETP* 68: 439–443.

23 Jansen, H.T., Trojahn, S., Saxton, M.W. et al. (2019). Hibernation induces widespread transcriptional remodeling in metabolic tissues of the grizzly bear. *Commun. Biol.* 2. doi: https://doi.org/10.1038/s42003-019-0574-4.

24 Papadimitriou, C. (1994). *Computational Complexity*. Reading, MA: Addison-Wesley.

25 Cygan, M., Fomin, F.V., Kowalik, L. et al. (2015). *Parameterized Algorithms*. New York: Springer.

26 Downey, R.G. and Fellows, M.R. (2013). *Fundamentals of Parameterized Complexity*. London: Springer.

27 Niedermeier, R. (ed.) (2006). *Invitation to Fixed Parameter Algorithms*. Oxford: Oxford University Press.

28 Lampis, M., Kaouri, G., and Mitsou, V. (2011). On the algorithmic effectiveness of digraph decompositions and complexity measures. *Discrete Optim.* 8: 129–138.

29 Downey, R.G. and Fellows, M.R. (1995). Fixed-parameter tractability and completeness I: basic results. *SIAM J. Comput.* 24: 873–921.

30 Ahlswede, R. (2014). *Storing and Transmitting Data*. Cham: Sringer.

31 Kolmogorov, A.N. (1956). On certain asymptotic characteristics of completely bounded metric spaces (in Russian). *Dokl. Akad. Nauk SSSR* 108 (3): 385–388.

32 Kolmogorov, A.N. and Tikhomirov, V.M. (1961). ε-entropy and ε-capacity of sets in functional spaces. *Am. Math. Soc. Transl. Ser. 2* 17: 277–364.

33 Vitushkin, A.G. (1961). *Theory of Transmission and Processing of Information*. Oxford: Pergamon Press.

12

DNA Computing: Versatile Logic Circuits and Innovative Bio-applications

Daoqing Fan, Erkang Wang, and Shaojun Dong

State Key Laboratory of Electroanalytical Chemistry, Changchun Institute of Applied Chemistry, Chinese Academy of Sciences, 5625 Renmin Street, Changchun, Jilin 130022, China

As smart molecular-level biocomputers, DNA logic circuits have been considered as one of the most promising alternatives to traditional silicon-based semiconductor circuits because of the low cost, predictable structure, facile design, and powerful parallel computing ability of DNA. And DNA computing has also exhibited magical power in various intelligent bio-applications (bioanalysis, diagnostics, drug load/release, and so on) in recent decades.

In this chapter, we will illustrate the tutorial definition, logical principle, and classification of DNA logic circuits initially. Then, we summarize representative achievements about DNA computing in our group (most) and other ones during the past decade, in which the comprehensive operating mechanisms of various advanced DNA logic circuits (arithmetic, non-arithmetic logic devices, cascade/concatenated circuits, and so on) based on different systems are summarized. After that, the innovative bio-applications of DNA computing are alternatively exhibited. Finally, current bottlenecks and future directions of this field are further suggested. We hope this chapter will benefit both researchers and newcomers of this field.

12.1 Definition, Logical Principle, and Classification of DNA Computing

DNA computing is a kind of molecular demonstration of Boolean logic, in which George Boole used binary numbers to perform logic operations that enabled the assignment of "false" to "0" and "true" to "1" in form of bits [1–3]. Since de Silva operated the first molecular AND gate using the fluorescent signals of organic molecule as outputs in 1993 [4], this area has attracted much attentions and gained extensive developments [5]. DNA logic circuits are biomolecular devices that execute Boolean logic; they are stimulated by binary-encoded inputs (0/1) and could generate binary outputs (0/1). This promising field was pioneered by Adleman's work using DNAs to solve Hamilton problems in 1994 [6]. But, unlike silicon circuits, DNA logic circuits use binary-encoded

Table 12.1 Truth table of various basic logic gates.

Input		AND	OR	INHIBIT	XOR	NAND	NOR	IMPLICATION	XNOR
					Output				
					Basic logic gates				
0	0	0	0	0	0	1	1	1	1
0	1	0	1	0	1	1	0	1	0
1	0	0	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0	1	1

DNAs as inputs (absence "0"/presence "1") and optical (fluorescent, visual, chemiluminescent)/electrochemical signals as outputs (low "0"/high "1") [7–10].

There are various basic logic gates, such as the simplest 1-input YES, 2-input AND, OR, INHIBIT, and XOR gates and the ones with opposite functions (NOT, NAND, NOR, IMPLICATION, and XNOR). Their corresponding logical principles are presented in Table 12.1 in detail. These basic logic gates are the building elements of complicated arithmetic/non-arithmetic logic devices with specific functions, and the cascade/concatenated logic circuits can be further achieved through their reasonable integration to realize more sophisticated logical computation. According to the operating principles, we herein made a general classification for current advanced DNA logic circuits, which was shown in Table 12.2. It should be noted that this classification was based on our scientific experience in this area and just used to depict this tutorial chapter more reasonably.

12.2 Advanced Arithmetic DNA Logic Devices

12.2.1 Half-Adder, Half-Subtractor

There are many different kinds of advanced arithmetic DNA logic devices, in which the half-adder/half-subtractor is one of the most typical ones. They are triggered by two inputs and could generate two different outputs. A half-adder could achieve the addition of two binary digits via the parallel operation of an XOR gate and an AND gate [11–14]. The XOR gate can generate a SUM (S) output, and the AND will produce a CARRY (C) one, respectively. Similarly, a half-subtractor can perform the subtraction of two bits through the parallel integration of an XOR gate and an INHIBIT gate, in which the former gate generates a DIFFERENCE (D) signal and the latter produces a BORROW (B) one, respectively. By taking advantage of the fluorescence quenching ability of graphene oxide (GO) (long-range resonance energy transfer) towards FAM-labeled DNA and fluorescence enhancing ability of G-quadruplex (G4) towards NMM (a kind of porphyrin dye), our group constructed half-adder and half-subtractor at the same time based on the universal GO/DNA platform (Figure 12.1a) [13].
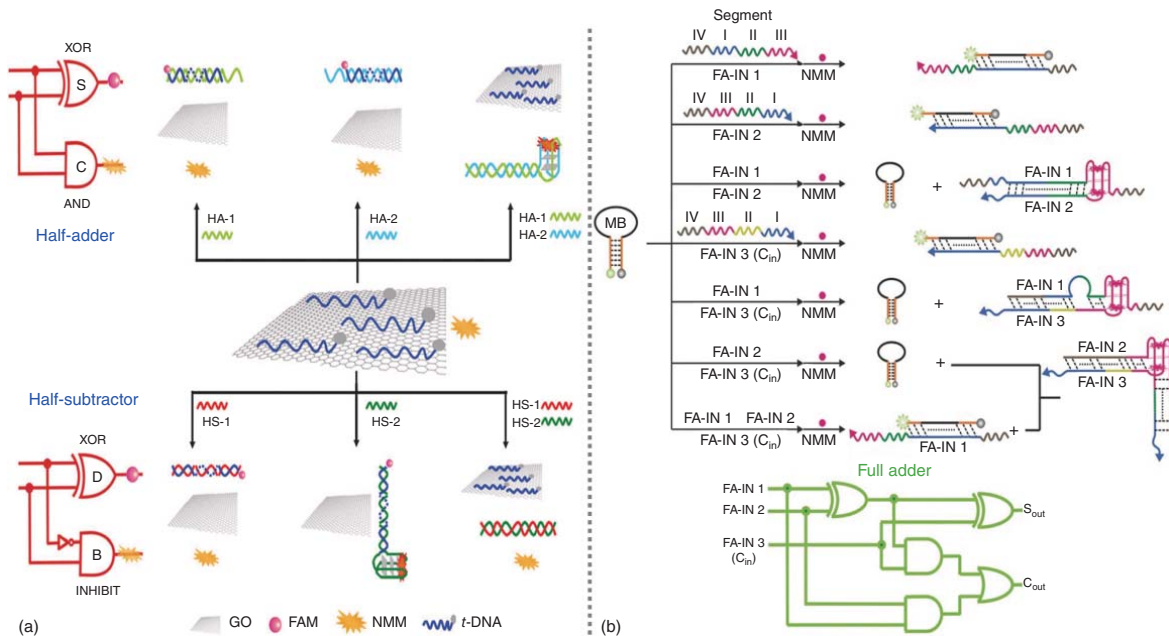
Figure 12.1 (a) The operation of half-adder/half-subtractor based on the universal GO/DNA platform. Source: From Wang et al. [13]. © 2014. Reproduced with permission of Royal Society of Chemistry. (b) The full-adder based on the integration of dual-fluorophore-labeled molecular beacon and G4/NMM. Source: From Li et al. [14]. © 2015. Reproduced with permission of Wiley-VCH.

Table 12.2 General classification of different advanced DNA logic circuits.

Advanced logic circuits	Logical composition or functions	
Arithmetic devices	Half-adder	AND+XOR
	Half-subtractor	INHIBIT+XOR
	Full-adder	AND+XOR+OR
	Full-subtractor	INHIBIT+XOR+OR
	Others	—
Non-arithmetic devices	Encoder/decoder	Binary data/code conversion
	Multiplexer/demultiplexer	Binary data compression/decompression
	Parity checker for natural numbers	Distinguish even/odd natural numbers less than 10
	Voter	Majority logic gate
	Keypad lock	Sequential logic
	Parity generator/checker (pG/pC)	Error detection through data transmission (cascade of XOR gates)
	Non-Boolean ternary logic gates	Three input/output states (0/1/2)
	Others	—
Concatenated circuits	The cascade or parallel integration of different logic gates and advanced devices	

12.2.2 Full-Adder, Full-Subtractor

As one kind of DNA logic circuits with the highest complexity, a full-adder could achieve the addition of three binary bits. And it requires the integration of two half-adders and another OR gate [12, 14]. In our previous work, we constructed full-adder and full-subtractor based on the same dual-labeled molecular beacon (MB) system that do not need any nanoquencher (such as GO) [14]. As shown in Figure 12.1b, any one of the three inputs, FA-IN 1, FA-IN 2, and FA-IN 3 (CARRY IN, C_{in}), could open the MB hairpin. In the coexistence of any two inputs, their mutual hybridization will prohibit the interaction of each input with MB, producing a weak FAM signal (SUM out). Meanwhile, the hybridization of any two inputs could yield an intact G4, generating a high fluorescent signal of NMM (CARRY out). While, if all three inputs were inputted, FA-IN 2–3 could hybridize with each other in the system, and the left FA-IN 1 could still open the MB. Thus, above operations could fulfill the requirements of a full-adder. Similarly, a full-subtractor that consists of two half-subtractors and one OR gate was further achieved through the reasonable sequence design [14].

These advanced arithmetic DNA logic devices described above play important roles in solving sophisticated computing problems, which will be competitive with semiconductor electronics in the long term and also lay solid foundation for intelligent bio-applications in the future.

12.3 Advanced Non-arithmetic DNA Logic Devices

Apart from above advanced arithmetic logic devices, the non-arithmetic ones are also essential for information processing at the molecular level. In the following, we will introduce typical non-arithmetic DNA logic circuits, including encoder/decoder, multiplexer/demultiplexer, the parity checker for distinguishing even/odd natural numbers, voter, keypad lock, parity generator/checker (pG/pC system) for error detection through data transmission, and non-Boolean ternary logic gates [15–27].

12.3.1 Data Conversion: Encoder/Decoder, Multiplexer/Demultiplexer

Encoders and decoders are advanced non-arithmetic logic devices and have widespread applications in molecular computing because of their ability to convert binary data/code into code/data. The encoder could compress information for transmission or storage via converting the data signal into a code, and the decoder just has opposite functions. In our work, taking 2-to-1 and 4-to-2 encoders and a 1-to-2 decoder as model devices, we for the first time combined the quenching ability of GO to DNA-templated AgNCs with G4-enhanced fluorescence intensity of porphyrin dyes together and constructed a label-free and enzyme-free platform for operating encoders/decoders (Figure 12.2A) [15]. Compared with previous DNA encoders/decoders, the above strategy largely reduced the time and costs and presented obvious advantages.
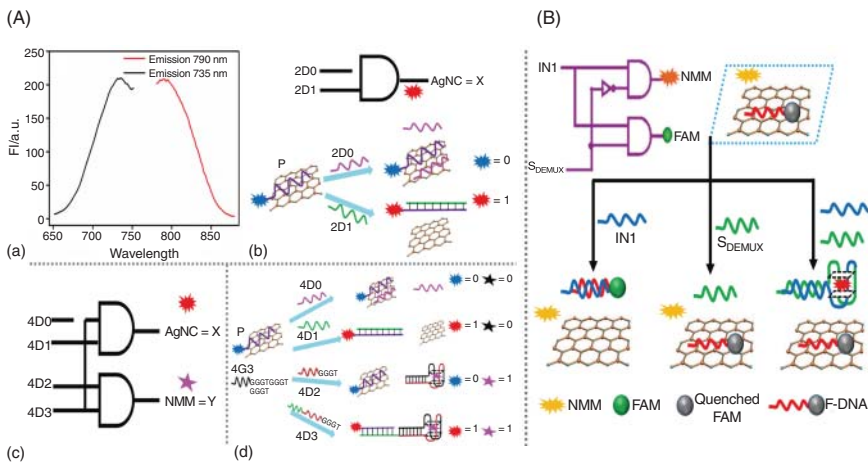
Figure 12.2 (A) The operation of encoder/decoder based on the universal GO/DNA-AgNC platform. (a) The excitation/emission spectra of DNA-AgNCs. (b) Schematic operation of 2-to-1 encoder. (c) Equivalent logic circuit of 4-to-2 encoder. (d) Schematic operation of 4-to-2 encoder based on GO/DNA-AgNCs/G4 system. Source: From Fan et al. [15]. © 2016. Reproduced with permission of Royal Society of Chemistry. (B) The multiplexer/demultiplexer based on the integration of GO/FAM-DNA and G4/NMM. Source: From Wu et al. [17]. © 2015. Reproduced with permission of Royal Society of Chemistry.

Different from encoders/decoders, the multiplexers/demultiplexers could act as controlled switches and are also widely used in electronic and signal processing systems for data compression and decompression [16, 17]. A 2-to-1 multiplexer can transmit two input signals into a single output channel. A 1-to-2 demultiplexer just plays the opposite role in transmitting a single input signal into two output channels. Analogously, by utilizing the GO/FAM-DNA and G4/NMM system, we achieved enzyme-free DNA 2-to-1 multiplexer and 1-to-2 demultiplexer based on the same platform (Figure 12.2B) [17].

12.3.2 Distinguishing Even/Odd Natural Numbers: The Parity Checker

Among various advanced non-arithmetic DNA logic devices, the parity checker that could distinguish the even/odd natural numbers is one of the most practical ones. As the natural numbers are endless, the parity checkers were usually constructed for discriminating natural numbers less than 10 (0–9). Based on a versatile 3D DNA nanoprism and the FRET between fluorophore and quencher, Nie and coworkers constructed a DNA parity checker successfully (Figure 12.3a) [18]. For its operation, 10 decimal numbers were transformed into 10 groups of different binary-coded decimal (BCD) code initially. By utilizing four ssDNA strands as inputs, the corresponding fluorescent intensity at 603 nm of all the odd numbers (1, 3, 5, 7, 9) was higher than the threshold value, producing the positive outputs, while the even ones (0, 2, 4, 6, 8) generated false outputs. Thus, the even/odd numbers can be easily distinguished.

12.3.3 DNA Voter and Keypad Lock

The reasonable integration of basic logic gates could form functional logic devices with specific functions, such as the DNA voter. A DNA voter is typically based on a majority logic gate (MAJ) [19, 20]. In Boolean logic, a *TRUE* decision will be made by a MAJ only in the presence of more than half of the inputs, which corresponds to the *PASS* of the proposal. Otherwise, the MAJ will make a *FALSE* decision, which corresponds to the *DENY* of the proposal. Based on the four-way junction structure and the fluorescence modulating ability of G4 toward PPIX (another kind of porphyrin dye), our group fabricated a label-free and enzyme-free 3-input DNA voter (Figure 12.3b) [19]. Afterward, the first DNA voter with "one-vote veto" function was also achieved on the basis of DNA inputs modulated split-G4 and its peroxidase-like G4 DNAzyme with catalytic ability toward colorimetric substrate TMB (3,3′,5,5′-tetramethylbenzidine) [20]. It could present visual/fluorescent outputs simultaneously, and complicated cascade DNA circuits (OR/INHIBIT) were further designed.

Compared with above sophisticated logic devices where the input sequence has almost no effect on the logical functions, the performance of sequential logic circuits depends not only on the composition of inputs but also their sequence [21]. One of the most famous devices is DNA keypad locks, as they can work as security devices to protect molecular-level information from illegal invasion. The keypad lock could only be unlocked by the correct "password"; the input order, otherwise, will keep locked. By utilizing the separation ability
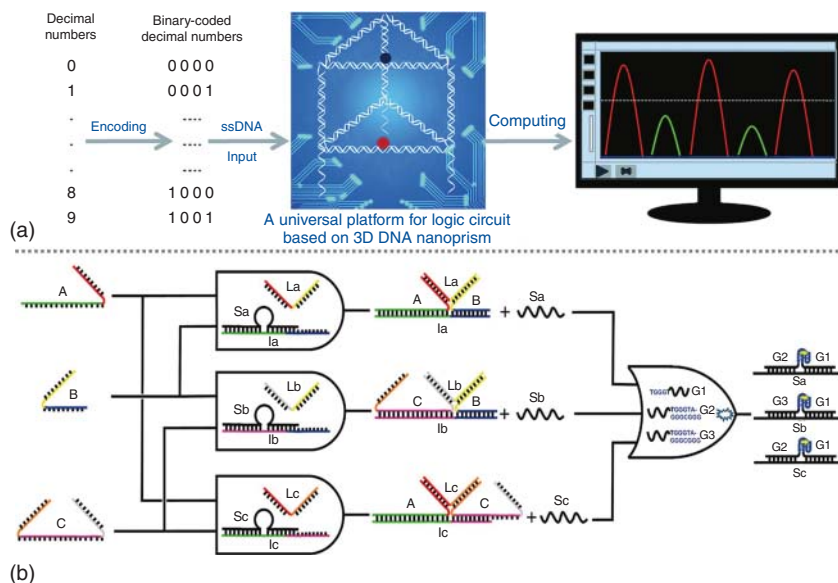
Figure 12.3 (a) The parity checker for identifying even/odd numbers from natural numbers less than 10 by employing the 3D DNA nanoprism platform. Source: From He et al. [18]. © 2015. Reproduced with permission of Royal Society of Chemistry. (b) Four-way-junction-based label-free and enzyme-free 3-input DNA voter. Source: From Zhu et al. [19]. © 2013. Reproduced with permission of American Chemical Society.

of silver microspheres and peroxidase-like property of G4zyme, we constructed the first 5-input DNA keypad lock with visual readout (the color of colorimetric substrate, TMB) (Figure 12.4a) [22]. Only the correct password "a-b-c-d-e" could unlock this keypad lock and generate dark-blue colored solution (the "open" state). If the wrong password was used, the DNA strands could not bind with the silver microspheres and will be washed out; then no complete G4 will form, producing pale-blue colored solution (the "locked" state). This five-digit keypad lock possesses much higher security level compared with widely reported three-digit ones.

12.3.4 Parity Generator/Checker (pG/pC) for Error Detection During Data Transmission

Apart from above illustrated advanced DNA logic devices with specific functions, another important logic device is the pG/pC for error detection through data transmission. During the transmission of any type of binary data, the appearance of bit errors is an inevitable problem. These errors that have serious effects on the normal logic computing can be detected via placing a pG at the sending terminal and a pC at the receiving terminal [23]. For the operating principle of the 3-bit pG/pC system, an even pG can generate an extra parity bit P and add it to the original binary bit, D1D2, making the total number of 1's (Σ) in the D1D2P string even. The 2-bit pG will assign a binary value to P (output of pG) according to the
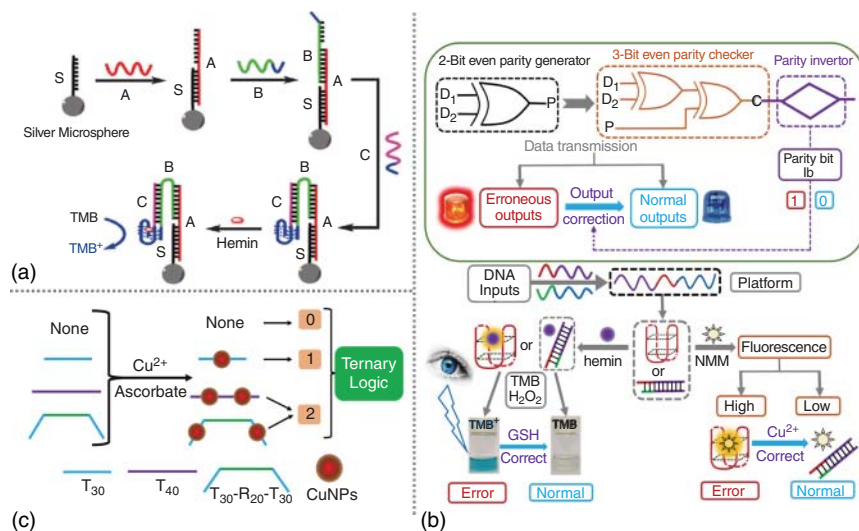
Figure 12.4 (a) The five-digit DNA keypad lock based on silver microspheres and catalytic property of DNA G4zyme. Source: From Zhu et al. [22]. © 2013. Reproduced with permission of Royal Society of Chemistry. (b) DNA-based pG/pC system for error detection through data transmission with fluorescent and visual readouts and "output-correction" function. Source: From Fan et al. [24]. © 2017. Reproduced with permission of Royal Society of Chemistry. (c) Ternary logic system based on poly-T templated fluorescent copper nanoparticles. Source: From Fan et al. [27]. © 2017. Reproduced with permission of Springer Nature.

truth table of an XOR logic gate (Figure 12.4b). Afterward, the D1D2P string will be transmitted to a 3-bit even pC and checked by it. If the transmitted bits in the D1D2P string were changed by unavoidable disturbances through data transmission (generation of bit errors), the number of 1's in the received wrong string will be changed to odd [23], and the pC will show an "alarm" signal (red light in Figure 12.4b), producing the output C = 1. However, if the transmission is normal, the D1D2P string will keep unchanged, and the number of 1's is still even. Then the pC will exhibit a "normal" signal (blue light in Figure 12.4b), producing the output C = 0.

Our group designed the first DNA-based pG/pC system with "output-correction" function by utilizing DNA inputs modulated split-G4/NMM and G4 DNAzyme as fluorescent and visual signal reporters, respectively [24]. Besides, an "output-correction" function was introduced into the pC for the first time, in which all the erroneous outputs could be corrected into normal states, which ensured the regular performance of downstream logic devices. On the basis of this work, a simple, enzyme-free, and G4-free pG/pC system relying on the fluorescent quenching ability of polydopamine nanosphere toward FAM-labeled DNA [25] and an electrochemical one based on "aptamer-nanoclaw"-modulated protein steric hindrance [26] were further successfully fabricated. These multifunctional pG/pC systems that could exhibit different kinds of output signals greatly fulfilled the computing requirements under diversified circumstances.

12.3.5 Non-Boolean Ternary Logic Gates

Till now, all of the above logic circuits are operated according to Boolean logic, in which each input/output has only two states (0 or 1). However, as for the limited computing capacity, the binary Boolean logic is facing challenges from multi-valued logic gates during the processing of high-density or unprecise molecular information. Taking the ternary OR and INHIBIT logic gates as model devices, we fabricated a simple, fast, label-free, and nanoquencher-free system for operating multivalued DNA logic gates using poly-thymine (T) templated copper nanoparticles (CuNPs) (red emission) as signal reporter (Figure 12.4c) [27]. The mixture of Cu^{2+} and ascorbic acid (AA) was used as the universal platform. Different poly-T strands and smartly designed polyadenine (A) strands were selectively taken as ternary inputs to produce the ternary output states (low/0, medium/1, high/2). All of the above ternary gates can be finished within 20 minutes as for the fast formation of CuNPs. This work provided a vivid prototype for multivalued logic gates and also opened novel horizons for corresponding intelligent biological applications.

12.4 Concatenated Logic Circuits

Different from the basic logic gates and functional logic devices that could only execute limited or specific logical functions, the concatenated logic circuits that achieved via the integration/combination of basic gates and advanced devices could execute more sophisticated computing functions that cannot be realized by any single element inside alone [28]. The concatenation of basic gates and advanced devices should abide by the Boolean logic principles initially; then the output of upstream logic gate/device should be one of the inputs of downstream gates/devices. For example, Chen et al. designed interesting concatenated logic circuits based on a DNA three-way junction (Figure 12.5a) [29]. It could work as a keypad lock with visual output (color of TMB) and an automatic reset function. A trigger strand was modified on magnetic beads via streptavidin–biotin interaction. Through toehold mediated strand displacement (TMSD) reaction, the trigger strand could hybridize with the hairpin HA and expose the second toehold. After sequential reaction, a DNA three-way junction with three parts of G4 DNAzyme can be obtained after magnetic separation. This phenomenon matches the concatenated logic circuit "AND-AND-AND." Similarly, our group constructed a label-free and enzyme-free platform for operating fluorescent concatenated "AND-OR-YES" logic circuit based on G-quadruplex/PPIX interaction and DNA TMSD reaction (Figure 12.5b) [30]. Apart from the visual fluorescent outputs, concatenated DNA circuits with electrochemical outputs were also reported. Feng et al. designed various multilevel electrochemical logic circuits based on aptamer–target interaction triggered by electrochemical rectification and applied them to logical detection of ATP/ADA biomarkers (Figure 12.5c) [31].
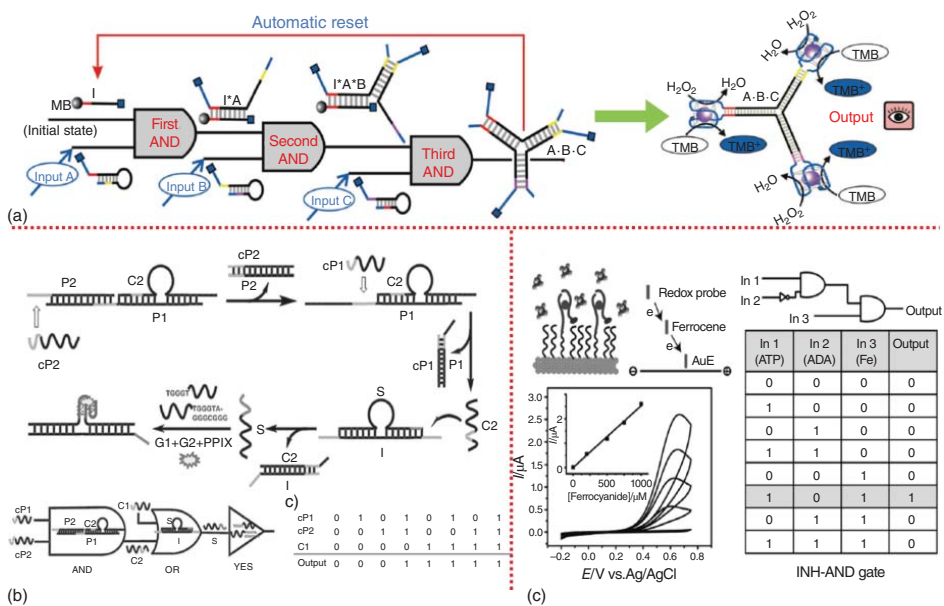
Figure 12.5 Concatenated DNA logic circuits with (a) visual. Source: From Chen et al. [29]. © 2014. Reproduced with permission of John Wiley & Sons. (b) fluorescent. Source: From Zhu et al. [30]. © 2013. Reproduced with permission of John Wiley & Sons. and (c) electrochemical outputs. Source: From Feng et al. [31]. © 2015. Reproduced with permission of John Wiley & Sons.

12.5   Innovative Multifunctional DNA Logic Library

Although various DNA logic gates and advanced logic devices were constructed based on many different systems, only limited logic devices can be achieved in a fragmentary way. For further advancements of this field, exploring innovative versatile DNA logic library in which multifarious basic logic gates, advanced devices, and even concatenated circuits can be alternatively operated on a universal system is one of the brightest directions and will definitely present obvious advantages.

For instance, Yao et al. integrated G4/NMM and DNA-AgNC together as a universal system and constructed series of DNA logic circuits, including encoder/decoder, multiplexer/demultiplexer, ternary gates, parity checker, and so on [32]. Our group for the first time proposed the concept of "contrary logic pairs" to systematically classify DNA gates with opposite functions into "positive + negative" gates (CLP = Pos + Neg) (Figure 12.6a). By utilizing two fluorescent substrates (Amplex Red, Scopoletin) of G4 DNAzyme as label-free signal reporters, we fabricated the first intelligent universal system that yields double results with half the effort for engineering a DNA CLP library and various DNA combinatorial logic circuits [33]. This work greatly simplified the operation and reduced the time/costs of current DNA gates' operation by at least half and also brought significantly improved computing efficiency.

The integration of nanomaterials and DNA computing will exhibit magical power in pushing forward the advancements of this field. Based on the peroxidase-like property of G4zyme and the luminescence quenching ability of oxidized TMB (OxTMB) towards the RGB (red, green, blue) colored upconversion nanoparticles, we operated a multicolor upconversion-chameleon-driven DNA logic library, and corresponding bioanalysis application was also proposed (Figure 12.6b) [34]. Subsequently, inspired by the ancient Roman mythical God Janus and relying on the above similar reaction mechanism, another versatile "DNA Janus logic pair" library was further reported recently [35]. It should be noted that, by using the cascade circuit "YES-INH-1-2 decoder" as the "computing core," we designed an "antioxidant indicator" with ratiometric responses that could distinguish the presence of antioxidants logically and intelligently (output changed from "10" to "01"), which provided a typical prototype for potential intelligent bio-applications.

12.6   Intelligent Bio-applications

It has been widely recognized that DNA computing could bring smart solutions to multifunctional biological applications by integrating the stringency of Boolean logic and biocompatibility of DNA and other biomaterials. Considering that many groups have pioneered excellent *in vivo* applications of DNA computing, we herein just demonstrated several typical *in vitro* bio-applications in this chapter.

Many researchers have proposed that DNA computing can be integrated with enzymes, proteins, and other biomaterials to achieve innovative functions.
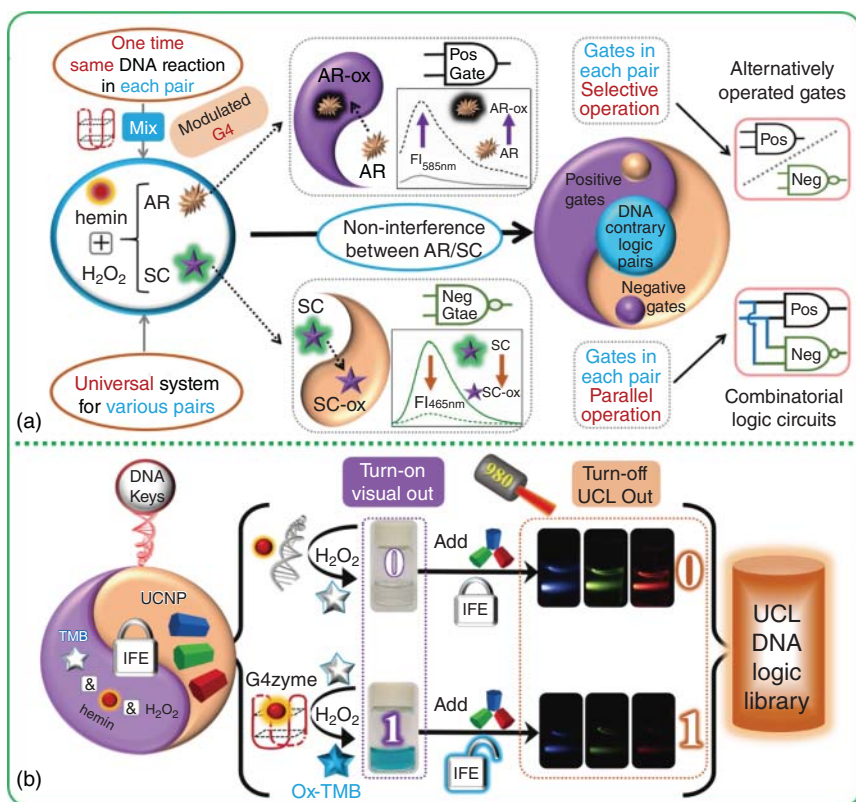
Figure 12.6 (a) Schematic illustration of the DNA "contrary logic pairs" library. Source: From Fan et al. [33]. © 2017. Reproduced with permission of Royal Society of Chemistry. (b) Operating principle of the versatile DNA logic library driven by multicolor upconversion chameleon. Source: From Fan et al. [34]. © 2019. Reproduced with permission of Royal Society of Chemistry.

Merkx's group used bivalent peptide–DNA conjugates as generic, non-covalent, and easily applicable molecular locks to realize OR/AND logic-controlled antibody activity (Figure 12.7A) [36]. By utilizing yeast as cellular model system, the reversible switch of antibody targeting is achieved with only nM-level DNA–peptide locks and strand displacement reaction. Furthermore, aptamer-mediated control of antibody reactivation was also illustrated. Deiters's group operated AND, OR, and NOR logic gates by taking advantage of zinc finger proteins, in which these gates were triggered by short oligonucleotide inputs and could lead to the activation or deactivation of a split luciferase enzyme (Figure 12.7B) [37]. Moreover, microRNAs, which act as potential cellular cancer markers, were used as inputs to perform logic-programmed protein activation.

Apart from peptides and proteins, Katz's group constructed a universal interface to bridge DNA computing and enzymatic computing together and operated various concatenated logic circuits (Figure 12.7C) [38]. After upstream enzymatic
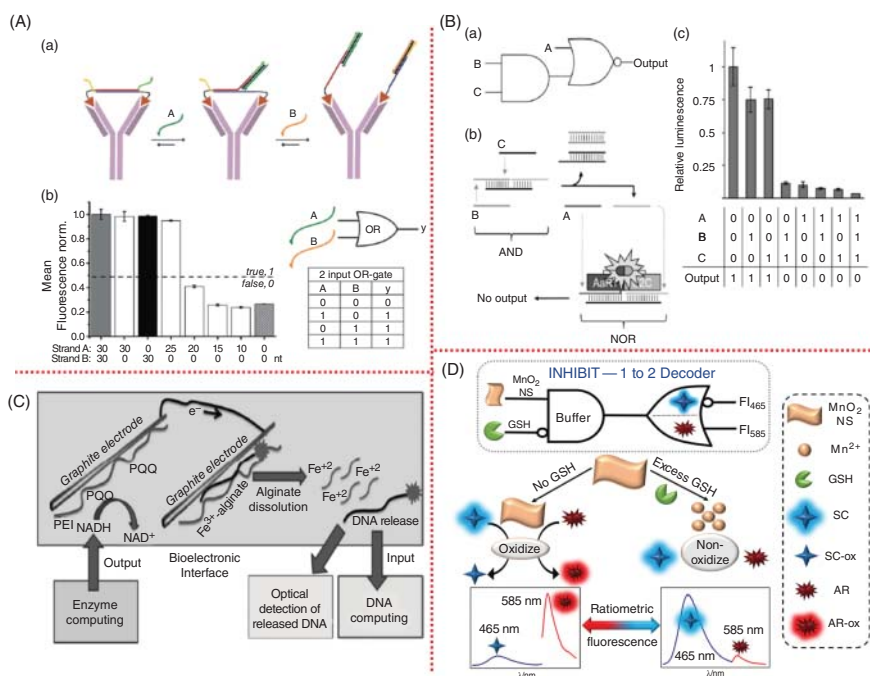
Figure 12.7 The integration of DNA computing with (A) peptides. (a) Equivalent logic circuit of the OR logic gate. (b) Fluorescent outputs of the 2-input OR gate and corresponding truth table. Source: From Janssen et al. [36]. © 2014. Reproduced with the permission of John Wiley & Sons. (B) proteins. (a) Equivalent logic circuit of the AND-NOR cascade gate. (b) Detailed scheme for operating the AND-NOR gate. (c) Relative luminescent outputs under different input variations. Source: From Prokup and Deiters [37]. © 2014. Reproduced with the permission of John Wiley & Sons. and (C) nature enzymes. Source: From Mailloux et al. [38]. © 2015. Reproduced with the permission of John Wiley & Sons. (D) The combination of molecular computing with nanozymes for intelligent bioanalysis of GSH. Source: From Fan et al. [39]. © 2017. Reproduced with permission of American Chemical Society.

computing, the generated NADH will be oxidized to NAD^+ on one electrode and reduce Fe^{3+} to Fe^{2+} on another electrode. Then the Fe^{3+}-cross-linked alginate polymer will dissolve and release the DNA outputs, which will initiate the downstream DNA computing system. The AND-AND-AND and OR-AND circuits were successfully realized, which also had the potential to probe the activity of different enzymes.

Finally, another work we would like to highlight here is the nanozyme-programmed molecular logic system for smart GSH detection (Figure 12.7D) [39]. Although there were no DNAs in this work, the combination of molecular logic and nanozyme will definitely inspire various interesting and useful logic-controlled bio-applications. Nanozymes are nanomaterials that possess different enzyme-mimicking properties; they have attracted widespread attentions from various bioanalytical and biomedical areas. In our work, the oxidase-like property of MnO_2 nanosheet (MnO_2 NS) was applied to fluorescent substrates for the first time. We found that MnO_2 NS could not only

largely quench the fluorescence of highly fluorescent Scopoletin (SC) but also surprisingly enhance that of nonfluorescent Amplex Red (AR) via oxidation reaction. If MnO_2 NS is premixed with GSH, it will be reduced to Mn^{2+} and lose the oxidase-like property, accompanied by subsequent increase in SC's fluorescence and decrease in AR's. Based on this phenomenon, a cascade logic circuit (INH-1-2 decoder) that could logically recognize the presence/absence of GSH before detection was fabricated. This cascade logic circuit endowed biosensing with the intelligence/stringency of Boolean logic and also opened up novel horizons for powerful logical biosensors based on nanozymes.

12.7 Prospects

To summarize, in this chapter, we demonstrated the tutorial logical principle and typical classification of different DNA logic circuits. After that, various representative examples about advanced arithmetic/non-arithmetic DNA logic devices and concatenated circuits constructed by our group and other ones over the past decade were presented. For the bottlenecks of this field, (i) the solution-based DNA reactions and derived limited computing speed will be one of the long-lasting problems, (ii) and most current bio-applications of DNA computing focus only on *in vitro* ones, such as bioanalysis and target recognition, and corresponding *in vivo* applications remain challenging.

For the future advancements of DNA computing, there might be two main directions from our point of view. One direction is the integration of already realized DNA logic systems with the semiconductor transistors. However, it is still difficult for DNA computing to compete with already mature semiconductor computers. The integration of them will combine the advantages of both fields to exhibit surprising power. Recently reported silicon-deposition technique on DNA origami by Fan's group [40] will be a promising and powerful bridge for above integration. Another direction is the smart *in vivo* bio-applications of DNA computing to nanozyme catalysis, genome-editing technique, intracellular imaging, drug load/release, and promising areas. Through reasonable design, these combinations will definitely pave the way for real smart disease diagnostics and intracellular therapy.

Acknowledgment

References

1 de Silva, P.A. and Uchiyama, S. (2007). *Nat. Nanotechnol.* 2: 399–410.
2 Shapiro, E. and Gil, B. (2007). *Nat. Nanotechnol.* 2: 84–85.

3 Andréasson, J. and Pischel, U. (2010). *Chem. Soc. Rev.* 39: 174–188.

4 de Silva, P.A., Gunaratne, N.H.Q., and McCoy, C.P. (1993). *Nature* 364: 42–44.

5 Stojanovic, M.N. and Stefanovic, D. (2003). *Nat. Biotechnol.* 21: 1069–1074.

6 Adleman, L. (1994). *Science* 266: 1021–1024.

7 Qian, L. and Winfree, E. (2011). *Science* 332: 1196–1201.

8 Seelig, G., Soloveichik, D., Zhang, D.Y., and Winfree, E. (2006). *Science* 314: 1585–1588.

9 Willner, I., Shlyahovsky, B., Zayats, M., and Willner, B. (2008). *Chem. Soc. Rev.* 37: 1153–1165.

10 Mailloux, S., Guz, N., Zakharchenko, A. et al. (2014). *J. Phys. Chem. B* 118: 6775–6784.

11 Stojanović, M.N. and Stefanović, D. (2003). *J. Am. Chem. Soc.* 125: 6673–6676.

12 Orbach, R., Wang, F., Lioubashevski, O. et al. (2014). *Chem. Sci.* 5: 3381–3387.

13 Wang, K., Ren, J., Fan, D. et al. (2014). *Chem. Commun.* 50: 14390–14393.

14 Li, H., Guo, S., Liu, Q. et al. (2015). *Adv. Sci.* 2: 1500054.

15 Fan, D., Zhu, J., Liu, Y. et al. (2016). *Nanoscale* 8: 3834–3840.

16 Orbach, R., Remacle, F., Levine, R., and Willner, I. (2014). *Chem. Sci.* 5: 1074–1081.

17 Wu, C., Wang, K., Fan, D. et al. (2015). *Chem. Commun.* 51: 15940–15943.

18 He, K., Li, Y., Xiang, B. et al. (2015). *Chem. Sci.* 6: 3556–3564.

19 Zhu, J., Zhang, L., Dong, S., and Wang, E. (2013). *ACS Nano* 7: 10211–10217.

20 Fan, D., Wang, K., Zhu, J. et al. (2015). *Chem. Sci.* 6: 1973–1978.

21 Jiang, X.J. and Ng, D.K. (2014). *Angew. Chem. Int. Ed.* 53: 10481–10484.

22 Zhu, J., Yang, X., Zhang, L. et al. (2013). *Chem. Commun.* 49: 5459–5461.

23 Bälter, M., Li, S., Nilsson, J.R. et al. (2013). *J. Am. Chem. Soc.* 135: 10230–10233.

24 Fan, D., Wang, E., and Dong, S. (2017). *Chem. Sci.* 8: 1888–1895.

25 Fan, D., Wang, E., and Dong, S. (2017). *ACS Appl. Mater. Interfaces* 9: 1322–1330.

26 Fan, D., Fan, Y., Wang, E., and Dong, S. (2018). *Chem. Sci.* 9: 6981–6987.

27 Fan, D., Wang, E., and Dong, S. (2017). *Nano Res.* 10: 2560–2569.

28 Andréasson, J. and Pischel, U. (2015). *Chem. Soc. Rev.* 44: 1053–1069.

29 Chen, J., Zhou, S., and Wen, J. (2015). *Angew. Chem. Int. Ed.* 127: 456–460.

30 Zhu, J., Zhang, L., Li, T. et al. (2013). *Adv. Mater.* 25: 2440–2444.

31 Feng, L., Lyu, Z., Offenhäusser, A., and Mayer, D. (2015). *Angew. Chem. Int. Ed.* 54: 7693–7697.

32 Gao, R.-R., Yao, T.-M., Lv, X.-Y. et al. (2017). *Chem. Sci.* 8: 4211–4222.

33 Fan, D., Wang, E., and Dong, S. (2017). *Mater. Horiz.* 4: 924–931.

34 Fan, D., Wang, E., and Dong, S. (2019). *Mater. Horiz.* 6: 375–384.

35 Fan, D., Wang, J., Wang, E., and Dong, S. (2019). *Chem. Sci.* 10: 7290–7298.

36 Janssen, B.M.G., van Rosmalen, M., van Beek, L., and Merkx, M. (2015). *Angew. Chem. Int. Ed.* 54: 2530–2533.

37 Prokup, A. and Deiters, A. (2014). *Angew. Chem. Int. Ed.* 53: 13192–13195.

38 Mailloux, S., Gerasimova, Y.V., Guz, N. et al. (2015). *Angew. Chem. Int. Ed.* 54: 6562–6566.

39 Fan, D., Shang, C., Gu, W. et al. (2017). *ACS Appl. Mater. Interfaces* 9: 25870–25877.

40 Liu, X., Zhang, F., Xinxin, Jing et al. (2018). *Nature* 559: 593–598.

13

Nucleic Acid-Based Computing in Living Cells Using Strand Displacement Processes

Lukas Oesinghaus and Friedrich C. Simmel

TU Munich, Physics Department, Am Coulombwall 4a, Garching 85748, Germany

13.1 Nucleic Acid Strand Displacement

Over the past two decades, nucleic acid strand displacement processes have become one of the most widely used dynamic processes in nucleic acid nanotechnology [1, 2]. Toehold-mediated strand displacement enables isothermal switching of DNA- or RNA-based devices between different conformations and has thus become the basis of a plethora of molecular machines, sensors, and computing devices. Over the past years, researchers have begun to explore whether this extremely useful concept can be also applied inside of living cells – predominantly with the task to interrogate or interfere with genetic circuits and to enable molecular information processing *in vivo*. In the present chapter – after a brief introduction to the basic properties and terminology of strand displacement processes – we will provide a brief review on these *in vivo* computing applications.

13.1.1 Basics

In dynamic DNA nanotechnology one is often confronted with the task of removing a DNA strand from a given DNA construct in order to either induce a mechanical change (converting a rigid double helical part of the construct into a flexible single strand) or to replace it with another DNA molecule. One of the ways to induce a transition from duplex to single-stranded DNA is heating, which results in a "melting" of the duplex above a certain melting temperature determined by the duplex stability (which in turn is dictated by the length and sequence of the duplex). Heating, however, is not regarded as the most elegant way to switch molecular devices – it is unspecific, typically slow, and usually not an option in the context of thermally unstable components or even *in vivo*.

Fortunately, it is also possible to replace one strand of a DNA duplex by another of the same sequence in an isothermal process called "branch migration." DNA duplexes are dynamic molecular structures – driven by thermal fluctuations, the most unstable base pairs at the ends of a DNA duplex (which, in contrast to the other base pairs, are only stabilized via base stacking with one neighbor) can

temporarily break in a process termed "fraying," which exposes some of the terminal bases in an unpaired form. Another DNA strand of the same sequence ("the invader") may now attach to the exposed base and then compete with the bound ("incumbent") strand for binding to its complement. In a random walk process termed "branch migration," the site at which invader and incumbent meet then diffuses along the duplex until either of the strands has completely displaced the other. Due to the relatively low probability to initiate and successfully complete such an invasion process, strand displacement is relatively slow, however.

Now one of the most widely used tricks in dynamic DNA nanotechnology is the utilization of a "toehold" for strand invasion (Figure 13.1a, b), which speeds up the strand displacement processes considerably. In this case, one of the strands of the duplex (the "gate" strand) to be invaded is extended by a short single-stranded section (the "toehold") to which the invader can attach. This has the advantage that the branch migration initiation frequency increases and, further, that the invader/gate duplex is thermodynamically more stable than the incumbent/gate duplex (simply because it has more base pairs).

Both experimental and theoretical studies have shown that the speedup of strand displacement process compared to "blunt end" strand invasion becomes exponentially faster with the toehold length but saturates above a certain threshold length (around eight nucleotides (nt) for DNA *in vitro* – but longer for RNA *in vivo*). The reverse reaction – incumbent replacing the invader – can be up to 6 orders of magnitude slower than the desired forward direction. Such leak reactions caused by fraying can be further reduced by "clamping" the ends of duplexes with GC pairs.

While the toehold sequence influences the speed of strand displacement initiation, the influence of duplex sequence on the kinetics of branch migration is subtle. Secondary structures formed by partly single-stranded invader and incumbent strands may influence each other and slow down branch migration considerably. Of course, mismatches – when present – play an important role as they may isolate different branch migration domains from each other. Surprisingly, studies on branch migration with unstructured nucleotides have resulted in a relatively wide spread of rate constants, whose sequence dependence has not been understood so far.

Nevertheless, computational modeling has improved our quantitative understanding of branch migration processes considerably over the past years. Most notably, strand displacement has been modeled using the coarse-grained molecular dynamics tool "OxDNA," [3] and the program package "Multistrand" provides Gillespie-type stochastic simulations of the kinetics of strand displacement processes [4].

13.1.2 Computing with Strand Displacement Processes

A strand displacement process can be regarded as a computational "primitive" , which links an "input" strand (the invader) to an "output" (the incumbent). In its original form discussed in the previous paragraph, this would not be very useful, but if the input and output are augmented with additional sequence domains, a wide variety of interesting functions can be realized.
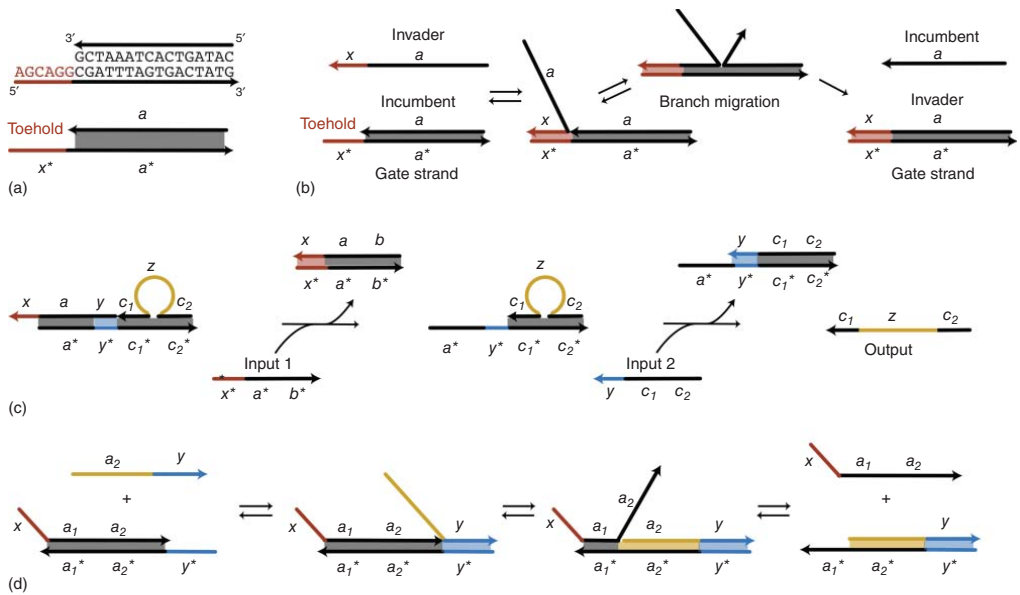
Figure 13.1 Toehold-mediated strand displacement reactions. (a) A DNA duplex with a single-stranded extension at one of its 5′ ends serves as a toehold. Subsequences are abbreviated in domain notation as indicated. (b) In toehold-mediated strand displacement, the invader initiates a three-way branch migration process at the toehold domain *x* as indicated. The process is essentially irreversible and results in the release of the incumbent from the original duplex, leaving an inert invader/gate waste duplex. (c) One of the ways to link strand displacement processes into circuits is to sequester sequence domains into secondary structures, in which they are unable to hybridize with the complements. In this example domain *z* is sequestered in a loop. Addition of input strands 1 and 2 leads to the release of an unstructured output strand, in which *z* is freely accessible. This process can also be interpreted as an AND gate (output = input 1 AND input 2). (d) In toehold exchange reactions, two invader strands compete for binding to a gate strand, to which both are complementary only in part. In the example the initially present toehold y^* is replaced by toehold $a_1{}^*$. The process is reversible and may proceed in both directions.

In one of the first examples by Seelig et al. [5], specific output sequence domains were sequestered by gate strands into inert loop conformations, in which they were unable to hybridize with complementary sequence domains (Figure 13.1c). Toehold-mediated strand displacement was then applied to break the loop conformation, which activated the output sequences for hybridization with downstream components. By making the release of the output dependent on two input sequences, AND logic could be implemented, whereas OR logic is trivially obtained by providing two gates that convert different inputs into the same output.

Briefly after that, Zhang et al. came up with the concept of "entropy-driven" DNA networks [6], in which only the number of nucleic acid strands changed during a reaction, but not the number of base pairs. This work also introduced the useful concept of a "toehold exchange" process. Toehold exchange differs from toehold-mediated strand displacement in that the invader does not completely displace the incumbent, but relies on the spontaneous dissociation of a few remaining base pairs that are not shared by invader and incumbent. This process deactivates the toehold of the invader and "activates" a toehold section on the incumbent, which can be used for wiring.

Another major advance for strand displacement computing was the demonstration of the "seesaw gates" by Qian and Winfree [7], which are based on reversible toehold exchange reactions (Figure 13.1d). With the seesaw concept, it became possible to implement fuel-driven signal amplification and thresholding, which further enabled the realization of large-scale molecular computing networks. Seesaw gates were impressively utilized for large digital logic networks [7] as well as neural network computation for pattern recognition [8, 9].

Strand displacement processes have also been used for analog computing such as in chemical controllers or for the implementation of chemical oscillators. Moreover, researchers have sought to improve the performance and kinetics of strand displacement-based computing processes by arranging their components on origami substrates [10–13].

13.1.3 Computing with Nucleic Acid Strand Displacement *In Vivo*

Until recently, DNA nanotechnology and DNA computing traditionally used DNA only in a nonbiological context, and DNA was merely regarded as a sequence-programmable molecular substrate for the realization of structural or computational functions. However, as nucleic acids of course do have a biological function, it makes sense to consider potential applications of DNA nanostructures and DNA computers in biochemistry and biology.

Nucleic acid-based computers can be naturally interfaced with genetic functions. Importantly, nucleic acids offer the unique possibility to rationally program interactions with their target molecules through the choice of their sequence. Furthermore, applying a negative design strategy [14], it is also possible to reduce undesired interactions and thus improve orthogonality of the components.

Strand displacement processes in the cellular context have to obey different boundary conditions depending on the cellular "chassis" they are implemented in, in particular on whether they are to be operated inside of bacteria or eukaryotes.

Bacteria grow and divide relatively fast and RNA degradation proceeds rapidly. Therefore RNA-based circuits in bacteria require constant production of RNA species, and circuit operation has to consider their continuous buildup and degradation. By contrast, eukaryotic cells do not grow and divide as quickly, making dilution less of an issue. For applications in mammalian cells, the circuit components can therefore be delivered from the outside using transfection agents and then operated within an approximately static cellular environment. In addition, chemically stabilized nucleic acid species can be used to increase the time available for strand displacement processes to finish.

In the remainder of the chapter, we will discuss two particularly promising approaches toward *in vivo* computing that utilize strand displacement processes: engineered riboregulators and switchable guide RNAs (gRNAs) for CRISPR (clustered regularly interspaced short palindromic repeats)-associated proteins.

13.2 Synthetic Riboregulators

13.2.1 First-Generation Riboregulators

RNA-based gene regulatory processes are an obvious area of application for strand displacement techniques in synthetic biology. One interesting class of molecules in this context is riboregulators, which have a similar operation principle as naturally occurring riboswitches. Riboswitches acting at the posttranscriptional level are aptamer-based regulatory regions found in the 5′ untranslated region (UTR) of many bacterial mRNA molecules (some are also found in mRNAs of archaea, plants, and fungi), in which the accessibility of the ribosome binding site (RBS) is dependent on the binding of a small molecule metabolite. In the presence of the metabolite, the riboswitch undergoes a conformational change, which leads to either sequestration or release of the RBS (depending on whether this is negative or positive regulation) [15].

Synthetic riboregulators have been created based on a similar scheme, but by making the conformational switch dependent on another RNA molecule instead of a metabolite. Maybe the first example of a riboregulator controlling the expression of green fluorescent protein (GFP) in *Escherichia coli* cells was provided by Isaacs et al. in 2004 [16]. They designed a "cis-repressing" RNA motif, which inhibited translation of an mRNA into GFP by sequestering the RBS within the stem of a hairpin structure placed in the 5′ UTR. Translation could be activated by the expression of a "trans-acting" RNA molecule, which was capable of binding to the hairpin of the cis-repressing RNA, breaking the secondary structure and thus releasing the RBS.

For the best riboregulators, this scheme resulted in a ≈20-fold change in gene expression upon activation. While not explicitly mentioned by Isaacs et al., the riboregulator design utilized the loop region of the cis-repressing RNA as an internal toehold for nucleation of the cis-repressing/trans-acting RNA interaction and thus was, in principle, also based on a strand invasion mechanism.

The design of the original riboregulator switches was guided by thermodynamic calculations using the mfold prediction tool. While such calculations are

extremely important for the design process, one has to consider that intracellular conditions differ from the idealized experimental conditions under which the thermodynamic data underlying the predictions were generated. An interesting *in vivo* SHAPE-Seq study of the riboregulator by Lucks and coworkers demonstrated, however, that the riboregulator switches essentially folded and performed *in vivo* as designed, with only a few differences between predicted and actually realized base pairing patterns [17].

13.2.2 Toehold Switch Riboregulators

The riboregulator design of the Isaacs paper was reengineered in 2014 by explicitly introducing an external toehold for strand invasion by a trigger RNA into the regulatory hairpin [18]. The mechanism of this "toehold switch" riboregulator is depicted in Figure 13.2. In its design, the RBS is sequestered in the hairpin loop of the riboregulator, while the start codon for translation (AUG) is placed into a bulge loop in the stem. As a result, the remaining sequence of the hairpin stem-loop can be freely chosen (i.e., it does not contain an anti-RBS sequence). In addition, the hairpin is extended at its 5′ end with a 12 nt toehold sequence. In the folded state the riboregulator is translationally inactive. Upon addition of a trigger RNA molecule, which is complementary to the toehold and part of the stem sequence, the hairpin structure is broken by strand invasion, exposing the RBS and AUG sequences. Now the ribosome can assemble on the mRNA molecule, scan for the start codon, and translate the coding region into a protein. In a first round of rational design, a set of 168 toehold switches was investigated that exhibited ON/OFF gene expression ratios of up to 300. Thermodynamic analysis of the structures resulted in a number of thermodynamic "predictors," based on which the toehold switches could be further improved in a "forward-design" step – resulting in 13 riboregulators with one exhibiting an ON/OFF ratio of more than 600.

Instead of synthetic trigger molecules, the authors were also able to use naturally occurring small RNAs as inputs and also mRNA molecules transcribed from a high copy number plasmid. For these natural RNA triggers, secondary structure had to be taken into account, and this also required an extension of the toehold length to more than 24 nt. In contrast to the synthetically triggered switches, activation by natural RNAs resulted in reduced ON/OFF ratios between ≈10 and 50.

The potential for mRNA detection by toehold switches was immediately applied by Pardee et al. for the realization of paper-based biosensors [19]. To this end, cell-free transcription–translation mix containing toehold switches coding for fluorescent proteins, and responding to a desired target RNA, was freeze-dried on a paper strip. Upon rehydration and addition of trigger RNA molecules, a fluorescent readout was generated on the paper. It was shown that this approach could be used for the specific detection of disease-related mRNAs – for instance, for the detection and distinction of mRNA from two different strains of Ebola virus.

While the original toehold switch only responded to single trigger RNAs, Green et al. later demonstrated that toehold switches could also be engineered
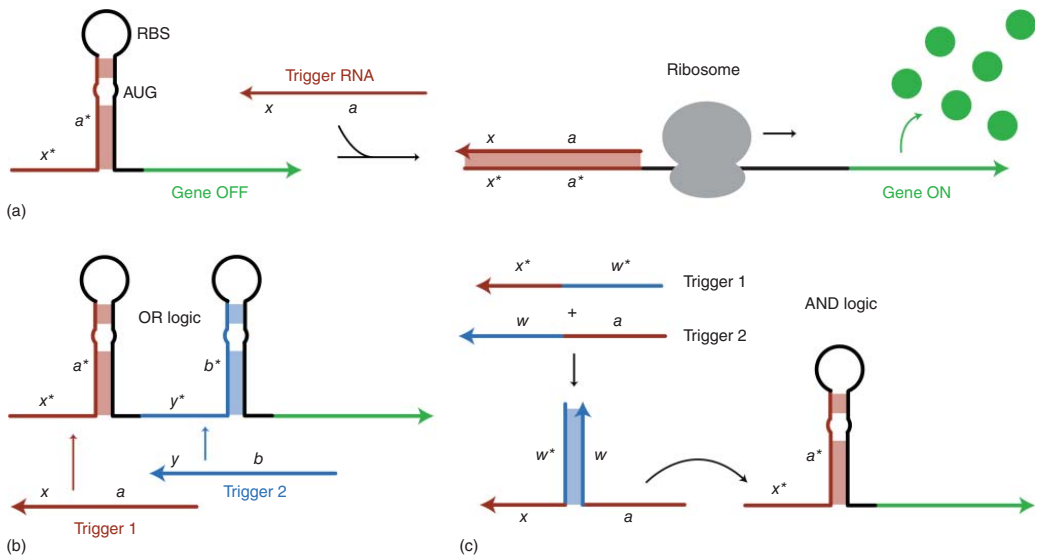
Figure 13.2 Toehold switch riboregulators [18] and input logic [20]. (a) Toehold switch riboregulators are placed as synthetic regulatory elements into the 5′ untranslated region of a (bacterial) mRNA. The ribosome binding site is sequestered in a hairpin loop and thus inaccessible to the ribosome. The sequence domains x* (the toehold) and a* can be freely chosen. An appropriate trigger RNA with sequence a–x can open the hairpin via toehold-mediated strand invasion and thus activate translation of the mRNA by the ribosome. (b) OR logic can be implemented by simply placing two toehold switch elements upstream of the coding region. Either trigger 1 or trigger 2 (or both) can release a ribosome binding site and thus switch on gene expression. (c) AND logic relies on a split trigger, which becomes capable of toehold-mediated strand displacement only after hybridization of the two trigger components 1 and 2 (which brings together the required sequence domains x and a).

for more complex RNA-based input logic (Figure 13.2b, c), making protein expression dependent on the presence of more than one trigger RNA [20]. For instance, AND logic can be realized by utilizing composite trigger strands, which only become active when hybridized together. In this approach, the toehold section of the trigger and the invading sequence are initially separated – a toehold sequence alone can only bind to the complementary toehold on the riboregulator, but cannot perform strand invasion by branch migration. On the other hand, the invading sequence without toehold cannot efficiently initiate branch migration.

OR logic can be implemented by putting several toehold switches responding to different trigger sequences in series. When any of the regulatory hairpins upstream of the start codon is opened, the ribosome will bind to the corresponding RBS and then plunge through the secondary structure constituted by the other toehold switches. Together with logical negation – utilizing antisense trigger molecules – general logical expressions in disjunctive normal form (DNF) can be realized.

Toehold switches can also be designed to act as translational repressors rather than activators [21]. In this case, the toehold switching principle is simply reversed, and the RBS is made initially accessible to the ribosome. Toehold-mediated strand invasion by a trigger RNA leads to a reorganization of the riboregulator, which sequesters the RBS in a stable secondary structure and thus represses translation.

13.2.3 Other Transcriptional and Translational Regulators

A different approach to translational regulation in mRNA molecules had been previously demonstrated by Mutalik et al., who adapted the naturally occurring RNA-IN/RNA-OUT system from *E. coli* for the rational design or riboregulators. In *E. coli* the short noncoding RNA-OUT molecule binds to the complementary RNA-IN sequence and therefore regulates the expression level of insertion sequence IS10 both by masking its RBS and by increasing RNA degradation. Interactions between RNA-IN and RNA-OUT are nucleated in the loop region of RNA-OUT, which contains a pyrimidine-uracil-nucleotide-purine (YUNR) motif – corresponding to an "internal" toehold in the hairpin loop. Based on this system, Mutalik et al. designed a library of 23 orthogonal regulator pairs, in which 5 nt in the RNA-OUT recognition loop were mutated [22]. When testing the library experimentally *in vivo*, strong variations in the efficiency of translational repression were observed – ranging from less than 5% to more than 90%. Analysis of the results demonstrated that the YUNR motif was not essential for the performance of the artificial RNA regulators (the YUNR U-turn enforces a sharp bend in phosphate backbone of a stem-loop structure, which exposes the bases following the bend to the solvent, which makes them more accessible for base pairing). In their study, Mutalik et al. found that also other 5 nt loop recognition sequences performed well, potentially because the AU-rich stem close to the loop provided enough flexibility for nucleation of the first base pairs.

Gene expression can also be regulated at the transcription level, e.g. by controlling the efficiency of transcription termination with transcriptional

"attenuators." Based on these, Julius Lucks and coworkers developed a rational approach toward the construction of RNA-based activators termed "small transcription activating RNAs (STARs)" [23]. Transcriptional terminators typically consist of a strong hairpin secondary structure followed by a stretch of unstructured uracil bases. In order to make this process switchable, Lucks et al. added an anti-anti-terminator/anti-terminator hairpin (aat/at-HP) upstream of the terminator. aat/at-HP folds quickly during transcription and therefore does not interfere with the terminator hairpin. STAR molecules are designed to break the aat/at hairpin in a loop-initiated strand invasion process (the hairpin loop acts, in a sense, as a toehold), upon which part of the terminator will be sequestered by the anti-terminator – which diminishes its termination activity and thus allows transcription to proceed.

More recently, Koeppl, Suess and coworkers also demonstrated the simultaneous use of transcriptional and translational regulation by combining the toehold switch and a STAR region within a single mRNA molecule [24]. In this way, gene expression was only active in the presence of the STAR and the toehold switch trigger RNA, comprising an AND gate of these two inputs.

13.3 Combining Strand Displacement and CRISPR Mechanisms

13.3.1 A Brief Introduction to CRISPR

In the past years, life science research has been transformed by the development of new tools for genetic engineering based on CRISPR elements [25, 26]. In particular the CRISPR-Cas9 technology has enabled the precise cutting of double-stranded DNA (dsDNA) molecules at arbitrary sequence locations, which can be freely programmed by the choice of an appropriate "gRNA" (see Figure 13.3a). Cas9 (CRISPR-associated protein 9) is a ∼160 kDa large protein, which comprises two distinct nuclease domains for cleavage of the target and nontarget DNA strand [27]. Cas9 binds to gRNA molecules containing a specific handle sequence, and the sequence contained in the "spacer" section of the gRNA directs the Cas9/gRNA complex to a sequence complementary "protospacer" region on a dsDNA (in the natural CRISPR system, the gRNA is composed of two parts – the crRNA containing the spacer and part of the handle and the tracrRNA containing the second part of the handle and a terminator). In addition to sequence complementarity of the protospacer, a short protospacer adjacent motif (PAM) is required on the 3′ side adjacent to the binding sequence.

Mechanistic studies have shown that the Cas9/gRNA complex first binds at the PAM sequence, where it melts the adjacent dsDNA and then displaces one of the DNA strands by the RNA spacer loaded in the complex (which is also termed "R-loop formation") [28]. Thus, the PAM plays a role somewhat reminiscent of a "toehold" for strand invasion, which in this case is driven by the Cas9 protein.

In typical gene editing applications, Cas9/gRNA is used to cut at specific locations on the genome and thus to create an artificial double-strand break, which can result either in a gene knockout due to indel formation or, less frequently,
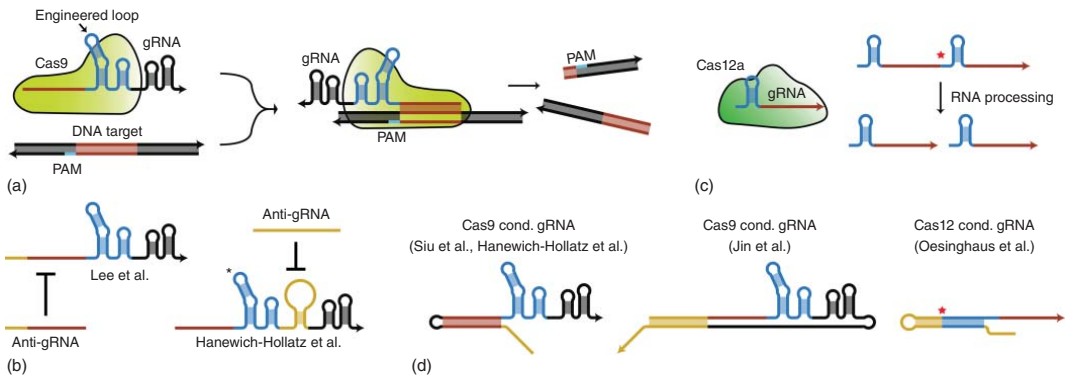
Figure 13.3 Switching guide RNAs using toehold-mediated strand displacement. (a) Cas9 binds to a gRNA that is recognized by its handle sequence (blue). The gRNA is a fusion of the natural crRNA and tracrRNA of Cas9 by an engineered loop. The spacer sequence (red) directs the Cas9/gRNA complex to a complementary DNA target (the protospacer) that needs to be adjacent to a protospacer adjacent motif (PAM) (cyan) to be recognized. Upon binding, Cas9 cleaves the DNA target. (b) Different antisense mechanisms used to inhibit Cas9 gRNAs. Yellow sequences indicate engineered domains that are not present in the regular gRNA. (c) Cas12a functions similarly to Cas9 in regard to DNA target cleavage, although its gRNA sequence is much shorter. Unlike Cas9, it has an RNase activity that cleaves gRNAs at their 5' end upon recognition, a feature that is used for crRNA array processing in the natural system. The cleavage site is indicated by a red star. (d) Designs of different conditional guide RNAs (cond. gRNAs) for Cas9 and Cas12a that are switchable via toehold-mediated strand displacement. Similar colors indicate complementary sequences. Yellow sequences indicate engineered domains that are not present in the regular gRNA.

the insertion of a new gene sequence via homology-directed repair if a suitable template is present. Using a catalytically inactive version of Cas9 ("dead Cas9" or dCas9), CRISPR can also be used for gene regulation rather than gene editing. In prokaryotes, dCas9/gRNA complexes targeted toward genes can either efficiently inhibit transcription initiation or block transcriptional elongation, a technique referred to as clustered regularly interspaced short palindromic repeats interference (CRISPRi) [29]. In eukaryotes, transcriptional inhibition, activation, and base editing can be achieved by fusing different functional domains to dCas9.

As CRISPR mechanisms involve short RNA molecules, namely, the gRNAs, as their central regulatory components, they are also amenable to modulation via strand displacement or strand invasion processes. The goal of such modulation could be the construction of synthetic circuits or to make the activity of gRNAs dependent on the presence of specific endogenous RNA sequences. Apart from Cas9, there is a wide range of other CRISPR-associated proteins that are of interest in this context. Cas12a, for example, is an alternative to Cas9 and functions according to similar principles, though the PAM and gRNA sequences are different. In contrast to Cas9, it processes its own CRISPR arrays using an RNase activity, which is useful for multiplex editing (cf. Figure 13.3b) [30, 31]. Cas13a has an RNA-dependent RNase activity that has been used for biosensor applications [32], and Cas1-Cas2 is an integrase complex that has been used for "storage" of dsDNA and RNA sequences in the genome of prokaryotes [33, 34]. Together, these Cas proteins create many new opportunities for the application of strand displacement techniques in synthetic biology.

Several groups have already demonstrated the use of antisense RNA complementary to the gRNA to inhibit dCas9/gRNA activity. In such applications, CRISPRi is used to suppress the expression of a gene, while anti-gRNA is used to sequester gRNA and promote its degradation (cf. Figure 13.3c). Tae Seok Moon and coworkers designed anti-gRNA molecules augmented by known binding sites for the RNA chaperone Hfq, which promotes interactions between the small RNAs [35]. They could show that anti-gRNA can be used to recover gene expression of initially dCas9/gRNA repressed genes and that several such processes could be operated in parallel. For purely synthetic circuits such as that one, a fixed anti-gRNA sequence is unproblematic. To make this type of antisense mechanism respond to endogenous RNA molecules, however, it is necessary for the gRNA spacer and the anti-gRNA to be sequence independent. Such sequence independence was achieved by Hanewich-Hollatz et al. by letting the anti-gRNA bind an engineered loop inside the gRNA handle rather than the spacer sequence (cf. Figure 13.3c). Notably, this approach was also demonstrated to work in mammalian cells [36].

In related work, Mückl et al. [37] directly used a single-stranded section of the gRNA as a toehold for strand invasion by a complementary anti-gRNA, which disrupted the gRNA handle and thus prevented dCas9 from binding. In this work, the antisense strand invasion concept was used to reversibly switch bacteria into a filamentous state and back to normal growth by first suppressing the expression of the cell division protein FtsZ via CRISPRi and then recovering its expression supported by anti-gRNAs.

Recently, several groups have demonstrated conditional activation of gRNAs in *E. coli* using strand displacement processes for both Cas9 and Cas12a. The conditional activation of gRNAs designed by Siu and Chen and Hanewich-Hollatz et al. relies on extensions at their 5′ end that fold back onto the spacer [36, 38]. An adjacent toehold allows for binding of an RNA trigger that binds the extension and releases the target sequence, thereby restoring activity (see Figure 13.3d). In addition to artificial trigger strands, Siu et al. also used both mRNAs and short RNAs as inputs. In this type of conditional gRNA design, the trigger needs to displace a domain that is complementary to the spacer. Therefore, either the trigger sequence or the spacer sequence is constrained, meaning that it would not be possible to, for example, sense a specific mRNA and activate a gRNA for another natural target in turn. Jin et al. extended Cas9 gRNAs at their 3′ end and designed this extension to fold back onto an extension at the 5′ end without pairing the spacer, thereby avoiding fixed sequences on the trigger [39]. This was sufficient to inhibit the activity of Cas9 *in vitro*, though it is unclear if the inhibition using this design will function sufficiently well *in vivo*.

All these approaches leave double-stranded extensions on the gRNA that can decrease the activity of the activated gRNAs. We used the RNA processing activity of Cas12a to create switchable gRNAs with full on-target activity [40]. Cas12a gRNAs were extended at their 5′ end and folded back onto the handle sequence to inhibit Cas12a binding. An RNA trigger restores the handle structure and promotes Cas12a binding. Cas12a then cleaves off the extension, restoring a wild-type gRNA. This approach is also amenable to the creation of multi-input AND gates analogous to what Green et al. demonstrated for toehold switches. Such AND gates are an alternative way of overcoming sequence constraints to allow for the sensing of endogenous mRNAs [40].

The wide range of effector choices enabled by using Cas9/Cas12a (indel formation, homology-directed repair, transcriptional inhibition, transcriptional activation, base editing, epigenetic editing) makes them highly promising for *in vivo* nucleic acid computing, especially also in eukaryotic cells.

13.4 Computing Via Nucleic Acid Strand Displacement in Mammalian Cells

Strand displacement reactions have been successfully employed for the activation and logical control of cellular processes inside of mammalian cells. As briefly mentioned above, in such applications, the required nucleic acid components can be delivered to the cells using transfection agents and do not have to be expressed *in vivo*. Afonin et al. demonstrated an interesting approach based on transfected RNA/DNA hybrids [41, 42] with single-stranded DNA toeholds (Figure 13.4).

Two of such hybrids could bind together via the DNA toeholds, which initiated a four-stranded branch migration process that resulted in the formation of an RNA and a DNA duplex. In one application of this concept, they assembled siRNA molecules *in vivo* (consisting of a 21 bp duplex with 2 nt overhangs at the 3′ ends), which then successfully silenced the expression of a target gene via
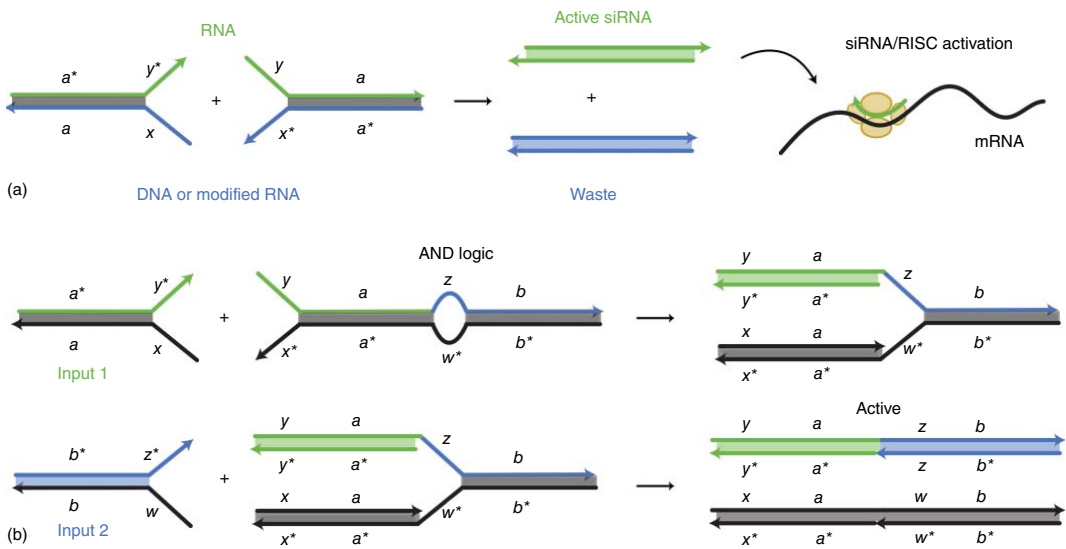
Figure 13.4 Activation of RNA interference via toehold-mediated strand displacement [41, 43]. (a) Hybrid duplexes composed of RNA and either DNA or chemically stabilized RNA hybridize with each other to form an active siRNA duplex and an inactive waste product. The siRNA duplex is further processed, and its guide strand becomes part of an RNA-induced silencing complex (RISC), which has gene silencing activity *in vivo*. (b) Upstream strand displacement reactions can be used to implement AND or OR input logic [43]. In the example shown, two inactive hybrid duplexes (input 1, input 2) have to react with an AND gate complex to generate an active all-RNA duplex that can be further processed (black lines indicate inert, chemically modified nucleic acids). Source: (a) From Afonin et al. [41] and Groves et al. [43]; (b) Based on Groves et al. [43].

RNA interference (RNAi). Gene silencing was only observed when the cells were transfected with both types of RNA/DNA hybrids necessary for formation of the siRNA.

Using a slightly different approach, Groves et al. demonstrated four-way DNA strand exchange reactions between dsDNA species inside of mammalian cells, which could be utilized for the execution of simple logical AND or OR gate operations [43]. They also demonstrated strand exchange between two RNA hybrids, which, similar to the approach by Afonin et al., resulted in siRNA products that led to gene knockdown. Here, the highest efficacy was observed for RNA hybrids containing RNA strands with phosphorothioate bonds and 2-*O*-methylribonucleotides, which increased their stability with respect to nucleases.

13.5 Outlook

13.5.1 Interfacing Nucleic Acid Computing with Synthetic Biology

As shown in the previous paragraphs, toehold-mediated strand displacement – a concept initially developed in the context of dynamic DNA nanotechnology – has been successfully utilized for the control of gene regulatory processes both in bacteria and in mammalian cells. Strand displacement processes are most naturally applied to RNA-based regulatory mechanisms such as riboregulators or riboswitches, CRISPR, or RNAi. A more widespread application of RNA strand displacement *in vivo* is currently still hindered by our lack of control of RNA degradation and RNA hybridization processes and the potential presence of RNA-binding proteins. Whereas for the realization of DNA strand displacement processes *in vitro* typically random sequences without any secondary structure are utilized, the design of dynamic RNA systems *in vivo* thus poses different challenges.

On the one hand, in bacteria, RNA without any secondary structure is degraded rapidly, while on the other hand secondary structure slows down hybridization and strand displacement reactions. Furthermore, undesired interactions with a plethora of other cellular RNA molecules have to be avoided or managed, and also the intracellular localization of RNA molecules has to be considered. We anticipate that specifically for such applications new design rules can be established that will improve intracellular stability and hybridization of artificial RNA molecules and structures, which may also involve the utilization (or, at least, consideration) of RNA–protein interactions. An important development in this context is the utilization of next-generation sequencing methods for the characterization of "RNA accessibility" *in vivo*. For instance, Contreras and coworkers recently developed an RNA-Seq-based method termed INTERFACE that allowed the high-throughput characterization of potential binding sites for small regulatory RNAs (sRNAs) in bacteria [44].

Apart from gaining a better mechanistic understanding to allow for improved rational design approaches, it is also conceivable that *in vivo* dynamic RNA systems could be optimized using directed evolution methods. Molecular evolution

could be applied, e.g. to balance hybridization and degradation pathways as well as to minimize intracellular crosstalk.

Apart from simple riboregulators, so far no attempts to generate and operate RNA-based molecular machines inside of cells have been published. However, based on recent advances in RNA origami [45], it is conceivable that more complex molecular devices can be generated by gene expression, which could further be actuated by RNA strand displacement or related processes.

As also mentioned above, implementation of dynamic processes in fast-growing bacteria poses challenges such as a rapid dilution of components in the cell and the complex superimposition of bacterial growth effects with the dynamics of the artificial system.

Slow-growing bacteria or eukaryotic cells may therefore turn out to be better chassis for dynamic RNA or DNA nanotechnology, as they represent a more constant biochemical background. It is then also possible to "transfect" all the components of a dynamic DNA or RNA system and to utilize chemically stabilized nucleic acids.

In this context, the use of endogenous RNAs as inputs opens up a wide range of possible applications in sensing and therapeutics. Labeling of mRNAs in fixed cells for imaging using catalytic hairpin assembly (a technique for signal amplification using strand displacement) is already well established [46, 47]. Recently, imaging of a synthetic mRNA target in living mammalian cells has been demonstrated using chemically modified DNA strand displacement probes [48]. If this can be extended to natural mRNAs, the high specificity of strand displacement could be used to image several endogenous mRNAs in living cells simultaneously.

Side effects due to off-target activities have been a major hurdle for the introduction of RNAi drugs to the market [49]. By making the activity of transfected siRNA or gRNA dependent on one or even multiple endogenous RNAs, RNAi and CRISPR therapeutics could be made tissue specific, reducing such off-target activity and making them viable for clinical applications. Conditional gRNAs could also be used for more fundamental research. There, transcribed, rather than chemically stabilized, conditional gRNAs could be useful, e.g. for developmental biology research. Continuous mutagenesis of barcodes by Cas9 has previously been used to track cell lineage in zebrafish and mice [50, 51]. Conditional gRNAs could extend this technique to recording not just a cell's lineage but also expression levels of individual endogenous RNAs during development.

For more complicated functionalities including multiple input RNAs and outputs, low and inhomogeneous concentrations of the multiple transfected components required for a traditional strand displacement circuit implementation are likely to pose a significant problem. This would be especially acute in a therapeutic context where the maximum amount of usable material is highly limited. By combining nucleic acid nanotechnology with nucleic acid computing principles, it might be possible to integrate multiple sensing and effector modules on a scaffold, thereby ultimately creating nucleic acid "robots" that evaluate a cell's state and execute a complex program in response.

References

1 Zhang, D.Y. and Seelig, G. (2011). Dynamic DNA nanotechnology using strand-displacement reactions. *Nat. Chem.* 3: 103–113.

2 Simmel, F.C., Yurke, B., and Singh, H.R. (2019). Principles and applications of nucleic acid strand displacement reactions. *Chem. Rev.* 119: 6326–6369.

3 Sulc, P., Ouldridge, T.E., Romano, F. et al. (2015). Modelling toehold-mediated RNA strand displacement. *Biophys. J.* 108: 1238–1247.

4 Schaeffer, J.M., Thachuk, C., and Winfree, E. (2015). Stochastic simulation of the kinetics of multiple interacting nucleic acid strands. In: *DNA Computing and Molecular Programming (DNA21), Lecture Notes in Computer Science (LNCS)*, vol. 9211, 194–211.

5 Seelig, G., Soloveichik, D., Zhang, D.Y., and Winfree, E. (2006). Enzyme-free nucleic acid logic circuits. *Science* 314: 1585–1588.

6 Zhang, D.Y., Turberfield, A.J., Yurke, B., and Winfree, E. (2007). Engineering entropy-driven reactions and networks catalyzed by DNA. *Science (New York, NY)* 318: 1121–1125.

7 Qian, L. and Winfree, E. (2011). Scaling up digital circuit computation with DNA strand displacement cascades. *Science (New York, NY)* 332: 1196–1201.

8 Qian, L., Winfree, E., and Bruck, J. (2011). Neural network computation with DNA strand displacement cascades. *Nature* 475: 368–372.

9 Cherry, K.M. and Qian, L. (2018). Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature* 559: 1–19.

10 Wickham, S.F.J., Bath, J., Katsuda, Y. et al. (2012). A DNA-based molecular motor that can navigate a network of tracks. *Nat. Nanotechnol.* 7: 169–173.

11 Chatterjee, G., Dalchau, N., Muscat, R.A. et al. (2017). A spatially localized architecture for fast and modular DNA computing. *Nat. Nanotechnol.* 12: 920–927.

12 Thubagere, A.J., Li, W., Johnson, R.F. et al. (2017). A cargo-sorting DNA robot. *Science* 357: eaan6558–eaan6511.

13 Chao, J., Wang, J., Wang, F. et al. (2019). Solving mazes with single-molecule DNA navigators. *Nat. Mater.* 18: 1–10.

14 Dirks, R.M., Lin, M., Winfree, E., and Pierce, N.A. (2004). Paradigms for computational nucleic acid design. *Nucleic Acids Res.* 32: 1392–1403.

15 Mandal, M. and Breaker, R.R. (2004). Gene regulation by riboswitches. *Nat. Rev. Mol. Cell Biol.* 5: 451–463.

16 Isaacs, F.J., Dwyer, D.J., Ding, C. et al. (2004). Engineered riboregulators enable post-transcriptional control of gene expression. *Nat. Biotechnol.* 22: 841–847.

17 Watters, K.E., Abbott, T.R., and Lucks, J.B. (2016). Simultaneous characterization of cellular RNA structure and function with in-cell SHAPE-Seq. *Nucleic Acids Res.* 44: e12.

18 Green, A.A., Silver, P.A., Collins, J.J., and Yin, P. (2014). Toehold switches: de-novo-designed regulators of gene expression. *Cell* 159: 925–939.

19 Pardee, K., Green, A.A., Ferrante, T. et al. (2014). Paper-based synthetic gene networks. *Cell* 159: 940–954.

20 Green, A.A., Kim, J., Ma, D. et al. (2017). Complex cellular logic computation using ribocomputing devices. *Nature* 548: 117–121.

21 Kim, J., Zhou, Y., Carlson, P.D. et al. (2019). De-novo-designed translational repressors for multi-input cellular logic. *biorxivorg*. *Nat. Chem. Biol.* 15: 1173–1182.

22 Mutalik, V.K., Qi, L., Guimaraes, J.C. et al. (2012). Rationally designed families of orthogonal RNA regulators of translation. *Nat. Chem. Biol.* 8: 447–454.

23 Chappell, J., Takahashi, M.K., and Lucks, J.B. (2015). Creating small transcription activating RNAs. *Nat. Chem. Biol.* 11: 214–220.

24 Lehr, F.-X., Hanst, M., Vogel, M. et al. (2019). Cell-free prototyping of AND-logic gates based on heterogeneous RNA activators. *ACS Synth. Biol.* 8: 2163–2173.

25 Makarova, K.S., Haft, D.H., Barrangou, R. et al. (2011). Evolution and classification of the CRISPR-Cas systems. *Nat. Rev. Microbiol.* 9: 467–477.

26 Wiedenheft, B., Sternberg, S.H., and Doudna, J.A. (2012). RNA-guided genetic silencing systems in bacteria and archaea. *Nature* 482: 331–338.

27 Anders, C., Niewoehner, O., Duerst, A., and Jinek, M. (2014). Structural basis of PAM-dependent target DNA recognition by the Cas9 endonuclease. *Nature* 513: 569–573.

28 Szczelkun, M.D., Tikhomirova, M.S., Sinkunas, T. et al. (2014). Direct observation of R-loop formation by single RNA-guided Cas9 and Cascade effector complexes. *Proc. Natl. Acad. Sci. U.S.A.* 111: 9798–9803.

29 Qi Lei, S., Larson, M.H., Gilbert, L.A. et al. (2013). Repurposing CRISPR as an RNA-guided platform for sequence-specific control of gene expression. *Cell* 152: 1173–1183.

30 Zetsche, B., Heidenreich, M., Mohanraju, P. et al. (2017). Multiplex gene editing by CRISPR-Cpf1 using a single crRNA array. *Nat. Biotechnol.* 35: 31–34.

31 Breinig, M., Schweitzer, A.Y., Herianto, A.M. et al. (2019). Multiplexed orthogonal genome editing and transcriptional activation by Cas12a. *Nat. Methods* 16: 51–54.

32 Gootenberg, J.S., Abudayyeh, O.O., Lee, J.W. et al. (2017). Nucleic acid detection with CRISPR-Cas13a/C2c2. *Science* 356: 438–442.

33 Shipman, S.L., Nivala, J., Macklis, J.D., and Church, G.M. (2017). CRISPR-Cas encoding of a digital movie into the genomes of a population of living bacteria. *Nature* 547: 345–349.

34 Schmidt, F., Cherepkova, M.Y., and Platt, R.J. (2018). Transcriptional recording by CRISPR spacer acquisition from RNA. *Nature* 562: 380–385.

35 Lee, Y.J., Hoynes-O'Connor, A., Leong, M.C., and Moon, T.S. (2016). Programmable control of bacterial gene expression with the combined CRISPR and antisense RNA system. *Nucleic Acids Res.* 44: 2462–2473.

36 Hanewich-Hollatz, M.H., Chen, Z., Hochrein, L.M. et al. (2019). Conditional guide RNAs: programmable conditional regulation of CRISPR/Cas function in bacterial and mammalian cells via dynamic RNA nanotechnology. *ACS Cent. Sci.* 5: 1241–1249.

37 Mückl, A., Schwarz-Schilling, M., Fischer, K., and Simmel, F.C. (2018). Filamentation and restoration of normal growth in *Escherichia coli* using a combined CRISPRi sgRNA/antisense RNA approach. *PLoS One* 13: e0198058.

38 Siu, K.-H. and Chen, W. (2019). Riboregulated toehold-gated gRNA for programmable CRISPR-Cas9 function. *Nat. Chem. Biol.* 15: 217–220.

39 Jin, M., Garreau de Loubresse, N., Kim, Y. et al. (2019). Programmable CRISPR-Cas repression, activation, and computation with sequence-independent targets and triggers. *ACS Synth. Biol.* 8: 1583–1589.

40 Oesinghaus, L. and Simmel, F.C. (2019). Switching the activity of Cas12a using guide RNA strand displacement circuits. *Nat. Commun.* 10: 2092.

41 Afonin, K.A., Viard, M., Martins, A.N. et al. (2013). Activation of different split functionalities on re-association of RNA-DNA hybrids. *Nat. Nanotechnol.* 8: 296–304.

42 Afonin, K.A., Desai, R., Viard, M. et al. (2014). Co-transcriptional production of RNA-DNA hybrids for simultaneous release of multiple split functionalities. *Nucleic Acids Res.* 42: 2085–2097.

43 Groves, B., Chen, Y.-J., Zurla, C. et al. (2016). Computing in mammalian cells with nucleic acid strand exchange. *Nat. Nanotechnol.* 11: 287–294.

44 Mihailovic, M.K., Vazquez-Anderson, J., Li, Y. et al. (2018). High-throughput in vivo mapping of RNA accessible interfaces to identify functional sRNA binding sites. *Nat. Commun.* 9: 4084.

45 Geary, C., Rothemund, P.W.K., and Andersen, E.S. (2014). RNA nanostructures. A single-stranded architecture for cotranscriptional folding of RNA nanostructures. *Science* 345: 799–804.

46 Choi, H.M.T., Chang, J.Y., Le Trinh, A. et al. (2010). Programmable in situ amplification for multiplexed imaging of mRNA expression. *Nat. Biotechnol.* 28: 1208–1212.

47 Choi, H.M.T., Schwarzkopf, M., Fornace, M.E. et al. (2018). Third-generation in situ hybridization chain reaction: multiplexed, quantitative, sensitive, versatile, robust. *Development* 145 (12): dev165753 26.

48 Chatterjee, G., Chen, Y.-J., and Seelig, G. (2018). Nucleic acid strand displacement with synthetic mRNA inputs in living mammalian cells. *ACS Synth. Biol.* 7: 2737–2741.

49 Setten, R.L., Rossi, J.J., and Han, S.P. (2019). The current state and future directions of RNAi-based therapeutics. *Nat. Rev. Drug Discovery* 18: 421–446.

50 McKenna, A., Findlay, G.M., Gagnon, J.A. et al. (2016). Whole-organism lineage tracing by combinatorial and cumulative genome editing. *Science* 353: aaf7907.

51 Kalhor, R., Kalhor, K., Mejia, L. et al. (2018). Developmental barcoding of whole mouse via homing CRISPR. *Science* 361 (6405):eaat9804.

14

Strand Displacement in DNA-Based Nanodevices and Logic

Antoine Bader and Scott L. Cockroft

University of Edinburgh, EaStCHEM School of Chemistry, Joseph Black Building, David Brewster Road,
Edinburgh EH9 3FJ, UK

14.1 An Introduction to Strand Displacement Reactions

A major step forward in DNA nanotechnology was achieved with the introduction of the toehold-based DNA strand displacement (DSD) reaction [1]. A recognition domain was added as a dangling end in the target duplex to facilitate the displacement of the incumbent strand (Figure 14.1). Indeed, this toehold region favors the co-localization of the invading strand and the target complex via complementary binding. Subsequent branch migration occurs through a series of reversible base pairing dissociation and association steps. The random walk of the branch migration eventually results in the incumbent strand being released to afford the most thermodynamically stable product.

Thus, several methods have been devised to provide fine control over the kinetics and selectivity of strand displacement reactions. A direct approach involves increasing the toehold binding strength, either by increasing the number of base pairs formed between toehold domains or by increasing the G-C content of the toehold [1, 3]. These approaches enhance toehold binding and thus favor the strand displacement reaction, since the reaction rate grows exponentially with the toehold binding strength. The rate of the bimolecular reaction is highly dependent on the toehold length, with rates ranging from $1\,\mathrm{M}^{-1}\,\mathrm{s}^{-1}$ to $6 \times 10^6\,\mathrm{M}^{-1}\,\mathrm{s}^{-1}$. Experimental toehold lengths typically range between five and eight nucleotides as these present the fastest displacement rates while keeping the length minimal to avoid incorrect triggering or unwanted associations.

14.1.1 External Control of Strand Displacement Reactions

Toehold sequestering also permits regulation of the strand displacement process. Several techniques have been devised to trap the toehold in an inactive state before eventually revealing the site for the strand displacement reaction to occur. Toeholds can be deactivated by hybridization [4, 5] such that they are trapped in secondary structures such as hairpins [6], bulge loops [7], or even DNA triplexes formed via pH-regulated Hoogsteen motifs [8]. Photochemical inputs have been

(a)

(b)
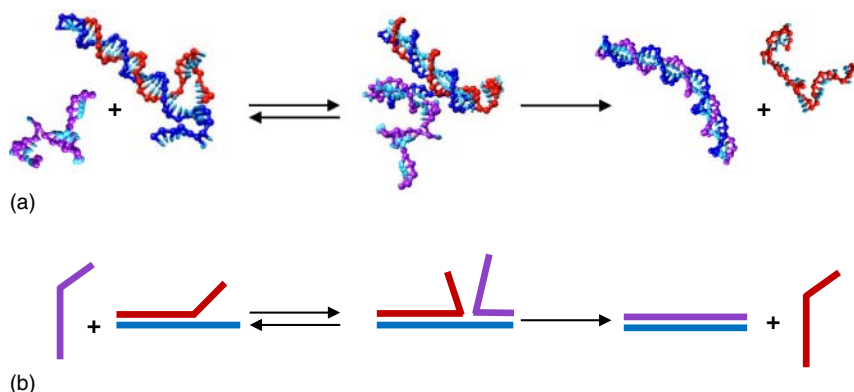
Figure 14.1 A prototypical DNA strand displacement reaction showing one of the ensembles of dynamic "walking" intermediates depicted using (a) OxDNA [2] coarse-grained modeling and (b) in simplified schematic form.

Figure 14.2 Allosteric toehold mechanism. A first input invades the target duplex, revealing a new domain used by the second input in a subsequent strand displacement reaction.

used to activate caged strands that subsequently react in a strand displacement reaction [9–11]. Yang et al. proposed an allosteric toehold mechanism for the regulation of DSD (Figure 14.2) [12]. The allosteric toehold is incorporated into a regulatory strand, thus separating the toehold and branch migration domains, which can be altered independently. The lengths of the allosteric toehold as well as the length of the invading motif can both be adjusted to tune the kinetics of the reaction.

The strand displacement reaction has also been demonstrated to be facilitated by cooperative hybridization of two invading strands. Either of the invading strands are too short to induce a full displacement of the output, but combined displacement originating from opposite toeholds releases the output (Figure 14.3a) [13]. Conversely, a single invading strand can displace multiple outputs from a unique target strand (Figure 14.3b) [5].

Alternatively, a strand capable of simple oligonucleotide replacement can be integrated in a strand displacement reaction through associative toehold activation (Figure 14.4) [14]. This mechanism makes use of a helper strand (orange) that provides a toehold domain that is not hardwired with the branch migration domain, but instead connected to it via three-way junction hybridization, allowing for greater design flexibility. This type of associative toehold was used by Genot et al. in a DNA-based system capable of calculating the product of binary matrix multiplication and weighted sums [15]. This strategy was also employed to operate DNA-based logic gates that induce the formation of split G-quadruplex.

(a)

(b)
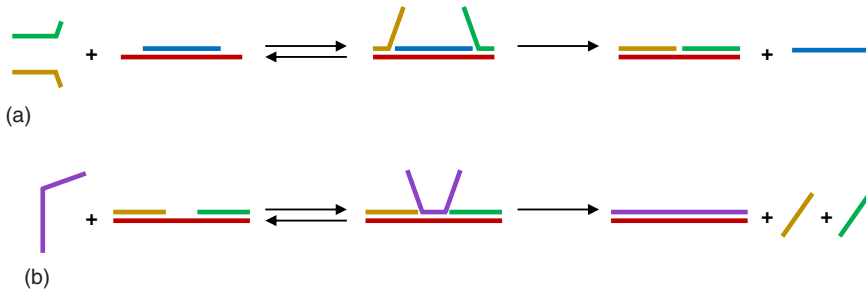
Figure 14.3 (a) Cooperative hybridization of the inputs induces the displacement of the output strand. (b) Combined displacement of two outputs by a single input strand.

Figure 14.4 Associative toehold mechanism. The helper strand (orange) facilitates the strand displacement reaction by bringing the invading strand (green) into proximity to its target domain (red).
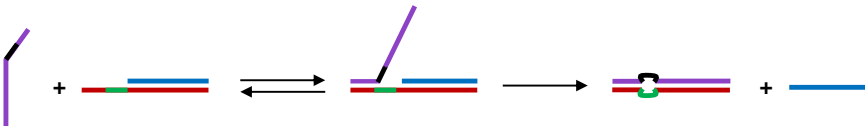
Figure 14.5 Remote toehold mechanism. A spacer separates the toehold and the branch migration domains. Modulation of the nature and size of the spacer allows to finely tune the kinetics of the strand displacement reaction.

The signal produced constitutes a scaffold that brings together two G-rich strands that form a bimolecular G-quadruplex [16, 17].

Another way of controlling the kinetics of a strand displacement reaction while keeping the toehold and branch migration domains in the same strand relies on the use of a remote toehold (Figure 14.5) [18]. In this technique, an inert spacer (DNA or PEG, green) physically separates the toehold and branch migration domains. Like the standard strand displacement mechanism, a first docking step brings the invading strand and the target duplex together. Once hybridized, additional internal diffusion is required before internal displacement occurs. The nature and the length of the spacer can be modulated to effectively tune the kinetics of the reaction.

Spatial segregation can also be used as a means of controlling interactions between DNA components of a reaction [19–21]. A typical approach employs surface-bound DNA molecules, either on a DNA scaffold or solid-phase beads. Alternative approaches for controlling the kinetics of toehold-mediated strand displacement reactions involve the introduction of mismatches within the branch migration domain [22–25]. Modulating the position and identity of the mismatch allows for fine kinetic tuning of the reaction.

Figure 14.6 The toehold exchange reaction. The process is fully reversible via reverse toehold binding and strand displacement.

14.1.2 The Toehold Exchange Mechanism

A strand displacement reaction is controlled by kinetics and thermodynamics. In order to reduce the contribution of the thermodynamic factor to the advancement of a reaction, Winfree developed a new mechanism called toehold exchange (Figure 14.6) [26]. It differs from the classical strand displacement reaction as the branch migration domain of the invading strand is shorter than the incumbent strand by a few nucleotides. Thus, after toehold binding and branch migration (Figure 14.6, step 1), an additional step consisting in the dissociation of the incumbent toehold from the target strand must occur to release the output (Figure 14.6, step 2).

The lengths of the invading and incumbent toeholds are critical in the regulation of the kinetics of a toehold exchange reaction. When the incumbent toehold is shorter than the invading toehold, reaction rates are similar to a standard toehold-based strand displacement reaction. However, when the incumbent toehold is longer than the invading toehold, the rate constant decreases strongly with the length of the incumbent toehold. This behavior is rationalized by the difference in binding energies between the toeholds. Indeed, when the binding energy difference between the toeholds favors a strong incumbent toehold, the probability of dissociation between the target strand and the incumbent strand decreases exponentially.

14.2 Dynamic Reconfiguration of Structural Devices

The introduction of the toehold-mediated strand displacement reaction has facilitated the development of new sequence-dependent DNA devices that couple information processing via DNA base pairing programmability to mechanical actuation. The first nanomachine that used DNA as a fuel to induce a programmed conformational change was demonstrated by Yurke et al. in 2000 (Figure 14.7a) [27]. A DNA tweezer was assembled by connecting two double-stranded DNA arms via a single-stranded DNA hinge. Both arms presented 24-base dangling ends labeled with fluorophores that quench each other through Förster resonance energy transfer (FRET). The tweezer was locked in a closed state using a DNA strand complementary to both dangling ends. This brought the FRET pairs into proximity, giving a low fluorescence response. The closing strand was then displaced by a complementary DNA fuel that induced reopening of the tweezer, leading to a high fluorescence response. The device then underwent several set and reset cycles as either the closing or the fuel strand was added. Further improvements on the device allowed the construction of a nanoactuator that operated in the reverse sense of the original
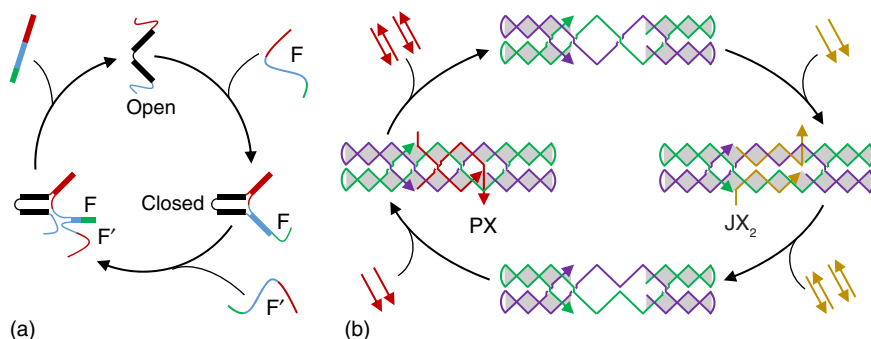
Figure 14.7 Programmed reconfiguration of DNA assemblies using the strand displacement reaction. (a) A DNA tweezer that can be switched between open and closed states. (b) Two-state conversion of a DNA nanomechanical device between PX (left) and JX$_2$ (right) states. The central strands confer the tile's identity. Those can be displaced and replaced with other strands to attain a new topology.

tweezer [28] and a DNA device displaying tweezer- and actuator-like properties that can be switched between three different states [29]. The two-state nature of the tweezer devices was used to operate nanomachines that can regulate biological [30, 31] and chemical reactions [32]. The operation of DNA tweezers was rendered autonomous by introducing a DNAzyme unit (catalytically active DNA domain capable of performing a chemical reaction) in the closing strand [33]. The DNAzyme activity conferred the device with continuous autonomous mechanical motion.

The robustness of a device is a key element to facilitate the distinction between the different transitional states. Robust systems are characterized by structurally well-defined topological states and therefore require the avoidance of flexible single-stranded DNA domains in structurally defining positions. Furthermore, the formation of by-products or other unwanted species during the operation of a device should be avoided. Seeman and coworkers took advantage of the interconversion between the paranemic PX crossover and its topoisomer JX$_2$ to operate a two-state nanomechanical device (Figure 14.7b) [34]. The crossover motif was shown to confer rigidity to these nanoassemblies, with conversion between the PX and the JX$_2$ motifs being achieved using a strand displacement reaction. The strand displacement reaction induced a 180° rotation of one end of the device relative to the other. The robustness of the device was demonstrated by several cycles of programmed interconversion between the two possible topological states. This device was then used to construct a machine that directs the synthesis of a DNA strand whose sequence is defined by the topological states adopted by the assembly [35], thus mimicking the function of the ribosome, despite lacking a translocational mechanism and hence limiting the product length to the size of the device. In a further development of such device, a PX-JX$_2$ cassette was built that integrates a DNA arm placed at a precise location in the assembly [36]. Programmed interconversion of the cassette allowed the arm to be maneuvered in a rotary fashion. This work demonstrated that a single device could be inserted and operated
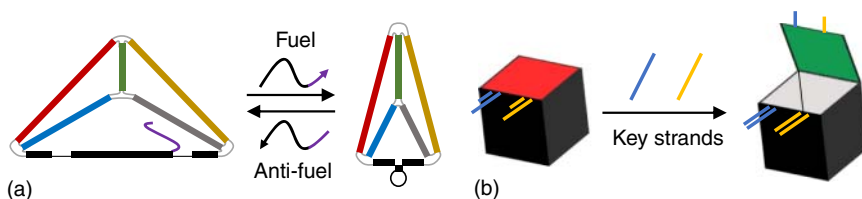
Figure 14.8 Programmed reconfiguration of DNA assemblies using the strand displacement reaction. (a) Conformational reconfiguration of a DNA tetrahedron. (b) Controlled opening of a DNA box. Both key strands are required to open the lid.

at a specific site on a substrate, which is of high importance for the development of nanorobotics with biological components.

Further developments in dynamic reconfiguration of structural devices were led by the desire to increase the number of topological states adopted by a nanodevice. Turberfield incorporated hairpin motifs at the edges of a DNA tetrahedron [37] to confer it with reconfiguration properties (Figure 14.8a) [38]. In the original state of the device, the hairpin is locked into a duplex state. A displacing strand (fuel) was subsequently added, freeing the hairpin motif that resulted in shortening the length of the edge of the tetrahedron. This introduced a distortion in the whole assembly, affording a different conformation. This process was reversed by simply adding the anti-fuel strand complementary to the hairpin. Introduction of independently addressable hairpins at two different positions on the tetrahedra facilitated switching between four different conformers. Gothelf and coworkers later devised a DNA tile actuator that was switched between eleven discrete states. Strand displacement reactions were used to selectively remove locking strands and access new topological states [39]. DNA catenanes have also been found to be capable of dynamic reconfiguration (Figure 14.9) [40]. Willner and coworkers demonstrated the reversible interconversion between the three possible topological states of a three-ring catenane triggered by strand displacement reaction [41]. The group also assembled a seven-ring DNA catenane that can exist as 16 different isomers [42]. The different isomers could theoretically be switched into another isomer via a total of 240 different switching pathways.

An evident application for such switchable nanodevices involves the encapsulation and release of small molecules for drug delivery. DNA cages have been described that can act as nanocontainers for small molecules. Andersen et al. assembled a DNA box of precisely controlled dimensions using the DNA origami technique (Figure 14.8b) [43]. The box presents a functional lid that was locked with sticky ends to the rest of the structure. The available toeholds were used to open the box lid in a programmable fashion. A simpler but similar device was constructed integrating dendritic DNA amphiphile molecules to a DNA scaffold [44]. The cube-core assembly was then capable of encapsulating hydrophobic guests, such as a fluorophore, into its hydrophobic cavity. Dendritic DNA strands were then displaced by eraser strands from the scaffold, leading to the liberation of the fluorophore into solution. An alternative option for cargo release involves the use of DNA-templated hydrogel synthesis [45, 46]. Based on this, Liedl et al. proposed
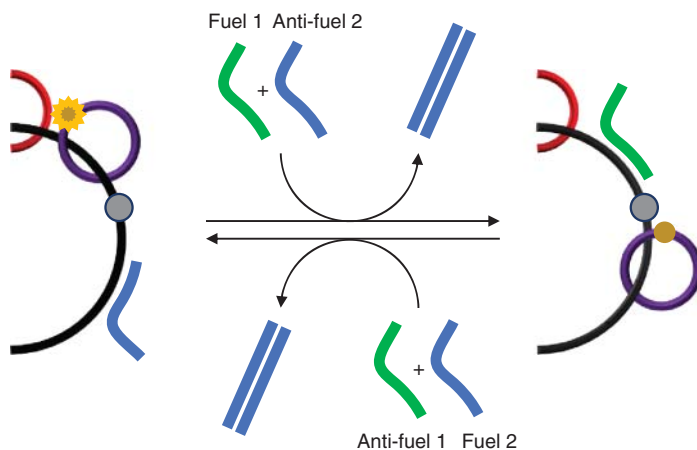
Figure 14.9 Dynamic reconfiguration of DNA-based interlocked catenanes using the appropriate fuel/anti-fuel strands. Transitioning between states is monitored by fluorescence.

a DNA-switchable gel for nanoparticle delivery [47]. Quantum dots were trapped in a DNA-cross-linked polyacrylamide gel and subsequently liberated after addition of complementary release strands that displaced cross-linking strands.

14.3 Stepped and Autonomous DNA Walkers

Bionanotechnology finds inspiration in nature with the aim of developing new materials and devices with functions that operate away from equilibrium. The activation of such devices induces either reconfiguration, information processing, motion, or a combination of these properties. A striking illustration of motion found in living organisms originates from molecular motors that convert fuel to perform mechanical work. Molecular motors perform tasks and achieve transport of cargo following processive movements. As an example, cytoskeletal motor proteins kinesin and dynein are responsible for the transport of various cellular cargos along microtubules in cells by converting ATP into mechanical work.

Synthetic molecular motors replicating this processive motion have been developed using DNA as a fuel for their operation. Commonly, a DNA walker system comprises a track assembled from complementary strands on which anchorage strands are tethered. These anchorages constitute the walking track followed by the DNA walker. Motion between stations is produced with the aid of fuel strands involved in strand displacement reactions that detach the walker from the tether strand to allow it to bind to the subsequent tether strand in the path. The first synthetic DNA walker powered by strand displacement was demonstrated by Shin and Pierce in 2004 (Figure 14.10) [48]. The track was assembled from six DNA strands into a duplex presenting four protruding branches that constitute the anchorages. The DNA walker is made up of two strands partially hybridized to each other, leaving two single-stranded domains that form the legs. The bipedal
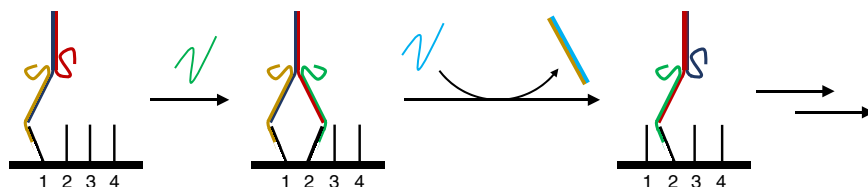
Figure 14.10 Programmed motion of a bipedal DNA walker. An attaching strand (in green) connects the second leg of the walker. Subsequently, a strand displacement reaction can occur that sees the attaching strand connecting the first leg (in gold) to the track being displaced by a detaching strand (in light blue). This way, the walker has moved from station 1 to station 2. Walking toward station 4 was achieved following the same stepwise mechanism.

walker was attached to the first station using an attachment strand complementary to both the anchorage strand and one leg of the walker. Similarly, the second leg was subsequently attached to the next anchorage on the track. A detaching strand was then used to displace the first attaching strand, releasing the leg of the walker and rendering it available for subsequent attachment to the third anchorage on the track. Stepwise motion was demonstrated between four stations. Although not autonomous, one impressive feature of this device resides in the length of the step taken by the bipedal walker, as it presents a finer step size than that of a kinesin protein on a microtubule [49]. Moreover, the step size can be conveniently adjusted by altering the length of the track.

The next big step for synthetic DNA motors was the attainment of autonomous locomotion. This was demonstrated using enzymes as a catalyst to turn over reactions resulting in movement [50–53]. However, achieving a similar task using DSD as a fuel poses two issues. Firstly, a single-stranded DNA fuel does not possess any catalytic activity and is thus consumed in a strand displacement reaction, which requires addition of new fuel strands. However, an autonomous system only allows for the addition of a triggering signal for the system to become operational, which contradicts the addition of new strands to the system. Secondly, the simultaneous presence of all triggering strands in the media may induce false triggering of the device or its incorrect operation due to unwanted crosstalk. Pierce and coworkers overcame these issues by programming the interactions between the components of a bipedal walking system using a DNA hairpin motif that sequesters the toehold required for a strand displacement reaction [54]. The DNA walker and the metastable hairpin anchorage motifs were tethered to a DNA scaffold. The DNA hairpin input could only interact with an anchorage strand once the latter had been unraveled from its hairpin state by the walker leg, leading to controlled motion of the walker. However, it was shown that the processive motion of the walker was stochastic as it could fully disassemble from the track and bind again at a random position where it could exhibit motion again. Omabegho et al. managed to tackle this by coordinating the movement of both legs of a DNA walker (Figure 14.11) [55]. Movement of one leg could be performed only once the other leg had finished its programmed transition from one station to the subsequent one. This way, they demonstrated the first autonomous bipedal walker with true motor behavior powered by strand displacement reactions.
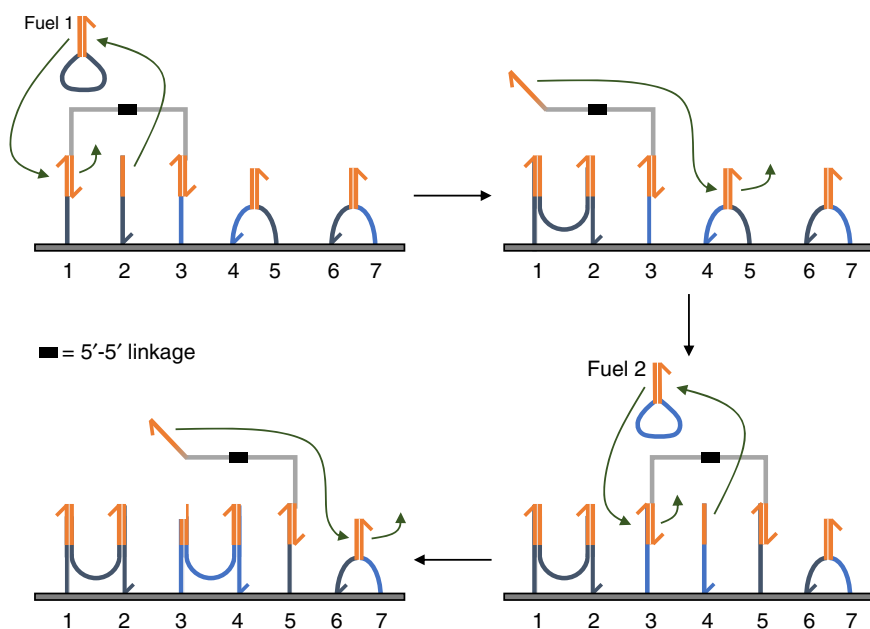
Figure 14.11 Autonomous directional motion of a DNA bipedal walking device. The movement of one leg from one station to the next is achieved by hybridization of a metastable DNA fuel strand that displaces the leg from its station. The leg can subsequently bind the next station by strand displacement reaction.

An equally important task for DNA walkers to achieve is transport. In a remarkable effort, Seeman and coworkers built a DNA walker capable of collecting different cargos in a programmable fashion (Figure 14.12) [56]. The walker possessed four legs required for motion and three arms tasked to grab a specific cargo. A walking track was constructed on the surface of an origami tile containing cargo-donating stations, each bearing a different set of nanoparticles. The donating stations were built from PX-JX$_2$ tiles that could be switched from the OFF state to the ON state by a strand displacement reaction. In the ON state, the cargo is brought in proximity to one arm of the device, allowing for its transfer from the cargo-donating station to the walker. A series of strand displacement reactions that would first activate the ON state of a donating station and then consecutively induce movement of the walker to the next station were used to program the motion of the walker along with cargo collection. The eight different possible collection paths were performed, demonstrating the possible programmable and sequential synthesis of a target product. Alternatively, prescribed motion of a DNA transporter across different states was reported on branched DNA tracks [57, 58] and across a network of tracks [59]. Muscat et al. reported the use of small molecule signals that bind to aptamer regions anchored on the track to direct the route of a molecular cargo [60]. A more sophisticated DNA transporter engineered by Qian and coworkers was shown to sort two types of cargos and transport them to discrete locations placed on opposite corners of
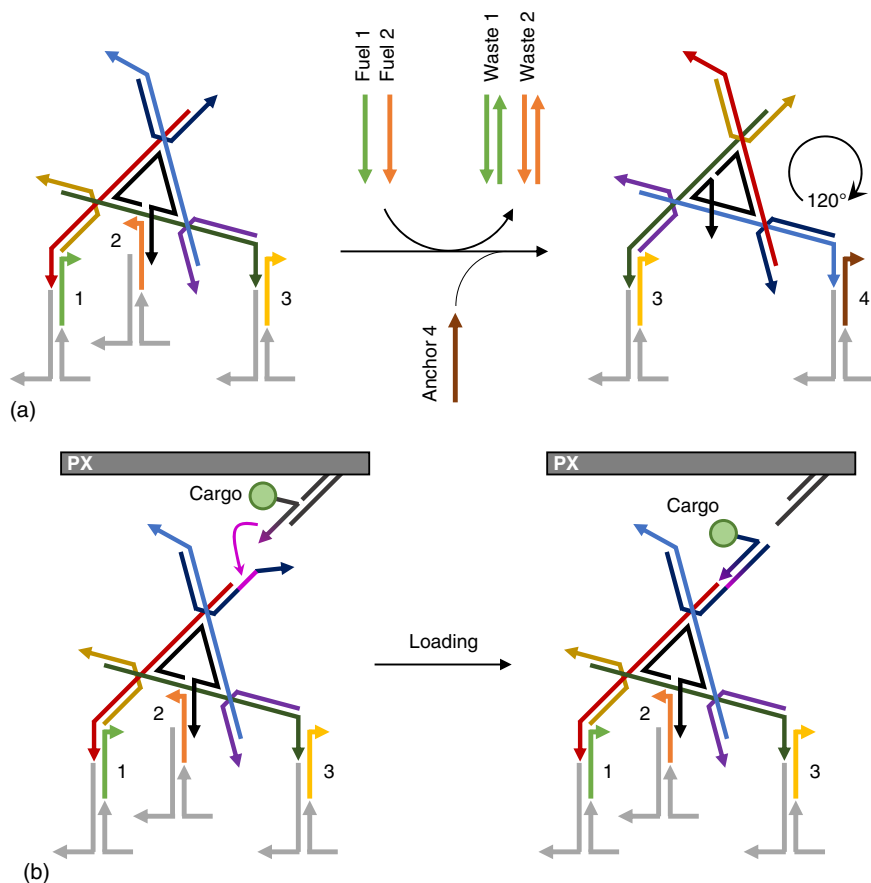
Figure 14.12 A DNA-based transporter. (a) Details of the movement of the walker. Motion is achieved by displacing specific anchor strands that bind the walker to the track. A subsequent 120° rotation brings a leg in proximity to the next station, and connection is attained using an anchor strand complementary to both the leg and the station strands. (b) Details of the cargo loading process. When the origami tile is in its PX state (ON), a donating transfer of a nanoparticle cargo is accomplished by strand displacement reaction.

a two-dimensional surface [61]. Multiple identical robots were capable of working in parallel on a single DNA origami surface, collectively achieving a unique cargo-sorting task. This was completed at a faster rate and with greater efficiency as the number of robots present on the surface increased.

14.4 Early Breakthroughs in DNA Computing

A broad definition of a computer describes it as a device capable of executing a specific set of instructions to produce a response. Conventional computers are composed of a motherboard that physically supports and electrically connects all the components of the system, such as the central processing unit (CPU),

the memory units, and other peripherals. The CPU constitutes the core of the computer, as it performs all the computation it is tasked to execute following a set of instructions registered in the memory, leading to the generation of an output. Current computers rely on a silicon-based technology, which has constantly evolved over the years to increase the complexity and number of transistors integrated into a CPU, giving it increasing computational power. Gordon Moore, founder of the semiconductor manufacturing company Intel, observed in 1965 that the number of transistors integrated per square inch in a computer circuit had doubled every year, which has since become known as Moore's law [62]. However, the race toward electronic miniaturization is currently facing challenges that may impede further developments [63], highlighting the need for new computer models.

In his famous talk "There's Plenty of Room at the Bottom" given in 1965, Richard Feynman suggested that information processing could be performed at the submicroscopic level using biological systems [64]. This would open the door to a new construction material since biological molecules are based on a carbon framework. Molecular computing emerged with the aim of bridging together computing, physics, and biotechnology to create new computational devices. DNA was soon regarded as the molecule of choice for that matter, owing to its programmable base pairing properties. Indeed, molecular computing relies on the fact that one can predict the interactive behavior of molecules, which is achievable with DNA. Through sequence design and self-assembly, DNA molecules can implement the input, computational program, and output functions required in computing.

14.4.1 Hamiltonian Paths

Following the concept introduced by Feynman [64] and later discussion by Conrad and Liberman [65, 66], the first experimental realization of DNA computation was demonstrated by Adleman [67]. He showed that DNA-based computations could be used to solve the Hamiltonian path problem, also commonly referred as the "travelling salesman problem," as represented by the directed graph G (Figure 14.13).

This is considered as an incomplete graph since not all the vertices are connected by edges. The directed graph is said to have a Hamiltonian path if a path

Figure 14.13 Directed graph G representing Adleman's Hamiltonian path problem.
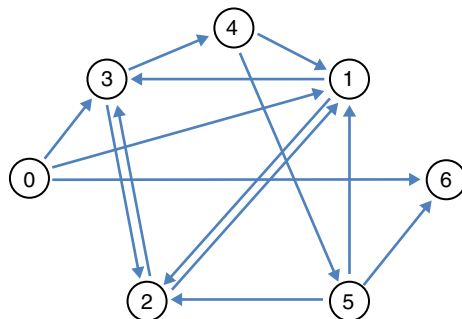
starting at specified vertex v_{in} and finishing at specified vertex v_{out} after visiting each vertex exactly once can be found. The algorithm used in classical computation to determine the existence of such a path proceeds as follows:

- *Step 1*. Generate random paths through the graph.
- *Step 2*. Keep paths starting at specified vertex v_{in} and finishing at specified vertex v_{out}.
- *Step 3*. Keep paths entering exactly n vertices (n being the number of vertices in the graph).
- *Step 4*. Keep paths entering all of the vertices at least once.
- *Step 5*. Give results of the computation; YES if paths exist, NO otherwise.

In his experiment, Adleman attributed a 20-nucleotide random DNA sequence to each vertex of the graph (seven in total). For each edge $i \rightarrow j$ of the graph, an oligonucleotide $O_i \rightarrow_j$ was designed in such a way that the first 10 nucleotides were complementary to the first 10 nucleotides of the O_i strand and the last 10 nucleotides were complementary to the last 10 nucleotides of the O_j strand. All the generated DNA strands were mixed together. The "vertices" strands were used as splints in a single ligation reaction to lead to the formation of DNA molecules encoding all of the possible random paths through the graph. Products of Step 1 were then submitted to a PCR amplification reaction using primers O_0 and O_6 such that only paths starting at vertex v_0 and finishing at vertex v_6 were retained. Step 3 was achieved by running an agarose gel to select the 140-base pair band, which corresponds to double-stranded DNA encoding paths entering exactly seven vertices. To implement the last step of the computation, the remaining path candidates were purified on successive affinity columns, each presenting a different "edge" DNA tag. This way, selection of the 140-bp strands entering all seven of the vertices was achieved. To identify the path that is a solution to the Hamiltonian graph, a graduated PCR was carried out. This technique revealed the order of edges appearing in the final duplex, hence providing the result of the computation (Figure 14.14).

Adleman's experiment is considered as a major breakthrough for DNA computation, since he demonstrated that DNA could be used to carry information and solve a computationally challenging problem. Despite the elegance of the experiment, some issues might arise when using DNA to perform computations. First, Adleman needed about seven days to achieve his experiment, which is quite a long time given the size of the graph. Some of the processes might be shortened and/or automated, but the time required to perform the same computation on a regular computer would be much shorter. However, when looking at individual steps, it appears that some of them such as Step 1 would take about the same amount of time if running the computation with either a small or a huge number of molecules. Another problem with molecular computation is the possibility of errors. Indeed, false hybridization between two strands whose sequences are not fully complementary might lead to a wrong path or a false negative.

If one had to compare computing at the molecular level to digital computing, it appears that individual operations proceed faster and in a much more
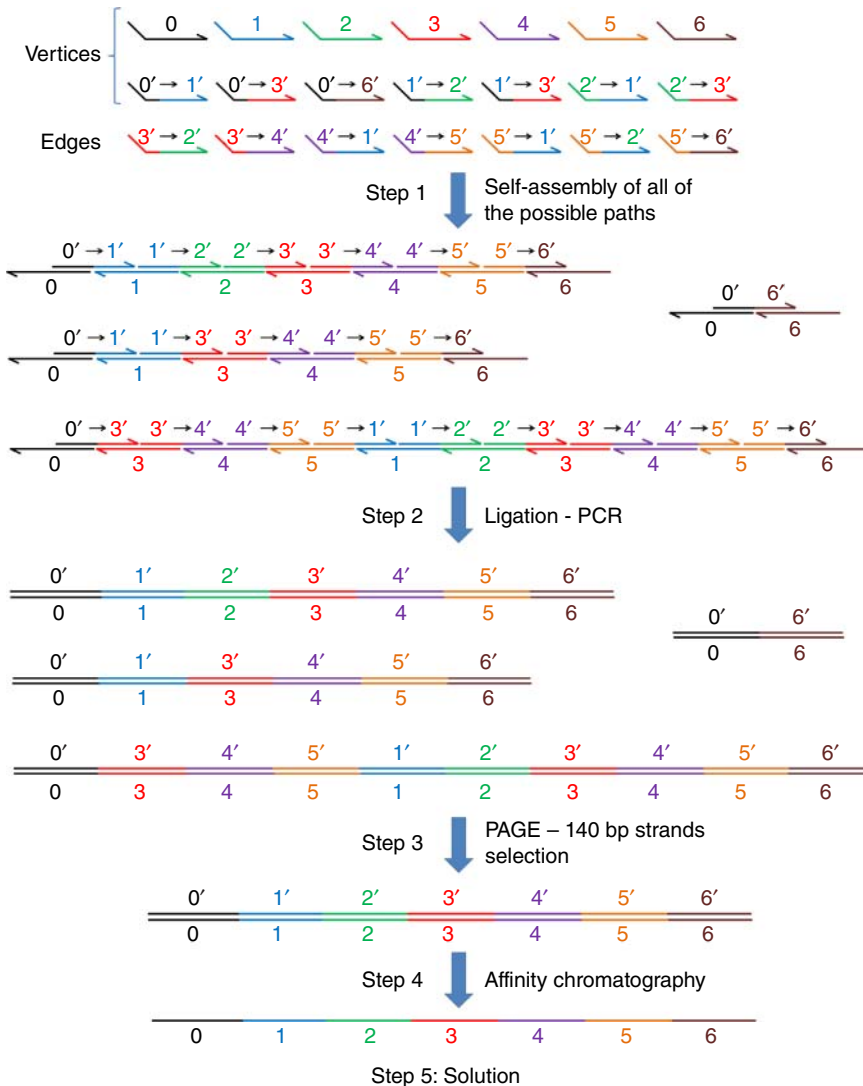
Figure 14.14 Adleman's DNA solution to the Hamiltonian path problem.

energy-efficient way at the molecular level. However, the biggest problem with DNA computing is that only one specific task can be achieved using a molecular algorithm. If a new task had to be computed, a new algorithm would need to be designed that operates with different processes. In contrast, digital computers offer much better flexibility in performing different tasks.

14.4.2 Satisfiability (SAT) Problem

The SAT problem involves determining whether a solution to a Boolean logic operation exists. In a paper published in 1995, Richard Lipton considers the following SAT problem: [68]
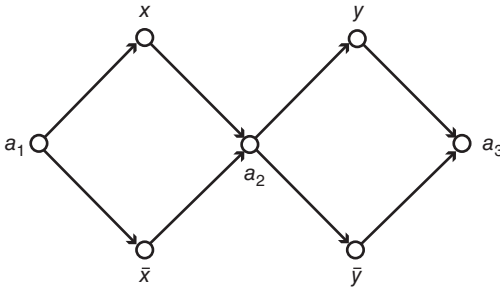
Figure 14.15 Graphical representation of a 2-SAT problem.

$$F = (x \lor y) \land (\bar{x} \lor \bar{y}) \tag{14.1}$$

The variables x and y are Boolean, meaning they can only take values 0 or 1. The symbol $\lor$ represents the logical OR operation ($x \lor y = 0$ only if $x = y = 0$), the symbol $\land$ represents the logical AND operation ($x \land y = 1$ only if $x = y = 1$), and $\bar{x}$ represents the "negation" of x ($\bar{x} = 0$ if $x = 1$ and vice versa). The aim is to find values for both variables in such a way that $F = 1$. A graph depicting this SAT problem and all the possible paths can be found in Figure 14.15.

In the graph, different paths start from the same vertex a_n and stop at the same vertex a_{n+1}. At each stage, the path has two choices, either the unprimed variable where it will encode 1 or the primed one where it will encode 0. For instance, the path $a_1 x a_2 \bar{y} a_3$ will encode 10. Instead of testing all the combinations of values for x and y, Lipton proposed a method using DNA to implement the computations. In a similar way to Adleman's experiment, random DNA sequences are assigned to each vertex and edge of the graph. These strands are partially complementary to each other, in such a way that the beginning of an edge strand is complementary to the end of a vertex strand and the beginning of the $n + 1$ vertex strand.

The experiment had to be conducted in separate test tubes, the first one being used to generate all the possible paths and the others to extract the solutions to the SAT problem. Tube t_0 comprises all the vertices and edge strands that can anneal through complementary sequences. A series of test tubes can then be constructed to extract the solution from the pool of random paths that were generated. Tube t_1 corresponds to $E(t_0, 1, 1)$, meaning that it will contain only strands encoding paths from t_0 that go through x (value = 1) in the first step. Therefore, values contained in t_1 are 10 and 11. A remainder tube t'_1 contains strands from t_0 encoding paths that are not present in t_1. Tube t_2 corresponds to $E(t'_1, 2, 1)$. Tube t_3 is the result of mixing t_1 and t_2 together, and values encoded by DNA in this tube satisfy the first OR clause. Tube t_4 corresponds to $E(t_3, 1, 0)$ and t_5 corresponds to $E(t'_4, 2, 0)$ with t'_4 being the remainder tube from t_3. The final tube t_6 is created by pouring t_4 and t_5 together. This tube contains all the values that satisfy both the first and the second clause: 01 and 10. The values encoded by DNA in the test tubes at each step are shown in Table 14.1.

Table 14.1 Paths and values encoded by the different test tubes used for solving a 2-SAT problem.

Test tube	Path	Values present
t_0	All generated	00, 01, 10, 11
t'_1	$E(t_0, 1, 1)$	10, 11
t_1	Remainder t_0	00, 01
t_2	$E(t'_1, 2, 1)$	01
t_3	$t_1 + t_2$	01, 10, 11
t_4	$E(t_3, 1, 0)$	01
t'_4	Remainder t_0	10, 11
t_5	$E(t'_4, 2, 0)$	10
t_6	$t_4 + t_5$	01, 10

This process is very time consuming in the case of more complex graphs. To overcome this issue, Adleman used successive affinity columns to extract the solution from the incorrect paths, in a similar way to which the Hamiltonian path problem experiment was performed. With this method, he was able to report the solution of a problem with 20 different parameters and 24 clauses in the equation [69].

14.5 DNA-Based Molecular Logic

14.5.1 Computing with Boolean Logic

Programming languages have been developed to program the behavior of interacting species. Molecular computing aims to use DNA to perform operations like those encountered in traditional silicon-based computers. Silicon-based computers function using a binary language to characterize the absence (value 0) or presence (value 1) of inputs and outputs that are connected to each other using a set of Boolean logic operations. These follow the rules of Boolean algebra, which was developed by the mathematician George Boole. Boolean algebra describes the various relationships existing between binary variables.

A Boolean operation is executed according to a propositional formula (Boolean expression), which describes how an output is generated from a set of inputs. The most common Boolean logic operators are the NOT, AND, and OR functions (Figure 14.16). The NOT function characterizes the negation, which produces an output in the absence of input and vice versa. The AND function describes the conjunction, where both inputs are required to produce an output. The OR function represents the disjunction, where at least one input needs to be present in the system to produce the desired output. Those operations can be theoretically assembled into logic circuits to eventually perform any desired logic operation.

	A	NOT A
	0	1
	1	0

(a) ¬ A

A	B	A AND B
0	0	0
1	0	0
0	1	0
1	1	1

(b) A ∧ B

A	B	A OR B
0	0	0
1	0	1
0	1	1
1	1	1

(c) A ∨ B
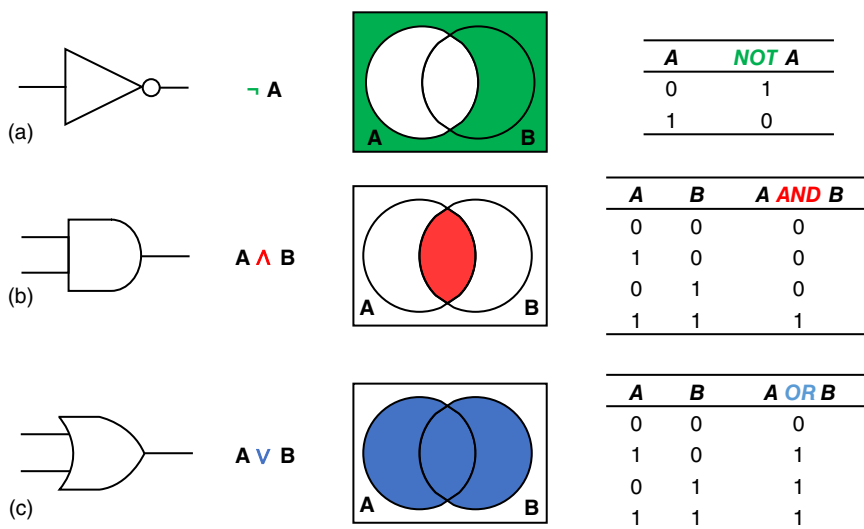
Figure 14.16 The Boolean logic operators NOT (a), AND (b), and OR (b). The different operators are represented with their respective circuit symbol, logical connective, Venn diagram, and truth table (from left to right).

DNA was promptly recognized as a substrate of high interest for performing Boolean logic owing to the base pairing programmability. Logic operations are directly implemented within DNA assemblies that are targeted by single-stranded inputs. Strand displacement reactions are used as the main mechanism for connecting multiple devices together as well as producing a single-stranded output.

14.5.2 Deoxyribozyme Logic Gates

An important step forward in performing DNA computation was achieved through the pioneering collaborative work of Stojanovic and Stefanovic, who reported the construction and the operation of the first logic gates known to use only DNA sequences as the inputs [70]. Their strategy was based on using deoxyribozymes, which are DNA molecules capable of various catalytic activities, such as phosphodiester bond formation or cleavage. Deoxyribozyme properties arise from their defined secondary and tertiary structures that bring the reactive moieties into proximity. Two previously studied deoxyribozymes have been used to build molecular logic gates [71, 72]. These deoxyribozymes have been modified by hybridizing short complementary strands to them. These strands inhibit the cleavage activity of a particular deoxyribozyme either by preventing them from folding into an active form or by interfering with their hybridization to a substrate strand. The target molecule is a chimeric DNA sequence containing a single ribonucleobase that indicates the cleavage site. A quencher and a fluorophore were attached, respectively, to the 5′ end and the 3′ end of the target strand. Five different logic gates (YES, NOT, AND, AND-NOT, and XOR) were constructed, and their operations monitored by fluorescence.
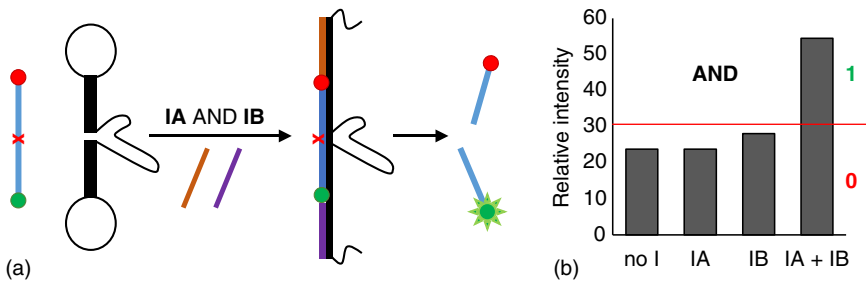
Figure 14.17 (a) Deoxyribozyme-based AND logic gate design with hairpins present at both ends of the gate strand. According to the Boolean truth table, the addition of both inputs will result in the opening of the two hairpins, allowing the fluorescent strand to hybridize. The latter is subsequently cleaved leading to fluorescence emission. (b) Relative fluorescence observed depending on the input added to the system. Only the addition of both inputs induces a characteristic increase in the fluorescence emission.

When adding the appropriate inputs, the deoxyribozyme present in the gate would see its activity restored by a strand displacement reaction, causing the target molecule to be cleaved. Thus, the interaction between the fluorophore and the quencher was modified accordingly, responding according to Boolean logic truth tables. An example of the operations for the AND gate is given in Figure 14.17.

The same approach has been employed by the same research team to build and operate deoxyribozyme-based logic gates that could operate with three inputs [73] and deoxyribozyme-based logic circuits [74, 75]. Similar logic operations have also been demonstrated using ribozymes [76]. They also reported a deoxyribozyme-based molecular automaton called MAYA that could play a tic-tac-toe game [77]. It encodes the 19 possible game combinations within 23 deoxyribozyme-based DNA logic gates arranged in a 3×3 grid. The automaton can take the most appropriate decision as a response to any move played by a human opponent. It implements a perfect strategy in such a way that it cannot be defeated, regardless of the moves played by the human opponent. MAYA was the first demonstration of a biomolecular device that could play a dynamic game. Moreover, its decision-making properties could be of high interest for performing tasks in a cellular environment.

The Willner research lab has also contributed significantly to the development of deoxyribozyme-based DNA logic gates. They introduced a versatile computing platform composed of two libraries of DNA strands, one consisting of DNAzyme subunits and the other of DNA substrates [78]. The addition of DNA inputs to the system directs the self-assembly of the various DNA strands into a computing module found either in an active DNAzyme form or in an unreactive duplex state depending on the nature of the inputs. The computational power of their design was illustrated by operating logic gates using cascades that fan out to perform parallel computations. The same approach was used to construct a multiplexer and a demultiplexer [79] as well as a full-adder [80], which are logic devices of much greater complexity than the consistent logic gates. The versatility of the DNAzyme platform was further enhanced by introducing an additional

level of control by modifying the pH of the media [81]. This way, a unique library of DNAzyme subunits and their substrates could display three different logic identities depending on the pH. Furthermore, the logic gates were reset to the single-stranded inputs and subunits by strand displacement reactions induced by anti-input strands. Alternatively, logic reversibility was reported by operating DNAzyme-based Toffoli and Fredkin logic gates [82]. Logic reversibility is achieved when a set of inputs can be unambiguously reconstructed from the corresponding set of outputs. However, thermodynamic reversibility could not be attained as cleaved DNA fragments are obtained from the operation of such DNAzyme logic gates.

14.5.3 Autonomous DNA Translators

A key step in the evolution of DNA logic systems has been the development of devices that can produce compatible outputs in response to a set of inputs, thereby enabling logic gates to be chained together into circuits. So-called DNA translators are capable of releasing oligonucleotides in response to the addition of multiple oligonucleotide sequences. Thus, a DNA sequence can be translated to another. The autonomous and self-sufficient nature of DSD reactions enables computations to be performed in sequence; different gates can be chained together by simply using the outputs displaced from one gate as the input for downstream computations. Moreover, the rate of the displacement can be controlled by varying the length and sequence of the toehold.

Three different approaches can be used for performing computations based on strand displacement reactions: toehold exchange, toehold sequestering, and solid-phase translation [21]. In the toehold exchange strategy, an input strand is translated into a new strand of a different sequence through branch migration and toehold binding (Figure 14.18). The strand displacement is not explicitly achieved since the input does not fully displace a strand. Instead, the input partially displaces another through toehold hybridization (Figure 14.18, step 1) such that the incumbent strand then freely dissociates due to the weak binding between its toehold domain and the complementary domain in the double strand (Figure 14.18, step 2).

Ghadiri and coworkers presented a series of DNA-based Boolean logic gates that used oligonucleotides as inputs and outputs, the latter being generated by strand displacement reactions [83]. Expanding this approach, the logic gates could be wired both linearly and in parallel to construct a half-adder and chained gate circuits. Using a similar approach, Zhang et al. constructed DNA logic gates on a circular scaffold that could present greater modularity over their single-stranded counterparts as multiple recognition domains can be adapted on the circular scaffold and outputs can be displaced from both sides [84]. Additionally, this type of scaffold allows for the precise positioning of fluorescent labels that can be used to monitor device operation. A similar approach was used by Famulok and coworkers to control the topological reconfiguration of pseudocatenanes following prescribed Boolean logic rules [85].

In a particularly notable example, the research group of Erik Winfree developed a simple DNA gate architecture that could be used to build large-scale circuits
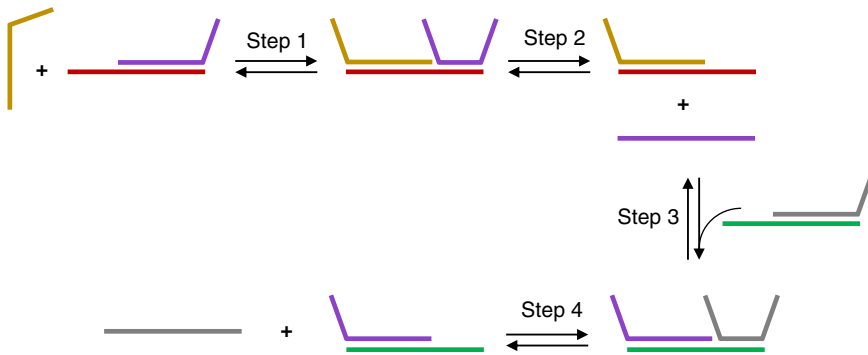
Figure 14.18 Strand displacement cascade illustrating a toehold exchange strategy.
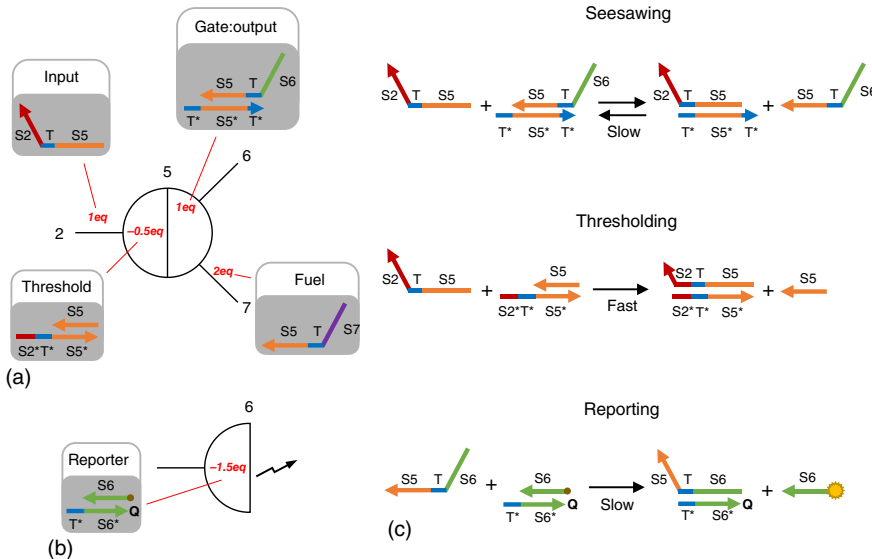
Figure 14.19 The DNA seesaw architecture. (a) Abstract seesaw gate formalism. Red numbers indicate initial concentration. (b) Abstract diagram of the reporter unit. (c) Reactions taking place in the seesaw mechanism. Seesawing describes the reversible displacement of a signal bound to the right by an input signal coming from the left. Thresholding describes the irreversible consumption of a signal molecule. The initial concentration of the thresholding unit confers the gate's identity. Reporting is used to monitor the activity of the logic circuit by fluorescence.

(Figure 14.19) [86]. The "seesaw" gates utilized reversible toehold exchange reactions in which a hybridized input containing a free toehold could be displaced to regenerate the original state of the gate. Thus, by shifting the position of the equilibrium, it was possible to switch activity from one side of the gate to the other. Boolean logic was implemented with the seesaw DNA motif to compute operations of AND and OR gates and circuits comprising four OR gates. A multilayer four-bit square root circuit operating with 130 DNA strands was also built [3]. The circuit used a combination of the AND and OR seesaw gates previously

designed to implement dual-rail logic. This way, both the TRUE and FALSE states were computed to calculate the square root of a four-digit binary number. The same architecture was employed for the construction of a neural network capable of playing a *read-your-mind* game where the device, after having been submitted to a set of four different DNA strands, is able to recognize a pattern in a similar DNA strand and tell which strand is the most similar from the pool of four [87]. This way, the system was shown to exhibit DNA-based associative memory.

The toehold sequestering technique provides an alternative solution-phase translation technique where a single-stranded toehold sequestered in a DNA duplex requires a toehold-mediated strand displacement reaction to release another toehold that is used in a subsequent steps (similar to Figure 14.18, but typically with direct, complete displacement). Seelig and coworkers developed a series of Boolean logic gates based on toehold sequestering and strand displacement that could be connected together to form logic circuits [5]. Due to possible gate failure, the authors introduced threshold and amplification gates to restore the signal induced by gate operations. The efficiency of the system was assessed by fluorescence. This work also showed that microRNAs could be used as inputs in DNA-based logic systems, suggesting the possible interfacing of such devices with biological nucleic acids, opening the door to potential applications in biotechnology.

In a solid-phase translation, the input releases the output after interacting with a gate attached to the solid-phase via toehold-mediated strand displacement reactions. The output released in solution can then be filtered off or used as an input in the next gate. Using this system, Frezza et al. reported the operations of DNA-based logic gates that were further wired together into a higher-order logic circuit (Figure 14.20) [19].

As an alternative to the use of a solid-phase support for the spatial isolation of the components of a circuit, Seelig et al. used a DNA origami scaffold to organize
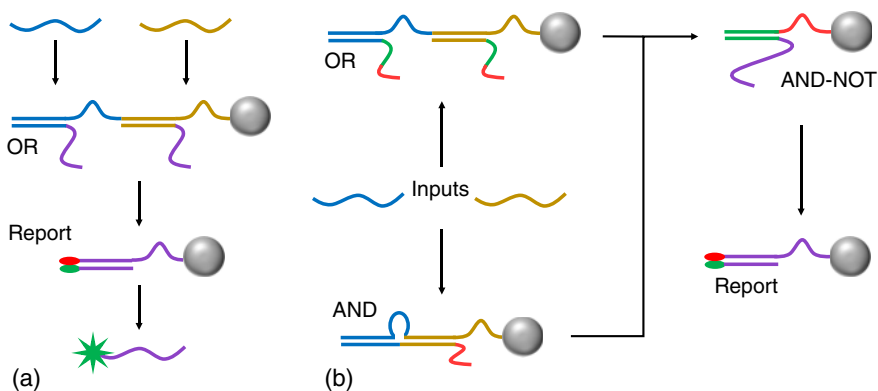
Figure 14.20 Operation of logic gates immobilized on a solid-phase support. (a) Operation of an OR logic gate. Both inputs are capable of displacing strands that contain a domain required for the activation of the reporting unit, leading to a fluorescence increase. (b) XOR circuit built from immobilized OR, AND, and AND-NOT logic gates.

the different logic units [88]. As the positions of the interacting sequences were co-localized, the computation time was shown to be reduced compared to a standard diffusive system. It was envisioned that such an assembly could be used to perform computations in a cellular environment, since transitional inputs could be less affected by DNA-binding molecules present in cells, thus enabling potentially bio-orthogonal *in vitro* computations.

14.5.4 Catalytic Systems for Signal Amplification

Two potential drawbacks of strand displacement-based DNA logic circuits arise from computational time and signal loss. Indeed, increasing the computational complexity and capability of a DNA circuit comes with an increased number of DNA components interacting with each other. As the number of components increases, so does the computational time, usually ranging from several minutes to a few hours. Additionally, a gradual signal loss can be observed at every computational step of a circuit. Thus, the development of new elements that could be introduced in a DNA circuit to improve the computational speed and the transmission of information is called for.

A first attempt in speeding up DNA reactions showed an enhancement of the strand displacement reaction by 2–4 orders of magnitude for the catalytic system over the classical scheme (Figure 14.21a) [89]. The target strand was topologically constrained into a bulge loop structure using a complementary strand. A catalyst strand was then added that opened the loop after strand displacement reaction and was released again after displacement by the invading strand, rendering it available for a subsequent catalytic cycle.

Winfree and coworkers improved this system by designing a new metastable fuel complex from two kissing loops stabilized by complementary strands (Figure 14.21b) [90]. The catalytic strand causes partial unwinding and relaxation of the fuel that subsequently undergoes a four-way branch migration to
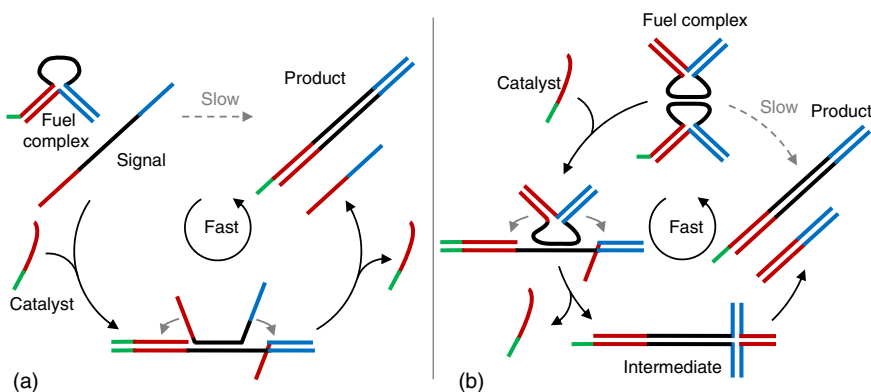
Figure 14.21 Early developments in DNA catalytic systems. (a) Kinetic control over the DNA hybridization is achieved by introduction of a catalyst that unravels the metastable bulge loop complex, allowing the subsequent invasion of the signal strand to afford the product. (b) Catalyzed decay of a metastable "kissing loop" complex. This method improves previous DNA catalytic systems by over 2 orders of magnitude.
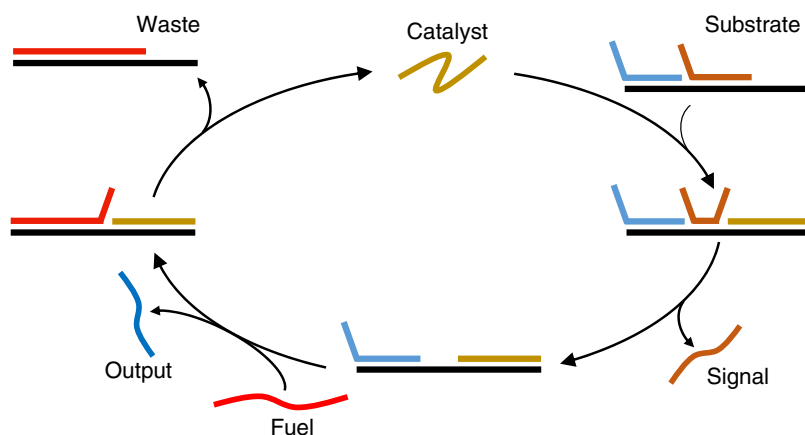
Figure 14.22 Entropy-driven catalytic DNA system. The catalyst first interacts with the substrate in a toehold exchange process that eventually releases the signal strand. A fuel strand subsequently targets the revealed domain to induce a double-sided strand displacement reaction that affords the desired output and regenerates the catalyst that can be fed into a new catalytic cycle.

afford the expected products. This work improved catalytic activity by over two orders of magnitude over the previous systems. The kissing loop catalyst was used as a signal amplification device in DNA-based logic circuits [5]. However, this device was deemed too complicated and lacking of modularity.

A new catalytic system was introduced that relies exclusively on entropic gain (Figure 14.22) [91]. A three-strand catalytic substrate was successively triggered by a catalytic strand and a fuel strand to release a signal and an output strand, thus regenerating the catalyst. A gain in configurational entropy is obtained from releasing the signal and the output while only the fuel strand is hybridized in the process. The entropic gain is not balanced by an enthalpic change as the number of base pairs remains identical at the end of the catalytic cycle, allowing the cycle to go forward. This catalytic system was demonstrated to be faster, simpler, and of higher modularity. This element was integrated into an AND logic gate that made use of two catalysts to produce an output.

The same team later demonstrated allosteric control over the catalytic activity of the system by trapping the active domain of the substrate into a metastable hairpin that can spontaneously open into an active state, thus inducing kinetic modulation of the catalytic cycle [92].

14.6 Future Prospects for Strand Displacement-Based Devices

14.6.1 DNA Chemical Reaction Networks

Computational processes are considered to be digital because a binary output is produced after the execution of a set of instructions, indicating the presence

or absence of response. However, the set of reactions involved in the transformation of a set of inputs into an output is intrinsically analog. Living organisms make use of such reactions to dynamically control their activity. Thus, important efforts have been made to develop theoretical models that explain biological behavior, giving rise to chemical reaction network (CRN) theory [93]. A CRN describes a set of reactions that turn a set of reactants into a set of products. CRNs incorporate the law of mass action for every reaction taking place. Hypothetically, all biological and chemical mechanisms can be described using a CRN, including all DNA-based computations and logic gates discussed above. In DNA systems, hybridization and strand displacement reactions go through a series of different intermediates before reaching completion. Those intermediates are not accounted for in the CRN formalism.

In molecular computation, the description of the molecular behavior of a DNA system by a CRN has allowed a universal descriptive language that models DNA interactions to be established. Following a reversed approach, Soloveichik et al. proposed to use CRNs as a prescriptive programming language [94], i.e. a set of rules that would define a DNA system rather than describe it. They used cascades of strand displacement reactions to couple chemical reactions between DNA species to reproduce the behavior of oscillating chaotic and algorithmic systems. Seelig and coworkers developed multistranded DNA gates to implement the reactions of non-catalytic, catalytic, and autocatalytic systems [95]. DNA gates were encoded in plasmid from a synthetic template and eventually nicked to afford the desired product. This strategy was investigated to overcome leaks and signal sequestration that are inherent to strand displacement-based systems.

Up to this point, DSD reactions were represented by an equivalent CRN and not executed directly. Researchers at Microsoft Research developed the DSD programming language that involves the key elements of a strand displacement reaction: sequence domains, toeholds, branch migration, and kinetic rates [96, 97]. The new programming language was implemented in a compiler that computes all possible interactions between a given population of DNA molecules of assigned domains. This was later integrated into the software Visual DSD [98] for rapidly designing and simulating DNA computational devices. Once computed, the interactions between a set of DNA species are rendered as a connected graph displaying the reactant, intermediate, and final products; a plot is produced that shows the real-time evolution of the concentration of all species in the system. The software is freely available and represents great progress for the engineering of future DNA circuits of higher complexity, since designed systems can be simulated and tuned before being experimentally executed. In particular, undesired side reactions can be readily identified and eliminated by altering the DNA sequences involved.

14.6.2 DNA Nanotechnology Goes *In Vivo*

Research in DNA computing is driven by the idea of using biological components of living cells, such as DNA and enzymes, in computation. It is believed that logical circuits made of biomolecules could become of greater importance compared

with standard silicon-based electronic circuits since it might be possible to perform operations in a massively parallel manner. Moreover, DNA computing could be used for many biological applications, first *in vitro* for biotechnology applications and eventually *in vivo* as smart drug delivery devices that could be activated by logical diagnosis of a disease.

Logic gates have been integrated in living cells with the aim of controlling specific biological processes. Transcriptional two-input AND gates consisting of a transcription factor and a chaperone protein found in type III secretion pathways of different bacteria strains such as *Salmonella* or *Pseudomonas* have been developed [99]. One of the inputs activates the expression of a transcription factor, while the other regulates the expression of the chaperone protein. When both inputs were applied in the system, the transcription factor turned on the output promoter.

Other synthetic networks reported the use of inputs that downregulated the activation of synthetic promoters that encode for RNA-binding protein or an RNA target unit [100]. Their interaction was in turn blocked, which led to inhibiting the release of the gene translator reporter.

In another logic-based gene regulation experiment, Lu et al. built logic modules based on plasmids inserted into *Escherichia coli* cells [101]. They would all contain promoter and terminator DNA sequences that start or stop gene expression. An output gene encoding for green fluorescent protein was used as the reporter. Recombinase enzymes were used as the inputs that would target terminators located between a promoter and the output gene. Sixteen binary Boolean logic gates have been developed. Interestingly, this approach was demonstrated to also provide the cells with DNA memory as the output would be maintained over many cell generations even without feeding the system with the appropriate inputs.

Logic-controlled bacteriophage serine integrase activity has also been used to control the flow of RNA polymerase along DNA and thus the transcription rate within cells [102]. DNA logic was also employed *ex vivo* for cell surface marker recognition to evaluate cell state [103] and even for controlling a molecule *in vivo* that targets cockroach cells [104].

Considering the operating power of DNA molecules, it is only logical for DNA computing to move toward biological environments. The DNA molecule represents a formidable computing scaffold owing to its programmable base pairing properties and, by virtue of its origin, could be interfaced with other biomolecules and processes within a cell. A wide variety of applications can be envisaged, ranging from targeted delivery, genetic regulation, and cellular evaluation to therapeutic applications. DNA computing devices capable of responding to their environment, making decisions, and initiating an appropriate response could be of fundamental importance for biotechnological and biomedical applications. A multitude of devices have been developed to date, and many reviews dealing with the topic have been published so far [105–110], demonstrating the ever-increasing appeal of the field. However, important challenges remain to be addressed. Firstly, difficulties are encountered with the cellular uptake of DNA-based devices. Delivery and absorption need to be specific, and pharmacokinetic and pharmacodynamic studies will be required

before one may consider using such devices on patients. DNA-based constructs may themselves be considered as drugs and therefore would have to follow the common drug development and approval process, which is demanding of both time and money. Additionally, despite being easily operated in a test tube, DNA devices may suffer from interfering interactions with the myriad of components present in the cellular environment, either preventing their correct operation or causing their degradation. Speed of operation is another important feature, as the devices must perform their function before being degraded. Should these issues be tackled, the combination of movement, defined structure, and computational and decision-making properties exhibited by DNA devices would be a great asset for therapeutic applications.

Acknowledgment

References

 1 Yurke, B. and Mills, A. (2003). *Genetic Program. Evolv. Mach.* 4: 111.
 2 Šulc, P., Romano, F., Ouldridge, T.E. et al. (2012). *J. Chem. Phys.* 137: 135101.
 3 Qian, L. and Winfree, E. (2011). *Science* 332: 1196–1201.
 4 Nakayama, S., Yan, L., and Sintim, H.O. (2008). *J. Am. Chem. Soc.* 130: 12560–12561.
 5 Seelig, G., Soloveichik, D., Zhang, D.Y., and Winfree, E. (2006). *Science* 314: 1585–1588.
 6 Dirks, R.M. and Pierce, N.A. (2004). *Proc. Natl. Acad. Sci. U.S.A.* 101: 15275–15278.
 7 Xing, Y., Yang, Z., and Liu, D. (2011). *Angew. Chem., Int. Ed.* 50: 11934–11936.
 8 Amodio, A., Zhao, B., Porchetta, A. et al. (2014). *J. Am. Chem. Soc.* 136: 16469–16472.
 9 Prokup, A., Hemphill, J., and Deiters, A. (2012). *J. Am. Chem. Soc.* 134: 3810–3815.
10 Hemphill, J. and Deiters, A. (2013). *J. Am. Chem. Soc.* 135: 10512–10518.
11 Huang, F., You, M., Han, D. et al. (2013). *J. Am. Chem. Soc.* 135: 7967–7973.
12 Yang, X., Tang, Y., Traynor, S.M., and Li, F. (2016). *J. Am. Chem. Soc.* 138: 14076–14082.
13 Zhang, D.Y. (2011). *J. Am. Chem. Soc.* 133: 1077–1086.
14 Chen, X. (2012). *J. Am. Chem. Soc.* 134: 263–271.
15 Genot, A.J., Bath, J., and Turberfield, A.J. (2013). *Angew. Chem., Int. Ed.* 52: 1189–1192.
16 Zhu, J., Zhang, L., Dong, S., and Wang, E. (2013). *ACS Nano* 7: 10211–10217.
17 Zhu, J., Zhang, L., Zhou, Z. et al. (2014). *Anal. Chem.* 86: 312–316.
18 Genot, A.J., Zhang, D.Y., Bath, J., and Turberfield, A.J. (2011). *J. Am. Chem. Soc.* 133: 2177–2182.

19 Frezza, B.M., Cockroft, S.L., and Ghadiri, M.R. (2007). *J. Am. Chem. Soc.* 129: 14875–14879.

20 Teichmann, M., Kopperger, E., and Simmel, F.C. (2014). *ACS Nano* 8: 8487–8496.

21 Picuri, J.M., Frezza, B.M., and Ghadiri, M.R. (2009). *J. Am. Chem. Soc.* 131: 9368–9377.

22 Zhang, D.Y., Chen, S.X., and Yin, P. (2012). *Nat. Chem.* 4: 208–214.

23 Machinek, R.R.F., Ouldridge, T.E., Haley, N.E.C. et al. (2014). *Nat. Commun.* 5: 5324.

24 Broadwater, D.W.B. and Kim, H.D. (2016). *Biophys. J.* 110: 1476–1484.

25 Zhang, D.Y. and Winfree, E. (2010). *Nucleic Acids Res.* 38: 4182–4197.

26 Zhang, D.Y. and Winfree, E. (2009). *J. Am. Chem. Soc.* 131: 17303.

27 Yurke, B., Turberfield, A.J., Mills, A.P. et al. (2000). *Nature* 406: 605.

28 Simmel, F.C. and Yurke, B. (2001). *Phys. Rev. E* 63: 041913.

29 Simmel, F.C. and Yurke, B. (2002). *Appl. Phys. Lett.* 80: 883–885.

30 Liu, M.; Fu, J.; Hejesen, C.; Yang, Y.; Woodbury, N.W.; Gothelf, K.; Liu, Y.; Yan, H. (2013). *Nat. Commun.* 4: 2127.

31 Dittmer, W.U. and Simmel, F.C. (2004). *Nano Lett.* 4: 689–691.

32 Chhabra, R., Sharma, J., Liu, Y., and Yan, H. (2006). *Nano Lett.* 6: 978–983.

33 Chen, Y., Wang, M., and Mao, C. (2004). *Angew. Chem., Int. Ed.* 43: 3554–3557.

34 Yan, H., Zhang, X., Shen, Z., and Seeman, N.C. (2002). *Nature* 415: 62–65.

35 Liao, S. and Seeman, N.C. (2004). *Science* 306: 2072–2074.

36 Ding, B. and Seeman, N.C. (2006). *Science* 314: 1583–1585.

37 Goodman, R.P., Schaap, I.A.T., Tardin, C.F. et al. (2005). *Science* 310: 1661–1665.

38 Goodman, R.P., Heilemann, M., Doose, S. et al. (2008). *Nat. Nanotechnol.* 3: 93–96.

39 Zhang, Z., Olsen, E.M., Kryger, M. et al. (2011). *Angew. Chem., Int. Ed.* 50: 3983–3987.

40 Elbaz, J., Wang, Z.-G., Wang, F., and Willner, I. (2012). *Angew. Chem., Int. Ed.* 51: 2349–2353.

41 Lu, C.-H., Qi, X.-J., Cecconello, A. et al. (2014). *Angew. Chem., Int. Ed.* 53: 7499–7503.

42 Lu, C.-H., Cecconello, A., Qi, X.-J. et al. (2015). *Nano Lett.* 15: 7133–7137.

43 Andersen, E.S., Dong, M., Nielsen, M.M. et al. (2009). *Nature* 459: 73–76.

44 Edwardson, T.G.W., Carneiro, K.M.M., McLaughlin, C.K. et al. (2013). *Nat. Chem.* 5: 868–875.

45 Qi, H., Ghodousi, M., Du, Y. et al. (2013). *Nat. Commun.* 4: 2275.

46 Lee, J.B., Peng, S., Yang, D. et al. (2012). *Nat. Nanotechnol.* 7: 816–820.

47 Liedl, T., Dietz, H., Yurke, B., and Simmel, F. (2007). *Small* 3: 1688–1693.

48 Shin, J.-S. and Pierce, N.A. (2004). *J. Am. Chem. Soc.* 126: 10834–10835.

49 Vale, R.D. and Milligan, R.A. (2000). *Science* 288: 88–95.

50 Yin, P., Yan, H., Daniell, X.G. et al. (2004). *Angew. Chem., Int. Ed.* 43: 4906–4911.

51 Venkataraman, S., Dirks, R.M., Rothemund, P.W.K. et al. (2007). *Nat. Nanotechnol.* 2: 490–494.

52 Tian, Y., He, Y., Chen, Y. et al. (2005). *Angew. Chem., Int. Ed.* 44: 4355–4358.

53 Bath, J., Green, S.J., and Turberfield, A.J. (2005). *Angew. Chem., Int. Ed.* 44: 4358–4361.

54 Yin, P., Choi, H.M.T., Calvert, C.R., and Pierce, N.A. (2008). *Nature* 451: 318–322.

55 Omabegho, T., Sha, R., and Seeman, N.C. (2009). *Science* 324: 67–71.

56 Gu, H., Chao, J., Xiao, S.-J., and Seeman, N.C. (2010). *Nature* 465: 202–205.

57 Muscat, R.A., Bath, J., and Turberfield, A.J. (2011). *Nano Lett.* 11: 982–987.

58 Wang, Z.-G., Elbaz, J., and Willner, I. (2012). *Angew. Chem., Int. Ed.* 51: 4322–4326.

59 Wickham, S.F.J., Bath, J., Katsuda, Y. et al. (2012). *Nat. Nanotechnol.* 7: 169–173.

60 Muscat, R.A., Bath, J., and Turberfield, A.J. (2012). *Small* 8: 3593–3597.

61 Thubagere, A.J., Li, W., Johnson, R.F. et al. (2017). *Science* 357: eaan6558.

62 Moore, G.E. (2006). *IEEE Solid-State Circuits Soc. Newslett.* 11: 33–35.

63 Waldrop, M. (2016). *Nature* 530: 144–147.

64 Editorial: "Plenty of room" revisited (2009). *Nat. Nanotechnol.* 4: 781.

65 Conrad, M. (1985). *Commun. ACM* 28: 464–480.

66 Conrad, M. and Liberman, E.A. (1982). *J. Theor. Biol.* 98: 239–252.

67 Adleman, L.M. (1994). *Science* 266: 1021–1024.

68 Lipton, R. (1995). *Science* 268: 542–545.

69 Braich, R.S., Chelyapov, N., Johnson, C. et al. (2002). *Science* 296: 499–502.

70 Stojanovic, M.N., Mitchell, T.E., and Stefanovic, D. (2002). *J. Am. Chem. Soc.* 124: 3555–3561.

71 Breaker, R.R. and Joyce, G.F. (1995). *Chem. Biol.* 2: 655–660.

72 Santoro, S.W. and Joyce, G.F. (1997). *Proc. Natl. Acad. Sci. U.S.A* 94: 4262–4266.

73 Lederman, H., Macdonald, J., Stefanovic, D., and Stojanovic, M.N. (2006). *Biochemistry* 45: 1194–1199.

74 Stojanović, M.N. and Stefanović, D. (2003). *J. Am. Chem. Soc.* 125: 6673–6676.

75 Stojanovic, M.N., Semova, S., Kolpashchikov, D. et al. (2005). *J. Am. Chem. Soc.* 127: 6914–6915.

76 Penchovsky, R. and Breaker, R.R. (2005). *Nat. Biotechechnol.* 23: 1424–1433.

77 Stojanovic, M.N. and Stefanovic, D. (2003). *Nat. Biotechnol.* 21: 1069–1074.

78 Elbaz, J., Lioubashevski, O., Wang, F. et al. (2010). *Nat. Nanotechnol.* 5: 417–422.

79 Orbach, R., Remacle, F., Levine, R.D., and Willner, I. (2014). *Chem. Sci.* 5: 1074–1081.

80 Orbach, R., Wang, F., Lioubashevski, O. et al. (2014). *Chem. Sci.* 5: 3381–3387.

81 Elbaz, J., Wang, F., Remacle, F., and Willner, I. (2012). *Nano Lett.* 12: 6049–6054.

82 Orbach, R., Remacle, F., Levine, R.D., and Willner, I. (2012). *Proc. Natl. Acad. Sci. U.S.A.* 109: 21228–21233.

83 Voelcker, N.H., Guckian, K.M., Saghatelian, A., and Ghadiri, M.R. (2008). *Small* 4: 427–431.

84 Zhang, C., Yang, J., and Xu, J. (2010). *Langmuir* 26: 1416–1419.

85 Li, T., Lohmann, F., and Famulok, M. (2014). *Nat. Commun.* 5: 4940.

86 Qian, L. and Winfree, E. (2011). *J. R. Soc. Interface* 8: 1281–1297.

87 Qian, L., Winfree, E., and Bruck, J. (2011). *Nature* 475: 368–372.

88 Chatterjee, G., Dalchau, N., Muscat, R.A. et al. (2017). *Nat. Nanotechnol.* 12: 920–927.

89 Turberfield, A.J., Mitchell, J.C., Yurke, B. et al. (2003). *Phys. Rev. Lett.* 90: 118102.

90 Seelig, G., Yurke, B., and Winfree, E. (2006). *J. Am. Chem. Soc.* 128: 12211–12220.

91 Zhang, D.Y., Turberfield, A.J., Yurke, B., and Winfree, E. (2007). *Science* 318: 1121.

92 Zhang, D.Y. and Winfree, E. (2008). *J. Am. Chem. Soc.* 130: 13921–13926.

93 Feinberg, M. (2019). *Foundations of Chemical Reaction Network Theory.* Springer International Publishing.

94 Soloveichik, D., Seelig, G., and Winfree, E. (2010). *Proc. Natl. Acad. Sci. U.S.A.* 107: 5393–5398.

95 Chen, Y.-J., Dalchau, N., Srinivas, N. et al. (2013). *Nat. Nanotechnol.* 8: 755–762.

96 Phillips, A. and Cardelli, L. (2009). *J. R. Soc. Interface* 6: S419–S436.

97 Lakin, M.R., Youssef, S., Cardelli, L., and Phillips, A. (2012). *J. R. Soc. Interface* 9: 470–486.

98 Lakin, M.R., Youssef, S., Polo, F. et al. (2011). *Bioinformatics* 27: 3211–3213.

99 Moon, T.S., Lou, C., Tamsir, A. et al. (2012). *Nature* 491: 249–253.

100 Auslander, S., Auslander, D., Muller, M. et al. (2012). *Nature* 487: 123–127.

101 Siuti, P., Yazbek, J., and Lu, T.K. (2013). *Nat. Biotechnol.* 31: 448–452.

102 Bonnet, J., Yin, P., Ortiz, M.E. et al. (2013). *Science* 340: 599–603.

103 Rudchenko, M., Taylor, S., Pallavi, P. et al. (2013). *Nat. Nanotechnol.* 8: 580–586.

104 Amir, Y., Ben-Ishay, E., Levner, D. et al. (2014). *Nat. Nanotechnol.* 9: 353–357.

105 Simmel, F. (2007). *Nanomedicine* 2: 817–830.

106 Jung, C. and Ellington, A.D. (2014). *Acc. Chem. Res.* 47: 1825–1835.

107 Linko, V., Ora, A., and Kostiainen, M.A. (2015). *Trends Biotechnol.* 33: 586–594.

108 Chen, Y.-J., Groves, B., Muscat, R.A., and Seelig, G. (2015). *Nat. Nanotechnol.* 10: 748–760.

109 Hockenberry, A.J. and Jewett, M.C. (2012). *Curr. Opin. Chem. Biol.* 16: 253–259.

110 Kahan, M., Gil, B., Adar, R., and Shapiro, E. (2008). *Physica D* 237: 1165–1172.

15

Development and Application of Catalytic DNA in Nanoscale Robotics

David Arredondo[1], Matthew R. Lakin[2], Darko Stefanovic[2], and Milan N. Stojanovic[3,4]

[1] *Center for Biomedical Engineering, University of New Mexico, 1 University of New Mexico, Albuquerque, NM, 87131, USA*
[2] *Department of Computer Science, Department of Chemical & Biological Engineering, Center for Biomedical Engineering, University of New Mexico, 1 University of New Mexico, Albuquerque, NM, 87131, USA*
[3] *Division of Experimental Therapeutics, Department of Medicine, Columbia University Medical Center, New York, NY 10032, USA*
[4] *Departments of Biomedical Engineering and Systems Biology, Columbia University, New York, NY 10027, USA*

15.1   Introduction

Biology often provides useful design paradigms that can be adapted by engineers for novel purposes, yet some designs continue to evade our complete understanding owing to their complexity. Naturally occurring molecular motor proteins, for example, are used pervasively by natural systems to transport cargo or generate mechanical force by using ATP hydrolysis to fuel specific conformational changes, resulting in directed motion [1, 2]. Instead of attempting to reverse-engineer such complex machinery, a more common approach to protein engineering is to take a naturally occurring protein and make small changes to its components and therefore improve or change its function. In contrast, the field of DNA nanotechnology uses Watson–Crick base pairing, a system with much simpler dynamics than protein folding, to construct devices from the ground up. The information processing nature of DNA may be utilized by DNA robots to direct their activity via biomolecular inputs, which may be used to program active assembly of molecular-scale objects [3]. The introduction of catalytic activity to DNA walking systems can be used to generate directional motion or motion against external force [4–6]. Recent reviews cover the various DNA walking mechanisms and their diverse applications [7, 8]. This chapter will mainly focus on DNAzyme-based implementations of molecular walkers and the theoretical analyses that seek to guide their design.

15.2   Brief History of DNAzymes

Catalytic DNAs (DNAzymes or deoxyribozymes) are a class of DNA oligonucleotides that can catalyze covalent modification of other molecules, primarily

nucleic acids. They contain a conserved sequence whose secondary structure acts as a catalytic core as well as recognition arms of variable sequence on either side of the core. A wide variety of catalytic core motifs have been identified via *in vitro* selection (SELEX), including the RNA-cleaving 8–17 and 10–23 DNAzymes [9]. DNAzymes can act on a suitable substrate, which typically consists of two domains complementary to the recognition arms connected by a single RNA base that is located opposite the catalytic core when the substrate is bound to the DNAzyme. In the presence of a metal ion cofactor, the DNAzyme will cleave the phosphodiester bond of the RNA base on the substrate strand. The kinetics of the reaction can be influenced by the type of ion and its concentration as well as environmental factors such as temperature and pH [9]. DNAzymes that can cleave DNA bases exist as well, but these are relatively slow and inefficient compared with RNA-cleaving DNAzymes [10].

Success in the application of DNAzymes for the purpose of information processing lends motivation to the development of DNAzyme-based walkers. One approach to information processing using DNAzymes is through the mimicry of digital logic circuits, in which Boolean logic calculations are carried out by arrays of DNA gates. The MAYA (Molecular Array of YES and AND gates) automata [11–13] are a series of molecular systems capable of playing games where decisions consist of Boolean calculations in response to single-stranded DNA inputs. A MAYA automaton exists as a set of separate solutions of various species of DNA gates. A gate may take up to three input DNA strands, the correct combination of which will result in the gate becoming a catalytically active DNAzyme. The active DNAzyme will then cleave a complementary strand containing a fluorophore–quencher pair, thus increasing the fluorescence of its solution. The game of tic-tac-toe is played by the first two iterations of MAYA (MAYA I and II) [11, 12] where nine separate wells represent the nine cells of a grid in a game of tic-tac-toe. The automaton receives an input strand, presented to all nine wells, that represents the choice of the human player. The gates in each well are chosen such that the automaton plays a predefined strategy – given a move by the opponent, a corresponding well will be chosen by the automaton in response until the game has ended. The third iteration (MAYA III) [13] plays a simpler game called "tit-for-tat," which consists of only four cells. There are a total of two turns each for the human and the automaton, starting with the human player, where they take turns choosing a cell. If either the human or the automaton choose a cell that has already been chosen, they lose. In contrast to MAYA I and II, each well contains an identical set of trainable gates. By first adding a training input to certain individual wells, the automaton is primed to react to the given player input; therefore, when the player input is added to all wells simultaneously, only the primed wells will react. This is an important advancement as it allows an untrained proto-automaton to be configured to use a certain game strategy via the introduction of training inputs and potentially reconfigured, as opposed to necessitating a different set of gates in each well that must be preconfigured to use only one strategy. Figure 15.1 shows how the MAYA III system used allosteric control to activate or inactivate DNAzymes based on the presence or absence of input oligonucleotides.
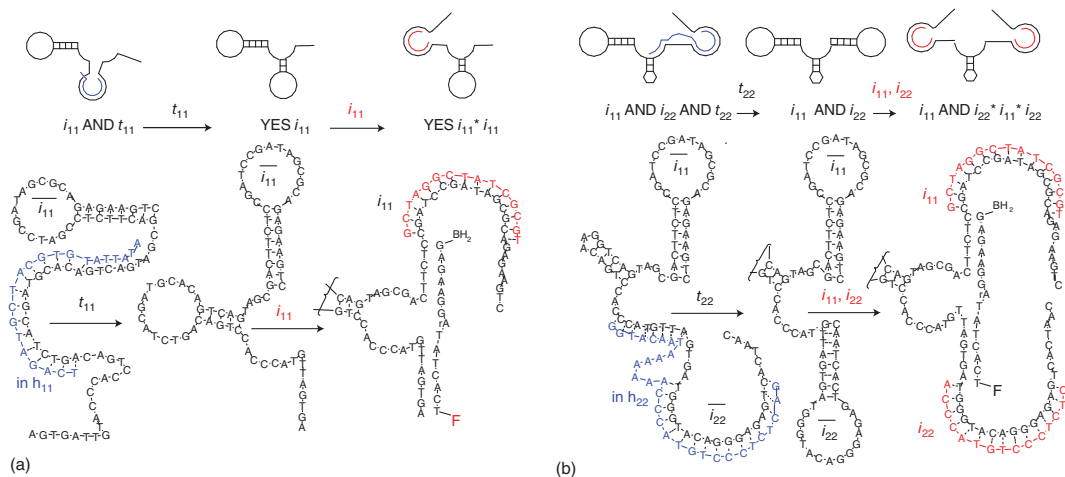
Figure 15.1 Basic molecular logic units and their activation during training and game play. (a) A two-input AND gate (i_{11} ANDt_{11}) is turned, upon addition of a training input (t_{11}), into a single-input YES gate (YESi_{11}). The training input is the complement of an oligonucleotide (inh_{11}) that is pre-complexed with the gate, inhibiting the enzymatic activity of the deoxyribozyme. The single-input YES gate is activated by the play input i_{11} to cleave the double end-labeled substrate. (b) A three-input ANDAND gate (i_{11}ANDi_{22}ANDt_{22}) is turned into a two-input AND gate (i_{11}ANDi_{22}) by a training input (t_{22}) complementary to an oligonucleotide (inh_{22}) that is pre-complexed with the gate. The two-input AND gate is activated by the play inputs i_{11} and i_{22} to cleave the double end-labeled substrate. Source: (a,b) From Pei et al. [13]. © 2010. Reproduced with permission of Springer Nature.

The DNA-based computing paradigm enables the representation of computational elements, such as logic gates, as DNA complexes that directly interface with cell biology. This allows the use of biomolecules as direct, not symbolic, inputs that can be interpreted and processed algorithmically to yield either symbolic output such as fluorescence for diagnostics or the output of molecules to directly interact with a biological system as treatment. To this end, modular components of potentially very large systems have been developed including input/output modules that react to certain biomolecules or other environmental stimuli [14–20] and DNA-based information processing modules capable of implementing logical algorithms [11–13, 21–27]. The use of DNAzyme walkers is attractive because of their ability to function as a modular component of a DNA-based computing system, such as those outlined above, while also exhibiting catalytic behavior that plays an important role in its locomotion, which will be discussed in Section 15.3.

15.3 Experimental Implementations

DNA walkers require a surface with attachment sites made of complementary DNA strands that are situated such that the walker can step between them. The surface itself can be made of DNA, which is useful for creating tracks with regular lattice spacing via DNA origami [28] or self-assembled tiles [29]. Other tracks such as DNA–AuNP conjugates are able to contain a high density of protruding DNA sites, although the sites are randomly spaced, which may or may not be of concern depending on the application [30]. The walking mechanism can be either autonomous or nonautonomous and in combination with the track design may yield processive motion (i.e. continuous directed motion) or a random walk.

Processive walking mechanisms generally employ irreversible reactions, such as DNAzyme cleavage [31], restriction endonuclease activity [5], or strand displacement [32], to destroy sites as they are visited on a linear track (1D lattice) so that the walker can only move in a given direction. Nonautonomous processive mechanisms require human interference for the purpose of adding input signals, such as DNA signals [33] or UV/visible light irradiation [34], that direct the motion of the walker at each step along a 1D track. Each step is irreversible once the system has equilibrated and requires further input from the user to continue its motion. A walker that cannot step back to previously visited sites is said to follow the "burnt-bridge" model. The processive motion of a burnt-bridge walker on a 1D track becomes a self-avoiding random walk once in two dimensions. Although this method renders the track unusable after a single traversal, DNAzyme implementations have found applications where one-time use is sufficient, such as target-initiated release of fluorescently labeled DNA from a nanoparticle (random walk) [30, 35] or one-time cargo transport on a linear track (processive) [36].

Gu et al. use input oligonucleotides to direct the activity of a trigonal DNA walker that will selectively pick up cargo from the track [3]. They show that with appropriate signals any combination of three cargoes can be picked up by the
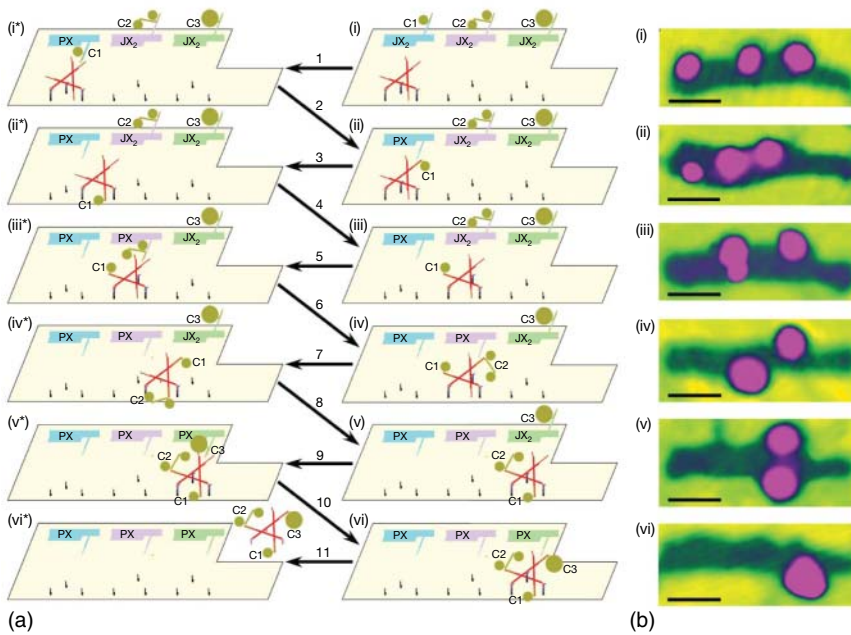
Figure 15.2 The molecular assembly line and its operation. (a) The basic components of the system are the origami tile (shown as a tan outline), programmable two-state DNA machines inserted in series into the file (shown in blue, purple, and green), and the walker (shown as a trigonal arrangement of DNA double helices in red). The cargo of the machines consists of a 5 nm gold particle (C1), a coupled pair of 5 nm particles (C2), and a 10 nm particle (C3) (indicated by green-brown dots), and their state can be labeled as PX (meaning ON or "donate" cargo) and JX_2 (meaning OFF or "do not donate" cargo). In the example shown, the walker collects cargo from each machine. (b) Atomic force micrographs of the system corresponding to the process steps sketched as states (i)–(vi) in (a). Atomic force microscopy (AFM) was performed by tapping in air; this mode of AFM results in only the nanoparticles and the origami being visible, and the individual nanoparticle components are not individually resolved. Owing to the washing procedures between steps, the AFM images are not of the same individual assembly line. Scale bars, 50 nm. Source: (a,b) From Gu et al. [3]. © 2010. Reproduced with permission of Springer Nature.

walker, resulting in eight possible final products, where a product is defined as a walker with a unique combination of attached cargoes. This system is able to reliably process input biomolecules that direct the stepwise assembly process of the walker (Figure 15.2).

Even in the simplest case of an unbiased random walk, the capability of DNA to encode information can lead to impressive behavior. For example, Thubagere et al. [37] implemented a diffusive walker consisting of a single strand of DNA that is able to sort dispersed cargo into designated locations. The walking domain of the strand is composed of two short foot domains on either side of a longer leg domain, and it walks by branch migration to adjacent sites, where the adjacent sites contain a complementary leg domain and only one foot domain. A secondary domain on the walking strand is able to pick up a fluorescent-labeled

cargo strand and drop it off at a designated goal site, where the pickup and drop-off occur via toehold mediated strand displacement. The multiple domains within this single strand encode multiple pieces of information, including which sites are available to step to, whether or not a cargo strand can be picked up, and where each cargo strand should be dropped off.

15.4 DNAzyme Walkers

DNAzyme walkers inherently modify the track on which they walk, simultaneously releasing DNA fragments from the track as the top half of the cleaved substrate unbinds from the upper recognition arm of the DNAzyme. The track may be designed such that the DNAzyme is effectively unable to rebind to previously cleaved substrate sites due to instability of the interaction between the DNAzyme and the cleaved substrate, resulting in a burnt-bridge walker. If, however, the DNAzyme can rebind to cleaved substrate sites, this results in rich dynamics dictated by the placement of cleaved and uncleaved sites as well as the geometrical constraints of the walker's legs.

The earliest implementation of a fully autonomous processive DNAzyme walker consists of a single DNAzyme on a one-dimensional track of four sites [31]. The DNAzyme consists of a 15-base (bottom) and a 7-base (top) recognition arm. Cleavage by the DNAzyme allows the complement of the 7-base recognition arm to dissociate from the complex, leaving the 15-base recognition arm still attached to the site. After dissociation of the 7-base complement, the 7-base recognition arm of the DNAzyme can attach to its complement on the next adjacent site, which leads to a branch migration of the DNAzyme to the new site and so on, as shown in Figure 15.3 [31]. Rate constants of migration and cleavage have been studied experimentally and characterized in terms of the catalytic core type, recognition arm lengths, and metal cation type and concentration [38]. This design has seen multiple recent applications. Cha et al. use the processive motion of the burnt-bridge DNAzyme walker to achieve cargo transport along a 1D track on a carbon nanotube [36]. Liu et al. use molecular logic to initiate the motion of an immobile walker on a gold nanoparticle in the presence of target miRNA in living cells, where the cleavage of substrate releases a fluorescently labeled strand that is monitored in real time [35] (Figure 15.4). Blanchard et al. [6] use a silica nanoparticle as the body for a highly polyvalent DNAzyme walker, consisting of thousands of hybridized DNAzymes. In this design, many walker legs attach to sites protruding from a gold substrate and can sustain over 100 pN of force as the body rolls about the surface cleaving substrate.

Another class of DNAzyme walkers called "spiders" has an inert body with multiple DNAzyme legs that walk on 2D tracks [39]. Unlike the burnt-bridge walkers, they are designed so that the DNAzyme cleavage permanently modifies the sites that have been visited by the walker while leaving behind a site that can still be visited by the walker. The cleavage of substrate leaves behind a shorter product strand that binds less stably to the walker leg and therefore reduces its mean residence time, creating an asymmetry between visited and unvisited sites that can

Figure 15.3 Scheme of a walking DNAzyme and its track. (a) The walking principle. (b) A construction where the walking DNAzyme is at one end of its track. Black lines: template (T); green lines: substrate (S); red/yellow lines: a 10–23 DNAzyme (red), with the catalytic core highlighted (yellow); blue dots indicate the bonds to be cleaved by the DNAzyme. Source: (a,b) From Tian et al. [31]. © 2005. Reproduced with permission of John Wiley & Sons. (Image courtesy of Chengde Mao.)
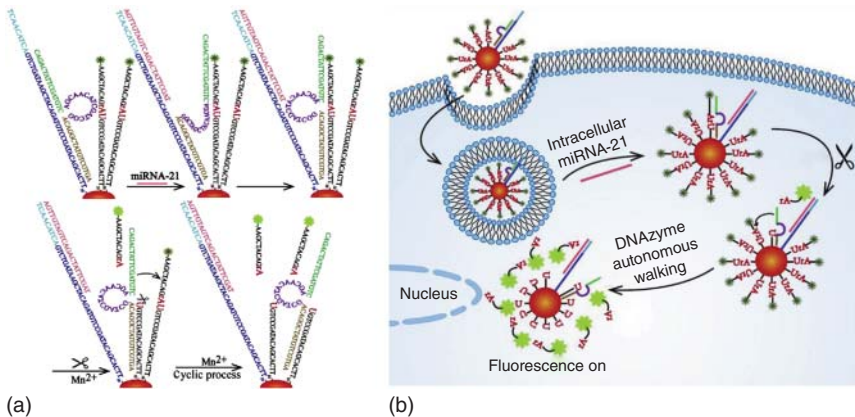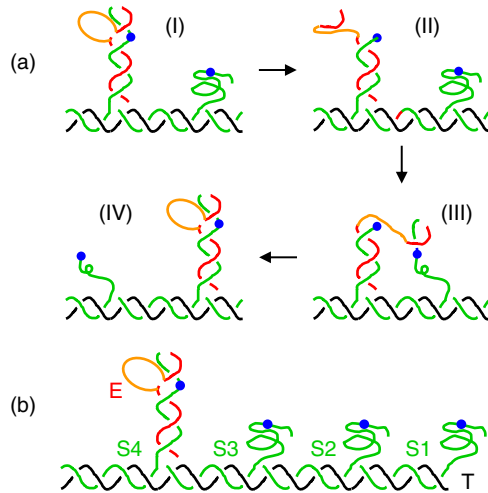
Figure 15.4 Domain-level (a) and schematic (b) representations of the miRNA-triggered DNAzyme. The DNAzyme is initially bound to a blocker strand. It is unable to cleave the blocker strand due to the lack of RNA-base at the DNAzyme core interface. The presence of miRNA-21 initiates branch migration of the walker by competing for the top half of the blocker strand, which is bound to the upper recognition arm of the DNAzyme. The DNAzyme then migrates to an adjacent fluorescently labelled substrate. The presence of metal ion cofactor (Mn^{2+}) induces cleavage of the substrate. Then the fluorescently labelled top half of the substrate may stochastically unbind from the DNAzyme's top recognition arm, allowing the DNAzyme to migrate to another adjacent substrate while releasing a fluorescent molecule from the complex. Source: From Liu et al. [35]. © 2019. Reproduced with permission of Elsevier.

be exploited to achieve a biased random walk [39, 40]. Although not completely processive, this paradigm grants the walker the ability to revisit sites while still enjoying periods of biased directional motion. Lund et al. employ this method to show that the motion of the walker can be guided by the landscape of its track via DNA sequence complementarity and that it can reach its goal on average faster than a purely diffusive walker [41]. The increased processivity of the molecular

spider over purely diffusive walkers has become a subject of theoretical analysis and computational simulation, as seen in Section 15.5.

15.5 Statistical Mechanics and Simulation

The DNAzyme spider has attracted additional attention from the field of statistical physics, as the characteristics of its biased random walk are an interesting system to analyze and model. Initial analyses pose the problem that the biased random walk is doomed to degenerate to an ordinary random walk, as its stochastic nature inevitably results in detachment from new substrate sites [40, 42]. This has resulted in a search for design parameters that may be exploited to increase the duration of transient superdiffusive motion of the spider, including optimization of the cleavage rate [43], polyvalency of the walker [44], and the cooperative effect of multiple walkers [45, 46]. While experimental researchers continue to find applications for the non-optimized DNAzyme walker such as target detection via product release [35], the computer science perspective promises design optimizations that may lead to new applications. For example, the burnt-bridge mechanism for processive motion does not allow multiple traversals of its track; however, an optimized spider may be able to exhibit effectively processive motion along a path and then return to its starting position.

The burnt-bridge self-avoiding random walk in one dimension is a simple solution to the challenge of designing an autonomous walker capable of continuous directed motion; however, optimizing a system for maximum superdiffusive motion with biased random walk remains nontrivial if we grant the walker the ability to revisit sites or to move on a two-dimensional track. Theoretical analysis and coarse-grained simulations of walkers provide insight that is useful for designing more complex systems such as on narrow strips or in two dimensions.

The starting point for theoretical analysis of a walker that is able to revisit sites after cleaving them (the spider) is the case of a single spider on a one-dimensional track. Antal and Krapivsky [40] perform this analysis for the case of a single spider consisting of one or more legs on a regular one-dimensional track. In their model, legs may only hop to adjacent sites and are unable to overtake one another in the case of multiple legs. They consider an asymmetric track with only visited (product) sites to the left of the origin and only unvisited (substrate) sites in the positive direction. They derive the probability for a spider to take a step in either direction when its front (or only) leg is attached to the leftmost substrate (i.e. at the boundary of visited and unvisited sites) as a function of the ratio of dwell times at product vs. substrate sites $r < 1$. They show that the maximal bias for a two-legged spider to step in the positive direction is $p_+ = 5/8$ in the limit $r \to 0$, while for a one-legged spider this probability is $p_+ = 1/2$ for all r values. Semenov et al. [42] simulate this system for a two-legged spider and develop a framework for understanding the behavior of this model. They propose that the spider exists in one of two metastases, either the boundary state (B) in which the spider is at the boundary between visited and unvisited sites and
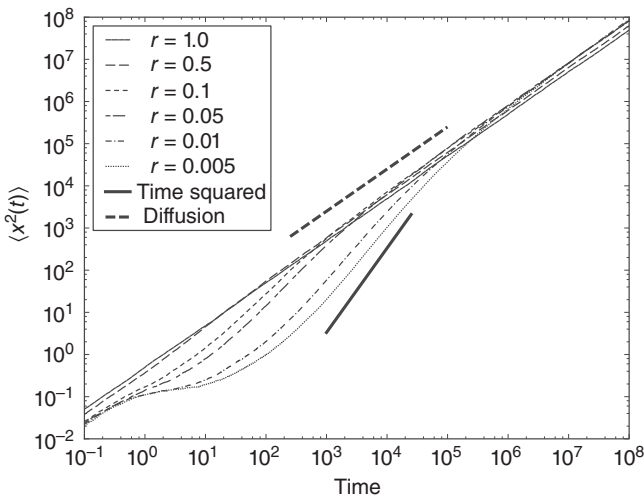
Figure 15.5 Mean squared displacement, $\langle x^2(t) \rangle$. Source: From Semenov et al. [42]. © 2011. Reproduced with permission of American Physical Society.

is biased in the forward direction or the diffusive state (D) in which the spider has moved away from the boundary and is performing an unbiased random walk through the contiguous region of product sites. When initialized on a symmetrical infinite track of substrates, the spider cleaves a contiguous region of product, termed the product sea. The average motion of the spiders is superdiffusive to begin and deteriorates toward ordinary diffusion as the product sea becomes larger and the spiders take increasingly large amounts of time on average to find the substrate boundary from the D state (Figure 15.5). The two-legged spider on a 1D track is only slightly biased toward the region of new substrates, and as the product sea grows, so does the time spent in the diffusive state. The major corollary of these findings is that the motion of a spider that does not employ the burnt-bridge method will not reliably exhibit superdiffusive directional motion. This poses the question: how can a system be designed to minimize the time spent by the spider in the diffusive state and increase its bias toward regions of new substrate?

Semenov et al. [45] explore the interaction between multiple spiders on the 1D track. They ask whether they can overcome the eventual decay to superdiffusive motion due to a growing product sea by injecting spiders at the origin, thus effectively shrinking the size of the product sea by filling it with new spiders. They show that superdiffusive motion is increased initially but nonetheless decays toward ordinary diffusion in the long time limit. Semenov et al. explore the first passage properties of a two-legged spider in the 2D plane [43]. They simulate the mean time a spider takes to reach various goals in 2D. They consider the mean time a spider takes to reach a circular boundary when starting in the center and the reverse situation of starting at the edge of the circle searching for a goal site in the center. They find that an optimal cleavage rate minimizes the mean first passage time of a spider to its goal. Rank et al. [46] show that teams of two-legged
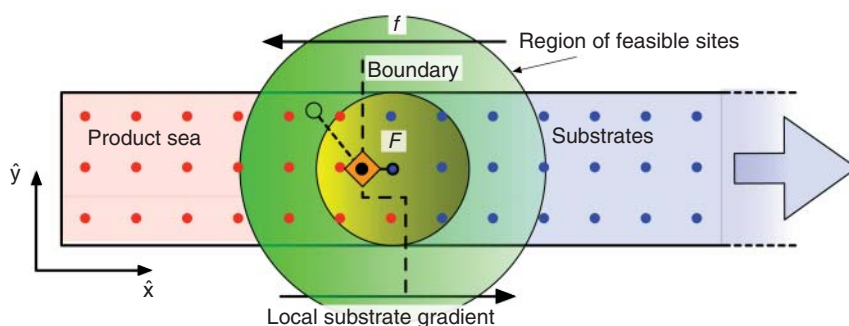
Figure 15.6 The irreversible catalysis of substrates to products leads to the emergence of a spatial asymmetry in substrate concentration at the *boundary* between the contiguous *product sea* and the contiguous region of unvisited substrates. A walker with $k_{cat} < 1s^{-1}$ has a residence time bias where leg–substrate binding durations are much longer than for leg–product bindings. Thus, over time, legs are more likely to be attached to local substrates than local products, not because they seek out substrates, but because legs attached to products quickly detach. Hence, walkers are effectively driven in the direction of greatest local substrate density, and near the boundary this is always in the $+\hat{x}$ direction. The irreversibility of substrate catalysis means the boundary itself also moves in the $+\hat{x}$ direction, causing walkers near the boundary to move ballistically away from the origin. Source: From Olah et al. [44]. © 2013. Reproduced with permission of American Physical Society.

spiders on their own independent 1D tracks exhibit greater superdiffusive motion when tethered together than when moving independently. Olah and Stefanovic simulate the motion of a single spider on a thin 2D strip walking against a force [44]. In this more detailed model, the position of the spider body is modeled as a Boltzmann distribution, which is affected by a force in the negative direction. This model shows that a three-legged spider on a track three sites wide can achieve superdiffusive motion against forces up to 4 pN, on the same order as the stall force of kinesin (Figure 15.6). An even more detailed computational model by Ouldridge et al. [47] explores the thermodynamic and kinematic molecular interactions of leg binding and unbinding, providing insight into nonequilibrium behavior of the walker.

15.6 Conclusions

While great progress has been achieved in the development of DNAzyme-based robotics, there is a disconnect between theory and experiment that could provide future direction for larger-scale experiments. Statistical physics has provided a framework for thinking about how to improve the superdiffusive transient of DNAzyme spiders, but these insights have not been verified experimentally nor have they guided subsequent experimental design. Furthermore, simulations have characterized only a few geometries including spiders on a 1D track [40, 42, 45], the single bipedal spider in 2D [43], the three-legged spider on a 2D strip [44], and multiple tethered bipedal spiders on independent 1D tracks [46]. There exists a massive unexplored design space for other geometric constraints
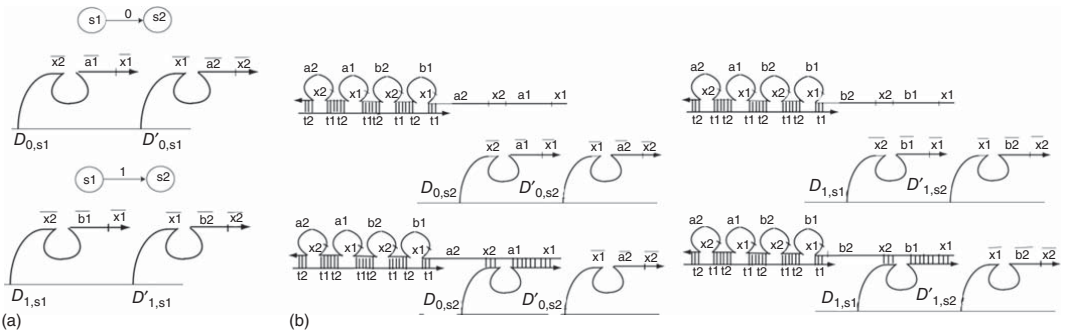
Figure 15.7 (a) Implementation of a state transition through DNAzymes. (b) D_{0,s_1} in the transition machinery for state transition at 0 combines with input nanostructure when the active input symbol encoded by the sticky end is 0. When the active input symbol encoded by the sticky end is 1, D_{0,s_1} in the transition machinery for state transition at 1 combines with the input nanostructure. Source: From Reif et al. [51]. © 2009. Reproduced with permission of Elsevier.

regarding walker leg lengths, attachments between walker bodies, and site variation on the track. It would be interesting to employ emerging techniques in computer science to search this design space to either optimize superdiffusive directional motion or to find other potentially useful emergent behavior of stochastic walkers.

Proposals for completely different applications of DNAzyme walkers have been developed and not yet implemented experimentally. Mo et al. [48] use DNAzyme walkers to improve upon previous designs [49, 50] of walkers that navigate tracks, evaluating Boolean circuits as they walk. They implement basic logic gates using spiders that are tagged with binary values that compete to reach an output location on the track composed of functional sites that may block spiders with certain binary values from progressing. In contrast to a walker with DNAzyme legs modifying a track, Reif and Suhu [51] propose a design for a DNAzyme-based finite state automaton (FSA), composed of a DNA nanostructure that is modified as it walks along a track of DNAzyme sites (Figure 15.7). They propose multiple applications for the FSA such as a "DNAzyme doctor" and a DNAzyme router. The "DNAzyme doctor" walks along a series of DNAzyme sites, each representing a logical AND necessary to diagnose a disease state. If all conditions are met, the walker will progress to the end of its track, releasing a signal that results in appropriate drug delivery. The DNAzyme router is composed of an FSA with different DNAzymes located at sites on a 2D tile lattice. The walking nanostructure can then be designed to follow any path along the sites in the lattice. The attractiveness of DNAzymes in nanoscale robotics lies in their simplicity and easily controllable catalytic activity. Without the requirement of natural proteins and enzymes, the dynamics of Watson–Crick base pairing combines the utility of catalytic activity with information encoding and processing, an indispensable tool for the future of DNA nanotechnology. With this utility, many other theoretical implementations of DNAzymes in nanoscale robotics will be envisioned, and we should strive to bring these visions to life.

References

1 Ait-Haddou, R. and Herzog, W. (2003). *Cell Biochem. Biophys.* 38: 191–214.

2 Geeves, M.A. (2016). *Biopolymers* 105: 483–491.

3 Gu, H., Chao, J., Xiao, S.-J., and Seeman, N.C. (2010). *Nature* 465: 202–205.

4 Bath, J., Green, S.J., and Turberfield, A.J. (2005). *Angew. Chem. Int. Ed.* 44: 4358–4361.

5 Yin, P., Yan, H., Daniell, X.G. et al. (2004). *Angew. Chem. Int. Ed.* 43: 4906–4911.

6 Blanchard, A.T., Bazrafshan, A.S., Yi, J. et al. (2019). *Nano Lett.* 19 (10): 6977–6986.

7 Xing, Y., Liu, B., Chao, J., and Wang, L. (2017). *RSC Adv.* 7: 47425–47434.

8 Chen, J., Luo, Z., Sun, C. et al. (2019). *TrAC, Trends Anal. Chem.* 120: 115626.

9 Santoro, S.W. and Joyce, G.F. (1997). *Proc. Natl. Acad. Sci. U.S.A.* 94: 4262–4266.

10 Gu, H., Furukawa, K., Weinberg, Z. et al. (2013). *J. Am. Chem. Soc.* 135: 9121–9129.

11 Stojanovic, M.N. and Stefanovic, D. (2003). *Nat. Biotechnol.* 21: 1069–1074.

12 Macdonald, J., Li, Y., Sutovic, M. et al. (2006). *Nano Lett.* 6: 2598–2603.

13 Pei, R., Matamoros, E., Liu, M. et al. (2010). *Nat. Nanotechnol.* 5: 773–777.

14 Green, E., Olah, M.J., Abramova, T. et al. (2006). *J. Am. Chem. Soc.* 128: 15278–15282.

15 Pei, R., Shen, A., Olah, M.J. et al. (2009). *Chem. Commun.* 22: 3193.

16 Chen, C., Pu, F., Huang, Z. et al. (2011). *Nucleic Acids Res.* 39: 1638–1644.

17 Liu, C., Sheng, Y., Sun, Y. et al. (2015). *Biosens. Bioelectron.* 70: 455–461.

18 Zhang, Z., Balogh, D., Wang, F., and Willner, I. (2013). *J. Am. Chem. Soc.* 135: 1934–1940.

19 Lu, C.-H. and Willner, I. (2015). *Angew. Chem. Int. Ed.* 54: 12212–12235.

20 Poje, J.E., Kastratovic, T., Macdonald, A.R. et al. (2014). *Angew. Chem. Int. Ed.* 53: 9222–9225.

21 Stojanovic, M.N., Mitchell, T.E., and Stefanovic, D. (2002). *J. Am. Chem. Soc.* 124: 3555–3561.

22 Stojanovic, M.N., Semova, S., Kolpashchikov, D. et al. (2005). *J. Am. Chem. Soc.* 127: 6914–6915.

23 Lederman, H., Macdonald, J., Stefanovic, D., and Stojanovic, M.N. (2006). *Biochemistry* 45: 1194–1199.

24 Banda, P., Teuscher, C., and Lakin, M.R. (2013). *Artif. Life* 19: 195–219.

25 Banda, P., Teuscher, C., and Stefanovic, D. (2014). *J. R. Soc. Interface* 11: 20131100.

26 Brown, C.W., Lakin, M.R., Horwitz, E.K. et al. (2014). *Angew. Chem. Int. Ed.* 53: 7183–7187.

27 Brown, C.W., Lakin, M.R., Stefanovic, D., and Graves, S.W. (2014). *ChemBioChem* 15: 950–954.

28 Rothemund, P.W.K. (2006). *Nature* 440: 297–302.

29 Winfree, E., Liu, F., Wenzler, L.A., and Seeman, N.C. (1998). *Nature* 394: 539–544.

30 Yang, K., Wang, H., Ma, N. et al. (2018). *ACS Appl. Mater. Interfaces* 10: 44546–44553.

31 Tian, Y., He, Y., Chen, Y. et al. (2005). *Angew. Chem. Int. Ed.* 44: 4355–4358.

32 Omabegho, T., Sha, R., and Seeman, N.C. (2009). *Science* 324: 67–71.

33 Sherman, W.B. and Seeman, N.C. (2004). *Nano Lett.* 4: 1203–1207.

34 Cheng, J., Sreelatha, S., Loh, I.Y. et al. (2014). *Methods* 67: 227–233.

35 Liu, C., Hu, Y., Pan, Q. et al. (2019). *Biosens. Bioelectron.* 136: 31–37.

36 Cha, T.-G., Pan, J., Chen, H. et al. (2014). *Nat. Nanotechnol.* 9: 39–43.

37 Thubagere, A.J., Li, W., Johnson, R.F. et al. (2017). *Science* 357: eaan6558.

38 Cha, T.-G., Pan, J., Chen, H. et al. (2015). *J. Am. Chem. Soc.* 137: 9429–9437.

39 Pei, R., Taylor, S.K., Stefanovic, D. et al. (2006). *J. Am. Chem. Soc.* 128: 12693–12699.

40 Antal, T. and Krapivsky, P.L. (2007). *Phys. Rev. E: Stat. Nonlinear Soft Matter Phys.* 76: 021121.

41 Lund, K., Manzo, A.J., Dabby, N. et al. (2010). *Nature* 465: 206–210.

42 Semenov, O., Olah, M.J., and Stefanovic, D. (2011). *Phys. Rev. E* 83: 021117.

43 Semenov, O., Mohr, D., and Stefanovic, D. (2013). *Phys. Rev. E* 88: 012724.

44 Olah, M.J. and Stefanovic, D. (2013). *Phys. Rev. E* 87: 062713.

45 Semenov, O., Olah, M.J., and Stefanovic, D. (2013). *Nat. Comput.* 12: 259–276.

46 Rank, M., Reese, L., and Frey, E. (2013). *Phys. Rev. E* 87: 032706.

47 Ouldridge, T.E., Hoare, R.L., Louis, A.A. et al. (2013). *ACS Nano* 7: 2479–2490.

48 Mo, D., Lakin, M.R., and Stefanovic, D. (2016). *Biosystems* 146: 10–25.

49 Dannenberg, F., Kwiatkowska, M., Thachuk, C., and Turberfield, A.J. (2015). *Nat. Comput.* 14: 195–211.

50 Wickham, S.F.J., Bath, J., Katsuda, Y. et al. (2012). *Nat. Nanotechnol.* 7: 169–173.

51 Reif, J.H. and Sahu, S. (2009). *Theor. Comput. Sci.* 410: 1428–1439.

16

DNA Origami Transformers

Reem Mokhtar, Tianqi Song, Daniel Fu, Shalin Shah, Xin Song, Ming Yang, and John Reif

Duke University, Department of Computer Science, LSRC Building D101, 308 Research Drive Campus Box 90129, Durham, NC 27708-0129, USA

16.1   Introduction

The field of DNA nanoscience [1] and subfield of structural DNA nanotechnology have enormously benefited from the DNA origami technique [2]. The discovery of the DNA origami technique has allowed researchers to assemble a wide variety of 2D and 3D nanostructures with arbitrary complex shapes with moderate to high yield. It uses a long single strand of DNA (typically a circular strand extracted from a bacteriophage) termed a "scaffold," self-assembled into a predetermined shape using short complementary single strands, known as "staples," which target different regions on the scaffold strand to bring them together and to form stabilizing crossovers. If well designed, the resulting self-assembled DNA origami is a thermodynamically stable structure [3], due to the inherent stability of double-stranded DNA [3–5], the careful placement of crossovers in the nanostructure [2, 6], and electrostatic interactions.

While heralded as an innovative technique for building nanostructures, there are few forays into the computational capabilities of the structures themselves. Despite that its partnered subfield, DNA computing, also unconventionally utilizes nucleic acids, the two remain mostly disjoint subfields of DNA nanoscience. At their narrow union, basic DNA nanostructures, such as rectangles constructed by DNA origami, act as templates for organizing computing components that still function according to traditional DNA nanoscience applications, such as nanophotonic imaging [7, 8], DNA walkers [9, 10], or spatially localized DNA computation on DNA nanostructures [11, 12] or cell membranes [13]. We suggest that it should be possible for DNA nanostructures to become more closely intertwined with strand-based DNA computing to process and perform mechanical actions or transduce states into physical forms, effectively as finite state machines.

In this chapter, we present work that develops a novel paradigm of DNA origami transformers, which are DNA origami nanostructures that can dynamically alter their conformation. We will review existing DNA nanostructures with

focus on evaluating their physical states. Biophysical models of DNA origami self-assembly are then discussed to provide context on how they motivate design primitives for DNA origami transformations. We then subsequently discuss the design of *zip* and *unzip* primitives that are based upon toehold-mediated strand displacement and then show how these techniques can also be extended to use strand-displacing polymerase and hairpin architectures for unbinding of domains. Zip and unzip primitives are also experimentally demonstrated upon a simple rectangular DNA origami. Finally, the chapter reviews potential applications of these techniques.

The current focus of most research into dynamic DNA origami nanostructures is limited to modular (rearrangement in relative positioning of static pieces) or mechanical ("loose" dynamically designed parts that respond to external stimuli) approaches to DNA origami transformations. Only some [14–17] prior work on DNA origami has investigated general methods for the design of dynamic DNA origami nanostructures that transform into different structures under programmed control. While designed structures have demonstrated a wide range of shape complexity, most lack dynamic movement or have limited degrees of freedom. To describe some of the limitations of existing DNA nanostructures, we introduce them as *static* or *bi-dynamic*. Static nanostructures [18–27] only have one distinct conformation, but these may be used to create dynamic superassemblies, such as by tiling or attachment of guest molecules including nanoparticles or other nanostructures [9, 21, 28–36]. Others may have limited or fixed range of movement, such as walkers [9, 10], and a set of structures that are bi-dynamic [15, 37–40], meaning they can be transformed between various states that each keep the same overall conformation of the nanostructure.

A DNA origami nanostructure can be viewed as *static* if it has only one distinct conformation. In prior work, "static DNA origami nanostructures" have been used as components of more complex nanoassemblies:

(i) As a breadboard [9, 28, 33], which is a static substrate used to spatially arrange molecules at specific locations on its surface. This technique has also been used to arrange nanoparticles in specific patterns, such as chiral assemblies that generated chiral dichroism [41] and to form plasmonic structures [42] with enhanced Raman signal between nanoparticles affixed close together on the DNA origami, as well as similar uses with carbon nanotubes [43] and metallic nanoparticles [29–32, 34–36].

(ii) As a static substrate where dynamic nanodevices (e.g. DNA walkers [9, 10] and DNA nanomanufacturing devices [28]) are attached. This provides a track for these dynamic nanodevices to traverse, via stators spaced out on a DNA origami surface, or to perform simple programmable tasks.

(iii) As components in larger tiling nanoassemblies (e.g. see "DNA tiles" [21] for more details), and strand displacement reactions have been used to carve out sections of larger tiling nanoassemblies (see "DNA bricks" [20] for more details, which use an alternative construction technique than DNA origami).

A DNA origami nanostructure can be viewed as *bi-dynamic* if it can be transformed between a single static DNA origami nanostructure and one or more

distinct static DNA origami nanostructures that are subcomponents of the original DNA origami nanostructure. In prior work, bi-dynamic DNA origami have been used as components of more complex nanoassemblies. Bi-dynamic DNA origami nanostructures have functioned as containers [40] with hinged lids using strand displacement reactions to transform between a single static DNA origami nanostructure (a closed container) and two static DNA origami nanostructures (a container with an open lid). For example, Torelli et al. [37] demonstrated a system where contact with a DNA target strand induces a localized strand displacement reaction that opens a flap that releases a molecular cargo. Bi-dynamic DNA origami nanostructures have functioned as payload carriers [39], making use of hybridization reactions to capture components or strand displacement reactions to drop components.

Other prior work has demonstrated various restricted transformations of DNA origami: Zhang et al. [16] were able to reconfigure and alternate between quasi-fractal patterns within a large DNA origami rectangular frame. Suzuki et al. and Yang et al. demonstrated the use of azobenzene-modified DNA to photo-regulate the assembly and disassembly of DNA origami units together [44, 45]. Zhou et al. [46] tested bending geometries and the mechanical functionality of DNA origami devices. Mitchell et al. [47] showed that a flat sheet of DNA origami could be rolled up into a nanotube and this transformation has since been commonly used to organize gold nanoparticles into chiral geometries [30, 48, 49]. This showed that the single layer of a DNA origami nanostructure can be used to induce dynamic structure reconfiguration. Chen et al. [50] showed that a rectangular DNA may be reversibly folded into the shape of a cylindrical tube by connecting two opposing sides via linker staple strands complementary to regions of the scaffold on the opposing ends. Unfolding occurs via toehold-mediated strand displacement using exposed toeholds on these linker strands. The folding and unfolding operations resulted in yields of 90–100%, respectively. Castro et al. [51] focused on designing nanostructures containing stiff and loose regions. Single strands of DNA were placed at locations adding flexibility between stiff parts or limiting the degrees of freedom, which the authors termed DOMs (DNA origami mechanisms). Using this mechanism, the authors were able to demonstrate the controllable bending of stiff arms of a DNA origami tweezer [46]. Han et al. [52] demonstrated that a DNA origami nanostructure in the shape of a Möbius strip could stretch to double its length. Using strand displacement reactions, staples with exposed toeholds were removed from the scaffold to let the strip extend from two bound "layers" of the Möbius strip to one longer strip. Gerling et al. [53] also demonstrated various 3D nanoassemblies composed of multiple 3D origami parts, where the attachments between parts (the parts are attached by base stacking) can be dynamically modified to provide for shape transformations of the nanoassemblies. The difference between all these prior works, and the work described here, is the use of fully transformable DNA origami nanostructures to perform total shape transformations. With the exception of these and the above cited bi-dynamic DNA origami demonstrations, most of the prior work on DNA origami has not been investigated with the purpose of developing general techniques for

dynamic DNA origami nanostructures that transform into different structures under programmed control, which is the focus of our work.

Expanding the presently known limitations of DNA origami shape change would be of great interest to molecular-scale robotics. Transformations on a local and global scale that are not necessarily modular nor mechanical should be investigated in order to expand the vocabulary of dynamism available to DNA origami and provide for previously unachievable capabilities of DNA origami in medical or nanorobotic applications. Structurally encoded computing, such as that inherent and envisioned in protein design, is still not fully realized at present with DNA origami. Much more flexible modes of interaction that are inherently encoded within the staple design of DNA origami nanostructures provide for an expanded set of capabilities to perform the processing and transmission of information on the nanoscale.

To do so, we must understand the biophysical models of DNA origami self-assembly that govern the invasion of staple strands into the scaffold strand during synthesis. There are considerable open questions on how to explain the relatively high yield and stability of assembled DNA origami nanostructures, as well as the specific procedure undertaken by the participating strands of the DNA origami self-assembly process.

Wei et al. used Förster resonance energy transfer (FRET) experiments to gain insight into DNA origami nanostructure stability. FRET pairs were placed at various points on Rothemund's rectangular origami nanostructure to understand the stabilizing effect that staple strand placement has, locally and globally [3]. It has been informally conjectured by a number of researchers, and in particular Arbona et al., that the process of DNA origami self-assembly (known as DNA origami folding), proceeding as the system cools, might consist of the basic phases illustrated in Figure 16.1.

Note that the figure illustrates a relatively simple example of a DNA origami nanostructure, consisting of two connected 2D rectangular DNA origami components, where, in each rectangular component, the scaffold strand layers are horizontally oriented (rasterized). Further note that along the length (in the horizontal direction) of each scaffold strand layer section, the staple strand's hybridization sites to the scaffold alternate in attachment either just above or just below in order to bring together disparate segments of the scaffold in this raster pattern. Note that at inner turns of the scaffold layer, the staple strands are hybridized to very proximate sections of the scaffold strand.

From the perspective of a single staple strand, it is conjectured by Arbona et al. that the incorporation of a staple strand onto the scaffold in DNA origami might undergo the following phases:

 (i) Initially the staple strand is not hybridized.
 (ii) There is an initial hybridization of the staple strand with a single complementary domain sequence of the scaffold strand.
(iii) Zippering: The staple strand makes a second hybridization to the scaffold at a contiguous location across a layer position of the scaffold strand, possibly via a strand displacement of another competing staple strand already hybridized at that second location of the scaffold. Note that this may occur earliest for
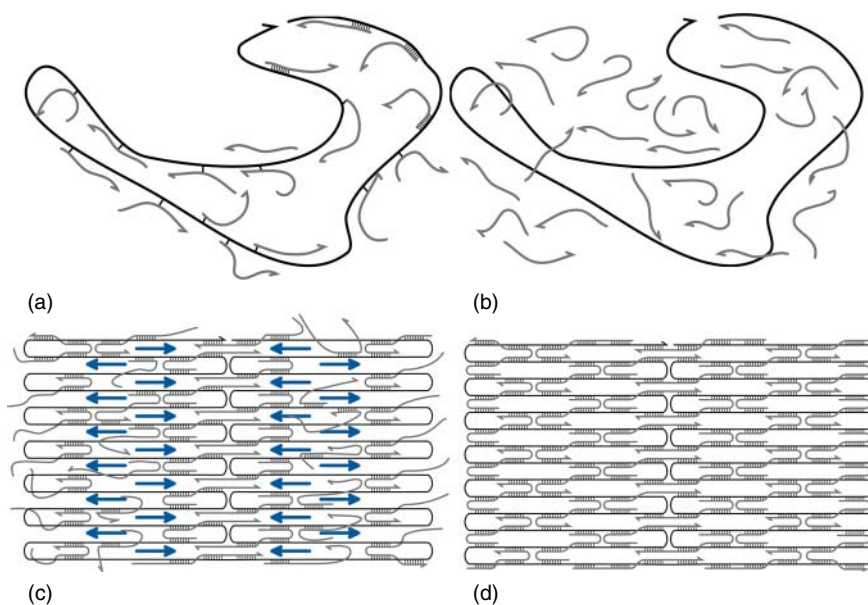
Figure 16.1 Conjectured self-assembly of DNA origami. (a) Phase 1: Synthesis begins with unbound scaffold and staples all diffusing freely in solution. (b) Phase 2: Staples begin to hybridize to the scaffold at single contiguous domains. (c) Phase 3: Staples begin to hybridize to secondary domains, preferring those near scaffold crossovers that are spatially nearby. (d) Phase 4: Synthesis completes through a continual, cooperative effect of hybridized staples spatially localizing the scaffold for remaining unbound staple to hybridize.

the staples closest to the scaffold crossovers, since the staple's hybridization sites are proximal to each other on the scaffold.

(iv) Further strand displacements and hybridizations involving a limited number of staples may occur to complete the self-assembly of the DNA origami nanostructure in the presence of significant steric hindrance and strain. Also note that a clean distinction of the phases is an idealization depending on location and melting temperature of staple strands.

On a different scale, from the perspective of the overall assembly, it is conjectured that DNA origami nanostructures might undergo the following phases:

(i) Initially, at high temperature, the scaffold and staple strands are (mostly) all single stranded.

(ii) Domains of individual staple strands hybridize to complementary sequences of the scaffold strand, so the scaffold strand is (nearly) fully hybridized, but nearly all the staples hybridized to the scaffold are only affixed at one domain of the staple strand (recall the staples are each designed to be eventually hybridized to two or more domains on the scaffold).

(iii) Zippering: Each layer of the scaffold strand is stitched together by staples in a zippering process, starting near the scaffold crossovers (where the staples have nearby pairs of hybridization sites on the scaffold).

(iv) Further (and not well understood) self-assembly may occur by additional shape transformations of sections of the DNA origami nanostructure in the presence of significant steric hindrance and strain.

Arbona et al. developed a model based on nearest-neighbor parameters [54] for the folding process of DNA origami nanostructures, modeling a zippering process (as briefly described above) in the assembly of a small DNA nanostructure that is a substructure of a full DNA origami nanostructure. This model considers the topology, sequence composition, and melting temperature of staples. Arbona et al. made experimental tests of the model using fluorescence data and found it characterized well many aspects of the first three phases of DNA origami folding but does not account for the final phase of DNA origami folding, which likely includes such issues as steric hindrance and strain. (Arbona's model of DNA origami folding and conjectured zippering process will have importance to our transformations of DNA origami.)

Next, we report on developed primitives whereby a DNA origami nanostructure may be transformed via a sequence of hybridization reactions from one stable structure to another stable structure with high yield and in a scalable fashion. We describe two basic transformation primitives, unzip and zip, which affect the partial disassembly (unzip) and reassembly (zip) of subset structures of a DNA origami nanostructure and can be performed repeatedly to achieve more complex 2D and 3D nanostructure transformations. Zip transformations can be executed by reactions where replacement staple strands hybridize to the scaffold, which may also introduce bends, twists, and other 3D shape changes. Unzip transformations can be executed by either a sequence of strand hybridization reactions, where extended staple strands are strand-displaced by replacement staple strands, or strand-displacing polymerase reactions. Zip and unzip techniques via toehold-mediated strand displacement are then experimentally demonstrated on small rectangular DNA origami and imaged via atomic force microscopy (AFM).

16.2 Design

In 2D DNA origami, the scaffold is typically folded into a number of relatively flat substructures, each of which will be termed a "sheet." Each sheet represents two "layers" of DNA origami or two regions of the scaffold intended to align together via staple strands. This is configured similar to a protein beta sheet. Within all sheets, the scaffold winds in a series of adjoining layers. In the interior of a sheet, each layer L adjoins two other layers (say, L' and L''), and so the staples attached to layer L are alternatively also attached L' or L''.

Two basic transformation primitives (zip and unzip) are developed with the intent that these can be performed repeatedly to achieve more complex transformations of DNA origami. These basic transformation primitives affect the partial disassembly (unzip) and reassembly (zip) of a sheet layer of DNA origami nanostructures containing a single scaffold layer, conceptually illustrated in Figure 16.2. Strand displacement reactions govern the addition or removal of
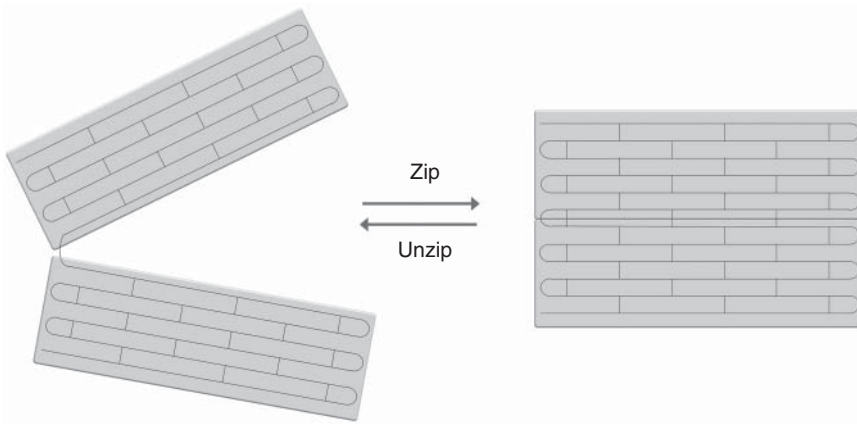
Figure 16.2 Simplistic illustration of primitives: a conceptual illustration of reversible transformation operating on a simple rectangular DNA origami with top and bottom layers is shown. Zip primitives join two separated pieces. Unzip primitives split two joined pieces.
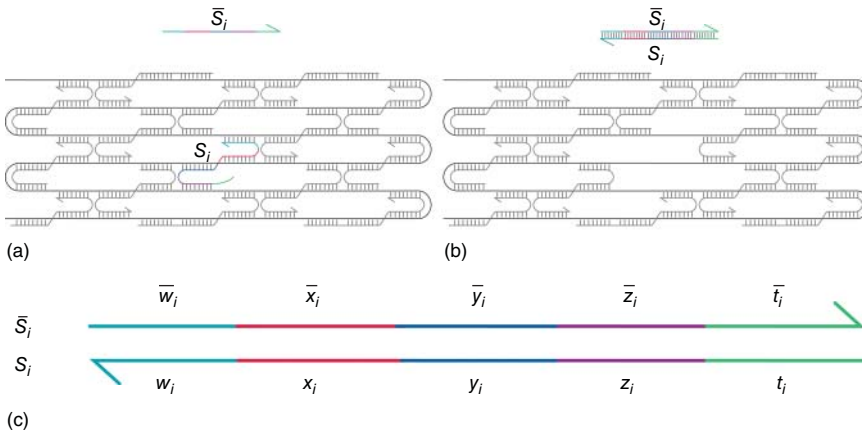
Figure 16.3 Strand displacement reaction. (a) Before strand displacement. (b) After strand displacement. (c) Staple strand s_i and complement $\overline{s_i}$. A structure (a) may transition to structure (b) via a strand displacement reaction of strand $\overline{s_i}$ that invades the structure to react with and remove strand s_i. Duplex waste with domains (c) is formed.

structure-defining staple strands. Each strand s_i along the transformation border between origami layers L and L' has domains w_i, x_i, y_i, z_i that hybridize to the scaffold and t_i for initiating toehold-mediated strand displacement reactions. Complementary strand $\overline{s_i}$ has similar regions to remove the strand as duplex waste from the structure during a transformation (see Figure 16.3).

Zip is a basic primitive transformation inspired by Arbona's biophysical model [55] of DNA origami assembly. In the zip transformation (Figure 16.4), a pair of disjoint (not connected via staple strands) scaffold layers L and L' of the partially assembled DNA origami are stitched together by the hybridizations to segments of a sequence of k staples $s_1, \ldots, s_k$. The scaffold layer L has a sequence of domains
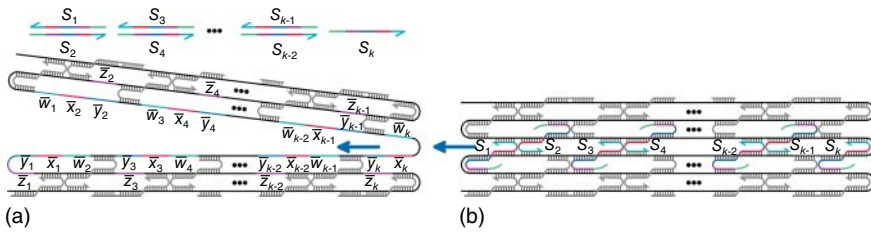
Figure 16.4 Zip transformation. (a) Before zip. (b) After zip. Strands $s_1, ..., s_k$ complementary to sections between layers L and L' are introduced into the system. Hybridization completes the assembly of the structure to form (b).

$\overline{w}_1, \overline{x}_2, \overline{y}_2, ..., \overline{w}_{k-1}, \overline{x}_k, \overline{y}_k$, which are initially unhybridized, and scaffold segment L also has a sequence of domains $\overline{y}_1, \overline{x}_1, \overline{w}_2, ..., \overline{y}_k, \overline{x}_k$, which are also initially unbound. For each $i = 1, ..., k$, the staple s_i has unbound domains complementary to w_{i-1}, x_i, and y_i to which it can hybridize. After these staple hybridizations are completed, the result of the zip transformation is a DNA origami nanostructure containing the adjoined pair of scaffold layer segments L and L'. By adjusting the locations of the hybridization sites of the staples, zip operations allow two initially disconnected layers of DNA origami to be adjoined either in parallel or at given angles (e.g. at right angles). The zip operation can be implemented by use of staples that have further domains to hybridize with more than two scaffold segments.

Unzip as a second basic primitive transformation of DNA origami was inspired by considering the reverse of Arbona's biophysical model [55] of DNA origami assembly, specifically in the case where an already assembled DNA origami is slowly heated and the structure slowly melts. The unzip transformation is essentially the reverse of the zip transformation and uses a sequence of isothermal toehold-mediated strand displacement reactions to remove staples with an extended toehold binding domain.

The unzip transformation (Figure 16.5) separates a pair of adjoined scaffold layers L and L', which were originally stitched together by existing hybridizations of segments of a sequence of k original staples $s_1, ..., s_k$. The scaffold layer L has a
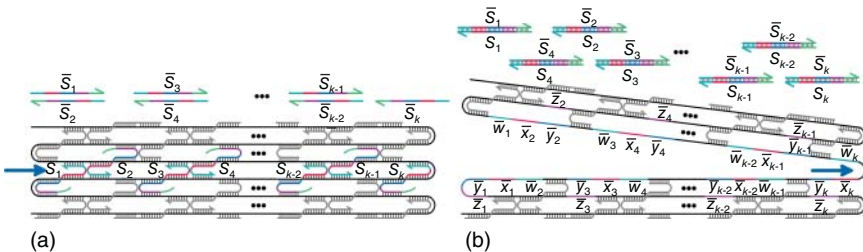
Figure 16.5 Unzip transformation. (a) Before unzip. (b) After unzip. Strands $\overline{s}_1, ..., \overline{s}_k$ complementary to already placed staple strands $s_1, ..., s_k$ are added to the system. A toehold domain on displacing strands hybridize to a matching domain on placed strands initiating a toehold-mediated strand displacement reaction that removes placed strands to form duplex waste and structure (b).

sequence of domains $\overline{w_1}, \overline{x_2}, \overline{y_2}, \ldots, \overline{w_k}$, and scaffold segment L' also has a sequence of domains $\overline{y_1}, \overline{x_1}, \overline{w_2}, \ldots, \overline{y_k}, \overline{x_k}$. For each $i = 1, \ldots, k$ the original staple s_i has domains complementary to w_i and y_i, which are initially hybridized to L and L' at these sites. Also the staple s_i is designed to have at one of its ends an exposed toehold domain t_i (which will be used by the unzip's toehold-mediated strand displacement reaction).

A displacing strand $\overline{s_i}$ is designed to have domains that are complementary to t_i, z_i, y_i, x_i, w_i. The exposed toehold t_i of original staple s_i is easily accessible when s_i is hybridized to the scaffold strand. The introduction of the displacing strand $\overline{s_i}$ will displace s_i by a toehold-mediated strand displacement reaction. For example, the original staple may be $s_i = t_i z_i y_i x_i w_i$, and the displacing strand may be $\overline{s_i} = \overline{t_i z_i}\, \overline{y_i}\, \overline{x_i}\, \overline{w_i}$. The hybridization of the original staple s_i with the displacing strand $\overline{s_i}$ strips the domain x_i of scaffold L and the domain y_i of scaffold L'. The result of the unzip transformation is two subset structures of a DNA origami nanostructure where scaffold layer segments L and L' are no longer adjoined due to the strand displacement of the staples $s_1, \ldots, s_k$ (originally in place) by the displacing strands $\overline{s_1}, \ldots, \overline{s_k}$.

The zip and unzip transformations may also be completed by DNA hairpin architectures. DNA hairpins are stem-loop structures that are formed by a single-stranded DNA (ssDNA). Complementary domains exist within the strand and form a double helix that becomes the stem. The stability of the hairpin structure is influenced by the length and the base composition of the double helix, the length of the loop, and the temperature. Domains in the loop can become inactive in DNA hybridization reactions if the length of the loop is short.

DNA hairpins can be used as staples inside the origami for enabling transformational capabilities. As shown in Figure 16.5, a standard DNA origami staple at a crossover can be modified such that before it crosses from one helix to another, it folds into a hairpin. As shown in Figure 16.5a, such a primitive can have a toehold domain t_1 and t_2 that can be used to open the hairpin and displace the staple off from the DNA origami using toehold-mediated strand displacement. Figure 16.5c demonstrates a full transition process. An input strand (red) is added to the sample that binds to the toehold domain t to initiate the strand displacement process that eventually completely opens the staple. This process can be repeated for all the staples along a helix until the origami unzips into two different layers.

The zip and unzip transformations may also be completed by strand-displacing DNA polymerases. DNA polymerases are enzymes that can synthesize new single DNA strands. Strand-displacing polymerases can break hybridization bonds of a duplex DNA during transcription of one strand that contains an exposed primer site within the duplex. Once the primer hybridizes with the unpaired section, the strand-displacing polymerase can synthesize a new DNA strand and displace the old one.

A strand-displacing polymerase enzyme such as Bst or phi29 can also be used for the purpose of unzipping. The strand displacement process here is slightly different than toehold-mediated strand displacement. In polymerase-based strand displacement, the input signal attaches to the complex to act as a primer for polymerase reaction. Upon its hybridization, polymerase enzyme comes in and starts
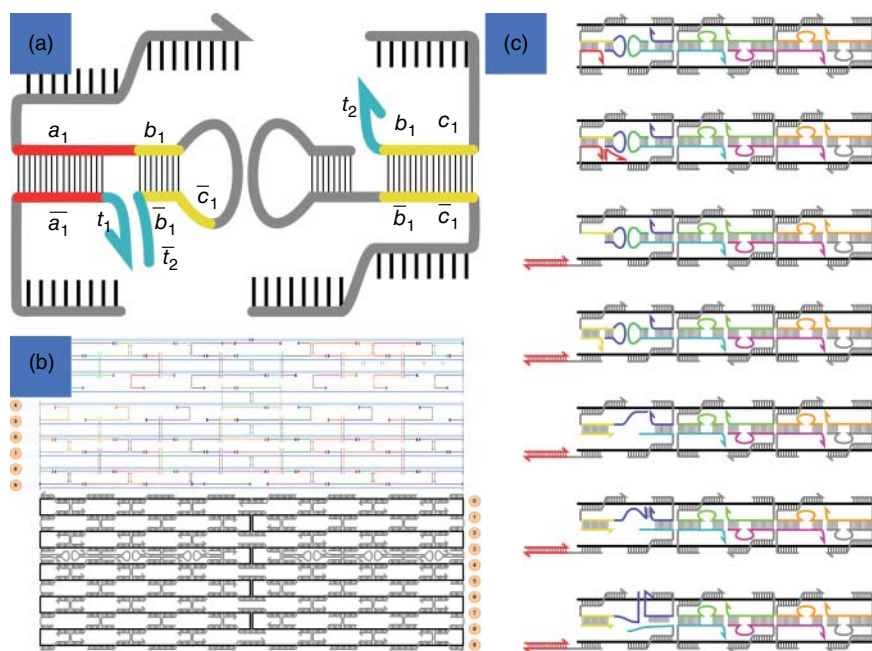
Figure 16.6 Zip and unzip by DNA hairpins. (a) A pair of adjacent hairpins. (b) Hybridization cascade for unzip using hairpins. (c) Hybridization cascade for unzip using hairpins within a rectangular DNA origami.

polymerization process while also displacing any incumbent strand on the complex. (See [56, 57] for more details.) Such polymerase-based architectures can also be used for the transformation process as shown in Figure 16.6. As shown, we can design a long ssDNA to act as a primer for all the staples adjoining two layers. Upon addition of polymerase, it starts the staple displacement process to unzip the origami into two layers. The only caveat with this approach is that polymerase enzymes operate at higher temperature, and therefore the thermal stability of DNA origami at such temperatures has to be accounted for (Figure 16.7).

16.3 Experimental Demonstrations

In some preliminary demonstrations, we designed a rectangular DNA origami nanostructure based on mini-M13 [58]. The first structure (M1) is missing staples that connect the top and bottom halves of the rectangular origami. The zip primitive is applied to M1 to join the two halves together. The second structure (M2) already has staples connecting the top and bottom halves attached during the annealing process. The unzip primitive is applied to M2 to split the two halves of the structure.

The origami nanostructure was made by combining 6.4 μl of 412 nM of scaffold and 100 μl of 240 nM of these staples (at 10× of the scaffold concentration) in the presence of 10 μl of 1 × TAE/Mg^{2+}. The mixture was then divided into two
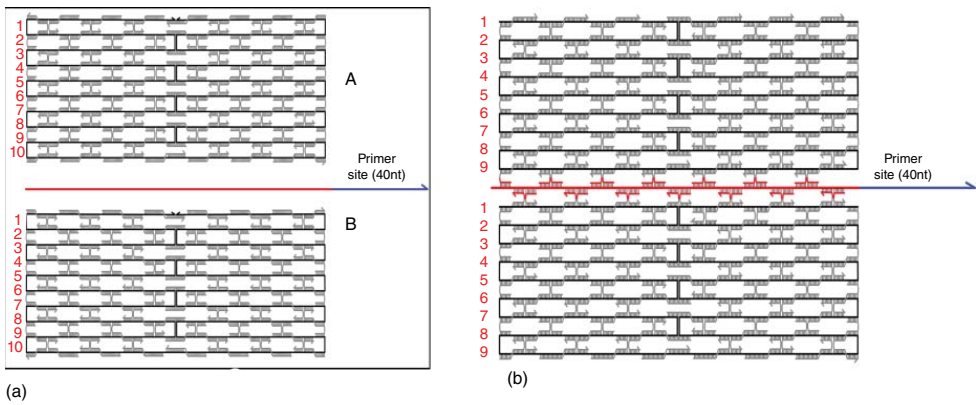
Figure 16.7 Zip and unzip by strand-displacing polymerase. (a) Unhybridized primer site for strand-displacing polymerase. (b) Two adjacent sections held together by a strand-displacing polymerase-compatible unzip strand.
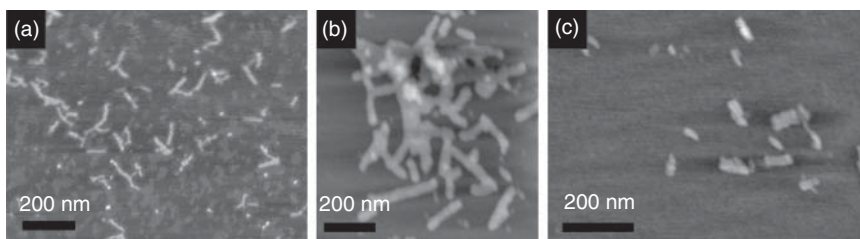
Figure 16.8 AFM characterization. (a) M1 before zip. (b) M1 after zip. (c) M2 after unzip. Note that base stacking between distinct resulting DNA origami can be observed, complicating the interpretation of the AFM images. Source: (a–c) John Reif.

equal parts. Only M2 was combined with the rest of the staples at 10× as well, called zipping staples, which have toeholds on both the 3′ and 5′ ends. Both solutions were annealed for 90 minutes and then left to sit at room temperature for 10 minutes. The objective is to add zipping staples to M1 and unzipping strands to M2 in order to demonstrate zipping and unzipping, respectively, which were added at 10× as well. After adding these strands, the mixtures were left to react between 24 and 72 hours and imaged multiple times using AFM on mica after being diluted to 7.9 nM. Figure 16.8 shows before and after AFM images of M1 and M2.

16.4 Applications

To motivate the description and concrete establishment of DNA origami transformation primitives, we envision and describe actions and mechanisms that could be possibly explored in the future that could rely on multiple transformations. Multistage DNA origami transformations would involve applying sequences of unzip and zip transformations to DNA origami. A general procedure may be as follows: (i) Begin with a self-assembled nanostructure in the same architecture as DNA origami. (ii) Apply a sequence of unzip operations to separate the origami into a number of sheets (e.g. shaped as rectangles or triangles). (iii) Apply a sequence of zip operations to connect together these sheets into a 3D origami. (iv) Repeat steps (i) and (ii) to induce other transformations.

For example, starting with a self-assembled 2D DNA origami consisting of a single rectangle, five unzip transformations can slice up the initial DNA origami rectangular nanostructure into six squares, which can be reassembled via zip transformations into a 3D DNA origami box. Zip and unzip primitives can be applied in the reverse order to return to the original rectangle while preserving all the same components throughout the entire process. Further complex devices potentially important for molecular-scale applications can be envisioned, such as a dynamic pick-and-move device, which would be a box with a jointed movable arm used for picking and moving molecular structures or nanoscale objects. The movable jointed arm may provide a novel and flexible device for nanomanufacturing. These objects could also share a universal starting point (Figure 16.9) by
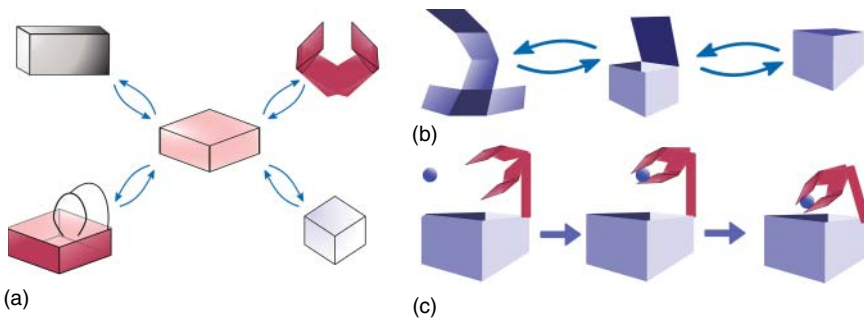
Figure 16.9 Dynamic devices created from a single DNA origami. (a) Examples of transformations from a 2D DNA origami to a long rectangular tube (top left), gripping device with two jointed arms (top right), box with a jointed loading arm (bottom left), or an elongation of the cube into other rectilinear aspect ratios (bottom right). (b) Steps that apply zip and unzip to reversibly transform a 2D conformation of six flat pieces of rectangular DNA origami into a 3D box. (c) Transformations control mechanical movement of a joint and gripping device to manipulate molecular structures or nanoscale objects.

using zip and unzip to manipulate the same starting structure, thereby greatly reducing the cost overheads of designing unique DNA origami nanostructures and mechanisms.

A dynamically reconfigurable network of chambers (Figure 16.10) could also be self-assembled from DNA origami and is dynamically transformed to modify the arrangement and network interconnections between the chambers, thus allowing for movement of molecules between chambers. The reconfigurable network of chambers may provide a method for much more controlled chemical reactions at the molecular scale than previously possible. Moreover, due to conservation of DNA as the substrate to perform such tasks through the zip and unzip primitives, these actions can be easily integrated with existing DNA computing circuits and provide circuitry an immediately useful, physical purpose.

Another possibility is to use localized molecular detection events (e.g. aptamer binding) and DNA computations (using DNA strand displacement methods) to control DNA origami transformations. Figure 16.11 shows a consolidated view of three domains within a DNA origami nanostructure that transforms its aspect ratio upon undergoing a strand displacement reaction. We can make use of controlling DNA-based computations that are localized in the sense that they are proximate to the transformed part of the DNA origami nanostructure. For example, the controlling DNA computation could be a localized hybridization chain reaction [11, 12, 59] based on hairpin logical computations that are used to trigger our DNA origami transformations. Local aptamer-based molecular sensing can be combined with these controlling DNA computations.

The resulting control may provide for initiation of the transformations but also may be used later to guide and control the transformations. As in the above example of reconfigurable chambers, an individual module of a DNA origami nanostructure can be selectively reconfigured with respect to other modules to
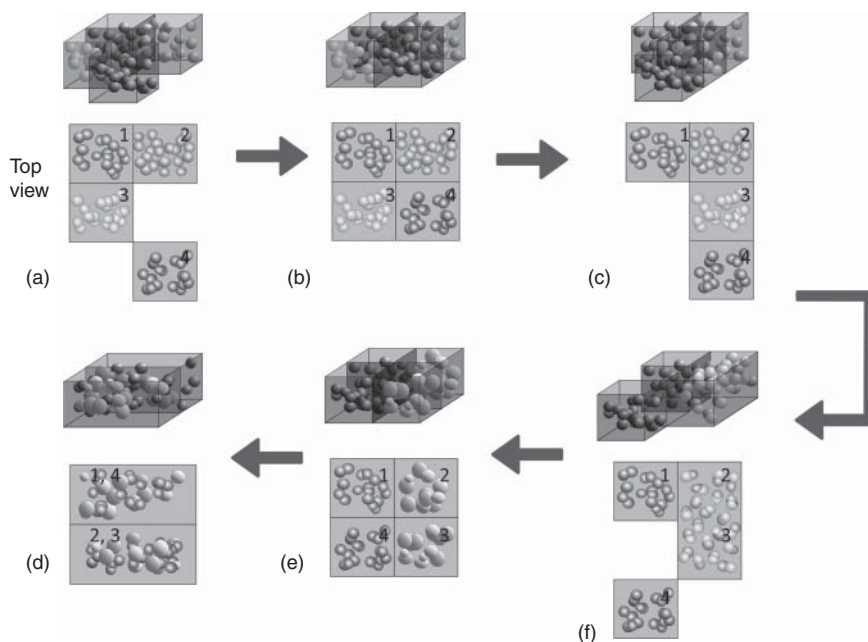
Figure 16.10 Dynamic network of nanocontainers using DNA origami transformers: DNA origami boxes are already synthesized and control a reaction of reagents 1, 2, 3, and 4. Through (a)–(c), boxes are freely manipulated without mixing. At step (d), the edge between boxes 2 and 3 is removed and allows the mixing of reagents. (e) The relative positioning of the boxes can again be manipulated to perform another reaction in (f) between boxes 1 and 4.

the nanostructure depending on the result of computations local to the individual module. This could instruct the module's operation depending on what reagent it holds within itself and could also facilitate inter-module communication to specific actions depending on all reagents present within the system. Overall, we suggest that a small number of (localized) triggers could guide a larger sequence of cascaded transformations, for example, to perform controlled or sequenced release of molecules (e.g. release of drugs *in vivo* for triggered nanoscale drug delivery or reactants for a multistage chemical reaction sequence) in a localized system.

However, to achieve such complex transformations, we implore that immediate future work could focus on autonomously sequencing unzip and zip actions according to input and desired output geometries and proceeding via cascading catalytic events where each unzip strand displacement reaction triggers the next reaction. The initial DNA origami structure could have four types of staple strands: initiator, helper, replaceable, and static. The initiator staple strand has an exposed toehold t that is easily accessible when it is hybridized to the scaffold strand in the source structure. One or more of these initiator strands can be present in a source structure. The entire series of strand displacements is initiated by an external input strand $\overline{s_{init}}$, which is complementary to s_{init}, and removes it from the scaffold via its exposed toehold. When an initiator strand hybridizes and displaces an initiator staple strand in the source structure, this
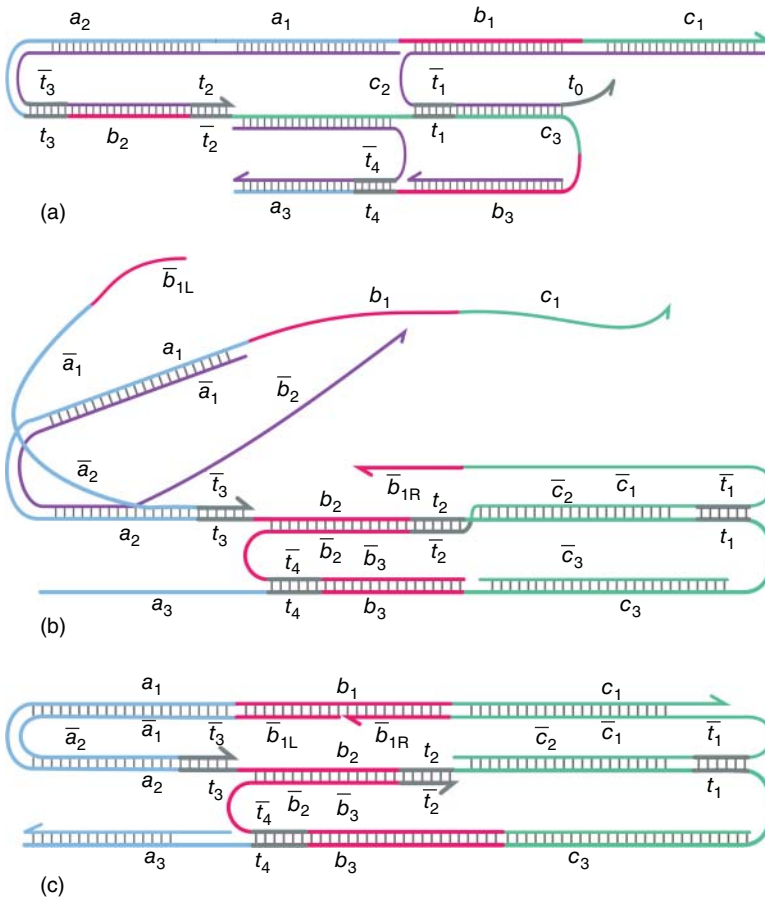
Figure 16.11 Initiated transformations: repeated units of these consolidated domains of a DNA nanostructure can be affected in the same way to induce global change in the physical conformation of the nanostructure. In this subunit, a source substructure (a) has an aspect ratio showing four consecutive domains in the top helix, a_2, a_1, b_1, c_1. A transformation with two invading strands with domains b_{1L} and b_{1R} (b) displaces existing domains to form a new subunit (c) with transformed aspect ratio of only three consecutive domains in the top helix, a_1, b_1, c_1.

exposes a new toehold on the scaffold. This new toehold would allow existing helper strands s_{help} to displace the replaceable strands $s_{replace}$. Static staple strands would remain hybridized in the target structure to the same region in the scaffold as the source. This should serve to trigger an isothermal transformation via the presence of one (or more) initiator strands. If these strands are not present, the transformation should not occur, despite the presence of helper transformation staple strands.

In this system, one of the technical challenges is ensuring that the design minimizes the occurrence of leak reactions (which is an unintended, spurious reaction that occurs due to one or many factors) that may allow a staple replacement

reaction to occur before it is correctly triggered by a prior replacement reaction. This could be enforced via the introduction of mismatches in the invading helper strands at key locations [60, 61].

16.5 Conclusion

In this chapter we have shown the generalization of primitive actions zip and unzip that serve to dynamically physically transform a DNA origami nanostructure into different conformations while maintaining all the same components as the beginning conformation. We have demonstrated these primitives in basic contexts, but we also envision more applications of more complex transformations. The concept draws heavily from the kinetic mechanisms of DNA origami formation and may also be useful in feedback toward the study of DNA origami folding kinetics. The primitives may be further applied in higher complexity objects to induce functional changes with respect to its physical shape or to induce nanomechanical motion. Even more so, these operations closely capture the dynamism that is naturally found in biology and should be expected to pave the way toward increasingly complex biomimetic nanotechnologies.

Acknowledgment

References

1 Seeman, N.C. (1982). Nucleic acid junctions and lattices. *J. Theor. Biol.* 99 (2): 237–247.

2 Rothemund, P.W. (2006). Folding DNA to create nanoscale shapes and patterns. *Nature* 440 (7082): 297.

3 Wei, X., Nangreave, J., Jiang, S. et al. (2013). Mapping the thermal behavior of DNA origami nanostructures. *J. Am. Chem. Soc.* 135 (16): 6165–6176.

4 Mergny, J.-L. and Lacroix, L. (2003). Analysis of thermal melting curves. *Oligonucleotides* 13 (6): 515–537.

5 Yakovchuk, P., Protozanova, E., and Frank-Kamenetskii, M.D. (2006). Base-stacking and base-pairing contributions into thermal stability of the DNA double helix. *Nucleic Acids Res.* 34 (2): 564–574.

6 Castro, C.E., Kilchherr, F., Kim, D.-N., and Shiao, E.L. (2011). A primer to scaffolded DNA origami. *Nat. Methods* 8 (3): 221.

7 Shah, S., Dubey, A.K, and John Reif, J.H. (2019). Programming Temporal DNA Barcodes for Single-Molecule Fingerprinting *Nano Letters* 19 (4): 2668–2673.

8 Garg, S., Shah, S., Bui, H., Song, T., Mokhtar, R., and Reif, J.H. (2018). Renewable Time-Responsive DNA Circuits. *Small*, 14 (33), 1801470.

9 Wickham, S.F.J., Endo, M., Katsuda, Y. et al. (Mar. 2011). Direct observation of stepwise movement of a synthetic molecular transporter. *Nat. Nanotechnol.* 6 (3): 166–169.

10 Lund, K., Manzo, A.J., Dabby, N. et al. (2010). Molecular robots guided by prescriptive landscapes. *Nature* 465 (7295): 206–210.

11 Bui, H., Shah, S., Mokhtar, R. et al. (2018). Localized DNA hybridization chain reactions on DNA origami. *ACS Nano* 12 (2): 1146–1155.

12 Chatterjee, G., Dalchau, N., Muscat, R.A. et al. (2017). A spatially localized architecture for fast and modular DNA computing. *Nat. Nanotechnol.* 12 (9): 920.

13 Song, T., Shah, S., Bui, H., Garg, S., Eshra, A., Yang, M., and Reif, J.H. (2019). Programming DNA-Based Biomolecular Reaction Networks on Cancer Cell Membranes, *Journal of the American Chemical Society (JACS)*, 141 (42), 16539–16543.

14 Yurke, B., Turber, A.J., Simmel, F.C., and Neumann, J.L. (2000). A DNA-fuelled molecular machine made of DNA. 406: 5.

15 Wei, B., Ong, L.L., Chen, J. et al. (2014). Complex reconfiguration of DNA nanostructures. *Angew. Chem. Int. Ed.* 126 (29): 7605–7609.

16 Zhang, F., Nangreave, J., Liu, Y., and Yan, H. (2012). Reconfigurable DNA origami to generate quasifractal patterns. *Nano Lett.* 12 (6): 3290–3295.

17 Wang, W., Yang, Y., Cheng, E. et al. (2009). A pH-driven, reconfigurable DNA nanotriangle. *Chem. Commun.* 7: 824–826.

18 Douglas, S.M., Dietz, H., Liedl, T. et al. (2009). Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature* 459 (7245): 414.

19 Dietz, H., Douglas, S.M., and Shih, W.M. (Aug. 2009). Folding DNA into twisted and curved nanoscale shapes. *Science* 325 (5941): 725–730.

20 Ke, Y., Ong, L.L., Shih, W.M., and Yin, P. (Nov. 2012). Three-dimensional structures self-assembled from DNA bricks. *Science* 338 (6111): 1177–1183.

21 Wei, B., Dai, M., and Yin, P. (2012). Complex shapes self-assembled from single-stranded DNA tiles. *Nature* 485 (7400): 623.

22 He, Y., Chen, Y., Liu, H. et al. (2005). Self-assembly of hexagonal DNA two-dimensional (2D) arrays. *J. Am. Chem. Soc.* 127 (35): 12202–12203.

23 Park, S.H., Barish, R., Li, H. et al. (2005). Three-helix bundle DNA tiles self-assemble into 2D lattice or 1D templates for silver nanowires. *Nano Lett.* 5 (4): 693–696.

24 Rothemund, P.W., Papadakis, N., and Winfree, E. (2004). Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biol.* 2 (12): e424.

25 Goodman, R.P., Berry, R.M., and Turberfield, A.J. (2004). The single-step synthesis of a DNA tetrahedron. *Chem. Commun.* 12: 1372–1373.

26 Shih, W.M., Quispe, J.D., and Joyce, G.F. (2004). A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron. *Nature* 427 (6975): 618.

27 Han, D., Pal, S., Nangreave, J. et al. (2011). DNA origami with complex curvatures in three-dimensional space. *Science* 332 (6027): 342–346.

28 Gu, H., Chao, J., Xiao, S.-J., and Seeman, N.C. (May 2010). A proximity-based programmable DNA nanoscale assembly line. *Nature* 465 (7295): 202–205.

29 Bui, H., Onodera, C., Kidwell, C. et al. (2010). Programmable periodicity of quantum dot arrays with DNA origami nanotubes. *Nano Lett.* 10 (9): 3367–3372.

30 Kuzyk, A., Schreiber, R., Fan, Z. et al. (2012). DNA-based self-assembly of chiral plasmonic nanostructures with tailored optical response. *Nature* 483 (7389): 311.

31 Schreiber, R., Kempter, S., Holler, S. et al. (Jul. 2011). DNA origami-templated growth of arbitrarily shaped metal nanoparticles. *Small* 7 (13): 1795–1799.

32 Schreiber, R., Do, J., Roller, E.-M. et al. (2014). Hierarchical assembly of metal nanoparticles, quantum dots and organic dyes using DNA origami scaffolds. *Nat. Nanotechnol.* 9 (1): 74.

33 Qian, L. and Winfree, E. (2014). Parallel and scalable computation and spatial dynamics with DNA-based chemical reaction networks on a surface. In: *DNA Computing and Molecular Programming. DNA 2014. Lecture Notes in Computer Science*, vol. 8727 (eds. S. Murata and S. Kobayashi), 114–131. Cham: Springer.

34 Zhang, Q., Jiang, Q., Li, N. et al. (2014). DNA origami as an in vivo drug delivery vehicle for cancer therapy. *ACS Nano* 8 (7): 6633–6643.

35 Jiang, Q., Song, C., Nangreave, J. et al. (2012). DNA origami as a carrier for circumvention of drug resistance. *J. Am. Chem. Soc.* 134 (32): 13396–13403.

36 Zhao, Y.-X., Shaw, A., Zeng, X. et al. (2012). DNA origami delivery system for cancer therapy with tunable release properties. *ACS Nano* 6 (10): 8684–8691.

37 Torelli, E., Marini, M., Palmano, S. et al. (Jul. 2014). A DNA origami Nanorobot controlled by nucleic acid hybridization. *Small* 10 (14): 2918–2926.

38 Liu, D., Bruckbauer, A., Abell, C. et al. (2006). A reversible pH-driven DNA nanoswitch array. *J. Am. Chem. Soc.* 128 (6): 2067–2071.

39 Douglas, S.M., Bachelet, I., and Church, G.M. (Feb. 2012). A logic-gated Nanorobot for targeted transport of molecular payloads. *Science* 335 (6070): 831–834.

40 Andersen, E.S., Dong, M., Nielsen, M.M. et al. (2009). Self-assembly of a nanoscale DNA box with a controllable lid. *Nature* 459 (7243): 73.

41 Shen, X., Asenjo-Garcia, A., Liu, Q. et al. (2013). Three-dimensional plasmonic chiral tetramers assembled by DNA origami. *Nano Lett.* 13 (5): 2128–2133.

42 Pilo-Pais, M., Watson, A., Demers, S. et al. (2014). Surface-enhanced Raman scattering plasmonic enhancement using DNA origami-based complex metallic nanostructures. *Nano Lett.* 14 (4): 2099–2104.

43 Maune, H.T., Han, S., Barish, R.D. et al. (Jan. 2010). Self-assembly of carbon nanotubes into two-dimensional geometries using DNA origami templates. *Nat. Nanotechnol.* 5 (1): 61–66.

44 Suzuki, Y., Endo, M., Yang, Y., and Sugiyama, H. (2014). Dynamic assembly/disassembly processes of photoresponsive DNA origami nanostructures directly visualized on a lipid membrane surface. *J. Am. Chem. Soc.* 136 (5): 1714–1717.

45 Yang, Y., Endo, M., Hidaka, K., and Sugiyama, H. (2012). Photo-controllable DNA origami nanostructures assembling into predesigned multiorientational patterns. *J. Am. Chem. Soc.* 134 (51): 20645–20653.

46 Zhou, L., Marras, A.E., Su, H.-J., and Castro, C.E. (Jan. 2014). DNA origami compliant nanostructures with tunable mechanical properties. *ACS Nano* 8 (1): 27–34.

47 Mitchell, J.C., Harris, J.R., Malo, J. et al. (2004). Self-assembly of chiral DNA nanotubes. *J. Am. Chem. Soc.* 126 (50): 16342–16343.

48 Benn, F., Haley, N.E.C., Lucas, A.E. et al. (2018). Chiral DNA origami nanotubes with well-defined and addressable inside and outside surfaces. *Angew. Chem. Int. Ed.* 130 (26): 7813–7816.

49 Huang, Y., Nguyen, M.-K., Natarajan, A.K. et al. (Dec. 2018). A DNA origami-based chiral Plasmonic sensing device. *ACS Appl. Mater. Interfaces* 10 (51): 44221–44225.

50 Chen, H., Weng, T.-W., Riccitelli, M.M. et al. (2014). Understanding the mechanical properties of DNA origami tiles and controlling the kinetics of their folding and unfolding reconfiguration. *J. Am. Chem. Soc.* 136 (19): 6995–7005.

51 Castro, C.E., Su, H.-J., Marras, A.E. et al. (2015). Mechanical design of DNA nanostructures. *Nanoscale* 7 (14): 5913–5921.

52 Han, D., Pal, S., Liu, Y., and Yan, H. (2010). Folding and cutting DNA into reconfigurable topological nanostructures. *Nat. Nanotechnol.* 5 (10): 712.

53 Gerling, T., Wagenbauer, K.F., Neuner, A.M., and Dietz, H. (2015). Dynamic DNA devices and assemblies formed by shape-complementary, non–base pairing 3D components. *Science* 347 (6229): 1446–1452.

54 Santa Lucia, J. Jr., and Hicks, D. (2004). The thermodynamics of DNA structural motifs. *Ann. Rev. Biophys. Biomol. Struct.* 33: 415–440.

55 Arbona, J.-M., Elezgaray, J., and Aime, J.-P. (2012). Modelling the folding of DNA origami. *EPL Europhys. Lett.* 100 (2): 28006.

56 Shah, S., Song, T., Song, X. et al. (2019). Implementing arbitrary CRNs using strand displacing polymerase. In: *International Conference on DNA Computing and Molecular Programming (DNA 2019), Chapter in: DNA Computing and Molecular Programming*, vol. 11648 (eds. C. Thachuk and Y. Liu), 21–36. Switzerland AG: Springer.

57 Song, T., Eshra, A., Shah, S. et al. (2019). Fast and compact DNA logic circuits based on single-stranded gates using strand-displacing polymerase. *Nat. Nanotechnol.*: 1–7.

58 Brown, S., Majikes, J., Martínez, A. et al. (2015). An easy-to-prepare mini-scaffold for DNA origami. *Nanoscale* 7 (40): 16621–16624.

59 Chandran, H., Gopalkrishnan, N., Phillips, A., and Reif, J. (2011). Localized hybridization circuits. In: *International Conference on DNA Computing and Molecular Programming, (DNA17), California Institute of Technology, Pasadena, California, September 19–23, Lecture Notes for Computer Science (LNCS), NYC, NY* (eds. L. Cardelli and W. Shih), 64–83. Berlin, Heidelberg: Springer-Verlag, LNCS.

60 Jiang, Y.S., Bhadra, S., Li, B., and Ellington, A.D. (2014). Mismatches improve the performance of Strand-displacement nucleic acid circuits. *Angew. Chem. Int. Ed.* 53 (7): 1845–1848.

61 Machinek, R.R., Ouldridge, T.E., Haley, N.E. et al. (2014). Programmable energy landscapes for kinetic control of DNA strand displacement. *Nat. Commun.* 5: 5324.

17

Nanopore Decoding for DNA Computing

Hiroki Yasuga[1], Kan Shoji[2,3], and Ryuji Kawano[2]

[1]*Ochanomizu University, Faculty of Core Research, 2-1-1, Otsuka, Bunkyo-ku, Tokyo 112-8610, Japan*
[2]*Tokyo University of Agriculture and Technology (TUAT), Department of Biotechnology and Life Science, 2-24-16, Naka-cho, Koganei-shi, Tokyo 184-8588, Japan*
[3]*Nagaoka University of Technology, Department of Mechanical Engineering, 1603-1 Kamitomioka, Nagaoka-shi, Niigata 940-2188, Japan*

Abbreviations

αHL	α-hemolysin
DNA	deoxyribonucleic acid
miRNA	microRNA
PCR	polymerase chain reaction
PEG	polyethylene glycol
PNA	peptide nucleic acid
RNA	ribonucleic acid
SCLC	small-cell lung cancer
SNR	signal-to-noise ratio
ssDNA	single-stranded DNA

17.1 Introduction

DNA computing is an area of natural computing based on the principle that polynucleotides contain information encoding amino acid sequence, with this information capable of being transferred and/or copied using chemical and enzymatic reactions. In DNA computing, the computing processes in the conventional digital electronic computing that are an input of data coded in binary, data conversion by the programmed data rewrite rules, and output of results as codes are substituted as below. First, input data are encoded as DNA strands. Then, the input data are converted by molecular biology processes such as DNA hybridization, enzymatic reactions, and strand displacement methods. Finally, the computational result is described as a DNA strand. In 1994, a computer scientist, Adleman, realized that the DNA polymerase works like a computer and proposed "DNA-based computing" based on a directed Hamiltonian path problem (Figure 17.1A) [1]. The problem solution involves finding a path among
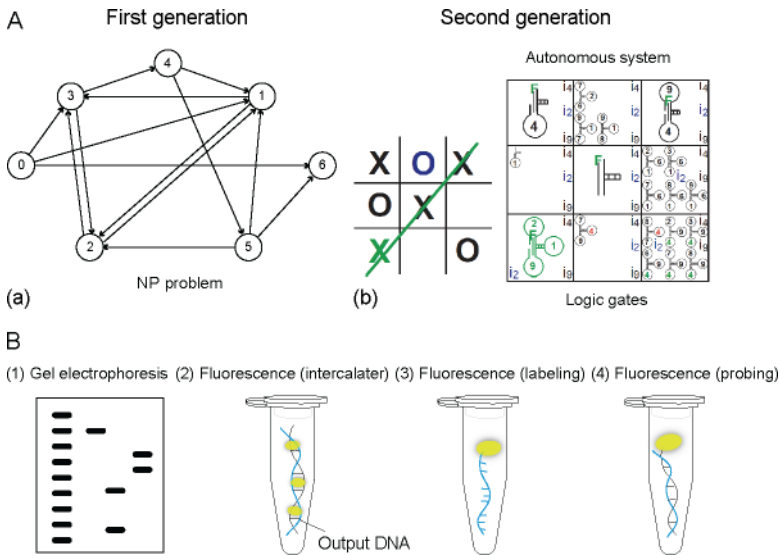
Figure 17.1 Conventional DNA computing and its decoding. (A) Adleman [1] described molecular computations using a Hamiltonian path presented as a problem with NP complexity (a). Source: From Adleman [1]. Tic-tac-toe board representation (b). Source: From Stojanovic and Stefanovic [2]. © 2003. Reproduced with the permission of Springer Nature. (B) Conventional decoding for DNA computing.

several cities on a map, such that each city is visited only once. This was initially transferred to DNA-related problems by preparing short DNAs (20-mers) and assigning each to different cities and paths. Five steps were experimentally performed in order to solve this problem:

Step 1: Generate random paths through the graph, with all DNAs mixed, hybridized, and fixed by ligation.
Step 2: Remove all paths that do not begin with a start node, "0," and end with an end node, "6," using polymerase chain reaction (PCR).
Step 3: The correct length of the DNAs is separated and retrieved by gel electrophoresis.
Step 4: Remove any paths that repeat nodes using magnetic bead purification.
Step 5: Supply an answer of "Yes" or "No" depending on whether any path remains. This type of problem displays NP complexity, requiring enormous computational resources relative to the number of cities. NP problems are considered beyond the scope of von Neumann-type computers. After proposing this groundbreaking idea, computer scientists collaborated with wet-lab scientists to study first-generation DNA-based computing [3–9], which required human intervention to implement each step (i.e. adding solutions, changing reaction temperatures, and observing results via gel electrophoresis).

The second generation of DNA computation allowed autonomous calculations. To create autonomous operations, a method involving "strand displacement" was developed [10, 11] that utilized differences in free energy (ΔG) associated

with DNA hybridization. When the hybridization energy of DNA strand A-B is larger than that of A-C, the A-C strand is displaced by A-B autonomously. Using this reaction, the operational procedure could be encoded in the DNA sequence by designing the reaction order in terms of the differences in ΔG. Benenson et al. [12, 13] proposed a finite automaton system using DNAs and restriction enzymes and that operated on a state transition autonomously. This operation was implemented in a 120 µL volume without additional procedures and at room temperature, with this calculation recorded in the *Guinness World Records* as representing the "smallest biological computing device."

Logic gate [14, 15] implementation is another approach used to construct autonomous DNA-based calculations, given that they are constructed according to a simple binary combination of OR, NOT, and AND gates. This method allows higher-level calculations by combining several logic gates, with any logic gate capable of construction through combining multiple NAND (negative-AND) gates. Several researchers have studied complex binary operations using DNAs and enzymes [16–19]. A popular application of logic gates involves a game of "tic-tac-toe" using nine wells in a 3×3 matrix. Stojanovic et al. [2, 20] constructed a DNA computational version of "tic-tac-toe" named "MAYA" (Molecular Array of YES and AND-AND-NOT gates) using DNAzyme. This algorithm involved a simplified symmetry-pruned game of tic-tac-toe encompassing 19 permissible game plays and using an array of 23 logic gates distributed over eight wells (Figure 17.1A).

Because of the characteristics of DNAs and biochemical reactions, the DNA computer has many advantages as below by comparison with conventional silicon-based computers. First, parallel processing can be provided. Because biochemical reactions based on DNAs and enzymes in the DNA computer can be performed simultaneously, the DNA computer has the potential to calculate faster than conventional silicon-based computers. Second, DNAs have high data density. Instead of a string of binary data is encoded with 1's and 0's in the silicon-based computer, a strand of DNA is encoded with the four nucleotides that are represented by A, T, C, and G, and each nucleotide is helically arranged every 0.35 nm. In theory, DNAs potentially provide the data density of around one zettabyte per gram of DNAs. Erlich and Zielinski reported DNA storage, which has a data density of 215 petabytes per gram of DNA [21]. Furthermore, biocompatibility can also be provided. Since biochemical reactions work well *in vivo* conditions, DNA computers attract attention as *in vivo* molecular computing devices, and many researchers have reported *in vivo* application of DNA computing for diagnosis, cell imaging, and drug delivery [22–24].

On the other hand, there are several issues while using DNA computing. One of the most significant issues is its accuracy. Because hybridization is not very precise and mismatching pairs are often obtained, if DNA computing works with massive data, the probability of error increases exponentially. Also, complex chemical reactions lead to errors and potentially produce false positive results. Another big problem associated with DNA computing is the time-consuming process of the data output. In conventional DNA computation, recognition of output molecules is mainly performed by four different methods (Figure 17.1B and Table 17.1): (i) gel-electrophoretic detection following PCR amplification,

Table 17.1 Comparison of conventional fluorescence and nanopore decoding methods of DNA computation.

	Gel electrophoresis [1, 12]	Fluorescence (intercalator) [25]	Fluorescence (labeling) [26]	Fluorescence (probing) [27]	Nanopore [28] [29]
Detection method	Optical	Optical	Optical	Optical	Electrical
Measurement time	Long	Short	Short	Long	Short
Sensitivity	Low	Low	High	High	High
Throughput	Low	Low	High	High	Potentially high
Pretreatment	Multistep	Few steps	Multistep	Multistep	Few steps
Generality of method	High	High	Medium	Medium	Low

(ii) fluorescence detection with isothermal amplification, (iii) fluorescence detection by direct labeling without amplification, and (iv) fluorescence probing without amplification. The pioneering works of Adleman [1] and Benenson et al. [12] in DNA computation used these methods. Despite the recent development of microscale rapid gel electrophoresis [31, 32], traditional gel electrophoresis is time consuming. As a substitute, several fluorescence techniques have been developed involving specific amplification of output DNA by isothermal reactions and observation by fluorescence labeling [2, 25]. Methods 1 and 2 require an amplification step involving enzymes, which requires long reaction times and temperature control, even under constant conditions at 37°C. Therefore, non-amplification methods (i.e. methods 3 and 4) can be used, where the output DNA is labeled and detected [26], and a specific fluorescence probe is used [27]. Although neither of these methods require an amplification step (considering their implementation at relatively high concentrations), the use of direct labeling or specific probe molecules is required.

Nanopore technology allows the rapid and electrical detection of oligonucleotides in the absence of labeling. Several studies reported methods related to nanopore decoding [28–30] and their applications in diagnosis or clinical settings based on DNA computing. Therefore, nanopore methods represent potential candidate methods for decoding DNA computations.

17.2 Application of Nanopore Technology for Rapid and Label-Free Decoding

Nanopore technology involves electrical measurement of ion current through a nanopore [33–40]. Biological (proteins) or solid-state nanopores ranging in size from 1 to ≥10 nm show open-pore current conductance (Figure 17.2a). When a molecule passes through or blocks a nanopore, the open-pore current reduces,
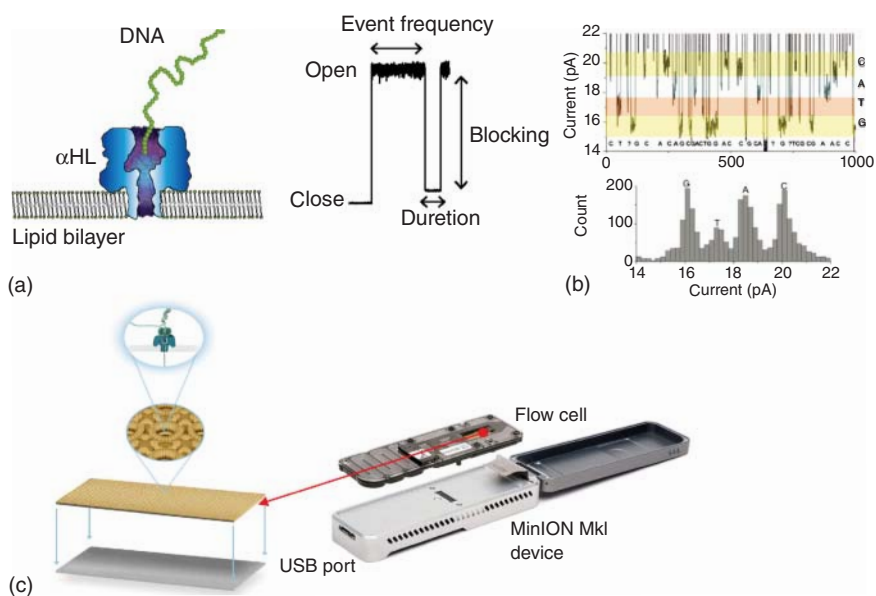
Figure 17.2 Applications of nanopore technology. (a) Schematic illustration of nanopore detection of DNA and current–time trace. (b) DNA detection using an αHL nanopore. Four different mononucleotides show individual blocking current levels. Source: From Astier et al. [41]. © 2006. Reproduced with the permission of American Chemical Society. (c) A commercialized nanopore sequencer with small flow cells. Source: From Lu et al. [42]. Reprinted with permission of Elsevier.

thereby demonstrating current-signal blockage. The blocking amplitude, duration time, and event frequency provide information regarding the size, mobility, and concentration of target molecules at the single-molecule level. α-Hemolysin (αHL), a channel toxin from *Staphylococcus aureus* [43], is conventionally used as a biological nanopore for detecting oligonucleotides based on its having a pore size comparable with that of single-stranded DNA (ssDNA) or ssRNA. Extensive studies reported using this nanopore as a label-free, rapid, and electrical method for single-oligonucleotide determination, with one application targeting nanopore sequencing (Figure 17.2b) [41, 44–47]. Since the first report in 1996 by Kasianowicz et al. [44], enormous efforts have been undertaken to apply this method [41, 45, 47–60]. In 2015, a company named Oxford Nanopore Technologies was launched and provided the first commercially available nanopore sequencer for general use (Figure 17.2c). Currently, nanopore technology can be utilized not only for single DNA/RNA detection but also for large-scale DNA sequencing [42].

Nanopore methods allow recognition of oligonucleotides rapidly, electrically, and without the necessity for labeling. Since 2016, several studies reported possible application of this technology for the detection of DNA computing output. The first study involved detection of the output of a NAND logic operation in a microdroplet system [61]. This method involved construction of a four-droplet system with a biological nanopore at the droplet interface bilayer and electrodes

in each droplet [62–65]. Input DNAs are injected into two input droplets, calculations are performed in the operation droplet, and these DNAs are subsequently passed through the output droplet, with the output monitored electrically by the nanopore (Figure 17.3). The important feature of this work was that output "1" or "0" was defined according to whether an ssDNA translocated through a nanopore. This method harnessed the unique property of the αHL nanopore, which allowed only ssDNAs to pass through. This method involves conversion of molecular information into electrical signals in a binary system, and while output times associated with fluorescence-based logic gates range from minutes to hours, this nanopore system requires only ~10 minutes without any labeling.

It is also noteworthy that the nanopore method combined with a microdroplet system facilitates connection of DNA logic gates. In general, DNA logic circuits suffer from unintended crosstalk reactions of DNA and enzymes as the type of molecules increases: it makes difficult to perform complex operations in a single aqueous solution system [10, 66]. To suppress such unintended crosstalk reactions, Yasuga et al. developed a microdroplet system with a mechanism of DNA relay (Figure 17.4) [29]. The DNA relay mechanism was based on electrical fusion and mechanical splitting in a split-and-contact method device equipped with electrodes [67]. In the proposed system, each component (i.e. input, output, and logic gate) was compartmented into a water-in-oil droplet covered with lipid molecules, and the respective droplets were prepared in mechanically sliding and fixed wells as shown in Figure 17.4. The ssDNA involved with binary data was relayed step by step from input via operation to output droplets: a droplet made contact with another droplet, followed by formation of droplet interface lipid bilayer; then, the lipid bilayer was ruptured by applying pulsed voltage, resulting in the fusion of the droplets and mixing of ssDNA over the droplets; the relay was terminated by mechanically splitting the two droplets. The ssDNA in the output droplet can be transferred as an input for the next logic gate. The DNA relay mechanism allowed integration of two types of logic gates, OR and NOT gates, demonstrating NOR logic gate. As silicon-based logic circuits have obtained sophisticated functionalities through the process of integration, the relay mechanism allowing the connection of DNA logic gates is expected to enable implementing complicated operation through integration of multiple DNA logic gates.

Following these preliminary studies, Ohara et al. [28, 68] proposed a complicated logic operation using enzyme reactions in the microdroplet of a nanopore system in order to accommodate the necessity for enzymatic reactions in most DNA computations. An enzyme-free system requires rigid operational conditions, because the temperature of the reaction and/or enzyme concentrations are strictly controlled in such reactions. However, enzyme-free operations can be implemented in one-to-one reaction, with one input molecule generating one output molecule using a chain displacement reaction. By contrast, operations involving enzymes can be implemented in versatile reactions, such as DNA polymerization, amplification, and transcription. Therefore, verification of nanopore decoding is important, especially in cases of DNA calculations involving enzymatic reactions. Ohara et al. [28] constructed an AND gate allowing the input molecules to amplify and transcribe DNA via T7 RNA polymerase when two
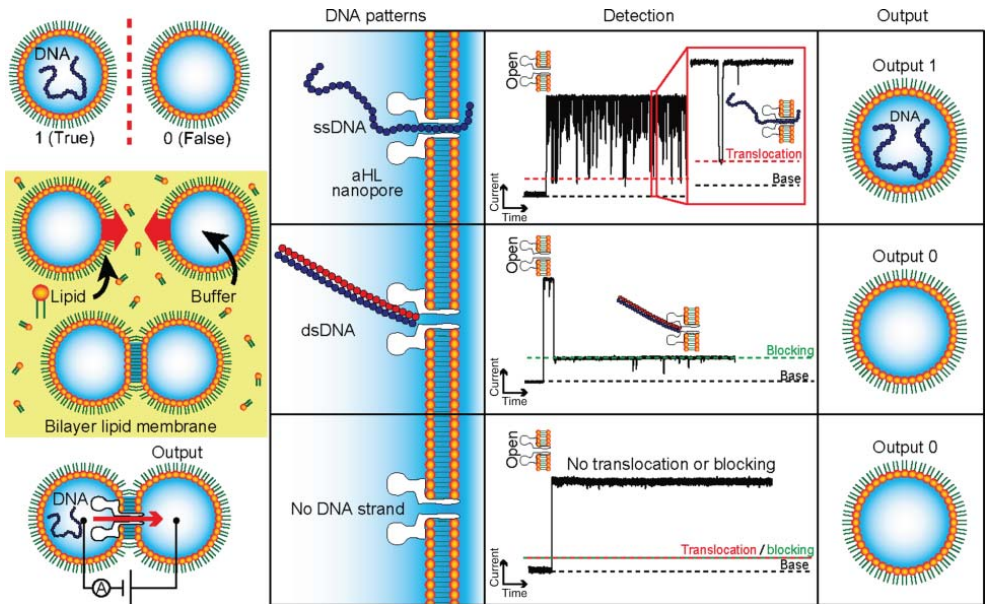
Figure 17.3 Nanopore decoder methodology. NAND operation in a droplet system for nanopore decoding. The input molecule moves from the input droplet to the output droplet through the nanopore.
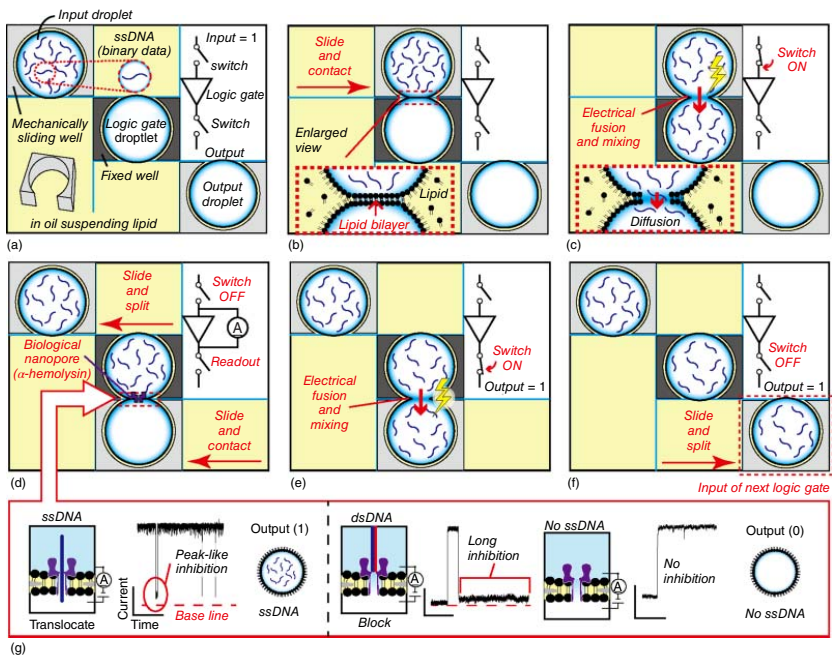
Figure 17.4 DNA relay mechanism. (a–f) Conceptual diagrams of the DNA relay mechanism using the mechanical splitting and electrical fusion of droplets, exemplifying YES logic gate. Binary data, i.e. ssDNA, is gradually relayed from input via operation to output droplets. (g) When necessary, the result of operation can be obtained by nanopore decoding at the interface operation and output droplets.

(a)

(b)
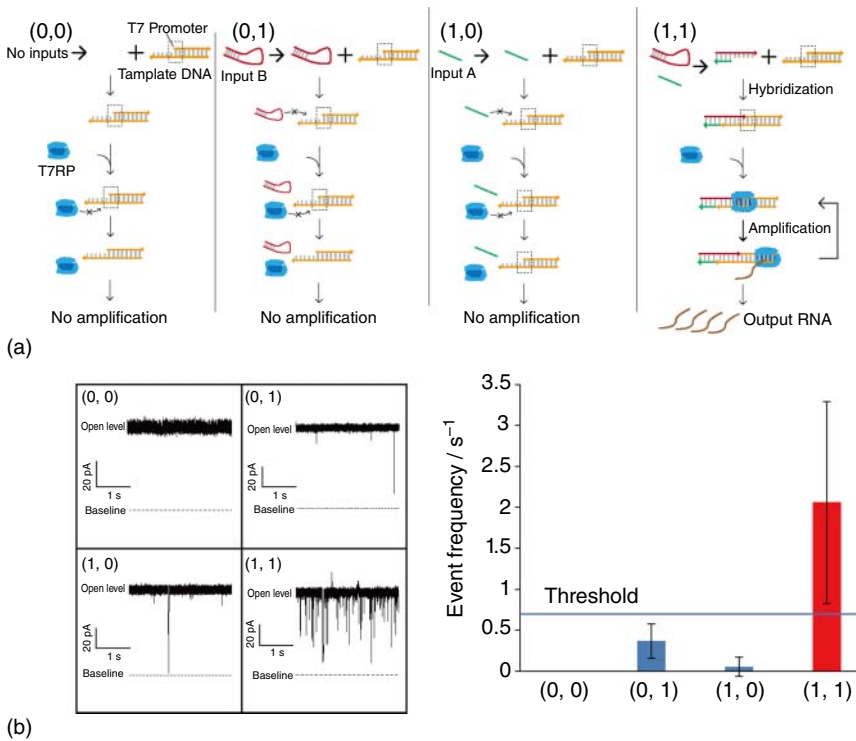
Figure 17.5 Nanopore decoder applications. (a) Four individual operations associated with a reverse transcription AND gate involving the T7 RNA polymerase. (b) Nanopore measurement enables rapid and label-free detection of the output molecules. The translocation frequency of the output molecules through the nanopore allows discrimination between a (1, 1) system and others. Source: (a,b) From Ohara et al. [28]. © 2017. Reproduced with the permission of American Chemical Society.

DNAs are input simultaneously (Figure 17.5a). Their study showed that four different operations represented by (0 0), (1 0), (0 1), and (1 1) were implemented in multiple microdroplet devices and that the output could be obtained after 90 minutes, which included a 60 minutes' enzymatic reaction (Figure 17.5b) [26], suggesting the efficacy of nanopore decoding for operations involving enzymes in microdroplet systems. However, the reaction efficiency differed between that performed in conventional plastic tubes and droplets with a surrounding lipid bilayer, with reduced efficiency observed in the droplet system [28]. Therefore, these operations need to be improved appropriately with respect to enzymatic reactions performed in a lipid droplet environment.

17.3 Application of Nanopore Decoding in Medical Diagnosis

The field of DNA computing was developed largely as a curiosity-driven exercise focused on solving mathematics-related problems, including cryptograms and

constructing various types of logic gates (AND, OR, NOT, XOR, and NAND). However, this field recently increased in importance due to its potential applications in medical diagnosis [22, 69, 70]. Benenson et al. [22] reported autonomous diagnosis and drug-release systems using DNA computing using the following "if-then" logic: "if" certain diagnostic conditions are true, such as low expression levels of certain mRNAs relative to those of others, "then" the antisense drug is released. After this pioneering study, several studies were undertaken focused on application of this technology to diagnosis and therapy [69]. Based on the favorable compatibility of nanopore technology with oligonucleotide detection, strategies utilizing this method for diagnosis using nanopores and DNA have been proposed.

Wang et al. demonstrated nanopore-based detection of microRNAs (miRNAs) from patients with lung cancer (Figure 17.6a) [71]. MiRNAs are short noncoding RNAs, the expression levels of which correlate with various diseases and represent potential early diagnostic markers for cancer [74]. Wang et al. used programmable oligonucleotide probes that formed partially hybridized structures with target miRNA (Figure 17.6b), resulting in partially hybridized DNA/RNA that exhibited different blocking levels against unzipping relative to non-hybridized oligonucleotides. They were specifically able to detect target miRNAs of the let-7 tumor family at picomolar levels, providing a practical demonstration of miRNA detection in cancer patients with cancer and emphasizing that the measurement accuracy of the αHL nanopore was higher than that of quantitative real-time PCR assays [71]. Subsequent studies reported development of specific tags, such as peptide nucleic acid (PNA) [75] or polyethylene glycol (PEG) [30], used to detect miRNA. The PNA probe method is unique and involves hybridization of a cationic probe to the target miRNA to form a double-stranded structure that can be captured by nanopores exhibiting opposite polarity [68]. Despite the efficacy of this method, simultaneous detection of multiple miRNAs remains challenging. Complementary DNA with a PEG tag was later used to target miRNAs, with PEG tags bound to target miRNAs showing different blocking levels during translocation through the αHL nanopore [32]. Although this method achieved accurate detection of four different miRNAs, it was difficult to measure the amounts of respective miRNAs, because differences in the blocking levels were too close to allow discrimination. Recently, the analysis of the duration of the blocking instead of the blocking current is proposed for the pattern recognition of miRNA expression using AND logic gate with nanopore technology [77]. Two different miRNAs (miR-20a and miR-17-5p) are simultaneously secreted from tumor cells of small-cell lung cancer (SCLC). In this report, designed diagnostic DNAs autonomously recognized miR-20a and miR-17-5p; four different operations, i.e. (0, 0), (0, 1), (1, 0), and (1, 1), could be discriminated as AND logic operation.

Another advantage of using nanopore technology for miRNA detection is its sensitivity. In conventional analytic methods, such as microscopic or electrochemical methods, sensitivity relies upon signal intensity. In fluorescence measurements, low-intensity results make it difficult to discriminate between noise and signal at low concentrations. Therefore, sensitivity is dependent upon the signal-to-noise ratio (SNR). On the other hand, nanopore measurements at low
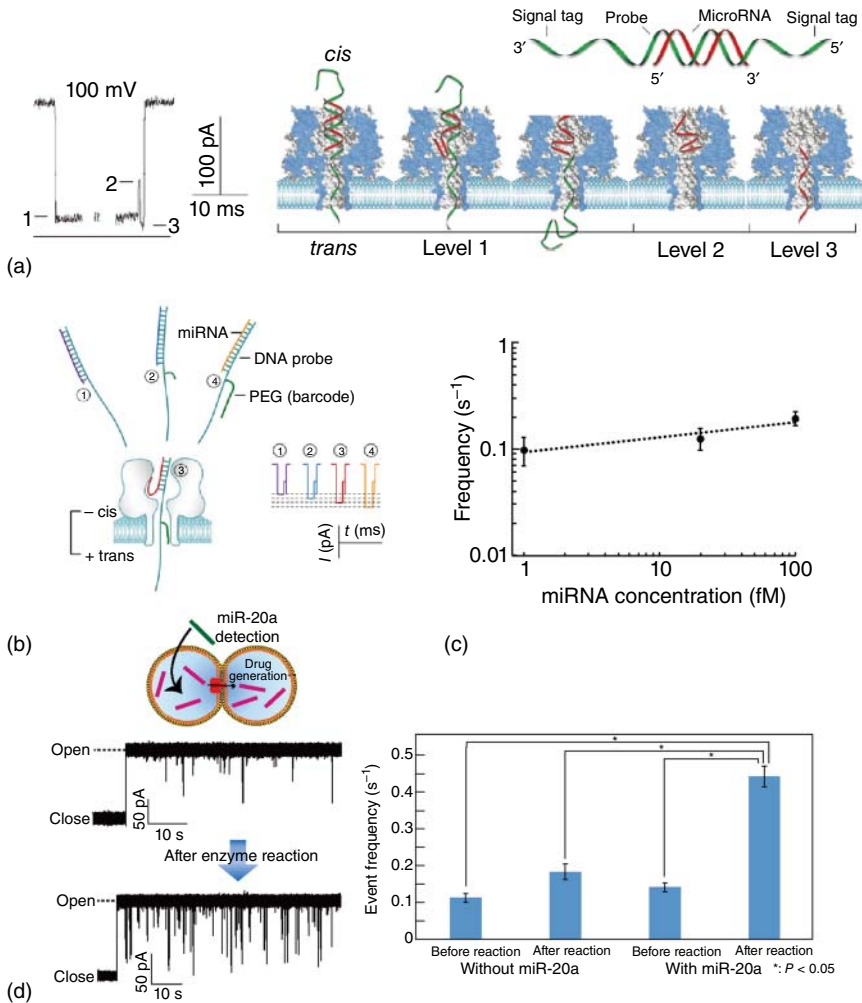
Figure 17.6 Nanopore detection of miRNAs. (a) MiRNA detection using DNA probes. The αHL nanopore recognizes the complex according to specific current signals. Source: (a) From Wang et al. [71]. © 2011. Reproduced with the permission of Springer Nature. (b) PEG-tagged DNA probes provide different blocking currents to allow discrimination between specific miRNAs. Source: (b) From Zhang et al. [30]. © 2014. Reproduced with the permission of American Chemical Society. (c) Detection of ultralow concentrations (on order of 1 fM) of miRNAs using nanopores and DNA computing. Source: (c) From Zhang et al. [72]. Reproduced with permission of Royal Society of Chemistry. (d) Theranostics system for SCLC using DNA computing. An antisense DNA drug generated upon miR-20a detection, followed by nanopore-based measurement to quantify drug concentration using a label-free and real-time method. Source: From Hiratani et al. [73]. © 2017. Reproduced with the permission of American Chemical Society.

concentrations result in no changes in the SNR of the current signals, but rather a decrease in the frequency of appearance of the signal. Ideally, even at very low target molecule concentrations, such as those involving single molecules, the signal will appear during continuous measurement. This implies that the sensitivity of nanopore measurements is dependent upon measurement time, which normally ranges from several minutes to several hours. Consequently, concentration limitations are approximately on the order of one picomolar, even after several attempts at enhancing translocation via the nanopore under asymmetric salt conditions [76].

A useful technique associated with DNA computation involves amplification, which can enhance detection of targets at low concentrations. Zhang et al. reported the successful detection of low concentrations of miR-20a, which is secreted in SCLC, by combining isothermal amplification of the oligonucleotide along with nanopore-based methods (Figure 17.6c) [72]. Their method amplified stable ssDNA from miR-20a at concentrations ranging from 1 fM to 10 pM using an isothermal enzymatic reaction, with the output DNA capable of quantification by nanopore measurement according to the translocation frequency. Based on this methodology, any cancer-specific miRNAs can potentially be specifically amplified and detected by changing the nucleotide sequences of the DNA template and primer according to the target miRNA.

Another interesting aspect of DNA computing used in clinical applications is "theranostics," which describes a system of simultaneously combining diagnosis and therapy. Benenson et al. [63] reported an autonomous diagnosis and drug-release system using DNA computing and involving a one-to-one reaction (i.e. a single input molecule generates a single output molecule), which is incompatible with the requirements of most therapies, where the concentration of drug molecule (output) needs to be higher than that of the diagnostic molecule (input). Hiratani et al. [73, 77] demonstrated a theranostic system for SCLC using isothermal amplification from target miRNA to an antisense oligonucleotide, which was treated as a DNA-based drug (Figure 17.6d). Isothermal amplification is an emerging technique in DNA computation that allows DNA amplification at a constant temperature. A previous study described generation and amplification of a DNA–drug molecule (output) using enzyme-mediated strand displacement amplification following detection of the target (miR-20a; input) from an SCLC patient, with the generated DNA drug (oblimersen) monitored and quantified by nanopore-based measurement in real time [73, 77]. The results of nanopore quantification showed that oblimersen was amplified by >20-fold from miR-20a, thereby meeting the dosage requirement for SCLC therapy and suggesting this autonomous amplification strategy as a potential candidate for broad-range theranostics using antisense oligonucleotides.

Nanopore decoding might also contribute to molecular robotics [78, 79]. Molecular robots represent next-generation biochemical machines composed of biomaterials, such as DNA, proteins, and lipids, with the prerequisites of sensors, intelligence, and actuators proposed as requirements for the construction of such robots. To develop sensors necessary to apply a level of "intelligence" to these robots, output decoding using nanopores will be a valuable tool used to construct the necessary parts.

17.4   Conclusions

In this chapter, we described recent developments in nanopore decoding methods for DNA computation and their applications in clinical fields. Nanopore technology does not require labeling, and the decoding time is relatively rapid compared with conventional fluorescence methods. However, for laboratory-scale measurements, reconstitution of biological nanopores in lipid bilayers requires training that might be time consuming. Although the droplet contact method enables rapid, reproducible, and stable nanopore measurements, it requires experience and training. Powerful strategies based on microfabrication have been recently introduced, allowing the preparation of massive numbers of nanopore chambers in a small device to acquire the required data exclusively from the appropriate chambers. This strategy addresses current nanopore-specific issues and can be potentially applied to other nanopore technologies, including nanopore decoding of DNA computation on an industrialized scale. Nanopore technology represents a valuable methodology for enhancing the decoding of DNA computations.

References

1 Adleman, L.M. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266: 1021–1024.

2 Stojanovic, M.N. and Stefanovic, D. (2003). A deoxyribozyme-based molecular automaton. *Nat. Biotechnol.* 21: 1069–1074.

3 Ezziane, Z. (2006). DNA computing: applications and challenges. *Nanotechnology* 17: R27–R39.

4 Braich, R.S., Chelyapov, N., Johnson, C. et al. (2002). Solution of a 20-variable 3-SAT problem on a DNA computer. *Science* 296: 499–502.

5 Ouyang, Q., Kaplan, P.D., Liu, S., and Libchaber, A. (1997). DNA solution of the maximal clique problem. *Science* 278: 446–449.

6 Head, T., Rozenberg, G., Bladergroen, R.S. et al. (2000). Computing with DNA by operating on plasmids. *Biosystems* 57: 87–93.

7 Faulhammer, D., Cukras, A.R., Lipton, R.J., and Landweber, L.F. (2000). Molecular computation: RNA solutions to chess problems. *Proc. Natl. Acad. Sci. U.S.A.* 97: 1385–1389.

8 Cukras, A.R., Faulhammer, D., Lipton, R.J., and Landweber, L.F. (1999). Chess games: a model for RNA based computation. *Biosystems* 52: 35–45.

9 Arita, M. and Kobayashi, S. (2002). DNA sequence design using templates. *New Generation Computing* 20: 263–277.

10 Zhang, D.Y. and Seelig, G. (2011). Dynamic DNA nanotechnology using strand-displacement reactions. *Nat. Chem.* 3: 103–113.

11 Seelig, G., Soloveichik, D., Zhang, D.Y., and Winfree, E. (2006). Enzyme-free nucleic acid logic circuits. *Science* 314: 1585–1588.

12 Benenson, Y., Adar, R., Paz-Elizur, T. et al. (2003). DNA molecule provides a computing machine with both data and fuel. *Proc. Natl. Acad. Sci. U. S. A.* 100: 2191–2196.

13 Benenson, Y., Paz-Elizur, T., Adar, R. et al. (2001). Programmable and autonomous computing machine made of biomolecules. *Nature* 414: 430–434.

14 Balzani, V., Credi, A., and Venturi, M. (2003). Molecular logic circuits. *ChemPhysChem* 4: 49–59.

15 Okamoto, A., Tanaka, K., and Saito, I. (2004). DNA logic gates. *J. Am. Chem. Soc.* 126: 9458–9463.

16 Chatterjee, G., Dalchau, N., Muscat, R.A. et al. (2017). A spatially localized architecture for fast and modular DNA computing. *Nat. Nanotechnol.* 12: 920–927.

17 Qian, L., Winfree, E., and Bruck, J. (2011). Neural network computation with DNA strand displacement cascades. *Nature* 475: 368–372.

18 Qian, L. and Winfree, E. (2011). Scaling up digital circuit computation with DNA strand displacement cascades. *Science* 332: 1196–1201.

19 Zhang, D.Y. and Winfree, E. (2009). Control of DNA strand displacement kinetics using toehold exchange. *J. Am. Chem. Soc.* 131: 17303–17314.

20 Stojanovic, M.N., Semova, S., Kolpashchikov, D. et al. (2005). Deoxyribozyme-based ligase logic gates and their initial circuits. *J. Am. Chem. Soc.* 127: 6914–6915.

21 Erlich, Y. and Zielinski, D. (2017). DNA fountain enables a robust and efficient storage architecture. *Science* 355: 950–953.

22 Benenson, Y., Gil, B., Ben-Dor, U. et al. (2004). An autonomous molecular computer for logical control of gene expression. *Nature* 429: 423–429.

23 Modi, S., Nizak, C., Surana, S. et al. (2013). Two DNA nanomachines map pH changes along intersecting endocytic pathways inside the same cell. *Nat. Nanotechnol.* 8: 459–467.

24 Chen, Y.J., Groves, B., Muscat, R.A., and Seelig, G. (2015). DNA nanotechnology from the test tube to the cell. *Nat. Nanotechnol.* 10: 748–760.

25 Li, X., Ding, T., Sun, L., and Mao, C. (2011). Ultrasensitive DNA detection by cycle isothermal amplification based on nicking endonuclease and its application to logic gates. *Biosens. Bioelectron.* 30: 241–248.

26 Saghatelian, A., Volcker, N.H., Guckian, K.M. et al. (2003). DNA-based photonic logic gates: AND, NAND, and INHIBIT. *J. Am. Chem. Soc.* 125: 346–347.

27 Park, K.S., Seo, M.W., Jung, C. et al. (2012). Simple and universal platform for logic gate operations based on molecular beacon probes. *Small* 8: 2203–2212, 2129.

28 Ohara, M., Takinoue, M., and Kawano, R. (2017). Nanopore logic operation with DNA to RNA transcription in a droplet system. *ACS Synth. Biol.* 6: 1427–1432.

29 Yasuga, H., Inoue, K., Kawano, R. et al. (2017). Serial DNA relay in DNA logic gates by electrical fusion and mechanical splitting of droplets. *PLoS One* 12.

30 Zhang, X., Wang, Y., Fricke, B.L., and Gu, L.Q. (2014). Programming nanopore ion flow for encoded multiplex microRNA detection. *ACS Nano* 8: 3444–3450.

31 Lo, C.T., Throckmorton, D.J., Singh, A.K., and Herr, A.E. (2008). Photopolymerized diffusion-defined polyacrylamide gradient gels for on-chip protein sizing. *Lab Chip* 8: 1273–1279.

32 Herr, A.E., Hatch, A.V., Throckmorton, D.J. et al. (2007). Microfluidic immunoassays as rapid saliva-based clinical diagnostics. *Proc. Natl. Acad. Sci. U.S.A.* 104: 5268–5273.

33 Gu, L.Q., Braha, O., Conlan, S. et al. (1999). Stochastic sensing of organic analytes by a pore-forming protein containing a molecular adapter. *Nature* 398: 686–690.

34 Shi, W., Friedman, A.K., and Baker, L.A. (2017). Nanopore sensing. *Anal. Chem.* 89: 157–188.

35 Osaki, T. and Takeuchi, S. (2017). Artificial cell membrane systems for biosensing applications. *Anal. Chem.* 89: 216–231.

36 Stoloff, D.H. and Wanunu, M. (2013). Recent trends in nanopores for biotechnology. *Curr. Opin. Biotechnol.* 24: 699–704.

37 Reiner, J.E., Balijepalli, A., Robertson, J.W. et al. (2012). Disease detection and management via single nanopore-based sensors. *Chem. Rev.* 112: 6431–6451.

38 Ying, Y.L., Cao, C., and Long, Y.T. (2014). Single molecule analysis by biological nanopore sensors. *Analyst* 139: 3826–3835.

39 Ding, T., Yang, J., Pan, V., Zhao, N., Lu, Z., Ke, Y., and Zhang, C. (2020). DNA nanotechnology assisted nanopore-based analysis. *Nucleic Acids Res.* 48: 2791–2806.

40 Shoji, K. and Kawano, R. (2020). Recent Advances in Liposome-Based Molecular Robots. *Micromachines* 11: 788.

41 Astier, Y., Braha, O., and Bayley, H. (2006). Toward single molecule DNA sequencing: direct identification of ribonucleoside and deoxyribonucleoside 5′-monophosphates by using an engineered protein nanopore equipped with a molecular adapter. *J. Am. Chem. Soc.* 128: 1705–1710.

42 Lu, H., Giordano, F., and Ning, Z. (2016). Oxford nanopore MinION sequencing and genome assembly. *Genomics Proteomics Bioinf.* 14: 265–279.

43 Song, L., Hobaugh, M.R., Shustak, C. et al. (1996). Structure of staphylococcal alpha-hemolysin, a heptameric transmembrane pore. *Science* 274: 1859–1866.

44 Kasianowicz, J.J., Brandin, E., Branton, D., and Deamer, D.W. (1996). Characterization of individual polynucleotide molecules using a membrane channel. *Proc. Natl. Acad. Sci. U.S.A.* 93: 13770–13773.

45 Akeson, M., Branton, D., Kasianowicz, J.J. et al. (1999). Microsecond time-scale discrimination among polycytidylic acid, polyadenylic acid, and polyuridylic acid as homopolymers or as segments within single RNA molecules. *Biophys. J.* 77: 3227–3233.

46 Deamer, D.W. and Branton, D. (2002). Characterization of nucleic acids by nanopore analysis. *Acc. Chem. Res.* 35: 817–825.

47 Branton, D., Deamer, D.W., Marziali, A. et al. (2008). The potential and challenges of nanopore sequencing. *Nat. Biotechnol.* 26: 1146–1153.

48 Meller, A., Nivon, L., and Branton, D. (2001). Voltage-driven DNA translocations through a nanopore. *Phys. Rev. Lett.* 86: 3435–3438.

49 Meller, A. and Branton, D. (2002). Single molecule measurements of DNA transport through a nanopore. *Electrophoresis* 23: 2583–2591.

50 Benner, S., Chen, R.J., Wilson, N.A. et al. (2007). Sequence-specific detection of individual DNA polymerase complexes in real time using a nanopore. *Nat. Nanotechnol.* 2: 718–724.

51 White, R.J., Ervin, E.N., Yang, T. et al. (2007). Single ion-channel recordings using glass nanopore membranes. *J. Am. Chem. Soc.* 129: 11766–11775.

52 Butler, T.Z., Pavlenok, M., Derrington, I.M. et al. (2008). Single-molecule DNA detection with an engineered MspA protein nanopore. *Proc. Natl. Acad. Sci. U.S.A.* 105: 20647–20652.

53 Clarke, J., Wu, H.C., Jayasinghe, L. et al. (2009). Continuous base identification for single-molecule nanopore DNA sequencing. *Nat. Nanotechnol.* 4: 265–270.

54 Derrington, I.M., Butler, T.Z., Collins, M.D. et al. (2010). Nanopore DNA sequencing with MspA. *Proc. Natl. Acad. Sci. U. S. A.* 107: 16060–16065.

55 Hall, A.R., Scott, A., Rotem, D. et al. (2010). Hybrid pore formation by directed insertion of alpha-haemolysin into solid-state nanopores. *Nat. Nanotechnol.* 5: 874–877.

56 Olasagasti, F., Lieberman, K.R., Benner, S. et al. (2010). Replication of individual DNA molecules under electronic control using a protein nanopore. *Nat. Nanotechnol.* 5: 798–806.

57 Cherf, G.M., Lieberman, K.R., Rashid, H. et al. (2012). Automated forward and reverse ratcheting of DNA in a nanopore at 5-A precision. *Nat. Biotechnol.* 30: 344–348.

58 Schreiber, J., Wescoe, Z.L., Abu-Shumays, R. et al. (2013). Error rates for nanopore discrimination among cytosine, methylcytosine, and hydroxymethylcytosine along individual DNA strands. *Proc. Natl. Acad. Sci. U.S.A.* 110: 18910–18915.

59 Jain, M., Fiddes, I.T., Miga, K.H. et al. (2015). Improved data analysis for the MinION nanopore sequencer. *Nat. Methods* 12: 351–356.

60 Kawano, R., Schibel, A., Cauley, C., and White, H. (2009). Controlling the translocation of single-stranded DNA through alpha-hemolysin ion channels using viscosity. *Langmuir* 25: 1233–1237.

61 Yasuga, H., Kawano, R., Takinoue, M. et al. (2016). Logic gate operation by DNA translocation through biological nanopores. *PLoS One* 11 (2): e0149667.

62 Funakoshi, K., Suzuki, H., and Takeuchi, S. (2006). Lipid bilayer formation by contacting monolayers in a microfluidic device for membrane protein analysis. *Anal. Chem.* 78: 8169–8174.

63 Heron, A.J., Thompson, J.R., Mason, A.E., and Wallace, M.I. (2007). Direct detection of membrane channels from gels using water-in-oil droplet bilayers. *J. Am. Chem. Soc.* 129: 16042–16047.

64 Bayley, H., Cronin, B., Heron, A. et al. (2008). Droplet interface bilayers. *Mol. BioSyst.* 4: 1191–1208.

65 Kawano, R., Tsuji, Y., Sato, K. et al. (2013). Automated parallel recordings of topologically identified single ion channels. *Sci. Rep.* 3 https://doi.org/10.1038/srep01995.

66 Zhang, D.Y. and Winfree, E. (2010). Robustness and modularity properties of a non-covalent DNA catalytic reaction. *Nucleic Acids Res.* 38: 4182–4197.

67 Tsuji, Y., Kawano, R., Osaki, T. et al. (2013). Droplet split-and-contact method for high-throughput transmembrane electrical recording. *Anal. Chem.* 85: 10913–10919.

68 Ohara, M., Sekiya, Y., and Kawano, R. (2016). Hairpin DNA unzipping analysis using a biological nanopore array. *Electrochem.* 84: 338–341.

69 Benenson, Y. (2012). Biomolecular computing systems: principles, progress and potential. *Nat. Rev. Genet.* 13: 455–468.

70 Jung, C. and Ellington, A.D. (2014). Diagnostic applications of nucleic acid circuits. *Acc. Chem. Res.* 47: 1825–1835.

71 Wang, Y., Zheng, D., Tan, Q. et al. (2011). Nanopore-based detection of circulating microRNAs in lung cancer patients. *Nat. Nanotechnol.* 6: 668–674.

72 Zhang, H.L., Hiratani, M., Nagaoka, K., and Kawano, R. (2017). MicroRNA detection at femtomolar concentrations with isothermal amplification and a biological nanopore. *Nanoscale* 9: 16124–16127.

73 Hiratani, M., Ohara, M., and Kawano, R. (2017). Amplification and quantification of an antisense oligonucleotide from target microRNA using programmable DNA and a biological nanopore. *Anal. Chem.* 89: 2312–2317.

74 Di Leva, G., Garofalo, M., and Croce, C.M. (2014). MicroRNAs in cancer. *Annu. Rev. Pathol.* 9: 287–314.

75 Tian, K., He, Z., Wang, Y. et al. (2013). Designing a polycationic probe for simultaneous enrichment and detection of microRNAs in a nanopore. *ACS Nano* 7: 3962–3969.

76 Wanunu, M., Morrison, W., Rabin, Y. et al. (2010). Electrostatic focusing of unlabelled DNA into nanoscale pores using a salt gradient. *Nat. Nanotechnol.* 5: 160–165.

77 Hiratani, M. and Kawano, R. (2018). DNA logic operation with nanopore decoding to recognize MicroRNA patterns in small cell lung cancer. *Anal. Chem.* 90: 8531–8537. (DOI: https://doi.org/10.1021/acs.analchem.8b01586).

78 Hagiya, M., Konagaya, A., Kobayashi, S. et al. (2014). Molecular robots with sensors and intelligence. *Acc. Chem. Res.* 47: 1681–1690.

79 Kawano, R. (2018). Synthetic ion channels and DNA logic gates as components of molecular robots. *ChemPhysChem* 19: 359–366.

18

An Overview of DNA-Based Digital Data Storage

Xin Song [1,2,3], Shalin Shah [1,3], and John Reif [1,3]

[1] *Duke University, Pratt School of Engineering, Department of Electrical and Computer Engineering, Hudson Hall, Science Dr, Box 90291, Durham, NC 27708, USA*
[2] *Duke University, Department of Biomedical Engineering, Durham, NC 27708, USA*
[3] *Duke University, Department of Computer Science, LSRC Building D101, 308 Research Drive Campus Box 90129, Durham, NC 27708-0129, USA*

18.1 Introduction

As conventional data storage technologies are no longer approaching the trend in scaling of bit density expected by the Moore's law [1], DNA has begun to be explored as an emerging molecular storage media to potentially tackle the problem of storing exponentially growing amounts of data generated by our modern society [2]. DNA, nature's carrier of genetic information, has appealing properties that can be leveraged for digital data storage at potentially very large scales for long terms.

18.1.1 Durability and Energy Efficiency

Compared to widely used magnetic, optical, and semiconductor storage media, DNA has a much longer lifespan (up to thousands of years) when preserved under proper conditions (temperature, humidity, oxidation) [3], offering a durable and energy-efficient alternative for large-scale data archiving (i.e. the data is mostly at rest and only infrequently accessed) with small environmental footprint.

18.1.2 Density and Coding Capacity

In DNA-based digital storage, data is encoded in DNA sequences (strands) by mapping the binary bitstream to a series of nucleotides based on specific encoding rules. Because each position on the strand corresponds to one of the four natural nucleotides (A, T, C, or G), the theoretical maximum coding capacity in DNA is 2 bits/base, assuming the use of only these natural nucleotides. Owing to the small size of DNA molecules (the length of a single base is about 0.34 nm and the width of DNA in the double-helical form is about 2 nm), a key benefit of storing data in DNA is its remarkably high density when dehydrated (at scales of petabytes/gram) [4], which allows enormous amounts of data to be stored in extremely small physical volumes.

18.1.3   Availability of Supporting Technologies

Advances in DNA synthesis and sequencing technologies have enabled the use of synthetic DNA as a versatile substrate for writing and reading arbitrarily encoded information at the molecular scale [5]. Furthermore, various tools and techniques from biotechnology can be readily adapted to enable flexible and high-throughput manipulations of the DNA storage content. For example, the polymerase chain reaction (PCR) can replicate DNA to make numerous copies of the storage data or be repurposed as a simple addressing mechanism to support highly specific data retrievals [6] from large and complex DNA storage pools.

18.2   Components of a DNA Storage System

A typical DNA storage workflow (Figure 18.1) consists of data encoding, writing, storage, retrieval, and decoding [5].

18.2.1   Data Encoding

Similar to other storage technologies, information stored in DNA is subject to certain degrees of noises due to errors that arise from different stages of the pipeline [7]. As a result, a good data encoding scheme not only ensures that the digital data is converted to oligonucleotides (oligos) with desired sequence properties (e.g. balanced GC content, free of homopolymers and secondary structures, sufficient orthogonality to minimize spurious interactions) but also incorporates some form of error detection and correction [8] to protect the data by adding logical redundancies. Common errors occurring in DNA storage can be categorized into base substitutions, insertions and deletion (indels), and complete loss of DNA strands (erasures). Thanks to the field of information theory, different types of error-correcting codes (e.g. Reed–Solomon code [9], fountain code [10], LDPC code [11]) can be adapted to deal with errors that are characteristic to the DNA storage channel. Based on recent experimental work, several statistical models [7, 12] have been proposed to better understand the error distributions and molecular bias associated with the DNA storage channel, providing helpful quantitative and qualitative metrics for evaluating various trade-offs involved in the design of practical DNA storage systems.

18.2.2   Data Writing

In the writing stage, the computer-designed sequences are synthesized to actual DNA strands that form the DNA data storage pool. The standard method of DNA synthesis [13] relies on the solid-phase phosphoramidite chemistry with array-based technology to offer high throughput. The synthesis process typically generates a large number of copies (generally in the millions) of each designed strand, which provides intrinsic physical redundancy to the data stored in DNA storage. Because the bases are sequentially appended to the growing strands with certain probabilities of success, the typical chemical synthesis of DNA

Figure 18.1 Pipeline of a typical DNA-based digital storage system. Source: Based on Song and Reif [5].

usually does not exceed about 200 bases in length, while longer strands may be obtained by subsequent assembly procedure [14]. Both short [15] and long [16] strands of synthetic DNA have been used to implement robust DNA data storage *in vitro*; however, due to limitations of current synthesis and sequencing technologies, large blocks of data are typically segmented and encoded into multiple fixed-length strands. Although the phosphoramidite synthesis is a long-established technique, it is slow and costly and thus a major hurdle to enabling practical DNA storage applications. Recently, aqueous enzymatic approaches [17] are being developed to potentially improve the speed and affordability of DNA synthesis at large scales.

18.2.3 Data Storage

Once the data-encoding oligos are manufactured, they can be stored and preserved for long terms using techniques such as dehydration, lyophilization, and encapsulation, and different materials such as filter paper and airtight steel mini-capsules can be used as physical containers for DNA storage [2]. For example, studies have shown promising results of preserving synthetic DNA storage molecules in silica nanospheres [9] or magnetic nanoparticles [18] and achieved error-free data recovery from the released DNA after simulated aging experiments.

18.2.4 Data Retrieval

In a large-scale data storage system, it is desirable to have the capability of retrieving a target subset of data in an efficient and scalable manner without the need to read all data from the entire storage. Such an ability of retrieving a select subset of arbitrary data (without traversing through the whole dataset) is referred to as *random access*. In recent demonstrations of DNA-based storage [6, 15], random access operations have been implemented via selective PCR amplifications. Specifically, unique primer targets are appended at the ends of DNA data payload strands (where the encoded data resides) to effectively serve as the addresses for PCR-based random access. As an example, if a shared address is assigned to strands encoding different chunks of a data file, the corresponding pair of PCR primers could then simultaneously enrich (in an exponential fashion during PCR amplifications) only the strands belonging to this target file from a mixed solution containing lots of other strands. In addition to the primer targets, the DNA data payload strands often also contain a short index [15] to assist the intra-file ordering so that the amplified (random-accessed) oligos can be correctly reassembled *in silico* for intact data reconstruction. In order to economically scale up the PCR-based addressing mechanism for large DNA storage, we recently designed an hierarchical architecture that combines the nested and semi-nested PCR [19] to virtually organize a large pool of data-encoding oligos in a multidimensional address space while offering various previously unexplored efficient random access patterns [20]. Other than PCR-based random access, data retrieval from a DNA storage may also be achieved via techniques such as magnetic bead extraction [21]. Although DNA may theoretically archive information

at densities orders of magnitude higher than that of conventional storage media, the storage capacity of a DNA storage pool is ultimately limited by the diffusion kinetics and stochastic interactions of DNA strands in a mixed solution. As a result, practical large-scale DNA storage systems may benefit from leveraging physically isolated storage pools instead of a single pool, and the same addressing scheme could then be shared across all individual pools for improved scalability and performance [22].

18.2.5 Data Decoding

Once the target oligos are retrieved from a DNA storage pool (following PCR random access and sample purification), their sequences can be read out via DNA sequencing [23]. The most commonly used sequencing platforms include the Illumina sequencers that base on the idea of sequencing by synthesis to generate reads in a massively parallel fashion. Nanopore-based platforms such as the portable MinION sequencer can support real-time sequencing of much longer DNA strands to offer potentially higher throughput at lower cost. Because these two technologies differ considerably in terms of cost, throughput, and error rates, their uses for DNA storage implementations require uniquely tailored encoding/decoding strategies with different trade-offs in efficiency and robustness. During sequencing, each synthesized oligo in the DNA pool may be detected (read) numerous times (referred to as sequencing coverage) although the actual reads corresponding to any given strand may slightly differ due to errors from the sequencing process. In fact, DNA sequencing can be viewed as a random sampling process, and the stochasticity of PCR (including PCR random access and the bridge amplification during the Illumina sequencing process) may inevitably result in a skewed distribution of the sample pool as compared with the originally synthesized DNA pool. Thus, besides erroneous reads, some oligos might become underrepresented or even failed to be detected by sequencing. To achieve reliable data decoding and reconstruction, DNA data storage typically leverages physical and/or logical redundancies to protect data against potential errors and loss. For instance, a large copy number of oligos from synthesis introduces physical redundancy to effectively reduce the variations in PCR amplification ratio [7]. Additionally, a higher read coverage (sequencing redundancy) improves the likelihood of data reconstruction by estimating a consensus for each original sequence [24], while clustering reads based on sequence similarities may maximize the use of available reads (including reads of incorrect lengths) to recover data under low coverage [15]. Logical redundancy can be introduced by the use of error detection codes (to filter out erroneous reads) or error correction codes (to repair erroneous reads). For example, an interleaved Reed–Solomon code could deal with both errors and erasures and thus may protect a DNA storage against single-base or burst errors on individual DNA strands as well as complete loss of strands. Codes such as the fountain code and other variations of the LDPC code have the advantage of speed when it comes to storing large amounts of data [8]. Although clever and efficient algorithms have been designed for both the encoding and decoding of these well-established codes, the level of logical redundancies must be specifically

tailored to the underlying error models of the DNA storage channel to optimize the error tolerance while minimizing the overhead in computation complexity and effective coding density.

18.3   Conclusions and Outlook

As a nascent technology, DNA-based digital data storage still faces a large gap (several orders of magnitude) in data reading and writing throughput to become competitive to mainstream storage media widely used today [2]. Although challenges remain, the active research in this field has already seen a rapid progress including the recent demonstrations of an end-to-end automation system [25] and a compact digital microfluidic device [22] for programmable DNA data storage and retrieval. Although the cost barriers in DNA synthesis and sequencing may not allow practical implementations of DNA storage at large economical scales very soon, we see a strong potential in DNA as an archival storage media for its attractive properties in terms of density, longevity, energy efficiency, and environmental impact [5]. Other recent variations of *in vitro* DNA digital data storage include the use of degenerate bases [26] and composite letters [27] to achieve higher logical densities via augmented encodings as well as recording information as nicks on native genomic DNA to reduce synthesis cost [28]. DNA-based memory devices have been constructed for *in vivo* applications such as intra/extracellular event tracing and recording [29]. As science and technologies forge forward, various other potential materials and strategies also arise to offer alternative insights for molecular data storage using different classes of molecules [30, 31].

Acknowledgments

References

1 Fontana, R.E. and Decad, G.M. (2018). Moore's law realities for recording systems and memory storage components: HDD, tape, NAND, and optical. *AIP Adv.* 8 (5): 056506.

2 Ceze, L., Nivala, J., and Strauss, K. (2019). Molecular digital data storage using DNA. *Nat. Rev. Genet.* 20 (8): 456–466.

3 Bonnet, J., Colotte, M., Coudy, D. et al. (2010). Chain and conformation stability of solid-state DNA: implications for room temperature storage. *Nucleic Acids Res.* 38 (5): 1531–1546.

4 Heckel, R. (2018). An archive written in DNA. *Nat. Biotechnol.* 36 (3): 236–237.

5 Song, X. and Reif, J. (2019). Nucleic acid databases and molecular-scale computing. *ACS Nano* 13 (6): 6256–6268.

6 Tabatabaei Yazdi, S.M.H., Yuan, Y., Ma, J. et al. (2015). A rewritable, random-access DNA-based storage system. *Sci. Rep.* 5 (1): 14138.

7 Heckel, R., Mikutis, G., and Grass, R.N. (2019). A characterization of the DNA data storage channel. *Sci. Rep.* 9 (1): 9663.

8 Mitzenmacher, M. (2011). Codes: protecting data against errors and loss. In: *Algorithms Unplugged* (eds. B. Vöcking, H. Alt, M. Dietzfelbinger, et al.), 203–220. Berlin, Heidelberg: Springer.

9 Grass, R.N., Heckel, R., Puddu, M. et al. (2015). Robust chemical preservation of digital information on DNA in silica with error-correcting codes. *Angew. Chemie Int. Ed.* 54 (8): 2552–2555.

10 Erlich, Y. and Zielinski, D. (2017). DNA fountain enables a robust and efficient storage architecture. *Science* 355 (6328): 950–954.

11 Chandak, S., Tatwawadi, K., Lau, B., Mardia, J., Kubit, M., Neu, J., Griffin, P., Wootters, M., Weissman, T., and Ji, H. (2019) Improved read/write cost tradeoff in DNA-based data storage using LDPC codes. 2019 57th *Annu. Allert. Conf. Commun. Control. Comput.*, 147–156.

12 Chen, Y.-J., Takahashi, C.N., Organick, L., Stewart, K., Ang, S.D., Weiss, P., Peck, B., Seelig, G., Ceze, L., and Strauss, K. (2020) Quantifying molecular bias in DNA data storage. *Nat. Commun.* 11 (1): 3264.

13 Hughes, R.A. and Ellington, A.D. (2017). Synthetic DNA synthesis and assembly: putting the synthetic in synthetic biology. *Cold Spring Harb. Perspect. Biol.* 9 (1): a023812.

14 Lopez, R., Chen, Y.-J., Dumas Ang, S. et al. (2019). DNA assembly for nanopore data storage readout. *Nat. Commun.* 10 (1): 2933.

15 Organick, L., Ang, S.D., Chen, Y.-J. et al. (2018). Random access in large-scale DNA data storage. *Nat. Biotechnol.* 36 (3): 242–248.

16 Yazdi, S.M.H.T., Gabrys, R., and Milenkovic, O. (2017). Portable and error-free DNA-based data storage. *Sci. Rep.* 7 (1): 5011.

17 Lee, H.H., Kalhor, R., Goela, N. et al. (2019). Terminator-free template-independent enzymatic DNA synthesis for digital information storage. *Nat. Commun.* 10 (1): 2383.

18 Chen, W.D., Kohll, A.X., Nguyen, B.H. et al. (2019). Combining data longevity with high storage capacity – layer-by-layer DNA encapsulated in magnetic nanoparticles. *Adv. Funct. Mater.* 29 (28): 1901672.

19 Green, M.R. and Sambrook, J. (2019). Nested polymerase chain reaction (PCR). *Cold Spring Harb. Protoc.* 2019 (2): 175–178.

20 Song, X., Shah, S., and Reif, J. (2019) Multidimensional data organization and random access in large-scale DNA storage systems. *bioRxiv*, 743369.

21 Stewart, K., Chen, Y.-J., Ward, D. et al. (2018). A content-addressable DNA database with learned sequence encodings. *DNA Comput. Mol. Program. DNA 2018. Lect. Notes Comput. Sci.* 11145: 55–70.

22 Newman, S., Stephenson, A.P., Willsey, M. et al. (2019). High density DNA data storage library *via* dehydration with digital microfluidic retrieval. *Nat. Commun.* 10 (1): 1706.

23 Shendure, J., Balasubramanian, S., Church, G.M. et al. (2017). DNA sequencing at 40: past, present and future. *Nature* 550 (7676): 345–353.

24 Church, G.M., Gao, Y., and Kosuri, S. (2012). Next-generation digital information storage in DNA. *Science* 337 (6102): 1628–1628.

25 Takahashi, C.N., Nguyen, B.H., Strauss, K., and Ceze, L. (2019). Demonstration of end-to-end automation of DNA data storage. *Sci. Rep.* 9 (1): 4998.

26 Choi, Y., Ryu, T., Lee, A.C. et al. (2019). High information capacity DNA-based data storage with augmented encoding characters using degenerate bases. *Sci. Rep.* 9 (1): 6582.

27 Anavy, L., Vaknin, I., Atar, O., Amit, R., and Yakhini, Z. (2019). Data storage in DNA with fewer synthesis cycles using composite DNA letters. *Nat. Biotechnol.*, 1–8.

28 Tabatabaei, S.K., Wang, B., Athreya, N.B.M., Enghiad, B., Hernandez, A.G., Leburton, J.-P., Soloveichik, D., Zhao, H., and Milenkovic, O. (2019). DNA punch cards: encoding data on native DNA sequences via nicking. *Nat. Commun.* 11 (1), 1742.

29 Sheth, R.U. and Wang, H.H. (2018). DNA-based memory devices for recording cellular events. *Nat. Rev. Genet.* 19 (11): 718–732.

30 Cafferty, B.J., Ten, A.S., Fink, M.J. et al. (2019). Storage of information using small organic molecules. *ACS Cent. Sci.* 5: 911–916.

31 Rosenstein, J.K., Rose, C., Reda, S., Weber, P.M., Kim, E., Sello, J., Geiser, J., Kennedy, E., Arcadia, C., Dombroski, A., Oakley, K., Chen, S.L., Tann, H., and Rubenstein, B.M. (2019). Principles of information storage in small-molecule mixtures. *IEEE Trans. Nanobioscience* 19 (3): 378–384.

19

Interfacing Enzyme-Based and DNA-Based Computing Systems: From Simple Boolean Logic to Sophisticated Reversible Logic Systems

Evgeny Katz

Clarkson University, Department of Chemistry and Biomolecular Science, Potsdam, NY, 13699, USA

19.1 Interfacing Enzyme-Based and DNA-Based Computing Systems is a Challenging Goal: Motivations and Approaches

In the research area of biomolecular computing, DNA computing [1–7] and enzyme-based computing [8, 9] have received exceptional attention representing two most important subareas of unconventional computing. However, these subareas based on the use of different biomolecular species performing different reactions are usually isolated from each other, rarely operating together in one integrated system. They represent different approaches to chemical information processing, having different advantages and disadvantages. DNA computing is believed to be a potential alternative to electronic computers [10, 11] for some computational tasks, due to the advantage of massive parallel data processing [12], a straightforward design of relatively complex circuits [13], and affordability. Among the most obvious applications of DNA-based logic circuits is the analysis of genetic alterations that can be transformed into clinical testing of infectious and genetic diseases [14–18]. Despite advances in the development of *in vitro* selection, functional DNAs are still limited in the diversity and efficiency of catalytic reactions and are inferior to proteins in terms of affinity and diversity of ligands that DNA can recognize [19, 20]. At the same time, enzymes are proven to be selective and sensitive receptors; they are known as the best catalysts, allowing rate enhancement up to 10^{17}-fold in comparison with non-catalyzed reactions [21]. However, enzyme-based computing is experimentally limited to the systems mimicking operation of only a few concatenated logic gates [8, 9], and the network complexity was restricted by enzymes cross-reactivity and noise build [22–24]. Combining enzyme and DNA computational systems in communicating enzyme/DNA circuits may enable (i) highly selective recognition of a diverse spectrum of biological molecules or disease biomarkers, (ii) catalytic signal amplification, (iii) massive parallel data processing, and (iv) complex computational information processing, particularly for biologically generated signals. So far, mixed enzyme/DNA computational systems have been limited

to those that involve enzymes directly acting on DNA, e.g., DNA polymerases, ligases, endonuclease, etc. [25–27]. However, DNA processing enzymes cannot detect such disease biomarkers as small biological molecules, sugars, peptides, etc. On the other hand, biocomputing systems based on general enzymes (not related to DNA) were successfully used for logic processing and binary sensing of various combinations of physiological biomarkers in the binary YES/NO format [28–31]. Therefore, a universal interface for connecting enzymatic logic gates with DNA information processing circuits is needed. The present chapter introduces the very first approach to enzyme/DNA interfaces, where the primary chemical signals are processed by an enzyme logic system and then the produced intermediate output signal activates a DNA logic system operating as a downstream computing element. The whole system operates as an integrated enzyme/DNA-based computing device [32]. The enzyme logic system as well as the DNA computing part can be represented by systems of different complexity, including very sophisticated molecular "devices" performing reversible logic operations [33].

19.2 Bioelectronic Interface Transducing Logically Processed Signals from an Enzymatic System to a DNA System

Figure 19.1 shows schematically a bioelectronic interface that recognizes NADH, which is produced as an output of an enzymatic system, and releases a DNA oligonucleotide, which can be processed by a downstream DNA computing system as an input [32]. The interface is based on two modified electrodes. The first electrode communicating with the enzyme computing system ("sensing electrode" at the left) was coated with adsorbed polyethyleneimine (PEI) and pyrroloquinoline quinone (PQQ) covalently attached to the PEI thin film [34]. The immobilized PQQ served as a catalyst for electrochemical oxidation of the biocatalytically produced NADH [35]. This process resulted in the formation of a negative potential of c. $-60\,\text{mV}$ (vs. Ag/AgCl) and the corresponding current sufficient for reduction of Fe^{3+} in the Fe^{3+}-cross-linked alginate film on the second connected electrode ("releasing electrode" at the right) [36–39]. Note that Fe^{2+} cations are not capable of alginate cross-linking and their formation results in the alginate thin-film dissolution and concomitant release of the entrapped DNA molecules. The released DNA was analyzed optically (note that the DNA was labeled with a fluorescent dye, shown in Figure 19.1) and used for activating the downstream DNA computing process. The final output generated by the DNA computing was also analyzed optically.

In this study [32] two enzyme logic systems have been employed, either of which produced NADH (Figure 19.2). For binary operation of the enzyme systems, digital input **0** was defined as the absence of the corresponding substrates, whereas digital input **1** was defined as experimentally optimized concentrations of the substrates. The first system (Figure 19.2a) operated as a cascade of reactions catalyzed by three enzymes – maltose phosphorylase (MPh), hexokinase
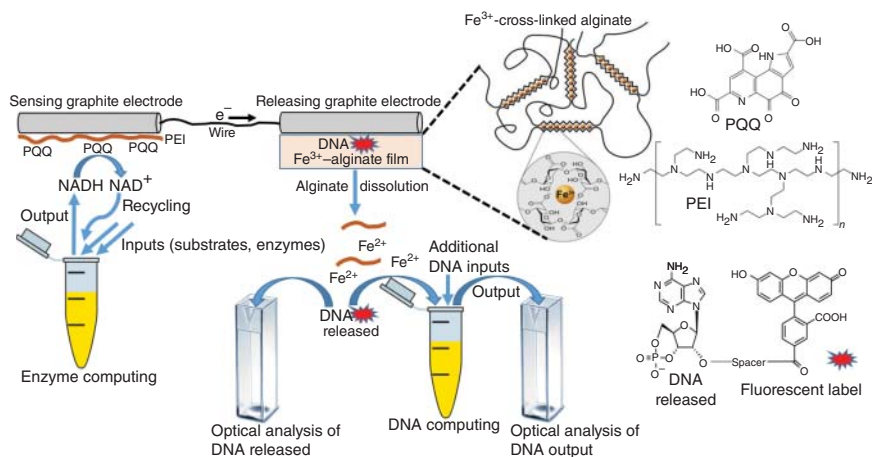
Figure 19.1 General scheme of the system operation: the enzyme computing system produces NADH as an output, which is oxidized at a PQQ-modified electrode generating a negative potential on the electrode. This sensing electrode is electrically connected to an Fe^{3+}-cross-linked alginate-modified electrode. When the negative potential is generated on the sensing electrode, it results in electrochemical reduction of Fe^{3+} to Fe^{2+} at the alginate electrode, thus leading to the alginate layer dissolution and release of the entrapped DNA. The released DNA labeled with a fluorescent dye was optically detected in the solution and used as an input for the downstream DNA computing system.

(HK), and glucose-6-phosphate dehydrogenase (G6PDH). It mimicked three concatenated Boolean AND logic gates (Figure 19.2c), and the high output signal (production of NADH) was observed only in the presence of all four input substrates (see legend to Figure 19.2). The second system (Figure 19.2b) operated as a 3-input OR gate connected to an AND gate (Figure 19.2d). For this system, the NADH production was activated in the presence of any of substrate inputs *A*, *B*, *C* with the mandatory presence of NAD^+ (input *D*). Upon enzymatic formation of NADH using any of the systems, it reacted with the PQQ electrode, producing a negative potential and reoxidized back to the NAD^+ state (Figure 19.1).

Figure 19.3 demonstrates the correct digital behavior of the enzymatic logic gate systems interacting with the PQQ electrode. The PQQ-modified sensing electrode was coupled with a reference electrode to measure the potential biocatalytically produced on the electrode (Figure 19.3a). A negative potential of c. -60 mV (digital **1**) was achieved when NADH was produced by either of the enzyme logic systems. Otherwise, the potential less negative than -10 mV (digital **0**) was measured. For the enzyme logic system mimicking three concatenated AND gates (Figure 19.2a,c), logic output **1** (c. -60 mV) was measured only in the presence of all reacting input species (input combination **1,1,1,1**). All other input combinations (15 different variants) resulted in the electrode potential less negative than -10 mV, digital **0** (Figure 19.3b). Operation of the enzyme system mimicking a 3-input OR gate followed by an AND gate (Figure 19.2b,d) resulted in output **1** (c. -60 mV) generated in the following input combinations (*A*,*B*,*C*,*D*):
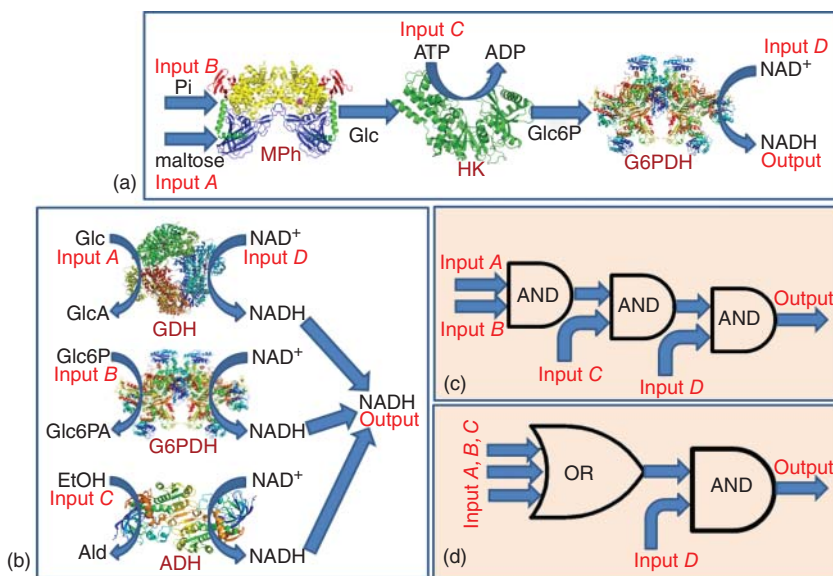
Figure 19.2 Two enzyme systems used in this study and their corresponding logic schemes. (a) A cascade of three AND gates made of maltose phosphorylase (MPh), hexokinase (HK), and glucose-6-phosphate dehydrogenase (G6PDH). The biocatalytic reaction of MPh was activated in the presence of maltose (Input *A*) and inorganic phosphate (Pi, Input *B*), resulting in glucose (Glc) and glucose-1-phosphate by-product formation. In the next reaction step catalyzed by HK, Glc is converted to glucose-6-phosphate (Glc6P) in the presence of ATP (Input *C*). Finally, Glc6P reduces NAD^+ (Input *D*) to NADH in the process biocatalyzed by G6PDH. Overall, the NADH production is only possible in the presence of all four input signals activating the enzyme-based system. (b) A combination of three parallel reactions biocatalyzed by three NAD^+-dependent enzymes: glucose dehydrogenase (GDH), G6PDH, and alcohol dehydrogenase (ADH). Each biocatalytic reaction was activated by the corresponding substrate: Glc (Input *A*), Glc6P (Input *B*), and ethanol (EtOH), (Input *C*). The NAD^+ cofactor (input *D*) was needed for all reactions; thus none of them could proceed in the absence of NAD^+. (c,d) The logic schemes corresponding to the biocatalytic cascades shown in (a) and (b), respectively. Source: (a–d) From Mailloux et al. [32]. Reproduced with the permission of John Wiley & Sons.

0,0,1,1; 0,1,0,1; 1,0,0,1; 0,1,1,1; 1,0,1,1; 1,1,0,1; 1,1,1,1, while all other input combinations resulted in output signal **0** (Figure 19.3c). The generated potential output patterns (Figure 19.3b,c) correspond to the expected operation of the enzyme logic systems.

To enable transfer of the output signal produced by the enzyme computing systems (Figure 19.2) into a DNA input signal, the PQQ electrode was connected to another electrode (Figure 19.1) that was coated with Fe^{3+}-cross-linked alginate film entrapping a fluorescently labeled DNA oligonucleotide output (DNA1: FITC-5′-TGC AGA CGT TGA AGG ATC CTC, where FITC is fluorescein isothiocyanate label bound to the DNA; see Figure 19.1). Generation of the negative potential on the PQQ electrode resulted in subsequent reduction of Fe^{3+} into Fe^{2+} at the alginate-coated electrode. It triggered the alginate film dissolution and release of DNA1. It was observed that when the potential of c. −60 mV (digital **1**
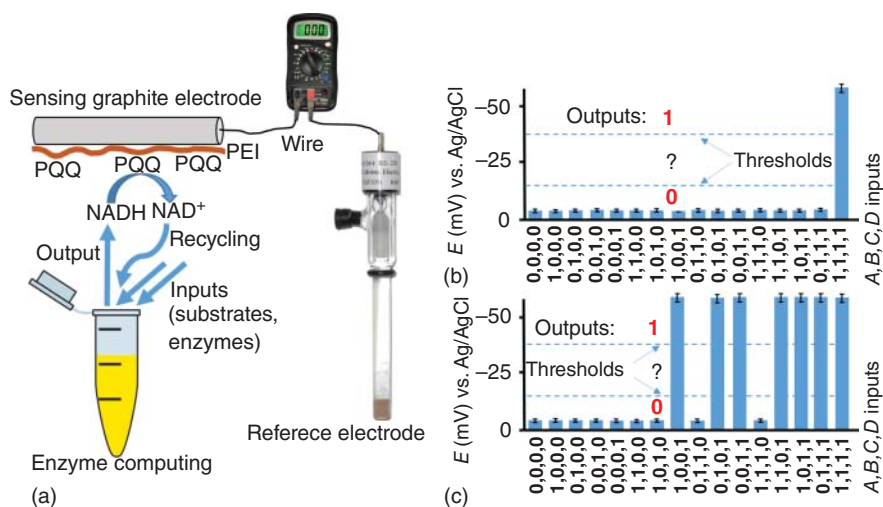
Figure 19.3 (a) Potential measurements on the sensing electrode – general scheme. (b,c) Electric potentials generated on the PQQ-modified electrode interfaced with the biocatalytic systems shown in Figure 19.2a,b, respectively, when different combinations of input signals were applied. The bars show the potential values achieved after 30 minutes of exposing the PQQ-modified electrode to the enzyme systems. The data are average of three independent experiments. The potential produced on the PQQ-modified electrode has a logarithmic dependence on the NADH concentration (according to the Nernst equation), thus resulting in very small variations of the measured potentials. Threshold lines separate logic output **0**, undefined area (?), and logic output **1**.

output of the enzymatic computing systems) was applied to the second electrode, then the alginate film was substantially degraded. At the same time, no visible changes in the film structure were observed at the potential of c. −5 mV (digital **0** output of the enzyme computing systems), on the same experimental timescale. Fluorescent signal of the solution containing released DNA1 was measured in the presence of different combinations of enzymatic system inputs (Figure 19.4). As expected, high fluorescence (digital **1**) was registered upon the electrochemically stimulated release of DNA1 in the presence of NADH. When no NADH was produced, the fluorescent signal remained low (digital **0**). The concentrations of the released (digital **1**) and leaking from the alginate film (digital **0**) DNA1 were reaching c. 5 and 0.8 nM, respectively, after 30 minutes. This result shows significant discrimination between the uncontrolled leakage and signal-stimulated release of DNA1 entrapped in the alginate-modified interface. It should be noted that there is perfect correlation between the output signals produced by the enzymatic systems in the form of the potentials (Figure 19.3) with the fluorescence of the released DNA1 (Figure 19.4). In other words, the enzyme-generated output was consistently converted into DNA1, which served as an input for DNA computing as detailed below.

Oligonucleotide DNA1 released by the interface was recognized by a 3-input deoxyribozyme AND gate (3-AND) (Figures 19.5 and 19.6). Deoxyribozyme logic gates controlled by oligonucleotide inputs are most well-developed DNA
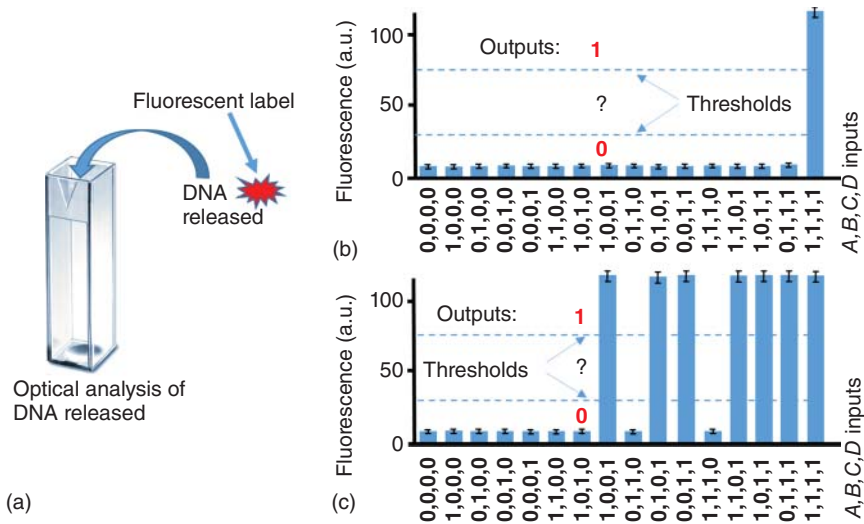
Figure 19.4 (a) Optical analysis of the DNA released (note that the DNA was labeled with a fluorescent dye). (b,c) Fluorescence signal corresponding to the dye-labeled DNA released from the alginate thin film when the PQQ-modified electrode was interfaced with the enzymatic logic gate systems shown in Figure 19.2a,b, respectively, when different combinations of input signals were applied. The bars show the fluorescence measured after 30 minutes of exposing the electrodes to the enzyme systems. The fluorescence is represented by normalized arbitrary values. The data are average of three independent experiments. Threshold lines separate logic output **0**, undefined area (?), and logic output **1**.
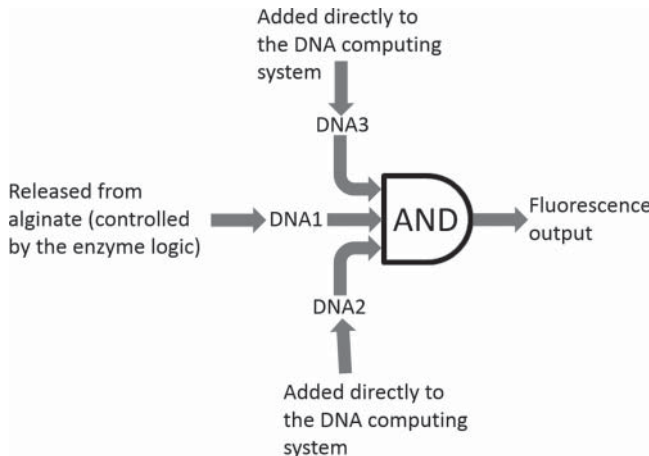
Figure 19.5 The general scheme illustrating the DNA-based 3-AND logic gate operation. More details are shown in Figure 19.6.
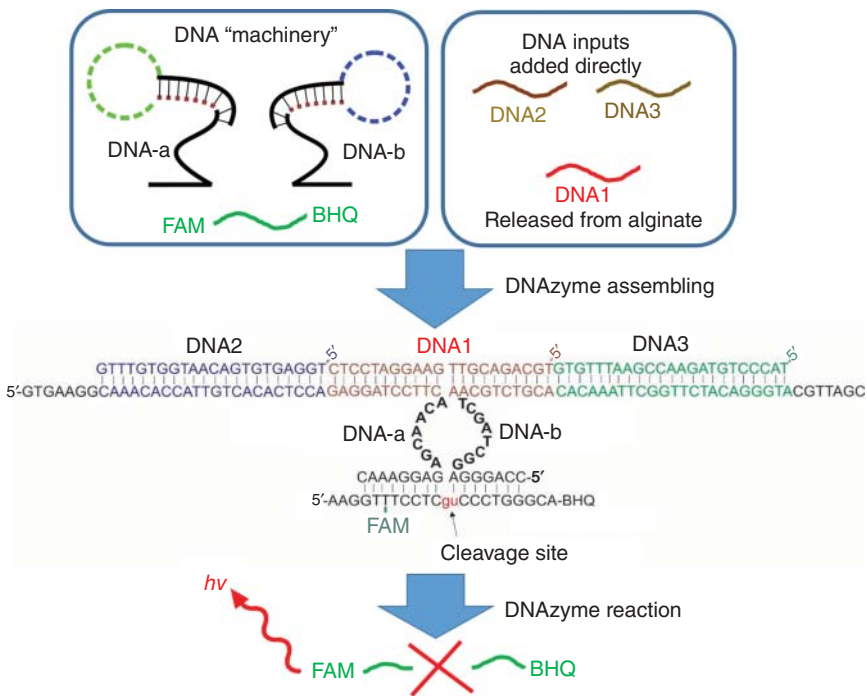
Figure 19.6 Principal scheme of a three-input deoxyribozyme AND gate. Strands DNA-a and DNA-b of the gate are dissociated in the absence of inputs. Dashed lines indicate the input-recognition fragments of the strands. The catalytic deoxyribozyme complex was formed in the presence of all three inputs (DNA1, DNA2, and DNA3). Note that DNA1 was released from the alginate electrode in the process stimulated by the enzyme logic system, while two other DNA inputs (DNA2 and DNA3) were added directly to the reacting solution performing the DNA computing operation. The DNAzyme catalytic core cleaves the fluorophore- and quencher-labeled FAM-DNA-BHQ substrate and increases fluorescent signal. Note that in this experiment DNA1 is not fluorescently labeled.

logic constructs up to date [40–47]. Indeed, such gates can be assembled in automaton that plays "tic-tac-toe" game with human [41]; they can be organized in multilayer computational cascades [43–46] and a molecular calculator with 7-segment digital display [47]. The design of 3-AND takes advantage of the concept of split (binary) deoxyribozyme sensors [48–51] and consists of two DNA strands folded in the stem-loop structures DNA-a and DNA-b (Figure 19.6). The strands are dissociated in the absence of input oligonucleotides (i.e., in the absence of DNA1, DNA2, and DNA3 inputs). However, hybridization of two oligonucleotide inputs to the loop fragments opens the hairpins, and the third input bridges DNA-a and DNA-b strands, which results in the formation of a catalytic core (Figure 19.6). The deoxyribozyme cleaves a fluorophore- and quencher-labeled substrate (FAM-BHQ in Figure 19.6), thus producing high fluorescence. In this study, DNA1, an output of the enzyme/DNA interface, was used as a bridging input for DNA-a and DNA-b. In experiments with the 3-AND gate, DNA1 without fluorescent label was used (otherwise, the release system

was operating in the way detailed above). Two other inputs (DNA2 and DNA3) mimicked the sequences of microRNAs shown to be promising molecular markers of human cancers [52].

In order to activate the DNA computing system (3-AND gate), DNA1 input was released from the alginate film upon its dissolution stimulated by the enzyme logic system, while two other DNA inputs (DNA2 and DNA3) were added directly to the reacting solution (Figures 19.5 and 19.7a). According to the truth table (Figure 19.7b), the 3-AND gate produces high fluorescence output (digital **1**) only in the presence of all three inputs being with the logic value **1**. In the performed experiments, the DNA1 input was produced *in situ* by the stimulated release (digital **1**) or leakage (digital **0**) from the alginate-modified electrode, and its concentration was set by the system as a function of logic operation of the enzyme systems. Two other inputs, DNA2 and DNA3, were either used in concentration of 10 nM or absent for digital **1** and **0**, respectively.

The full logic network includes six independent logic inputs, that is, four inputs (A, B, C, D) in the enzyme part and two inputs in the DNA part (DNA2 and DNA3 (see the oligonucleotide sequences in Figure 19.6), while DNA1 is not an independent input. Therefore, the full truth table includes $2^6 = 64$ variants of logic input combinations. Figure 19.7c shows a simplified representation of the logic process considering only the DNA logic part. Logic value **0** and **1** for the intermediate output/input DNA1 can be realized with various combinations of the enzyme inputs A, B, C, D, also being different for two enzyme-based logic systems (Figure 19.2). For simplicity and for minimizing number of experiments A, B, C, D, enzyme inputs were used in combinations **0,0,0,0** and **1,1,1,1** for realizing the DNA1 digital values **0** and **1**, respectively. This simplification is justified by very small DNA1 signal variations for all combinations of the A, B, C, D inputs generating either by output **0** or **1** (Figure 19.4). In other words, the leakage of DNA1 and the release of DNA1 are almost the same regardless of the input combinations, resulting in the leakage and release, respectively.

The correct digital response of 3-AND gate was registered at all possible DNA input combinations (Figure 19.7c). Importantly, the high output signal (last bar in Figure 19.7c corresponding to **1,1,1** input combination) could be statistically distinguished from the low output (about fourfold fluorescence increase) obtained for all other input combinations. The fluorescent results were supported by the analysis of the samples by gel electrophoresis [32]. The data proves the expected digital response of 3-AND gate and the possibility of using an electrode-released oligonucleotide for transferring the signal from enzymatic to DNA computational systems.

This study demonstrates the possibility to design an interface that enables communication between enzymatic and DNA-based computing systems. The whole system includes two individual logic subsystems (enzyme based and DNA based) connected electrically to allow the output signal produced by the enzyme logic gates operate as the input signal for the DNA logic gates. The system operated in two distinct steps, first the enzyme logic process and then the DNA logic process, representing the very first realization of the enzyme/DNA logic interface connecting a non-DNA processing enzyme computation system with DNA logic gates. This interface is "universal" because it is compatible with a variety of both
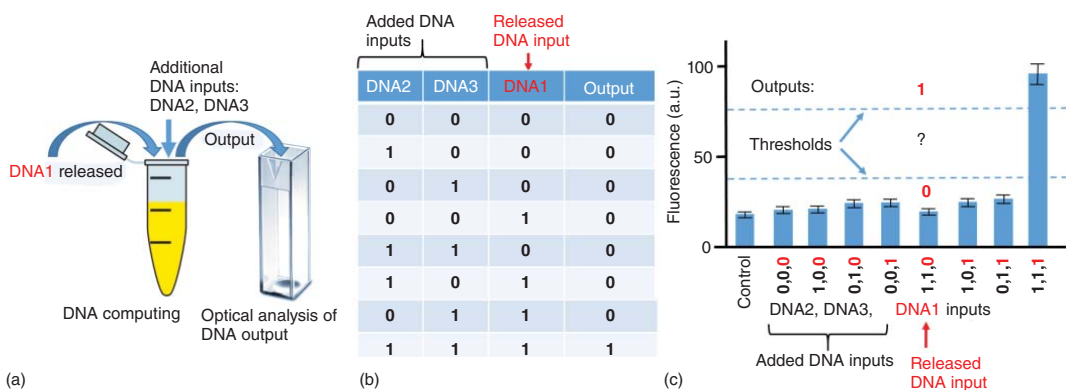
Figure 19.7 Digital performance of the DNA logic gate. (a) General scheme. (b) Truth table. (c) Fluorescent response of DNA logic gate in the presence of all possible DNA input combinations. The concentrations of the inputs were as follows. For low inputs (digital **0**): DNA2, 0 nM; DNA3, 0 nM; DNA1 was produced *in situ* with the concentration set by the system corresponding to output **0** (when substrate inputs *A,B,C,D* for the enzyme systems were **0,0,0,0**). For high inputs (digital **1**): DNA2, 10 nM; DNA3, 10 nM; DNA1 was produced *in situ* with the concentration set by the system corresponding to output **1** (when inputs *A,B,C,D* for the enzyme systems were **1,1,1,1**). Digital values for DNA1 are shown in red. Control bar corresponds to a sample containing only FAM-DNA-BHQ substrate. The bars show the fluorescence measured after 30 minutes of exposing the electrode to the enzyme systems. The fluorescence is represented by normalized arbitrary values. The data are average of three independent experiments. Threshold lines separate logic output **0**, undefined area (?), and logic output **1**.

enzymatic and DNA molecular logic circuits. NADH communicating between the enzyme system and the interface electrode allows great versatility for the selection of enzymes participating in the biocomputing process, since NADH is produced in a broad variety of reactions. In addition, it is possible to replace NADH with other reducing molecules (e.g., glucose) [53]. The deoxyribozyme gate-based computational systems are also known to show great versatility and complexity [40–47]. The limitations of the interface are the following: (i) The enzyme-based computing system must produce NADH or other reductive species as an output. (ii) The DNA-based computing system must accept nM-range concentration of oligonucleotide as an input. However, the amount of the released DNA could be increased if larger electrodes or thicker alginate films are used for the DNA entrapment and release. (iii) In its current design, the signal can be transferred only in one direction: from the enzyme to DNA system. (iv) Only one kind of DNA (or a set of DNA sequences) can be released per an electrode pair. More DNA outputs could be released in the controlled way, if a multielectrode array is applied [54]. Despite the limitations, the designed enzyme/DNA system can find some important practical applications. Indeed, the enzymatic and DNA-based computing systems used in this study proved to be relevant to diagnosis of human disorders [1, 17, 18, 55, 56], as well as to very complex information processing [40–47]. The studied system [32] reports on the final outcome results achieved after the system came to the saturation (the end of the process) similarly to most other studies in the field [8, 9, 57–60]. It would be interesting to study the time-dependent output production. The time-dependent outputs were studied experimentally and modeled theoretically for some multistep biocatalytic reactions applied for logic operations [61, 62]. Also time-dependent dissolution of alginate thin film and concomitant release of loaded substances (particularly DNA species) were reported recently [63–65]. The present system includes a number of processes with complex kinetics (biocatalytic cascades, electric potential formation, reductive dissolution of alginate, oligonucleotide release, and finally DNA reactions). Study of the combination of these time-dependent processes and their kinetics could become the subject of challenging investigation. It should be also noted that the alginate film dissolution and subsequent DNA release could be achieved in non-electrochemical systems using direct chemical communication with enzyme logic systems. Similar release processes have been studied using enzyme systems producing H_2O_2 [63–65] or citrate [66, 67] as the final output, thus excluding the electrical interface between enzyme logic gates and DNA computing systems.

19.3 The Bioelectronic Interface Connecting Enzyme-Based Reversible Logic Gates and DNA-Based Reversible Logic Gates: Realization in a Flow Device

While the Section 19.2 explains integration of the enzyme and DNA computing systems represented by relatively simple Boolean logic gates (OR, AND) and their simple combinations, the present section demonstrates scaling up the complexity of the computing processes represented by multicomponent/multistep

reversible logic operations. Notably, the reversible systems are only logically reversible, but not physically reversible. The present section is aimed at demonstrating the concerted operation of logically reversible processes performed first by an enzyme-based system and then continued in a DNA-based system activated by enzyme-generated outputs through the bioelectronic interface similar to one illustrated earlier. The system discussed here integrates *reversible* logic gate (represented by the Fredkin gate [68] with 3-input/3-output information channels) and *reversible* DNA gate (represented by the Feynman gate [69, 70] with 2-input/2-output information channels).

The (bio)molecular realizations of the Fredkin [71–74] and Feynman [75–80] gates have been already reported by several research groups, being designed and studied as single (stand-alone) computing units. However, their integration in complex circuits, particularly when one of the units operates with enzymes and the second with DNA molecules, is very challenging, and the present system is a unique biomolecular device performing reversible logic operations with enzymes and DNA molecules in the concerted way. It is particularly important to note that the modularity of the used flow system allows for easy substitution of the used logic gates with alternative logic systems that have different logic functionality. Overall, the designed system in specific, as well as the approach in general, represents one more step advancing the complexity and operational flexibility of biomolecular systems while working to mimic their electronic computing counterparts. The following sections explain the processes used to mimic the Fredkin gate with enzyme reaction and then Feynman gate with DNA reactions, as well as their integration in the continuous process with the help of a bioelectronic interface.

19.4 Enzyme-Based Fredkin Gate Processing Biomolecular Signals Prior to the Bioelectronic Interface

While many sophisticated reversible logic gates (e.g., Toffoli, Peres, and Feynman gates) can be assembled by combining together several basic Boolean operations, such as AND, XOR, etc., the Fredkin gate represents a special kind of logic processes, which cannot be easily realized by combinations of trivial Boolean functions. When the Fredkin gate is assembled from XOR, NOT, and AND logic operations, their unique logic operations translate into a very complicated set of logic combinations, which transpires into the need for complicated circuitry that is nearly impossible to directly mirror enzymatically [81]. In the Fredkin gate the logic value of Input *A* controls the directions of Inputs *B* and *C*. When the Control Input *A* has **0** value, Inputs *B* and *C* are directly copied to the corresponding Outputs *Q* and *R*; otherwise when Input *A* has **1** value, Inputs *B* and *C* switch their directions, and they are copied to Outputs *R* and *Q*, respectively (note their opposite order). Thus, the Fredkin gate represents a controlled-swap gate (CSWAP) with the swapping (exchanging) of the output channels depending on the Control channel. It should be noted that the logic value read at Output *P* indicates whether Inputs *B* and *C* are directly transposed to Outputs *Q* and *R* or switched

Input (A)	Input (B)	Input (C)	Output (P)	Output (Q)	Output (R)
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

(a)

(b) Fredkin gate
$P = A$
$Q = \bar{A}B \oplus AC$
$R = \bar{A}C \oplus AB$

(c)
$P = A$
$Q = \bar{A}B \oplus AC$
$R = \bar{A}C \oplus AB$
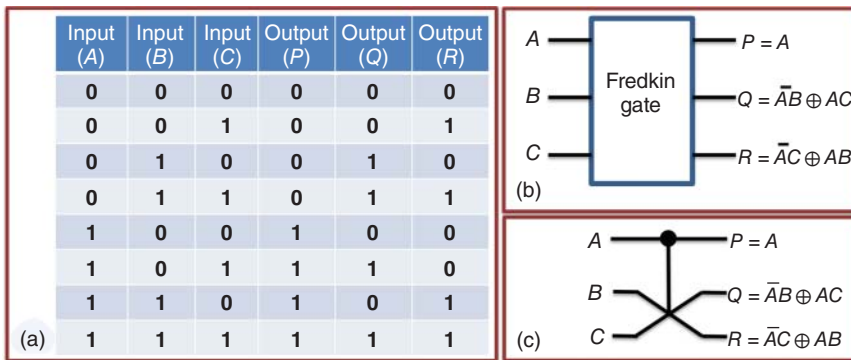
Figure 19.8 The truth table (a), block diagram (b), and equivalent electronic circuitry (c) for the Fredkin gate. Standard Boolean notations are used in the schemes [82]. Source: From Fratto and Katz [71]. Reproduced with the permission of John Wiley & Sons.

between these channels. Figure 19.8 shows the truth table and schematics of the Fredkin gate.

The Fredkin gate was recently realized in an enzyme-based system that was assembled from flow cells modified with various enzymes (Figure 19.9). For one of the input signals (Control Input A), the logic values **1** and **0** were defined as the presence of glucose (Glc) and lactate (Lac), respectively (note that it was not absence or presence of one substance, but presence of different substances to define **0** and **1** inputs (Figure 19.9). Control Input A, when represented by Lac (logic value **0**), resulted in no reaction in the cells functionalized with glucose oxidase (GOx) and horseradish peroxidase (HRP), thus producing no optical changes in the output channel and resulting in Output P being **0**. When Input A is represented by Glc (logic value **1**), the biocatalytic reactions in the cells functionalized with GOx and HRP resulted in the formation of H_2O_2 and finally results in oxidation of ABTS, (2,2′-azinobis[3-ethylbenzothiazoline-6-sulfonic acid]-diammonium salt), catalyzed by HRP to yield ABTS$_{ox}$, thus increasing the optical absorbance at $\lambda_{max} = 415$ nm, which was considered as output **1**. This process mimicked the ID (Identity, YES) gate copying Input A to Output P. Control Input A was also split and directed to the flow cells functionalized with lactate dehydrogenase (LDH) and glucose dehydrogenase (GDH) operating in parallel. Both Data Inputs B and C were represented with similar solutions containing NAD$^+$ for the logic value **1** or being a background solution only for the logic value **0**. These signals were applied to different pairs of LDH and GDH cells through the different tubing. This was possible due to the modular design of the flow system and cannot be realized in a single solution due to the inability to distinguish the difference in logic inputs if the inputs are represented by the same chemical signal. The separation of the pathways allows the logically different inputs to be represented by the same chemical substance (NAD$^+$ in the present case), since the routing of the system ensures the channels will remain separate. The two Data Inputs were each split into two pathways and passed through the cells functionalized with LDH and GDH enzymes (Figure 19.9). The LDH and GDH enzymes were mute (giving no reactions) if NAD$^+$ was absent
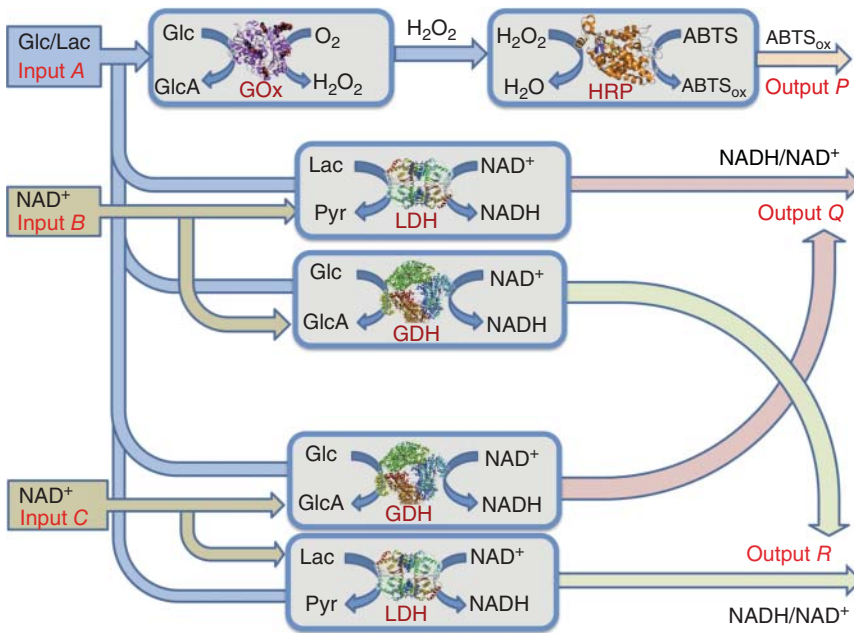
Figure 19.9 Experimental realization of the biocatalytic Fredkin gate in the flow device (GlcA, gluconic acid; Pyr, pyruvate; all other abbreviations and processes are explained in the text). Source: From Fratto and Katz [71]. Reproduced with the permission of John Wiley & Sons.

(meaning logic **0** for Inputs *B* and *C*) regardless of the logic value of Input *A*. When NAD^+ was present (logic value **1** for Inputs *B* and *C*), the LDH or GDH cells were activated depending on the presence of Lac or Glc in Input *A*. The change of Input *A* from **0** (presence of Lac) to **1** (presence of Glc) resulted in the inhibition of LDH reaction and activation of GDH reaction, thus redirecting Data Inputs *B* and *C* from one output channel to the other output channels (Figure 19.9). Each output channel (*Q* and *R*) was connected with the tubing to LDH and GDH pathways corresponding to Inputs *B* and *C* channels, thus allowing Outputs *Q* and *R* to be activated by Inputs *B* and *C*, respectively, if Input *A* is **0**. Alternatively, Outputs *Q* and *R* responded to Inputs *C* and *B*, respectively (note the switch in their order), if Input *A* is **1**, thus following the signal pattern shown in the truth table (Figure 19.8a), corresponding to the features of the Fredkin gate. The experimental realization of the Fredkin gate in the flow device is shown in Figure 19.10.

When the enzyme-based Fredkin gate was used as the primary information processing step (Figure 19.11A) and then connected to the DNA computing system, the enzymatically produced output signals in the form of NADH were measured first optically (Figure 19.11B) and then used to activate an electrochemical system (Figure 19.11C). This system includes two identical electrochemical flow cells, which have inlets connected to output channels *Q* and *R* of the Fredkin gate. Each cell has two graphite electrodes, one modified with PQQ and another
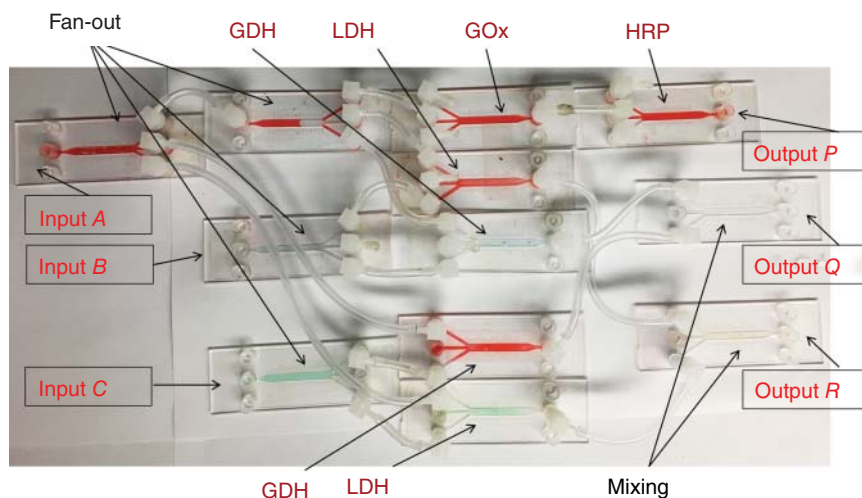
Figure 19.10 Experimental realization of the Fredkin gate (photo of the flow cell circuitry). Different colored dyes are used in this image to illustrate the experimental realization including the mixing of channels where it is applicable. Note that the colors are used for the illustration only and do not correspond to the real view of the system upon its operation. The abbreviated enzyme names were explained in the text. The fan-out channels served for mixing and distributing flows in the system. Source: From Fratto and Katz [71]. Reprinted with the permission of John Wiley & Sons.

coated with Fe^{3+}-cross-linked alginate with the loaded DNA. The only difference was in the sequence of the DNA loaded in the thin alginate film: one alginate electrode was loaded with DNA1 (labeled with a fluorescent dye FAM) and another with DNA2, which are input signals for the next information processing step – DNA-based Feynman gate.

PQQ is a well-known electrocatalyst for oxidation of NADH [35]. When Outputs Q or P appear at the logic value **1**, thus containing in the solution NADH at concentration of c. $160\,\mu M$, the PQQ-modified electrodes start NADH oxidation, and a negative potential of c. $-70\,mV$ (vs. Ag/AgCl reference) is produced on the active electrode. On the other hand, the PQQ electrodes have c. $0\,mV$ potential when the solution pumped to the electrochemical cells from the Fredkin gate does not contain NADH (logic value **0**). Since the appearance of NADH in the output solutions is directly translated into the potential formation on the PQQ electrodes, it is obvious that the pattern of the potential outputs measured for different combinations of Inputs A, B, and C repeats the same pattern observed optically for the NADH formation in Outputs Q and R (compare the bar diagrams in Figure 19.11B,C). In simpler terms, the potentials measured on the PQQ electrodes repeat the same logic as Outputs Q and R after the Fredkin gate, thus conserving the logic of the outputs, but being measured differently. The PQQ electrodes in both electrochemical cells were connected electrically to the alginate-modified electrodes loaded with DNA. Anytime when the PQQ electrode becomes negatively polarized in the presence of NADH, it produces a current, resulting in Fe^{3+} reduction in the alginate film on the connected electrode.
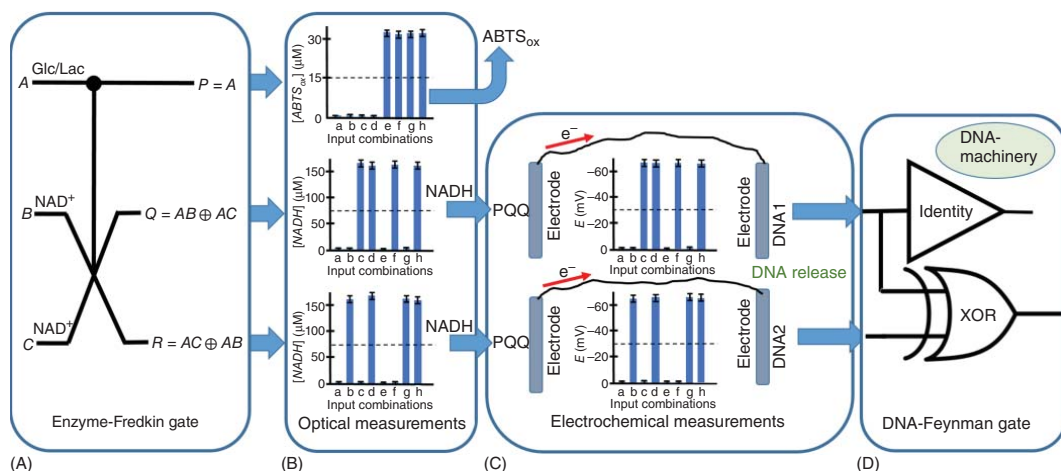
Figure 19.11 The block scheme of the entire system including (A) the enzyme-based Fredkin gate, (C) the electrochemical cells operating as an interface between the enzyme system and DNA system, and (D) the DNA-based Feynman gate composed of the ID and XOR gates operating in parallel. (B) The optically measured outputs (P, Q, and R) generated by the Fredkin gate upon application of various combinations of Inputs A, B, and C. Output P corresponding to the formation of $ABTS_{ox}$ was measured at $\lambda = 415$ nm, while Outputs Q and R corresponding to the production of NADH were measured at $\lambda = 340$ nm. The bars in the diagrams correspond to the following combinations of the A, B, C inputs: (a) **0,0,0**; (b) **0,0,1**; (c) **0,1,0**; (d) **0,1,1**; (e) **1,0,0**; (f) **1,1,0**; (g) **1,0,1**; (h) **1,1,1**. The optically measured outputs were recalculated to the corresponding concentrations of $ABTS_{ox}$ and NADH. (C) In addition to the schematically shown electrochemical cells composed of the PQQ-modified and alginate/DNA-modified electrodes, the electrochemical outputs are shown. The bars in the diagrams correspond to the following combinations of the A, B, C inputs: (a) **0,0,0**; (b) **0,0,1**; (c) **0,1,0**; (d) **0,1,1**; (e) **1,0,0**; (f) **1,1,0**; (g) **1,0,1**; (h) **1,1,1**. These outputs were measured as electrical potentials produced on the PQQ electrodes (measured vs. Ag/AgCl reference), and they were consistent with the optically measured production of NADH in Outputs Q and R of the Fredkin gate. The dash lines in all bar diagrams represent the thresholds separating **0** and **1** output signals. Source: (a–d) From Guz et al. [33]. Reproduced with the permission of John Wiley & Sons.
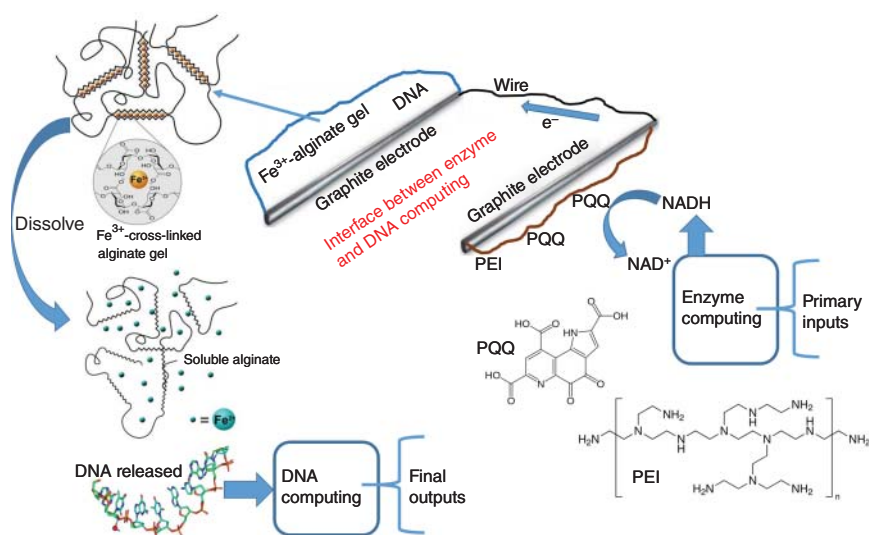
Figure 19.12 Operation of the electrochemical interface between the enzyme and DNA computing systems. The PQQ-modified electrode generated negative potential in the presence of enzymatically produced NADH. This potential resulted in the current passing from the PQQ electrode to the alginate electrode, thus resulting in the reduction of Fe^{3+} ions and dissolution of alginate hydrogel and release of the entrapped DNA. The released DNA was applied as a signal to the DNA computing system. Note that two electrochemical cells activated by Outputs Q and R generated by the enzyme system released two different DNA samples, DNA1 and DNA2, both operating as new input signals to the DNA system. Structures of PQQ and PEI are shown. Source: From Guz et al. [33]. Reproduced with thepermission of John Wiley & Sons.

When Fe^{3+} is reduced to yield Fe^{2+} cations, the alginate film is dissolved, and the loaded DNA is released to the solution (Figure 19.12). This process repeats the pattern of Outputs Q and R generated by the Fredkin gate for different combinations of primary Inputs A, B, and C, thus translating the biocatalytically produced output signal in the form of NADH to the released DNA operating as an input signal to the downstream DNA computing system. Overall, the negative potential formation on the PQQ electrode and the corresponding alginate dissolution and concomitant DNA release repeat the pattern of the logic outputs from the Fredkin gate, thus being extensions of Outputs Q and R (Figure 19.11A–C).

19.5 Reversible DNA-Based Feynman Gate Activated by Signals Produced by the Enzyme-Based Fredkin Gate

The DNA molecules released from the alginate-modified electrodes activated by logic output signals produced by the Fredkin gate were used to activate the next step in the information processing, which was represented by the DNA-based Feynman logic gate [33]. In other words, the downstream DNA-based Feynman gate was activated by the chemical outputs (represented by NADH) produced
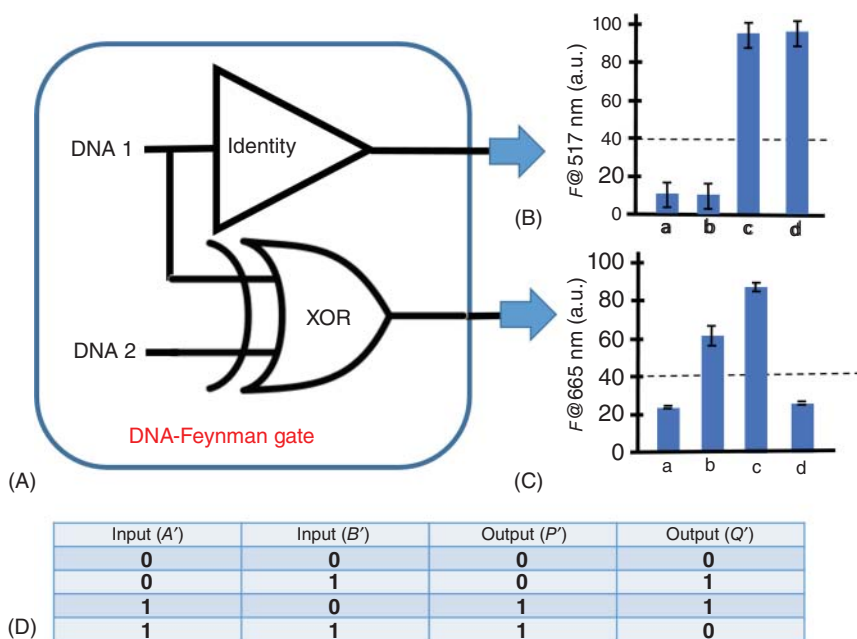
Figure 19.13 (A) Logic scheme (including the ID and XOR gates operating in parallel) of the Feynman gate and the output signals produced by the DNA system. (B) The bar diagram showing the output fluorescent signals ($\lambda_{max} = 517$ nm) produced by the DNA-based ID gate with the following DNA inputs: (a) **0,0**; (b) **0,1**; (c) **1,0**; (d) **1,1**. The output signals were measured as fluorescence produced by the FAM-labeled DNA released from the alginate electrode. Note that four input combinations are shown for consistency with the truth table (D), while the input signals processed by the ID gate are only **0** and **1**. (C) The bar diagram showing the fluorescent output signals produced by the DNA-based XOR gate with the following combinations of the DNA inputs: (a) **0,0**; (b) **0,1**; (c) **1,0**; (d) **1,1**. The output signals were measured as fluorescence ($\lambda_{max} = 665$ nm) generated by the DNA reactions. The dashed lines in both bar diagrams represent the thresholds separating **0** and **1** output signals. (D) The truth table for the Feynman (CNOT) gate. Inputs *A′* and *B′* and Outputs *P′* and *Q′* correspond to the DNA-based part of the system. Note that Inputs *A′* and *B′* in the DNA system correspond to the Outputs *Q* and *R* of the enzyme system. Source: (a–d) From Guz et al. [33]. Reproduced with the permission of John Wiley & Sons.

in situ by the upstream enzyme-based Fredkin gate, which was activated in its turn by the primary input signals *A*, *B*, and *C*. The logically reversible Feynman gate can be represented by a combination of ID and XOR gates operating in parallel (Figures 19.11D and 19.13A). The Feynman gate (also known as a controlled-NOT [CNOT] gate) includes 2-input and 2-output channels [83]. The truth table for the Feynman gate (Figure 19.13D) corresponds to the experimentally observed output signals generated by the system mimicking the Feynman gate (Figure 19.13B,C). DNA1 and DNA2 released from the alginate electrodes served as input signals for the Feynman gate. A number of other oligonucleotides (A1, A2, B1, B2 Q-F) (Figure 19.14) (see the list of oligonucleotide sequences in Appendix 19.A) served as "machinery" (meaning non-variable components of the logic system) for the Feynman gate. One of the
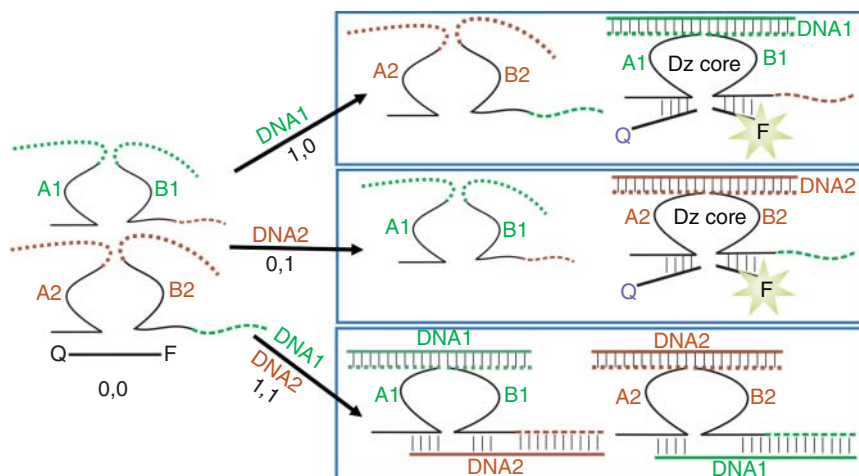
Figure 19.14 Schematic representation of the DNA reactions mimicking XOR function of the Feynman gate realized for different combinations of the DNA1 and DNA2 inputs. The produced fluorescence was used as the final output signal. The sequences of the used DNAs are given in Appendix 19.A. All DNAs except DNA1 and DNA2 were the constant (nonvariable) parts of the solution ("machinery"). Source: From Guz et al. [33]. Reproduced with the permission of John Wiley & Sons.

oligonucleotides, Q-F, included a quencher (BHQ2) and a fluorescent dye (Qz6) (Q and F, respectively) covalently bound to different ends of the oligonucleotide. The distance separating the fluorophore and quencher was short enough to allow the effective quenching of the photoexcited state of the dye, thus inhibiting its fluorescence. The DNA information processing system (Feynman gate) was realized in a single solution fed by the flows coming from both electrochemical cells releasing DNA inputs DNA1 and DNA2. When no inputs were added to the DNA "machinery" system, meaning **0,0** input combination for DNA1 and DNA2, the fluorescence observed in the system was only negligibly higher than the control (Q-F alone) due to a background reaction corresponding to the logic output **0**. When one of the DNA inputs, DNA1 or DNA2 (input combinations **0,1**; **1,0**), was added to the "machinery" system, it hybridized with the oligonucleotides present in the solution and produced a deoxyribozyme catalytic core that cut the Q-F sequence, thus separating the fluorophore and quencher. The fluorophore separated from the quencher produced a high fluorescence signal considered as logic output **1**. Upon addition of both DNA inputs (input combination **1,1**), the DNA inputs were both hybridized to the oligonucleotides present in the solution, resulting in catalytically mute structures not capable of cutting the Q-F sequence. The fluorescence was quenched, thus resulting in the logic output **0**. Minor fluorescence observed upon application of **0,0** and **1,1** input combinations originates from the incomplete quenching of the fluorophore attached to the oligonucleotide. The realized DNA reactions and the obtained logic outputs mimic the XOR gate operation, which is a part of the Feynman gate (Figure 19.13A). The second part of the Feynman gate represented by the ID gate was realized in a very simple way. One of the DNA

inputs (DNA1) was labeled with a fluorescent dye (FAM). When it was released from the alginate electrode, its fluorescence was observed and considered as the output value **1**. Obviously, the absence of this fluorescence, in case DNA1 was not released from alginate electrode, was considered as the output **0**. It should be noted that the fluorescence observed in the XOR gate ($\lambda_{max} = 665$ nm) and fluorescence measured in the ID gate ($\lambda_{max} = 517$ nm) were produced by the different dyes (Qz6 and FAM, respectively) and they did not interfere with each other. The experimental results measured with the DNA logic system for the ID and XOR channels are shown as bar diagrams demonstrating the fluorescent outputs for different combinations of the DNA inputs (Figure 19.13B,C). The experimental result is shown for the simplified logic process when only four combinations of the DNA inputs were used to activate the DNA-based Feynman gate. The full set of the data was represented with eight input combinations, starting from the primary inputs *A*, *B*, and *C* used for the activation of the enzyme-based Fredkin gate. The number of shown input combinations was reduced due to their redundancy. In experimental practice, some of the logically triggered releases produce the same effect on the DNA computing system; therefore to avoid redundancy and unnecessary complexity in the shown data, some input combinations and the corresponding output signals are not shown. Indeed, any combination of the primary inputs that produces NADH will yield a negative potential on the PQQ electrode, resulting in the release of DNA from the affected alginate electrode. Another reason for the reduced number of logic variants illustrated in Figure 19.13C is the clear demonstration of the XOR logic of the DNA computing system. If the whole set of the inputs was to be shown, the output would not be a clear representation of an XOR function, it would in fact be much more complex. It should be noted that XOR functions have been already realized with DNA systems [84–89]; however, they have never been included in the complex network similar to the present study.

19.6 Conclusions and Perspectives

Further developments in the area of biomolecular computing will certainly be motivated by practical applications. While several potential applications have already emerged (e.g., using genetically encoded computing circuits [90, 91] or simply borrowing ideas from DNA computing systems, such as their use as gene regulators in living cells [92]), novel applications are needed to justify the continuing research in the genre of biomolecular computing. In turn, the quest for novel applications will require new structural and functional features to be integrated into the biocomputing systems to yield increased complexity and enhanced functionality. The present study demonstrated possibility to increase the complexity of biomolecular computing systems, particularly by integration of parts based on the enzyme- and DNA-based reactions. While the present design does not pretend to be practically useful for any specific application, it demonstrated advantages of the modular design, where complex information processing steps can be realized by combinations of individual cells performing

simple functions. These individual cells, performing enzyme reactions that can electrochemically stimulate a release processes and DNA reactions, can be easily exchanged to other functional elements, thus allowing different functions depending on specific needs. Additionally, the number of input and output channels can be increased, thus increasing the system complexity. Considering that the present systems were realized as a combination of relatively large flow cells, the next experimental step will include miniaturization of the systems to a microfluidic lab-on-a-chip system. This lab-on-a-chip system would retain the ability to perform complex computational functions and allow for biomolecular release processes that are logically controlled by complex combinations of variable input signals. The bioelectronic (bioelectrochemical) interface connecting enzyme and DNA processes could also be miniaturized using electrode microarrays. Overall, the present approach, while already demonstrating a significant advancement in sophisticated biocomputing processes, can foster novel and complex systems that show functionality similar to microelectronic circuitries. It can be noted that it is hard to expect a miniaturization of the system that is comparable with modern electronics. Yet, when considering that the advantage of the biomolecular circuitries originates from their unique activation by metabolites representing physiological changes in living organisms, the importance cannot be overlooked. Even more challenging approach to the reverse operation, where a DNA computing system activates an enzyme logic system, should be considered as the extension of the developed system. The first step in this direction has been already done with a sensing electrode activated with a DNA input signal [65].

The concept discussed in the present chapter can go far beyond of the specific experimental realization described above. Enzyme/DNA microelectrode chips can be integrated with highly sophisticated biological systems responding to complex patterns of biological signals and releasing DNA species for computing operations. In other words, the electronic interface can potentially connect biological systems with the DNA computing. The technology for such interface is mostly available (being developed in the research areas different from the DNA computing, such as implantable microelectronics) [93]; thus the question should be asked is: "Why this should be done?" rather than "Could it be done?" Electronic chips can be functionalized with biological cells [94, 95], including nerve cells (Figure 19.15a), and the biologically activated microelectrodes can be used to trigger the DNA release followed by the DNA computations. The research efforts, based on already available technology, can go as far as implanting the chips in living organisms (even in human brain) and using them for interfacing with the DNA computing (Figure 19.15b). Obviously, this is far beyond the framework of the present biotechnology, but it illustrates possible perspectives and research extensions.

Another intriguing possibility would be to use "DNA computing" not within the binary-gate paradigm that much unconventional computing work has focused on, but rather for storing/encoding complex information as is the role of DNA in nature, especially in the framework of the biological systems/processes → electronics → DNA → electronics → biological systems/processes transduction (realizing sort-of bio-silico "android" systems), by using the developments described in the chapter.

(a)

(b)
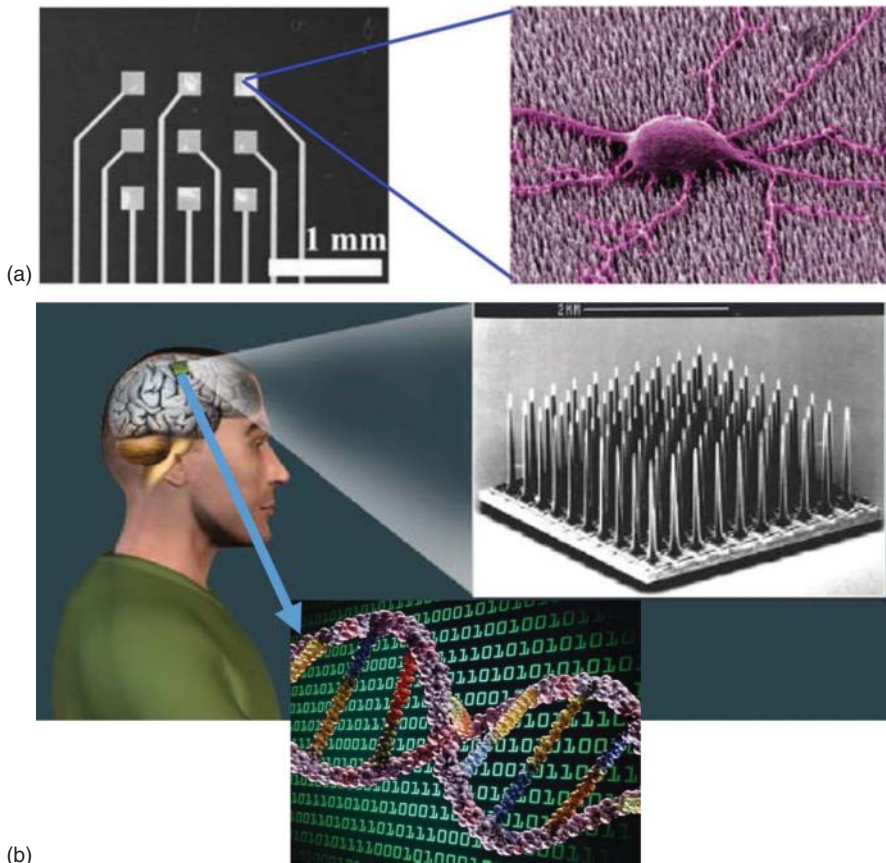
Figure 19.15 (a) Nerve cells deposited on a microelectrode array. (b) Beyond the present science–interfacing human brain with DNA computing using a microelectrode array.

19.A Appendix

19.A.1 Oligonucleotides Used in the System Mimicking Feynman Gate

DNA1:   6-FAM-5′-TAG AGT AACC T CAC ACG GAA TGT TTC
DNA2:   5′-GCTTA CAA CCAAT GAA GGA TCC TC
A1:   GAA ACA TTC CGT A CAA CGA GGTTGTGC
B1:   GAG GAT CCT TC GTTG GTG A GG CTA GCT GTG AGG TTA
         CTC TA
A2:   GAG GAT CCT TC A CAA CGA GG TTG TGC
B2:   GAA ACA TTC CGT GTT GGT GA GG CTA GCT ATT GG TTG
         TAAGC
Q-F: Qz6-CAGCACAACCguCACCAACCG-BHQ2

Nucleotides complementary to inputs DNA1 or DNA2 are underlined.

The following abbreviations are used: BHQ2 – "Black Hole Quencher®" is a trademark registered in the United States for a fluorescence quencher; 6-FAM attached to the DNA is a fluorescein derivative; Qz6 is a Quasar 670 fluorescent dye.

References

1 Gerasimova, Y.V. and Kolpashchikov, D. (2015). *Chem. Commun.* 51: 870–872.

2 Pei, R., Matamoros, E., Liu, M. et al. (2010). *Nat. Nanotechnol.* 5: 773–777.

3 Stojanovic, M.N., Stefanovic, D., and Rudchenko, S. (2014). *Acc. Chem. Res.* 47: 1845–1852.

4 Stojanovic, M.N. and Stefanovic, D. (2011). *J. Comput. Theor. Nanosci.* 8: 434–440.

5 Benenson, Y. (2012). *Nat. Rev. Genet.* 13: 455–468.

6 Kahan, M., Gil, B., Adar, R., and Shapiro, E. (2008). *Physica D* 237: 1165–1172.

7 Ezziane, Z. (2006). *Nanotechnology* 17: R27–R39.

8 Katz, E. and Privman, V. (2010). *Chem. Soc. Rev.* 39: 1835–1857.

9 Katz, E. (2015). *Curr. Opin. Biotechnol.* 34: 202–208.

10 Benenson, Y. (2009). *Mol. BioSyst.* 5: 675–685.

11 Stojanovic, M.N. (2011). *Isr. J. Chem.* 51: 99–105.

12 Adleman, L.M. (1994). *Science* 266: 1021–1024.

13 Santini, C.C., Bath, J., Turberfield, A.J., and Tyrrell, A.M. (2012). *Int. J. Mol. Sci.* 13: 5125–5137.

14 Wang, D., Fu, Y., Yan, J. et al. (2014). *Anal. Chem.* 86: 1932–1936.

15 Gil, B., Kahan-Hanum, M., Skirtenko, N. et al. (2011). *Nano Lett.* 11: 2989–2996.

16 Hemphill, J. and Deiters, A. (2013). *J. Am. Chem. Soc.* 135: 10512–10518.

17 Konry, T. and Walt, D.R. (2009). *J. Am. Chem. Soc.* 131: 13232–11323.

18 Cornett, E.M., Campbell, E.A., Gulenay, G. et al. (2012). *Angew. Chem. Int. Ed.* 51: 9075–9077.

19 Schlosser, K. and Li, Y. (2009). *Chem. Biol.* 16: 311–322.

20 Toh, S.Y., Citartan, M., Gopinath, S.C.B., and Tang, T.-H. (2014). *Biosens. Bioelectron.* 64: 392–403.

21 Radzicka, A. and Wolfenden, R. (1995). *Science* 267: 90–93.

22 Privman, V., Zavalov, O., Halámková, L. et al. (2013). *J. Phys. Chem. B* 117: 14928–14939.

23 Privman, V., Arugula, M.A., Halámek, J. et al. (2009). *J. Phys. Chem. B* 113: 5301–5310.

24 Privman, V., Strack, G., Solenov, D. et al. (2008). *J. Phys. Chem. B* 112: 11777–11784.

25 Benenson, Y., Paz-Elizur, T., Adar, R. et al. (2001). *Nature* 414: 430–434.

26 Kolpashchikov, D.M. and Stojanovic, M.N. (2005). *J. Am. Chem. Soc.* 127: 11348–11351.

27 Beyer, S. and Simmel, F.C. (2006). *Nucleic Acids Res.* 34: 1581–1587.

28 Katz, E., Wang, J., Privman, M., and Halámek, J. (2012). *Anal. Chem.* 84: 5463–5469.

29 Wang, J. and Katz, E. (2011). *Isr. J. Chem.* 51: 141–150.

30 Halámková, L., Halámek, J., Bocharova, V. et al. (2012). *Analyst* 137: 1768–1770.

31 Halámek, J., Bocharova, V., Chinnapareddy, S. et al. (2010). *Molec. BioSys.* 6: 2554–2560.

32 Mailloux, S., Gerasimova, Y.V., Guz, N. et al. (2015). *Angew. Chem. Int. Ed.* 54: 6562–6566.

33 Guz, N., Fedotova, T.A., Fratto, B.E. et al. (2016). *ChemPhysChem* 17: 2247–2255.

34 Moore, A.N.J., Katz, E., and Willner, I. (1996). *Electroanalysis* 8: 1092–1094.

35 Katz, E., Lötzbeyer, T., Schlereth, D.D. et al. (1994). *J. Electroanal. Chem.* 373: 189–200.

36 Mailloux, S., Halámek, J., and Katz, E. (2014). *Analyst* 139: 982–986.

37 Mailloux, S., Guz, N., Zakharchenko, A. et al. (2014). *J. Phys. Chem. B* 118: 6775–6784.

38 Mailloux, S., Guz, N., Gamella Carballo, M. et al. (2014). *Anal. Bioanal.Chem.* 406: 4825–4829.

39 Jin, Z., Güven, G., Bocharova, V. et al. (2012). *ACS Appl. Mater. Interfaces* 4: 466–475.

40 Stojanovic, M.N., Mitchell, T.E., and Stefanovic, D. (2002). *J. Am. Chem. Soc.* 124: 3555–3561.

41 Stojanovic, M.N. and Stefanovic, D. (2003). *Nat. Biotechnol.* 21: 1069–1074.

42 Macdonald, J., Li, Y., Sutovic, M. et al. (2006). *Nano Lett.* 6: 2598–2603.

43 Stojanovic, M.N., Semova, S., Kolpashchikov, D. et al. (2005). *J. Am. Chem. Soc.* 127: 6914–6915.

44 Yashin, R., Rudchenko, S., and Stojanovic, M.N. (2007). *J. Am. Chem. Soc.* 129: 15581–15584.

45 Brown, C.W. III, Lakin, M.R., Horwitz, E.K. et al. (2014). *Angew. Chem. Int. Ed.* 53: 7183–7187.

46 Brown, C.W. III, Lakin, M.R., Stefanovic, D., and Graves, S.W. (2014). *ChemBioChem* 15: 950–954.

47 Poje, J.E., Kastratovic, T., Macdonald, A.R. et al. (2014). *Angew. Chem. Int. Ed.* 53: 9222–9225.

48 Kolpashchikov, D.M. (2007). *ChemBioChem* 8: 2039–2042.

49 Gerasimova, Y.V., Cornett, E., and Kolpashchikov, D.M. (2010). *ChemBioChem* 11: 811–817.

50 Mokany, E., Bone, S.M., Young, P.E. et al. (2010). *J. Am. Chem. Soc.* 132: 1051–1059.

51 Gerasimova, Y.V. and Kolpashchikov, D.M. (2010). *Chem. Biol.* 17: 104–106.

52 Liu, Z., Sall, A., and Yang, D. (2008). *Int. J. Mol. Sci.* 9: 978–999.

53 Mailloux, S., Halámek, J., Halámková, L. et al. (2013). *Chem. Commun.* 49: 4755–4757.

54 Gamella, M., Zakharchenko, A., Guz, N. et al. (2017). *Electroanalysis* 29: 398–408.

55 Gerasimova, Y.V., Cornett, E.M., Edwards, E. et al. (2013). *ChemBioChem* 14: 2087–2090.

56 Gerasimova, Y.V. and Kolpashchikov, D.M. (2013). *Angew. Chem. Int. Ed.* 52: 10586–10588.

57 Katz, E. (ed.) (2012). *Molecular and Supramolecular Information Processing: From Molecular Switches to Logic Systems*. Weinheim, Germany: Wiley-VCH.

58 Katz, E. (ed.) (2012). *Biomolecular Information Processing – From Logic Systems to Smart Sensors and Actuators*. Weinheim, Germany: Wiley-VCH.

59 Szacilowski, K. (2012). *Infochemistry – Information Processing at the Nanoscale*. Chichester: Wiley.

60 de Silva, A.P. (2013). *Molecular Logic-Based Computation*. Cambridge: Royal Society of Chemistry.

61 Privman, V., Domanskyi, S., Mailloux, S. et al. (2014). *J. Phys. Chem. B* 118: 12435–12443.

62 Halámek, J., Bocharova, V., Arugula, M.A. et al. (2011). *J. Phys. Chem. B* 115: 9838–9845.

63 Okhokhonin, A.V., Domanskyi, S., Filipov, Y. et al. (2018). *Electroanalysis* 30: 426–435.

64 Gamella, M., Privman, M., Bakshi, S. et al. (2017). *ChemPhysChem* 18: 1811–1821.

65 Gamella, M., Guz, N., and Katz, E. (2016). *Electroanalysis* 28: 2692–2696.

66 Mailloux, S., Zavalov, O., Guz, N. et al. (2014). *Biomater. Sci.* 2: 184–191.

67 Bocharova, V., Zavalov, O., MacVittie, K. et al. (2012). *J. Mater. Chem.* 22: 19709–19717.

68 Fredkin, E. and Toffoli, T. (1982). *Int. J. Theor. Phys.* 21: 219–253. (First published in 1981 as a DARPA report: http://www.dtic.mil/dtic/tr/fulltext/u2/a101383.pdf).

69 O'Brien, J.L., Pryde, G.J., White, A.G. et al. (2003). *Nature* 426: 264–267.

70 Monroe, C., Meekhof, D.M., King, B.E. et al. (1995). *Phys. Rev. Lett.* 75: 4714–4717.

71 Fratto, B.E. and Katz, E. (2016). *ChemPhysChem* 17: 1046–1053.

72 Orbach, R., Remacle, F., Levine, R.D., and Willner, I. (2012). *Proc. Natl. Acad. Sci. U.S.A.* 109: 21228–21233.

73 Roy, S. and Prasad, M. (2010). *Opt. Eng.* 49 (Art. No.: 065201).

74 Klein, J.P., Leete, T.H., and Rubin, H. (1999). *Biosystems* 52: 15–23.

75 Fratto, B.E. and Katz, E. (2015). *ChemPhysChem* 16: 1405–1415.

76 Katz, E. and Fratto, B.E. (2017). Enzyme-based reversible logic gates operated in flow cells. In: *Advances in Unconventional Computing*, vol. 2 (ed. A. Adamatzky), 29–59. Springer.

77 Cervera, J. and Mafé, S. (2010). *ChemPhysChem* 11: 1654–1658.

78 Remón, P., Ferreira, R., Montenegro, J.-M. et al. (2009). *ChemPhysChem* 10: 2004–2007.

79 Moseley, F., Halámek, J., Kramer, F. et al. (2014). *Analyst* 139: 1839–1842.

80 Zhou, C., Wang, K., Fan, D. et al. (2015). *Chem. Commun.* 51: 10284–10286.

81 De Vos, A. (2010). *Reversible Computing: Fundamentals, Quantum Computing, and Applications*. Weinheim, Germany: Willey-VCH.

82 Roth, C.H. Jr., and Kinney, L.L. (2010). *Fundamentals of Logic Design*. Cengage Learning.

83 Garipelly, R., Kiran, P.M., and Kumar, A.S. (2013). *Int. J. Emerging Technol. Adv. Eng.* 3: 417–423.

84 Stojanovic, M.N. and Stefanovic, D. (2003). *J. Am. Chem. Soc.* 125: 6673–6676.

85 Gao, W., Zhang, L., Liang, R.P., and Qiu, J.D. (2015). *Chem. Eur. J.* 21: 15272–15279.

86 Gao, W., Zhang, L., Zhang, Y.M. et al. (2014). *J. Phys. Chem. C* 118: 14410–14417.

87 Zadegan, R.M., Jepsen, M.D.E., Hildebrandt, L.L. et al. (2015). *Small* 11: 1811–1817.

88 Liu, Y.Z., Dong, B.R., Wu, Z.T. et al. (2014). *Chem. Commun.* 50: 12026–12029.

89 Yang, C.N., Chen, Y.L., Lin, H.Y., and Hsu, C.-Y. (2013). *Chem. Commun.* 49: 8860–8862.

90 Roybal, K.T., Rupp, L.J., Morsut, L. et al. (2016). *Cell* 164: 770–779.

91 Haefliger, B., Prochazka, L., Angelici, B., and Benenson, Y. (2016). *Nat. Commun.* 7 (Art. No.: 10709).

92 Green, A.A., Silver, P.A., Collins, J.J., and Yin, P. (2014). *Cell* 159: 925–939.

93 Katz, E. (ed.) (2014). *Implantable Bioelectronics - Devices, Materials and Applications*. Weinheim, Germany: Wiley-VCH.

94 Elad, T., Lee, J.H., Belkin, S., and Gu, M.B. (2008). *Microb. Biotechnol.* 1: 137–148.

95 Wijffels, R.H., Buitelaar, R.M., Bucke, C., and Tramper, J. (eds.) (1996). *Immobilized Cells: Basics and Applications*. Amsterdam: Elsevier.

20

Conclusions and Perspectives: Further Research Directions and Possible Applications

Evgeny Katz

Clarkson University, Department of Chemistry and Biomolecular Science, Clarkson Avenue 8, Potsdam, NY 13699, USA

DNA computing shows great potential and has many advantages in the field of unconventional information processing, primarily due to its ability to perform millions of calculations simultaneously using molecules instead of electronics. DNA computing has the potential to execute orders of magnitude more powerful functions than traditional silicon circuitry and has a wide range of applications in medical diagnosis, *in situ* information analysis, and artificial intelligence. Various strategies for engineering DNA switches and their applications for DNA computing have been studied. Such switches are likely to drive many innovations in the fields of medicine, green chemistry, and nanotechnology. In the field of synthetic chemical circuits, DNA-based logic gates played a crucial role.

Though the use of DNA for computing can have the benefit of performing millions of operations simultaneously with very high energy efficiency, it also has several challenges. The amount of DNA required is substantial even for a simple formulation. Therefore, solving the large size problems becomes impractical owing to the requirement of a large amount of DNA. Unlike the traditional silicon-based computers in which memory reallocation is performed readily, reuse of DNA material is challenging in DNA computing, as specific designs of DNA are required.

Molecular computing systems present a design hierarchy. Starting from employing data representation methods such as fractional encoding, researchers build basic molecular circuit modules such as logic gates and clock generators. Based on that, more complex functions, e.g., digital signal processing and neural network computation, can be realized. There are various applications in silicon-based hardware, which can be implemented by the interaction of chemical materials, which will be the main focus of future research.

The recent progress in DNA/RNA nanotechnology has provided exciting opportunities to precisely manipulate naturally existing signal pathways and networks. By discovering and adopting new rules of nucleic acid-based molecular design and programming, we are now in a position to rewire signal pathways in cells *in vivo*. Nevertheless, numerous challenges must be overcome

before achieving the ultimate goal of reconstructing a modularized, transplantable, and versatile integrated circuitry systems for synthetic biology. DNA patterning systems could create a new generation of lifelike materials capable of self-construction, communication, and healing.

Majority of the DNA logic gates, however, explore only two to five layers of integration, which faces significant signal reduction as the signal propagates along the chain of communicating gates. At least partially, this problem can be mitigated by localizing logic gates in a specific order and at precise positions on a DNA tile for efficient communication as it is used in electronic processors. An energy input is required to "push" the signal through the DNA association, an approach that has not been realized yet. Alternatively, parallel computation using multiple small-scale integrated circuits can be explored. While all the technical problems can be eventually addressed given the appropriate time and effort, the future of molecular computation depends on the practical usability of DNA computers. Particularly, biocompatible and biodegradable DNA-based logic constructs can be used for manipulating biological molecules and objects (cells), which can eventually find applications in addressing biological and biomedical problems.

Despite numerous studies and success in the design of various systems for information processing, the DNA computer is far from matching the reliability of conventional silicon-based computers owing to several challenges such as poor scaling and limited ability to handle real-world problems. The comparative analysis of existing DNA computing and data storage models illustrated their pros and cons, which is opening up the new directions in materials science and biomedical applications.

For the future advancements of DNA computing, there might be two main directions. One direction is the integration of already realized DNA logic systems with the semiconductor transistors, although it is still difficult for DNA computing to compete with already mature semiconductor computers. The integration of them will combine the advantages of both fields to exhibit high efficiency in the information processing. Another direction is the smart *in vivo* bio-applications of DNA computing to nanozyme catalysis, genome-editing technique, intracellular imaging, drug load/release, and other bio-related areas. Through rational design, these combinations will definitely pave the way for real smart disease diagnostics and intracellular therapy. The prospective diagnostic and theranostic applications of molecular logic are more likely to have societal impact in the near future than DNA-based computing. Molecular logic-based screening for diseases and infections may help address inefficiencies in healthcare systems and may contribute to the realization of personalized medicine as an approach to patient care.

DNA has been recognized as a promising natural medium for information storage. Indeed, the DNA molecules were created by nature to keep the genetic code, which can be easily "written" and "read" by biomolecular systems. With information retention times that range from thousands to millions of years, volumetric density 10^3 times greater than flash memory, and energy of operation 10^8 times less, DNA is a memory storage material viable and compelling alternative to electronic memory. Recent research in the area of information storage with DNA molecules resulted in the proof-of-the-concept systems, while the practical use

of the DNA memory systems is only limited by technological problems. Both processes in the information storage with DNA, "writing" and "reading" information, are available, but they are not as simple as needed to be implemented with the present computer technology. In other words, the DNA memory is technically possible, but it is not convenient enough to be integrated with standard Si-based computers operated by end users.

Further developments in the area of biomolecular computing will certainly be motivated by practical applications. While several potential applications have already emerged (e.g., using genetically encoded computing circuits or simply borrowing ideas from DNA computing systems, such as their use as gene regulators in living cells), novel applications are needed to justify the continuing research in the genre of biomolecular computing. In turn, the quest for novel applications will require new structural and functional features to be integrated into the biocomputing systems to yield increased complexity and enhanced functionality.

So far, DNA computing is still in its infancy. Most of the reported works are conceptual with little attention given to practical application. Further efforts are required to bring the novel concepts to real-life applications.

Index