The Evolution of DNA-Based Molecular Computing

Fei Wang, Qian Li, and Chunhai Fan

Abstract The first demonstration of DNA computing was realized by Adleman in 1994, aiming to solve hard combinational problems with DNA molecules. This pioneering work initiated the evolution of the field of DNA computing during the last three decades. Up to date, the implemented functions of DNA computing have been expanded to logic operations, neural network computations, time-domain oscillator circuits, distributed computing, etc. Herein, the history of DNA computing is briefly reviewed, followed by discussions on opportunities and challenges of DNA-based molecular computing, especially from the perspective of algorithm design. Future directions and design strategies for next-generation DNA computing is also discussed.

1 A Brief History of DNA Computing

Nature-evolved DNA molecules are the primary information-carrying medium of life [1]. The computing power of DNA relies primarily on its structural potential. In 1953, Watson and Crick first proposed the double-helix structure of DNA, which marks a key step to uncover the secret of life [2]. In 1982, Seeman for the first time proposed a rational design of Holliday junction-like branched DNA structure [3], pioneering the endeavor to construct human-defined structures using DNA beyond the secret of life. This work and the subsequent progress in DNA nanotechnology provide insights and toolbox for the design of dynamic structures to implement computing algorithms.

In 1994, Adleman proposed a DNA-based algorithm to solve a Hamiltonian path problem [4]. This work for the first time demonstrated the feasibility of carrying out computations using synthetic DNA molecules, thereby signaling the start of DNA computing. The parallelism far beyond that of conventional silicon-based computers attracted wide interest. Following this work, efforts were made to explore the parallel

F. Wang · Q. Li · C. Fan (✉)
School of Chemistry and Chemical Engineering, Frontiers Science Center for Transformative Molecules, Institute of Translational Medicine, Shanghai Jiao Tong University, Shanghai 200240, China
e-mail: fanchunhai@sjtu.edu.cn

© The Author(s) 2023

N. Jonoska and E. Winfree (eds.), *Visions of DNA Nanotechnology at 40 for the Next 40*,
Natural Computing Series, https://doi.org/10.1007/978-981-19-9891-1_3

computation ability of DNA molecules to solve various mathematically complex problems, including Boolean satisfiability problem (SAT) [5–9], maximal clique problem [10], traveling salesman problem [11], etc. During this period, it was realized that the available molecular parallelism was not enough to combat the slow clock speed of biochemical operations and the redundancy required to combat the high intrinsic error rate of the operations. Evolutionary computation models were proposed to overcome the limitations, although not experimentally demonstrated [10, 12–15].

Subsequently, a variety of new molecular mechanisms for implementing computational algorithms were proposed, which greatly enriched the toolbox of DNA computing [16–24]. In 2000, as a milestone in DNA computing, an enzyme-free DNA "tweezer" was proposed by Yurke et al., which could switch between ON and OFF states through strand displacement reactions [25]. Based on accurate base pairing principle, with tunable reaction kinetics [26–31] and spontaneous execution, strand displacement reactions immediately enable various modular design of DNA computing architectures [16, 18, 32–34]. In 2004, Dirks and Pierce realized triggered amplification by hybridization chain reaction (HCR) [17], which has been broadly applied in computing [35], biosensing [36], and self-assembly [37, 38]. In 2006, Seelig et al. proposed a toehold-mediated strand displacement scheme to construct enzyme-free logic circuits [18]. In 2007, Zhang et al. developed a signal amplification reaction network that uses DNA strand as catalyst [39].

With the advances of DNA computing toolbox, various computational functions (e.g., automaton [40, 41], logic computing [42, 43], neural network computing [44, 45], cargo-sorting [46], and maze solving [35]) have been realized. Figure 1 presents a timeline of representative advances in DNA computing, classified mainly according to the realized functions and design principles. In 2003, Benenson et al. reported a molecular automaton that uses DNA both as data and fuel [40]. In 2010, Pei et al. first developed a programmable computing device to play a game [41]. In 2011, Qian and Winfree proposed a simple yet modular computing unit "seesaw" motif, with which large-scale digital computation [42] and neural network computation [44] were implemented experimentally with improved performance. In 2012, Padirac et al. developed switchable memories using DNA and DNA processing enzymes [19]. In addition to state jumps, they also implemented time-domain programming. Recently, temporal dynamics programming was further developed to implement predator-prey reaction network [47], adaptive immune response simulator [48], and enzyme-free oscillators [49]. Logic computing has also been developed with the introduction of DNA origami-based logic gates [50], spatially localized logic gates [33], integrated gene logic chip [51], single-stranded gates [52], and DNA switching circuits [53]. Meanwhile, task-oriented DNA molecular algorithms have been demonstrated in recent years, such as edge detection [54], cargo-sorting [46], maze solving [35], and pattern recognition [45].

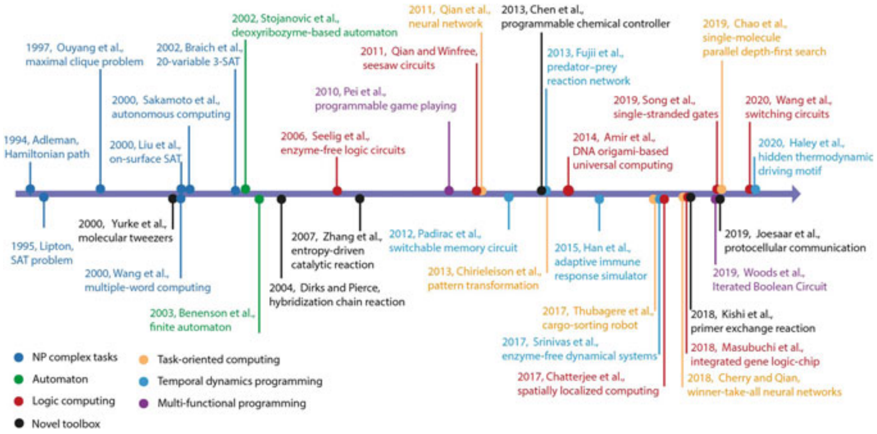

Fig. 1 Timeline of representative advances in the field of DNA computing [4–10, 16–19, 21, 25, 30, 32, 33, 35, 39–41, 44, 47–56]

2 Opportunities and Challenges

Highly ordered arrangement of physically addressable computing units at the nanoscale facilitates high-performance computing of silicon-based conventional computers. To complete a calculation task, the corresponding electronic transmission paths are activated by addressing to map the algorithm to the hardware. In contrast, DNA computing functions are mainly implemented with interactions between DNA and DNA [18], other biological molecules (e.g., RNAs [57], proteins [19, 51], and small molecules [58]), or environmental conditions (e.g., light [59], pH [60], and ions [61]). These molecule computing units are addressed chemically rather than physically, since they are mixed in solution with indistinguishable locations. To carry out a required computing task, a DNA-based chemical reaction network is programmed to run according to specified rules (algorithms) by designing and controlling the interactions between DNA and above elements. The differences in underlying implementations between DNA and solid-state computing devices (e.g., electronic and optical computers) result in unique advantages and challenges for further development of DNA computing.

2.1 *Bridge Between Matter and Information*

Computing relies on information processing, transfer, and storage. For conventional storage media, information is stored in the specified (magnetic, optical, electrical) states of matters by spatial manipulation of these storage media. In contrast, the aperiodic nucleobases make a DNA strand itself a piece of information. This not only leads to ultrahigh information density in DNA, but also bridges matter and information.

With aperiodic nucleobases and regularly sized hydrogen bonds, DNA double helix is an aperiodic crystal, which supports reliable information storage and transfer [62]. The information stored in DNA is transferred into RNA and proteins through genetic transcription and translation. The stereoregular duplex structure allows DNA to be accessed and processed by sequence-independent enzymes for self-replication and degradation. Thus, DNA links a wide matter world at the molecular level with the sequence information it carries.

2.2 Massive Parallelism

DNA computing was initially proposed to solve mathematically complex problems, such as Hamilton path problem [4], SAT problem [5], and maximum clique problem [10], taking advantage of specific and highly parallel binding between DNA molecules. Due to the stochasticity of molecular interactions, every individual molecule of the same population randomly follows certain permitted reaction paths. According to the law of large numbers, all possibilities are covered as long as the number of molecules is sufficient. Compared with algorithms that search for every possible combination one by one until the answer is found, DNA-based molecular algorithms can greatly reduce the time complexity.

It should be noted that the parallelism underlying molecular interactions relies on the availability of participating molecules. For example, a 500-node traveling salesman problem has more than 10^{1000} potential solutions, which is beyond the number of available molecules, as 1 L of 1 M solution could only provide 6×10^{23} manipulative molecules. Besides, as proposed by Back et al. [12], a huge number of DNA molecules can participate a calculation in parallel to generate a random population of candidate solutions, followed by a filtering step to remove all DNA molecules not representing a solution to the problem. Such “filtering approach” becomes infeasible as the problem size grows, since it becomes difficult to select a small number of answer products from a large number of non-answer products. Theoretically, this limitation of parallelism in problem size could be overcome by evolutionary algorithms [12, 13, 63].

2.3 Scalability

Conventional computing devices are based on the integration and spatial arrangement of same building blocks. For these systems, scaling is realized by integrating more building blocks. In contrast, DNA computing relies on specific interactions of orthogonal sequences. The 4-base coding nature and base pairing rule of DNA sequences support a rich orthogonal molecule library for large-scale computing. However, with the increasing number of required molecule types, the Hamming distances between DNA strands become smaller, leading to transient or stable unwanted binding of

DNA strands [15]. In addition, molecules participating in a basic function need to search for each other through diffusion. Increasing molecule types will increase the chance for a molecule to bind to an unwanted molecule and thus decrease the probability to be found by its target molecule, which would result in the decrease of computing speed together with increases of leakage. Therefore, scalability of DNA computing is limited to a certain finite size that cannot be extended by simply adding more computing units.

3 Directions for Future Development and Potential Approaches

3.1 Scaling-Up

The number of computing elements in a computing system determines its executable program complexity. Similar to other computational machines, the improvement of circuit scale is an important direction of evolution. The first modern electronic computer ENIAC contained 18,000 electron tubes [64]. Nowadays, tens of billions of transistors are integrated into an everyday mobile phone chip. For DNA computing, the effective scope is still solving problems that contain a small number of nodes or variables, using less than a few hundred participating DNA strands. Therefore, circuit scaling-up is an important requisite for functional evolution of DNA computing, which raises challenges including: (1) the specificity and efficiency of molecular reactions deteriorated with the increase of participating molecules; (2) the lack of a scalable computing architecture to realize the automatic mapping of complex algorithms to hardware implementations. For scaling up DNA computing, several strategies have been proposed and further explorations could be worthwhile.

3.1.1 Spatial Separation

A whole reaction can be split into different compartments by spatial separation. As a result, the molecules from different compartments are restricted from meeting each other; therefore, the reactions can be carried out efficiently in each compartment. Both semiconductor circuits and cells use the spatial separation strategy to control the material and information transmission pathways to complete complex computing tasks. In DNA computing, spatial separation has also been explored [65–68]. In 2011, Chandran et al. proposed a theoretical framework to implement parallel and scalable computation, using localized strand displacement reactions on the surface of DNA nanostructure [65]. In 2016, Genot et al. realized simultaneous observation of 10^4 reactions, by encapsulating a computational reaction system with various input conditions into droplets [67]. The molecules in each droplet reacted independently,

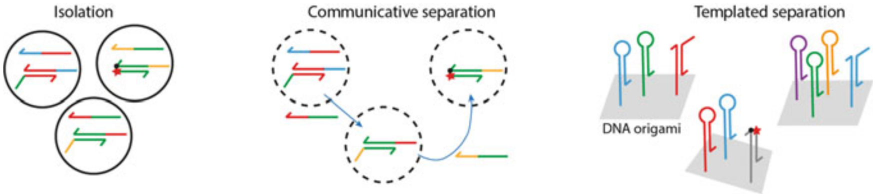

Fig. 2 Three types of spatial separation that may inspire design of DNA computing systems

since the oil film blocked molecular communication between droplets (Fig. 2a), which enabled parallel and high-resolution mapping of circuits.

Communicative separation allows different spatial positions to store different chemical information with input/output communication. In 2019, Joesaar et al. encapsulated basic logic blocks into proteinosomes as protocells to mimic the function of natural cells [55]. As the output of one reaction, the released DNA strands in one protocell could pass through protocell membranes, diffuse in solution, and then enter another protocell, triggering the reactions in the downstream as input strands. This approach was supported by protocell–protocell communications (Fig. 2b), which holds potential for high-performance DNA computing with distributed systems. However, the computation demonstrated so far is limited to several protocells with single logic gates inside. Circuit size in each protocell and communication efficiency between protocells remain to be explored.

The addressability of DNA origami at nanometer precision facilitated the confinement of molecule reactions on DNA origami surface, which enables templated separation of reactions on different origamis (Fig. 2c). In 2017, Chatterjee et al. proposed spatialized information propagation by using DNA origami as the canvas to design circuits [33]. The circuit on DNA origami receives input signals and fuel strands from the solution and releases output strands for readout, allowing signal communication between origamis. However, circuits across multiple origamis via inter-origami communication have not been realized. Due to the lack of threshold and amplifier functions in such spatial-separated systems, this strategy still faces challenges when increasing scale of the circuit. DNA self-assembly may provide alternative solutions to construct localized response elements with noise suppression and signal amplification functions.

3.1.2 Combination of Order and Disorder

Collision events of DNA molecules in solution are disordered, which brings a high degree of parallelism to DNA computation together with inaccuracy at some extent. The parallelism means subdividing a deterministic space into a probability space for one calculation. Taking the search algorithm as an example, in sequential computing, each time a possible path is explored, and a specific output is generated. For parallel DNA molecular reactions, all possible paths are explored simultaneously. As the

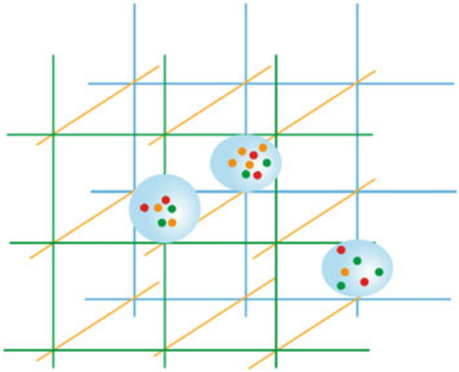

Fig. 3 Schematic illustration of high-level ordered and low-level disordered architecture

number of possible reaction paths increases, detectors with higher sensitivity and accuracy are required to obtain the solution for a problem. In addition, as the number of molecules increases in the system, the probability for a single molecule of being at the state of non-specific binding is higher, which will consequently reduce the speed and probability along the correct reaction path and increase the probability of signal leakage.

Living systems provide unique examples for coordinating reactions involving a large amount and variety of molecules. In cells, spatial compartments are utilized to confine reactions into small reaction containers. With a certain degree of fluidity, the skeletal structures of cells provide a heterogeneous environment for disordered reactions. Cells aggregate to form ordered tissues, organs, and finally the organisms. With the combination of disorder and order, organisms have evolved advanced computing capabilities (e.g., learning, thinking, and decision-making). High-precision manufacturing technologies, including DNA nanotechnology, hold potential to build highly ordered containers for DNA molecular reactions. The ordered organization of computing modules, together with the disordered molecular reactions, will provide high parallelism and overall coordination to the computing system (Fig. 3). It is possible to develop more complex artificial DNA-based computing systems in vitro with improved synthetic intelligence.

3.1.3 Reversible and Directional Reaction

Currently, the generate-and-test approach is the most widely utilized one to experimentally demonstrate DNA computing process. When the scale of potential solutions exceeds the amounts of available molecules, a problem becomes theoretically infeasible. In fact, the faithful readout of the result is also limited by the proportion of correct calculation result. A solution to a problem could be viewed as a correct assembly of DNA molecules, and a high yield of the correct DNA assembly will be of benefit to the filtration of correct answer. If the yield is too low, a solvable problem may be misinterpreted as no solution. Condon and coworkers proposed a strategy

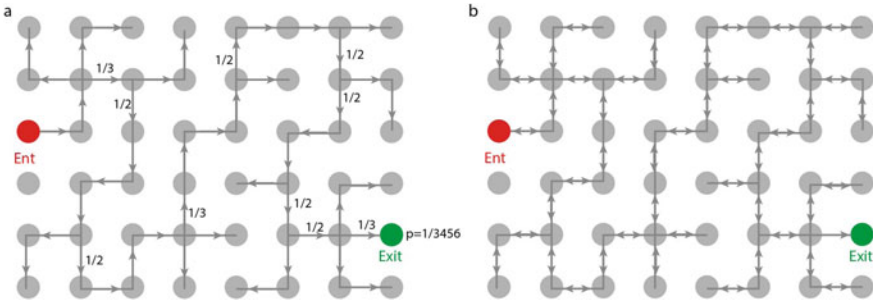

Fig. 4 Schematic illustrations of irreversible search algorithm (**a**) and reversible search algorithm (**b**) for a maze

using reversible strand displacement reactions for DNA computations that are space and energy efficient [69]. Thubagere et al. proposed a random walk-based algorithm for cargo-sorting [46], in which a single-stranded DNA robot picks up cargo irreversibly through toehold-mediated strand displacement reactions. Carrying the cargo, the robot performs a reversible random walk among adjacent tracks via toehold exchange until reaching the goal track for cargo drop-off.

The reversible motion strategy mentioned above could be extended to solve complex optimization problems. Taking a maze-solving problem for example, assuming the probability of stepping in each allowed direction is equal, the probability for a navigator to reach the exit of the maze shown in Fig. 4a is 1/3456. As each individual navigator randomly follows a correct or wrong path for the maze solution, it only requires 16 steps to reach the exit (Fig. 5a). However, the probability suggests that only ~ 0.03 nM correct assembly would be obtained with 100 nM navigators, making it hard to detect the correct solution. Using a reversible search algorithm, the navigator could return to a node that has been visited, while its last step to exit is irreversible (Fig. 4b). Through this approach, there is no dead end except the exit; thus, every navigator is capable of reaching the exit. In a simulation, more than 50% navigators reached the exit within 1000 steps (Fig. 5b). With possible repeated visit of an intermediate node, it takes remarkably more time to reach the exit for the reversible navigator than the irreversible one. However, the arrived percentage of reversible navigators exceeds that of irreversible navigators in 100 steps (Fig. 5c). Sacrificing time for higher success probability may provide an approach. For a reversible system, time sacrifice will probably lead to higher success probability with higher yields of correct DNA assemblies, which may provide solutions to complex tasks beyond the practical computing power.

3.1.4 More Efficient Molecular Searching Modes

In homogeneous solutions, molecules recognize each other mainly through diffusion. This is why DNA computing systems constructed from diffusive building blocks

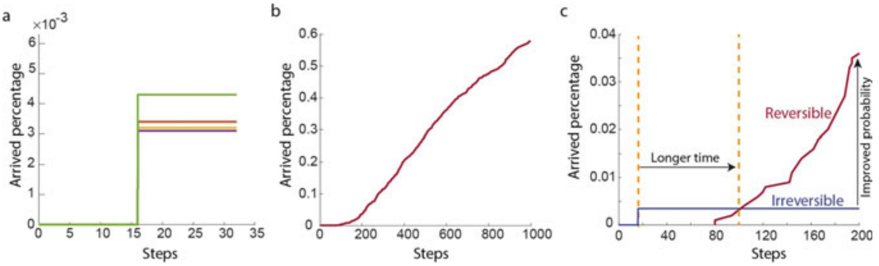

Fig. 5 Simulated overall success rates of maze solving. **a** All navigators that could reach the exit undergoes 16-step propagation in the irreversible algorithm. **b** Arrived percentage increases with step numbers in the reversible algorithm. **c** Comparison of arrived percentage in the two propagation modes within 200 steps

usually face a limitation in reaction rates toward the correct pathway. Commonly, tens of hours are needed to finish calculation when hundreds of DNA strands are involved [45]. Fast computing under low DNA concentrations could be realized by introducing new molecular searching modes. As a successful demonstration, calculations were completed in minutes under nanomolarity concentrations [33], by using DNA origami to confine the diffusion of each computing element into nanoscale regions.

Inspiration may also be obtained from the natural systems. In the cellular environment, the searching process of a protein toward its target DNA segments generally involves complex motions, including sliding, hopping, and intersegmental transfer as well as diffusion [70] (Fig. 6). When the DNA strand is stretched, protein prefers 1-dimensional lateral search. When the DNA strand is coiled, which is the native state, proteins can transfer between spatially adjacent segments. Combining these searching modes, even at very low concentrations, proteins can realize fast target search throughout the whole cell. Learning from these natural molecular searching modes, high-performance DNA computing may be developed based on novel molecular interaction mechanisms. The accurate spatial addressing property of DNA nanotechnology and tunable mechanical rigidity of DNA nanostructures may offer novel scaffolds to construct molecular machines with new molecule motion modes. With these molecular machines, more fast and efficient molecular recognition may be possible, which would realize the increase of the executable program complexity of DNA computing systems.

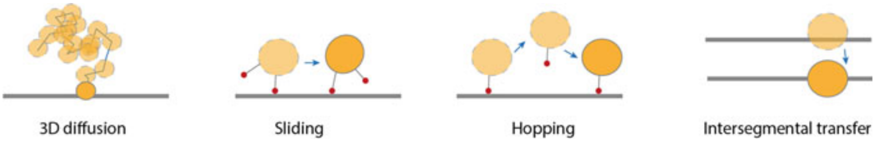

Fig. 6 Possible DNA searching modes of proteins in cells that may inspire more efficient molecular interactions for DNA computing in solution

3.2 Updating and Reusing

As a natural computer, life accumulates memories from past experiences, renews, and upgrades itself. Electrical computing chips also update their states either triggered by periodic clock or aperiodic pulse signals. Most proposed DNA computing devices are disposable and cannot be reused, because computing elements are permanently destroyed after calculation. Therefore, DNA computing is less developed in this respect, and developing renewable and reusable circuits will greatly expand the application scope of DNA computing.

3.2.1 Hardware Resetting

In DNA computing, the reset of the hardware state is based on chemical reactions, which is an important challenge that limits the sustainable use of computing devices. Spontaneous chemical reactions follow the tendency of energy change. By adding new strands to trigger reverse strand displacement [71], or by using the action of nicking enzyme to change the energy state of the system [72], the reaction can be reversed, and the input signal can be degraded, thereby realizing the reset of the computing device. Although resettable circuit implementations have been validated, the recovered concentration of input strands for next computing cycle reduced rapidly, making the circuit incapable of recycling. To perform sequential operations like electronic computers, further exploration on the design of a resettable DNA computing system is needed. Meanwhile, in combination with the parallelism of molecular reactions, highly parallel computing within one clock cycle may be developed. In this direction, to improve the reset efficiency, it is necessary to simplify the molecular structure design with further understanding of the underlying mechanisms.

3.2.2 Iteration and Update of Molecular Reaction Networks

Neuromorphic computing empowers artificial devices to learn from new inputs and realize self-update. Recently, DNA circuits-based neural network computing has been demonstrated [44, 45, 57]. However, the weight values of these neural networks were trained in silico. This one-time-use feature makes DNA circuits unable to renew themselves and thus uncapable of learning.

Evolutionary DNA algorithms have been proposed to overcome parallelism limitations by dividing the selection of the final answer from a single huge pool into recursive selections from various small pools [12, 13]. For example, in Systematic Evolution of Ligands by EXponential Enrichment (SELEX), a destructive process is performed to remove intermediate results that do not fit the constraints of a problem. These iterative selections allow evolution of computing results to approach the solution, which is different from the Adleman-style computing with a single selection at the final step [73–76]. The evolutionary strategy also provides possibilities to achieve molecular model training in machine learning. Rondelez's team has developed a series of evolutionary DNA reaction networks using a toolbox of DNA processing enzymes, i.e., polymerase, nickase, and exonuclease [19, 47, 77–79]. With a rich library of DNA processing enzymes to generate, transfer, and degrade DNA signals, it is promising to experimentally implement more complicated evolutions with DNA-based reaction networks to mimic biological systems in vitro.

4 Summary

DNA computing has evolved, slowly but steadily during the last 30 years. Despite the remarkable progress, challenges remain in many facets, such as function diversity, feasible circuit size, and computing efficiency. In particular, DNA computing relies on molecular diffusion and recognition of DNA molecules, which is fundamentally different with conventional and other type of computing systems that use a universal signal (e.g., electron or photon). We envision that next-generation DNA computing with molecular intelligence may evolve with inspiration from both natural living systems and electronic computers.

Acknowledgements We thank Erik Winfree, Darko Stefanovic, and other two anonymous reviewers for many helpful suggestions during the revision of the manuscript.

References

1. T. van der Valk, P. Pecnerova, D. Diez-Del-Molino, A. Bergstrom, J. Oppenheimer, S. Hartmann, G. Xenikoudakis, J.A. Thomas, M. Dehasque, E. Saglican, F.R. Fidan, I. Barnes, S. Liu, M. Somel, P.D. Heintzman, P. Nikolskiy, B. Shapiro, P. Skoglund, M. Hofreiter, A.M. Lister, A. Gotherstrom, L. Dalen, Million-year-old DNA sheds light on the genomic history of mammoths. Nature **591**(7849), 265–269 (2021)
2. J.D. Watson, F.H.C. Crick, Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. Nature **171**(4356), 737–738 (1953)
3. N.C. Seeman, Nucleic acid junctions and lattices. J. Theor. Biol. **99**(2), 237–247 (1982)
4. L.M. Adleman, Molecular computation of solutions to combinatorial problems. Science **266**(5187), 1021–1024 (1994)
5. R.J. Lipton, DNA solution of hard computational problems. Science **268**(5210), 542–545 (1995)
6. Q.H. Liu, L.M. Wang, A.G. Frutos, A.E. Condon, R.M. Corn, L.M. Smith, DNA computing on surfaces. Nature **403**(6766), 175–179 (2000)
7. K. Sakamoto, H. Gouzu, K. Komiya, D. Kiga, S. Yokoyama, T. Yokomori, M. Hagiya, Molecular computation by DNA hairpin formation. Science **288**(5469), 1223–1226 (2000)
8. L.M. Wang, Q.H. Liu, R.M. Corn, A.E. Condon, L.M. Smith, Multiple word DNA computing on surfaces. J. Am. Chem. Soc. **122**(31), 7435–7440 (2000)
9. R.S. Braich, N. Chelyapov, C. Johnson, P.W.K. Rothemund, L. Adleman, Solution of a 20-variable 3-SAT problem on a DNA computer. Science **296**(5567), 499–502 (2002)
10. Q. Ouyang, P.D. Kaplan, S. Liu, A. Libchaber, DNA solution of the maximal clique problem. Science **278**(5337), 446–449 (1997)

11. J.Y. Lee, S.Y. Shin, T.H. Park, B.T. Zhang, Solving traveling salesman problems with DNA molecules encoding numerical values. Biosystems **78**(1–3), 39–47 (2004)
12. T. Back, J.N. Kok, G. Rozenberg, Evolutionary computation as a paradigm for DNA-based computing. Evol. Comput., 15–40 (2002)
13. W.P. Stemmer, The evolution of molecular computation. Science **270**(5241), 1510 (1995)
14. A.D. Ellington, M.P. Robertson, K.D. James, J.C. Cox, Strategies for DNA computing. DNA Based Comp. **3**, 173–184 (1997)
15. K.D. James, A.R. Boles, D. Henckel, A.D. Ellington, The fidelity of template-directed oligonucleotide ligation and its relevance to DNA computation. Nucleic Acids Res. **26**(22), 5203–5211 (1998)
16. M.N. Stojanovic, T.E. Mitchell, D. Stefanovic, Deoxyribozyme-based logic gates. J. Am. Chem. Soc. **124**(14), 3555–3561 (2002)
17. R.M. Dirks, N.A. Pierce, Triggered amplification by hybridization chain reaction. Proc. Natl. Acad. Sci. U.S.A. **101**(43), 15275 (2004)
18. G. Seelig, D. Soloveichik, D.Y. Zhang, E. Winfree, Enzyme-free nucleic acid logic circuits. Science **314**(5805), 1585–1588 (2006)
19. A. Padirac, T. Fujii, Y. Rondelez, Bottom-up construction of in vitro switchable memories. Proc. Natl. Acad. Sci. U.S.A. **109**(47), E3212–E3220 (2012)
20. J.Y. Kishi, T.E. Schaus, N. Gopalkrishnan, F. Xuan, P. Yin, Programmable autonomous synthesis of single-stranded DNA. Nat. Chem. **10**(2), 155–164 (2018)
21. Y.J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, G. Seelig, Programmable chemical controllers made from DNA. Nat. Nanotechnol. **8**(10), 755–762 (2013)
22. C.D. Mao, T.H. LaBean, J.H. Reif, N.C. Seeman, Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. Nature **407**(6803), 493–496 (2000)
23. P. Yin, H.M. Choi, C.R. Calvert, N.A. Pierce, Programming biomolecular self-assembly pathways. Nature **451**(7176), 318–322 (2008)
24. X. Chen, A.D. Ellington, Shaping up nucleic acid computation. Curr. Opin. Biotechnol. **21**(4), 392–400 (2010)
25. B. Yurke, A.J. Turberfield, A.P. Mills, F.C. Simmel, J.L. Neumann, A DNA-fuelled molecular machine made of DNA. Nature **406**(6796), 605–608 (2000)
26. B. Yurke, A.P. Mills, Using DNA to Power Nanostructures. Genet. Prog. Evolvable Mach. **4**(2), 111–122 (2003)
27. N. Srinivas, T.E. Ouldridge, P. Sulc, J.M. Schaeffer, B. Yurke, A.A. Louis, J.P.K. Doye, E. Winfree, On the biophysics and kinetics of toehold-mediated DNA strand displacement. Nucleic Acids Res. **41**(22), 10641–10658 (2013)
28. D.Y. Zhang, E. Winfree, Control of DNA strand displacement kinetics using toehold exchange. J. Am. Chem. Soc. **131**(47), 17303–17314 (2009)
29. R.R.F. Machinek, T.E. Ouldridge, N.E.C. Haley, J. Bath, A.J. Turberfield, Programmable energy landscapes for kinetic control of DNA strand displacement. Nat. Commun. **5**(1), 1–9 (2014)
30. N.E.C. Haley, T.E. Ouldridge, I. Mullor Ruiz, A. Geraldini, A.A. Louis, J. Bath, A.J. Turberfield, Design of hidden thermodynamic driving for non-equilibrium systems via mismatch elimination during DNA strand displacement. Nat. Commun. **11**(1), 2562 (2020)
31. J. Li, A. Johnson-Buck, Y.R. Yang, W.M. Shih, H. Yan, N.G. Walter, Exploring the speed limit of toehold exchange with a cartwheeling DNA acrobat. Nat. Biotechnol. **13**(8), 723–729 (2018)
32. L. Qian, E. Winfree, A simple DNA gate motif for synthesizing large-scale circuits. J. R. Soc. Interface **8**(62), 1281–1297 (2011)
33. G. Chatterjee, N. Dalchau, R.A. Muscat, A. Phillips, G. Seelig, A spatially localized architecture for fast and modular DNA computing. Nat. Nanotechnol. **12**(9), 920–927 (2017)
34. H. Lv, Q. Li, J. Shi, C. Fan, F. Wang, Biocomputing based on DNA strand displacement reactions. ChemPhysChem. **22**(12), 1151–1166 (2021)
35. J. Chao, J. Wang, F. Wang, X. Ouyang, E. Kopperger, H. Liu, Q. Li, J. Shi, L. Wang, J. Hu, L. Wang, W. Huang, F.C. Simmel, C. Fan, Solving mazes with single-molecule DNA navigators. Nat. Mater. **18**(3), 273–279 (2019)

36. S. Bi, S.Z. Yue, S.S. Zhang, Hybridization chain reaction: a versatile molecular tool for biosensing, bioimaging, and biomedicine. Chem. Soc. Rev. **46**(14), 4281–4298 (2017)
37. W. Meng, R.A. Muscat, M.L. McKee, P.J. Milnes, A.H. El-Sagheer, J. Bath, B.G. Davis, T. Brown, R.K. O'Reilly, A.J. Turberfield, An autonomous molecular assembler for programmable chemical synthesis. Nat. Chem. **8**(6), 542–548 (2016)
38. C.A. Figg, P.H. Winegar, O.G. Hayes, C.A. Mirkin, Controlling the DNA Hybridization Chain Reaction. J. Am. Chem. Soc. **142**(19), 8596–8601 (2020)
39. D.Y. Zhang, A.J. Turberfield, B. Yurke, E. Winfree, Engineering entropy-driven reactions and networks catalyzed by DNA. Science **318**(5853), 1121–1125 (2007)
40. Y. Benenson, R. Adar, T. Paz-Elizur, Z. Livneh, E. Shapiro, DNA molecule provides a computing machine with both data and fuel. Proc. Natl. Acad. Sci. U.S.A. **100**(5), 2191–2196 (2003)
41. R. Pei, E. Matamoros, M. Liu, D. Stefanovic, M.N. Stojanovic, Training a molecular automaton to play a game. Nat Nanotechnol **5**(11), 773–777 (2010)
42. L. Qian, E. Winfree, Scaling up digital circuit computation with DNA strand displacement cascades. Science **332**(6034), 1196–1201 (2011)
43. H. Su, J. Xu, Q. Wang, F. Wang, X. Zhou, High-efficiency and integrable DNA arithmetic and logic system based on strand displacement synthesis. Nat. Commun. **10**(1), 5390 (2019)
44. L. Qian, E. Winfree, J. Bruck, Neural network computation with DNA strand displacement cascades. Nature **475**(7356), 368–372 (2011)
45. K.M. Cherry, L.L. Qian, Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. Nature **559**(7714), 370–376 (2018)
46. A.J. Thubagere, W. Li, R.F. Johnson, Z. Chen, S. Doroudi, Y.L. Lee, G. Izatt, S. Wittman, N. Srinivas, D. Woods, E. Winfree, L. Qian, A cargo-sorting DNA robot. Science **357**(6356), eaan6558 (2017)
47. T. Fujii, Y. Rondelez, Predator-prey molecular ecosystems. ACS Nano **7**(1), 27–34 (2013)
48. D. Han, C. Wu, M. You, T. Zhang, S. Wan, T. Chen, L. Qiu, Z. Zheng, H. Liang, W. Tan, A cascade reaction network mimicking the basic functional steps of adaptive immune response. Nat. Chem. **7**(10), 835–841 (2015)
49. N. Srinivas, J. Parkin, G. Seelig, E. Winfree, D. Soloveichik, Enzyme-free nucleic acid dynamical systems. Science **358**(6369), eaal2052 (2017)
50. Y. Amir, E. Ben-Ishay, D. Levner, S. Ittah, A. Abu-Horowitz, I. Bachelet, Universal computing by DNA origami robots in a living animal. Nat. Nanotechnol. **9**(5), 353–357 (2014)
51. T. Masubuchi, M. Endo, R. Iizuka, A. Iguchi, D.H. Yoon, T. Sekiguchi, H. Qi, R. Iinuma, Y. Miyazono, S. Shoji, T. Funatsu, H. Sugiyama, Y. Harada, T. Ueda, H. Tadakuma, Construction of integrated gene logic-chip. Nat. Nanotechnol. **13**(10), 933 (2018)
52. T. Song, A. Eshra, S. Shah, H. Bui, D. Fu, M. Yang, R. Mokhtar, J. Reif, Fast and compact DNA logic circuits based on single-stranded gates using strand-displacing polymerase. Nat. Nanotechnol. **14**(11), 1075–1081 (2019)
53. F. Wang, H. Lv, Q. Li, J. Li, X. Zhang, J. Shi, L. Wang, C. Fan, Implementing digital computing with DNA-based switching circuits. Nat. Commun. **11**(1), 121 (2020)
54. S.M. Chirieleison, P.B. Allen, Z.B. Simpson, A.D. Ellington, X. Chen, Pattern transformation with DNA circuits. Nat. Chem. **5**(12), 1000–1005 (2013)
55. A. Joesaar, S. Yang, B. Bogels, A. van der Linden, P. Pieters, B. Kumar, N. Dalchau, A. Phillips, S. Mann, T.F.A. de Greef, DNA-based communication in populations of synthetic protocells. Nat. Nanotechnol. **14**(4), 369–378 (2019)
56. D. Woods, D. Doty, C. Myhrvold, J. Hui, F. Zhou, P. Yin, E. Winfree, Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. Nature **567**(7748), 366–372 (2019)
57. C. Zhang, Y. Zhao, X. Xu, R. Xu, H. Li, X. Teng, Y. Du, Y. Miao, H.C. Lin, D. Han, Cancer diagnosis with DNA molecular computation. Nat. Nanotechnol. **15**(8), 709–715 (2020)
58. J. Chen, S. Zhou, J. Wen, Concatenated logic circuits based on a three-way DNA junction: a keypad-lock security system with visible readout and an automatic reset function. Angew. Chem. Int. Ed. **54**(2), 446–450 (2015)

59. X. Xiong, M. Xiao, W. Lai, L. Li, C. Fan, H. Pei, Optochemical control of DNA-switching circuits for logic and probabilistic computation. Angew. Chem. Int. Ed. **60**(7), 3397–3401 (2020)
60. Y.J. Guo, D.B. Yao, B. Zheng, X.B. Sun, X. Zhou, B. Wei, S.Y. Xiao, M. He, C.X. Li, H.J. Liang, pH-Controlled detachable DNA circuitry and its application in resettable self-assembly of spherical nucleic acids. ACS Nano **14**(7), 8317–8327 (2020)
61. R. Orbach, F. Wang, O. Lioubashevski, R.D. Levine, F. Remacle, I. Willner, A full-adder based on reconfigurable DNA-hairpin inputs and DNAzyme computing modules. Chem. Sci. **5**(9), 3381–3387 (2014)
62. E. Schrodinger, *What is Life? The Physical Aspect of the Living Cell* (University Press, Cambridge, 1951)
63. D.J.F. Jeng, I. Kim, J. Watada, DNA-based evolutionary algorithm for cable trench problem. Knowl. Based Intell. Inform. Eng. Syst. Pt 3, Proc. **4253**, 922–929 (2006)
64. A.W. Burks, From ENIAC to the stored-program computer: two revolutions in computers, in *A History of Computing in the Twentieth Century* (Elsevier, 1980), pp. 311–344
65. H. Chandran, N. Gopalkrishnan, A. Phillips, J. Reif, Localized hybridization circuits. DNA Comput. Mole. Program. **6937**, 64–83 (2011)
66. L. Qian, E. Winfree, in *Parallel and Scalable Computation and Spatial Dynamics with DNA-based Chemical Reaction Networks on a Surface (DNA Computing and Molecular Programming*. 20th International Conference, DNA 20. Proceedings: LNCS 8727), (2014), pp. 114–131
67. A.J. Genot, A. Baccouche, R. Sieskind, N. Aubert-Kato, N. Bredeche, J.F. Bartolo, V. Taly, T. Fujii, Y. Rondelez, High-resolution mapping of bifurcations in nonlinear biochemical circuits. Nat. Chem. **8**(8), 760–767 (2016)
68. M. Weitz, J. Kim, K. Kapsner, E. Winfree, E. Franco, F.C. Simmel, Diversity in the dynamical behaviour of a compartmentalized programmable biochemical oscillator. Nat. Chem. **6**(4), 295–302 (2014)
69. C. Thachuk, A. Condon, in *Space and Energy Efficient Computation with DNA Strand Displacement Systems, DNA Computing and Molecular Programming* (Berlin, Heidelberg, 2012), pp. 135–149.
70. M. Klein, T.J. Cui, I. MacRae, C. Joo, M. Depken, Skipping and sliding to optimize target search on protein-bound DNA and RNA. BioRxiv (2020). https://doi.org/10.1101/2020.06.04.133629
71. S. Garg, S. Shah, H. Bui, T. Song, R. Mokhtar, J. Reif, Renewable time-responsive DNA circuits. Small **14**(33), 1801470 (2018)
72. C. Zhang, Z. Wang, Y. Liu, J. Yang, X. Zhang, Y. Li, L. Pan, Y. Ke, H. Yan, Nicking-assisted reactant recycle to implement entropy-driven DNA circuit. J. Am. Chem. Soc. **141**(43), 17189–17197 (2019)
73. D. Faulhammer, A.R. Cukras, R.J. Lipton, L.F. Landweber, Molecular computation: RNA solutions to chess problems. Proc. Natl. Acad. Sci. U.S.A. **97**(4), 1385–1389 (2000)
74. A. Drame-Maigne, A.S. Zadorin, I. Golovkova, Y. Rondelez, Quantifying the performance of micro-compartmentalized directed evolution protocols. *Life* **10**(2), (2020)
75. R. Deaton, R.C. Murphy, J. Rose, M. Garzon, D.R. Franceschetti, S. Stevens, in A DNA based implementation of an evolutionary search for good encodings for DNA computation. Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97) (IEEE, 1997), pp. 267–271
76. J.H. Chen, D.H. Wood, Computation with biomolecules. Proc. Natl. Acad. Sci. U.S.A. **97**(4), 1328–1330 (2000)
77. A. Dramé-Maigné, R. Espada, G. McCallum, R. Sieskind, Y. Rondelez, Directed evolution of enzymes based on in vitro programmable self-replication. BioRxiv (2021). https://doi.org/10.1101/2021.04.22.440993

78. H.Q. Dinh, N. Aubert, N. Noman, T. Fujii, Y. Rondelez, H. Iba, An effective method for evolving reaction networks in synthetic biochemical systems. IEEE Trans. Evol. Comput. **19**(3), 374–386 (2015)

79. N. Aubert, C. Mosca, T. Fujii, M. Hagiya, Y. Rondelez, Computer-assisted design for scaling up systems based on DNA reaction networks. J. R. Soc. Interface **11**(93), 20131167 (2014)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.