



## OPEN ACCESS

## EDITED BY

Quan Zou,  
University of Electronic Science and  
Technology of China, China

## REVIEWED BY

Takashi Nakakuki,  
Kyushu Institute of Technology, Japan  
Junwei Sun,  
Zhengzhou University of Light Industry,  
China

## \*CORRESPONDENCE

Xiaolong Shi,  
✉ xshi@gzhu.edu.cn

## SPECIALTY SECTION

This article was submitted to  
Computational Genomics,  
a section of the journal  
Frontiers in Genetics

RECEIVED 10 December 2022

ACCEPTED 13 February 2023

PUBLISHED 22 February 2023

## CITATION

Chen C, Wen J, Wen Z, Song S and Shi X  
(2023), DNA strand displacement based  
computational systems and  
their applications.  
*Front. Genet.* 14:1120791.  
doi: 10.3389/fgene.2023.1120791

## COPYRIGHT

© 2023 Chen, Wen, Wen, Song and Shi.  
This is an open-access article distributed  
under the terms of the [Creative  
Commons Attribution License \(CC BY\)](#).  
The use, distribution or reproduction in  
other forums is permitted, provided the  
original author(s) and the copyright  
owner(s) are credited and that the original  
publication in this journal is cited, in  
accordance with accepted academic  
practice. No use, distribution or  
reproduction is permitted which does not  
comply with these terms.

# DNA strand displacement based computational systems and their applications

Congzhou Chen<sup>1</sup>, Jinda Wen<sup>2</sup>, Zhibin Wen<sup>2</sup>, Sijie Song<sup>2</sup> and Xiaolong Shi<sup>2\*</sup>

<sup>1</sup>School of Computer Science, Beijing University of Technology, Beijing, China, <sup>2</sup>Institute of Computing Science and Technology, Guangzhou University, Guangzhou, China

DNA computing has become the focus of computing research due to its excellent parallel processing capability, data storage capacity, and low energy consumption characteristics. DNA computational units can be precisely programmed through the sequence specificity and base pair principle. Then, computational units can be cascaded and integrated to form large DNA computing systems. Among them, DNA strand displacement (DSD) is the simplest but most efficient method for constructing DNA computing systems. The inputs and outputs of DSD are signal strands that can be transferred to the next unit. DSD has been used to construct logic gates, integrated circuits, artificial neural networks, etc. This review introduced the recent development of DSD-based computational systems and their applications. Some DSD-related tools and issues are also discussed.

## KEYWORDS

DNA strand displacement, DNA computing, integrated circuits, artificial neural networks, cancer detection

## 1 Introduction

DNA, as a natural material, is biocompatible and programmable. With the development of biotechnology, DNA can be precisely synthesized, controlled, and detected by various tools (Beckwitt et al., 2018; Palluk et al., 2018; Del Grosso et al., 2020). Its sequence can be programmed as a 4-bit of encoding information (ATGC four nucleotides), which is information denser than the 2-bit electronic devices (0 and 1) (Ceze et al., 2019). In this strategy, 1 gram of DNA can store about one EB of data (Sun et al., 2019). Furthermore, DNA, as an information storage carrier, can be kept at  $-25^{\circ}\text{C}$  for 10,000 years. These properties make DNA a perfect information material.

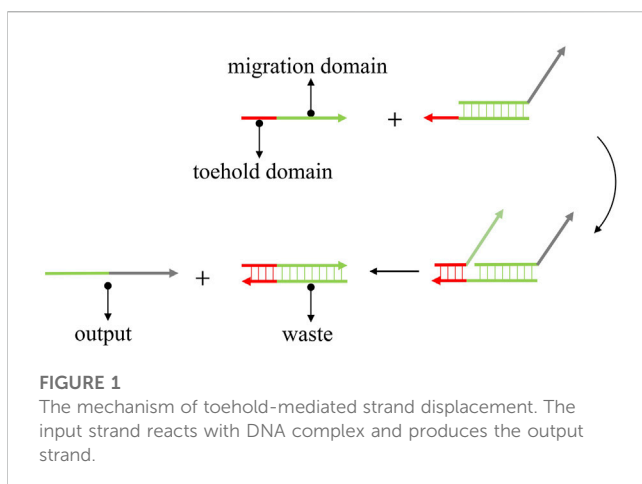
DNA can be used to construct many computational devices. Their sizes are predictable and controllable. The length of one double helix of B-type DNA is 3.4 nm, and the width is 2 nm. A single strand of DNA is flexible, whereas the double strand of DNA is among the stiffest polymers, with a persistence length of 50 nm in 0.1 M aqueous NaCl (Chirkov et al., 2022). Therefore, DNA strands and their complexes can be utilized as computational devices (Chen et al., 2020; Chen et al., 2022a; Xu et al., 2022).

So far, many DNA computing models have been proposed. According to DNA structures, these models can be classified as the single-strand-based computational model (Adleman, 1994; Liu et al., 2000), DNA-tile-based model (Mao et al., 2000; Rothmund et al., 2004), DNA origami-based model (Woods et al., 2019; Amir et al., 2014), and mixture model (Xu, 2016). Among them, the single-strand-based computational model is the simplest and easiest method to construct. Researchers do not have to design

complex or large structures. Single DNA strands are utilized as the inputs and outputs, which can be programmed and cascaded to solve complex problems. The most popular single-strand-based computational method is DNA strand displacement (DSD) reaction.

DSD is the ideal technology for the single-strand-based computational model. DSD was first proposed by Yurke et al. (2000). They constructed a DNA tweezer that can transfer two states *via* DNA strand displacement reactions. DSD involves a short single-strand domain (toehold domain) and the replacement of paired double strand (migration domain). The input strand can react with the DNA complex that is mediated by the toehold domain and produce the output strand. As shown in Figure 1. The red part is the toehold domain; the green part is the migration domain. The input strand consists of a single strand with the toehold and an unpaired migration domain. While the DNA complex contains an unpaired toehold domain and a paired migration domain. The input strand will react with the complex, beginning in the toehold domain. The mechanism of DSD involves the strand's thermodynamic stabilization process. The incompletely paired strands (DNA complex) will be replaced by fully paired strands.

DSD reactions can be implemented to solve computational problems. Many mathematical operations, such as exponential operations (Salehi et al., 2018), multiplication operations (Genot et al., 2013), Boolean operations (Zhao et al., 2021), and satisfiability problems (Liu et al., 2000) have been completed *via* DSD. Additionally, DSD reactions are the approximate representation of arbitrary chemical reactions. As a result, many DSD-based chemical reaction systems have been built, including oscillators, chaotic systems, and feedback digital logic (Soloveichik et al., 2010). DSD reaction is enzyme-free, and it can be used for logical gates, including AND, OR, YES, NOT, NOR, NAND, XOR, Threshold, Inhibited gates, etc. (Okamoto et al., 2004; Seelig et al., 2006; Carell, 2011; Song et al., 2019). The DSD-based gates can be cascaded into integrated circuits to solve complex computing problems. What's more, integrated DSD circuits can function as artificial neural networks (ANN) and perform machine learning (ML) algorithms. The Multilayer Perceptron (MLP) (Arredondo and Lakin, 2022), Support Vector Machine (SVM) (Lopez et al., 2018), Hopfield network (Qian et al., 2011), and Convolutional Neural Network (CNN) (Xiong et al., 2022) had been architected *via* DSD reactions.



DSD can be combined with other technologies and utilized in a wide range of applications. So far, DSD has been combined with CRISPR technology (Ishino et al., 1987; Montagud-Martínez et al., 2021), DNA origami (Rothemund, 2006; Zhang et al., 2022), enzymes (Bucci et al., 2022; Schaffter and Strychalski, 2022), proteins (Fern and Schulman, 2017), etc. These combinations effectively expand the application scenarios for DSD. It has been applied in information storage (Lin et al., 2020; Banal et al., 2021), encryption (Zhu et al., 2022), medical treatment (Peng et al., 2018), biosensing (Li et al., 2021), etc.

In this review, we first introduced the DSD computing systems and their ability to solve computational issues. Focusing on DSD integrated circuits, DSD-based artificial neural networks. Then, we presented the applications of DSD, including molecules and technologies that combine with DSD. Last, we introduced some useful tools for DSD. Defects and problems of DSD were also discussed. Figure 2 shows the abstract of this review.

## 2 DSD computational systems

### 2.1 Arithmetical operations

Fractional coding and Matrix multiplication are two important operations for the computing system. Fractional coding is the foundation of other complex arithmetical operations. It is a numeric format for representing numbers with a decimal part. While matrix multiplication is vital for the implementation of deep networks. Matrix operations are widely used in deep learning because they are an efficient way to represent and manipulate complex mathematical relationships.

Salehi et al. (2018) constructed two types of fractional coding. One is the unipolar fractional coding, the other is the bipolar fractional coding. The definition is illustrated as formulas 1, 2:

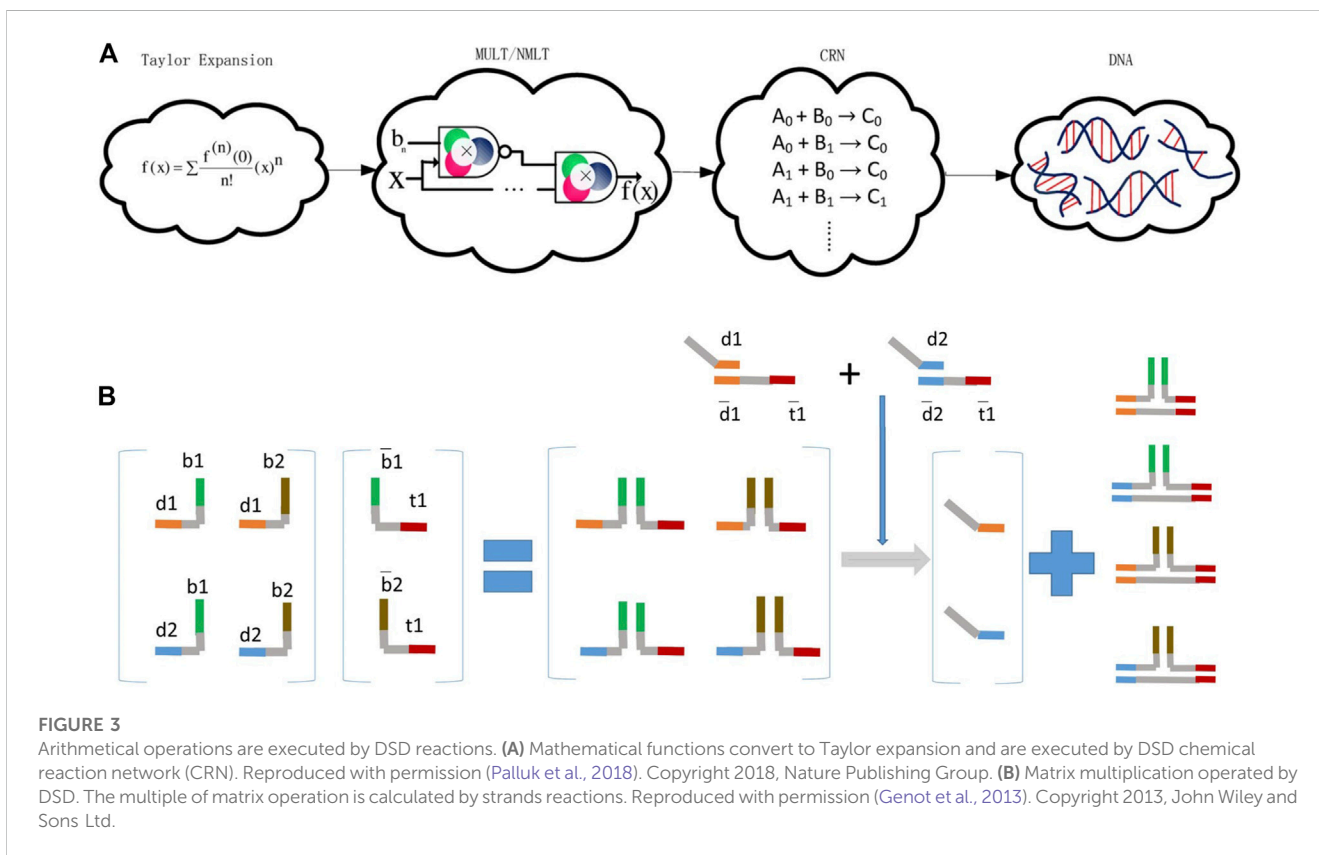
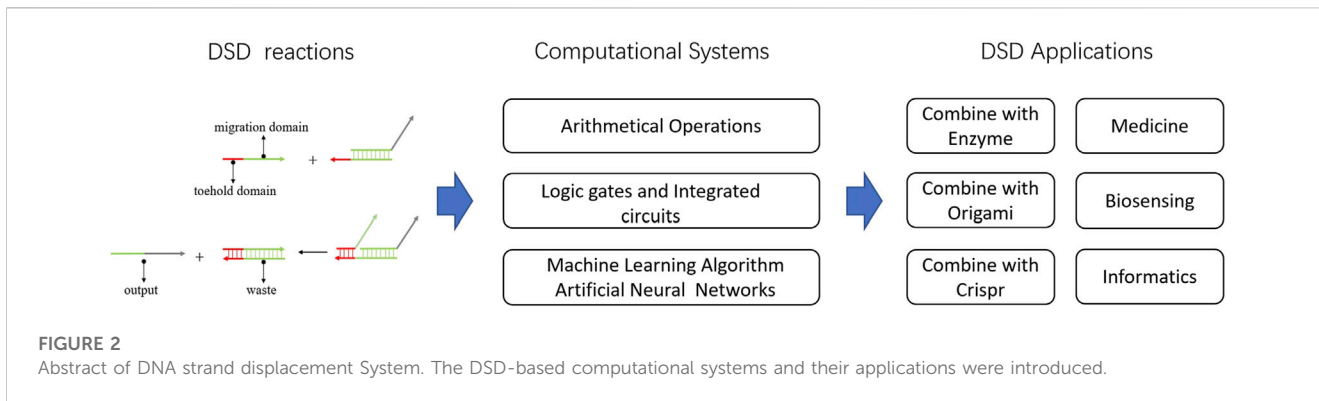
$$x = \frac{[X_1]}{[X_0] + [X_1]} \quad (1)$$

$x$  is the variable. A pair of molecular  $X_0$  and  $X_1$  is assigned to  $x$ . The value of the variable is determined by the ratio of the concentration for the assigned pair. Therefore, the value of  $x$  is confined to unit interval  $[0, 1]$ . This strategy is unipolar fractional coding.

$$x = \frac{[X_1] - [X_0]}{[X_0] + [X_1]} \quad (2)$$

Similarly, the value of the variable  $x$  locates in the range of  $[-1, 1]$ , represented by formula 2. This is the definition of bipolar fractional encoding. Further, they implemented mathematical functions, including exponentials, sigmoid, sine, cosine, and tanh functions (Salehi et al., 2018). They defined the five basic DSD reactions as the multiplication (Mult), Nor-Mult (NMult), multiplexer, bipolar Mult, and bipolar NMult units. Then, mathematical functions can be transformed into Taylor expansions, and constructed by these five units, Figure 3A.

Matrix multiplication can be performed *via* DSD reactions. Reference (Genot et al., 2013) is the first research that explicitly illustrated the implementation of matrix multiplication with DSD.

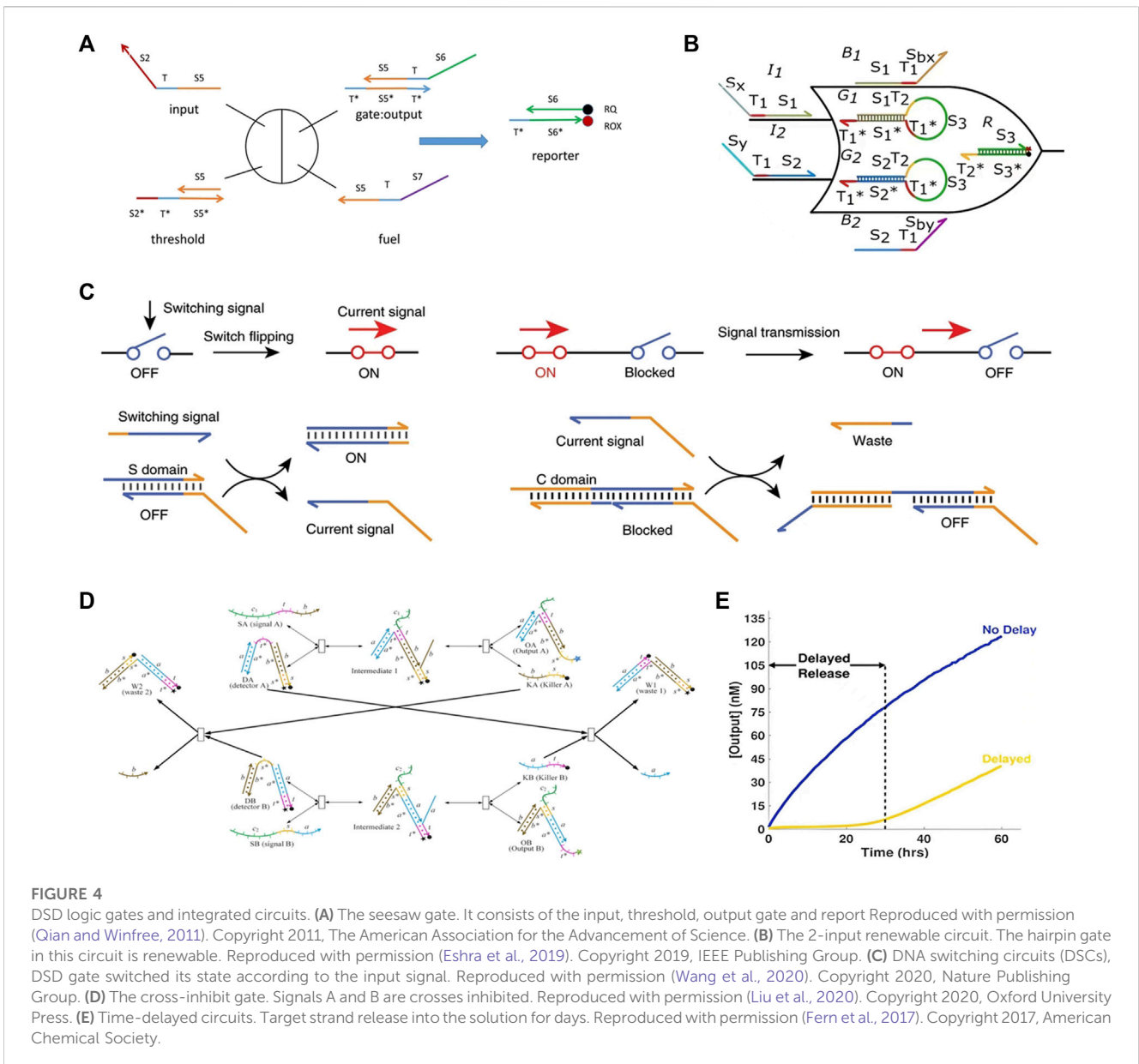


Genot et al. (2013) designed the typical multiplication of a  $2 \times 2$  vector by a  $2 \times 1$  matrix. They utilized the combinations of toehold and migration domains, and implemented the multiplication of a  $2 \times 2$  vector by a  $2 \times 1$  matrix. In this research, toehold and migration domains were dynamically and combinatorially linked to form DNA complexes, which represent the matrices.  $n$  toehold domains and  $n$  substitution domains can form  $n^2$  substitution complexes, which greatly reduce the number of required chains and increase the computational power. The operation processes are shown in Figure 3B, the multiplication

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \times \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \quad (3)$$

The elements in Matrix  $M$  were represented by strands  $\{d_1b_1, d_1b_2, d_2b_1, d_2b_2\}$ , Matrix  $X$  was represented by  $\{b_1t_1, \bar{b}_2t_2\}$ , the cap means its complementary sequences. The output of this matrix multiplication is matrix  $Y$ , represented by two strands  $\{Y_1, Y_2\}$ .

DSD-based chemical reaction networks can be designed in a programmable language for solving mathematical problems. Tang et al. (2021) designed weighted reactions, sum reactions, threshold modules *via* DSD, and solved a three-parameter 0–1 knapsack problem. Lopiccolo et al. (2021) implemented a last-in, first-out stack structure *via* DSD. This stacked structure stores two signals, and signals are released into the solution by order of the input strand. Later, the stack can be rebooted by the activation strand.



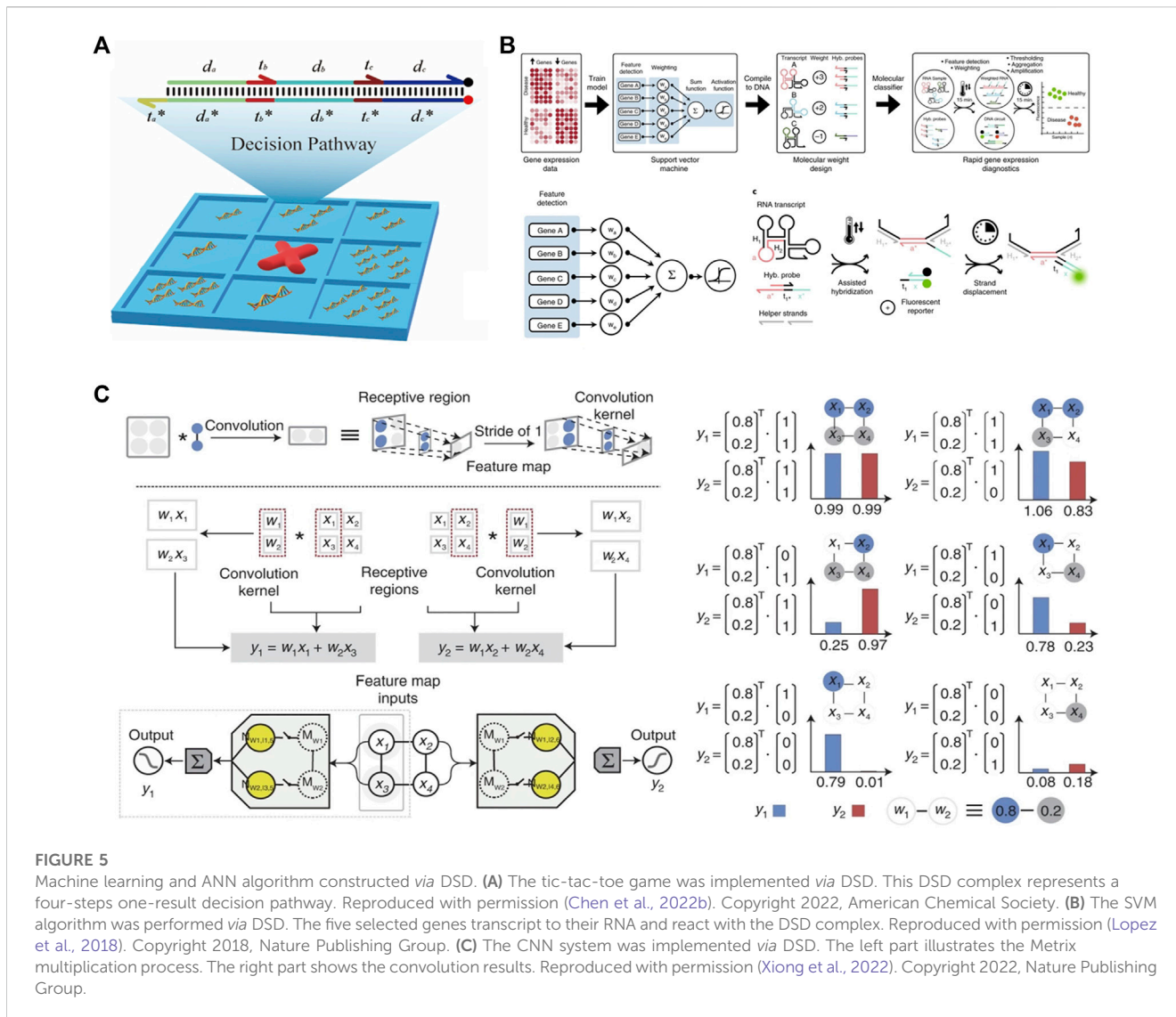
**FIGURE 4** DSD logic gates and integrated circuits. **(A)** The seesaw gate. It consists of the input, threshold, output gate and report. Reproduced with permission (Qian and Winfree, 2011). Copyright 2011, The American Association for the Advancement of Science. **(B)** The 2-input renewable circuit. The hairpin gate in this circuit is renewable. Reproduced with permission (Eshra et al., 2019). Copyright 2019, IEEE Publishing Group. **(C)** DNA switching circuits (DSCs), DSD gate switched its state according to the input signal. Reproduced with permission (Wang et al., 2020). Copyright 2020, Nature Publishing Group. **(D)** The cross-inhibit gate. Signals A and B are crosses inhibited. Reproduced with permission (Liu et al., 2020). Copyright 2020, Oxford University Press. **(E)** Time-delayed circuits. Target strand release into the solution for days. Reproduced with permission (Fern et al., 2017). Copyright 2017, American Chemical Society.

## 2.2 Logic gates and integrated circuits

Logic gates are the primary unit of integrated circuits. They are the foundation of modern computer systems. DSD and modified DSD reactions can be programmed and utilized as logic gates. An outstanding research of improved DSD reaction is the seesaw gate, proposed by Qian and Winfree (2011). The mechanism of the seesaw gate is the reversible DSD reaction. Figure 4A depicts the reaction principle. The input strand will first react with a threshold gate until all the thresholds are consumed. As the threshold chain has a longer toehold domain, which guarantees the first step. Then, excessive input can react with the output gate and produce the output strand. Last, the output strand will react with the report gate and give the fluorescent report signal. Because the fuel strand and input strand have the same toehold and migration domain. Fuel strand can replace the input strand out from the byproduct of second step.

There are many topics involving DSD logic gates, including callability, signal restoration, time-responsive, etc. are one of them. Eshra et al. (2017) modified the seesaw gate motif into a hairpin, called the “DNA hairpin-seesaw gate”. As shown in Figure 4B, the inner two hairpin motifs. The sequences of them from 5’ to 3’ are {S<sub>1</sub>, T<sub>2</sub>, S<sub>3</sub>, T<sub>1</sub>\*, S<sub>1</sub>\*, T<sub>1</sub>\*} and {S<sub>2</sub>, T<sub>2</sub>, S<sub>3</sub>, T<sub>1</sub>\*, S<sub>2</sub>\*, T<sub>1</sub>\*} (cap “\*” represents the complementary sequences). S<sub>1</sub> pairs with S<sub>1</sub>\* to form the hairpin, S<sub>2</sub> pairs with S<sub>2</sub>\* to form the other hairpin. The two toehold domains T<sub>1</sub>\* and T<sub>2</sub> locate at the two sides of these motifs, similar to the original seesaw gate. Further, they added two extractors to initialize the hairpin-seesaw gate, which realized the renewable process. This hairpin gate can be reused more than three times in consecutive calculations. They constructed a 2-input renewable circuit via this motif (Eshra et al., 2019), as shown in Figure 4B. Based on DSD’s AND gate and XOR gate, half-adder or full-adder circuits can be assembled. Xiao et al. (2020) constructed a three bits full-adder. Xie et al. (2022) used three dual-track logic gates and assembled a four bits full-adder.





**FIGURE 5**

Machine learning and ANN algorithm constructed *via* DSD. (A) The tic-tac-toe game was implemented *via* DSD. This DSD complex represents a four-steps one-result decision pathway. Reproduced with permission (Chen et al., 2022b). Copyright 2022, American Chemical Society. (B) The SVM algorithm was performed *via* DSD. The five selected genes transcript to their RNA and react with the DSD complex. Reproduced with permission (Lopez et al., 2018). Copyright 2018, Nature Publishing Group. (C) The CNN system was implemented *via* DSD. The left part illustrates the Matrix multiplication process. The right part shows the convolution results. Reproduced with permission (Xiong et al., 2022). Copyright 2022, Nature Publishing Group.

In 2020, Wang et al. (2020) designed DNA switching circuits (DSCs). The input chain interacts with the gate to generate an output strand. Then the output strand will propagate and arrive at the next gate. Gates will change their states according to the output strand. Therefore, the ON and OFF states are switched, which can represent Yes and No. The output strand propagates like a current passing through the DSCs' gate. Besides, the DSC scheme does not use the dual-track strategy to design NOT gates. (Dual-track strategy refers to a design methodology used in digital circuits that employs two parallel design approaches or paths. The two paths operate in parallel and regularly interact with each other so that both are progressing toward the final goal. For example, the two paths of full-adder are add-path and carry-path. Furthermore, the full-adder is the fundamental component of digital computing.) As a result, its required chains are reduced by 3/4 compared to current DSD circuits. The implementation scheme is shown in Figure 4C.

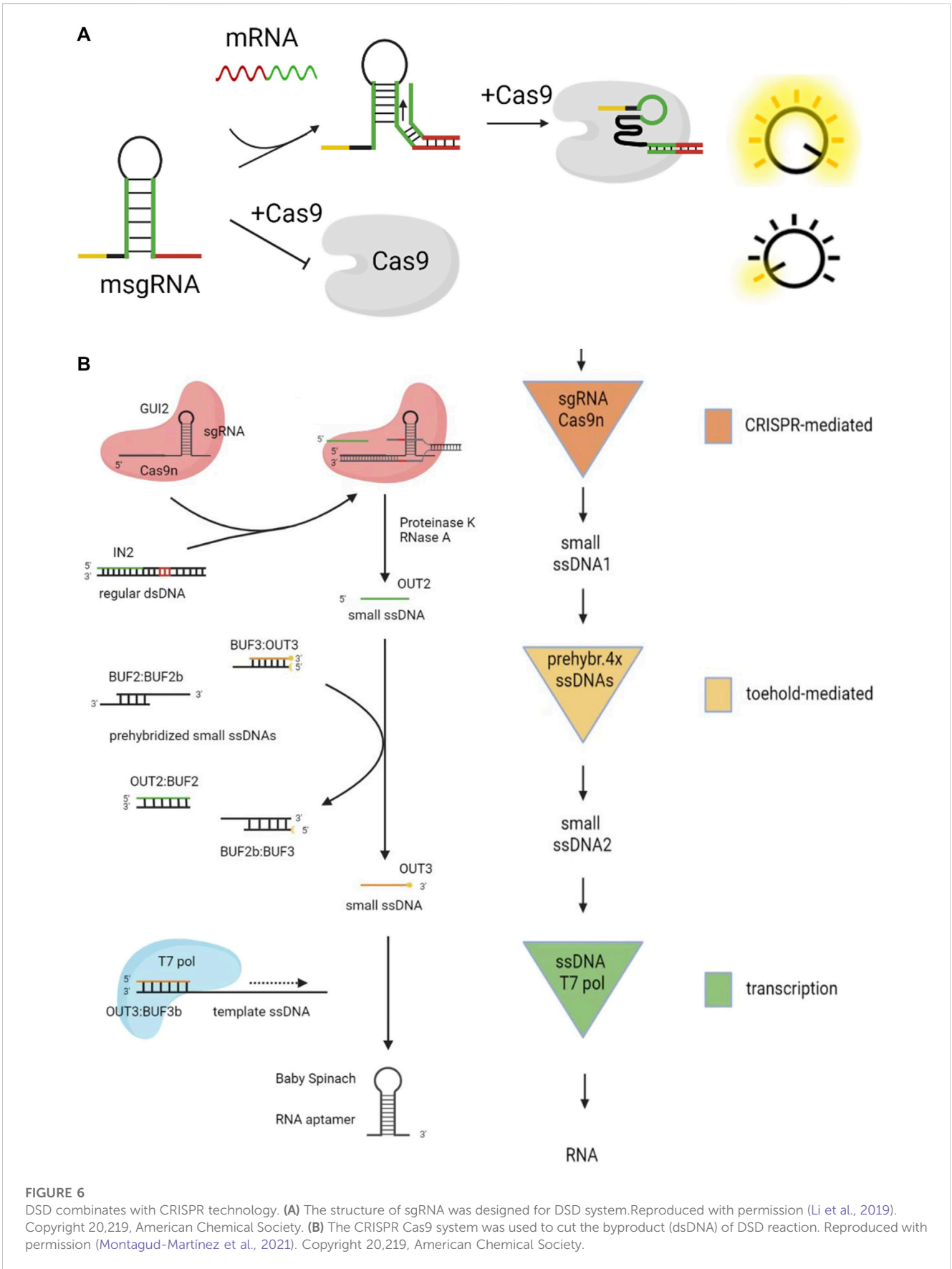
Liu et al. (2020) created a cross-inhibit gate and then used it to perform four-input time-sensitive circuits. Signal strand A can react with detector DA to produce output OA and kill strand KA, as shown in Figure 4D. Afterward, KA can react with detector DB, which inhibits the

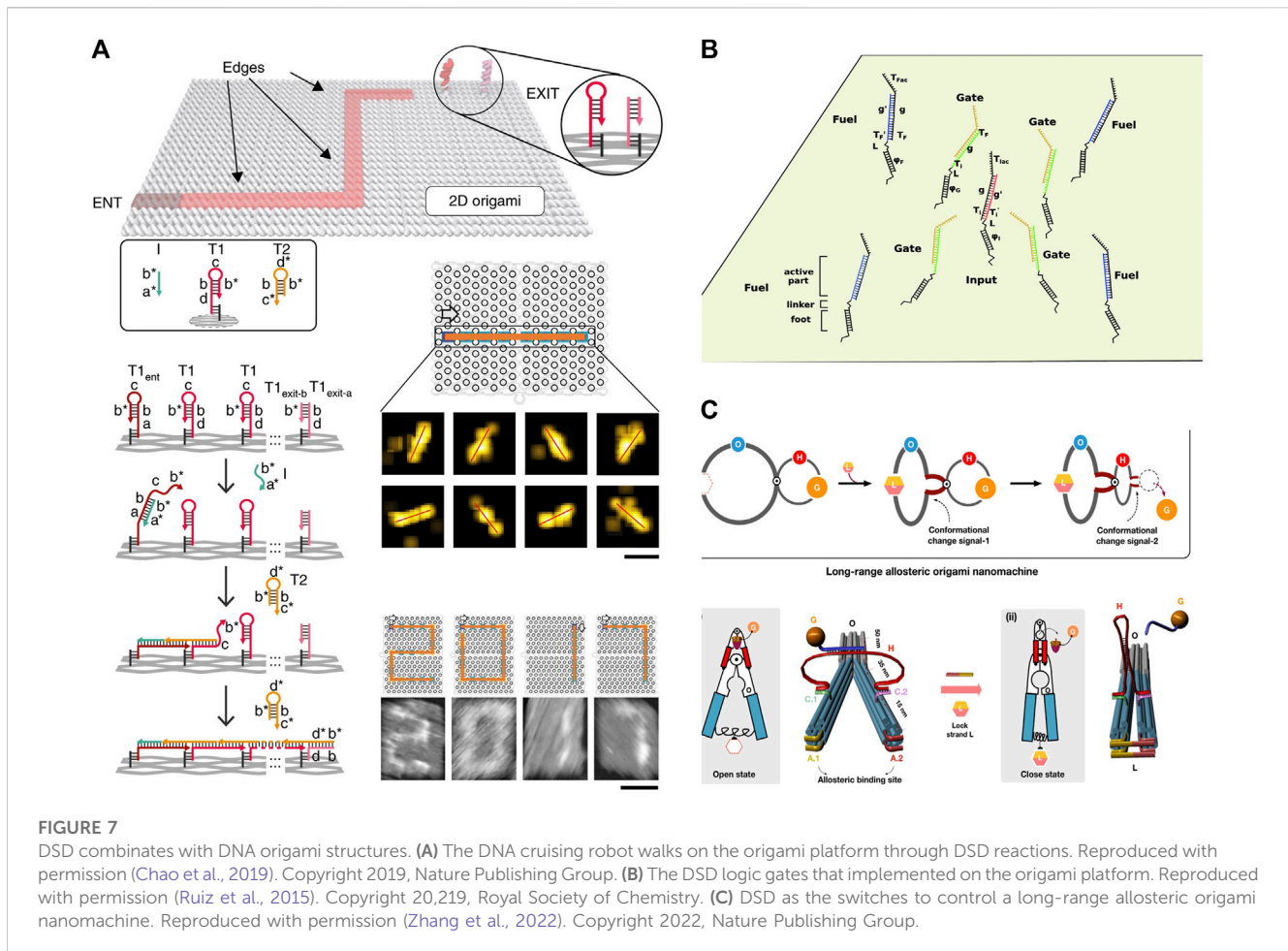
reaction of DB and signal strand B. As a result, signal A inhibited signal B. If signal B is first added, then the situation is the opposite. This DSD-based cross-inhibit strategy is a simple and effective method for constructing time-sensitive circuits. Fern et al. (2017) also designed time-delayed DSD circuits. They created a simple DSD circuit with eight strands that can release target DNA strands into solution at a constant rate for hours to days. The result is illustrated in Figure 4E.

### 2.3 ML and ANN algorithms

Machine learning algorithms and artificial neural networks can be implemented *via* DSD reactions. The decision tree is a classical classification ML algorithm. It processes the classification pathway based on the known probability of occurrence situations. Its classification pathway can be executed *via* DSD reactions. Chen et al. (2022b) designed a domino-like DSD sequential system, which can execute four steps of the decision pathway. As shown in Figure 5A.

Support Vector Machine (SVM) is another powerful ML algorithm. It is a two-class classification algorithm. Its basic





approach is to find the maximum classification interval in the feature space. Lopez et al. (2018) performed a gene classification SVM system. The selected genes for each class are pretrained on a silicon computer. Then, these genes transcript to RNA and reaction with DNA complexes under the guidance of SVM algorithm. The weights of each gene are implemented by transcript times, as shown in Figure 5B. Using seesaw gates and the dual-track strategy, Qian et al. (2011) further constructed a Hopfield network. This network contains twenty-four circuits and “remembers” four patterns.

In 2018, Qian et al. extended the computational border of the seesaw gate. They constructed a winner-take-all neural network without using the dual-track strategy (Cherry and Qian, 2018). This network is a three-layer, fully connected artificial network. The inputs are 100 bits DNA strands in  $10 \times 10$  patterns. The hidden layer consists of 20 distinct DNA molecules that can react with these 100 bits DNA strands *via* DSD reactions. The output layer shows the classification results of handwritten digits ‘1’ to ‘9’. This is the first research attempt using DSD to construct complex ANN.

Convolutional Neural Network (CNN) can also be implemented *via* DSD reactions. The essence of the convolution operation is matrix multiplication. Xiong et al. (2022) take the same matrix multiplication strategy as Genot et al. (2013). They designed the Matrix operation DSD system, including the multiplication of two matrices, matrix Addition, and matrix Subtraction.

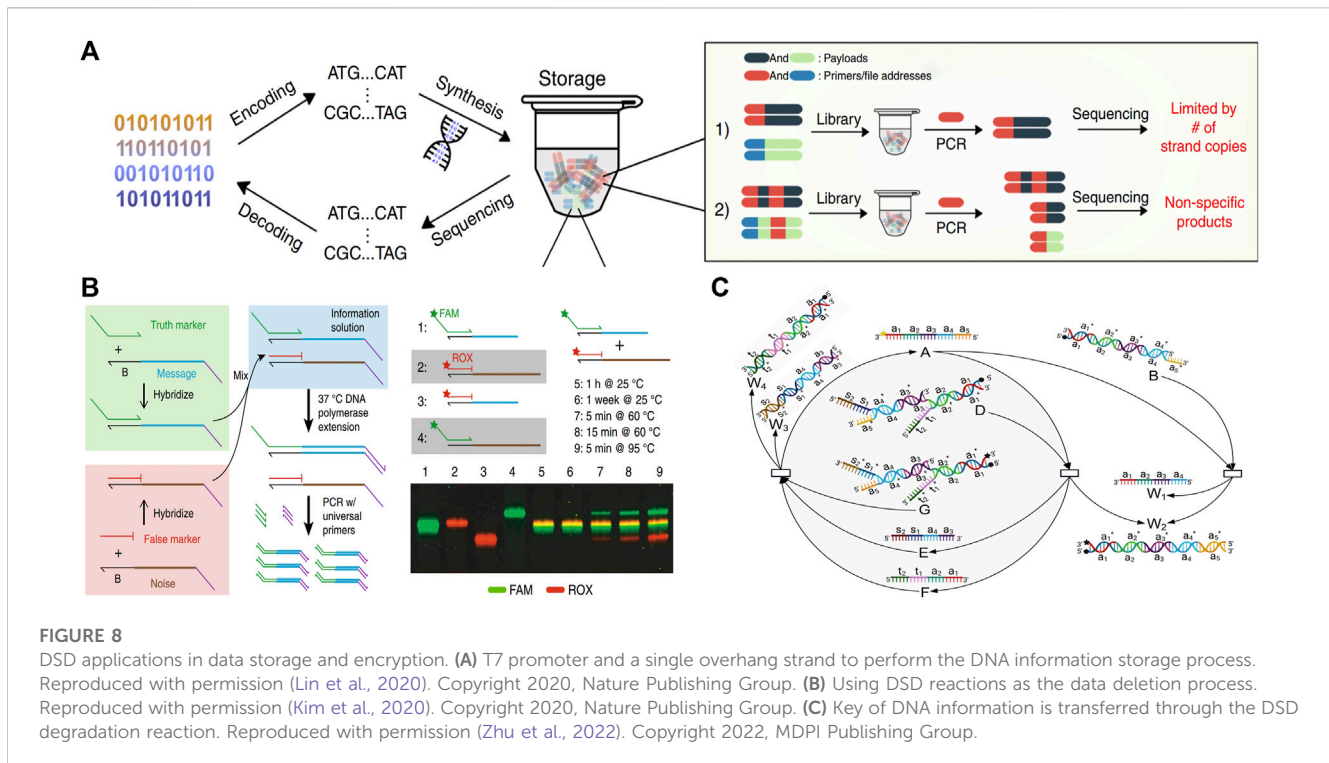
The matrix multiplication is  $|X_{2 \times 2}| \times |W_{2 \times 1}|$ , results are  $y_1$  and  $y_2$ . Then,  $y_1$  and  $y_2$  are added or subtracted to produce the final results  $Y$  which is the result of this convolution process. They implemented this kernel  $|W_{2 \times 1}|$  to convolute the input ‘image’ (image is transferred to 144 bits DNA strands in  $12 \times 12$  patterns). Then the image indentation result can be illustrated by  $\sum_1^n Y$  ( $n$  is the convolutional times). As shown in Figure 5C. Using this DSD-based CNN method, they successfully identified oracle bones as well as English letters and Arabic numerals.

### 3 Applications of DSD computing

DNA is biocompatible, and DNA structures can be endocytosed by cells. Furthermore, DNA can be modified and linked with drugs, proteins, and other molecules. DSD computational systems combined with molecules and biotechnologies have applications in medicine, biosensors, informatics, and other fields.

#### 3.1 DSD combines with CRISPR technology

CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats) was first found from *Escherichia coli* bacteria by Ishino et al. (1987). The CRISPR system consists of an artificially designed



sgRNA (single-guide RNA) and a Cas protein, which combine to form a complex. sgRNA has a guide sequence of about 20 nt that matches the target gene, then the PAM sequence in the upstream can be cleaved, repressed, and activate the target gene with the help of Cas protein. As sgRNA is a segment of RNA around 100 nt, it is possible to program it.

The combination of CRISPR and DSD enables some in-cell logical circuits, even intracellular gene regulation can be realized. Jin et al. (2019) added the toehold domain for sgRNA at the 5' end (guide sequence) and 3' end (scaffold structure) without affecting its activity. Because the guide sequence of sgRNA is sensitive to its formation, the binding and unbinding of the toehold domain can be utilized for controlling its activity. Then, they used this designed CRISPR as the switch to control a DSD system.

Li et al. (2019) designed the scaffold structure of sgRNA, as shown in Figure 6A. Regular sgRNA consists of a guide sequence (the yellow part) and a scaffold sequence (the black part). They added the mRNA sensing sequence (the green part) and the toehold domain (the red part) in the sgRNA, which is named msgRNA. This structure was designed to disrupt the scaffold of sgRNA, making it impossible to bind with the Cas9 protein. When mRNA is added, the strand displacement reaction opens the hairpin of msgRNA, restoring sgRNA activity. Hao et al. disrupted the structure of sgRNA through a blocking strand. Then they added a replacement strand to react with the blocking strand, thus restoring the activity of sgRNA (Hao et al., 2020).

The fully paired double-stranded DNA (dsDNA) is generally considered the waste product of the DSD reaction. However, Roser et al. made the waste strands useful through CRISPR technology (Montagud-Martínez et al., 2021), as shown in Figure 6B. The regular dsDNA is cut by CRISPR Cas9 and produces output2. Then output2 will react with prehybridized ssDNAs through DSD reaction and produce output3. Last, output3 can regulate

the target gene with the help of T7 pol. The whole system includes CRISPR-mediated reactions, toehold-mediated reactions, and the transcript process.

From the above-mentioned designs, we can conclude that DSD reactions can be used as the switch to control the CRISPR process. These above-mentioned studies have a common strategy, which is to program the DSD reaction domain in sgRNA sequences. This strategy is direct and efficient.

## 3.2 DSD combines with molecules

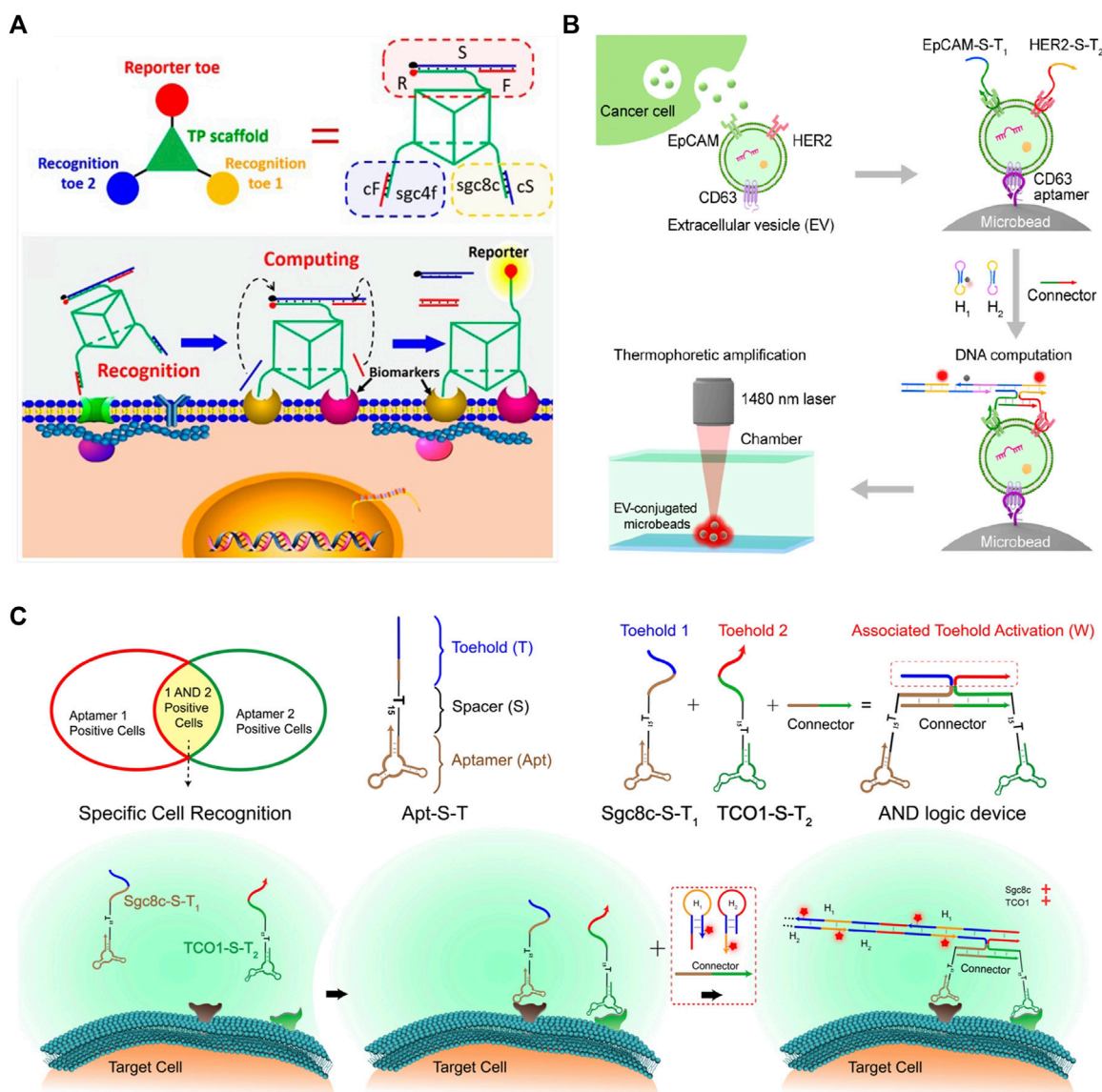
Based on DSD reactions, DNA structures can be designed as molecular robots, molecular machines, and many other molecular devices. These devices can be precisely controlled *via* the DSD reaction. DSD combines with various molecules can make these devices efficient for many applications.

### 3.2.1 DSD combines with origami

DNA origami is a powerful tool for designing arbitrary shapes of DNA structures (Rothenmund, 2006). It involves a long scaffold strand and hundreds of short strands that help to bind to pre-designed structures. Every location of the origami structure can be programmed through the binding strands. Therefore, DNA origami can be utilized as the platform for DSD reactions.

Lund et al. (2010) designed a molecular robot; called the molecular spider. The three legs of this spider are specific DNA enzyme strands that can cruise on a DNA origami *via* DSD reactions. Chao et al. (2019) designed a DNA cruising robot that enables molecular reactions along the designed paths on a DNA origami platform. The robot can find the paths of the DNA origami maze by the controlled DSD reactions. They observed all the walking paths of





**FIGURE 9** DSD applications in Medicine and Biosensing. **(A)** The 3D DNA nanomachine is attached on the cancer cell surface and detects SELEX cancer cells. Reproduced with permission (Peng et al., 2018). Copyright 2018, American Chemical Society. **(B)** The DNA computational device that can identify tEVs through DSD reactions. Reproduced with permission (Li et al., 2021). Copyright 2021, American Chemical Society. **(C)** The DSD AND gate are attached to the cancer cells. Reproduced with permission (Chang et al., 2019). Copyright 2019, American Chemical Society.

the DNA cruising robot in the maze using AFM and DNA-PAINT imaging characterization. This DSD and origami combination method provides a strategy for single-molecule diagnosis and treatment. As shown in Figure 7A. Ruiz et al. (2015) tethered DSD system on a DNA origami platform, making the reactions faster than reactions in solution. Besides, reactions between gates are limited among their neighbors. They then devised many logic gates through this method. As shown in Figure 7B.

Zhang et al. (2022) used the DSD reactions as switches to control an origami nanomachine. They designed a long-range allosteric origami nanomachine. DNA complexes were programmed at the binding sites. Then the allosteric nanomachine can open or close by the input strands via DSD reactions. As shown in Figure 7C.

DSD reaction is hard to visualize. The common method uses the fluorescent signal as the reporter. However, the above-mentioned devices made these processes visual. The DSD switch on/off status can be represented by the shapes of origami. The track paths of the cruising robot and the spider on the DNA origami substrate can be visualized via AFM. When DSD combines with origami, these devices become interesting and credible.

### 3.2.2 DSD combines with enzymes

Enzymes in DSD computational systems can be used as regulators to control their reactions. Bucci et al. (2022) designed a blocker strand that can bind to the toehold domain, which prevented the DSD reactions. Then they took two steps to recover the activity of DSD.

First, they used the RNase H enzyme to cut the block strand, thus exposing the toehold domain. Second, they used formamidopyrimidine DNA glycosylase or uracil-DNA glycosylase to degrade the block strand. The degradation rates of these two glycosylases are different. Therefore, the reaction rate of the DSD system can be controlled.

With the help of transcriptase, DSD system can be deployed in cells. Schaffter et al. (Schaffter and Strychalski, 2022) designed a DNA strand with a specific sequence that can be transcribed to RNA complexes in cells. This specific DNA strand is endocytosed by cells, and then transcribed in cells. Therefore, the entire RNA circuit was implemented in cells. Fern et al. (Fern and Schulman, 2017) designed a hairpin structure at the 5' and 3' end of the DSD complexes, which can prevent the binding enzymes from being disrupted by other proteins in serums.

Enzymes can regulate the expression or reaction of the DSD system. With the help of enzymes, a sophisticated regulatory network with multiple coordinated functions can be realized. However, the redundant enzyme could complicate the reaction environment, which might cause instability in the system.

### 3.3 DSD application in informatics

The ATGC four nucleotides can be used as 4-bit coding information, and synthetic DNA strands have been utilized as data storage material. Generally, a strip of DNA data consists of an addressing domain, an information domain, and a correction domain. DSD reactions can be deployed on these domains. The input strand of the DSD can be utilized as the information encryption, information reader, initiator, etc. The DSD-based information processing is simple and effective.

Lin et al. (2020) inserted a T7 promoter and a single overhang strand into the information domain. The overhanging strand can accomplish the tagging, locking, replacement, and deletion processes for the DNA information with the help of the T7 promoter. Additionally, the DNA data can be read out through the PCR process without disrupting the original DNA strand. Lin increases the information capacity and reduces coding complexity by using the DSD system, which avoids the impact of primers on DNA coding regions. As illustrated in Figure 8A.

Kim et al. (2020) used the DSD reactions to extract image data that was stored in DNA strands. They encoded a false message-encoded strand that has the same primers and length as the true message-encoded strand. Correct information can be extracted from the true strands through DSD reactions and the PCR process, while the false strand cannot. However, if the whole system is heated to above 95° and kept for 5 min, the correct information and the error information will be mixed together, resulting in decoding failure, and the related data cannot be recovered, which plays the role of fast erasure of the target information. As shown in Figure 8B.

Banal et al. (2021) sealed DNA information in impervious silica capsules. The surface of silica gel is labeled with single overhanging DNA strands, which represent the characteristics of the stored information. When information retrieval is required, a complementary strand with magnetic bead-modified DNA strands will be added to the solution. These two strands will react through the DSD reaction, and the silica capsule data will be captured by magnetic adsorption.

Zhu et al. (2022) used the DSD reactions as the encryption approach. This encryption approach includes the conversion of information to DNA sequences through Huffman coding. Then, the key to this DNA information is transferred through a degradation reaction. Finally, the DSD transferred key is extended *via* a catalysis reaction, which increases the decryption complexity. The key transfer reactions are illustrated in Figure 8C.

### 3.4 DSD applications in Medicine and Biosensing

DSD can be designed as a tool for cancer detection. DNA strands modified with aptamers can target the cancer cell surface, and the detection signals or drugs can be released through the DSD reaction. In these applications, the ligands target specific cells, and the DSD system is utilized as the switch to turn on the drug release process or cancer detection process.

Peng et al. (2018) constructed a three-dimensional DNA-based nanomachine that can target cancer cells. This nanomachine can identify the DNA aptamers that are produced by SELEX cancer cells. When the nanomachine binds to the surface of cancer cells, it will produce the fluorescent signal through the DSD reaction. As shown in Figure 9A. This DSD-based cancer detection method is ultra-sensitive to cancer cells and can be used as a tool for early cancer identification.

Molecular profiling of tumor-derived extracellular vesicles (tEVs) is a vital cancer biomarker. Li et al. (2021) used the thermophoresis mediated DNA computing device to identify tEVs. They constructed an AND gate that consists of EpCAM-S-T2 and HER2-S-T2 proteins. tEVs causes the overexpression of two proteins. The whole tEVs are binding to a Microbead *via* CD63 aptamer. Last, they added the DNA hairpin strands into the solution, and then the AND gate produced the fluorescent signals, which can represent the expression level of tEVs.

Chang et al. (2019) identified and narrowed down a cancer cell-type subpopulation from large populations of similar cells through the DSD logic device. They programmed two cancer-expression proteins, aptamers Sgc8c-S-T1 and TCO1-S-T2, on an AND logic gate. These two aptamers can identify the target cancer cell and attach themselves to its surface. Then, input strands can react with the AND gate and produce fluorescent signals. The scheme of this method is illustrated in Figure 9B. These two methods (Li et al., 2021)- (Chang et al., 2019) have a similarity. They both used two aptamers to identify cancer and then assembled an AND gate. The final signal reports were accomplished *via* DSD AND gate reactions.

## 4 Tools of DSD

Visual DSD (Lakin et al., 2011) is computer simulation software created for DSD researchers. The authors designed specific syntax conventions for DSD language, which can program DSD complexes or single DNA strands in various situations. The DSD syntax can be programmed and constructed to form complex chemical reaction networks (CRN). Besides, the initial condition, CRN conditions, and other parameters can be set at will. Additionally, there are many thermodynamical algorithms provided for different experiments,

such as deterministic simulation, stochastic simulation, spatial simulation, etc. Visual DSD generates all possible reactions and products automatically. It helps researchers construct complex DNA reaction networks without manual design, and the results can be visualized.

Leakages could be occurred among with DSD reactions. If partially paired sequences exist between two DNA complexes, they could react with each other without the help of toehold mediation. Visual DSD does not take the leakage into consideration. Zarubiieva et al. (2022) proposed a leak analysis method for Visual DSD. This method, named DSD leaks, consists of a leak reaction enumeration algorithm and leak probability prediction. They extended the logic programming functionality of Visual DSD.

Badelt et al. (2020) proposed a domain-level DSD reaction simulation software; they named it “peppercorn”. Peppercorn is more general than Visual DSD. It considers the natural connection to nucleic acid biophysics, and it is still suitable for structure analysis. The authors implemented three different algorithms for different situations. They presented an enumeration algorithm for the DSD reaction network. The condensation algorithm for CRN uses slow reactions. The approximate rate model for DNA domain level systems. Last, they performed multiple case studies that compared their model with real experiments. Peppercorn analysis examines the DSD reactions at the domain level, which can be rigorously analyzed without knowing the specific type of nucleic acid or polymer.

## 5 Summary

DNA computing is a promising technology that combines DNA nanotechnology and computer science. It exploits the massively parallel nature of molecules. DSD is the simplest DNA computing strategy and involves the inputs and outputs of signal DNA strands. What’s more, DSD can be programmed and cascade into complex chemical reaction networks. DSD integrated circuits have been implemented to realize machine learning and artificial neural networks. With the help of molecules and biotechnologies, the

DSD computational system can be used in various applications. DSD with CRISPR technology has been applied to construct intracellular circuits and biosensing. DSD had been demonstrated its ability to detect cancer cells. DSD is also useful in data storage and encryption. Although DSD is already widely applied, it has a lot of development potential. DSD is a simple tool, and its development depends on the cross-fertilization of other biotechnologies.

## Author contributions

This review was conceived by CC and XS. All authors wrote this manuscript and approve of its publication.

## Funding

This work was supported by the National Key R&D Program of China (grant 2019YFA0706402), and the National Natural Science Foundation of China under grants 62272009, 62172302, 61772376, and 62072129.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

- Aleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *science* 266 (5187), 1021–1024. doi:10.1126/science.7973651
- Amir, Y., Ben-Ishay, E., Levner, D., Ittah, S., Abu-Horowitz, A., and Bachelet, I. (2014). Universal computing by DNA origami robots in a living animal. *Nat. Nanotechnol.* 9 (5), 353–357. doi:10.1038/nnano.2014.58
- Arredondo, D., and Lakin, M. R. (2022). Supervised learning in a multilayer, nonlinear chemical neural network. *IEEE Trans. Neural Netw. Learn. Syst.* 8, PP. doi:10.1109/TNNLS.2022.3146057
- Badelt, S., Grun, C., Sarma, K. V., Wolfe, B., Shin, S. W., and Winfree, E. (2020). A domain-level DNA strand displacement reaction enumerator allowing arbitrary non-pseudoknotted secondary structures. *J. R. Soc. Interface* 17 (167), 20190866. doi:10.1098/rsif.2019.0866
- Banal, J. L., Shepherd, T. R., Berleant, J., Huang, H., Reyes, M., Ackerman, C. M., et al. (2021). Random access DNA memory using Boolean search in an archival file storage system. *Nat. Mater.* 20 (9), 1272–1280. doi:10.1038/s41563-021-01021-3
- Beckwitt, E. C., Kong, M., and Van Houten, B. (2018). Studying protein-DNA interactions using atomic force microscopy. *Seminars Cell & Dev. Biol.* 73, 220–230. doi:10.1016/j.semcdb.2017.06.028
- Bucci, J., Irmisch, P., Del Grosso, E., Seidel, R., and Ricci, F. (2022). Orthogonal enzyme-driven timers for DNA strand displacement reactions. *J. Am. Chem. Soc.* 144, 19791–19798. doi:10.1021/jacs.2c06599
- Carell, T. (2011). Molecular computing: DNA as a logic operator. *Nature* 469 (7328), 45–46. doi:10.1038/469045a
- Ceze, L., Nivala, J., and Strauss, K. (2019). Molecular digital data storage using DNA. *Nat. Rev. Genet.* 20 (8), 456–466. doi:10.1038/s41576-019-0125-3
- Chang, X., Zhang, C., Lv, C., Sun, Y., Zhang, M., Zhao, Y., et al. (2019). Construction of a multiple-aptamer-based DNA logic device on live cell membranes via associative toehold activation for accurate cancer cell identification. *J. Am. Chem. Soc.* 141 (32), 12738–12743. doi:10.1021/jacs.9b05470
- Chao, J., Wang, J., Wang, F., Ouyang, X., Kopperger, E., Liu, H., et al. (2019). Solving mazes with single-molecule DNA navigators. *Nat. Mater.* 18 (3), 273–279. doi:10.1038/s41563-018-0205-3
- Chen, C., Xu, J., Ruan, L., Zhao, H., Li, X., and Shi, X. (2022). DNA origami frame filled with two types of single-stranded tiles. *Nanoscale* 14, 5340–5346. doi:10.1039/d1nr05583f
- Chen, C., Xu, J., and Shi, X. (2020). Multiform DNA origami arrays using minimal logic control. *Nanoscale* 12, 15066–15071. doi:10.1039/d0nr00783h
- Chen, X., Liu, X., Wang, F., Li, S., Chen, C., Qiang, X., et al. (2022). Massively parallel DNA computing based on domino DNA strand displacement logic gates. *ACS Synth. Biol.* 11, 2504–2512. doi:10.1021/acssynbio.2c00270



- Cherry, K. M., and Qian, L. (2018). Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature* 559 (7714), 370–376. doi:10.1038/s41586-018-0289-6
- Chirkov, N. S., Lin, S. Y., Michailov, A. V., Miller, R., and Noskov, B. A. (2022). DNA penetration into a lysozyme layer at the surface of aqueous solutions. *Int. J. Mol. Sci.* 23 (20), 12377. doi:10.3390/ijms232012377
- Del Grosso, E., Prins, L. J., and Ricci, F. (2020). Transient DNA-based nanostructures controlled by redox inputs. *Angew. Chem. Int. Ed.* 59 (32), 13238–13245. doi:10.1002/anie.202002180
- Eshra, A., Shah, S., and Reif, J. (2017). “DNA hairpin gate: A renewable DNA seesaw motif using hairpins,”. arXiv preprint arXiv:1704.06371.
- Eshra, A., Shah, S., Song, T., and Reif, J. (2019). Renewable DNA hairpin-based logic circuits. *IEEE Trans. Nanotechnol.* 18, 252–259. doi:10.1109/tnano.2019.2896189
- Fern, J., Scalise, D., Cangialosi, A., Howie, D., Potters, L., and Schulman, R. (2017). DNA strand-displacement timer circuits. *ACS Synth. Biol.* 6 (2), 190–193. doi:10.1021/acssynbio.6b00170
- Fern, J., and Schulman, R. (2017). Design and characterization of DNA strand-displacement circuits in serum-supplemented cell medium. *ACS Synth. Biol.* 6 (9), 1774–1783. doi:10.1021/acssynbio.7b00105
- Genot, A. J., Bath, J., and Turberfield, A. J. (2013). Combinatorial displacement of DNA strands: Application to matrix multiplication and weighted sums. *Angew. Chem. Int. Ed.* 52 (4), 1189–1192. doi:10.1002/anie.201206201
- Hao, Y., Li, J., Li, Q., Zhang, L., Shi, J., Zhang, X., et al. (2020). Programmable live-cell CRISPR imaging with toehold-switch-mediated strand displacement. *Angew. Chem. Int. Ed.* 59 (46), 20612–20618. doi:10.1002/anie.202009062
- Ishino, Y., Shinagawa, H., Makino, K., Amemura, M., and Nakata, A. (1987). Nucleotide sequence of the *iap* gene, responsible for alkaline phosphatase isozyme conversion in *Escherichia coli*, and identification of the gene product. *J. Bacteriol.* 169 (12), 5429–5433. doi:10.1128/jb.169.12.5429-5433.1987
- Jin, M., Garreau de Loubresse, N., Kim, Y., Kim, J., and Yin, P. (2019). Programmable CRISPR-cas repression, activation, and computation with sequence-independent targets and triggers. *ACS Synth. Biol.* 8 (7), 1583–1589. doi:10.1021/acssynbio.9b00141
- Kim, J., Bae, J. H., Baym, M., and Zhang, D. Y. (2020). Metastable hybridization-based DNA information storage to allow rapid and permanent erasure. *Nat. Commun.* 11, 5008. doi:10.1038/s41467-020-18842-6
- Lakin, M. R., Youssef, S., Polo, F., Emmott, S., and Phillips, A. (2011). Visual DSD: A design and analysis tool for DNA strand displacement systems. *Bioinformatics* 27 (22), 3211–3213. doi:10.1093/bioinformatics/btr543
- Li, Y., Deng, J., Han, Z., Liu, C., Tian, F., Xu, R., et al. (2021). Molecular identification of tumor-derived extracellular vesicles using thermophoresis-mediated DNA computation. *J. Am. Chem. Soc.* 143 (3), 1290–1295. doi:10.1021/jacs.0c12016
- Li, Y., Teng, X., Zhang, K., Deng, R., and Li, J. (2019). RNA strand displacement responsive CRISPR/Cas9 system for mRNA sensing. *Anal. Chem.* 91 (6), 3989–3996. doi:10.1021/acs.analchem.8b05238
- Lin, K. N., Volkel, K., Tuck, J. M., and Keung, A. J. (2020). Dynamic and scalable DNA-based information storage. *Nat. Commun.* 11, 2981. doi:10.1038/s41467-020-16797-2
- Liu, C., Liu, Y., Zhu, E., Zhang, Q., Wei, X., and Wang, B. (2020). Cross-inhibitor: A time-sensitive molecular circuit based on DNA strand displacement. *Nucleic acids Res.* 48 (19), 10691–10701. doi:10.1093/nar/gkaa835
- Liu, Q., Wang, L., Frutos, A. G., Condon, A. E., Corn, R. M., and Smith, L. M. (2000). DNA computing on surfaces. *Nature* 403, 175–179. doi:10.1038/35003155
- Lopez, R., Wang, R., and Seelig, G. (2018). A molecular multi-gene classifier for disease diagnostics. *Nat. Chem.* 10 (7), 746–754. doi:10.1038/s41557-018-0056-1
- Lopiccolo, A., Shirt-Ediss, B., Torelli, E., Castronovo, M., Fellermann, H., Krasnogor, N., et al. (2021). A last-in first-out stack data structure implemented in DNA. *Nat. Commun.* 12, 4861. doi:10.1038/s41467-021-25023-6
- Lund, K., Manzo, A. J., Dabby, N., Michelotti, N., Johnson-Buck, A., Nangreave, J., et al. (2010). Molecular robots guided by prescriptive landscapes. *Nature* 465 (7295), 206–210. doi:10.1038/nature09012
- Mao, C., LaBean, T. H., Reif, J. H., and Seeman, N. C. (2000). Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature* 407 (6803), 493–496. doi:10.1038/35035038
- Montagud-Martínez, R., Heras-Hernández, M., Goiriz, L., Daros, J. A., and Rodrigo, G. (2021). CRISPR-mediated strand displacement logic circuits with toehold-free DNA. *ACS Synth. Biol.* 10 (5), 950–956. doi:10.1021/acssynbio.0c00649
- Okamoto, A., Tanaka, K., and Saito, I. (2004). DNA logic gates. *J. Am. Chem. Soc.* 126 (30), 9458–9463. doi:10.1021/ja047628k
- Palluk, S., Arlow, D. H., De Rond, T., Barthel, S., Kang, J. S., Bector, R., et al. (2018). De novo DNA synthesis using polymerase-nucleotide conjugates. *Nat. Biotechnol.* 36 (7), 645–650. doi:10.1038/nbt.4173
- Peng, R., Zheng, X., Lyu, Y., Xu, L., Zhang, X., Ke, G., et al. (2018). Engineering a 3D DNA-logic gate nanomachine for bispecific recognition and computing on target cell surfaces. *J. Am. Chem. Soc.* 140 (31), 9793–9796. doi:10.1021/jacs.8b04319
- Qian, L., Winfree, E., and Bruck, J. (2011). Neural network computation with DNA strand displacement cascades. *nature* 475 (7356), 368–372. doi:10.1038/nature10262
- Qian, L., and Winfree, E. (2011). Scaling up digital circuit computation with DNA strand displacement cascades. *science* 332 (6034), 1196–1201. doi:10.1126/science.1200520
- Rothemund, P. W. K., Papadakis, N., and Winfree, E. (2004). Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biol.* 2 (12), e424. doi:10.1371/journal.pbio.0020424
- Rothemund, P. W. (2006). Folding DNA to create nanoscale shapes and patterns. *Nature* 440, 297–302. doi:10.1038/nature04586
- Ruiz, I. M., Arbona, J. M., Lad, A., Mendoza, O., Aime, J. P., and Elezgaray, J. (2015). Connecting localized DNA strand displacement reactions. *Nanoscale* 7 (30), 12970–12978. doi:10.1039/c5nr02434j
- Salehi, S. A., Liu, X., Riedel, M. D., and Parhi, K. K. (2018). Computing mathematical functions using DNA via fractional coding. *Sci. Rep.* 8, 8312. doi:10.1038/s41598-018-26709-6
- Schaffter, S. W., and Strychalski, E. A. (2022). Cotranscriptionally encoded RNA strand displacement circuits. *Sci. Adv.* 8 (12), eabl4354. doi:10.1126/sciadv.abl4354
- Seelig, G., Soloveichik, D., Zhang, D. Y., and Winfree, E. (2006). Enzyme-free nucleic acid logic circuits. *science* 314 (5805), 1585–1588. doi:10.1126/science.1132493
- Soloveichik, D., Seelig, G., and Winfree, E. (2010). DNA as a universal substrate for chemical kinetics. *Proc. Natl. Acad. Sci.* 107 (12), 5393–5398. doi:10.1073/pnas.0909380107
- Song, T., Eshra, A., Shah, S., Bui, H., Fu, D., Yang, M., et al. (2019). Fast and compact DNA logic circuits based on single-stranded gates using strand-displacing polymerase. *Nat. Nanotechnol.* 14 (11), 1075–1081. doi:10.1038/s41565-019-0544-5
- Sun, L., He, J., Luo, J., and Coy, D. H. (2019). DNA and the digital data storage [J]. *Health Sci. J.* 13 (3), 1–7.
- Tang, Z., Yin, Z., Wang, L., Cui, J., Yang, J., and Wang, R. (2021). Solving 0-1 integer programming problem based on DNA strand displacement reaction network. *ACS Synth. Biol.* 10, 2318–2330. doi:10.1021/acssynbio.1c00244
- Wang, F., Lv, H., Li, Q., Li, J., Zhang, X., Shi, J., et al. (2020). Implementing digital computing with DNA-based switching circuits [J]. *Nat. Commun.* 11, 121. doi:10.1038/s41467-019-13980-y
- Woods, D., Doty, D., Myhrvold, C., Hui, J., Zhou, F., Yin, P., et al. (2019). Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. *Nature* 567 (7748), 366–372. doi:10.1038/s41586-019-1014-9
- Xiao, W., Zhang, X., Zhang, Z., Chen, C., and Shi, X. (2020). Molecular full adder based on DNA strand displacement. *IEEE Access* 8, 189796–189801. doi:10.1109/access.2020.3031221
- Xie, N., Li, M., Wang, Y., Lv, H., Shi, J., Li, J., et al. (2022). Scaling up multi-bit DNA full adder circuits with minimal strand displacement reactions. *J. Am. Chem. Soc.* 144, 9479–9488. doi:10.1021/jacs.2c03258
- Xiong, X., Zhu, T., Zhu, Y., Cao, M., Xiao, J., et al. (2022). Molecular convolutional neural networks with DNA regulatory circuits. *Nat. Mach. Intell.* 4 (7), 625–635. doi:10.1038/s42256-022-00502-7
- Xu, J., Chen, C., and Shi, X. (2022). Graph computation using algorithmic self-assembly of DNA molecules. *ACS Synth. Biol.* 11, 2456–2463. doi:10.1021/acssynbio.2c00120
- Xu, J. (2016). Probe machine. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 1405–1416. doi:10.1109/TNNLS.2016.2555845
- Yurke, B., Turberfield, A. J., Mills, A. P., Simmel, F. C., and Neumann, J. L. (2000). A DNA-fueled molecular machine made of DNA. *Nature* 406 (6796), 605–608. doi:10.1038/35020524
- Zarubiieva, I., Spaccasassi, C., Kulkarni, V., and Phillips, A. (2022). Automated leak analysis of nucleic acid circuits. *ACS Synth. Biol.* 11, 1931–1948. doi:10.1021/acssynbio.2c00084
- Zhang, C., Ma, X., Zheng, X., Ke, Y., Chen, K., Liu, D., et al. (2022). Programmable allosteric DNA regulations for molecular networks and nanomachines. *Sci. Adv.* 8 (5), eabl4589. doi:10.1126/sciadv.abl4589
- Zhao, S., Yu, L., Yang, S., Tang, X., Chang, K., and Chen, M. (2021). Boolean logic gate based on DNA strand displacement for biosensing: Current and emerging strategies. *Nanoscale Horizons* 6 (4), 298–310. doi:10.1039/d0nh00587h
- Zhu, E., Luo, X., Liu, C., and Chen, C. (2022). An operational DNA strand displacement encryption approach. *Nanomater. (Basel)* 12, 877. doi:10.3390/nano12050877