IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 May 2024, accepted 3 June 2024, date of publication 11 June 2024, date of current version 26 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3412919

RESEARCH ARTICLE

Matrix-Chain Multiplication Based on Combinatorial Allosteric DNA Strand Displacement

HENGYAN GUO[1], MINGLIANG WANG[2], AND XUEDONG ZHENG[1,3]

[1]College of Computer Science, Shenyang Aerospace University, Shenyang 110136, China
[2]BOGO Information Technology Company Ltd., Dalian 116600, China
[3]Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, Dalian University, Dalian 116622, China

Corresponding author: Xuedong Zheng (xuedongzheng@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61972266 and Grant 62272079, in part by the 111 Project under Grant D23006, in part by the Natural Science Foundation of Liaoning Province under Grant 2022-KF-12-14, in part by the Scientific Research Fund of Liaoning Provincial Education Department under Grant LJKZZ20220147, and in part by the Major Open Project of Key Laboratory for Advanced Design and Intelligent Computing of the Ministry of Education under Grant ADIC2023ZD002.

ABSTRACT DNA computing has gained widespread attention for leveraging the unique properties of DNA molecules to perform computational operations. As a fundamental tool for analyzing data and optimizing models, matrix operation plays an important role in intensive computational tasks and is a focus of DNA-based numerical computation. However, complex computing tasks are often achieved through transmitting and processing signals successively, which requires matrix operation to perform calculations sequentially. Therefore, it is important to find a way to perform successive matrix operation to ensure computational sustainability in molecular computing. In this paper, we present a successive DNA matrix operation method based on the mechanism of combinatorial allosteric DNA strand displacement. In this mechanism, the input signal and the output signal are completely decoupled in the base arrangement of the DNA domain, which makes it easy to implement successive DNA matrix operation and easily realize the connection of DNA signal processing units. Based on this mechanism, some basic DNA logic gates, such as AND gate, OR gate, and INHIBIT gate, were constructed first, then Boolean matrix multiplication was realized and, finally, matrix-chain multiplication was completed to illustrate successive DNA matrix operation. This study provides a new way to implement successive DNA matrix operation and enriches the toolbox for achieving intensive computational tasks through molecular computing.

INDEX TERMS DNA computing, biocomputing, DNA matrix computation, allosteric DNA strand displacement.

I. INTRODUCTION

DNA computing [1] is a rapidly developing interdisciplinary research field that integrates the principles of biology science and computer science, unlocking new possibilities to construct non-silicon-based computing systems at the nanoscale. After decades of development, DNA computing has made significant progress in various fields including nanotechnology [2], medical diagnosis [3], [4], artificial intelligence [5], [6], and information storage and processing [7], [8]. Due to the natural characteristics of DNA molecules, such as powerful computing parallelism, excellent programmability, and high specificity brought by the principle of Watson–Crick base pairing, DNA computing is considered particularly suitable for constructing molecular computing devices, providing a new way for achieving complex computing tasks on molecular levels.

As an electronic computer completes tasks by integrating basic processing units, DNA computing also achieves complex calculations by assembling primitive functional logic

The associate editor coordinating the review of this manuscript and approving it for publication was Lin Lin.

gates [9], [10] including YES, AND, OR, INHIBIT, and so on. By receiving and processing input signals transmitted from upstream reactions and generating output signals to trigger downstream reactions, these simple and effective DNA logic gates can be combined to form DNA circuits or DNA reaction networks [10] to perform complex computational operations as electronic circuits. So far, many DNA-based computing systems have been proposed to perform logic and arithmetic computations, such as half adder and subtractor [11], full adder and subtractor [12], encoder and decoder [13], [14], digital comparator [15], multiplexer and demultiplexer [16], parity generator/checker [17], multiple cascade logic circuit [18], and backtracking operation [19].

To perform DNA-based complex computation, various bio-engineered methods and biochemical reaction mechanisms, including DNA strand displacement [20], DNAzyme cleavage [21], allosteric regulation [22], [23], protein catalysis [24], self-assembly [25], G-quadruplex [26], nanoparticle [27] and DNA origami [28] have been applied and developed, establishing a solid foundation for the building of DNA molecular computing devices. Combining these bio-engineered methods and biochemical reaction mechanisms enables DNA computing to perform complex numerical operations in a programmable and modular manner as electronic circuits [29]. For example, Qian and Winfree constructed a large-scale DNA circuit to compute the integer portion of the square root of a four-bit binary number [30], making an incredible breakthrough in DNA computing to complete numerical calculations. Later on, with the help of strand-displacing DNA polymerase, Song et al. developed a fast and compact DNA logic circuit to calculate the binary square root with a four-bit input [31], improving both the speed of computation and the number of DNA strands needed. Furthermore, Zhou et al. constructed a large-scale DNA-based logic system [32] that can compute the cube root of a 10-bit binary number (within the decimal number 1000) for the first time, which shows the power of DNA computing in executing complex numerical computation.

Apart from numerical computation, DNA-based computing systems have also been programmed as molecular neural networks [5], [6], [33], which have unparalleled advantages in dealing with complex problems due to their brain-like working patterns. The computing process of molecular neural networks consists of several fundamental numerical operations, with matrix calculation being a crucial component among them. Matrix operation serves as a fundamental tool for analyzing data and optimizing models, playing a pivotal role in intensive computational tasks, and representing a key focus in DNA-based numerical computation. In 1997, Oliver first introduced a conceptual framework for performing matrix multiplication [34]. Based on the strategy of associative toehold activation [35], Genot et al. proposed a combinatorial displacement of DNA strands to achieve matrix multiplication which is suited to implement linear operations [36]. To reduce leakage in DNA strand displacement, Xu et al. introduced a DNA matrix operation method based on the substrate-binding mechanism of DNAzyme requiring the connection of two auxiliary strands [37], where the results of matrix calculation were represented by fluorescence signals. Although these previous works offer insights and bio-engineered methods for executing DNA matrix operation, complex computational tasks are often achieved through successive transmission and processing of signals, which require matrix operation to perform the computations sequentially. Therefore, it is important to develop and combine new DNA computing mechanisms to perform successive matrix operation to ensure computational sustainability in a programmable and modular manner.

Notably, conformational signals are essential in the control of signal transmission and processing in DNA reaction networks, and allosteric regulation has found broad applications in DNA computing, such as allosteric DNAzyme [38], [39], dynamic allosteric control [40], programmable allosteric DNA regulation [22], intramolecular conformational motions [41], conformational cooperative regulation mechanism [42]. These allosteric mechanisms provide more possibilities to achieve successive matrix operation using DNA molecules.

In this work, we aim to explore a new DNA-based mechanism, named combinatorial allosteric DNA strand displacement, to perform successive DNA matrix operation to adapt to intensive computational tasks. In this mechanism, the input DNA strands and the output DNA strands are completely decoupled in the base arrangement of the DNA domain, which makes it easy to implement successive DNA matrix operation and can easily realize the connection of DNA signal processing units, which is particularly suitable for successive linear operations. Based on this mechanism, some basic DNA logic gates, such as YES, AND, OR, and INHIBIT, were constructed first, then DNA matrix multiplication and weighted summation were achieved and finally, DNA matrix-chain multiplication was completed. And the calculation results of the DNA matrix are non-one-shot, implying that the upstream products can continue to participate in the next operations, enabling cascading operations of matrices. Therefore, the excellent scalability and strong decoupling capability of this mechanism lay the foundation for constructing more complex and extensive computing platforms, which hold significant potential in large-scale molecular information processing system.

II. COMBINATORIAL ALLOSTERIC DNA STRAND DISPLACEMENT

In performing successive DNA matrix operation, the ideal basic processing unit should possess some essential characteristics, such as good decoupling of input and output, multiple signal transductions, and programmable integration. Therefore, the design of combinatorial allosteric DNA strand displacement is developed as shown in Fig. 1A, where the DNA complex S/O is defined as the foundational signal processing structure, consisting of two DNA strands S and O. As shown in Fig. 1A, the strand S mainly consists of four

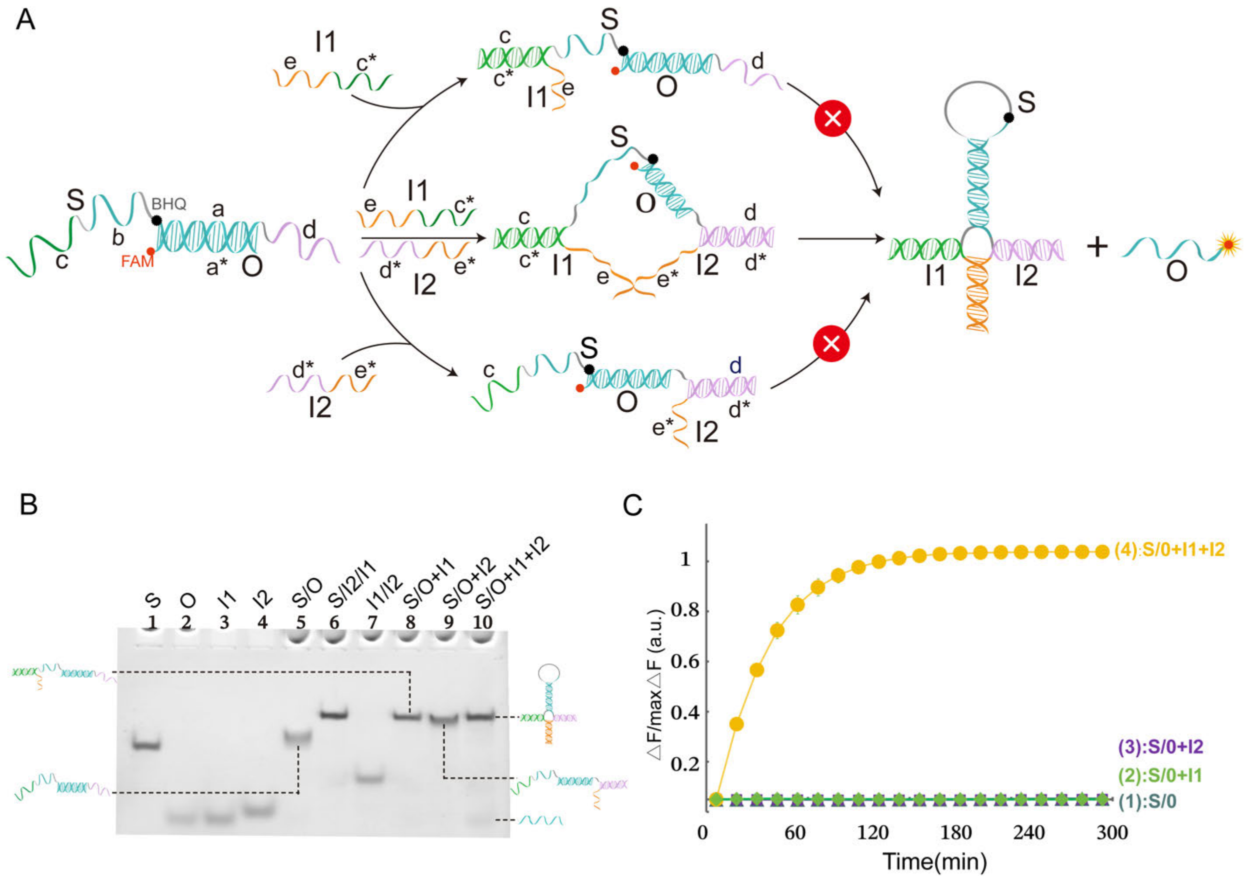


FIGURE 1. **(A) Schematic representation of the combinatorial allosteric DNA strand displacement. The 3' end of strand O is labelled with the fluorophore FAM, and the quencher BHQ is modified at the complementary position of strand S. In this paper, all single DNA strands are arranged from left to right or from top to bottom, representing the 5' to 3' orientation. (B) PAGE gel analysis of the combinatorial allosteric DNA strand displacement. The involved DNA strands and complexes were labelled above the lane number. Here, DNA complex is represented by its elements linked by slashes, and the symbol + denotes the addition of DNA strands. Lane 1, single strand S; lane 2, single strand O; lane 3, input DNA strand I1; lane 4, input DNA strand I2; lane 5, DNA complex S/O used as the fundamental signal processing structure; lane 6, DNA complex S/I2/I1; lane 7,duplex I1/I2; lane 8, products of DNA complex S/O mixed with input strand I1; lane 9, products of DNA complex S/O mixed with input strand I2; lane 10; products of DNA complex S/O triggered by input DNA strands I1 and I2. The concentrations of the reactants [S] = [I1] = [I2] = [S/O] = [I1/I2] = [S/I2/I1] = 1μM, and [O] = 2μM. (C) Time-dependent normalized fluorescence changes (ΔF/MaxΔF) during the reaction process. Curves (1) to (4) demonstrate the processing unit responses to different inputs. Here, the symbol + denotes the addition of the strand. The concentrations of the reactants [I1] = [I2] = [S/O] = 0.4μM. All data represent the average of three replicates. Error bars represent one standard deviation from triplicate analyses. The sampling interval is 6 minutes, with 50 cycles, resulting in a duration of 300 minutes.**

functional domains: domain a with 18-nt length that carries the output strand O through base pairing; domain b with 12-nt length that can hybridize with a portion of domain a, facilitating the formation of a hairpin structure in strand S through an internal strand displacement induced by the proximity effect; and domains c and d, both with 10-nt length, serving as binding domains for the input DNA strands. The input strand I1 has two functional domains: input domain c* at the 3' end and linking domain e at the 5' end, and it can partially hybridize to strand S by base pairing between domains c* and c. The input strand I2 has a similar domain design to the strand I1: input domain d* at the 5' end and linking domain e* at the 3' end, and it can partially hybridize to the strand S by base paring between domains d* and d. Based on the domain design, for the existence of strand O, the internal strand displacement possibly induced by the proximity effect is blocked and the output will not be generated in the initial state. Once both input strands I1 and I2 are introduced into the system, the binding of strands I1, I2, and S/O will pull the domains c and d on both sides of strand S close, triggering a conformational change in strand S. This conformational change induces a self-complementary hybridization of the functional domains a and b of the strand S, facilitating internal strand displacement in the complex S/O and releasing strand O as the output. After that, the strand S forms a stable triple-stranded DNA complex S/I2/I1 by binding with the two input strands I1 and I2, creating a three-way DNA junction. However, only one input strand, I1 or I2, does not cause a change in the configuration of strand S and the output strand O cannot be released, serving as a natural AND gate. Notably, in this mechanism, there are no common or complementary DNA domains between the input DNA strand and the output DNA strand, which means the input signal and the output signal are completely decoupled in the base arrangement of DNA domains. The decoupling of input and output signals allows the processing unit to perform more complex computational

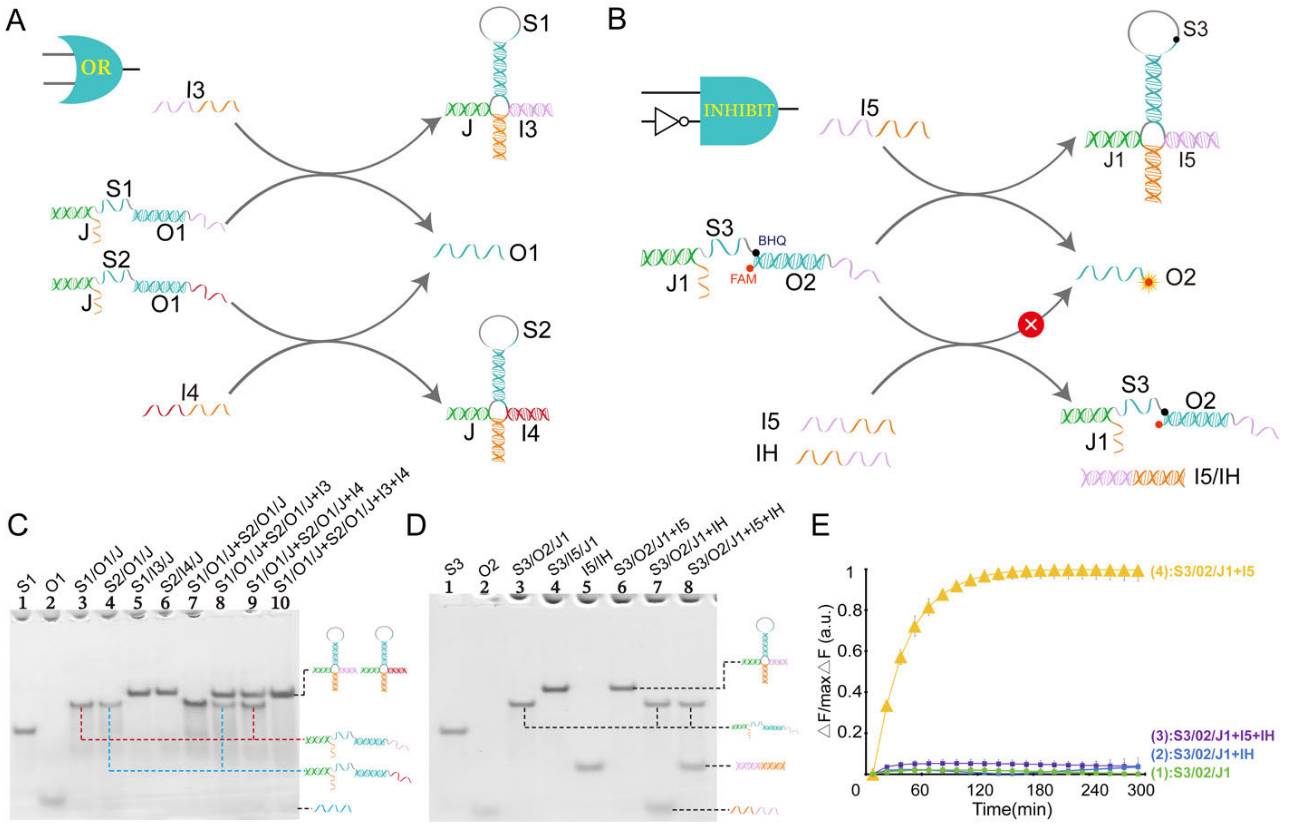


FIGURE 2. (A) Schematic representation of OR gate. (B) Schematic representation of INHIBIT gate. The 3′ end of strand O2 is labelled with the fluorophore FAM, and the quencher BHQ is modified at the complementary position of strand S3. (C) PAGE gel analysis of OR gate. Lane 1, single strand S1; lane 2, single strand O1; lane 3, DNA complex S1/O1/J; lane 4, DNA complex S2/O1/J; lane 5, DNA complex S1/I3/J; lane 6, DNA complex S2/I4/J; lane 7, mixture of DNA complexes S1/O1/J and S2/O1/J; lane 8, products of the OR gate triggered by input strand I3; lane 9, products of the OR gate triggered by input strand I4; lane 10, products of the OR gate triggered by both input strands I3 and I4. The concentrations of the reactants [S1] = [S1/O1/J] = [S2/O1/J] = [S1/I3/J] = [S2/I4/J] = [I3] = [I4] = 1μM, and [O1] = 2μM. (D) PAGE gel analysis of INHIBIT gate. Lane 1, single strand S3; lane 2, single strand O2; lane 3, DNA complex S3/O2/J1; lane 4, DNA complex S3/I5/J1; lane 5, duplex I5/H; lane 6, products of the INHIBIT gate with the addition of input strand I5; lane 7, products of the INHIBIT gate with the addition of inhibitory strand IH; lane 8, products of the INHIBIT gate with the addition of both input strand I5 and inhibitory strand IH. The concentrations of the reactants [S3] = [I5] = [IH] = [S3/O2/J1] = [S3/I5/J1] = [I5/IH] = 1μM, and [O2] = 2μM. (E) Time-dependent normalized fluorescence changes (ΔF/MaxΔF) during the reaction process of the INHIBIT gate. Curves (1) to (4) demonstrate the INHIBIT gate responses to different inputs. The concentrations of the reactants [S3/O2/J1] = [I5] = [IH] = 0.4μM. All data represent the average of three replicates. Error bars represent one standard deviation from triplicate analyses. The sampling interval is 6 minutes, with a total of 50 cycles, resulting in a duration of 300 minutes.

tasks through signal combinations, and it makes it easier to construct DNA reaction networks by connecting the basic processing units. In addition, although the displacement is not direct, from the perspective of black-box functionality, the two input DNA strands I1 and I2 achieve the displacement of the output strand O through allosteric mechanisms of the foundational signal processing structure. Based on the above two characteristics of the mechanism, we refer to the mechanism as combinatorial allosteric DNA strand displacement.

The correctness of the mechanism was experimentally confirmed through PAGE gel. As shown in Fig. 1B, the foundational signal processing structure S/O can be observed clearly in lane 5 as a single gel band. When only one input DNA strand, I1 or I2, existed, no conformational change happened, resulting in the formation of triple-stranded DNA complexes, S/O/I1 or S/I2/O (lanes 8 and 9). Once the two input DNA strands I1 and I2 coexisted in the system, the allosteric mechanism was triggered and the output DNA strand O was released as shown in lane 10, concurrently forming the gel band corresponding to the DNA complex S/I2/I1.

A fluorescence assay was also conducted to monitor the combinatorial allosteric DNA strand displacement in real time as shown in Fig. 1C. In the presence of both the input strands I1 and I2, a significant fluorescence signal was produced (curve 4). In contrast, no remarkable increase of fluorescence signal could be observed in curves 1-3 when no input DNA strand existed or only one input DNA strand was added. The results demonstrate the successful performance of the mechanism of combinatorial allosteric DNA strand displacement.

III. OR GATE AND INHIBIT GATE

To further illustrate the flexibility of the mechanism of combinatorial allosteric DNA strand displacement, two basic logic

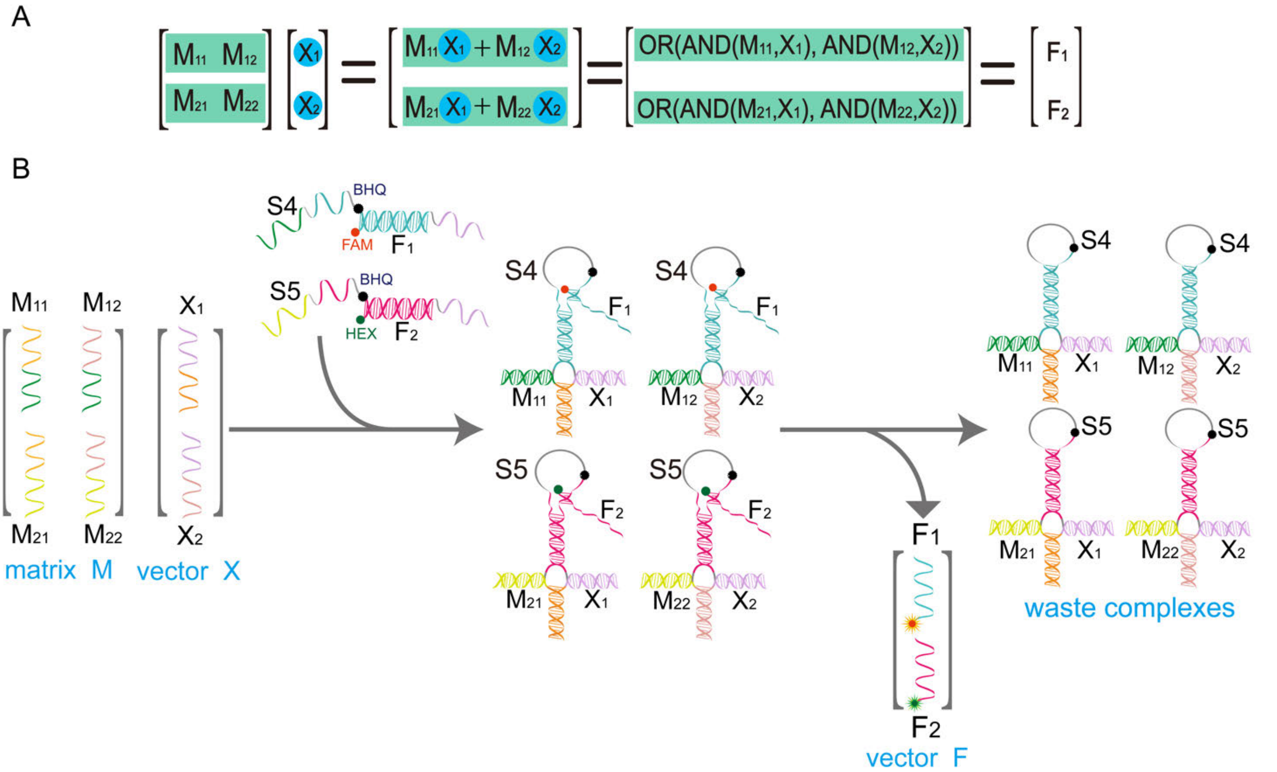


FIGURE 3. (A) Mathematical principles of matrix multiplication. (B) Schematic diagram illustrating the principles of the matrix M multiplied by vector X. The 3′ end of strand F1 is labelled with the fluorophore FAM, the quencher BHQ is modified at the complementary position of the substrate strand S4. The 3′ end of strand F2 is labelled with the fluorophore HEX, and the quencher BHQ is modified at the complementary position of the substrate strand S5.

gates, an OR gate, and an INHIBIT gate, were established. Moreover, the OR gate is also one of the essential processing units for DNA matrix multiplication. The OR gate is designed to respond to either input DNA strand, I3 or I4. Accordingly, as illustrated in Fig. 2A, two triple-stranded complexes, S1/O1/J and S2/O1/J, are designed to serve as receptors for each input strand, implementing the OR logic gate by the combination of two basic processing units. When only input strand I3 is present, the two functional domains of strand I3 hybridize with the linking domain of strand J and the binding domain of strand S1 respectively, causing strand S1 to undergo internal strand displacement and release output strand O1. A similar process occurs when the input is strand I4. Thus, whenever either input strand I3 or I4 is present, the system will generate an output strand, functioning as an OR gate. As shown in Fig. 2C, the OR gate has undergone PAGE gel experiments for all possible input combinations. From lane 7 in Fig. 2C, no output was generated when there were no input strands. Lanes 8 and 9 demonstrate that when only I3 or I4 was present, the triple-stranded DNA complex S1/I3/J or S2/I4/J formed, producing the output strand O1. Lane 10 shows that when both I3 and I4 were input simultaneously, the gel band corresponding to a mixture of the complexes S1/O1/J and S2/O1/J disappeared and a gel band of strand O1 was produced, which means the gate made a correct response to the inputs. So, in all cases, the OR gate has produced the correct outputs.

Next, as shown in Fig. 2B, the INHIBIT gate is composed of three DNA strands S3, O2, and J1, and the output is true if and only if the input strand I5 itself exists. The inhibitory strand IH is designed to have a fully complementary base arrangement with the input strand I5, and once the strand IH and the strand I5 coexist, strand IH preferentially hybridizes with strand I5, resulting in the logic gate having no output. The reaction results were experimentally confirmed through PAGE gel as shown in Fig. 2D. In the absence of input strand I5, the gate complex S3/O2/J1 stably existed in the solution, and the gel band of complex S3/O2/J1 can be observed in lane 3. When input strand I5 was added to the solution, the complex S3/O2/J1 reacted with input strand I5, forming new gel bands corresponding to the complex S3/I5/J1 as shown in lane 6. When only the inhibitory strand IH was added, the gel bands corresponding to the complex S3/O2/J1 still existed, and only the gel band corresponding to the inhibitory strand IH was observed in lane 7. When both input strand I5 and inhibitory strand IH were added, the gel bands corresponding to the complex S3/O2/J1 complex still existed and the formation of the double-stranded DNA I5/IH was observed in lane 8, which confirmed the function of the INHIBIT gate. From Fig. 2E, the fluorescence assay further demonstrates the

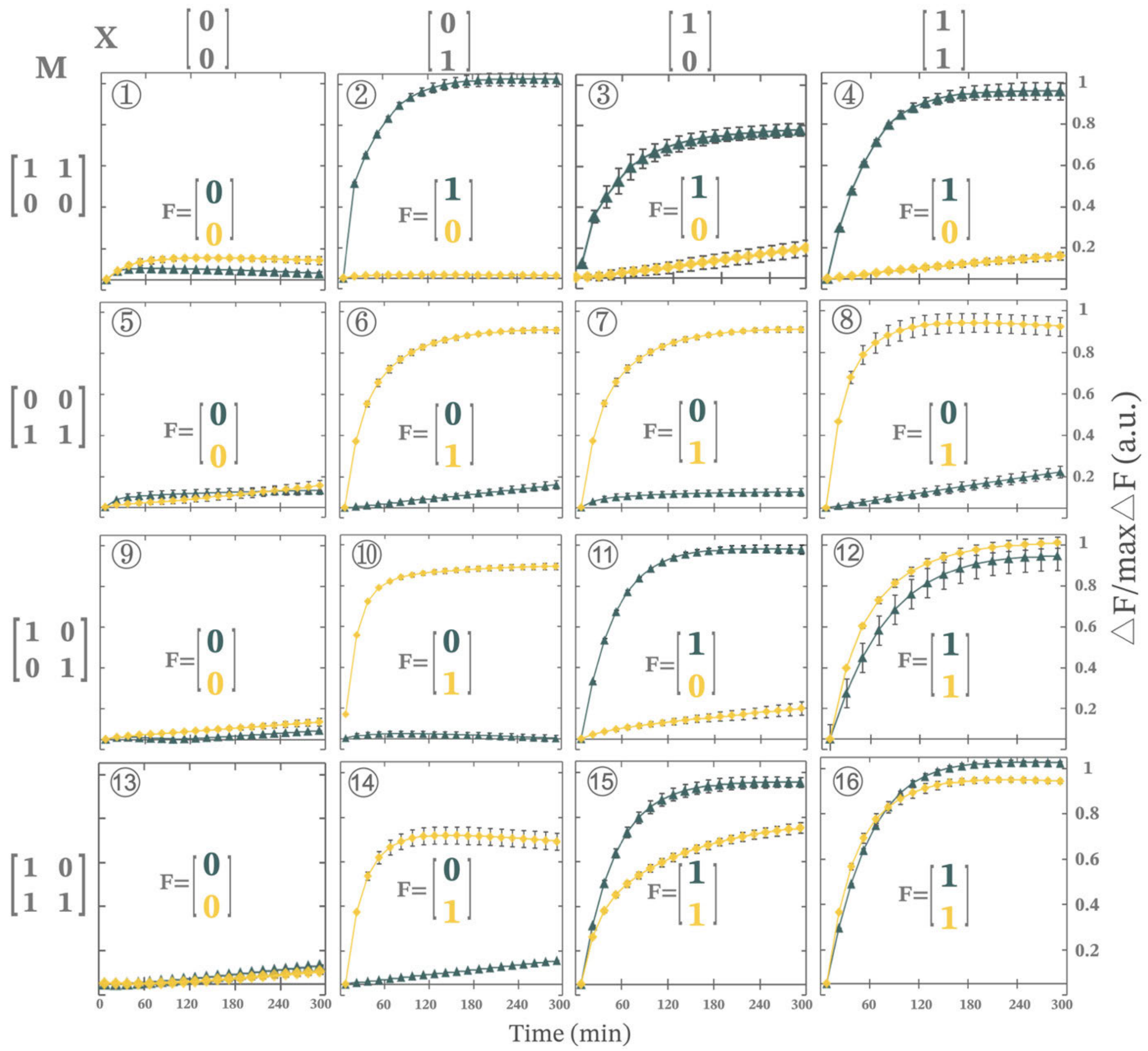


FIGURE 4. Time-dependent normalized fluorescence changes (ΔF/MaxΔF) of the 16 sets of experimental results corresponding to the multiplication of four distinct matrices M with four distinct vectors X. The brown-blue curve represents the value of the first-row element F_1 in the result vector F, indicated by the fluorophore FAM. The yellow curve represents the value of the second-row element F_2 in the result vector F, indicated by the fluorophore HEX. The concentrations of all complexes, along with those of single strands, were set at 0.4μM. All data represent the average of three replicates. Error bars represent one standard deviation from triplicate analyses. The sampling interval is 6 minutes, with 50 cycles, resulting in a duration of 300 minutes.

feasibility of this INHIBIT logic gate. In addition, to verify the possibility of the assembly of basic logic gates, a two-layer cascading DNA circuit was established as shown in Fig. S1 of the Supplementary, which confirms that the basic processing unit has good scalability.

IV. MATRIX MULTIPLICATION

To demonstrate the computational power of combinatorial allosteric DNA strand displacement, we applied it to perform Boolean matrix multiplication. In general, multiplying two n × n Boolean matrices requires n^3 AND operations. Each element of the matrix product needs a fluorescent signal to report the calculation result. The larger the dimension of matrix multiplication, the more fluorescent channels are required. Hence, without losing generality, we focused on validating Boolean matrix multiplication operation between a 2 × 2 matrix and a 2 × 1 vector, which only requires two fluorescent channels.

From the mathematical operations depicted in Fig. 3A, each element F_i in the result vector F can be calculated through the dot product of the i-th row vector (M_{i1}, M_{i2}) in matrix M and vector X, where two AND operations and an OR operation are needed. However, for the value 1 or 0 is represented by the presence or absence of DNA strands respectively, the OR operation is natural and no extra operations are needed. Therefore, the core of DNA Boolean matrix multiplication is to construct AND gates, combinatorially connecting the row vector of the matrix M with the column vector X to

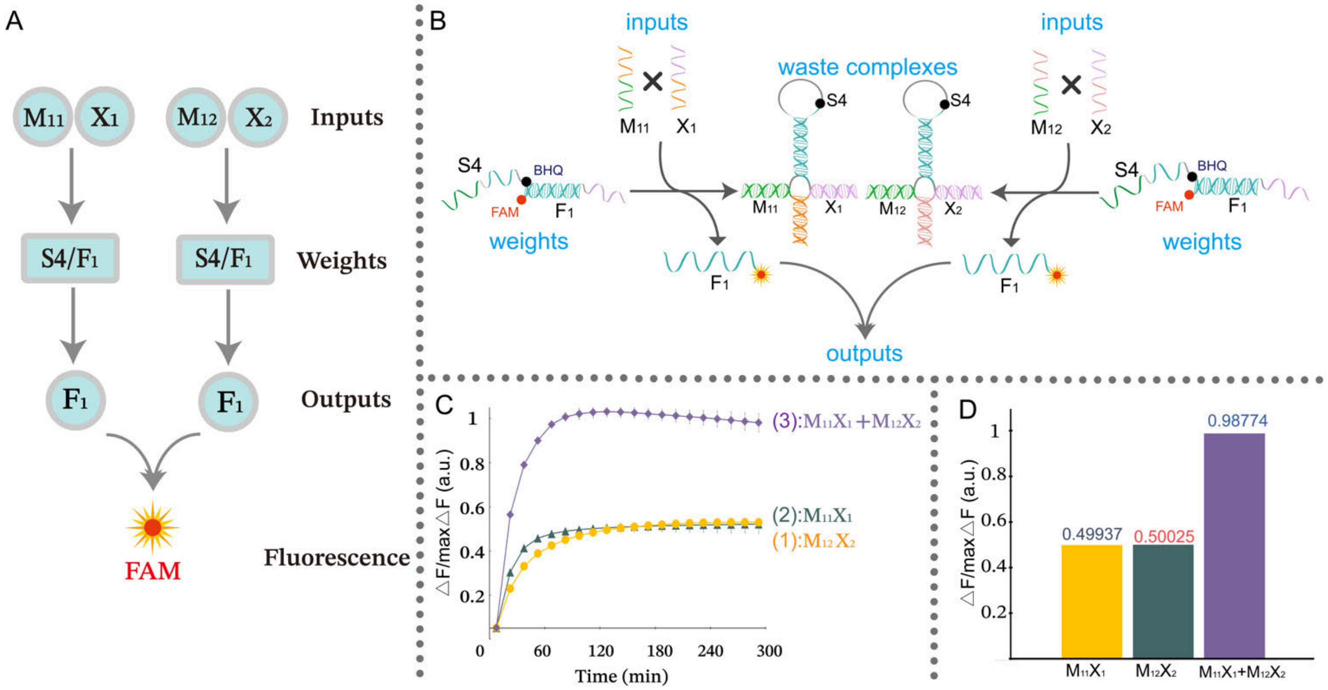


FIGURE 5. (A) Illustration of a weighted sum of matrix multiplication for F_1. (B) Schematic diagram of the weighted summation process for F_1. The specific location of fluorescent labelling is the same as those for Boolean matrix multiplication. (C) Time-dependent normalized fluorescence changes ($\Delta F/\mathrm{Max}\Delta F$) during the reaction process. Curves (1)-(3) show the weighted summation results under different input combinations. The concentrations of the reactants $[M_{11}] = [M_{12}] = [X_1] = [X_2] = 0.4\mu M$, and $[S4/F_1] = 0.8\mu M$. All data represent the average of three replicates. Error bars represent one standard deviation from triplicate analyses. The sampling interval is 6 minutes, with 50 cycles, resulting in a duration of 300 minutes. (D) Histogram of the fluorescence data of the weighted summation process of F_1.

complete the dot product operation. As shown in Fig. 3B, single DNA strands with functional domains represent the elements of matrix M and vector X, and DNA complexes $S4/F_1$ and $S5/F_2$ serve as the AND gates. In the computational process, the DNA complex $S4/F_1$ ($S5/F_2$) serves to connect the first-row (second-row) vector of matrix M with vector X through the binding domains. Simultaneously, the elements in the first-row (second-row) vector of matrix M could be linked with the corresponding elements in vector X if they have complementary linking domains. When both matrix elements and vector elements exist, the DNA complex $S4/F_1$ ($S5/F_2$), driven by a conformational change, displaces the DNA strand F_1 (F_2), resulting in the output of the computational results. Compared with the previous work using a combinatorial displacement mechanism [36], no slow branch migration occurs, which is the main reason for leakage. Moreover, the calculation results are represented by single DNA strands, not fluorescence signals [37], which can serve as the next input for successive matrix operation, endowing the DNA matrix operation with the ability to achieve high-intensity computational tasks through modular assembly.

We conducted a total of 32 independent experiments, with 16 sets of fluorescence assay data presented in Fig. 4 and the remaining 16 sets shown in Fig. S2. From fluorescence curves, it can be observed that for each instance of the matrix M being multiplied by the vector X, the experiment consistently yielded correct results. It is worth noting that when elements within the same column of matrix M react with the same element in vector X, a competitive relationship is established. For instance, both M_{11} and M_{21} will simultaneously react with X_1. Due to the complexity of base sequence design, ensuring that the reaction rates of these competitive reactions are equal poses a challenge. As a result, this competitive relationship will reduce the concentration and reaction rate of the corresponding reaction, resulting in a slightly inferior outcome for the reaction with weaker competitiveness compared to the other, as shown in the fifteenth case in Fig. 4, where the amplitude of the rise and the final percentage of the product in the F_2 curve (HEX) is inferior to the F_1 curve (FAM). The experimental results were also verified by PAGE gel as shown in Table S1, Fig. S3 and Fig. S4.

In addition, as shown in Fig. S5 and Fig. S6, we verified DNA matrix multiplication between a 2 × 3 Boolean matrix and a 3-dimensional vector, which indicates that the mechanism of combinatorial allosteric DNA strand displacement has strong horizontal scalability in DNA matrix multiplication.

V. WEIGHTED SUM OF BOOLEAN MATRIX MULTIPLICATION

To validate the accuracy of DNA matrix multiplication based on combinatorial allosteric DNA strand displacement, we designed a weighted summation for Boolean matrix multiplication. That is, after multiplying the row vector of the

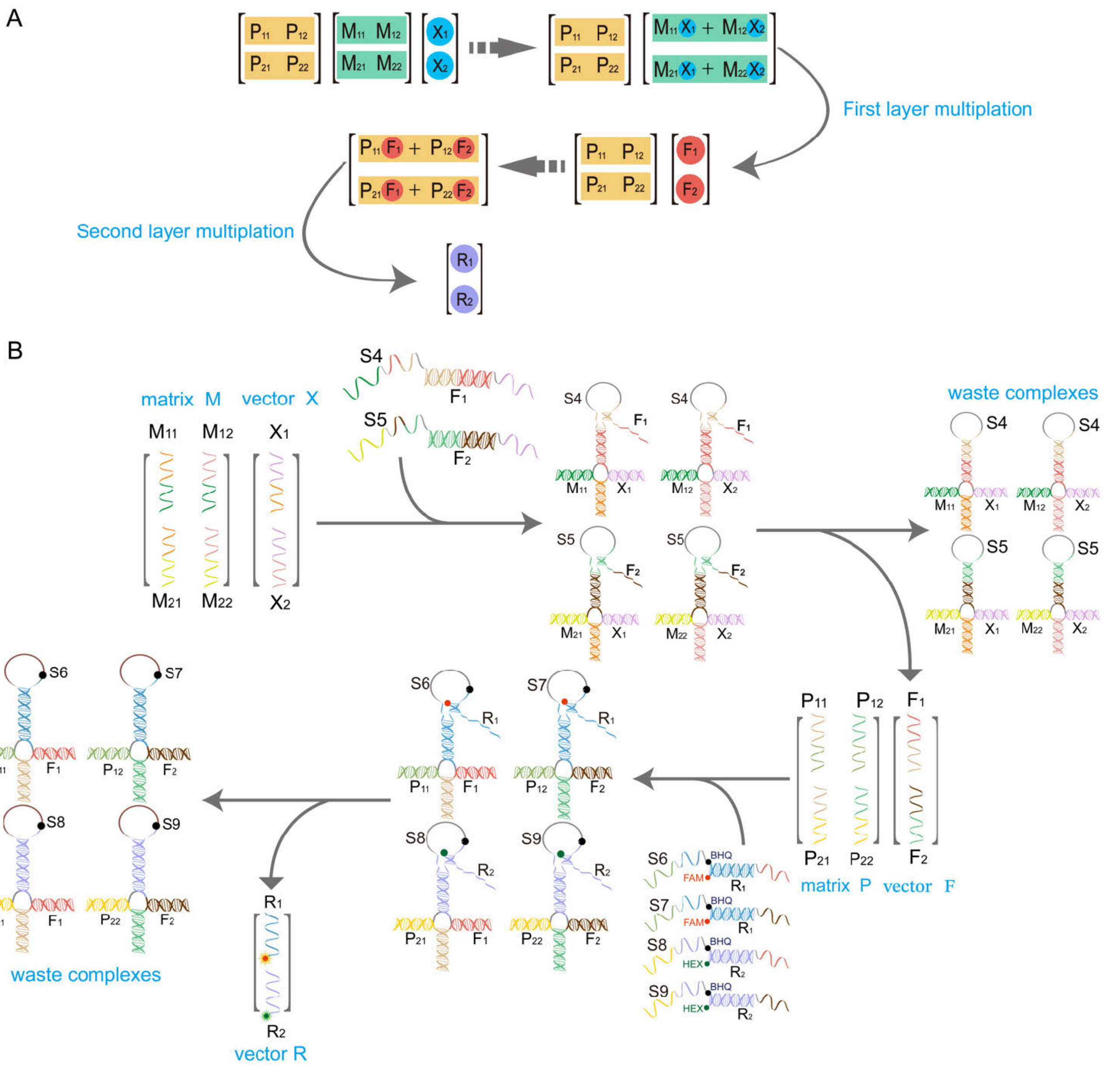


FIGURE 6. (A) Mathematical principle of matrix-chain multiplication. (B) Schematic diagram of matrix-chain multiplication. The functional domains of strands F_1 and F_2 in the figure have been recolored to match the colors of the functional domains of DNA strands in downstream reactions. The 3′ end of strand R_1 is labelled with the fluorophore FAM and the quencher BHQ is modified at the complementary position of the substrate strand S6 (S7). The 3′ end of strand R_2 is labelled with the fluorophore HEX, and the quencher BHQ is modified at the complementary position of the substrate strand S8 (S9).

matrix M with the corresponding elements in the vector X, the weighted summation is calculated according to the respective weights. In this scheme, we fix the weights at 1, which means the output signal is contingent upon the presence or absence of the matrix elements. Fig. 5A illustrates the weighted summation process of the first-row element F_1 in the result vector F, where the input represents the elements involved in the operation within the matrix, the combination of input vector elements is multiplied by their respective weights, and the same fluorescence signal is released to measure the results.

As shown in Fig. 5B, when only the input combination $M_{11}X_1$ is introduced, it binds to the weighted complex S4/F_1 to displace the single strand F_1, generating a fluorescence signal, and the same process occurs when only input combination $M_{12}X_2$ is present. When both input combinations are simultaneously introduced into the solution, the strand F_1 is displaced, resulting in a fluorescence signal approximately twice that of either input case alone. Thus, the calculation of Boolean weighted sum is achieved. We validated the rationality of the weighted summation for F_1 through quantitative fluorescence experiments as shown in Fig. 5C and Fig. 5D. From the histogram in Fig. 5D, it can be intuitively observed that when both input combinations, $M_{11}X_1$ and $M_{12}X_2$, were added simultaneously, the fluorescence value reached a

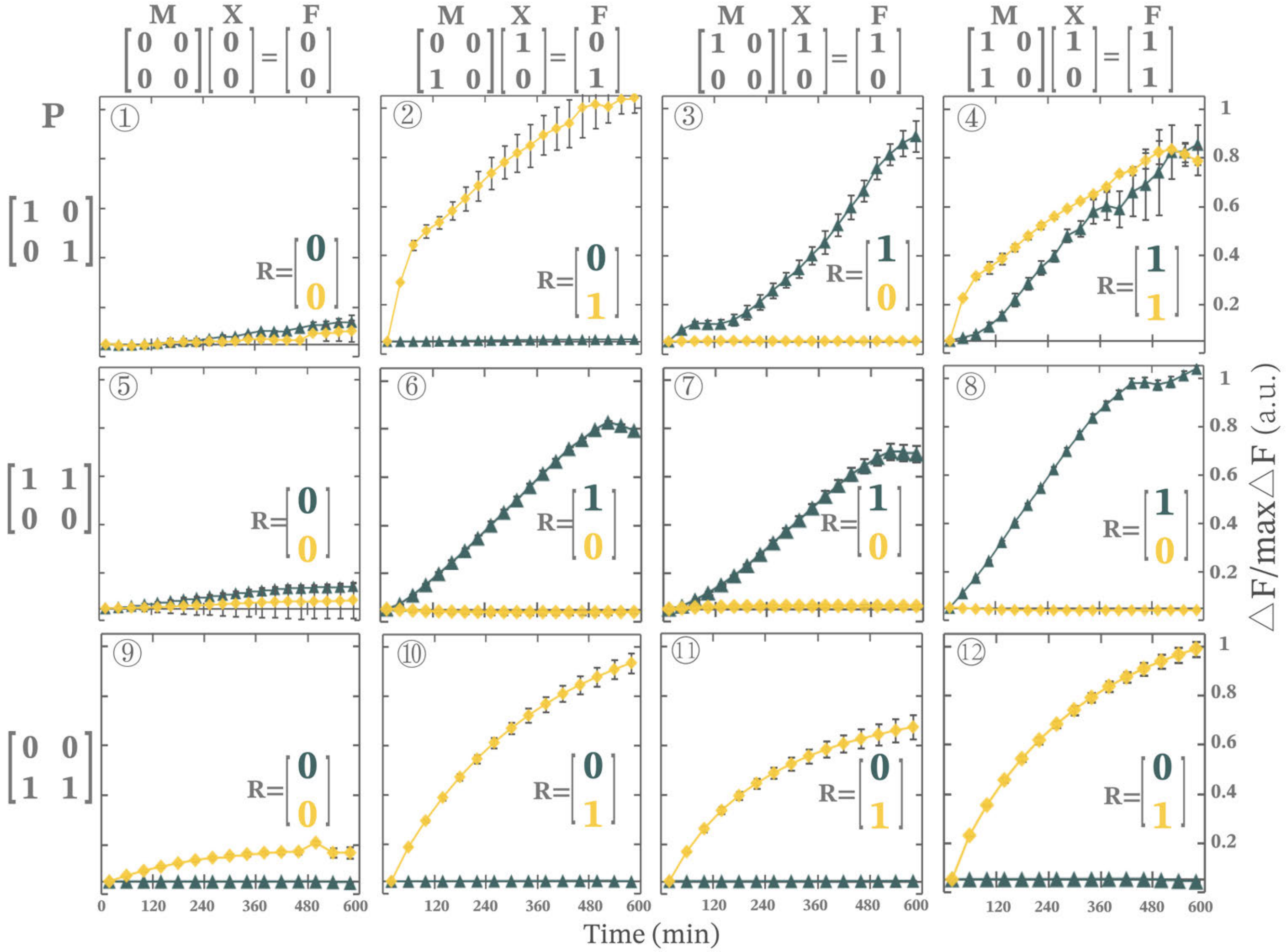


FIGURE 7. Time-dependent normalized fluorescence changes (ΔF/MaxΔF) of the 12 sets of experimental results corresponding to the matrix-chain multiplication. The brown-blue curve represents the value of the first-row element R_1 in the result vector R, indicated by the fluorophore FAM. The yellow curve represents the value of the second-row element R_2 in the result vector R, indicated by the fluorophore HEX. The concentrations of all complexes, along with those of single strands, were set at 0.4μM. All data represent the average of three replicates. Error bars represent one standard deviation from triplicate analyses. The sampling interval is 6 minutes, with 100 cycles, resulting in a duration of 600 minutes.

maximum, and the resulting fluorescence signal was approximately twice as large as when either input combination, $M_{11}X_1$ or $M_{12}X_2$, was added individually. The weighted sum chart for the second-row element F_2 of the result vector F is provided in Fig. S7. These experiments demonstrate the feasibility of our basic mechanism in weighted sum of Boolean matrix multiplication.

VI. MATRIX-CHAIN MULTIPLICATION

To verify the feasibility of our architecture on successive matrix operation, we designed a matrix-chain multiplication operation. In this process, the result vector generated by the previous matrix multiplication operation will be used as the input vector for the next matrix operation, continuing to participate in subsequent matrix operation. The mathematical principle of implementing successive matrix multiplications is depicted in Fig. 6A, where the matrix chain is composed of two Boolean matrixes P and M, the input is the vector X, and the result is vector R.

From Fig. 6B, the matrix-chain multiplication is performed through a two-layer cascading DNA circuit, where the upstream DNA products F_1 and F_2 obtained from the first-layer matrix multiplication are taken as the input DNA strands for the downstream matrix P. To perform the second-layer matrix multiplication accurately, the fundamental processing units consist of four complexes to perform AND operations: S6/R_1, S7/R_1, S8/R_2, and S9/R_2. It is worth noting that verifying all possibilities in the matrix-chain multiplication would be highly labour- and time-intensive (a total of 16 × 64 situations need to be investigated). Therefore, we only verified a subset of the possible results, a total of 12 independent experiments.

As illustrated in Fig. 7, the experimental results of the 12 independent experiments were confirmed by fluorescence assays. From Fig. 7, the fluorescence signals show that the cascading DNA circuit through modular assembly of basic processing units could generate correct results for the 12 different cases of matrix-chain multiplication.

It is worth noting that although some by-products exist in the matrix multiplication calculation of the first layer, as illustrated in Fig. S3B and Fig. S4A, the decoupling of the input and output in terms of base arrangement within the basic processing units ensures that these by-products have virtually no impact on the results of subsequent matrix calculations. To further confirm the degree of influence of these by-products on the matrix-chain multiplication, we performed a leakage analysis of the calculation process as shown in Fig. S8. From Fig. S8A, when vector X was absent, mixing the matrix M, P, and DNA complexes performing AND operations did not produce any significant output. Similarly, when the matrix M in the first layer is absent, mixing the matrix P, vector X, and DNA complexes performing AND operations also did not generate any output as shown in Fig. S8B. Therefore, when performing matrix-chain multiplication, DNA signals are transmitted and processed successively between different matrices, and there is no significant crosstalk between upstream and downstream signals. The successive transmission and processing of signals in the process of the matrix-chain multiplication demonstrate the excellent programmability and modular assembly capability of the combinatorial allosteric DNA strand displacement.

VII. CONCLUSION

The rapid development of DNA computing has opened new possibilities for utilizing biological molecules to perform intensive numerical calculations at the nanoscale. Matrix operation, as a fundamental tool for executing numerical calculations, occupies an important position in DNA numerical computing, providing a novel tool and method for realizing artificial intelligence based on biocomputing. Previous research on DNA matrix operation has achieved a series of successes. However, these methods may be constrained by the specificity of DNAzymes and the reaction conditions, increasing the complexity of experiments. In this work, we propose a simple and effective method of combinatorial allosteric DNA strand displacement. This method utilizes the conformational change of hairpin DNA molecules triggered by the combinatorial input DNA strands to induce DNA strand displacement and form output DNA strands. This method decouples the input and output of the DNA strand in terms of the base arrangement, enabling the construction of DNA circuit in a programmable modular way. Based on this method, we have constructed a series of primitive DNA logic gates and cascading DNA circuits, which have been further applied to DNA matrix multiplication. Due to the non-interfering design of the input and output of DNA strands at the domain level, this method is particularly suitable for successive linear operations. We achieved Boolean matrix multiplication and Boolean weighted sum function. Moreover, as the computational results remain in the form of DNA strands, we have successfully implemented matrix-chain multiplication through the cascading of DNA circuits. This enables the construction of complicated DNA neural networks through the successive transmission and processing of DNA signals. The advantage of applying combinatorial allosteric DNA strand displacement to DNA matrix multiplication lies in its ability to perform successive matrix operation simply and accurately, enabling dense numerical computations. Furthermore, its excellent scalability and robust decoupling capability lay the foundation for building more complex and extensive computing platforms.

VIII. EXPERIMENTAL DETAILS

A. MATERIALS

All DNA strands were purchased from Sangon Biotech Co., Ltd. (Shanghai, China). Unmodified DNA strands were purified by polyacrylamide gel electrophoresis (PAGE), and modified DNA strands with fluorophore were purified by high-performance liquid chromatography (HPLC). The sequences of all strands are listed in Table S2 and simulated using NUPACK as shown in supplementary Fig. S9 and Fig. S10. DNA strands were dissolved in water as stock solution and quantified using a Nanodrop One spectrophotometer (Thermo Fisher Scientific Inc. USA), and absorption intensities were recorded at $\lambda = 260$nm. Other chemicals were of analytical grade and were used without further purification.

B. PREPARATION OF DNA COMPLEXES

All DNA complexes were formed by annealing: firstly, the mixture of the required DNA single strands in 1 × TAE/Mg^{2+} buffer (40 mM Tris, 20 mM acetic acid, 1 mM EDTA2Na and 12.5 mM $Mg(OAc)_2$, pH 8.0) was heated at 95°C for 4 min, 65°C for 30 min, 50°C for 30 min, 37°C for 30 min, 22°C for 30 min, and preserved at 20°C; and then the substrates were added into the annealed mixture and incubated at constant temperature 20°C for 4h. Note that no PAGE purifications were applied in all experiments.

C. DNA STRAND DISPLACEMENT

Reactions of combinatorial allosteric DNA strand displacement were triggered in 1× TAE/Mg^{2+} buffer (40 mM Tris, 20 mM acetic acid, 1 mM EDTA2Na, and 12.5 mM $Mg(OAc)_2$, pH 8.0). The input DNA strands were added to a solution containing DNA complexes and reacted for >2h at 25°C. Next, the displaced products were stored at 25°C for native PAGE or fluorescence detection.

D. NATIVE PAGE

Samples were mixed with 6 × loading buffer (Takara) or 36% glycerine solution and run on 12% native polyacrylamide gel in 1 × TAE/Mg^{2+} buffer (40 mM Tris, 20 mM acetic acid, 1 mM EDTA2Na, and 12.5 mM $Mg(BAc)_2$, pH 8.0) at 90V for no >3h at 4°C.

E. FLUORESCENCE ASSAY

The fluorescent results were obtained using real-time PCR (Agilent, Palo Alto, USA, G8830A) equipped with a 96-well fluorescence plate reader. Two types of fluorophores, FAM and HEX, were modified on DNA reporter strands. The reactions were performed in 1 × TAE/Mg^{2+} buffer (40mM Tris, 20mM acetic acid, 1mM EDTA2Na, and 12.5mM $Mg(OAc)_2$, pH 8.0) and a typical 20ul reaction volume at 25°C. All fluorescence experiments were performed thrice to ensure reproducibility. The detection time interval was 6 min. The fluorescence data were obtained by averaging the values from three replicates of the experimental results, and then the data were normalized as $\Delta F/\text{Max}\Delta F$ where ΔF represented the fluorescence intensity change at each time point and $\text{Max}\Delta F$ indicated the highest value of fluorescence intensity change throughout the reaction period.

ACKNOWLEDGMENT

(Hengyan Guo and Mingliang Wang contributed equally to this work.)

REFERENCES

[1] Q. Ma, C. Zhang, M. Zhang, D. Han, and W. Tan, "DNA computing: Principle, construction, and applications in intelligent diagnostics," *Small Struct.*, vol. 2, no. 11, Nov. 2021, Art. no. 2100051.

[2] N. C. Seeman and H. F. Sleiman, "DNA nanotechnology," *Nat. Rev. Mater.*, vol. 3, no. 1, 2017, Art. no. 17068.

[3] C. Zhang, Y. Zhao, X. Xu, R. Xu, H. Li, X. Teng, Y. Du, Y. Miao, H.-C. Lin, and D. Han, "Cancer diagnosis with DNA molecular computation," *Nature Nanotechnol.*, vol. 15, no. 8, pp. 709–715, Aug. 2020.

[4] A. Ebrahimi, H. Ravan, and M. Mehrabani, "Multiplex monitoring of Alzheimer associated miRNAs based on the modular logic circuit operation and doping of catalytic hairpin assembly," *Biosensors Bioelectron.*, vol. 170, Dec. 2020, Art. no. 112710.

[5] K. M. Cherry and L. Qian, "Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks," *Nature*, vol. 559, no. 7714, pp. 370–376, Jul. 2018.

[6] L. Qian, E. Winfree, and J. Bruck, "Neural network computation with DNA strand displacement cascades," *Nature*, vol. 475, no. 7356, pp. 368–372, Jul. 2011.

[7] L. Organick, "Random access in large-scale DNA data storage," *Nature Biotechnol.*, vol. 36, no. 3, pp. 242–248, Mar. 2018.

[8] L. Anavy, I. Vaknin, O. Atar, R. Amit, and Z. Yakhini, "Data storage in DNA with fewer synthesis cycles using composite DNA letters," *Nature Biotechnol.*, vol. 37, no. 10, pp. 1229–1236, Oct. 2019.

[9] D. Fan, J. Wang, E. Wang, and S. Dong, "Propelling DNA computing with materials' power: Recent advancements in innovative DNA logic computing systems and smart bio-applications," *Adv. Sci.*, vol. 7, no. 24, Dec. 2020, Art. no. 2001766.

[10] Y. Zhang, Y. Feng, Y. Liang, J. Yang, and C. Zhang, "Development of synthetic DNA circuit and networks for molecular information processing," *Nanomaterials*, vol. 11, no. 11, p. 2955, Nov. 2021.

[11] Y. Zhao, Y. Liu, X. Zheng, B. Wang, H. Lv, S. Zhou, Q. Zhang, and X. Wei, "Half adder and half subtractor logic gates based on nicking enzymes," *Mol. Syst. Des. Eng.*, vol. 4, no. 6, pp. 1103–1113, 2019.

[12] N. Xie, M. Li, Y. Wang, H. Lv, J. Shi, J. Li, Q. Li, F. Wang, and C. Fan, "Scaling up multi-bit DNA full adder circuits with minimal strand displacement reactions," *J. Amer. Chem. Soc.*, vol. 144, no. 21, pp. 9479–9488, Jun. 2022.

[13] S. Liu, M. Li, X. Yu, C.-Z. Li, and H. Liu, "Biomacromolecular logic gate, encoder/decoder and keypad lock based on DNA damage with electrochemiluminescence and electrochemical signals as outputs," *Chem. Commun.*, vol. 51, no. 67, pp. 13185–13188, 2015.

[14] M. N. Mattath, D. Ghosh, C. Dong, T. Govindaraju, and S. Shi, "Mercury mediated DNA–Au/Ag nanocluster ensembles to generate a gray code encoder for biocomputing," *Mater. Horizons*, vol. 9, no. 8, pp. 2109–2114, 2022.

[15] C. Zhou, H. Geng, and C. Guo, "Design of DNA-based innovative computing system of digital comparison," *Acta Biomaterialia*, vol. 80, pp. 58–65, Oct. 2018.

[16] C. Wu, K. Wang, D. Fan, C. Zhou, Y. Liu, and E. Wang, "Enzyme-free and DNA-based multiplexer and demultiplexer," *Chem. Commun.*, vol. 51, no. 88, pp. 15940–15943, 2015.

[17] C. Zhang, L. Ge, X. Zhang, W. Wei, J. Zhao, Z. Zhang, Z. Wang, and X. You, "A uniform molecular low-density parity check decoder," *ACS Synth. Biol.*, vol. 8, no. 1, pp. 82–90, Jan. 2019.

[18] C. W. Brown, M. R. Lakin, E. K. Horwitz, M. L. Fanning, H. E. West, D. Stefanovic, and S. W. Graves, "Signal propagation in multi-layer DNAzyme cascades using structured chimeric substrates," *Angew. Chem. Int. Ed.*, vol. 53, no. 28, pp. 7183–7187, 2014.

[19] W. Zhong, W. Tang, Y. Tan, J. Fan, Q. Huang, D. Zhou, W. Hong, and Y. Liu, "A DNA arithmetic logic unit for implementing data backtracking operations," *Chem. Commun.*, vol. 55, no. 6, pp. 842–845, 2019.

[20] F. C. Simmel, B. Yurke, and H. R. Singh, "Principles and applications of nucleic acid strand displacement reactions," *Chem. Rev.*, vol. 119, no. 10, pp. 6326–6369, May 2019.

[21] R. Orbach, B. Willner, and I. Willner, "Catalytic nucleic acids (DNAzymes) as functional units for logic gates and computing circuits: From basic principles to practical applications," *Chem. Commun.*, vol. 51, no. 20, pp. 4144–4160, 2015.

[22] C. Zhang, X. Ma, X. Zheng, Y. Ke, K. Chen, D. Liu, Z. Lu, J. Yang, and H. Yan, "Programmable allosteric DNA regulations for molecular networks and nanomachines," *Sci. Adv.*, vol. 8, no. 5, Feb. 2022, Art. no. eabl4589.

[23] X. Zheng, J. Yang, C. Zhou, C. Zhang, Q. Zhang, and X. Wei, "Allosteric DNAzyme-based DNA logic circuit: Operations and dynamic analysis," *Nucleic Acids Res.*, vol. 47, no. 3, pp. 1097–1109, Feb. 2019.

[24] J. Liu, S. Liu, C. Zou, S. Xu, and C. Zhou, "Research progress in construction and application of enzyme-based DNA logic gates," *IEEE Trans. Nanobiosci.*, vol. 22, no. 2, pp. 245–258, Apr. 2023.

[25] D. Woods, D. Doty, C. Myhrvold, J. Hui, F. Zhou, P. Yin, and E. Winfree, "Diverse and robust molecular algorithms using reprogrammable DNA self-assembly," *Nature*, vol. 567, no. 7748, pp. 366–372, Mar. 2019.

[26] A. Whitaker and B. Freudenthal, "Transcriptional regulation via strand displacement DNA repair in G-quadruplexes," *FASEB J.*, vol. 36, no. S1, May 2022, doi: 10.1096/fasebj.2022.36.s1.0r292.

[27] D.-D. Wang, J. Zhang, Q.-Q. Yu, K. Zhang, T.-T. Chen, and X. Chu, "Biomineralized zeolitic imidazolate Framework-8 nanoparticles enable polymerase-driven DNA biocomputing for reliable cell identification," *Anal. Chem.*, vol. 94, no. 11, pp. 4794–4802, Mar. 2022.

[28] J. Yang, K. Chen, X. Li, and C. Zhang, "A molecular computing model using 3D dumbbell DNA origami," in *Proc. BIBE ; Int. Conf. Biol. Inf. Biomed. Eng.*, Shanghai, China, Jun. 2018, pp. 1–4.

[29] S. A. Salehi, X. Liu, M. D. Riedel, and K. K. Parhi, "Computing mathematical functions using DNA via fractional coding," *Sci. Rep.*, vol. 8, no. 1, p. 8312, May 2018.

[30] L. Qian and E. Winfree, "Scaling up digital circuit computation with DNA strand displacement cascades," *Science*, vol. 332, no. 6034, pp. 1196–1201, Jun. 2011.

[31] T. Song, A. Eshra, S. Shah, H. Bui, D. Fu, M. Yang, R. Mokhtar, and J. Reif, "Fast and compact DNA logic circuits based on single-stranded gates using strand-displacing polymerase," *Nature Nanotechnol.*, vol. 14, no. 11, pp. 1075–1081, Nov. 2019.

[32] C. Zhou, H. Geng, P. Wang, and C. Guo, "Ten-input cube root logic computation with rational designed DNA nanoswitches coupled with DNA strand displacement process," *ACS Appl. Mater. Interface*, vol. 12, no. 2, pp. 2601–2606, Jan. 2020.

[33] S. Okumura, G. Gines, N. Lobato-Dauzier, A. Baccouche, R. Deteix, T. Fujii, Y. Rondelez, and A. J. Genot, "Nonlinear decision-making with enzymatic neural networks," *Nature*, vol. 610, no. 7932, pp. 496–501, Oct. 2022.

[34] J. S. Oliver, "Matrix multiplication with DNA," *J. Mol. Evol.*, vol. 45, no. 2, pp. 161–167, Aug. 1997.

[35] X. Chen, "Expanding the rule set of DNA circuitry with associative toehold activation," *J. Amer. Chem. Soc.*, vol. 134, no. 1, pp. 263–271, Jan. 2012.

[36] A. J. Genot, J. Bath, and A. J. Turberfield, "Combinatorial displacement of DNA strands: Application to matrix multiplication and weighted sums," *Angew. Chem.*, vol. 125, no. 4, pp. 1227–1230, Jan. 2013.

[37] S. Xu, Y. Liu, S. Zhou, Q. Zhang, and N. K. Kasabov, "DNA matrix operation based on the mechanism of the DNAzyme binding to auxiliary strands to cleave the substrate," *Biomolecules*, vol. 11, no. 12, p. 1797, Nov. 2021.

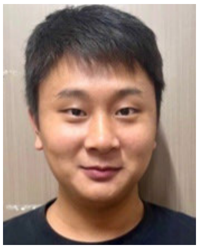
[38] X. Liu, Q. Zhang, X. Zhang, Y. Liu, Y. Yao, and N. Kasabov, "Construction of multiple logic circuits based on allosteric DNAzymes," *Biomolecules*, vol. 12, no. 4, p. 495, Mar. 2022.

[39] Y. Hu, C. Li, M. Hu, Z. Zhang, R. Fu, X. Tang, and T. Wu, ''Allosteric nucleic acid enzyme: A versatile stimuli-responsive tool for molecular computing and biosensing nanodevices,'' *Small*, vol. 19, no. 27, Jul. 2023, Art. no. 2300207.

[40] D. Y. Zhang and E. Winfree, ''Dynamic allosteric control of noncovalent DNA catalysis reactions,'' *J. Amer. Chem. Soc.*, vol. 130, no. 42, pp. 13921–13926, Oct. 2008.

[41] W. Lai, L. Ren, Q. Tang, X. Qu, J. Li, L. Wang, L. Li, C. Fan, and H. Pei, ''Programming chemical reaction networks using intramolecular conformational motions of DNA,'' *ACS Nano*, vol. 12, no. 7, pp. 7093–7099, Jul. 2018.

[42] Y. Liang, Y. Qie, J. Yang, R. Wu, S. Cui, Y. Zhao, G. J. Anderson, G. Nie, S. Li, and C. Zhang, ''Programming conformational cooperativity to regulate allosteric protein-oligonucleotide signal transduction,'' *Nature Commun.*, vol. 14, no. 1, p. 4898, Aug. 2023.



HENGYAN GUO received the B.E. degree in computer science and technology from Jilin Agricultural Science and Technology University, in 2021. He is currently pursuing the master's degree in computer technology with Shenyang Aerospace University. His research interests include DNA strand displacement reactions and DNA-based numerical computation.

MINGLIANG WANG was born in Baishan, Jilin, China, in 1990. He received the B.S. degree in mathematics from Jilin University of Finance and Economics and the M.E. degree in applied mathematics from Arizona State University, in 2022. His research interests include biocomputing, deep learning, and numerical computation.

XUEDONG ZHENG received the Ph.D. degree from Huazhong University of Science and Technology, Wuhan, China, in 2010. He is currently a Professor with Shenyang Aerospace University. His research interests include DNA computing, DNA nanotechnology, and nonlinear dynamics.