A Loser-Take-All DNA Circuit
Kellen R. Rodriguez,∥ Namita Sarraf,∥ and Lulu Qian*

ABSTRACT: DNA-based neural networks are a type of DNA circuit capable of molecular pattern recognition tasks. Winner-take-all DNA networks have been developed to scale up the complexity of molecular pattern recognition with a simple molecular implementation. This simplicity was achieved by replacing negative weights in individual neurons with lateral inhibition and competition across neurons, eliminating the need for dual-rail representation. Here we introduce a new type of DNA circuit that is called loser-take-all: an output signal is ON if and only if the corresponding input has the smallest analog value among all inputs. We develop a DNA strand-displacement implementation of loser-take-all circuits that is cascadable without dual-rail representation, maintaining the simplicity desired for scalability. We characterize the impact of effective signal concentrations and reaction rates on the circuit performance, and derive solutions for compensating undesired signal loss and rate differences. Using these approaches, we successfully demonstrate a three-input loser-take-all circuit with nine unique input combinations. Complementary to winner-take-all, loser-take-all DNA circuits could be used for recognition of molecular patterns based on their least similarities to a set of memories, allowing classification decisions for patterns that are extremely noisy. Moreover, the design principle of loser-take-all could be more generally applied in other DNA circuit implementations including k-winner-take-all.

KEYWORDS: DNA strand displacement, DNA neural network, molecular pattern recognition, winner-take-all, loser-take-all, signal reversal

RESULTS AND DISCUSSION
A Loser-Take-All Function. In this work, we introduce a new DNA circuit architecture that is called loser-take-all. It is closely related to winner-take-all, but computes an inverse function:

Received: July 10, 2021
Published: October 8, 2021
where $x_i \in \mathbb{R}^+, i, j \in \{1, 2, ..., n\}$ for a circuit with $n$ inputs. An output signal is ON if and only if the corresponding input has the smallest analog value among all inputs. This function has been explored in electrical circuits.\(^3\) We show that the implementation of a loser-take-all (LTA) neural network expands the functionality of DNA-based neural networks while maintaining the simplicity desired for scalability.

Loser-take-all (LTA) can be seen as complementary to winner-take-all (WTA). For example, even in theory the classification accuracy of a WTA neural network using averaged training patterns as weights can be fairly low for certain classes of patterns when the similarity between distinct classes is high (Figure 1a, left). In these cases, it will be useful to identify the original input signal with the smallest analog value. However, implementation of negative weights requires dual-rail representation where a pair of species is used to indicate the positive or negative values of a weight ($w_i^+ = w_i$ when $w_i > 0$ and $w_i^- = -w_i$ when $w_i < 0$), doubling the circuit size.\(^7\) A few other issues arise when applying the dual-rail technique for implementing a loser-take-all neural network, which we will discuss later. Simpler implementations could be achieved by using an annihilator where positive and negative weights result in two distinct weighted sum species (e.g., $s^+ = \sum_{w_i > 0} w_i x_i$ and $s^- = \sum_{w_i < 0} w_i x_i$) that stoichiometrically consume each other when reacting with the annihilator.\(^10,14\)

However, this approach does not allow the weighted sum function to be cascaded with the winner-take-all function—when one weighted sum species is present before the other it will react with downstream circuit components before annihilation takes place. For implementing $1 - x_i$ it is not a variable and should be present before $x_i$ arrives, which would be consumed by the downstream winner-take-all layer and result in false output.

DNA Strand-Displacement Implementation. Here we show a loser-take-all implementation that is both cascadable and requires no dual-rail representation (Figure 1b). First, reversal of each input signal is accomplished by computing the average of all input signals excluding itself:

$$y_i = \frac{\sum_{j \neq i} x_j}{n-1}$$  \hspace{1cm} (2)

where $x_i \in \mathbb{R}^+, i, j \in \{1, 2, ..., n\}$ for a circuit with $n$ inputs. It is clear that the largest $y_i$ corresponds to the smallest $x_i$: suppose $x_1 < x_2 < \cdots < x_n$, then $c - x_1 > c - x_2 > \cdots > c - x_n$ where $c = \sum x_i$. Next, the output of the circuit is computed as a winner-take-all function of the reversed input signals:

$$z_i = \begin{cases} 
1 & \text{if } y_j > y_i \ \forall j \neq i \\
0 & \text{otherwise}
\end{cases}$$  \hspace{1cm} (3)

DNA strand-displacement implementation of the winner-take-all function has been previously developed.\(^3\) It involves pairwise annihilation that facilitates competition between all input strands until there is only one winner left, and subsequent signal restoration that recovers the concentration of the winner species through a catalytic reaction that utilizes a gate and fuel species. Sequential operation of these two types of reactions was approximated by controlling the rate of annihilation ($k_j$) to be much faster than that of signal restoration ($k_i$) via a longer toehold domain ($s^*$ $T^*$ on $A_{n-i}$ and $T^*$ on $G_{n-i}$ as shown in Figure 2). Here we develop a DNA strand-displacement implementation of signal reversal that is composable with the winner-take-all implementation (Figure 2). In a three-input loser-take-all circuit, each input strand (e.g., $X_1$) irreversibly reacts with one of two signal reversal gates (e.g., $GY_{12}$ and $GY_{13}$) to produce one of two output strands (e.g., $Y_2$ and $Y_3$). Output strands that contain the same toehold and branch migration domains for reacting with an annihilator (e.g., output strands from $GY_{12}$ and $GY_{13}$ both contain domains $s$ $T$ $Sy_2$) collectively represent a reversed signal (e.g., $Y_2$). When signal reversal gates are in excess, the concentration of a reversed signal at reaction completion is
expected to be the average of all but one initial input concentrations (e.g., \( Y_2 = (X_1|_{t=0} + X_3|_{t=0})/2 \)) if it is not consumed by any downstream reactions.

**Demonstration of Signal Reversal.** We first demonstrated signal reversal by connecting the signal reversal gates directly to reporters (i.e., replacing each SYi domain with SZi).

To compute the average of certain inputs, it is important that each input strand reacts with all signal reversal gates at the same rate. As the effective rate constant of an irreversible strand-displacement reaction mainly depends on the toehold length and sequence,

\[ k_s \ll k_r \]

the same toehold sequence was employed in all signal reversal gates. To evaluate how well input signals can evenly split to produce output signals, we tested each input strand with a pair of signal reversal gates. No more than 10% difference was observed between each pair of output signals (Figure 3a). Considering stoichiometry inaccuracy and experimental noise, this difference is unsurprising. However, despite having almost no difference between the two outputs produced by input \( X_1 \), their concentrations were approximately 40% lower than expected. Given that the data was normalized based on the fluorescence level of \( 1 \times \) standard concentration, this could be due to the effective concentration of the input strand being lower than that of the output strand, which is not uncommon with unpurified DNA strands.

Nonetheless, when all three input strands and six signal reversal gates were mixed together, the smallest input signal resulted in the largest output signal for two unique combinations of input signals that we tested (Figure 3b), suggesting a successful demonstration of signal reversal.

**Rate Measurements for Evaluating the Fairness of Competition among Reversed Signals.** Equal reaction rates are essential not only for signal reversal but also for winner-take-all, as they ensure fair competition among distinct signal species. To promote equal reaction rates, the toeholds on all annihilators were designed to have the same sequence. As discussed above, the toeholds on annihilators need to be longer than that on signal restoration gates so as to approximate sequential operation, and thus a common \( s^* \) domain was introduced in all input signals to allow for a common extended toehold \( (s^* T^*) \) on all annihilators (Figure 2). To investigate the impact of branch migration sequence on strand displacement rate, we measured the rate of signal...
restoration (Figure 4a)—signal restoration gates share the same branch migration domains (Syi) as the annihilators and fuels.

Figure 4. Rate measurements in signal restoration (a) without and (b) with the presence of signal reversal gates and annihilators. Abstract reaction diagrams indicate the reactions involved in each experiment. Simulation and fluorescence kinetics data are shown as solid and dotted trajectories, respectively. Standard concentration 1X = 50 nM. Initial concentrations of all signal reversal gates, annihilators, signal restoration gates, fuels, and reporters were 2X, 4X, 1X, 2X, and 2X, respectively.

their outputs are directly measurable by reporters. We estimated a 2.4-fold difference in strand displacement rate constant $k_s$ (specified in eq S3) across three signal restoration pathways. As expected, signal restoration slowed when signal reversal gates and annihilators were present (Figure 4b). In particular, signal strands $Y_i$ were anticipated to react with the annihilators, which allowed us to estimate toehold dissociation rate constant $k_r$ (specified in eq S2) across three annihilation reaction pathways. Like $k_s$, a 2-fold difference was estimated for $k_r$. We suspect this is due to spurious interactions that temporarily inhibit the toeholds (also known as toehold occlusion) having different impacts on annihilators with different branch migration sequences.

Concentration Adjustments for Compensating Rate Differences and Improving ON–OFF Separations.

Applying the estimated $k_s$ and $k_r$ in simulations, we predicted that the overall behavior of the loser-take-all circuit would bias toward identifying input $X_2$ as the smallest signal; this was indeed shown in experiments where output $Z_2$ turned ON the fastest when the concentration of $X_2$ was 0 (Figure 5a). It would be possible to reduce the difference in reaction rates by carefully redesigning the DNA sequences guided by sequence-level kinetics simulations, but it would be challenging given the complexity of the circuit. We thus chose to explore the possibility of exploiting concentration adjustments to compensate for the rate differences. We hypothesized and verified by simulations that reducing the concentration of annihilator $Anh_{13}$ to half would reduce the competition between reversed signals $Y_i$ and $Y_3$ while simultaneously promoting both of them to compete with $Y_5$—this introduced bias would negate the observed bias in producing output $Z_2$. This hypothesis was supported by experimental observations: with the adjustment in annihilator concentration, similar kinetics were achieved in all three outputs that turned ON when the corresponding input strand had the lowest concentration (Figure 5b). In general, when the production of a particular output $Z_i$ is faster than the others, reducing the concentrations of annihilators $Anh_{ij}$, $\forall j$, $k \neq i$ would help balance the rate bias. Alternatively, more experiments with distinct Syi domains could be performed to allow the selection of sequences with similar $k_s$ and $k_r$.

Considering the lower effective concentration of input strands indicated by experiments on the signal reversal layer of the circuit (Figure 3a), we further hypothesized and verified by simulations that doubling the concentration of inputs would speed up the circuit while resulting in a better separation between outputs that are supposed to turn ON and those that are supposed to stay OFF. The second hypothesis was also supported by experiments: with the additional adjustment in

![Figure 4](https://doi.org/10.1021/acssynbio.1c00318)
![Figure 5](https://doi.org/10.1021/acssynbio.1c00318)
input concentrations, larger gaps between ON and OFF outputs were achieved within 2 h (Figure 5c).

Demonstration of Loser-Take-All. With the above adjustments in annihilator and input concentrations, we demonstrated the three-input loser-take-all function with nine unique input combinations (Figure 6). If all rate constants in each of the circuit layers (signal reversal, pairwise annihilation, and signal restoration) were equal, input combinations with the same values but different orders (shown in each row of Figure 6) would lead to the same kinetics in output trajectories but different identities. Because the rate constants were not equal and the concentration adjustments could not fully account for the differences, the ON−OFF separation in outputs was better in some cases than others. For example, with the same smallest input, swapping the identities of the two larger inputs could result in a slightly better performance (Figure 5c, rightmost plot vs Figure 6, rightmost plot in the first row). Regardless of the quantitative variations, the loser-take-all computation was qualitatively correct for the nine example input cases—output corresponding to the smallest input reached at least 60% reaction completion within 2 h while the other outputs remained below 40% (Figure 6).

Robustness of Loser-Take-All Compared to Winner-Take-All DNA Circuits. Admittedly, the loser-take-all behavior was not as robust as the previously demonstrated winner-take-all behavior. A main reason is that the difference between any reversed signals is reduced to \(1/(n-1)\) of that between the original inputs in a loser-take-all circuit with \(n\) inputs. As shown in the bar chart in Figure 6, a 50% reduction is expected in the three-input circuit, making the competition more challenging. Besides that, we investigated another possible reason: in winner-take-all circuits, all signal strands had exactly one toehold domain; in loser-take-all circuits, signal reversal was designed to be irreversible, which requires two toehold domains in input strands. We suspected that these input strands with two toeholds could have increased spurious interactions, particularly with the annihilators. For example, if single-stranded DNA is sufficiently stretchable, it is conceivable that an input strand could bind to an annihilator by both toeholds, which would lead to slower dissociation rate and thus more significant toehold occlusion compared to a single toehold. For this reason, we explored two alternative designs. In the first we removed one toehold from the input strands, allowing signal reversal reactions to be reversible (Figure S2a). Interestingly, the performance of the circuit became worse (Figure S2b), suggesting that the common s domain needed for reacting with the extended toeholds on annihilators introduced significant crosstalk among signal reversal reactions in the reverse direction. In the second design we changed the 3’ end toehold on input strands to a different sequence, which is now distinct from the toeholds on annihilators (Figure S3a). This design did result in mild improvement in circuit performance (Figure S3b), indicating reduced effect of toehold occlusion.

Simulation Analysis of a 100-bit, Three-Memory Loser-Take-All Neural Network. Finally, we evaluated the performance of a loser-take-all neural network in simulation for processing three similar classes of 100-bit patterns: MNIST digits 1, 4, and 8. Analysis of all test patterns in their weighted sum space suggested that 20% more patterns can be correctly classified by the loser-take-all neural network (Figure S5) than the winner-take-all neural network (Figure S4). However, due to the reduced differences among the weighted sums after their strengths are reversed, the percentage of patterns that are experimentally feasible (defined by a 15% margin between two closest weighted sums) is only mildly larger in the loser-take-all neural network. The behavior of the network was simulated with four example patterns per class among the experimentally feasible ones (Figure S6). Examples of digits 1 and 8 were correctly identified to be least similar to 4; examples of digit 4 resulted in ambiguous classification where the separation between the two fastest output trajectories was small, indicating that the input pattern was similarly different from 1 and 8. These simulation results suggest that in principle loser-take-all neural networks are useful for processing classes of patterns that are too similar to be recognized by winner-take-all neural networks, and that alternative strategies for implementing signal reversal that maintain or increase the difference among distinct weighted sums of input will improve the performance of the neural network.

Complexity Analysis of Three Distinct Implementations for Loser-Take-All Neural Networks. In the above simulations, the loser-take-all neural network was implemented by combining signal reversal with summation (Figures S6a and S7a), which resulted in \(n^2 - 2n\) more distinct species compared to the implementation of winner-take-all neural networks, where \(n\) is the number of memories (i.e., classes of patterns). Alternatively, the signal reversal function can be combined
with weight multiplication, resulting in the same number of distinct species compared to winner-take-all neural networks for arbitrary memories and input patterns (Figure S7b, upper bound). However, for specific classes of patterns with many zeros, this implementation is not efficient, as it always requires \( m \times n \) weight species regardless of the patterns, where \( m \) is the number of all bits in the patterns (Figure S7b, lower bound).

For processing a few classes of complex patterns (e.g., \( m \geq n^2 \)), the strategy of signal reversal generally leads to simpler implementations than the dual-rail technique (Figure S7c). Moreover, there are two issues with the dual-rail implementation of winner-take-all or loser-take-all neural networks. First, only half of the dual-rail circuit producing \( z_i^{ON} \) can be implemented while the other half producing \( z_i^{OFF} \) cannot, as it would require a function other than winner-take-all. Second, an “always ON” signal used for generating a set of constants needed for the dual-rail implementation must arrive at the same time as the input signals, otherwise false output will be produced (i.e., output corresponding to the largest constant will turn ON), which cannot be reversed after the input signals have arrived. For this reason, the dual-rail implementation is only correct if autonomous operation of the neural network is not required (e.g., for a diagnostic task for which human intervention is allowed but not for a therapeutic task for which the circuit must respond to changes in a molecular environment that are unknown to humans).

**Future Work.** It is worthwhile to mention that other than the rate constants estimated from experiments that were designed to quantify the difference in distinct signal restoration and annihilation reaction pathways, we applied the same rate constants from previous work\(^{10,11} \) to all simulations. In some cases, the simulation did not fully explain the experimental data, which was especially true for outputs that were supposed to stay OFF. Because these outputs were mainly produced by signals that bypassed the annihilators and became amplified by signal restoration gates and fuels, this observation calls for signal restoration gates and fuels, this observation calls for extending to keep consistent binding energies.

More importantly, our results suggest two criteria for improving the robustness of DNA circuits: an ideal circuit design should tolerate at least 2-fold variation in any reaction rates and at least 40% variation in any signal concentrations. In theory, the computational power of rate-independent chemical reaction networks has been explored, where correct output is guaranteed for any reaction rates.\(^{23,24} \) In experimental work, robust DNA circuits have been developed where both reaction rate and signal concentration requirements allow a wide range.\(^{17,18} \) For example, in seesaw logic circuits, correct computation can be achieved so long as one type of reaction rate is at least 10 times larger than the other and all signals representing logic OFF and ON are within 0–0.3x and 0.7–1x, respectively. By contrast, no DNA neural networks demonstrated thus far have similar tolerance in reaction rates and concentrations. Hopefully our work will motivate future research in DNA neural networks with ever-increasing robustness, for example by considering alternative architectures that utilize binary weights.\(^{22} \)

**METHODS**

**Sequence Design.** Every strand of DNA in the loser-take-all system consisted of two types of functionally independent components: short toehold domains and long branch-migration domains. All domain sequences were drawn from a set of sequences that had been designed in line with heuristics that had already been validated experimentally.\(^{17,18} \) The heuristics are summarized as follows: every domain utilized a code of A, T, and C in order to minimize undesired interactions between strands and secondary structure. To minimize synthesis errors, the domain sequences were limited to at most four consecutive A’s or T’s and at most three consecutive C’s. The toehold domains are five nucleotides long and universally used in all strands. The branch-migration domains are 15 nucleotides long, and the sequences had a range of C-content of 30–70% in order to ensure similar melting temperatures between the double-stranded complexes. Lastly, the sequences had a minimum of 30% difference in nucleotides between all pairs of branch-migration domains and no pair could share a matching sequence longer than 35% of the length of those domains to prevent spurious branch-migrations from reaching completion.

Same as the previously designed winner-take-all circuits,\(^9 \) the toehold was extended by two nucleotides in the annihilator molecules in order to increase the binding energies and effective rates of strand displacement for the annihilation reactions. This was done to ensure that the annihilation reaction was faster than signal restoration in order to reduce the amplification of the signals in the winner-take-all reaction before all but the winner species were annihilated. To prevent bias and allow fair competition in the pairwise annihilation, the annihilator molecules had the same 2-nucleotide toehold extension to keep consistent binding energies.

The DNA sequences for all molecules in the loser-take-all system were analyzed with NUPACK\(^{26} \) in order to verify that the desired structures would form and additionally to check that no spurious structures would form.

**Sample Preparation.** DNA oligonucleotides were purchased from Integrated DNA Technologies (IDT). Reporter strands with fluorophores and quenchers were ordered with high-performance liquid chromatography (HPLC) purification, while gate, fuel, annihilator, and input strands were ordered unpurified (standard desalting). Strands were shipped with formulation service LabReady (100 \( \mu \)M in IDTE buffer at pH 8.0). They were stored at 4.0 °C.

**CONCLUSIONS**

By introducing the concept of loser-take-all DNA circuits and experimentally demonstrating a three-input loser-take-all function, we have advanced the architecture of DNA-based neural networks. Like the previously developed winner-take-all DNA circuits, only two-stranded motifs are used here and no dual-rail representation is required, both of which are desired properties for simplicity and scalability. In contrast to winner-take-all, loser-take-all allows for recognition of molecular patterns based on their least similarities to a set of memories, opening up possibilities for analyzing and responding to highly noisy patterns that cannot be correctly recognized by winner-take-all DNA neural networks. Furthermore, the design principle of loser-take-all DNA circuits could be more generally applied to other implementations including k-winner-take-all (k-WTA). For example, a three-input 2-WTA function could be implemented by introducing fan-out in the signal restoration layer of the loser-take-all circuit. This k-WTA implementation is potentially more feasible for experimental demonstration than previous proposals\(^4,5 \) and could enable more sophisticated pattern classification tasks traditionally performed by multilayer linear threshold circuits.\(^11 \)
Annihilator and gate complexes were annealed at 45 μM with a 1:1 ratio of top and bottom strands. Reporters were annealed at 20 μM with a 1:2:1 ratio of top and bottom strands. All complexes were annealed in TE buffer with 12.5 mM MgCl₂. Annealing took place in a thermocycler (Eppendorf). Samples were heated to 90 °C for 5 min and then cooled to 20 °C at a rate of 0.1 °C per 6 s.

As the excess top strands of the reporter complexes do not interfere with designed molecular interactions in the circuit, reporter complexes do not need to be purified. However, an excess of either the top or bottom strands of the annihilator or gate complexes would affect the circuit behavior. Therefore, annihilator and gate complexes were purified using 12% polyacrylamide gel electrophoresis (PAGE). The gels were run at 150 V for roughly 6 h. Bands containing the complexes were cut out from the gel, diced into smaller pieces, and incubated at 150 V for roughly 6 h. Bands containing the complexes were then collected. The absorbance of each collected sample at 260 nm was measured using a NanoDrop (Thermo Fisher). Along with the extinction coefficients of the complexes, these data were used to calculate the concentration of each complex.

**Fluorescence Kinetics Experiments.** Fluorescent data was collected on a microplate reader (Synergy H1, Biotek). A 96-well plate (Corning) was used for experiments, with 110 μL of reaction mixture per well. The standard concentration for the experiments was 50 nM. Excitation/emission wavelengths were 496 nm/525 nm for fluorophore ATTO488, 555 nm/582 nm for fluorophore ATTO550, and 598 nm/629 nm for fluorophore ATTO590. Readings were taken every 2 min for the duration of the experiment.

**Data Normalization.** The raw fluorescent data from the fluorescent kinetic experiments were normalized to relative concentrations of the respective output signal using internal controls for each set of experiments. These controls consisted of a negative control sample with no input where each output signal was in a minimum OFF state as well as three positive control samples each with one of the reversed signals Y₁ to Y₃ at 50 nM (Figures 3, 4, S4, and S5) or 100 nM (Figures S6 and S7) where each output signal was in a maximal ON state. The controls were conducted in parallel with the corresponding set of experiments that used the same circuit components, but varying input signals. The negative control was used to set the baseline 0× relative concentration by averaging the first five data points. Similarly, the last five data points of the positive control samples at the point where the trajectories had reached a plateau were averaged to set the 1× relative concentration.

**ASSOCIATED CONTENT**

1. **Supporting Information**

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acssynbio.1c00318.

Supporting notes on modeling and simulation, sequence-level design diagrams, alternative designs and experimental data for reducing toehold occlusion, simulation and complexity analysis on implementations of loser-take-all neural networks, and tables of DNA sequences (PDF)

**REFERENCES**

1. Benenson, Y. Biomolecular computing systems: principles, progress and potential. Nat. Rev. Genet. 2012, 13, 455–468.
2. Zhang, D. Y.; Seelig, G. Dynamic DNA nanotechnology using strand-displacement reactions. Nat. Chem. 2011, 3, 103–113.
3. Scalise, D.; Schulman, R. Controlling matter at the molecular scale with DNA circuits. Annu. Rev. Biomed. Eng. 2019, 21, 469–493.
4. Kim, J.; Hopfield, J. J.; Winfree, E. Neural network computation by in vitro transcriptional circuits. Adv. Neural Inform. Proc. Syst. 2004, 17, 681–688.
5. Genot, A. J.; Fuji, T.; Rondelez, Y. Scaling down DNA circuits with competitive neural networks. J. R. Soc., Interface 2013, 10, 20130212.
6. Lakin, M. R.; Stefanovic, D. Supervised learning in adaptive DNA strand displacement networks. ACS Synth. Biol. 2016, 5, 885–897.
7. Qian, L.; Winfree, E.; Bruck, J. Neural network computation with DNA strand displacement cascades. Nature 2011, 475, 368–372.
8. Chen, S. X.; Seelig, G. A DNA neural network constructed from molecular variable gain amplifiers. International Conference on DNA-Based Computers. 2017, 10467, 110–121.
9. Cherry, K. M.; Qian, L. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. Nature 2018, 559, 370–376.
10. Lopez, R.; Wang, R.; Seelig, G. A molecular multi-gene classifier for disease diagnostics. Nat. Chem. 2018, 10, 746–754.
11. Maass, W. On the computational power of winner-take-all. Neural Computation 2000, 12, 2519–2535.
(12) LeCun, Y.; Cortes, C.; Burges, C. J. MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/index.html (accessed 2021-10-07).
(13) Patel, G. N.; DeWeerth, S. P. An analog VLSI loser-take-all circuit. Proceedings of ISCAS’95-International Symposium on Circuits and Systems 1995, 850–853.
(14) Zhang, D. Y.; Seelig, G. DNA-based fixed gain amplifiers and linear classifier circuits. International Workshop on DNA-Based Computers 2011, 6518, 176–186.
(15) Yurke, B.; Mills, A. P. Using DNA to power nanostructures. Genetic Programming and Evolvable Machines 2003, 4, 111–122.
(16) Zhang, D. Y.; Winfree, E. Control of DNA strand displacement kinetics using toehold exchange. J. Am. Chem. Soc. 2009, 131, 17303–17314.
(17) Thubagere, A. J.; Thachuk, C.; Berleant, J.; Johnson, R. F.; Ardelean, D. A.; Cherry, K. M.; Qian, L. Compiler-aided systematic construction of large-scale DNA strand displacement circuits using unpurified components. Nat. Commun. 2017, 8, 1–12.
(18) Qian, L.; Winfree, E. Scaling up digital circuit computation with DNA strand displacement cascades. Science 2011, 332, 1196–1201.
(19) Srinivas, N.; Parkin, J.; Seelig, G.; Winfree, E.; Soloveichik, D. Enzyme-free nucleic acid dynamical systems. Science 2017, 358, eaal2052.
(20) Schaeffer, J. M.; Thachuk, C.; Winfree, E. Stochastic simulation of the kinetics of multiple interacting nucleic acid strands. International Workshop on DNA-Based Computers. 2015, 9211, 194–211.
(21) Berleant, J.; Berlind, C.; Badelt, S.; Dannenberg, F.; Schaeffer, J.; Winfree, E. Automated sequence-level analysis of kinetics and thermodynamics for domain-level DNA strand-displacement systems. J. R. Soc., Interface 2018, 15, 20180107.
(22) Zhang, D. Y. Cooperative hybridization of oligonucleotides. J. Am. Chem. Soc. 2011, 133, 1077–1086.
(23) Chen, H.-L.; Doty, D.; Soloveichik, D. Rate-independent computation in continuous chemical reaction networks. Proceedings of the 5th Conference on Innovations in Theoretical Computer Science; Association for Computing Machinery: 2014; pp 313–326.
(24) Doty, D.; Rogers, T. A.; Soloveichik, D.; Thachuk, C.; Woods, D. Thermodynamic binding networks. International Conference on DNA-Based Computers. 2017, 10467, 249–266.
(25) Vasic, M.; Chalk, C.; Khurshid, S.; Soloveichik, D. Deep molecular programming: a natural implementation of binary-weight ReLU neural networks. International Conference on Machine Learning, July 13–18, 2020; pp 9701–9711.
(26) Zadeh, J. N.; Steenberg, C. D.; Bois, J. S.; Wolfe, B. R.; Pierce, M. B.; Khan, A. R.; Dirks, R. M.; Pierce, N. A. NUPACK: analysis and design of nucleic acid systems. J. Comput. Chem. 2011, 32, 170–173.