frontiers in Genetics

ORIGINAL RESEARCH
published: 17 May 2021
doi: 10.3389/fgene.2021.644484

Stable DNA Sequence Over Close-Ending and Pairing Sequences Constraint

Xue Li[1†], Ziqi Wei[2†], Bin Wang[1] and Tao Song[3*]*

[1] The Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian, China, [2] School of Software, Tsinghua University, Beijing, China, [3] College of Computer and Communication Engineering, China University of Petroleum, Qingdao, China

DNA computing is a new method based on molecular biotechnology to solve complex problems. The design of DNA sequences is a multi-objective optimization problem in DNA computing, whose objective is to obtain optimized sequences that satisfy multiple constraints to improve the quality of the sequences. However, the previous optimized DNA sequences reacted with each other, which reduced the number of DNA sequences that could be used for molecular hybridization in the solution and thus reduced the accuracy of DNA computing. In addition, a DNA sequence and its complement follow the principle of complementary pairing, and the sequence of base GC at both ends is more stable. To optimize the above problems, the constraints of Pairing Sequences Constraint (PSC) and Close-ending along with the Improved Chaos Whale (ICW) optimization algorithm were proposed to construct a DNA sequence set that satisfies the combination of constraints. The ICW optimization algorithm is added to a new predator–prey strategy and sine and cosine functions under the action of chaos. Compared with other algorithms, among the 23 benchmark functions, the new algorithm obtained the minimum value for one-third of the functions and two-thirds of the current minimum value. The DNA sequences satisfying the constraint combination obtained the minimum of fitness values and had stable and usable structures.

Keywords: DNA computing, DNA sequence design, constraint, WOA, ICW

OPEN ACCESS

Edited by:
Pan Zheng,
University of Canterbury, New Zealand

Reviewed by:
Effirul Ikhwan Ramlan,
Ulster University, United Kingdom
Leyi Wei,
Shandong University, China

***Correspondence:**
Bin Wang
wangbinpaper@gmail.com
Tao Song
tsong@upc.edu.cn

[†] These authors have contributed equally to this work

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

Received: *21 December 2020*
Accepted: *12 April 2021*
Published: *17 May 2021*

Citation:
Li X, Wei Z, Wang B and Song T (2021) Stable DNA Sequence Over Close-Ending and Pairing Sequences Constraint. Front. Genet. 12:644484. doi: 10.3389/fgene.2021.644484

INTRODUCTION

DNA computing is a new and promising interdisciplinary subject based on computational science and molecular biology, which shows great potential in solving NP problems (Wang et al., 2019; Zhu et al., 2020). At the end of the 20th century, Adleman (1994) used DNA molecules for calculation and solved the Hamiltonian problem (Heidari, 2014). The successful solution of this problem led DNA computing to become a field of great development. It has since been widely used to solve problems in many domains, including PCR amplification (Sze and Schloss, 2019), DNA sequencing (Shendure et al., 2017), bioinformatics (Zou and Liu, 2019), prediction of disease genes (Zeng et al., 2020a,b), image encryption (Zhou et al., 2020), and DNA data storage (Zhang et al., 2019; Cao et al., 2021), among others.

Benenson et al. (2004) made decisions through simple Boolean logic and successfully used RNA interference to construct molecular computing cores in human kidney cells. Yaakov et al. (Rinaudo et al., 2007) announced a breakthrough DNA computer, which can theoretically release anticancer drugs into cancer cells. In 2017, researchers used the CRISPR-Cas system (Shipman et al., 2017) to encode the pixel values of black-and-white images and short films into the genome of living bacterial populations; they minimized the technical limitations of the information storage system. Thubagere et al. (2017) developed a DNA robot that can control DNA to perform specific actions, such as picking and sorting goods in solution. Han et al. (2017) developed a new strategy called single-stranded origami (ssOrigami), which uses a single-stranded DNA or RNA as long as thin as a noodle to implement a self-folding structure without a topological junction, which could allow drugs to travel directly to the site of injury within the cell. Li et al. (2018) developed a nanorobot based on DNA origami technology that can be used to carry thrombin to accurately target tumor cells, and more broadly, this technology can be used for many types of cancer. Cherry and Qian (2018) used the extended seesaw motif DNA neural network for pattern discrimination and constructed a neural network using DNA sequence to realize the recognition of handwritten digits in model organisms. Palluk et al. (2018) proposed a strategy for the synthesis of oligonucleotides using a template-independent polymerase terminal deoxyribonucleotide transferase and obtained a scheme to repeatedly write a definite sequence. Schickinger et al. (2018) proposed DNA origami for creating a tethered multifluorophore movement experiment and explain interactions between cells. This is easy to use and has a wide range of applications. In order to avoid errors in DNA storage, Deng et al. (2019) adapted a hybrid coding system, which is composed of improved variable length run-length limited (vl-rll) codes and optimized photograph low-density parity check codes (LDPCs). Wang et al. (2020) proposed a deep learning framework, SeqEnhDL, to classify cell type–specific enhancers based on sequence features. This framework can transform folding changes of any DNA sequence into deep learning model features. Zhang et al. (2020) constructed a DNA molecular lock by using the characteristics of enzyme mutual inhibition and realized the information protection function at the molecular level.

In addition, in the face of massive data, the current computer is limited in terms of data storage and computing speed. Biomolecular computers have attracted the interest of scientists. DNA computer, as one of the biomolecular computers, has received much attention due to its small size, large storage capacity, fast operation, low energy consumption, and high parallelism. DNA computers use DNA (deoxyribonucleic acid) as a basis to bind enzymes for biochemical reactions that eventually generate DNA sequences carrying specific genetic information. These sequences are used to perform computation and solve the problem. DNA computing is encoded by A, T, C, and G, which is different from the binary combination of the traditional computer.

The design of DNA sequences is the key to perform DNA computing, and the quantity and quality of sequences can directly affect the accuracy and efficiency of calculations. Therefore, a good coding method is of great significance to improve the reliability and accuracy of DNA computing. Deb et al. (2002) proposed a fast, non-dominated sorting genetic algorithm, which was used to solve a class of multi-objective optimization problems. With the help of linear coding, Gaborit and King (2005) developed a DNA code that met the anti-complement constraint and GC content requirement. Thus, they constructed an appropriate DNA sequence set. Shin et al. (2005) carried out multi-objective optimization for DNA sequences, including continuity, similarity, hairpin structure, H-measure, and GC content. Because the traditional algorithm cannot address the heterogeneity and conflict of DNA sequences, scientists designed a multi-objective optimization algorithm based on the artificial bee colony (MO-ABC) (Chaves-González et al., 2013), in which six kinds of conflict problems were solved, and finally, a reliable DNA sequence was generated. In 2014, in order to obtain effective DNA sequences, they proposed to use the multi-objective differential evolution algorithm (DEPT) (Chaves-González and Vega-Rodríguez, 2014) to optimize DNA sequences. In 2015, they used a hybrid multi-objective heuristic algorithm (H-MO-TLBO) (Chaves-González, 2015) to design DNA sequences. Yang et al. (2017) proposed to add the niche exclusion mechanism to improve the invasive weed optimization algorithm, which enhanced robustness and obtained the optimal sequence. Wang et al. (2018) improved the fast non-dominated sorting genetic algorithm II (INSGA-II) and achieved a high convergence rate and reliable DNA sequences. In 2019, in order to further optimize the DNA sequence, Chaves-González and Martínez-Gil (2019) introduced an algorithm called pMO-ABC that harvested different numbers of DNA sequences. In 2020, to decrease the error rate, Cao et al. (2020) presented a new constraint, namely, uncorrelated address, and constructed a set of effective DNA codes. Yin et al. (2020) considered multiple constraints to ensure accurate hybridization of DNA sequences.

In this study, to obtain a high-quality DNA sequence set, the constraints of Close-ending and Pairing Sequences Constraint (PSC) were added to the original constraint combination to form a new combinatorial constraint. The PSC addresses non-specific hybridization that occurs within the DNA sequences set, and the constraint adds a sliding method to ensure that each base can be traversed. Adding PSC to the DNA sequence set reduces the probability of interaction between sequences. The reason for the Close-ending constraint is that the G base and the C base in the sequence have three hydrogen bonds, and there are only two hydrogen bonds between the A base and the T base, so the stability of the AT end of the sequence is less than that of the G-C end. The DNA sequence reacts according to the principle of base complementary pairing. When G-C base is selected as the terminal of the sequence, the desired structure can be achieved when the DNA sequence continues to react. In addition, the constraint also include continuity, hairpin structure, H-measure, similarity, GC content, melting temperature, triplet-bases unpaired. The first four constraints are used as objective functions to calculate the fitness value; the remaining constraints are used to narrow the solution space. At the same time, we enrich and improve the WOA algorithm. In

addition, the predatory behavior of another marine mammal, manta ray (Zhao et al., 2020), is added to expand the predation range and maintain the diversity of the population. To improve the global search ability, the sine cosine model (Mirjalili, 2016) is combined with chaos (Shan et al., 2005) to further expand the solution space. After 23 benchmark functions, it is proved that the Improved Chaos Whale (ICW) optimization algorithm is meaningful. It reached the optimal value in most test functions. Under the combined action of the new constraint combination and the improved algorithm, excellent DNA sequences can be selected as elites. These elite sequences have a minimum value of zero in continuity and hairpin structure and the current minimum value in H-measure, followed by the minimum melting temperature change. In the evaluation of NUPACK, the concentrations of all DNA sequences before and after entering the solution were normalized to total values. All sequences showed stable and usable structures, indicating that the DNA sequences have good stability.

The rest of this article is arranged as follows. The second part introduces the constraints of constructing the DNA sequence set, including the new Close-ending constraint and PSC. The third part introduces the ICW optimization algorithm. In the fourth part, the results of fitness analysis and NUPACK evaluation are given. The last part is the summary and conclusion.

THE CONSTRAINTS ON DNA SEQUENCE DESIGN

To ensure the accuracy of DNA calculation and avoid non-specific hybridization of sequences, constraints must be imposed on DNA sequences. The construction of useful and high-quality DNA sequence set is dependent on strict constraints, which can enhance the robustness of the sequences. Continuity and hairpin structure constraints can effectively prevent sequences from generating secondary structures. The addition of triplet-bases unpaired and PSC to the sequences without secondary structure can not only avoid the self-complementary reaction but also effectively avoid the reaction between the sequences to generate other structures. On this basis, by adding similarity and H-measure, well-structured sequences be obtained that reduce unnecessary hybridization with other sequences. The addition of GC content and melting temperature constraints can keep the sequences in a thermodynamically stable state. Combined with Close-ending constraint, the formed DNA double strand is also stable in structure. Applying these constraints can lead to good sequences. In this study, we adopted all the above constraints in the design sequences.

Continuity

Continuity (Chaves-González et al., 2013) refers to the fact that the same bases are displayed side by side in a confined area. The continuous presence of the same base in a limited region can cause the DNA sequence to stack or distort. To avoid such a secondary structure of the DNA sequence, it is necessary to select a DNA sequence with little continuity. The continuity can be visualized with the following example. Assuming that the DNA sequence threshold is 4, then in the sequence TTAGGGATCCATTTTT, the last sub-sequence with an underscore will trigger the threshold. To improve the quality of DNA sequences, such sequences will be removed from the sequence set. The mathematical formula is as follows:

$$f_{con}(L) = \sum_{p=1}^{m} Con(L_p) \tag{1}$$

$$Con(x) = \sum_{i=1}^{n-CT} T(cont_a(x, i), CT) \tag{2}$$

$$cont_a(x, i) = \begin{cases} c \ if \ \exists c, \ s.t. \ x_i \neq a, x_{i+j} = a \ for \ 1 \leq j \leq c \\ \quad and \ x_{i+c+1} \neq a \\ 0 \ otherwise \end{cases} \tag{3}$$

where m is the number of DNA sequence sets; L_p is a sequence in the DNA set L; n is the number of bases in the current DNA sequence; CT is a specific continuity threshold; $T\ (b,\ CT)$ is a threshold function; when $b > CT$, the result is b; otherwise, the result is 0. $cont_a\ (x,\ i)$ returns the number of consecutive bases, where a ∈{A, T, C, G}.

Hairpin Structure

Hairpin structure (Chaves-González et al., 2013) is a secondary structure caused by the stacking of the DNA sequence itself, which may lead to inaccurate calculation. The hairpin structure is composed of a hair ring and hair stem. The number of bases for the hairpin to form the smallest ring is R_{min}, and P_{min} is the minimum length of the hairpin stem. The mathematical formula for calculating the hairpin value is as follows:

$$f_{hairpin}(L) = \sum_{\rho=1}^{m} Hairpin(L_p) \tag{4}$$

$$Hairpin(x) = \sum_{q=P_{min}}^{(n-R_{min})} \sum_{r=R_{min}}^{n-2q} \sum_{i=1}^{n-2q-r} T \left(\sum_{j=1}^{PL_{qri}} cb(x_{i+j}, x_{n-j}), \frac{PL_{qri}}{2} \right) \tag{5}$$

$$PL_{qri} = min(p + i, l - r - i - p) \tag{6}$$

where r is the ring length of the hairpin structure, and q is the stem length. m is the number of DNA sequence sets, and n is the number of bases in a DNA sequence. For $T\ (a,\ y)$, when $a > y$, the result is a; otherwise, it is 0. The function $cb\ (a,\ b)$ means that when a and b are complementary, the result is 1; otherwise, the result is 0. The equations should be inserted in editable format from the equation editor.

H-Measure

H-measure (Chaves-González, 2015) is a parameter to measure the degree of sequence hybridization. The parameter records

the number of complementary bases of two sequences. The calculation formula is as follows:

$$f_{H-measure}(L) = \sum_{i=1}^{m} \sum_{j=1, i\neq j}^{m} H - measure(L_i, L_j) \tag{7}$$

where m represents the size of sequence L sets; and L_i and L_j represent two sequences in opposite directions. The H-measure is classified into two types: continuous and discontinuous.

$$H_{measure(x,y)} = Max_{g,i}(h_{dis}(x, shift)(y(-)^g y, t)) + h_{cont}(x, shift)(y(-)^g y, t)) \tag{8}$$

where x and y represent different DNA sequences. The shift function defines the offset from y to t.

$$h_{dis}(x, y) = T\left(\sum_{i=1}^{n} bp(x_i, x_j), H_{dis} \times length_{nb}(y)\right) \tag{9}$$

$$h_{cont}(x, y) = \sum_{i=1}^{n} T(cbp(x, y, i), H_{cont} \tag{10}$$

where H_{dis} is a number between 0 and 1; H_{con} is a positive integer from 1 to n; and the function *cbp (x, y, i)* represents the length of a continuous base pair starting from the ith base of the sequence.

$$bp(x, y) = \begin{cases} 1 & x = y \\ 0 & otherwise \end{cases} \tag{11}$$

$$cbp(x, y, i) = \begin{cases} c \ if \ \exists c, \ s.t. \ bp(x_i, y_j) = 0 \ bp(x_{i+j}, y_{i+j}) = 1 \\ \quad for \ 1 \leq j \leq c \ and \ bp(x_{i+c+1}, y_{i+c+1}) = 0 \\ 0 \ otherwise \end{cases} \tag{12}$$

Similarity

Similarity (Chaves-González, 2015) is an important index to evaluate sequence diversity. The higher the similarity, the more likely it is that non-specific hybridization will occur. It can calculate the number of the same base after the shift of two identical sequences. The higher the number of the same base, the more similar is the coding. Similarity is divided into discontinuous similarity and the largest continuous common subset. The formula for calculating similarity is as follows:

$$f_{sim}(L) = \sum_{i=1}^{n} \sum_{j=1}^{n} Max_{g,t}(s_{dis}(x, Shift(y(-)^g y, t)) + s_{cont}(x, Shift(y(-)^g y, t))) \tag{13}$$

where L is the set of DNA sequences; n is the number of set L; and x and y are the different sequences in set L. (-) indicates a gap. Shift represents the offset of the encoding y through t. $g \in [0, 3]$.

$$s_{dis}(x, y) = T\left(\sum_{i=1}^{n} eq(x_i, y_i), DS \times n\right) \tag{14}$$

$$s_{cont}(x, y) = \sum_{i=1}^{n} T(ceq(x, y, i), CS) \tag{15}$$

$$eq(x, y) = \begin{cases} 1 \ x = y \\ 0 \ otherwise \end{cases} \tag{16}$$

$$ceq(x, y, i) = \begin{cases} c \ if \ \exists c, \ s.t. \ eq(x_i, y_j) = 0 \ eq(x_{i+j}, y_{i+j}) = 1 \\ \quad for \ 1 \leq j \leq c \ and \ eq(x_{i+c+1}, y_{i+c+1}) = 0 \\ 0 \ otherwise \end{cases} \tag{17}$$

For *T (a, value)*, when $a > value$, the result is a; otherwise, it is 0. *ceq (x, y, i)* is the length of the continuous common subsequence starting from the ith base of the sequence. DS is a real number from 0 to 1; CS is a positive integer from 1 to n.

Melting Temperature

The melting temperature (Yang et al., 2017) of DNA is an important parameter. In the process of DNA denaturation, double stranded DNA molecules undergo physical changes. In the process of denaturation from double strand to single strand, the temperature at which half of the DNA molecules are released is called the melting temperature. This behavior is an important constraint to ensure the thermodynamic stability of DNA molecules. The melting temperature is usually calculated by the gas chromatography content method and the nearest neighbor method. In this article, the melting temperature is calculated by the nearest neighbor method. The calculation formula is as follows:

$$T_m = \Delta H^\circ / (\Delta S^\circ + R \ln C_t) \tag{18}$$

where ΔH and ΔS represent the standard enthalpy change and entropy change in the hybridization reaction, respectively, and the calculation method is the same as that of the free energy change. C_t is the molar concentration of DNA molecules. When the molecule is a symmetric sequence, its molar concentration is $C_t/4$. R is the gas constant of 1.987 cal/Kmol.

GC Content

GC content (Yang et al., 2017) is the ratio of guanine and cytosine in a DNA sequence. GC content is an important constraint; it can directly affect the stability of a DNA sequence. In a DNA sequence, the number of bases is expressed by n; the number of guanine is a; and the number of cytosine is b. Generally speaking, the GC content (t) of a sequence is

$$t = \frac{a + b}{n} * 100\% \tag{19}$$

Then, in the sequence GTTCGTACTGATCGTAGC, the GC content is (5+4) /18*100%; that is, 50%.

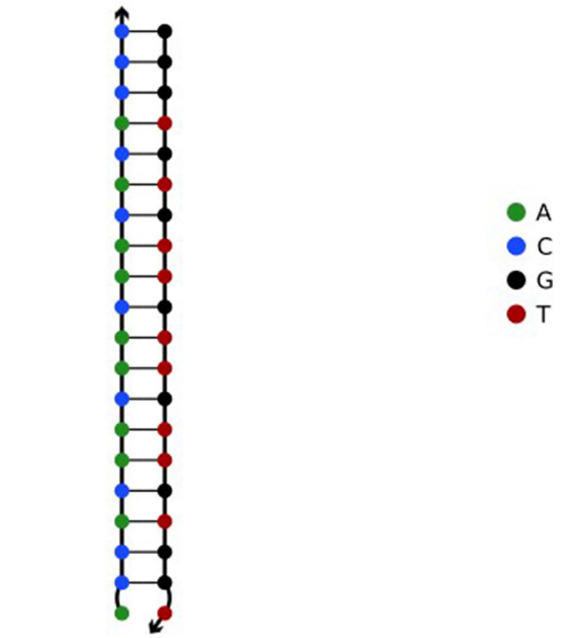

FIGURE 1 | Sequence ACCACAACAACAACACACCC and its complementary sequence.

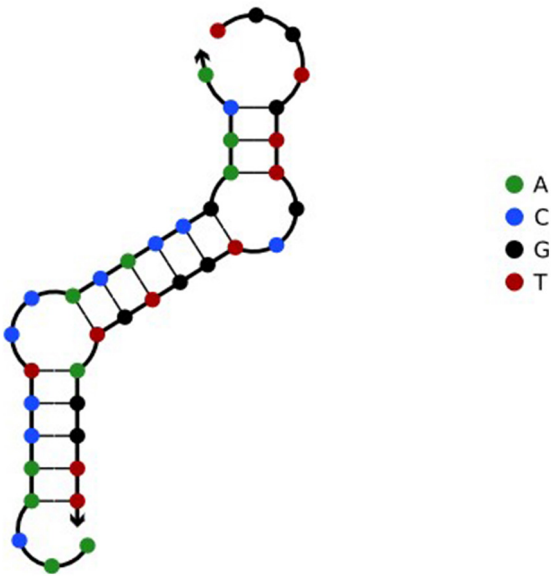

FIGURE 2 | Sequence a: AACAACCTCCACACCGAACA and b: TGGTGTTGCTGGTGTAGGTT.

Triplet-Bases Unpaired

Triplet-bases unpaired (Li et al., 2020) is employed to avoid the complementary reaction of DNA sequence in solution. The sequence entered into NUPACK is evaluated, and the result is denoted by i. $i = C_{output}/C_{input}$; C_{output} is the sum of the sequence concentrations in the solution after NUPACK input; and C_{input} is the sum of the DNA sequence concentrations when NUPACK is input. The closer i is to 1, the higher is the sequence quality. X is a DNA sequence; n is the base number of X; Y is the inverse sequence of X; and x, y are the subsequences of X and Y. It is expressed by the following formula:

$$f_{pair}(x) = \begin{cases} pair(x) & subcb(x, y, k) = 3 \\ x & subcb(x, y, k) \neq 3 \end{cases} \quad (20)$$

where $x = (x_i, x_{i+1}, x_{i+2})$; $y = (y_j, y_{j+1}, y_{j+2})$; and $i, j \in [1, n-2]$. The function $subcb(x, y, k)$ calculates the number of base complementary pairs starting from the kth base.

Close-Ending

In a pair of complementary DNA sequences, the sequence usually creates a gap at one end of the A-T base pair, resulting in an unstable structure (Chaves-González and Vega-Rodríguez, 2014) of the sequence in solution. There are three hydrogen bonds between the G and C bases in the DNA sequence, but there are only two hydrogen bonds between A and T, so the bond strength formed by A-T is less than that between G-C. The structure evaluated in NUPACK is shown in **Figure 1**.

As shown in **Figure 1**, one end of the sequence is G-C. Here, the reaction takes place according to the principle of complementary base pairs. However, there is a gap at one end of the A base and T base, which is less stable than that of GC (Zgarbova et al., 2014). Therefore, by choosing GC base as the port of the sequence, the DNA sequence in the solution can achieve the desired structure. Based on the above, the Close-ending constraint is proposed. Assuming that X is a DNA sequence and X_i is the ith base in the DNA sequence, then

$$X = x_1 x_2 \ldots x_i (x_1, x_i \epsilon \{G, C\}) \quad (21)$$

PSC

In solution, DNA reacts in accordance with the principle of base pairing. When the optimized sequences are put into the solution, there will be a reaction between different sequences. Seven optimized sequences in the article (Chaves-González and Martínez-Gil, 2019) were put into NUPACK for evaluation. The sequence a:AACAACCTCCACACCGAACA reacted with sequence b:TGGTGTTGCTGGTGTAGGTT, and $i(ab) = 0.57/2 = 0.285$. The structural results are shown in **Figure 2**.

As shown in **Figure 2**, sequences a and b reacted in solution to form another structure. $i = 0.285$ means that the sequences react with each other in the solution. In DNA computing, if a DNA sequence in the solution reacts because of the complementary base pairs, the proportion of the DNA sequence in the solution will be reduced, which affects the accuracy of DNA computing. To solve for the reaction between DNA sequences in solution, the PSC is proposed. Different from the Hamming distance constraint, this constraint adds a sliding method to ensure that

every base in the sequence can be traversed. Compared with the H-measure, the comparison between the original sequence and inverse sequence has lower complexity, and the method is simple. Take the a and b sequences as an example. The H-measure calculated values are all 17, while the PSC calculated values are 0. The sequence set, which is constrained by the PSC, reduces the probability of interaction between sequences. This constraint is expressed by the formula:

$$f(L) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} Indep(L_i, L_j') \tag{22}$$

where n is the number of DNA sequences in the L set; L_i is the ith sequence; L_j is the jth sequence; and $L_j{}'$ is the inverse sequence of L_j. x and y represent two different sequences, and the function *cbp (x, y, i)* represents the length of a continuous base pair starting from the ith base of the sequence, where x' represents the sequence of five consecutive bases in the x sequence and y' represents the sequence of five consecutive bases in the y sequence. The value of M is four. The value of M is explained in the **Supplementary Material**.

$$Indep_{(x,y)} = \begin{cases} x, & d = 0 \\ 0, & d > M \end{cases} \tag{23}$$

$$d(x, y) = \sum_{a=1}^{m-4} \sum_{b=1}^{m-4} T(cbp(x_a', y_b', i), M) \tag{24}$$

ALGORITHM

The Whale Optimization Algorithm and Chaos Map

The whale optimization algorithm (WOA) (Mirjalili and Lewis, 2016), a kind of meta starting algorithm, is an effective swarm intelligence optimization algorithm. Compared with other group optimization algorithms, the WOA algorithm has the advantages of simple structure and less adjustment parameters. The predation method is to select a random or the current optimal whale position to simulate the behavior of whale predation. The main inspiration of the whale algorithm design is from the unique whale predation method: bubble net predation. In 2016, Mirjalili et al. simulated the predatory behavior of whales with the contraction closed mechanism and spiral update position and selected the random number p as the boundary of the two behaviors. To ensure the scientificity and fairness of the data, 0.5 was taken as the threshold value.

When $p < 0.5$, the whales use the contraction closure mechanism to prey. In particular, we need to determine the absolute value of A in relation to 1. If the absolute value of A is less than 1, select the current optimal whale position to simulate whale hunting behavior; otherwise, the random position of the whale is selected to simulate the predatory behavior of the whale. It is expressed as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{25}$$

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & |A| < 1 \\ \vec{X_{rand}} - \vec{A} \cdot \vec{D} & |A| \geq 1 \end{cases} \tag{26}$$

and

$$\vec{C} = 2 \cdot \vec{r} \tag{27}$$

$$\vec{D} = \begin{cases} |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| & |A| < 1 \\ |\vec{C}\vec{X_{rand}} - \vec{X}| & |A| \geq 1 \end{cases} \tag{28}$$

where $\vec{a}$ decreases from 2 to 0 as the number of iterations increases; and $\vec{r}$ is a random number from 0 to 1.

When $p \geq 0.5$, to further expand the scope of whale predation, the whale algorithm is improved by adding a new predator–prey method. The added predator–prey mechanism is somersault foraging by learning manta ray simulation. The general predation formula is as follows:

$$\vec{X}(t+1) = \begin{cases} \vec{D'} \cdot e^l \cos(2\pi l) + \vec{X}^*(t) & c < 0.5 \\ \vec{X}(t) + S \cdot (r_1 \cdot \vec{X}^*(t) - r_2 \cdot \vec{X}(t)) & c \geq 0.5 \end{cases} \tag{29}$$

where $\vec{D'}$ represents the distance between the current whale and the optimal whale; $l \in [-1,1]$; the default value of S is 2; and r_1, r_2, and c are random numbers between 0 and 1,

$$\vec{D'} = |\vec{X'}(t) - \vec{X}| \tag{30}$$

Chaos has the characteristics of randomness, regularity, and ergodicity. When solving the function optimization problem, the diversity of the population can be maintained, and the global search ability can be improved. Tent (Shan et al., 2005)has better ergodic uniformity and can improve the search speed of the algorithm, and it can generate more evenly distributed values between [0,1]. The formula is as follows:

$$z_{i+1} = \begin{cases} \frac{z_i}{u} & 0 \leq z_i < 0 \\ \frac{1-z_i}{1-u} & u \leq z_i \leq 1 \end{cases} \tag{31}$$

When $u = 1/2$, tent is the most classical form. The sequence of this form has uniform distribution and approximately uniform distribution density for different parameters. Therefore, the formula of the tent chaotic map in this paper is

$$z_{i+1} = \begin{cases} 2z_i & 0 \leq z_i < \frac{1}{2} \\ 2(1 - z_i) & \frac{1}{2} \leq z_i \leq 1 \end{cases} \tag{32}$$

The sine cosine algorithm is a new intelligent optimization algorithm proposed by Mirjalili in 2016. It is based on the mathematical model of outward sine and cosine wave or the wave in the direction of the optimal solution. Using multiple random variables and adaptive variables to calculate the current solution location, different regions in the space can be searched, effectively avoiding local optimization and converging to the global optimum.

ICW Optimization Algorithm

Based on the above algorithm introduction, the ICW optimization algorithm is proposed. To expand the predator–prey of whales, somersault foraging method was added. In this strategy, the position of the candidate solution is regarded as a pivot. Each individual tends to somersault around the pivot to a new position. Therefore, each individual always updates its surrounding location until it finds the best location so far. Because the swarm intelligence algorithm has the disadvantage of falling into local optimization, in order to obtain better global optimization ability and increase the search range, the sine cosine mathematical model is introduced, which makes the optimization direction fluctuate outward or to the direction of the optimal solution. The chaos is added to the sine cosine model to further expand the coverage of the solution space. This makes it easy for the algorithm to escape from the local optimal solution, thus maintaining the diversity of the population and improving the global search ability. In general, this algorithm achieves the desired outcome.

The details of the algorithm are as follows:

Step 1. Introduce the parameters and generate the initial population.
Step 2. Calculate the fitness value of the current population and find the optimal solution, which is the minimum fitness value.
Step 3. Obtain the parameters for every iteration.
Step 4. Generate random number p and determine the relationship between p and 0.5. If $p < 0.5$, enter step 5; otherwise, enter step 6.
Step 5. Generate $|A|$ and determine the relationship between $|A|$ and 1. If $|A| < 1$, select the current optimal location for the update operation. Otherwise, select a random location for the update operation.
Step 6. Generate c and determine the relationship between c and 0.5. If $c < 0.5$, use the spiral upward mechanism to update the position; otherwise, use the somersault mode to update the position.
Step 7. Use sines and cosines with chaos to increase the global search capability and get the set of desirable populations.
Step 8. Select populations that satisfy the constraint combinations.
Step 9. Increase the number of iterations to determine the relationship between the current iteration number and the maximum iteration number. If the current iteration number is less than the maximum, step 2 is performed; otherwise, step 10 is performed.
Step 10. Calculate the fitness function of the desirable populations and select the optimal populations as the final result.

In this work, the conversion method between numbers and letters is as follows: 0-C,1-T,2-A,3-G.

The flow chart of the ICW optimization algorithm is shown in **Figure 3**, and the **Figure 4** presents the pseudo-code of a general implementation.

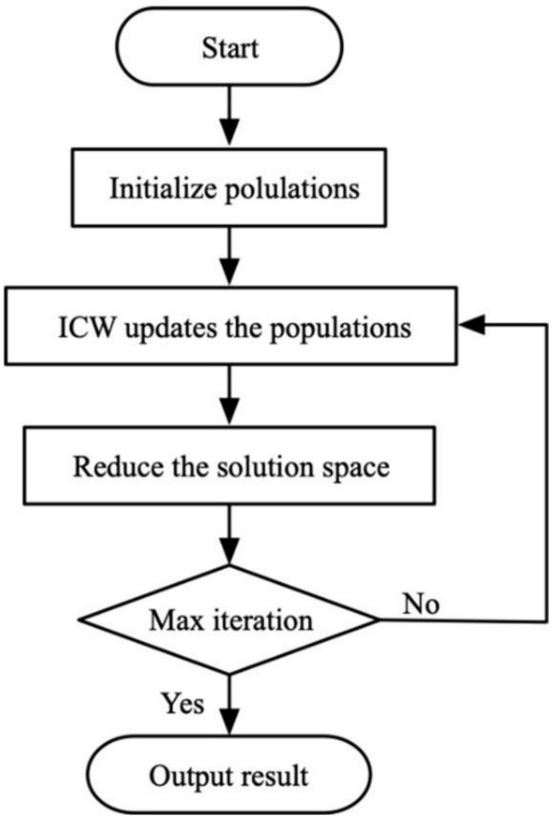

FIGURE 3 | Flow chart of ICW.

Benchmark Test Functions

To better demonstrate the performance of the ICW optimization algorithm, we tested 23 benchmark functions that are widely used. It is worth noting that because each algorithm focuses on solving different types of problems, not every algorithm can get the minimum value of all functions.

For the 23 test functions, the functions can be divided into three categories according to the function types (Digalakis and Margaritis, 2001), namely, unimodal benchmark function (F1–F7), multimodal benchmark function (F8–F13), and fixed-dimension multimodal benchmark function (F14–F23). The equations of the different types of functions are listed in the **Supplementary Material**. In the table of the **Supplementary Material**, Function represents benchmark functions; Dim represents function dimension; rang represents the definition domain value of Function; and f_{min} represents the optimal solution of Function. The unimodal benchmark function has only one global optimal value, so it can be used to test the benchmark development ability of the algorithm. The multimodal benchmark function has one optimal value and several local optimal values. The number of optimal values increases with the increase of the dimension, so it can be used to test the exploration ability of the algorithm and its ability to jump out of the local optimum. Like the multimodal benchmark function, the fixed-dimension multimodal function has only

```
Inputs:The population size N and maximum number of
iterations t
Outputs:DNA sequences that satisfy combination constraint
Initialize the random population Xi (i=1,2…N)
While(t<T)
Calculate the fitness values of each agent.
if 1 (p<0.5)
  if 2(|A|<1)
     Update the position of the current search agent by the
     Eq.(26)
     else if 2(|A|⩾1)
     Select a random search agent(X_rand)
     Update the position of the current search agent by the
     Eq.(26)
  end if 2
else if1(p⩾0.5)
  if3(c<0.5)
     Update the position of the current search by the Eq.(29)
  else if 3(c⩾0.5)
     Update the position of the current search by the Eq.(29)
  end if 3
end if 1
Expand population by sines and cosines with chaos
End while
```

FIGURE 4 | Pseudo-code of the ICW optimization algorithm.

one global optimal value and many local optimal solutions. However, the solution space of the multi-modal function with fixed dimension is very small, so the step size should be adjusted adaptively.

We compared our algorithm with the algorithms HSWOA (Li et al., 2020), WOA (Mirjalili and Lewis, 2016), GA (Heidari et al., 2019), PSO (Yin et al., 2020), FPA (Yang et al., 2014), GWO (Mirjalili et al., 2014), FA (Gandomi et al., 2011), MFO (Mirjalili, 2015), TLBO (Rao et al., 2011), and DE (Yin et al., 2020). We compared the average (AVG) and standard deviation (STD) of 23 benchmark functions. Other conditions remain unchanged, and each function is iterated 500 times and run 30 times. This operation ensures the validity of the test. The test results are shown in **Tables 1**, **2**. **Table 1** shows the results of F1–F13 functions calculated by different algorithms, and **Table 2** shows the results of the remaining functions.

As shown in **Tables 1**, **2**, the ICW optimization algorithm is better than the other algorithms in most function values. In particular, our algorithm achieves the optimal values for unimodal functions F1–F4, F9, F11 and F16, F18, F19 of the multimodal functions. In addition, in the functions F5, F7, F10, F11, F15–F17, F19, F21, and F22, the improved algorithm is significantly better than the other algorithms in the table. With respect to the original algorithm WOA, by comparing the average value of the two algorithms, it can be found that the improved ICW optimization algorithm significantly enhances the development ability and gives results that are closer to the global optimization. For standard deviation, the new algorithm has better stability. Compared with the other functions in the table, the ICW optimization algorithm occupies nearly 75% of the optimal value. So, this algorithm has a strong competitive advantage.

To further illustrate that the improved algorithm has more advantages than the WOA algorithm, the single peak test function, multiple test function, and fixed image of the multimodal function are drawn. We do not select the function that reaches the optimal value, which ensures universal optimization of the new algorithm and enhances the persuasiveness. As shown in the figure, F7, F12, and F20 are selected **Figure 5**. The unimodal benchmark function F7 can jump out of the local optimal solution and converge to the global optimum. The multimodal benchmark function F12 and the fixed-dimension multimodal benchmark function F20 can converge to the local optimal solution quickly.

RESULTS AND ANALYSIS

In this article, the results are obtained by running the algorithm in MATLAB R2018a. The computer had the Windows 10 operating system, Intel (R) CPU2.70 GHz, and ARM 8.00 GB. The minimum stem length and ring length of the hairpin structure were 6. The minimum continuity threshold was set to 2. In the continuous case, the threshold of similarity and H-measure were six, and the threshold of discontinuous similarity and discontinuous H-measure were 0.17. In addition, Tm was obtained by using the proximity model. The concentration of DNA was set at 10 nM, and salt concentration was set at 1 M.

This part compares ICW with other algorithms, namely NACST/Seq (Shin et al., 2005), DEPT (Chaves-González and Vega-Rodríguez, 2014), MO-ABC (Chaves-González et al., 2013), pMO-ABC (Chaves-González and Martínez-Gil, 2019), and HSWOA (Li et al., 2020). We compare the average values of the above algorithms for continuity, hairpin structure, H-measure, similarity, and melting temperature. We also input the optimal sequence of the algorithm into NUPACK for experimental simulation and compared the simulation results.

Table 3 compares the sequences obtained by our proposed algorithm with those of other algorithms. Seven sequences were used in **Table 3**; each sequence has 20 bases. According to the average values of continuity, hairpin structure, H-measure, similarity, and Tm in **Table 3**, **Figures 6–9**, and **Table 4**, the results show that our work outperforms other algorithms in terms of continuity and hairpin structure. In the aspect of H-measure, the results obtained by our algorithm are much better than other algorithms, which indicates that our algorithm can effectively avoid non-specific hybridization. In the reaction solution, the DNA sequence can also maintain the maximum value. In general, it can effectively avoid non-specific hybridization between sequences. The GC content is always maintained at 50%, representing that the sequences obtained by our algorithm have stable thermodynamic properties.

In **Figure 6**, the result shows the average fitness of continuity and hairpin for different algorithms. From the bar graph, it can be

TABLE 1 | Result of benchmark functions (F1–F13) with 30 dimensions.

ID	Metric	ICW	HSWOA	WOA	GA	PSO	FPA	GWO	FA	MFO	TLBO	DE
F1	AVG	**0.00E+00**	2.71E−91	1.41E-03	1.03E+03	1.83E+04	2.01E+03	1.18E−27	7.11E−03	1.01E+03	2.17E−89	1.33E−03
	STD	0.00E+00	1.24E−90	4.91E-30	5.79E+02	3.01E+03	5.60E+02	1.47E−27	3.21E−03	3.05E+03	3.14E−89	5.92E−04
F2	AVG	**0.00E+00**	7.03E−58	1.06E-21	2.47E+01	3.58E+02	3.22E+01	9.71E−17	4.34E−01	3.19E+01	2.77E−45	6.83E−03
	STD	0.00E+00	2.94E−57	2.93E-21	5.68E+00	1.35E+03	5.55E+00	5.60E−17	1.84E−01	2.06E+01	3.11E−45	2.06E−03
F3	AVG	**0.00E+00**	1.11E+04	5.39E-07	2.65E+04	4.05E+04	1.41E+03	5.12E−05	1.66E+03	2.43E+04	3.91E−18	3.97E+04
	STD	0.00E+00	4.54E+03	2.93E-06	3.44E+03	8.21E+03	5.59E+02	2.03E−04	6.72E+02	1.41E+04	8.04E−18	5.37E+03
F4	AVG	**0.00E+00**	3.22E+01	0.07E+00	5.17E+01	4.39E+01	2.38E+01	1.24E−06	1.11E−01	7.00E+01	1.68E−36	1.15E+01
	STD	0.00E+00	0.07E+01	0.39E+00	1.05E+01	3.64E+00	2.77E+00	1.94E−06	4.75E−02	7.06E+00	1.47E−36	2.37E+00
F5	AVG	**2.54E+01**	2.76E+01	2.78E+01	1.95E+04	1.96E+07	3.17E+05	2.70E+01	7.97E+01	7.35E+03	2.54E+01	1.06E+02
	STD	0.25E0+00	0.58E+00	0.76E+00	1.31E+04	6.25E+06	1.75E+05	7.78E−01	7.39E+01	2.26E+04	4.26E−01	1.01E+02
F6	AVG	0.11E+00	0.33E+00	3.11E+00	9.01E+02	1.87E+04	1.70E+03	8.44E−01	6.94E−03	2.68E+03	3.29E−05	1.44E−03
	STD	0.03E+00	0.18E+04	0.53E+00	2.84E+02	2.92E+03	3.13E+02	3.18E−01	3.61E−03	5.84E+03	8.65E−05	5.38E−04
F7	AVG	**6.19E-05**	1.29E−03	1.42E-03	1.91E−01	1.07E+01	3.41E−01	1.70E−03	6.62E−02	4.50E+00	1.16E−03	5.24E−02
	STD	6.37E-05	1.20E−03	1.14E−03	1.50E−01	3.05E+00	1.10E−01	1.06E−03	4.23E−02	9.21E+00	3.63E−04	1.37E−02
F8	AVG	−1.23E+04	−1.11E+04	−5.08E+03	**−1.26E+04**	−3.86E+03	−6.45E+03	−5.97E+03	−5.85E+03	−8.48E+03	−7.76E+03	−6.82E+03
	STD	5.26E+02	1.50E+03	6.95E+02	4.51E+00	2.49E+02	3.03E+02	7.10E+02	1.16E+03	7.98E+02	1.04E+03	3.94E+02
F9	AVG	**0.00E+00**	9.01E+00	0.00E+00	9.04E+00	2.87E+02	1.82E+02	2.19E+00	3.82E+01	1.59E+02	1.40E+01	1.58E+02
	STD	0.00E+00	1.44E+01	0.00E+00	4.58E+00	1.95E+01	1.24E+01	3.69E+00	1.12E+01	3.21E+01	5.45E+00	1.17E+01
F10	AVG	**8.88E-16**	2.78E−15	7.40E+00	1.36E+01	1.75E+01	7.14E+00	1.03E−13	4.58E−02	1.74E+01	6.45E−15	1.21E−02
	STD	0.00E+00	1.80E−15	9.89E+00	1.51E+00	3.67E−01	1.08E+00	1.70E−14	1.20E−02	4.95E+00	1.79E−15	3.30E−03
F11	AVG	**0.00E+00**	**0.00E+00**	2.89E-04	1.01E+01	1.70E+02	1.73E+01	4.76E−03	4.23E−03	3.10E+01	**0.00E+00**	3.52E−02
	STD	0.00E+00	0.00E+00	1.58E+03	2.43E+00	3.17E+01	3.63E+00	8.57E−03	1.29E−03	5.94E+01	0.00E+00	7.20E−02
F12	AVG	5.06E-03	6.90E−00	3.39E-01	4.77E+00	1.51E+07	3.05E+02	4.83E−02	3.13E−04	2.46E+02	**7.35E−06**	2.25E−03
	STD	2.54E-03	7.06E−00	0.21E+00	1.56E+00	9.88E+06	1.04E+03	2.12E−02	1.76E−04	1.21E+03	7.45E−06	1.70E−03
F13	AVG	0.10E-00	4.52E−00	1.88E+00	1.52E+01	5.73E+07	9.59E+04	5.96E−01	**2.08E−03**	2.73E+07	7.89E−02	9.12E−03
	STD	9.40E-02	9.14E−00	3.66E+01	4.52E+00	2.68E+07	1.46E+05	2.23E−01	9.62E−04	1.04E+08	8.78E−02	1.16E−02

Bold values represent the optimal value of the function in the table.

TABLE 2 | Results of benchmark functions (F14–F23) with 30 dimensions.

ID	Metric	ICW	HSWOA	WOA	GA	PSO	FPA	GWO	FA	MFO	TLBO	DE
F14	AVG	3.73E-00	2.18E−00	2.11E+00	**9.98E−01**	1.39E+00	9.98E−01	4.17E+00	3.51E+00	2.74E+00	**9.98E−01**	1.23E+00
	STD	4.17E-00	2.11E−00	2.49E+00	4.52E−16	4.60E−01	2.00E−04	3.61E+00	2.16E+00	1.82E+00	4.52E−16	9.23E−01
F15	AVG	**3.99E-04**	5.25E−04	5.72E−04	3.33E−02	1.61E−03	6.88E−04	6.24E−03	1.01E−03	2.35E−03	1.03E−03	5.63E−04
	STD	1.30E-04	2.54E−04	3.24E−04	2.70E−02	4.60E−04	1.55E−04	1.25E−02	4.01E−04	4.92E−03	3.66E−03	2.81E−04
F16	AVG	**−1.03E+00**	−1.03E+00	**−1.03E−00**	−3.78E−01	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**
	STD	1.94E-10	1.32E−08	4.20E−07	3.42E−01	2.95E−03	6.78E−16	6.78E−16	6.78E−16	6.78E−16	6.78E−16	6.78E−16
F17	AVG	**3.98E-01**	3.97E−01	3.97E−01	5.24E−01	4.00E−01	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**	**3.98E−01**
	STD	8.20E-07	1.19E−07	2.70E−05	6.06E−02	1.39E−03	1.69E−16	1.69E−16	1.69E−16	1.69E−16	1.69E−16	1.69E−16
F18	AVG	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	3.10E+00	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**
	STD	1.65E-14	5.98E-07	4.22E-15	0.00E+00	7.60E−02	0.00E+00	4.07E−05	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F19	AVG	**−3.86E+00**	**−3.86E+00**	−3.85E+00	−3.42E+00	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**
	STD	6.85E-06	3.19E−03	2.70E−03	3.03E−01	1.24E−03	3.16E−15	3.14E−03	3.16E−15	1.44E−03	3.16E−15	3.16E−15
F20	AVG	−3.2821	−3.27	−2.98105	−1.61351	−3.11088	**−3.2951**	−3.25866	−3.28105	−3.23509	−3.24362	−3.27048
	STD	0.057049	0.060296	0.376653	0.46049	0.029126	0.019514	0.064305	0.063635	0.064223	0.15125	0.058919
F21	AVG	**−10.142**	−8.7129	−7.04918	−6.66177	−4.14764	−5.21514	−8.64121	−7.67362	−6.8859	−8.64525	−9.64796
	STD	0.011404	2.4655	3.629551	3.732521	0.919578	0.008154	2.563356	3.50697	3.18186	1.76521	1.51572
F22	AVG	**−10.3907**	−8.1948	−8.18178	−5.58399	−6.01045	−5.34373	−10.4014	−9.63827	−8.26492	−10.2251	−9.74807
	STD	0.015752	2.9941	3.29202	2.605837	1.962628	0.053685	0.000678	2.293901	3.076809	0.007265	1.987703
F23	AVG	−10.527	−8.7416	−9.34238	−4.69882	−4.72192	−5.29437	−10.0836	−9.75489	−7.65923	−10.0752	**−10.5364**
	STD	0.011063	2.8113	2.414737	3.256702	1.742618	0.356377	1.721889	2.345487	3.576927	1.696222	8.88E−15

Bold values represent the optimal value of the function in the table.

TABLE 3 | Comparing sequences from NACS/Seq, DEPT, MO-ABC, pMO-ABC, and HSWOA.

Seq.	C	P	H	S	Tm	GC%	Seq.	C	P	H	S	Tm	GC%
Our work							**NACST/Seq [21]**						
CCTCTCCATCCTTATCCTTC	0	0	35	66	60.88	50	CTCTCATCTCTCCGTTCTTC	0	0	37	58	61.43	50
CCAGACCAATACAGAACCAC	0	0	50	57	62.54	50	TATCCTGTGGTGTCCTTCCT	0	0	45	57	64.46	50
CTCCTCTTCTCCTTCTTCTC	0	0	28	82	60.72	50	GTATTCCAAGCGTCCGTGTT	0	0	55	49	65.29	50
CACAACCAATCACTCTCACC	0	0	38	65	63.10	50	TCTCTTACGTTGGTTGGCTG	0	0	51	53	64.63	50
CCACCTGACCGACTAATAAC	0	0	48	61	62.02	50	CTCTTCATCCACCTCTTCTC	0	0	43	58	61.38	50
CCAACCACTCTTCTACAACC	0	0	37	72	62.47	50	ATTCTGTTCCGTTGCGTGTC	0	0	52	56	65.82	50
CCTTCTTCTCTCTCTCTCTC	0	0	30	75	60.13	50	AAACCTCCACCAACACACCA	9	0	55	43	66.71	50
DEPT [23]							**MO-ABC [22]**						
CCATTCCTTAACCTCTCTCC	0	0	59	39	61.39	50	GTAAGGAAGGCAAGGCAGAA	0	0	42	54	64.70	50
ACACACACACACACACACAC	0	0	27	49	65.85	50	GTTGGTGGTTGTTGGTGGTT	0	0	46	36	66.00	50
GGAAGGAGGAGGAAGAAGAA	0	0	37	45	62.83	50	GGAGACGGAATGGAAGAGTA	0	0	44	55	62.93	50
GAGAGAAGAGAAGAGGCCAA	0	0	39	53	63.17	50	CCATTCTTCTCTTCTCTCCC	9	0	67	22	61.39	50
ACCACAACAACAACACACCC	9	0	29	55	65.96	50	AGGAGAGGAGAGGAGGAAAA	16	0	31	53	63.80	50
GGAGCAATGGAGAATAAGGG	9	0	48	47	62.42	50	ATAAGAGAGAGAGAGAGGGG	16	0	34	51	61.11	50
CCATACCAGCCAACCGAAAA	16	0	42	56	65.33	50	GAGCCAACAGCCAACCAAAA	16	0	48	45	66.40	50
pMO-ABC [28]							**HSWOA [36]**						
GGTGGTATTGGTGGTATTGG	0	0	47	47	62.64	50	CTCGTCTAACCTTCTTCAGC	0	0	63	51	62.28	50
CTTCTCTTCTCTTGCCGCTT	0	0	39	56	64.70	50	CTGTGTGGAATGCAAGGATG	0	0	64	48	63.82	50
CTCTCTCTCTCACTCTCTCA	0	0	41	48	61.32	50	CGAGCGTAGTGTAGTCATCA	0	0	63	69	63.56	50
AACAACCTCCACACCGAACA	0	0	62	32	66.69	50	AGTTACAGGACACCACCGAT	0	0	65	51	66.39	50
TGTGGTTGGTTAGTCGGTTG	0	0	46	49	63.80	50	CAGTAGCAGTCATAACGAGC	0	0	64	56	62.69	50
TGGTGTTGCTGGTGTAGGTT	0	0	48	51	66.46	50	GCATAGCACATCGTAGCGTA	0	0	59	54	64.60	50
CTCTCATTCCTTCTTACCCC	16	0	43	51	61.40	50	TGGACCTTGAGAGTGGAGAT	0	0	62	50	64.44	50

TABLE 4 | Comparing the melting temperatures of the various algorithm sequences.

	ICW	NACST/Seq	DEPT	MO-ABC	pMO-ABC	HSWOA
Var	1.24	4.33	3.37	4.37	4.91	1.86

clearly seen that the sequence obtained by the ICW optimization algorithm was better than that obtained by other algorithms in terms of continuity and hairpin structure. ICW optimization algorithm obtained the minimum value. Therefore, the sequence obtained by our algorithm can effectively avoid the generation of secondary structures.

Using **Table 3**, we calculated the average fitness of different algorithms for H-measure and similarity (**Figure 7**). It can be clearly seen from the graph that our algorithm obtains the minimum H-measure. However, our algorithm is different from other comparable algorithms in terms of the similarity.

After comparing the above average fitness values, we also studied the evaluation results in NUPACK. Most DNA solution

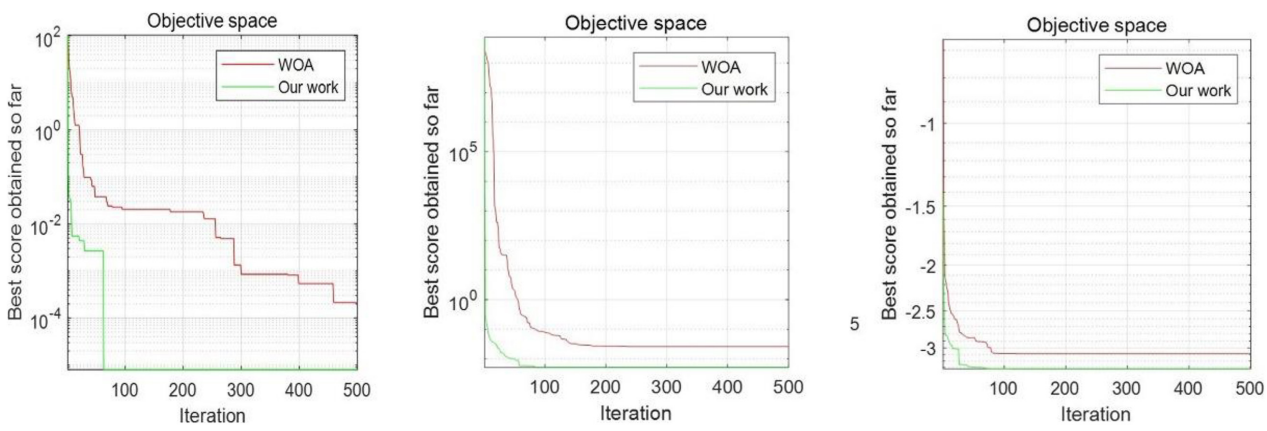

FIGURE 5 | Test functions of functions F7, F12, and F20.

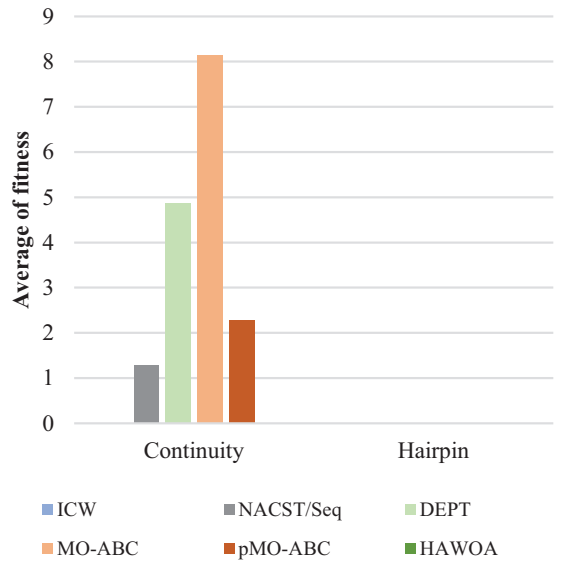

FIGURE 6 | Average fitness of continuity and hairpin for ICW and other algorithms.

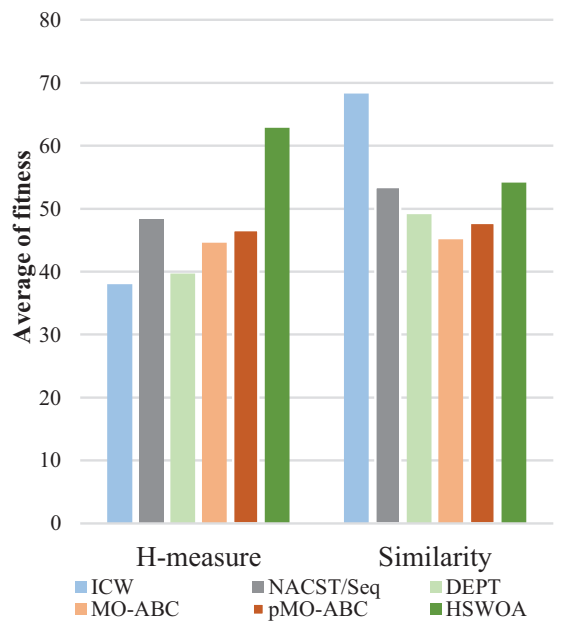

FIGURE 7 | Average fitness of H-measure and similarity for ICW and other algorithms.

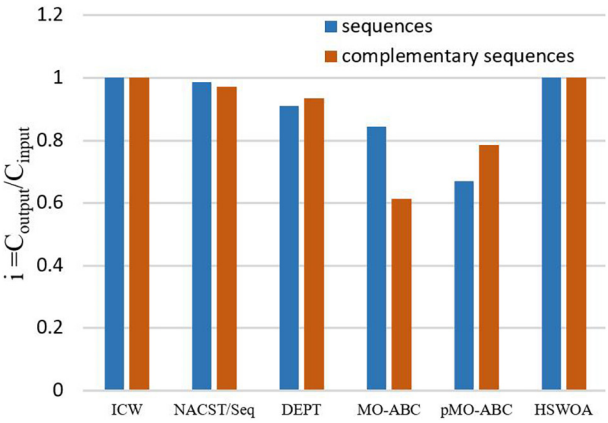

FIGURE 8 | Value of i for different algorithms.

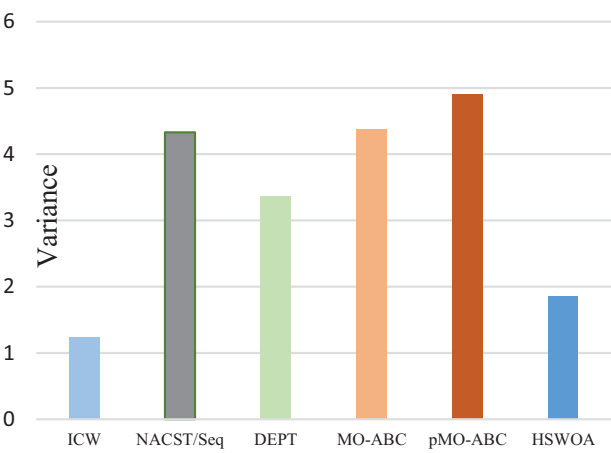

FIGURE 9 | Variance of different algorithms.

experiments are performed at room temperature, so the temperature was set at 25°C. We input the seven sequences (optimal sequences of each algorithm) of different algorithms in **Table 3** into NUPACK. After evaluation, the values of C_{input} and C_{output} were calculated, and the value of i was obtained, as shown in **Figure 8**.

Figure 8 shows the values of i for different algorithms. It should be noted that the closer i is to 1, the higher is the quality of the sequences. In the figure, each algorithm has two columnar regions. The first represents seven sequences of the algorithms in **Table 3**, and the second column region represents the complement sequence of these seven sequences. First, consider the sequences in **Table 3**. In the histogram, ICW and HSWOA, which is our previous work, both have the value 1 for i. In the HSWOA algorithm, the triplet-bases unpaired constraint is added to form a constraint combination to control a single sequence so that the sequence does not react with itself. In the ICW optimization algorithm, PSC is added to form a new constraint combination to reduce the reaction between different sequences. From the value of i, the new algorithm with constraint is better performing than other algorithms.

In addition, considering the double stranded structure of DNA, the complements of seven DNA sequences were evaluated. The evaluation results are shown in **Figure 8**. In the figure, the values of the ICW optimization algorithm and HSWOA algorithm are 1, indicating that there is no reaction between complementary sequences, which is what we expected. In addition, seven DNA sequences and

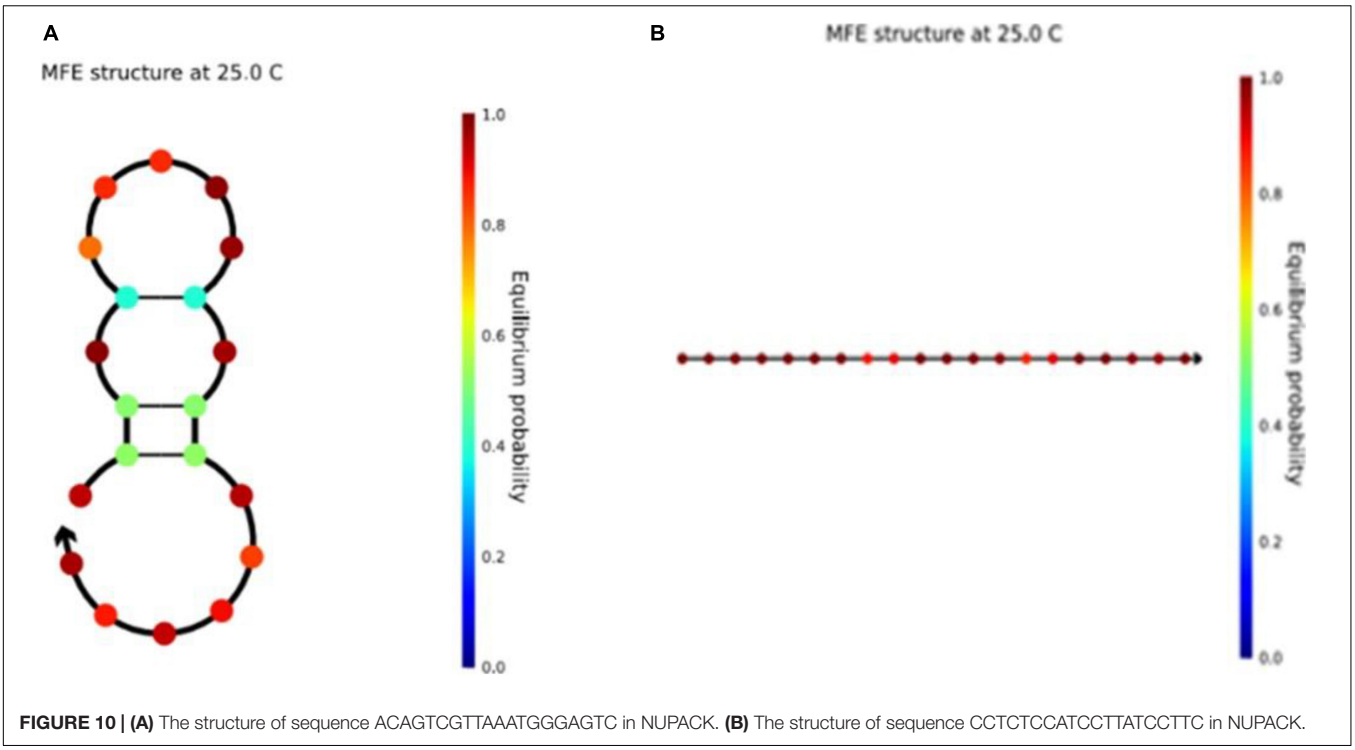

FIGURE 10 | (A) The structure of sequence ACAGTCGTTAAATGGGAGTC in NUPACK. **(B)** The structure of sequence CCTCTCCATCCTTATCCTTC in NUPACK.

complements were input into NUPACK for evaluation, finding the Fourteen sequences were complete reactions, which were complementary to each other.

Table 4 and **Figure 9** show the variance of the melting temperature of different algorithms. The values in the table are calculated from the Tm values in **Table 4**. In DNA computing, to ensure the consistency of the biochemical reactions of DNA molecules, all the DNA molecules participating in the biochemical reactions should be uniform. In other words, if the variance is small, the melting temperature change will be small, and the probability of achieving the desired result will be increased.

The significance of this work is evaluated again by the variance of melting temperature (Tm) and the value of *i*. In order to ensure the consistency of the work, in the comparison of indicators, the control group still chooses sequence a and sequence b and compares with the average value of the indicators in this work. The variance of the Tm of the DNA sequence obtained by the pMO-ABC algorithm is 1.65, while the variance of the Tm in this work is 1.24, which is reduced by 33.1%. Smaller variance of Tm is more conducive to controlling the temperature during the reaction, and smaller temperature fluctuation is more conducive to the reaction. The i value of the two sequences of a and b is $i = 0.285$, while the i of the set of DNA sequences in this work is all 1. The closer i is to 1, the higher the sequence quality.

It is worth noting that the sequence structure of the unstable available structure in the previous article (Li et al., 2020) is shown in **Figure 10A**, and this structure has also been changed in this work. Seven optimized sequences satisfying the new combinatorial constraints were input into NUPACK at the same time. We found that all the sequences were linear structures. They were stable and available structures that could improve the accuracy of DNA calculation. As shown in the figure, the color bar on the right shows the stability of the sequence. The closer the color is to red, the more stable it is. Compared with the two graphs, the sequence structure of **Figure 10B** is more stable. In the graph of the two DNA sequences, the figures represent the stable structure of the sequences. In addition, other DNA sequence structures obtained from our work are presented in the **Supplementary Material**.

CONCLUSION

In this study, we input existing DNA sequences into NUPACK and evaluated them. It was found that the DNA sequences may react with each other owing to base complementary pairing. Therefore, we propose a new constraint, PSC, to solve this problem. In addition, due to the double strand structure of DNA, if the A-T base is located at one end of the DNA sequence and its complementary sequence, there may be a gap, which leads to a decrease of the accuracy of calculation. The proposed Close-ending constraint can effectively avoid the generation of such sequences. These two new constraints were fused into the previous constraint combination to form a new combination constraint. Then, the new ICW optimization algorithm was used to obtain the sequences satisfying the new combination constraints, and the sequence results were analyzed. The analysis results show that the current minimum is obtained on continuity, hairpin, and H-measure. This shows that the sequence greatly improves the ability to avoid secondary structures. In terms

of the Tm value, the minimum variance was obtained by calculation, which ensured that the DNA molecules participating in biochemical reactions were more uniform and improved the thermodynamic stability. When the sequences were input into NUPACK for evaluation, the concentration of the obtained sequence in the solution was the same as before, indicating that the DNA sequence did not react with itself or other sequences in the solution, and the DNA sequence was stable and available in the solution, which improved the accuracy of DNA calculation.

In the future, to produce stable DNA sequence and ensure the accuracy of DNA computing, we will take the optimization of the DNA sequence set as the primary task. With respect to the algorithm, we will further optimize it. At the same time, reducing the similarity of DNA sequence sets and exploring the linear structure of DNA sequences in solution will also be two important aspects. In addition, to expand the application of DNA computing, we will also work with the machine learning (Song et al., 2019; Gong et al., 2020) and bioinformatics (Zou et al., 2020).

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

XL and ZW designed the DNA sequences, analyzed the data and the performance, and wrote the manuscript. BW and TS supervised the work, evaluated the performance, and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

FUNDING

This work is supported by the National Key Technology R&D Program of China (No. 2018YFC0910500), the National Natural Science Foundation of China (Nos. 61425002, 61751203, 61772100, 61972266, 61802040, and 61672121), the High-level Talent Innovation Support Program of Dalian City (Nos. 2017RQ060 and 2018RQ75), and the Innovation and Entrepreneurship Team of Dalian University (No. XQN202008).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.644484/full#supplementary-material

REFERENCES

Adleman, L. M. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266, 1021–1024. doi: 10.1126/science.7973651

Benenson, Y., Gil, B., Ben-Dor, U., Adar, R., and Shapiro, E. (2004). An autonomous molecular computer for logical control of gene expression. *Nature* 429, 423–429. doi: 10.1038/nature02551

Cao, B., Li, X., Zhang, X., Wang, B., Zhang, Q., and Wei, X. (2020). Designing uncorrelated address constrain for DNA storage by DMVO algorithm. *IEEE/ACM Trans. Comput. Biol. Bioinform.* doi: 10.1109/tcbb.2020.3011582

Cao, B., Zhang, X., Wu, J., Wang, B., Zhang, Q., and Wei, X. (2021). Minimum free energy coding for DNA storage. *IEEE Trans. Nanobiosci.* 20, 212–222. doi: 10.1109/tnb.2021.3056351

Chaves-González, J. M. (2015). Hybrid multiobjective metaheuristics for the design of reliable DNA libraries. *J. Heuristics* 21, 751–788. doi: 10.1007/s10732-015-9298-x

Chaves-González, J. M., and Martínez-Gil, J. (2019). An efficient design for a multi-objective evolutionary algorithm to generate DNA libraries suitable for computation. *Interdiscip. Sci.* 11, 542–558. doi: 10.1007/s12539-018-0303-6

Chaves-González, J. M., and Vega-Rodríguez, M. A. (2014). DNA strand generation for DNA computing by using a multi-objective differential evolution algorithm. *Biosystems* 116, 49–64. doi: 10.1016/j.biosystems.2013.12.005

Chaves-González, J. M., Vega-Rodríguez, M. A., and Granado-Criado, J. M. (2013). A multiobjective swarm intelligence approach based on artificial bee colony for reliable DNA sequence design. *Eng. Appl. Artif. Intell.* 26, 2045–2057. doi: 10.1016/j.engappai.2013.04.011

Cherry, K. M., and Qian, L. (2018). Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature* 559:370. doi: 10.1038/s41586-018-0289-6

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2002). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *IEEE Trans Evol Comput.* 6, 182–197.

Deng, L., Wang, Y., Noor-A-Rahim, M., Guan, Y. L., Shi, Z., Gunawan, E., et al. (2019). Optimized code design for constrained DNA data storage with asymmetric errors. *IEEE Access* 7, 84107–84121. doi: 10.1109/access.2019.2924827

Digalakis, J. G., and Margaritis, K. G. (2001). On benchmarking functions for genetic algorithms. *Int. J. Comput. Math.* 77, 481–506. doi: 10.1080/00207160108805080

Gaborit, P., and King, O. D. (2005). Linear constructions for DNA codes. *Theor. Comput. Sci.* 334, 99–113. doi: 10.1016/j.tcs.2004.11.004

Gandomi, A. H., Yang, X.-S., and Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Comput. Struct.* 89, 2325–2336. doi: 10.1016/j.compstruc.2011.08.002

Gong, F. M., Ma, Y. H., Zheng, P., and Song, T. (2020). A deep model method for recognizing activities of workers on offshore drilling platform by multistage convolutional pose machine. *J. Loss Prev. Process Ind.* 64:104043. doi: 10.1016/j.jlp.2020.104043

Han, D. R., Qi, X. D., Myhrvold, C., Wang, B., Dai, M. J., Jiang, S. X., et al. (2017). Single-stranded DNA and RNA origami. *Science* 358:eaao2648.

Heidari, A. (2014). An investigation of the role of DNA as molecular computers: a computational study on the Hamiltonian path problem. *Int. J. Sci. Eng. Res.* 5, 1884–1889.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., and Chen, H. (2019). Harris hawks optimization: algorithm and applications. *Future Gener. Comput. Syst.* 97, 849–872. doi: 10.1016/j.future.2019.02.028

Li, S., Jiang, Q., Liu, S., Zhang, Y., Tian, Y. H., Song, C., et al. (2018). A DNA nanorobot functions as a cancer therapeutic in response to a molecular trigger in vivo. *Nat. Biotechnol.* 36, 258–264. doi: 10.1038/nbt.4071

Li, X., Wang, B., Lv, H., Yin, Q., Zhang, Q., and Wei, X. (2020). Constraining DNA sequences with a triplet-bases unpaired. *IEEE Trans. Nanobiosci.* 19, 299–307. doi: 10.1109/tnb.2020.2971644

Mirjalili, S. (2015). Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* 89, 228–249. doi: 10.1016/j.knosys.2015.07.006

Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* 96, 120–133. doi: 10.1016/j.knosys.2015.12.022

Mirjalili, S., and Lewis, A. (2016). The whale optimization algorithm. *Adv. Eng. Softw.* 95, 51–67.

Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. *Adv. Eng. Softw.* 69, 46–61.

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.* 36:645. doi: 10.1038/nbt.4173

Rao, R. V., Savsani, V. J., and Vakharia, D. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* 43, 303–315. doi: 10.1016/j.cad.2010.12.015

Rinaudo, K., Bleris, L., Maddamsetti, R., Subramanian, S., Weiss, R., and Benenson, Y. (2007). A universal RNAi-based logic evaluator that operates in mammalian cells. *Nat. Biotechnol.* 25, 795–801. doi: 10.1038/nbt1307

Schickinger, M., Zacharias, M., and Dietz, H. (2018). Tethered multifluorophore motion reveals equilibrium transition kinetics of single DNA double helices. *Proc. Natl. Acad. Sci. U.S.A.* 115, E7512–E7521.

Shan, L., Qiang, H., Li, J., and Wang, Z.-Q. (2005). Chaotic optimization algorithm based on Tent map. *Control Decis.* 20, 179–182.

Shendure, J., Balasubramanian, S., Church, G. M., Gilbert, W., Rogers, J., Schloss, J. A., et al. (2017). DNA sequencing at 40: past, present and future. *Nature* 550, 345–353. doi: 10.1038/nature24286

Shin, S. Y., Lee, I. H., Kim, D., and Zhang, B. T. (2005). Multiobjective evolutionary optimization of DNA sequences for reliable DNA computing. *IEEE Trans. Evol. Comput.* 9, 143–158. doi: 10.1109/tevc.2005.844166

Shipman, S. L., Nivala, J., Macklis, J. D., and Church, G. M. (2017). CRISPR-Cas encoding of a digital movie into the genomes of a population of living bacteria. *Nature* 547:345. doi: 10.1038/nature23017

Song, T., Pan, L., Wu, T., Zheng, P., Wong, M. D., and Rodríguez-Patón, A. (2019). Spiking neural P systems with learning functions. *IEEE Trans. NanoBioscience* 18, 176–190. doi: 10.1109/tnb.2019.2896981

Sze, M. A., and Schloss, P. D. (2019). The impact of DNA polymerase and number of rounds of amplification in PCR on 16S rRNA gene sequence data. *Msphere* 4, e00163–19.

Thubagere, A. J., Li, W., Johnson, R. F., Chen, Z., Doroudi, S., Lee, Y. L., et al. (2017). A cargo-sorting DNA robot. *Science* 357:eaan6558. doi: 10.1126/science.aan6558

Wang, Y., Jaime-Lara, R., Roy, A., Sun, Y., Liu, X., and Joseph, P. V. (2020). SeqEnhDL: sequence-based classification of cell type-specific enhancers using deep learning models. *bioRxiv* [Preprint] doi: 10.1101/2020.05.13.093997

Wang, Y., Noor-A-Rahim, M., Gunawan, E., Guan, Y. L., and Poh, C. L. (2019). Construction of bio-constrained code for DNA data storage. *IEEE Commun. Lett.* 23, 963–966. doi: 10.1109/lcomm.2019.2912572

Wang, Y., Shen, Y., Zhang, X., Cui, G., and Sun, J. (2018). An improved non-dominated sorting genetic algorithm-II (INSGA-II) applied to the design of DNA codewords. *Math. Comput. Simul.* 151, 131–139. doi: 10.1016/j.matcom.2018.03.011

Yang, G., Wang, B., Zheng, X., Zhou, C., and Zhang, Q. (2017). IWO algorithm based on niche crowding for DNA sequence design. *Interdiscip. Sci.* 9, 341–349. doi: 10.1007/s12539-016-0160-0

Yang, X.-S., Karamanoglu, M., and He, X. (2014). Flower pollination algorithm: a novel approach for multiobjective optimization. *Eng. Optim.* 46, 1222–1237. doi: 10.1080/0305215x.2013.832237

Yin, Q., Cao, B., Li, X., Wang, B., Zhang, Q., and Wei, X. (2020). An intelligent optimization algorithm for constructing a DNA storage code: NOL-HHO. *Int. J. Mol. Sci.* 21:2191. doi: 10.3390/ijms21062191

Zeng, X., Lin, Y., He, Y., Lu, L., Min, X., and Rodriguez-Paton, A. (2020a). Deep collaborative filtering for prediction of disease genes. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 17, 1639–1647.

Zeng, X., Zhong, Y., Lin, W., and Zou, Q. (2020b). Predicting disease-associated circular RNAs using deep forests combined with positive-unlabeled learning methods. *Brief. Bioinform.* 21, 1425–1436. doi: 10.1093/bib/bbz080

Zgarbova, M., Otyepka, M., Sponer, J., Lankas, F., and Jurecka, P. (2014). Base pair fraying in molecular dynamics simulations of DNA and RNA. *J. Chem. Theory Comput.* 10, 3177–3189. doi: 10.1021/ct500120v

Zhang, S., Huang, B., Song, X., Zhang, T., Wang, H., and Liu, Y. (2019). A high storage density strategy for digital information based on synthetic DNA. *3 Biotech* 9:342.

Zhang, X., Zhang, Q., Liu, Y., Wang, B., and Zhou, S. (2020). A molecular device: a DNA molecular lock driven by the nicking enzymes. *Comput. Struct. Biotec.* 18, 2107–2116. doi: 10.1016/j.csbj.2020.08.004

Zhao, W., Zhang, Z., and Wang, L. (2020). Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. *Eng. Appl. Artif. Intell.* 87:103300. doi: 10.1016/j.engappai.2019.103300

Zhou, S., He, P., and Kasabov, N. (2020). A dynamic DNA Color image encryption method based on SHA-512. *Entropy* 22:1091. doi: 10.3390/e22101091

Zhu, E., Jiang, F., Liu, C., and Xu, J. (2020). Partition independent set and reduction based approach for partition coloring problem. *IEEE T. Cybern.* doi: 10.1109/TCYB.2020.3025819

Zou, Q., and Liu, Q. (2019). Advanced machine learning techniques for bioinformatics. *IEEE Comput. Archit. Lett.* 16, 1182–1183. doi: 10.1109/tcbb.2019.2919039

Zou, Q., Lin, G., Jiang, X., Liu, X., and Zeng, X. (2020). Sequence clustering in bioinformatics: an empirical study. *Brief. Bioinform.* 21, 1–10.

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.