applied sciences

MDPI

Article

An Improved Marker Code Scheme Based on Nucleotide Bases for DNA Data Storage

Jian Tong, Guojun Han * and Yi Sun

School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China; tjian.stu@foxmail.com (J.T.)
* Correspondence: gjhan@gdut.edu.cn

Abstract: Due to the rapid growth in the global volume of data, deoxyribonucleic acid (DNA) data storage has emerged. Error correction in DNA data storage is a key part of this storage technology. In this paper, an improved marker code scheme is proposed to correct insertion, deletion, and substitution errors in deoxyribonucleic acid (DNA) data storage. To correct synchronization (i.e., insertion and deletion) errors, a novel base-symbol-based synchronization algorithm is proposed and used. In the improved scheme, the marker bits are encoded as the information part of the LDPC code, and then mapped into marker bases to correct the synchronization errors. Thus marker bits not only assist in regaining synchronization, but also play a role in LDPC decoding to improve decoding performance. An improved low-complexity normalized min-sum (INMS) algorithm is proposed to correct residual substitution errors after regaining synchronization. The simulation results demonstrate that the improved scheme provides a substantial performance improvement over the concatenated marker code scheme and concatenated watermark code scheme. At the same time, the complexity of the INMS algorithm was reduced, while its bit error rate (BER) performance was approximate to that of the belief propagation (BP) algorithm.

Keywords: DNA data storage; insertion/deletion; low-density parity-check (LDPC) codes; marker codes; synchronization

Citation: Tong, J.; Han, G.; Sun, Y. An Improved Marker Code Scheme Based on Nucleotide Bases for DNA Data Storage. *Appl. Sci.* **2023**, *13*, 3632. https://doi.org/10.3390/app13063632

Academic Editor:Takahito Ohshiro

Received: 15 February 2023
Revised: 7 March 2023
Accepted: 7 March 2023
Published: 12 March 2023

1. Introduction

In the Big Data era, a huge amount of data is generated every day. It is predicted that the global data volume will reach 175 ZB by 2025 [1,2], and the data volume will exceed the storage capacity. Therefore, there is an urgent need to find new storage media. The possibility of storing data in DNA was first proposed by Clelland in 1999 [3]. Due to its durability and ultrahigh density, DNA is a great potential storage medium. One gram of DNA can store 215 GB of data, and the data stored in DNA can be preserved for tens of thousands of years [4]. Therefore, DNA data storage has attracted considerable attention from the data storage community in recent years, especially in the area of cold data storage, such as government documents and historical archives [5].

Channel error correction is a key part of DNA storage technology. In DNA synthesis and sequencing, insertion, deletion, and substitution errors of nucleotide bases occur [6]. Since DNA molecules use four nucleotide bases (i.e., adenine (A), thymine (T), cytosine (C), and guanine (G)) to store genetic information [7], DNA is a quadratic channel corrupted by insertions, deletions, and substitutions. To correct synchronization (i.e., insertion and deletion) errors, Matthew C. Davey proposed the concatenated watermark code scheme [8], and Daniel Marco proposed the concatenated marker code scheme [9]. To correct DNA channel errors, a modified concatenated watermark code scheme and a modified concatenated marker code scheme were proposed in [10,11], respectively. The former uses watermark codes as inner codes and low-density parity-check (LDPC) codes [12] as outer codes. The latter employs the marker codes as inner codes and LDPC codes as outer codes.

In these schemes, watermark and marker codes only assist in regaining synchronization, and they play no role in LDPC decoding. Therefore, these schemes have the disadvantage in error correction performance.To improve error correction performance, we designed an improved error correction scheme for DNA data storage.

In Ref. [13], a novel marker code scheme, called the embedded marker code scheme, was proposed for channels corrupted by insertions, deletions, and additive white Gaussian noise (AWGN), which are binary channels. On the basis of the embedded marker code scheme, we propose an improved scheme for DNA channels. In the improved scheme, marker codes are used as inner codes and LDPC codes are used as outer codes. Marker bits are encoded as part of the information bits. Compared to the concatenated marker code scheme, marker bits in the proposed scheme additionally assist in LDPC decoding. To correct synchronization errors, a base maximal a posteriori (MAP) detector was adopted. Correspondingly, a novel base-symbol-based synchronization algorithm is provided for the detector. In LDPC decoding, we propose an improved normalized min–sum (INMS) algorithm to reduce decoding complexity. There were some biological constraints on DNA synthesis and sequencing that needed to be considered in our proposed scheme. The long nucleotide chains cannot be synthesized in modern DNA synthesis technology, and the maximal length of nucleotides is 250 [14]. To maintain stability, the GC content of the synthesized sequence must be close to 50% (i.e., 40∼60%) [15].

Our main contributions are summarized as follows.

- We propose an improved marker code scheme for DNA channels based on the embedded marker code scheme. In this scheme, a novel base-symbol-based synchronization algorithm is proposed to correct synchronization errors.
- In LDPC decoding, we propose an improved decoding algorithm to reduce decoding complexity.

The paper is organized as follows. In Section 2, the DNA channel model is described. In Section 3, related works are introduced. In Section 4, the improved marker code scheme's development is presented. In Section 5, the base-symbol-based synchronization and the low-complexity INMS algorithms are outlined. Section 6 shows the simulation results. Lastly, Section 7 concludes the paper.

2. DNA Channel Model

Insertion and deletion errors occur mainly during DNA synthesis, and substitution errors occur mainly in DNA sequencing [16]. Thus, the error probability is influenced by synthesis and sequencing techniques. Nanopore and Illumina sequencing are advanced sequencing methods. Compared with nanopore sequencing, Illumina sequencing has the advantage of high reading accuracy, i.e., Illumina sequencing has lower probabilities of base insertion, deletion, and substitution errors. In this paper, we used a channel model on the basis of Illumina sequencing [17], as shown in Figure 1.

In Figure 1, the channel input is a nucleotide base (corresponding to s). Let the maximal insertion length be I_m. With probability P_i, the insertion event occurs, i.e., one or two bases are inserted in the input base where the length of the inserted bases obeys a geometric distribution. With probability P_d, the deletion event occurs, which means that the input base is deleted. With probability $P_t = 1 - P_i - P_d$, the input base is transmitted, i.e., no insertion or deletion event occurs, but the substitution error may occur according to the transfer probabilities among bases. Let γ be the parameter related to the sequencing; then, $p_1 = \gamma$ and $p_2 = 1.5\gamma$ are satisfied, where $\gamma \in (0, 2/3)$ [17]. The substitution error probabilities for T and G are higher than those for A and C. In a word, the input base may be corrupted by insertion, deletion, and substitution errors. Therefore, the channel output is zero, one, or two nucleotide base(s) (corresponding to r in Figure 1). Generally, input data are a base sequence consisting of many bases, and the bases enter the channel one by one.

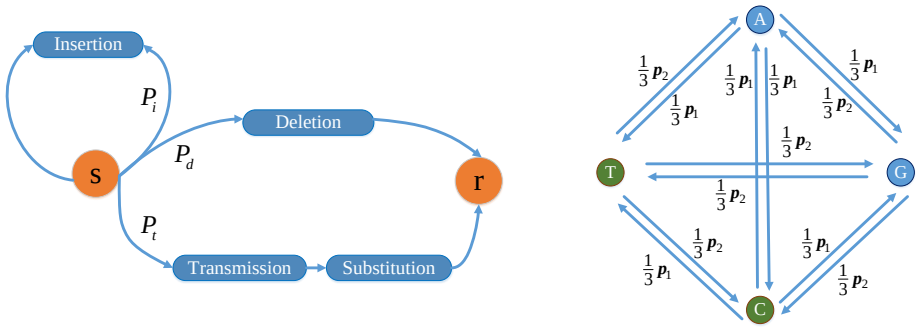

Figure 1. DNA channel model based on Illumina sequencing. s, input base; r, output base.

3. Related Works

In the latest related works [10,11], two error correction schemes were used to correct insertion, deletion, and substitution errors in DNA channels, including the concatenated watermark code [10] and the concatenated marker code scheme [11]. We introduce these schemes in this section.

3.1. Concatenated Watermark Code Scheme

The procedure of the concatenated watermark code scheme is illustrated in Figure 2a. The LDPC code was sparsified to obtain sparse code word s (in which 0 is much more than 1); then, s was superposed with the watermark code to obtain the bias sequence. The watermark code, which is a fixed pseudorandom sequence, is known to the receiver. The principle of this scheme is that the receiver exploits a known watermark code to regain synchronization for error correction. More details about encoding and decoding are in [10].

The storage efficiency of this scheme can be expressed as follows:

$$\eta_1 = R_L R_S, \tag{1}$$

where R_L denotes the code rate of the LDPC code, and R_S denotes the sparsification efficiency. For example, if 4 bits are converted into 5 bits in the sparsification, then $R_S = 0.8$.

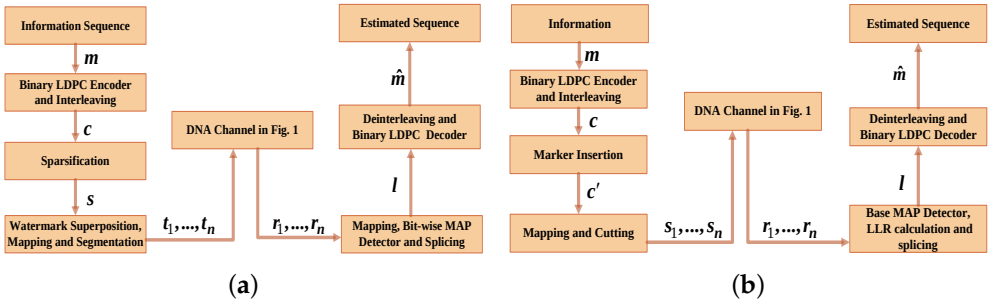

Figure 2. The procedure of the concatenated watermark code scheme and the concatenated marker code scheme. (**a**) Concatenated watermark code scheme; (**b**) Concatenated marker code scheme.

3.2. Concatenated Marker Code Scheme

The procedure of the concatenated marker code scheme is shown in Figure 2b. In this scheme, marker codes (known by the receiver) are inserted uniformly (i.e., equally spaced between any two marker codes) in LDPC code word c. Information-carrying sections of the code are interspersed with synchronization-providing marker sections. Similar to the watermark code, the marker code serves to regain synchronization in decoding. In fact, marker codes can be considered irregular watermark codes [8]. Compared with concatenated watermark code schemes, concatenated marker code schemes can provide better synchronization performance [8]. More details about this scheme are found in [4].

In this scheme, the storage efficiency can be expressed as follows:

$$\eta_2 = R_L \frac{D_E}{D_M + D_E}, \tag{2}$$

where D_E denotes the interval between two adjacent marker codes, and D_M denotes the length of the marker codes.

4. Improved Marker Code Scheme

The procedure of the improved marker code scheme is shown in Figure 3. The encoding process is shown in Figure 4. First, binary sequence $\boldsymbol{m'}$, including information bits and marker bits, is encoded as the information part of the LDPC code. Then, the LDPC code is interleaved to obtain the codeword $\boldsymbol{c}$. Second, sequence $\boldsymbol{c}$ is arranged to guarantee that the marker bits are uniformly distributed. In other words, D_M marker bits are placed at each interval D_E. Next, the arranged sequence is mapped into a nucleotide sequence and cut into several short nucleotide chains $\boldsymbol{s_1, \ldots, s_n}$ because DNA long chains are difficult to synthesize. Lastly, sequences $\boldsymbol{s_1, \ldots, s_n}$ enter the channel and correspondingly output sequences $\boldsymbol{r_1, \ldots, r_n}$.

Due to synchronization errors in a DNA channel, the first step in decoding is to regain synchronization. This step is performed in the base MAP detector, where a new base-symbol-based synchronization algorithm is used. The details of this algorithm are described in the next section. The base detector outputs several log-likelihood ratio (LLR) sequences; then, these sequences are spliced together to obtain a new LLR sequence, $\boldsymbol{l}$. The LDPC decoder receives sequence $\boldsymbol{l}$ and corrects residual substitution errors. To further improve correction performance, the LDPC code decoder feeds back information to the detector for resynchronization. The error correction performance is improved with the joint decoding between the detector and the LDPC decoder.

Let the code length of the LDPC code be N, and the check bit length of the LDPC code be M. If N can be divided by $D_M + D_E$, the storage efficiency of the improved marker code scheme can be expressed as follows:

$$\eta_3 = \frac{N - M - \left(\frac{N}{D_M + D_E}\right) D_M}{N} \tag{3}$$

$$= R_L - \frac{D_M}{D_M + D_E}. \tag{4}$$

If N cannot divided by $D_M + D_E$, the storage efficiency is expressed as follows:

$$\eta_3 = \frac{N - M - \left(\frac{N}{D_M + D_E} + 1\right) D_M}{N} \tag{5}$$

$$= R_L - \frac{D_M}{D_M + D_E} - \frac{D_M}{N}. \tag{6}$$

Thus, the storage efficiency of this scheme is expressed as follows:

$$\eta_3 = \begin{cases} R_L - \frac{D_M}{D_M + D_E}, & if\ N = 0\ mod(D_M + D_E) \\ R_L - \frac{D_M}{D_M + D_E} - \frac{D_M}{N}, & otherwise \end{cases}. \tag{7}$$

Compared to the concatenated marker code scheme, in the proposed scheme, marker bits are inserted before LDPC encoding, which means that the marker codes are part of the LDPC code. Therefore, marker codes in the proposed scheme not only assist in regaining synchronization, but also improve the decoding performance of the LDPC code.

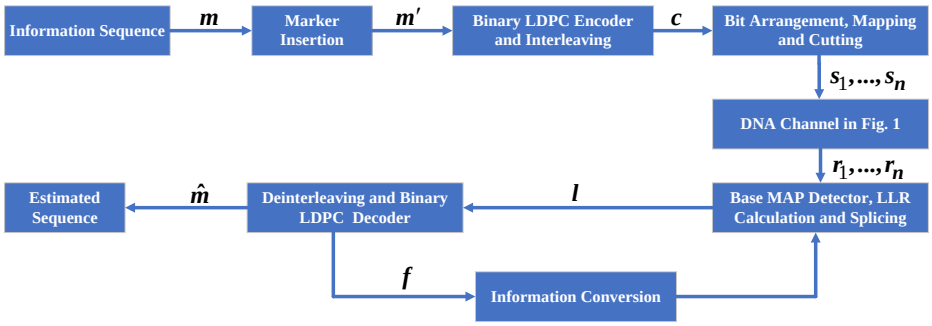

Figure 3. The procedure of the improved marker code scheme.

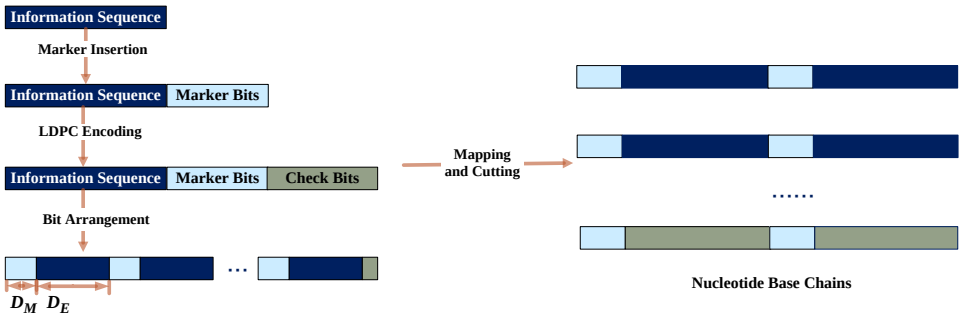

Figure 4. The encoding process of the improved marker code scheme.

5. Decoding Algorithm

5.1. Base-Symbol-Based Synchronization Algorithm

A conventional forward/backward (FB) algorithm is usually used to correct the synchronization errors [8–11]. A novel base-symbol-based synchronization (i.e., FB) algorithm is proposed and used to regain synchronization in the improved scheme.

In this subsection, we describe the algorithm in the case of the maximal insertion length (of the channel) $I_m = 2$. Suppose that channel input sequence $\boldsymbol{s} = \{s_1, s_2, \ldots, s_T\}$ and output sequence $\boldsymbol{r} = \{r_1, r_2, \ldots, r_Q\}$. These are both nucleotide base sequences, but their lengths, T and Q, may be different. Let the position offset be $O_k = n$, $k \in [1, T]$ if the $(k+n)$-th base of the output sequence corresponds to the k-th base of the input sequence. Let $O_{\max}$ be the maximal allowed position offset. Thus, $O_k \in [-O_{\max}, O_{\max}]$. Sequence $\{O_k\}$ forms the hidden states of the hidden Markov model. To express the transfer probabilities among the bases, we define the following function:
if $s_k =$ A or C,

$$F(s_k, r_{k+n}) = \begin{cases} 1-p_1, & if\ r_{k+n} = s_k \\ \frac{1}{3}p_1, & otherwise \end{cases}, \tag{8}$$

if $s_k =$ T or G,

$$F(s_k, r_{k+n}) = \begin{cases} 1-p_2, & if\ r_{k+n} = s_k \\ \frac{1}{3}p_2, & otherwise \end{cases}. \tag{9}$$

As in [8], we define the coefficients as follows:

$$\alpha_k(n) = P(\mathbf{r}_1^{k+n}, O_k = n), \tag{10}$$

$$\beta_k(n) = P(\mathbf{r}_{k+1+n}^{Q} | O_k = n), \tag{11}$$

where $\boldsymbol{r}_1^{k+n} = \{r_1, \ldots, r_{k+n}\}$ and $\boldsymbol{r}_{k+1+n}^{Q} = \{r_{k+1+n}, \ldots, r_Q\}$. These coefficients can be obtained with forward and backward recursion. Suppose $\lambda = 1/(1-P_i^2)$; thus, the transfer probabilities between two adjacent hidden states are P_t, P_d, $\lambda P_i P_d$, $\lambda P_i P_t$, $\lambda P_i^2 P_d$, and $\lambda P_i^2 P_t$.

For example, the transfer probability from state $O_{k-1} = n-2$ to state $O_k = n$ is $\lambda P_i^2 P_t$; thus, the probability of conversion from $\alpha_{k-1}(n-2) = P(r_1, \ldots, r_{(k-1)+(n-2)}, O_{k-1} = n-2)$ to $\alpha_k(n) = P(r_1, \ldots, r_{(k-1)+(n-2)}, r_{(k-1)+(n-1)}, r_{(k-1)+n}, r_{k+n}, O_k = n)$ is $\lambda P_i^2 P_t \cdot \frac{1}{16} \sum_{s_k} P(s_k) F(s_k, r_{k+n})$. Suppose

$$\Phi(x) = \sum_{s_k} P(s_k) F(s_k, x), x \in \{\mathrm{A}, \mathrm{T}, \mathrm{C}, \mathrm{G}\}. \tag{12}$$

The calculations are given as follows.

$$\begin{aligned} \alpha_k(n) = & \frac{\lambda P_i^2 P_t}{16} \Phi(r_{k+n}) \alpha_{k-1}(n-2) \\ & + \left[\frac{\lambda P_i^2 P_d}{16} + \frac{\lambda P_i P_t}{4} \Phi(r_{k+n}) \right] \alpha_{k-1}(n-1) \\ & + \left[\frac{\lambda P_i P_d}{4} + P_t \Phi(r_{k+n}) \right] \alpha_{k-1}(n) \\ & + P_d \alpha_{k-1}(n+1). \end{aligned} \tag{13}$$

$$\begin{aligned} \beta_k(n) = & \frac{\lambda P_i^2 P_t}{16} \Phi(r_{k+1+n}) \beta_{k+1}(n+2) \\ & + \left[\frac{\lambda P_i^2 P_d}{16} + \frac{\lambda P_i P_t}{4} \Phi(r_{k+1+n}) \right] \beta_{k+1}(n+1) \\ & + \left[\frac{\lambda P_i P_d}{4} + P_t \Phi(r_{k+1+n}) \right] \beta_{k+1}(n) \\ & + P_d \beta_{k+1}(n-1). \end{aligned} \tag{14}$$

The probabilities of certain states can be derived after recurring all coefficients. In other words, the a posteriori probabilities of the channel input bases can be obtained and expressed as follows:

$$\begin{aligned} P(s_k|\boldsymbol{r}) = & \frac{\lambda P_i^2 P_t}{16} \sum_{n=-O_{\max}}^{O_{\max}} \alpha_{k-1}(n-2) \beta_k(n) F(s_k, r_{k+n}) \\ & + \frac{\lambda P_i^2 P_d}{16} \sum_{n=-O_{\max}}^{O_{\max}} \alpha_{k-1}(n-1) \beta_k(n) \\ & + \frac{\lambda P_i P_t}{4} \sum_{n=-O_{\max}}^{O_{\max}} \alpha_{k-1}(n-1) \beta_k(n) F(s_k, r_{k+n}) \\ & + \frac{\lambda P_i P_d}{4} \sum_{n=-O_{\max}}^{O_{\max}} \alpha_{k-1}(n) \beta_k(n) \\ & + P_t \sum_{n=-O_{\max}}^{O_{\max}} \alpha_{k-1}(n) \beta_k(n) F(s_k, r_{k+n}) \\ & + P_d \sum_{n=-O_{\max}}^{O_{\max}} \alpha_{k-1}(n+1) \beta_k(n). \end{aligned} \tag{15}$$

The a posteriori probabilities of the bases need to be converted into the soft information of bits (which is the input of the binary LDPC decoder). Since 2 bits are mapped into 1 base, there are 24 mapping rules between base symbols and bits. Suppose that the mapping rule (00-A, 01-C, 10-T, 11-G) is adopted. The first bit is defined as the low bit, and the second bit

is defined as the high bit. Thus, several low and high bits form the channel input sequence. For nonmarker bits, the LLRs of the low bits are calculated as follows:

$$L_l = \frac{P(s_k = \mathrm{T}|\boldsymbol{r}) + P(s_k = \mathrm{G}|\boldsymbol{r})}{P(s_k = \mathrm{A}|\boldsymbol{r}) + P(s_k = \mathrm{C}|\boldsymbol{r})}, \tag{16}$$

The LLRs of the high bits are calculated as follows:

$$L_h = \frac{P(s_k = \mathrm{C}|\boldsymbol{r}) + P(s_k = \mathrm{G}|\boldsymbol{r})}{P(s_k = \mathrm{A}|\boldsymbol{r}) + P(s_k = \mathrm{T}|\boldsymbol{r})}. \tag{17}$$

For marker bits, the LLRs of the low and high bits are set to the larger absolute values because the positions and values of the marker bits are known to the decoder. If the marker bit is 1, the LLR is positive. If the marker bit is 0, the LLR is negative. Through the above method, an LLR sequence is obtained; then, $\boldsymbol{l}$ passes to the decoder. Lastly, the LDPC decoder outputs the estimated sequence $\hat{\boldsymbol{m}}$, as shown in Figure 3.

The joint decoding between decoder and detector is achieved with information feedback. In the LDPC decoder, let $\boldsymbol{l'}$ be the LLR sequence used for hard decisions. Thus, the feedback LLR sequence $\boldsymbol{f} = \boldsymbol{l'} - \boldsymbol{l}$. Suppose that the LLRs of the low bits are f_1', and those of the high bits are f_2'. Thus, the a priori probabilities (which are used for the base MAP detector in the next synchronization) of the nonmarker bases are updated as follows:

$$P(\mathrm{A}) = \frac{1}{1+e^{f_1'}} \times \frac{1}{1+e^{f_2'}}, \tag{18}$$

$$P(\mathrm{C}) = \frac{1}{1+e^{f_1'}} \times \frac{e^{f_2'}}{1+e^{f_2'}}, \tag{19}$$

$$P(\mathrm{T}) = \frac{e^{f_1'}}{1+e^{f_1'}} \times \frac{1}{1+e^{f_2'}}, \tag{20}$$

$$P(\mathrm{G}) = \frac{e^{f_1'}}{1+e^{f_1'}} \times \frac{e^{f_2'}}{1+e^{f_2'}}. \tag{21}$$

The a priori probabilities of the marker bits are determined and fixed to large absolute values in decoding (i.e., they are unchanged even at resynchronization).

Every time the detector is used, the number of synchronizations is increased by 1. There is an upper limit to the number of synchronizations (joint decoding) that is predetermined. Let this upper limit be S_m. In the first synchronization, the a priori probabilities of the nonmarker bases used by the detector are initialized to $P(\mathrm{A}) = 0.25$, $P(\mathrm{C}) = 0.25$, $P(\mathrm{T}) = 0.25$, $P(\mathrm{G}) = 0.25$. In the second and subsequent synchronizations, the a priori probabilities of the nonmarker bases are calculated with (18)–(21).

5.2. Low-Complexity INMS

5.2.1. INMS

In related works, the belief propagation (BP) algorithm was used to correct residual substitution errors after regaining synchronization [10,11]. However, the BP algorithm has high complexity [18]. To reduce decoding complexity, we propose an improved decoding algorithm for our proposed scheme called INMS.

Notations in the algorithm development are as follows: k denotes the length of the information bits, n denotes code length, σ denotes the correction factor, L_j denotes the initial LLR of the j-th bit, L_j^{tot} denotes the total LLR, $I_{\max}$ denotes the maximal number of the iterations, M denotes the set of all marker bits, $N_c(j)$ denotes the set of all check nodes connected with the j-th bit, $N_v(i)$ denotes the set of all variable nodes connected with the i-th check node, $N_c(j)/i$ denotes $N_c(j)$ except for the i-th check node, and $N_v(i)/j$ denotes $N_v(i)$ except for the j-th variable node. The detailed procedure of the INMS algorithm is given in Algorithm 1.

In Algorithm 1, marker bits are fixed; thus, their $L_{c\to v}$, $L_{v\to c}$, and L^{tot} are not calculated because the marker bits are known to the decoder. Therefore, marker bit nodes reduce decoding complexity while providing reliable information to other nodes.

Algorithm 1 INMS for the binary LDPC code decoder.

Initialize:
for $1 \le j \le n, i \in N_c(j)$ **do**
  $L_j = l_j, L_{v\to c}(i,j) = l_j$
end for
for $iter = 1 : I_{\max}$ **do**
  Stop if $\hat{c}H^T = 0$ or $iter = I_{\max}$
Check node update:
  for $1 \le i \le n-k, j \in N_v(i)$ **do**
    if $j \notin M$ **then**
      $L_{c\to v}(i,j) = \sigma \times \min_{j' \in N_v(i)/j} |L_{v\to c}(i,j')| \times \prod_{j' \in N_v(i)/j} \mathrm{sgn}(L_{v\to c}(i,j'))$
    else $L_{c\to v}(i,j) = 0$
    end if
  end for
Variable node update:
  for $1 \le j \le n, i \in N_c(j)$ **do**
    if $j \notin M$ **then**
      $L_{v\to c}(i,j) = L_j + \sum_{i' \in N_c(j)/i} L_{c\to v}(i,j')$
    else $L_{v\to c}(i,j) = L_j$
    end if
  end for
Hard decision:
  for $j = 1 : n$ **do**
    if $j \notin M$ **then**
      $L_j^{tot} = L_j + \sum_{i \in N_c(j)} L_{c\to v}(i,j)$
    else $L_j^{tot} = L_j$
      $\hat{c}_j = \begin{cases} 1, \text{ if } L_j^{tot} \ge 0 \\ 0, \text{ else} \end{cases}$
    end if
  end for
end for

5.2.2. Algorithm Complexity

In this subsubsection, we give the quantitative analysis of algorithm complexity. A regular LDPC code was used for the analysis that had a length of N, a code rate of 0.5, a row weight of R, and a column weight of C. We defined the complexity of an addition operation as ρ, the complexity of a multiplication operation as ω, and the complexity of a table lookup operation as δ. Table 1 shows the complexity of the BP, NMS (i.e., normalized min-sum algorithm), and INMS algorithms. The INMS algorithm was divided into four steps: syndrome calculation, check node update, variable node update, and verdict LLR update. In syndrome calculation, three algorithms have the same complexity, $(2R-1)N\rho/2$. In the check node update, the INMS algorithm reduces the table lookup operation compared to the BP algorithm, and reduces the complexity of $CM\omega$ compared to the NMS algorithm. In variable node update, the complexity of the INMS algorithm is reduced by $C(C-1)M\rho$ compared to that with the BP and NMS algorithms. In verdict LLR update, the complexity of the INMS algorithm is reduced by $CM\rho$ compared to that with the BP and NMS algorithms. Compared to the BP algorithm, the total complexity of the

INMS algorithm is reduced by $C^2M\rho + (0.5R^2N - 1.5RN + CM)\omega + 0.5R^2N\delta$. Therefore, the INMS algorithm could achieve improved complexity.

Table 1. Algorithm complexity comparison.

	BP	**NMS**	**INMS**
Syndrome calculation	$(2R-1)N\rho/2$	$(2R-1)N\rho/2$	$(2R-1)N\rho/2$
Check node update	$R(R-1)N\omega/2 + R^2N\delta/2$	$RN\omega$	$(RN-CM)\omega$
Variable node update	$C(C-1)N\rho$	$C(C-1)N\rho$	$C(C-1)(N-M)\rho$
Verdict LLR update	$CN\rho$	$CN\rho$	$C(N-M)\rho$

6. Simulation Results

In this section, we compared the error correction performance of the proposed scheme, concatenated watermark code scheme, and concatenated marker code scheme. Moreover, we compared the decoding performance of the proposed INMS algorithm to that of the flooding BP algorithm. We used Microsoft Visual Studio 2019 as the simulation platform.

To ensure fairness, three schemes were compared at equal levels of storage efficiency. At any same storage efficiency, the comparison was valid. In our simulations, we compared the bit error rate (BER) performance of the schemes at a storage efficiency of 0.72 (which was determined with LDPC code rates R_L, D_M, and D_E). A quasicyclic LDPC code with code length of 4544 and code rate of 0.9 was used. In the proposed scheme, suppose that $D_M = 4$ and $D_E = 18$ (which makes the storage efficiency be 0.72). To guarantee the balance of GC content, {AC} and {TG} were selected as the marker bases. In other words, the nucleotide base sequences contained two types of markers, {AC} and {TG}. Since it is difficult to synthesize long chains exceeding 250 nucleotide bases [14], the long sequence was cut into several segments with a length of 110 bases. We assumed that the maximal number of LDPC decoding iterations $I_{max} = 30$, and the maximal allowed position offset Omax = 80 (according to the length of the segments).

Figure 5a shows the variation in BER with probabilities $P_i + P_d$, at $\gamma = 0.002$. Figure 5b shows the variation in BER with parameter γ related to substitution errors at $P_i = 0.0005$ and $P_d = 0.0065$. Figure 5 shows that the BP algorithm was used in all three schemes. Figure 5 demonstrates that the proposed scheme could achieve lower BER than those of the concatenated marker code scheme and concatenated watermark code scheme in the DNA channel. For example, in Figure 5b, at $\gamma = 0.004$, the BER of the proposed scheme was about 0.003 lower than that of the concatenated marker code scheme and about 0.025 lower than that of the concatenated watermark code scheme.

Figure 6a shows the variation in the BER with probabilities $P_i + P_d$, at $\gamma = 0.002$. Figure 6b shows the the variation in the BER with parameter γ related to the substitution errors at $P_i = 0.0005$ and $P_d = 0.0065$. Figure 6 shows a comparison of the performance of the INMS and BP algorithms in the proposed scheme, and the BER performance of the INMS algorithm was approximate to that of the BP algorithm, which verified the effectiveness of the INMS algorithm. In Figure 6b, the BER of the INMS algorithm was higher than that of the BP algorithm when $\gamma > 0.003$ because the INMS algorithm used approximate replacement in the check node update and thus lost some performance. However, the BER of the INMS algorithm was lower than that of the BP algorithm when $\gamma < 0.003$ because the INMS algorithm achieved a better reduction in the correlation effect among the messages after several iterations. A similar case appeared in [19]. The performance of the INMS algorithm was approximate to that of the BP algorithm in all cases. An appropriate correction factor was selected for the INMS algorithm to obtain the above results. Here, $\sigma = 0.45$, which was obtained with numerous simulations.

Combining Figures 5 and 6 obviously shows that BER performance could be significantly improved with several synchronizations. In Figure 5a, for example, at $P_i + P_d = 0.005$, the BER at the maximal number of synchronizations $Sm = 3$ was reduced by almost 10 times compared to the BER at $Sm = 1$. At the same time, several synchronizations (i.e., joint

decoding) also led to an increase in complexity; thus, we had to consider both performance and complexity in the proposed scheme, and select a suitable Sm.

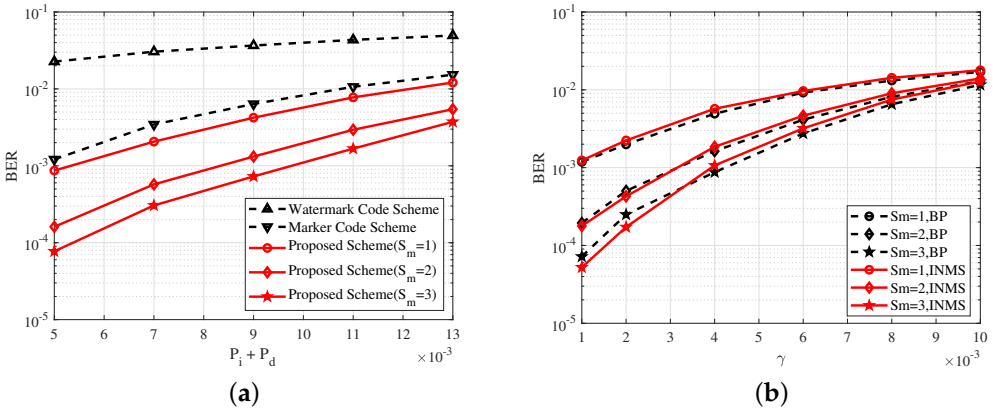

Figure 5. (**a**) BER performance of the concatenated watermark code scheme, the concatenated marker code scheme, and the proposed scheme at $\gamma = 0.002$. (**b**) BER performance of the watermark code scheme and the proposed scheme at $P_i = 0.0005$ and $P_d = 0.0065$. S_m denotes the maximal number of synchronizations.

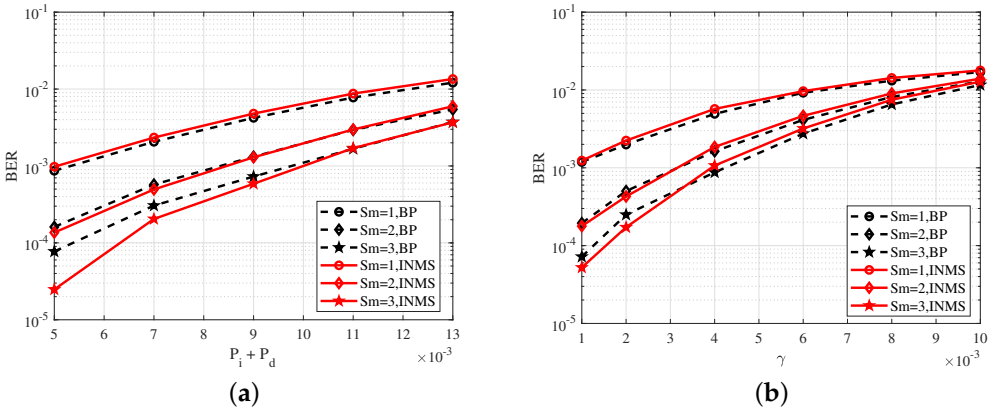

Figure 6. (**a**) BER performance of the INMS and BP algorithms at $\gamma = 0.002$. (**b**) BER performance of the INMS and BP algorithms at $P_i = 0.0005$ and $P_d = 0.0065$. The S_m denotes the maximal number of synchronizations.

7. Conclusions

In this paper, we proposed an improved marker code scheme for DNA channels. In this scheme, the marker bits are encoded as the information part of the LDPC code and then mapped into marker bases to correct the synchronization errors. Thus, marker bits not only assist in regaining synchronization, but also play a role in LDPC decoding to improve decoding performance. On the basis of the fact that DNA is a quadratic channel, a novel base-symbol-based synchronization algorithm is proposed. The simulation results demonstrate that the improved scheme achieved substantial performance improvement over the concatenated marker code scheme and concatenated watermark code scheme. Moreover, the INMS algorithm was proposed to reduced decoding complexity in the proposed scheme. The BER performance of this algorithm was approximate to that of the BP algorithm, but its complexity was significantly reduced. The simulation results demonstrated the effectiveness of the INMS algorithm.

Author Contributions: Conceptualization, methodology and original draft preparation, J.T.; supervision, review and editing, G.H.; validation, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Siddiqa, A.; Karim, A.; Gani, A. Big data storage technologies: A survey. *Front. Inf. Technol. Electron. Eng.* **2017**, *18*, 1040–1070. [CrossRef]
2. Tazeen, N.; Sandhya, K. A Survey on Some Big Data Applications Tools and Technologies. *Int. J. Recent Technol. Eng.* **2021**, *9*, 239–242. [CrossRef]
3. Clelland, C.; Risca, V.; Bancroft, C. Hiding messages in DNA microdots. *Nature* **1999**, *399*, 533–534. [CrossRef] [PubMed]
4. Taluja, S.; Bhupal, J.; Krishnan, S.R. A Survey Paper on DNA-Based Data Storage. In Proceedings of the 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 24–25 February 2020; pp. 1–4.
5. Bornholt, J.; Lopez, R.; Carmean, D.M.; Ceze, L.; Seelig, G.; Strauss, K. A DNA-based archival storage system. *Comput. Archit. News* **2016**, *44*, 637–649. [CrossRef]
6. Chandak, S.; Tatwawadi, K.; Lau, B-; Mardia, J.; Kubit, M.; Neu, J.; Griffin, P.; Wootters, M.; Weissman, T.; Ji, H. Improved read/write cost tradeoff in DNA-based data storage using LDPC codes. In Proceedings of the 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 24–27 September 2019; pp. 147–156.
7. Shomorony, R.H. DNA-Based Storage: Models and Fundamental Limits. *IEEE Trans. Inf. Theory* **2021**, *67*, 3675–3689. [CrossRef]
8. Davey, M.C.; Mackay, D.J.C. Reliable communication over channels with insertions, deletions, and substitutions. *IEEE Trans. Inf. Theory* **2001**, *47*, 687–698. [CrossRef]
9. Ratzer, E.A. Marker codes for channels with insertions and deletions. *Ann. Telecommun.* **2005**, *60*, 29–44. [CrossRef]
10. Weigang, C.; Mingzhe, H.; Jianting, Z.; Qi, G.; Panpan, W.; Xinchen, Z.; Siyu, Z.; Lifu, S.; Yingjin, Y. An artificial chromosome for data storage. *Natl. Sci. Rev.* **2021**, *8*, nwab028.
11. Nakata, R.; Kaneko, H. Synchronization and Asymmetric Error Correction for Nanopore Sequencing. In Proceedings of the IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Penghu, Taiwan, 16–18 June 2021; pp. 1–2.
12. Gallager, R. Low-density parity-check codes. *IRE Trans. Inf. Theory* **1962**, *8*, 21–28. [CrossRef]
13. Han, G.; Guan, Y.L.; Cai, K.; Chan, K.S.; Kong, L. Embedded Marker Code for Channels Corrupted by Insertions, Deletions, and AWGN. *IEEE Trans. Magn.* **2013**, *49*, 2535–2538. [CrossRef]
14. Ma, S.; Tang, N.; Tian, J. DNA synthesis, assembly and applications in synthetic biology. *Curr. Opin. Chem. Biol.* **2012**, *16*, 260–267. [CrossRef] [PubMed]
15. Ross, M.G.; Russ, C.; Costello, M. Characterizing and measuring bias in sequence data. *Genome Biol.* **2013**, *14*, R51. [CrossRef] [PubMed]
16. Organick, L.; Ang, S.D.; Chen, Y.J.; Lopez, R.; Yekhanin, S.; Makarychev, K.; Strauss, K.; Racz, M.Z.; Seelig, G.; Ceze, L.; et al. Random access in large-scale DNA data storage. *Nat. Biotechnol.* **2018**, *36*, 242–248. [CrossRef] [PubMed]
17. Deng, L.; Wang, Y.; Noor-A-Rahim, M.D.; Guan, Y.L.; Shi, Z.; Gunawan, E.; Poh, C.L. Optimized Code Design for Constrained DNA Data Storage With Asymmetric Errors. *IEEE Access* **2019**, *7*, 84107–84121. [CrossRef]
18. Roberts, M.K.; Sunny, E. Investigations on performance analysis of various soft decision based LDPC decoding algorithms. In Proceedings of the 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 23–24 November 2017; pp. 175–179.
19. Jinghu, C.; Dholakia, A.; Eleftheriou, E.; Fossorier, M.P.C.; Hu, X.-Y. Reduced-Complexity Decoding of LDPC Codes. *IEEE Trans. Commun.* **2005**, *53*, 1288–1299.