Error Rate-Based Log-Likelihood Ratio Processing for Low-Density Parity-Check Codes in DNA Storage

XIAOZHOU LU¹, JAEHO JEONG², (Graduate Student Member, IEEE), JAE-WON KIM², (Member, IEEE), JONG-SEON NO², (Fellow, IEEE), HOSUNG PARK³, (Member, IEEE), ALBERT NO⁴, (Member, IEEE), AND SUNGHWAN KIM¹, (Member, IEEE)

¹School of Electrical Engineering, University of Ulsan, Ulsan 44610, South Korea
²Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
³Department of Computer Engineering and Department of ICT Convergence System Engineering, Chonnam National University, Gwangju 61186, South Korea
⁴Department of Electronic and Electrical Engineering, Hongik University, Seoul 04066, South Korea

Corresponding author: Sunghwan Kim (sungkim@ulsan.ac.kr)

This work was supported by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT1802-09.

ABSTRACT Due to the advantages of high storage densities and longevity, DNA storage has become one of the attractive technologies for future data storage systems. However, the writing/reading cost is still high and more efficient techniques for DNA storage are required. In this paper, we propose improved log-likelihood ratio (LLR) processing schemes based on observed statistics for low-density parity-check (LDPC) code decoding to reduce reading cost while encoding schemes are kept unchanged. Due to the mismatch between the real channel and the observed statistics and also the limit of maximum decoder input value, scaling the magnitude of LLR can lead to a better error correcting performance. Therefore, we propose two strategies: 1) directly scaling LLRs and 2) scaling pairwise substitution error rates, which changes the magnitude of LLRs. We also suggest the relation between substitution error rate and scaling values in the strategies by using curve fitting methods. Simulation results show that the error correcting performance from the proposed LLR calculation is better than that from the conventional scheme. Finally, we verify that the proposed LLR methods can be generally applied in DNA storage systems, and present practical methods to calculate error rates.

INDEX TERMS DNA storage, error correction codes, low-density parity-check (LDPC) codes, log-likelihood ratio (LLR), error rate.

I. INTRODUCTION

With the rapid increase in the total amount of data, the demand for data storage is increasing. Since conventional storage requires exponentially increasing costs to store massive amounts of data and is susceptible to damage over a relatively long period, DNA storage has been proposed for high storage densities, longevity in ideal condition, and efficient data duplication based on PCR technology [1]–[3]. In DNA storage, binary data are synthesized into DNA sequences with four-base nucleotides (A, T, G, C) at a certain writing cost [4] and are then stored in a DNA pool. During the reading process, Illumina sequencer reads randomly sampled DNA sequences from the pool [2]. The DNA sequences obtained by the sequencer are regarded as reads [4], [5], which may not be equal to the original sequence due to errors in synthesizing and sequencing processes. A reading cost was defined to evaluate performance, which can be obtained by dividing the total number of bases in the minimum reads required to recover the original DNA sequences by the number of information bits [4].

Currently, the error rates during DNA synthesis and sequencing are high [6]. To address this issue, simple error-correction channel codes [7] such as Reed-Solomon (RS) codes [6], Bose–Chaudhuri–Hocquenghem (BCH) codes [4], and fountain codes [8], have been considered for
DNA storage system. To further improve the error-correcting capability, it is natural to apply the modern coding techniques such as low-density parity-check (LDPC) codes [9], which can achieve the channel capacity limit. Since balanced GC-content (45%–55%) and short homopolymer length (≤3) are essential for DNA synthesis/sequencing [8], Deng et al. proposed variable-length run-length limited codes combined with LDPC codes [10]. Also, Chandak et al. proposed a DNA storage system with LDPC codes [4], where a simple channel model with Poisson distribution was considered. A machine learning based DNA basecaller [11] was also proposed, which uses convolutional code in DNA storage.

For better error-correction capability of LDPC codes, the input of LDPC decoders should be soft information such as log-likelihood ratio (LLR). In conventional radio frequency communication systems, calculation of LLRs or soft information is already known since those channel models such as additive white Gaussian noise (AWGN) channel are given. Unfortunately, there is no research about exact channel models in DNA storage so far, and it is not possible to calculate exact LLR. This situation motivates us to find how to calculate LLR even though we do not know exact channel model in DNA storage.

We present to calculate the ratios by using error statistics. To solve the mismatch between exact channel and error statistics, we propose two scaling methods which are to directly change magnitude of LLR and to change pairwise error rate for adjusting the magnitude of LLRs. Simulation results show that the performance of the proposed methods is better than or equal to that of the conventional method. To verify application of the proposed LLR calculation methods to DNA storage, we suggest an example and results in the example are almost same with those in the simulation results. The main contributions of our work are summarized as below.

- We propose how to calculate LLRs by using error rates on the assumption that there is no exact channel model in DNA storage. Some research [4], [12] started to use LDPC codes in DNA storage but they assumed simple symmetric channels [4] and asymmetric channels [12] which are not exact channel models in DNA storage.
- We propose two scaling methods suitable for decoding of LDPC codes since input range in LDPC decoders depends on the error rate and there is a mismatch between exact channel model and LLR calculation.
- We suggest relation between scaling values in two scaling and the error rates by using curve fitting methods.
- Finally, we suggest a verification example how to apply the proposed LLR calculation to new experiments.

The paper is organized as follows. In Section II, we introduce the system model briefly. In Section III, LLR calculation and two scaling methods are proposed in detail. In Section IV, the results of the comparison between Chandak et al.’s work [4] and our work are presented, and in Section V, we apply two experiments to our proposed strategies with their simulation results and present practical calculation of error rates. We conclude the paper in Section VI.

II. SYSTEM MODEL

The proposed system model of DNA storage is shown in Fig. 1. Since encoding, multiple sequence alignment (MSA), and decoding are similar with those in [4], we briefly explain them in relation to the proposed LLR calculation.

A. DNA BASE MAPPING AND MSA

The encoding scheme for the proposed systems has the following steps [4]. First, the binary file is encoded by large block LDPC codes, which can achieve excellent error correction performance. These binary data are split into fragments with a specific length and mapped to the DNA bases. Two bits (‘00’, ‘01’, ‘10’, and ‘11’) in binary data are mapped to the DNA bases A, C, G, and T, respectively. Then, a synchronization marker [4] can be added to the middle of each fragment to detect deletion or insertion errors on both sides of the marker. Finally, we add an address index protected by BCH code [4] to the front of each sequence to distinguish the randomly sampled reads after sequencing.

According to the consistency of the bases, the MSA [4], [13] is the general alignment of three or more DNA sequences at the same address to great extent. During the MSA processing, some positions of DNA sequences are substituted by blanks, which are regarded as erasures E. Note that there are other recent alignment algorithms such as pairwise sequence alignment and structural alignment. However, these algorithms are not very different from MSA, so that the proposed method can be applied to the DNA storage systems with other alignment algorithms. The most important reason we consider MSA in our work is for fair comparison since the referenced system [4] considered MSA as an alignment method.

B. LDPC DECODING

Since the LDPC decoding in this work is similar to those in previous works [4], [14], we briefly summarize it in this
section. The sum-product algorithm [14] for binary LDPC decoding is introduced by the following steps.

First, the initial LLR for a variable node $v$ is given as

$$M_v^{(0)} = \log \left( \frac{p( x = 0 | y )}{p( x = 1 | y )} \right),$$

where $x$ is a transmitted bit and $y$ is a received signal. Let $M_v^{(l)}$ be the message from a variable node $v$ to a check node $c$ at the $l$-th iteration. Similarly, $M_c^{(l)}$ is the message from a check node $c$ to a variable node $v$ at the $l$-th iteration. For convenience, all $M_v^{(0)}$ values are set to zero, and the number of iterations $l$ is larger than 0.

Second, the message $M_v^{(l)}$ is calculated as

$$M_v^{(l)} = M_v + \sum_{c \in C_v \setminus \{v\}} M_c^{(l-1)},$$

where $C_v$ is a set of check nodes connected to the variable node $v$.

Third, the message $M_c^{(l)}$ is calculated as

$$M_c^{(l)} = 2 \tanh^{-1} \left( \prod_{v' \in V \setminus \{v\}} \tanh \left( \frac{1}{2} M_v^{(l)} \right) \right),$$

where $V_c$ is a set of variable nodes connected to the check node $c$.

The message to estimate a variable node $v$ at the $l$-th iteration is calculated as

$$M_v^{(l)} = M_v + \sum_{c \in C_v} M_c^{(l)},$$

C. CONVENTIONAL LLR CALCULATION

Since LLR calculation depends on the channel model, we briefly introduce the conventional LLR calculation under the binary AWGN channel and binary symmetric channel model with Poisson distribution [4].

Under the binary AWGN channel model, the LLR $L$ is expressed as

$$L = \log \frac{p(x = 0 | y)}{p(x = 1 | y)} = \log \frac{1}{\sqrt{2\pi}\sigma} \exp \left[ -\frac{y^2}{2\sigma^2} \right] = \log \exp \left[ -\frac{(y-1)^2}{2\sigma^2} + \frac{(y-1)^2}{2\sigma^2} \right] = \frac{2y}{\sigma^2},$$

where $\sigma^2$ denotes the variance of the noise.

In [4], it was assumed that reads pass through a binary symmetric channel with error probability $\epsilon$ and an ideal Poisson random sampling model is considered. A memoryless approximation was also considered for the channel. The input and output of the channel are a bit and a tuple $(k_0, k_1)$, respectively, where $k_b$ is the number of times that the bit is read as $b$. Then, the transition probability for the channel is given as

$$p((k_0, k_1) | 0) = \frac{e^{-\lambda}(k_0,k_1)!}{(k_0)!k_1!} \left( \frac{k_0 + k_1}{k_0} \right)(1 - \epsilon)^{k_0}\epsilon^{k_1},$$

$$p((k_0, k_1) | 1) = \frac{e^{-\lambda}(k_0,k_1)!}{(k_0)!k_1!} \left( \frac{k_0 + k_1}{k_1} \right)(1 - \epsilon)^{k_1}\epsilon^{k_0},$$

where $\lambda$ is the ratio of reading cost to writing cost [4]. Then, the LLR information can be calculated as

$$L = \log \frac{p((k_0, k_1) | 0)}{p((k_0, k_1) | 1)} = (k_0 - k_1) \log \frac{1 - \epsilon}{\epsilon}.$$
We assume that the error statistics are provided after alignment for LDPC decoders. Based on occurrence rate is too high, and it makes the LDPC decoding worse.

### III. PROPOSED PREPROCESSING FOR LDPC CODES IN DNA STORAGE

In this section, we first explain the definition of the LLR and LLR calculation method based on the substitution error rate. Next, we explain two scaling strategies of the proposed LLR calculation for LDPC decoders.

#### A. LLR CALCULATION

Note that error statistics are required to compute the LLR. We assume that the error statistics are provided after alignment. Then, we derive a new calculation method for LLRs in LDPC decoding to improve decoding performance based on error statistics.

Let \( x = (x_1, x_2, \ldots, x_n) \) be an encoded DNA codeword of length \( n \) and \( y = (y_1, y_2, \ldots, y_n) \) be a DNA read. Note that \( y \) can include erasure symbols. Let \( b^1 = A, b^2 = C, b^3 = G, b^4 = T, \) and \( b^5 = E \), where \( E \) is an erasure symbol. Then, \( x_j \) is an element in \( \{b^1, b^2, b^3, b^4\} \) and \( y_j \) is an element in \( \{b^1, b^2, b^3, b^4, b^5\} \). For all \( j, 1 \leq j \leq n \), each DNA base symbol is treated as two binary bits denoted by \( x_j = (x_j^1, x_j^2) \). Suppose \( y_j = b^i \); then, the initial LLRs, \( L_j^1 \) and \( L_j^2 \), are given as

\[
L_j^1 = \log \frac{p(x_j^1 = 0 | y_j = b^i)}{p(x_j^1 = 1 | y_j = b^i)} \tag{2}
\]

and

\[
L_j^2 = \log \frac{p(x_j^2 = 0 | y_j = b^i)}{p(x_j^2 = 1 | y_j = b^i)}. \tag{3}
\]

According to Bayes’ theorem, the two LLRs in (2) and (3) can be expressed as

\[
L_j^1 = \log \frac{p(x_j^1 = 0, y_j = b^i) / p(y_j = b^i)}{p(x_j^1 = 1, y_j = b_i) / p(y_j = b^i)} \tag{4}
\]

and

\[
L_j^2 = \log \frac{p(x_j^2 = 0, y_j = b^i) / p(y_j = b^i)}{p(x_j^2 = 1, y_j = b_i) / p(y_j = b^i)}. \tag{5}
\]

Thus, the LLRs in (4) and (5) are expressed as

\[
L_j^1 = \log \frac{p(x_j^1 = 0, y_j = b^i)}{p(x_j^1 = 1, y_j = b_i)} = \log \frac{p(x_j = A, y_j = b^i) + p(x_j = C, y_j = b^i)}{p(x_j = G, y_j = b_i) + p(x_j = T, y_j = b^i)}. \tag{6}
\]

and

\[
L_j^2 = \log \frac{p(x_j^2 = 0, y_j = b^i)}{p(x_j^2 = 1, y_j = b_i)} = \log \frac{p(x_j = A, y_j = b^i) + p(x_j = G, y_j = b^i)}{p(x_j = C, y_j = b_i) + p(x_j = T, y_j = b^i)}. \tag{7}
\]

since two bits ‘00’, ‘01’, ‘10’, and ‘11’ in binary data are mapped to \( A, C, G, \) and \( T \) in DNA symbols. Then, the four terms in (6) and (7) can be generally expressed as \( p(x_j = b^k, y_j = b^i) \). The probability \( p(x_j = b^k, y_j = b^i) \) is calculated as

\[
p(x_j = b^k, y_j = b^i) = \frac{o(x_j = b^k, y_j = b^i)}{N}, \tag{8}
\]

where \( N \) is the number of all aligned reads after MSA and \( o(x_j = b^k, y_j = b^i) \) denotes the number of occurrences that symbols of \( x_j \) and \( y_j \) are \( b^k \) and \( b^i \), respectively.

Then, the average substitution error rate is given as

\[
S_E = \frac{\sum_s \sum_{i,j} o(x_j = b^k, y_j = b^i)}{Nn}. \tag{9}
\]

To calculate exact substitution error rates, we consider the case that the symbol \( x_j \) in (8) is known and total reads are used. These assumptions are not valid in practice. In Section V,
we discuss more on error rate estimation without these ideal assumptions.

During the process to obtain the statistics, we found that there is a mismatch of substitution errors between statistical and theoretical analyses due to the lack of channel models in DNA storage systems. Since insertion and deletion errors exist and the alphabet of binary codeword bits is different from the alphabet of DNA symbols, the substitution error rate must be scaled for LDPC decoding. Moreover, the larger the substitution error, the smaller the LLR, so you need to adjust the LLR to fit the input range of the LDPC decoder. To solve this issue, we propose two scaling strategies, for which the main ideas are described in the following subsection.

**B. FIRST SCALING METHOD FOR LLR**

In Subsections III.B and III.C, we introduce two scaling methods, namely Strategy 1 (S1) and Strategy 2 (S2), respectively. To compare these methods with the conventional schemes, we consider eight experiments in [4]. In [4], Experiments 1–2 and Experiment 8 have payload lengths of 164 bits and 174 bits without a marker, which is a sequence ‘ACT’ at the center of each payload, respectively. Experiments 3–5, Experiment 6, and Experiment 7 have payload lengths of 174 bits, 168 bits, and 180 bits, respectively, including a 6-bit marker. In these eight experiments, we first find an optimal linear scaling strategy to minimize the number of reads required to calculate the reading cost. Next, we explain the relation between substitution error rate \( S_E \) and the scaling coefficients.

From the LLRs \( L_j^1 \) and \( L_j^2 \) in (6) and (7), we define the first scaled LLRs \( L_j^{1\prime} \) and \( L_j^{2\prime} \) as

\[
L_j^{1\prime} = \alpha L_j^1 \\
L_j^{2\prime} = \alpha L_j^2,
\]

where \( \alpha \) is a scaling coefficient. When \( 0 \leq \alpha \leq 1 \), this scaling makes the magnitude of the LLRs smaller. The values of \( S_E \) and the best scaling coefficient \( \alpha \) for each experiment are listed in Table 1. ‘Exp.’ in Table 1 means Experiment. According to Table 1, as \( S_E \) increases, \( \alpha \) also increases. To interpret the relation between \( S_E \) and \( \alpha \), we consider curve fitting methods.

LLR values in eight experiments imply that we need to scale down the LLR values. To find \( \alpha \), we increase \( \alpha \) by 0.01 from 0 to 1 and check whether all 30 trials are successful with the minimum number of reads. Finally, we use the median of such \( \alpha \) values since the median provides robust results. Moreover, these \( \alpha \) values can provide margin to determine new \( \alpha \) for new experiments, which will be discussed in Section V.

To determine the optimal curve fitting between scaling coefficient \( \alpha \) and \( S_E \), we use the Curve Fitting Toolbox in MATLAB. In our work, we define \( S_E \) as the independent variable and \( \alpha \) as a variable dependent on \( S_E \). The most important term in curve fitting is \( R^2 \), which is a statistical measure that indicates how much variation of a dependent variable (e.g., \( S_E \)) is changed by independent variables (e.g., \( S_E \)) in a regression model. The formula for \( R^2 \) is defined as \( R^2 = 1 - \frac{\text{sum of squares of residuals}}{\text{total variation of } S_E} \). If the value of \( R^2 \) is near 1, it shows that \( S_E \) is strongly correlated with \( \alpha \). Since there are various curve fitting results between \( S_E \) and \( \alpha \), we choose the curve fitting method resulting in the largest \( R^2 \). The best curve fitting method with the associated equation and parameters is shown in Fig. 5.

The best fit is the linear one-term curve, which can be expressed as the equation \( f(x) = ax + c \), where independent variable \( x \) is the substitution error rate \( S_E \) and the dependent variable \( f(x) \) is the scaling coefficient \( \alpha \). From the results of Fig. 5, when new experiment with LDPC codes and substitution error rates is performed, we can assign proper scaling values in LLR calculations. When the substitution error rate is larger than 4.7\%, the coefficient \( \alpha \) is larger than 1, which increases magnitude of LLRs.

While the first scaling is directly applied to the magnitude of the LLRs, the second scaling addresses pairwise substitution error rate.

| Table 1. Relation between substitution error rate \( S_E \) and scaling coefficient \( \alpha \) |
|-----------------------------------------------|
| Exp. 1 | 0.68 | 0.53 |
| Exp. 2 | 0.49 | 0.49 |
| Exp. 3 | 1.51 | 0.61 |
| Exp. 4 | 1.67 | 0.63 |
| Exp. 5 | 1.86 | 0.66 |
| Exp. 6 | 1.68 | 0.63 |
| Exp. 7 | 3.00 | 0.79 |
| Exp. 8 | 1.72 | 0.64 |

**FIGURE 5. Linear one-term curve fitting with \( S_E \) and scaling coefficient \( \alpha \).**
TABLE 2. Relation between substitution error rate $S_E$ and scaling coefficient $\beta$.

| $S_E$ (%) | $\beta$ |
|----------|---------|
| Exp. 1   | 0.68    | 12.50 |
| Exp. 2   | 0.49    | 16.67 |
| Exp. 3   | 1.51    | 5.56  |
| Exp. 4   | 1.67    | 5.00  |
| Exp. 5   | 1.86    | 4.55  |
| Exp. 6   | 1.68    | 5.00  |
| Exp. 7   | 3.00    | 2.78  |
| Exp. 8   | 1.72    | 4.76  |

C. SECOND SCALING METHOD FOR LLRs

From the pairwise substitution error rate, $p(x_i = b^k, y_j = b^j)$ in (8) for $i \neq k$, the substituted substitution error rate $p'(x_i = b^k, y_j = b^j)$ is defined as

$$p'(x_i = b^k, y_j) = \beta p(x_i = b^k, y_j) \quad \text{for} \quad i \neq k,$$

where $\beta$ is a scaling coefficient satisfying $\beta > 0$. Then, the scaled LLRs, $L'^{(1)}_j$ and $L'^{(2)}_j$, are calculated from $L^j$ in (6) and $L^j_k$ in (7) by using the scaled substitution error rate defined in (10). For example, when $y_j$ is $C$, the scaled LLRs are given as

$$L'^{(1)}_j = \log \frac{\beta p(x_i = A, y_j = C) + p(x_i = C, y_j = C)}{\beta p(x_i = G, y_j = C) + p(x_i = T, y_j = C)},$$

and

$$L'^{(2)}_j = \log \frac{\beta p(x_i = A, y_j = C) + p(x_i = G, y_j = C)}{p(x_i = C, y_j = C) + \beta p(x_i = T, y_j = C)}.$$

This scaling also makes the magnitude of the LLRs smaller.

TABLE 3. Comparison of results for Experiments 1–8.

|                | Ref. [4] | S1 | S2 |
|----------------|----------|----|----|
| $\text{M}_{\text{read}}$ | $\text{R}_{\text{cost}}$ | $\text{M}_{\text{read}}$ | $\text{R}_{\text{cost}}$ | $\text{M}_{\text{read}}$ | $\text{R}_{\text{cost}}$ |
| Exp. 1         | 35000    | 2.73 | 35000 | 2.73 | 35000 | 2.73 |
| Exp. 2         | 63000    | 3.51 | 62500 | 3.49 | 62500 | 3.49 |
| Exp. 3         | 53500    | 3.49 | 53500 | 3.49 | 53500 | 3.49 |
| Exp. 4         | 72000    | 4.69 | 70000 | 4.56 | 70000 | 4.56 |
| Exp. 5         | 127000   | 8.28 | 125500 | 8.18 | 127000 | 8.28 |
| Exp. 6         | 58500    | 3.81 | 58500 | 3.81 | 58500 | 3.81 |
| Exp. 7         | 57000    | 3.68 | 57000 | 3.68 | 56000 | 3.65 |
| Exp. 8         | 58500    | 3.81 | 58500 | 3.81 | 58500 | 3.81 |

are synthesized by CustomArray and sequenced by Illumina iSeq technology. The detailed experimental parameters and procedures are described in Section 3 and 4 of Supplementary Material in [4], respectively. The code is available at https://github.com/shubhamchandak94/LDPC_DNA_storage. Among the parameters, we explain any changes or important ones belows.

In these experiments, we consider 30 trials of experiments instead of 20 trials and find the minimum number of reads that successfully decode all trials, since the 20 trials in [4] are too few. We consider the number of incremental reads as 500. We denote the minimum number of reads to make all trials successful as $\text{M}_{\text{read}}$. Then, the reading cost $\text{R}_{\text{cost}}$ is defined as $\text{R}_{\text{cost}} = (\text{M}_{\text{read}} \times \text{oligo length}) / (\text{file size (bytes)} \times 8)$. We compare the referenced system in [4] and the two proposed methods. The simulation results with the minimum number of reads $\text{M}_{\text{read}}$ and the reading cost $\text{R}_{\text{cost}}$ are given in Table 3 and Fig. 7. 'Ref. [4]' in Table 3 and Fig. 7 stands for referenced system [4].
TABLE 4. Numbers of successful decoding trials with different numbers of reads in Experiment 2.

| No. Reads | Ref. [4] | S1 | S2 |
|-----------|----------|----|----|
| 60500     | 20       | 19 | 22 |
| 61000     | 27       | 25 | 27 |
| 61500     | 25       | 26 | 28 |
| 62000     | 27       | 27 | 27 |
| 62500     | 28       | 30 | 30 |
| 63000     | 30       |    |    |

Table 3 and Fig. 7 show that the two scaling methods lead to improvements in Experiments 2, 4, 5, and 7 and similar performances in Experiments 1, 3, 6, and 8. To compare decoding performances, we list the number of successful decoding trials among 30 trials for the three LLR calculations in the experiments as the number of reads increases. Here, we mainly list Experiments 2, 4, 5, and 7 in Tables 4, 5, 6, and 7, respectively, where ‘No. Reads’ in the tables denotes the number of reads.

V. DISCUSSION ON GENERALIZATION OF THE PROPOSED SCALING METHODS AND PRACTICAL CALCULATION OF ERROR RATES

To verify that the proposed scaling methods are suitable for DNA storage with LLR calculation, we must apply the scaling methods to new experiments and compare results. However, since there are no new experiments with similar environments, it is unclear that the proposed scaling methods are generally applied to DNA storage. In this section, we suggest one application to verify the proposed scaling method. Moreover, we explain how to obtain error rates in more practical application later.

A. APPLICATION OF THE PROPOSED SCALING METHODS

In this subsection, we verify the application of our proposed scaling methods S1 and S2 for decoders with other experiments in DNA storage, which require the soft information.

Instead of running new experiments, we fit the scaling curve based on six experiments only, and test the scaling using the two remaining experiments. Since the number of cases to choose six experiments among eight experiments is 28, we consider only one case in which we already know the results of Experiments 1–6 and derive curve fitting functions for the two scaling methods in a manner similar to that explained in Section III. Then, we derive the optimal scaling coefficients of Experiments 7–8 using the curve fitting results of the other six experiments.

First, we apply the proposed scaling strategy S1 to Experiments 7–8. According to the optimal scaling coefficients of Experiments 1–6 in Table 1, we perform curve fitting of...
Experiments 1–6 with a linear one-term function, as shown in Fig. 8.

When Experiments 7–8 with LDPC codes and substitution error rate are provided, we can assign proper scaling values in LLR calculation. The substitution error rate $S_E$ and scaling coefficients $\alpha$ of Experiments 7–8 derived by the linear one-term function in Fig. 8, are listed in Table 8.

In Table 8, coefficients $\alpha$ for Experiments 7–8 are 0.78 and 0.64. Compared to Table 1, where we assume that true symbols are known, the differences are $-0.01$ and 0, respectively. Therefore, it is obvious that the performance of Experiment 8 is same. The performance of Experiment 7 is also same, which will be shown in Table 10.

Similarly, we apply Experiments 7–8 for the proposed scaling strategy $S_2$; according to the optimal scaling coefficients of Experiments 1–6 in Table 2, we perform curve fitting of Experiments 1–6 with a Gaussian two-term function, as shown in Fig. 9.

The substitution error rate $S_E$ and scaling coefficients $\beta$ of Experiments 7–8 derived by the Gaussian two-term function in Fig. 9 are listed in Table 9.

Note that $\beta$ values are 2.74 and 4.89 for Experiments 7–8 in Table 9, where the differences from Table 2 are $-0.04$ and 0.13, respectively. From Tables 8–9, the ratios of the gaps to the original scaling values are very small. To obtain the simulation results of Experiments 7–8 with the scaling values in Tables 8–9, we consider similar conditions to those used in Section IV. We still consider 30 trials of Experiments 7–8, and the incremental number of reads is 500. For the three LLR calculation methods which are Ref. [4], $S_1$, and $S_2$, the simulation results with the minimum number of reads $M_{\text{read}}$ and reading cost $R_{\text{cost}}$ are shown in Table 10 and Fig. 10.

From Table 10 and Fig. 10, we know that Experiments 7–8 have similar performances in Section IV. This means that our proposed scaling methods, $S_1$ and $S_2$, can be generally applicable to estimate scaling values for the decoders of new experiments in DNA storage. Similar to Section IV, we list the number of successful decoding trials among 30 trials for the calculations of the three LLR types in Experiment 7 as the numbers of reads increase in Table 11.

### B. PRACTICAL CALCULATION OF ERROR RATES

In Section III.A, we determine scaling values with statistics, where we compute statistics based on the assumption that the encoded symbol $x_j$ in (8) is known. These statistics in
Section III.A can provide actual error rates, but it is not easy to obtain these error rates in practice. In this subsection, we suggest two calculation methods of error rates.

The issue in the statistics calculation is that we cannot know the encoded symbol \( x_j \). However, we can estimate the symbol \( x_j \) using whole reads after MSA. Thus, the first calculation method of error rates (E1) is to estimate the symbol \( \hat{x}_j \) at each position by using majority rule, and use the estimated symbols \( \hat{x}_j \) in the calculation. In other words, we replace the probability \( p(x_j = b^k, y_j = b^j) \) in (8) with

\[
p(\hat{x}_j = b^k, y_j = b^j) = \frac{o(\hat{x}_j = b^k, y_j = b^j)}{N}. \tag{11}
\]

In all eight experiments, we find that the estimated symbols \( \hat{x}_j \) are the same as the true encoded symbols \( x_j \). This implies that the probability in (11) is equivalent to the probability (8). Then, scaling values obtained from E1 are the same as ones in Tables 1 and 2. Moreover, \( M_{\text{read}} \) and \( R_{\text{cost}} \) from E1 are also the same as ones in Table 3. The advantage of E1 is that it computes error rates without knowing the encoded symbols. However, it requires to check all reads, which results in computational issues.

The second calculation method of error rates (E2) is to use partial reads instead of all reads. Similar to E1, we estimate symbol \( \hat{x}_j \) at each position using majority rule, but from partial reads. Then, the estimated probability from partial reads is given as

\[
\hat{p}(\hat{x}_j = b^k, y_j = b^j) = \frac{o(\hat{x}_j = b^k, y_j = b^j)}{N'}, \tag{12}
\]

where \( N' \) is the number of partial reads. We choose the minimum number of reads \( M_{\text{read}} \) in Table 3 out of various \( N' \) for eight experiments. Since each trial provides different statistics, we denote the substitution error rate for the \( m \)-th trial by \( S'_E \), which is given as

\[
S'_E = \frac{1}{30} \sum_{m=1}^{30} S'_E, \tag{14}
\]

TABLE 12. Average error rates and scaling values from partial reads.

| \( N \)  | \( N' \)  | \( S'_E(\%) \) | \( \alpha \) | \( \beta' \) |
|---------|---------|----------------|----------|---------|
| Exp. 1  | 3108470 | 35000          | 0.40     | 0.49    | 19.38   |
| Exp. 2  | 5070796 | 62500          | 0.34     | 0.48    | 21.52   |
| Exp. 3  | 311894 | 53500          | 1.29     | 0.59    | 6.47    |
| Exp. 4  | 1278533 | 70000         | 1.35     | 0.60    | 6.18    |
| Exp. 5  | 1083234 | 125500         | 1.32     | 0.60    | 6.32    |
| Exp. 6  | 303779  | 58500          | 1.37     | 0.61    | 6.00    |
| Exp. 7  | 67752   | 56000          | 2.20     | 0.70    | 3.85    |
| Exp. 8  | 608211  | 58500          | 1.16     | 0.58    | 7.23    |

We summarize the number of total reads (\( N \)), the number of partial reads (\( N' \)), and average error rate for 30 trials (\( S'_E \)) in Table 12. We also present parameters \( \alpha' \) and \( \beta' \) which are obtained from curve-fitting functions in Fig. 5 and Fig. 6 using \( S'_E \).

Since the average substitution error rate calculated from partial reads in Table 12 is lower than the error rate in Table 1 and Table 2, \( \alpha' \) and \( \beta' \) tend to decrease and increase, respectively. However, scaling values in each trial are not always the same since scaling values are determined by not \( S'_E \) but \( S'_E \) at the \( m \)-th trial. After comparing the 30 \( S'_E \) with one \( S'_E \) in eight experiments, we find that 30 substitution error rates are near the average values at each experiment. In Table 13, the substitution error rate for 30 trials \( S'_E \) can be defined as

\[
S'_E = \frac{1}{30} \sum_{m=1}^{30} S'_E. \tag{14}
\]
we increase $M_{\text{read}}$ by 500 to decode successfully in the trial 24 when $\alpha'$ is 0.49. We obtain the minimum $M_{\text{read}} = 35500$. From these results, we can infer that there is slight mismatch in $\alpha$ scaling (S1) in Experiment 1. However, since the results using E2 (error rates from partial reads) are consistent with the results with true error rates (and the result with E1) in most cases, the error-rate calculation from partial reads can be used in practice. The results using E2 are also omitted here because of almost the same as Table 3.

### VI. CONCLUSION

In this work, new LLR estimation schemes have been proposed to reduce the reading cost in DNA storage with LDPC codes. Considering the mismatch of substitution errors between the statistic and theoretical analysis resulting in larger magnitude of LLRs, we proposed two strategies for scaling LLR magnitude. The proposed scaling values were obtained from eight experiments, but proper scaling values can be assigned via curve fitting equations. Since recent error correcting codes with powerful error correcting capability need soft information such as LLR, the proposed LLR calculation methods can be applied to any modern codes in DNA storage.

### ACKNOWLEDGMENT

The authors would like to thank S. Chandak [4] for guidance in setting up the simulation environment and helpful discussion.

### REFERENCES

[1] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney, “Towards practical, high-capacity, low-maintenance information storage in synthesized DNA,” *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013.

[2] L. Organick et al., “Random access in large-scale DNA data storage,” *Nature Biotechnol.*, vol. 36, no. 3, pp. 242–248, 2018.

[3] S. M. H. Tabatabaei Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, “A rewriterable, random-access DNA-based storage system.” *Sci. Rep.*, vol. 5, no. 1, p. 14138, Sep. 2015.

[4] S. Chandak, H. Ji, K. Tatwawadi, B. Lau, J. Mardia, M. Kubit, J. Neu, P. Griffin, M. Wootters, and T. Weissman, “Improved read/write cost tradeoff in DNA-based data storage using LDPC codes,” in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput.* (Allerton), Monticello, IL, USA, Sep. 2019, pp. 147–156.

[5] S. M. H. T. Yazdi, R. Gabrys, and O. Milenkovic, “Portable and error-free DNA-based data storage,” *Sci. Rep.*, vol. 7, no. 1, p. 5011, Jul. 2017.

[6] R. Heckel, G. Mikutis, and R. N. Grass, “A characterization of the DNA data storage channel,” *Sci. Rep.*, vol. 9, no. 1, p. 9663, Dec. 2019.

[7] M. Blawat, K. Gaedke, I. Hütter, X.-M. Chen, B. Turczyk, S. Inverso, B. W. Pruitt, and G. M. Church, “Forward error correction for DNA data storage,” *Procedia Comput. Sci.*, vol. 80, no. 3, pp. 1011–1022, Jan. 2016.

[8] Y. Erlich and D. Zielinski, “DNA fountain enables a robust and efficient storage architecture,” *Science*, vol. 355, no. 6328, pp. 950–954, Mar. 2017.

[9] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low density parity check codes,” *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, Mar. 1997.

[10] L. Deng, Y. Wang, M. Noor-A-Rahim, Y. L. Guan, Z. Shi, E. Gunawan, and C. L. Poh, “Optimized code design for constrained DNA data storage with asymmetric errors,” *IEEE Access*, vol. 7, pp. 84107–84121, 2019.

[11] S. Chandak, J. Neu, K. Tatwawadi, J. Mardia, B. Lau, M. Kubit, R. Hulett, P. Griffin, M. Wootters, T. Weissman, and H. Ji, “Overcoming high nanopore basecaller error rates for DNA storage via basecaller-decoder integration and convolutional codes,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8822–8826.

[12] P. Fei and Z. Wang, “LDPC codes for portable DNA storage,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 76–80.

[13] T. Lassmann, O. Frings, and E. L. L. Sonnhammer, “Kalign2: High-performance multiple alignment of protein and nucleotide sequences allowing external features,” *Nucleic Acids Res.*, vol. 37, no. 3, pp. 858–865, Feb. 2009.

[14] A. Shokrollahi, “LDPC codes: An introduction.” Digital Fountain, Fremont, CA, USA, Tech. Rep., Apr. 2003, vol. 2, pp. 17–51.

---

**TABLE 13.** Average substitution error rates from partial statistics of Experiment 1.

| Trial | $S_{\text{E}}$(%) | Trial | $S_{\text{E}}$(%) | Trial | $S_{\text{E}}$(%) |
|-------|------------------|-------|------------------|-------|------------------|
| 1     | 0.41             | 11    | 0.39             | 21    | 0.40             |
| 2     | 0.41             | 12    | 0.42             | 22    | 0.41             |
| 3     | 0.41             | 13    | 0.41             | 23    | 0.40             |
| 4     | 0.41             | 14    | 0.40             | 24    | 0.41             |
| 5     | 0.40             | 15    | 0.41             | 25    | 0.40             |
| 6     | 0.38             | 16    | 0.41             | 26    | 0.40             |
| 7     | 0.40             | 17    | 0.42             | 27    | 0.40             |
| 8     | 0.40             | 18    | 0.40             | 28    | 0.37             |
| 9     | 0.42             | 19    | 0.41             | 29    | 0.40             |
| 10    | 0.41             | 20    | 0.40             | 30    | 0.41             |

**XIAOZHOU LU** received the B.S. degree from the University of Ulsan, South Korea, in 2018, where he is currently pursuing the M.S. and Ph.D. degrees with the School of Electrical Engineering. His main research interests include 5G communication, error correction codes, and storage systems.

**JAEHO JEONG** (Graduate Student Member, IEEE) received the B.S. degree in computer science from Yonsei University, Seoul, South Korea, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Seoul National University, Seoul. His current research interests include error-correcting codes and DNA storage.

**JAE-WON KIM** (Member, IEEE) received the B.S. and Ph.D. degrees in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2014 and 2020, respectively. He is currently a Senior Engineer with Samsung Electronics, South Korea. His research interests include error-correcting codes, coding theory, and index coding.
JONG-SEON NO (Fellow, IEEE) received the B.S. and M.S.E.E. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1981 and 1984, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1988. He was a Senior MTS with Hughes Network Systems, from 1988 to 1990. He was an Associate Professor with the Department of Electronic Engineering, Konkuk University, Seoul, from 1990 to 1999. He joined the faculty of the Department of Electrical and Computer Engineering, Seoul National University, in 1999, where he is currently a Professor. He became an IEEE Fellow through the IEEE Information Theory Society, in 2012. He was a recipient of the IEEE Information Theory Society Chapter of the Year Award, in 2007. From 1996 to 2008, he served as a Founding Chair of the Seoul Chapter of the IEEE Information Theory Society. He was a General Chair of Sequence and Their Applications 2004 (SETA2004), Seoul. He also served as a General Co-Chair of the International Symposium on Information Theory and Its Applications 2006 (ISITA2006) and the International Symposium on Information Theory 2009 (ISIT2009), Seoul. He served as a Co-Editor-in-Chief of the IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS, from 2012 to 2013. He became a member of the National Academy of Engineering of Korea (NAEK), in 2015, where he is currently a Division Chair of Electrical, Electronic, and Information Engineering. His area of research interests includes error-correcting codes, cryptography, sequences, LDPC codes, interference alignment, and wireless communication systems.

HOSUNG PARK (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 2007, 2009, and 2013, respectively. He was a Postdoctoral Researcher with the Institute of New Media and Communications, Seoul National University, in 2013, and the Qualcomm Institute, The California Institute for Telecommunications and Information Technology, University of California at San Diego, La Jolla, CA, USA, from 2013 to 2015. He is currently an Associate Professor with the Department of Computer Engineering, Chonnam National University, Gwangju, South Korea, as of 2015. His research interests include channel codes for communications systems, coding for memory/storage, coding for distributed storage, communication signal processing, compressed sensing, and network information theory.

SUNGHWAN KIM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Seoul National University, South Korea, in 1999, 2001, and 2005, respectively. He was a Postdoctoral Visitor with the Georgia Institute of Technology (GeorgiaTech), from 2005 to 2007, and a Senior Engineer with Samsung Electronics, from 2007 to 2011. He is currently a Professor with the School of Electrical Engineering, University of Ulsan, South Korea. His main research interests are channel coding, modulation, massive MIMO, visible light communication, quantum information, and storage system.