Dynamic sliding window encoding for data storage on DNA under biological and indexing constraints

Chloé Berton
IMT Atlantique
Unite Inserm 1101 LaTIM
Brest, France
chloe.berton@imt-atlantique.fr

Gouenou Coatrieux
IMT Atlantique
Unite Inserm 1101 LaTIM
Brest, France
gouenou.coatrieux@imt-atlantique.fr

Dominique Lavenier
GENSCALE, INRIA
Rennes, France
dominique.lavenier@irisa.fr

Sahar Haddad
Unite Inserm 1101 LaTIM
Brest, France
sahar.haddad@inserm.fr

***Abstract*—In this paper, we propose a first dynamic sliding window encoding (DSWE) for storing encrypted data in a DNA form taking into account biological constraints and prohibited nucleotide motifs used for data indexing. Its originality is twofold. First, it takes advantage of variable length DNA codewords to avoid homopolymers longer than N bases when encoding binary data. Second, it relies on a sliding window to prevent the creation of prohibited motifs of nucleotides, adding non-coding bases when necessary. Contrarily to existing schemes, scaling DSWE to high values of N and of numbers of prohibited motifs is extremely simple. It is furthermore independent of the cryptosystem. We provide the theoretical information rate of our proposal for a given number of prohibited motifs and a maximum homopolymer length. Experiments show that in general, it offers much higher performances than existing schemes.**

***Index Terms*—Confidentiality, DNA data storage, dynamic encoding, encryption, variable length DNA codewords.**

I. Introduction

World digital data production has been growing exponentially. It is envisioned to reach 175ZB of data [1] in 2025. Storing all these data is a challenge as current electronic technologies (flash memory, hard disk drives) are reaching their limits, particularly in terms of density (capacity $< 10^{12}$ bytes), energy, environmental costs, and sustainability ($<$ 20 years). Data storage on deoxyribonucleic acid (DNA) molecules has recently shown great promise [2]. It could be a million times denser with 10^{17} bytes per gram of DNA, for a 100 times longer lifespan and near-zero energy consumption (storage at room temperature). The basic principle of DNA data storage consists of three main steps: i) digital data conversion into DNA sequences of bases, or nucleotides, (A,C,G,T) ; ii) DNA sequences synthesis ; iii) DNA molecules or strands storage inside specific containers. To retrieve data, DNA strands are sequenced, generating digital DNA sequences that are decoded to recover the original information. Fig. 1 provides an overview of a common DNA storage chain with digital and biological functionalities in blue and green, respectively. These functionalities are not secured. For instance, data confidentiality is at risk at all stages. An unauthorized user can spy on the sequencing [3] or synthesis [4] device, or secretly clone/steal the stored DNA molecules. This is why an encryption/decryption step should be added as shown in red in Fig. 1. But, doing so has consequences on data encoding into nucleotides due to constraints of DNA synthesis and sequencing technologies and of data indexing. Indeed, cryptosystems convert clear data into an encrypted bitstream that is a uniformly distributed sequence of bits. Since it is impossible to predict the outcome of a cryptosystem, there is a risk that, when encoded in base-4, an encrypted bitstream leads to a DNA sequence that does not comply with these constraints. The first class of constraints are biological: i) an uneven proportion of G and C bases (G-C content) will cause errors during data retrieval including dropouts (missing bases) during sequencing [5]; ii) synthesis and sequencing processes are sensitive to homopolymers (repetition of the same nucleotide), the longer these ones the higher the probability of generating errors is at these steps. Another important constraint is the unwanted presence of certain base motifs in DNA sequences. For instance, for file indexing, each sequence can be framed by primers, usually 18 to 30 bases long, that are unique to a sequence or group of sequences that belong to a file. Based on these primers, data random access is possible by selecting a subset of sequences before sequencing. Thus primers should not exist in the middle of a sequence to avoid loss of information.

To the best of our knowledge, only a few works have been interested in encoding data in DNA molecules while taking into account biological constraints. Some of them only consider a maximum homopolymer length along with a balanced G-C content [6] [7], others add to their encoding scheme the possibility to limit the appearance of secondary structures [8] where a part of a sequence is complementary to another, or more generally prohibited motifs [9]. The mCGR solution [9] is the only one to consider prohibited motifs along with balanced G-C content and homopolymer size constraints. Basically, its strategy consists in building a dictionary that associates to a fixed length block of information bits a fixed length base-4 or DNA codeword. To create this dictionary, they follow a “fractal” strategy to go across the base-4 codeword space to find codewords that satisfy the previous three constraints without having to test all codewords. They next clean the dictionary by removing all codewords that if concatenated do not respond to the constraints. Even though mCGR encoding is quite fast, because of this cleaning operation, its information rate that is the number of bits

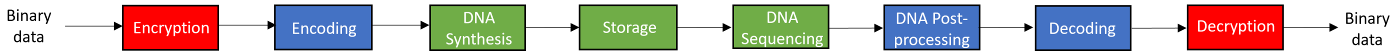


Fig. 1. A common DNA storage chain with its digital (blue) and biological (green) functionalities. Encryption (red) is used to ensure data confidentiality.

encoded per base (BPB) significantly drops when the number of prohibited motifs increases. Additionally, this solution does not consider data confidentiality, data are not encrypted.

In this work, we propose a novel encoding for DNA storage of encrypted data while considering the above biological constraints (G-C content and homopolymers) as well as prohibited motifs. Its originality stands on a two-step Dynamic Sliding Window Encoding that converts binary data into DNA sequences of variable length. It relies on the use of a set of dictionaries of small dimensions and on an informed encoding/decoding strategy based on a sliding window. Basically, a binary block of fixed length is encoded into a variable length DNA sequence using a dictionary which is selected according to the last encoded DNA base to avoid homopolymers, while a sliding window of fixed length is used to detect if a prohibited motif is about to appear. If it is the case, a non-coding base is added to the sequence. This base does not encode any information but will also be identified by the decoder. This latter is said "informed" as it repeats in part the encoding process to decide which dictionary to use and if a base it is reading is a non-coding one. Beyond, the G-C balance is ensured due to the fact we encode uniformly distributed encrypted data. As we will see, compared to mCGR, our proposal offers a greater information rate. Moreover, it is easy and more flexible to adapt when changing the maximum homopolymer size and does not require the complex elaboration of a dictionary as in [7] [10] and [9] with the possibility to manage a larger number of prohibited motifs. In addition, it is independent of the chosen cryptosystem.

The rest of this paper is organized as follows. Our dynamic encoding sliding windows encoding (DSWE) is presented in Section 2. Section 3 provides and discusses experimental results and comparison assessment of our proposal with the only concurent scheme [9]. Section 4 concludes this paper.

II. DNA DYNAMIC SLIDING WINDOW ENCODING

Herein, we first introduce the DNA storage chain we work with along with its underlying technologies, before presenting the Dynamic Sliding Window Encoding (DSWE) we propose.

A. DNA storage chain

The DNA data storage chain considered in this work is depicted in Fig. 1. Binary data are first encrypted before being DSWE-encoded into nucleotide sequences. For experimentation, we have opted for the well-known Advanced Encryption Standard (AES) [11], the symmetric key algorithm of choice for NIST and NSA US agencies. AES is a block-cipher cryptosystem that encrypts a fixed-length plaintext block P_i of 128-bits into a ciphertext C_i of same size such as: $C_i = Enc(P_i, K_{AES})$, where K_{AES} is the encryption key. To decrypt C_i, the AES algorithm works in a reverse way such as: $P_i = Dec(C_i, K_{AES})$. An important aspect is that AES outputs are independent and identically distributed 128-bit blocks, i.e., one bit has an equal probability to be '0' or '1' and is independent from other ciphertext bits. Therefore, if one applies a direct base-4 encoding, each of the nucleotide value $\{A, C, G, T\}$ has a probability of 25% to appear at any position in a sequence and independently from any other nucleotide. This will allow the preservation of the G-C balance. Once encrypted data are DSWE-encoded, obtained DNA sequences are framed with primers for data indexing.

In this work, biological functionalities are simulated using the simulator proposed in [12]. This one mimics: DNA synthesis based on phosphoramidite chemistry; in vitro storage of DNA molecules; and DNA strand sequencing with MinION devices based on the Oxford Nanopore Technology (ONT). It takes as input a DNA sequence flanked by primers and produces several digital strands copies of this sequence introducing three types of nucleotide errors: insertion, deletion and substitution. As depicted in Fig. 1, digital data are post-processed through the consensus algorithm from [13], taking as input these multiple erroneous copies to produce a unique DNA sequence. Then DSWE decoding converts back DNA nucleotides into binary data.

B. Dynamic Sliding Window Encoding (DSWE)

As depicted in Fig. 2, our Dynamic Sliding Window Encoding (DSWE) scheme consists in applying iteratively two encodings: Dynamic Encoding (DE) and Sliding Window Encoding (SWE). DE and SWE are two variable length DNA encodings the purpose of which is to avoid the creation of homopolymers of a given maximum size N and the occurrence of M prohibited motifs of m nucleotides stored in a dictionary P_m, respectively. In the sequel, we first detail DE and then explain how it is combined with SWE to achieve DSWE.

1) DE encoding: The objective of this encoding is to avoid the generation of homopolymers longer than a given maximum length N, that is to say the repetition of a nucleotide. Our basic idea is to dynamically encode binary data using different dictionaries or codebooks depending on the last base of the DNA sequence that has been already encoded. Such a dictionary associates fixed length binary blocks of $2(N-1)$ bits to variable length (VL) DNA codewords of $N-1$ or N bases. Codewords have been designed in such a way that, if concatenated with the lastly encoded base, it is not possible to generate a homopolymer of size greater than N.

To be more clear, let us consider a maximum homopolymer length $N = 4$. Our dynamic encoder takes as input a binary block S_k of $2*(N-1) = 6$ bits, $(b_0^k b_1^k b_2^k b_3^k b_4^k b_5^k)$, and the last DNA encoded base B^l of the DNA sequence. It outputs

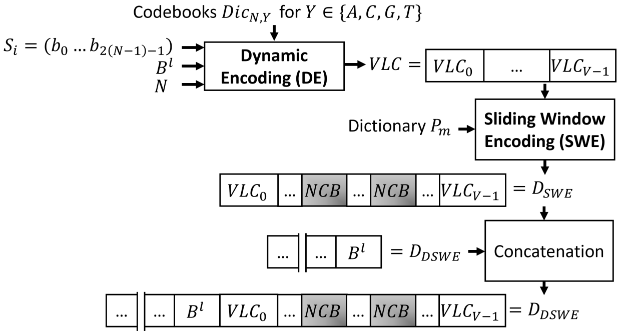
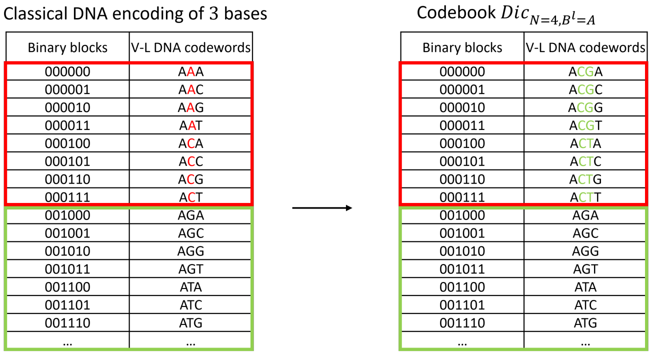

Codebooks $Dic_{N,Y}$ for $Y \in \{A,C,G,T\}$
$S_i = (b_0 \dots b_{2(N-1)-1})$
B^l
N
Dynamic Encoding (DE)
$VLC =$
VLC_0
...
VLC_{V-1}
Dictionary P_m
Sliding Window Encoding (SWE)
VLC_0 ... NCB ... NCB ... VLC_{V-1} $= D_{SWE}$
... ... B^l $= D_{DSWE}$
Concatenation
... ... B^l VLC_0 ... NCB ... NCB ... VLC_{V-1} $= D_{DSWE}$

Fig. 2. Dynamic Sliding Window Encoding (DSWE) of a binary block S_i. First, S_i is encoded into a variable length DNA codeword VLC using DE (Dynamic Encoding), taking into account the last encoded base B^l of the DNA sequence D_{DSWE}, a maximal homopolymer size N and codebooks $\{Dic_{N,Y}\}_{Y=A,C,G,T}$. Then, SWE (Sliding Window Encoding) is applied leading to the addition or not of Non-Coding Bases (NCBs) to avoid prohibited motifs from the dictionary P_m. The resulting DNA codeword D_{SWE} is concatenated to the already encoded DNA sequence D_{DSWE}.

a DNA codework D_k of length $N-1 = 3$ or $N = 4$ bases, i.e. $D_k = (B_0^k B_1^k B_2^k)$ or $D_k = (B_0^k B_1^k B_2^k B_3^k)$, where $B_p^k \in \{A, C, G, T\}$, $\forall p$. S_k is encoded into D_k using one of four dictionaries $(Dic_{4,A}, Dic_{4,C}, Dic_{4,G}, Dic_{4,T})$, chosen depending on the value of B^l to avoid the repetition of this base. To explain the construction principle of these dictionaries, let us consider $Dic_{4,A}$ that is used when $B^l = A$. This one encodes blocks of 6 bits into 64 possible DNA codewords. At first, $Dic_{4,A}$ contains all classical DNA codewords of size 3, as shown in Fig. 3. Classical DNA Encoding associates two bits to one base as follows for example: $(00 \rightarrow A), (01 \rightarrow C), (10 \rightarrow G), (11 \rightarrow T)$. In second, classical DNA codewords starting with bases AA are re-encoded in VL DNA codewords to avoid $(A...A)$ homopolymers. DNA codewords starting with AC are also re-encoded to guarantee that no codeword is the prefix of another. This ensures there will be no ambiguity at the decoding stage about the encoded binary sequence. To avoid homopolymers, these codewords are built such that $B_0^k = A$ and $B_1^k \in C, G, T$. As a result, for $Dic_{4,A}$, 56 binary blocks are encoded using the Classical Encoding into 3-base DNA Codewords, while 8 binary blocks are encoded into 4-base DNA Codewords. Other codebooks $Dic_{4,C}$, $Dic_{4,G}$ and $Dic_{4,T}$ are similarly constructed. We give in Fig. 4 the rules for the re-encoding of B_1^k into 2 bases for any values of N. It can be seen that DE Encoding is easy to design. Regarding the initialization of the encoding, that is to say the encoding of the first DNA sequence base, we use the Classical DNA Encoding.

2) SWE Encoding: The purpose of this Sliding Window Encoding scheme (SWE) is to generate DNA sequences without creating prohibited motifs, such as primers. Let us consider a binary sequence S that is progressively DE-converted into a DNA sequence D and a dictionary P_m which contains M prohibited motifs of m bases. To avoid that such a motif occurs when adding to D a new base B_j, we propose to use a sliding window W to verify if the last $m-1$ encoded nucleotides, i.e. $(B_{j-m+1}...B_{j-1})$, equal the $m-1$ high order bases of one of the prohibited motifs. For instance, the u^{th} prohibited motif is such that $M_u = (ATCGAT)$ of size $m = 6$, with the following $m-1$ high-order bases: $(ATCGA)$. Thus, if W does not correspond to an element of P_m, then B_j is concatenated to D. Otherwise, B_j must take a base value to avoid the occurrence of a non-authorized motif. We denote such a base as a non-coding base (NCB). Once B_j is encoded, the sliding window advances by one base. Notice that as long as a prohibited motif is detected, a non-coding base is added. On its side, the decoder follows the same strategy. Before considering that B_j encodes data, it checks if the previous $m-1$ bases do not equal the first bases of a prohibited motif.



Classical DNA encoding of 3 bases

Binary blocks	V-L DNA codewords
000000	AAA
000001	AAC
000010	AAG
000011	AAT
000100	ACA
000101	ACC
000110	ACG
000111	ACT
001000	AGA
001001	AGC
001010	AGG
001011	AGT
001100	ATA
001101	ATC
001110	ATG
...	...

Codebook $Dic_{N=4,B^l=A}$

Binary blocks	V-L DNA codewords
000000	ACGA
000001	ACGC
000010	ACGG
000011	ACGT
000100	ACTA
000101	ACTC
000110	ACTG
000111	ACTT
001000	AGA
001001	AGC
001010	AGG
001011	AGT
001100	ATA
001101	ATC
001110	ATG
...	...

Fig. 3. Re-encoding of classically encoded DNA codewords into variable length DNA codewords in the case of the codebook $Dic_{N=4,B^l=A}$ to avoid DNA codewords beginning with (AA). 8 codewords framed in red must be re-encoded: their second base (red letter) is encoded into 2 bases in green.

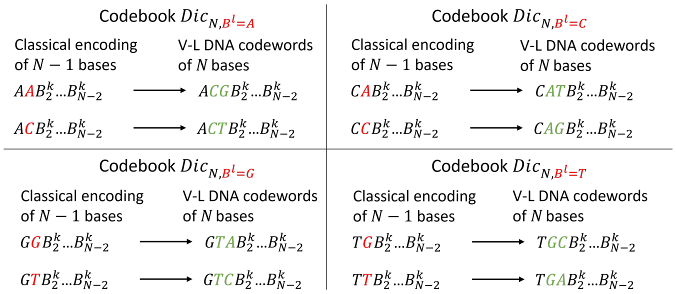


Fig. 4. Re-encoding of some DNA codewords in codebooks Dic_{N,B^l} so that no codeword begins with $(B^l B^l)$. Dictionaries typically encode blocks of bits with the Classical Encoding, except for DNA codewords beginning with B^l and a base in red. Such red bases are re-encoded into bases in green.

3) DSWE Encoding : To sum up, let us consider that DSWE is looking for the encoding of a binary block S_i of $2(N-1)$ bits having already encoded a DNA sequence of d bases: $D_{DSWE} = (D_{DSWE,0}...D_{DSWE,d-1})$. As it can be seen from Fig. 2, the process begins with the Dynamic Encoding of S_i into a variable length DNA codeword VLC of size N or $(N-1)$ nucleotides, using one of the codebooks $\{Dic_{N,B^l}\}_{B^l=A,C,G,T}$ chosen by looking at the last encoded base B^l of D_{DSWE} (i.e. $D_{DSWE,d-1}$). Then, DSWE continues with the SWE encoding of VLC, considering the dictionary P_m. It outputs a variable length DNA sequence

D_{SWE} where one or several Non-Coding Bases (NCB) may have been added if necessary. In fact, the bases of VLC are added to D_{DSWE} one by one. Regarding VLC_0, a sliding window W is created with the last $m-1$ bases of D_{DSWE}. If the window does not match to the $m-1$ high order bases of a motif from P_m, VLC_0 is added to D_{DSWE}, otherwise a non-coding base is concatenated to D_{DSWE}. The sliding window W is shifted by one base to consider the last $m-1$ bases of D_{DSWE} until all bases of VLC have been added to D_{DSWE}. Regarding DSWE initialization, as for DE, the first two bits of S are encoded with the Classical DNA Encoding.

The decoding DSWE process works in the reverse way. Let us consider DSWE is looking for the decoding of the d^{th} base D_d of the sequence D_{DSWE}, having already decoded $d-1$ bases from a DNA sequence D_{DSWE} into a global binary sequence S_{DSWD}. To do so, the process starts by applying the SWE decoding taking as input the window W that contains the $m-1$ bases preceding D_d in D_{DSWE}. If W does not correspond to the $m-1$ higher bases of one of the motifs of P_m then D_d is assumed to be a base of a VL DNA codeword. Such a codeword can be of N or $N-1$ bases and its DE decoding depends on the value of the base that precedes it in D_{DSWE} (i.e., the base B^l from the DSWE encoding point of view) to select the appropriate decoding dictionary (i.e., Dic_{N,B^l}). Finally, the output of the decoding of a VL DNA codeword, a binary block of $2(N-1)$ bits is concatenated to the DSWE-decoded sequence S_{DSWD}.

C. Theorical performance of DSWE

As DSWE produces DNA sequences of variable length, the theoretical information rate, that is the number of bits encoded per base, is probabilistic. However, because we encode encrypted data, data are uniformly distributed (i.e., $Pr(b_i^k = 0) = Pr(b_i^k = 1) = \frac{1}{2}$). Thus, it is possible to estimate this rate in average for a given maximal homopolymer length N and M prohibited motifs of m bases, such as:

$$R_{DSWE} = R_N * P_{B_j} \tag{1}$$

where: R_N is the DE information rate for a maximal homopolymer size N and P_{B_j} the probability the base B_j is a coding nucleotide. As we encode encrypted binary data, all VLC codewords of DE are equiprobable. Additionally, for any values of $N > 2$, blocks of $2(N-1)$ bits are classically encoded into $N-1$ bases for $\frac{7}{8}^{th}$ of binary blocks, and into N bases for an 8^{th} of inputs (re-encoded VL DNA codewords as seen in Fig. 4), leading to an information rate (in BPB) of:

$$R_N = \frac{2*(N-1)}{(N*\frac{1}{8}+(N-1)*\frac{7}{8})} = \frac{16*(N-1)}{8*N-7} \tag{2}$$

P_{B_j}, the probability that B_j is a coding base, is given by:

$$P_{B_j} = (1 - \frac{|P_{mW}|}{4^{m-1}}) \tag{3}$$

where: $|P_{mW}|$ is the number of prohibited motifs with distinct $m-1$ high order bases. Indeed, B_j is a non-coding base if it is preceded by the $m-1$ high order bases of a prohibited motif. In the case one has $|P_m|$ high order sequences, the probability of such a high order sequence occurs is of $\frac{|P_m|}{4^{m-1}}$.

The proposed solution as well as all experiments do not include an error-correction code. It would be typically applied on the binary encrypted data just before DSWE encoding, adding thus some information overhead.

III. EXPERIMENTAL RESULTS

We implemented our algorithms in Python3 (version 3.8.10) using the following dataset: the test image "Lena.png" of 48 Kbytes, a random *lorem ipsum* text of 95 Kbytes and the pdf files of two scientific articles of 2.4 MBytes and 567 KBytes. Each data sample was first encrypted 5 times with 5 distinct AES-256 keys. Then, each of the 5 encrypted bitstreams was separated into 10000 fixed sized binary blocks of X bits, with $X \in \{100, 200, 1000, 2000\}$, separately DSWE-encoded into DNA sequences considering the different parameters' values $N = \{2, 3, 4\}$, $M = \{0, 10, 20, 30, 40, 50, 60, 70, 76\}$ prohibited motifs of $m = 6$ bases with distinct $m-1$ high order bases. In the sequel, we denote the parametrization of our scheme as follows: $DSWE(N, M, m)$.

A. Experimental DSWE information rate and comparison

We computed the experimental information rate of our DSWE scheme as the mean value of the information rates obtained for all encoded sequences. It is defined such as:

$$R_{DSWE}^E = \frac{\sum_{i=1}^{e} \frac{|S_i|}{|D_i|}}{e} \tag{4}$$

where: e is the number of binary blocks; S_i is the i^{th} binary block of the encrypted bitstream; D_i is the DE-encoded DNA sequence of S_i; and $|x|$ is the size operator in bits if x is a binary block, or in bases for a DNA sequence.

We give in Table I the information rate of our scheme considering different values of $N \in \{2, 3, 4\}$ and $M \in [0, 76]$ and compared it with the one of the mCGR method from [9] which, to our knowledge, is the only other existing work that considers prohibited motifs, the G-C content and a maximal homopolymer size constraints. mCGR generates codebooks of DNA codewords of size l with: a fixed G-C content or in an interval; homopolymers of maximum size N; and a list of prohibited motifs of up to l bases. We use its publicly available implementation considering a G-C content balance of $40-60\%$ and the generation of a codebook with DNA codewords of l nucleotides, $l = 6$ or 10. Authors of [9] provided the maximum information rate R_{mCGR}^{opt} their scheme may reach, it is given by: $R_{mCGR}^{opt} = (\lfloor log_2|C| \rfloor)/l$ where: $|C|$ is the size of the codebook C; and $\lfloor log_2|C| \rfloor$ is the integer part of $log_2|C|$. For $l = 10$ this rate cannot be higher than 1.9 BPB. As seen in Table I, the information rate of DSWE decreases linearly but slowly and outperforms mCGR for any values of M and $N > 2$. Its rate tends to the ideal rate of 2 BPB with the increase of N while mCGR peaks at 1.9 BPB. For instance, when $M = 76$, $DSWE(3, 76, 6)$ achieves an experimental rate of 1.73 BPB while mCGR provides a rate of 0.75 BPB and 1.3 BPB with DNA codewords of size

TABLE I

Experimental information rates in BPB (bits per base) of DSWE and of [9] for a given maximal homopolymer size N and number M of prohibited motifs of $m = 6$ bases. DSWE rate is given in average and standard deviation (STD). Rate of [9] is given for DNA codewords of length $l = 10$ bases.

M	N	DSWE rate	DSWE rate STD	Rate of [9] for $l = 10$
0	2	1.334	0.015	**1.8**
	3	**1.882**	0.010	1.8
	4	**1.921**	0.008	1.9
20	2	1.302	0.017	**1.7**
	3	**1.839**	0.011	1.8
	4	**1.880**	0.01	1.8
40	2	1.239	0.019	**1.6**
	3	**1.803**	0.011	1.7
	4	**1.843**	0.011	1.7
76	2	1.167	0.020	**1.2**
	3	**1.730**	0.013	1.3
	4	**1.768**	0.011	1.3

$l = 6$ and $l = 10$, respectively. Indeed, mCGR collapses with large numbers of prohibited motifs with its dictionary cleaning procedure. Moreover, our solution is more flexible and can take into account any value higher than 1 for N, while mCGR cannot consider values of N higher than the length l of its DNA codewords. One can also note that compared to DE encoding rate (when $M = 0$ for DSWE, in Table I), we have an information rate loss of 0.15 BPB in average in the case DSWE works with $M = 76$ prohibited motifs. To conclude, DSWE ensures a dense information rate.

B. G-C content, runtime and data recovery

To be correctly synthetized and read, DNA sequences must have a balanced base rate, with a required 40% to 60% rate of G and C bases [5]. We verified that, for any maximal homopolymer length N and number of motifs M, our 10000 test DNA sequences of 1000 bits all have a G-C content close to 50%, even in the case of short DNA sequences of 21 bases. These results are consistent with requirements.

In terms of runtime, DSWE encodes the "Lena" image in 1.3 to 6.9 seconds depending on N and M using a non-optimized implementation. mCGR achieves better runtime performances (0.2 to 1.1 seconds) than our solution but with a much smaller information rate. Notice also that mCGR does not include an encryption module.

Using the simulator from [12] with the consensus procedure from [13], we evaluated the robustness of DSWE to common errors that exist in a DNA storage chain (see Section II-A). Despite its dynamic character, with DNA sequences of variable length, DSWE offers at least the same performance than [9]. Whatever the values of M and $N \leq 4$, both methods successfully recover encoded DNA sequences with $35 \sim 40$ strand copies. Thus, DSWE does not seem to be more vulnerable to error propagation than fixed-length codeword based encodings while achieving a much better information rate.

IV. Conclusion

In this paper, we proposed a new dynamic sliding window encoding (DSWE) of encrypted data that produces DNA sequences of variable length. DSWE is able: to avoid homopolymers longer than N bases using variable length DNA codewords; to prevent the creation of the prohibited motifs during encoding using a sliding window; to ensure a balanced G-C content even in short sequences. Compared to state-of-the-art, our method reaches much better information rates for any maximal homopolymer lengths $N > 2$ and number M of prohibited motifs. For a given value of M, its information rate tends to the 2 BPB ideal rate as N grows. DSWE is also flexible and easy to deploy. Furthermore, we showed that data recovery is not impacted despite the dynamic nature of DSWE.

Acknowledgment

This work was supported in part by the French Government support granted to the Labex CominLabs and managed by the ANR through the "Investing for the Future" Program under Grant ANR-10-LABX-07-01.

References

[1] D. Reinsel, J. Gantz, and J. Rydning, "Data age 2025: the digitization of the world from edge to core," *Seagate*, vol. 16, 2018.

[2] P. Y. De Silva and G. U. Ganegoda, "New trends of digital data storage in dna," *BioMed research international*, vol. 2016, 2016.

[3] P. Ney, K. Koscher, L. Organick, L. Ceze, and T. Kohno, "Computer security, privacy, and dna sequencing: compromising computers with synthesized dna, privacy leaks, and more," in *26th USENIX Security Symposium (USENIX Security 17)*. Usenix, 2017, pp. 765–779.

[4] S. Faezi, S. R. Chhetri, A. V. Malawade, J. C. Chaput, W. Grover, P. Brisk, and M. A. Al Faruque, "Acoustic side channel attack against dna synthesis machines," in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2020, pp. 186–187.

[5] M. G. Ross, C. Russ, M. Costello, A. Hollinger, N. J. Lennon, R. Hegarty, C. Nusbaum, and D. B. Jaffe, "Characterizing and measuring bias in sequence data," *Genome biology*, vol. 14, no. 5, pp. 1–20, 2013.

[6] Y. Erlich and D. Zielinski, "Dna fountain enables a robust and efficient storage architecture," *science*, vol. 355, no. 6328, pp. 950–954, 2017.

[7] Y. Wang, M. Noor-A-Rahim, E. Gunawan, Y. L. Guan, and C. L. Poh, "Construction of bio-constrained code for dna data storage," *IEEE Communications Letters*, vol. 23, no. 6, pp. 963–966, 2019.

[8] Z. Ping, S. Chen, G. Zhou, X. Huang, S. J. Zhu, H. Zhang, H. H. Lee, Z. Lan, J. Cui, T. Chen *et al.*, "Towards practical and robust dna-based data archiving using the yin–yang codec system," *Nature Computational Science*, vol. 2, no. 4, pp. 234–242, 2022.

[9] H. F. Löchel, M. Welzel, G. Hattab, A.-C. Hauschild, and D. Heider, "Fractal construction of constrained code words for dna storage systems," *Nucleic acids research*, vol. 50, no. 5, pp. e30–e30, 2022.

[10] T. T. Nguyen, K. Cai, K. A. S. Immink, and H. M. Kiah, "Capacity-approaching constrained codes with error correction for dna-based data storage," *IEEE Transactions on Information Theory*, vol. 67, no. 8, pp. 5602–5613, 2021.

[11] N. Mouha and M. Dworkin, "Review of the advanced encryption standard," *Technical report, National Institute of Standards and Technology*, 2021.

[12] B. Hamoum, E. Dupraz, L. Conde-Canencia, and D. Lavenier, "Channel model with memory for dna data storage with nanopore sequencing," in *2021 11th International Symposium on Topics in Coding (ISTC)*. IEEE, 2021, pp. 1–5.

[13] D. Lavenier, "Constrained consensus sequence algorithm for dna archiving," *arXiv preprint arXiv:2105.04993*, 2021.