Robust image coding on synthetic DNA: Reducing sequencing noise with inpainting

Eva Gil San Antonio, Mattia Piretti, Melpomeni Dimopoulou, Marc Antonini

Robust image coding on synthetic DNA: Reducing sequencing noise with inpainting

Eva Gil San Antonio
Université Côte d'Azur
Laboratoire I3S / CNRS
Sophia Antipolis, France
gilsanan@i3s.unice.fr

Mattia Piretti
Université Côte d'Azur
Laboratoire I3S / CNRS
Sophia Antipolis, France
mattia.piretti@etu.univ-cotedazur.fr

Melpomeni Dimopoulou
Université Côte d'Azur
Laboratoire I3S / CNRS
Sophia Antipolis, France
dimopoulou@i3s.unice.fr

Marc Antonini
Université Côte d'Azur
Laboratoire I3S / CNRS
Sophia Antipolis, France
am@i3s.unice.fr

***Abstract*—The aggressive growth of digital data threatens to exceed the capacity of conventional storage devices. The need for new means to store digital information has brought great interest in novel solutions as it is DNA, whose biological properties allow the storage of information at a high density and preserve it without any information loss for hundreds of years when stored under specific conditions. Despite being a promising solution, DNA storage faces two major obstacles: the large cost of synthesis and the high error rate introduced during sequencing. While most of the works focus on adding redundancy aiming for effective error correction, this work combines noise resistance to minimize the impact of the errors in the decoded data and post-processing to further improve the quality of the decoding.**

***Index Terms*—Image coding, DNA, barcode, inpainting, sequencing**

I. Introduction

The exponential growth of digital information during the past years has raised the need to develop novel solutions to fit the continuously growing storage requirements. Traditional storage devices are facing important limitations in capacity and longevity which prove them insufficient with respect to the increasing volume of the generated data. 90% of the data available on the internet has been generated during the past 2 years while 80% of this information is very infrequently accessed and it is characterized as "cold"! Despite the low demand of this data it still needs to be safely stored for security and regulatory compliance purposes. Any attempt to cover the cold data storage needs using conventional storage devices such as Hard Disk Drives or tape involves the construction of big data centers (such as the one recently built by Facebook) and would entail extremely high investments on storage systems as well as a complex infrastructure for security, power supply and environmental controls to guarantee optimal conditions in the building. In addition to that, the lifespan of storage media reaches 20 years at most, meaning that every few years, data must be migrated to new devices to ensure reliability. As an alternative solution, the use of DNA has arouse great interest for the past years [1]. DNA is a complex molecule that contains all the genetic information in living organisms constituted by the succession of four types of nucleotides (nts): Adenine (A), Thymine (T), Guanine (G) and Cytosine (C). The information density that DNA molecules can retain for long periods of time compared to existing devices makes it a promising candidate. Theoretically, DNA can store 455 Exabytes in 1 gram and under the optimal conditions, it persists for centuries without any damage. Such is the case of the woolly mammoth, whose genetic information has been decoded after 40.000 years preserved in permafrost. Broadly, the process of DNA storage can be described as follows. First of all, digital data has to be encoded into quaternary using the four DNA symbols A, C, T and G. DNA synthesis is an error-free procedure as long as the synthesized sequences are shorter than 300 nts, meaning that the encoded sequence must be cut into smaller chunks (oligos), which will contain some meta-information included in headers to preserve their location inside the initial sequence. Oligos are then synthesised and stored in hermetically sealed capsules to prevent contact with oxygen and water and guarantee their durability. Whenever the information needs to be retrieved, the stored oligos will be read using specific devices called sequencers. Although this is an error-prone process, sequencing error can be reduced if some constraints are respected:

- No homopolymers longer than a certain length. This length varies between 3 and 6 depending on the sequencing device.
- G,C content: % G,C $\leq$ % A,T
- No pattern repetition

Hence, achieving a better quality sequencing depends on using an encoding algorithm able to guarantee that the restrictions above are fulfilled as it happens with the code proposed in [2]. The high cost of the processes of DNA synthesis and sequencing remains one of the main drawbacks of DNA data storage. However, this problem started diluting during the past years with the release of nanopore technologies to sequence DNA strands [3], whose portability, affordability, and speed in data production that latest sequencers offer are breathing new life in the idea of DNA data storage allowing it to take one step closer to reality. However, this device also has the drawback of an increased error rate respect to other more expensive and slower sequencers. In order to deal with such noise, respecting the biological constraints mentioned before is not enough and while most of the works propose error correction techniques based on adding redundancy and consequently increasing the cost, our work combines a bio-

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 863320.

logically constrained encoding algorithm with noise resistance and post-processing techniques. In section II we present some interesting works proposed by the state of the art. In section III-B we discuss how the quaternary code is constructed and in III-C we briefly describe the mapping algorithm to assign the VQ indexes to DNA codewords. Sections III-D and IV present how the quaternary sequence is formatted to respect the synthesis restrictions and the creation of ad hoc barcodes to ensure the integrity of the headers in presence of noise, respectively. Section V describes the inpainting solution we proposed to restore the decoded image. Finally, results are presented in section VI and in section VII we discuss about the conclusions and future works.

II. State of the art

During the past years different methods have been proposed to address the problem of DNA storage. In [4] and [5] it has been proposed to divide the encoded data into segments containing overlapping regions so each fragment is represented multiple times. Other works like [6] suggest the use of Reed-Solomon codes to guarantee the recovery of missing oligos and [7] performs forward error correction by creating more than one dictionary so each symbol is encoded by more than one quaternary word. However, all those methods are based on introducing redundancy, which translates into an increase of the global cost. Other works focused on image storage combine compression techniques and DNA coding avoiding non-necessary redundancy. Recently, [8] proposed method for storing quantized images in DNA integrating Huffman coding in the encoding and applying image processing techniques to correct discolorations in the reconstructed image. In [9] we proposed an encoding scheme for quantized images that includes a biologically constrained code which respects the biological restrictions linked to DNA synthesis and sequencing and a new algorithm to map the vector quantization (VQ) indexes to DNA words in a way that minimizes the impact of errors in the decoded image, adding resistance to noise and preventing from relying completely on error correction. While the vast majority of the proposed works up to the date focus on Ilumina sequencing due to its higher accuracy, we introduce nanopore sequencing in our workflow, attempting to speed up the process as well as reduce its cost. In this paper we extend the work presented in [9] by applying the proposed mapping algorithm on the DWT subbands of an image and adding an extra step of image post-processing based on the inpainting algorithm in [10], reducing the visual distortion of the decoded image.

III. Overview of the overall storage workflow

A. The main workflow

Our proposed workflow, depicted in figure 1, consists of 4 main parts: compression, encoding, the biological processes of synthesis and sequencing, and decoding. Firstly, the data is compressed with a discrete wavelet transform (DWT) and each of the resulting subbands is quantized independently using Vector Quantization (VQ). To achieve an optimal compression, a bit allocation algorithm similar to the ones described in [11] provides the optimal quantization codebooks for each wavelet subband, allowing to store the maximum possible bits in each nucleotide for a given encoding rate. Next step of the workflow consists on encoding the quantized subbands into a quaternary code using an algorithm robust to the error-prone process of DNA sequencing which is described in section III-B. The encoded oligos are then processed biologically in vitro so that in the third part of the process they are being synthesized into DNA and stored safely into special capsules which keep the DNA safe from corruptions and data loss for hundreds of years. DNA synthesis is generally an error-free process as long as the oligos to be synthesized are not longer than 300 nucleotides (nts). This justifies the need for cutting our oligos into smaller DNA chunks with the process of formatting which is further described in III-D. The retrieval of the data stored in the DNA is called sequencing and it is likely to introduce errors while reading the DNA strands. This yields that the addition of extra redundancy is necessary to protect the reliability of the decoding. This redundancy is added thanks to a biological process called PCR[1] amplification before sequencing by creating many copies of the synthesized oligos. The sequencing provides us a set of many noisy copies of our synthesized oligos retrieved. The last part of the scheme is the decoding of the sequenced oligos. To be able to reconstruct the data from all the noisy copies, the pool of oligos is filtered, discarding all the sequences with corrupted headers that we are not able to locate in the initial encoded sequence. The remaining oligos are clustered according to their decodable headers and from each cluster we obtain a consensus which is created by assigning to each position the nucleotide that appears in the majority of the oligos belonging to the cluster. Finally, the initial data is reconstructed using the consensus sequences.

B. Creation of the quaternary code

To create the quaternary code we will use the encoding algorithm proposed in [12]. This algorithm creates words by merging predefined symbols, ensuring that the biological constraints of DNA are respected and consequently, robustifying the code to sequencing noise. Those symbols have been selected in such way that their concatenation always lead to viable words i.e. respect all the sequencing constraints. The codewords of the code are built using permutations of the elements from the two following dictionaries:

- $D_1 = \{AT, AC, AG, TA, TC, TG, CA, CT, GA, GT\}$
- $D_2 = \{A, T, C, G\}$

Concretely, we construct words of even length l by combining $\frac{l}{2}$ pair-symbols from D_1 and adding to the previous combinations an additional symbol from D_2 to obtain words of odd length. Building a quaternary codebook $\mathcal{C}^*$ in this way we ensure that the content of C and G in the final sequence will not go over 40% while also avoiding homopolymers. Given

[1]Polymerase Chain Reaction

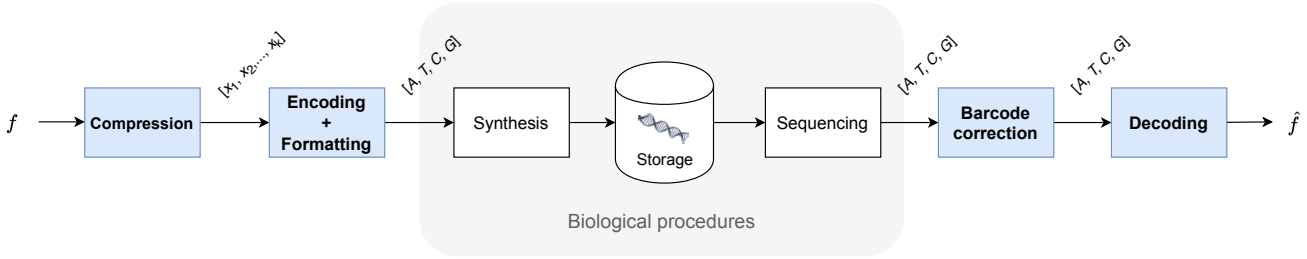

Fig. 1. General DNA coding workflow

that, the words prone to sequencing errors are discarded, making the constructed code incomplete but robust to distortion. Additionally, this encoding algorithm has the asset of allowing the encoding of any kind of input data and not only binary as it happens with most of the proposed solutions up to the date.

C. Controlled-mapping resistant to noise

In [9] we presented an extension of the method proposed in [13]. The main goal is to map the input vectors obtained from a Vector Quantization (VQ) algorithm and the quaternary codewords from our code in a way that the impact of an error in the quaternary sequence is minimized. The idea is to map quantization vectors with a small Euclidean distance to codewords which have a small Hamming distance. In this way, in case an error occurs during sequencing and assuming that the sequencing noise is reasonably small, a correct codeword will be transformed to another one which will have a small Hamming distance with the correct one. Consequently, the decoded erroneous vector will have a small Euclidean distance compared to the correct one, reducing the visual distortion that an error creates in the decoded image. Interested readers should refer to [9] and [13] for more informations.

D. Formatting of the oligos

The synthesis of DNA is a biological process which is error free if the oligos to be synthesized are shorter than 250-300 nts. It is therefore clear that when storing digital data into DNA, the encoded information needs to be cut into small chunks so to ensure reliability of the DNA synthesis. This division of the encoded data yields the need for adding at each oligo some information about the position of the data in the input image. This can be achieved by inserting some special headers in the oligo format. Dividing the information into smaller packets which include header fields as well as the creation of many copies to ensure robustness to errors resembles a lot to the way that data packets are transmitted in digital networks. The oligo formatting which has been used in this study is specific for the needs of the applied encoding workflow. More precisely, as briefly explained in previous sections, the encoding uses a DWT to reduce the spatiotemporal redundancies and each subband is independently quantized to vector indices using VQ. Those indices are then encoded into DNA codewords using a code which is robust to sequencing errors. Consequently, since the different DWT subbands are independently treated, for the formatting we create B different Subband Information Oligos (SIO) which will contain information about their specific encoding as for example the subband type and level and the VQ parameters of codebook size (K) and codewords length (ℓ). Furthermore, a Global Information Oligo (GIO) will contain all the global information for the encoding such as the image size and the number of DWT levels that were used in the encoding. Since the SIO and GIO only store headers for the decoding while the length of the oligo is relatively big, in those two types of oligos there is an empty field left which can be filled with any needed extra information. An example would be to replicate many times the same headers so to introduce some extra redundancy which can improve the decoding. This extra redundancy does not affect a lot the total encoding cost as this type of padding fields will occur only in the GIO and SIO oligos. Finally, the data will be cut and formatted into Data Oligos (DO) including an additional offset header which encodes the position of the data in the input image. The proposed formatting is illustrated in figure 2.

Fig. 2. General format of a Data Oligo. The header contains the information related to the type of oligo and DWT. The offset encodes the position of the data in the initial subband.

IV. DNA BARCODES

One of the main challenges when storing data into DNA is to ensure decodability of the data. Since the high throughput DNA sequencing is a procedure which introduces much noise in the oligos the full retrieval of the stored information can be at stake. One of the main problems when encoding images into DNA is the fact that if an error occurs in some important headers, the decoding becomes challenging if not impossible. Thus, robustifying those headers is highly important. In this work we have implemented an algorithm for constructing error correcting DNA barcodes with our proposed encoding algorithm that has been described in III-B. A set of barcodes $\mathcal{B}$ includes all those codewords among which the Levenshtein

distance [14], given by:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

is high enough to allow correction in case that errors of any type (insertion, deletion or substitution), up to a certain amount, corrupt some of those. DNA barcoded is a method which is highly used having diverse applications to biological studies. Interesting studies on DNA barcodes can be found in [15], [16] and [17]. To better understand the purpose of barcodes, we will analyse a simple example. Lets suppose the case depicted in figure 3, where some information encoded by one of the 5 possible codewords in the barcode set is corrupted by an insertion between the 3^{rd} and the 4^{rth} nucleotide. This will shift the last 3 nucleotides of the codeword pushing the last nucleotide out of the codeword frame. The produced codeword does not exist in the barcode set. This happens thanks to the high distance between the codewords of the barcode set. Thus if this codeword is received in the decoding it is clear that some error has occured. It is important to mention the fact that the barcode generation algorithm is implicit in the decoding process. To correct the error one can compute the Levenshtein Distance between the received codeword and all the codewords in the barcode set, correcting it to the closest codeword existing in the barcode set. For our work, this barcoding method can be used for robustifying headers containing information about the DWT such as the subband type and subband level, information about the VQ such as the number of vectors K and the length of vectors ℓ that was used for each subband, as well as information for the offsets for each chunk of data in each oligo. The construction of all the possible barcode sets using constrained codebooks created by our encoding algorithm (see III-B) is achieved by the following procedure:

- Initialization: Add the first codeword c_1 of our codebook $\mathcal{C}^*$ in a first codeword set $\mathcal{B}_1$, set $S = 1$.
- For each next codeword c_k for all $k = 1, 2, \ldots, K$: Check the distance between all the codewords in each of the existing barcode sets $\mathcal{B}_i$ with $i = 1, 2, \ldots, S$. If the distances between all codewords in a barcode set $\mathcal{B}_i$ is bigger than $d_{min} = 2 * \mu + 1$, with μ being the number of errors that can be corrected by the barcode, then this codeword is added to the barcode set $\mathcal{B}_i$. If the previous condition wasn't met for any existing barcode set, create a new barcode set $\mathcal{B}_{S+1}$ containing this codeword.

Once all the possible barcode sets have been created, we select the set $\mathcal{B}_i$ with the biggest cardinality. Figure 5(b) ilustrates the decoded image after barcode correction. From this figure it is clear that although the encoding of the headers using barcodes allows us to find the correct location of each oligo in the final image it is necessary to apply additional mechanisms of error correction to improve the quality of the retrieved data.

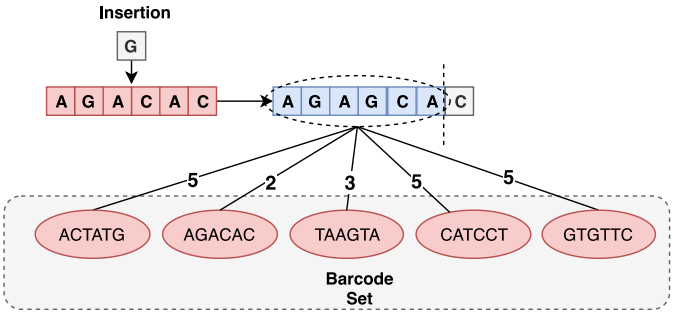

Fig. 3. Barcode Example

V. Improving visual distortion using inpainting

Image inpainting is an approach to repair and restore damaged images in a visually plausible way. The main difference from other restoration techniques (for example haze-removal) is that in the case of image inpainting there is no information that can be gained from inside the damaged area. All the information that the algorithms can employ has to come from either the undamaged parts of the image or, at most, from the contour between these and the target area. There are many families of inpainting techniques, each better suited to handle different types of damage. The one selected for our work is a Texture Synthesis algorithm, themselves a subcategory of more general Exemplar Based methods. These approaches aim to repair occlusions in the image by sampling and copying existing pixel values (referred to as patches) from the viable parts of the image onto the damaged ones. While these techniques are effective at repairing real world textures, they can cause artefacts due to the order in which patches are selected. As this could prove very problematic for our needs, we relied on the algorithm in [10]. This region filling approach utilizes an edge-driven method to order the patch selection and filling process, ensuring that the propagation of structure into the damaged area is consistent and avoids both artefacts and texture overshooting.

A. Proposed inpainting in wavelet domain

Our algorithm is specialized to handle the type of damage we incur in when decoding and deformatting DNA oligos that underwent a noisy sequencing process. As such, it differs from standard Texture Synthesis implementations in two ways. Firstly, it is built to be completely automatic. A series of damage identification steps try to identify and mark the target area of the image. Secondly, the inpainting is conducted on each single subband, obtained from the DWT decomposition, rather than on the whole image. Both of these differences are a product of the way in which we encode our images onto DNA. As the subbands are formatted and encoded separately, the noise is applied on each subband rather than on the whole image. This causes problems when trying to rely on traditional inpainting. First and foremost, damaged pixels in the more meaningful subbands can end up affecting the whole image making the definition of a mask, either automatically

or manually, impossible. The shape and size of the damage is also uncommon. Most inpainting algorithms are built to handle either large occluded areas or smaller, thinner damages. Our occlusions appear as either large spots and lines (caused by damage to a meaningful subband) or noisy checkerboard and crisscross patterns (caused by damages to the subbands carrying the details of the image). To sidestep all this, we act on the subbands before reconstructing them, which allows to employ a traditional inpainting approach in a more constrained environment and facilitates the damage detection.

B. Automatic detection of the damage in the subbands

The detection of the damaged areas is done using a 2-step algorithm, performed on each subband separately. The first step is done to detect errors caused by the erroneous decoding of single values, due to substitutions during the sequencing. This is done by comparing the value for each pixel to that of its neighbors. If the deviation between them is too high, it is likely that the pixel was damaged. This first step is not sufficient to detect extensive damage, for example in the case of one or more data (see III-D) being lost due to undecodable headers. In such cases, entire neighborhoods might be affected, and the first step is not able to reliably detect damaged areas. To handle this, a second detection step is performed. It can be observed that neighborhoods damaged in this way tend to have a very high internal variance. As such, during this second step the pixels whose neighborhoods present a standard deviation that is higher than average are detected as potentially damaged. At the end of the two steps we will have a binary mask that can be overlaid over the original subband (see figure 4). This identifies the pixels that the inpainting algorithm recognizes as the target area, and which will in turn will be filled from the source area, effectively the rest of the image.

VI. Experimental results

A. Simulation of the sequencing noise

As stated in previous sections, the sequencing of the DNA using nanopore technologies remains the main source of errors in the workflow presented in this work. The accuracy of this technology has improved from around 85% when the nanopore sequencing was first introduced to 95% or even higher [18]. We adjusted the rates of substitutions, insertions and deletions provided in [19] to fit the decreased error rate of nanopore sequencing, resulting in the following values: 2.3% of deletions, 1.01% of insertions and 1.5% of substitutions. The previous percentages were used for the simulation of the sequencing noise, in which 80% of the noise was concentrated in the first and last 20nt of each oligo [20]. It is important to denote that the simulation of the noise based on the statistics of real nanopore sequencing experiments constitutes a proof of concept of the methods presented in this work.

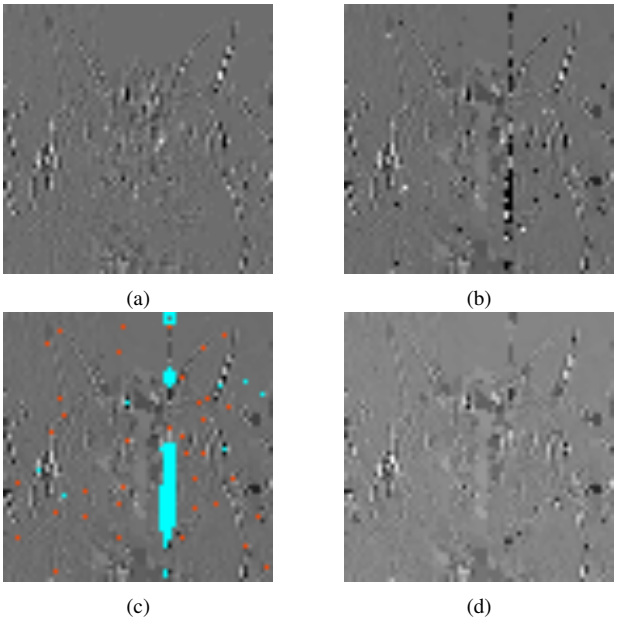

Fig. 4. Visual results on a DWT subband. (a) Original subband, (b) Noisy subband after nanopore sequencing simulation, (c) Output of the automatic detection of the damaged areas (1st step in red, 2nd step in blue), (d) Inpainted subband.

B. Workflow of the experiment

The image was compressed, quantized, encoded and formatted as explained in sections III and IV. We then simulated the nanopore sequencing noise and introduced it to the formatted oligos by creating 200 noisy copies of each input oligo. The purpose of this last step is to mimic the process of PCR amplification and production of multiple noisy reads by the nanopore sequencer. The result of this procedure is a set of multiple copies of the encoded oligos containing different error realisations as would occur in a real wet lab experiment. To decode the noisy data we start by correcting the barcoded information so to be able to distinguish each oligo type and locate them in the image. Once the barcoded headers are corrected, the noisy copies are clustered according to their headers. Each cluster is then cleaned by discarding the noisy oligos with high average Levenshtein distance to their cluster (which could be due to a low quality of the oligo or to an erroneous barcode correction). The remaining oligos in each cluster after filtering are then aligned and a consensus sequence is retrieved from each cluster as the most representative version of each oligo. The consensus algorithm is based on majority voting, assigning to each position inside the sequence the most frequent symbol along the cluster. Using those consensus sequences we reconstruct a noisy version of the input image which will then be post-processed for smoothing the damaged areas. We used the proposed algorithm for the automatic detection of the damage in each DWT subband and the selected areas are inpainted to provide the final result of the workflow.

(a) (b) (c)

Fig. 5. Visual results of the experiment: (a) Quantized image without sequencing noise, (b) Visual impact of sequencing noise in the image encoded using the controlled mapping proposed in [9] after barcode correction, (c) Post-processed image using inpainting.

C. Results of the simulation

For our experiments, we carried out the process described above using a compression quality of 4.9708 bits/nt. Figure 5(a) shows the quantized image after compression. This image has a PSNR = 48.12 dB and a SSIM = 0.991 compared to the original uncompressed image. Figure 5(b) shows the decoded image with sequencing noise and 5(c) corresponds to the final image after inpainting. The post-processing of the image led to a PSNR = 38.7 dB and a SSIM = 0.94, which constitute an improvement of 2.5 dB and 0.2 respectively compared to the original quantized image with sequencing noise.

VII. Conclusions

In this paper we presented a general workflow to store digital images into DNA combined with post-processing to further improve the quality of the decoding. While most works up to the date focus on Ilumina sequencing due to its higher accuracy, we introduce nanopore sequencing in our workflow, attempting to speed up the process as well as reduce its cost. More precisely, we carried out a study of the performance of a noise-resistant DNA-coding algorithm proposed in [9] in the presence of simulated nanopore sequencing noise. This encoding optimally assigns Vector indices of an image that has been quantized using VQ so to reduce the visual impact of sequencing errors. We also introduce an algorithm for detecting any remaining damaged areas and apply an inpainting algorithm to improve the quality of the decoding using post-processing. The results of this study are very promising given the high error-rates imposed by the nanopore sequencers showing significant visual improvement. However, since this study provides results on simulated nanopore noise, a wet-lab experiment is a priority future step to verify in practice the efficiency of the proposed encoding.

References

[1] Andy Extance, "How dna could store all the world's data," Nature, vol. 537, no. 7618, 2016.

[2] Melpomeni Dimopoulou, Marc Antonini, Pascal Barbry, and Raja Appuswamy, "A biologically constrained encoding solution for long-term storage of images onto synthetic DNA," in EUSIPCO, 2019.

[3] Miten Jain, Hugh E Olsen, Benedict Paten, and Mark Akeson, "The oxford nanopore minion: delivery of nanopore sequencing to the genomics community," Genome biology, vol. 17, no. 1, pp. 239, 2016.

[4] Nick Goldman, Paul Bertone, Siyuan Chen, Christophe Dessimoz, Emily M LeProust, Botond Sipos, and Ewan Birney, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," Nature, vol. 494, no. 7435, pp. 77, 2013.

[5] James Bornholt, Randolph Lopez, Douglas M Carmean, Luis Ceze, Georg Seelig, and Karin Strauss, "A DNA-based archival storage system," ACM SIGOPS Operating Systems Review, vol. 50, no. 2, pp. 637–649, 2016.

[6] Robert N Grass, Reinhard Heckel, Michela Puddu, Daniela Paunescu, and Wendelin J Stark, "Robust chemical preservation of digital information on DNA in silica with error-correcting codes," Angewandte Chemie International Edition, vol. 54, no. 8, pp. 2552–2555, 2015.

[7] Meinolf Blawat, Klaus Gaedke, Ingo Huetter, Xiao-Ming Chen, Brian Turczyk, Samuel Inverso, Benjamin W Pruitt, and George M Church, "Forward error correction for DNA data storage," Procedia Computer Science, vol. 80, pp. 1011–1022, 2016.

[8] Chao Pan, SM Yazdi, S Kasra Tabatabaei, Alvaro G Hernandez, Charles Schroeder, and Olgica Milenkovic, "Image processing in dna," arXiv preprint arXiv:1910.10095, 2019.

[9] "A quaternary code mapping resistant to the sequencing noise for dna image coding," in MMSP (submitted), 2020.

[10] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," IEEE Transactions on Image Processing, vol. 13, no. 9, pp. 1200–1212, 2004.

[11] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers (speech coding)," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 36, no. 9, pp. 1445–1453, 1988.

[12] Melpomeni Dimopoulou, Marc Antonini, Pascal Barbry, and Raja Appuswamy, "DNA coding for image storage using image compression techniques," in CORESA 2018, 2018.

[13] JR Boisson De Marca, NS Jayant, et al., "An algorithm for assigning binary indices to the codevectors of a multi-dimensional quantizer," in 1987 IEEE International Conference on Communications (ICC'87). , 1987, pp. 1128–1132.

[14] Vladimir I Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in Soviet physics doklady, 1966, vol. 10, pp. 707–710.

[15] John A Hawkins, Stephen K Jones, Ilya J Finkelstein, and William H Press, "Error-correcting dna barcodes for high-throughput sequencing," bioRxiv, p. 315002, 2018.

[16] Tilo Buschmann and Leonid V Bystrykh, "Levenshtein error-correcting barcodes for multiplexed dna sequencing," BMC bioinformatics, vol. 14, no. 1, pp. 272, 2013.

[17] Daniel Ashlock and Sheridan K Houghten, "Dna error correcting codes: No crossover.," in 2009 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology. IEEE, 2009, pp. 38–45.
[18] Oxford Nanopore Technologies, "New research algorithms yield accuracy gains for nanopore sequencing," 2020, (https://nanoporetech.com/about-us/news/new-research-algorithms-yield-accuracy-gains-nanopore-sequencing).
[19] Jingwen Zeng, Hongmin Cai, Hong Peng, Haiyan Wang, Yue Zhang, and Tatsuya Akutsu, "Causalcall: Nanopore basecalling using a temporal convolutional network," Frontiers in Genetics, vol. 10, pp. 1332, 2020.
[20] Miten Jain, Ian T Fiddes, Karen H Miga, Hugh E Olsen, Benedict Paten, and Mark Akeson, "Improved data analysis for the minion nanopore sequencer," Nature methods, vol. 12, no. 4, pp. 351–356, 2015.