A quaternary code mapping resistant to the sequencing noise for DNA image coding

Melpomeni Dimopoulou, Eva Gil, Marc Antonini

A quaternary code mapping resistant to the sequencing noise for DNA image coding

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

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

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

***Abstract*—The exponential growth in the generation of digital information creates a big challenge for data storage given the capacity limitations of conventional storage devices. Recent works have proposed DNA as a means of digital data storage proposing a novel solution for long-term storage. Although having many advantages, DNA storage is a challenging topic due to the error-prone process of DNA sequencing (reading). To deal with this error most existing works focus on the introduction of error-correction methods. However, most of those methods introduce important redundancy without promising full error correction for the widely used sequencing method using the Nanopore sequencer. This work focuses on noise resistance rather than error-correction proposing a new algorithm for optimally assigning VQ indices to DNA codewords while reducing the visual impact of substitution errors that are caused during sequencing.**

I. Introduction

Data explosion is leading to an increasing demand of storage resources which is inconsistent with the capacity and longevity limits of traditional storage units. Interestingly enough, studies estimate that 90% of the internet data has been only generated in the last two years while 80% of this information consists the so called "cold" data [1] which is very rarely or even never accessed but still needs to be safely stored for security and compliance reasons. The archiving of cold data requires tons of hardware which should be replaced every 5 to 10 years to ensure reliability. Every year millions of data centers worldwide are wasting huge amounts of energy and money for the migration of data into new back-up drives in order to protect all this information. Therefore, it is clear that the need of finding new ways for storing data is crucial and to this end recent studies have proposed the storage of data in form of DNA strands as a potential alternative to the current methods. The biological properties of the DNA molecule allow an efficient data storage for hundreds of years without loss of information and without additional costs to maintain its correctness. An interesting example of DNA's incredible longevity is a woolly mammoth which had been trapped into permafrost whose DNA has been successfully decoded after 40.000 years. DNA data storage is clearly a very promising multidisciplinary solution which consists of both data science algorithms and some delicate biological processes. More precisely, the DNA data storage can be described by the following main steps. To begin with, the digital data has to be transformed into a quaternary encoding of the symbols A, T, C and G which denote the four different building blocks (nucleotides) of DNA. This sequence is then synthesized into DNA by the biological process of DNA synthesis. This process is performed in vitro and is error free as long as the DNA strands to be synthesized are shorter than 300 nucleotides (nts). Consequently, before synthesis the encoded quaternary strand needs to be cut into small chunks (oligos) and formatted to contain special headers that denote the position of each oligo in the initial long sequence. The synthesized oligos are then stored into some safe containers (capsules) which prevent contacts with water and oxygen and promise long term preservation. To retrieve the stored information in the decoding part of the workflow, the stored DNA strands will be read in vitro using the biological process of DNA sequencing. The sequencers are special machines which can read the DNA content of the capsules providing to the output the quaternary data. Finally, the quaternary sequence is transformed back to its initial representation to retrieve the stored information. The biggest challenge of this workflow is imposed by the process of DNA sequencing as the sequencers can introduce errors in the decoding process. This error can be reduced if the encoding performed in the first step of the workflow respects the following rules:

- No homopolymers: A nucleotide should not be repeated more than three times in a row.
- G,C content: % G,C $\leq$ % A,T
- No pattern repetition: Repetition of short patterns should be avoided.

Consequently, the encoding algorithm should be carefully selected to provide an encoding that respects all the above rules. In this paper, we present an extension of our previously proposed encoding [2] by applying a more sophisticated mapping function which is resistant to substitution errors which can be created during the sequencing. This special mapping allows the reduction of the total distortion in the decoded image in the case that it is corrupted by errors. In section II we present some interesting works proposed by the state of the art. In section III we describe the algorithm which is

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

978-1-7281-9320-5/20/$31.00

used for constructing a robust quaternary code while in section IV we explain the mapping algorithm that is used for the encoding into a quaternary sequence. More precisely in IV-A we introduce some basic notions to facilitate understanding of the following section IV-B where we present the proposed algorithm for optimally mapping input quantization vectors to the quaternary codewords. In section VI we present the methods used in our experiments and we demonstrate our experimental results. Finally in section VII we discuss about the conclusions and future works.

II. State of the art

As discussed in the previous section DNA data storage is a very promising yet very challenging process as the decoding of DNA is error-prone and thus the encoding is constrained by important restrictions and should provide a code which is robust to errors. In the state of the art there have been proposed some very interesting algorithms for encoding an input stream into a robust quaternary representation ([3], [4], [5], [6], [7], [8]). In our latest work in [2] we proposed the use of Vector Quantization to further improve the efficiency of the compression. In this work we propose an extension of our solution in [2] by adding a special mapping algorithm which is resistant to noise. This mapping is strongly inspired by the works in [9] and [10] where the authors proposed an algorithm for assigning binary words to vectors of a multidimensional quantizer in such a way so that in case of an error the decoded vector would be close to the correct one improving this way the quality of the decoded signal. In this paper we extend this algorithm to a constrained quaternary dictionary to be applied for DNA coding.

III. Creating the quaternary code

For the creation of the quaternary code we will use the algorithm proposed in our previous work in [8]. The main idea of this algorithm lies on the construction of words using some pair-symbols, the concatenation of which can provide viable words. As viable words we consider the words which when assembled in an encoded strand they provide a quaternary sequence which respects all the constraints imposed by the process of DNA sequencing and will therefore be robust to sequencing noise. The codewords of the code are build using permutations of the elements from the two following dictionaries:

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

More precisely, words of an even length l are constructed using all possible permutations of $\frac{l}{2}$ pair-symbols from dictionary D_1. Equivalently, words of an odd length l are constructed using all possible permutations using $\frac{l-1}{2}$ pair-symbols from D_1 to represent the first $l-1$ nucleotides and a single symbol from D_2 to represent the last nucleotide of the words. This method for building quaternary words ensures that the codewords produced contain an appropriate percentage of G and C which does not exceed 40-60% while also avoiding homopolymers. It is clear that due to the biological restrictions imposed, some quaternary words will be omitted from the produced DNA code as they are considered as non-viable for being more prone to sequencing errors. Consequently, the constructed code will be less efficient than a complete quaternary code but it is more robust to distortion. Another important asset of this algorithm is the low complexity while it provides a valid fixed length code that can be applied to any type of input data in contrast to most of the existing state of the art algorithms that can only be applied to binary.

IV. A controlled code-mapping resistant to sequencing noise

A. Introduction to the proposed mapping

Sequencing of DNA strands is an error-prone process and can create insertions, deletions and substitutions of nucleotides causing important noise in the decoded data. Illumina and Nanopore are the two most widely used types of sequencers. While Illumina is the most reliable solution providing a low error-rate in the decoding, scientists are turning towards the more recent Nanopore machine due to its low cost, fast throughput and user-friendly, small size. Unfortunately this type of sequencer introduces more noise in the decoded sequence compared to the Illumina, creating the need for finding new ways to deal with errors in the decoded sequence. Existing studies have proposed some interesting methods for error correction by adding redundancy in the encoded sequence, while always using robust encoding to reduce the possibility of sequencing noise. However, the high error-rate introduced by sequencers can't easily be completely eliminated. Thus, in this study we take a very first step in proposing an algorithm which provides a solution resistant to sequencing noise. In other words, this algorithm aims to reduce the impact of the remaining sequencing errors. In this first attempt, we will only focus on the noise of substitutions.

In [9] DeMarca *et al.* have proposed an algorithm for assigning binary words to codevectors of a multi-dimensional quantizer in such a way so to be resistant to channel noise. Inspired by this idea and under the assumption that the noise of substitutions introduced by sequencers can be modeled as the one introduced by noisy channels of transmission we extend the algorithm proposed in [9] which is applied to binary codewords to an application which is using a quaternary code. Before describing the proposed mapping algorithm it is important to introduce some basic notions and definitions. Let $\mathcal{V} = \{v_1, v_2, \dots, v_K\}$ be a set of ℓ-dimensional vectors v_k. To encode those K vectors, a code $\mathcal{W} = \{w_1, w_2, \dots, w_K\}$ of K quaternary codewords is necessary. Since this study focuses on DNA data storage this code $\mathcal{W}$ is constructed using the dictionaries D_1 and D_2 providing a constrained encoding as explained in section III. Consequently, the size K of the code $\mathcal{W}$ is restricted to specific values as the words should be created according to the constraints imposed by the process of DNA coding. Consequently $K \in \{k_1, k_2, \dots\}$ where:

$$k_1 = 10 \text{ and } k_{i+1} = \begin{cases} 4k_i, & i\text{: odd} \\ 10k_{i-1}, & i\text{: even} \end{cases}$$

Given the two sets $\mathcal{V}$ and $\mathcal{W}$, one can define a mapping function $\mathcal{M} : \mathcal{V} \mapsto \mathcal{W}$ for assigning codewords from $\mathcal{W}$ to codevectors of $\mathcal{V}$. Using this function one can encode a sequence of input blocks (codevectors) to a quaternary representation of A, T, C and G.

Assuming a substitution error in the encoded quaternary stream, some correct codeword w_c will be transformed to an erroneous codeword w_e and under the hypothesis that the error rate produced by the sequencer will be reasonably small, the Hamming distance between the correct and the erroneous codeword will be $d_H(w_c, w_e) = 1$. Therefore, for each correct codeword w_c of length l there are different erroneous codewords of distance 1. The set of all possible erroneous codewords can be, according to coding theory, represented in the Hamming Space as a sphere $H(w_c)$ of radius 1 the center of which is the correct codeword w_c. An example of such a sphere is depicted in figure 1. In the case where the code $\mathcal{W}$ contained all the possible arrangements of A, T, C, G, there would be 4^l different spheres the cardinality of which would be $|H(w_c)| = 3l$. However, as explained in section III, since the code used for DNA coding excludes some words which can't be viable due to the encoding constraints, in this work we consider K different spheres with varying cardinality. In other words, some codewords w_e that would normally belong to some sphere of center w_c might be omitted due to the fact that they do not respect the rules of DNA coding. As a result, a substitution can cause two different possible types of error:

- **Decodable error:** The substitution transforms a correct codeword w_c to an erroneous word w_e which exists in the constrained code $\mathcal{C}$ proposed in section III and therefore $w_e \in H(w_c)$. Decoding will then provide an erroneous vector v_e instead of the correctly decoded vector v_c. In the case that the Euclidean distance $d_E(v_c, v_e)$ is small, the produced error will not significantly affect the visual quality of the decoded image. To this end we propose a special mapping algorithm which allows the assignment of codeword indices to vectors such that the possible errors will lead to a minimum distortion. This algorithm is analytically described in section IV-B.
- **Undecodable error:** The substitution transforms a correct codeword w_c to an erroneous codeword w_e which does not exist in the constrained code $\mathcal{C}$. In this case decoding is not possible and thus the application of some error correction is necessary to allow decoding. The applied correction techniques are further described in section V.

B. The proposed controlled mapping algorithm

The main goal of this algorithm is to map codewords w_q from a code $\mathcal{W}$ which differ from each other at exactly one position (Hamming distance of 1) to codevectors v_q from a vector set $\mathcal{V}$ which are close in terms of Euclidean distance. The objective of this mapping lies in the fact that in case of an error during sequencing and assuming that the sequencing noise is small enough, a correct codeword will be transformed to another one which will have a small Hamming distance with the correct one. Thus, if those two codewords are assigned to input vectors which are close, the error will not significantly affect the image quality. To further define which vectors can be considered as close vectors we introduce a set $S(v)$ which contains the closest vectors to the vector v in terms of Euclidean distance. The central idea of the proposed mapping is to assign codewords of the same sphere to vectors which belong to the same neighborhood as shown in figure 3. However, such an assignment is not possible for every neighborhood S and thus it is necessary to perform this assignment according to some priority. To this end, we define an empirical function $F(v)$ introduced by [9] for a vector v as:

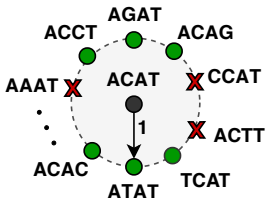


Fig. 1: Example of a Hamming sphere. The cross-elements denote non viable words that would belong in the Hamming sphere but are omitted due to the constrained quaternary code.

$$F(v) = \frac{p(v)}{\alpha^{\beta}(v)}$$

where $p(v)$ is the probability of v in the input sequence, and

$$\alpha(v) = \sum_{j|v_j \in S(v)} d_E(v_j, v)$$

with $\beta \geq 0$ a trade-off parameter. Therefore vectors with a higher value for F are considered to belong in a denser neighborhood and should consequently get higher priority to be assigned to the same sphere of words. The algorithm can be very roughly described by the following parts:

- For each codeword w_q: Create a sphere containing the B_q codewords which have a Hamming distance of 1 compared to w_q. Define $B = \max_i(B_i)$, $i \in \{1, \ldots, L\}$.
- For each input vector v_q:
  - Find a set $S(v_q)$ of B neighboring vectors v_n which are the closest to v_q in terms of Euclidean distance $d_E(v_q, v_n)$.
  - Compute the following empirical function $F(v_q)$.
- Use algorithm 1 to progressively perform assignment of vectors v_q to codewords w_q such that vectors with a bigger $F(v_q)$ as well as their neighboring vectors $v_n \in S(v_q)$ will be assigned to the same sphere of codewords whenever possible as depicted in figure 1. If this is not possible assignment is performed such that vectors are assigned to codewords which have a small Hamming distance from the codewords already assigned to their neighboring vectors.
- Optimization of the first assignment:
  - Exchange the previously mapped codewords between each pair of vectors.

– For each exchange check if the average distortion has decreased. If true keep this change, else keep the initial state of mapping.

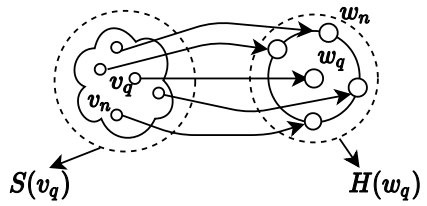


Fig. 2: Mapping vectors to codewords

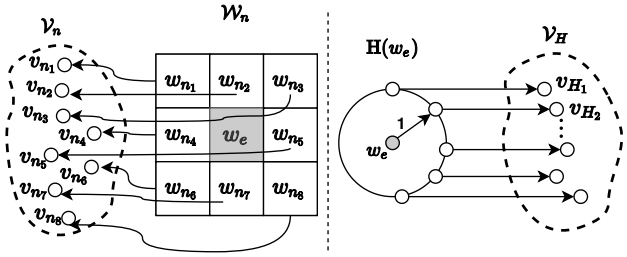


Fig. 3: ACD decoding schema

V. Proposed decoding of undecodable words

As discussed in section IV-A in the case in which a substitution error creates an undecodable word it is necessary to employ some error correction to allow decoding. In this section we will describe two possible methods for correcting undecodable words. The first method which we will call Simple Correction Decoding (SCD) is working according to the two following assumptions.

- A substitution error will produce an erroneous word which will not differ from the original one in more than one nucleotide.
- Using the mapping algorithm proposed in the previous section it is probable that codewords that are close in terms of Hamming distance are assigned to codevectors which are close in terms of Euclidean distance.

Let us assume a word w_c which is corrupted by a substitution error producing an undecodable erroneous codeword w_e. Given the first assumption, $d_H(w_c, w_e) = 1$. To allow decoding, w_e should be first corrected to a decodable word w_d. The idea behind SCD is to correct w_e to some word w_d we chose as most appropriate the codeword $w_d \in \mathcal{W}$ which is assigned to the median vector-index of the input image. This choice has two possible outcomes. Either w_d will be equal to w_c creating no visual distortion in the decoded image, or $w_d \neq w_c$ and $d_H(w_c, w_d) = \{1, 2\}$.

By making use of one extra assumption which indicates that since the input data is an image there can be correlations between neighboring elements we propose a second more sophisticated decoding, which we will call Advanced Correction Decoding (ACD). ACD works in two decoding cycles. In the first cycle it performs a first decoding by omitting the undecodable words. Then in a second round the algorithm decodes the remaining undecodable words using the following steps:

- Define as $\mathcal{V}_H$ the set of vectors v_{H_j} to which have been assigned the words in $H(w_e)$.
- Define $\mathcal{W}_n$ as the set of neighboring words w_{n_i} around w_e with $0 \leq i \leq 8$ as depicted in figure 3
- Define set $\mathcal{V}_n$ as the set of vectors v_{n_i} with $0 \leq |\mathcal{V}_n| \leq 8$ to which the codewords in $\mathcal{W}_n$ have been assigned.
- Define set $S = \mathcal{W}_n \cap H(w_e) = \{w_1, w_2, \ldots, w_z\}$
  - If $|S| = 1$, $w_e \leftarrow w_1 \in S$
  - If $|S| > 1$
    - Define $f(w_z)$ as the frequency of a word w_z, $w_z \in S$
    - $w_e \leftarrow w_z \in S$ such that $w_z = \arg\max_z f(w_z)$
  - if $|S| = \varnothing$
    - Compute $D(v_{H_j}) = \frac{\sum_{i=1}^{|\mathcal{V}_n|} d_E(v_{H_j}, v_{n_i}) f(\mathcal{M}^{-1}(v_{n_i}))}{|\mathcal{V}_{H_j}|}$, for $v_{H_i} \in \mathcal{V}_H$, $\forall v_{H_j} \in \mathcal{V}_{\mathcal{H}}$
    - $w_e \leftarrow w_d$ where $w_d = \mathcal{M}(v_d)$ with $v_d := \arg\min_{v_{H_j}}(D(v_{H_j}))$

VI. Experimental results

In our study we performed multiple comparisons in order to prove the efficiency of the proposed mapping algorithm. For each comparison we ran 10 different realisations of random error added on the same input image and have plot the improvement in PSNR in function of the error rate. The points in each plot correspond to the mean value of the 10 different realisations of noise. As explained in IV-B, the main assumption behind the proposed mapping algorithm is the restriction of the errors to one error per quaternary word. Thus, for the noise addition, we used a uniform random distribution to select a percentage of words w_c from the encoded sequence and we introduced an error of one nucleotide in each w_c. To begin with, to set an upper bound in the performance of the controlled mapping we tested the improvement of PSNR compared to the non-controlled case by adding only decodable errors. More specifically, in this experiment we added random substitutions making sure that the created erroneous words exist in the constrained codebook. Figure 4a, shows the improvement of PSNR (ΔPSNR) for different parameters of K and ℓ for VQ in function of the introduced error rate. As observed in this figure, the controlled mapping can improve the PSNR by at least 3 and at most 7.5 dB in the optimal case where no corrections are needed. The addition of only decodable errors might not be a realistic case in practice but this plot reveals the true potential of the proposed mapping algorithm. This is because in the case of an undecodable error the correct codeword will be transformed to an erroneous one w_e with $d_H(w_c, w_e) = 1$. This second more realistic experiment is presented in figure 4b which depicts the improvement of PSNR in function of different error rates comparing the controlled and non-controlled mapping using SCD for correcting the undecodable words. As expected the undecodable errors decreases the improvement of PSNR providing a best case of ΔPSNR$=$ 3dB. To improve those results

and since SCD is a simple decoding where the undecodable word is corrected to the codeword mapped to the median vector-index of the quantized input image, we also tested the case of ACD. The result is shown in figure 4c which depicts the improvement in PSNR comparing the controlled mapping using ACD to the non-controlled mapping using SCD. This case reveals the total improvement which can be achieved providing a ΔPSNR betwenn 2.5 and 5dB. On the right column of figure 4 we also present the visual improvement for each of the previous cases on the input image which was used in our experiments. The visual results are presented for an error rate of 3%, which is the percentage of substitutions introduced by the sequencer of the Nanopore (according to a study in [11]), and for the values of K and ℓ which provided the best PSNR improvement in each case.

VII. Conclusion

In this work we have introduced a new noise resistant controlled mapping for optimally assigning DNA quaternary words to vectors obtained by VQ. This new mapping has been tested on a constrained codebook according to the needs of DNA data storage. The obtained results exhibit a very promising increase of PSNR providing a noticeable visual improvement. An interesting future step of this work would be extending this algorithm to treat other kinds of errors such as insertions and deletions.

References

[1] M. Shacklett, "Unstructured data: A cheat sheet," 2014, iDC Data Age 2025 report, sponsored by Seagate;.

[2] M. Dimopoulou and M. Antonini, "Efficient Storage of Images onto DNA Using Vector Quantization," Data Compression Conference (DCC) 2020, Mar. 2020, poster. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02549709

[3] 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, p. 77, 2013.

[4] M. Blawat, K. Gaedke, I. Huetter, X.-M. Chen, B. Turczyk, S. Inverso, B. W. Pruitt, and G. M. Church, "Forward error correction for DNA data storage," Procedia Computer Science, vol. 80, pp. 1011–1022, 2016.

[5] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. 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.

[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] S. H. T. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access DNA-based storage system," Scientific reports, vol. 5, p. 14138, 2015.

[8] M. Dimopoulou, M. Antonini, P. Barbry, and R. Appuswamy, "A biologically constrained encoding solution for long-term storage of images onto synthetic DNA," in EUSIPCO 2019, 2019.

[9] J. B. De Marca, N. 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.

[10] N. Farvardin, "A study of vector quantization for noisy channels," IEEE Transactions on Information Theory, vol. 36, no. 4, pp. 799–809, 1990.

[11] J. Zeng, H. Cai, H. Peng, H. Wang, Y. Zhang, and T. Akutsu, "Causalcall: Nanopore basecalling using a temporal convolutional network," Frontiers in Genetics, vol. 10, p. 1332, 2020.

Algorithm 2 Phase 1: For the remaining indices

Definitions:

Set $\mathcal{V} = \{v_1, v_2, \ldots, v_K\}$ of codevectors v_k , $|\mathcal{V}| = K$
Set $\mathcal{W} = \{w_1, w_2, \ldots, w_L\}$ of quaternary words w_l, $|\mathcal{W}| = L$
Define a mapping function $\mathcal{M} : \mathcal{V} \mapsto \mathcal{W}$
Set $C \subseteq \mathcal{V}$ of vectors $v_i : \mathcal{M}(v_i) = \varnothing$
$H(w_l)$: Set containing codeword w_l and the B_i codewords w_n that differ from w_l in one nucleotide (Hamming distance of 1)
Define $B = \max_i(B_i)$ with $i \in \{1, 2, \ldots, L\}$
$S(v_k)$: Set containing vector v_k and its B closest neigboring vectors v_n
Function $\alpha(v_q) = \sum_{j|v_j \in S(v_q)} d(v_j, v_q)$ with $\beta \geq 0$ a trade-off parameter.
Empirical function: $F(v_q) = p(v_q)/\alpha^{\beta}(v_q)$ where $p(v_q)$ is the probability of v_q in the input sequence

Phase 0: For the first (B+1) indices

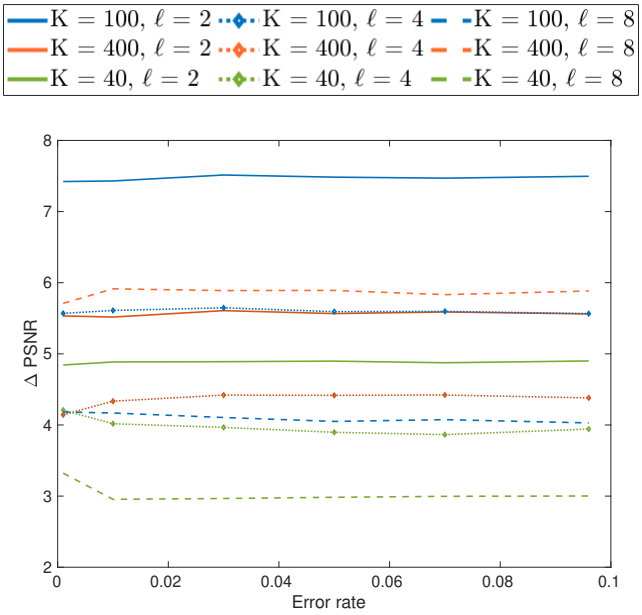
Initialise: $C = \mathcal{V}$
$v_q := \{v_i \in C : v_i = \arg\max_v F(v)\}$
$\mathcal{M}(v_q) = w_q$
$\mathcal{M}(v_n) = w_n$, $\forall v_n \in S(v_q)$, $w_n \in H(w_q)$

Phase 1: For the remaining indices

while $C \neq \varnothing$ **do**
  $C - \{v_q, v_n\}$
  $v_q := \{v_i \in C : v_i = \arg\max_v F(v)\}$
  if $\mathcal{M}(v_n) = \varnothing, \forall v_n \in S(v_q)$ **then**
    if $\exists w_q : \forall w_n \in H(w_q), \mathcal{M}^{-1}(w_n) = \varnothing$ **then**
      $\mathcal{M}(v_q) = w_q$
      $\mathcal{M}(v_n) = w_n$,$\forall v_n \in S(v_q)$, $\forall w_n \in H(w_q)$
    else
      $H := \arg\max_s(|H_s|)$, $\forall s \in \{1, 2, \ldots, K\}$,
      $\mathcal{M}^{-1}(H_s) = \varnothing$
      $\mathcal{M}(v_q) = w_q$
      $\mathcal{M}(v_n) = w_n, \forall w_n \in H(w_q) : \mathcal{M}^{-1}(w_n) \neq \varnothing$
      and $v_n = \arg\min_v(d(v_q, v))$ with $v \in S(v_q)$
    end if
  else
    if $\mathcal{M}(v_q) = \varnothing$ but $\mathcal{M}(\Gamma_q) \neq \varnothing$ with $\Gamma_q \subseteq S(v_q)$ **then**
      if $\exists w_q$: $\mathcal{M}(\Gamma_q) \subseteq H(w_q)$ **then**
        $\mathcal{M}(v_q) = w_q$
      else
        $H := \arg\max_s(|H_s \cap \mathcal{M}(\Gamma_q)|)$ $\forall s \in \{1, 2, \ldots, K\}$
        if $|\arg\max_s(|H|)| \geq 2$ **then**
          Let $w_{nj}, j = 1, 2, \ldots,$ be the tied indices
          Define $a = v_n | \mathcal{M}(v_n) \in \mathcal{M}(\Gamma_q) \cap H(w_q)$
          Define $b = v_n | \mathcal{M}(v_n) \in \mathcal{M}(\Gamma_q) \cap H(w_{nj})$
          Assign to vector v_q a word w_q such that:

$$\sum_a d(v_n, v_q) = \min_q \sum_b d(v_n, v_q)$$

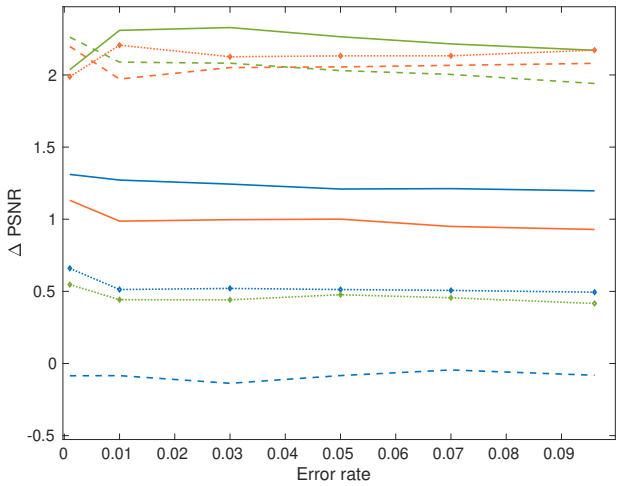
        end if
      end if
    end if
  end if
end while



Non-controlled mapping
PSNR = 15.63 dB

Controlled mapping
PSNR = 23.07 dB

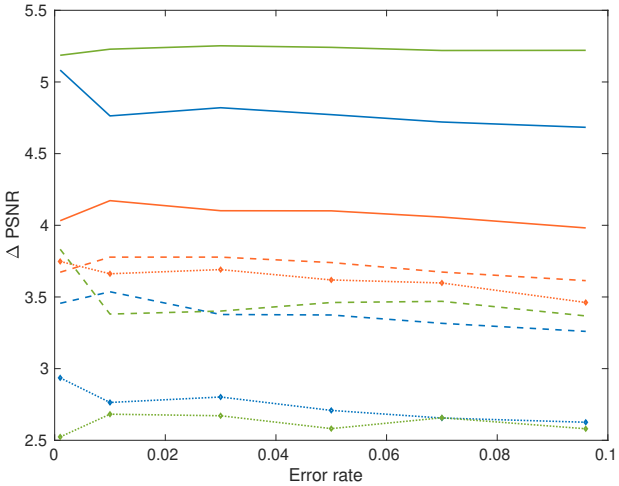
a) Controlled mapping vs.non-controlled mapping (decodable errors only)



Non-controlled mapping + SCD
PSNR = 16.72 dB

Controlled mapping + SCD
PSNR = 19.12 dB

b) Controlled mapping SCD vs. non-controlled mapping SCD



Non-controlled mapping + SCD
PSNR = 16.61 dB

Controlled mapping + ACD
PSNR = 21.61 dB

c) Controlled mapping ACD vs. non-controlled mapping SCD

Fig. 4: Left column: PSNR improvement in function of the error rate for different cases of VQ parameters K and ℓ with $\beta = 0$. This value of β was reported in [9] to perform optimally for a non-uniform input probability distribution. The three different plots correspond to different experiments as explained in the legends a, b and c. Right column: Visual results for each of the cases a, b and c for the values of K and ℓ that provided the best performance. The selected error rate for those images is 3% accordingly to the percentage of substitutions caused by the Nanopore sequencer [11].