Introduction to DNA data storage

Xavier Pic

January 22nd, 2025

EURECOM
Sophia Antipolis

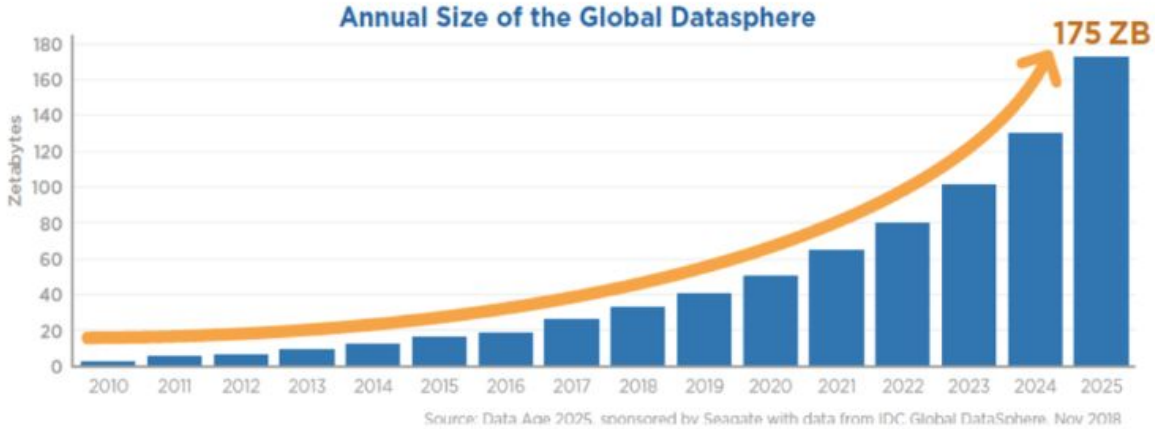
Contents

- Motivations and principle
- Examples of DNA coders
  - Channel Coders (SFC4 and C3)
  - JPEG-based Image source coders (JPEG DNA SFC4 and JPEG2000 DNA)
  - Direct transcoders adapted to DNA (JPEG DNA VM, CMOSS)

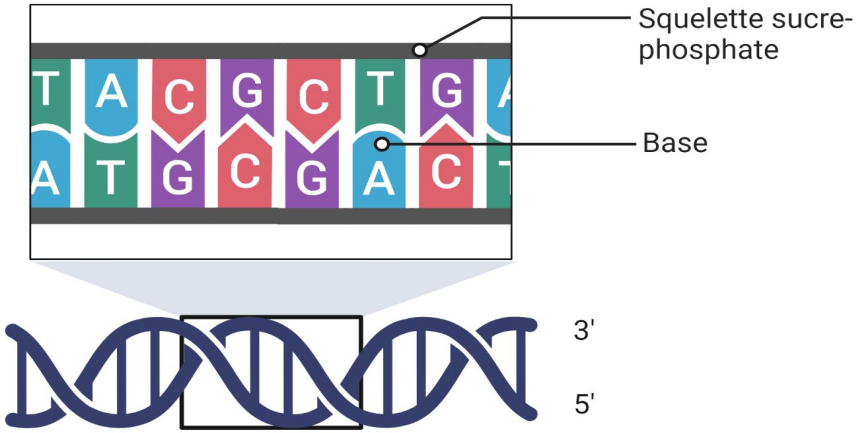
Motivations and principle

Conventional Storage Infrastructures

- Data storage demand is **ever expanding**
- Conventional Storage **cannot** be produced in **sufficient quantity**
- High **energy consumption**



The DNA molecule



Bases nucléiques

Purines:

A Adenine

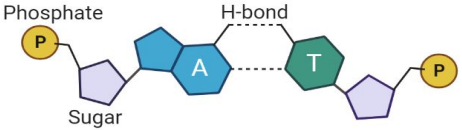
G Guanine

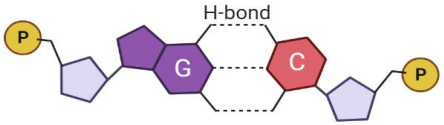
Pyrimidines:

T Thymine

C Cytosine

Appariement des bases complémentaires





Two key biochemical processes

Synthesis:

- **Chemical fabrication** of a DNA molecule
- Composed of nucleotides
- Takes as input **any quaternary sequence** composed of As, Ts, Cs and Gs to synthesize in a molecule

Sequencing:

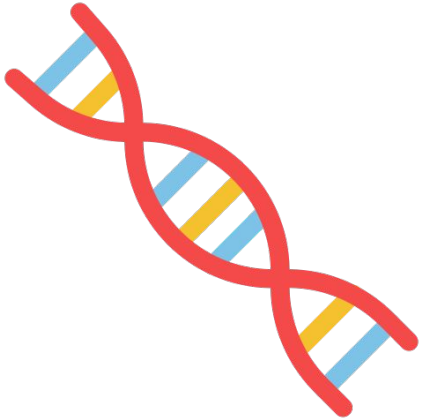
- **Chemical analysis** of a DNA molecule
- **Extracts** the quaternary **sequence** of As, Ts, Cs and Gs composing the molecule
- Takes as input any DNA molecule
- Outputs the associated quaternary representation

Two key biochemical processes

ATCGTGACTGTG

Synthesis →

← **Sequencing**

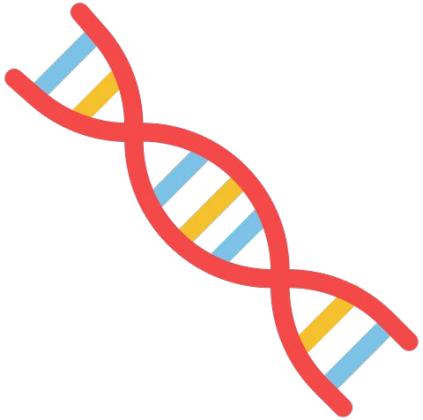


Through the biologist’s lens

ATCGTGACTGTG

Synthesis →

← **Sequencing**



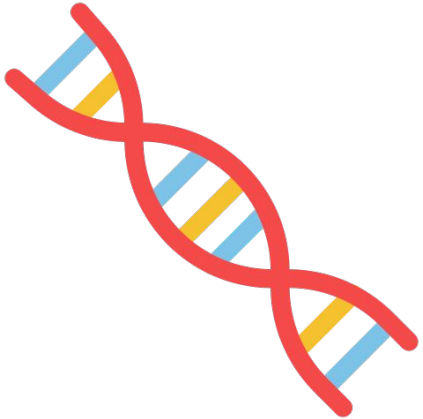
Genome analysis, genetic modification, etc…

Through the computer scientist’s lens

ATCGTGACTGTG

Synthesis →

← **Sequencing**



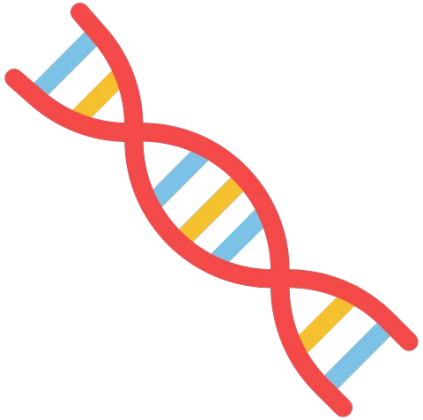
Through the computer scientist's lens

ATCGTGACTGTG

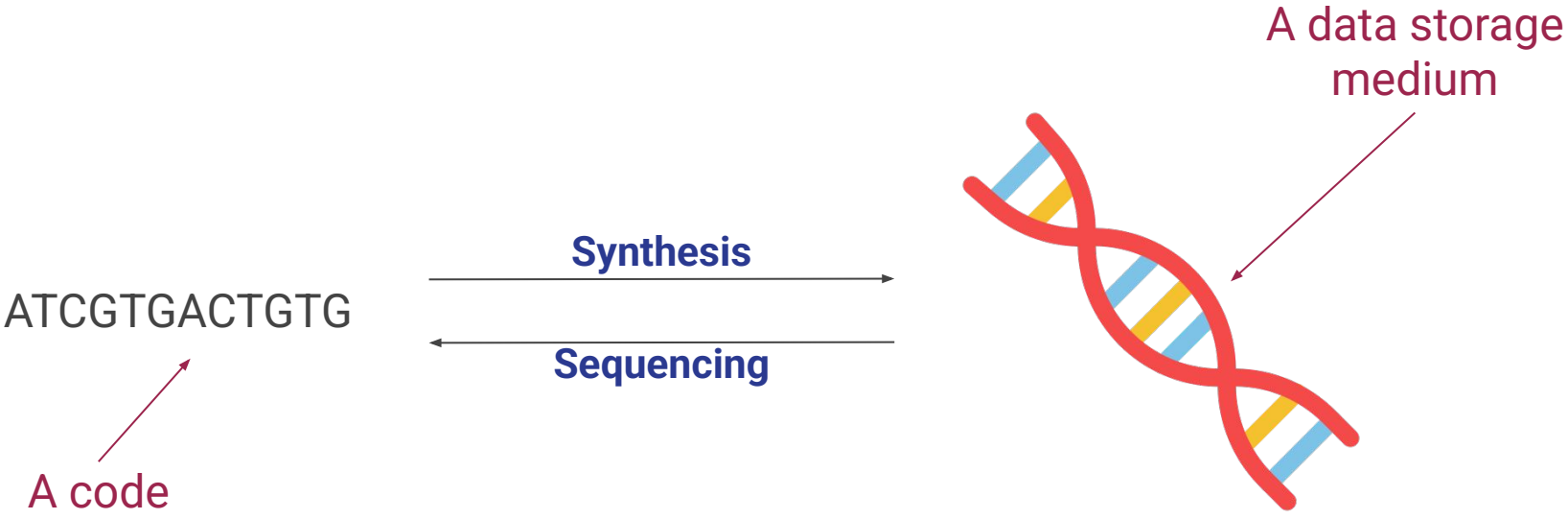
A code

Synthesis

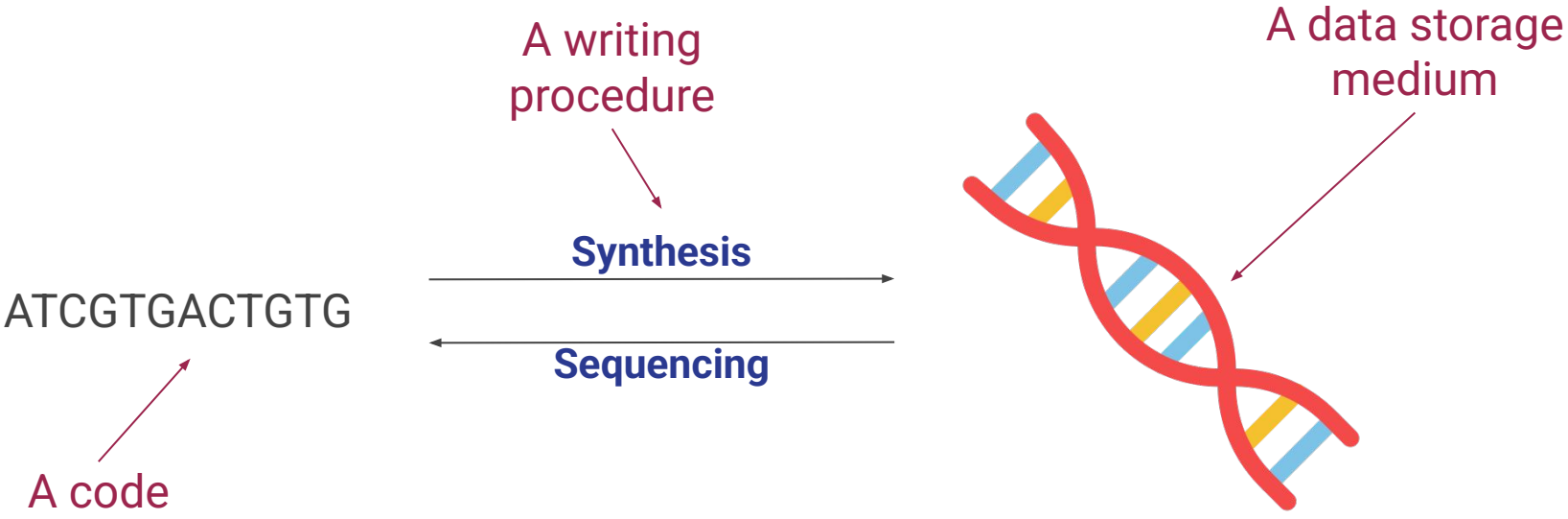
Sequencing



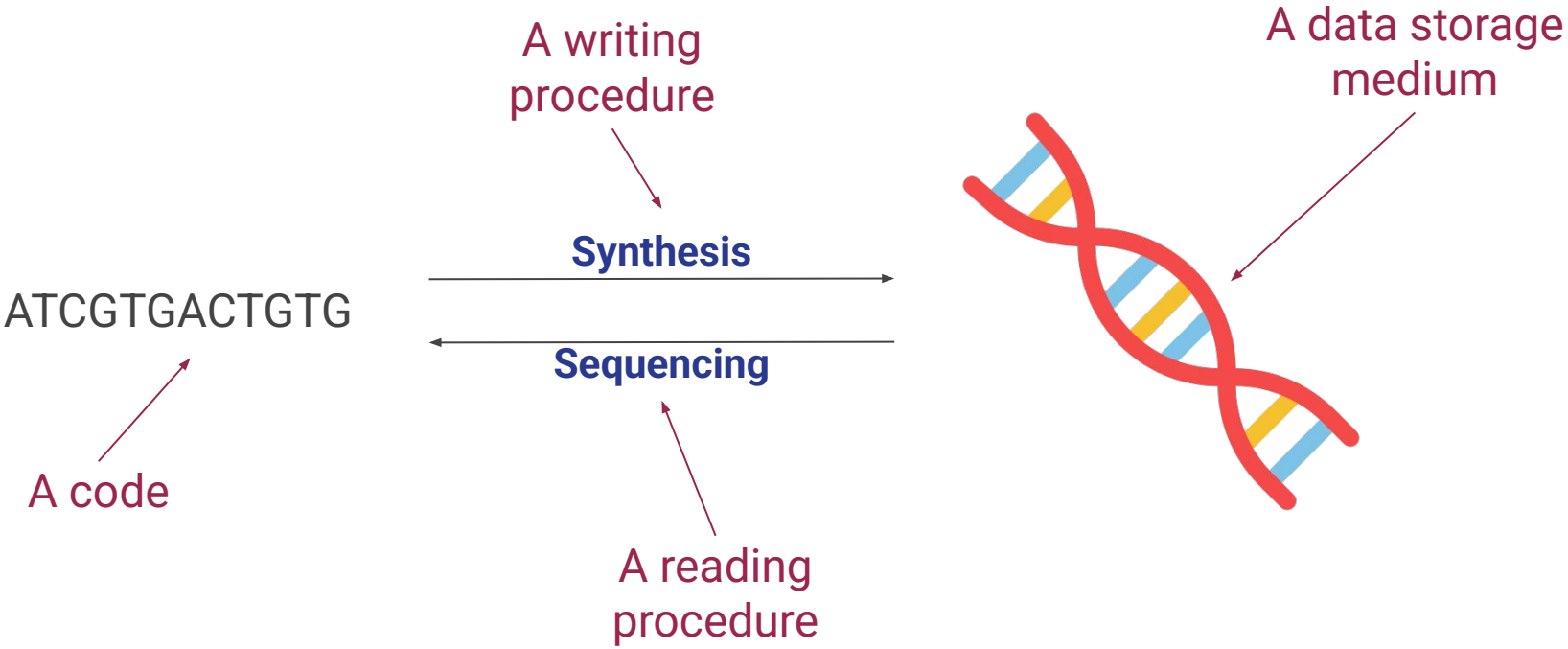
Through the computer scientist's lens



Through the computer scientist's lens



Through the computer scientist's lens



DNA Data storage - General Principle

- Writing data:
  - Encode a file into a set of quaternary sequences (oligos)
  - Synthesize the DNA molecules represented by these sequences
  - Store the molecules
- Reading data:
  - Retrieve the quaternary sequences from the stored DNA molecules (sequencing)
  - Decode the file from the set of sequenced oligos

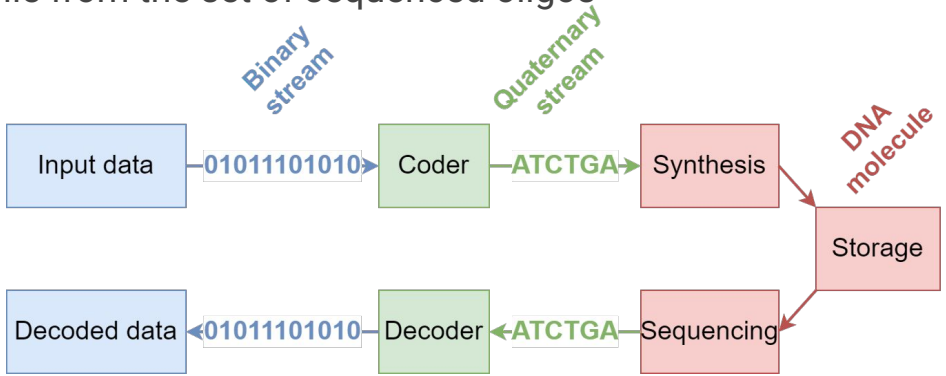


Fig: General workflow of DNA data storage

DNA Data storage - Biochemical operations

- Synthesis : Create a set of DNA molecules
- PCR : Copy molecules of DNA
- Sequencing : Read the structure of the DNA molecules

Operation	Cost	Speed	Errors
Synthesis	Very High	Very Slow	Few
PCR	Low	Fast	Low
Sequencing	High	Slow	Many

DNA Coding - Principle

Binary File (.png)

```
00000000: 10001001 01010000 01001110 01000111 00001101 00001010  .PNG..
00000006: 00011010 00001010 00000000 00000000 00000000 00001101  ......
0000000c: 01001001 01001000 01000100 01010010 00000000 00000000  IHDR..
00000012: 00000011 00000000 00000000 00000000 00000010 00000000  ......
00000018: 00001000 00000010 00000000 00000000 00000000 10111010  ......
0000001e: 10010100 10011100 01101101 00000000 00000000 00000000  ..m...
00000024: 00000100 01100111 01000001 01001101 01000001 00000000  .gAMA.
0000002a: 00000000 10110001 10001111 00001011 11111100 01100001  .....a
00000030: 00000101 00000000 00000000 00000000 00000001 01110011  .....s
00000036: 01010010 01000111 01000010 00000000 10101110 11001110  RGB...
0000003c: 00011100 11101001 00000000 00000000 00000000 00010100  ......
00000042: 01110100 01000101 01011000 01110100 01110011 01101111  tEXtso
00000048: 01110101 01110010 01100011 01100101 00000000 01001011  urce.K
0000004e: 01101111 01100100 01100001 01101011 00100000 01010000  odak P
00000054: 01000011 01000100 00110000 00111001 00111001 00110010  CD0992
0000005a: 01000100 10011101 10000000 11010110 00000000 00001000  D.....
00000060: 00000000 00000000 01001001 01000100 01000001 01010100  ..IDAT
00000066: 01111000 11011010 11101100 11011101 01010111 10001100  x...W.
0000006c: 01011111 11011001 10011101 00100111 01110110 11000011  _..'v.
00000072: 00001111 10110110 01100001 11001111 10101100 01110110  ..a..v
00000078: 01000110 00110011 00100011 01101001 10010100 00111010  F3#i.:
0000007e: 01000111 01110110 00110111 01110011 11001110 00111001  Gv7s.9
00000084: 10010011 11000101 10101010 01100010 11100101 10011100  ...b..
0000008a: 01110011 11001110 00111001 11100111 01100010 01010101  s.9.bU
00000090: 10010001 01000101 10110010 00011000 10001010 00111001  .E...9
00000096: 00110110 10011011 11011101 11101100 01101110 10110101  6...n.
0000009c: 01011010 11010010 10001100 01110110 01110110 01110110  Z..vvv
000000a2: 00001100 00101111 10110000 00110000 00001100 11111000  ./.0..
000000a8: 11000001 10000000 10000001 00000101 01101100 11101100  ....l.
000000ae: 10001011 11100001 00000111 00000011 00000110 10001100  ......
000000b4: 01111101 01011110 01111111 11001110 00111101 00101101  }^..=-
000000ba: 10001110 01110110 01110111 00100100 01101011 01100111  .vw$kg
000000c0: 01110111 11101100 00000111 10001011 00111000 10111000  w...8.
000000c6: 10111000 01110101 11111111 11100111 10011110 11110000  .u....
000000cc: 00111011 10111111 11110000 11111101 11111101 11001110  ;.....
000000d2: 11101111 01011100 11111110 01100111 00100101 10111111  .\.g%.
000000d8: 11111111 11110111 11111011 01111111 10111111 11111111  ......
000000de: 11110111 11111011 01111111 10111111 11111111 11110111  ......
000000e4: 11111011 01111111 10111111 11111111 11110111 11111011  ......
000000ea: 01111111 11111111 00111111 11111011 11110111 10011111  ..?...
000000f0: 11111101 11111010 00011111 10100101 11000101 00100101  .....%
000000f6: 10100101 00100101 10000101 10111111 00101111 11111111  .%../.
000000fc: 01001001 01001011 11111110 10111111 01011101 11111110  IK..].
00000102: 01010011 10110101 11110110 11111111 11010101 00101100  S...,
00000108: 01111110 11001011 11111101 00111111 11011100 11001000  ~..?..
```

ENCODING

DECODING

DNA File (.fasta)

```
>Oligo0
ATGTGCAGAATTCCGTTGAATTCCCACAAAGAGAATTCAAGAGAAGAGCGTCTCGACTATCGAGATCACATGTGCTGTGTGTAGTACTATACTGTC
>Oligo1
AATAATAAATAATAAATAATAACCGGATACTGGCTAATACCAAATAATAAATAATAAATAATAAATAATAAATAATAAATAATAAATAATAAATAA
>Oligo2
AATATATCGTAGTGTAGCTGCACGTACGACTCACGACAGCTGCACGTGCGCGAGTCGTGCATCTGATGCAGACAGCAGCAGCGATCGCGCTAGTCG
>Oligo3
AATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAACTATTCATAATATAAGCAAATAATTCACAATAATAATAAATAATAATAAA
>Oligo4
AACAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAATAAATAATAATAAATAGGATCAAATAATTCACAAT
>Oligo5
AAAATACTTCATAATAATATTGAATAATAATAAATAATAATAAATAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATA
>Oligo6
AAATAATAATAAATAATAATAAATAATAATAAATATTATTGAATAATAATGAATAATAATAAATAATAATAATAAATAATAATAATAAATAA
>Oligo7
AATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAATAAATAATAATAAATAAGATCGAATAATAATGAATAATAATAAATAAT
>Oligo8
ATAATAATCAATAATAATAAATAATAATAAATAATAATAAATAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAAACA
>Oligo9
AAATAATAAATAATAATAAATAACACGTAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAA
>Oligo10
AATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATCCACAATAATAATAAATAATAATAAATAATAATAAATAATAAT
>Oligo11
ATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATATCGAATAATAATA
>Oligo12
AAATAAATAATTCTCAATCCGCTTAAATATACGAGAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAATAA
>Oligo13
AATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAGATAATAATAATAAATAATAATAAATAATAATAAA
>Oligo14
ATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATATTAAAT
>Oligo15
AAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATA
>Oligo16
AAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAA
>Oligo17
AATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAAAT
>Oligo18
ATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATA
>Oligo19
AAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAA
>Oligo20
AATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAAT
>Oligo21
ATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATAAATAATAATA
```

DNA Coding - Constraints

The following operations can increase the errors in biochemical processes (synthesis, PCR, sequencing):

- Homopolymers (repetition of the same nucleotide)
- Motifs (patterns)
- Unbalanced GC content
- Large sequences (>300 nts)
- Undesired sequences

ADAPT AND CONSTRAIN THE CODING SYSTEM

DNA Coding - Why do we need to compress the data?

Errors can occur in the biochemical processes (synthesis, sequencing, PCR):

- High sequencing cost
- Very high synthesis cost
- Low synthesis and sequencing speed (writing and reading cost)

REDUCE THE SIZE OF THE ENCODED DNA FILES

Examples of DNA coders

Summary of the presentation - DNA coding methods

- DNA Adapted Channel coders:
  - SFC4
  - C3
  - CMOSS
- Image coders adapted to DNA:
  - JPEG DNA VM
  - JPEG DNA SFC4
  - JPEG 2000 DNA
  - HiDNA

Examples of DNA Channel Coders

SFC4

DNA Channel Coders - Entropy coder

- **Variable length** coder
- Adapted to **i.i.d. sources**
- Length of the codeword depends on the **probability of appearance** of the associated symbol
- Better performance for compression

Symbol	Probability	Code
a	0.40	0
b	0.05	1010
c	0.18	110
d	0.07	1011
e	0.20	111
f	0.10	100

Fig: Example of a binary Huffman code

DNA Channel Coders - The Goldman Coder

- Ternary Huffman Entropy Coder
- No repeated nucleotides (no homopolymers)

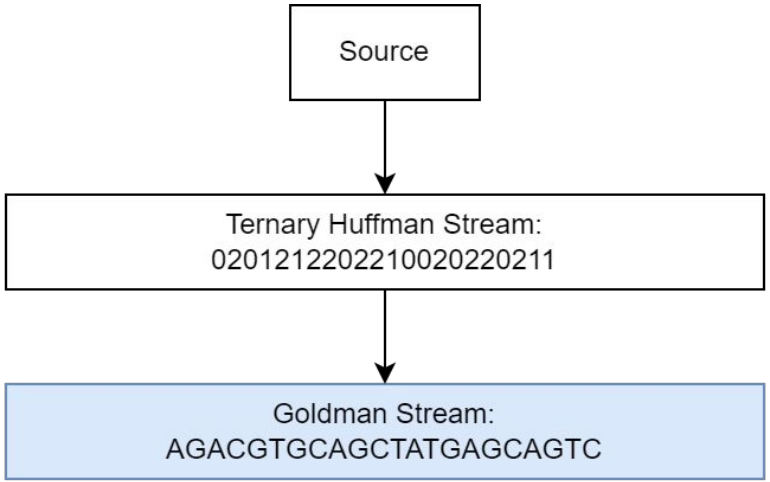


Fig: The data is first encoded into a ternary code

Previous nucleotide	0	1	2
A	T	C	G
T	A	C	G
C	A	T	G
G	A	T	C

Fig: Ternary to DNA translation table

N. Goldman, P. Bertone, and S. Chen, “Towards practical, high-capacity, low-maintenance information storage in synthesized dna,” *Nature*, 2013.

Performance metrics - DNA entropy coders

Expected length:

$$\mathcal{L}(C) = \sum_{x \in \Omega} p(X = x) l(C_x)$$

Ω - The set of symbols in the source
C_x - The codeword associated to the symbol
$l(C_x)$ - The length of a codeword

The **average size of a code** output by the entropy coder for a source

In the case of DNA coding, it is expressed in **nucleotides per symbol**

SFC4 - Characterization of the source and principle

- A source *S* composed of elements of an alphabet *A* has to be encoded
- The frequency of appearance of each element of *A* in *S* is computed
- A tree *T* is initialized as a single node (the root) where all the symbols are allocated
- The Shannon Fano tree is built by subdividing the leaves of the tree

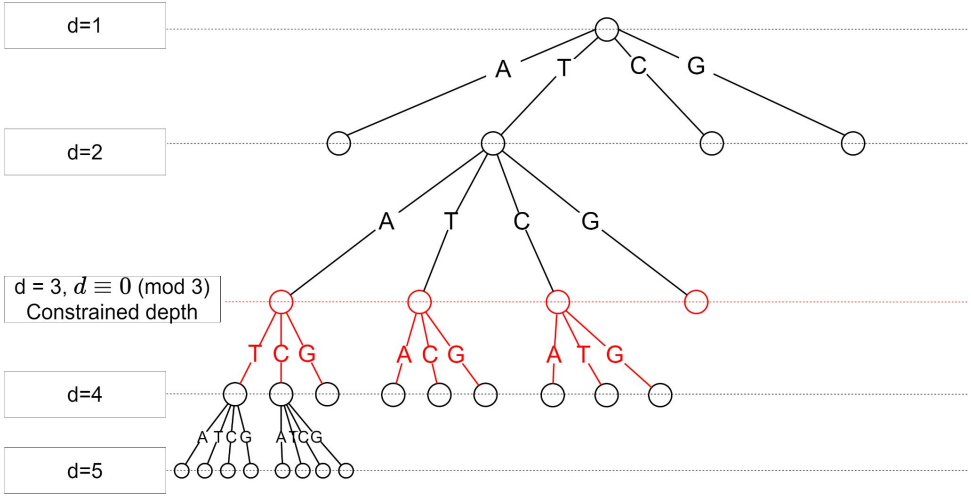


Fig.: Example of an SFC4 coding tree, constrained at depths multiples of 3

SFC4 - Coding tree

Fig.: Example of an SFC4 coding tree, constrained at depths multiples of 3

SFC4 - Coding tree

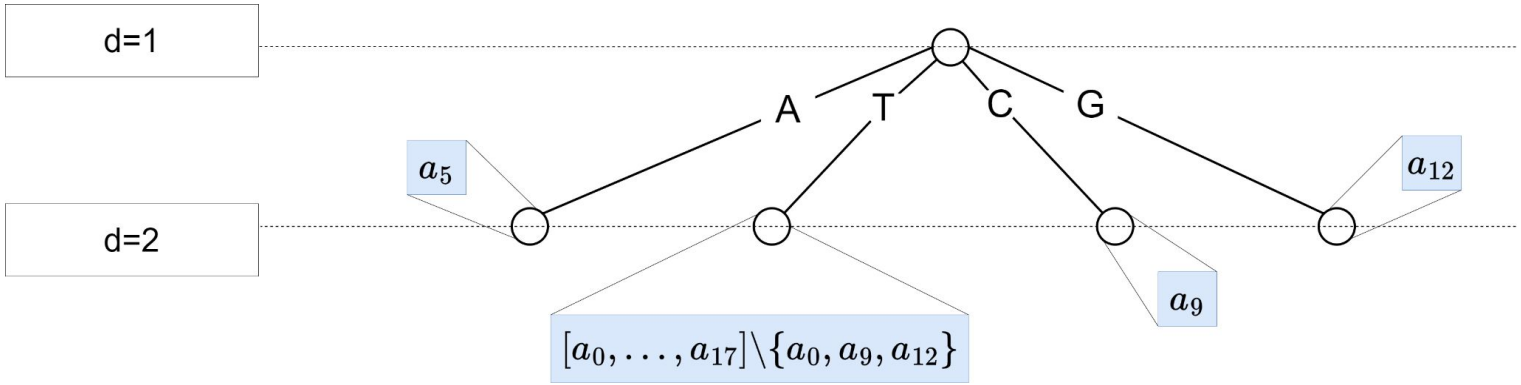


Fig.: Example of an SFC4 coding tree, constrained at depths multiples of 3

SFC4 - Coding tree

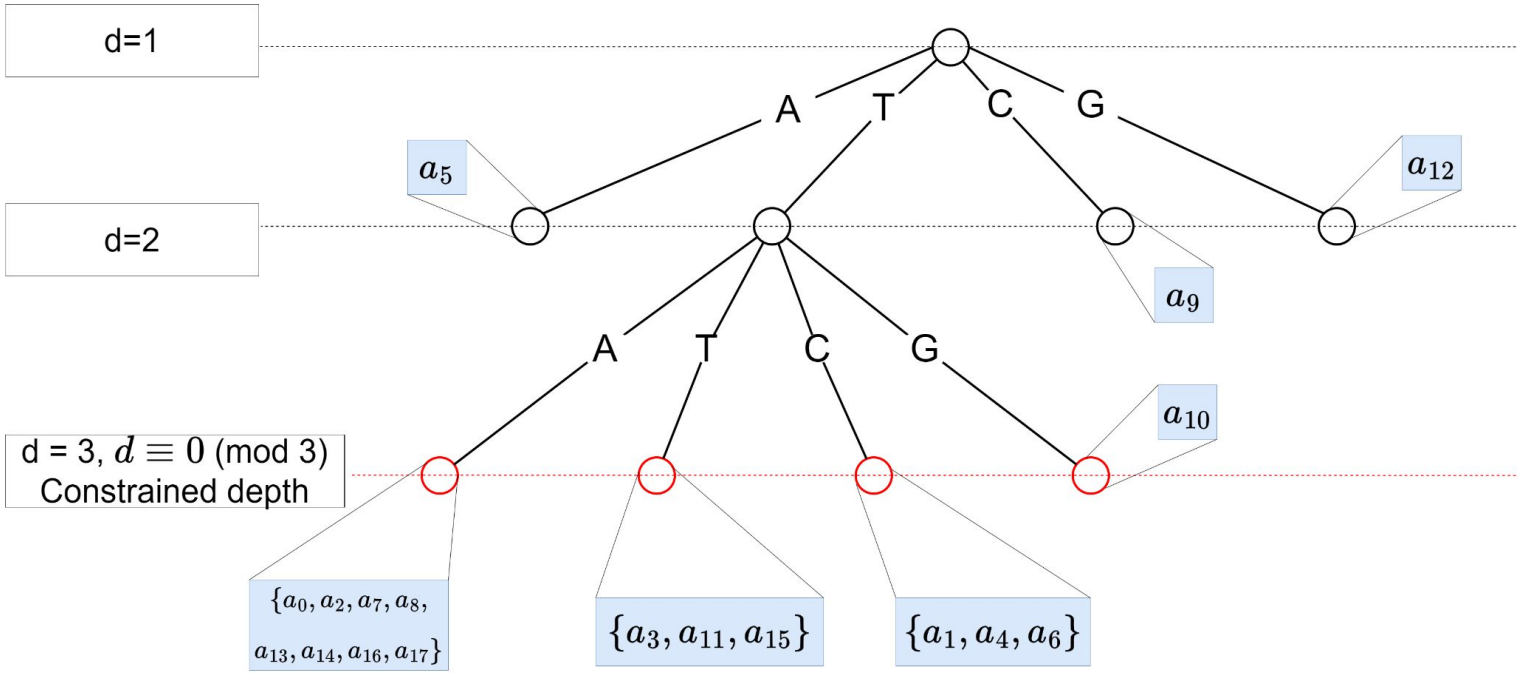
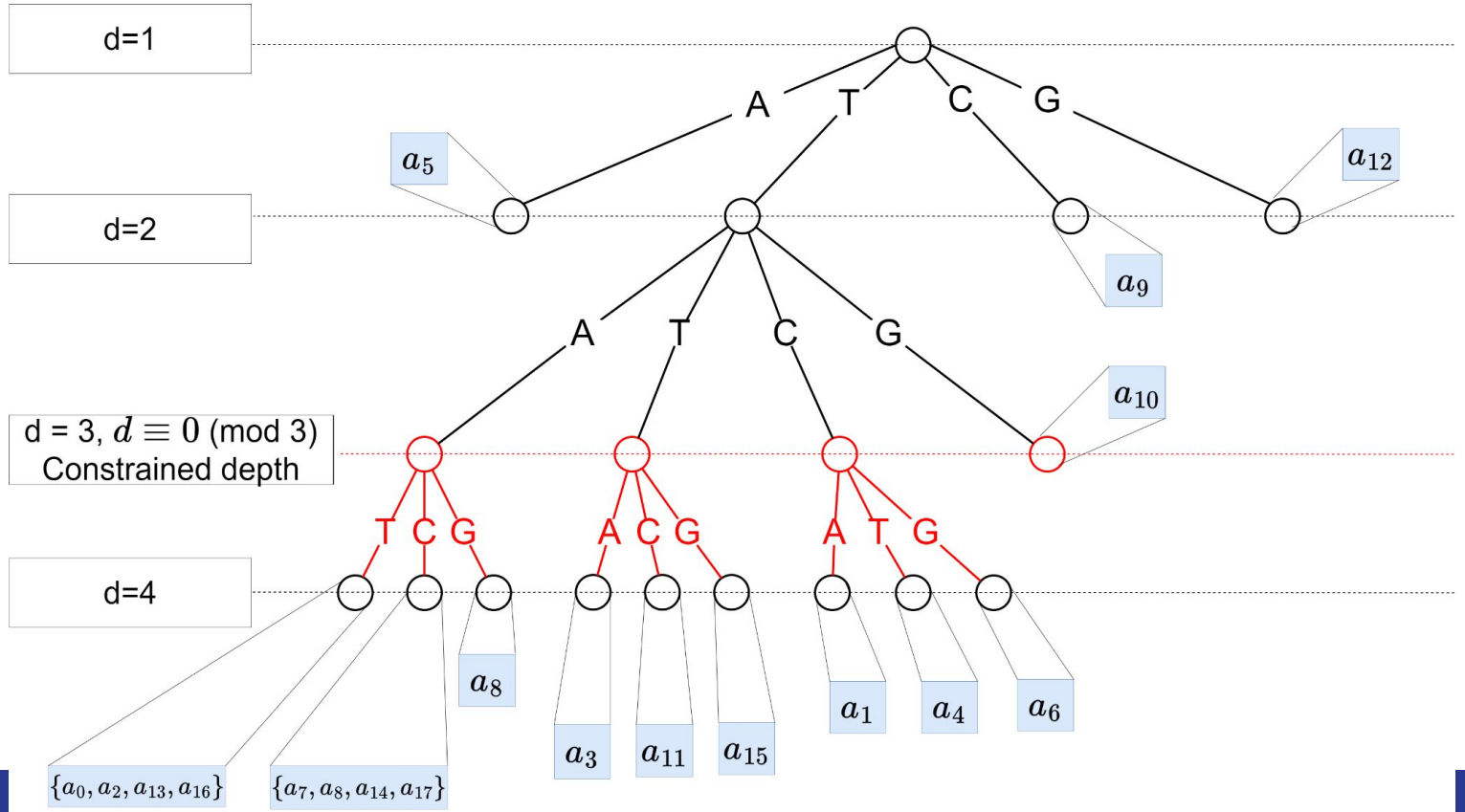


Fig.: Example of an SFC4 coding tree, constrained at depths multiples of 3

SFC4 - Coding tree



SFC4 - Coding tree

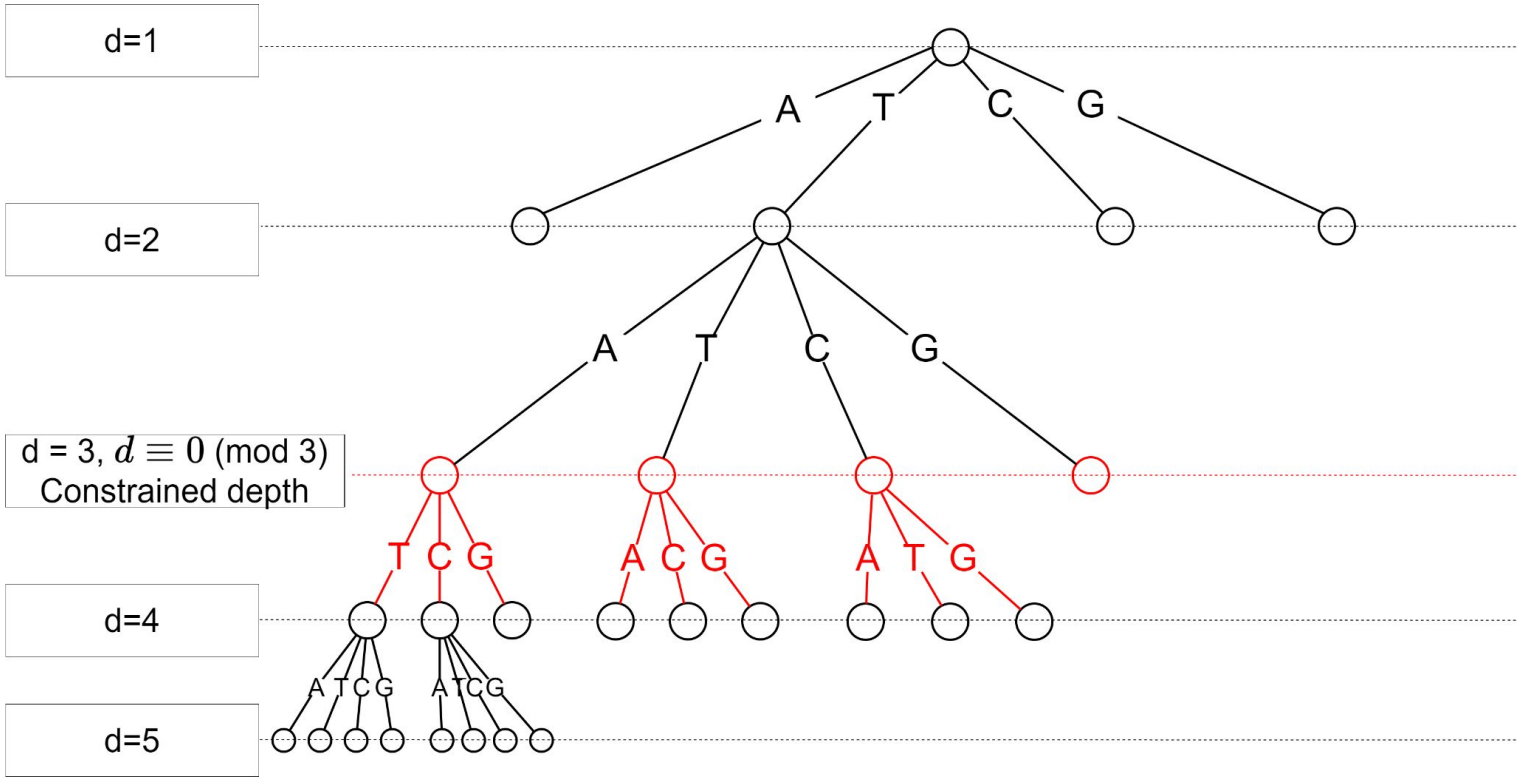


Fig.: Example of an SFC4 coding tree, constrained at depths multiples of 3

SFC4 - Results on I.I.D. Gaussian Sources

Expected length:

- Average results on a series of **100 i.i.d. gaussian sources**
- Each source contained **10000 samples**
- Quantized to the **nearest integer**

$H_4(X)$	$\boldsymbol{L(C_{SFC4})}$	$H_3(X)$	$L(C_{HG})$
3.48	**3.81**	4.39	4.45

Fig: Ternary to DNA translation table

Kodak dataset, weblink: https://r0k.us/graphics/kodak/

Examples of DNA Channel Coders

C3DNA

C3 - Principle of the MQ coder’s arithmetic coder

- Source: [1,0,1,1]
- Initialize an interval to [0, 1]

Received bits	1011	
Event	**Actions**	**Intervals**
None	Start	I = [0, 1)

C3 - Principle of the MQ coder’s arithmetic coder

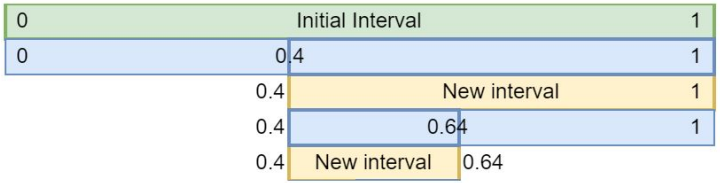
- Source: [1,0,1,1]
- Initialize an interval to [0, 1]
- First received bit: 1
  - Subdivide the interval into two smaller intervals of size relative to the probability of appearance of a zero or a one.
  - Select the subdivision associated to the received bit (1)

Received bits	1011	
Event	**Actions**	**Intervals**
None	Start	I = [0, 1)
Received 1	Subdivison	[0, 0.4), [0.4, 1)
	Selection	I = [0,4, 1)

C3 - Principle of the MQ coder’s arithmetic coder

- Source: [1,0,1,1]
- Initialize an interval to [0, 1]
- First received bit: 1
- Second received bit: 0
  - Subdivide the interval into two smaller intervals of size relative to the probability of appearance of a zero or a one.
  - Select the subdivision associated to the received bit (0)

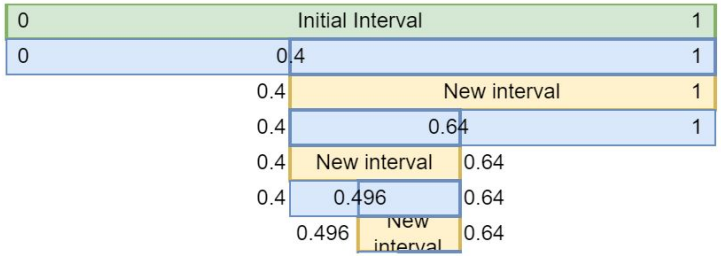
Received bits	1011	
Event	**Actions**	**Intervals**
None	Start	I = [0, 1)
Received 1	Subdivison	[0, 0.4), [0.4, 1)
	Selection	I = [0,4, 1)
Received 0	Subdivison	[0,4, 0,6*0,4+0.4 =0.64),[0.64, 1)
	Selection	I = [0.4, 0.64)



C3 - Principle of the MQ coder’s arithmetic coder

- Source: [1,0,1,1]
- Initialize an interval to [0, 1]
- First received bit: 1
- Second received bit: 0
- Third received bit: 1
  - Subdivide the interval into two smaller intervals of size relative to the probability of appearance of a zero or a one.
  - Select the subdivision associated to the received bit (1)

Received bits	1011	
Event	**Actions**	**Intervals**
None	Start	I = [0, 1)
Received 1	Subdivison	[0, 0.4), [0.4, 1)
	Selection	I = [0,4, 1)
Received 0	Subdivison	[0,4, 0,6*0,4+0.4 =0.64),[0.64, 1)
	Selection	I = [0.4, 0.64)
Received 1	Subdivison	[0.4, 0.496), [0.496, 0.64)
	Selection	I = [0.496, 0.64)

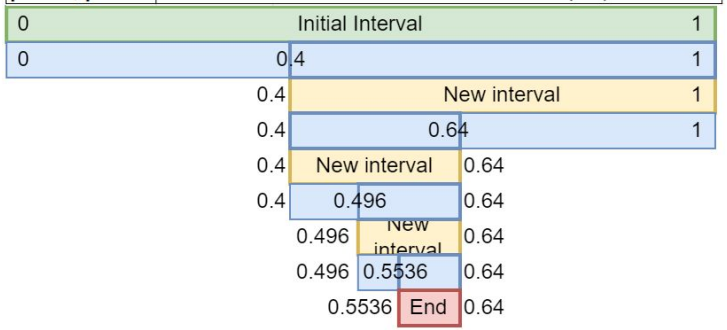


C3 - Principle of the MQ coder’s arithmetic coder

- Source: [1,0,1,1]
- Initialize an interval to [0, 1]
- First received bit: 1
- Second received bit: 0
- Third received bit: 1
- Final received bit: 1

The **final interval** represents the whole input source [1, 0, 1, 1]

Received bits	1011	
Event	**Actions**	**Intervals**
None	Start	I = [0, 1)
Received 1	Subdivison	[0, 0.4), [0.4, 1)
	Selection	I = [0,4, 1)
Received 0	Subdivison	[0,4, 0,6*0,4+0.4 =0.64),[0.64, 1)
	Selection	I = [0.4, 0.64)
Received 1	Subdivison	[0.4, 0.496), [0.496, 0.64)
	Selection	I = [0.496, 0.64)
Received 1	Subdivison	[0.496, 0.5536), [0.5536, 0.64)
	Selection	I = [0.5536, 0.64)
Output	x=0.625	
p0=0.4, p1=0.6	x is the element of I with the shortest binary representaion	



C3 - MQ coder-based arithmetic coder for DNA

- Transcode the registries of the binary MQ-coder
- Constrained fixed length code of length 3 (48 elements)
- Transcode integers into a base 48 representation
- Code the coefficients of this representation

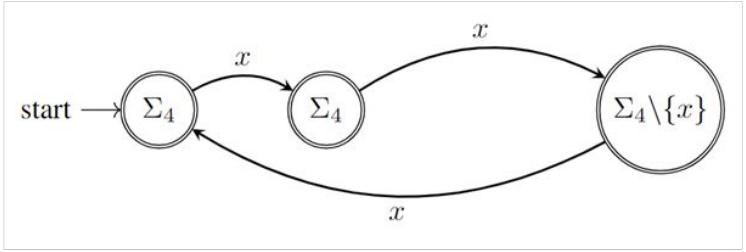


Fig. Automata generating the coding dictionary: **homopolymers are avoided**

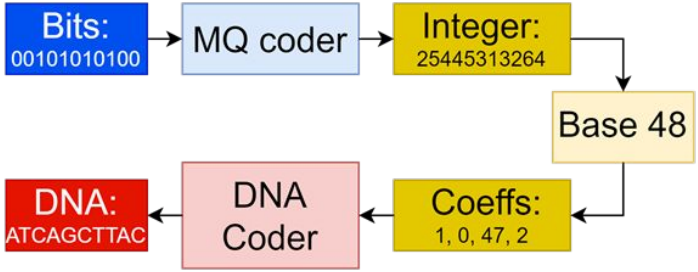


Fig. Adaptation of the MQ-coder to DNA

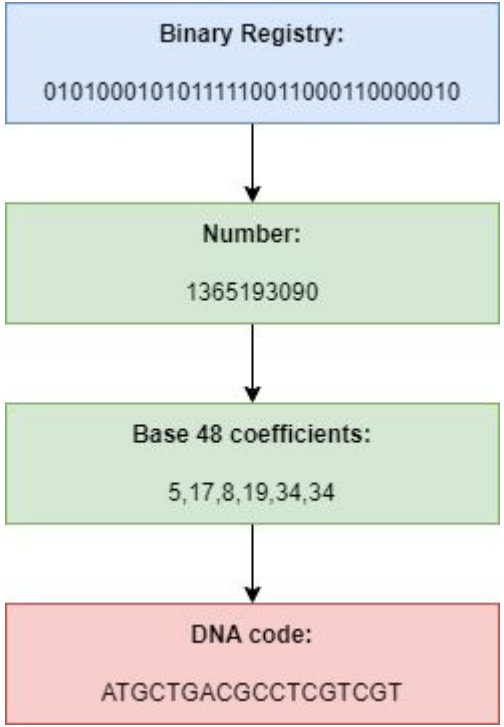
$$C_3 = \{AAT, AAC, AAG, ATA, ATC, ATG, ACA, \ ..., \ GCT, GCG, GGA, GGT, GGC\}$$

X. Pic, M. Dimopoulou, E. Gil San Antonio and M. Antonini, “MQ-Coder inspired Arithmetic Coder for Synthetic DNA data storage”, ICIP 2023

C3 - Encoding a registry

- The registry is first converted into a number
- The number is represensented into base 48
- The coefficients in base 48 are encoded into DNA, one by one, with the C3 coder:

$$C_3 = \{AAT, AAC, AAG, ATA, ATC, ATG, ACA, \ ..., \ GCT, GCG, GGA, GGT, GGC\}$$

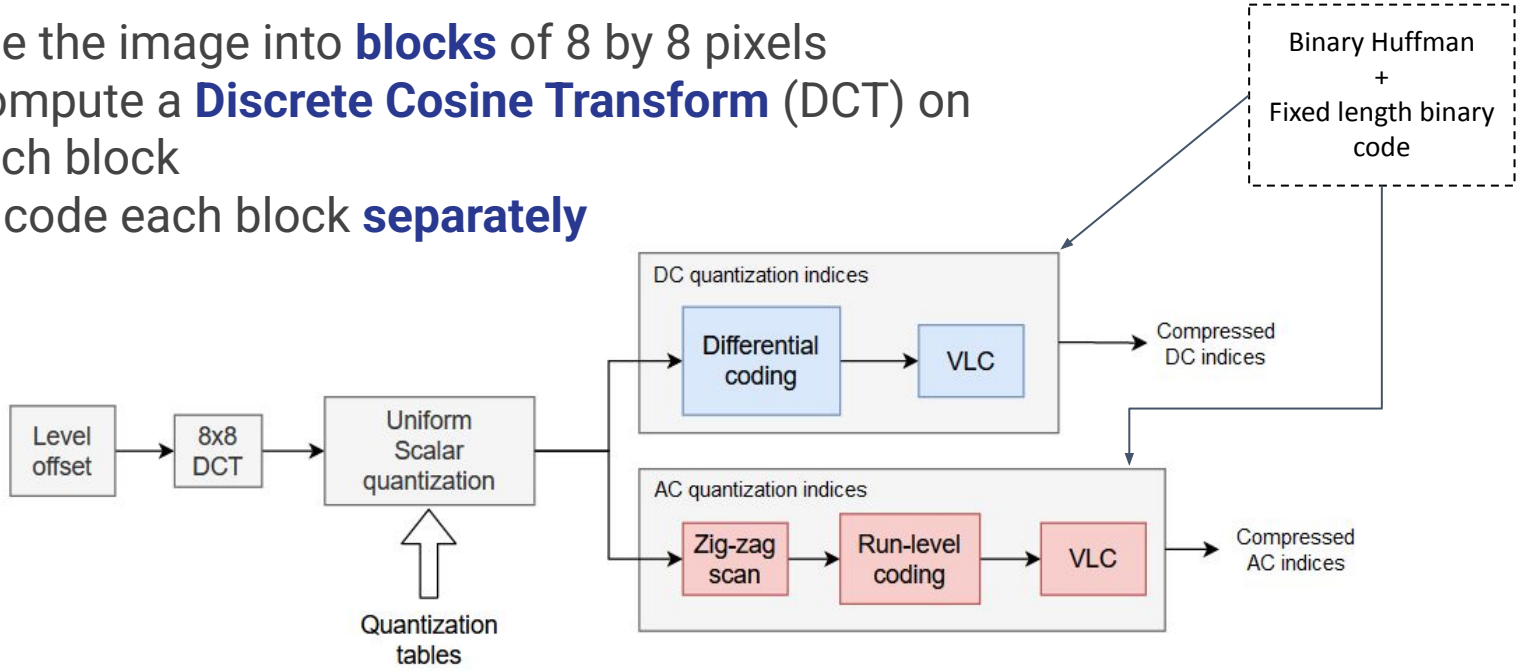


Examples of DNA Image Coders

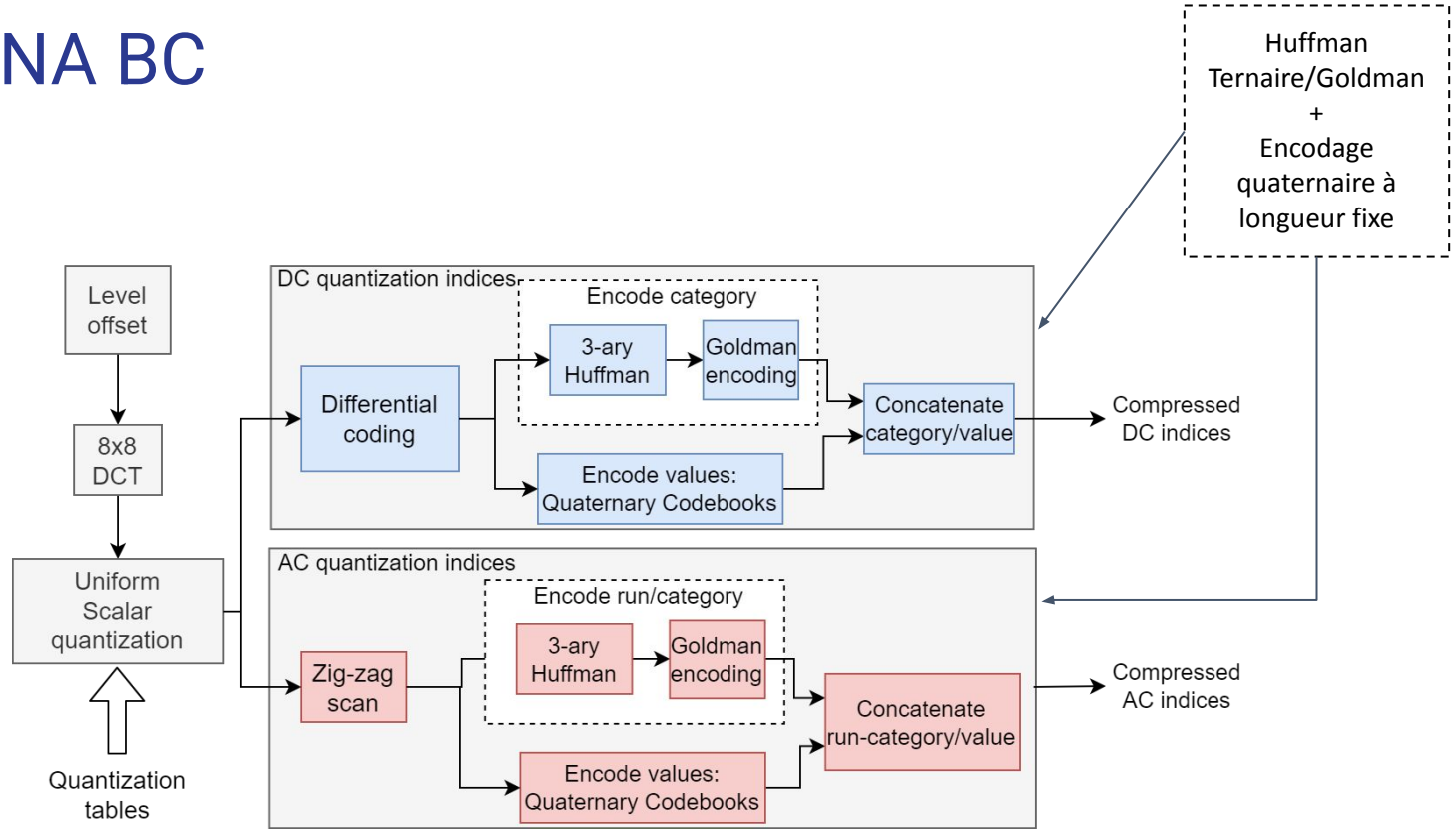
JPEG DNA SFC4 and JPEG2000 DNA

Image coders - JPEG

- Tile the image into **blocks** of 8 by 8 pixels
- Compute a **Discrete Cosine Transform** (DCT) on each block
- Encode each block **separately**

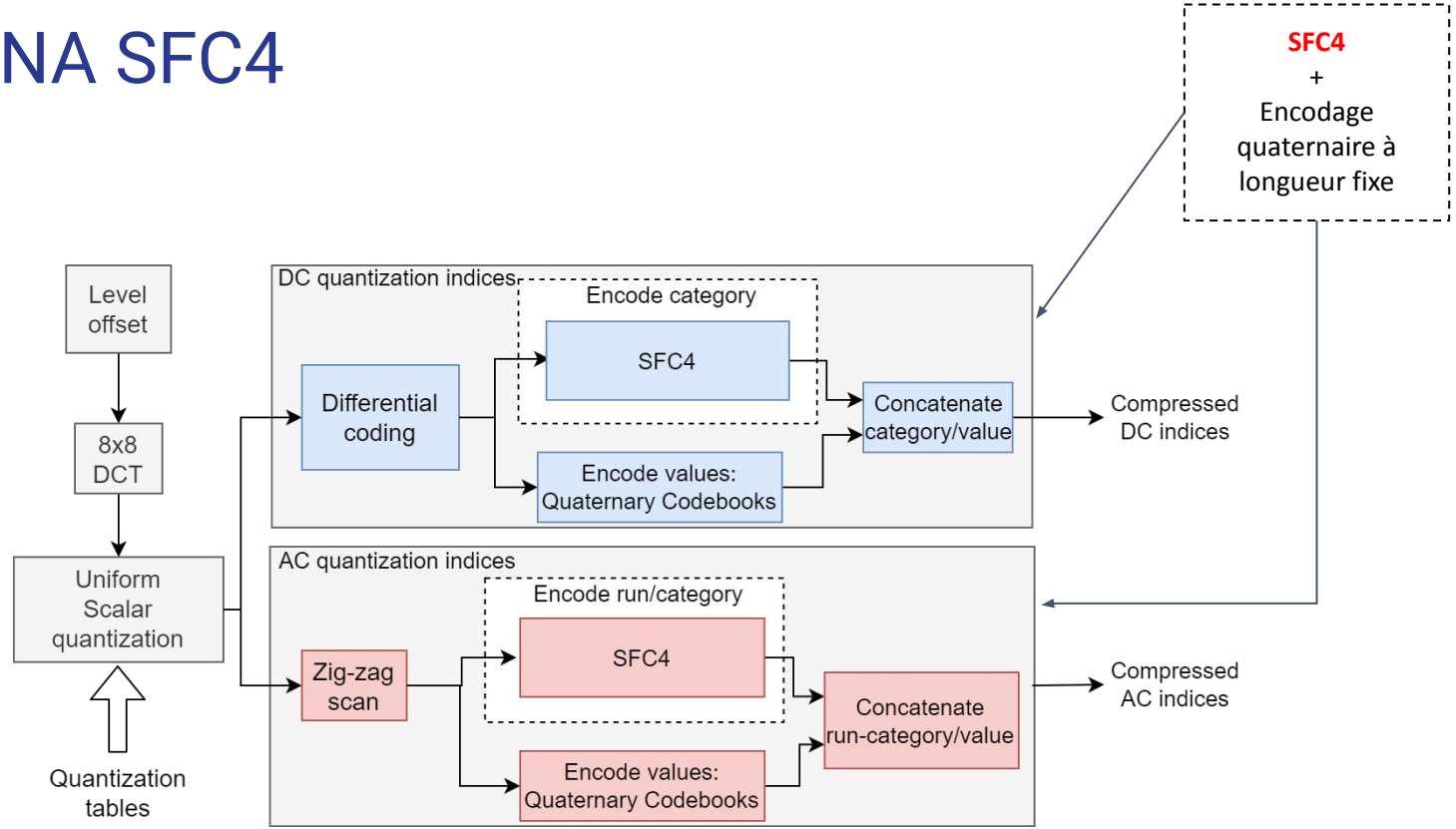


JPEG DNA BC



1. M. Dimopoulou, E. Gil San Antonio, M. Antonini, “A JPEG-based image coding solution for data storage on DNA”, *EUSIPCO*, 2021.
2. X. Pic, E. Gil San Antonio, M. Dimopoulou, M. Antonini, M93103 - JpegDNA Python Library

JPEG DNA SFC4



X. Pic and M. Antonini, “A constrained shannon-fano entropy coder for image storage in synthetic DNA”, EUSIPCO 2022

Metrics for DNA coding

Performance metrics - DNA Channel coders

Compression rate:

$$cr_{btsnt} = \frac{\#bits}{\#nucs}$$

The **number of bits** in the source that can be encoded in a **single nucleotide**

Performance metrics - DNA image coders

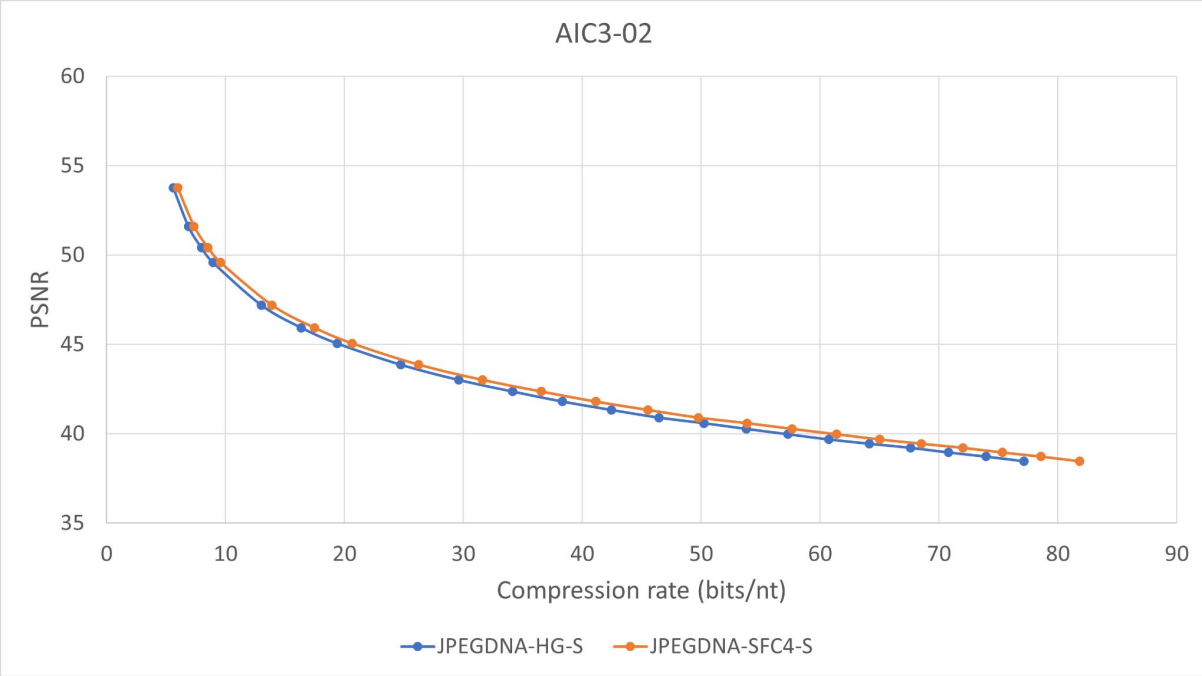
Nucleotide rate:

$$r_{ntpp} = \frac{\#nucs}{H \times W}$$

The **average number of nucleotides** to encode a **pixel**

JPEG DNA SFC4 - Results

- Novel codec improves the rate by 5% on average
- Tested on different datasets (kodak and JPEG AIC 03)



1. Kodak dataset, weblink: https://r0k.us/graphics/kodak/
2. M. Testolina, V. Hosu, M. Jenadeleh, D. Lazzarotto, D. Saupe, T. Ebrahimi, “JPEG AIC-3 Dataset: Towards Defining the High Quality to Nearly Visually Lossless Quality Range”, *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*, Ghent, Belgium, 2023

JPEG2000 DNA - Principle

- The **binary MQ coder** is associated to the new **C3 coder**
- The **overheads** are transcoded into DNA using a **fixed length** DNA adapted coder
- The decoder consists in reverting all previous operations
- **Closed loop modification** of JPEG2000

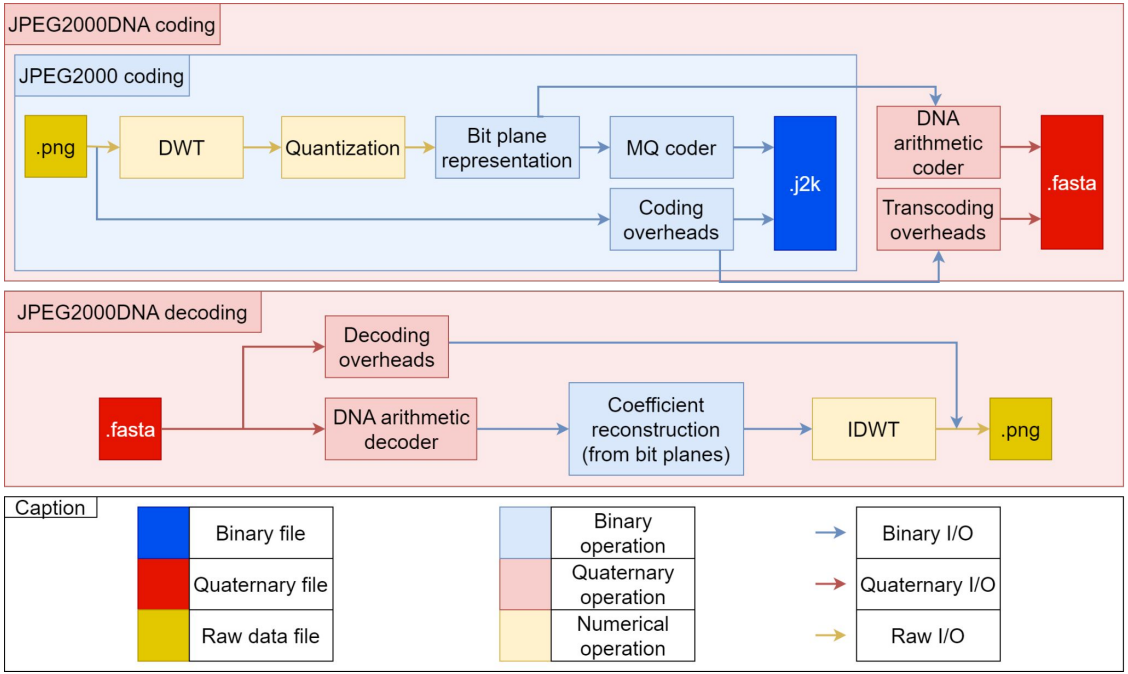
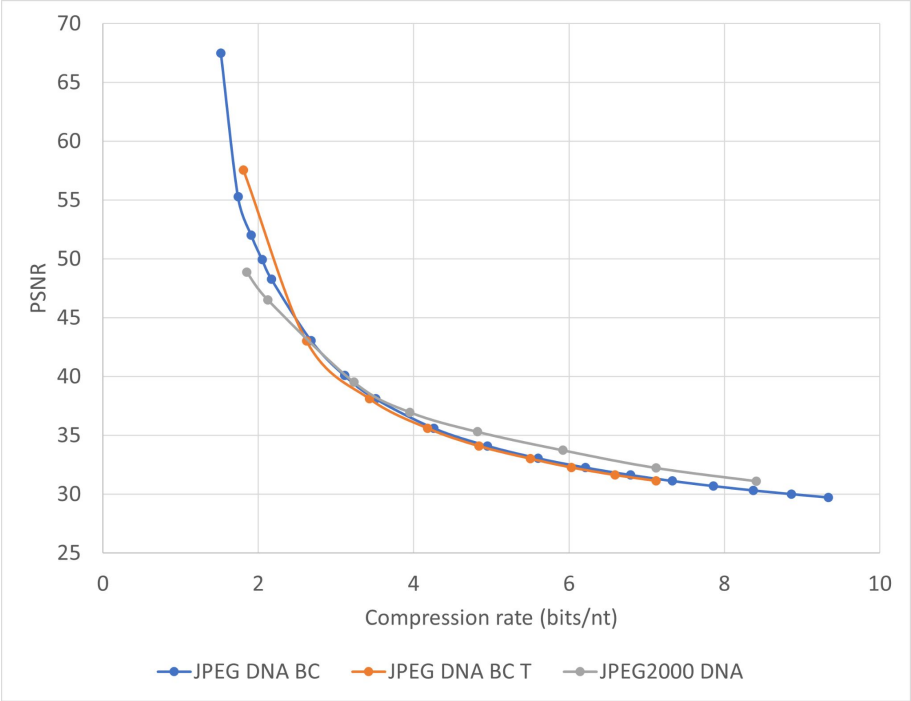


Fig: Adaptation to DNA of the JPEG2000 codec

X. Pic, M. Dimopoulou, E.G. San Antonio and M. Antonini, “MQ-Coder Inspired Arithmetic Coder for Synthetic DNA Data Storage”, ICIP 2023

JPEG2000 DNA - Results

- Tested across the kodak dataset
- On the right, an example of encoding one image of the kodak dataset (kodim7) using the JPEG DNA 2000 codec
- Performance mostly similar with the original JPEG DNA BC software



Kodak dataset, weblink: https://r0k.us/graphics/kodak/

Examples of DNA Channel Coders

JPEG DNA VM

DNA Fountain Codes - Raptor Code

- **Cut** the memory into bytes
- Randomly **Select and sum** a fixed number of these bytes
- Seed and **encode** into DNA
- **Repeat** the bytes **selection**, **sum** and **encoding** until all bytes are encoded

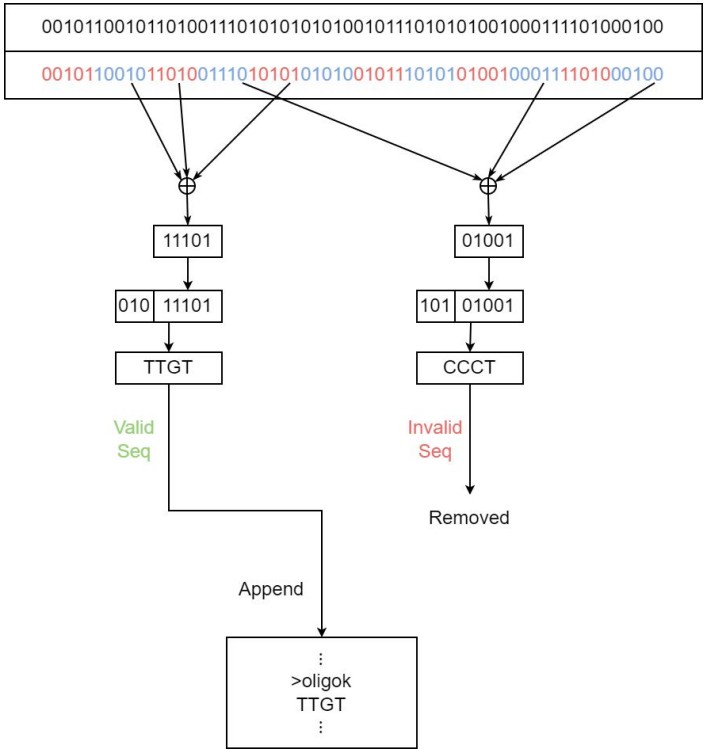


Fig: A DNA adapted fountain code

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

Fountain code adapted to DNA

- Fountain code
- Encode the XORed chunks into DNA with the following table:

Input bits	DNA code
00	A
01	T
10	C
11	G

Selection of the oligos

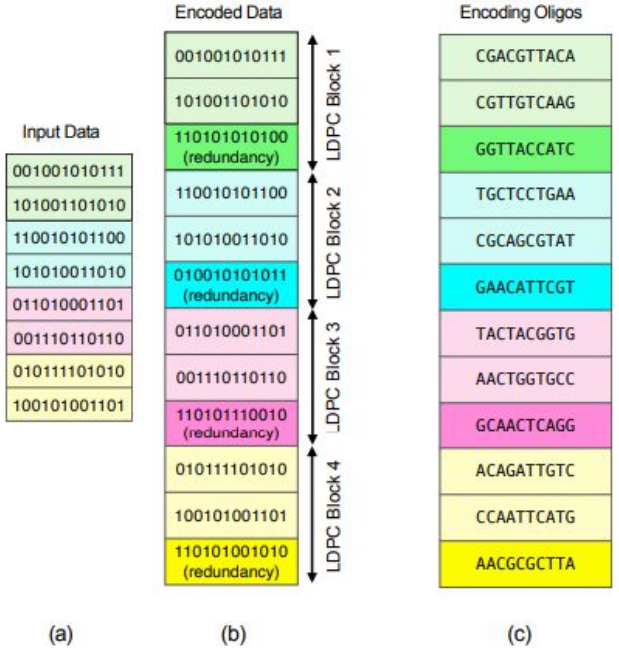
- Code is **not constrained**
- Some generated oligos will not respect the constraints
- **Remove all the oligos** that don’t respect the constraints
- Check that there is still enough **redundancy**, otherwise:
  - Start the whole process all over
  - Change the seed for the random selection of chunks

Examples of DNA Channel Coders

CMOSS

CMOSS - Encoding in columns

- Data is first organized in columns
- Each line in the column has a set number of bits
- Lines are grouped in blocks
- Redundancy is introduced (LDPC block)
- Each line is encoded into DNA using a motif



Eugenio Marinelli, Yiqing Yan, Lorenzo Tattini, Virginie Magnone, Pascal Barbry, Raja Appuswamy, “CMOSS: A Reliable, Motif-based Columnar Molecular Storage System”, 17th ACM International Systems and Storage Conference

CMOSS - A motif-based coding system

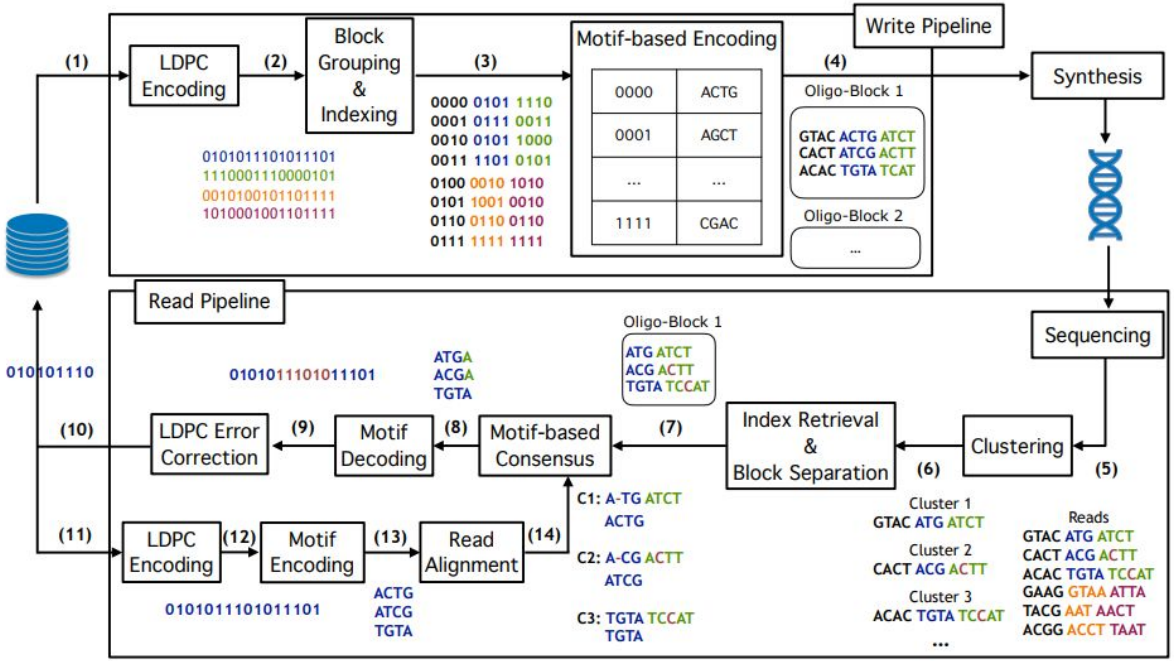
- Preconstructed set of motifs that can be used to encode data
- Translation table
- Motifs are synthesized in large numbers
- Ligated (concatenated) into larger oligos

Eugenio Marinelli, Yiqing Yan, Lorenzo Tattini, Virginie Magnone, Pascal Barbry, Raja Appuswamy, “CMOSS: A Reliable, Motif-based Columnar Molecular Storage System”, 17th ACM International Systems and Storage Conference

CMOSS - A consensus-driven coding system

- PCR allows the cheap copy of DNA molecules
- Easier to decode noised data when you have a lot of copies of it
  - Try to sequence all of the molecules
  - Operate a clusterization and consensus (merge the data from all the copies)
- Compute the minimal coverage (average number of copies) necessary to read, to be sure to decode the data

CMOSS - General workflow



Eugenio Marinelli, Yiqing Yan, Lorenzo Tattini, Virginie Magnone, Pascal Barbry, Raja Appuswamy, “CMOSS: A Reliable, Motif-based Columnar Molecular Storage System”, 17th ACM International Systems and Storage Conference

Examples of DNA Image Coders

AI-based Image coder (HiDNA)

Motivations

- Learning-based models such as JPEG AI are very promising
- AI-based methods can be used to adapt to noise

T. Ladune, P. Philippe, F. Henry, G. Clare, T. Leguay, “COOL-CHIC: Coordinate-based Low Complexity Hierarchical Image Codec”, ICCV 2023

Principle - Synthesis Model

- The synthesis model learns to reconstruct an image from a latent space
- Latent space is composed of a series of matrices of different sizes

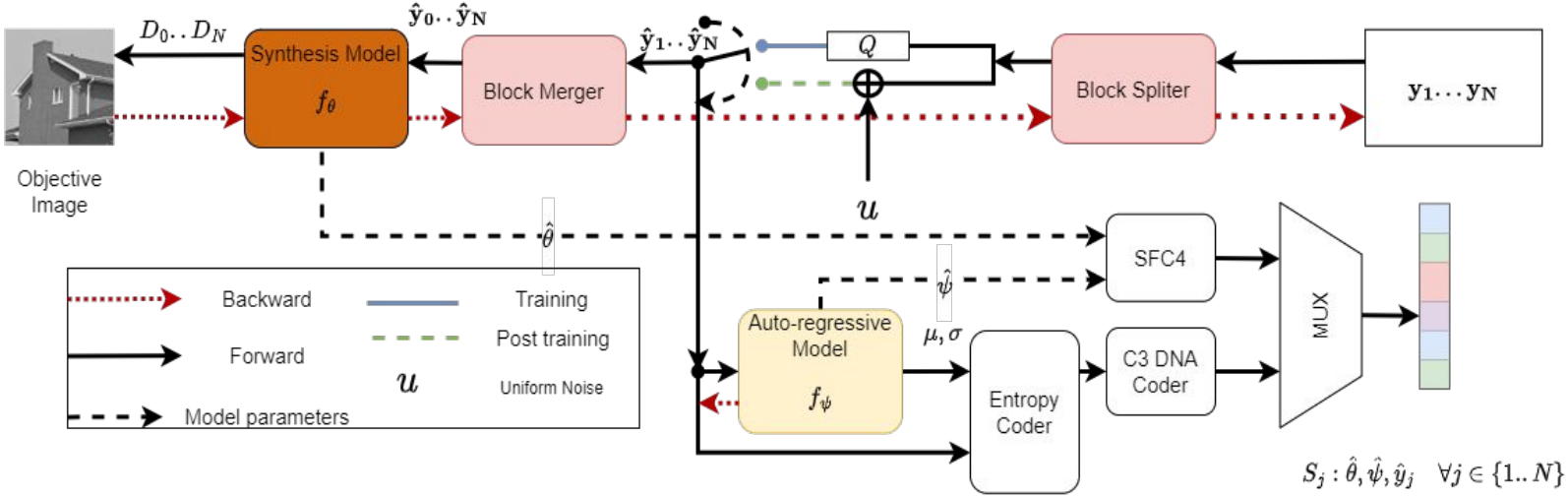


Fig: Workflow of the synthesis training and encoding processes in HiDNA

1. T. Ladune, P. Philippe, F. Henry, G. Clare, T. Leguay, “COOL-CHIC: Coordinate-based Low Complexity Hierarchical Image Codec”, ICCV 2023
2. T.H. Le, X. Pic, J. Mateos and M. Antonini, “Implicit Neural Multiple Description for DNA-based data storage”, ICASSP 2023

Synthesis model - Training phase

A loss based on the minimization of the **rate** of the latent space and the **reconstruction quality** of the image:

$$L = D(I, \hat{I}) + \lambda \times H(y)$$

where:

- I: Input image
- $\hat{I}$: Reconstructed image
- y: Latent space
- D: Distortion metric
- H: Entropy

Principle - Latent Space Coding

- Fully connected neural network (ARM Model)
- Learns the Probability Density Function of each element of the latent space, depending on its context
- The PDF is used to instantiate an arithmetic, binary entropy coder, associated with a DNA adapted coder

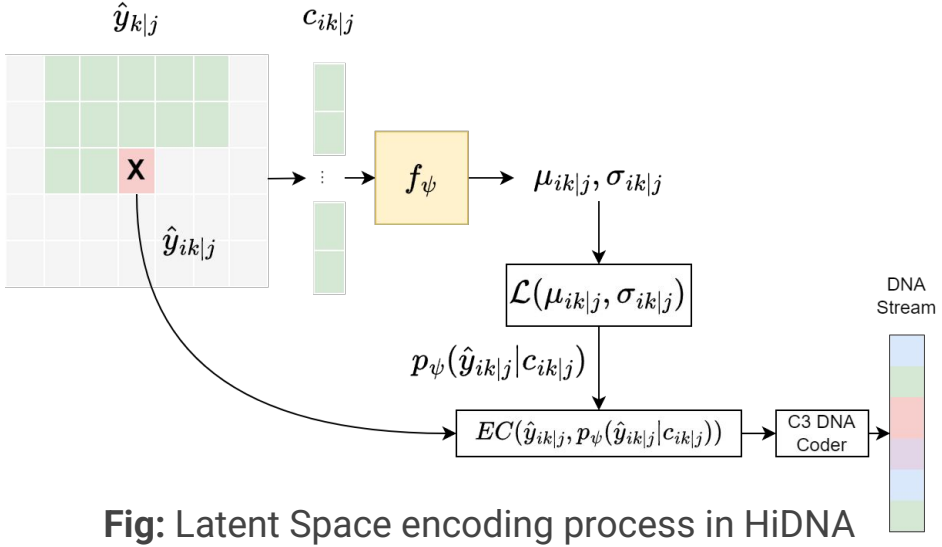
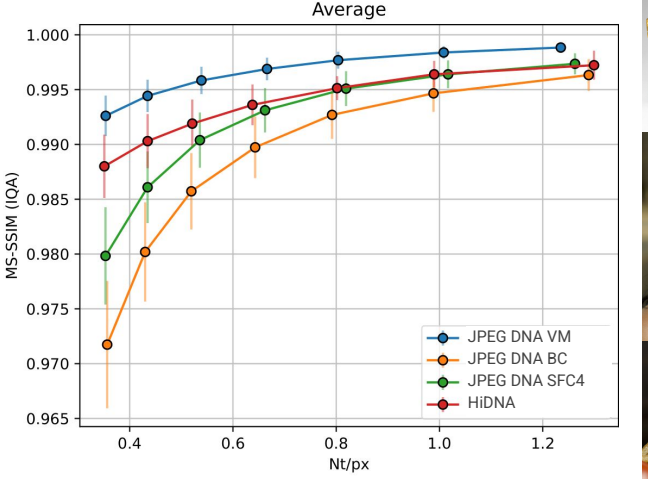


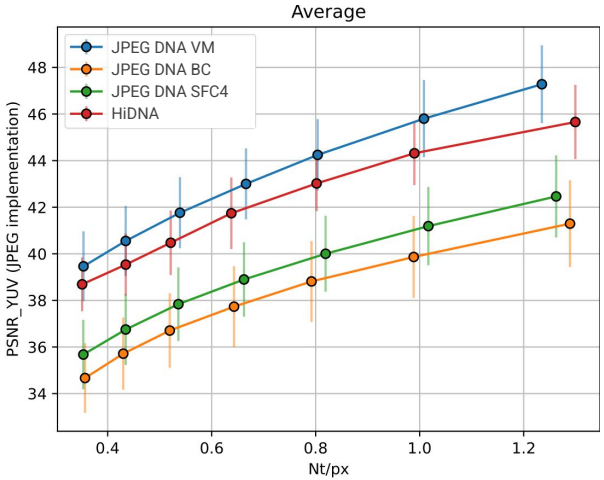
Fig: Latent Space encoding process in HiDNA

Results

- HiDNA overperforms all other state of the art source coders

- Only the JPEG DNA VM (**open loop** adaptation of JPEG XL to DNA) performs better





M. Testolina, V. Hosu, M. Jenadeleh, D. Lazzarotto, D. Saupe, T. Ebrahimi, “JPEG AIC-3 Dataset: Towards Defining the High Quality to Nearly Visually Lossless Quality Range”, *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*, Ghent, Belgium, 2023

Conclusion

- A series of coding softwares have been adapted to DNA data storage or implemented
  - For images (source coders and joint source/channel coders):
    - JPEG DNA SFC4, JPEG 2000 DNA, HiDNA
  - For any kind of data (channel coders):
    - C3, SFC4, Raptor Code
- Fairly new field of research -> Still opportunities for improvement in several fields (coding theory, information theory)

References

- PEPR MoleculArxiv (French Gvt - ANR)
- Molecular-Scale Data Storage and Archiving (USA - IARPA)
- DNA Data Storage Alliance (SNIA Technology Groups)
- Companies and labs: Wyss Institute, Twist Bioscience, Catalog DNA, Biomemory, PearCode

Thank you !

EURECOM - Data Science

xavier.pic@eurecom.fr
xavpic.github.io

References

- Y. Erlich and D. Zielinski, “DNA fountain enables a robust and efficient storage architecture” Science, vol. 355, no. 6328, pp.950–954, 2017
- George M. Church, Yuan Gao, and Sriram Kosuri, “Next-Generation Digital Information Storage in DNA”, Science
- N. Goldman, P. Bertone, and S. Chen, “Towards practical, high-capacity, low-maintenance information storage in synthesized DNA,” Nature, 2013
- A. Banerjee, A. Wachter-Zeh and E. Yaakobi, "Insertion and deletion correction in polymer-based data storage", IEEE Trans. Inf. Theory
- R. Gabrys, S. Pattabiraman and O. Milenkovic, "Mass error-correction codes for polymer-based data storage", IEEE Int. Symp. Inf. Theory (ISIT)
- 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", Angew. Chem. Int. Ed.
- M. Dimopoulou, E. Gil San Antonio, M. Antonini, “A JPEG-based image coding solution for data storage on DNA”, EUSIPCO, 2021
- Davi Lazzarotto, Jorge Encinas Ramos, Michela Testolina, Touradj Ebrahimi, “Storing images and point clouds on DNA support with fountain codes”, Applications of Digital Image Processing XLVII
- Eugenio Marinelli, Yiqing Yan, Lorenzo Tattini, Virginie Magnone, Pascal Barbry, Raja Appuswamy, “CMOSS: A Reliable, Motif-based Columnar Molecular Storage System”, 17th ACM International Systems and Storage Conference
- X. Pic and M. Antonini, “A constrained shannon-fano entropy coder for image storage in synthetic DNA”, EUSIPCO 2022
- X. Pic, M. Dimopoulou, E. Gil San Antonio and M. Antonini, “MQ-Coder inspired Arithmetic Coder for Synthetic DNA data storage”, ICIP 2023
- T.H. Le, X. Pic, J. Mateos and M. Antonini, “Implicit Neural Multiple Description for DNA-based data storage”, ICASSP 2023