STORAGE DEVELOPER CONFERENCE

SDC 22

Fremont, CA
September 12-15, 2022

BY Developers FOR Developers

A SNIA Event

DNAssim: A Full System Simulator for DNA Storage

Alessia Marelli[1], Thomas Chiozzi[1] , Lorenzo Zuolo[1],Nicholas Battistini[1], Piero Olivo[2], Cristian Zambelli[2], Rino Micheloni[1,2]

[1] DNAalgo

[2] Università degli studi di Ferrara

DNA DATA STORAGE ALLIANCE
A SNIA Technology Affiliate

Outline

- The need of new storage media
- What is DNA storage
- Error sources
- Edit Distance
- Why DNAssim
- Encoding & decoding
- SW/HW co-simulation
- Conclusions

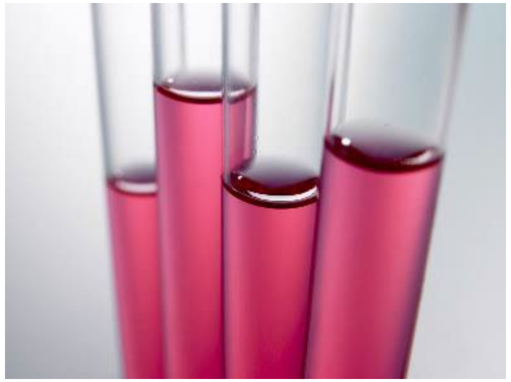
Need of new storage media

Why DNA?

DNAalgo

- More and more applications are data hungry
  - Earth is covered with data centers
- DNA storage enables

Longevity



Low power

Capacity

DNA issues

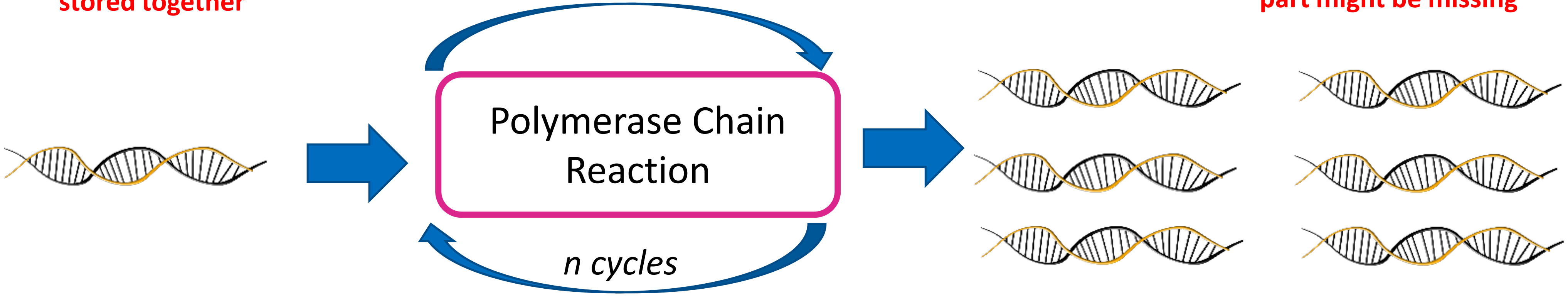
- Nothing comes for free, so the main DNA storage issues are

A huge amount of data are stored together

Data are read without order

Channel IDS

PCR replicas: some part might be missing



Polymerase Chain Reaction

n cycles

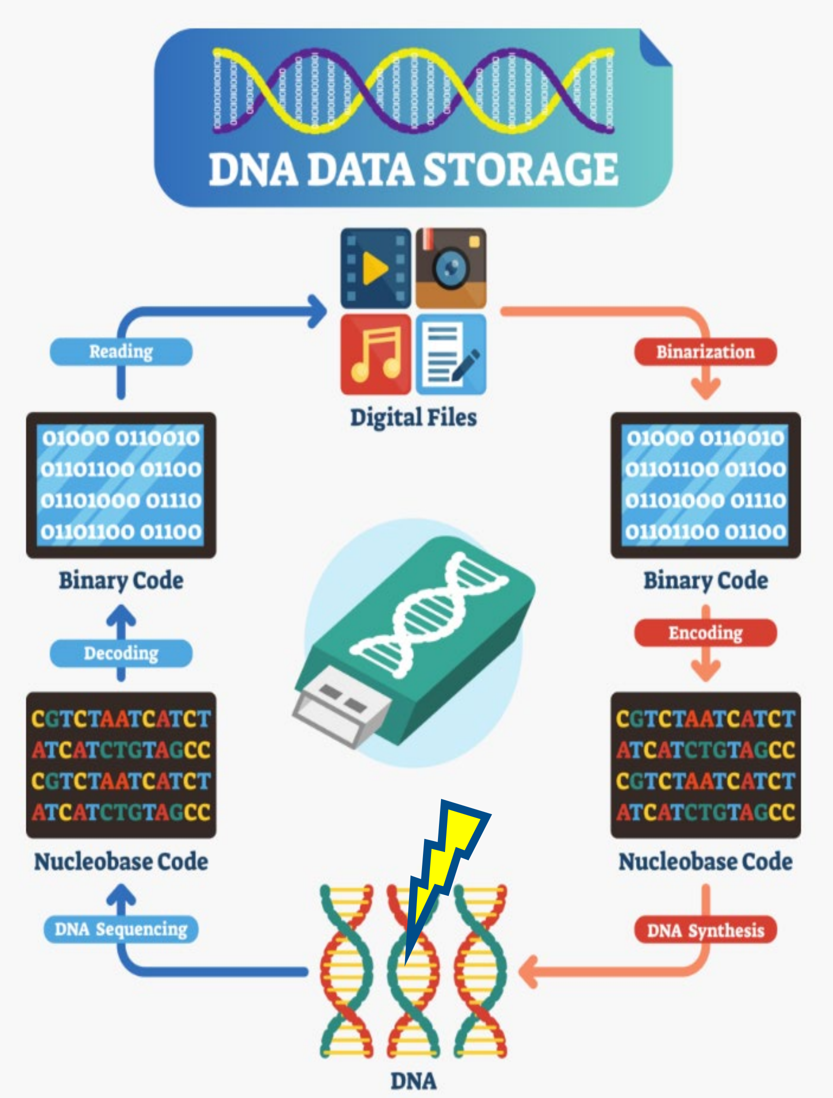
- At DNAalgo we believe that data “manipulation” is the only way for making DNA storage reliable and fast enough for the storage industry; without reliability and speed, DNA storage won’t go too far from Today’s proof-of-concept stage

DNA data storage

DNA storage

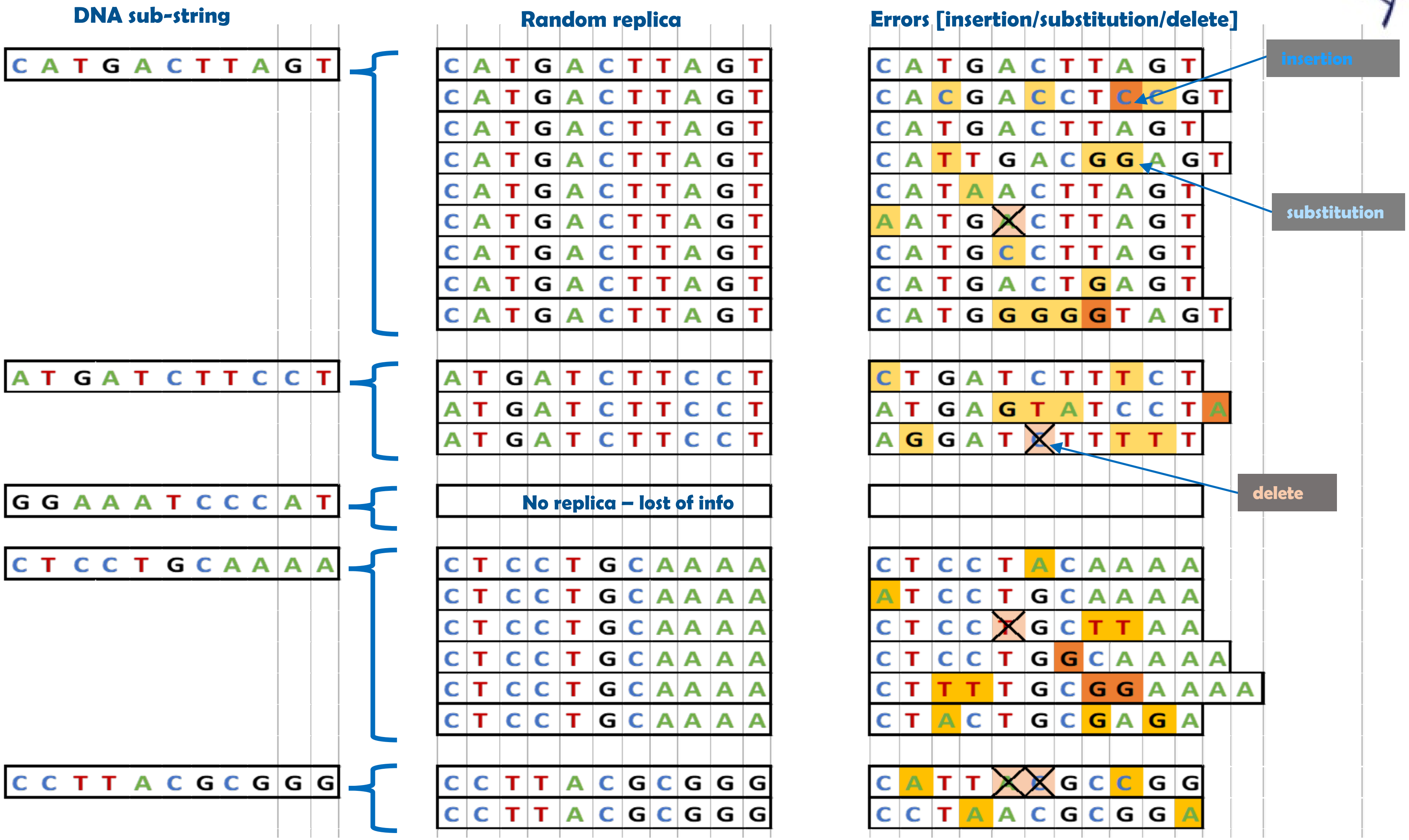
DNAalgo

- During synthesis, sequencing and storing some errors might occur.
- Errors can be insertion, deletion and substitution
- In addition to that, in order to sequence the information PCR is applied so that each strand is read a variable number of times (also 0 times)



https://dnaalgo.com/

Information Channel example



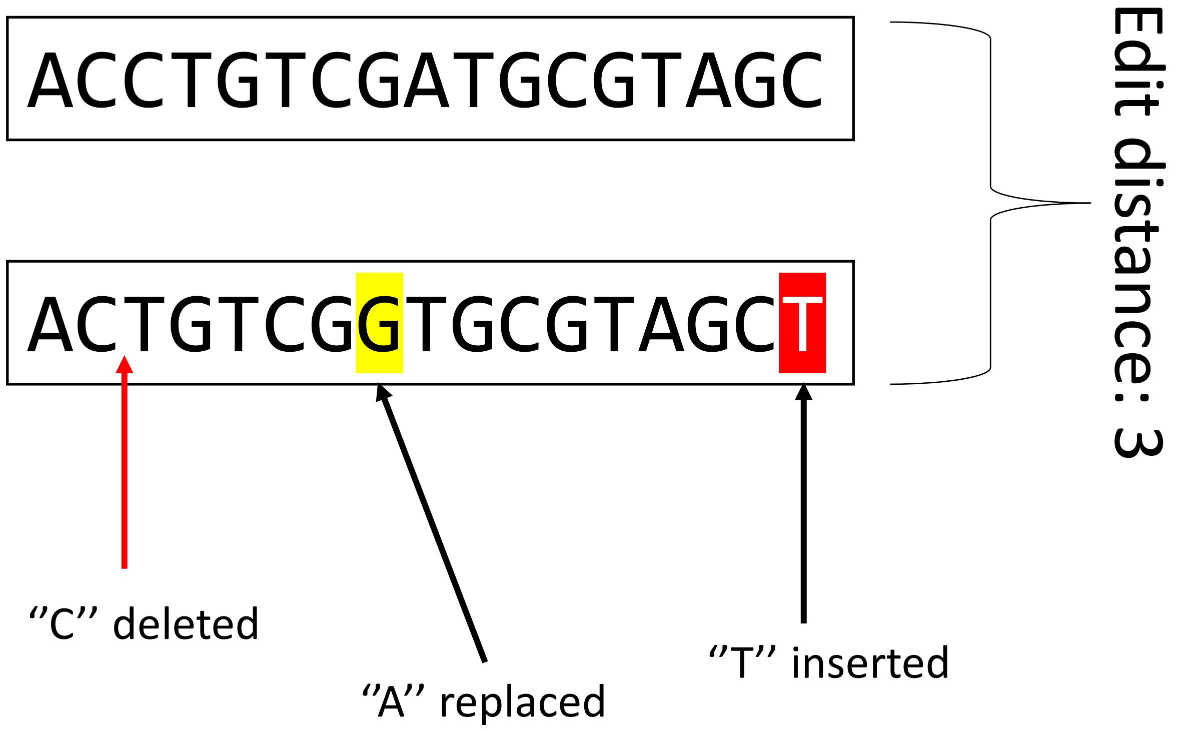
DNA sub-string	Random replica	Errors [insertion/substitution/delete]
CATGACTTAGT	CATGACTTAGT	CATGACTTAGT
	CATGACTTAGT	CACGACCTCCGT
	CATGACTTAGT	CATGACTTAGT
	CATGACTTAGT	CATTGACGGAGT
	CATGACTTAGT	CATAACTTAGT
	CATGACTTAGT	AATG~~A~~CTTAGT
	CATGACTTAGT	CATGCCTTAGT
	CATGACTTAGT	CATGACTGAGT
	CATGACTTAGT	CATGGGGGTAGT
ATGATCTTCCT	ATGATCTTCCT	CTGATCTTTCT
	ATGATCTTCCT	ATGAGTATCCTA
	ATGATCTTCCT	AGGAT~~C~~TTTTT
GGAAATCCCAT	No replica – lost of info	
CTCCTGCAAAA	CTCCTGCAAAA	CTCCTACAAAA
	CTCCTGCAAAA	ATCCTGCAAAA
	CTCCTGCAAAA	CTCC~~T~~GCTTAA
	CTCCTGCAAAA	CTCCTGGCAAAA
	CTCCTGCAAAA	CTTTTGCGGAAAA
	CTCCTGCAAAA	CTACTGCGAGA
CCTTACGCGGG	CCTTACGCGGG	CATT~~A~~~~C~~GCCGG
	CCTTACGCGGG	CCTAACGCGGA

DNA DATA STORAGE ALLIANCE
A SNIA Technology Affiliate

STORAGE DEVELOPER CONFERENCE

Edit distance

Edit distance: example



ACCTGTCGATGCGTAGC

ACTGTCGGTGCGTAGCT

"C" deleted

"A" replaced

"T" inserted

Edit distance: 3

- In an IDS channel, the metric used is the Levensthein distance.
- Algorithm to compute it and also recovering messages can be much harder when dealing with this distance

Evaluating the edit distance

- To evaluate the edit distance, one can use a well known **dynamic programming algorithm.**
- We describe the traditional algorithm in the next slides using an example between strings (NOTE: you can treat DNA sequences as strings in the {A,C,G,T} alphabeth).
- If you have two strings, one of length N and the other of length M, you can evaluate the edit distance by recursively evaluating a matrix of size (N+1) x (M+1).
- The matrix if filled according to a formula.
- The output of the algorithm will be last cell of the matrix.

Wagner Fischer algorithm

DNAalgo

- **Example:** compare the 5 symbol string paolo with the 6 symbol string `Paolo!`. We use le letter x for `paolo`, y for `Paolo!`.
- **Initialize the matrix:** first row and first column are an increasing sequence.

		p	a	o	l	o
	0	1	2	3	4	5
P	1					
a	2					
o	3					
l	4					
o	5					
!	6					

Initialized values

Will contain output of the algorithm

Wagner Fischer algorithm: cell evaluation

- The value D[i, j] is evaluated as follows:

1. $s = \begin{cases} 0, & x_i = y_j \\ 1, & x_i \neq y_j \end{cases}$, depends on the input strings x and y.

2. $D[i,j] = \min \begin{pmatrix} D[i-1,j-1]+s, \\ D[i-1,j]+1, \\ D[i,j-1]+1 \end{pmatrix}$

- NOTE: with x_i we mean the i-th letter in the string x.

Wagner Fischer algorithm

- **Recursively evaluate the matrix:** canonic evaluation is row by row.
- **Outcome:** last cell of the matrix.

		p	a	o	l	o
	0	1	2	3	4	5
P	1	D[1, 1]				
a	2					
o	3					
l	4					
o	5					
!	6					

S = 1

D[1,1] = min(0+1, 2, 2)

Wagner Fischer algorithm

- **Recursively evaluate the matrix**

		p	a	o	l	o
	0	1	2	3	4	5
P	1	1	2			
a	2					
o	3					
l	4					
o	5					
!	6					

S = 1

D[2,1] = min(2, 3, 2)

Wagner Fischer algorithm

- **Recursively evaluate the matrix**

		p	a	o	l	o
	0	1	2	3	4	5
P	1	1	2	3	4	5
a	2	2	1			
o	3					
l	4					
o	5					
!	6					🟨

S = 0

D[2,2] = min(1, 3, 3)

Wagner Fischer algorithm

- **Recursively evaluate the matrix**

		p	a	o	l	o
	0	1	2	3	4	5
P	1	1	2	3	4	5
a	2	2	1	2	3	4
o	3	3	2	1	2	3
l	4	4	3	2	1	2
o	5	5	4	3	2	1
!	6	6	5	4	3	**2**

Edit distance between ”paolo” and ”Paolo!”

Introducing DNAssim

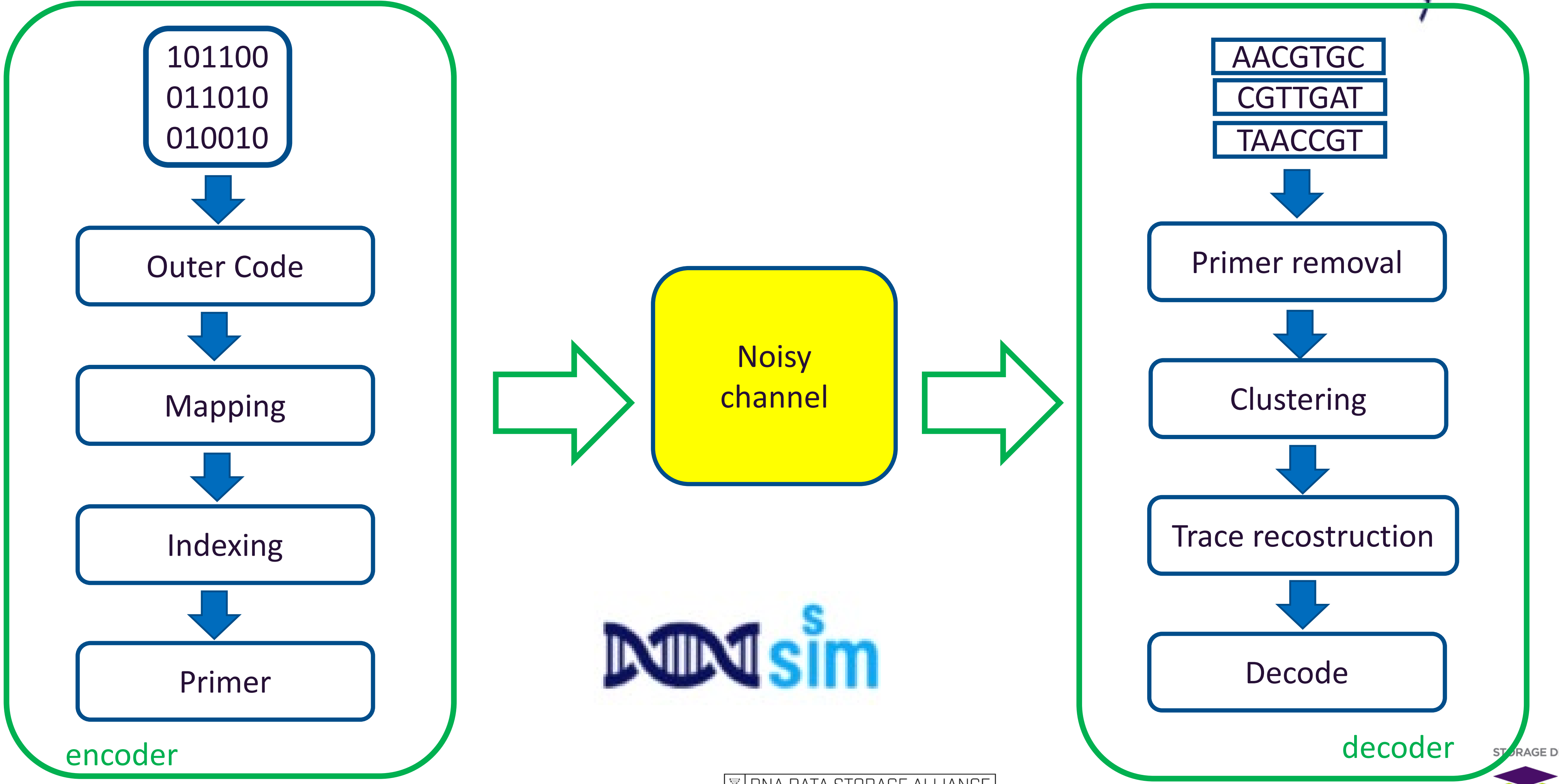
Why a simulator?

DNAalgo

- While encoding and decoding can be described by a set of equations, errors are not deterministic and must be modeled.
- Encoding and Decoding can be optimized if tailored to a specific noise model.
- Because of the intrinsic statistical behavior of the noise, a simulator is required for figuring out the impact of encoding/decoding algorithms.

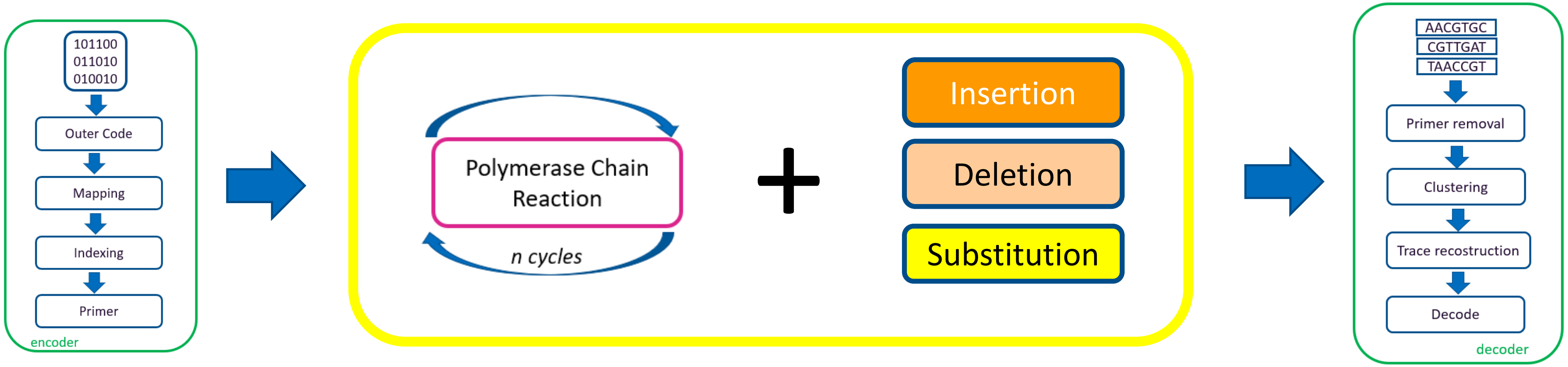
Introducing DNAssim (DNAssim)

DNAalgo



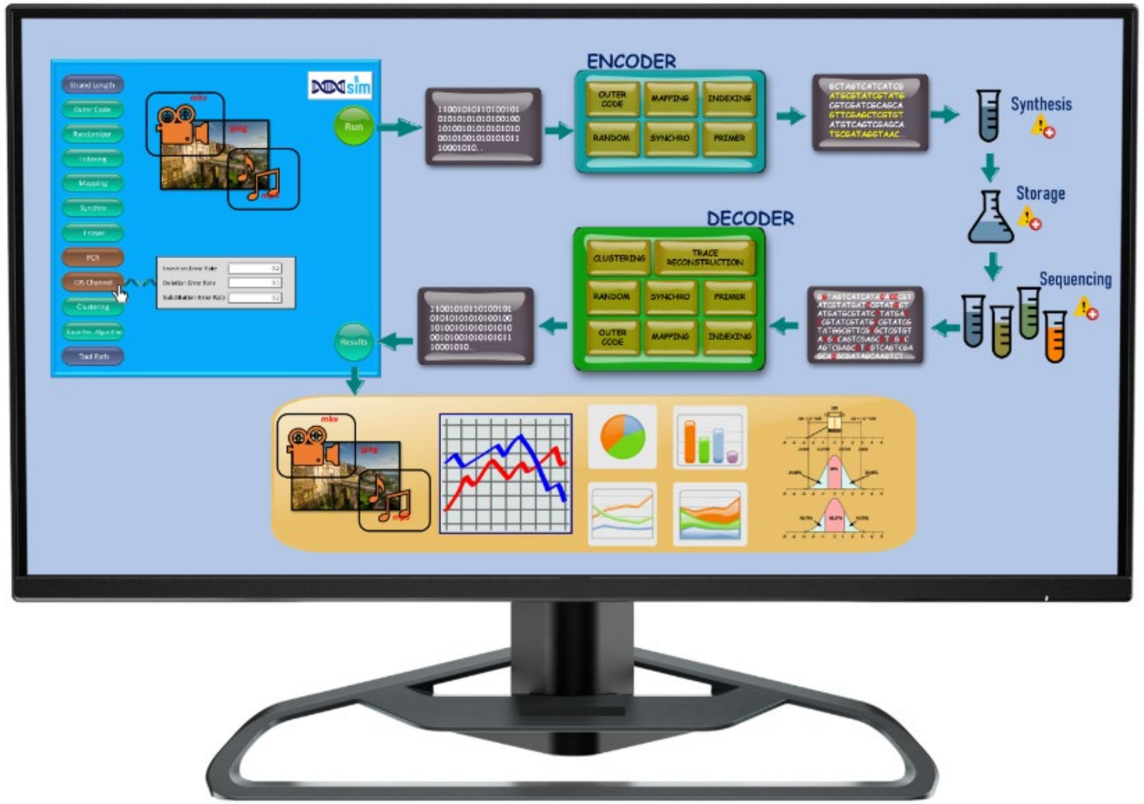
Noise Model

- Noise can be modeled as PCR (Polymerase Chain Reaction) + IDS (**I**nsertion **D**eletion **S**ubstitution) Channel
- PCR is represented by a variable number of strand replicas
  - Tunable multiplicity
- IDS channel translates into a statistical number of apply insertion, deletion and substitution for each strand
  - Tunable substitution/insertion/deletion probabilities



Simulation tool

- DNAssim is managed by a Graphical User Interface (GUI), where all the different parameters and options can be chosen
- When simulation is completed a bunch of graphs and texts are output in order to analyze results.



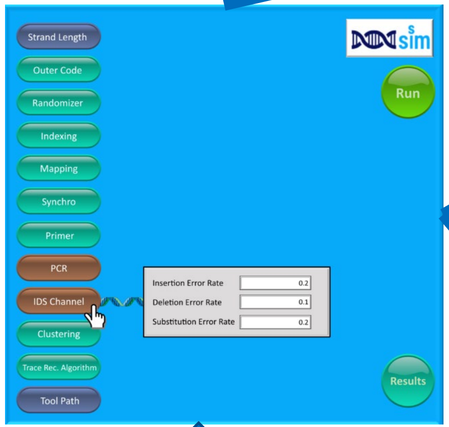
Simulation tool

https://dnaalgo.com/

Comparing results

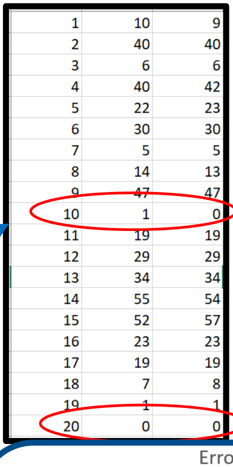
Outer Code: A
Trick on code: A,C
Mapping: natural
Randomizer: on
Indexing: A
Cluster: A
Trace Rec: A
Test 1

Outer Code: B
Trick on code: B,C
Mapping: natural
Randomizer: on
Indexing: A
Cluster: D
Trace Rec: A
Test 2

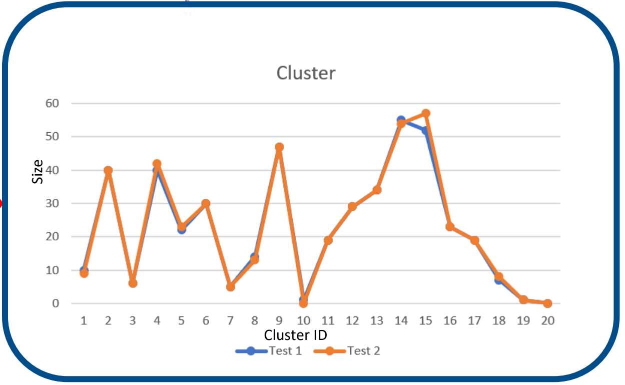


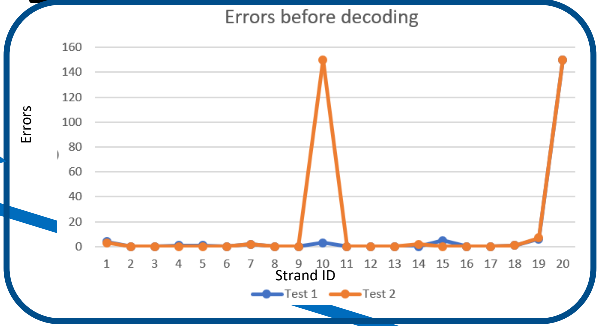
Insertion Error Rate 0.2
Deletion Error Rate 0.1
Substitution Error Rate 0.2

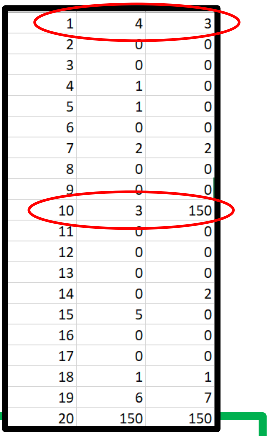
Percentage of G/C : 45%
Number of initial strands: 20
Number of sequenced strands: 450



1	10	9
2	40	40
3	6	6
4	40	42
5	22	23
6	30	30
7	5	5
8	14	13
9	47	47
10	1	0
11	19	19
12	29	29
13	34	34
14	55	54
15	52	57
16	23	23
17	19	19
18	7	8
19	1	1
20	0	0







1	4	3
2	0	0
3	0	0
4	1	0
5	1	0
6	0	0
7	2	2
8	0	0
9	0	0
10	3	150
11	0	0
12	0	0
13	0	0
14	0	2
15	5	0
16	0	0
17	0	0
18	1	1
19	6	7
20	150	150

Missing strands: 1
Errors before decoding: 23
Number of recovered strand: 19

Test 1

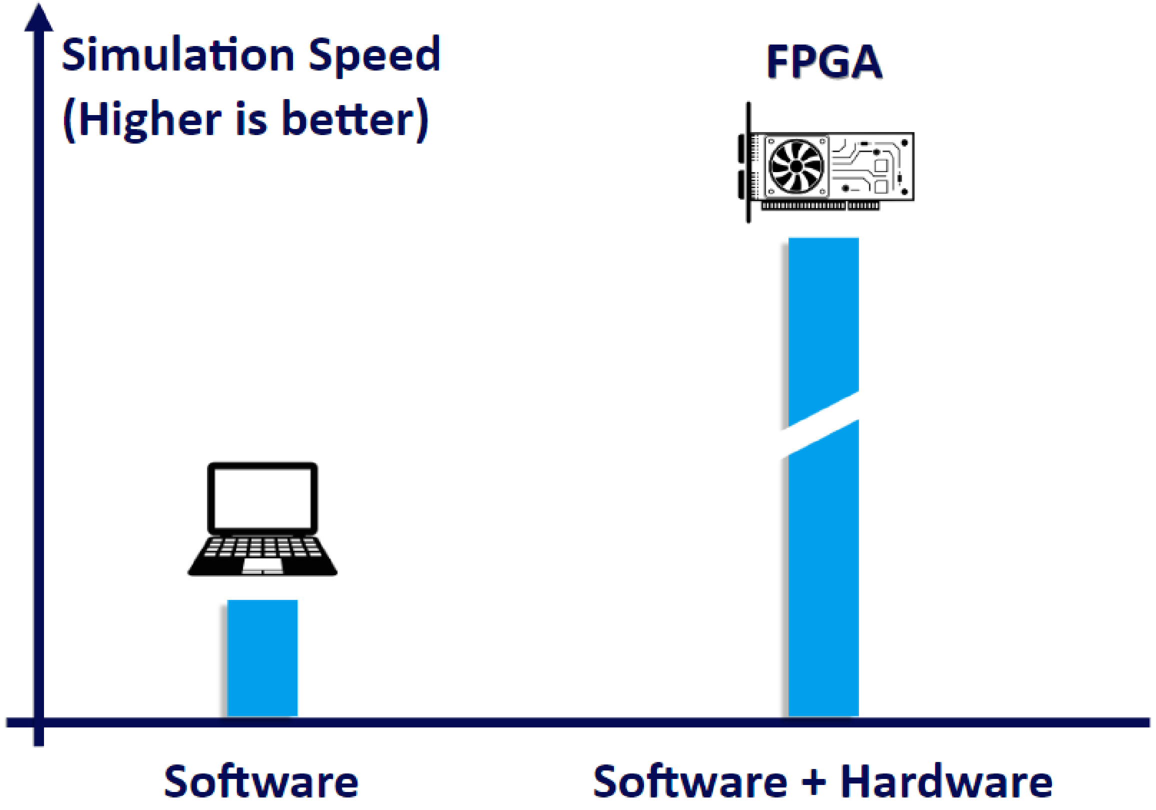
Missing strands: 2
Errors before decoding: 15
Number of recovered strand: **20**

Test 2

SW/HW co-simulation

HW/SW co-simulation

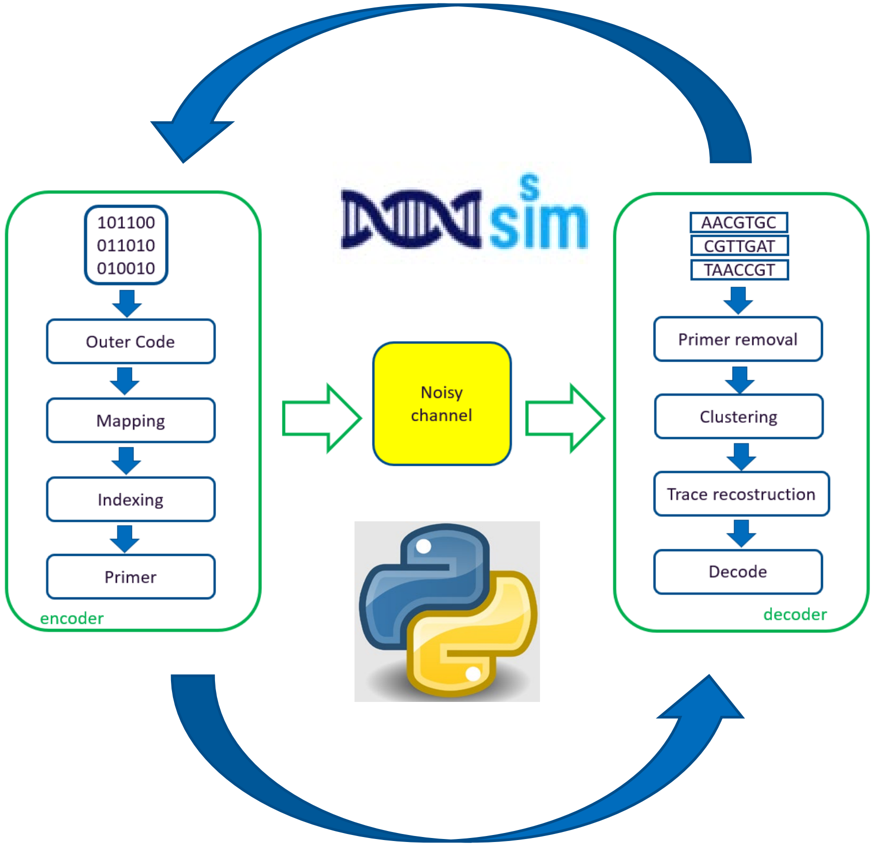
Because of the number and complexity of the steps involved in the DNA storing process, the number of simulations is huge and a “pure software” simulator can easily run out of gas. To overcome this limitation, at DNAalgo we developed a custom co-simulation (i.e. mix of hardware and software) platform



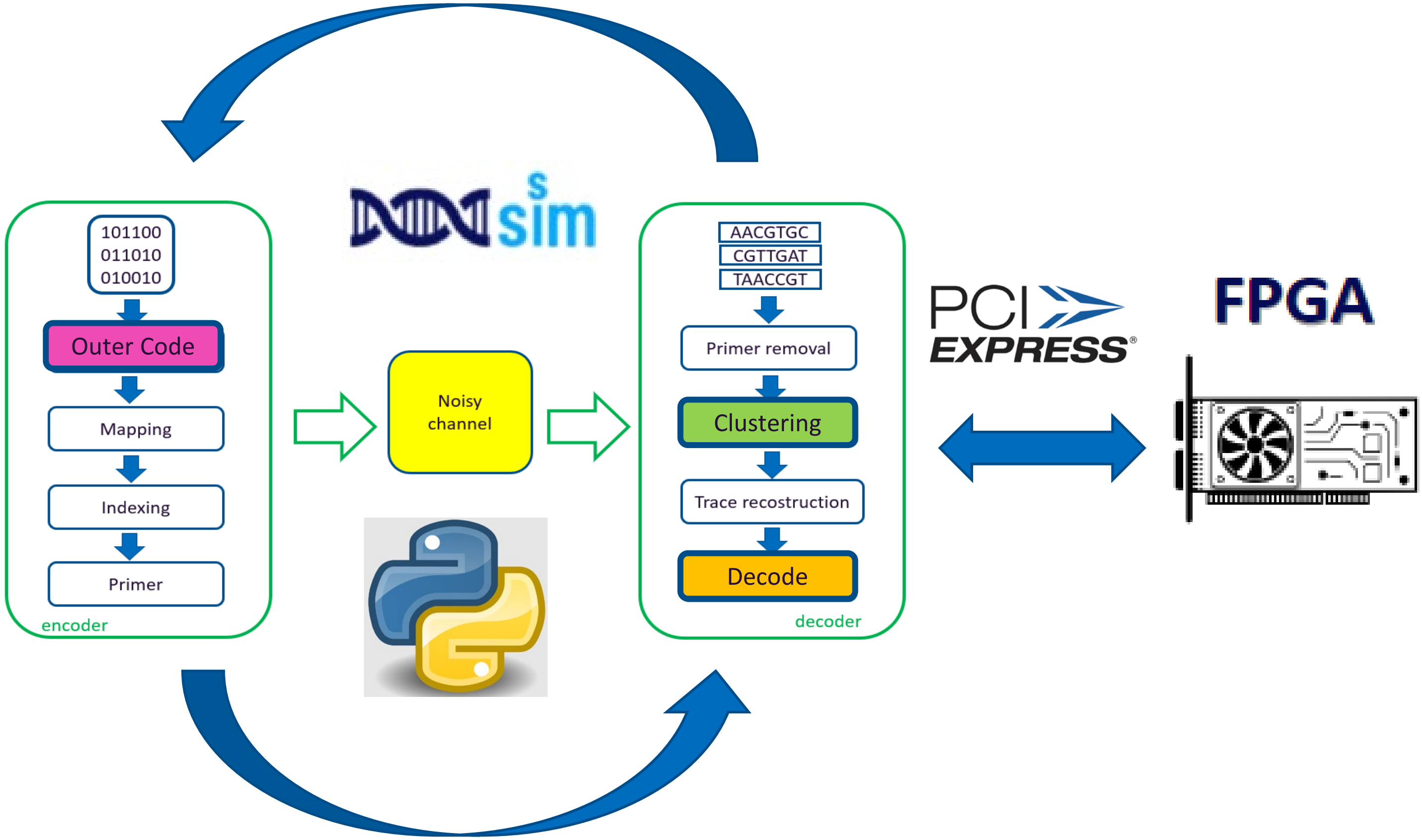
https://dnaalgo.com/

HW/SW co-simulation

SW

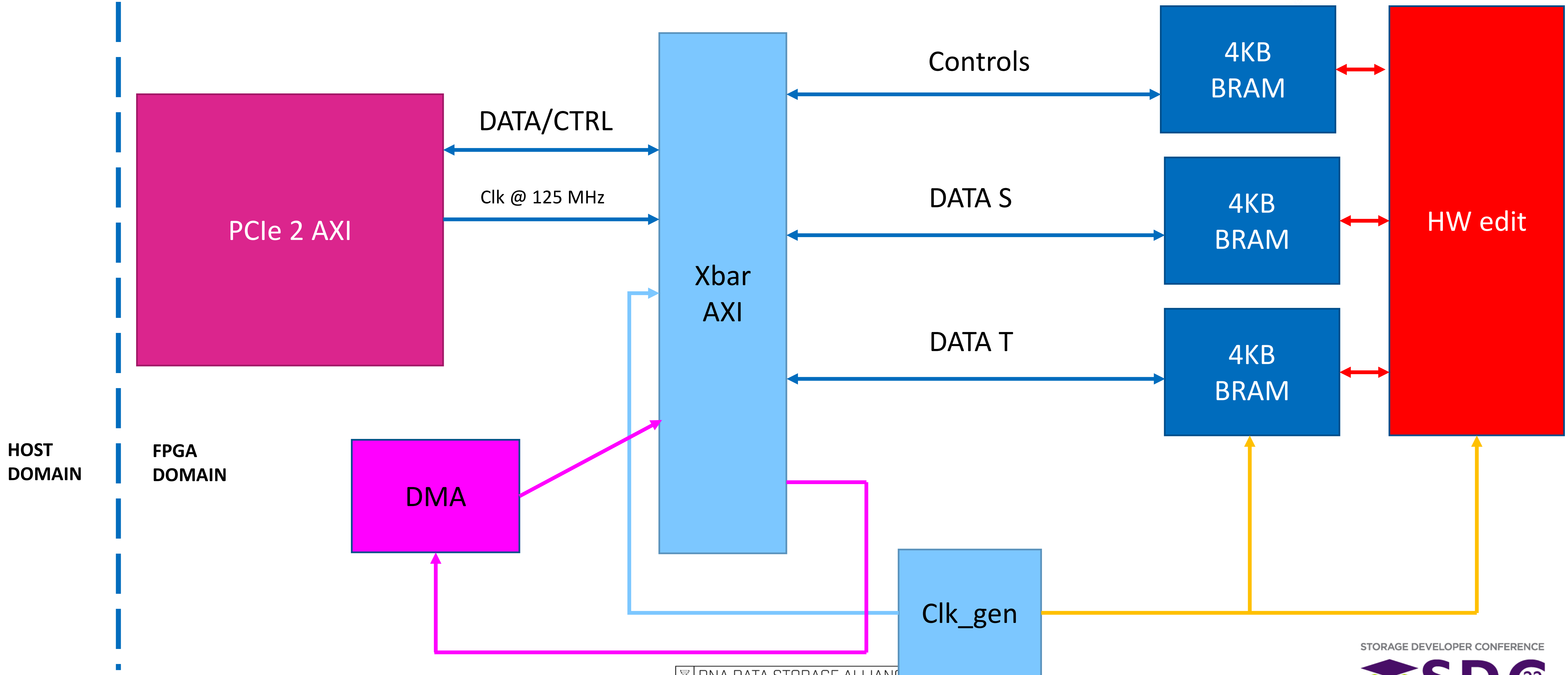


HW/SW



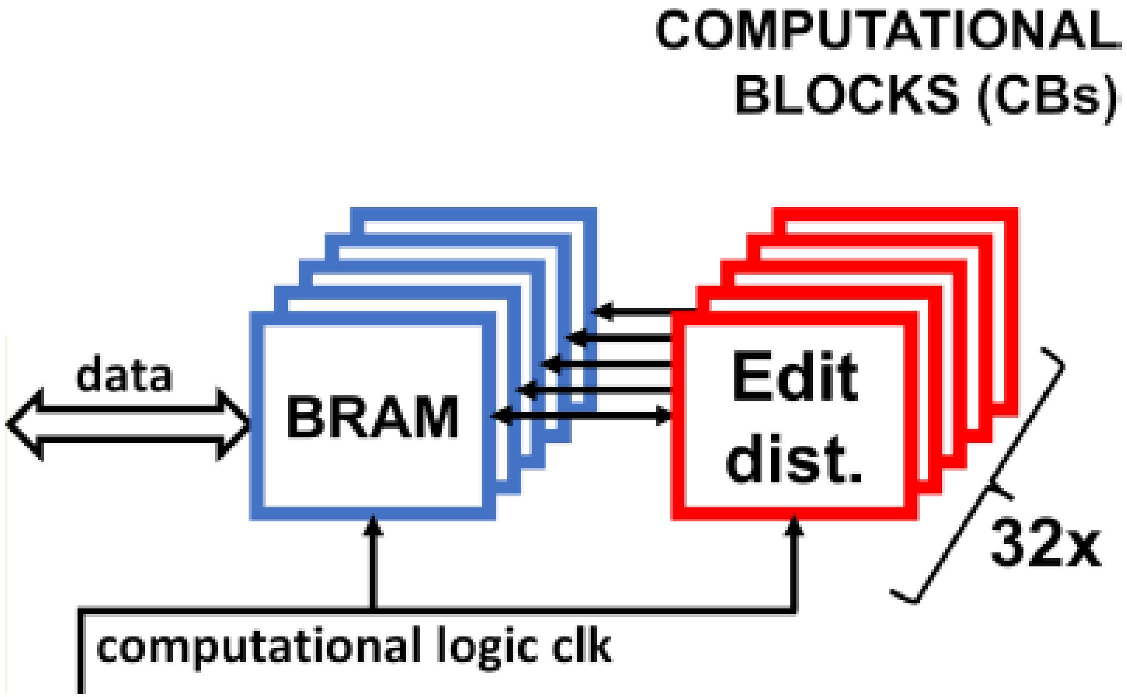
HW acceleration: block diagram

DNAalgo



HW acceleration: parallelism

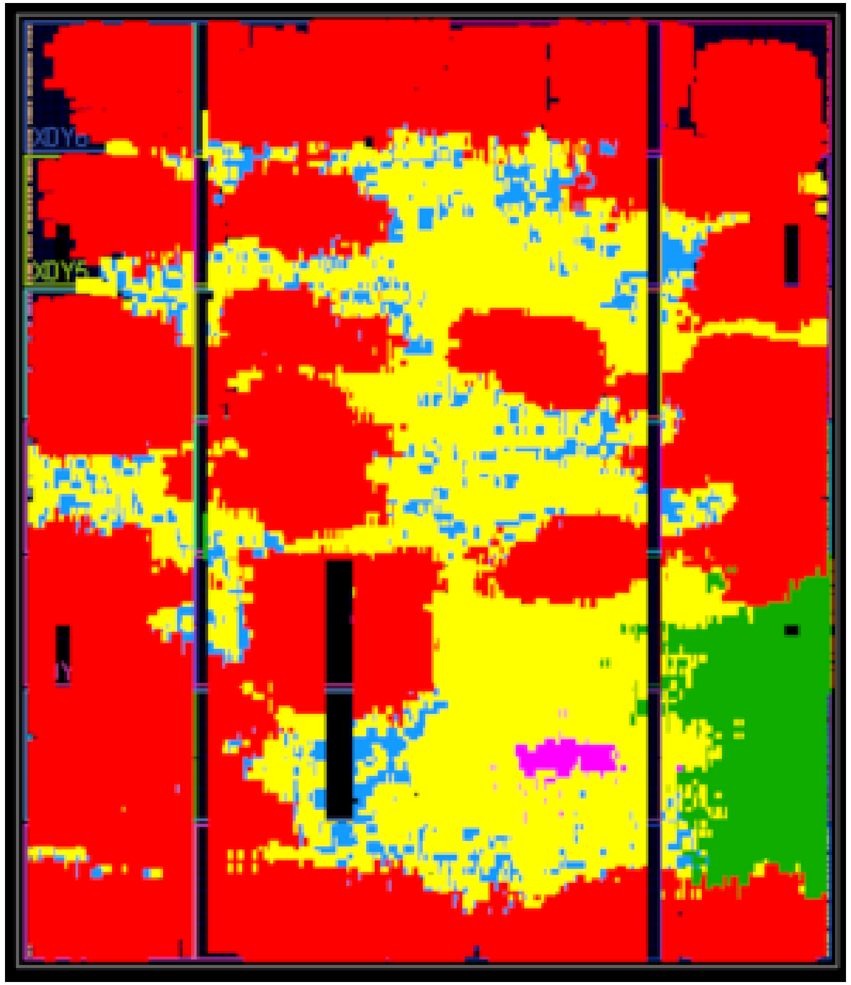
- The hardware design to speed up the computation of the edit distance is based on BRAM blocks instantiated for each computational block that can store up to 4 KB of data
- First generation: 32 BRAMs implemented coupled with 32 CBs allow calculating up to 224 results (DNA pairs).
- 87.4% occupation of the BRAMs



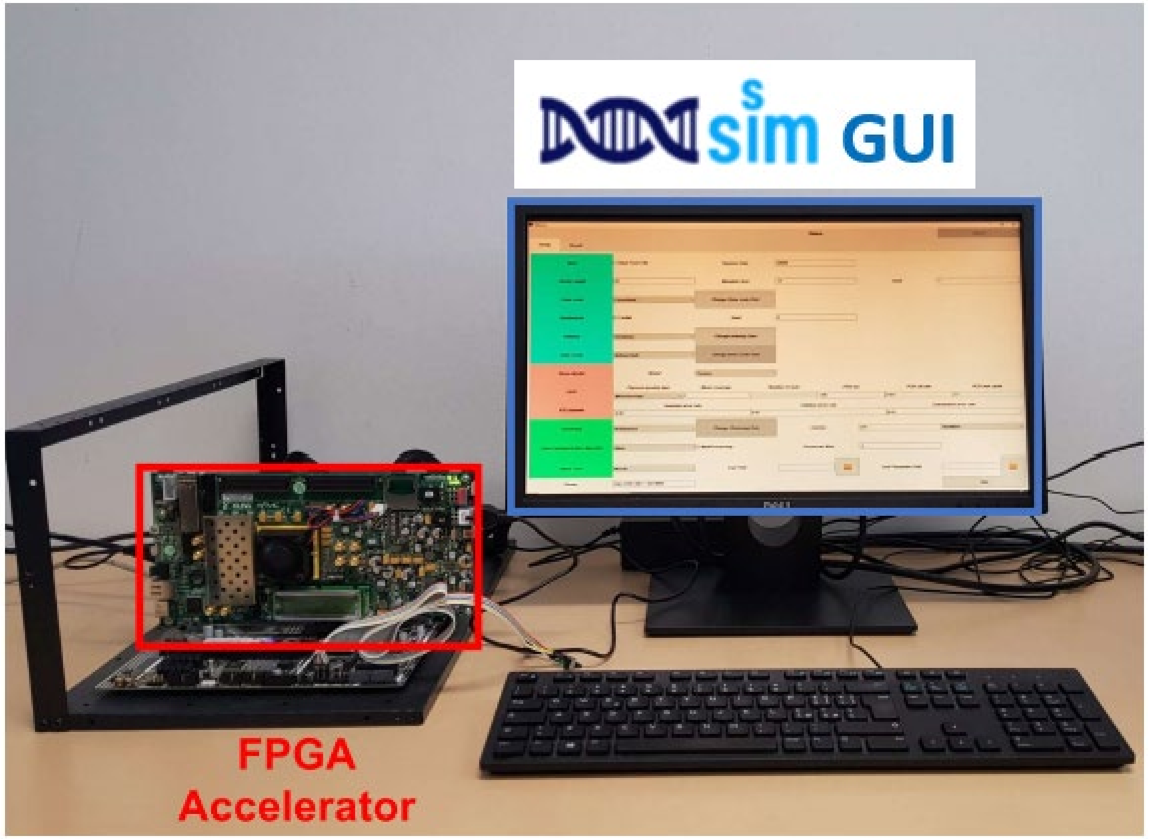
FPGA utilization

DNAalgo

- Floorplan of the XC7VX485T FPGA implementing a 32 CBs edit distance hardware accelerator.
- Red -> Computational Blocks
- Green -> PCIe I/F
- Magenta -> DMA
- Blue -> BRAM
- Yellow -> AXI Xbar



A photograph of the test rig used to assess the performance of the DNAssim framework. The Graphical User Interface (GUI) of the software engine and the FPGA-based hardware accelerator attached to the host motherboard are highlighted



Conclusions

- A new media is needed to store all data produced every day
- DNA storage is a promising candidate
- Encoding and Decoding involve multiple functions -> much more complicated w.r.t. Flash or HDD
- Noise channel can be modeled as a combination of PCR + IDS channel
- DNAssim is used to find the best encoding and decoding combinations tailored to a specific error model
- DNA simulations are accelerated by a combination of HW/SW (co-simulation)

DNAalgo

THANK YOU!

https://dnaalgo.com/