



Technion
Israel Institute of Technology



Toward
Efficient Genomic Compression
using Channel Codes for
Joint Alignment and Reconstruction

Yuval Cassuto

Joint w/ Yotam Gershon

Technion – Electrical and Computer Engineering

Simons Workshop on Application-Driven Coding Theory

March 5, 2024

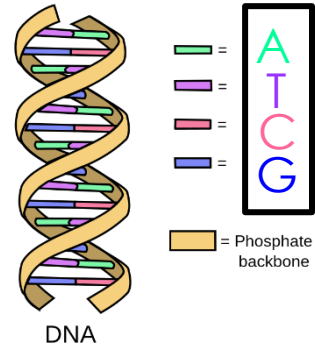
Genomic Data in Medicine and Science

- Virus detection and identification (e.g. Covid-19)
- Prenatal genetic diagnosis
- Species evolution studies
- Many more...
- Personalized medicine

Key features of DNA information

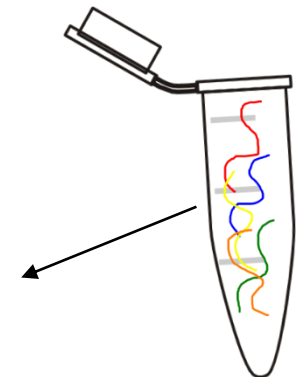
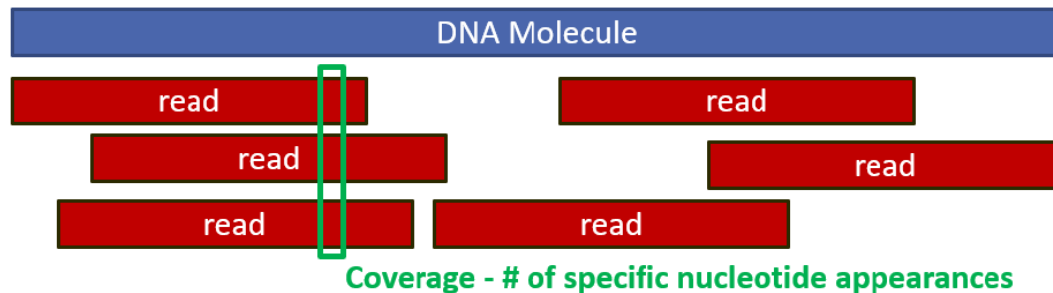
1. The nature of the info

- **Vast similarities** between individuals
 - 99.99% to our neighbor
 - 97% to a chimpanzee

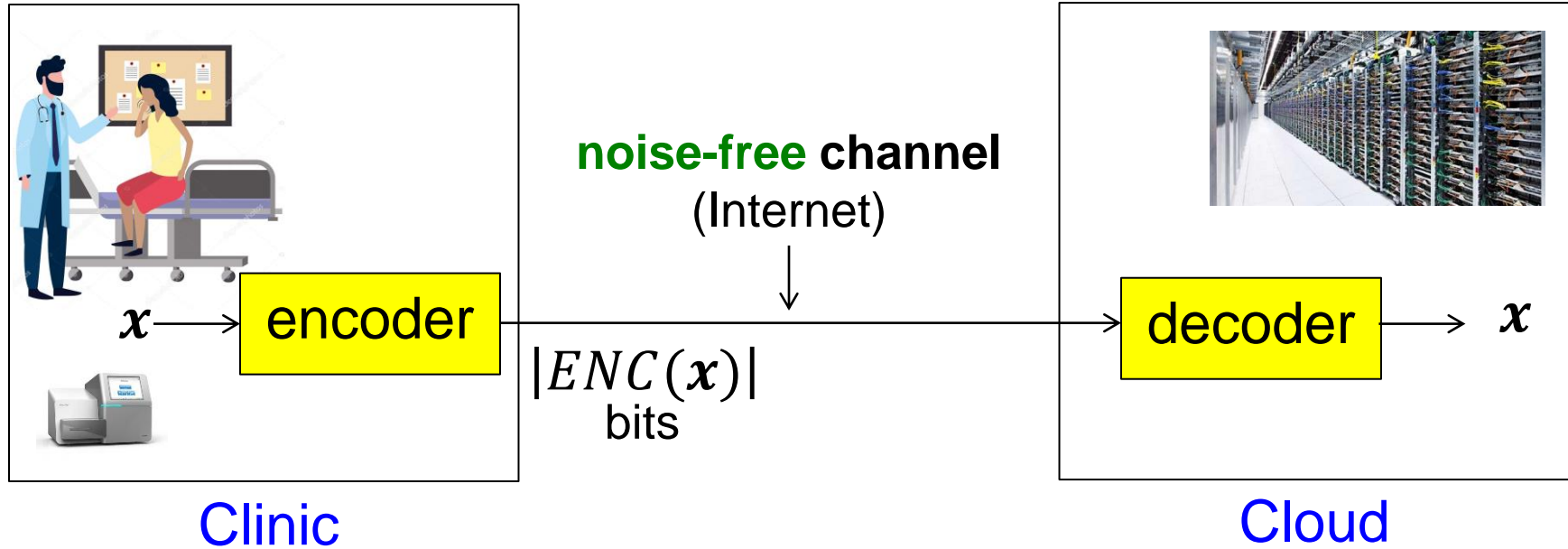


2. The format of the info

- Sequencing **short reads**

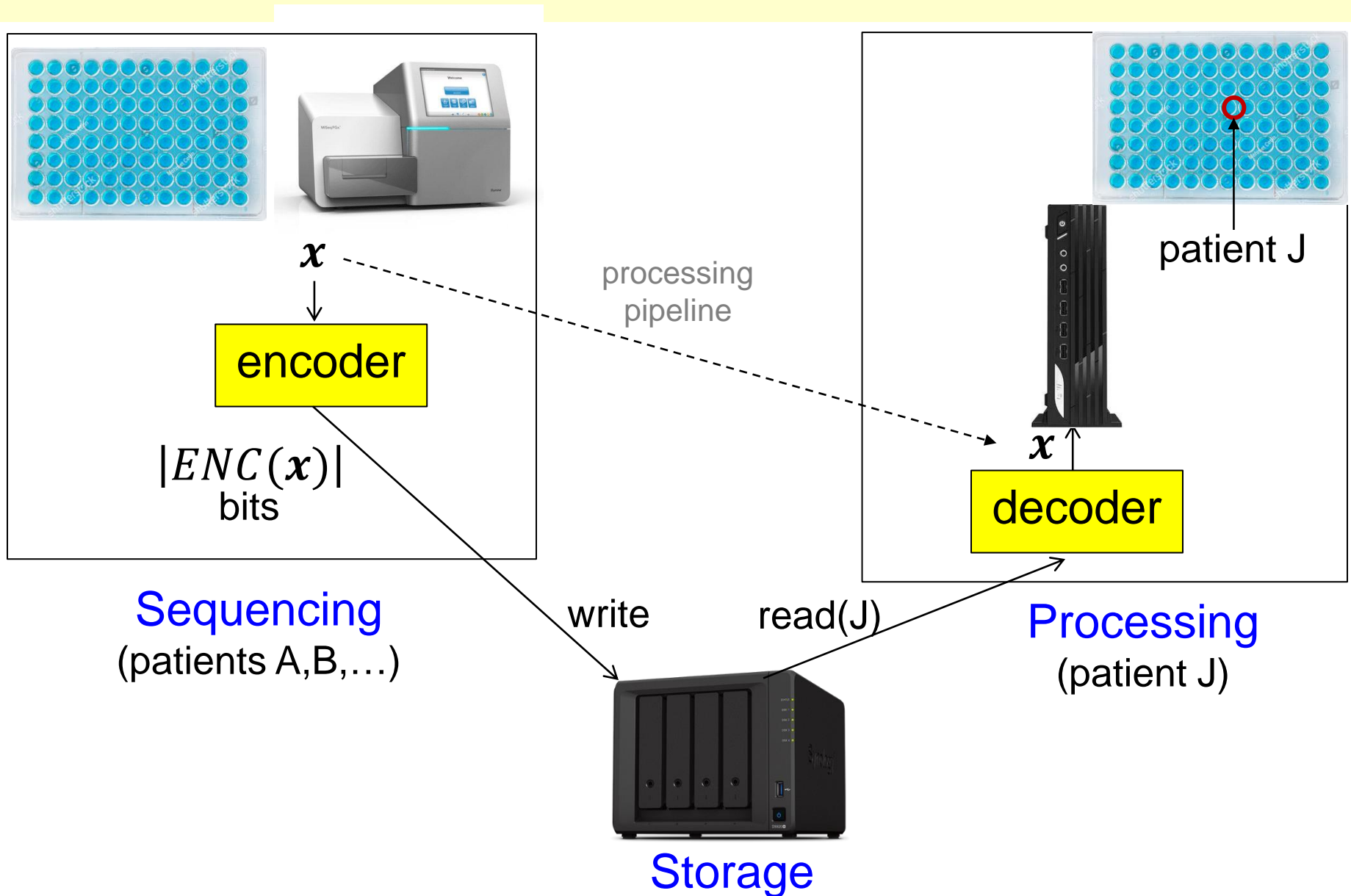


Encoding DNA: communication



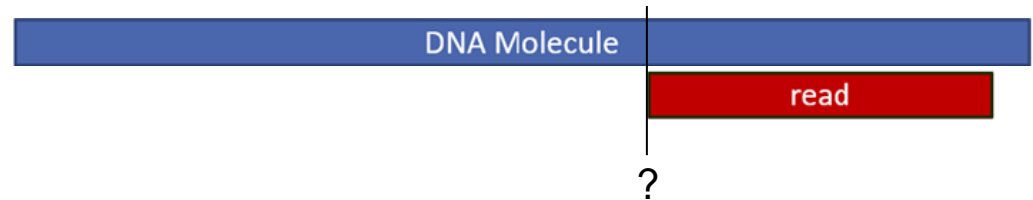
- DNA sampled in **clinic** but stored and processed in **cloud**
- Huge amount of data, compression is essential
 - A single uncompressed genome: ~GB
 - Much bigger due to high read coverage (e.g. x50)

Encoding DNA: storage



The challenge of read compression

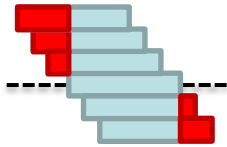
1. Compressing reads individually is too weak, but
2. Encoder does not know read locations within sequence



Genomic read compression: Known methods

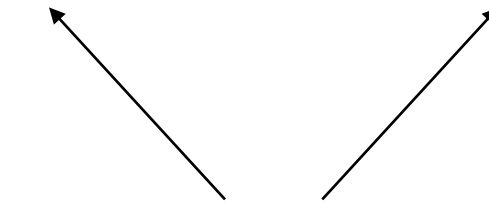
1. Reference-free

Exploit similarities among
the compressed reads



2. Reference-based

Exploit similarities
between the reads and a
reference sequence

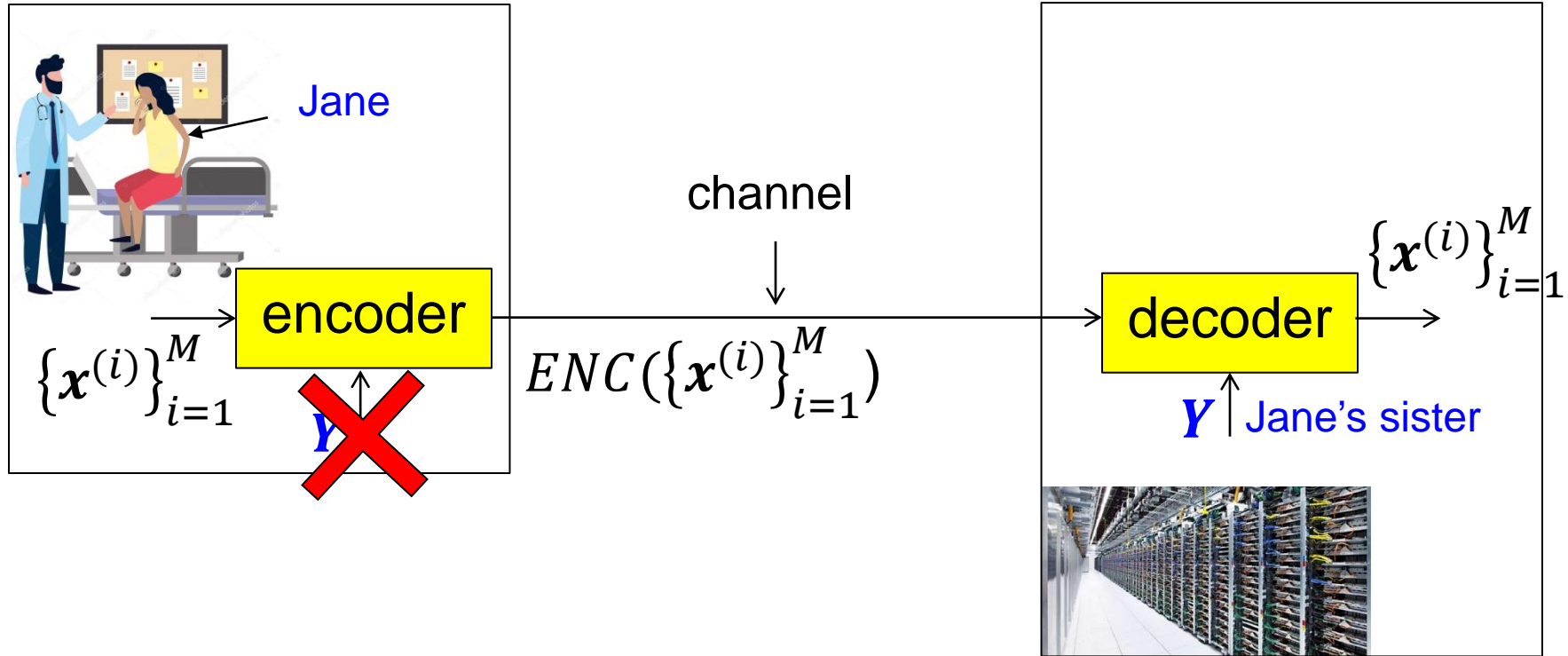


In both methods:

Extremely high encoding complexity!



The proposed approach: Decoder-only reference



- Y contains segments “similar” to x (few differences: substitutions, deletions, insertions)
- Only decoder has Y
 - Limited resources in clinic
 - Privacy

Problem Formulation

Goal: encode $\{\mathbf{x}^{(i)}\}_{i=1}^M$ from \mathbf{X} s.t a decoder with access to \mathbf{Y} will perfectly reconstruct them with high probability.

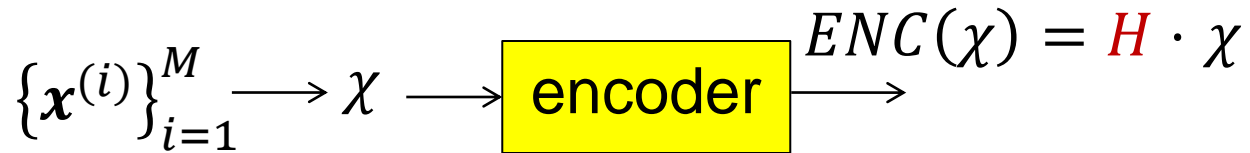
Definition: A **(M, n, \mathcal{R}, P_s) -code** is a pair of encoder-decoder $(\mathcal{E}, \mathcal{D})$ for a set $\{\mathbf{x}^{(i)}\}_{i=1}^M$ of length- n reads, such that:

1. The encoded size is $|\mathcal{E}(\{\mathbf{x}^{(i)}\}_{i=1}^M)| = nM \cdot \mathcal{R}$ (fixed rate)
2. The decoding-success probability satisfies

$$\Pr \left\{ \mathcal{D} \left[\mathcal{E} \left(\{\mathbf{x}^{(i)}\}_{i=1}^M \right), \mathbf{Y} \right] = \{\mathbf{x}^{(i)}\}_{i=1}^M \right\} \geq P_s$$

A practical code construction

batch
of
reads



Multi-layer code construction:

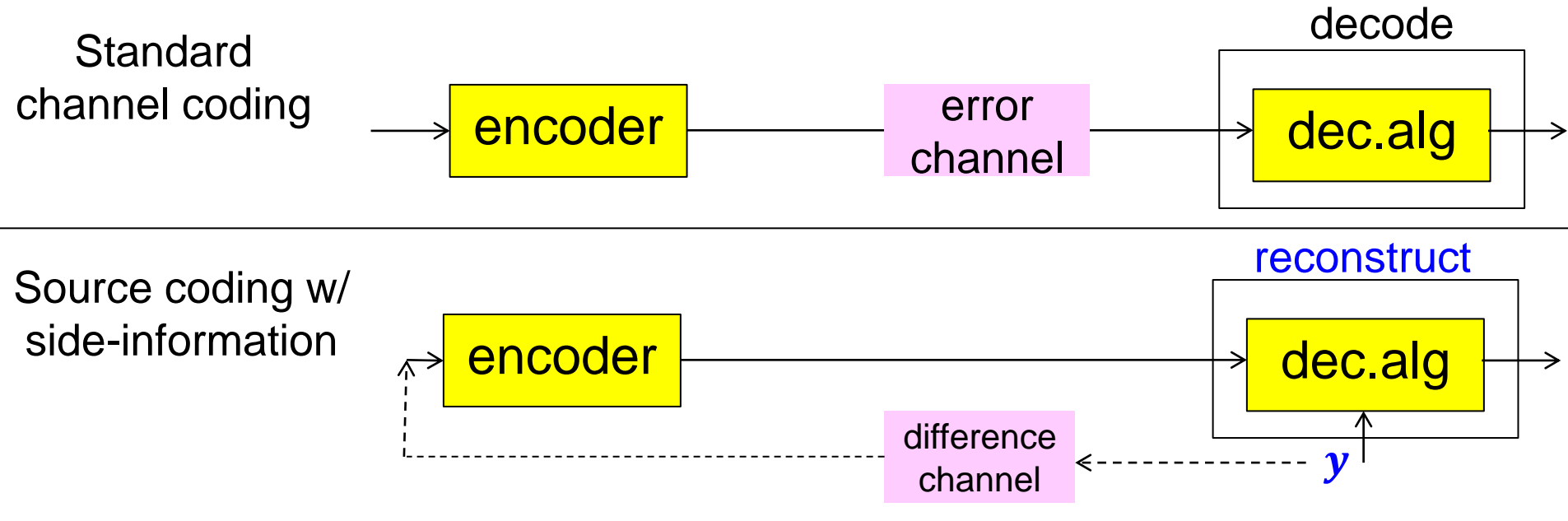
1. Read alignment $H_1 \cdot \mathbf{x}^{(i)}$
2. Read reconstruction $H_2 \cdot \mathbf{x}^{(i)}$
3. Read validation $H_3 \cdot \mathbf{x}^{(i)}$
4. Error/failure correction $H_4 \cdot \chi$

Analysis:

How to find layer parameters to reach success probability P_S with minimal rate.

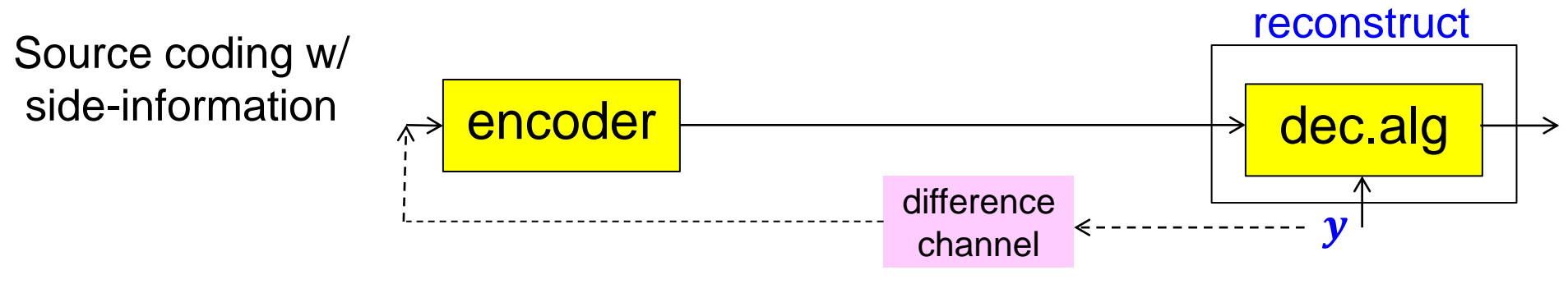
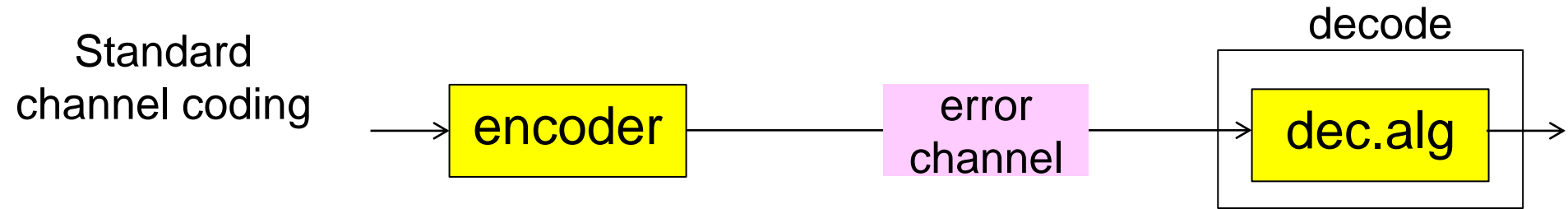
$$H = \begin{array}{|c|c|} \hline H_1 & H_1 \\ \hline H_2 & H_2 \\ \hline H_3 & H_3 \\ \hline \hline & H_4 \\ \hline \end{array} \quad \text{-----}$$

Joint Alignment and Reconstruction

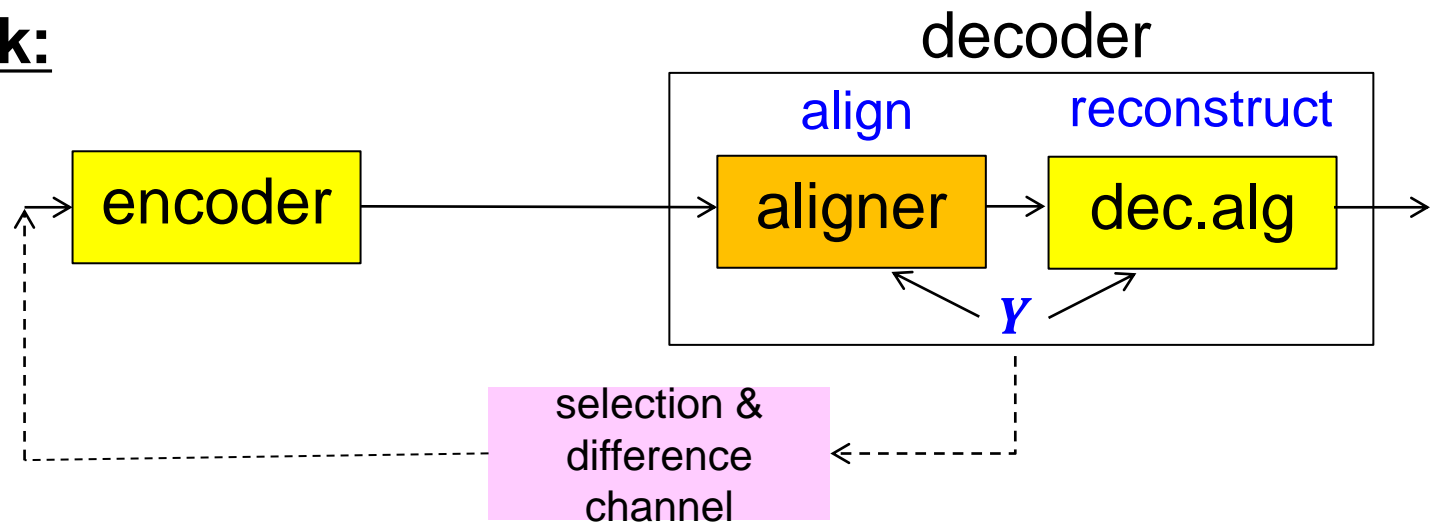


In this talk:

Joint Alignment and Reconstruction



In this talk:

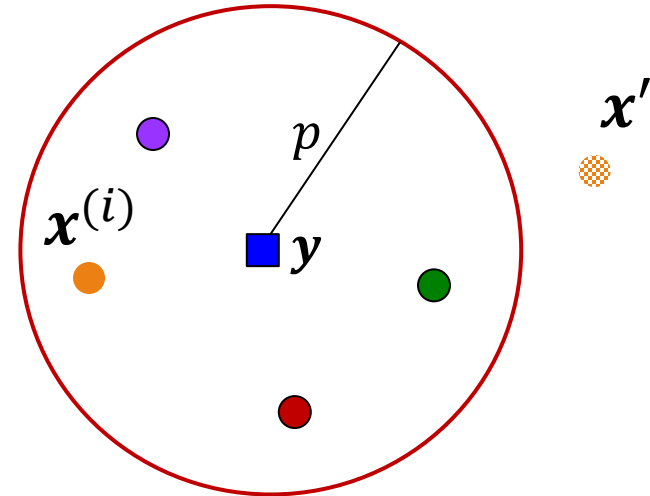


Reconstruct from side information

Inputs:

1) y

2) $\text{synd}(x^{(i)}) = s$ (coset / color)



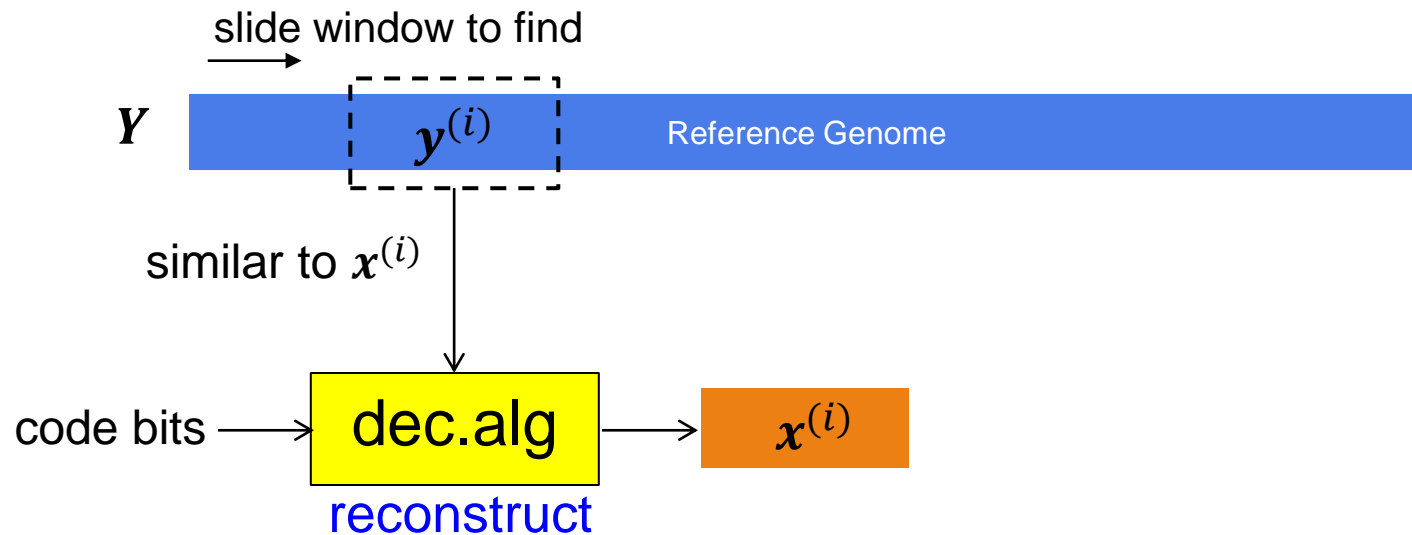
Output:

(Unique) word with color s in ball around y .

Align & Reconstruct @ Decoder

To reconstruct

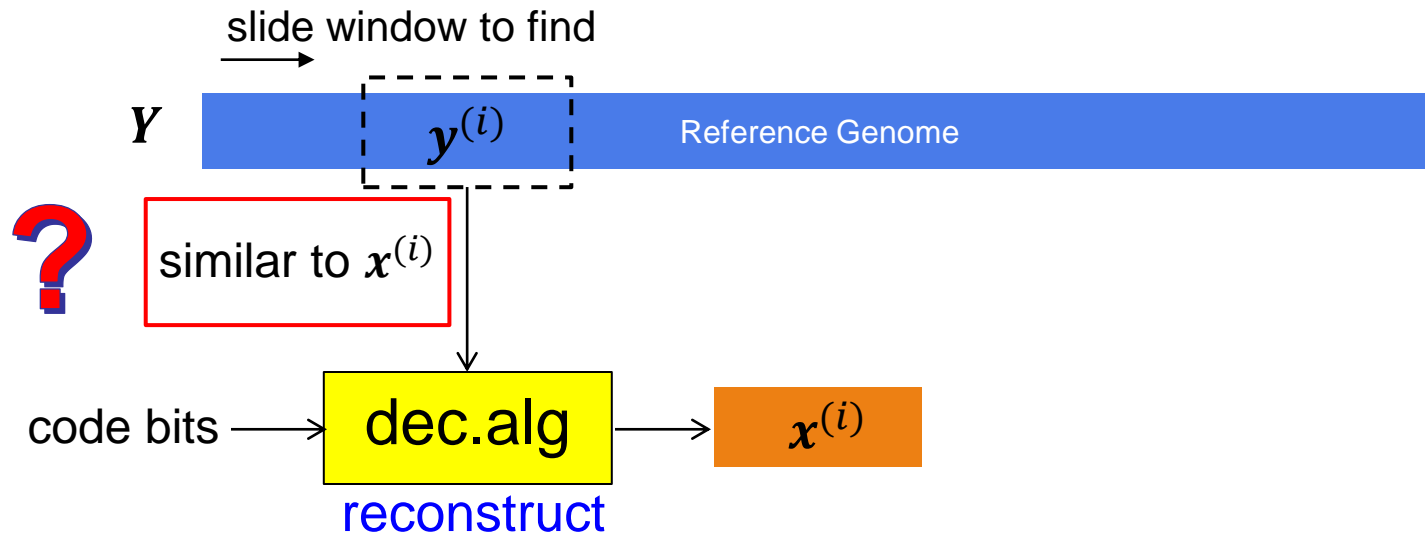
$x^{(i)}$



Align & Reconstruct @ Decoder

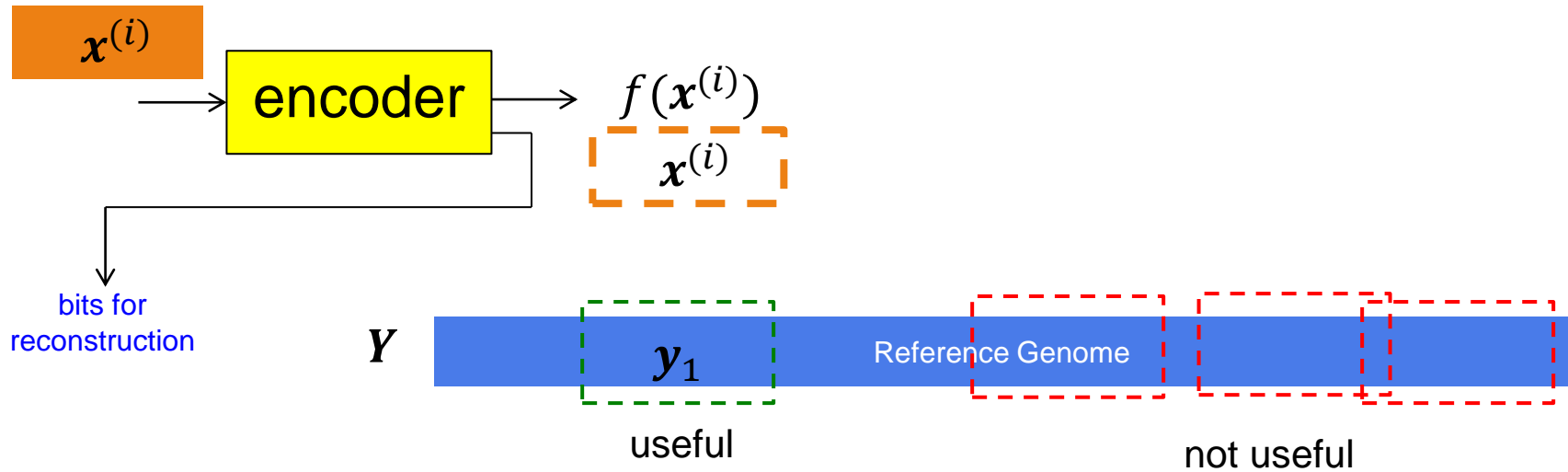
To reconstruct

$x^{(i)}$



Decoder Alignment Problem

Decoder aligns $\mathbf{x}^{(i)}$ to \mathbf{Y} using $f(\mathbf{x}^{(i)})$



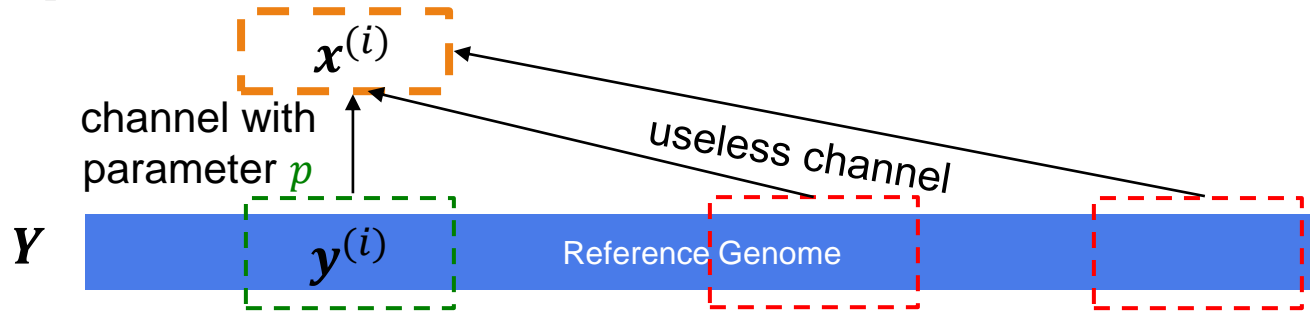
The alignment problem:

Find segments \mathbf{y} that are **likely** useful for reconstructing $\mathbf{x}^{(i)}$

Decoder Alignment Model

Assumption:

Each $\mathbf{x}^{(i)}$ is the output of $\mathbf{y}^{(i)}$ from a **difference channel** with parameter p .



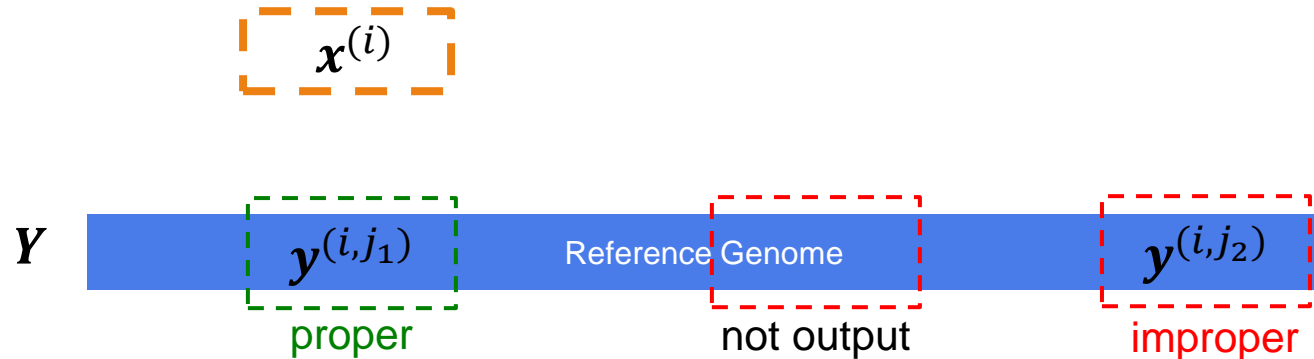
The alignment problem now:

Find $\mathbf{y}^{(i)}$ with high probability, and reject most useless segments.

Proper and Improper Alignments

Alignment operation:

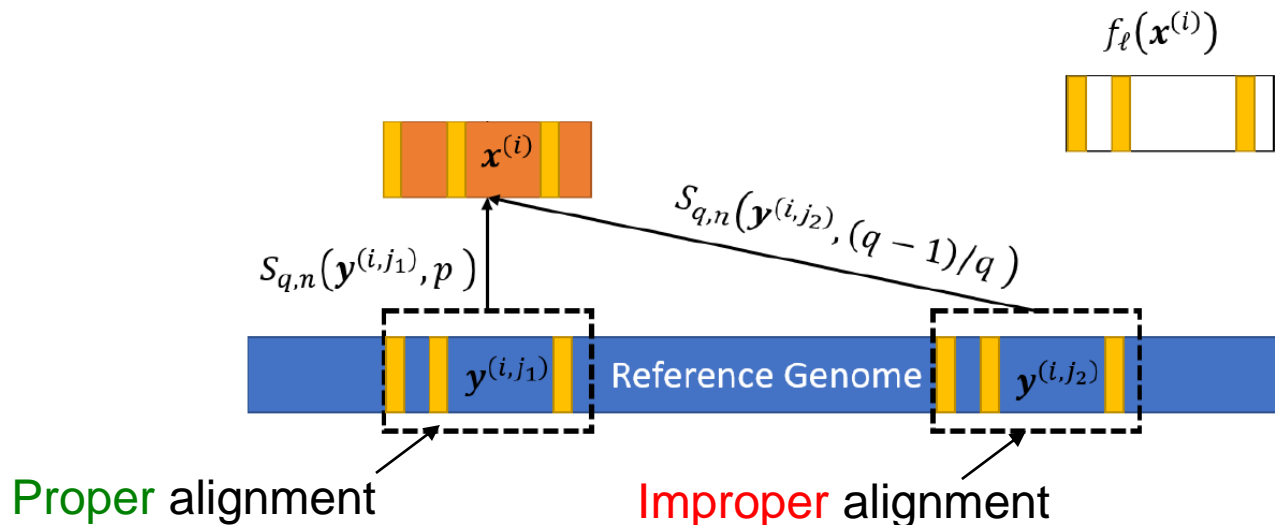
Given $f(\mathbf{x}^{(i)})$, output alignment positions $\mathbf{y}^{(i,1)}, \mathbf{y}^{(i,2)}, \dots, \mathbf{y}^{(i,K_i)}$



- **Proper** alignment: the $\mathbf{y}^{(i,j_1)}$ that equals $\mathbf{y}^{(i)}$ (if found)
- **Improper** alignments: the remaining K_i positions $\mathbf{y}^{(i,j)}$

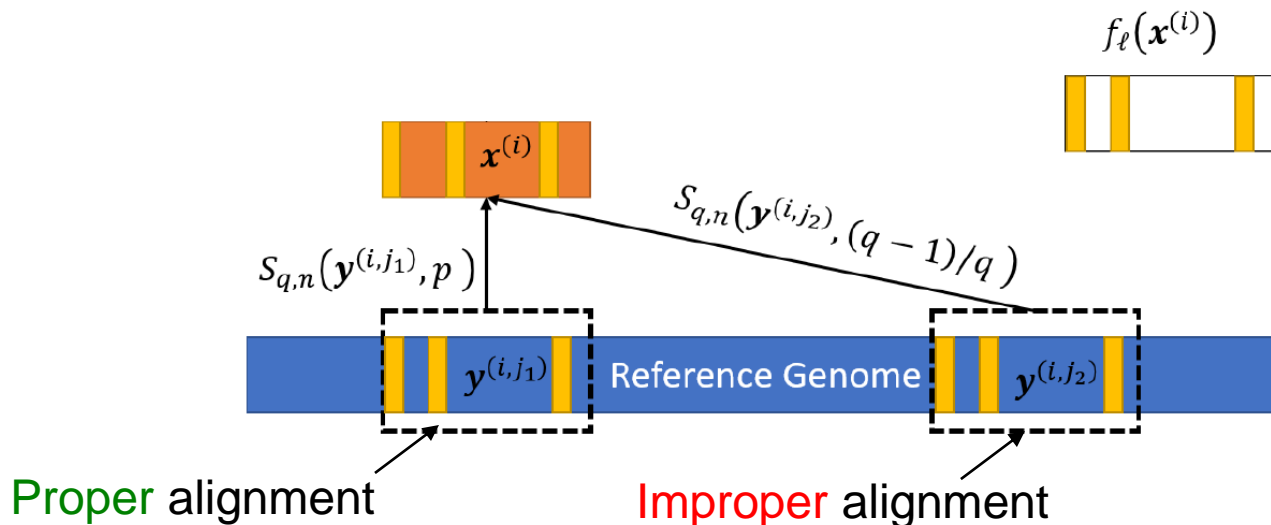
Decoder Alignment – substitution differences

- Substitution differences $\Rightarrow q$ -ary symmetric channels
 - Parameter p for proper alignment
 - Parameter $(q - 1)/q$ (useless) for improper alignments
- Take $f_\ell(\mathbf{x}^{(i)})$ as length- ℓ sample of $\mathbf{x}^{(i)}$
- Output set of candidates $Y^{(i)} = \left\{ \mathbf{y}^{(i,j)} \mid d_H \left(f_\ell(\mathbf{x}^{(i)}), f_\ell(\mathbf{y}^{(i,j)}) \right) \leq \mathbf{T} \right\}_{j=1}^{K_i}$



Decoder Alignment – substitution differences

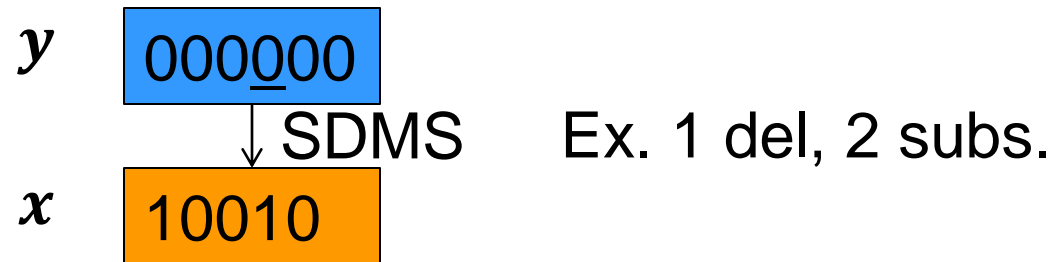
- Substitution differences $\Rightarrow q$ -ary symmetric channels
 - Parameter p for proper alignment
 - Parameter $(q - 1)/q$ (useless) for improper alignments
- Take $f_\ell(\mathbf{x}^{(i)})$ as length- ℓ sample of $\mathbf{x}^{(i)}$
- Output set of candidates $Y^{(i)} = \left\{ \mathbf{y}^{(i,j)} \mid d_H \left(f_\ell(\mathbf{x}^{(i)}), f_\ell(\mathbf{y}^{(i,j)}) \right) \leq T \right\}_{j=1}^{K_i}$



Tradeoff between finding proper and avoiding improper alignments

Difference Model with Deletions

- Model: *single deletion multiple substitutions (SDMS)* per read



- Justification: deletions are rare compared to substitutions
- Q: How to align under SDMS errors?

Alignment Metrics

$q = 2$
full segments x, y

- $d_H(x, y)$ not good anymore. Ex. $d_H(1010, 0101) = 4$.
- Levenshtein distance (Levenshtein, 1965) –
 - Disadvantages: **prohibitive complexity (dynamic programming)**
- Shifted-Hamming distance (Xin et al, 2015) –

$$d_{SH}(x, y) = \sum_{i=1}^n \bigwedge_{j=0}^{r-1} x_i \oplus y_{i+j}$$

- Matching each index with r adjacent indexes (for r deletions)
- Advantage: **linear complexity**
- Disadvantage: **high rate of false alignment**

A New Alignment Distance

$$\mathbf{x} \in \Sigma^n, \mathbf{y} \in \Sigma^{n+1}$$

- Definition: cumulative Hamming distance

$$\phi_j(\mathbf{x}, \mathbf{y}; t) \triangleq \sum_{i=1}^t x_i \oplus y_{i+j}$$

- Definition: differential cumulative Hamming distance

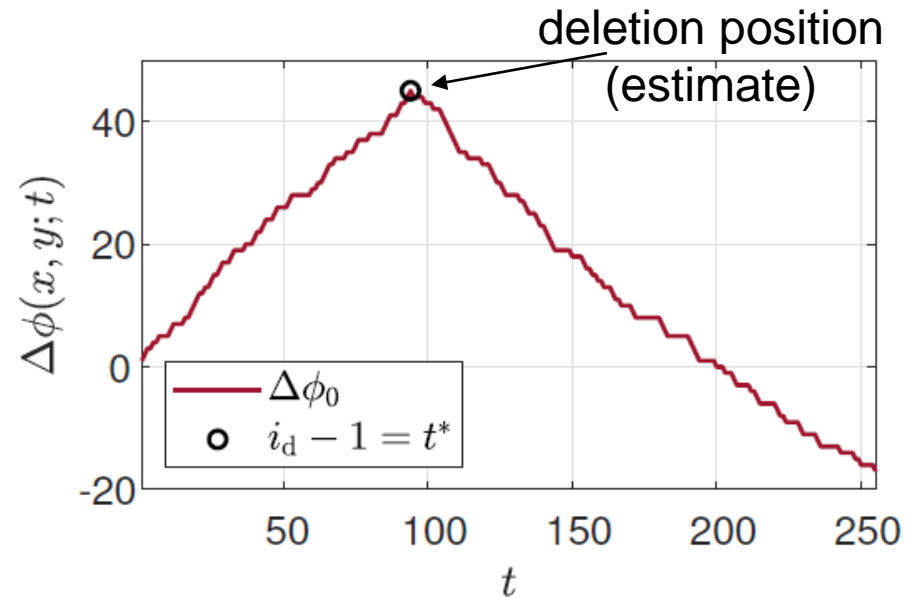
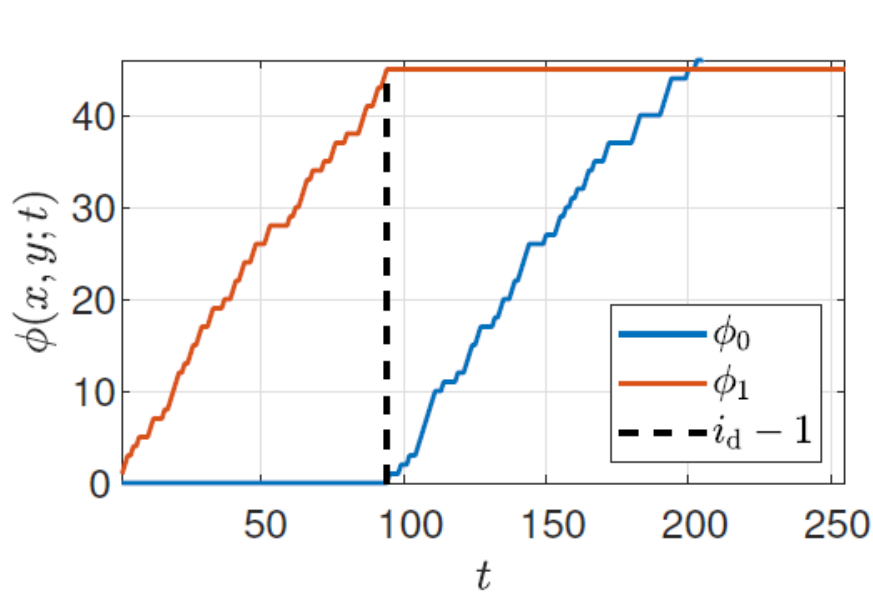
$$\Delta\phi_j(\mathbf{x}, \mathbf{y}; t) = \phi_{j+1}(\mathbf{x}, \mathbf{y}; t) - \phi_j(\mathbf{x}, \mathbf{y}; t)$$

- Definition: *shift-compensating distance (r=1)*

$$d_{s.c}(\mathbf{x}, \mathbf{y}) = \phi_1(\mathbf{x}, \mathbf{y}; n) - \max_{0 \leq t \leq n} \{\Delta\phi_0(\mathbf{x}, \mathbf{y}; t)\}$$

$$= \phi_1(\mathbf{x}, \mathbf{y}; n) - [\phi_1(\mathbf{x}, \mathbf{y}; t^*) - \phi_0(\mathbf{x}, \mathbf{y}; t^*)]$$

S.C Distance: Graphical Example



- $\phi_0(t)$ counts substitutions until index t , and random matches thereafter
- $\phi_1(t)$ counts random matches until index t , and substitutions thereafter
- $d_{s.c}$ counts substitutions while compensating for the partial shift due to deletion

Exact Distance Distribution

Theorem:

Define R.V D_n : S.C distance between random $\mathbf{x} \in \Sigma^n, \mathbf{y} \in \Sigma^{n+1}$

Then,

$$P(D_n = r) = \frac{1}{4^n} \sum_{m=0}^{n-r} \sum_{t=m}^n \sum_{k=0}^{n-m} \sum_{w=0}^k \sum_{l=0}^{t-1} A_1(t, w, m, l) \sum_{v=0}^{n-t-(k-w)} A_2(v, t, k-w, r-l)$$

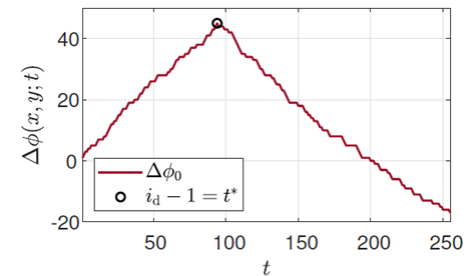
where $A_1(\cdot, \cdot, \cdot, \cdot)$ and $A_2(\cdot, \cdot, \cdot, \cdot)$ are closed-form combinatorial expressions.

Distribution of S.C Distance:

Proof idea

Suppose **unrelated** vectors $\mathbf{x} \in \Sigma^n, \mathbf{y} \in \Sigma^{n+1}$ (independent Bernoulli 1/2)

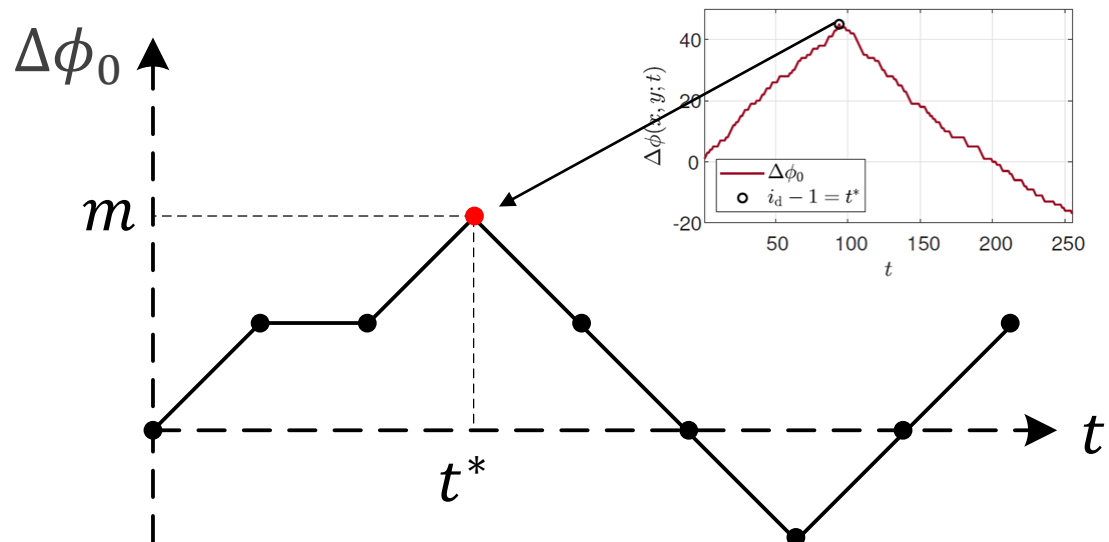
- $\Delta\phi_0(t) = \sum_{i=0}^t \Delta_i, \Delta_0 = 0, \Delta_i \in \{0, \pm 1\}$ w.p $\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right\}$
- Hence: $\Delta\phi_0(t) =$ symmetric **random walk** with null steps



Distribution of S.C Distance: Proof idea

Suppose **unrelated** vectors $\mathbf{x} \in \Sigma^n, \mathbf{y} \in \Sigma^{n+1}$ (independent Bernoulli 1/2)

- $\Delta\phi_0(t) = \sum_{i=0}^t \Delta_i, \Delta_0 = 0, \Delta_i \in \{0, \pm 1\}$ w.p $\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right\}$
- Hence: $\Delta\phi_0(t) =$ symmetric **random walk** with null steps
- Given t^*, m : count R.Ws that attain global max m at time t^*



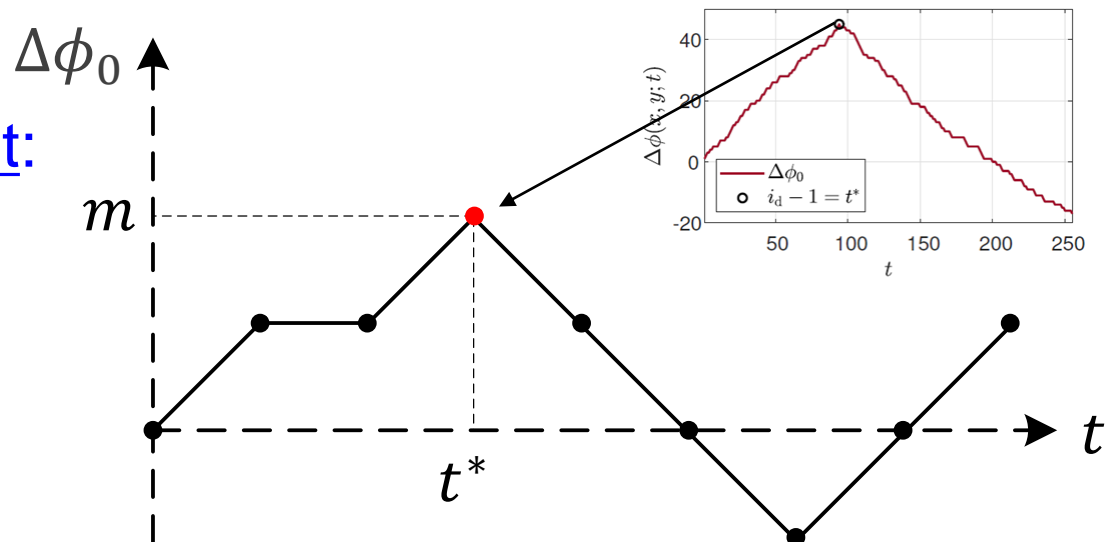
Distribution of S.C Distance: Proof idea

Suppose **unrelated** vectors $\mathbf{x} \in \Sigma^n, \mathbf{y} \in \Sigma^{n+1}$ (independent Bernoulli 1/2)

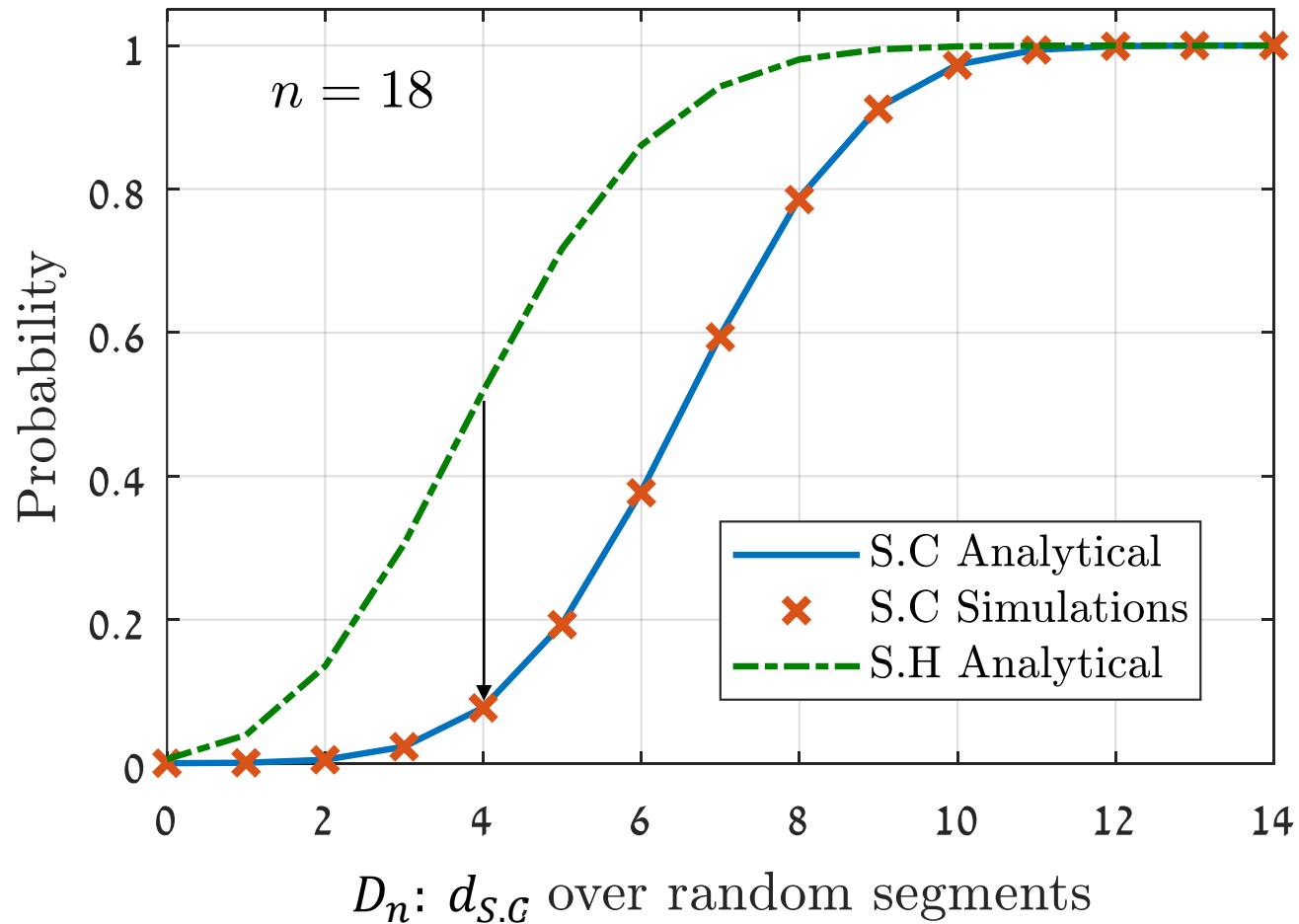
- $\Delta\phi_0(t) = \sum_{i=0}^t \Delta_i, \Delta_0 = 0, \Delta_i \in \{0, \pm 1\}$ w.p $\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right\}$
- Hence: $\Delta\phi_0(t) =$ symmetric **random walk** with null steps
- Given t^*, m : count R.Ws that attain global max m at time t^*

Additional proof ingredient:

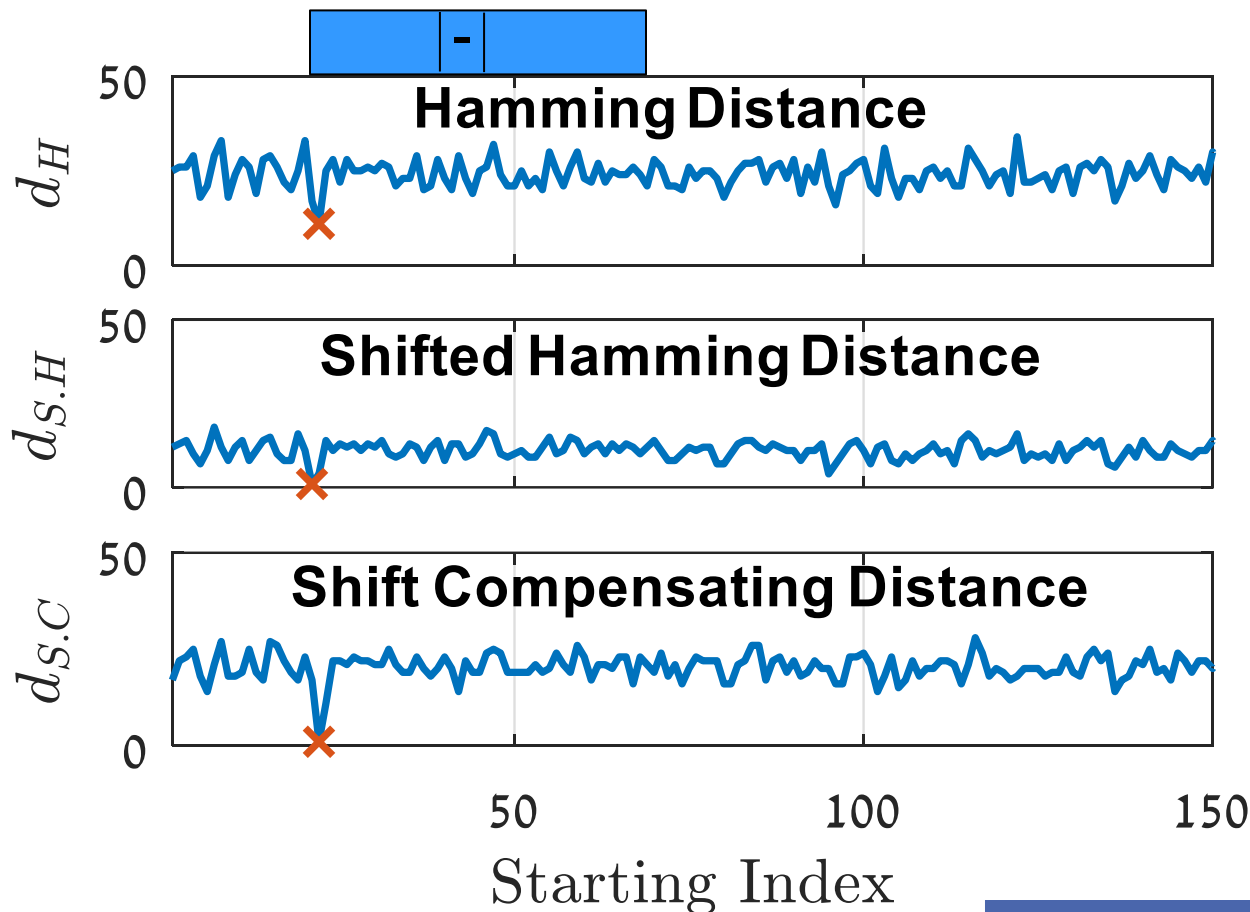
Counting how many \mathbf{x}, \mathbf{y} pairs map to each (m, t) random walk.



S.C-Distance Advantage



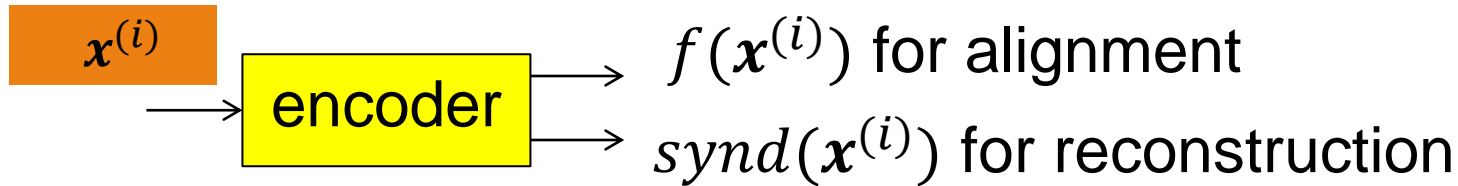
S.C Alignment Advantage



Distance	Random	True
Hamming	High	High
S.H	Low	Low
S.C (ours)	High	Low

Sharing bits between alignment and reconstruction

So far, separate encoded bits for alignment and reconstruction:

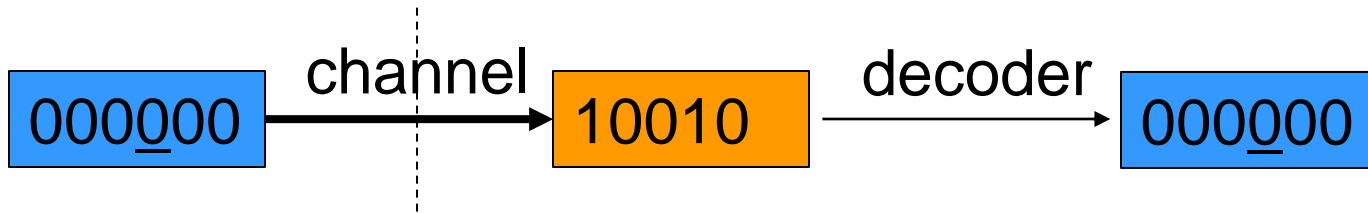


Potential savings:

Use same bits for both alignment and reconstruction.

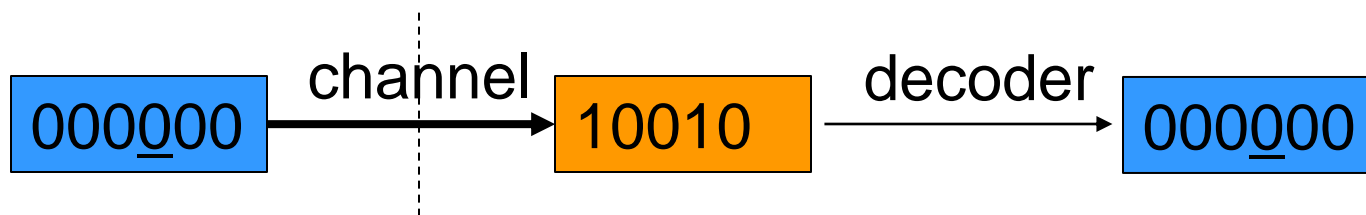
Reconstruction from SDMS Errors

- In channel coding: need SDMS-correcting code

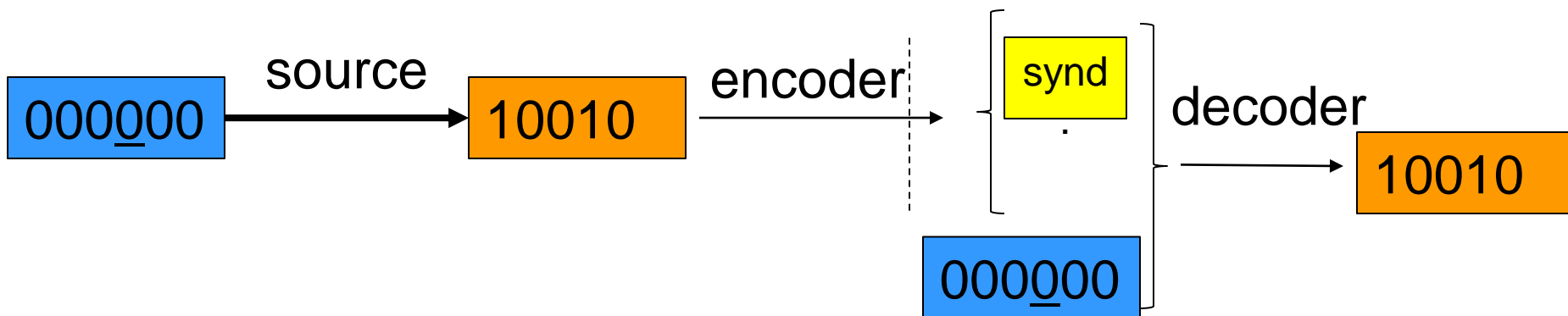


Reconstruction from SDMS Errors

- In channel coding: need SDMS-correcting code

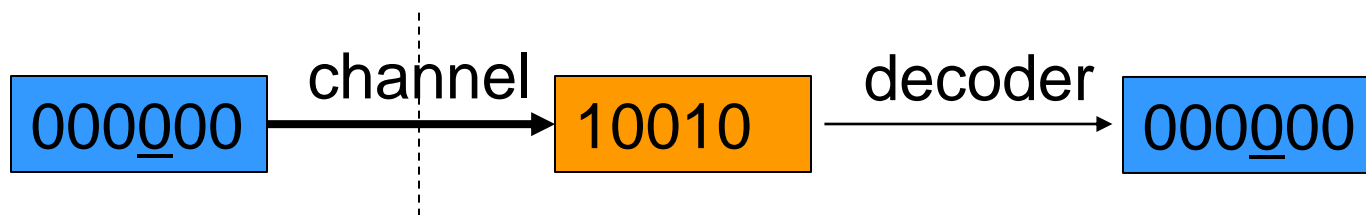


- In source coding:



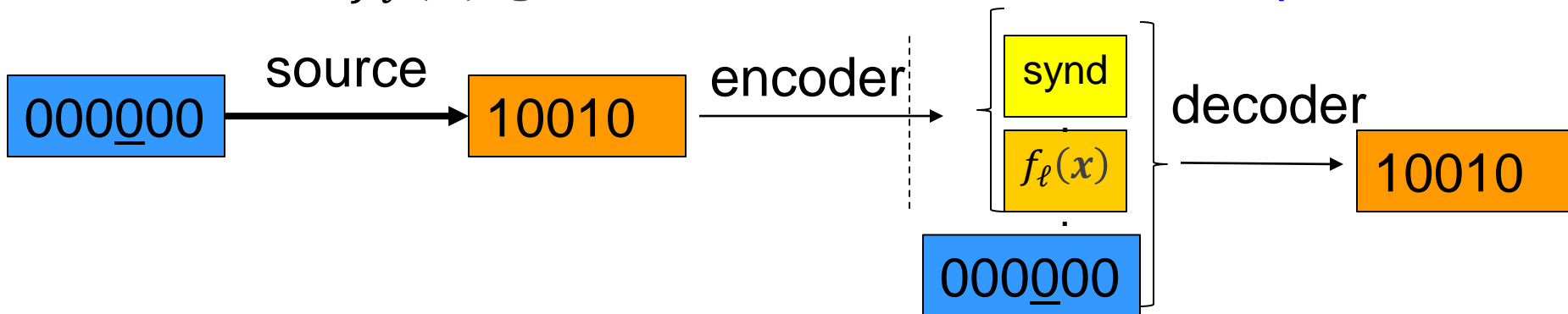
Reconstruction from SDMS Errors

- In channel coding: need SDMS-correcting code



- In source coding:

Observation: $f_\ell(x)$ gives information on deletion position



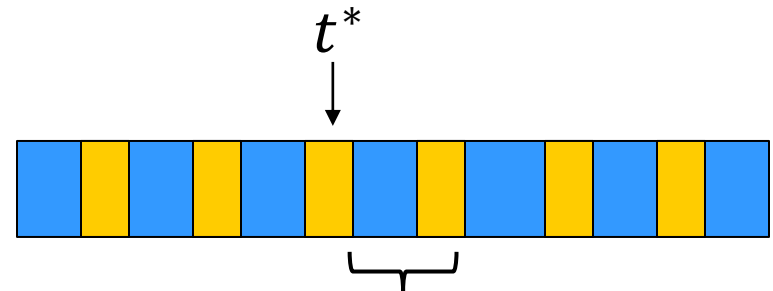
Estimate deletion interval from

$$d_{s.c}(f_\ell(\mathbf{x}^{(i)}), \mathbf{y}^{(i,j)})$$

Can calculate S.C distance on $f_\ell(\cdot)$ instead of full segments

Then, recall from S.C distance:

$$t^* = \operatorname{argmax}_{0 \leq t \leq \ell} \{\Delta \phi'_0(f_\ell(\mathbf{x}), \mathbf{y}); t\}$$

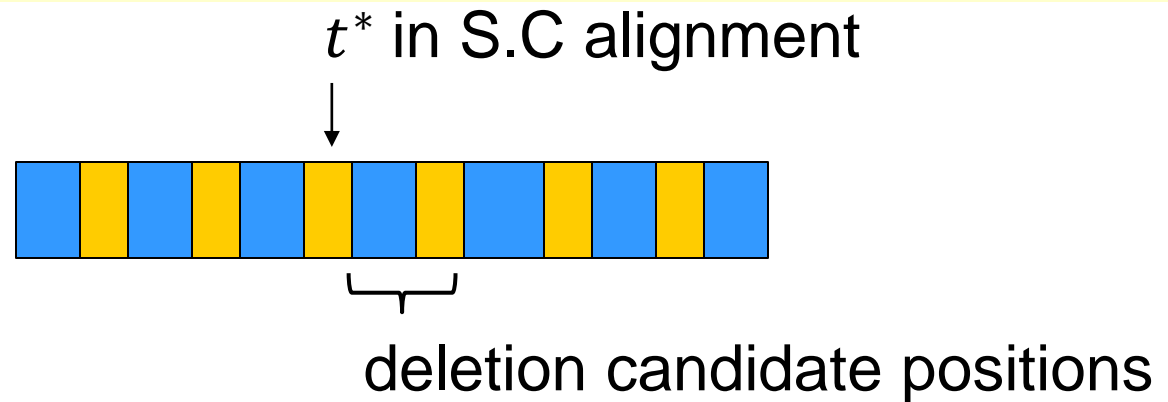


deletion candidate positions

y:	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉
x:	x ₁	x ₂	x ₃	x ₄	x ₆	x ₇	x ₈	x ₉	-
Candidates									
z⁽¹⁾:	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉	-
z⁽²⁾:	x ₁	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉	-
z⁽³⁾:	x ₁	x ₂	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉	-
z⁽⁴⁾:	x ₁	x ₂	x ₃	x ₅	x ₆	x ₇	x ₈	x ₉	-
z⁽⁵⁾:	x ₁	x ₂	x ₃	x ₄	x ₆	x ₇	x ₈	x ₉	-
z⁽⁶⁾:	x ₁	x ₂	x ₃	x ₄	x ₅	x ₇	x ₈	x ₉	-
z⁽⁷⁾:	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₈	x ₉	-
z⁽⁸⁾:	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₉	-
z⁽⁹⁾:	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	-






small ambiguity in deletion position \Rightarrow few added substitutions

SDMS Reconstruction Algorithm



Reconstruction algorithm:

1. Invoke **substitutions decoder** over each word in y 's **deletion-candidate list**
2. Apply a **majority** rule on decoder outputs

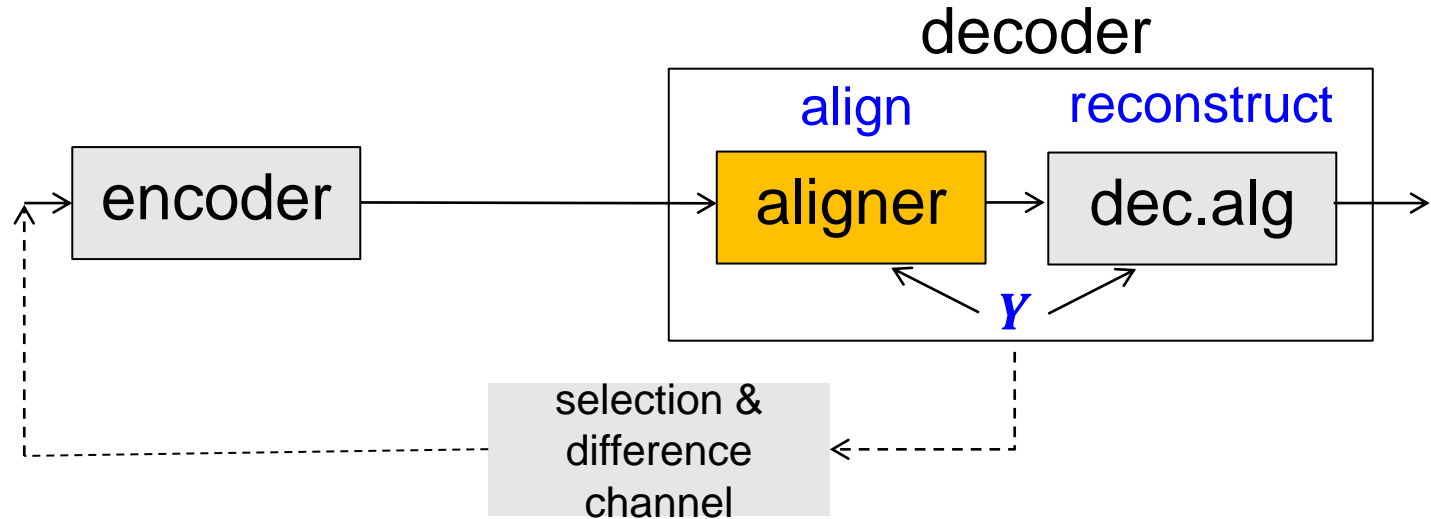
x' 

 x 



y:	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
x:	x_1	x_2	x_3	x_4	x_6	x_7	x_8	x_9	-
Candidates									
$z^{(1)}:$	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	-
$z^{(2)}:$	x_1	x_3	x_4	x_5	x_6	x_7	x_8	x_9	-
$z^{(3)}:$	x_1	x_2	x_4	x_5	x_6	x_7	x_8	x_9	-
$z^{(4)}:$	x_1	x_2	x_3	x_5	x_6	x_7	x_8	x_9	-
$z^{(5)}:$	x_1	x_2	x_3	x_4	x_6	x_7	x_8	x_9	-
$z^{(6)}:$	x_1	x_2	x_3	x_4	x_5	x_7	x_8	x_9	-
$z^{(7)}:$	x_1	x_2	x_3	x_4	x_5	x_6	x_8	x_9	-
$z^{(8)}:$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_9	-
$z^{(9)}:$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	-

Contribution summary

1. A new problem: joint alignment + reconstruction
2. Efficient and flexible coding scheme
3. Alignment algorithms with proven detection capabilities

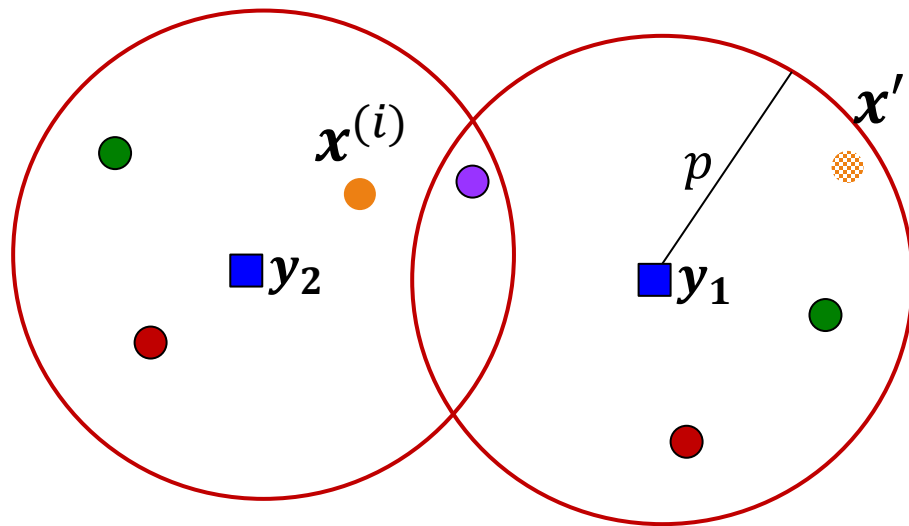
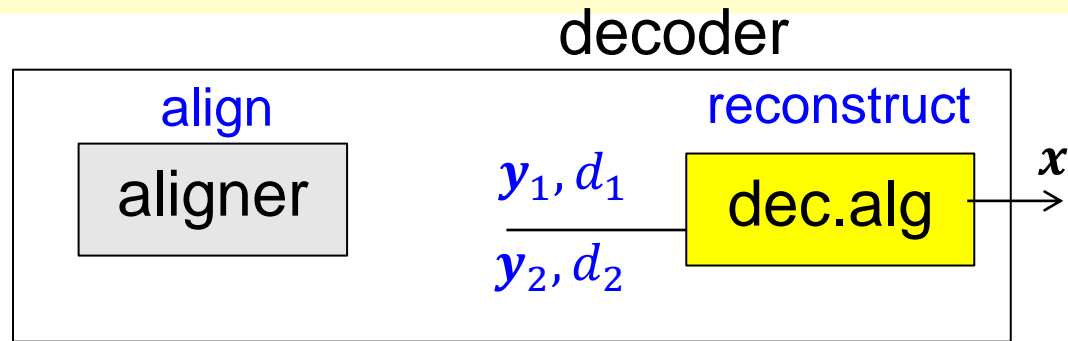
Open directions: **alignment**



S.C distance can be extended to any number of deletions/insertions

- No longer linear time
- But linear-time approximation works well in practice
- Theory?

Open directions : **reconstruction**



Reconstruct a read jointly from multiple alignments

- Develop soft/iterative decoders

To read more

- “Genomic compression with read alignment at the decoder”, JSAIT special issue in memory of Alexander Vardy, 2023.
- “Genomic compression with decoder alignment under single deletion and multiple substitutions”, YG&YC, ISIT2022
- “Distributed source coding of fragmented genomic sequencing data”, YG&YC, ISIT2021