Correcting a Single Deletion in Reads from a Nanopore Sequencer

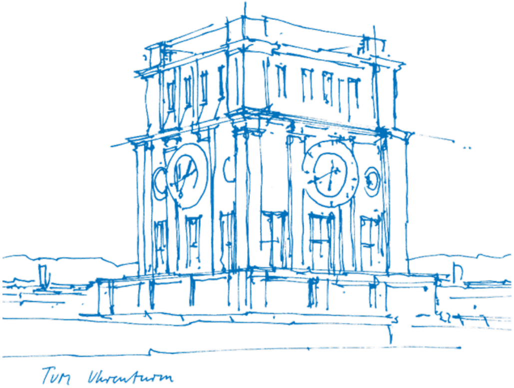
Anisha Banerjee[1]

Joint work with
Yonatan Yehezkeally[1], Antonia Wachter-Zeh[1], and Eitan Yaakobi[2]

[1]Technical University of Munich
Institute for Communications Engineering

[2]Technion – Israel Institute of Technology
Department of Computer Science

July 8, 2024



Outline

Motivation

- Need for dense, reliable, robust storage media
  - Molecular storage paradigms *e.g., DNA storage.*[1]

[1] R. Heckel *et al.*, "A characterization of the DNA data storage channel," *Scientific Reports*, 2019
[2] D. Deamer *et al.*, "Three decades of nanopore sequencing," *Nat. Biotech.*, 2016
[2] A. H. Laszlo *et al.*, "Decoding long nanopore sequencing reads of natural DNA," *Nat. Biotech.*, 2014

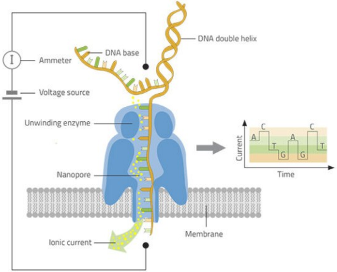
Motivation

- Need for dense, reliable, robust storage media
  - Molecular storage paradigms *e.g., DNA storage.*[1]
- Faster, cheaper sequencers in development.

[1] R. Heckel *et al.*, "A characterization of the DNA data storage channel," *Scientific Reports*, 2019
[2] D. Deamer *et al.*, "Three decades of nanopore sequencing," *Nat. Biotech.*, 2016
[2] A. H. Laszlo *et al.*, "Decoding long nanopore sequencing reads of natural DNA," *Nat. Biotech.*, 2014

Motivation

- Need for dense, reliable, robust storage media
  - ▶ Molecular storage paradigms *e.g., DNA storage.*[1]
- Faster, cheaper sequencers in development.
- Nanopore sequencing[2] :
  - \+ Can read longer DNA strands
  - \+ More portable
  - — High error rates

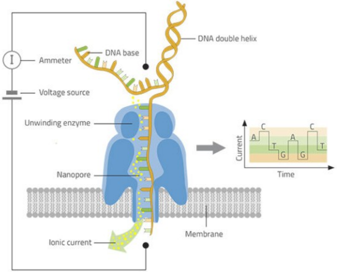


Source: *"Decoding DNA with a pocket-sized sequencer," Science in School.* `https://www.scienceinschool.org/article/2018/decoding-dna-pocket-sized-sequencer/`

[1] R. Heckel *et al.*, "A characterization of the DNA data storage channel," *Scientific Reports*, 2019
[2] D. Deamer *et al.*, "Three decades of nanopore sequencing," *Nat. Biotech.*, 2016
[2] A. H. Laszlo *et al.*, "Decoding long nanopore sequencing reads of natural DNA," *Nat. Biotech.*, 2014

Motivation

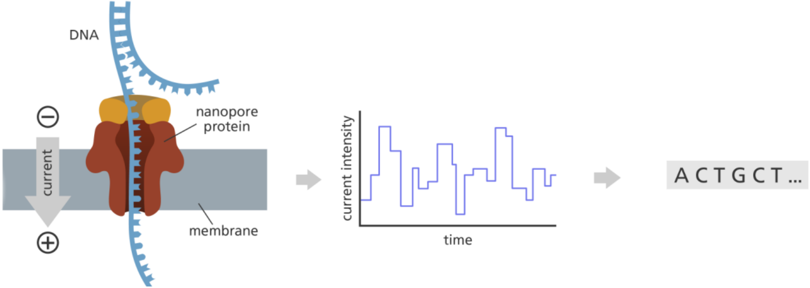
- Need for dense, reliable, robust storage media
  - Molecular storage paradigms *e.g., DNA storage.*[1]
- Faster, cheaper sequencers in development.
- Nanopore sequencing[2] :
  - \+ Can read longer DNA strands
  - \+ More portable
  - – High error rates
- *Aim:* Design coding techniques tailored for nanopore sequencing!



Source: *"Decoding DNA with a pocket-sized sequencer," Science in School.* `https://www.scienceinschool.org/article/2018/decoding-dna-pocket-sized-sequencer/`

[1] R. Heckel *et al.*, "A characterization of the DNA data storage channel," *Scientific Reports*, 2019
[2] D. Deamer *et al.*, "Three decades of nanopore sequencing," *Nat. Biotech.*, 2016
[2] A. H. Laszlo *et al.*, "Decoding long nanopore sequencing reads of natural DNA," *Nat. Biotech.*, 2014

Nanopore Sequencing



Source: *"What is Oxford Nanopore Technology (ONT) sequencing?,"* https://www.yourgenome.org/facts/what-is-oxford-nanopore-technology-ont-sequencing/

- Sources of noise [MDK18] :
  - Nanopore holds $\ell > 1$ nucleotides at a time $\rightarrow$ Intersymbol interference (ISI)!
  - Strand moves irregularly $\rightarrow$ backtracking & skipping.
  - Noisy measurements

[MDK18] W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018

Prior Work

- [MDK18] $\rightarrow$ Mathematical model, capacity bounds.
- [HCW21] $\rightarrow$ Algorithm to compute capacity of an abstracted, deterministic channel model.
- [MVS22] $\rightarrow$ Finite-state semi-Markov channel model with major noise sources.

[MDK18] W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018
[HCW21] R. Hulett *et al.*, "On coding for an abstracted nanopore channel for DNA storage," in *ISIT*, 2021
[MVS22] B. McBain *et al.*, "Finite-state semi-markov channels for nanopore sequencing," in *ISIT*, 2022
[CVVY21] Y. M. Chee *et al.*, "Coding for transverse-reads in domain wall memories," in *ISIT*, 2021
[BYWY24] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," *TIT*, 2024
[YEY24] O. Yerushalmi *et al.*, "The capacity of the weighted read channel," in *ISIT*, 2024
[CIV24] Y. M. Chee *et al.*, "Coding scheme for noisy nanopore sequencing with backtracking and skipping errors," in *ISIT*, 2024

Prior Work

- [MDK18] → Mathematical model, capacity bounds.
- [HCW21] → Algorithm to compute capacity of an abstracted, deterministic channel model.
- [MVS22] → Finite-state semi-Markov channel model with major noise sources.
- [CVVY21] considered a similar channel for racetrack memories:
  → Computed information limits, proposed error-correcting codes.

[MDK18] W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018
[HCW21] R. Hulett *et al.*, "On coding for an abstracted nanopore channel for DNA storage," in *ISIT*, 2021
[MVS22] B. McBain *et al.*, "Finite-state semi-markov channels for nanopore sequencing," in *ISIT*, 2022
[CVVY21] Y. M. Chee *et al.*, "Coding for transverse-reads in domain wall memories," in *ISIT*, 2021
[BYWY24] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," *TIT*, 2024
[YEY24] O. Yerushalmi *et al.*, "The capacity of the weighted read channel," in *ISIT*, 2024
[CIV24] Y. M. Chee *et al.*, "Coding scheme for noisy nanopore sequencing with backtracking and skipping errors," in *ISIT*, 2024

Prior Work

- [MDK18] → Mathematical model, capacity bounds.
- [HCW21] → Algorithm to compute capacity of an abstracted, deterministic channel model.
- [MVS22] → Finite-state semi-Markov channel model with major noise sources.
- [CVVY21] considered a similar channel for racetrack memories:
  → Computed information limits, proposed error-correcting codes.
- [BYWY24] → Optimal single-substitution-correcting code for this channel.

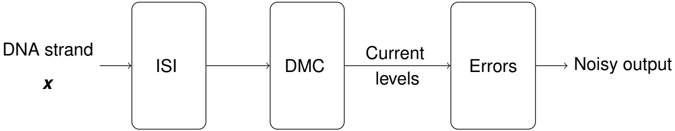
[MDK18] W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018
[HCW21] R. Hulett *et al.*, "On coding for an abstracted nanopore channel for DNA storage," in *ISIT*, 2021
[MVS22] B. McBain *et al.*, "Finite-state semi-markov channels for nanopore sequencing," in *ISIT*, 2022
[CVVY21] Y. M. Chee *et al.*, "Coding for transverse-reads in domain wall memories," in *ISIT*, 2021
[BYWY24] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," *TIT*, 2024
[YEY24] O. Yerushalmi *et al.*, "The capacity of the weighted read channel," in *ISIT*, 2024
[CIV24] Y. M. Chee *et al.*, "Coding scheme for noisy nanopore sequencing with backtracking and skipping errors," in *ISIT*, 2024

Prior Work

- [MDK18] $\rightarrow$ Mathematical model, capacity bounds.
- [HCW21] $\rightarrow$ Algorithm to compute capacity of an abstracted, deterministic channel model.
- [MVS22] $\rightarrow$ Finite-state semi-Markov channel model with major noise sources.
- [CVVY21] considered a similar channel for racetrack memories:
  $\rightarrow$ Computed information limits, proposed error-correcting codes.
- [BYWY24] $\rightarrow$ Optimal single-substitution-correcting code for this channel.
- At ISIT'24 $\rightarrow$ [YEY24] , [CIV24] , ...

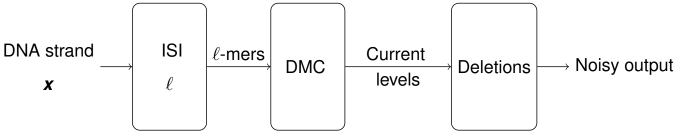
[MDK18] W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018
[HCW21] R. Hulett *et al.*, "On coding for an abstracted nanopore channel for DNA storage," in *ISIT*, 2021
[MVS22] B. McBain *et al.*, "Finite-state semi-markov channels for nanopore sequencing," in *ISIT*, 2022
[CVVY21] Y. M. Chee *et al.*, "Coding for transverse-reads in domain wall memories," in *ISIT*, 2021
[BYWY24] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," *TIT*, 2024
[YEY24] O. Yerushalmi *et al.*, "The capacity of the weighted read channel," in *ISIT*, 2024
[CIV24] Y. M. Chee *et al.*, "Coding scheme for noisy nanopore sequencing with backtracking and skipping errors," in *ISIT*, 2024

Simplified Model



W. Mao *et al.*, “Models and information-theoretic bounds for nanopore sequencing,” *TIT*, 2018

Simplified Model

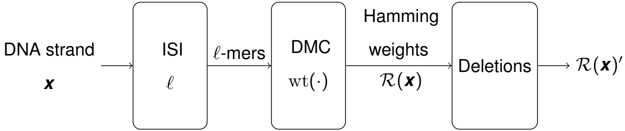


We model

- ISI as length-ℓ-sliding window
- Skipping effects as deletions

W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018

Simplified Model



We model

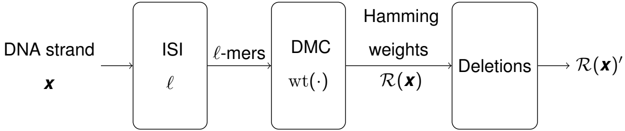
- ISI as length-ℓ-sliding window
- Skipping effects as deletions

Assumptions:

- Input $\boldsymbol{x} \in \{0, 1\}^n$
- DMC maps ℓ-mers to Hamming weights.

W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018

Simplified Model



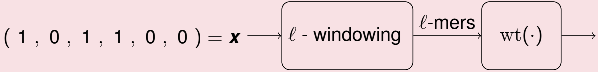
We model

- ISI as length-ℓ-sliding window
- Skipping effects as deletions

Assumptions:

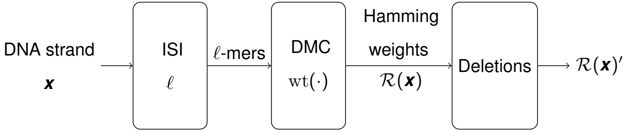
- Input $\boldsymbol{x} \in \{0,1\}^n$
- DMC maps ℓ-mers to Hamming weights.

Example ($\ell = 3$)



W. Mao *et al.*, “Models and information-theoretic bounds for nanopore sequencing,” *TIT*, 2018

Simplified Model



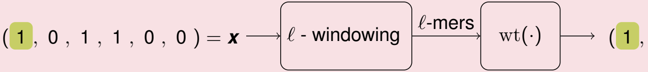
We model

- ISI as length-ℓ-sliding window
- Skipping effects as deletions

Assumptions:

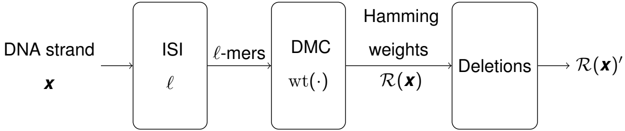
- Input $\boldsymbol{x} \in \{0, 1\}^n$
- DMC maps ℓ-mers to Hamming weights.

Example ($\ell = 3$)



W. Mao *et al.*, “Models and information-theoretic bounds for nanopore sequencing,” *TIT*, 2018

Simplified Model



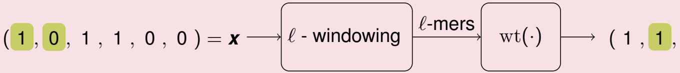
We model

- ISI as length-ℓ-sliding window
- Skipping effects as deletions

Assumptions:

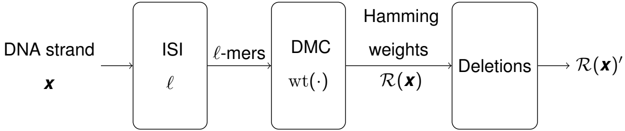
- Input $\boldsymbol{x} \in \{0, 1\}^n$
- DMC maps ℓ-mers to Hamming weights.

Example ($\ell = 3$)



W. Mao *et al.*, “Models and information-theoretic bounds for nanopore sequencing,” *TIT*, 2018

Simplified Model



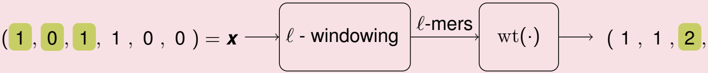
We model

- ISI as length-ℓ-sliding window
- Skipping effects as deletions

Assumptions:

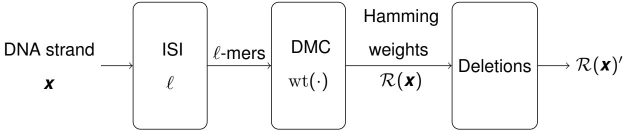
- Input $\boldsymbol{x} \in \{0, 1\}^n$
- DMC maps ℓ-mers to Hamming weights.

Example ($\ell = 3$)



W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018

Simplified Model



We model

- ISI as length-ℓ-sliding window
- Skipping effects as deletions

Assumptions:

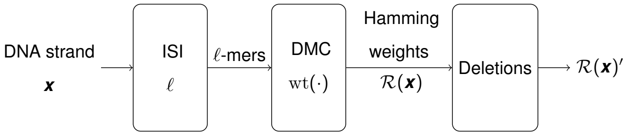
- Input $\boldsymbol{x} \in \{0,1\}^n$
- DMC maps ℓ-mers to Hamming weights.

Example ($\ell = 3$)

$(1, 0, 1, 1, 0, 0) = \boldsymbol{x} \longrightarrow$ $\boxed{\ell\text{ - windowing}}$ $\xrightarrow{\ell\text{-mers}}$ $\boxed{\mathrm{wt}(\cdot)}$ $\longrightarrow (1, 1, 2, 2,$

W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *T-IT*, 2018

Simplified Model



We model

- ISI as length-ℓ-sliding window
- Skipping effects as deletions

Assumptions:

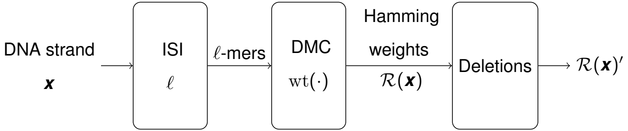
- Input $\boldsymbol{x} \in \{0, 1\}^n$
- DMC maps ℓ-mers to Hamming weights.

Example ($\ell = 3$)

$$(1, 0, 1, 1, 0, 0) = \boldsymbol{x} \longrightarrow \boxed{\ell\text{ - windowing}} \xrightarrow{\ell\text{-mers}} \boxed{\mathrm{wt}(\cdot)} \longrightarrow (1, 1, 2, 2, 2,$$

W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018

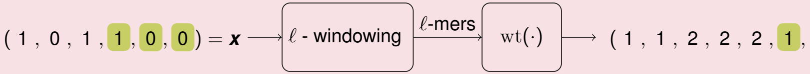
Simplified Model



We model
- ISI as length-ℓ-sliding window
- Skipping effects as deletions

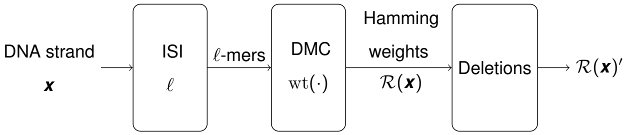
Assumptions:
- Input $\boldsymbol{x} \in \{0, 1\}^n$
- DMC maps ℓ-mers to Hamming weights.

Example ($\ell = 3$)



W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018

Simplified Model



We model

- ISI as length-ℓ-sliding window
- Skipping effects as deletions

Assumptions:

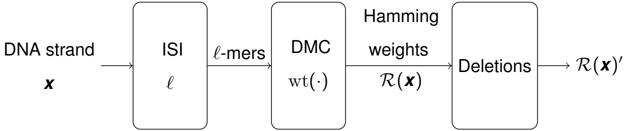
- Input $\boldsymbol{x} \in \{0, 1\}^n$
- DMC maps ℓ-mers to Hamming weights.

Example ($\ell = 3$)

$$(1, 0, 1, 1, 0, 0) = \boldsymbol{x} \longrightarrow \boxed{\ell \text{ - windowing}} \xrightarrow{\ell\text{-mers}} \boxed{\mathrm{wt}(\cdot)} \longrightarrow (1, 1, 2, 2, 2, 1, 0,$$

W. Mao *et al.*, “Models and information-theoretic bounds for nanopore sequencing,” *TIT*, 2018

Simplified Model



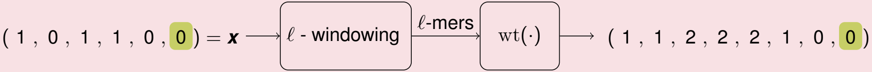
We model

- ISI as length-ℓ-sliding window
- Skipping effects as deletions

Assumptions:

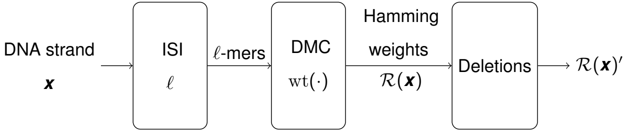
- Input $\boldsymbol{x} \in \{0,1\}^n$
- DMC maps ℓ-mers to Hamming weights.

Example ($\ell = 3$)



W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018

Simplified Model



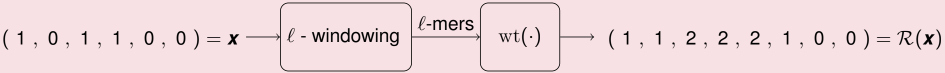
We model

- ISI as length-ℓ-sliding window
- Skipping effects as deletions

Assumptions:

- Input $\boldsymbol{x} \in \{0, 1\}^n$
- DMC maps ℓ-mers to Hamming weights.

Example ($\ell = 3$)



W. Mao *et al.*, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, 2018

Simplified Model

Definition

The ℓ-***read vector*** of any $\boldsymbol{x} \in \Sigma_2^n$ is

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n))$$

where $\mathrm{wt}(\cdot)$ denotes Hamming weight.

Y. M. Chee *et al.*, "Coding for transverse-reads in domain wall memories," in *ISIT*, 2021

Simplified Model

Definition

The ℓ-***read vector*** of any $\boldsymbol{x} \in \Sigma_2^n$ is

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n))$$

where $\mathrm{wt}(\cdot)$ denotes Hamming weight.

Our Results

- $\geq \log n - \ell - o(1)$ min redundancy to correct 1 deletion.

Y. M. Chee *et al.*, "Coding for transverse-reads in domain wall memories," in *ISIT*, 2021

Simplified Model

Definition

The ℓ-***read vector*** of any $\boldsymbol{x} \in \Sigma_2^n$ is

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n))$$

where $\mathrm{wt}(\cdot)$ denotes Hamming weight.

Our Results

- $\geq \log n - \ell - o(1)$ min redundancy to correct 1 deletion.
- 1-deletion correcting code, optimal up to additive constant.

Y. M. Chee *et al.*, "Coding for transverse-reads in domain wall memories," in *ISIT*, 2021

Simplified Model

Definition

The ℓ-***read vector*** of any $\boldsymbol{x} \in \Sigma_2^n$ is

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n))$$

where $\mathrm{wt}(\cdot)$ denotes Hamming weight.

Our Results

- $\geq \log n - \ell - o(1)$ min redundancy to correct 1 deletion.
- 1-deletion correcting code, optimal up to additive constant.
- No redundancy to reconstruct from two noisy ℓ-read vectors.

Y. M. Chee *et al.*, "Coding for transverse-reads in domain wall memories," in *ISIT*, 2021

Error-correcting Code

Definition

$\mathcal{C}$ is a t- **deletion ℓ-read code** if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$,

$$D_t(\mathcal{R}(\boldsymbol{x})) \cap D_t(\mathcal{R}(\boldsymbol{y})) = \emptyset,$$

where $D_t(\cdot)$ is the t-deletion ball.

Error-correcting Code

Definition

$\mathcal{C}$ is a t- **deletion ℓ-read code** if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$,

$$D_t(\mathcal{R}(\boldsymbol{x})) \cap D_t(\mathcal{R}(\boldsymbol{y})) = \emptyset,$$

where $D_t(\cdot)$ is the t-deletion ball.

- *Aim:* Find an optimal 1-deletion ℓ-read code.

Naive Deletion Correction

Definition (Read vector)

The ℓ-*read vector* of any $\boldsymbol{x} \in \Sigma_2^n$ is of length $n + \ell - 1$ and denoted by

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n)).$$

Here $\mathcal{R}(\boldsymbol{x})_i = \mathrm{wt}(\boldsymbol{x}_{i-\ell+1}^i)$ where for $j \notin [n]$, let $x_j = 0$.

Naive Deletion Correction

Definition (Read vector)

The *ℓ-read vector* of any $\boldsymbol{x} \in \Sigma_2^n$ is of length $n + \ell - 1$ and denoted by

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n)).$$

Here $\mathcal{R}(\boldsymbol{x})_i = \mathrm{wt}(\boldsymbol{x}_{i-\ell+1}^i)$ where for $j \notin [n]$, let $x_j = 0$.

- $\mathcal{R}(\boldsymbol{x})_i - \mathcal{R}(\boldsymbol{x})_{i-1} = x_i - x_{i-\ell}$.

Naive Deletion Correction

Definition (Read vector)

The ℓ-*read vector* of any $\boldsymbol{x} \in \Sigma_2^n$ is of length $n + \ell - 1$ and denoted by

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n)).$$

Here $\mathcal{R}(\boldsymbol{x})_i = \mathrm{wt}(\boldsymbol{x}_{i-\ell+1}^i)$ where for $j \notin [n]$, let $x_j = 0$.

- $\mathcal{R}(\boldsymbol{x})_i - \mathcal{R}(\boldsymbol{x})_{i-1} = x_i - x_{i-\ell}$.
- $\boldsymbol{x}$ uniquely determined from the n-prefix of $\mathcal{R}(\boldsymbol{x})$.

Naive Deletion Correction

Definition (Read vector)

The ℓ-*read vector* of any $\boldsymbol{x} \in \Sigma_2^n$ is of length $n + \ell - 1$ and denoted by

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n)).$$

Here $\mathcal{R}(\boldsymbol{x})_i = \mathrm{wt}(\boldsymbol{x}_{i-\ell+1}^i)$ where for $j \notin [n]$, let $x_j = 0$.

- $\mathcal{R}(\boldsymbol{x})_i - \mathcal{R}(\boldsymbol{x})_{i-1} = x_i - x_{i-\ell}$.
- $\boldsymbol{x}$ uniquely determined from the n-prefix of $\mathcal{R}(\boldsymbol{x})$ mod 2.

Naive Deletion Correction

Definition (Read vector)

The ℓ-*read vector* of any $\boldsymbol{x} \in \Sigma_2^n$ is of length $n + \ell - 1$ and denoted by

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n)).$$

Here $\mathcal{R}(\boldsymbol{x})_i = \mathrm{wt}(\boldsymbol{x}_{i-\ell+1}^i)$ where for $j \notin [n]$, let $x_j = 0$.

- $\mathcal{R}(\boldsymbol{x})_i - \mathcal{R}(\boldsymbol{x})_{i-1} = x_i - x_{i-\ell}$.
- $\boldsymbol{x}$ uniquely determined from the n-prefix of $\mathcal{R}(\boldsymbol{x})$ mod 2. $\leftarrow$ Use VT code!

Naive Deletion Correction

Definition (Read vector)

The ℓ-*read vector* of any $\boldsymbol{x} \in \Sigma_2^n$ is of length $n+\ell-1$ and denoted by

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n)).$$

Here $\mathcal{R}(\boldsymbol{x})_i = \mathrm{wt}(\boldsymbol{x}_{i-\ell+1}^i)$ where for $j \notin [n]$, let $x_j = 0$.

- $\mathcal{R}(\boldsymbol{x})_i - \mathcal{R}(\boldsymbol{x})_{i-1} = x_i - x_{i-\ell}$.
- $\boldsymbol{x}$ uniquely determined from the n-prefix of $\mathcal{R}(\boldsymbol{x})$ mod 2. $\leftarrow$ Use VT code!

Construction

$$\mathcal{C}(n,\ell,a) = \{\boldsymbol{x} \in \Sigma_2^n : \sum_{i=1}^n i(\mathcal{R}(\boldsymbol{x})_i \bmod 2) = a \pmod{n+1}\},$$

where $a \in \{0, \ldots, n\}$.

Naive Deletion Correction

Definition (Read vector)

The ℓ-*read vector* of any $\boldsymbol{x} \in \Sigma_2^n$ is of length $n + \ell - 1$ and denoted by

$$\mathcal{R}(\boldsymbol{x}) = (\mathrm{wt}(x_1), \mathrm{wt}(\boldsymbol{x}_1^2), \ldots, \mathrm{wt}(\boldsymbol{x}_{n-1}^n), \mathrm{wt}(x_n)).$$

Here $\mathcal{R}(\boldsymbol{x})_i = \mathrm{wt}(\boldsymbol{x}_{i-\ell+1}^i)$ where for $j \notin [n]$, let $x_j = 0$.

- $\mathcal{R}(\boldsymbol{x})_i - \mathcal{R}(\boldsymbol{x})_{i-1} = x_i - x_{i-\ell}$.
- $\boldsymbol{x}$ uniquely determined from the n-prefix of $\mathcal{R}(\boldsymbol{x})$ mod 2. $\leftarrow$ Use VT code!

Construction

$$\mathcal{C}(n, \ell, a) = \{\boldsymbol{x} \in \Sigma_2^n : \sum_{i=1}^{n} i(\mathcal{R}(\boldsymbol{x})_i \bmod 2) = a \pmod{n+1}\},$$

where $a \in \{0, \ldots, n\}$.

$\rightarrow$ Requires $\log_2(n+1)$ redundant bits.

Connection to Sticky Deletion

- **ℓ-sticky deletion** $\rightarrow$ delete in a run with $\geq \ell$ bits.

Connection to Sticky Deletion

- **ℓ-sticky deletion** $\rightarrow$ delete in a run with $\geq \ell$ bits.

Example ($\ell = 3$)

$$\boldsymbol{x} = (1, 0, \underline{1, 1, 1, 1}, 0, 0, 0)$$

$$\mathcal{R}(\boldsymbol{x}) = (1, 1, 2, 2, \boxed{3, 3}, 2, 1, 0, 0, 0)$$

Connection to Sticky Deletion

- **ℓ-sticky deletion** $\to$ delete in a run with $\geq \ell$ bits.

Example ($\ell = 3$)

$$\boldsymbol{x} = (1, 0, \underline{1, 1, 1, 1}, 0, 0, 0) \xrightarrow[\text{deletion}]{\ell-\text{sticky}} \boldsymbol{x}' = (1, 0, \underline{1, 1, 1}, 0, 0, 0)$$

$$\mathcal{R}(\boldsymbol{x}) = (1, 1, 2, 2, \boxed{3, 3}, 2, 1, 0, 0, 0) \xrightarrow{\ell\text{-deletion}} \mathcal{R}(\boldsymbol{x})' = (1, 1, 2, 2, \boxed{3}, 2, 1, 0, 0, 0)$$

Connection to Sticky Deletion

- **ℓ-sticky deletion** $\rightarrow$ delete in a run with $\geq \ell$ bits.

Example ($\ell = 3$)

$$\boldsymbol{x} = (1, 0, 1, 1, 1, 1, \underline{0, 0, 0}) \xrightarrow[\text{deletion}]{\ell-\text{sticky}} \boldsymbol{x}' = (1, 0, 1, 1, 1, 1, \underline{0, 0})$$

$$\mathcal{R}(\boldsymbol{x}) = (1, 1, 2, 2, 3, 3, 2, 1, \boxed{0, 0, 0}) \xrightarrow{\text{0-deletion}} \mathcal{R}(\boldsymbol{x})' = (1, 1, 2, 2, 3, 3, 2, 1, \boxed{0, 0})$$

Connection to Sticky Deletion

- **ℓ-sticky deletion** $\rightarrow$ delete in a run with $\geq \ell$ bits.

Example ($\ell = 3$)

$$\boldsymbol{x} = (1,0,1,1,1,1,\underline{0,0,0}) \xrightarrow[\text{deletion}]{\ell-\text{sticky}} \boldsymbol{x}' = (1,0,1,1,1,1,\underline{0,0})$$

$$\mathcal{R}(\boldsymbol{x}) = (1,1,2,2,3,3,2,1,\boxed{0,0,0}) \xrightarrow{\text{0-deletion}} \mathcal{R}(\boldsymbol{x})' = (1,1,2,2,3,3,2,1,\boxed{0,0})$$

- ℓ-Sticky deletion in $\boldsymbol{x} \rightarrow 0/\ell$-deletion in $\mathcal{R}(\boldsymbol{x})$!
  - A 1-deletion ℓ-read code also corrects an ℓ-sticky deletion in message.

Minimum Redundancy

Lemma

Any single-deletion ℓ-read code is also a single-ℓ-sticky-deletion-correcting code.

[FVY15] A. Fazeli *et al.*, “Generalized sphere packing bound,” *TIT*, 2015

Minimum Redundancy

Lemma

Any single-deletion ℓ-read code is also a single-ℓ-sticky-deletion-correcting code.

From [FVY15] and concentration arguments,

Theorem

The redundancy of any single-ℓ-sticky-deletion-correcting code is bounded from below by

$$\log_2 n - \ell - o(1).$$

[FVY15] A. Fazeli *et al.*, "Generalized sphere packing bound," *TIT*, 2015

Minimum Redundancy

Lemma

Any single-deletion ℓ-read code is also a single-ℓ-sticky-deletion-correcting code.

From [FVY15] and concentration arguments,

Theorem

The redundancy of any single-deletion ℓ-read code is bounded from below by

$$\log_2 n - \ell - o(1).$$

[FVY15] A. Fazeli *et al.*, "Generalized sphere packing bound," *TIT*, 2015

Minimum Redundancy

Lemma

Any single-deletion ℓ-read code is also a single-ℓ-sticky-deletion-correcting code.

From [FVY15] and concentration arguments,

Theorem

The redundancy of any single-deletion ℓ-read code is bounded from below by

$$\log_2 n - \ell - o(1).$$

$\rightarrow$ Naive VT-like construction optimal up to additive constant!

[FVY15] A. Fazeli *et al.*, "Generalized sphere packing bound," *TIT*, 2015

Comparison with Classical Channel

- ‘Classical’ $\rightarrow \ell = 1$.
- ‘Nanopore’ $\rightarrow \ell \geq 2$.

Comparison with Classical Channel

- ‘Classical’ $\rightarrow \ell = 1$.
- ‘Nanopore’ $\rightarrow \ell \geq 2$.

	Classical	**Nanopore**
1 substitution	$\log n$	[1] $\log \log n$
1 deletion	$\log n$	$\log n - \ell$

[1] A. Banerjee *et al.*, “Error-correcting codes for nanopore sequencing,” *TIT*, 2024

Comparison with Classical Channel

- 'Classical' $\rightarrow \ell = 1$.
- 'Nanopore' $\rightarrow \ell \geq 2$.

	Classical	**Nanopore**
1 substitution	$\log n$	[1] $\log \log n$
1 deletion	$\log n$	$\log n - \ell$

🥲

[1] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," *TIT*, 2024

Comparison with Classical Channel

- 'Classical' $\rightarrow \ell = 1$.
- 'Nanopore' $\rightarrow \ell \geq 2$.

	Classical	**Nanopore**
1 substitution	$\log n$	[1] $\log \log n$
1 deletion	$\log n$	$\log n - \ell$
1 deletion, 2 reads	[2] $\log \log n$	?

[1] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," *TIT*, 2024
[2] J. Chrisnata *et al.*, "Correcting deletions with multiple reads," *TIT*, 2022

Comparison with Classical Channel

- 'Classical' $\rightarrow \ell = 1$.
- 'Nanopore' $\rightarrow \ell \geq 2$.

	Classical	Nanopore
1 substitution	$\log n$	[1] $\log \log n$
1 deletion	$\log n$	$\log n - \ell$
1 deletion, 2 reads	[2] $\log \log n$	0

🥲

😁

[1] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," *TIT*, 2024
[2] J. Chrisnata *et al.*, "Correcting deletions with multiple reads," *TIT*, 2022

Correction with Two Reads

Lemma

When $\ell \geq 2$, for any two distinct $\boldsymbol{x}, \boldsymbol{y} \in \Sigma_2^n$,

$$|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| \leq 1.$$

[L01] V. Levenshtein, “Efficient reconstruction of sequences,” *TIT*, 2001
[BYWY24] A. Banerjee *et al.*, “Error-correcting codes for nanopore sequencing,” *TIT*, 2024
[CMNY22] K. Cai *et al.*, “Coding for sequence reconstruction for single edits,” *TIT*, 2022

Correction with Two Reads

Lemma

When $\ell \geq 2$, for any two distinct $\boldsymbol{x}, \boldsymbol{y} \in \Sigma_2^n$,

$$|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| \leq 1.$$

Proof sketch:

- [L01] $\rightarrow$ Distinct $\boldsymbol{x}, \boldsymbol{y}$ have $|D_1(\boldsymbol{x}) \cap D_1(\boldsymbol{y})| \leq 2.$

[L01] V. Levenshtein, "Efficient reconstruction of sequences," *TIT*, 2001
[BYWY24] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," *TIT*, 2024
[CMNY22] K. Cai *et al.*, "Coding for sequence reconstruction for single edits," *TIT*, 2022

Correction with Two Reads

Lemma

When $\ell \geq 2$, for any two distinct $\boldsymbol{x}, \boldsymbol{y} \in \Sigma_2^n$,

$$|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| \leq 1.$$

Proof sketch:

- [L01] $\rightarrow$ Distinct $\boldsymbol{x}, \boldsymbol{y}$ have $|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| \leq 2$.

[L01] V. Levenshtein, “Efficient reconstruction of sequences,” *TIT*, 2001
[BYWY24] A. Banerjee *et al.*, “Error-correcting codes for nanopore sequencing,” *TIT*, 2024
[CMNY22] K. Cai *et al.*, “Coding for sequence reconstruction for single edits,” *TIT*, 2022

Correction with Two Reads

Lemma

When $\ell \geq 2$, for any two distinct $\boldsymbol{x}, \boldsymbol{y} \in \Sigma_2^n$,

$$|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| \leq 1.$$

Proof sketch:

- [L01] $\rightarrow$ Distinct $\boldsymbol{x}$, $\boldsymbol{y}$ have $|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| \leq 2$.
- [BYWY24] $\rightarrow$ For $\ell \geq 2$, distinct $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy $d_H(\mathcal{R}(\boldsymbol{x}), \mathcal{R}(\boldsymbol{y})) \geq 2$.

[L01] V. Levenshtein, "Efficient reconstruction of sequences," *TIT*, 2001
[BYWY24] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," *TIT*, 2024
[CMNY22] K. Cai *et al.*, "Coding for sequence reconstruction for single edits," *TIT*, 2022

Correction with Two Reads

Lemma

When $\ell \geq 2$, for any two distinct $\boldsymbol{x}, \boldsymbol{y} \in \Sigma_2^n$,

$$|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| \leq 1.$$

Proof sketch:

- [L01] $\rightarrow$ Distinct $\boldsymbol{x}, \boldsymbol{y}$ have $|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| \leq 2$.
- [BYWY24] $\rightarrow$ For $\ell \geq 2$, distinct $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy $d_H(\mathcal{R}(\boldsymbol{x}), \mathcal{R}(\boldsymbol{y})) \geq 2$.
- [CMNY22] $\rightarrow$ $|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| = 2$ holds iff

$$\begin{aligned} \mathcal{R}(\boldsymbol{x}) &= (\quad \boldsymbol{a} \quad \alpha\beta\alpha\beta\ldots\alpha\beta \quad \boldsymbol{b} \quad) \\ \mathcal{R}(\boldsymbol{y}) &= (\quad \boldsymbol{a} \quad \beta\alpha\beta\alpha\ldots\beta\alpha \quad \boldsymbol{b} \quad) \end{aligned}$$

[L01] V. Levenshtein, "Efficient reconstruction of sequences," *TIT*, 2001
[BYWY24] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," *TIT*, 2024
[CMNY22] K. Cai *et al.*, "Coding for sequence reconstruction for single edits," *TIT*, 2022

Correction with Two Reads

Lemma

When $\ell \geq 2$, for any two distinct $\boldsymbol{x}, \boldsymbol{y} \in \Sigma_2^n$,

$$|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| \leq 1.$$

Proof sketch:

- [L01] $\rightarrow$ Distinct $\boldsymbol{x}, \boldsymbol{y}$ have $|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| \leq 2$.
- [BYWY24] $\rightarrow$ For $\ell \geq 2$, distinct $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy $d_H(\mathcal{R}(\boldsymbol{x}), \mathcal{R}(\boldsymbol{y})) \geq 2$.
- [CMNY22] $\rightarrow$ $|D_1(\mathcal{R}(\boldsymbol{x})) \cap D_1(\mathcal{R}(\boldsymbol{y}))| = 2$ holds iff

$$\begin{aligned} \mathcal{R}(\boldsymbol{x}) &= (\quad \boldsymbol{a} \quad \alpha\beta\alpha\beta\ldots\alpha\beta \quad \boldsymbol{b} \quad) \\ \mathcal{R}(\boldsymbol{y}) &= (\quad \boldsymbol{a} \quad \beta\alpha\beta\alpha\ldots\beta\alpha \quad \boldsymbol{b} \quad) \end{aligned}$$

$\rightarrow$ Never occurs!

[L01] V. Levenshtein, “Efficient reconstruction of sequences,” *TIT*, 2001
[BYWY24] A. Banerjee *et al.*, “Error-correcting codes for nanopore sequencing,” *TIT*, 2024
[CMNY22] K. Cai *et al.*, “Coding for sequence reconstruction for single edits,” *TIT*, 2022

Reconstruction from Two Reads

- Can reconstruct from two noisy reads with no additional redundancy!

[BYWY23] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," in *ISIT*, 2023

Reconstruction from Two Reads

- Can reconstruct from two noisy reads with no additional redundancy!

Algorithm: Reconstruct

Input: n, ℓ, set $\{\mathcal{R}', \mathfrak{R}'\} \subseteq D_1(\mathcal{R}(\boldsymbol{x}))$ for some $\boldsymbol{x} \in \Sigma_2^n$
Output: $\mathcal{R}(\boldsymbol{x})$

[BYWY23] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," in *ISIT*, 2023

Reconstruction from Two Reads

- Can reconstruct from two noisy reads with no additional redundancy!

Algorithm: Reconstruct

Input: n, ℓ, set $\{\mathcal{R}', \mathfrak{R}'\} \subseteq D_1(\mathcal{R}(\boldsymbol{x}))$ for some $\boldsymbol{x} \in \Sigma_2^n$
Output: $\mathcal{R}(\boldsymbol{x})$

init

    Let i and j be the first and last indices at which $\mathcal{R}'$ and $\mathfrak{R}'$ disagree.
    $\widehat{\mathcal{R}}(\boldsymbol{x}) \leftarrow (\mathcal{R}'_1, \ldots, \mathcal{R}'_{i-1}, \mathfrak{R}'_i, \mathcal{R}'_i, \ldots, \mathcal{R}'_{n+\ell-2})$;
    $\widetilde{\mathcal{R}}(\boldsymbol{x}) \leftarrow (\mathcal{R}'_1, \ldots, \mathcal{R}'_j, \mathfrak{R}'_j, \mathcal{R}'_{j+1}, \ldots, \mathcal{R}'_{n+\ell-2})$.

[BYWY23] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," in *ISIT*, 2023

Reconstruction from Two Reads

- Can reconstruct from two noisy reads with no additional redundancy!

Algorithm: Reconstruct

Input: n, ℓ, set $\{\mathcal{R}', \mathfrak{R}'\} \subseteq D_1(\mathcal{R}(\boldsymbol{x}))$ for some $\boldsymbol{x} \in \Sigma_2^n$

Output: $\mathcal{R}(\boldsymbol{x})$

init

Let i and j be the first and last indices at which $\mathcal{R}'$ and $\mathfrak{R}'$ disagree.

$\widehat{\mathcal{R}}(\boldsymbol{x}) \leftarrow (\mathcal{R}'_1, \ldots, \mathcal{R}'_{i-1}, \mathfrak{R}'_i, \mathcal{R}'_i, \ldots, \mathcal{R}'_{n+\ell-2})$;

$\widetilde{\mathcal{R}}(\boldsymbol{x}) \leftarrow (\mathcal{R}'_1, \ldots, \mathcal{R}'_j, \mathfrak{R}'_j, \mathcal{R}'_{j+1}, \ldots, \mathcal{R}'_{n+\ell-2})$.

if $\widehat{\mathcal{R}}(\boldsymbol{x})$ *is the ℓ-read vector of any vector in* Σ_2^n **then**

$\mathcal{R}(\boldsymbol{x}) \leftarrow \widehat{\mathcal{R}}(\boldsymbol{x})$.

- Efficient verification process [BYWY23] .

[BYWY23] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," in *ISIT*, 2023

Reconstruction from Two Reads

- Can reconstruct from two noisy reads with no additional redundancy!

Algorithm: Reconstruct

Input: n, ℓ, set $\{\mathcal{R}', \mathfrak{R}'\} \subseteq D_1(\mathcal{R}(\boldsymbol{x}))$ for some $\boldsymbol{x} \in \Sigma_2^n$
Output: $\mathcal{R}(\boldsymbol{x})$

```
init
    Let i and j be the first and last indices at which R' and ℜ' disagree.
    R̂(x) ← (R'_1, ..., R'_{i−1}, ℜ'_i, R'_i, ..., R'_{n+ℓ−2});
    R̃(x) ← (R'_1, ..., R'_j, ℜ'_j, R'_{j+1}, ..., R'_{n+ℓ−2}).
if R̂(x) is the ℓ-read vector of any vector in Σ_2^n then
    R(x) ← R̂(x).
else
    R(x) ← R̃(x).
```

Where: line 2: Let i and j be the first and last indices at which $\mathcal{R}'$ and $\mathfrak{R}'$ disagree; line 3: $\widehat{\mathcal{R}}(\boldsymbol{x}) \leftarrow (\mathcal{R}'_1, \ldots, \mathcal{R}'_{i-1}, \mathfrak{R}'_i, \mathcal{R}'_i, \ldots, \mathcal{R}'_{n+\ell-2})$; line 4: $\widetilde{\mathcal{R}}(\boldsymbol{x}) \leftarrow (\mathcal{R}'_1, \ldots, \mathcal{R}'_j, \mathfrak{R}'_j, \mathcal{R}'_{j+1}, \ldots, \mathcal{R}'_{n+\ell-2})$; line 5: **if** $\widehat{\mathcal{R}}(\boldsymbol{x})$ *is the ℓ-read vector of any vector in* Σ_2^n **then** $\mathcal{R}(\boldsymbol{x}) \leftarrow \widehat{\mathcal{R}}(\boldsymbol{x})$ **else** $\mathcal{R}(\boldsymbol{x}) \leftarrow \widetilde{\mathcal{R}}(\boldsymbol{x})$.

- Efficient verification process [BYWY23] .

[BYWY23] A. Banerjee *et al.*, "Error-correcting codes for nanopore sequencing," in *ISIT*, 2023

Introduction

Channel Model

Minimum Redundancy

Multiple Reads

Conclusion

Summary

Results

For a simplified model of nanopore sequencing, we show

- $\log n - \ell - o(1)$ min redundancy needed to correct 1 deletion.
- Explicit construction, optimal up to additive constant.
- $\ell \geq 2 \rightarrow$No redundancy to recover from two noisy reads.

Summary

Results

For a simplified model of nanopore sequencing, we show

- $\log n - \ell - o(1)$ min redundancy needed to correct 1 deletion.
- Explicit construction, optimal up to additive constant.
- $\ell \geq 2 \rightarrow$No redundancy to recover from two noisy reads.

Future work

- Multiple deletions & combination with substitutions.
- Levenshtein's sequence reconstruction for ℓ-read vectors.

Summary

Results

For a simplified model of nanopore sequencing, we show

- $\log n - \ell - o(1)$ min redundancy needed to correct 1 deletion.
- Explicit construction, optimal up to additive constant.
- $\ell \geq 2 \rightarrow$No redundancy to recover from two noisy reads.

Future work

- Multiple deletions & combination with substitutions.
- Levenshtein's sequence reconstruction for ℓ-read vectors.

Thank you!

References I

[1] R. Heckel, G. Mikutis, and R. N. Grass, "A characterization of the DNA data storage channel," *Scientific Reports*, vol. 9, no. 9663, Jul. 2019.

[2] D. Deamer, M. Akeson, and D. Branton, "Three decades of nanopore sequencing," *Nat. Biotech.*, vol. 34, no. 5, pp. 518–524, May 2016.

[3] A. H. Laszlo *et al.*, "Decoding long nanopore sequencing reads of natural DNA," *Nat. Biotech.*, vol. 32, no. 8, pp. 829–833, Aug. 2014.

[4] W. Mao, S. N. Diggavi, and S. Kannan, "Models and information-theoretic bounds for nanopore sequencing," *TIT*, vol. 64, no. 4, pp. 3216–3236, Apr. 2018.

[5] R. Hulett, S. Chandak, and M. Wootters, "On coding for an abstracted nanopore channel for DNA storage," in *ISIT*, Jul. 2021, pp. 2465–2470.

[6] B. McBain, E. Viterbo, and J. Saunderson, "Finite-state semi-markov channels for nanopore sequencing," in *ISIT*, Jun. 2022, pp. 216–221.

[7] Y. M. Chee, A. Vardy, V. K. Vu, and E. Yaakobi, "Coding for transverse-reads in domain wall memories," in *ISIT*, Jul. 2021, pp. 2924–2929.

References II

[8] A. Banerjee, Y. Yehezkeally, A. Wachter-Zeh, and E. Yaakobi, "Error-correcting codes for nanopore sequencing," *TIT*, 2024.

[9] O. Yerushalmi, T. Etzion, and E. Yaakobi, "The capacity of the weighted read channel," in *ISIT*, 2024, (arXiv preprint arXiv:2401.15368).

[10] Y. M. Chee, K. A. Schouhamer Immink, and V. K. Vu, "Coding scheme for noisy nanopore sequencing with backtracking and skipping errors," in *ISIT*, 2024.

[11] A. Fazeli, A. Vardy, and E. Yaakobi, "Generalized sphere packing bound," *TIT*, vol. 61, no. 5, pp. 2313–2334, May 2015.

[12] J. Chrisnata, H. M. Kiah, and E. Yaakobi, "Correcting deletions with multiple reads," *TIT*, vol. 68, no. 11, pp. 7141–7158, Nov. 2022.

[13] V. Levenshtein, "Efficient reconstruction of sequences," *TIT*, vol. 47, no. 1, pp. 2–22, Jan. 2001.

[14] K. Cai, H. M. Kiah, T. T. Nguyen, and E. Yaakobi, "Coding for sequence reconstruction for single edits," *TIT*, vol. 68, no. 1, pp. 66–79, Jan. 2022.

[15] A. Banerjee, Y. Yehezkeally, A. Wachter-Zeh, and E. Yaakobi, "Error-correcting codes for nanopore sequencing," in *ISIT*, Jun. 2023, pp. 364–369.