An Information Theory for Out-of-Order Information: DNA Data Storage and Beyond

Ilan Shomorony
University of Illinois at Urbana-Champaign

Stanford ISL Colloquium – 03/16/23
The wonders of high-throughput sequencing

Genome sequencing

RNA-seq

Metagenomics

DNA storage

high-throughput sequencer

- No location/ordering information!
- Reordering problem based on:
  - Reference (side information)
  - Overlaps
  - Coding
The wonders of high-throughput sequencing

Outline:
1. Information-theoretic limits of out-of-order channels
2. Efficiently finding overlaps

- No location/ordering information!
Why store data in DNA?

- Good candidate for archival storage

Longevity

High information density

Grass et al. 2015
Can we study the capacity of this channel?

Distinctive property: out-of-order (shuffled) observations
Noisy Shuffling Channel

shuffling

noisy channel
Example: BSC Shuffling Channel

What is the capacity of this channel?

\[
X^n = \underbrace{\ldots}_{\beta \log n} \\
\text{shuffling channel} \\
Y^n = \underbrace{\ldots}
\]

- Maximum rate \( R = \frac{\log |C|}{n} \) with \( P_e \to 0 \) as \( n \to \infty \)
Example: BSC Shuffling Channel

\[ X^n = \]

\[ Y^n = \]

\[ C = 1 - H(p) - \frac{1}{\beta} \]

\( \beta \log n \)

shuffling channel

BSC \((p)\)

cost of lack of ordering
Simple scheme: index + individual encoding

- Encode **unique index** into each block
- Individually encode block with an optimal BSC code

\[
\frac{n}{\beta \log n}
\]

index length:
\[
\log \left(\frac{n}{\beta \log n}\right) \approx \log n
\]

encode with BSC code.

\[
\beta \log n
\]

Rate:
\[
\frac{n}{\beta \log n} \left(C_{BSC} \cdot \beta \log n - \log n\right) / n
\]
Converse idea

- Consider using code for BSC shuffling channel on a regular BSC

\[ R + \frac{1}{\beta} \leq 1 - \text{H}(p) \]

How much of \( \sigma \) can we decode?

\[ R_{\text{extra}} = \frac{I(\sigma; Y^n)}{n} \]
Example: BSC Shuffling Channel

\[ X^n = \]

\[ \beta \log n \]

\[ \text{shuffling channel} \]

\[ Y^n = \]

\[ C = 1 - H(p) - \frac{1}{\beta} \]

I. Shomorony and R. Heckel, DNA-Based Storage: Models and Fundamental Limits, IT Transactions, 2021
General DNA Storage Channel

\[ X^n = \{ \cdots \} \]

Poisson (\(\lambda\)) sampling channel

\[ \mathcal{Y} = \{ \cdots \} \]

\[ C = (1 - e^{-\lambda})(E[C_{DMC}] - 1/\beta) \]

Lenz, Siegel, Wachter-Zeh, Yaakobi, The Noisy Drawing Channel: Reliable Data Storage in DNA Sequences
Weinberger and Merhav, The DNA Storage Channel: Capacity and Error Probability Bounds
The wonders of high-throughput sequencing

- **Genome sequencing**
- **RNA-seq**
- **Metagenomics**
- **DNA storage**

High-throughput sequencer

- No location/ordering information!
- Reordering problem based on:
  - **Coding**
  - **Overlaps**
  - Reference (side information)
What is the capacity of this channel?
Shotgun Sequencing Channel

• Cannot place unique identifiers
• Overlaps can help reordering

\[ C_{SSC} = 1 - e^{-\lambda(1-1/\beta)} \]

sequencing depth: \[ \lambda = \frac{|\mathcal{Y}| \cdot \beta \log n}{n} \]

\[ C_{shuf} = (1 - e^{-\lambda})(1 - 1/\beta) \]

A. Ravi, A. Vahid and I. Shomorony, Coded Shotgun Sequencing, JSAIT, 2022
Shotgun Sequencing Channel

\[ X^n = \text{Shotgun Sequencing Channel} \]

\[ Y = \{ \text{Overlaps neutralize lack of ordering information} \} \]

\[ C_{ SSC} = 1 - e^{-\lambda(1-1/\beta)} \]

\[ C_{ shuf} = (1 - e^{-\lambda})(1 - 1/\beta) \]

\[ \lambda = \frac{|Y| \cdot \beta \log n}{n} \]
Coding for the Shotgun Sequencing Channel

\[ X^n = \]

Better approach: interleave a pilot sequence in \( X^n \)

any read contains same amount of pilot symbols
Coding for the Shotgun Sequencing Channel

- Better approach: interleave a pilot sequence in $X^n$

```
\begin{align*}
X^n &= \begin{array}{c}
\text{de Bruijn sequence} \\
\text{data blocks}
\end{array}
\end{align*}
```

any read contains same amount of pilot symbols

unique alignment

interleaved pilot: doesn’t take advantage of overlaps

$C_{SSC}$

$\lambda$

0.5

1

2 4 6
Coding for the Shotgun Sequencing Channel

$X^n = \{ \text{reads} \}$

$Y = \{ \text{merged reads} \}$

$\beta \log n$

$C_{SSC} = 1 - e^{-\lambda(1-1/\beta)}$

$\lambda = \frac{|Y| \cdot \beta \log n}{n}$

- Achievability
  - Random code: i.i.d. Ber(1/2) codewords
  - Decoding: merge reads into “islands” + align to codewords

$X^n(j) = \{ \text{merged reads} \}$
Shotgun Sequencing Channel

\[ X^n = \]

\[ y = \{ \]

\[ \beta \log n \}

\[ C_{SSC} = 1 - e^{-\lambda(1 - 1/\beta)} \]

Shuffling Channel (noise-free)

\[ X^n = \]

\[ y = \{ \]

\[ \beta \log n \}

\[ C_{shuf} = (1 - e^{-\lambda})(1 - 1/\beta) \]

\[ \lambda = \frac{|Y| \cdot \beta \log n}{n} \]
The wonders of high-throughput sequencing

- Genome sequencing
- RNA-seq
- Metagenomics
- DNA storage

- High-throughput sequencer

- No location/ordering information!
- Reordering problem based on:
  - Coding
  - Overlaps
  - Reference (side information)
The wonders of high-throughput sequencing

Genome sequencing

RNA-seq

Metagenomics

DNA storage

Outline:
1. Information-theoretic limits of out-of-order channels
2. Efficiently finding overlaps

• No location/ordering information!
Finding pairwise overlaps

- Computational bottleneck in bioinformatics
- Other applications: document comparison (plagiarism detection)
Finding pairwise overlaps

Given two noisy sequences, find best overlap between them

Dynamic programming solution: $O(m^2n^2)$ time
Finding overlaps in practice

$X_1 \xrightarrow{\text{sketching}}$ Can estimate overlap in time linear in $B$

$X_2 \xrightarrow{}$

$X_3 \xrightarrow{}$

$\vdots$

$X_m \xrightarrow{n, B \text{ bits}}$ $O(m^2B)$ time instead of $O(m^2n^2)$

- **Fundamental tradeoff:** Sketch size and overlapping accuracy
Distributed source coding formulation

- We focus on estimating the **overlap** size $\theta \in (0,1)$

\[ X \]

\[ X_1 \]
\[ (1-\theta)n \quad \theta n \quad (1-\theta)n \]

\[ X_2 \]

\[ \text{Enc}_1 \rightarrow \hat{X}_1 \]
\[ B \]

\[ \text{Enc}_2 \rightarrow \hat{X}_2 \]
\[ B \]

\[ \text{Dec} \rightarrow \hat{\theta} \]

- **Distortion:**

\[ E[(\theta - \hat{\theta})^2] \leq D \]

- **$B(D)$:** Minimum $B$ required to achieve distortion $D$ (as $n \rightarrow \infty$)

Standard approach to sketching

- Sketches based on locality-sensitive hashing (min-hash)

$X_1 \cdots \theta = 0.3 \cdots X_2$

$\begin{array}{cccccc}
 h_1 & 9 & 13 & 7 & 14 & 28 & 91 \\
 h_2 & 31 & 86 & 53 & 6 & 7 & 18 \\
 h_3 & 6 & 7 & 41 & 42 & 32 & 9 \\
 h_4 & 40 & 2 & 61 & 3 & 8 & 30 \\
 h_5 & 17 & 97 & 13 & 52 & 48 & 5
\end{array}$

$\begin{array}{cccccc}
 h_1 & 0 & 3 & 5 & 2 & 1 \\
 h_2 & 9 & 3 & 4 & 2 & 7 & 0 \\
 h_3 & 5 & 3 & 4 & 5 & 1 & 0 \\
 h_4 & 90 & 25 & 72 & 81 & 13 & 87 \\
 h_5 & 23 & 1 & 74 & 19 & 23 & 52
\end{array}$

overlap estimate $= 2/5$
Standard approach to sketching

- Sketches based on locality-sensitive hashing (min-hash)

Achievable sketch size: \[ B \leq \frac{3 \log n}{D} \]
Previous approach based on min-hash achieves

\[ B(D) \leq \frac{3 \log n}{D} \]

**Theorem:** The function \( B(D) \) satisfies

\[ K_1 \cdot \log \left( \frac{1}{D} \right) \leq B(D) \leq K_2 \cdot \log^2 \left( \frac{1}{D} \right) \]

New sketching idea: Lexicographic Hashing

I. Shomorony and G. Kamath, Sketching and Sequence Alignment: A Rate-Distortion Perspective, ISIT, 2022
New idea: Lexicographic Hashing

- Encode location of longest run of zeros
- Overlap estimate: $\hat{\theta} = 1 - (\hat{X}_1 - \hat{X}_2)$
New idea 2: Multiple lexicographic orderings

- Encode location of lexicographically first suffix

\[ \hat{\theta} = 1 - \frac{147}{256} = \frac{109}{256} = 0.426 \]
Fundamental Limits of Sketching

- Previous approach based on min-hash achieves

\[ B(D) \leq \frac{3 \log n}{D} \]

Minimum sketch size

**Theorem:** The function \( B(D) \) satisfies

\[ K_1 \cdot \log \left( \frac{1}{D} \right) \leq B(D) \leq K_2 \cdot \log^2 \left( \frac{1}{D} \right) \]

New sketching idea:

Lexicographic Hashing

I. Shomorony and G. Kamath, Sketching and Sequence Alignment: A Rate-Distortion Perspective, ISIT, 2022
Practical overlap finding: LexicHash

- Sequence alignment tool based on lexicographic hashing
- More accurate overlap estimates
- Avoids need to select parameter $k$ from min-hash

G. Greenberg, A. Ravi, I. Shomorony, LexicHash: Sequence Similarity Estimation via Lexicographic Comparison of Hashes (submitted)
Practical overlap finding: LexicHash

- Sequence alignment tool based on lexicographic hashing
- More accurate overlap estimates
- Avoids need to select parameter $k$ from min-hash
- Significant improvements in overlap classification:

![Graphs showing precision and recall for S. pyogenes and P. falciparum with different AUC values for LH, MH (k=10), MH (k=12), MH (k=16).]

G. Greenberg, A. Ravi, I. Shomorony, LexicHash: Sequence Similarity Estimation via Lexicographic Comparison of Hashes (submitted)
The wonders of high-throughput sequencing

Outline:
1. Information-theoretic limits of out-of-order channels
2. Efficiently finding overlaps

- No location/ordering information!
Out-of-order information?

- Packet networks

<table>
<thead>
<tr>
<th>Packets:</th>
<th></th>
<th>Shuffling channel</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10110110</td>
<td>10100111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00010101</td>
<td>10110110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11011011</td>
<td>11011101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000100</td>
<td>00101000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11011101</td>
<td>11101001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00101000</td>
<td>1101001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11101001</td>
<td>10100111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10100111</td>
<td>10000100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Out-of-order information?

- Intermittent broadcast communication?
Out-of-order information?

- Dataset alignment?
Out-of-Order Information Theory?

- Packets: 01100011101010101, 11010110
- Random delay on each path
- Shuffling channel
- 01100011101010101, 11010110
- Random delay on each path
- Shuffling channel
- High-throughput sequencer
- Thank you!