Codes for Cost-Efficient DNA Synthesis

Andreas Lenz\textsuperscript{1}, Yi Liu\textsuperscript{2}, Cyrus Rashtchian\textsuperscript{3}, Paul H. Siegel\textsuperscript{2}, Andrew Tan\textsuperscript{2} Antonia Wachter-Zeh\textsuperscript{1}, Eitan Yaakobi\textsuperscript{4}

\textsuperscript{1}Institute for Communications Engineering, Technische Universität München, Germany
\textsuperscript{2}Department of Electrical and Computer Engineering, University of California, San Diego, USA
\textsuperscript{3}Computer Science and Engineering Department, University of California, San Diego, USA
\textsuperscript{4}Computer Science Department, Israel Institute of Technology, Haifa, Israel

Non-Volatile Memory Workshop
7-10 March 2021
Outline

System Model

Problem Formulation

Synthesis, Subsequences, Graphs

Relations to Cost-Constrained Systems

Fixed Length Sequences

Conclusion
Data Storage in DNA

User Binary Data
000001101011001
110100010010101
101000111110100

Encoding
DNA strands
GCTATGAGTACT
ATGATTGACTCT
GATGGCATAGCT

DNA Synthesizer

Decoding
DNA Sequencer

DNA strands
ATAATTGAGTCT
GCTGGCATAGCT
GATAGCTTAGCT
ATGATTGACTCT
GATGGCATACCT

Storage Container

Lenz, Liu, Rashtchian, Siegel, Tan, Wachter-Zeh, Yaakobi - "Codes for Cost-Efficient DNA Synthesis"
DNA Synthesis

Array-based synthesis

- Nucleotide-by-nucleotide parallel synthesis of DNA strands
- Flushing of nucleotides and selection of strands

<table>
<thead>
<tr>
<th>Cycle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S)</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>T</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>T</td>
</tr>
<tr>
<td>(x_1)</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>T</td>
<td>(\text{CTACG})</td>
<td></td>
</tr>
<tr>
<td>(x_2)</td>
<td>A</td>
<td>G</td>
<td>T</td>
<td>A</td>
<td>(\text{AGTA})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x_3)</td>
<td>C</td>
<td>T</td>
<td>(\text{CTT})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure**: Synthesis of \(x_1 = \text{(CTACG)}\), \(x_2 = \text{(AGTA)}\), \(x_3 = \text{(CTT)}\) using Synthesis sequence \(S = \text{(ACGTACGT)}\).
DNA Synthesis

Array-based synthesis

- Parallel synthesis of $x_1, x_2, \ldots, x_k \in \Sigma^*_q$ ($q = 4$ for DNA)
- Common synthesis sequence $S$

**Synthesis time**

\[
t(S, x_1, \ldots, x_k) = \min_t \text{ s.t. } x_i \text{ is subsequence of } S_{1:t}
\]

- E.g. $S = (ACGTACGTA)$, $x_1 = (CTACG)$, $x_2 = (AGTA)$, $x_3 = (CTT)$.
  \[\Rightarrow t(S, x_1, x_2, x_3) = 8.\]

**Problem formulation**

- Synthesis code $C \subseteq \Sigma^*_q$
- Goal: maximize information synthesized in time $T$

\[
N(S, T) = \max |C| \text{ s.t. } t(S, x) \leq T \forall x \in C
\]

Lenz, Liu, Rashtchian, Siegel, Tan, Wachter-Zeh, Yaakobi - "Codes for Cost-Efficient DNA Synthesis"
DNA Synthesis

\[ N(S, T) = \max |C| \text{ s.t. } t(S, x) \leq T \ \forall \ x \in C \]

- Maximum information rate per cost time

\[ R(S) = \lim_{T \to \infty} \frac{\log N(S, T)}{T} \]

- \( R(S) \) measured in bits/synthesis cycle

**Contribution**

- Characterize \( R(S) \) for **arbitrary** periodic sequences \( S \) using cost-constrained systems
- Highlight parallels to the subsequence spectrum
Subsequence Graph

- Vertices: Symbols of $S$
- Edges: each vertex has $q$ outgoing edges, one to the next appearance of each $\sigma \in \Sigma_q$

**Figure**: Subsequence graph $G(S)$ for Synthesis sequence $S = (ACGTACGT)$.

**Observation**

$x$ can be synthesized using $S$ iff it is obtained by some path through $G(S)$

**Lemma (informal)**

$N(S, T) = \#\text{paths through } G(S_{1:T})$
Subsequence Spectrum

- Subsequences of $S$: All words obtained by deleting symbols from $S$.
- E.g.

$$D((ACG)) = \{(ACG), (AC), (AG), (CG), (A), (C), (G), ()\}$$

Lemma

$$N(S, T) = |D(S_{1:T})|$$

- Proof follows from left-alignment of subsequences in subsequence graph

Equivalences

$$N(S, T) \triangleq \text{largest synthesis code} \iff \# \text{paths through } G(S) \iff \# \text{subsequences}$$

- Alternating sequence $A_q = (0, 1, 2, \ldots, q - 1, 0, 1, 2, \ldots)$

$$\implies \arg \max_S N(S, T) = A_q$$
Periodic Synthesis Sequences

- Periodic sequence: $S = (ssss \ldots)$, where $s \in \Sigma^L$ is the period
- E.g.: Synthesis sequence $S = (AACAAC \ldots)$

Periodic subsequence graph

- Vertices: Symbols of one period
- Edges: as in original graph, with step size as weight
Periodic Synthesis Sequences

- Periodic subsequence graph $\tilde{G}(S)$

**Lemma (informal)**

$$N(S, T) = \#\text{paths through } \tilde{G}(S) \text{ with sum weight at most } T$$

- Edge weights: $\tau_{i,j}$ define costs from vertex $i$ to $j$
- $\tilde{G}(S)$ is a cost-constrained system
Periodic Synthesis Sequences

- Define $C(S)$: Combinatorial capacity of cost-constrained system $\tilde{G}(S)$

Theorem

Maximal synthesis information rate per cost time

$$\lim_{T \to \infty} \frac{\log N(S, T)}{T} =: R(S) = C(S)$$

- Use results from constrained coding theory to find $R(S)$:

Combinatorial capacity of cost-constrained systems

- Given $\tilde{G}(S)$ with edge weights $\tau_{i,j}$
- Define $L \times L$ matrix $[P(z)]_{i,j} = \begin{cases} 0, & \text{if there is no edge from vertex } i \text{ to } j \\ 2^{-z\tau_{i,j}}, & \text{otherwise} \end{cases}$
- $z_0$ : largest real solution to $\det(1 - P(z)) = 0$
  $\implies C(S) = z_0$
Periodic Synthesis Sequences

- Example: Alternating sequence $S = (ACGTACGT \ldots )$

\[
P(z) = \begin{pmatrix}
2^{-4z} & 2^{-z} & 2^{-2z} & 2^{-3z} \\
2^{-3z} & 2^{-4z} & 2^{-z} & 2^{-2z} \\
2^{-2z} & 2^{-3z} & 2^{-4z} & 2^{-z} \\
2^{-z} & 2^{-2z} & 2^{-3z} & 2^{-4z}
\end{pmatrix}
\]

$\implies R(S) = z_0 \approx 0.947$ bit/cycle

- Uncoded: 0.5 bit/cycle
- Arbitrary $q$: $\sum_{i=1}^{q} 2^{-zi} = 1$

Lenz, Liu, Rashtchian, Siegel, Tan, Wachter-Zeh, Yaakobi - "Codes for Cost-Efficient DNA Synthesis"
Explicit Codes: Varn Codes

- Varn codes: Explicit prefix-free codes for cost-constrained systems
- Construction via tree expansion

With codebook size $4^8$ we attain an information rate of $R \approx 0.928$ bit/cycle vs. capacity $R(S) \approx 0.947$ bit/cycle

Lenz, Liu, Rashtchian, Siegel, Tan, Wachter-Zeh, Yaakobi - "Codes for Cost-Efficient DNA Synthesis"
Fixed Length Sequences

- Restrict $x \in \Sigma^q$ to have fixed length $n$.
- Largest synthesis code with fixed length: $N(S, T, n)$
- Information rate per cycle:

$$R(S, \alpha) = \lim_{T \to \infty} \frac{N(S, T, \alpha T)}{T}$$

- $0 \leq \alpha \leq 1$

Equivalences

$N(S, T, n)$ largest synthesis code $\leftrightarrow$ #length-$n$ paths through $G(S)$ $\leftrightarrow$ #length-$n$ subsequences
Fixed Length Sequences

- Example: Binary alternating sequence $S = (ACACAC \ldots)$

$$R(S, \alpha) = \begin{cases} 
\alpha, & \alpha \leq \frac{2}{3} \\
\alpha h(\alpha^{-1} - 1), & \alpha > \frac{2}{3}
\end{cases}.$$
Conclusion

Summary

• Time-efficient synthesis
• Optimal synthesis codes ↔ paths through graph ↔ subsequences
• Solution for arbitrary periodic sequences using constrained coding techniques
• Implies results for asymptotic growth rate of number of subsequences

Outlook

• Incorporate error-correction
• Variable synthesis sequence

Thank you!