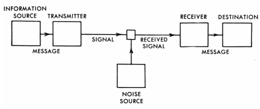
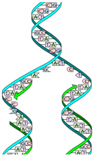
Part 1: DNA-Based Data Storage and Computing

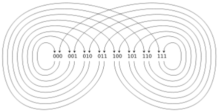
Olgica Milenkovic
University of Illinois, Urbana-Champaign

North American School of Information Theory, Texas, 2018

May 2018

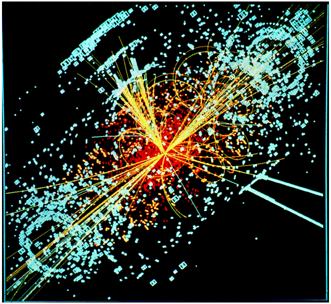






The Era of Massive Data

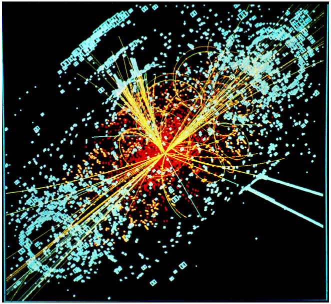
- Large Hadron Collider: 600 million collisions/s, 0.5 PB per week.



Credit: In search of the God particle, Wikipedia.

The Era of Massive Data

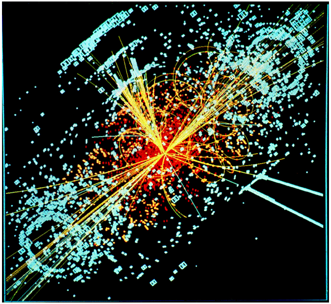
- Large Hadron Collider: 600 million collisions/s, 0.5 PB per week.
- DNA sequencing data: $30 - 50$ TB per week.



Credit: In search of the God particle, Wikipedia.

The Era of Massive Data

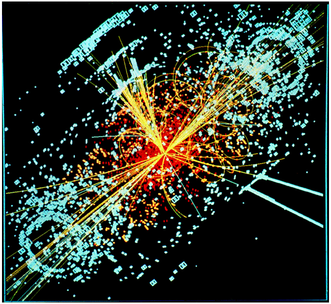
- Large Hadron Collider: 600 million collisions/s, 0.5 PB per week.
- DNA sequencing data: $30 - 50$ TB per week.
- Sloan Digital Sky Survey: $1 - 2$ TB per week.



Credit: In search of the God particle, Wikipedia.

The Era of Massive Data

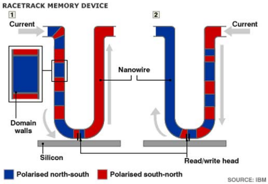
- Large Hadron Collider: 600 million collisions/s, 0.5 PB per week.
- DNA sequencing data: $30 - 50$ TB per week.
- Sloan Digital Sky Survey: $1 - 2$ TB per week.
- Social networks (Twitter, Facebook, LinkedIn), NASA weather surveys, consumer and stock market data, Internet sources...



Credit: In search of the God particle, Wikipedia.

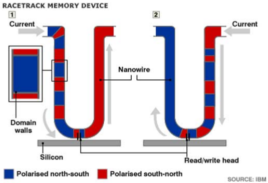
How to Cope?

- Pushing the limits of existing storage media: Magnetic tapes, disks, flash, 3D flash,...



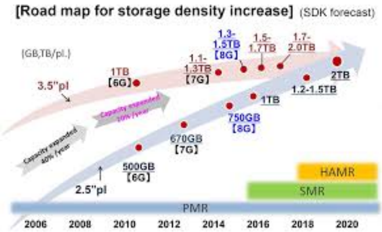
How to Cope?

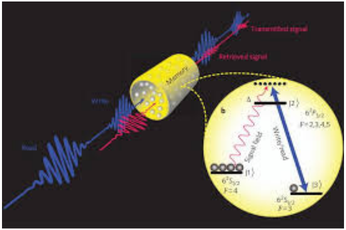
- Pushing the limits of existing storage media: Magnetic tapes, disks, flash, 3D flash,...
- Data compression: New initiatives by NIH (BD2K Targeted Software Development for Genomic Data Compression) and other efforts.



How to Cope?

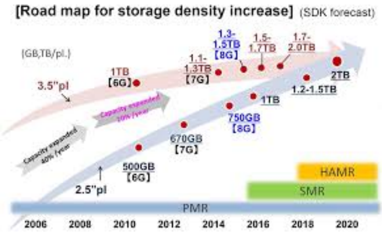
- New storage media: quantum memories, nanofilm storage, polymer-based storage?

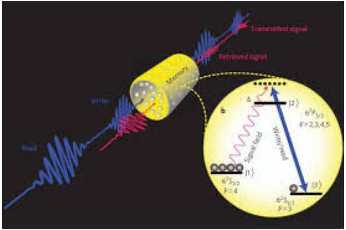




How to Cope?

- New storage media: quantum memories, nanofilm storage, polymer-based storage?
- Data compression: What densities are possible?





DNA-Based Data Storage

Looking for Alternative Storage Media: DNA

- DNA is extremely durable: Can still "read" mammoth, Neanderthal, and 700,000 old horse bone DNA!

Looking for Alternative Storage Media: DNA

- DNA is extremely durable: Can still "read" mammoth, Neanderthal, and 700,000 old horse bone DNA!
- DNA information content of Human cell: 6.4 GB. Mass of a cell: ~ 3 picograms. No. of cells: $15 - 40 \times 10^{12}$.

Looking for Alternative Storage Media: DNA

- DNA is extremely durable: Can still "read" mammoth, Neanderthal, and 700,000 old horse bone DNA!
- DNA information content of Human cell: 6.4 GB. Mass of a cell: ~ 3 picograms. No. of cells: $15 - 40 \times 10^{12}$.
- Can one store information in DNA?

Looking for Alternative Storage Media: DNA

- DNA is extremely durable: Can still "read" mammoth, Neanderthal, and 700,000 old horse bone DNA!
- DNA information content of Human cell: 6.4 GB. Mass of a cell: ~ 3 picograms. No. of cells: $15 - 40 \times 10^{12}$.
- Can one store information in DNA?
- This question has been raised before: "There is plenty of room at the bottom," R. Feynman.

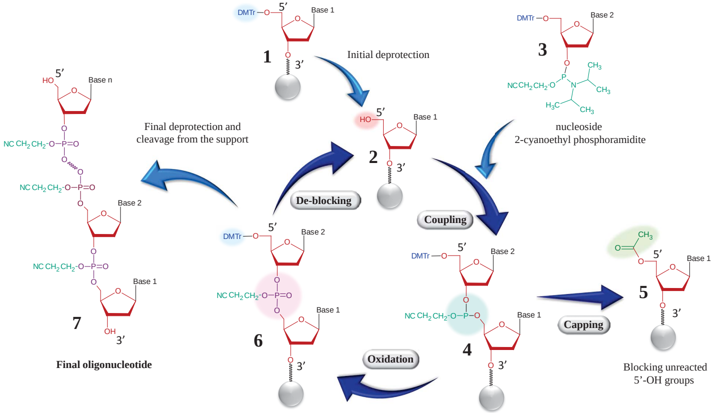
We can write...

We can "write" in DNA using what is called the process of DNA Synthesis.

We can write...

We can “write” in DNA using what is called the process of DNA Synthesis.

Biochemistry of synthesis: Stitching together bases from the set $\{A, T, G, C\}$ through deprotection & coupling cycles.



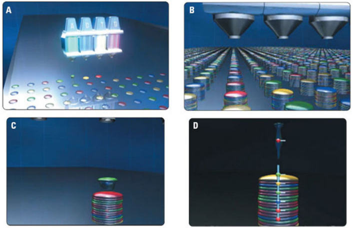
We Can Write...

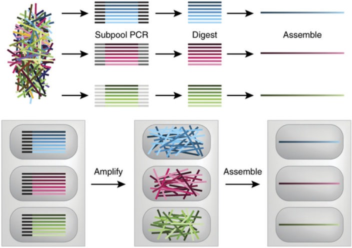
Commercial synthesis: Agilent, Gen9, IDT, Twist Bioscience (recent feature on "Rewriting DNA Synthesis on Silicon").

We Can Write...

Commercial synthesis: Agilent, Gen9, IDT, Twist Bioscience (recent feature on “Rewriting DNA Synthesis on Silicon”).

- DNA microarray-based short string (oligo) pool synthesis (left): Cost effective, large scale. Moderate error rates.
- Long strand (gBlocks) synthesis (right): Assembles short blocks. Chemical error-correction.

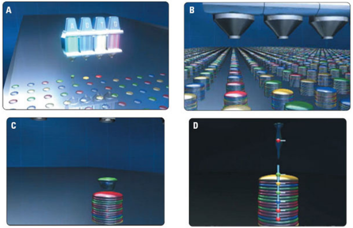


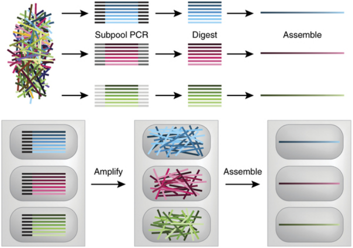


We Can Write...

Commercial synthesis: Agilent, Gen9, IDT, Twist Bioscience (recent feature on "Rewriting DNA Synthesis on Silicon").

- DNA microarray-based short string (oligo) pool synthesis (left): Cost effective, large scale. Moderate error rates.
- Long strand (gBlocks) synthesis (right): Assembles short blocks. Chemical error-correction.
- Types of synthesis errors: Deletions, insertions, substitutions.





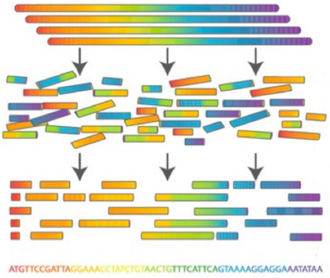
We Can Read...

Illumina (MiSeq): Best overall performance of modern sequencing technologies in terms of yield and accuracy. Relatively small error rates (substitutions and rare deletions). Short read length.

We Can Read...

Illumina (MiSeq): Best overall performance of modern sequencing technologies in terms of yield and accuracy. Relatively small error rates (substitutions and rare deletions). Short read length.

Steps: Cloning /// Shearing /// Reading of unordered pool /// Computer aided alignment of overlapping fragments /// Consensus



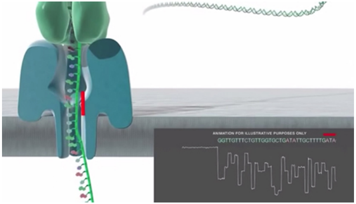
We Can Read...

Oxford Nanopore - MinIon: Longer read lengths, portable architecture.
Context-dependent deletion, insertion and substitution errors.

We Can Read...

Oxford Nanopore - MinIon: Longer read lengths, portable architecture. Context-dependent deletion, insertion and substitution errors.

Key properties: Biological pore(s) and motor, base calling using deep learning techniques.



We Can Amplify and Enable Random Access...

Polymerase Chain Reaction (PCR): Cheap, fast, “exponential” information replication.

We Can Amplify and Enable Random Access...

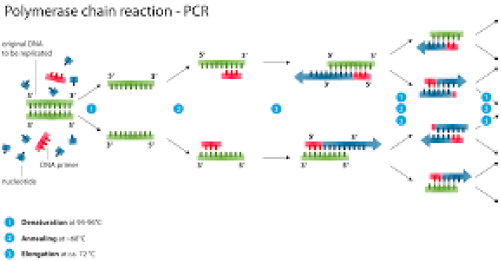
Polymerase Chain Reaction (PCR): Cheap, fast, "exponential" information replication.

Primers - Key enablers of PCR: Short DNA strands that initiate replication at "strand-matching" locations (red blocks).

We Can Amplify and Enable Random Access...

Polymerase Chain Reaction (PCR): Cheap, fast, “exponential” information replication.

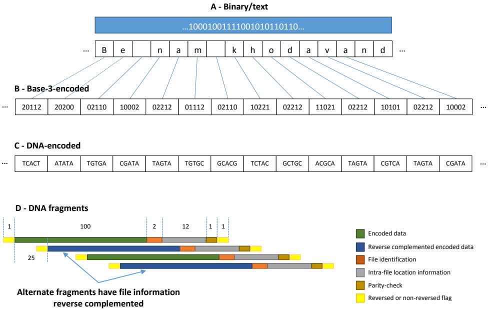
Primers - Key enablers of PCR: Short DNA strands that initiate replication at “strand-matching” locations (red blocks).



DNA Data Storage Platforms

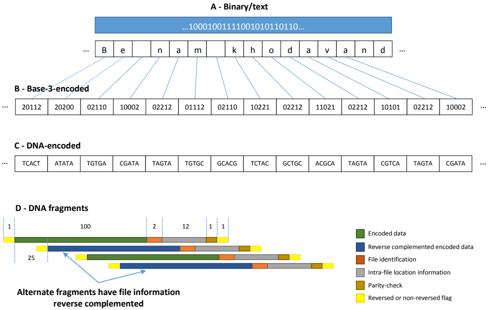
“Double Helix Serves Double Duty”, NY Times, Jan 2013

- Church *et al.* (Science, 2012) and later Goldman *et al.* (Nature, 2013) stored 739 KB of data in synthetic DNA, mailed it and recreated the original digital files using Illumina readers.



"Double Helix Serves Double Duty", NY Times, Jan 2013

- Church *et al.* (Science, 2012) and later Goldman *et al.* (Nature, 2013) stored 739 KB of data in synthetic DNA, mailed it and recreated the original digital files using Illumina readers.
- Digital archival storage systems that will safely store the equivalent of one million CDs in a gram of DNA for 10,000 years.

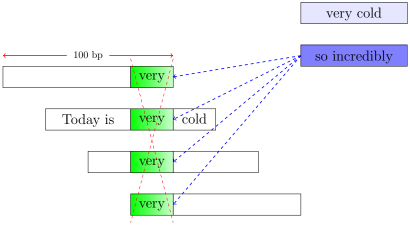


Can one Randomly Access and Rewrite the Data?

- Problem 1: Random access was impossible in first implementation - need to "read" whole book to find one sentence.
- Problem 2: To perform editing, need to change large number of reads (fragments).
- Problem 3: The first schemes were sensitive to contextual errors.

Storage format of Goldman *et al.*: overlapping reads akin to sequencer output.

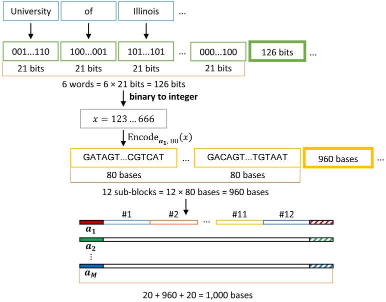
AAATTTTGCGCTATTGCCCAATT**GCC**GGGTTAAAATATATGAGACTCTAAA...



"Data Storage on DNA Can Keep it Safe for Centuries," NY Times, Dec 2015

A fully operational random access and rewritable DNA-based memory with *Sanger sequencing*.

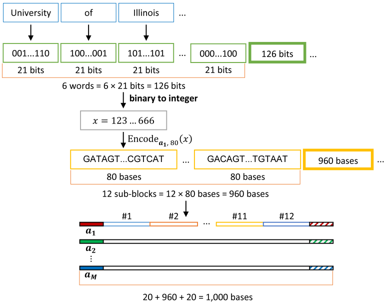
- Yazdi et.al., 2015 - First random access, rewritable DNA-based storage system. Encoded Wikipedia entries for six US universities.



"Data Storage on DNA Can Keep it Safe for Centuries," NY Times, Dec 2015

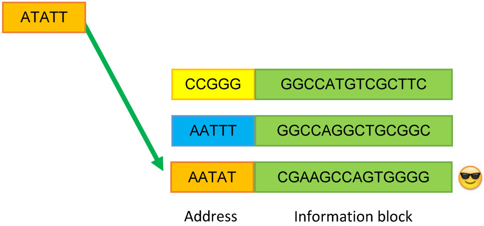
A fully operational random access and rewritable DNA-based memory with *Sanger sequencing*.

- Yazdi et.al., 2015 - First random access, rewritable DNA-based storage system. Encoded Wikipedia entries for six US universities.
- Large scale testing of our methods: Microsoft Research/Twist Bioscience, 2016. Coverage by Spectrum, Nature, New Scientist etc.



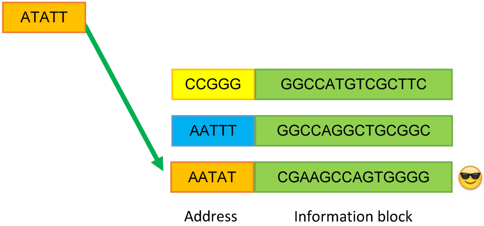
Random Access via Addressing and PCR

- The addressing system: Primers=Addresses, used in PCR reaction. Random access equals exponential amplification.



Random Access via Addressing and PCR

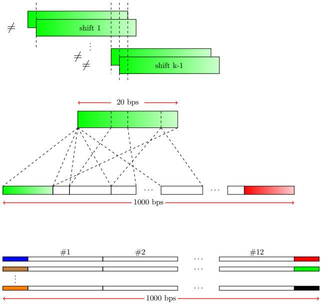
- The addressing system: Primers=Addresses, used in PCR reaction. Random access equals exponential amplification.



- How to avoid addressing errors?

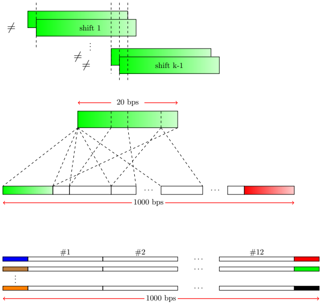
Address Properties

- Addresses need to be sufficiently different (Hamming, Levenshtein distance) and avoided elsewhere in the blocks.



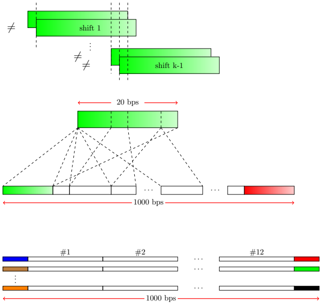
Address Properties

- Addresses need to be sufficiently different (Hamming, Levenshtein distance) and avoided elsewhere in the blocks.
- Addresses should not fold: Needed for accurate amplification.



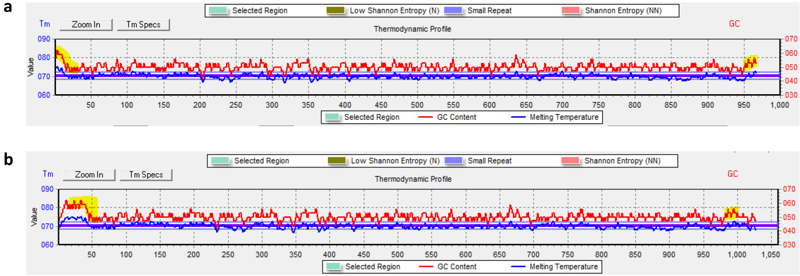
Address Properties

- Addresses need to be sufficiently different (Hamming, Levenshtein distance) and avoided elsewhere in the blocks.
- Addresses should not fold: Needed for accurate amplification.
- Addresses should have balanced GC content: Needed for stable melting temperature.



GC Imbalance Hurts!

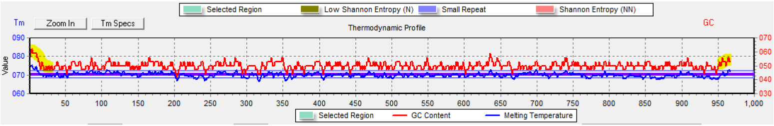
- Synthesis constraints identification with IDT.



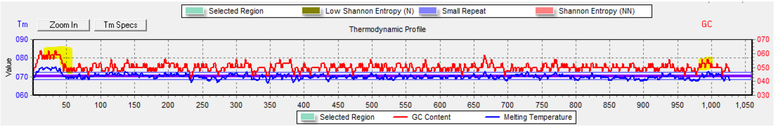
GC Imbalance Hurts!

- Synthesis constraints identification with IDT.
- Balancing constraints: GC-content has to be balanced in small block-lengths at the $3'$ and $5'$ ends of the strings, longer blocks allowed within the sequence (blocklength$=8$).

a



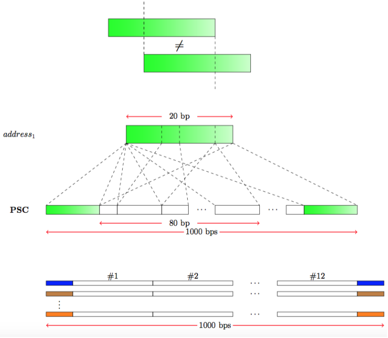
b



The Constrained Coding Components

Definition. A sequence $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}_q^n$ is self uncorrelated if no proper prefix of $\mathbf{a}$ matches its suffix, i.e., $(a_1, \ldots, a_i) \neq (a_{n-i+1}, \ldots, a_n)$, for all $1 \leq i < n$.

Extension: A mutually uncorrelated (cross-bifix-free) code is a set of sequences such that for any two sequences $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$ in the code no proper prefix of $\mathbf{a}$ appears as a suffix of $\mathbf{b}$ and vice versa [L70, G60, B12].



Address Sequence Construction

Enumeration and construction of strings of a given length that contain no elements of some fixed set of strings as subwords [GO80's]:

Take addresses as "forbidden words" to ensures specific random access. Relax constraints.

MU vs. Weakly MU: A k-weakly mutually uncorrelated (WMU) code is a set of sequences such that for any two sequences $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$ in the code no proper prefix of $\mathbf{a}$ of length $\geq k$ appears as a suffix of $\mathbf{b}$ and vice versa [TKM16].
Construction of balanced WMU codes Hamming distance constraints?

For $\mathbf{a} = (a_1, \ldots, a_s), \mathbf{b} = (b_1, \ldots, b_s) \in \{0,1\}^s$, define

$$\Psi(\mathbf{a}, \mathbf{b}) : \{0,1\}^s \times \{0,1\}^s \to \{\mathtt{A}, \mathtt{T}, \mathtt{C}, \mathtt{G}\}^s$$

according to:

$$\text{for } 1 \leq i \leq s,\ c_i = \begin{cases} \mathtt{A} & \text{if } (a_i, b_i) = (0,0) \\ \mathtt{C} & \text{if } (a_i, b_i) = (0,1) \\ \mathtt{T} & \text{if } (a_i, b_i) = (1,0) \\ \mathtt{G} & \text{if } (a_i, b_i) = (1,1) \end{cases}$$

Address Sequence Construction

Decoupling the construction: Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \{0,1\}^s$ be two binary block code of length s. Encode all pairs $(\mathbf{a}, \mathbf{b}) \in \mathcal{C}_1 \times \mathcal{C}_2$ using $\mathcal{C}_3 = \{\Psi(\mathbf{a}, \mathbf{b}) \mid \mathbf{a} \in \mathcal{C}_1, \mathbf{b} \in \mathcal{C}_2\}$. Then:

1. $\mathcal{C}_3$ is balanced if $\mathcal{C}_2$ is balanced.
2. $\mathcal{C}_3$ is a k-WMU code if either $\mathcal{C}_1$ or $\mathcal{C}_2$ is a k-WMU code.
3. If d_1 and d_2 are the minimum Hamming distances of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, then the minimum Hamming distance of $\mathcal{C}_3$ is at least $\min(d_1, d_2)$.

See also [LY17] for MU codes.

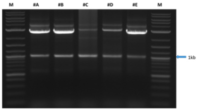
Information sequence encoding?

Information Sequence Encoding for Texts

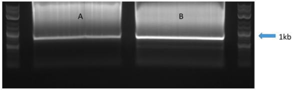
Modification based on [WI90's], and new approach in [TGM17].

Random Access and Rewriting Experiments

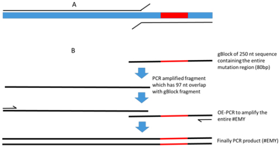
- Random access achieved via PCR, addresses used as primers.



PCR of five primers



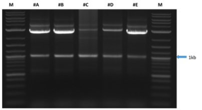
PCR of selected string from pool (A) and in individual well (B)

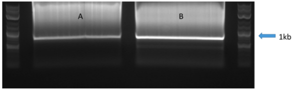


Rewriting via the gBlock process

Random Access and Rewriting Experiments

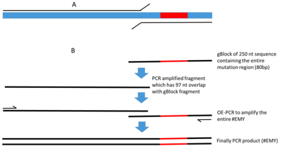
- Random access achieved via PCR, addresses used as primers.
- Context identification and rewriting performed via gBlock or OE-PCR methods.





PCR of five primers

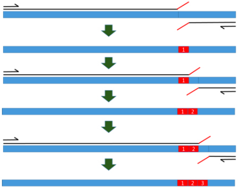
PCR of selected string from pool (A) and in individual well (B)



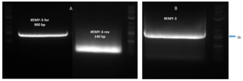
Rewriting via the gBlock process

Random Access and Rewriting Experiments

- Random access achieved via PCR, addresses used as primers.



Cheap 80-primer sequential rewriting

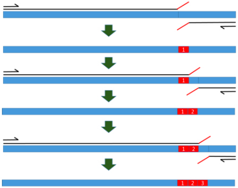


A) Two PCR products of rewrite. B) The generated PCR rewrite with correct size of 1kb.

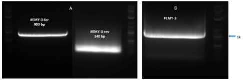
Sequencing results of 10 plasmids (5 from original, 5 from rewrite) with primer in forward direction of the insert. The rewritten region is covered in the red square

Random Access and Rewriting Experiments

- Random access achieved via PCR, addresses used as primers.
- Context identification and rewriting performed via gBlock or OE-PCR methods.



Cheap 80-primer sequential rewriting

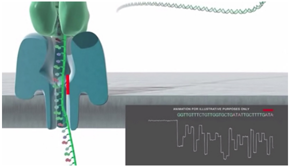


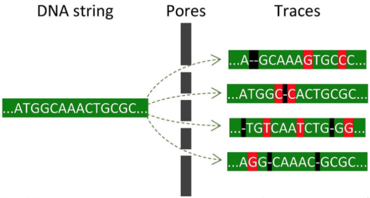
A) Two PCR products of rewrite. B) The generated PCR rewrite with correct size of 1kb.

Sequencing results of 10 plasmids (5 from original, 5 from rewrite) with primer in forward direction of the insert. The rewritten region is covered in the red square

Reading with Nanopores

MinION Oxford Nanopore (R7): Sequence traces (reads).





Reading with Nanopores

Major Problem: Very large number of sequence-dependent indel and substitution errors (R.7 flowcell, ~ 10%, R 9.4 flowcell ~ 4%)!

Example Statistics:

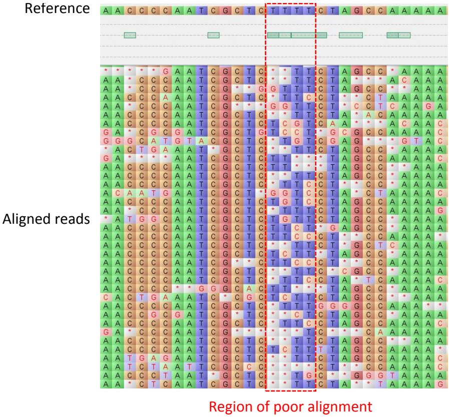
Block (length)	Number of reads	Sequencing Coverage depth		Number of errors: (substitution, insertion, deletion)		
		Average	Maximum	Per read (average)	Consensus	
					Nanopolish	Our method
1 (1,000)	201	176.145	192	(107, 14, 63)	(14,32,5)	(0, 0, 2)
2 (1,000)	407	315.521	349	(123, 12, 70)	(75,99,40)	(0, 0, 0)
3 (1,000)	490	460.375	482	(80, 23, 42)	(10,45,0)	(0, 0, 0)
4 (1,000)	100	81.763	87	(69, 18, 37)	(1,54,1)	(0, 0, 0)
5 (1,000)	728	688.663	716	(88, 20, 48)	(4,45,3)	(0, 0, 0)
6 (1,000)	136	120.907	129	(79, 21, 42)	(390,102,61)	(0, 0, 0)
7 (1,000)	577	542.78	566	(83, 26, 41)	(3,31,3)	(0, 0, 0)
8 (1,000)	217	199.018	207	(83, 20, 46)	(18,51,1)	(0, 0, 0)
9 (1,000)	86	56.828	75	(60, 16, 30)	(404,92,54)	(0, 0, 0)
10 (1,000)	442	396.742	427	(91, 18, 52)	(388,100,59)	(0, 0, 0)
11 (1,000)	114	101.826	110	(79, 23, 42)	(16,23,18)	(0, 0, 0)
12 (1,000)	174	162.559	169	(94, 23, 50)	(14,59,1)	(0, 0, 0)
13 (1,060)	378	352.35	366	(88, 26, 44)	(7,55,4)	(0, 0, 0)
14 (1,000)	222	189.918	203	(69, 22, 34)	(15,34,3)	(0, 0, 0)
15 (1,000)	236	222.967	232	(92, 24, 45)	(15,46,2)	(0, 0, 0)
16 (1,000)	198	182.99	195	(103, 16, 61)	(15,62,4)	(0, 0, 1)
17 (880)	254	240.273	250	(77, 19, 42)	(359,95,44)	(0, 0, 0)

Sequence Alignment

- First Step: "Merge traces" into one consensus sequence.

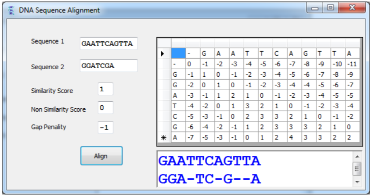
Sequence Alignment

- First Step: "Merge traces" into one consensus sequence.
- Bioinformatics: Sequence alignment [NW, SW].
  Computer Science: Reconstructing sequences from traces [Batu et.al. 2004].



Sequence Alignment

DP: Optimal, but of very high complexity.



Works poorly on real data!

AGTTGCGCACTTTAA...

GTGCCCAACTTTTA...

AATGCGGACTTTAA...

AATTGGCAACTTA...

ATGCCACTTAA...

A...

GTTGCGCACTTTAA...

GTGCCCAACTTTTA...

ATGCGGACTTTAA...

ATTGGCAACTTA...

TGCCACTTAA...

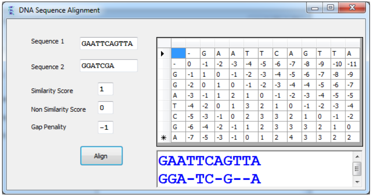
AG... or AA...

- Reference-free sequence alignment: CLUSTAL, KALIGN, MUSCLE, TCOFFEE, etc.

Sequence Alignment

DP: Optimal, but of very high complexity.

Works poorly on real data!

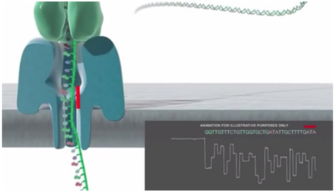


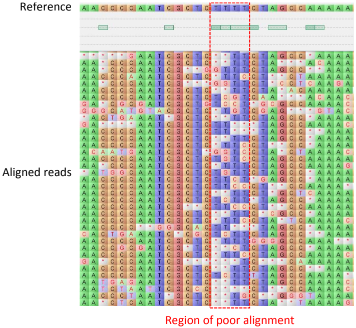
AGTTGCGCACTTTAA...
GTGCCCAACTTTTA...
AATGCGGACTTTAA...
AATTGGCAACTTA...
ATGCCACTTAA...
A...
GTTGCGCACTTTAA...
GTGCCCAACTTTTA...
ATGCGGACTTTAA...
ATTGGCAACTTA...
TGCCACTTAA...
AG... or AA...

- Reference-free sequence alignment: CLUSTAL, KALIGN, MUSCLE, TCOFFEE, etc.
- "Simple" trace reconstruction: [Batu et.al. 2004], [Holenstein et.al. 2016]

How Do We Handle Deletions?

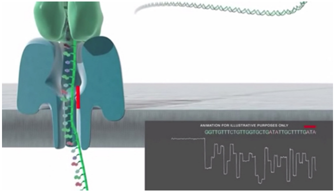
- Key idea: Use addresses (tags) as pilot sequences to identify good quality reads.

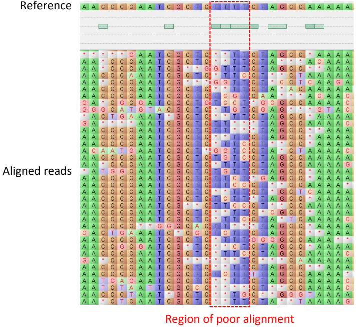




How Do We Handle Deletions?

- Key idea: Use addresses (tags) as pilot sequences to identify good quality reads.
- Align good reads. Use balancing property to correct errors, repeat. Use homopolymer coding.

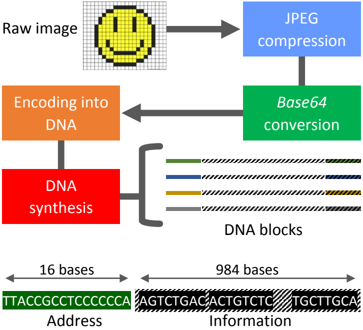


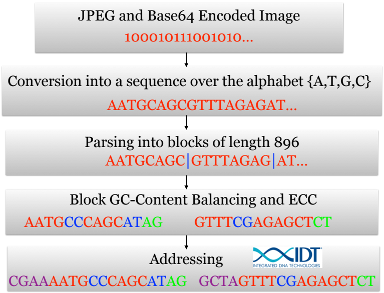


Data Encoding

Encoded images in compressed format into DNA.

- Compression, Base64 conversion, error-correcting and constrained coding (balancing GC content and forbidden address sequences).

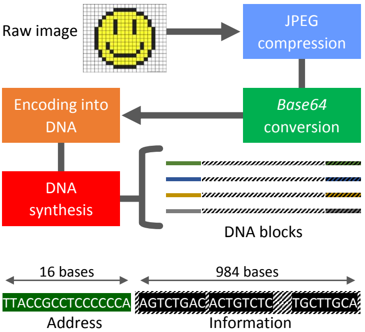


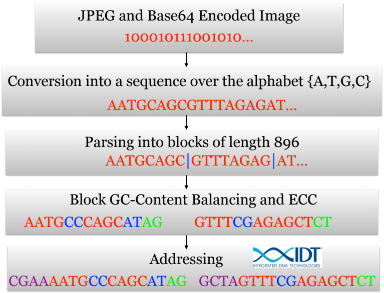


Data Encoding

Encoded images in compressed format into DNA.

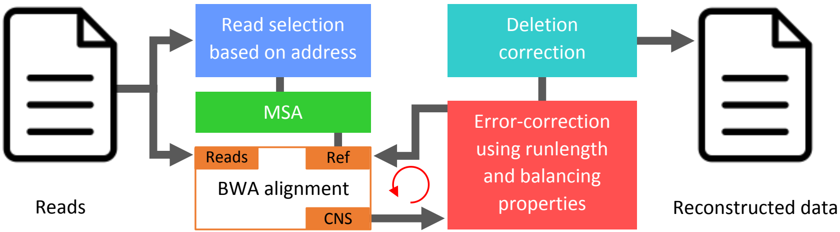
- Compression, Base64 conversion, error-correcting and constrained coding (balancing GC content and forbidden address sequences).
- Careful address design.





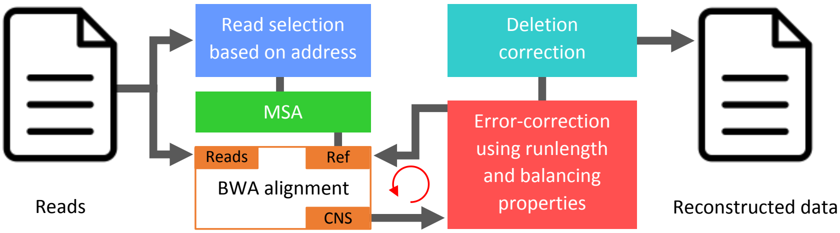
Our Readout Solution Summary

- Select best reads for first alignment: Best reads=highest quality addresses!



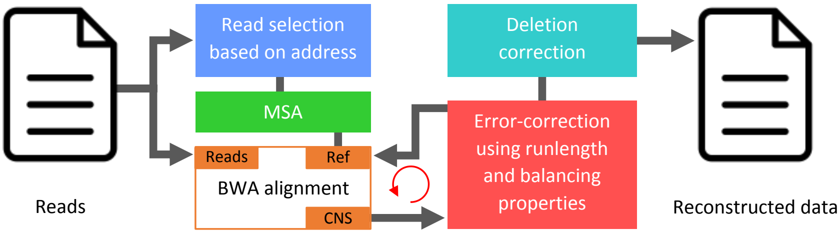
Our Readout Solution Summary

- Select best reads for first alignment: Best reads=highest quality addresses!
- Perform alignment: Roughly 30 traces (reads) involved.



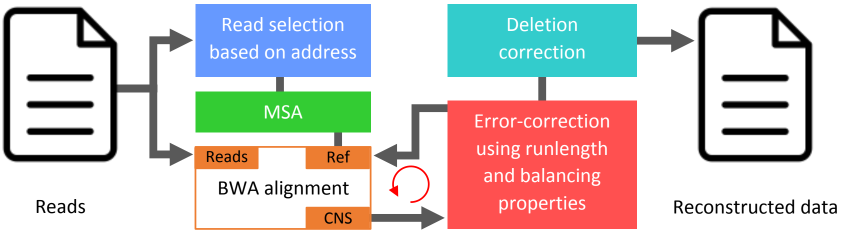
Our Readout Solution Summary

- Select best reads for first alignment: Best reads=highest quality addresses!
- Perform alignment: Roughly 30 traces (reads) involved.
- Adjust sequence balance. Balancing also limits runlengths!



Our Readout Solution Summary

- Select best reads for first alignment: Best reads=highest quality addresses!
- Perform alignment: Roughly 30 traces (reads) involved.
- Adjust sequence balance. Balancing also limits runlengths!
- Repeat while recruiting new traces.



Deletion Correction Through Balancing

C_{est}=Current estimate of the consensus sequence

- Initial alignment:

$C_{\textbf{est}}$ TTCACCCAAAAACCCGAAAACCGCTTCAGCGA
Trace1 TTCACCCCAAAACCGAAAACCGCTTCACGA
Trace2 TTCACCCAAAAACCCGAAAACCGCTTCAGCGA
Trace3 TTCACCCAAAAACCCGAAAACCGCTTCAGCGA

Deletion Correction Through Balancing

C_{est}=Current estimate of the consensus sequence

- Initial alignment:
  C_{est} TTCACCCAAAAACCCGAAAACCGCTTCAGCGA
  Trace1 TTCACCCCAAAACCGAAAACCGCTTCACGA
  Trace2 TTCACCCAAAAACCCGAAAACCGCTTCAGCGA
  Trace3 TTCACCCAAAAACCCGAAAACCGCTTCAGCGA
- After MSA
  C_{est} $T^2C^1A^1C^4A^4\ldots$
  Trace1 $T^2C^1A^1C^3A^4\ldots$
  Trace2 $T^2C^1A^1C^4A^4\ldots$
  Trace3 $T^2C^1A^1C^3A^5\ldots$

Deletion Correction Through Balancing

C_{est}=Current estimate of the consensus sequence

- Initial alignment:

  C_{est} TTCACCCAAAAACCCGAAAACCGCTTCAGCGA
  Trace1 TTCACCCCAAAACCGAAAACCGCTTCACGA
  Trace2 TTCACCCAAAAACCCGAAAACCGCTTCAGCGA
  Trace3 TTCACCCAAAAACCCGAAAACCGCTTCAGCGA

- After MSA

  C_{est} $T^2C^1A^1C^4A^4\ldots$
  Trace1 $T^2C^1A^1C^3A^4\ldots$
  Trace2 $T^2C^1A^1C^4A^4\ldots$
  Trace3 $T^2C^1A^1C^3A^5\ldots$

- After Balancing

  C_{est} $T^2C^1A^1C^3A^4\ldots$
  Trace1 $T^2C^1A^1C^3A^4\ldots$
  Trace2 $T^2C^1A^1C^3A^4\ldots$
  Trace3 $T^2C^1A^1C^3A^5\ldots$

Reading with Nanopores

Consensus may still have errors: Runlengths of As increase or decrease (protein-A interaction). Runlengths of Gs may form G quadruplexes.

Example Statistics:

Block (length)	Number of reads	Sequencing Coverage depth		Number of errors: (substitution, insertion, deletion)		
		Average	Maximum	Per read (average)	Consensus	
					Nanopolish	Our method
1 (1,000)	201	176.145	192	(107, 14, 63)	(14,32,5)	(0, 0, 2)
2 (1,000)	407	315.521	349	(123, 12, 70)	(75,99,40)	(0, 0, 0)
3 (1,000)	490	460.375	482	(80, 23, 42)	(10,45,0)	(0, 0, 0)
4 (1,000)	100	81.763	87	(69, 18, 37)	(1,54,1)	(0, 0, 0)
5 (1,000)	728	688.663	716	(88, 20, 48)	(4,45,3)	(0, 0, 0)
6 (1,000)	136	120.907	129	(79, 21, 42)	(390,102,61)	(0, 0, 0)
7 (1,000)	577	542.78	566	(83, 26, 41)	(3,31,3)	(0, 0, 0)
8 (1,000)	217	199.018	207	(83, 20, 46)	(18,51,1)	(0, 0, 0)
9 (1,000)	86	56.828	75	(60, 16, 30)	(404,92,54)	(0, 0, 0)
10 (1,000)	442	396.742	427	(91, 18, 52)	(388,100,59)	(0, 0, 0)
11 (1,000)	114	101.826	110	(79, 23, 42)	(16,23,18)	(0, 0, 0)
12 (1,000)	174	162.559	169	(94, 23, 50)	(14,59,1)	(0, 0, 0)
13 (1,060)	378	352.35	366	(88, 26, 44)	(7,55,4)	(0, 0, 0)
14 (1,000)	222	189.918	203	(69, 22, 34)	(15,34,3)	(0, 0, 0)
15 (1,000)	236	222.967	232	(92, 24, 45)	(15,46,2)	(0, 0, 0)
16 (1,000)	198	182.99	195	(103, 16, 61)	(15,62,4)	(0, 0, 1)
17 (880)	254	240.273	250	(77, 19, 42)	(359,95,44)	(0, 0, 0)

Homopolymer Codes

- Definition. The integer sequence of a vector $\mathbf{x} \in \mathbb{F}_4^n$ is the sequence of the length of the runs in $\mathbf{x}$.

  Example: $\mathbf{x} = (0, 0, 1, 3, 3, 2, 1, 1) \rightarrow I(\mathbf{x}) = (2, 1, 2, 1, 2)$.
  The alternating sequence is the sequence of symbols in $\mathbf{x}$, with all runs set to one. For above $\mathbf{x}$, we have $S(\mathbf{x}) = (0, 1, 3, 2, 1)$.

Homopolymer Codes

- Definition. The integer sequence of a vector $\mathbf{x} \in \mathbb{F}_4^n$ is the sequence of the length of the runs in $\mathbf{x}$.

  Example: $\mathbf{x} = (0,0,1,3,3,2,1,1) \rightarrow I(\mathbf{x}) = (2,1,2,1,2)$.
  The alternating sequence is the sequence of symbols in $\mathbf{x}$, with all runs set to one. For above $\mathbf{x}$, we have $S(\mathbf{x}) = (0,1,3,2,1)$.

- Suppose that $\mathbf{C}_H(n,t)$ is a code that can correct up to t substitution errors. Let

$$\mathbf{C}(n,t) = \{\mathbf{x} \in \mathbb{F}_4^n : I(\mathbf{x}) \bmod 2 \in \mathbf{C}_H(n,t)\}.$$

  The code $\mathbf{C}(n,t)$ can correct up to t asymmetric (decreasing) run-preserving deletions.

Homopolymer Codes

- Definition. The integer sequence of a vector $\mathbf{x} \in \mathbb{F}_4^n$ is the sequence of the length of the runs in $\mathbf{x}$.

  Example: $\mathbf{x} = (0,0,1,3,3,2,1,1) \rightarrow I(\mathbf{x}) = (2,1,2,1,2)$.
  The alternating sequence is the sequence of symbols in $\mathbf{x}$, with all runs set to one. For above $\mathbf{x}$, we have $S(\mathbf{x}) = (0,1,3,2,1)$.

- Suppose that $\mathbf{C}_H(n,t)$ is a code that can correct up to t substitution errors. Let

$$\mathbf{C}(n,t) = \{\mathbf{x} \in \mathbb{F}_4^n : I(\mathbf{x}) \bmod 2 \in \mathbf{C}_H(n,t)\}.$$

  The code $\mathbf{C}(n,t)$ can correct up to t asymmetric (decreasing) run-preserving deletions.

- Related to sticky deletions [B90's], [DA05].

Readout Time

- The sequencing time was ~ 10 hours, but only "junk" reads generated after ~ 6 hours.

Readout Time

- The sequencing time was ~ 10 hours, but only "junk" reads generated after ~ 6 hours.
- How long shall we sequence for?

Readout Time

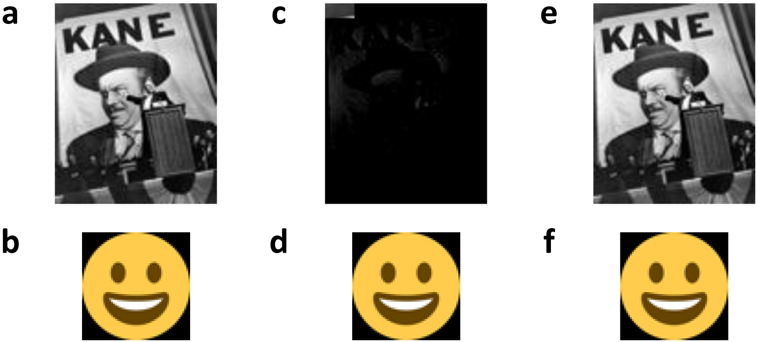
- The sequencing time was ~ 10 hours, but only "junk" reads generated after ~ 6 hours.
- How long shall we sequence for?
- Definition. The k-deck of a sequence $\mathbf{x}$ is the multiset of all subsequences of length k of $\mathbf{x}$.

Readout Time

- The sequencing time was ~ 10 hours, but only "junk" reads generated after ~ 6 hours.
- How long shall we sequence for?
- Definition. The k-deck of a sequence **x** is the multiset of all subsequences of length k of **x**.
- Hybrid reconstruction: One is given a small number M of "long" asymmetric traces ($o(n)$ deletions). What is the smallest value of k for a k-deck that along with the M long traces ensures unique reconstruction?

Resolving the Portability Problem

Example images of Citizen Kane poster (1946) and Smiley. Only three deletions left after iterative alignment.
Error-free decoding is possible with coding efficiency 88%



Native DNA-Based Data Storage

Resolving the Synthesis Problem?

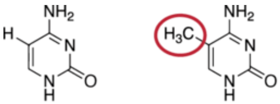
- Our system: Store information in native DNA (e.g., *E. coli*). How?

Resolving the Synthesis Problem?

- Our system: Store information in native DNA (e.g., *E. coli*). How?
- Content cannot be changed easily: –ATGCC– has to remain –ATGCC–.

Resolving the Synthesis Problem?

- Our system: Store information in native DNA (e.g., *E. coli*). How?
- Content cannot be changed easily: –ATGCC– has to remain –ATGCC–.
- Structure? Symbol alphabet?



Cytosine **methylated Cytosine**

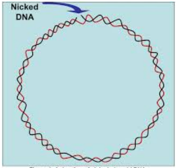
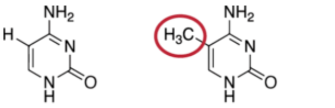


Figure 1: Animation of nicked plasmid DNA.
A nick can be enzymatically induced or caused by shearing during plasmid preparation.

Resolving the Synthesis Problem?

- Our system: Store information in native DNA (e.g., *E. coli*). How?
- Content cannot be changed easily: –ATGCC– has to remain –ATGCC–.
- Structure? Symbol alphabet?
- Nicking using programmable DNA-guided artificial restriction enzymes (in collaboration with Zhao lab).



Cytosine methylated Cytosine

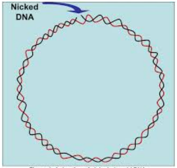
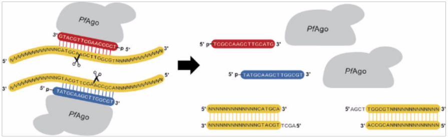
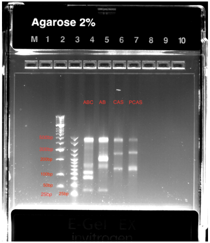


Figure 1: Animation of nicked plasmid DNA.
A nick can be enzymatically induced or caused by shearing during plasmid preparation.

Think of Punch Cards...

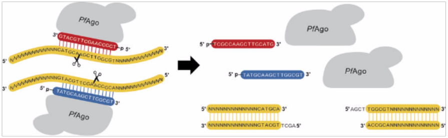
- Our approach: Do not nick or nick sense or antisense strand (ternary alphabet).

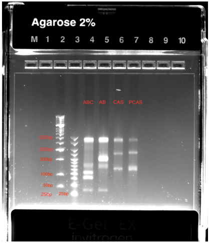




Think of Punch Cards...

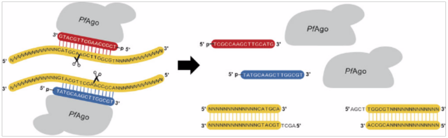
- Our approach: Do not nick or nick sense or antisense strand (ternary alphabet).
- Reuse nicking enzyme on a large number of DNA "registers": PfAgo DNA-guided enzyme.

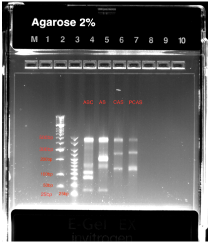




Think of Punch Cards...

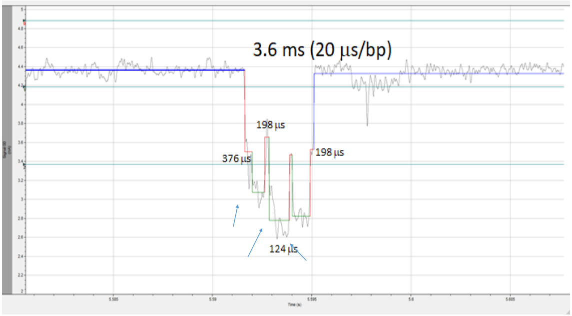
- Our approach: Do not nick or nick sense or antisense strand (ternary alphabet).
- Reuse nicking enzyme on a large number of DNA "registers": PfAgo DNA-guided enzyme.
- How do we know that we have the "right nicks"?





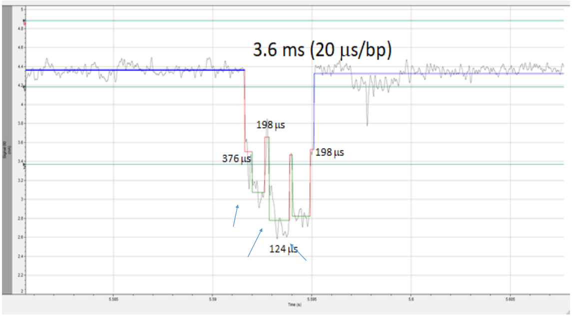
How do we Read the Nicks?

- Detect Nicks via Illumina Sequencers: Denature nicked DNA, Sanger sequence (expensive).



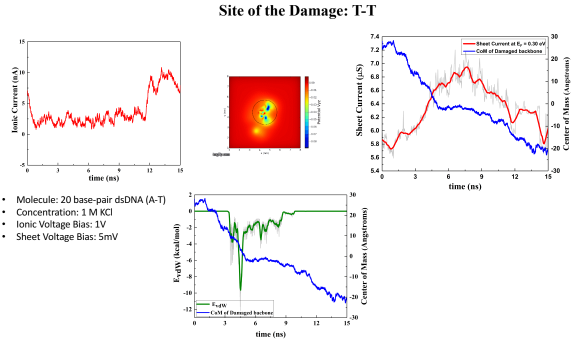
How do we Read the Nicks?

- Detect Nicks via Illumina Sequencers: Denature nicked DNA, Sanger sequence (expensive).
- Detect Nicks with Nanopores? Collaboration with Radenovic lab, EPFL.



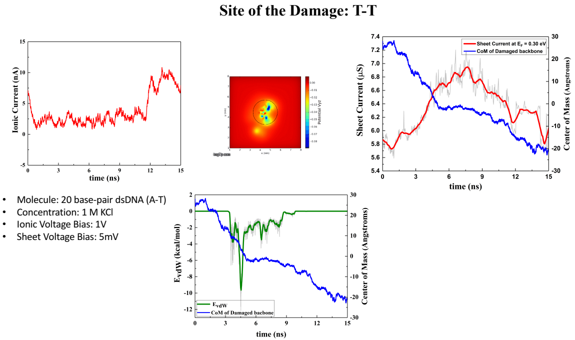
How do we Read the Nicks?

- Detect Nicks via Illumina Sequencers: Denature nicked DNA, Sanger sequence (expensive).



How do we Read the Nicks?

- Detect Nicks via Illumina Sequencers: Denature nicked DNA, Sanger sequence (expensive).
- Simulate Nicks with Nanopores? Collaboration with Leburton lab, EPFL.



Sources of Errors

- Erroneous Nicking: Shifts in the positions of the nick (nicking window), missing nicks (deletions).

Sources of Errors

- Erroneous Nicking: Shifts in the positions of the nick (nicking window), missing nicks (deletions).
- Erroneous Readout: Shifts in the positions of the nick (nicking window), missing and inserted nicks (deletions and insertions).

Sources of Errors

- Erroneous Nicking: Shifts in the positions of the nick (nicking window), missing nicks (deletions).
- Erroneous Readout: Shifts in the positions of the nick (nicking window), missing and inserted nicks (deletions and insertions).
- Instability Errors: Do not want to nick one strand exclusively, as it may cause the backbone to break down.

New Coding Solutions I

- Ternary Codes for Swap and Deletion correction: Related to codes in the Damerau distance [GYM18].

New Coding Solutions I

- Ternary Codes for Swap and Deletion correction: Related to codes in the Damerau distance [GYM18].
- Hard to Accommodate Instability Issues: Revert to codewords described in terms of sets.

New Coding Solutions I

- Ternary Codes for Swap and Deletion correction: Related to codes in the Damerau distance [GYM18].
- Hard to Accommodate Instability Issues: Revert to codewords described in terms of sets.
- Formal Definition: The *nicking codebook* is a set of subsets $S_i \subset [n]$, $i = 1, \ldots, M$, of fixed size k such that for any $i \neq j$, one has $|S_i \cap S_j| \leq s$, where M and s are code design parameters.

New Coding Solutions I

- Ternary Codes for Swap and Deletion correction: Related to codes in the Damerau distance [GYM18].
- Hard to Accommodate Instability Issues: Revert to codewords described in terms of sets.
- Formal Definition: The *nicking codebook* is a set of subsets $S_i \subset [n]$, $i = 1, \ldots, M$, of fixed size k such that for any $i \neq j$, one has $|S_i \cap S_j| \leq s$, where M and s are code design parameters.
- Introduced by Babai and Frankl: Let q be a prime power and $1 \leq s \leq k \leq q$. Set $n = kq$. Let ξ be a primitive element of $\mathbb{F}_q$ and $\mathcal{A} = \{0, 1, \xi, \ldots, \xi^{k-2}\}$. Then $|\mathcal{A}| = k$. For each polynomial $f \in \mathbb{F}_q[x]$, define

$$A_f := \{(a, f(a)) : a \in A\}.$$

We also have $|A_f| = k$. Let

$$\mathcal{C}(k, q, s) := \{A_f : f \in \mathbb{F}_q[x], \deg(f) \leq s - 1\}.$$

Then $\mathcal{C}(k, q, s)$ is a collection of q^s k-subsets of the set $X := \mathcal{A} \times \mathbb{F}_q$ and satisfies the property that every two sets intersect at at most $s - 1$ elements.

Set Discrepancy Theory

- Set discrepancy problem [Spencer'85, Lovasz'86]: Given a set of m subsets $\{A_1, \ldots, A_m\}$ of fixed size k over a ground set $[n-1]$, find a labeling $\ell : [n-1] \to \{-1, +1\}$ which minimizes

$$\max_{1 \le i \le m} \Big| \sum_{x \in A_i} \ell(x) \Big|, \text{ i.e.}$$

$$\min_{\ell : [n-1] \to \{-1,+1\}} \max_{1 \le i \le m} \Big| \sum_{x \in A_i} \ell(x) \Big|.$$

Set Discrepancy Theory

- Set discrepancy problem [Spencer'85, Lovasz'86]: Given a set of m subsets $\{A_1, \ldots, A_m\}$ of fixed size k over a ground set $[n-1]$, find a labeling $\ell : [n-1] \to \{-1, +1\}$ which minimizes

$$\max_{1 \le i \le m} \Big| \sum_{x \in A_i} \ell(x) \Big|, \text{ i.e.}$$

$$\min_{\ell : [n-1] \to \{-1,+1\}} \max_{1 \le i \le m} \Big| \sum_{x \in A_i} \ell(x) \Big|.$$

- In our case, the optimal mapping $\ell : [n-1] \to \{-1, +1\}$ ensures balance of nicks.

Set Discrepancy Theory

- Set discrepancy problem [Spencer'85, Lovasz'86]: Given a set of m subsets $\{A_1, \ldots, A_m\}$ of fixed size k over a ground set $[n-1]$, find a labeling $\ell : [n-1] \to \{-1, +1\}$ which minimizes

$$\max_{1 \le i \le m} \Big| \sum_{x \in A_i} \ell(x) \Big|, \text{ i.e.}$$

$$\min_{\ell : [n-1] \to \{-1,+1\}} \max_{1 \le i \le m} \Big| \sum_{x \in A_i} \ell(x) \Big|.$$

- In our case, the optimal mapping $\ell : [n-1] \to \{-1, +1\}$ ensures balance of nicks.
- Babai-Frankl sets can be shown to have zero discrepancy!

Set Discrepancy Theory

- Set discrepancy problem [Spencer'85, Lovasz'86]: Given a set of m subsets $\{A_1, \ldots, A_m\}$ of fixed size k over a ground set $[n-1]$, find a labeling $\ell : [n-1] \to \{-1, +1\}$ which minimizes

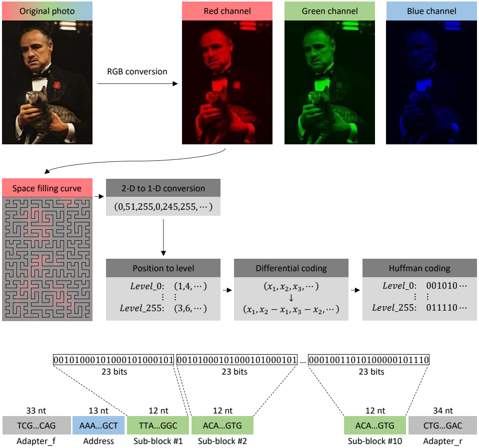
$$\max_{1 \le i \le m} \Big| \sum_{x \in A_i} \ell(x) \Big|, \text{ i.e.}$$

$$\min_{\ell:[n-1]\to\{-1,+1\}} \max_{1 \le i \le m} \Big| \sum_{x \in A_i} \ell(x) \Big|.$$

- In our case, the optimal mapping $\ell : [n-1] \to \{-1, +1\}$ ensures balance of nicks.
- Babai-Frankl sets can be shown to have zero discrepancy!
- Can extend the results further using Steiner systems.

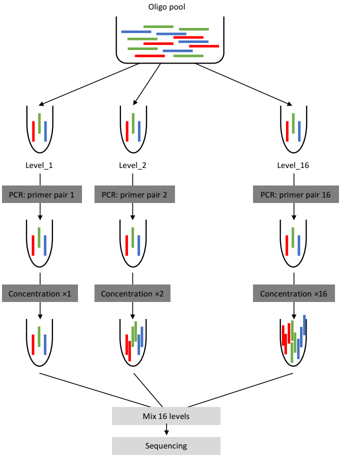
Other Directions: Concentration Based Coding

- Concentration based encoding: Image processing in DNA.



Other Directions: Concentration Based Coding

- Concentration based encoding: Image processing in DNA.

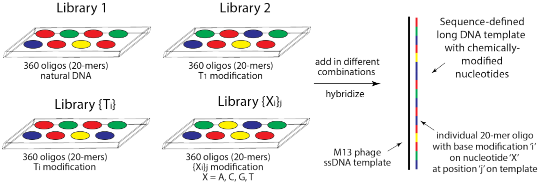


Other Directions: Concentration Based Coding

- Concentration based encoding: Image processing in DNA.

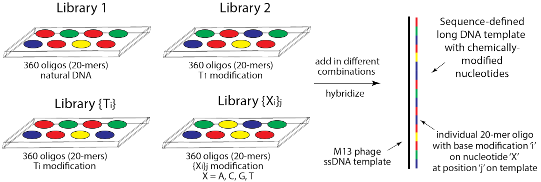
Other Directions: Enlarging the Code Alphabet

- Enlarging the code alphabet: Nonstructural, chemical modifications (with Schroeder lab).

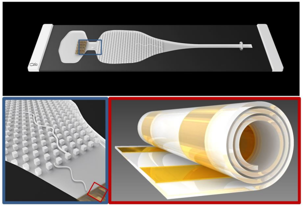


Other Directions: Enlarging the Code Alphabet

- Enlarging the code alphabet: Nonstructural, chemical modifications (with Schroeder lab).

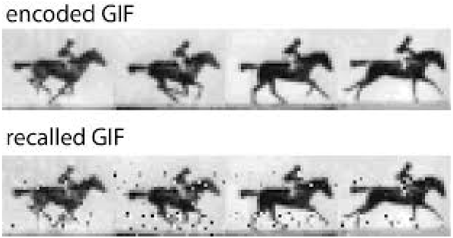


- Integration with nanoelectronics: Changing random access approaches (with Li lab).



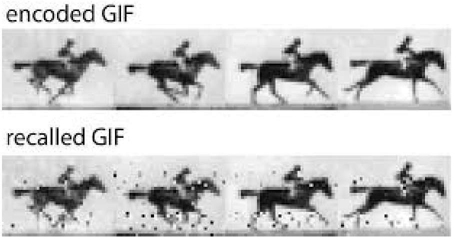
DNA Storage in Living Cells, Nature, 2017

- Low-density storage using CRISPR-Cas, *E. coli*: Church et al., 2017.



DNA Storage in Living Cells, Nature, 2017

- Low-density storage using CRISPR-Cas, *E. coli*: Church et al., 2017.

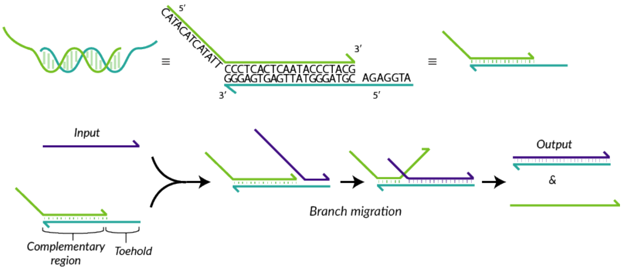


- Fountain DNA Storage: Erlich et al., 2017 (Reed-Solomon in Grass et.al.: oligos treated as symbols over a large alphabet, redundancy at the oligo level).

Native DNA-Based Computing

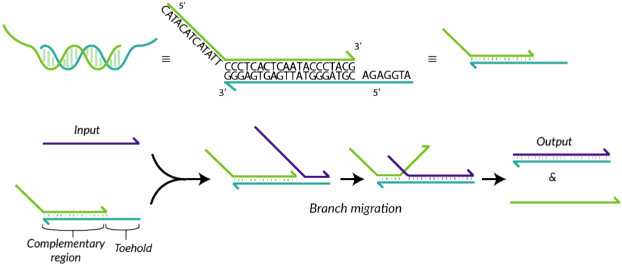
Some Computing Schemes

- Nick displacement: New computing paradigm akin to strand displacement.



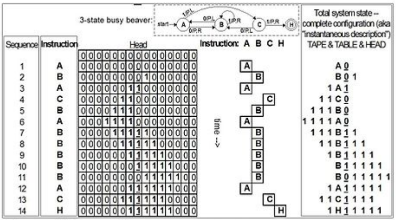
Some Computing Schemes

- Nick displacement: New computing paradigm akin to strand displacement.
- Easy-to-implement operations: Incrementing/decrementing, comparison (with Soloveichik lab).



Some Computing Schemes

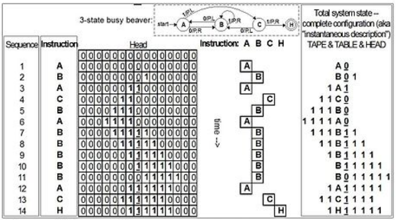
- Minsky's register machine: Need Incrementing/decrementing, comparison only. Computation power of Turing machines.



Progress of the computation (state-trajectory) of a 3-state busy beaver

Some Computing Schemes

- Minsky's register machine: Need Incrementing/decrementing, comparison only. Computation power of Turing machines.
- Large number of registers: Copies of the same native genomic sequence (e.g., *E. coli*).



Progress of the computation (state-trajectory) of a 3-state busy beaver

Acknowledgment

Collaborators:
Alvaro Hernaez, Keck Center, UIUC.
Jean-Pierre Lebourton, UIUC.
Xiuling Li, UIUC.
Jian Ma, CMU.
Aleksandra Radenovic, EPFL.
Marc Riedel, UMN.
Charles Schroeder, UIUC.
David Soloveichik, UT Austin.
Huimin Zhao, UIUC.
Students:
Hoang Dau, Ryan Gabrys, Hossein Tabatabei Yazdi, Yongbo Yuan.

Funding:
CIA
UIUC SRI
NSF
DARPA Molecular Informatics