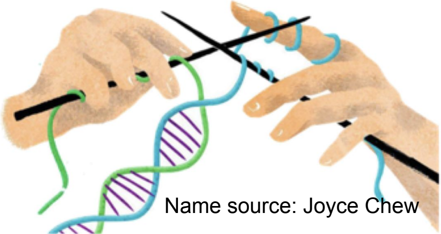
Cal-DISKS

Calvin/Berkeley Distributed *k*-mer Sketcher

Elizabeth Koning
Advised by Joel Adams



Name source: Joyce Chew

Genetic Sequences

… are strings of A, C, G, T characters representing DNA

- Used in Computational Biology problems, including:
  - Metagenomics (DNA of many different organisms)
  - Genome assembly (properly ordering genomic sequences)
- Involves massive amounts of data
  - Human genome: 3 billion base pairs (>750 megabytes)
  - Often many terabytes of data for metagenomes

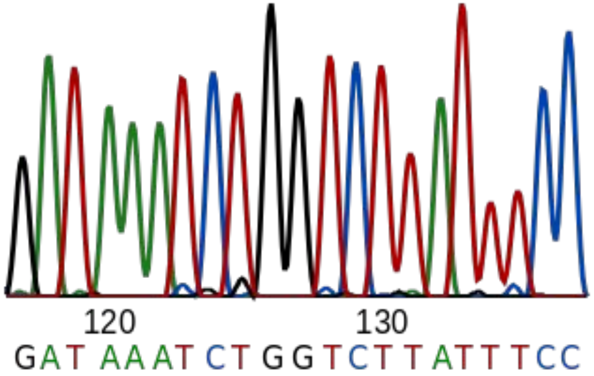


Image: EMBL-EBI

k-mer Counting

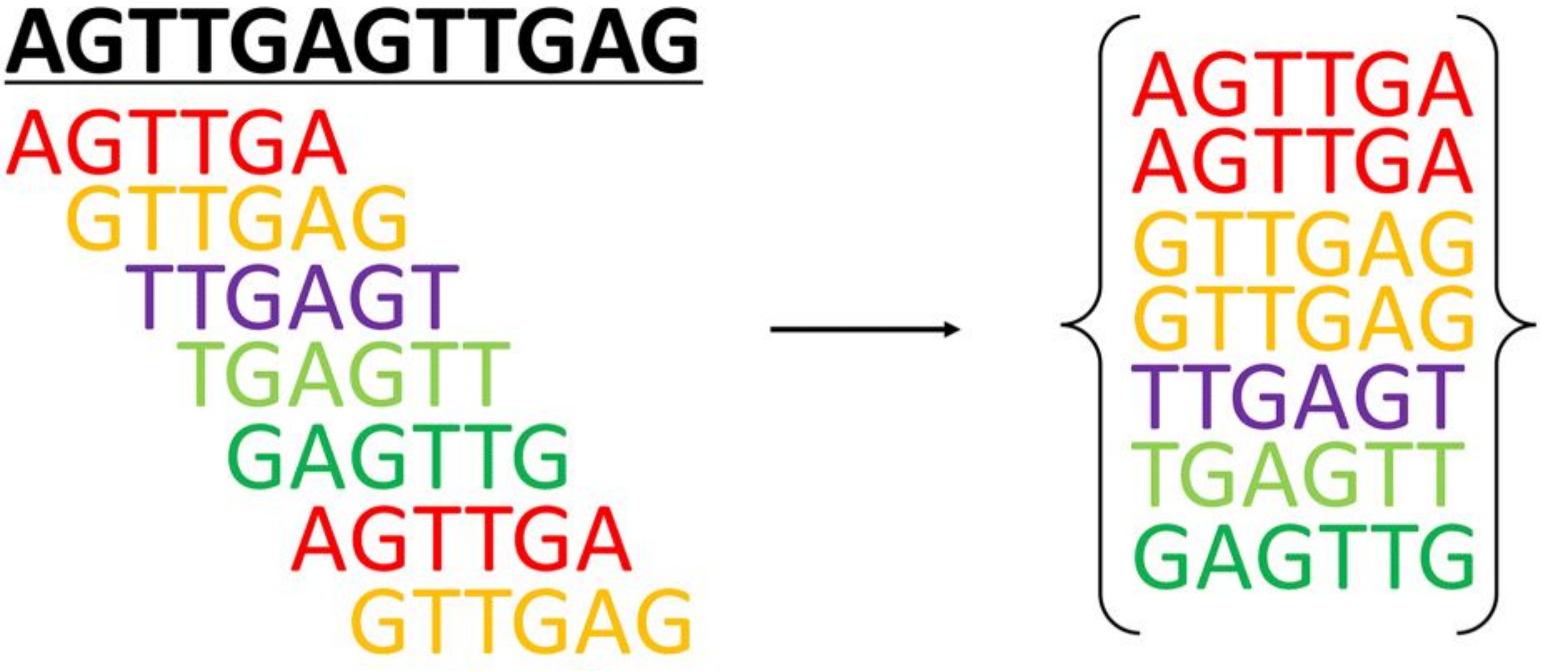


Image: doi:10.1186/s12864-019-5467-x

k-mer Usage

How similar are two DNA sequences?

Two applications:

1. *Alignment*: Find initial k-mer matches before performing nucleotide-level comparison of the genetic sequences. (If you need very accurate analysis)
2. *Similarity*: Computing the percentage of similarity between two sequences without performing computationally-intensive nucleotide-level alignment. (If you just need a rough estimation)

Problem for both applications:

We have many, many k-mers

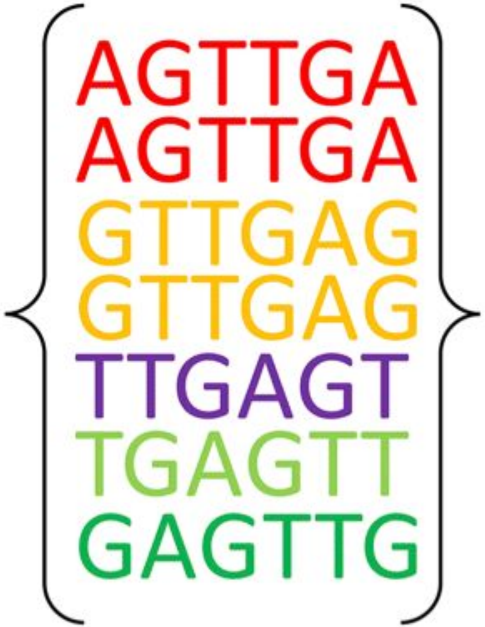


Image: doi:10.1186/s12864-019-5467-x

k-mer Sketcher

Strategy to reduce the number of k-mers:

- Select a subset of a sequence’s k-mers
  - ➔ The subset is a “sketch” of its sequence and is orders of magnitude smaller than the full representation

Tradeoff:

- Smaller sketch→ less storage and processing time
- Larger sketch → higher accuracy

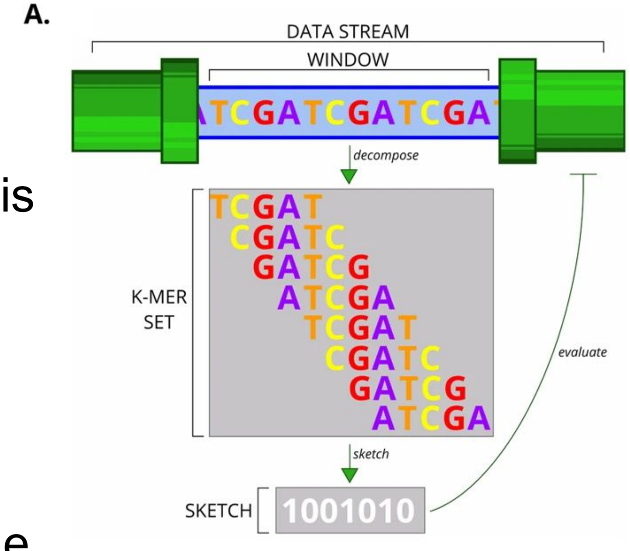


Image: doi:10.1186/s13059-019-1809-x

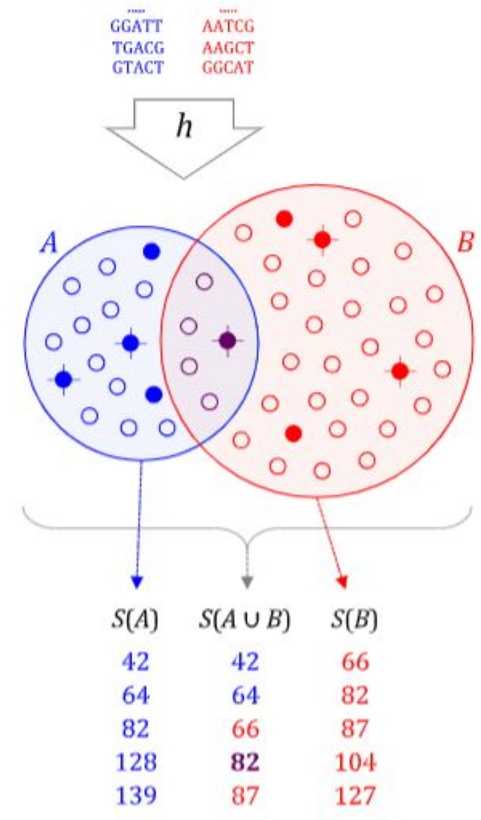
Sketching Technique

Use MinHash data structure

- Use a hash function to hash each k-mer
- Store set of minimum k-mer hash-values

Benefit:

- Sketches are smaller to store and faster to compare to other sketches
- Results are more accurate than a random sample



$S(A)$	$S(A \cup B)$	$S(B)$
42	42	66
64	64	82
82	66	87
128	**82**	104
139	87	127

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \approx \frac{|S(A \cup B) \cap S(A) \cap S(B)|}{|S(A \cup B)|}$$

Image: doi:10.1186/s13059-016-0997-x

Calculating a Discard Threshold

- Problem:
  - Storing and discarding most k-mers
- Goal:
  - Predict which k-mers will be discarded before storage

Calculating a Discard Threshold

- Problem:
  - Storing and discarding most k-mers
- Goal:
  - Predict which k-mers will be discarded before storage

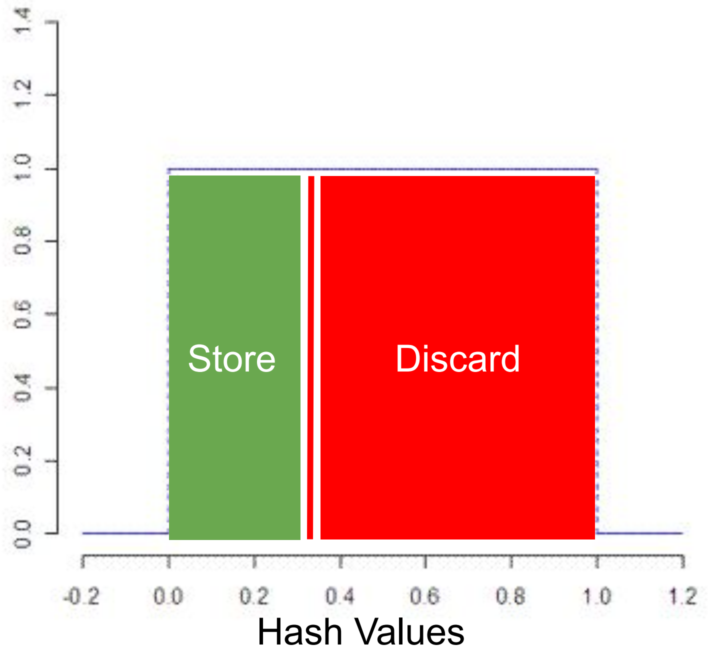
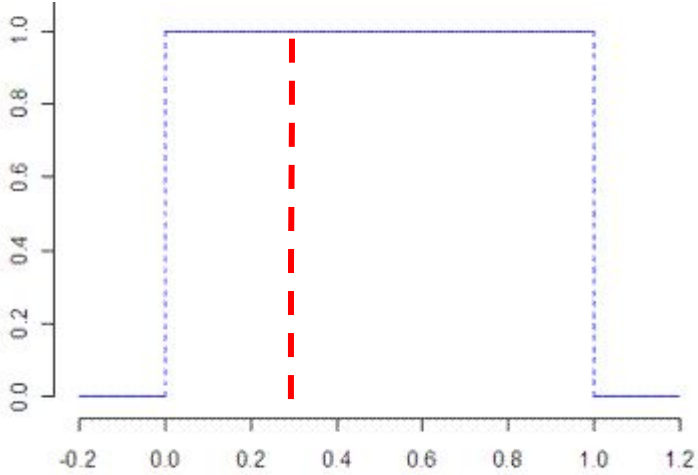


Image: https://www.statisticshowto.com/uniform-distribution

What Threshold Value To Use?

Calculate *threshold* as *expected cutoff* for hash values

$$threshold = \frac{desired\ \#\ of\ values}{unique\ values} * max\ hash\ value - min\ hash\ value + 1$$



Unique Values Calculation

To calculate the expected number of unique *k*-mers:

$$E(u) = n - n * \frac{(n-1)^k}{n^k}$$

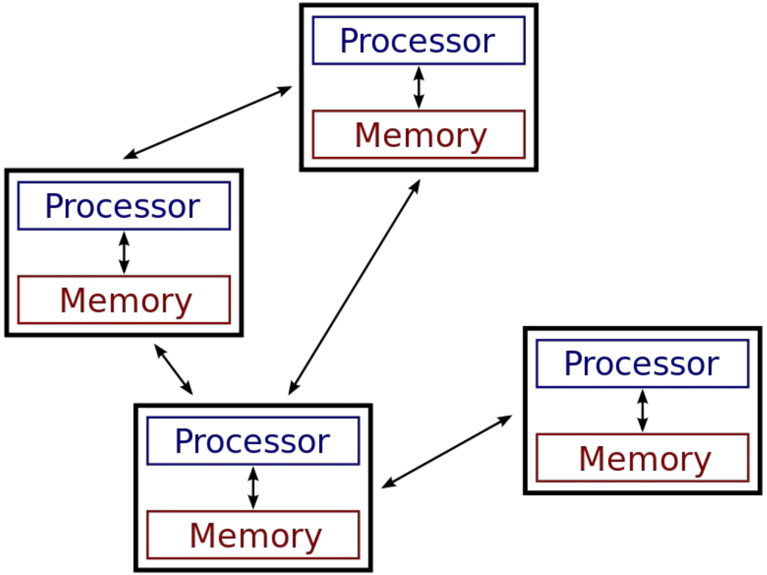
E(u) = expected unique k-mers

n = universe of k-mers

k = number of k-mers in the file

Distributed Computing

- Multiple processes run on separate computers
- Processes communicate via message passing (MPI)
- Communication is often run time bottleneck



Berkeley Supercomputer

Image: https://www.nersc.gov/news-publications/nersc-news/nersc-center-news/2016/cori-supercomputer-now-fully-installed-at-berkeley-lab/

Input File

File A

seq 1
CCACACCAAAGAG...

seq 2
GAGATTCAGCAATG...

seq 3
CTCGAAGAGATGGA...

seq 4
CGGCGTTAAGTTTA...

seq 5
CGCCGATAACCCCA...

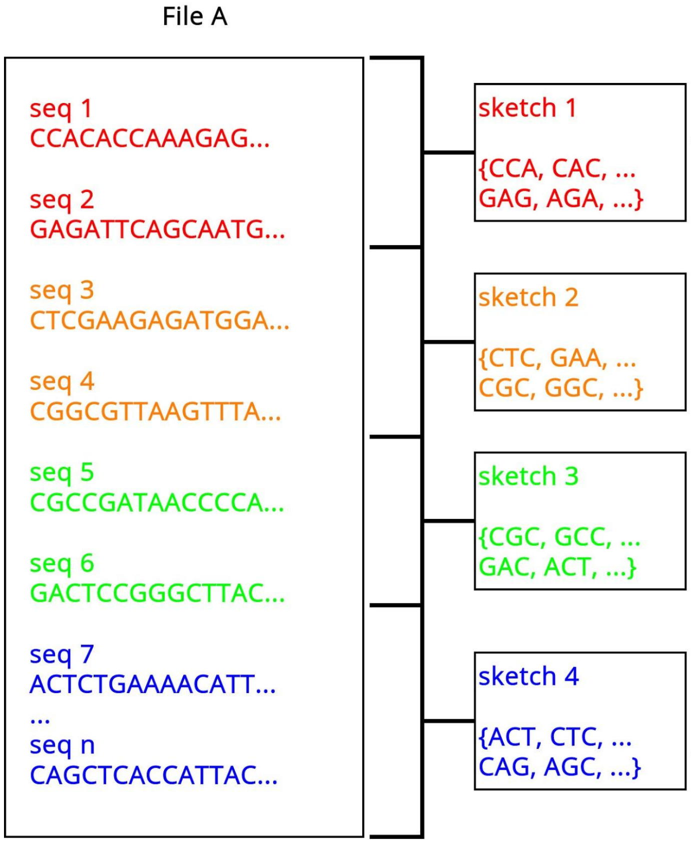
seq 6
GACTCCGGGCTTAC...

seq 7
ACTCTGAAAACATT...
...
seq n
CAGCTCACCATTAC...

DNA Sequencing Machine

- File of reads from DNA sequencing
- May be from one or multiple organisms

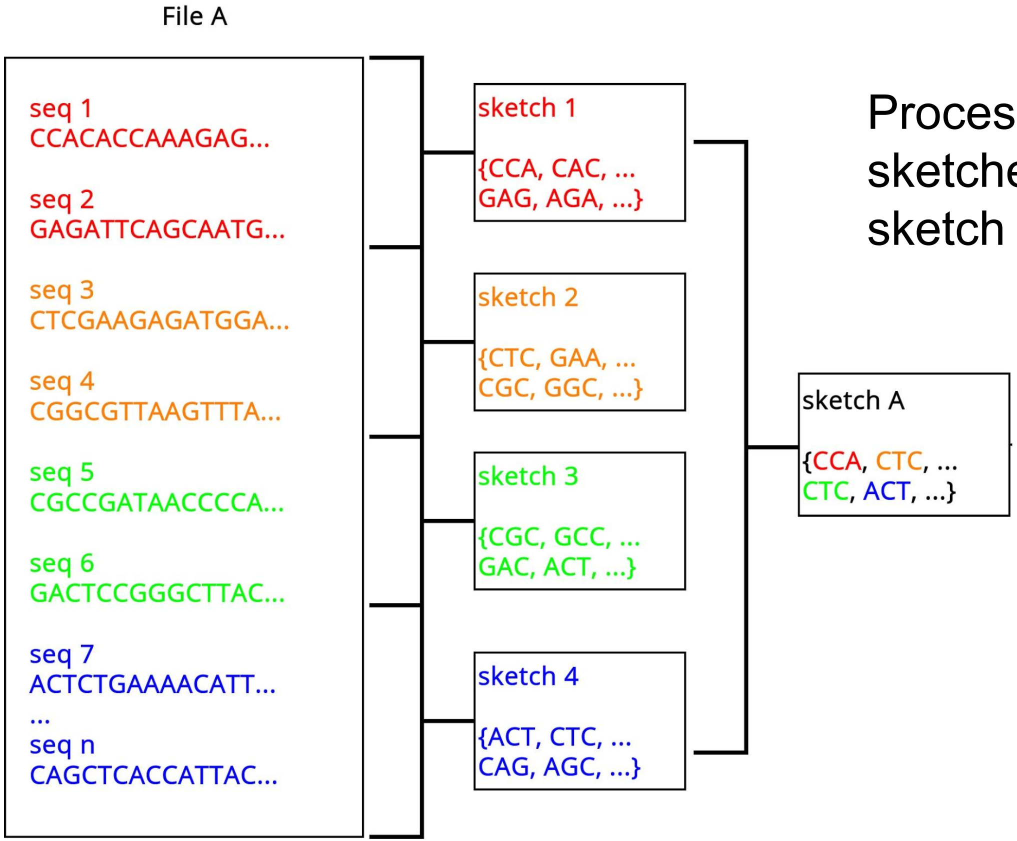
Parallel I/O using MPI



Each process simultaneously:

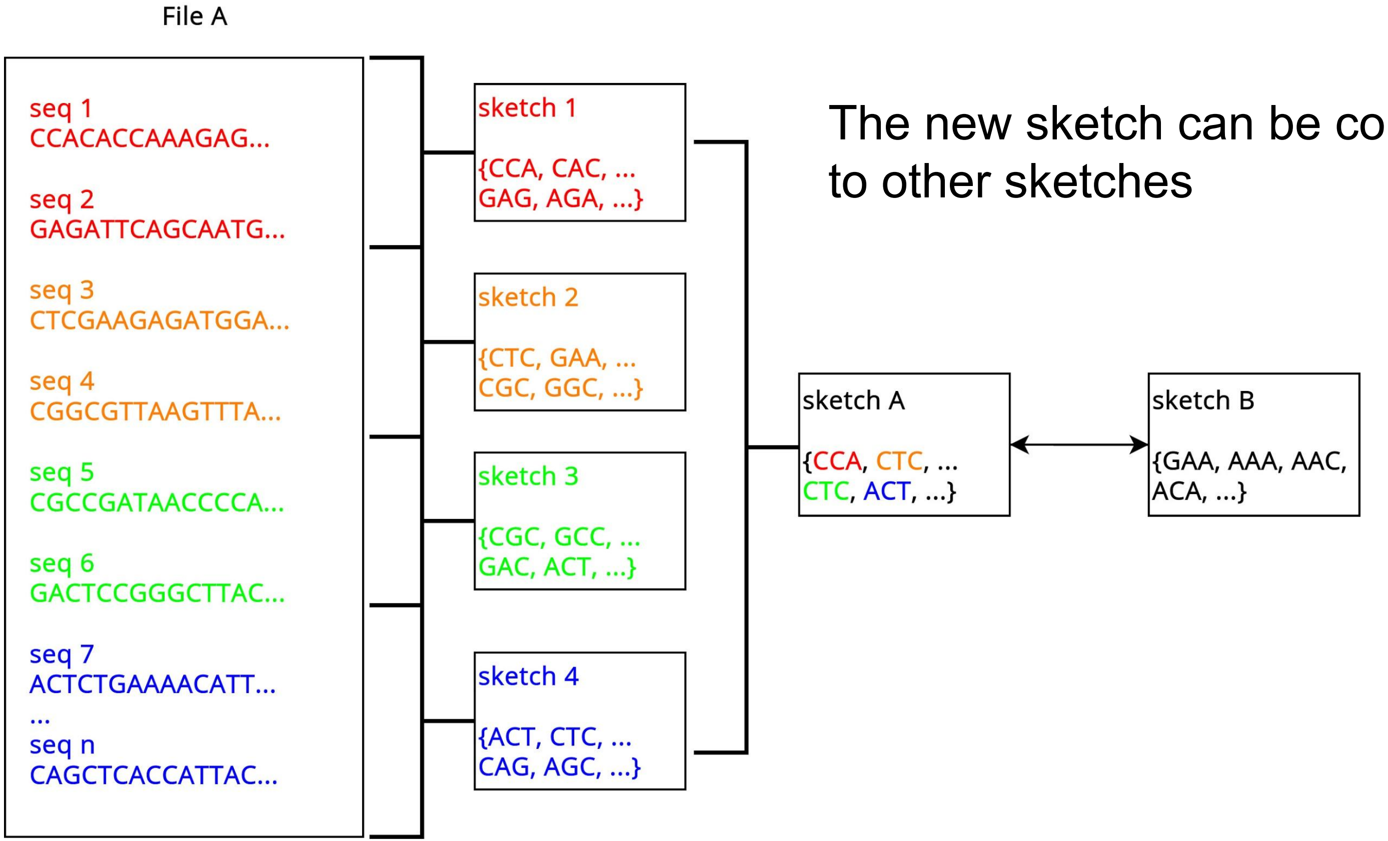
1. Reads a chunk of the file.
2. Creates a local sketch based on its chunk

Aggregating Sketches



Process 0 gathers the local sketches into a single global sketch

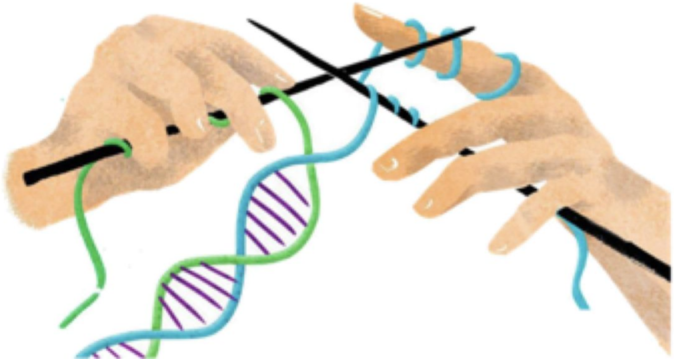
Parallel I/O using MPI



The new sketch can be compared to other sketches

Conclusions

- What I've done:
  - Threshold calculation
  - Parallel I/O
- What comes next:
  - Finalizing the sketch aggregation
  - Expected efficiency for large datasets



Thank you!

Giulia Guidi and Ben Brock
Aydın Buluç and Kathy Yelick
UC Berkeley

Joel Adams
Calvin University