Long Read Mapping & Sketching

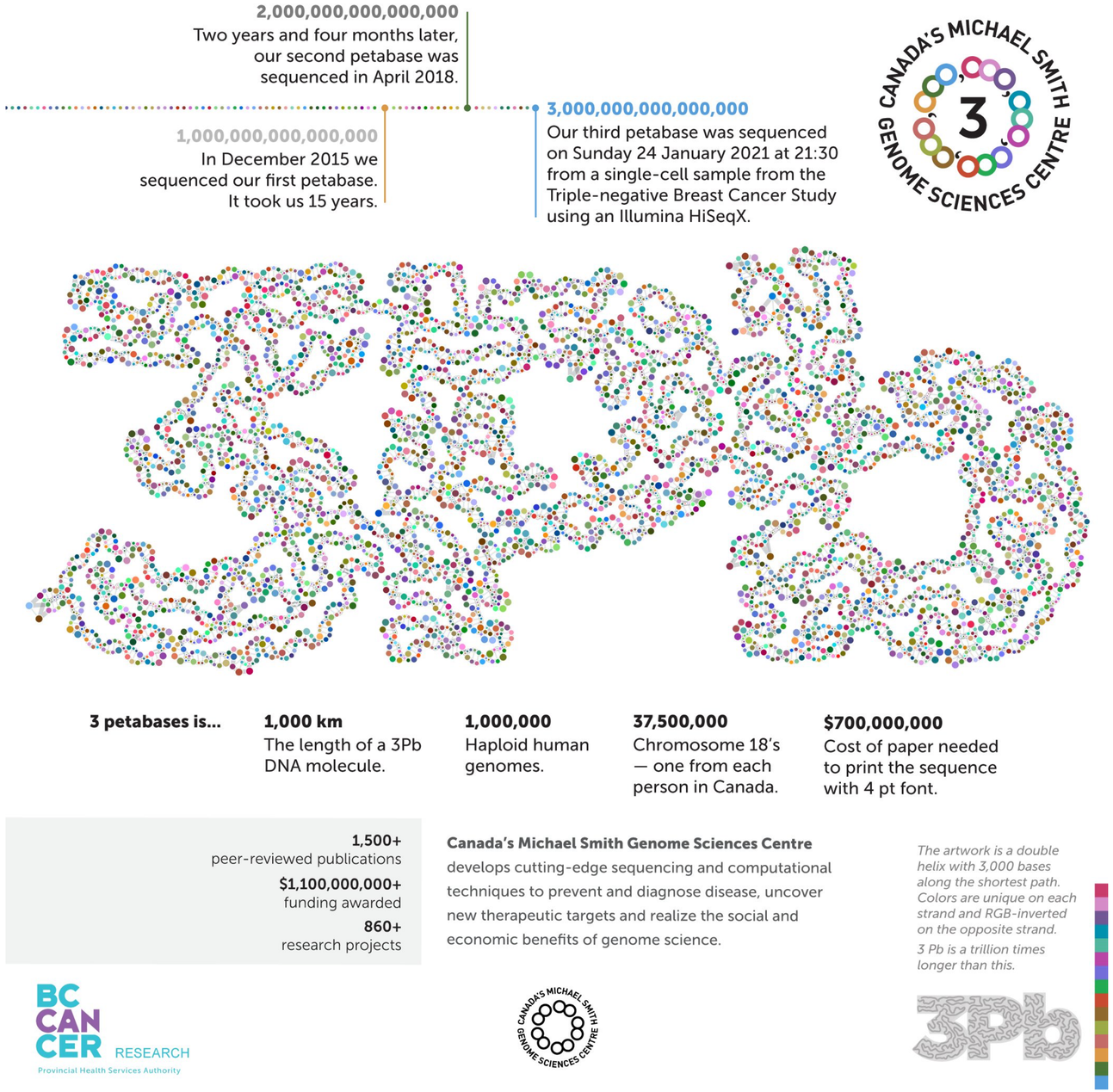
Alex Sweeten

October 4, 2023

Lecture 11: Applied Comparative Genomics

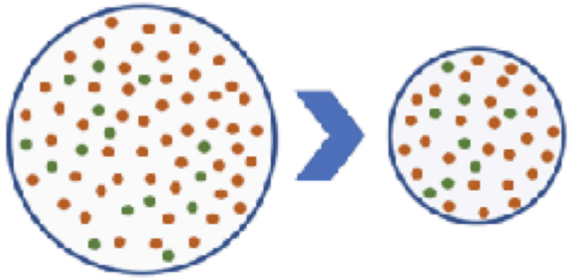
Motivation

- Genomics has a scaling problem
  - Genomes are large!
  - Polynomial time algorithms $O(n^2)$ or worse are often unusable
  - Linear or sub-linear is the ideal



Strategies

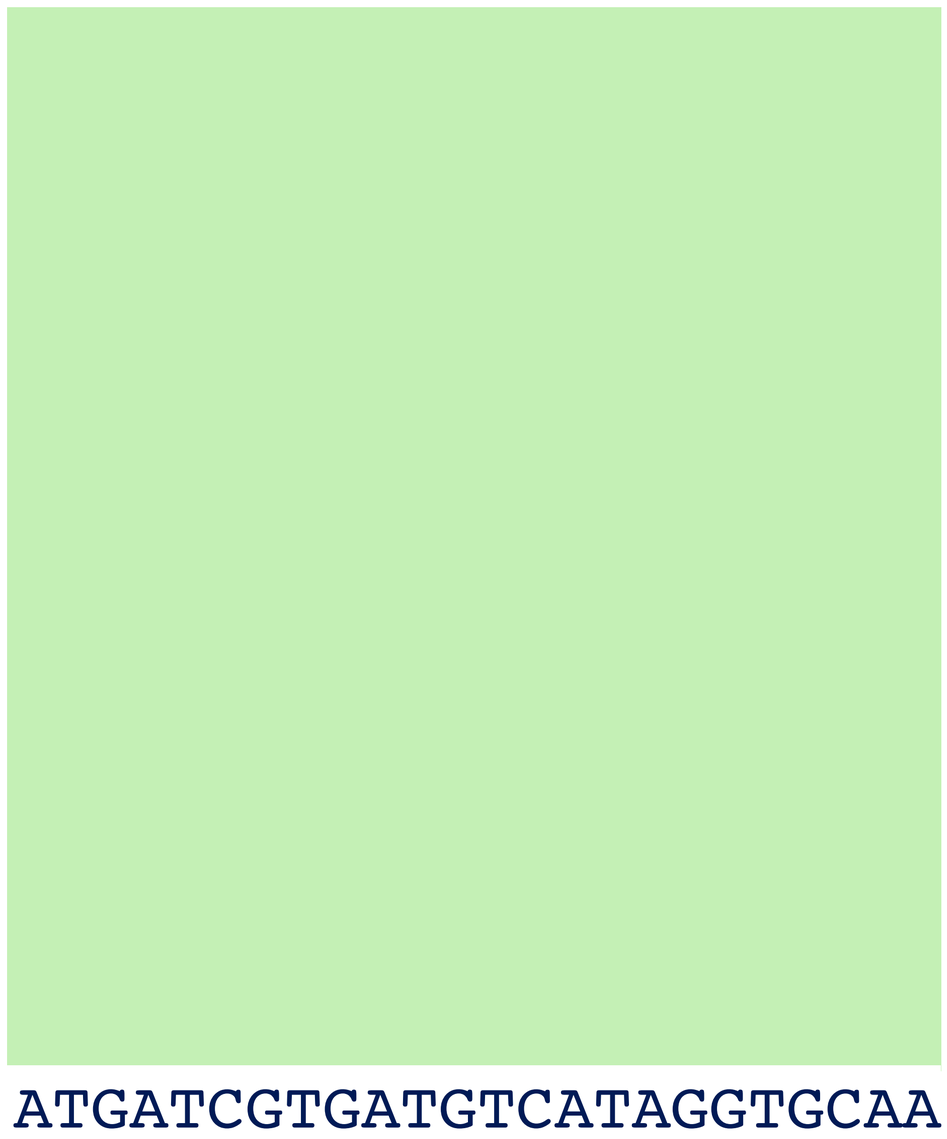
- Heuristics
  - Approximate solution instead of an exact solution
- Sketching
  - Use a “smart” subset of the original data



Sequence Alignment

- Global Alignment: Smith-Waterman (1970)
- Local Alignment: Needleman-Wunsch (1981)
- Dynamic programming
  - Fill in a m x n matrix
  - $O(n^2)$

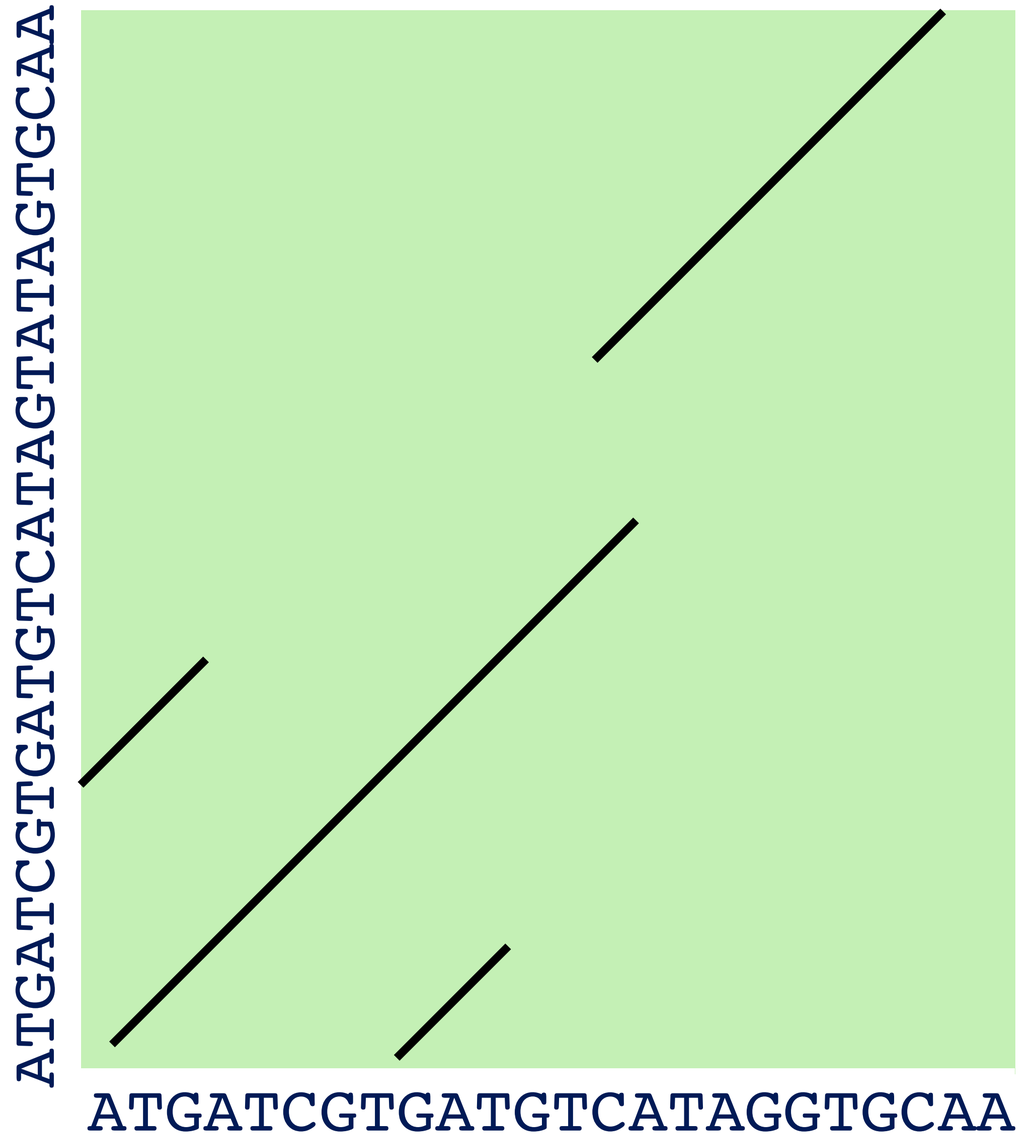
	A	G	C	T	-
A	0	5	5	5	3
G	5	0	4	5	3
C	5	4	0	5	3
T	5	5	5	0	3
-	3	3	3	3	∞



Sequence Alignment

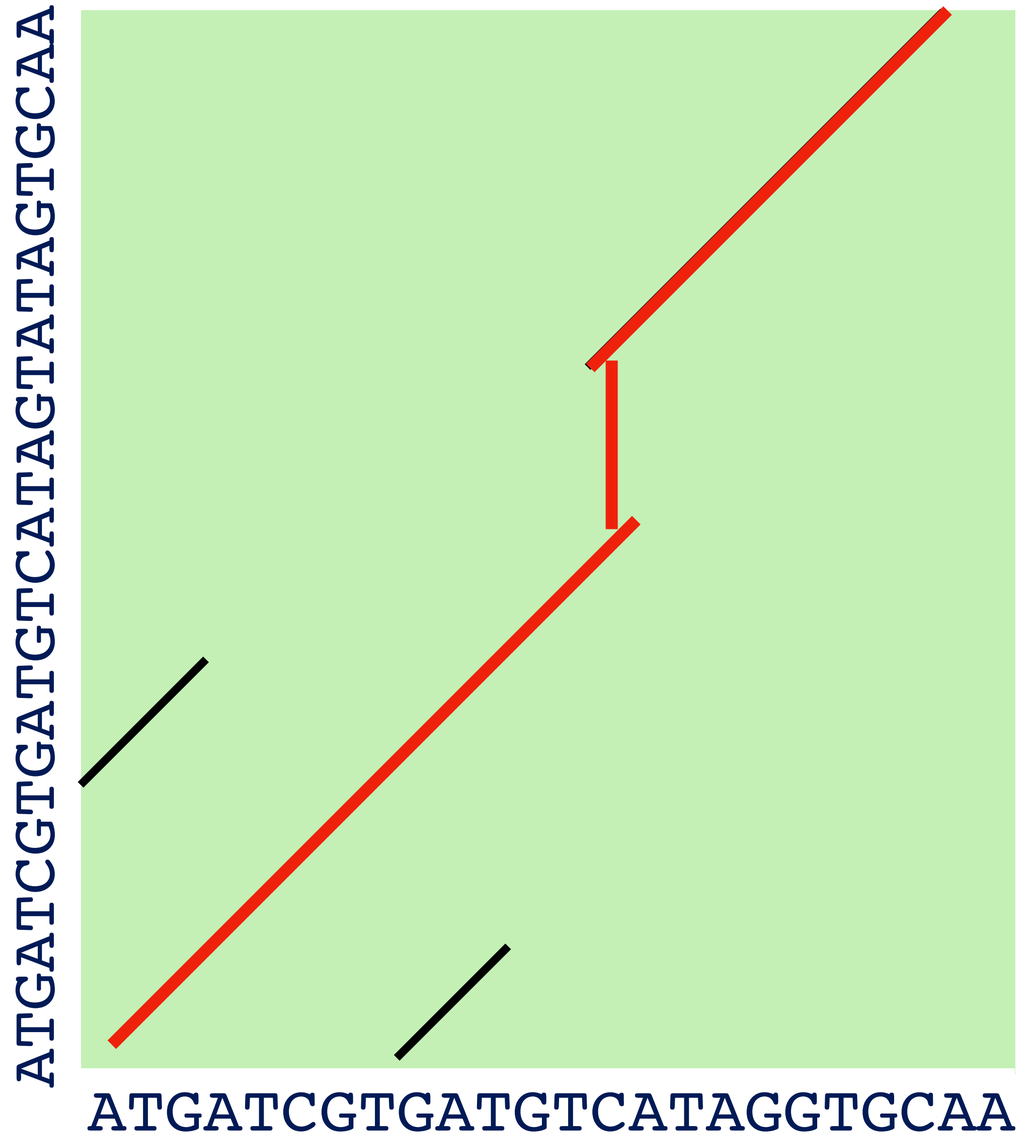
- Global Alignment: Smith-Waterman (1970)
- Local Alignment: Needleman-Wunsch (1981)
- Dynamic programming
  - Fill in a m x n matrix
  - $O(n^2)$

	A	G	C	T	-
A	0	5	5	5	3
G	5	0	4	5	3
C	5	4	0	5	3
T	5	5	5	0	3
-	3	3	3	3	∞



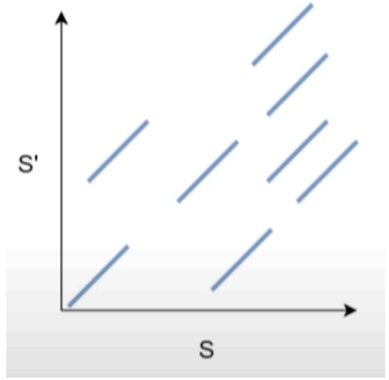
Sequence Alignment

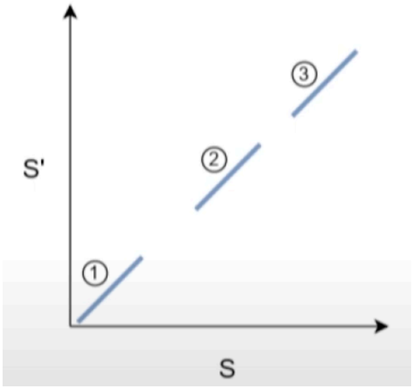
```
ATGATCGTGATGTCATAGTATAGTGCAA
||||||||||||||||||    ||||||
ATGATCGTGATGTCATAG----GTGCAA
```

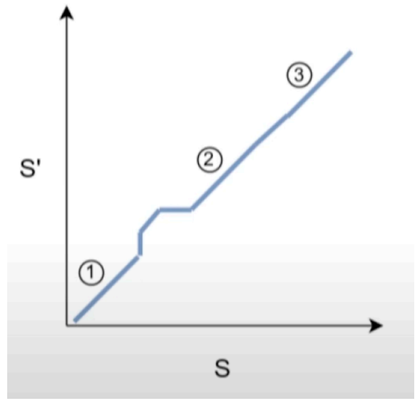


Seed Chain Extend

- Heuristic way to align sequences
- Choose *k*-mers as "seeds" that are shared between each sequence
- Chain seeds together to get "anchors"
- Extend anchors using sparse dynamic programming

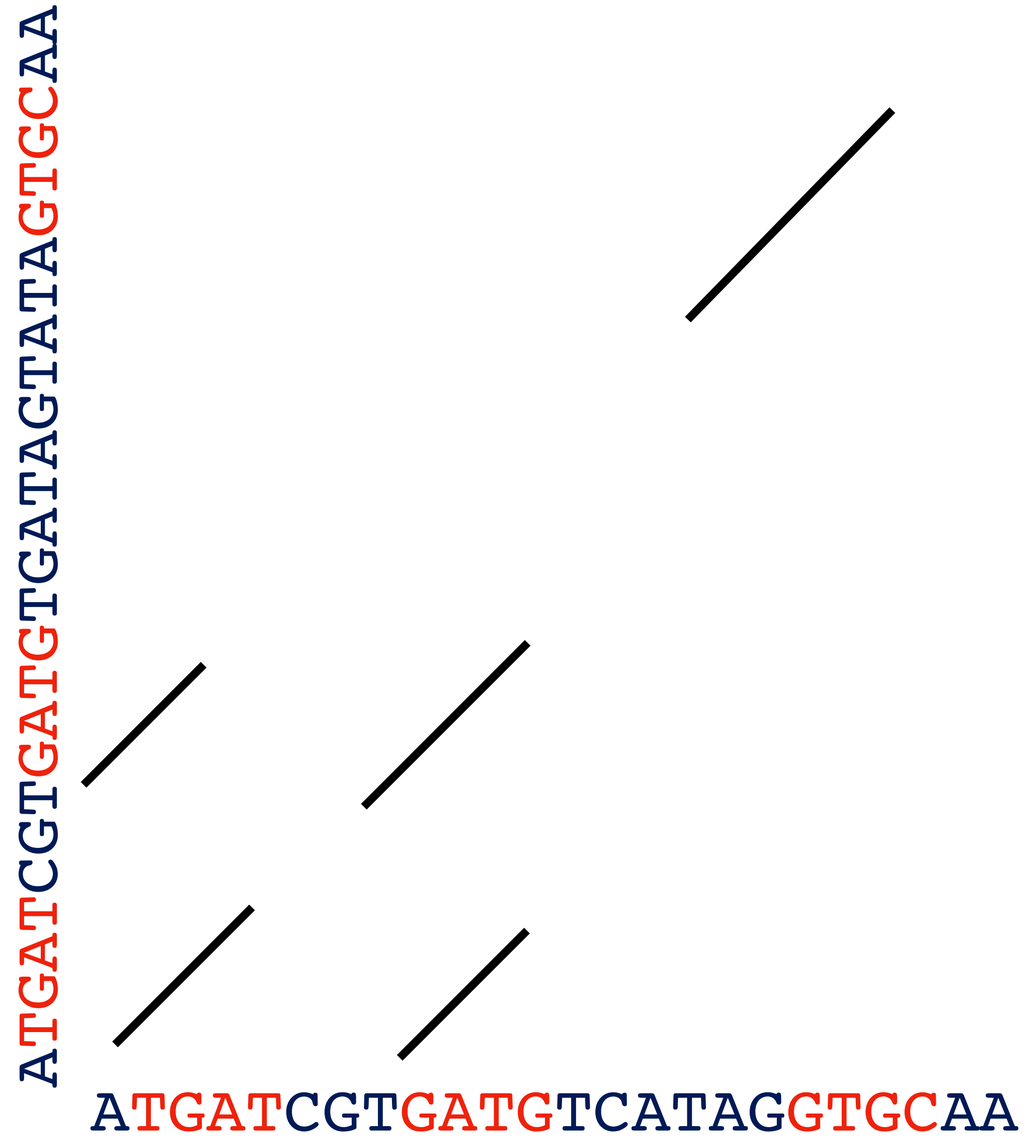






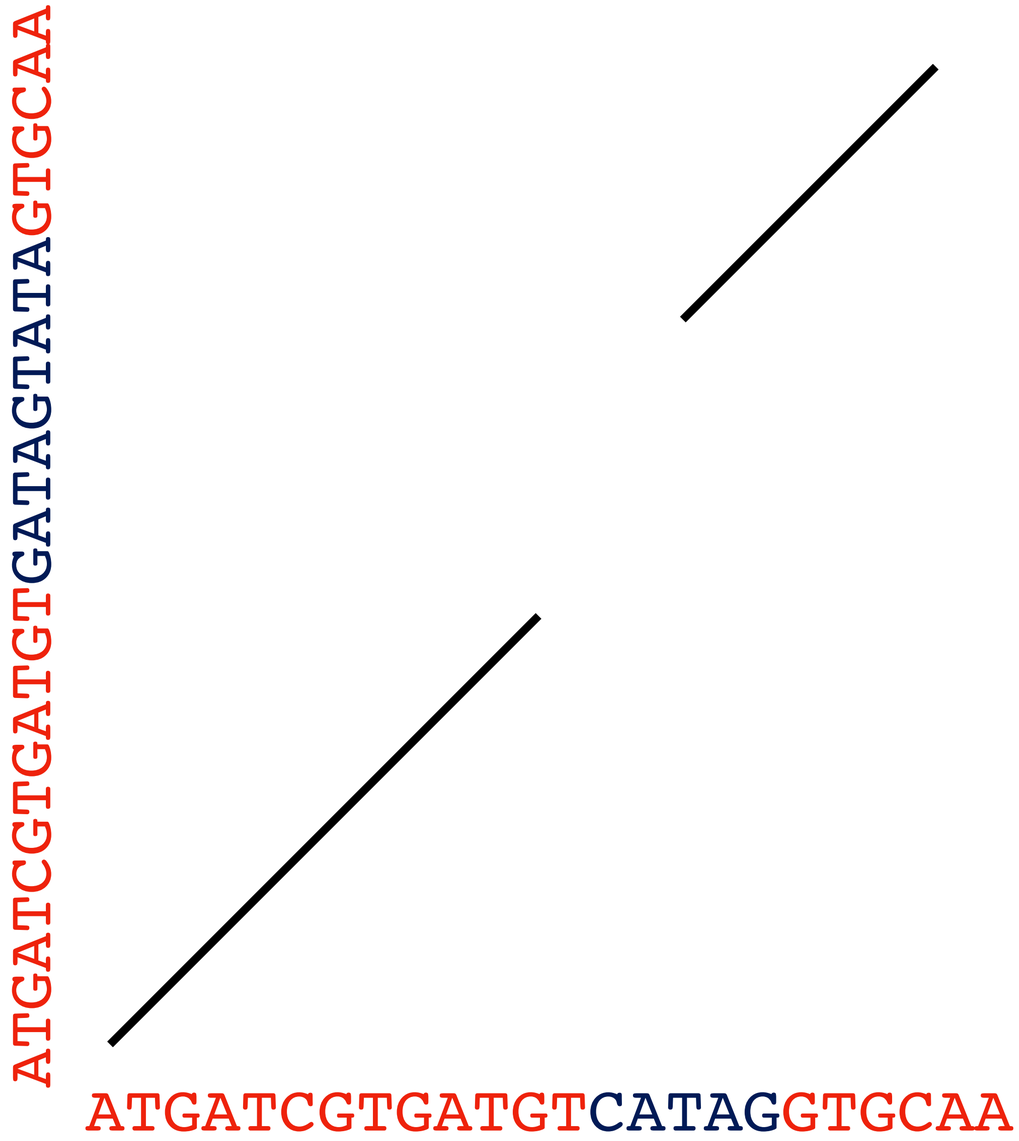
Seed Chain Extend

- Seeds:
  - TGAT
  - GATG
  - GTGC



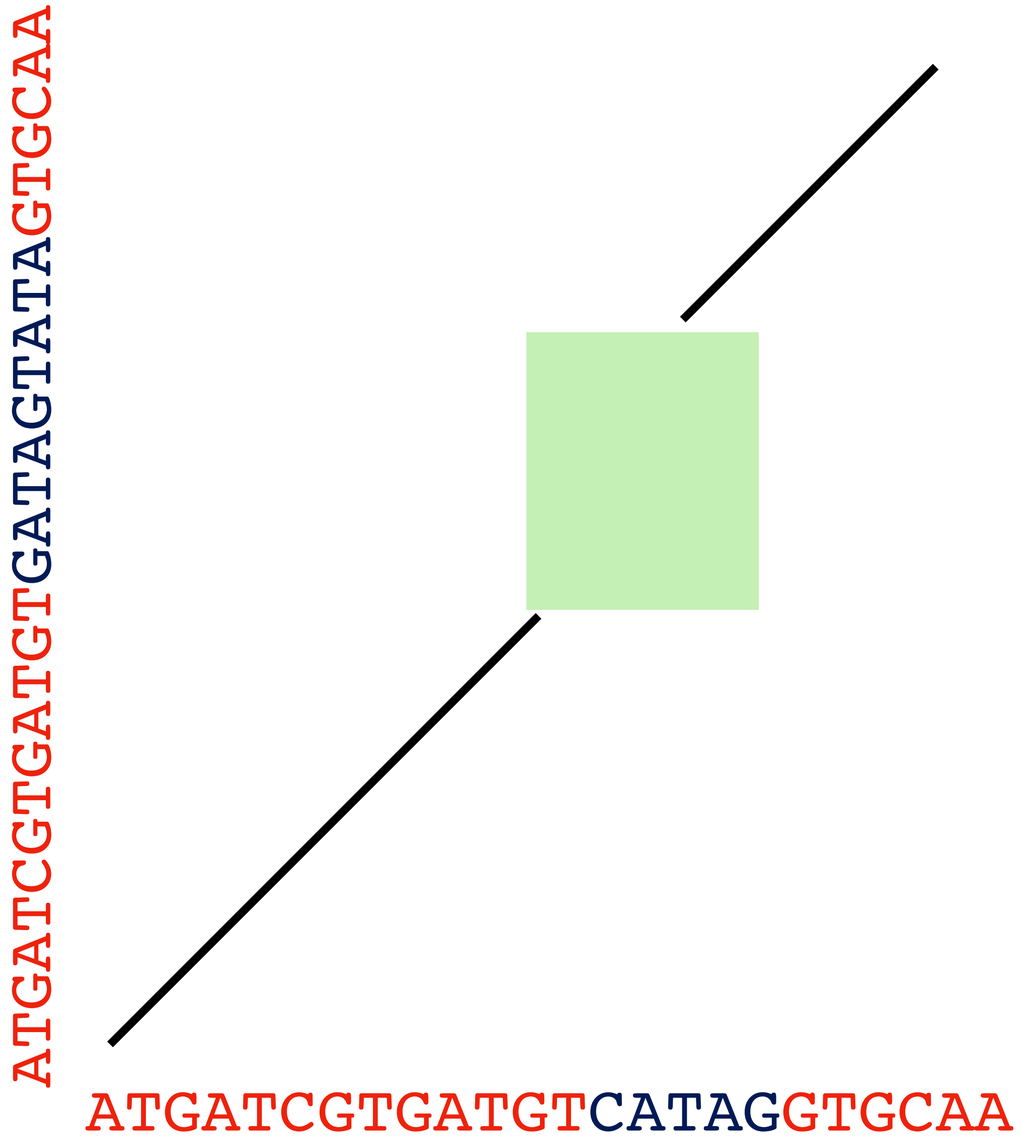
Seed **Chain** Extend

- Seeds:
  - TGAT
  - GATG
  - GTGC
- Chained Seeds:
  - ATGATCGTGATGT
  - GTGCAA



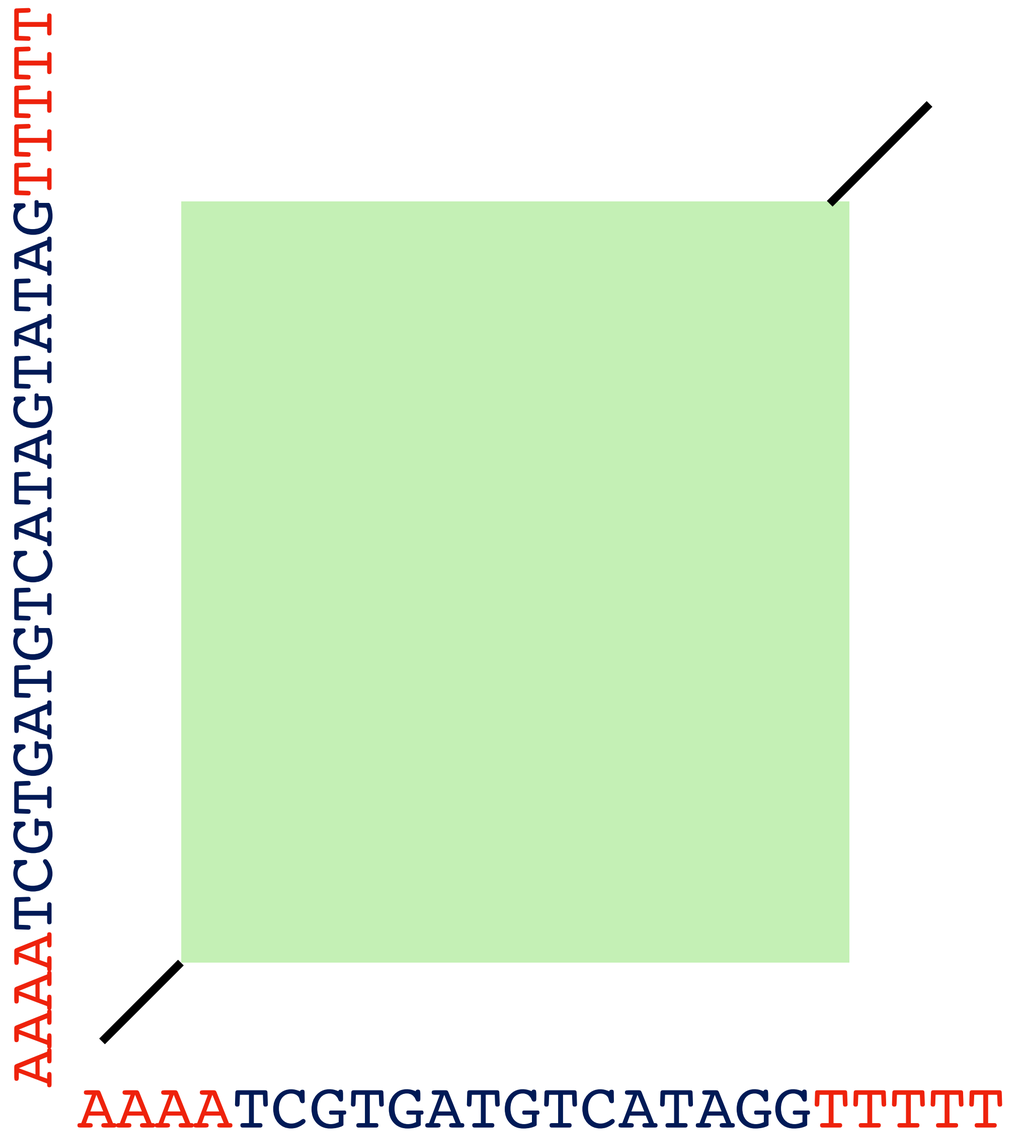
Seed Chain **Extend**

- Seeds:
  - TGAT
  - GATG
  - GTGC
- Chained Seeds:
  - ATGATCGTGATGT
  - GTGCAA



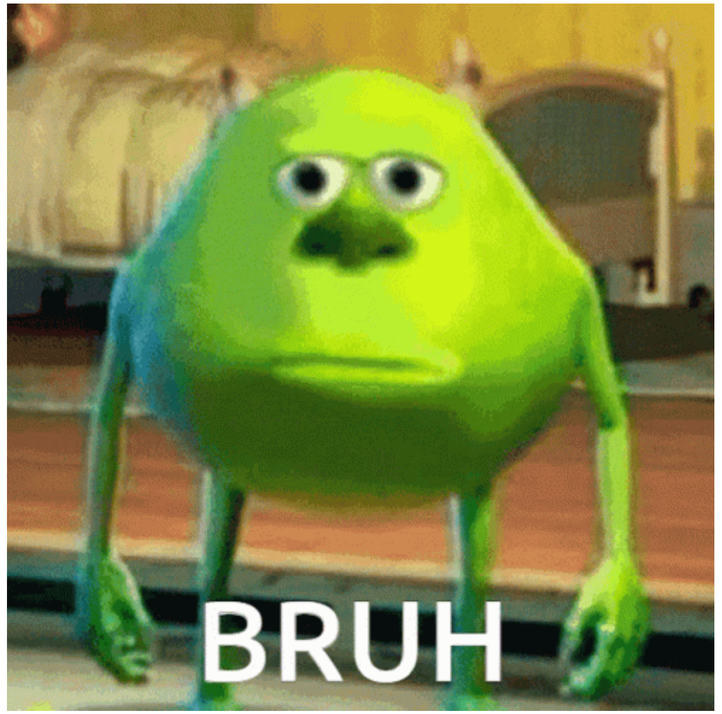
Runtime

- $O(n^2)$ in the worst case
- Impacting factors:
  - Bad seeding
  - Repetitive sequence
  - High divergence



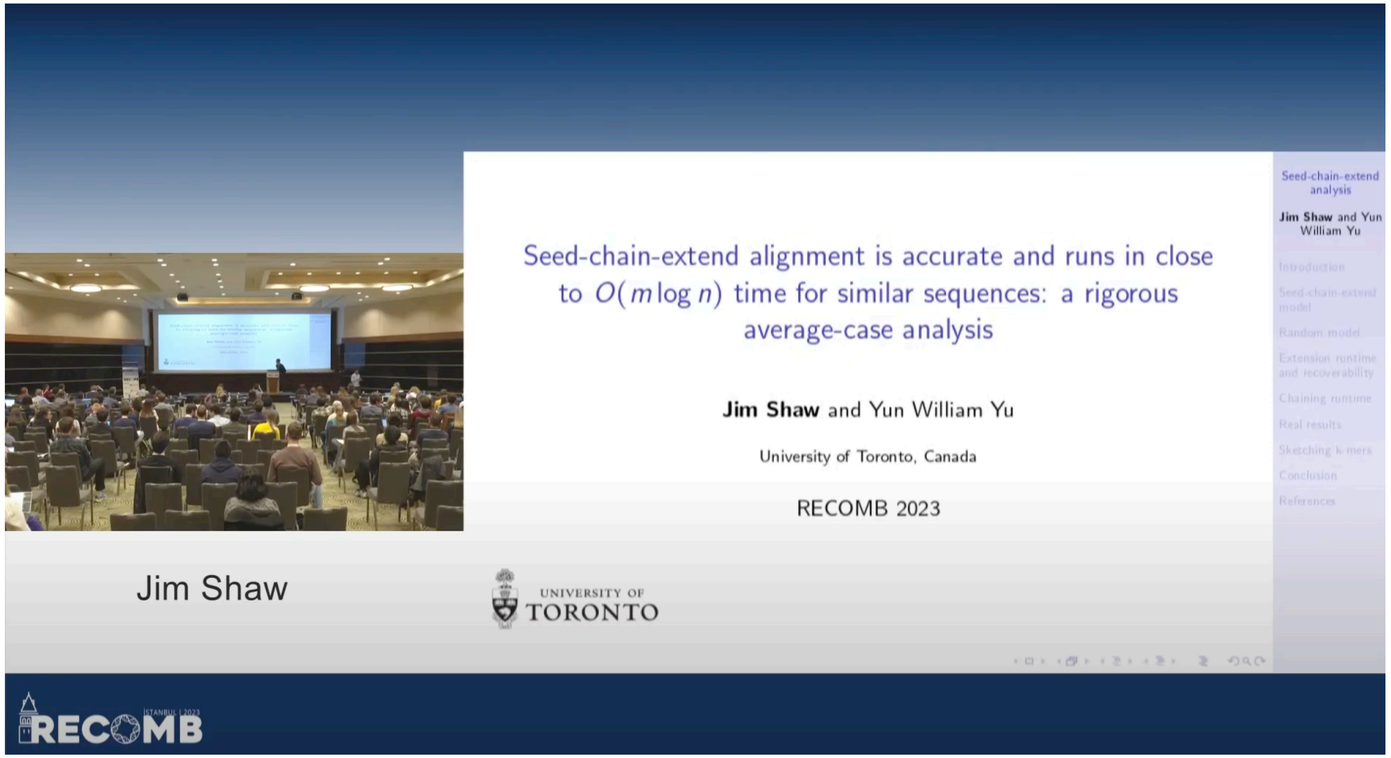
Runtime

- $O(n^2)$ in the worst case
- Impacting factors:
  - Bad seeding
  - Repetitive sequence
  - High divergence



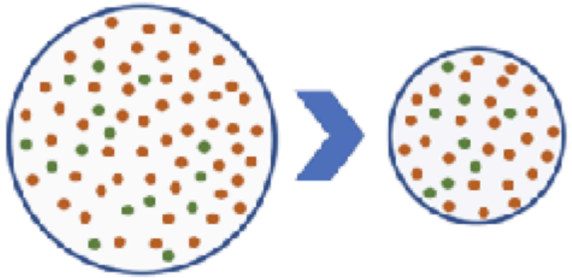
Runtime

- $O(n^2)$ in the worst case
- Doesn't explain good runtime in practice



Sketching

- "Smart" way to subsample a sequence
- Has guarantees on what data will be kept in the sketch
- Used to "seed" the alignment in seed-chain-extend



Winnowing

- Convert document into pieces with an assigned weight, seperate by weight
- Advantages:
  - Easy to implement
  - Guaranteed spacing between pieces

Winnowing: Local Algorithms for Document Fingerprinting (Schleimer et al., 2003)

Minimizers

- Given a tuple (*w*,*k*), *w* > *k*, output the "smallest" *k*-mer in w
- Smallest:
  - Lexicographic
  - Random (hash function)
- Important to use canonical *k*-mers

(*10*,*7*)

GTGATTACAT

GTGATTA

TGATTAC

GATTACA

ATTACAT

Reducing storage requirements for biological sequence comparison (Roberts et al., 2004)

Minimizers

- Given a tuple (*w*,*k*), *w* > *k*, output the "smallest" *k*-mer in w
- Smallest:
  - Lexicographic
  - Random (hash function)
- Important to use canonical *k*-mers

(*10*,*7*)

GTGATTACAT

GTGATTA

TGATTAC

GATTACA

ATTACAT

Reducing storage requirements for biological sequence comparison (Roberts et al., 2004)

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

ATGATCGTGATGTCATAGTATAGTGCAA

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

ATGATCGTGATGTCATAGTATAGTGCAA

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

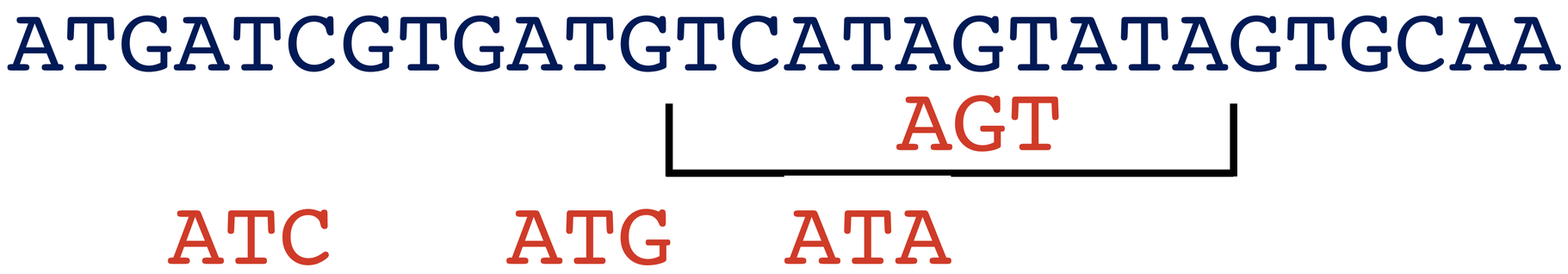
- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?



Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

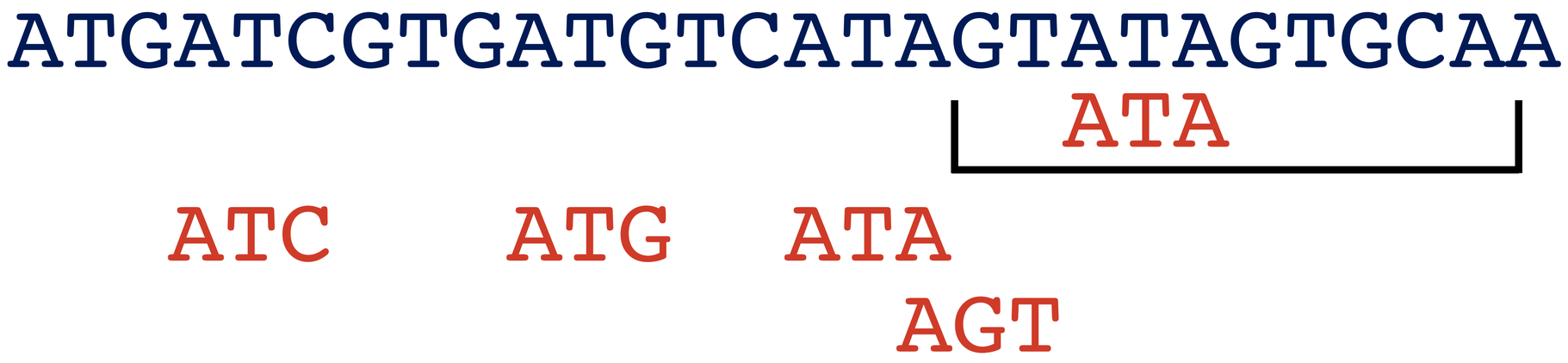
ATGATCGTGATGTCATAGTATAGTGCAA

AGT

ATC ATG ATA

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?



Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

ATGATCGTGATGTCATAGTATAGTGCAA

ATA

ATC ATG ATA

AGT

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

ATGATCGTGATGTCATAGTATAGTGCAA

- 5

ATC ATG ATA ATA

AGT

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

ATGATCGTGATGTCATAGTATAGTGCAA

- 5 ATC ATG ATA ATA AGT

- Window guarantee: Every 10-mer window must have a 3-mer minimizer!
- Good for read mapping. Why?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

ATGATCGTGATGTCATAGTATAGTGCAA

- 5 ATC ATG ATA ATA AGT
- 26 possible 3-mers
- 26 -> 5 ~81% reduction in number of 3-mers

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

ATGATCGTGATGTCATAGTATAGTGCAA

- 5 ATC ATG ATA ATA AGT
- 26 possible 3-mers
- 26 -> 5 ~81% reduction in number of 3-mers

IS THIS OPTIMAL?

Minimizer Example

- How many unique (10,3) minimizers exist on this genome?

ATGATCGTGATGTCATAGTATAGTGCAA

- 5 ATC ATG ATA ATA AGT
- 26 possible 3-mers
- 26 -> 5 ~81% reduction in number of 3-mers

IS THIS OPTIMAL? NO!

Minimizer Density

- Optimal minimizer density: 1/w (1 minimizer per length of window)

Minimizer Density

- Optimal minimizer density: 1/w (1 minimizer per length of window)

- 3

- 26 -> 3 ~89% reduction in number of 3-mers

Minimizer Density

- Optimal minimizer density: 1/w (1 minimizer per length of window)
- 3

- 26 -> 3 ~89% reduction in number of 3-mers
- Density in practice = 2/(w + 1)

Minimizer Density

- Why is lexicographic ordering bad in practice?

Paradigm	Constant
Optimal	1
DOCKS	1.737
Random (Hash)	1.999
Lexicographic	2.236

Minimizer Density

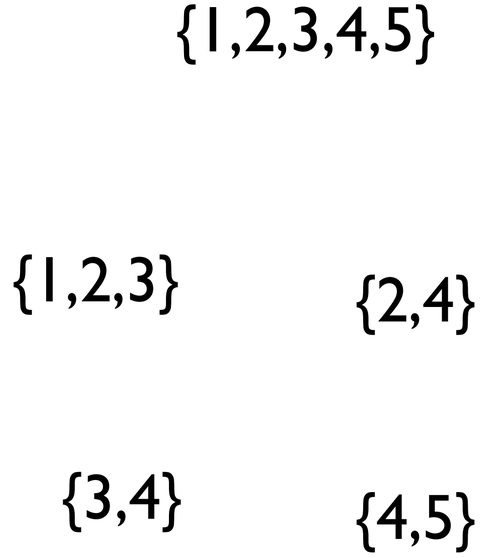
- Why is lexicographic ordering bad in practice?
- HOMOPOLYMERS!

AAAAAAAAAAAAA

Paradigm	Constant
Optimal	1
DOCKS	1.737
Random (Hash)	1.999
Lexicographic	2.236

Minimizer Density

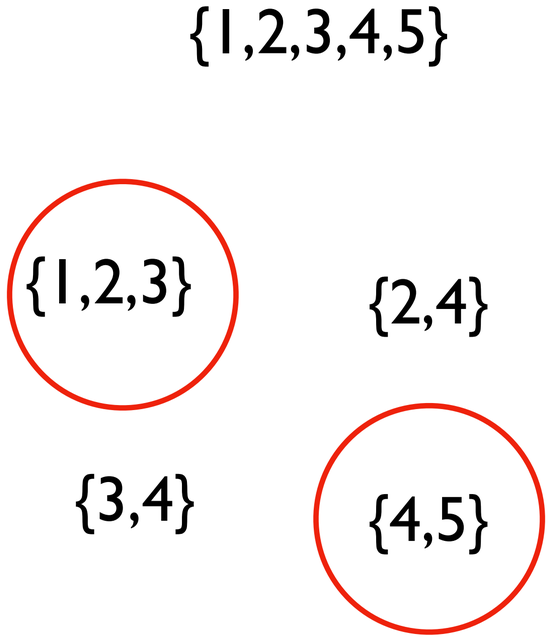
- Why is lexicographic ordering bad in practice?
- DOCKS uses a Universal Hitting Set:
  - Given tuple (*w,k*), find the smallest set $U_{k,w}$ of *k*-mers s.t. any window *w* must contain an element of $U_{k,w}$
  - Approximation of set-cover



Compact universal k-mer hitting sets. (Orenstein et. al, Workshop on Algorithms in Bioinformatics 2016)

Minimizer Density

- Why is lexicographic ordering bad in practice?
- DOCKS uses a Universal Hitting Set:
  - Given tuple (*w,k*), find the smallest set $U_{k,w}$ of *k*-mers s.t. any window *w* must contain an element of $U_{k,w}$
  - Approximation of set-cover



Compact universal k-mer hitting sets. (Orenstein et. al, Workshop on Algorithms in Bioinformatics 2016)

Context Dependency

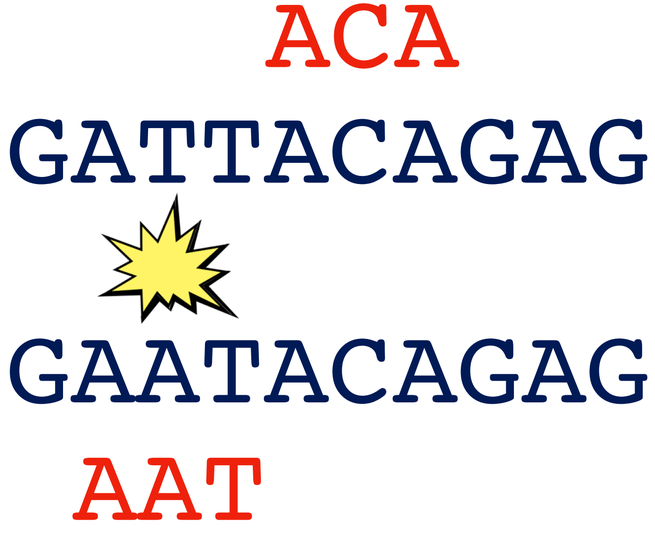
- There are *w* - *k* potential minimizers per window
- Mutations in a window can affect minimizer schemes, even if they don't affect that k-mer!
- Sketching methods exist that are context <u>independent</u>

ACA

GATTACAGAG

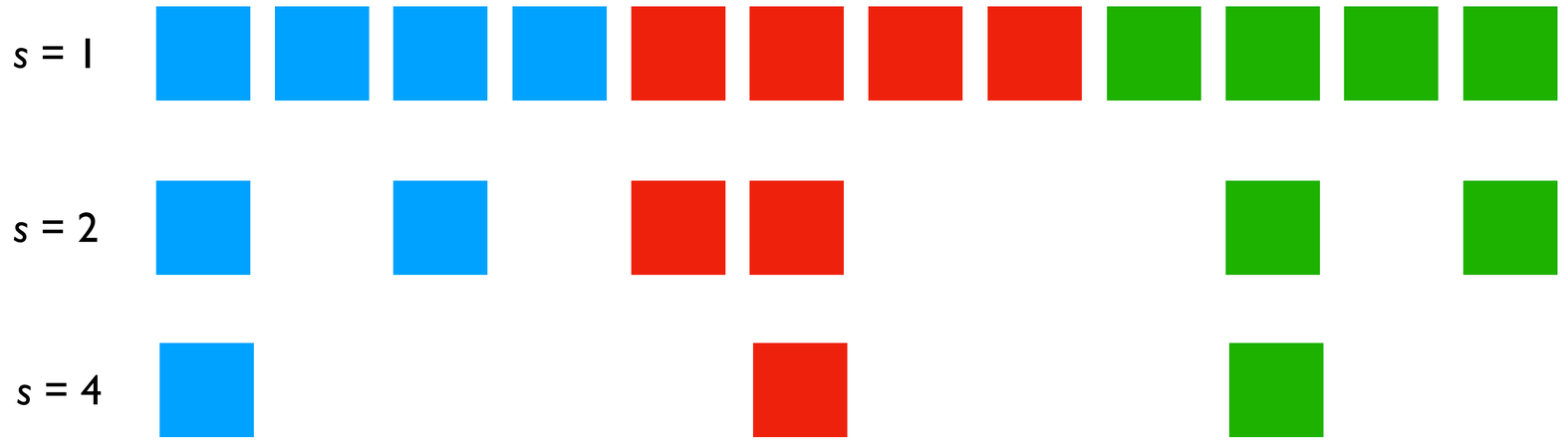
Context Dependency

- There are *w* - *k* potential minimizers per window
- Mutations in a window can affect minimizer schemes, even if they don't affect that k-mer!
- Sketching methods exist that are context <u>independent</u>



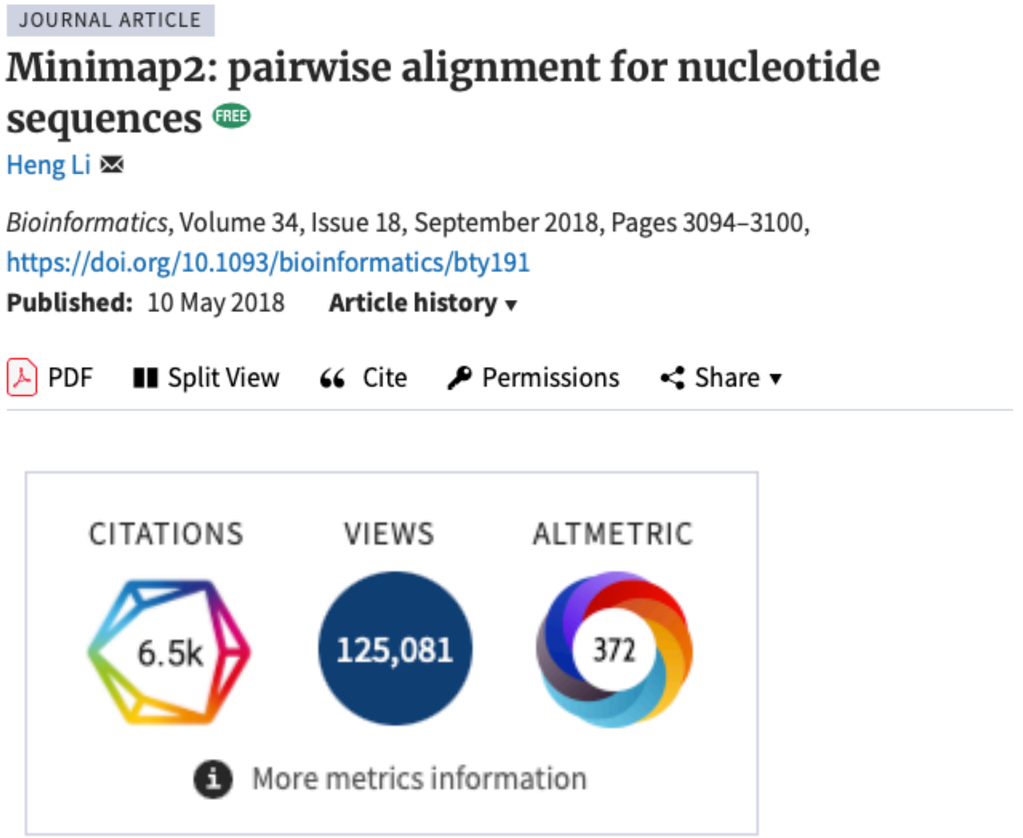
Context Independent Sketches

- Modimizers
  - hash(*k*-mer) % *s*
  - Density inversely proportional to *s*
- Syncmers
  - Minimizer at position *t*
- Give up window guarantee!



Minimap2

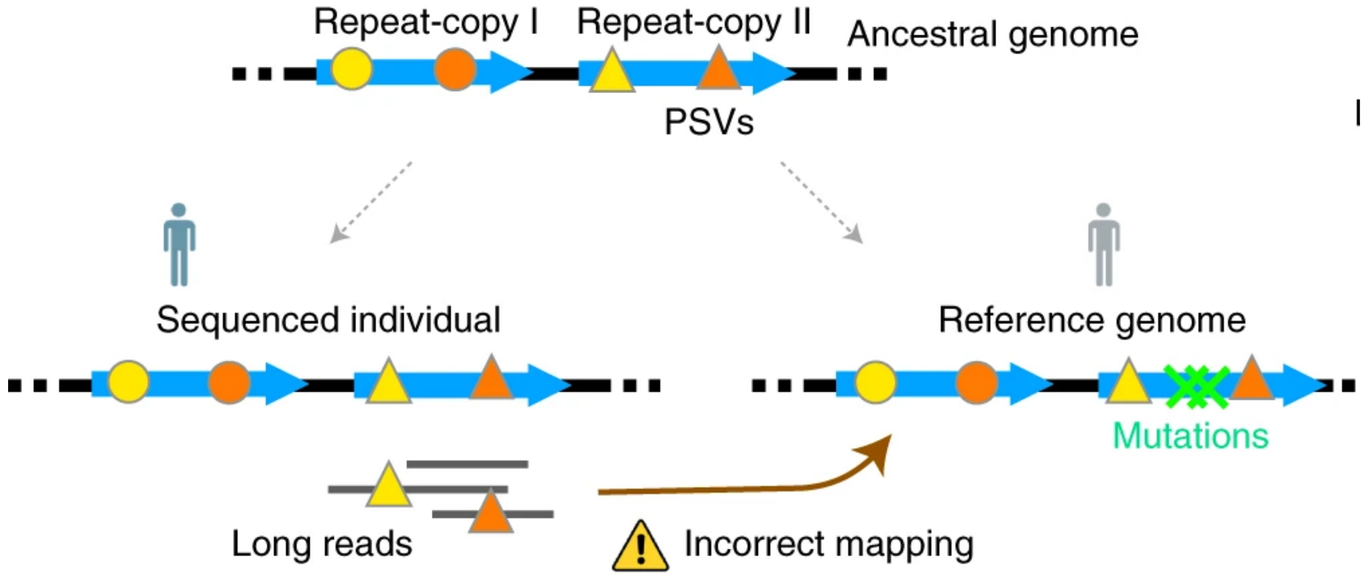
- De facto standard for long read mapping
- Wide tolerance for gaps, good for spliced alignment too!
- Not so great at divergent/repetitive seqeunces



Minimap2: pairwise alignment for nucleotide sequences. (Li, Bioinformatics 2018)

WinnowMap

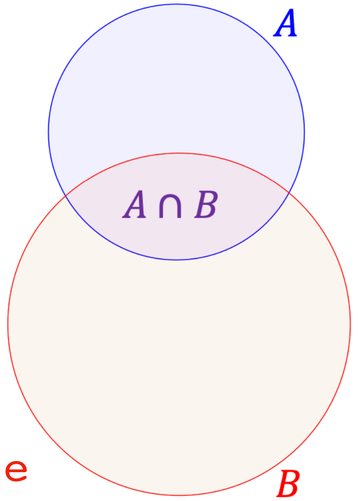
- Rare *k*-mers are more important than homopolymers & common k-mers!
- Include a weight parameter with each minimizer
- Improves upon minimap2 when dealing with repetitive seqeunces!



Weighted minimizer sampling improves long read mapping (Jain et al., Bioinformatics 2020)

Resemblance of Genomes

- Andrei Broder wanted to mathematically determine if two documents were "roughly the same"
- Break text into "shingles", perform set operations on those shingles



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

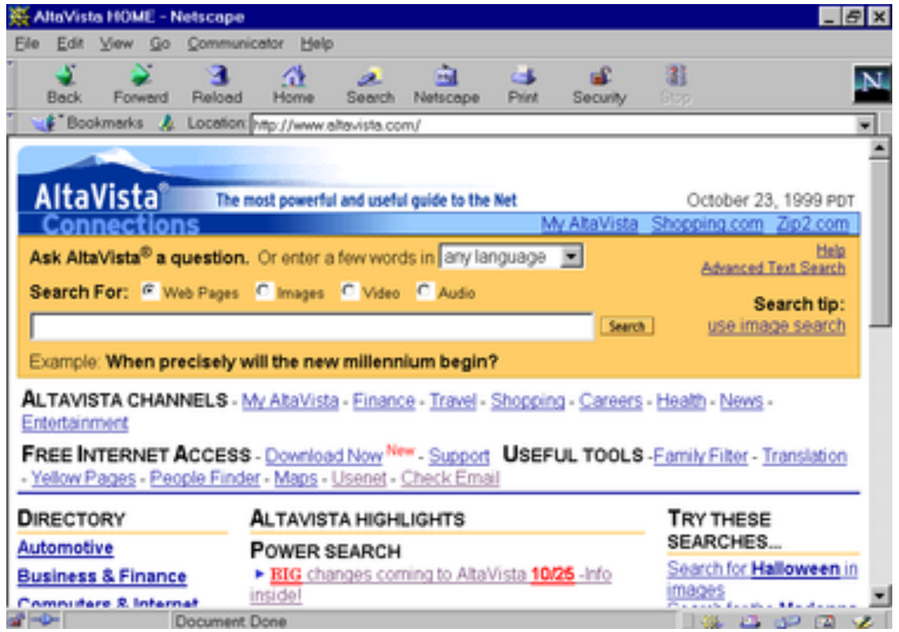
```
A rose is a rose is a rose
A rose is a
  rose is a rose
       is a rose is
          a rose is a
            rose is a rose
```

```
A rose is a flower is a rose
A rose is a
  rose is a flower
       is a flower is
          a flower is a
            flower is a rose
```

On the Resemblance and Containment of Documents. (Broder, Compression and Complexity of Sequences 1997)

Resemblance of Genomes

- Andrei Broder wanted to mathematically determine if two documents were "roughly the same"
- Break text into "shingles", perform set operations on those shingles

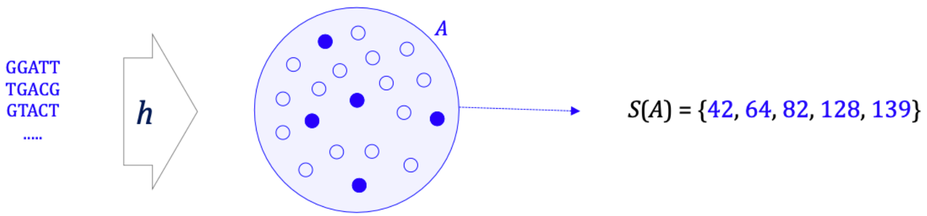


> Notice also that resemblance is not transitive (a well-known fact bemoaned by grandparents all over), but neither is our informal idea of "roughly the same;" for instance consecutive versions of a paper might well be "roughly the same," but version 100 is probably quite far from version 1.

On the Resemblance and Containment of Documents. (Broder, Compression and Complexity of Sequences 1997)

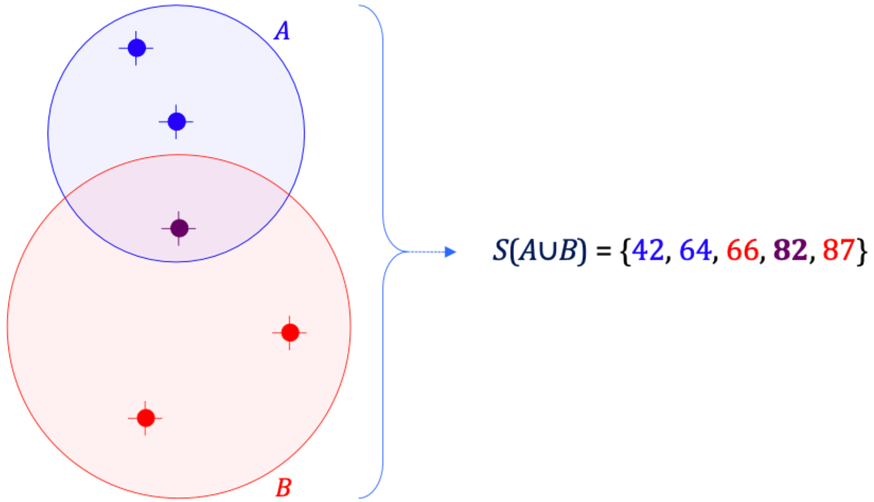
Resemblance of Genomes

- Shingles naturally translate into *k*-mers
- Sketch using MinHash: minimum *s* hashes in a set



k=16

s=500

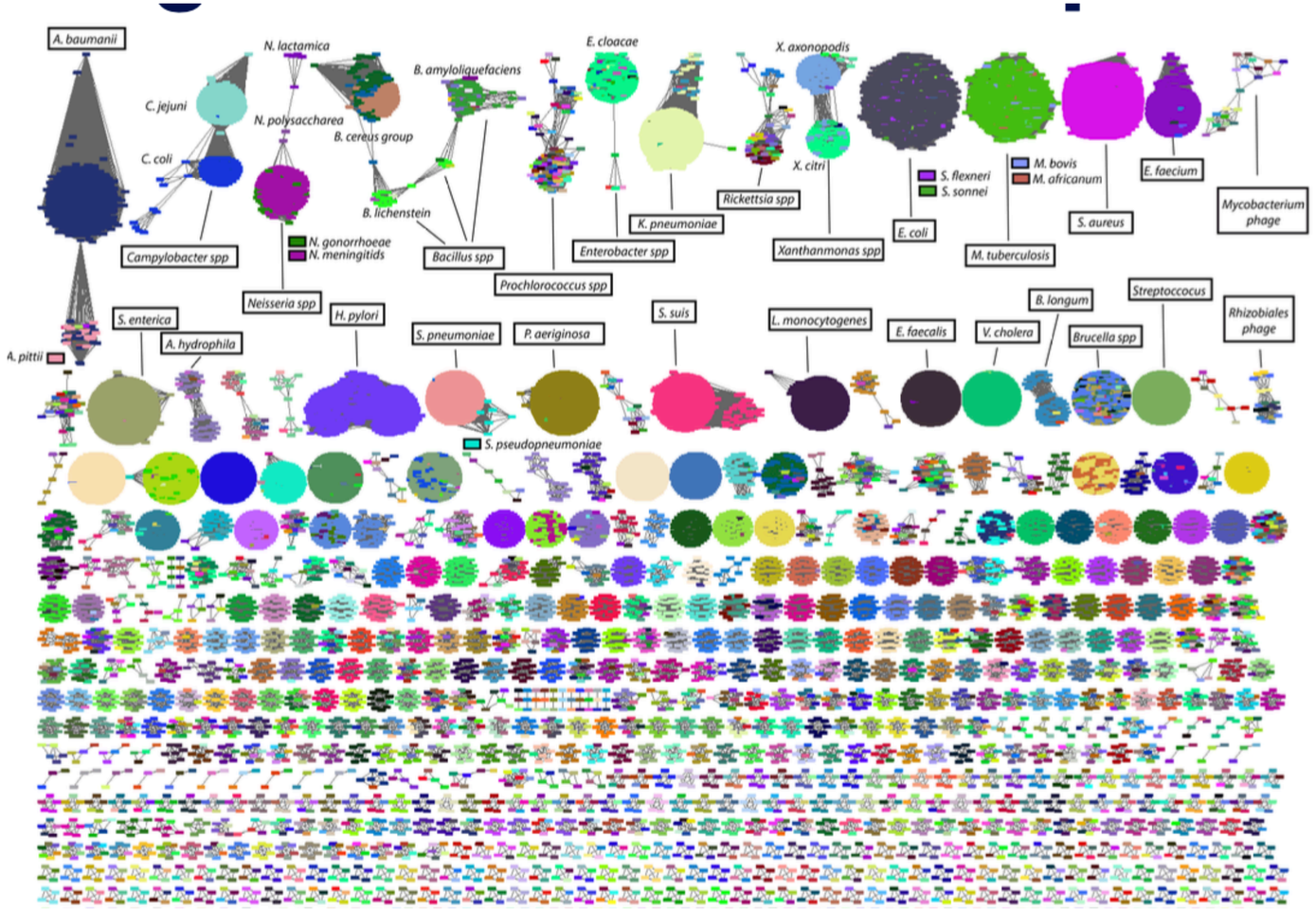


$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \approx \frac{|S(A \cup B) \cap S(A) \cap S(B)|}{|S(A \cup B)|}$$

Mash: fast genome and metagenome estimation using MinHash (Ondov et al., Genome Biology 2016)

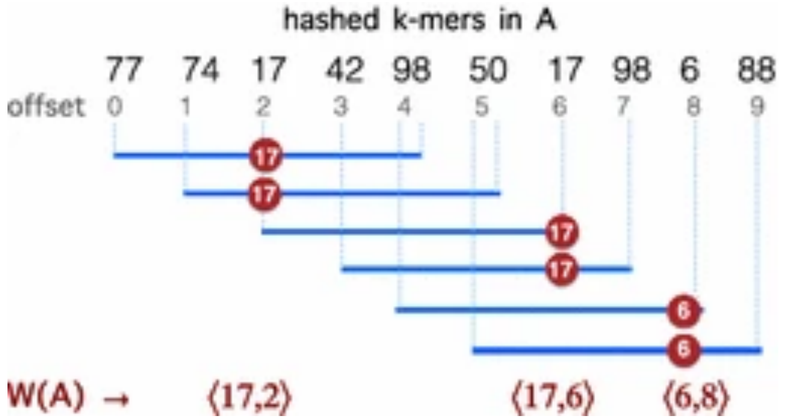
Clustering NCBI RefSeq

- 1.5 billion distances computed in 46 CPU h
- sketch db < 100 MB



Combine Mash with Minimap

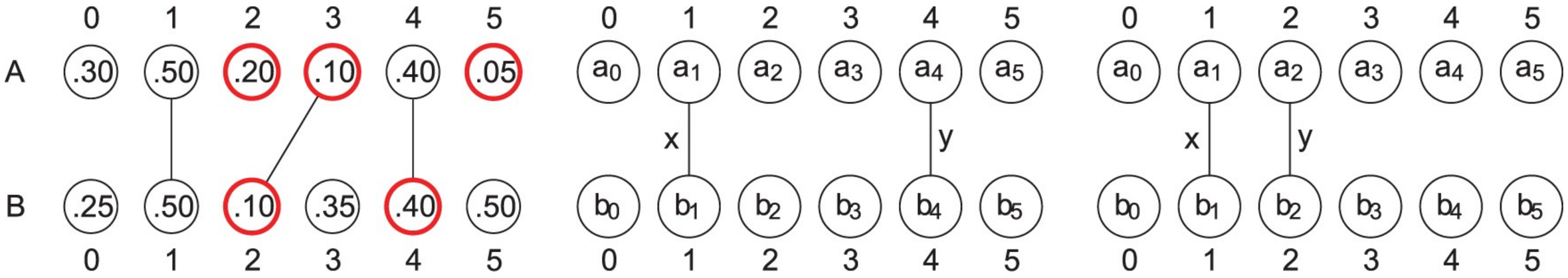
- Obtain minimizers both from reference and query sequences
- Apply minhash on top to control density of sampled *k*-mers
- 290% faster mapping than Bowtie2



A fast approximate algorithm for mapping long reads to large reference databases. (Jain et al., RECOMB 2017)

Oops 😱

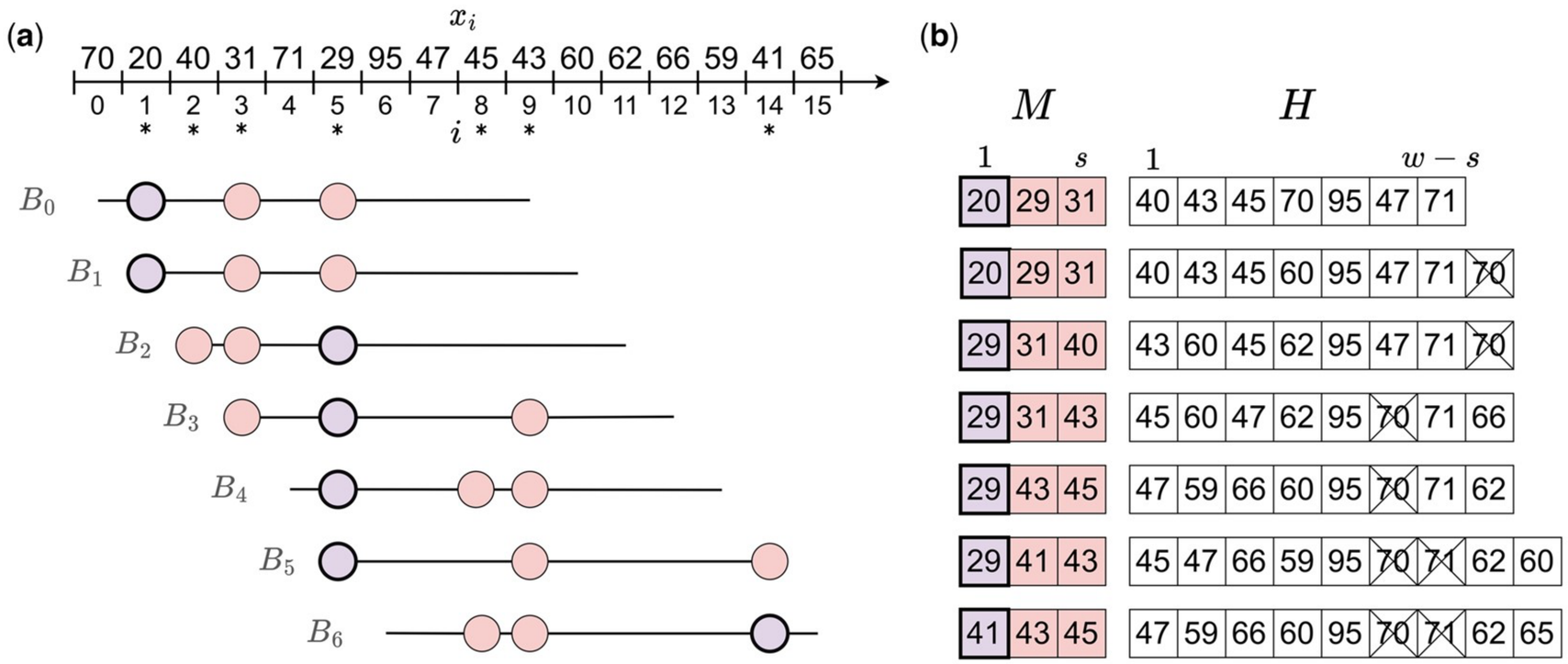
- Turns out minimizers are biased when used for Jaccard!



The minimizer Jaccard estimator is biased and inconsistent. (Balbasi et al., Bioinformatics 2022)

MashMap 3.0

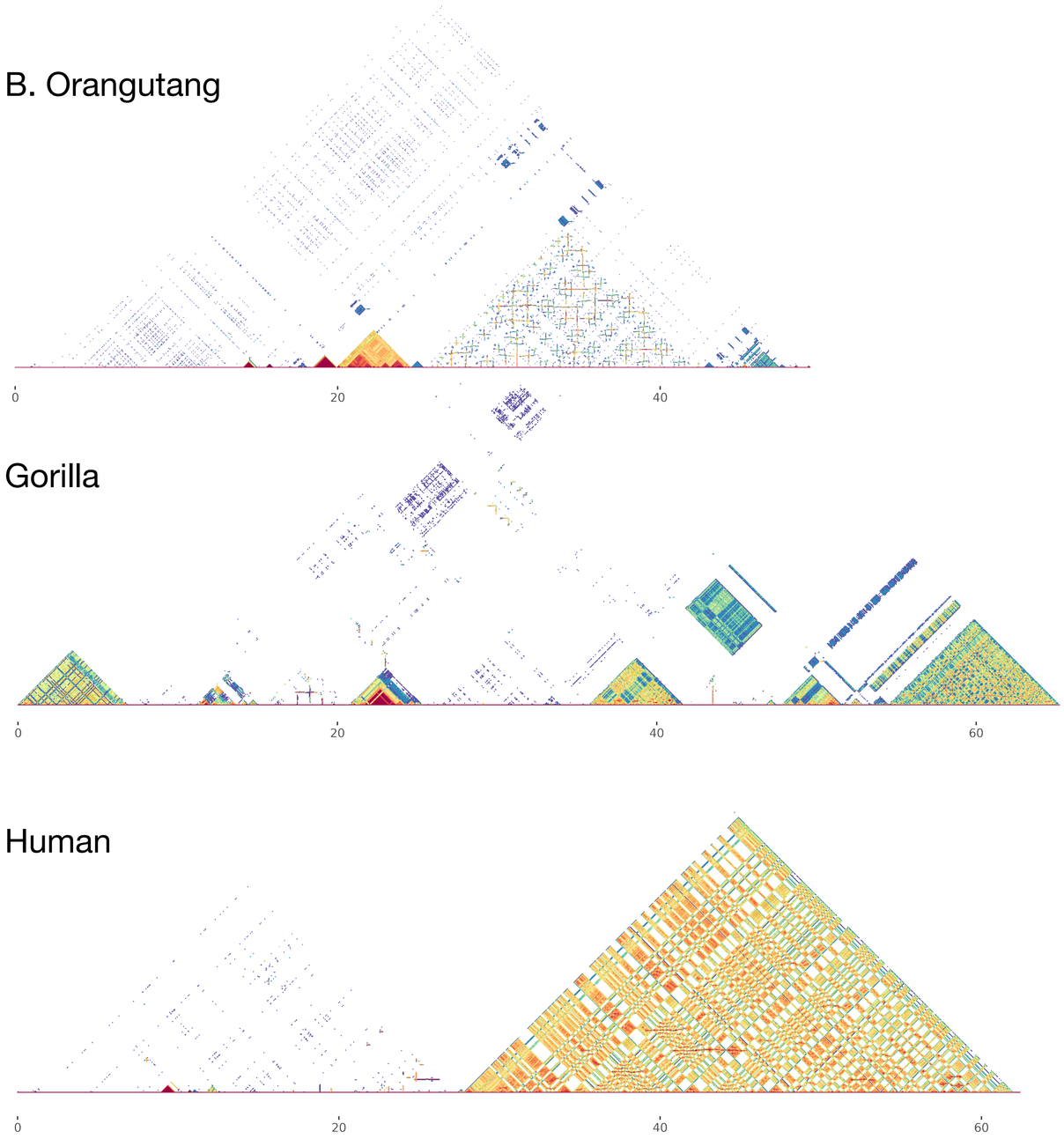
- Replace minimizers with “minmers”
- One of the smallest *k*-mers in a window
- Construct a rolling minhash
- Removes bias in Jaccard



Minmers are a generalization of minimizers that enable unbiased local Jaccard estimation (Kille et al., Bioinformatics 2023)

Applications

- Evolution is rapid on the Y chromosome!
- Complex repeat structures visualized with triangles
- Plots produced by using the Jaccard distance to estimate sequence similarity



Publications in Progress!