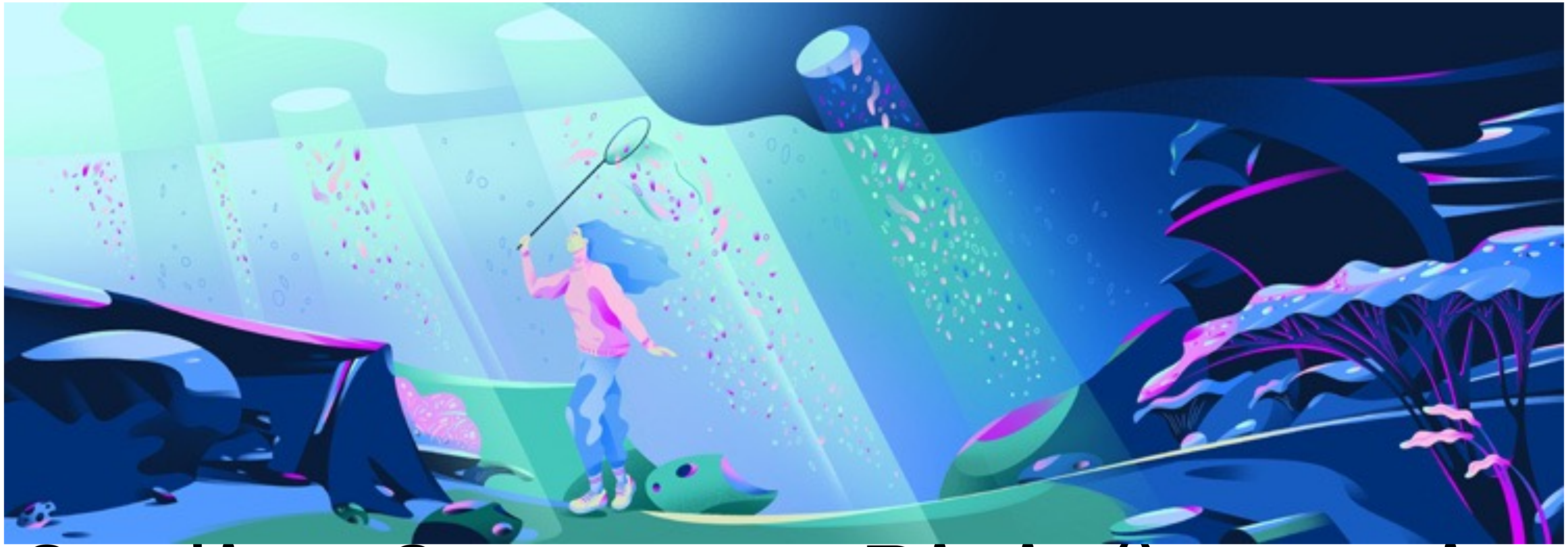


@Manon Sauzara

Scaling Sequence Bioinformatics with Logan and Logan-Search

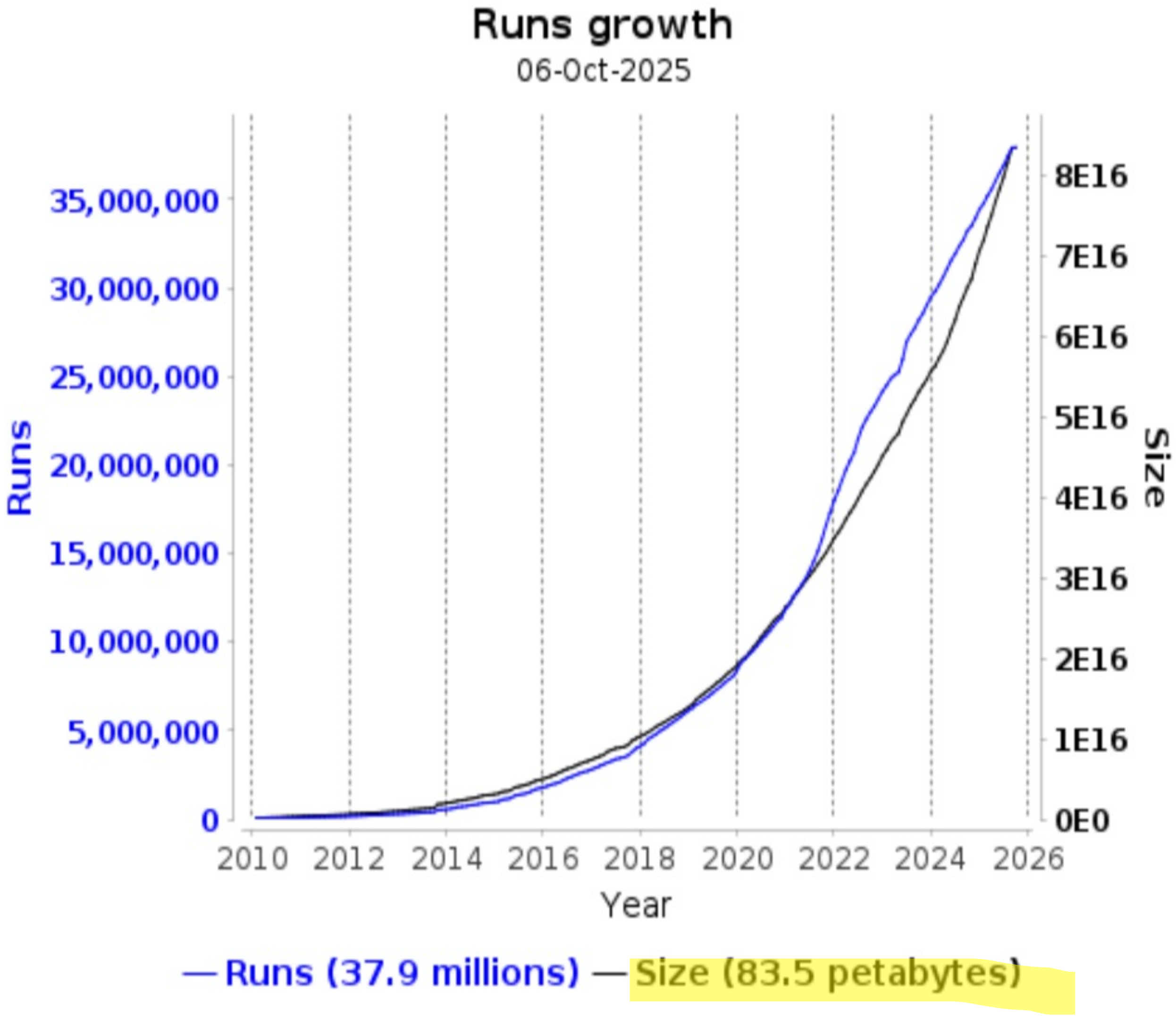
Pierre Peterlongo

PEPI Ibis, oct. 2025

Evolution

ENA

European Nucleotide Archive



https://www.ebi.ac.uk/ena/browser/about/statistics

SRA: Open Science at Its Best

A. Babaian

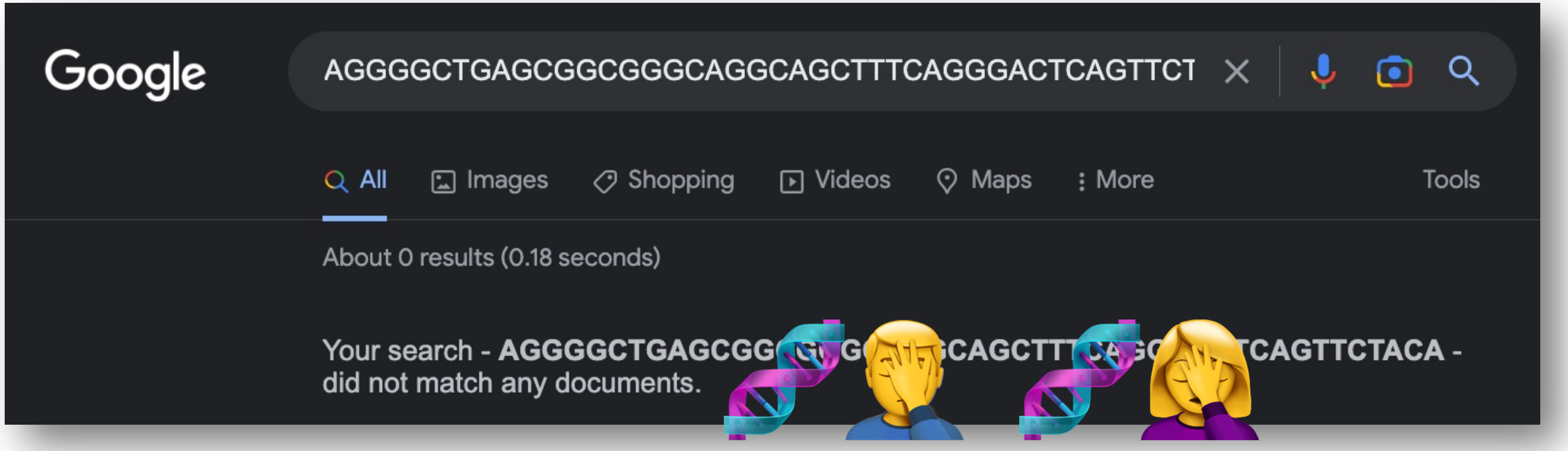
*"Earth's genetic biodiversity is the **shared heritage of all living organisms**, and as scientists we are **responsible for liberating and sharing** this heritage with everyone. Freely, and openly."*

Billions dollard data

“Library of Alexandria”
for genetics

My objective / obsession

- Propose a genomic search engine
- Covering the *Library of Alexandria* for genetics, SRA



AGGGGCTGAGCGGCGGGCAGGCAGCTTTCAGGGACTCAGTTCT

About 0 results (0.18 seconds)

Your search - AGGGGCTGAGCGG G CAGCTTT CAG TCAGTTCTACA - did not match any documents.

From reads to unitigs and contigs

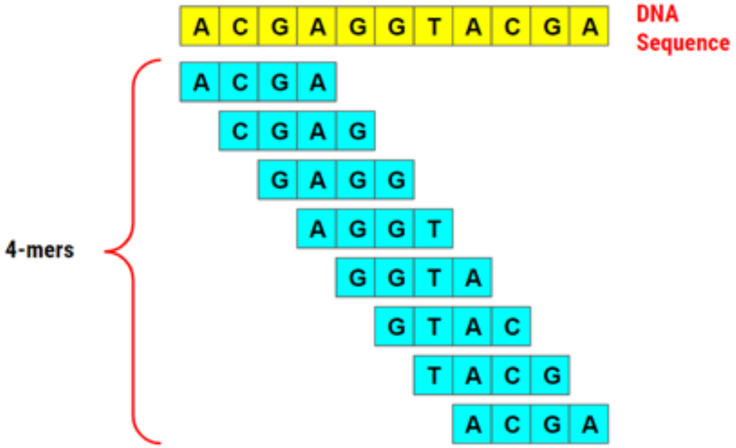
Logan

A taste of algorithmics

Assembly, kmers, and de Bruijn Graphs

k-mers, why and what.

- **No word in DNA**: ....ACGGCGCATCTGGTCGTATGACGAGCAGCT...
- Split to subsequences of fixed length *k* (called **k-mers**)



$$k = 31$$

- Nb words in Oxford English dictionary: ~600,000
- Number distinct 31-mers: 4,611,686,018,427,387,904

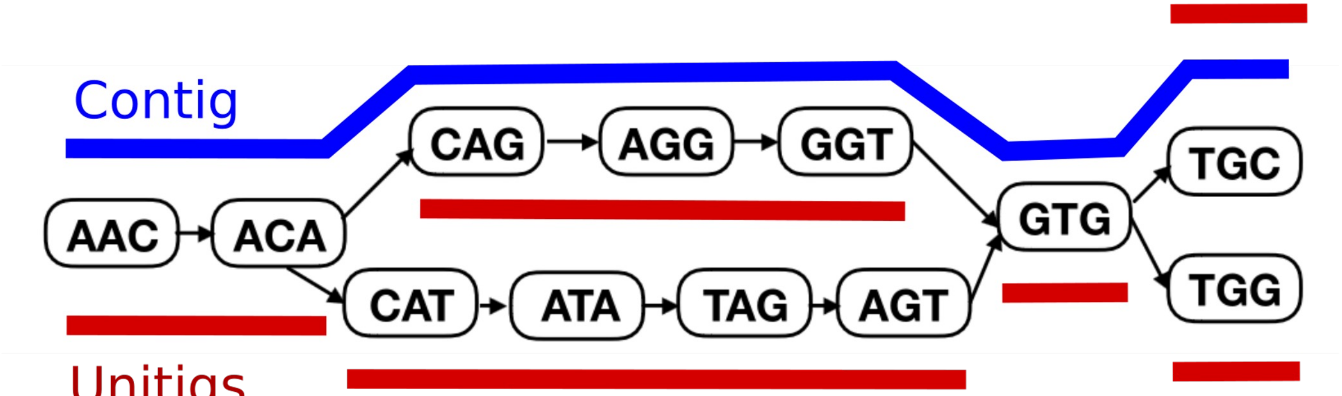
Unitigs & Contigs

Contigs: typical output of genome assembly methods

consensus sequence

No clear definition

Unitig: simple path in the de Bruijn graph



Graph from Menegaux, Vert

Realization: logan

Logan: Outline

PI: Rayan Chikhi

- 50 petabases of reads were downloaded & assembled on AWS cloud

- Reconstructed all contigs and unitigs in the entire SRA (27 millions samples)
  - 0.3 PB of contigs
  - 2 PB of unitigs

Search something on Logan data

- Sequence alignment with DIAMOND (`--sensitive`) streaming all of Logan contigs
- 11 hours on 60k cloud vCPUS (10k$)

With Téo Lemane

Search engine on logan data: Logan-search

Querying using kmers

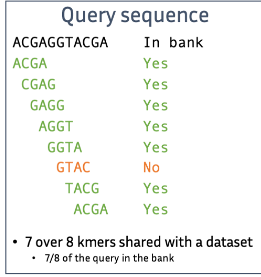
Querying using kmers

Query sequence

```
ACGAGGTACGA    In  bank
ACGA           Yes
 CGAG          Yes
  GAGG         Yes
   AGGT        Yes
    GGTA       Yes
     GTAC      No
      TACG     Yes
       ACGA    Yes
```

- 7 over 8 kmers shared with a dataset
  - 7/8 of the query in the bank

Comparing using kmers?



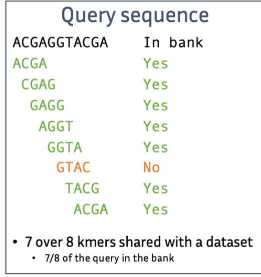
Query sequence

```
ACGAGGTACGA     In bank
ACGA            Yes
 CGAG           Yes
  GAGG          Yes
   AGGT         Yes
    GGTA        Yes
     GTAC       No
      TACG      Yes
       ACGA     Yes
```

- 7 over 8 kmers shared with a dataset
  - 7/8 of the query in the bank

Alignments

kmers



Query sequence

```
ACGAGGTACGA      In bank
ACGA             Yes
 CGAG            Yes
  GAGG           Yes
   AGGT          Yes
    GGTA         Yes
     GTAC        No
      TACG       Yes
       ACGA      Yes
```

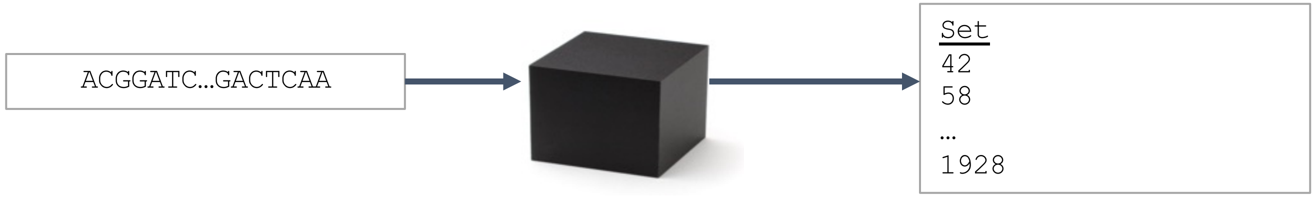
- 7 over 8 kmers shared with a dataset
  - 7/8 of the query in the bank

Genomic research engine: conceptual view

index

Atomic question

- Given a queried kmer, in which sets does it exist?



ACGGATC…GACTCAA

Set
42
58
…
1928

: A bloom filter

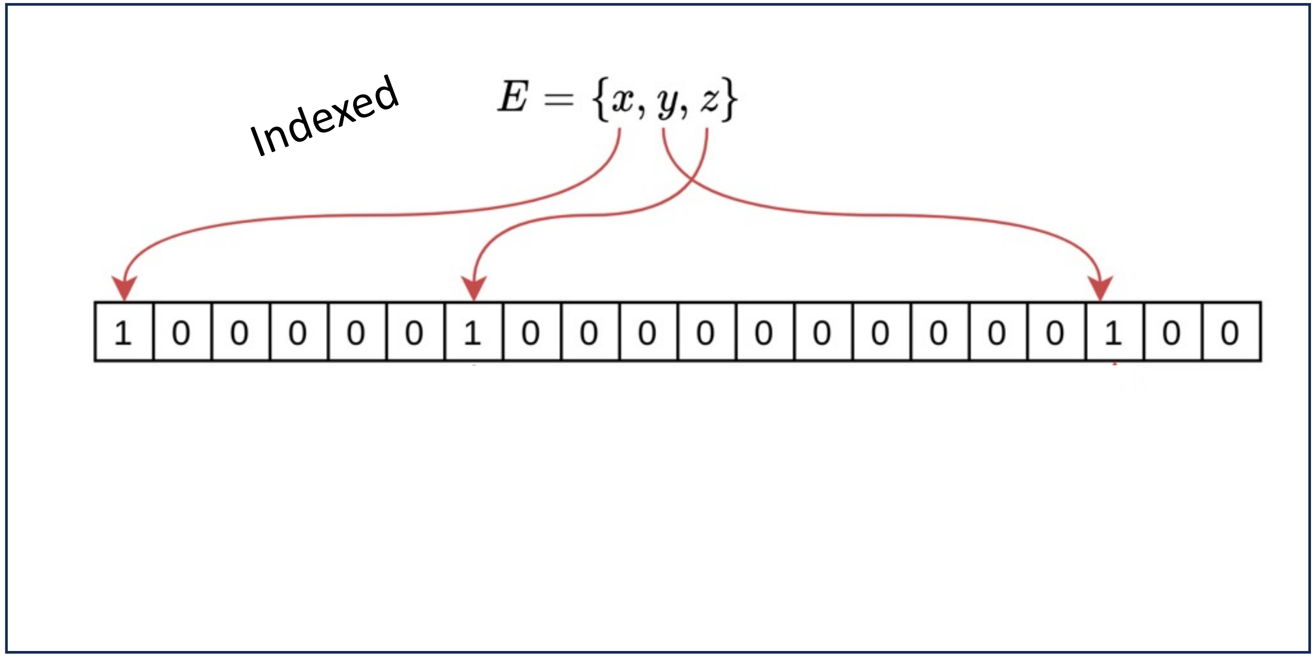
Bloom Filter

A bit vector B of fixed size

Add one element:
B[hash(element)] = 1

Query one element: B[hash(element)]
0: absent
1: present (possibly a False Positive)

1
0
0
1
0
0
1
1



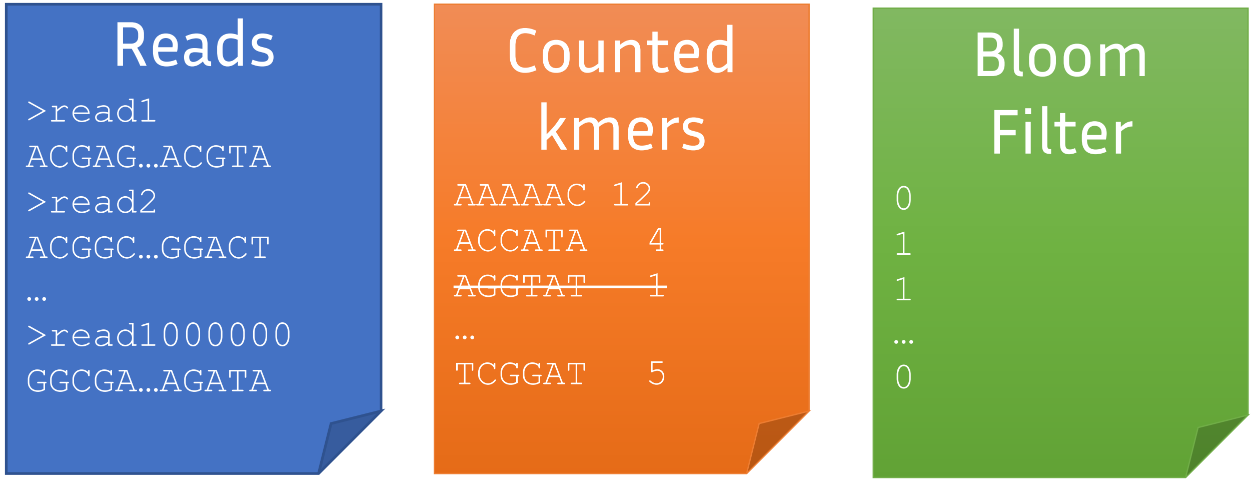
Indexing: conceptual view

One read set:

- Extract & count **kmers**
- Filter kmers
- Generate a bloom filter

***N* read sets**:

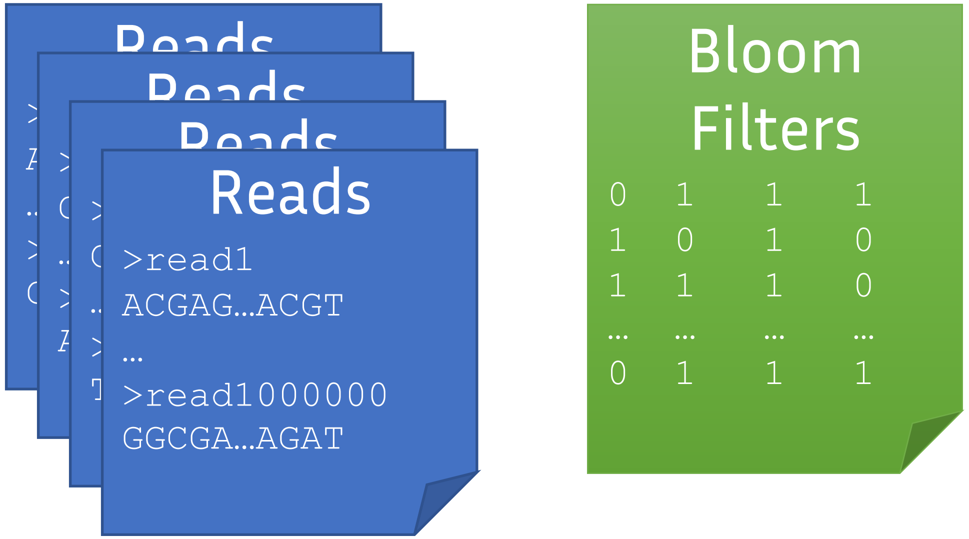
- Create *N* bloom filters
- This is the index



>read1
ACGAG…ACGTA
>read2
ACGGC…GGACT
…
>read1000000
GGCGA…AGATA

AAAAAC 12
ACCATA 4
~~AGGTAT 1~~
…
TCGGAT 5

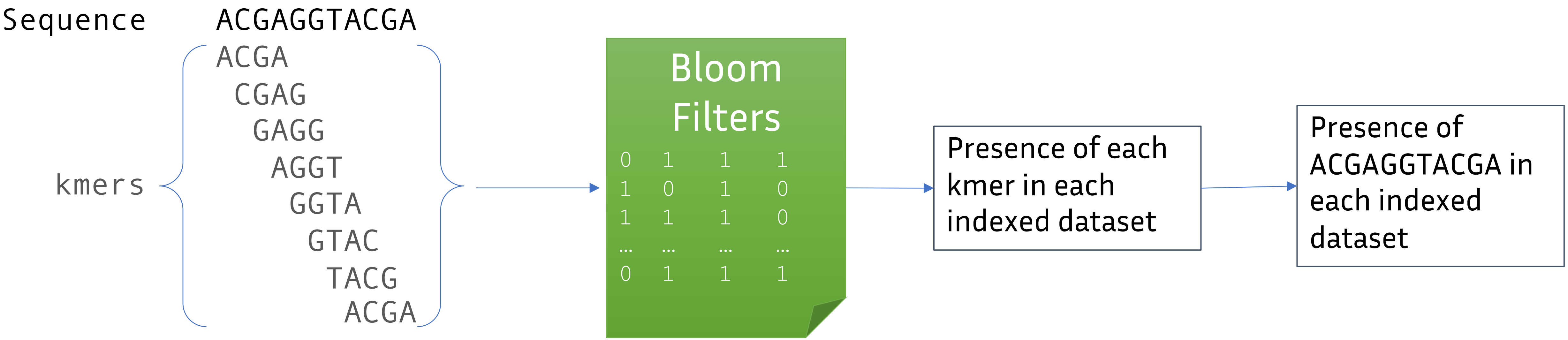
0
1
1
…
0



>read1
ACGAG…ACGT
…
>read1000000
GGCGA…AGAT

0	1	1	1
1	0	1	0
1	1	1	0
…	…	…	…
0	1	1	1

Indexing: conceptual view



Sequence ACGAGGTACGA

kmers

ACGA
CGAG
GAGG
AGGT
GGTA
GTAC
TACG
ACGA

Bloom Filters

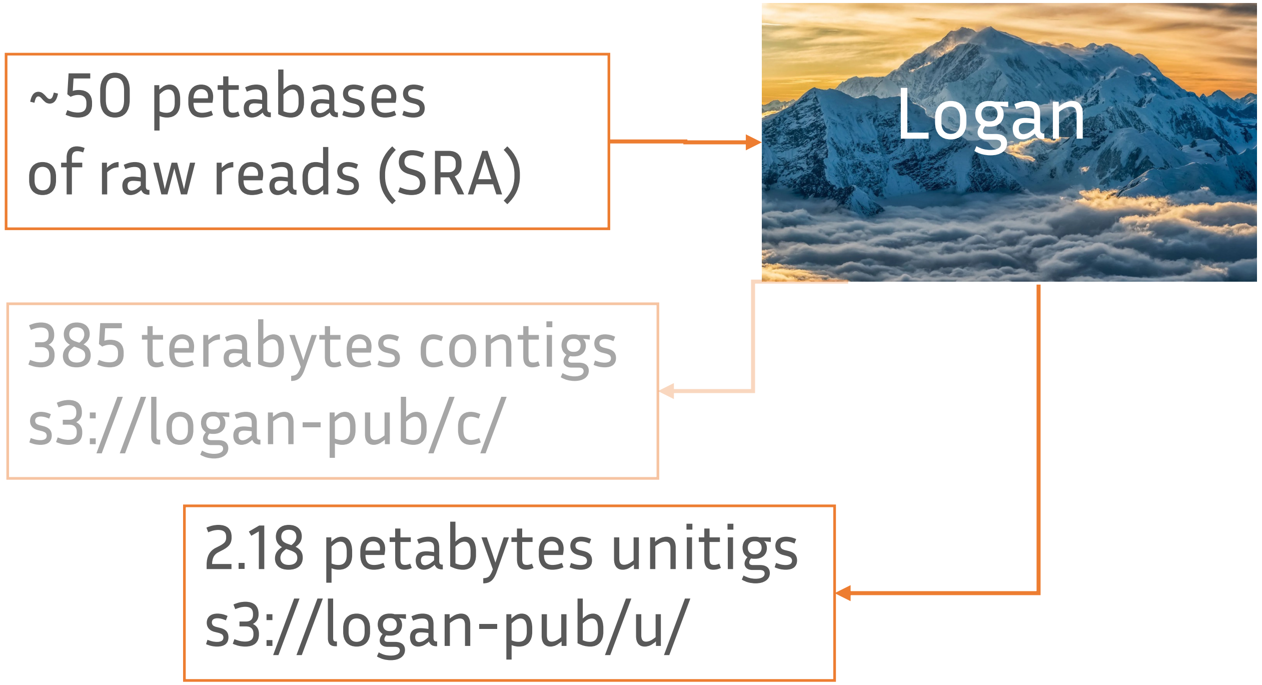
```
0  1  1  1
1  0  1  0
1  1  1  0
…  …  …  …
0  1  1  1
```

Presence of each kmer in each indexed dataset

Presence of ACGAGGTACGA in each indexed dataset

Realization: logan-search

Logan



~50 petabases
of raw reads (SRA)

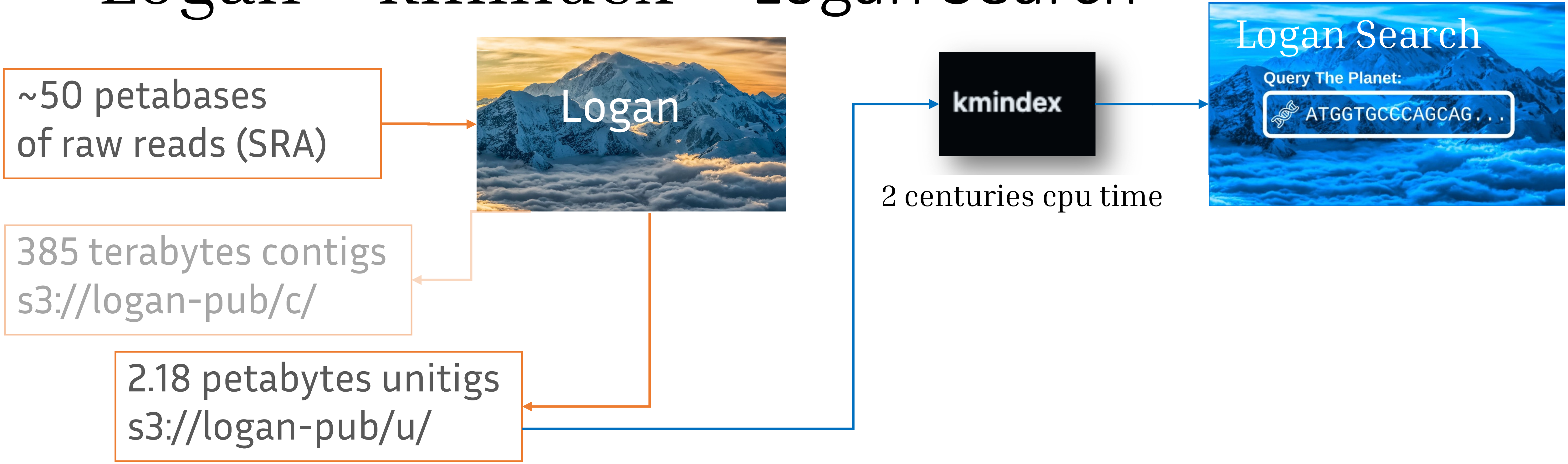
Logan

385 terabytes contigs
s3://logan-pub/c/

2.18 petabytes unitigs
s3://logan-pub/u/

27 million samples
2 million billions k-mers

Logan + kmindex = Logan Search



~50 petabases
of raw reads (SRA)

385 terabytes contigs
s3://logan-pub/c/

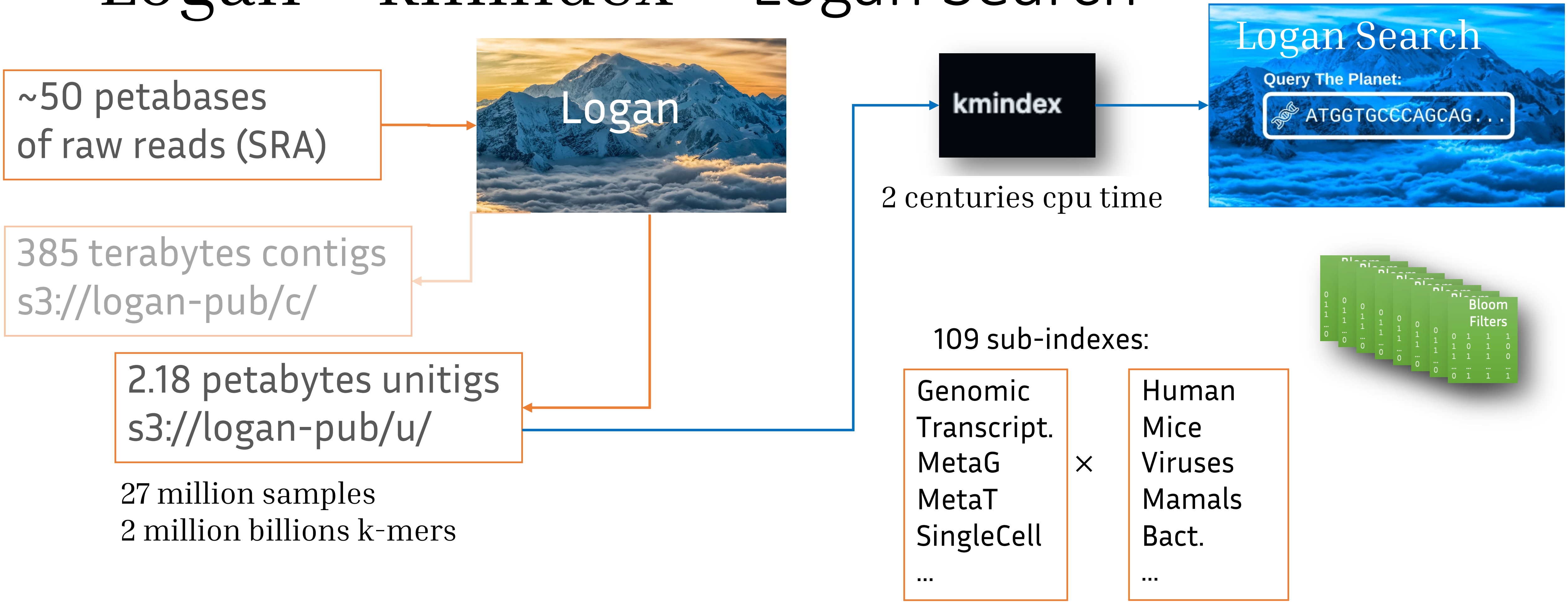
2.18 petabytes unitigs
s3://logan-pub/u/

2 centuries cpu time

27 million samples
2 million billions k-mers

Chikhi, R., Raffestin, B., Korobeynikov, A., Edgar, R. C., & Babaian, A. (2024).
Logan: Planetary-Scale Genome Assembly Surveys Life's Diversity.
bioRxiv, 2024-07.

Logan + kmindex = Logan Search



~50 petabases
of raw reads (SRA)

385 terabytes contigs
s3://logan-pub/c/

2.18 petabytes unitigs
s3://logan-pub/u/

27 million samples
2 million billions k-mers

2 centuries cpu time

109 sub-indexes:

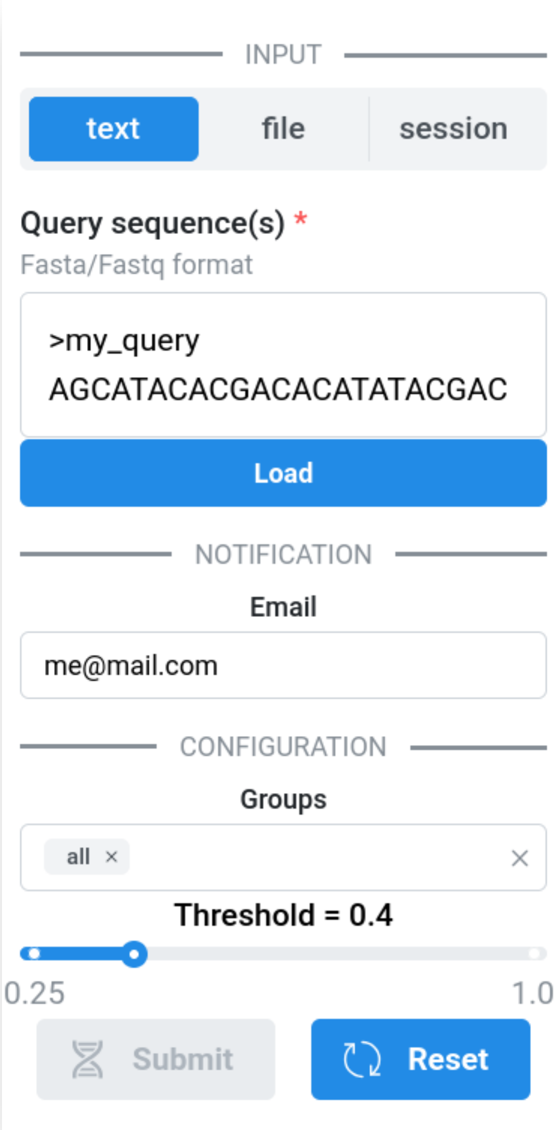
Genomic
Transcript.
MetaG
MetaT
SingleCell
...

×

Human
Mice
Viruses
Mamals
Bact.
...

Chikhi, R., Raffestin, B., Korobeynikov, A., Edgar, R. C., & Babaian, A. (2024).
Logan: Planetary-Scale Genome Assembly Surveys Life's Diversity.
bioRxiv, 2024-07.

Logan-search answer: perform a query



Query sequence(s) *
Fasta/Fastq format

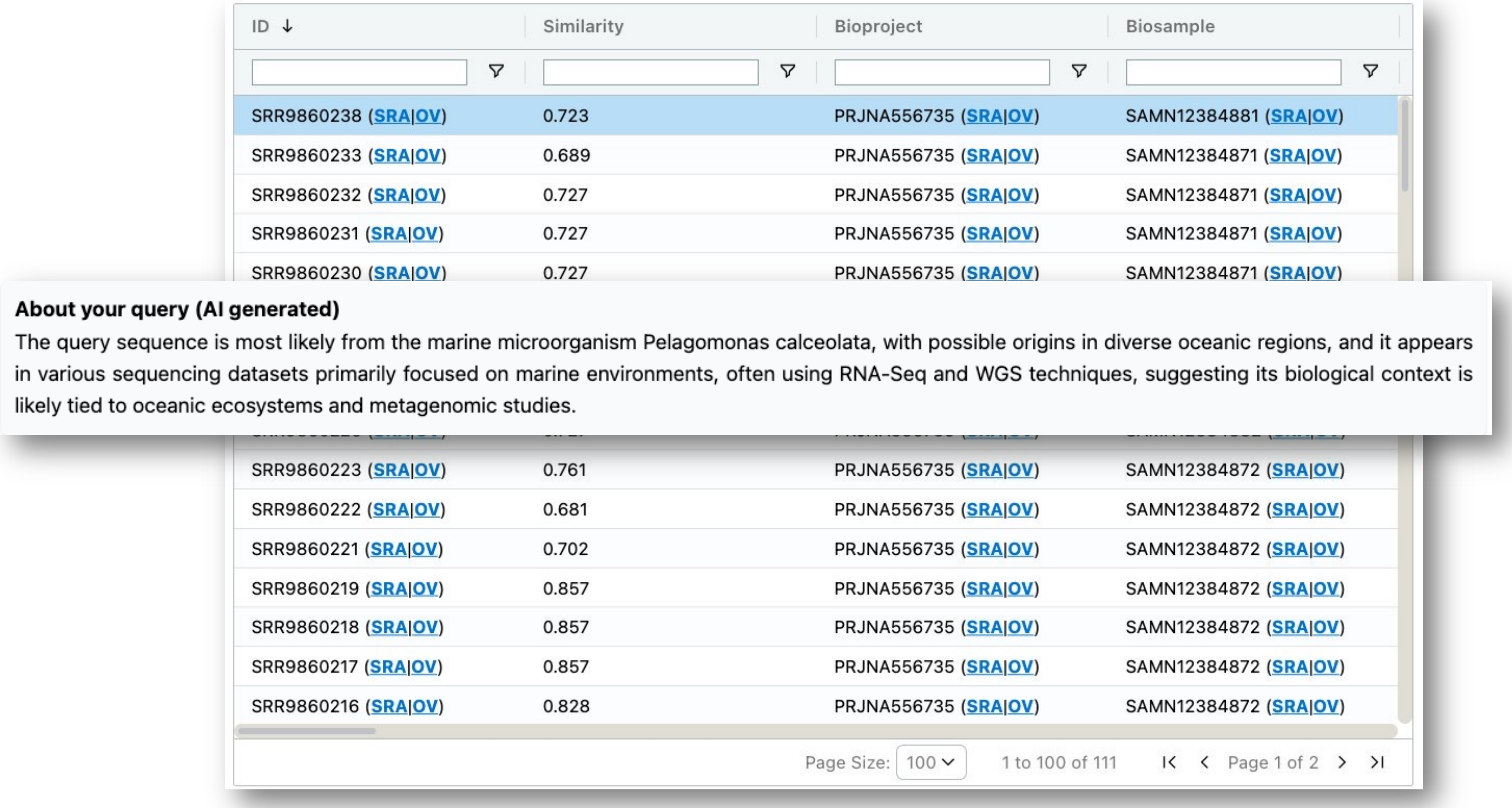
```
>my_query
AGCATACACGACACATATACGAC
```

Email: me@mail.com

Groups: all ×

Threshold = 0.4

Logan-search answer: perform a query



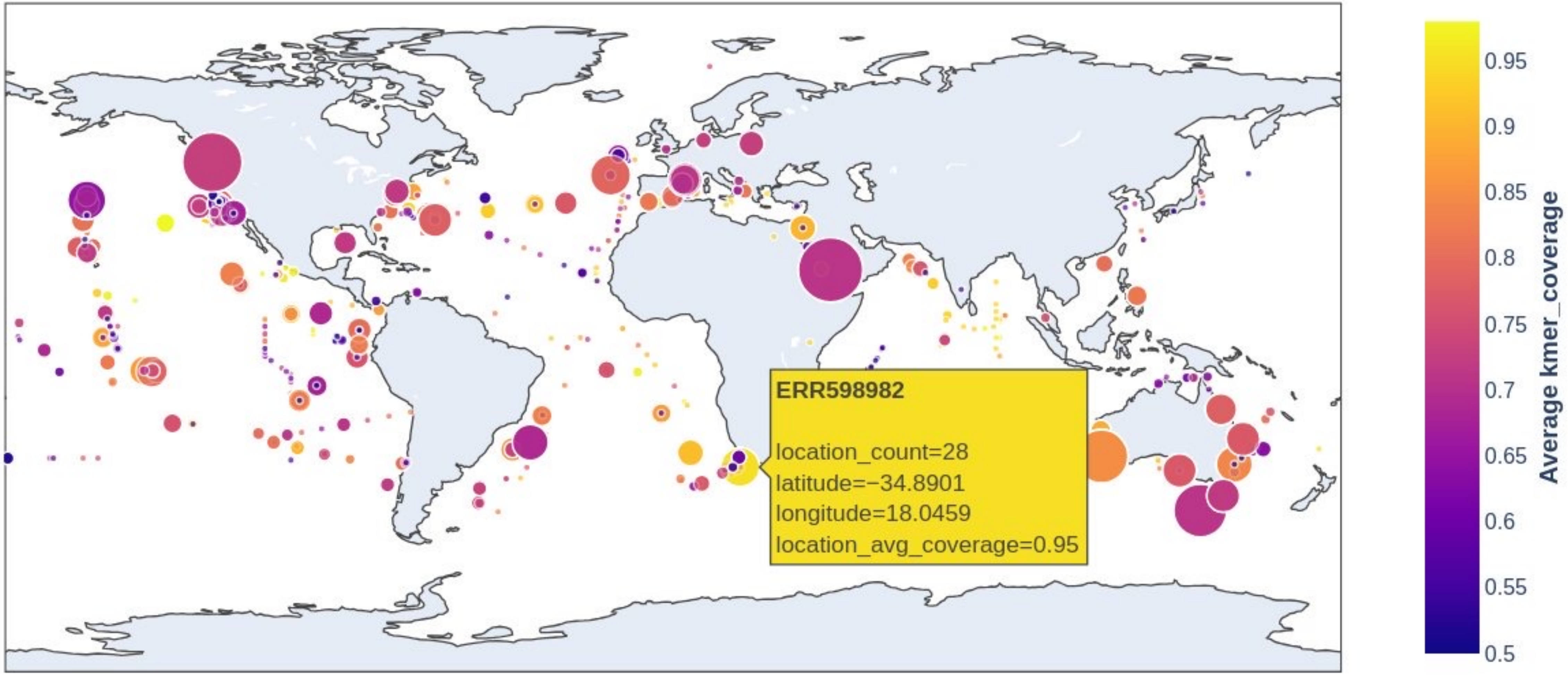
ID ↓	Similarity	Bioproject	Biosample
SRR9860238 (SRA|OV)	0.723	PRJNA556735 (SRA|OV)	SAMN12384881 (SRA|OV)
SRR9860233 (SRA|OV)	0.689	PRJNA556735 (SRA|OV)	SAMN12384871 (SRA|OV)
SRR9860232 (SRA|OV)	0.727	PRJNA556735 (SRA|OV)	SAMN12384871 (SRA|OV)
SRR9860231 (SRA|OV)	0.727	PRJNA556735 (SRA|OV)	SAMN12384871 (SRA|OV)
SRR9860230 (SRA|OV)	0.727	PRJNA556735 (SRA|OV)	SAMN12384871 (SRA|OV)
SRR9860223 (SRA|OV)	0.761	PRJNA556735 (SRA|OV)	SAMN12384872 (SRA|OV)
SRR9860222 (SRA|OV)	0.681	PRJNA556735 (SRA|OV)	SAMN12384872 (SRA|OV)
SRR9860221 (SRA|OV)	0.702	PRJNA556735 (SRA|OV)	SAMN12384872 (SRA|OV)
SRR9860219 (SRA|OV)	0.857	PRJNA556735 (SRA|OV)	SAMN12384872 (SRA|OV)
SRR9860218 (SRA|OV)	0.857	PRJNA556735 (SRA|OV)	SAMN12384872 (SRA|OV)
SRR9860217 (SRA|OV)	0.857	PRJNA556735 (SRA|OV)	SAMN12384872 (SRA|OV)
SRR9860216 (SRA|OV)	0.828	PRJNA556735 (SRA|OV)	SAMN12384872 (SRA|OV)

Page Size: 100 · 1 to 100 of 111 · Page 1 of 2

About your query (AI generated)

The query sequence is most likely from the marine microorganism Pelagomonas calceolata, with possible origins in diverse oceanic regions, and it appears in various sequencing datasets primarily focused on marine environments, often using RNA-Seq and WGS techniques, suggesting its biological context is likely tied to oceanic ecosystems and metagenomic studies.

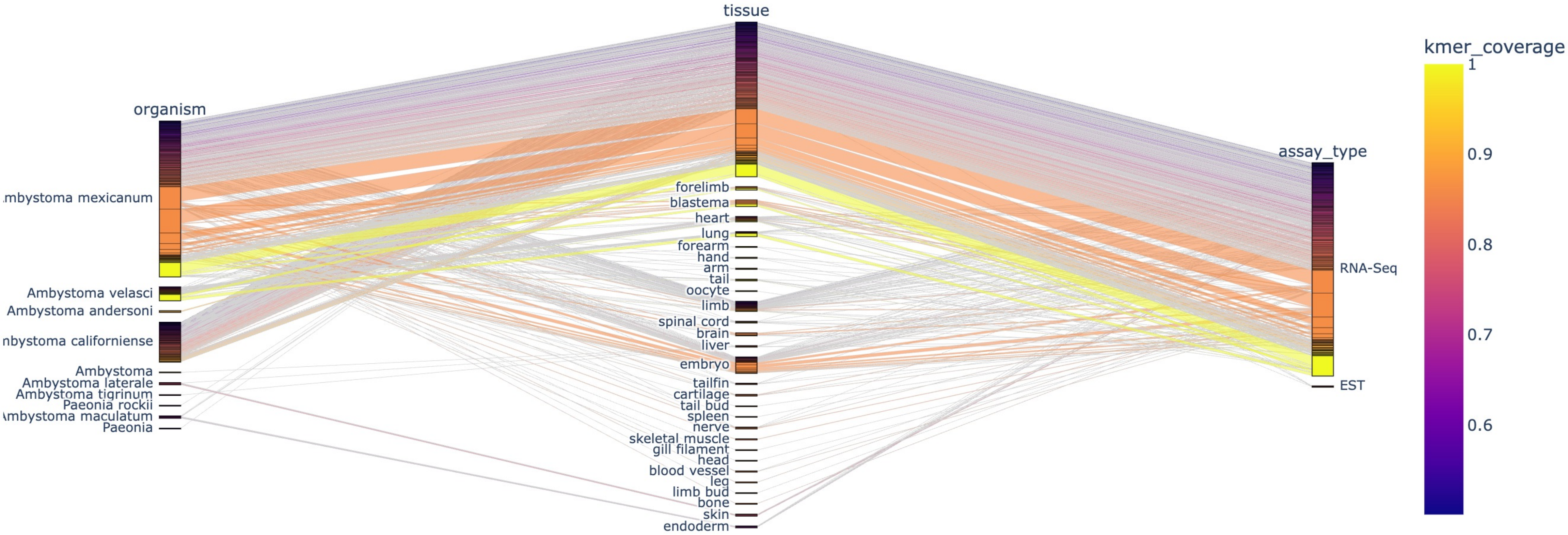
Query 1 kbp of *Pelagomonas calceolata* (point size is the number of hits per location)



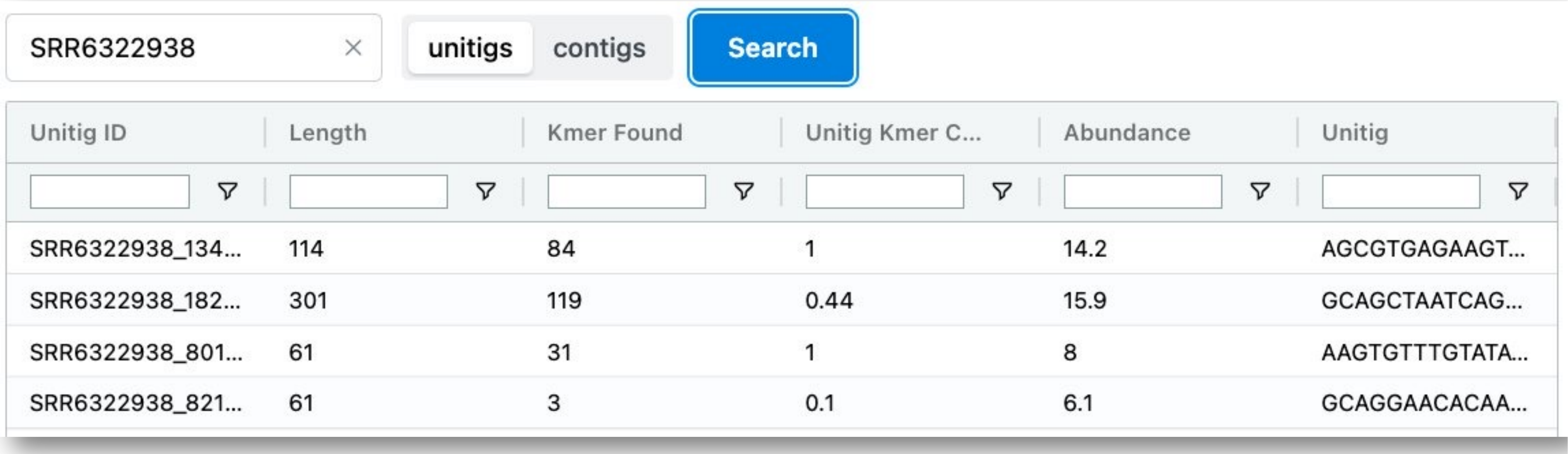
ERR598982

location_count=28
latitude=−34.8901
longitude=18.0459
location_avg_coverage=0.95

Logan-search answer: exploit metadata



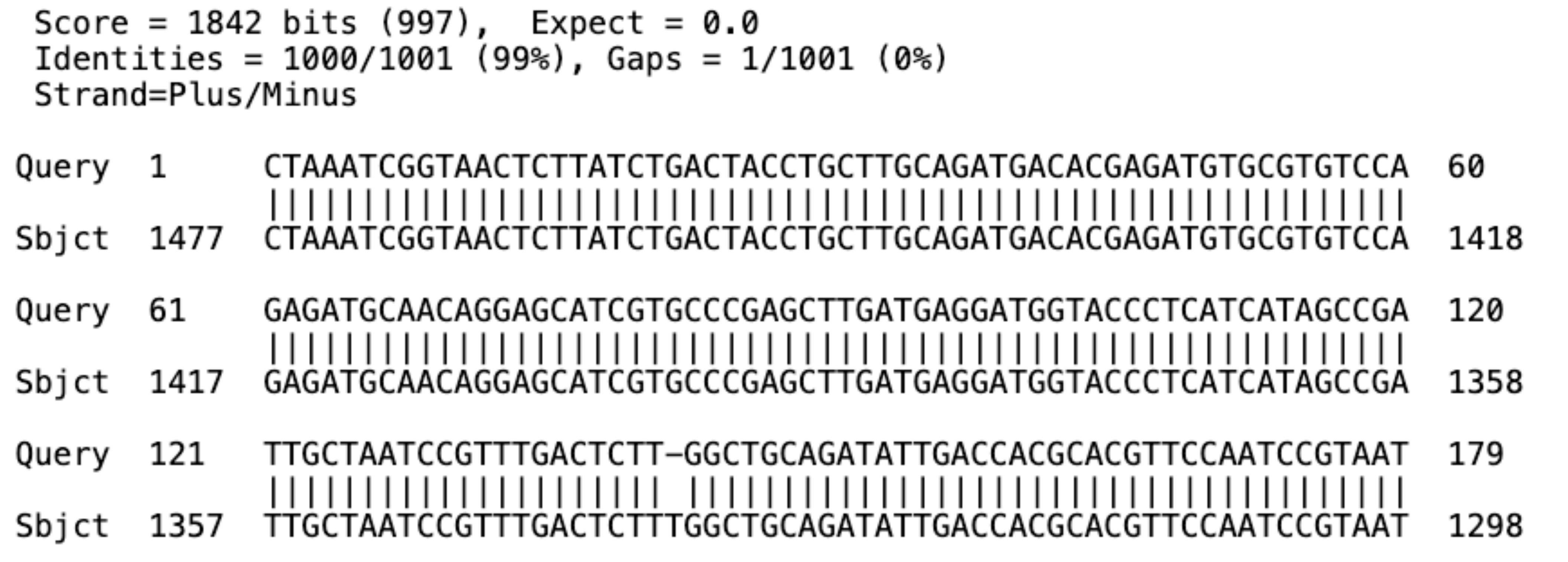
Logan-search answer: Align query to contigs or unitigs



Unitig ID	Length	Kmer Found	Unitig Kmer C...	Abundance	Unitig
SRR6322938_134...	114	84	1	14.2	AGCGTGAGAAGT...
SRR6322938_182...	301	119	0.44	15.9	GCAGCTAATCAG...
SRR6322938_801...	61	31	1	8	AAGTGTTTGTATA...
SRR6322938_821...	61	3	0.1	6.1	GCAGGAACACAA...

Anthony Baire, Pierre Marijon, Francesco Andreace and Pierre Peterlongo (2024).
Back to sequences: Find the origin of k-mers.
Journal of Open Source Software, 9(101), 7066, https://doi.org/10.21105/joss.07066

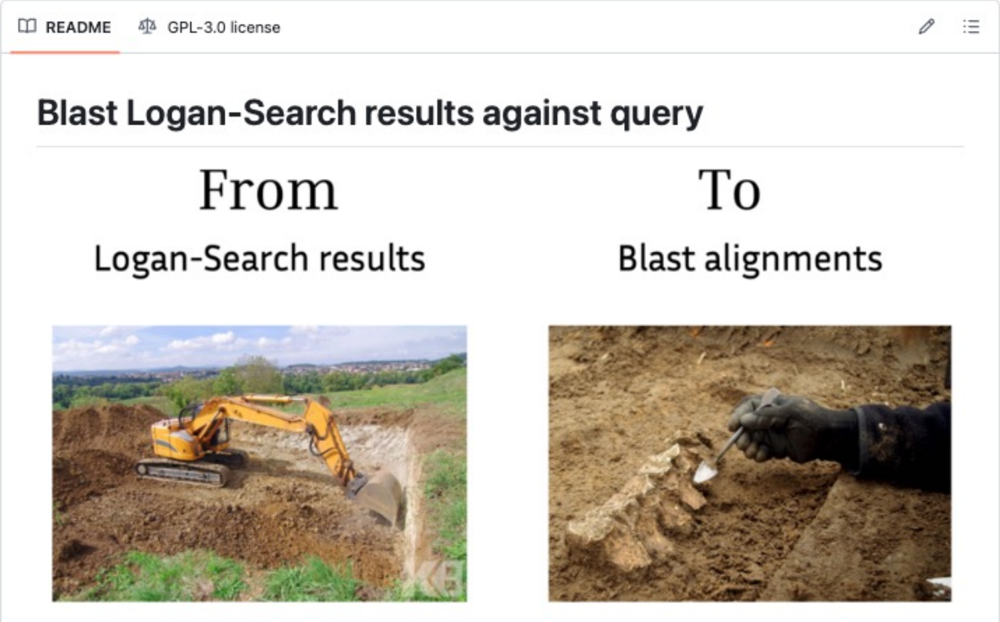
Logan-search answer: Align query to contigs or unitigs



```
 Score = 1842 bits (997),  Expect = 0.0
 Identities = 1000/1001 (99%), Gaps = 1/1001 (0%)
 Strand=Plus/Minus

Query  1     CTAAATCGGTAACTCTTATCTGACTACCTGCTTGCAGATGACACGAGATGTGCGTGTCCA  60
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  1477  CTAAATCGGTAACTCTTATCTGACTACCTGCTTGCAGATGACACGAGATGTGCGTGTCCA  1418

Query  61    GAGATGCAACAGGAGCATCGTGCCCGAGCTTGATGAGGATGGTACCCTCATCATAGCCGA  120
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  1417  GAGATGCAACAGGAGCATCGTGCCCGAGCTTGATGAGGATGGTACCCTCATCATAGCCGA  1358

Query  121   TTGCTAATCCGTTTGACTCTT-GGCTGCAGATATTGACCACGCACGTTCCAATCCGTAAT  179
             ||||||||||||||||||||| ||||||||||||||||||||||||||||||||||||||
Sbjct  1357  TTGCTAATCCGTTTGACTCTTTGGCTGCAGATATTGACCACGCACGTTCCAATCCGTAAT  1298
```

Anthony Baire, Pierre Marijon, Francesco Andreace and Pierre Peterlongo (2024).
Back to sequences: Find the origin of k-mers.
Journal of Open Source Software, 9(101), 7066, https://doi.org/10.21105/joss.07066

From logan-search results to blast



Blast Logan-Search results against query

From

Logan-Search results

To

Blast alignments

```
logan_blast.sh -s kmviz-b2bce461-ca13…
```

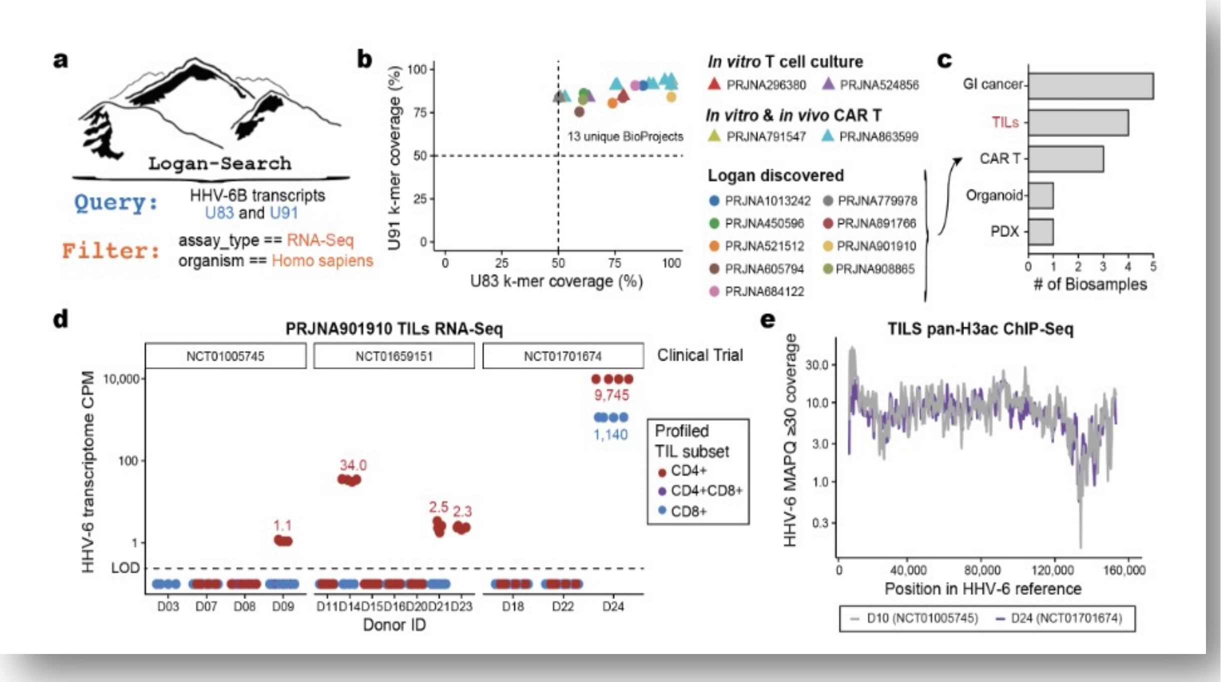
From petabytes to insights

An example of *what have we found*
(Check the paper for more)

Logan-search uncovers novel biological associations of viruses

- Query: HHV-6B transcripts (5 minutes)
- Where: human RNA-seq

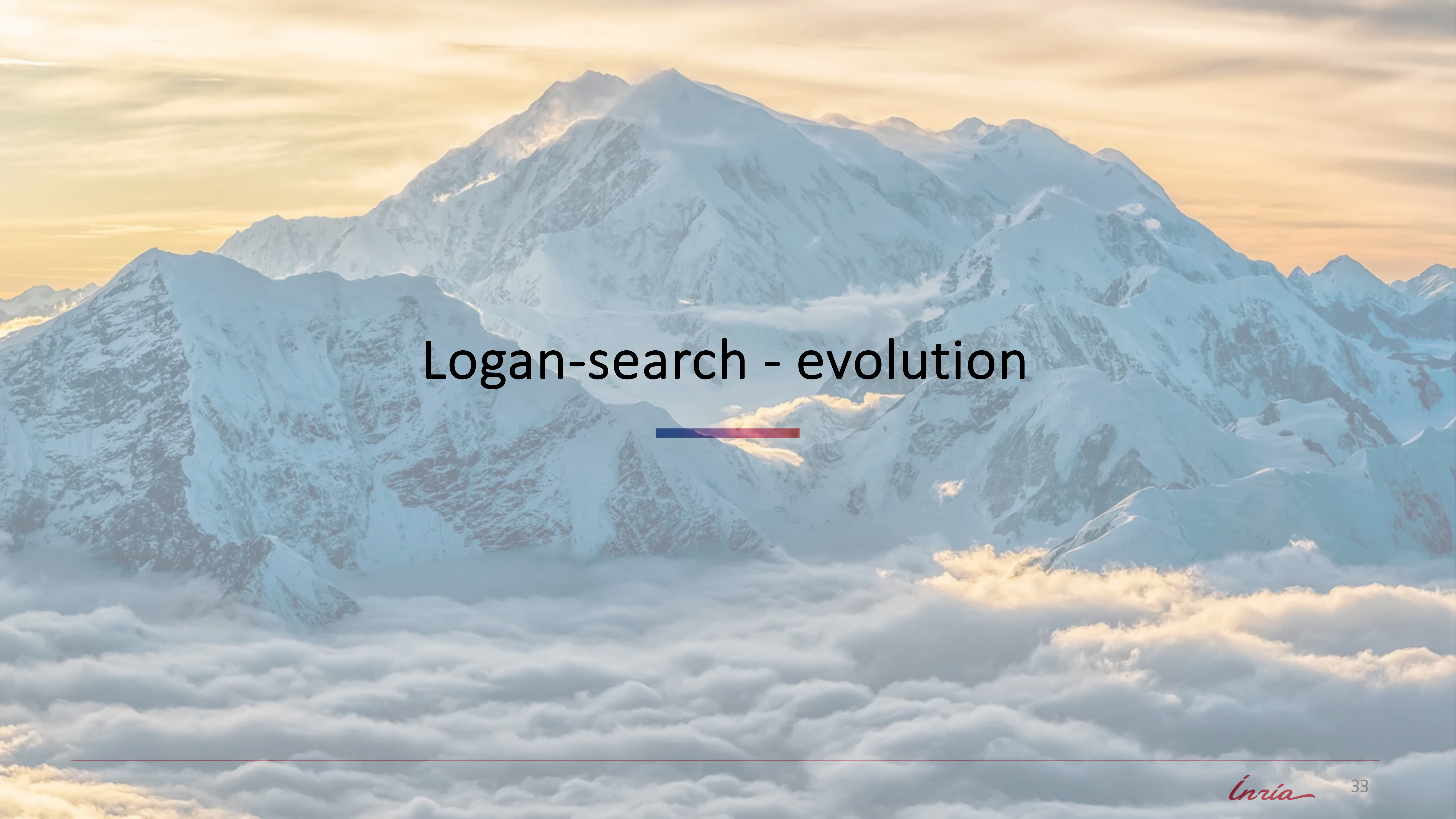
- Found:
  - 13 distinct bioprojects
  - 4 served as positive controls



Focus on biosamples about TIL therapies

(Tumor Infiltrating Lymphocytes)

- Observation: **HHV-6 reactivation in TIL therapies**



Logan-search - evolution

Logan, work in progress

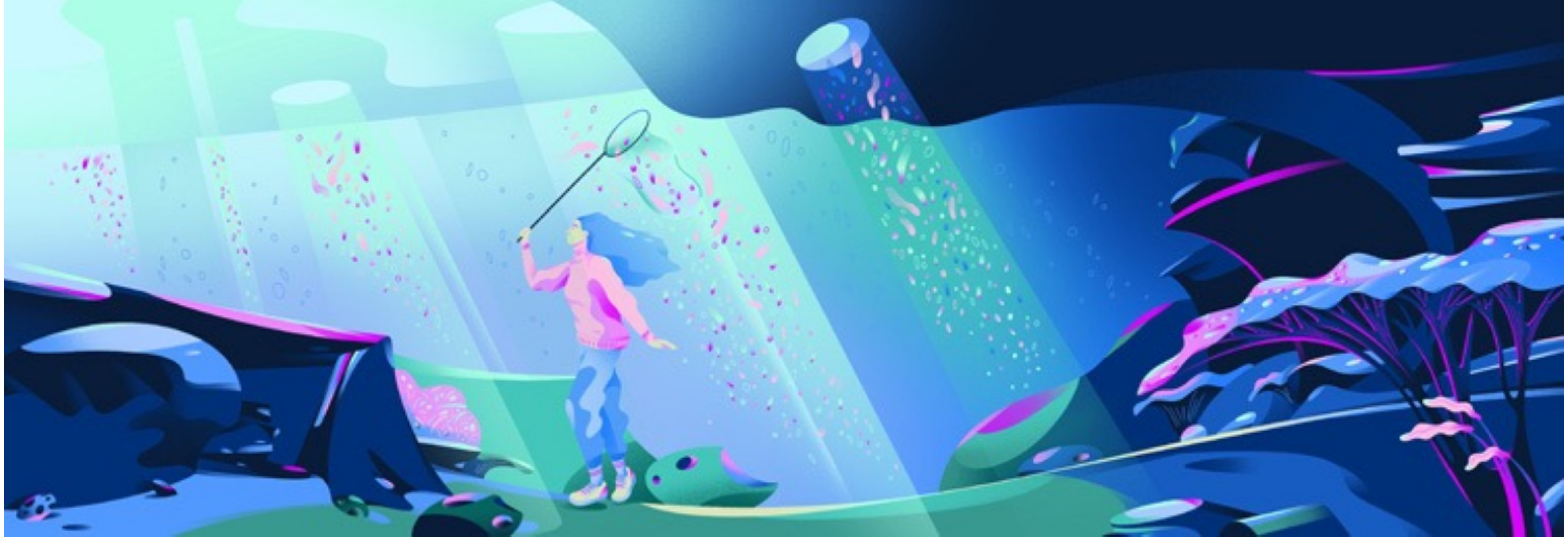
Alix Regnier
Sebastien Bellenous

Today

- Not compressed
  - Whole index: 1PB
- Housed by Microsoft Azure
  - File system not adapted
  - One query: 1kb max, 5minutes
- Indexes SRA as of Dec 2023.
  - Almost doubled
- LLM Analyze:
  - Based on metadata

Tomorrow

- Compressed
  - Whole compressed index: 300TB (?)
- Housed by TACC
  - Galaxy
  - Hope: instant and bigger queries
- Updated unitigs/contigs and index
- LLM Analyze
  - Based on scientific papers
  - Interactive



@Manon Sauzara

Thanks!!!

Logan-Search: https://logan-search.org/

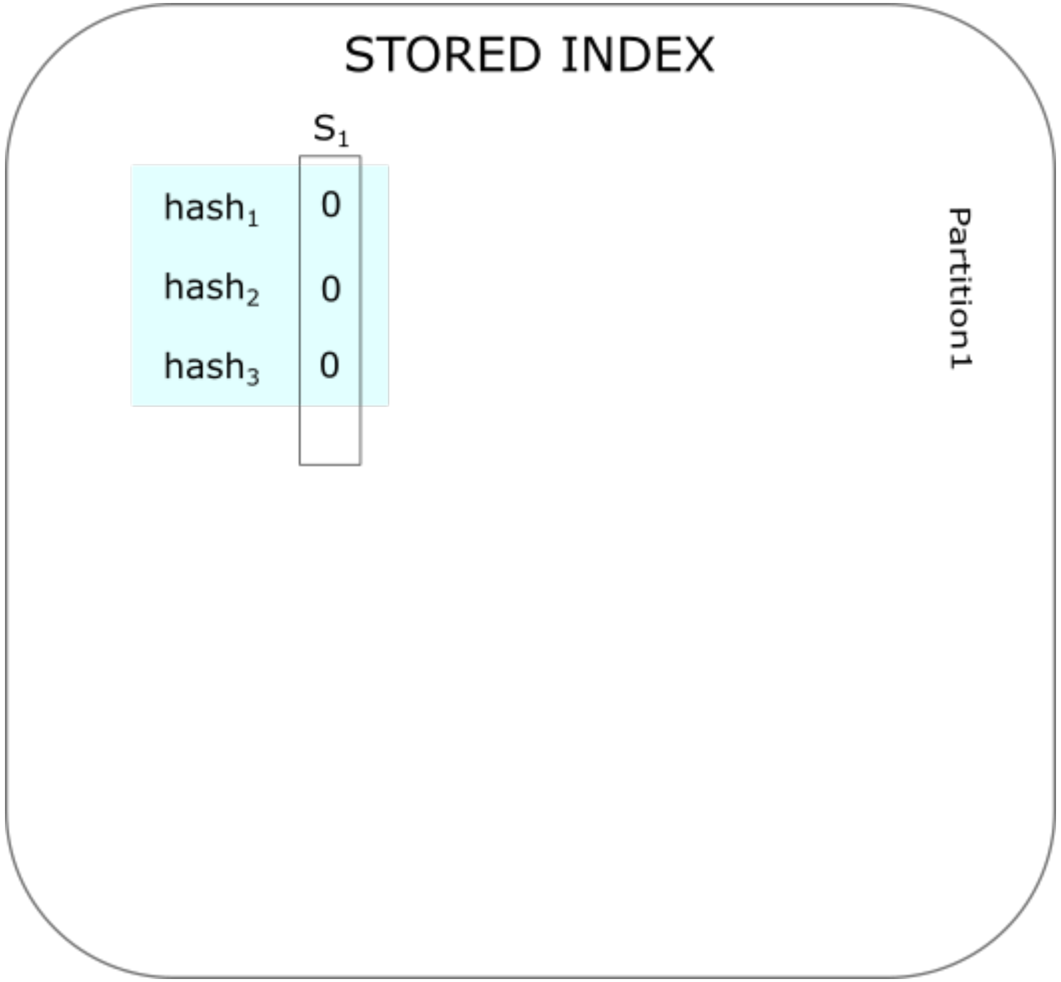
Logan-Search to Blast: github.com/pierrepeterlongo/blast_logan_search_results

Pierre Peterlongo

BONUS

A few technical details about kmindex construction and structure

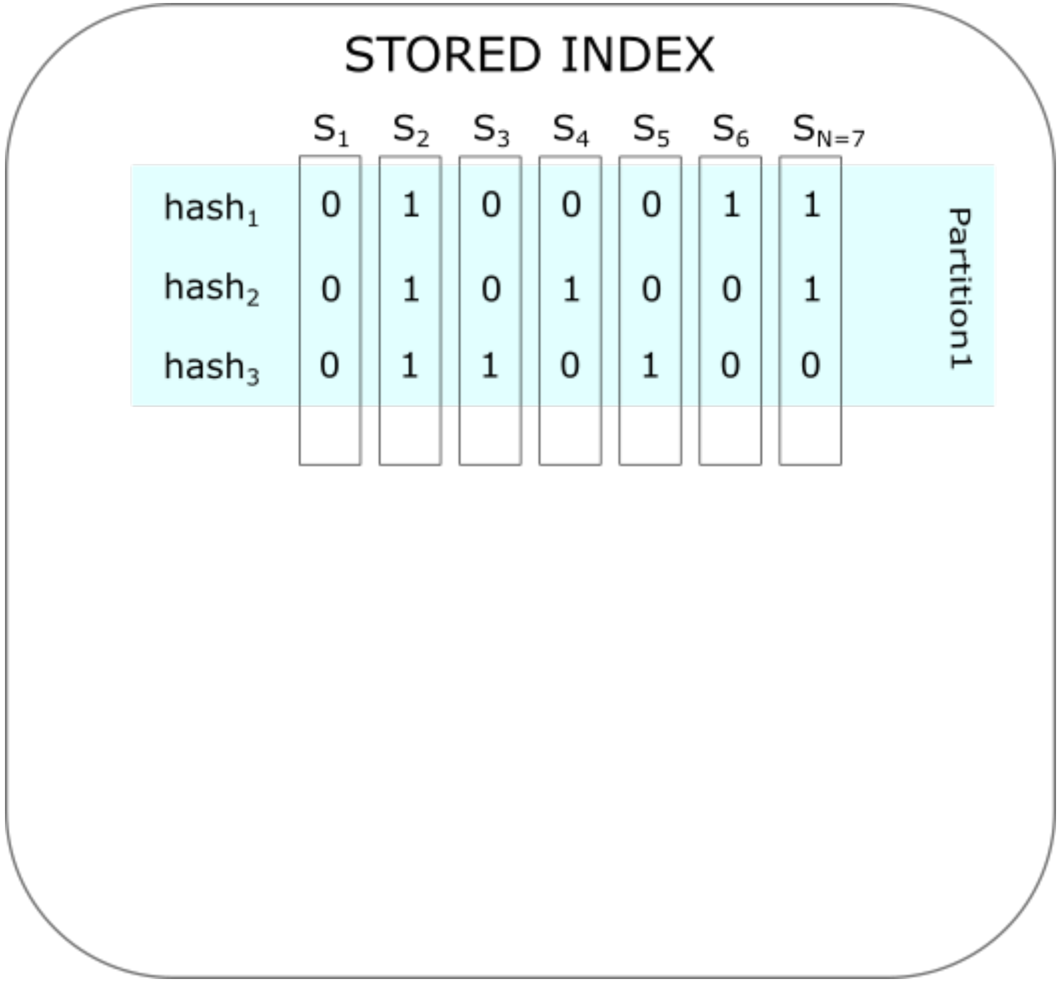
- If we have time…



STORED INDEX

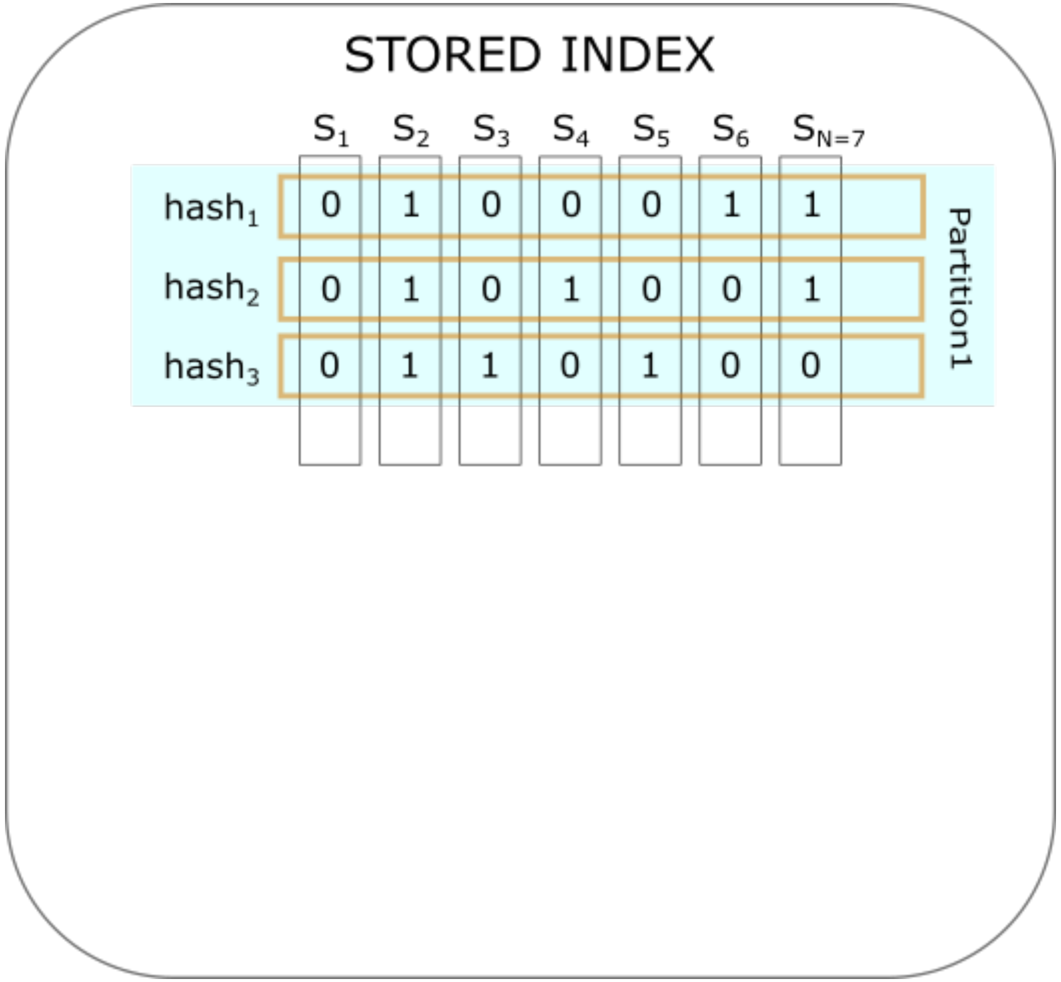
	S_1
$hash_1$	0
$hash_2$	0
$hash_3$	0

Partition1



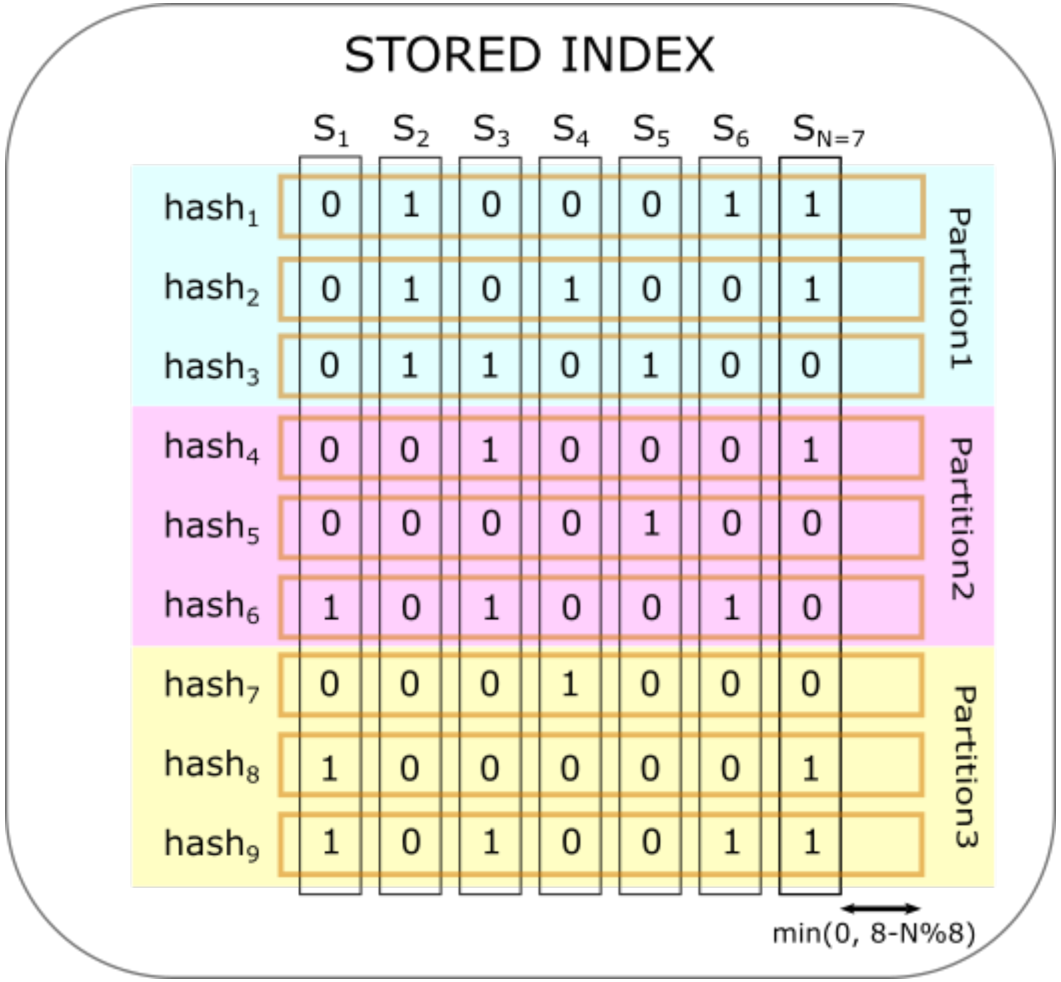
STORED INDEX

	S_1	S_2	S_3	S_4	S_5	S_6	$S_{N=7}$	
$hash_1$	0	1	0	0	0	1	1	Partition1
$hash_2$	0	1	0	1	0	0	1	
$hash_3$	0	1	1	0	1	0	0	



STORED INDEX

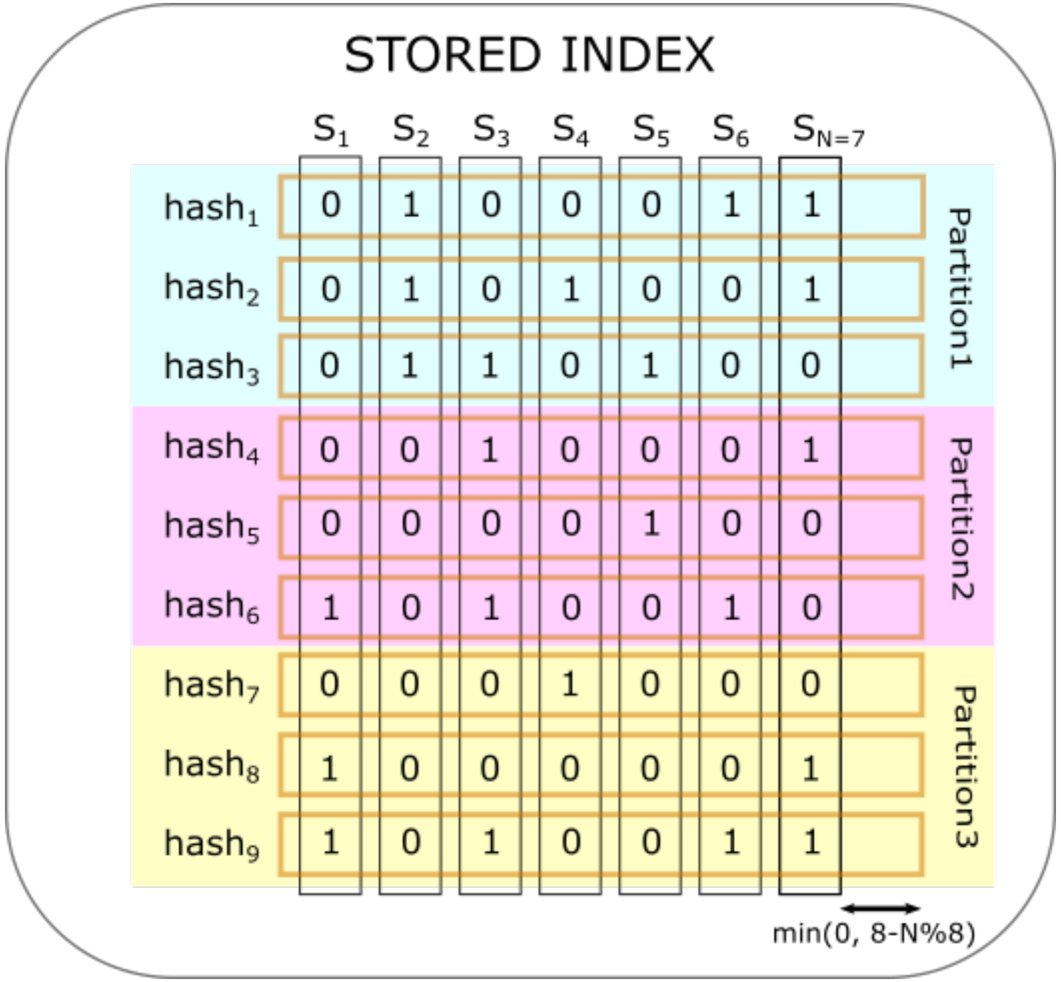
	S_1	S_2	S_3	S_4	S_5	S_6	$S_{N=7}$
hash_1	0	1	0	0	0	1	1
hash_2	0	1	0	1	0	0	1
hash_3	0	1	1	0	1	0	0



STORED INDEX

	S_1	S_2	S_3	S_4	S_5	S_6	$S_{N=7}$	
$hash_1$	0	1	0	0	0	1	1	Partition1
$hash_2$	0	1	0	1	0	0	1	
$hash_3$	0	1	1	0	1	0	0	
$hash_4$	0	0	1	0	0	0	1	Partition2
$hash_5$	0	0	0	0	1	0	0	
$hash_6$	1	0	1	0	0	1	0	
$hash_7$	0	0	0	1	0	0	0	Partition3
$hash_8$	1	0	0	0	0	0	1	
$hash_9$	1	0	1	0	0	1	1	

How to distribute k-mers into partitions?



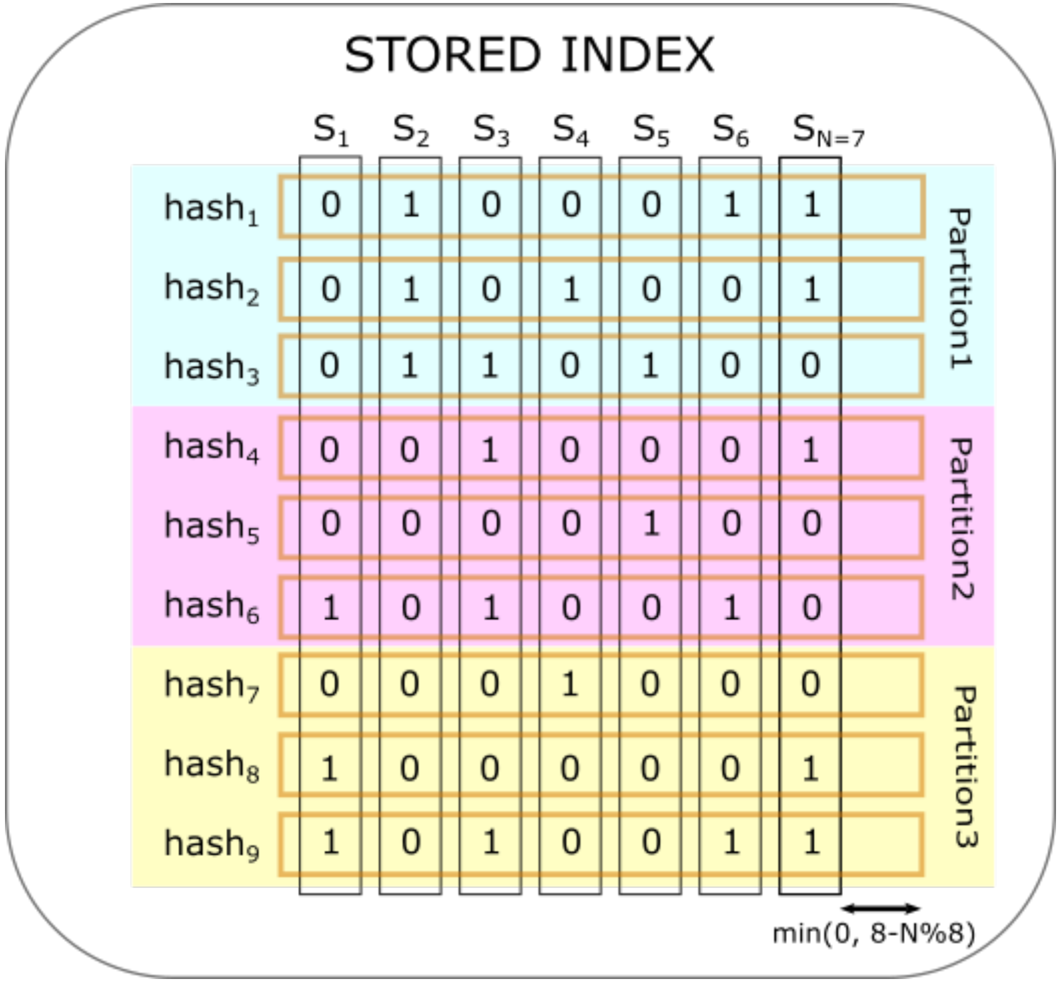
STORED INDEX

	S_1	S_2	S_3	S_4	S_5	S_6	$S_{N=7}$	
$hash_1$	0	1	0	0	0	1	1	Partition1
$hash_2$	0	1	0	1	0	0	1	
$hash_3$	0	1	1	0	1	0	0	
$hash_4$	0	0	1	0	0	0	1	Partition2
$hash_5$	0	0	0	0	1	0	0	
$hash_6$	1	0	1	0	0	1	0	
$hash_7$	0	0	0	1	0	0	0	Partition3
$hash_8$	1	0	0	0	0	0	1	
$hash_9$	1	0	1	0	0	1	1	

How to distribute k-mers into partitions?

ATAACTCGACA -> ?

$\boxed{k = 11, m = 4}$

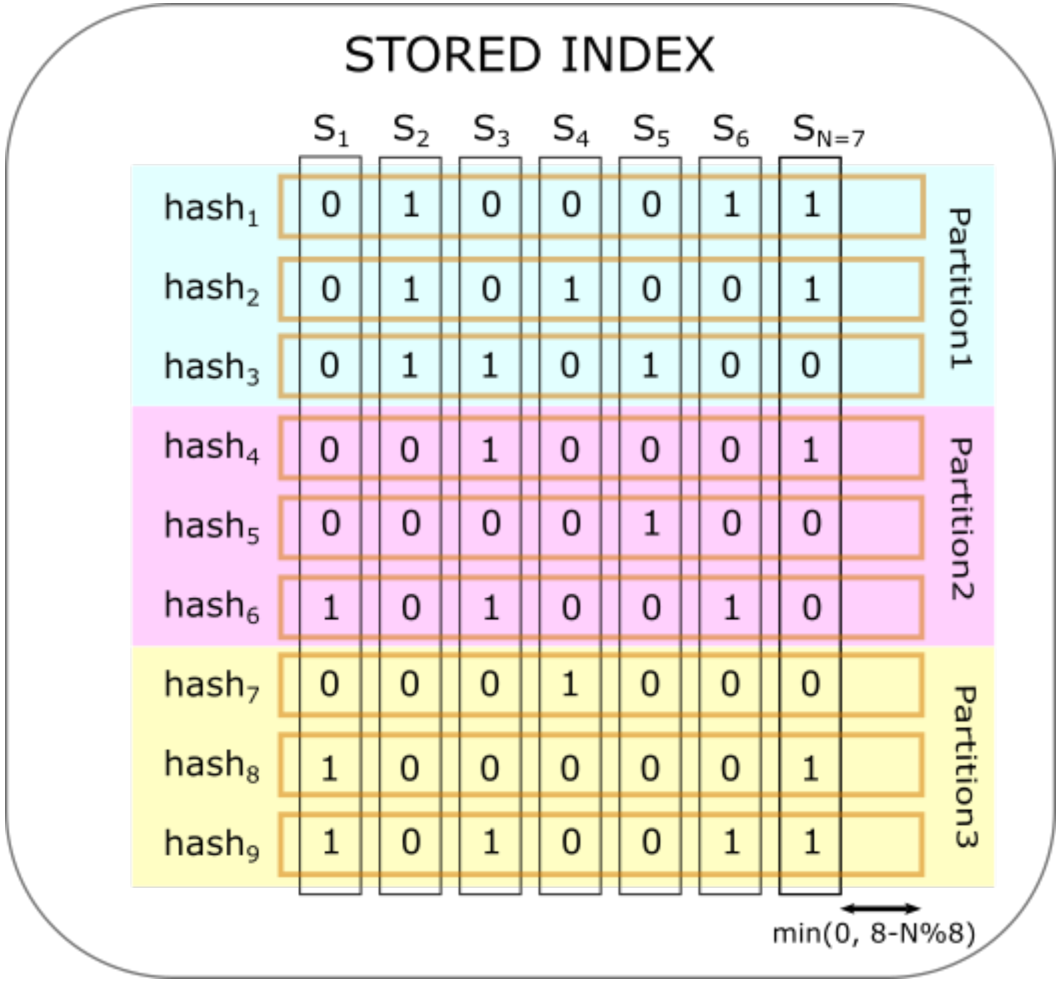


	S_1	S_2	S_3	S_4	S_5	S_6	$S_{N=7}$
$hash_1$	0	1	0	0	0	1	1
$hash_2$	0	1	0	1	0	0	1
$hash_3$	0	1	1	0	1	0	0
$hash_4$	0	0	1	0	0	0	1
$hash_5$	0	0	0	0	1	0	0
$hash_6$	1	0	1	0	0	1	0
$hash_7$	0	0	0	1	0	0	0
$hash_8$	1	0	0	0	0	0	1
$hash_9$	1	0	1	0	0	1	1

How to distribute k-mers into partitions?

ATAACTCGACA -> AACT
TAACTCGACAT -> ?

$\boxed{k = 11, m = 4}$



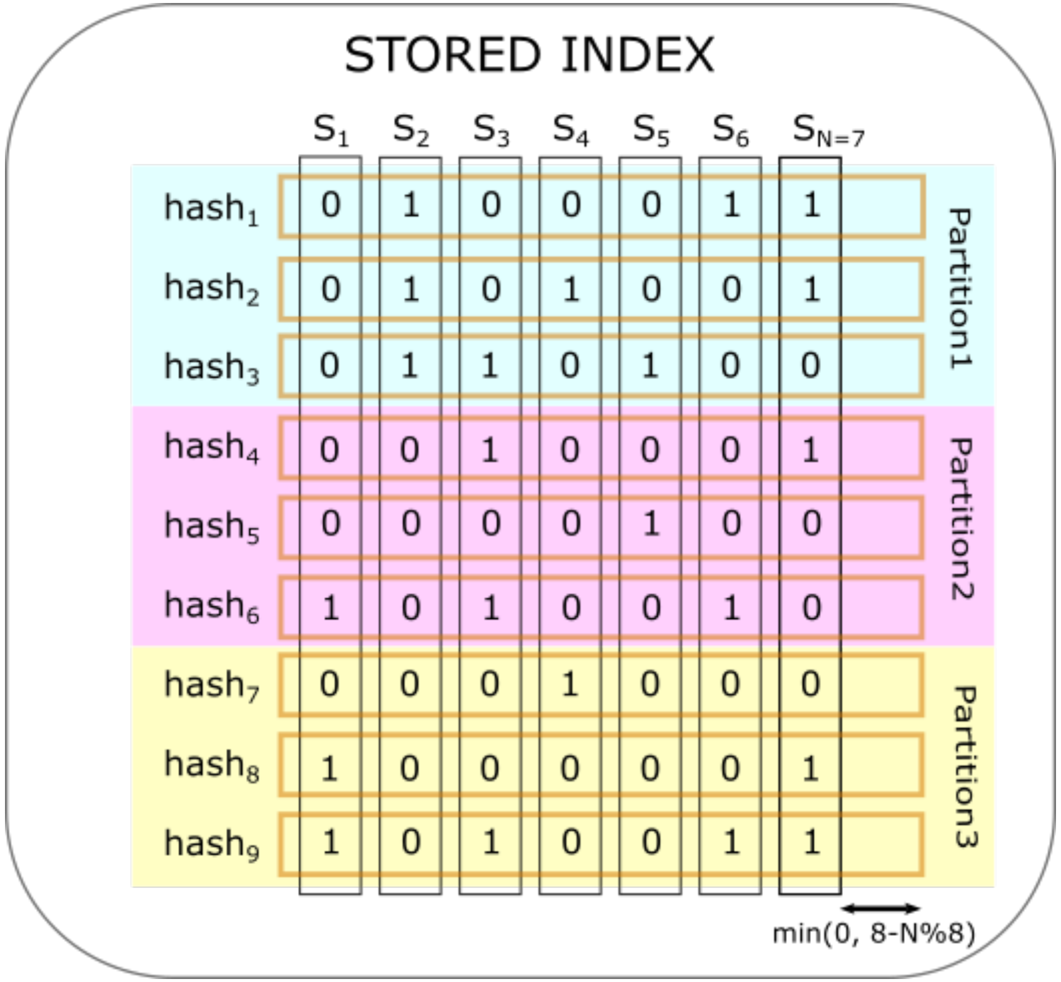
STORED INDEX

	S_1	S_2	S_3	S_4	S_5	S_6	$S_{N=7}$	
$hash_1$	0	1	0	0	0	1	1	Partition1
$hash_2$	0	1	0	1	0	0	1	
$hash_3$	0	1	1	0	1	0	0	
$hash_4$	0	0	1	0	0	0	1	Partition2
$hash_5$	0	0	0	0	1	0	0	
$hash_6$	1	0	1	0	0	1	0	
$hash_7$	0	0	0	1	0	0	0	Partition3
$hash_8$	1	0	0	0	0	0	1	
$hash_9$	1	0	1	0	0	1	1	

How to distribute k-mers into partitions?

AT**AACT**CGACA -> AACT
T**AACT**CGACAT -> AACT
AACTCGACATA -> AACT
ACTCG**ACAT**AG -> ACAT
CTCG**ACAT**AGT -> ACAT
TCG**ACAT**AGTA -> ACAT
...

$\boxed{k = 11, m = 4}$



STORED INDEX

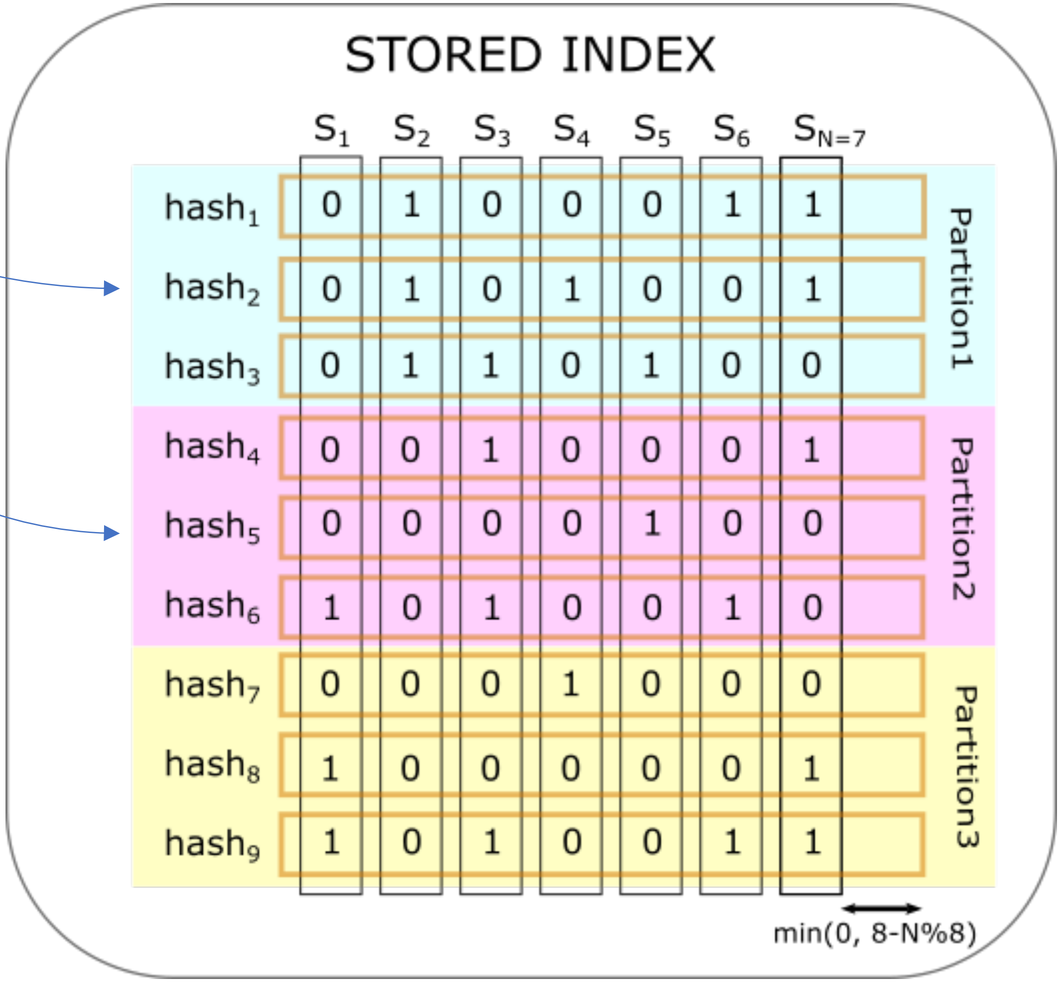
	S_1	S_2	S_3	S_4	S_5	S_6	$S_{N=7}$	
$hash_1$	0	1	0	0	0	1	1	Partition1
$hash_2$	0	1	0	1	0	0	1	
$hash_3$	0	1	1	0	1	0	0	
$hash_4$	0	0	1	0	0	0	1	Partition2
$hash_5$	0	0	0	0	1	0	0	
$hash_6$	1	0	1	0	0	1	0	
$hash_7$	0	0	0	1	0	0	0	Partition3
$hash_8$	1	0	0	0	0	0	1	
$hash_9$	1	0	1	0	0	1	1	

How to distribute k-mers into partitions?

```
ATAACTCGACA        ->  AACT
 TAACTCGACAT       ->  AACT
  AACTCGACATA      ->  AACT
   ACTCGACATAG     ->  ACAT
    CTCGACATAGT    ->  ACAT
     TCGACATAGTA   ->  ACAT
```

...

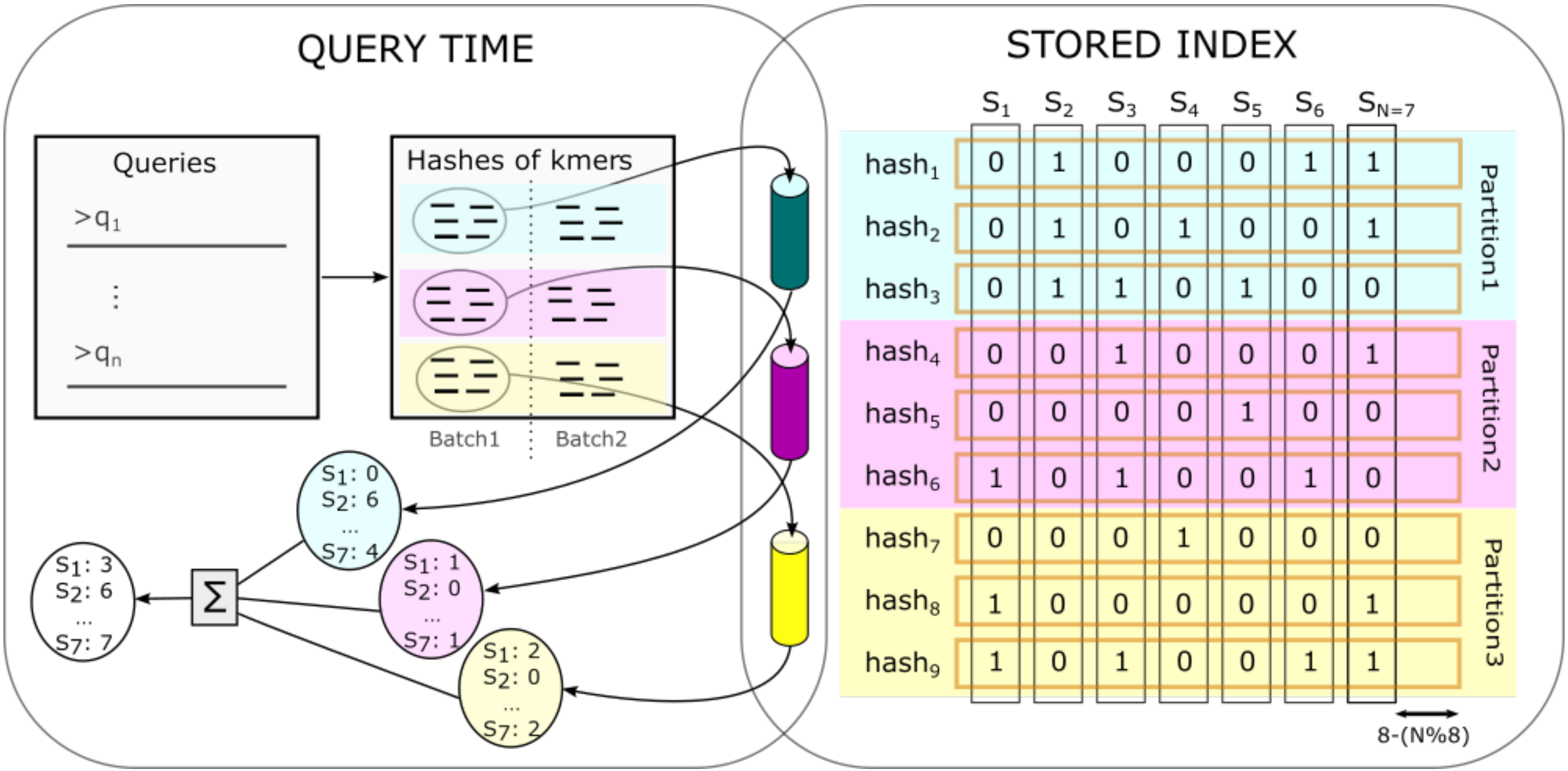
$\boxed{k = 11, m = 4}$



STORED INDEX

	S_1	S_2	S_3	S_4	S_5	S_6	$S_{N=7}$	
$hash_1$	0	1	0	0	0	1	1	Partition1
$hash_2$	0	1	0	1	0	0	1	
$hash_3$	0	1	1	0	1	0	0	
$hash_4$	0	0	1	0	0	0	1	Partition2
$hash_5$	0	0	0	0	1	0	0	
$hash_6$	1	0	1	0	0	1	0	
$hash_7$	0	0	0	1	0	0	0	Partition3
$hash_8$	1	0	0	0	0	0	1	
$hash_9$	1	0	1	0	0	1	1	

min(0, 8-N%8)



	S_1	S_2	S_3	S_4	S_5	S_6	$S_{N=7}$	
$hash_1$	0	1	0	0	0	1	1	Partition1
$hash_2$	0	1	0	1	0	0	1	
$hash_3$	0	1	1	0	1	0	0	
$hash_4$	0	0	1	0	0	0	1	Partition2
$hash_5$	0	0	0	0	1	0	0	
$hash_6$	1	0	1	0	0	1	0	
$hash_7$	0	0	0	1	0	0	0	Partition3
$hash_8$	1	0	0	0	0	0	1	
$hash_9$	1	0	1	0	0	1	1	

Back to sequences: Find the origin of k-mers

https://github.com/pierrepeterlongo/back_to_sequences

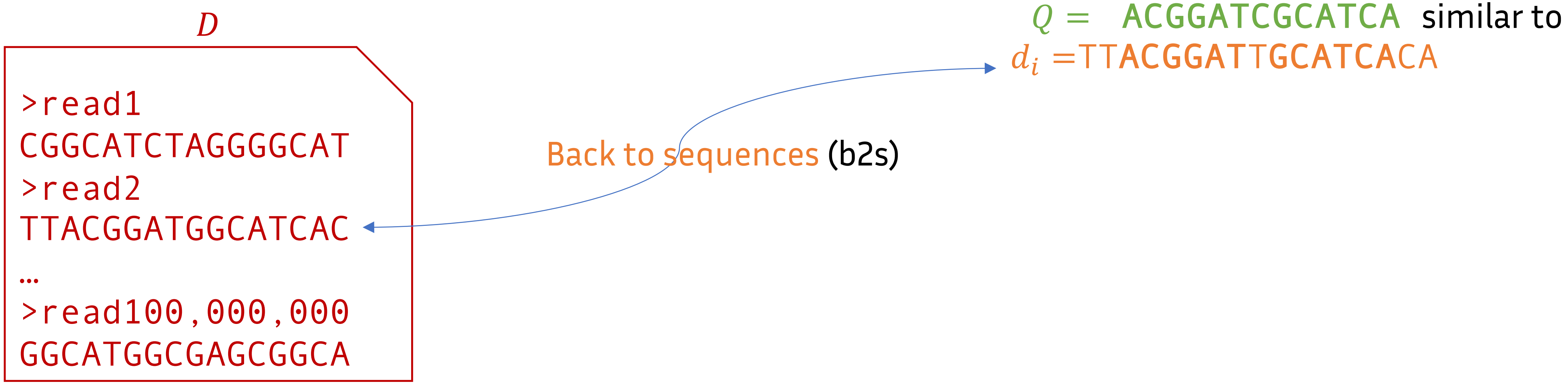
Baire et al., (2024). Back to sequences: Find the origin of k-mers. Journal of Open Source Software, 9(101), 7066, https://doi.org/10.21105/joss.07066

Find similar **sequences**

Kmindex enables to know to which dataset D my query Q is similar

"Super, but Q is similar to which sequences d_i from D?"

$Q =$ ACGGATCGCATCA



D

```
>read1
CGGCATCTAGGGGCAT
>read2
TTACGGATGGCATCAC
…
>read100,000,000
GGCATGGCGAGCGGCA
```

Back to sequences (b2s)

$Q =$ ACGGATCGCATCA similar to
d_i =TTACGGATTGCATCACA

Back to sequences

- IN:
  - A query Q (seen as a set of kmers)
  - A bank D
- OUT:
  - Sequences d_i from the bank similar to the query

- Optionally:
  - Abundance of kmers from Q in D
  - Mapping positions of kmers from Q in each d_i