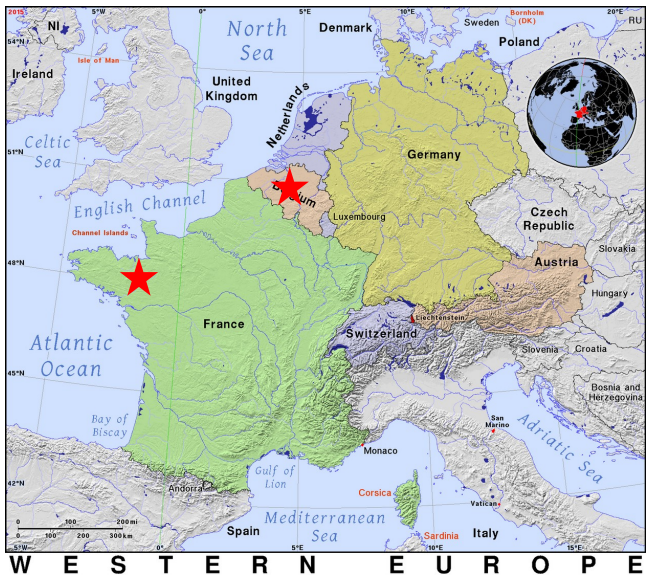
About me: Ph.D., 2021-2024



Jean-François Flot
Université Libre de Bruxelles
Focus: assembling wild genomes

Dominique Laveneir
Université de Rennes
Focus: computational methods

Metagenome assembly from long reads

About me: postdoc, since 2025

Institut Pasteur, Paris

Rayan Chikhi
Institut Pasteur, Paris
Focus: massive genomics

Index & Search the Logan database

The Logan project: indexing and querying **all** the (meta)genomic data ever published

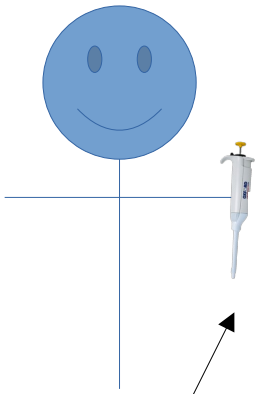
Roland Faure[1]

[1]Institut Pasteur

December 2025

The Logan database

J. Doe sequenced something



DNA/RNA

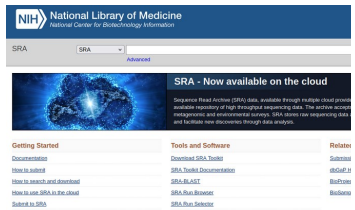
Sequencer

Sequencing Data

```
ACAGCAGT
CAGTCGTA
CGTATAA
```

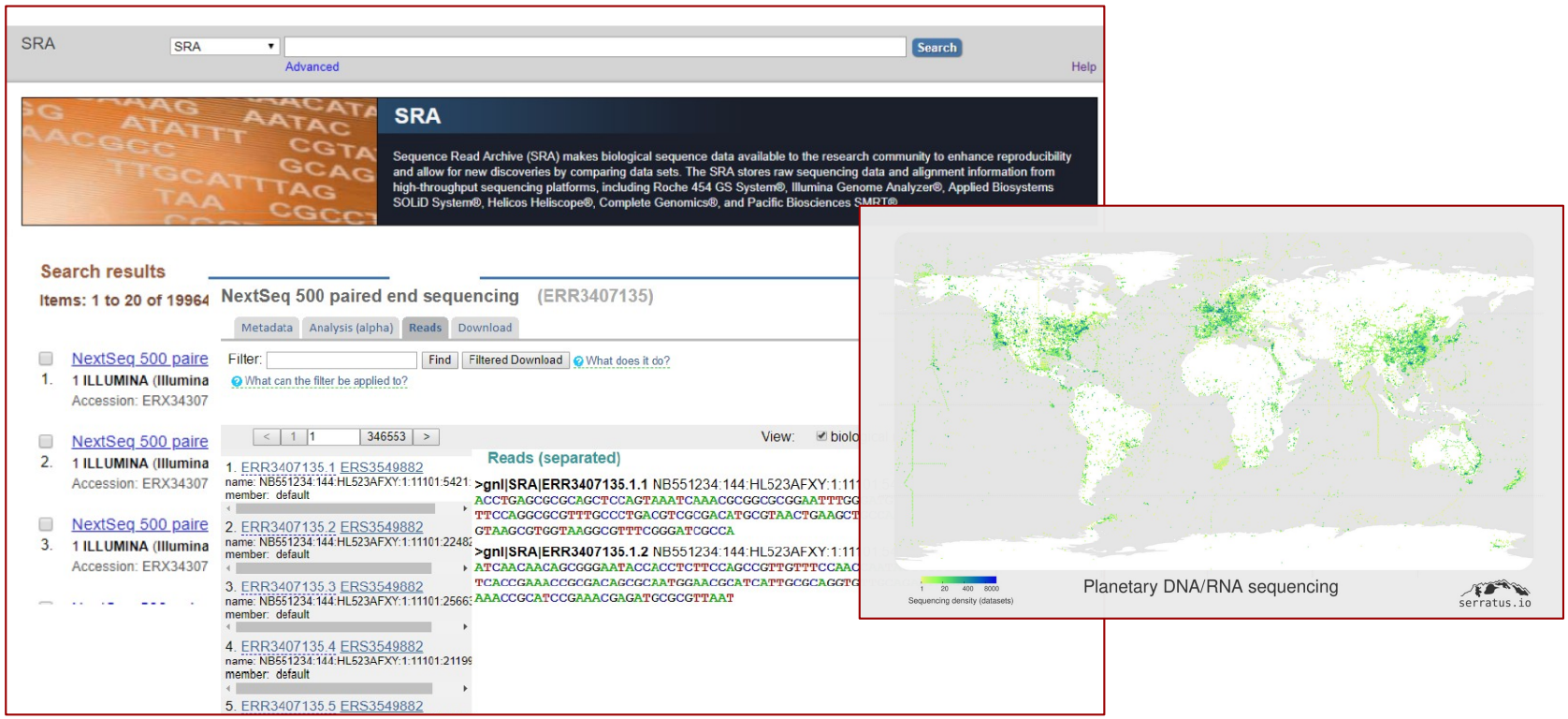
Article

Supporting data
e.g. SRR12345



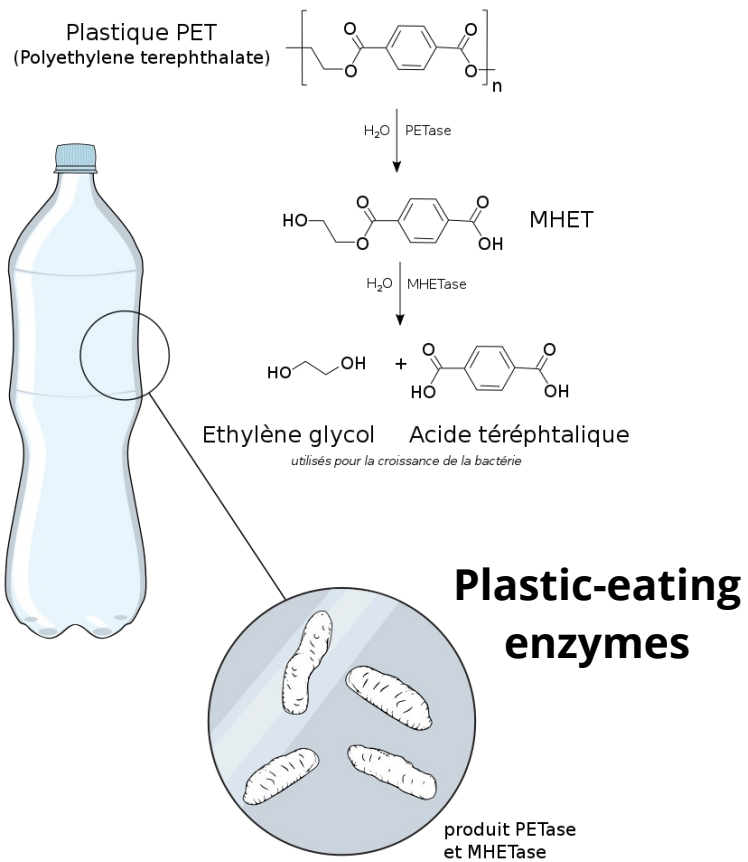
The SRA database

SRA: All public sequencing reads, **50 PBases** (as of Dec 2023)

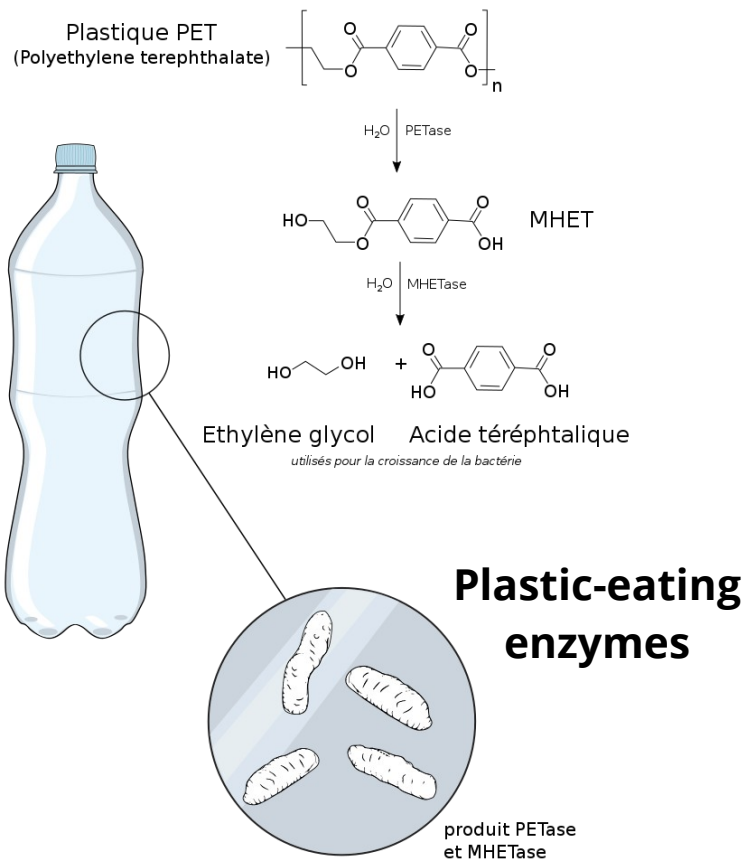


Slide Credits: Teo Lemane

Plastic-eating enzymes: PETases



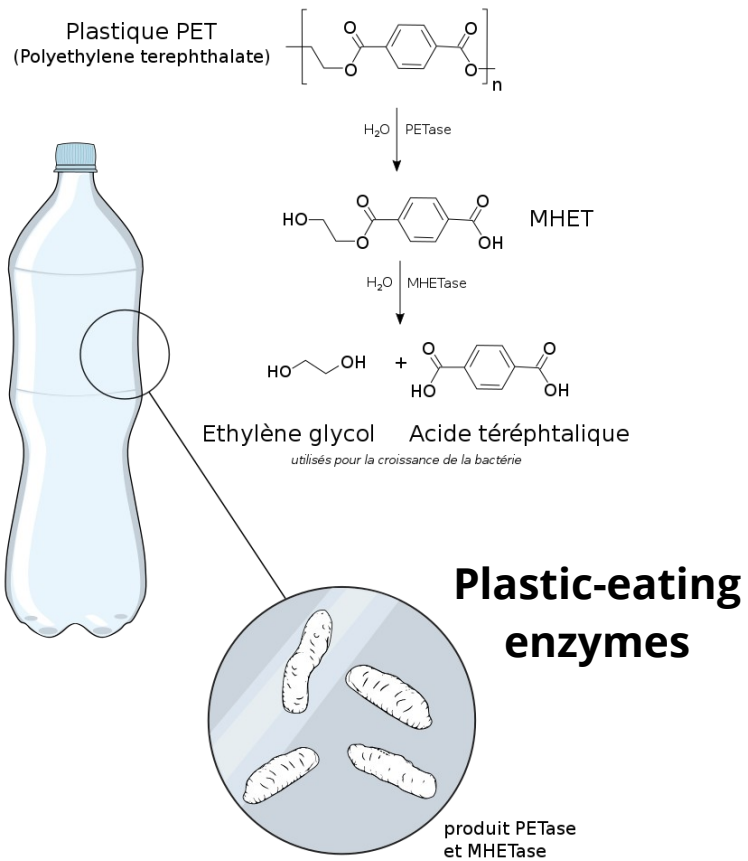
Plastic-eating enzymes: PETases



We know of 200 such enzymes but there is much we ignore, and they are hard to find

Artem Babaian

Plastic-eating enzymes: PETases

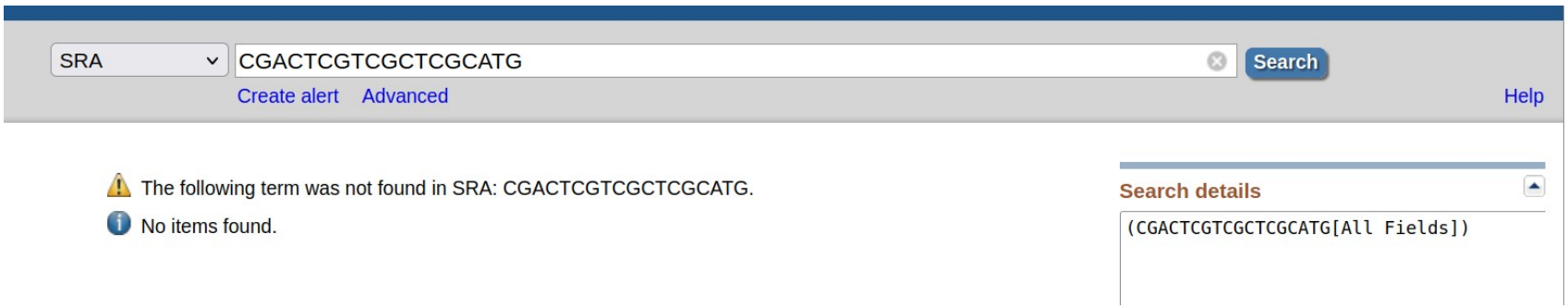


We know of 200 such enzymes but there is much we ignore, and they are hard to find

There must be more of them in the SRA!

Artem Babaian

The SRA is not queryable



Quiz: What is the size of SRA?

A: 50 TB
(all Wikipedia text in every language)

B: 500 TB
(all books ever written in human history)

C: 5 PB
(all movies ever made in 1080p)

D: 50 PB
(Internet indexed by Google in 2016)

Quiz: What is the size of SRA?

A: 50 TB
(all Wikipedia text in every language)

B: 500 TB
(all books ever written in human history)

C: 5 PB
(all movies ever made in 1080p)

D: 50 PB
(Internet indexed by Google in 2016)

The SRA now

Slide Credits: Teo Lemane

- 27 millions accessions

The Logan database

27 millions accessions, **50PB**

For all accessions

Raw reads → Contigs/Unitigs

```
ACAGCAGT
CAGTCGTA
CGTATAA
```

→

```
ACAGCAGT
    CAGTCGTA
        CGTATAA
```

→ `ACAGCAGTCGTATAA`

Logan

27 millions accessions, **~1PB**

How much did it cost to assemble Logan?

A: $5,000

B: $50,000

C: $500,000

D: $5,000,000

How much did it cost to assemble Logan?

A: $5,000

B: $50,000

C: $500,000

D: $5,000,000

The Logan database

- 2.18 million parallel CPUs, 30h wall-clock time
- All the assemblies are available online

The Logan database

- 2.18 million parallel CPUs, 30h wall-clock time
- All the assemblies are available online

Downloading

To download one accession, type:

```
wget https://s3.amazonaws.com/logan-pub/c/[accession]/[accession].contigs.fa.zst
```

The Logan database

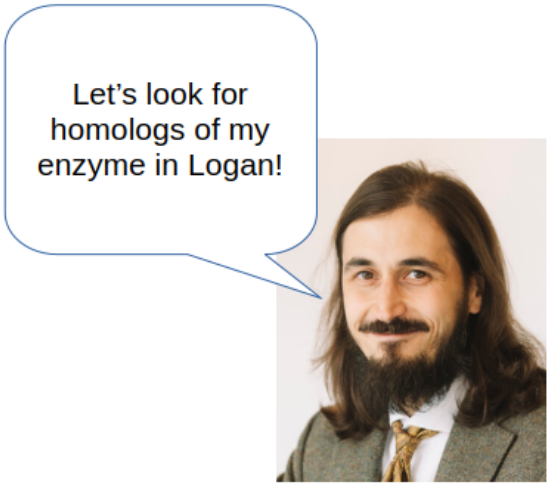
- 2.18 million parallel CPUs, 30h wall-clock time
- All the assemblies are available online

Downloading

To download one accessi

```
wget https://s3.amazo
```

```
a.zst
```



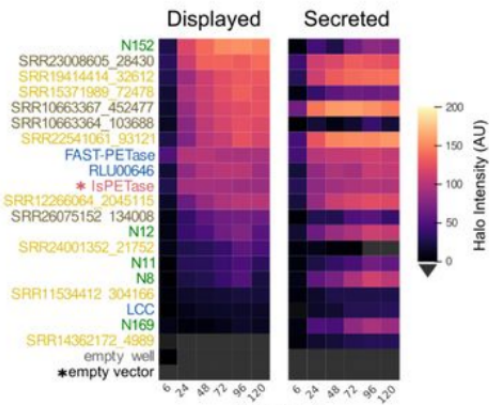
Artem Babaian

Searching through Logan

- Downloaded the 1PB of contigs
- Aligned the known enzymes against the contigs (DIAMOND)

Searching through Logan

- Downloaded the 1PB of contigs
- Aligned the known enzymes against the contigs (DIAMOND)
- 1.12 billion hits, 215 million clusters 90% identity
- Some discovered enzyme have better activity than known ones



How much did it cost to align on Logan?

A: $10

B: $100

C: $1,000

D: $10,000

How much did it cost to align on Logan?

A: $10

B: $100

C: $1,000

D: $10,000

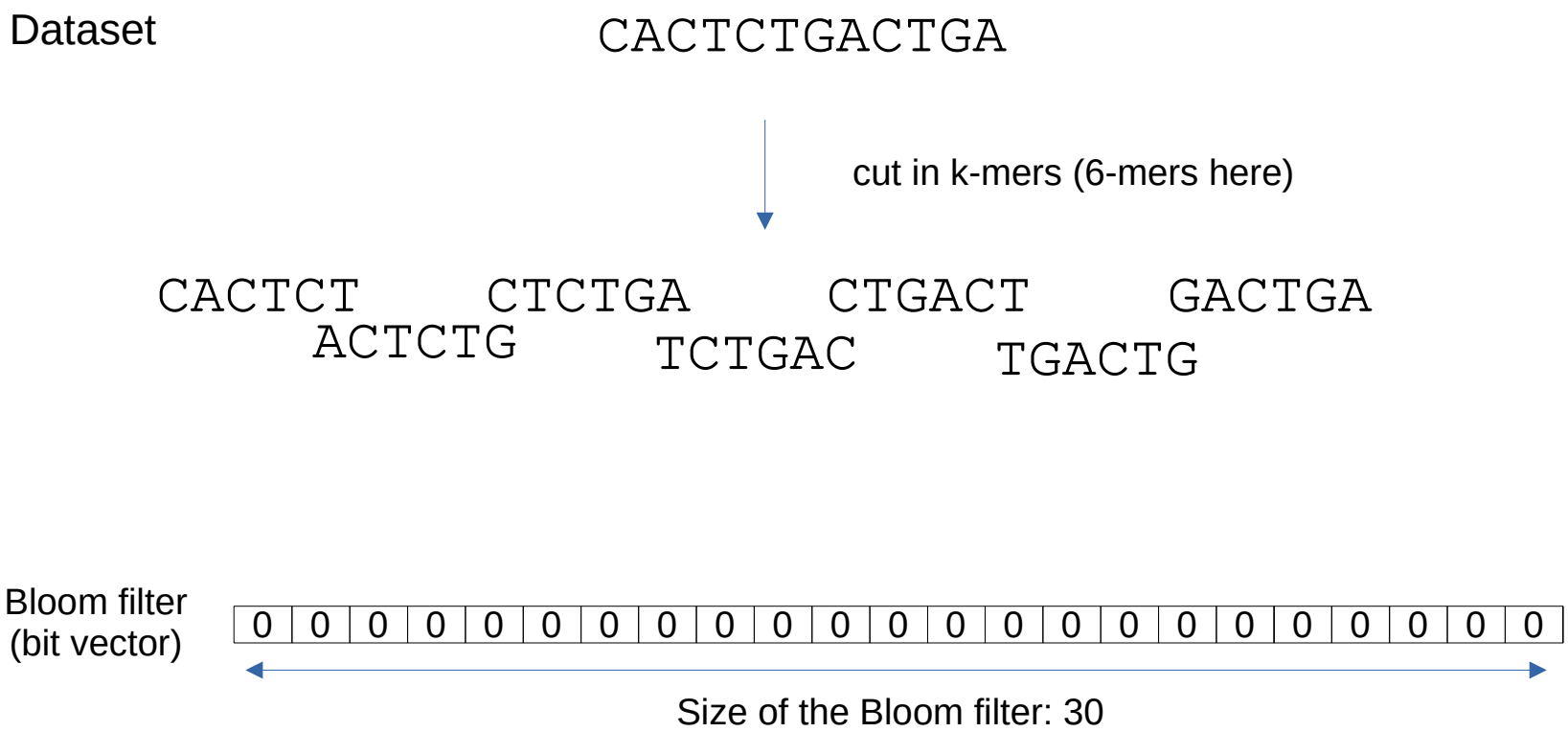
Indexing nucleotides

Let's index nucleotides

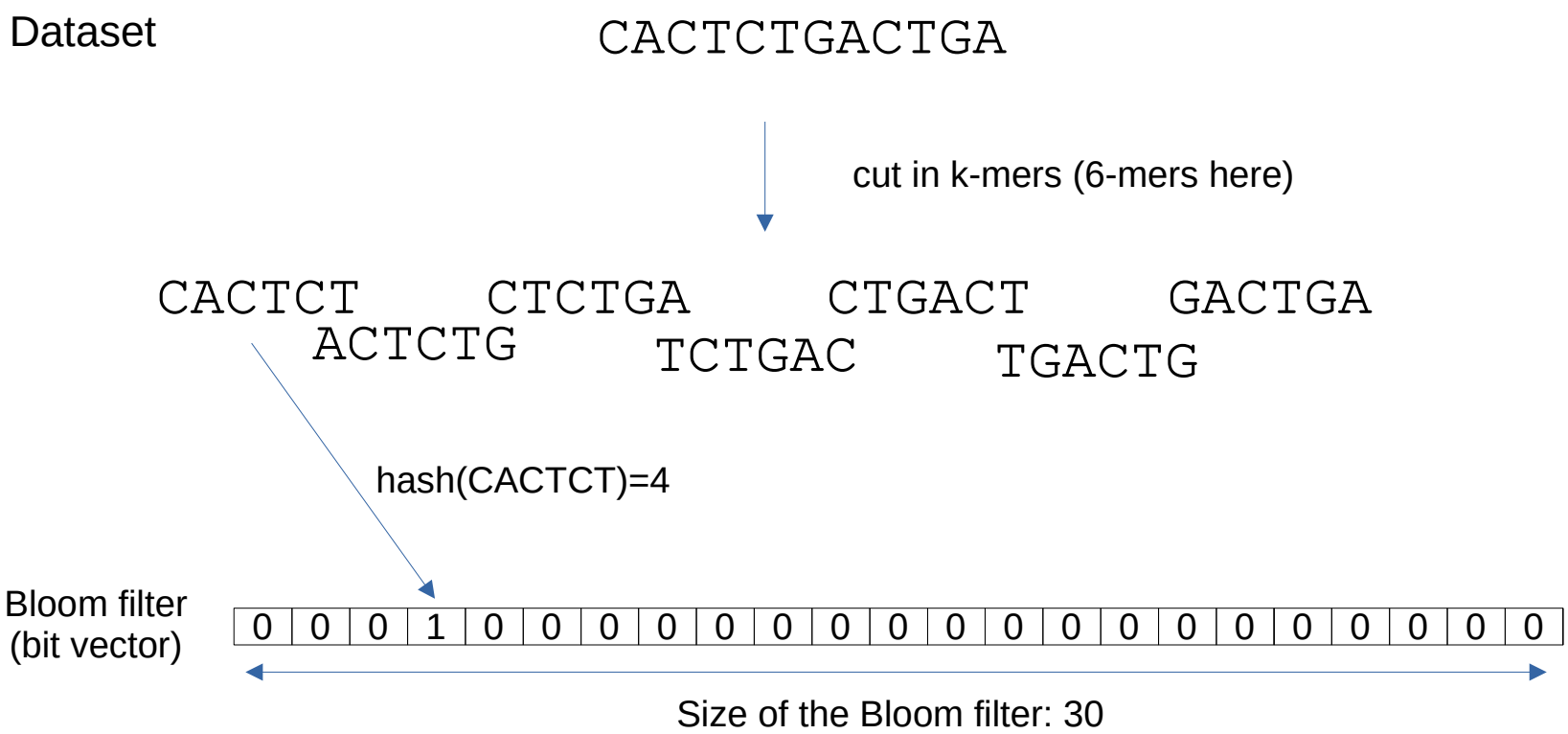
- Reminder: **1PB** of data, 27 million datasets
- Query: sequence
- Answer:
  - Difficulty level 1: datasets containing similar sequences
  - Difficulty level 2: the actual similar sequences

Téo Lemane

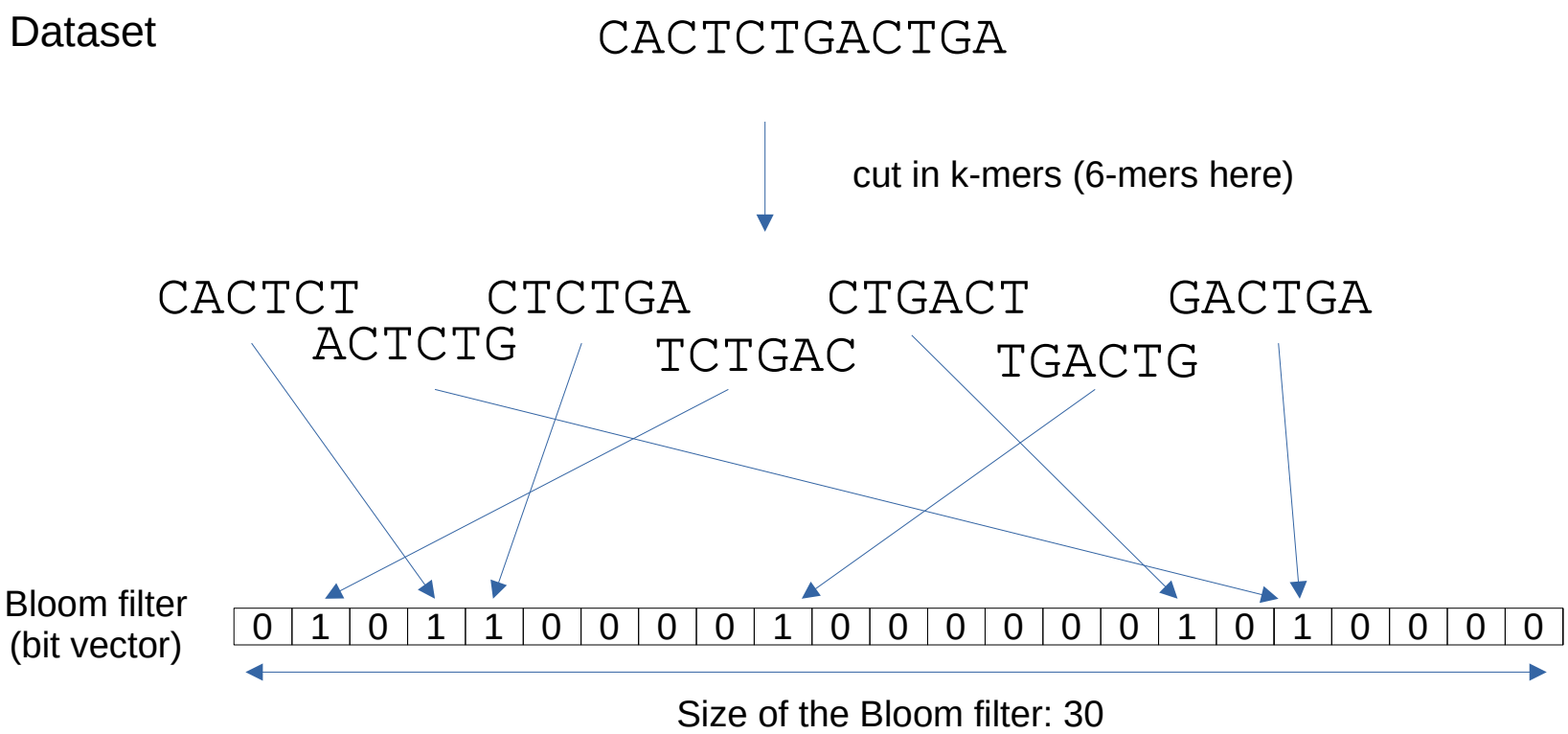
Indexing k-mers efficiently: bloom filters



Indexing k-mers efficiently: bloom filters



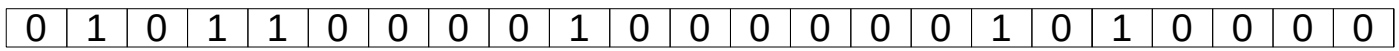
Indexing k-mers efficiently: bloom filters



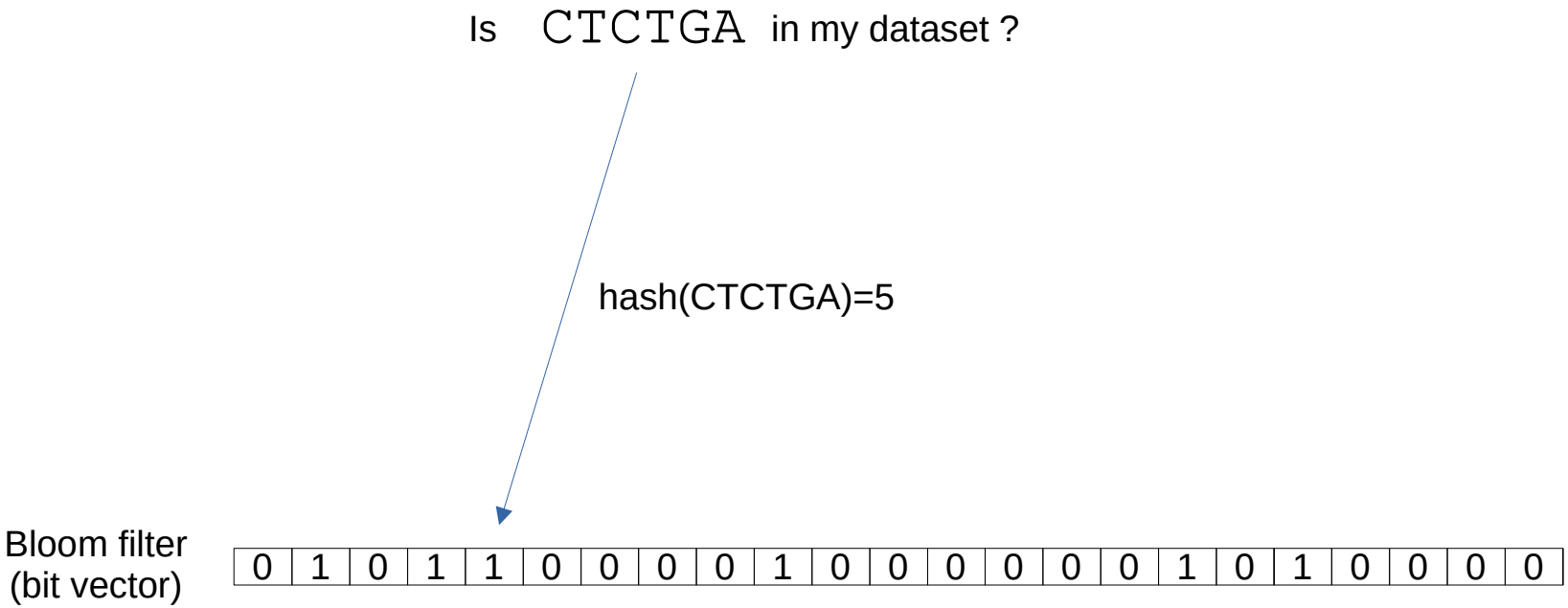
Indexing k-mers efficiently: bloom filters

Is `CTCTGA` in my dataset ?

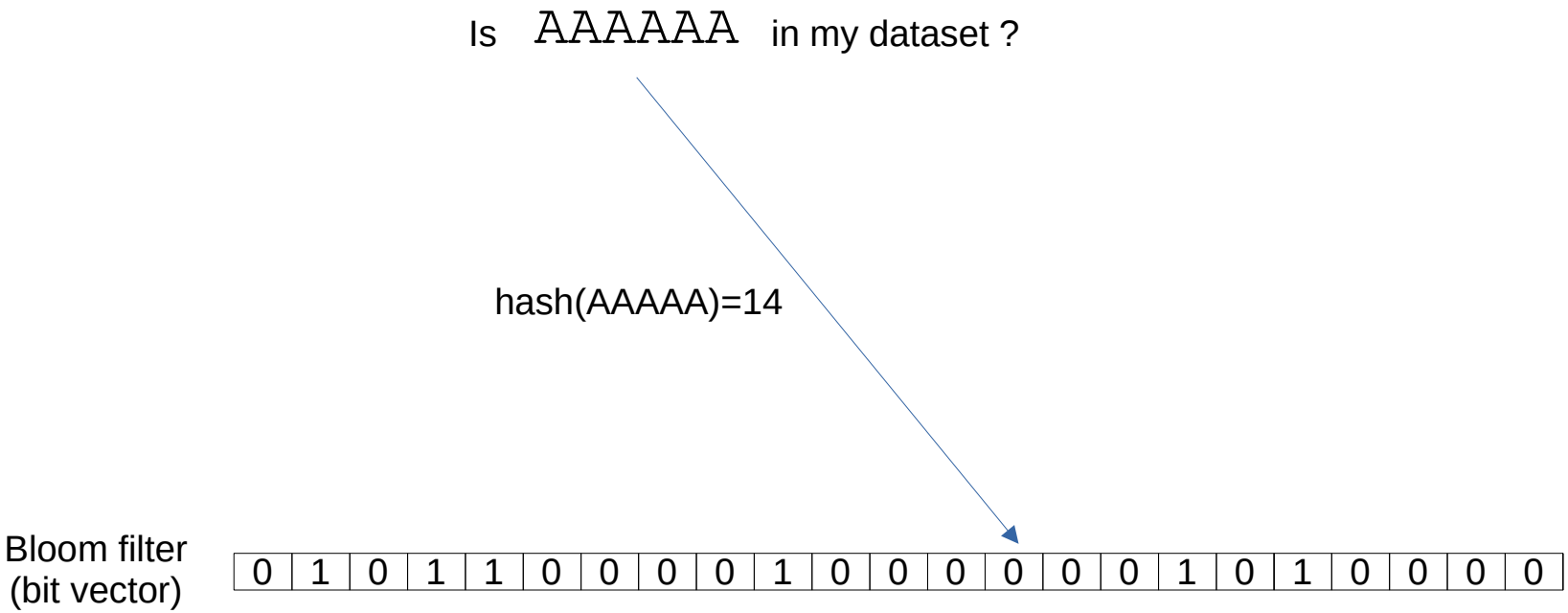
Bloom filter
(bit vector)



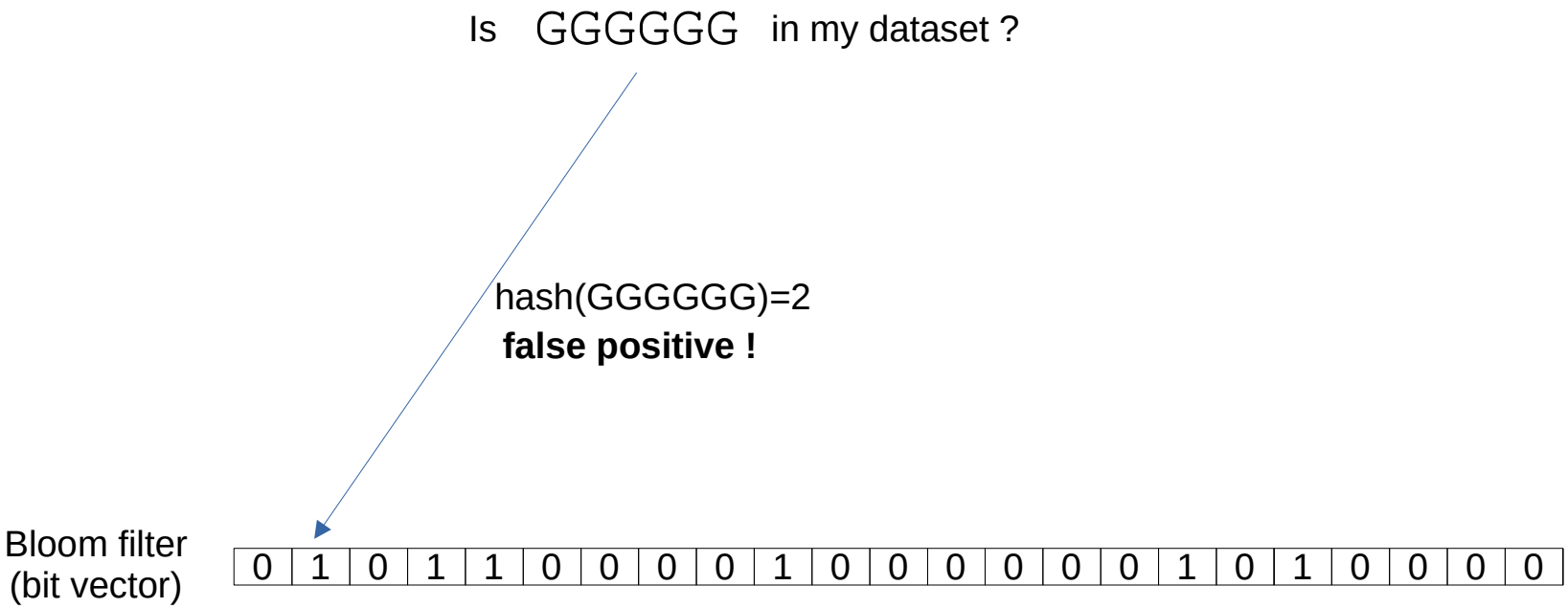
Indexing k-mers efficiently: bloom filters



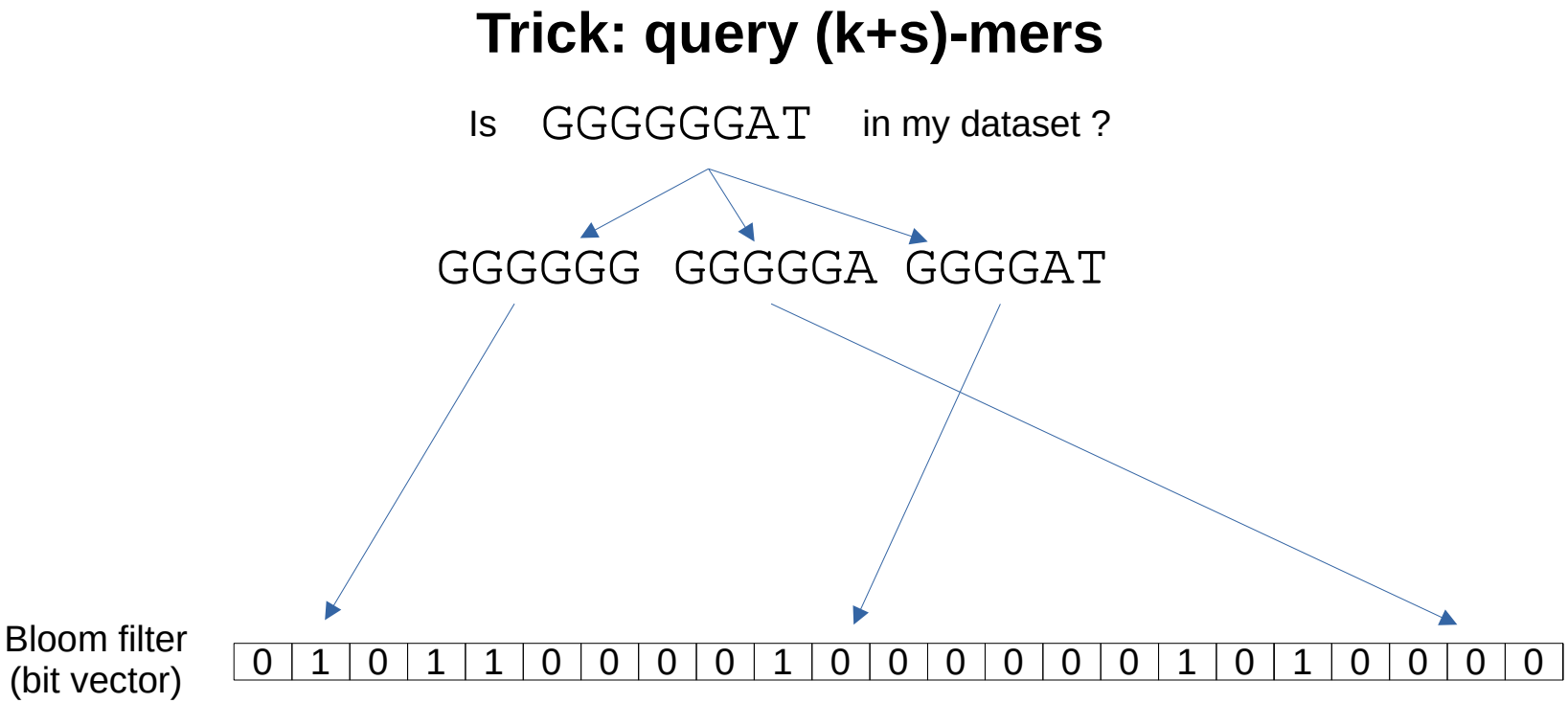
Indexing k-mers efficiently: bloom filters



Indexing k-mers efficiently: bloom filters



Indexing k-mers efficiently: bloom filters



Indexing strategy

- Index 26-mers of all datasets in Bloom filters
- At query time, query 31-mers

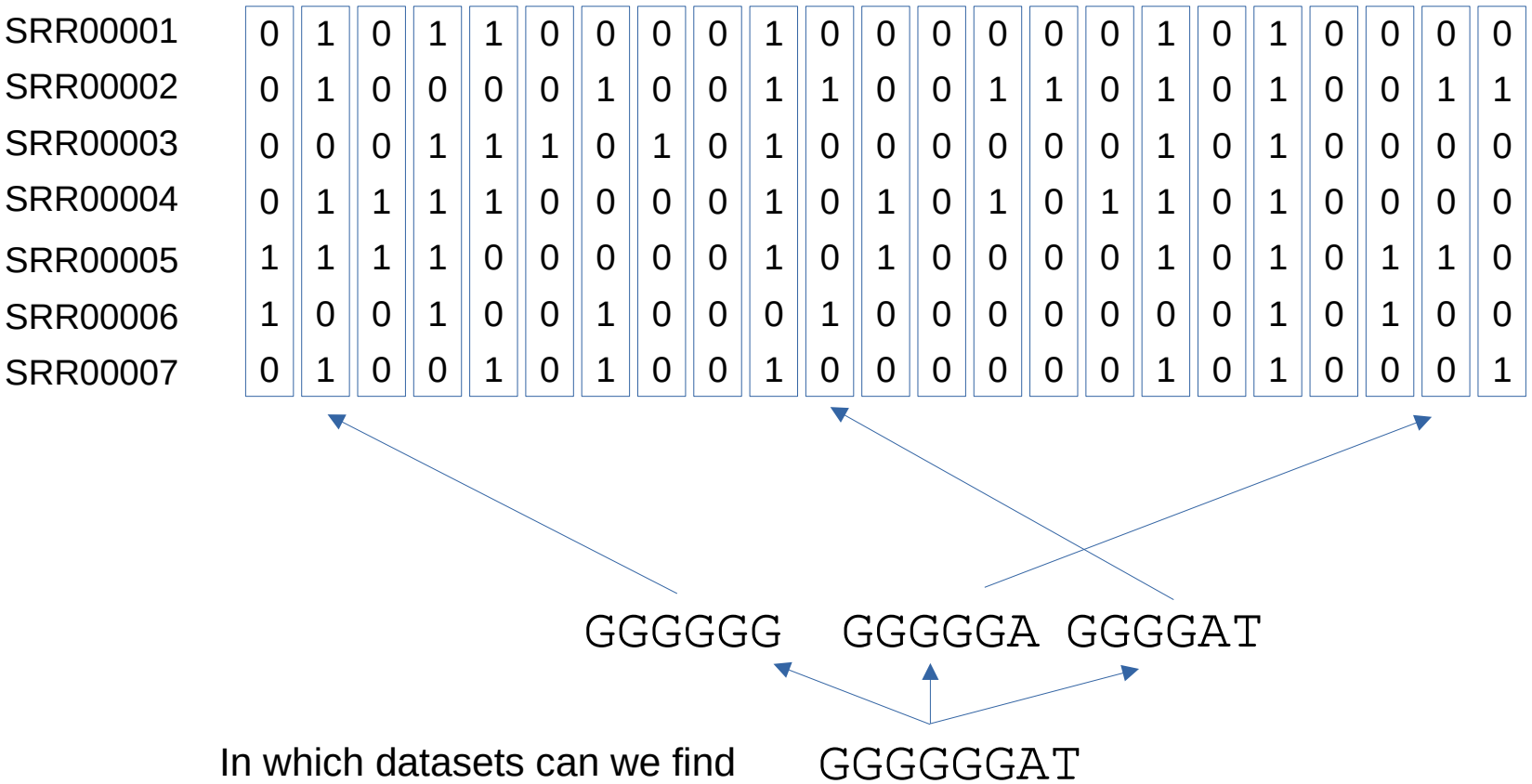
SRR00001	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
SRR00002	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	0	1	1
SRR00003	0	0	0	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
SRR00004	0	1	1	1	1	0	0	0	0	1	0	1	0	1	0	1	1	0	1	0	0	0	0
SRR00005	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0
SRR00006	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
SRR00007	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1
	.										.												
	.										.												
	.										.												

- In total, 1PB

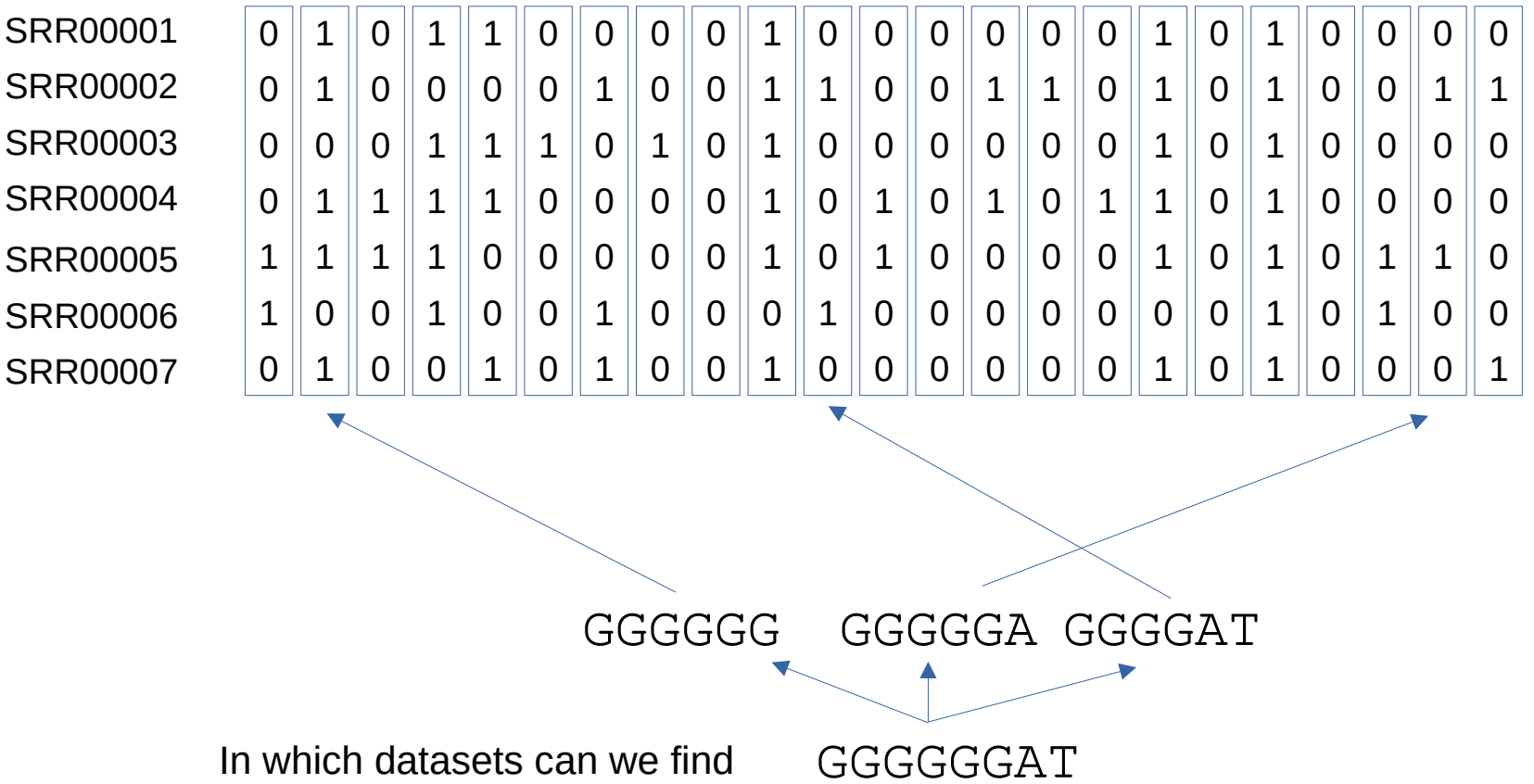
kmindex

SRR00001	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
SRR00002	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	0	1	1
SRR00003	0	0	0	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
SRR00004	0	1	1	1	1	0	0	0	0	1	0	1	0	1	0	1	1	0	1	0	0	0	0
SRR00005	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0
SRR00006	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
SRR00007	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	1

kmindex


SRR00001 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0
SRR00002 0 1 0 0 0 0 1 0 0 1 1 0 0 1 1 0 1 0 1 0 0 1 1
SRR00003 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0
SRR00004 0 1 1 1 1 0 0 0 0 1 0 1 0 1 0 1 1 0 1 0 0 0 0
SRR00005 1 1 1 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 1 0 1 1 0
SRR00006 1 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0
SRR00007 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 1
GGGGGG
GGGGGA
GGGGAT
In which datasets can we find
GGGGGGAT

kmindex



- Index of 1PB

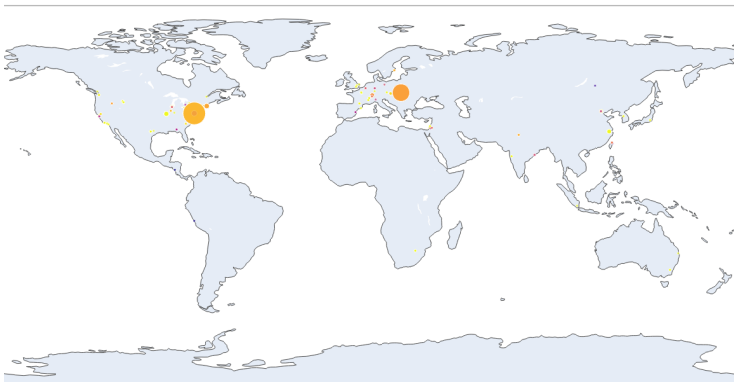
Logan-search.org

- "Find all datasets that share x% of 31-mers with my query"

Table | Map | Plot | BLAST-like alignment | Help

kmer_coverage > 0.7 AND assay_type IN ('WGS','WGA')

ID	kmer_coverage	bioproject	biosample
SRR16173100 (SRA|OV)	1	PRJNA768258 (SRA|OV)	SAMN22020249 (SRA|OV)
SRR14416387 (SRA|OV)	1	PRJNA727098 (SRA|OV)	SAMN19006852 (SRA|OV)
SRR15166888 (SRA|OV)	1	PRJNA746342 (SRA|OV)	SAMN20203734 (SRA|OV)
ERR3505101 (SRA|OV)	1	PRJEB27179 (SRA|OV)	SAMEA5930711 (SRA|OV)
SRR3740047 (SRA|OV)	1	PRJNA327431 (SRA|OV)	SAMN05335350 (SRA|OV)
ERR2398673 (SRA|OV)	1	PRJEB22684 (SRA|OV)	SAMEA104683220 (SRA|OV)
SRR14655031 (SRA|OV)	1	PRJNA732827 (SRA|OV)	SAMN13221434 (SRA|OV)
SRR13306852 (SRA|OV)	1	PRJNA687219 (SRA|OV)	SAMN17141292 (SRA|OV)
SRR14416320 (SRA|OV)	1	PRJNA727098 (SRA|OV)	SAMN19006784 (SRA|OV)
ERR2399820 (SRA|OV)	1	PRJEB22684 (SRA|OV)	SAMEA104685216 (SRA|OV)
SRR13011679 (SRA|OV)	1	PRJNA668406 (SRA|OV)	SAMN16710006 (SRA|OV)
ERR2395390 (SRA|OV)	1	PRJEB22684 (SRA|OV)	SAMEA104681707 (SRA|OV)
ERR2394036 (SRA|OV)	1	PRJEB22684 (SRA|OV)	SAMEA104680353 (SRA|OV)
SRR25617432 (SRA|OV)	1	PRJNA1004049 (SRA|OV)	SAMN36920974 (SRA|OV)
SRR25659779 (SRA|OV)	1	PRJNA1006165 (SRA|OV)	SAMN37013649 (SRA|OV)
SRR8369137 (SRA|OV)	1	PRJNA511728 (SRA|OV)	SAMN03178087 (SRA|OV)
SRR11318504 (SRA|OV)	1	PRJNA612868 (SRA|OV)	SAMN14389256 (SRA|OV)
SRR25507401 (SRA|OV)	1	PRJNA1001958 (SRA|OV)	SAMN36828017 (SRA|OV)



How much does it cost to query Logan-search?

A: $1

B: $10

C: $100

D: $1,000

How much does it cost to query Logan-search?

A: $1

B: $10

C: $100

D: $1,000

Indexing sequences: limits & future

- Limit: slow to get the actual sequences
- Limit: query and target need to share 31-mers

Indexing sequences: limits & future

- Limit: slow to get the actual sequences
- Limit: query and target need to share 31-mers

- Soon another strategy for indexing nucleotides

Indexing sequences: limits & future

- Limit: slow to get the actual sequences
- Limit: query and target need to share 31-mers

- Soon another strategy for indexing nucleotides
- Fundamental sensitivity / speed tradeoff in 1PB of data

Indexing proteins

Obtaining all the proteins of SRA

- Ran prodigal on all assemblies: 100 billion proteins
- Clustered with MMseqs2 at 50% identity: 3 billion representative proteins

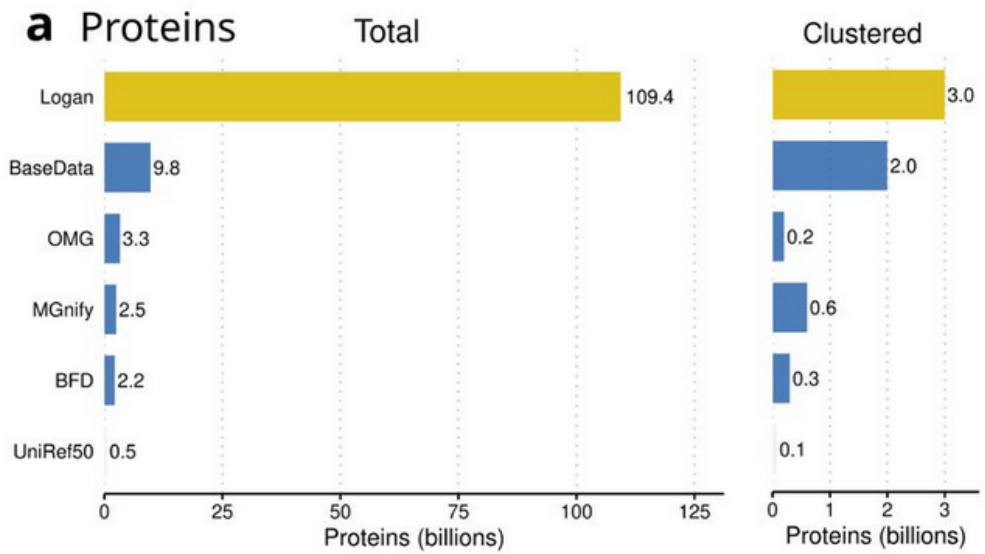


Figure from the Logan preprint

Obtaining all the proteins of SRA

- Ran prodigal on all assemblies: 100 billion proteins
- Clustered with MMseqs2 at 50% identity: 3 billion representative proteins

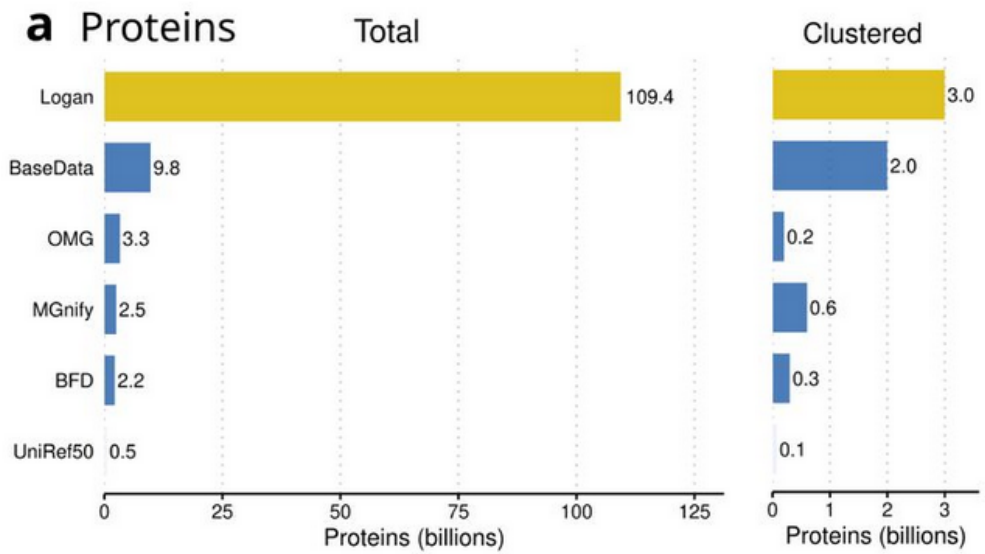


Figure from the Logan preprint

- Query: a protein ; Answer: all similar proteins in Logan

How to compare (3 billion) proteins?

Strategy 1: sequence comparisons

How to compare (3 billion) proteins?

Strategy 1: sequence comparisons

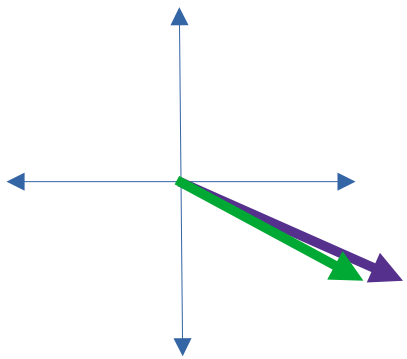
```
MRIFGFFITLVAIIGQ
    |||||||||||
MRIKGFFITLIAIIFQ
```

Strategy 2: embedding comparisons

MRIKGFFITLIAIIFQ

MRIFGFFITLVAIIGQ

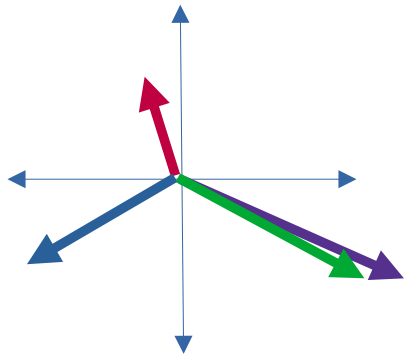
Protein Language Model embedding



Indexing 3 billion proteins

MRIKGFFITLIAIIFQ
MRIFGFFITLVAIIGQ
MSIYHMKVRTITGKDMTLQP
MTFFLYISPMISILIGFK
.....

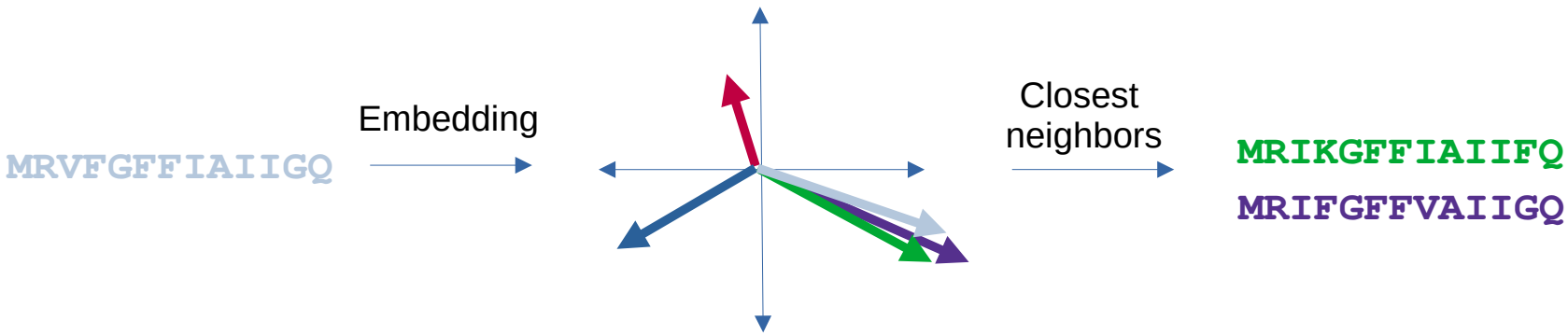
Embedding



Vector database

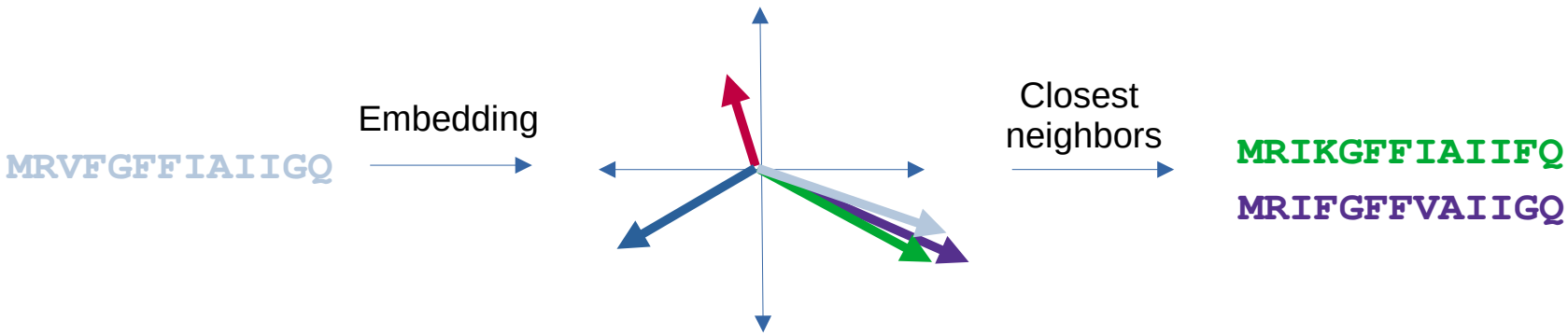
- Protein Language Model: gLM2
- 512-dimension vectors
- 3k GPU.hours
- Space taken by final database: 1.5TB

Querying 3 billion proteins



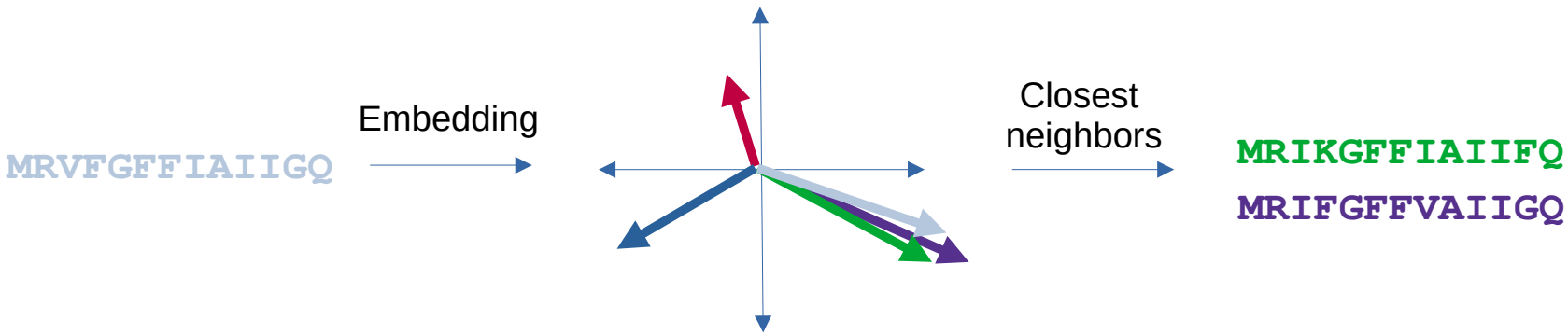
- Embed query & compare to existing vectors

Querying 3 billion proteins



- Embed query & compare to existing vectors
- Compare to 3 billion vectors using industrial vector databases

Querying 3 billion proteins



- Embed query & compare to existing vectors
- Compare to 3 billion vectors using industrial vector databases
- Actually just bruteforce comparison

Protein search: performance

- Query: known PETase enzymes

Protein search: performance

- Query: known PETase enzymes
- Throughput: $\sim$ 0.1 query/second, CPU, parallelizable
- 150Mb of RAM
- 3M results (but buggy, do not trust this yet)

Protein search: performance

- Query: known PETase enzymes
- Throughput: $\sim$ 0.1 query/second, CPU, parallelizable
- 150Mb of RAM
- 3M results (but buggy, do not trust this yet)
- (Possible but expensive to go back to sequences)

Limits

- Only works on human-dataset for now, but everything else available soon

Limits

- Only works on human-dataset for now, but everything else available soon
- Only full proteins match

Limits

- Only works on human-dataset for now, but everything else available soon
- Only full proteins match
- 90% proteins missing in the database because of the protein calling

Applications

- Functional insights

Applications

- Functional insights
- Phylogeny

Applications

- Functional insights
- Phylogeny
- Ecology: geographical metadata assiociated

Applications

- Functional insights
- Phylogeny
- Ecology: geographical metadata assiociated
- ...