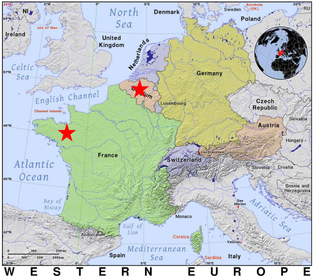
About me: Ph.D., 2021-2024



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

Dominique Lavenier
Université de Rennes
Focus: computational methods

Metagenome assembly from long reads

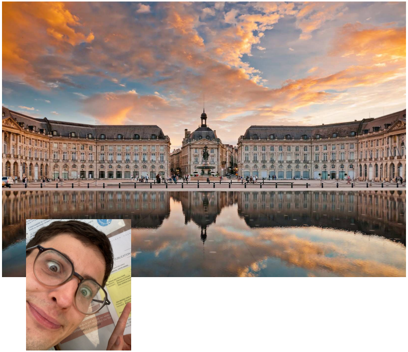
About me: postdoc, since 2025

Institut Pasteur, Paris

Rayan Chikhi
Institut Pasteur, Paris
Focus: massive genomics

Index & Search the Logan database

About me: future in Bordeaux??



Inria??

cnrs??

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

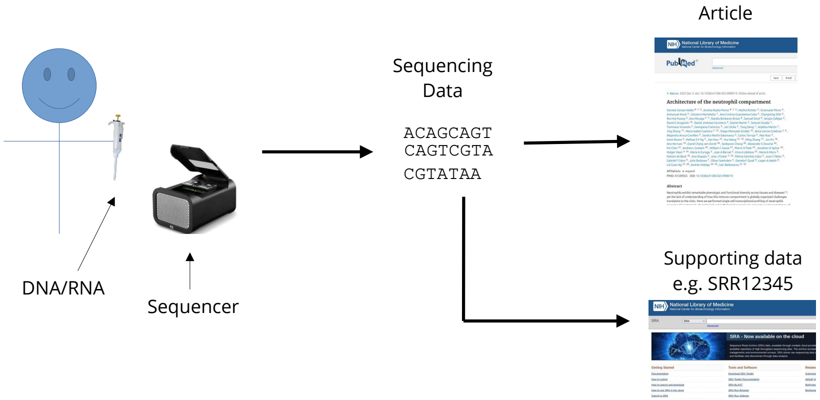
Roland Faure[1]

[1]Institut Pasteur

January 2026

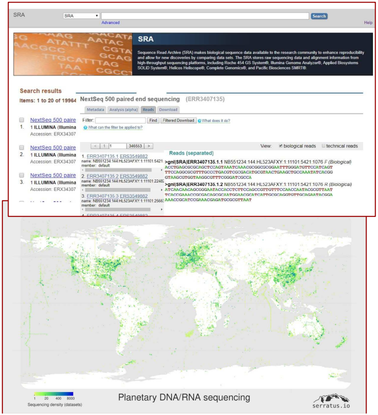
The Logan database

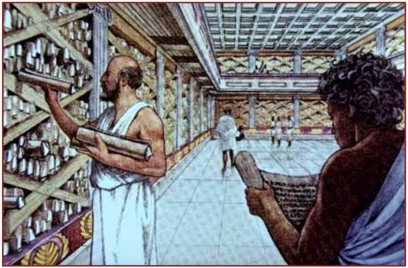
J. Doe sequenced something



The SRA database

SRA: All public sequencing reads, 80 PB of data

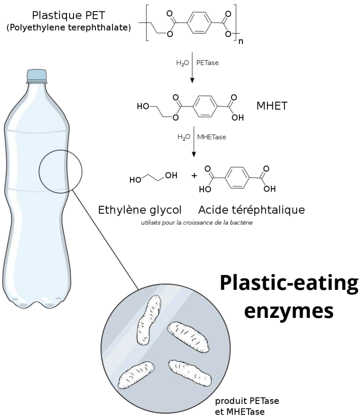




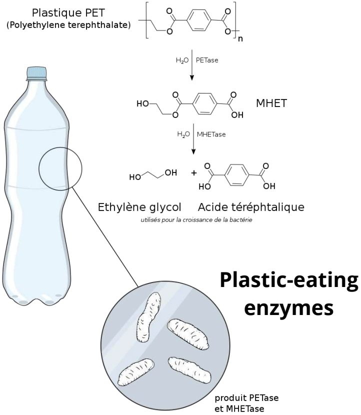
“Library of Alexandria” for genetics

Slide Credits: Rayan Chikhi

Plastic-eating enzymes: PETases



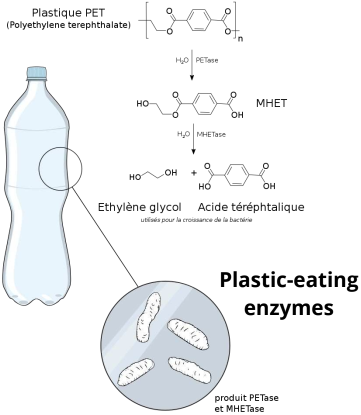
Plastic-eating enzymes: PETases



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

Artem Babaian

Plastic-eating enzymes: PETases

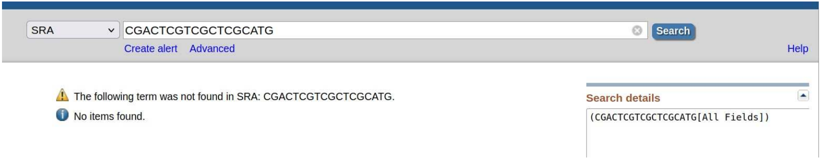


We know of 200 such enzymes but there is much we ignore, and they are hard to find

There must be more of them in the SRA!

Artem Babaian

The SRA is not queryable



The SRA now

Slide Credits: Teo Lemane

- 27 millions accessions

Quiz: At 1 Gbit/s, how much time to download the SRA?

A: 20 days	B: 20 weeks
C: 20 months	D: 20 years

Quiz: At 1 Gbit/s, how much time to download the SRA?

A: 20 days

B: 20 weeks

C: 20 months

D: 20 years

The Logan database

SRA

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

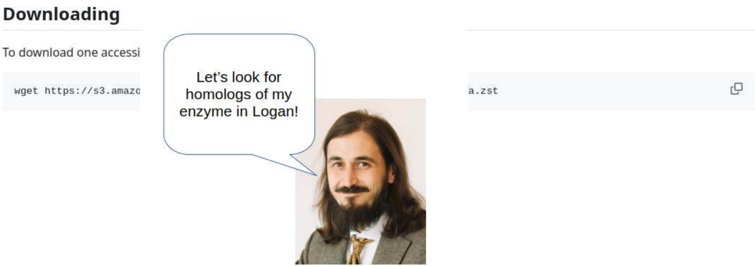
Downloading

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



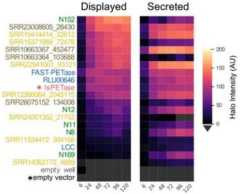
Artem Babaian

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

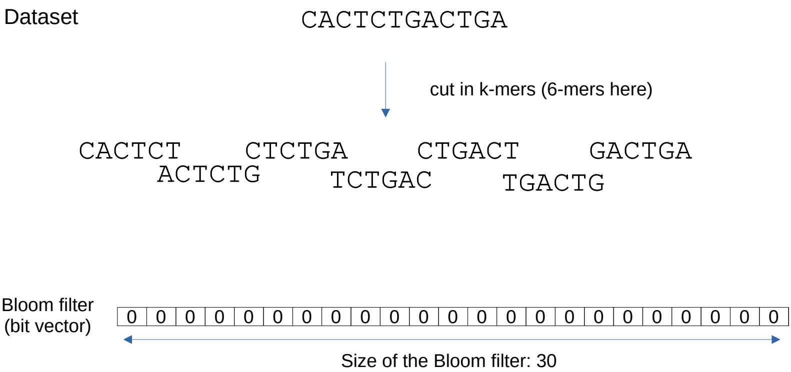
Indexing nucleotides

Let's index nucleotides

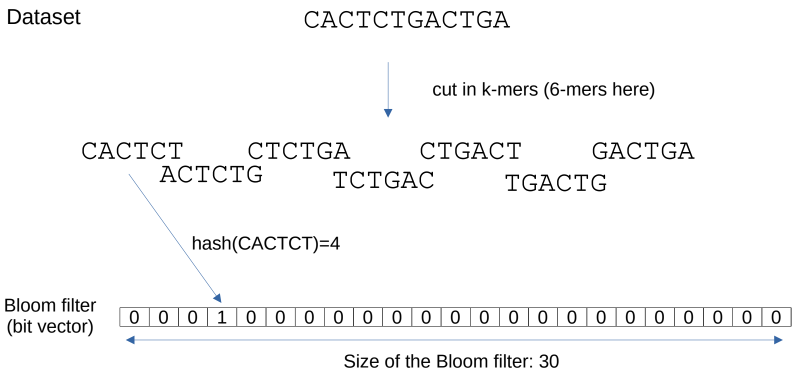
- Reminder: **1PB** of data, 27 million datasets
- Query: sequence
- Answer:
  - Difficulty level 1: datasets containing similar sequences
  - Difficulty level 2: the actual similar sequences

Téo Lemane

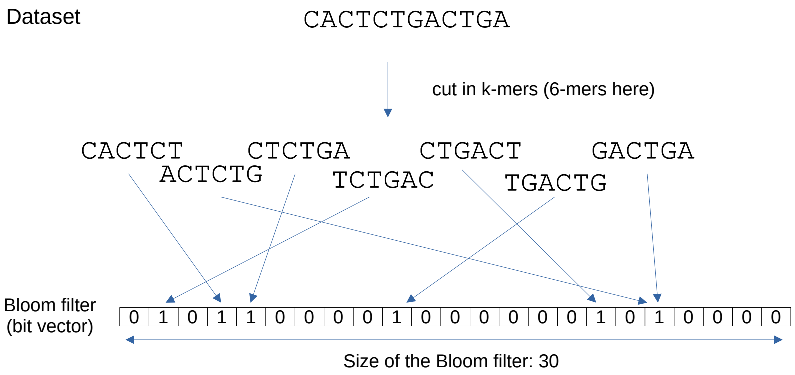
Indexing k-mers efficiently: bloom filters



Indexing k-mers efficiently: bloom filters



Indexing k-mers efficiently: bloom filters



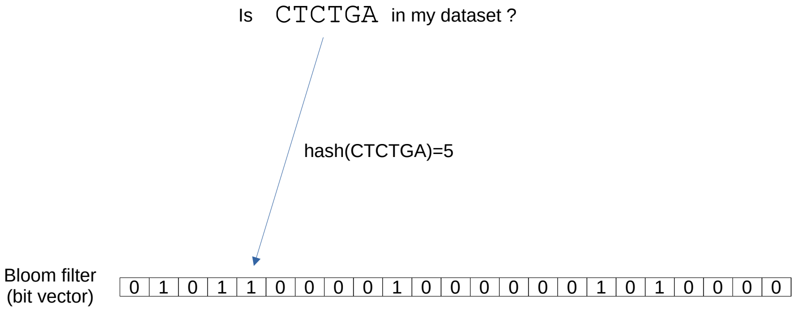
Indexing k-mers efficiently: bloom filters

Is `CTCTGA` in my dataset ?

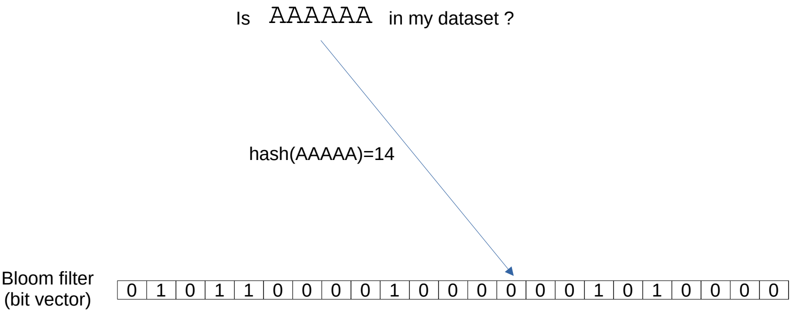
Bloom filter (bit vector)

0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0

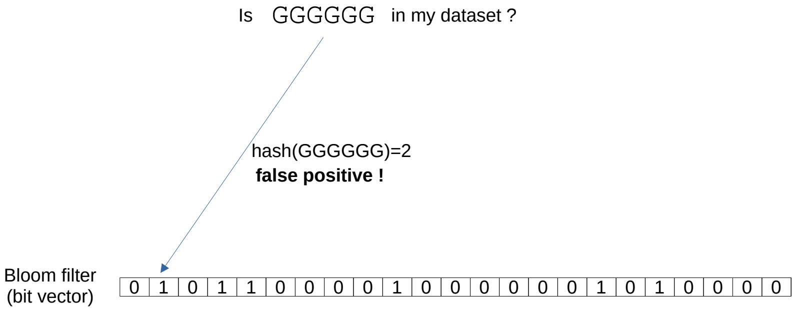
Indexing k-mers efficiently: bloom filters



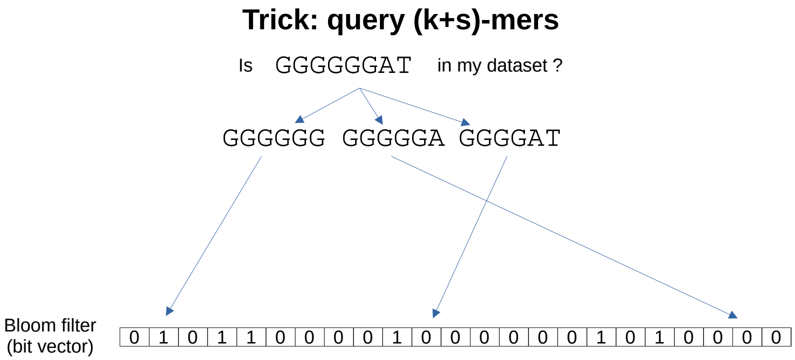
Indexing k-mers efficiently: bloom filters



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

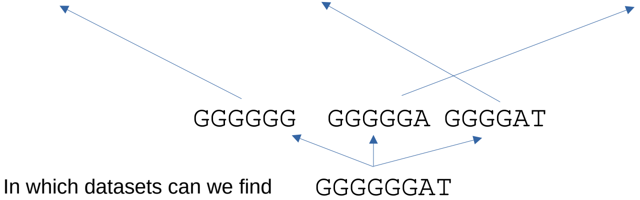
SRR00001	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
SRR00002	0	1	0	0	0	0	1	0	0	1	1	0	0	1	1	0	1	0	1	0	0	1	1
SRR00003	0	0	0	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
SRR00004	0	1	1	1	1	0	0	0	0	1	0	1	0	1	0	1	1	0	1	0	0	0	0
SRR00005	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	1	0
SRR00006	1	0	0	1	0	0	1	0	0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

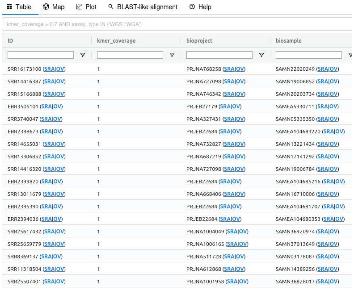
SRR00001	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
SRR00002	0	1	0	0	0	0	1	0	0	1	1	0	0	1	1	0	1	0	1	0	0	1	1
SRR00003	0	0	0	1	1	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0
SRR00004	0	1	1	1	1	0	0	0	0	1	0	1	0	1	0	1	1	0	1	0	0	0	0
SRR00005	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	1	0
SRR00006	1	0	0	1	0	0	1	0	0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



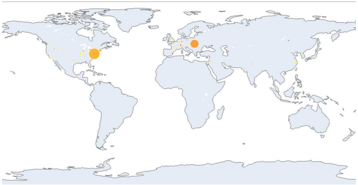
- Index of 1PB

Logan-search.org

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



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

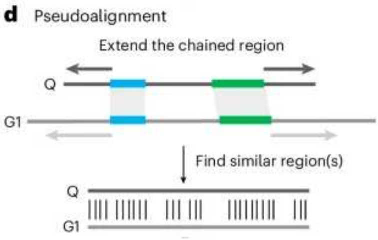
C: $100

D: $1,000

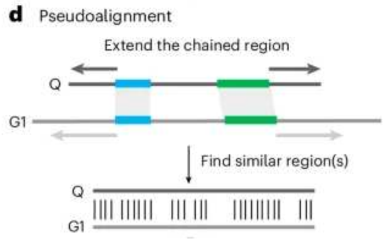
Indexing sequences: limits & future

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

Another strategy to index sequences: LexicMap

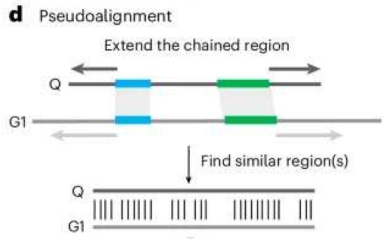


Another strategy to index sequences: LexicMap



- Available on Galaxy soon

Another strategy to index sequences: LexicMap



- Available on Galaxy soon

- More sensitive but slow (several hours)

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

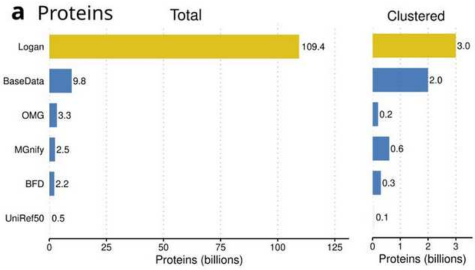
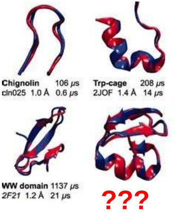


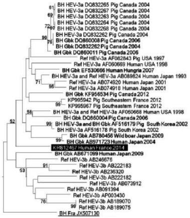
Figure from the Logan preprint

What can we do with these proteins?

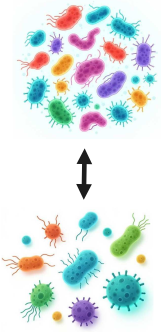
Discover new proteins



Improve protein phylogenies



Compare datasets



My work: looking for proteins by similarity

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

Artem Babaian

How to compare (3 billion) proteins?

Strategy 1: sequence comparisons

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

How to compare (3 billion) proteins?

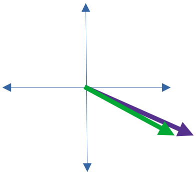
Strategy 1: sequence comparisons

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

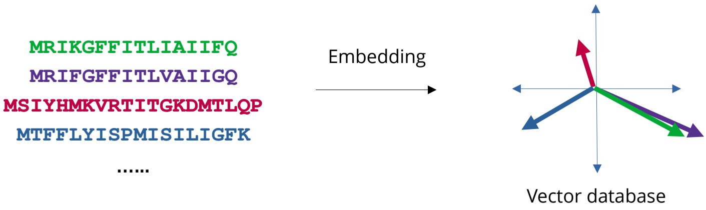
Strategy 2: embedding comparisons

MRIKGFFITLIAIIFQ
MRIFGFFITLVAIIGQ

↓ Protein Language Model embedding

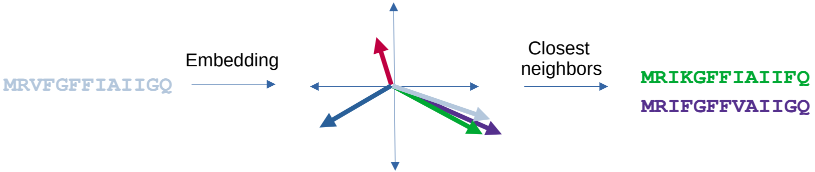


Indexing 3 billion proteins



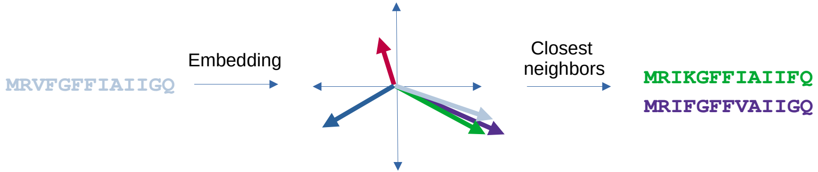
- Protein Language Model: gLM2
- 512-dimension vectors
- 3k GPU.hours
- Space taken by final database: 1.5TB

Querying 3 billion proteins



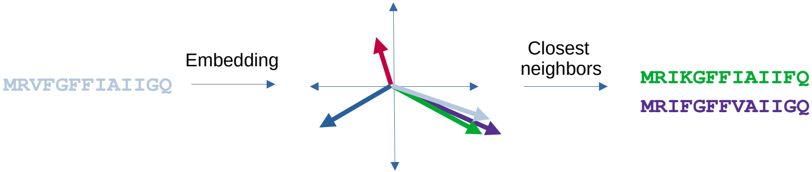
- Embed query & compare to existing vectors

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: 301 known papillomavirus proteins

Protein search: performance

- Query: 301 known papillomavirus proteins
- 4h computation, 120 GB RAM ($\sim$ $10)

Protein search: performance

- Query: 301 known papillomavirus proteins
- 4h computation, 120 GB RAM ($\sim$ \$10)
- 50k homologs in Logan

Protein search: performance

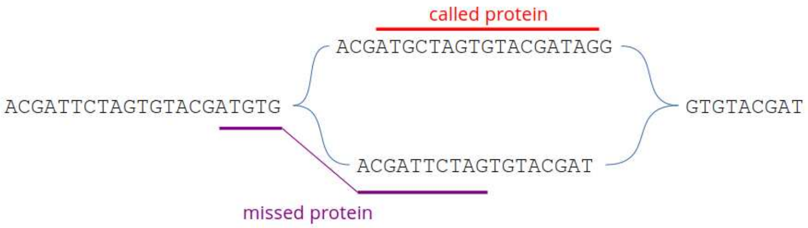
- Query: 301 known papillomavirus proteins
- 4h computation, 120 GB RAM ($\sim$ \$10)
- 50k homologs in Logan
- Soon available online on Galaxy and downloadable ($\sim$ 5TB)
- Contact me if you are interested now

Limits

- Only full proteins match

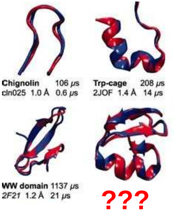
Limits

- Only full proteins match
- 90% proteins missing in the database because of the protein calling

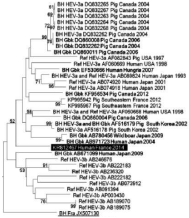


Future developments of Logan

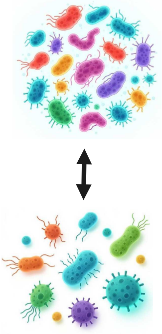
Discover new proteins



Improve protein phylogenies



Compare datasets



Take-home message

- Logan centralizes all public datasets
- Nucleotide and proteins

Take-home message

- Logan centralizes all public datasets
- Nucleotide and proteins

- Contact us if you need help using Logan tools
- Contact us if you want some features
- You can propose features too