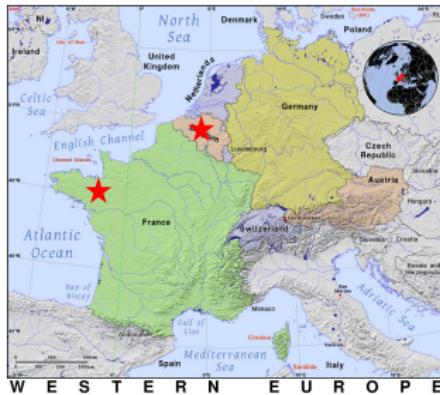


About me: Ph.D., 2021-2024



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

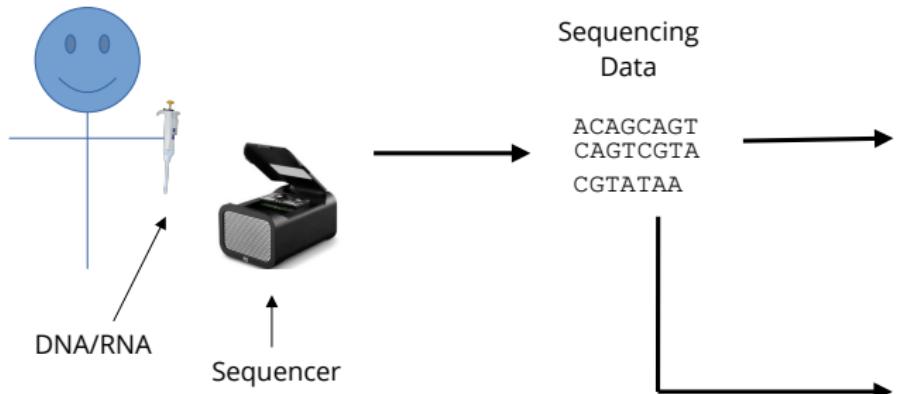
Roland Faure¹

¹Institut Pasteur

December 2025

The Logan database

J. Doe sequenced something



The SRA database

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

The figure shows a screenshot of the SRA (Sequence Read Archive) search interface. At the top, there is a search bar with "SRA" selected, a dropdown menu for "Advanced", a "Search" button, and a "Help" link. Below the search bar is a banner with the text "SRA" and a brief description: "Sequence Read Archive (SRA) makes biological sequence data available to the research community to enhance reproducibility and allow for new discoveries by comparing data sets. The SRA stores raw sequencing data and alignment information from high-throughput sequencing platforms, including Roche 454 GS System®, Illumina Genome Analyzer®, Applied Biosystems SOLiD System®, Helicos Helicosell®, Complete Genomics®, and Pacific Biosciences SMRT®".

Search results

Items: 1 to 20 of 19984 NextSeq 500 paired end sequencing (ERR3407135)

Metadata Analysis (alpha) Reads Download

Filter: [] Find: [] Filtered Download: [] What does it do?

NextSeq 500 paired

1. 1 ILLUMINA (Illumina)
Accession: ERX34307

NextSeq 500 paired

2. 1 ILLUMINA (Illumina)
Accession: ERX34307

NextSeq 500 paired

3. 1 ILLUMINA (Illumina)
Accession: ERX34307

NextSeq 500 paired

4. 1 ILLUMINA (Illumina)
Accession: ERX34307

NextSeq 500 paired

5. 1 ILLUMINA (Illumina)
Accession: ERX34307

Reads (separated)

View: bold

ERR3407135_1.EBS2549882
name: NB551234_144HL523AFXY_1.1101:21192
member: default

>gnl|SRA|ERR3407135_1.1 NB551234_144HL523AFXY_1.1101:21192
ACCTGAGCGGCAGTCGCGTAAATCAACCGCGGCGCGCGAATTGGCG
TTCCAGGGCGCTTGTGGCGTGGCGTGGCGTGGCGTGGCGTGGCGT
GTAAGCGCGTGGCGTGGCGTGGCGTGGCGTGGCGTGGCGTGGCGT
GTAAGCGCGTGGCGTGGCGTGGCGTGGCGTGGCGTGGCGTGGCGT
>gnl|SRA|ERR3407135_2 NB551234_144HL523AFXY_1.1101:21192
name: NB551234_144HL523AFXY_1.1101:21192
member: default

>gnl|SRA|ERR3407135_3 NB551234_144HL523AFXY_1.1101:2586
name: NB551234_144HL523AFXY_1.1101:2586
member: default

>gnl|SRA|ERR3407135_4 NB551234_144HL523AFXY_1.1101:21192
name: NB551234_144HL523AFXY_1.1101:21192
member: default

>gnl|SRA|ERR3407135_5 NB551234_144HL523AFXY_1.1101:21192
name: NB551234_144HL523AFXY_1.1101:21192
member: default

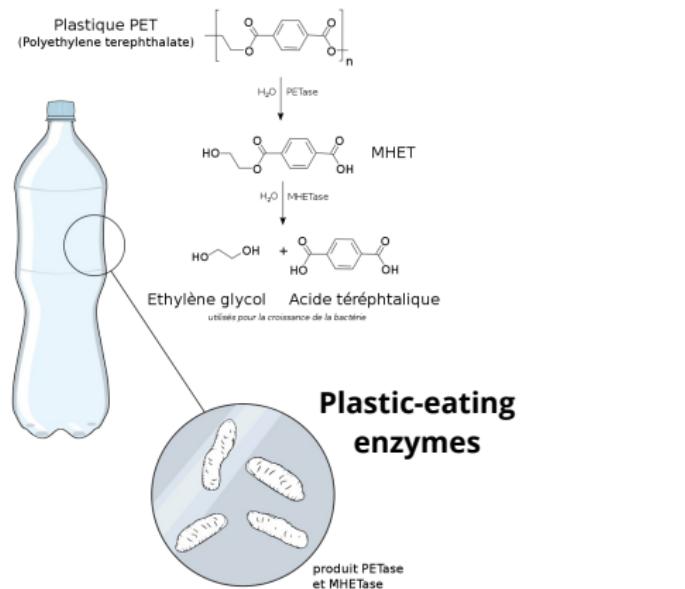
Planetary DNA/RNA sequencing

Sequencing density (reads)

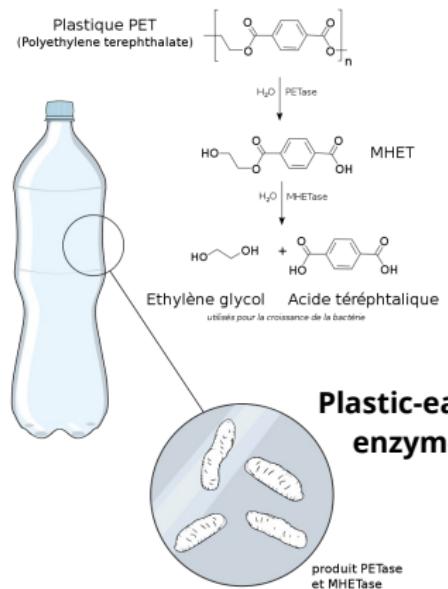
A world map showing sequencing density across the globe, with a color scale from blue (low density) to red (high density).

Slide Credits: Teo Lemane

Plastic-eating enzymes: PETases



Plastic-eating enzymes: PETases

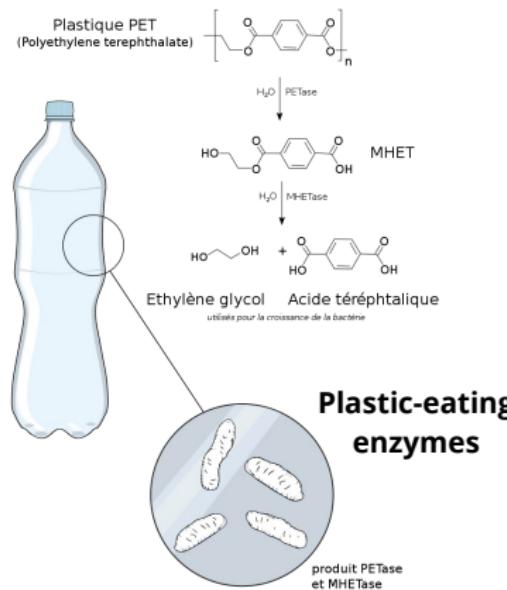


We know of 200 such enzymes but there is much we ignore, and they are hard to find



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

SRA

CGACTCGTCGCTCGCATG

Create alert Advanced

Search

Help

⚠ The following term was not found in SRA: CGACTCGTCGCTCGCATG.

ℹ No items found.

Search details

(CGACTCGTCGCTCGCATG[All Fields])

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)

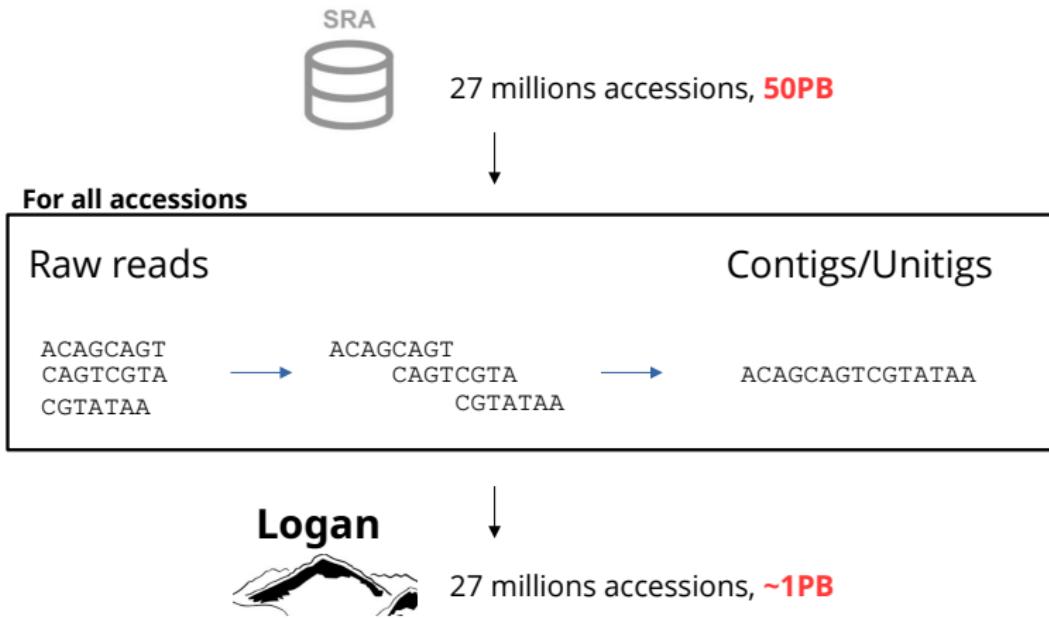
The SRA now



Slide Credits: Teo Lemane

- ▶ 27 millions accessions

The Logan database



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
```

Let's look for
homologs of my
enzyme in Logan!



Artem Babaian

a.zst

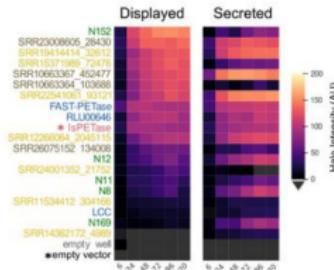


Searching through Logan

- ▶ 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

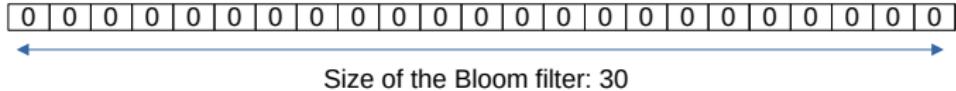
Dataset

CACTCTGACTGA

cut in k-mers (6-mers here)

CACTCT CTCTGA CTGACT GACTGA
ACTCTG TCTGAC TGACTG

Bloom filter
(bit vector)



Indexing k-mers efficiently: bloom filters

Dataset

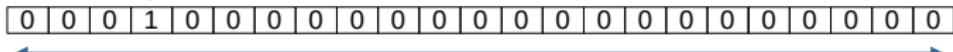
CACTCTGACTGA

↓
cut in k-mers (6-mers here)

CACTCT CTCTGA CTGACT GACTGA
 ACTCTG TCTGAC TGACTG

hash(CACTCT)=4

Bloom filter
(bit vector)



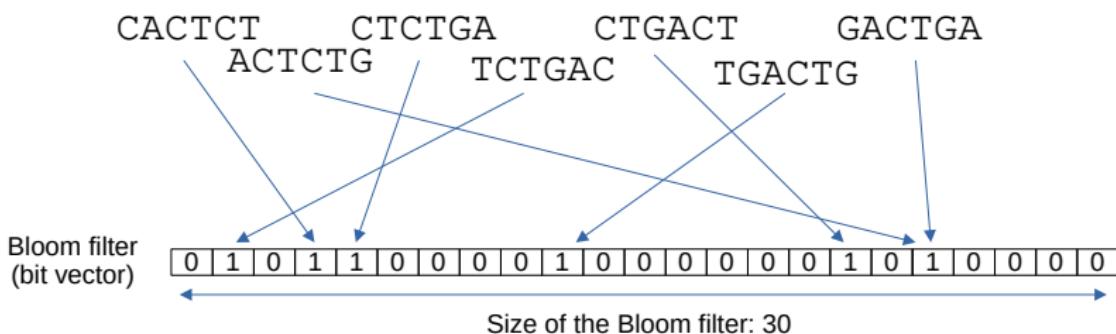
Size of the Bloom filter: 30

Indexing k-mers efficiently: bloom filters

Dataset

CACTCTGACTGA

cut in k-mers (6-mers here)



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

Is CTCTGA in my dataset ?

hash(CTCTGA)=5

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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Indexing k-mers efficiently: bloom filters

Is AAAAAA in my dataset ?

hash(AAAAA)=14

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
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Indexing k-mers efficiently: bloom filters

Is GGGGGG in my dataset ?

hash(GGGGGG)=2
false positive !

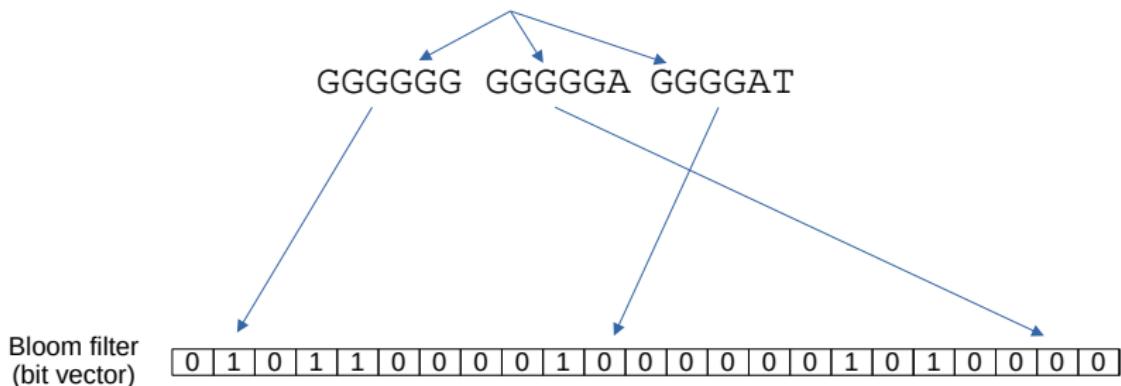
Bloom filter
(bit vector)

0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Indexing k-mers efficiently: bloom filters

Trick: query (k+s)-mers

Is GGGGGGAT in my dataset ?



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 0 1 0 1 0 0 0 0
SRR00002	0 0 0 0 0 0 1 0 0 1 0 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 0 1 0 1 0 0 0 0
SRR00004	0 1 1 1 1 0 0 0 0 0 1 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 0 1 0 1 0 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 1 0 0
SRR00007	0 1 0 0 1 0 1 0 0 1 0 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	0
SRR00002	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	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	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	0
SRR00005	1	1	1	1	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	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	1	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	0

kmindex

SRR00001
SRR00002
SRR00003
SRR00004
SRR00005
SRR00006
SRR00007

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

In which datasets can we find

GGGGGG

GGGGGA GGGGAT

GGGGGGAT

kmindex

SRR00001
SRR00002
SRR00003
SRR00004
SRR00005
SRR00006
SRR00007

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

GGGGGG
GGGGGA
GGGGAT
GGGGGGAT

In which datasets can we find

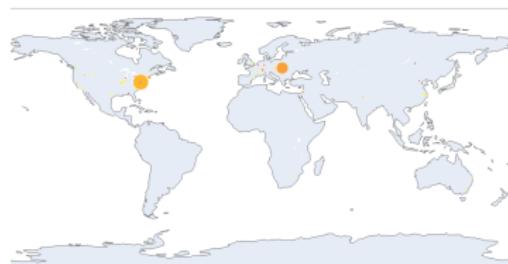
- ▶ Index of 1PB

► “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	0.9409	PRJNA768258	SAMN21020249
SRR14416307	0.9409	PRJNA727098	SAMN17606852
SRR15166688	0.9409	PRJNA746324	SAMN20307374
ER03505101	0.9409	PRJEB27179	SAMHEA1030711
SRR3740047	0.9409	PRJNA327431	SAMNG0333530
ER02399873	0.9409	PRJEB22849	SAMEA104883220
SRR14655031	0.9409	PRJNA732327	SAMN173221434
SRR13598852	0.9409	PRJNA687219	SAMN17141292
SRR14416320	0.9409	PRJNA727098	SAMN17606764
ER02399830	0.9409	PRJEB22849	SAMEA104883216
SRR13014679	0.9409	PRJNA66949	SAMN16170006
ER02395390	0.9409	PRJEB22849	SAMEA104881707
ER02394936	0.9409	PRJEB22849	SAMEA104881553
SRR25617432	0.9409	PRJNA100409	SAMNS6920974
SRR25659779	0.9409	PRJNA1006185	SAMNS37013649
SRR2607137	0.9409	PRJNA211728	SAMNG0317007
SRR11378509	0.9409	PRJNA6612988	SAMN17430256
SRR25507461	0.9409	PRJNA1001958	SAMNH366290117



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

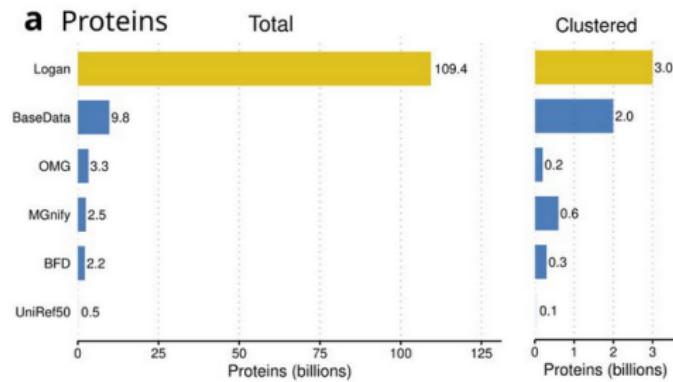


Figure from the Logan preprint

Obtaining all the proteins of SRA

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

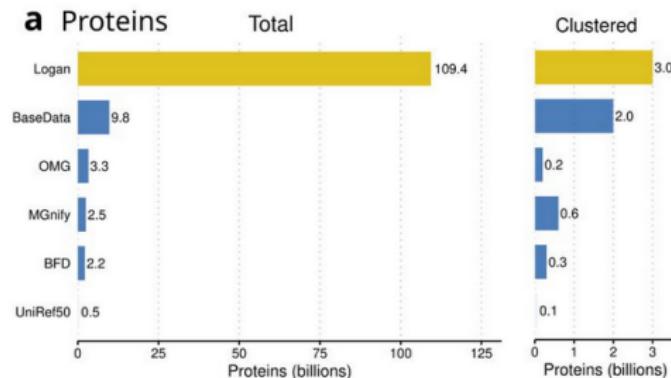


Figure from the Logan preprint

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

How to compare (3 billion) proteins?

Strategy 1: sequence
comparisons

MRIF**GFFITLVAAI**I GQ
|||||||
MRIK**GFFITLIAII**I FQ

How to compare (3 billion) proteins?

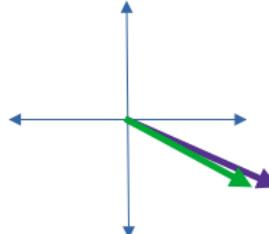
Strategy 1: sequence comparisons

MRIF**GFFITLVAII**GQ
|||||||
MRIK**GFFITLIAII**FQ

Strategy 2: embedding comparisons

MRIKGFFITLIAIIIFQ
MRIFGFFITLVAIIIGQ

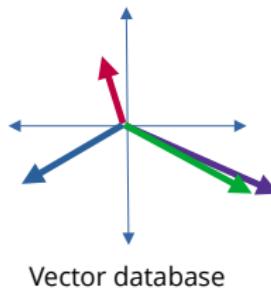
Protein Language Model embedding



Indexing 3 billion proteins

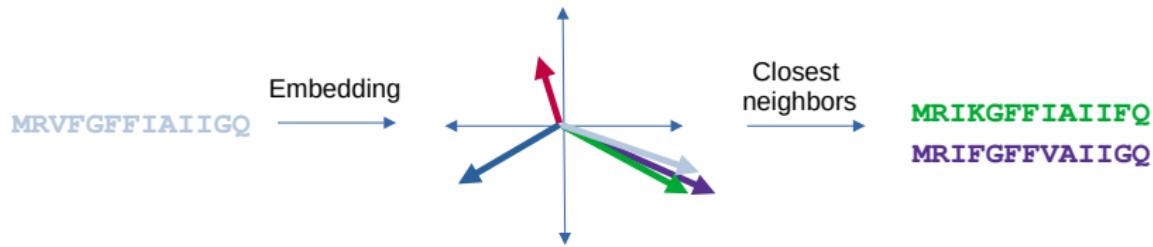
MRIKGFFITLIAIIFQ
MRIFGFFFITLVAIIGQ
MSIYHMKVRTITGKDMTLQP
MTFFFLYISPMISILIGFK
.....

Embedding



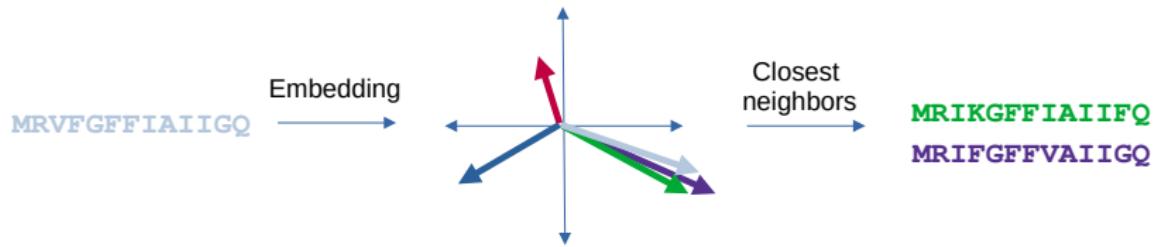
- ▶ Protein Language Model: gLM2
- ▶ 512-dimension vectors
- ▶ 3k GPU.hours
- ▶ Space taken by final database: 1.5TB

Querying 3 billion proteins



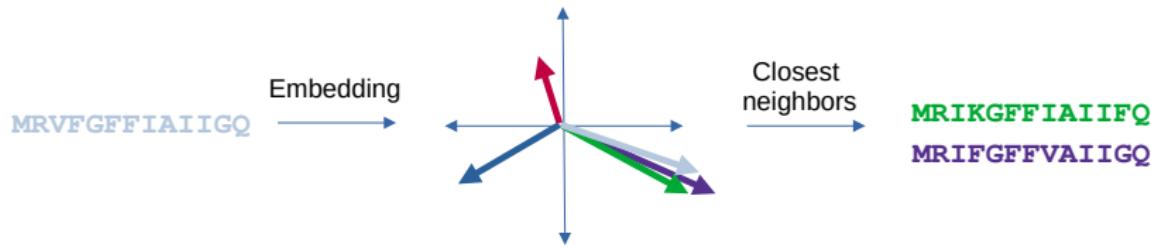
- ▶ Embed query & compare to existing vectors

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: ~ 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: ~ 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 associated

Applications

- ▶ Functional insights
- ▶ Phylogeny
- ▶ Ecology: geographical metadata associated
- ▶ ...

