Big data

Rayan Chikhi

Institut Pasteur

Workshop on Genomics 2024

Hello again again!

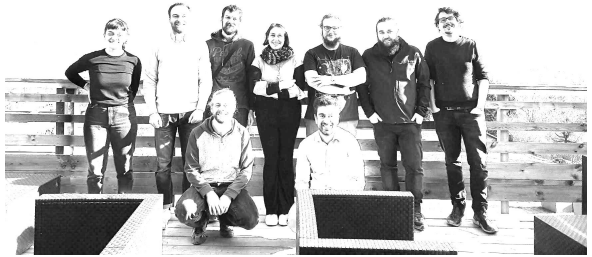
- I’m Rayan and I do bioinformatics!

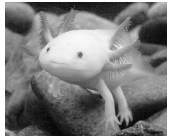
Sequence Bioinformatics

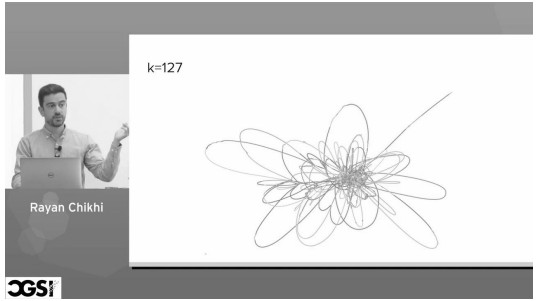
INSTITUT PASTEUR

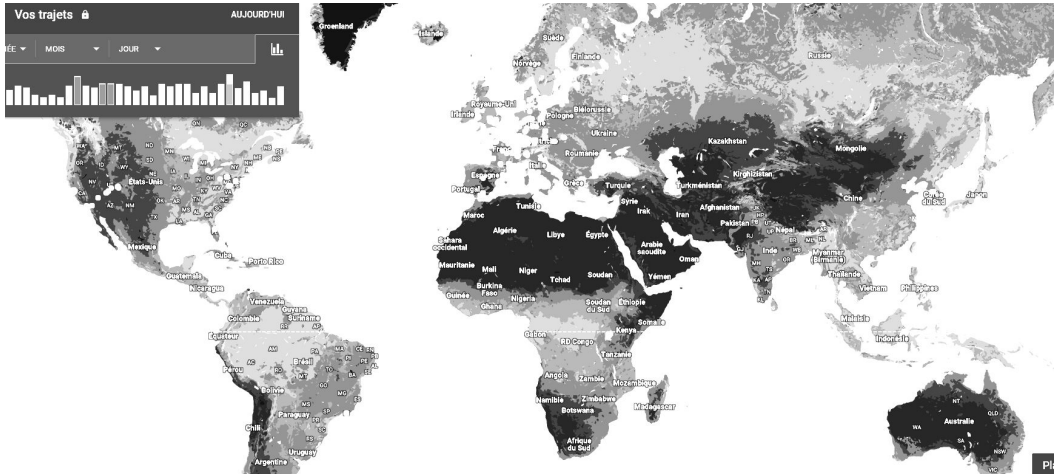
@RayanChikhi on Twitter

http://rayan.chikhi.name

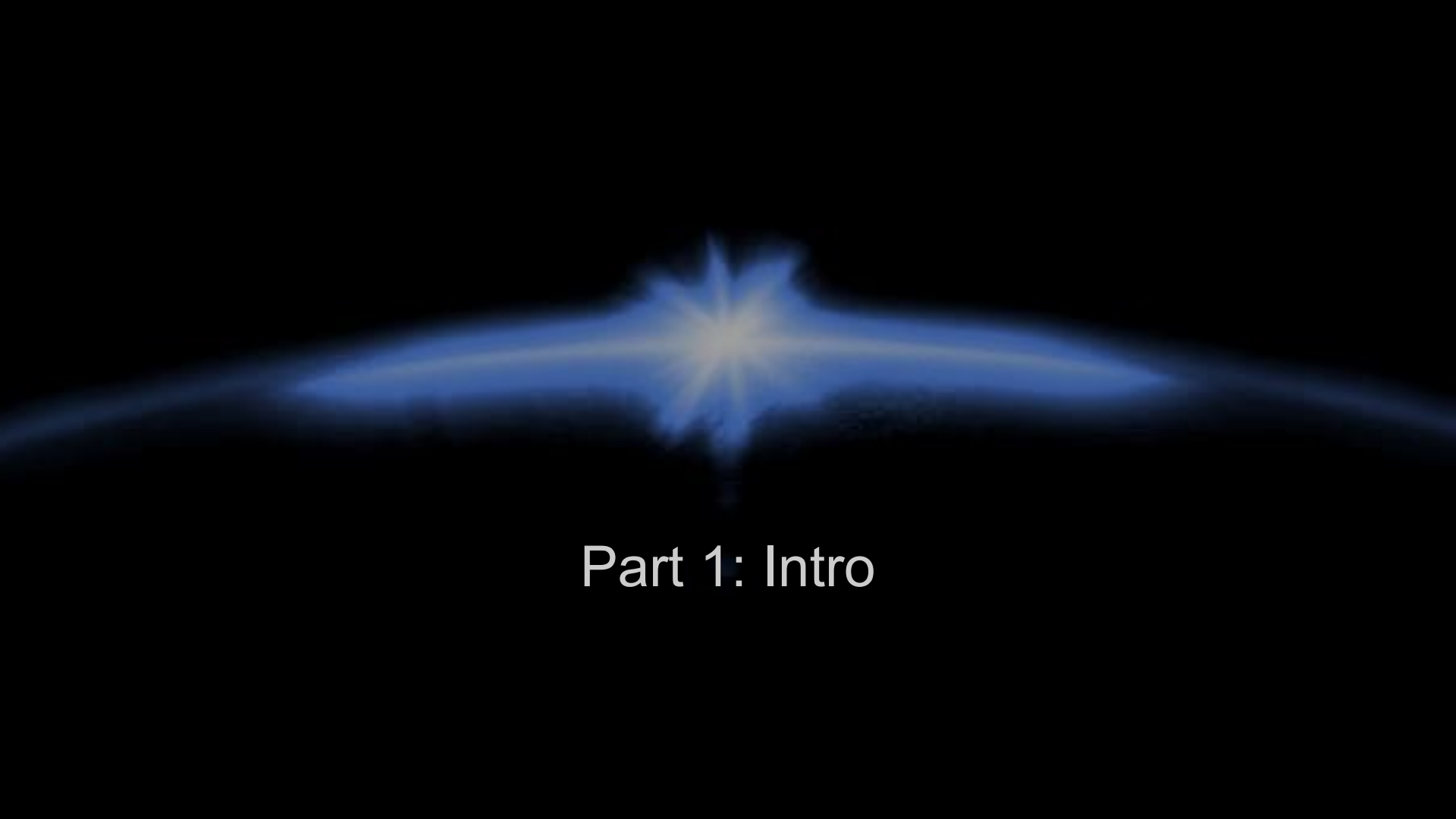






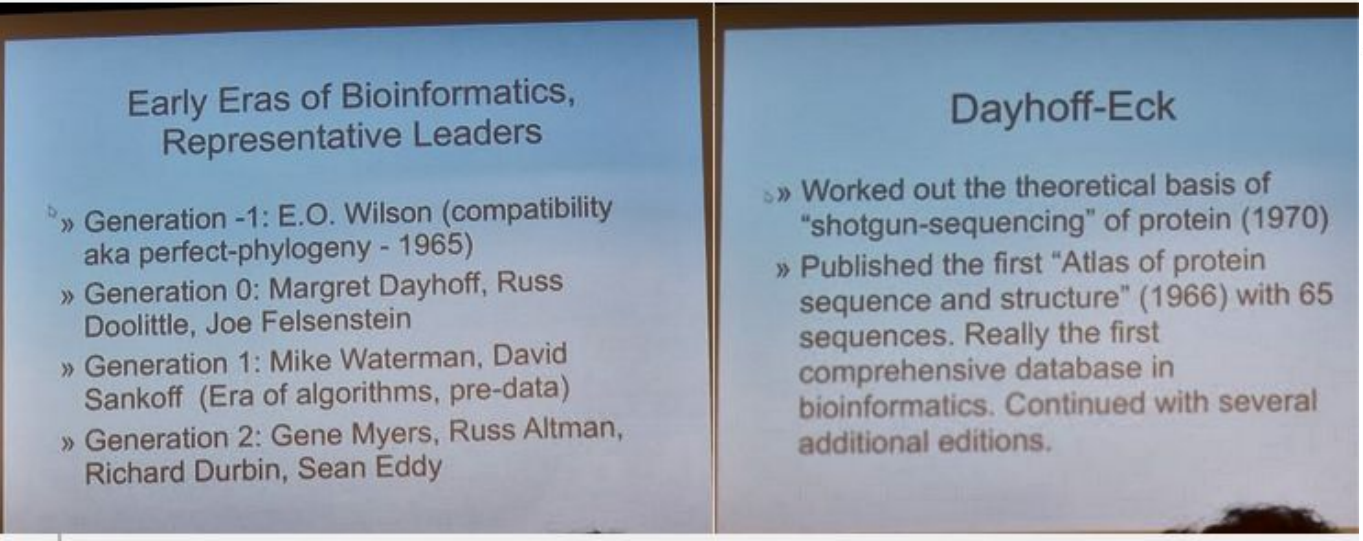


High expectations from last year - This won’t be the greatest big data talk, just a tribute



Part 1: Intro

Founding members of biological big data



Slides: Dan Gusfield

technologies to support advances in biology and medicine, most notably the creation of protein and nucleic acid databases and tools to interrogate the databases. She originated one of the first substitution matrices, point accepted mutations (*PAM*). The one-letter code used for amino acids was developed by her, reflecting an attempt to reduce the size of the data files used to describe amino acid sequences in an era of punch-card computing.

Margaret Oakley Dayhoff

Born	Margaret Belle Oakley March 11, 1925 Philadelphia, Pennsylvania
Died	February 5, 1983

Big data is the natural flow of biology

Data size

1972: single gene sequenced

2000: 1 high-quality human genome

2013: many low-quality human genomes

2021: 10 petabases of reads analyzed

2022: 1 million humans VCFs

2022: 50 high-quality human genomes

2024–: ?

The pGpOpApTp summary paragraph

The Nucleotide Sequence of *Saccharomyces cerevisiae* 5.8 S Ribosomal Ribonucleic Acid

(Received for publication, November 20, 1972)

GERALD M. RUBIN*

From the Medical Research Council Laboratory of Molecular Biology, Cambridge, CB2 2QH, England

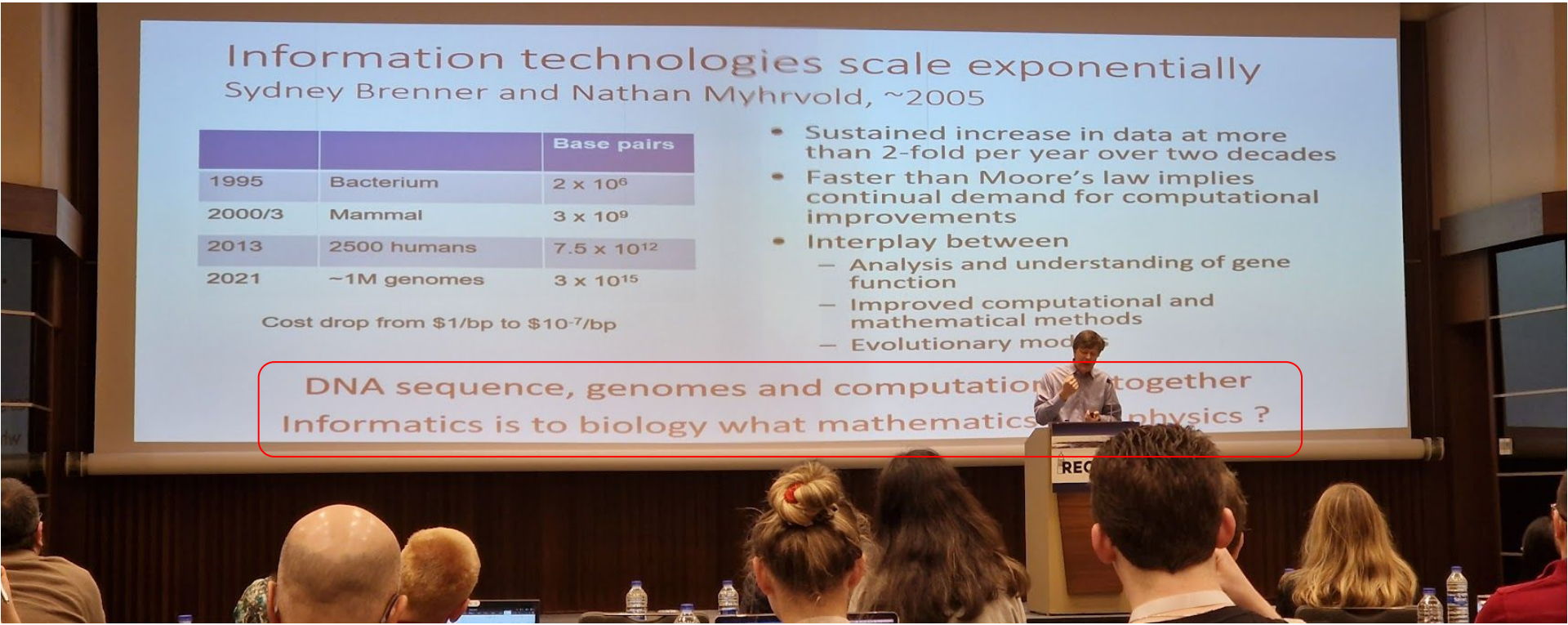
SUMMARY

The nucleotide sequence of *Saccharomyces cerevisiae* 5.8 S ribosomal RNA (also known as the 7 S or 1RNA species) has been determined to be pApApApCpUpUpUpCpApApCpA pApCpGpGpApUpCpUpCpUpUpGpGpUpUpCpUpCpGpC pApUpCpGpApUpGpApApGpApApCpGpCpApGpCpGpApA pApUpGpCpGpApUpApCpGpUpApApUpGpUpGpApAΨpUpG pCpApGpApApUpUpCpCpGpUpGpApApUpCpApUpCpGpA pApUpCpUpUpUpGpApApCpGpCpApCpApUpUpGpCpGpC pCpCpCpUpUpGpGpUpApUpUpCpCpApGpGpGpGpGpCpA pUpGpCpCpUpGpUpUpUpGpApGpCpGpUpCpApUpUpU.

Low Phosphate Medium—Inorganic phosphate was precipitated (as $MgNH_4PO_4$) from 10% Bacto-yeast extract and 20% Bacto-peptone by the addition of 10 ml of 1 M $MgSO_4$ and 10 ml of concentrated aqueous ammonia per liter. The phosphates were allowed to precipitate at room temperature for 30 min, and the precipitate was removed by filtration through Whatman No. 1 filter paper. The filtrate was adjusted to pH 5.8 with HCl and autoclaved. Sterile glucose was added to a final concentration of 2%.

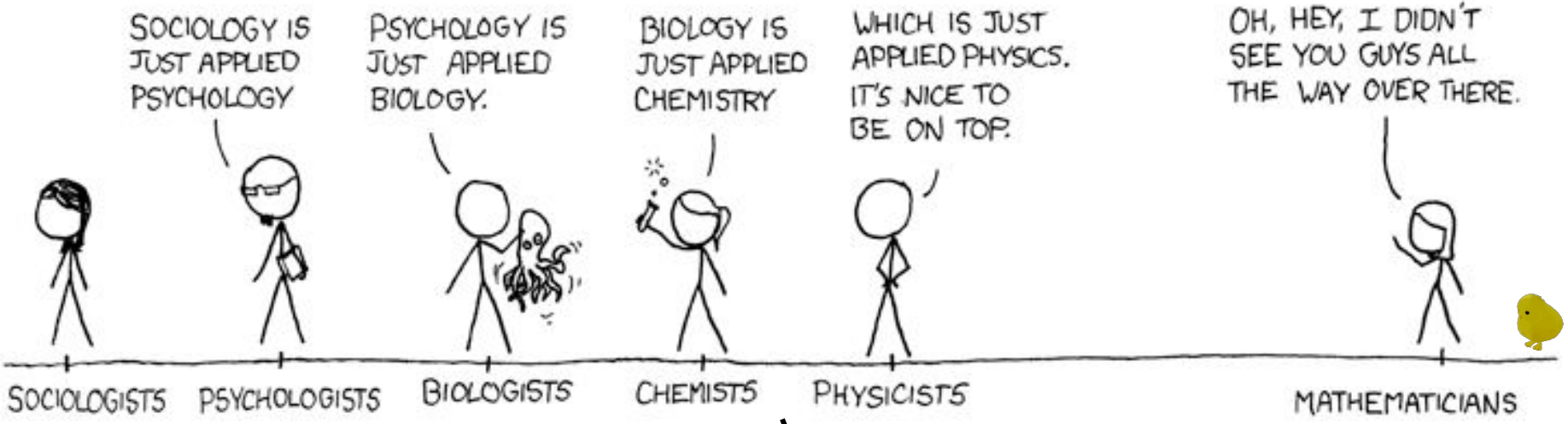
Credit: @SynBio1

Is big data just a *technical* matter?!



"Informatics is to biology,
what mathematics is to physics"

Richard Durbin, RECOMB 2023 keynote

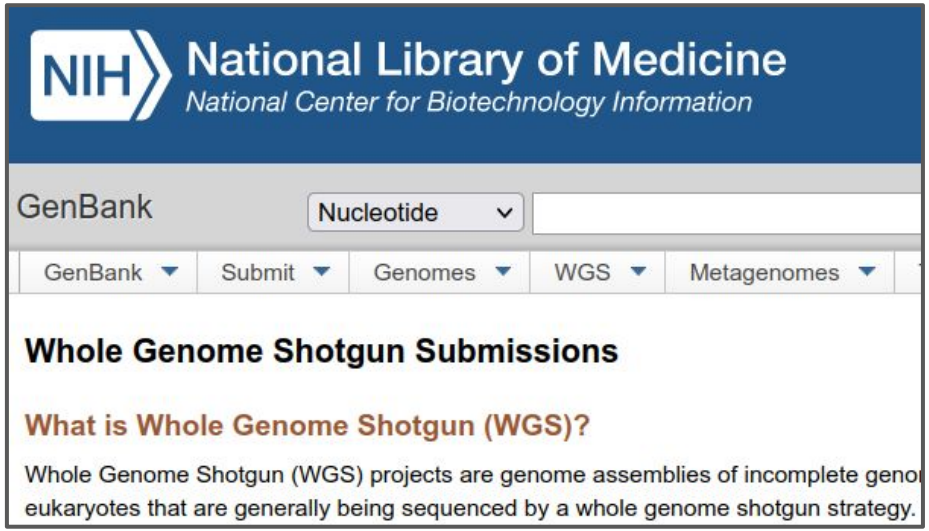


Informatics?

“purity”

“usefulness”

Big data in biology: NCBI GenBank & WGS



Type: genome assemblies of >500,000 species
Size: 1.2 terabytes (TB) (2022)

All sequences are *annotated*

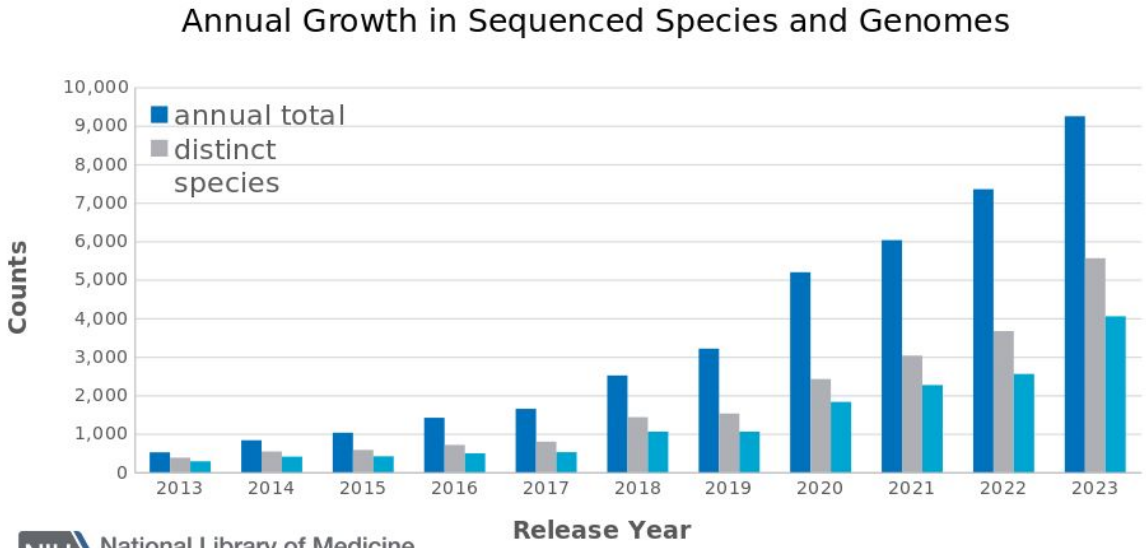
Type: genome assemblies
Size: 16 TB (2022)

Unannotated

Genome issues for comparative analyses

ALL EUKARYOTIC GENOMES (Cumulative: Dec 2023):

GenBank genomes (all):	36,593 (15,453 species)
GenBank (with annotation):	6,817 (3,801 species)



National Library of Medicine
National Center for Biotechnology Information

Slide credit: Terence Murphy, NCBI

GenBank eukaryotic genome submissions (2021):

- 55% are contaminated
- 80% lack annotation
- 20% have annotation
  - 58% have >50% proteins annotated as "*HYPOTHETICAL*"

NCBI

NCBI SRA

Type: reads
Size: 50 PB

SRA | SRA | Search
Advanced | Help

SRA

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 Heliscope®, Complete Genomics®, and Pacific Biosciences SMRT®.

Search results

Items: 1 to 20 of 19964

1. NextSeq 500 paire
   1 ILLUMINA (Illumina
   Accession: ERX34307
2. NextSeq 500 paire
   1 ILLUMINA (Illumina
   Accession: ERX34307
3. NextSeq 500 paire
   1 ILLUMINA (Illumina
   Accession: ERX34307

NextSeq 500 paired end sequencing (ERR3407135)

Metadata | Analysis (alpha) | Reads | Download

Filter: | Find | Filtered Download | What does it do?
What can the filter be applied to?

< 1 1 346553 >

View: ☑ biological reads ☐ technical reads

1. ERR3407135.1 ERS3549882
   name: NB551234:144:HL523AFXY:1:11101:5421:
   member: default
2. ERR3407135.2 ERS3549882
   name: NB551234:144:HL523AFXY:1:11101:2248
   member: default
3. ERR3407135.3 ERS3549882
   name: NB551234:144:HL523AFXY:1:11101:2566
   member: default
4. ERR3407135.4 ERS3549882
   name: NB551234:144:HL523AFXY:1:11101:2119
   member: default
5. ERR3407135.5 ERS3549882
   name: NB551234:144:HL523AFXY:1:11101:2350
   member: default

Reads (separated)

>gnl|SRA|ERR3407135.1.1 NB551234:144:HL523AFXY:1:11101:5421:1076 *F (Biological)*
ACCTGAGCGCGCAGCTCCAGTAAATCAAACGCGGCGCGGAATTTGGGATGTTCCATCAGT
TTCCAGGCGCGTTTGCCCTGACGTCGCGACATGCGTAACTGAAGCTGCCAAATATCACGG
GTAAGCGTGGTAAGGCGTTTCGGGATCGCCA

>gnl|SRA|ERR3407135.1.2 NB551234:144:HL523AFXY:1:11101:5421:1076 *R (Biological)*
ATCAACAACAGCGGGAATACCACCTCTTCCAGCCGTTGTTTCCAACCAATACGCGTTAAT
TCACCGAAACCGCGACAGCGCAATGGAACGCATCATTGCGCAGGTGTTGCAGAATACGGA
AAACCGCATCCGAAACGAGATGCGCGTTAAT

Units

yotta	[Y]	10^{24}	=	1 000 000 000 000 000 000 000 000
zetta	[Z]	10^{21}	=	1 000 000 000 000 000 000 000
exa	[E]	10^{18}	=	1 000 000 000 000 000 000
peta	[P]	10^{15}	=	1 000 000 000 000 000
tera	[T]	10^{12}	=	1 000 000 000 000
giga	[G]	10^{9}	=	1 000 000 000
mega	[M]	10^{6}	=	1 000 000
kilo	[k]	10^{3}	=	1 000
hecto	[h]	10^{2}	=	100
deca	[da]	10^{1}	=	10

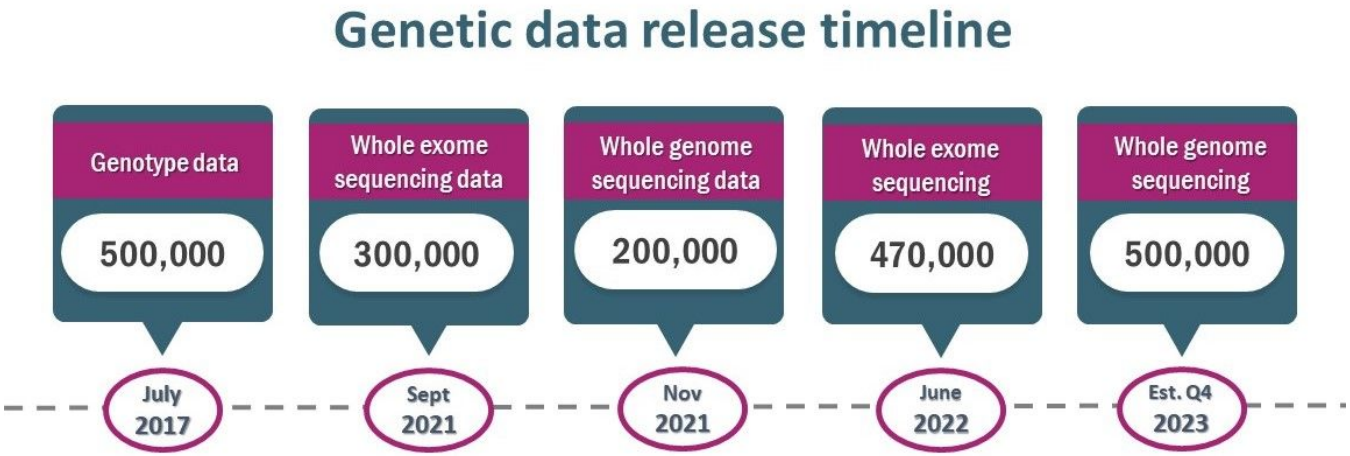
UK Biobank

Size: 25+ PB

source: https://twitter.com/uk_biobank/status/1578023831578427393

Type: reads*

* but many use just the SNPs



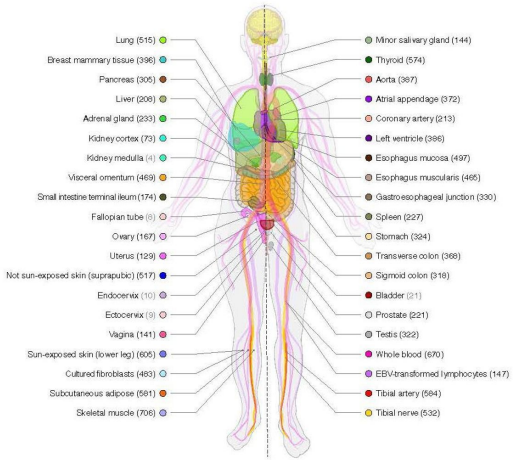
GTEx

Size: 150 TB

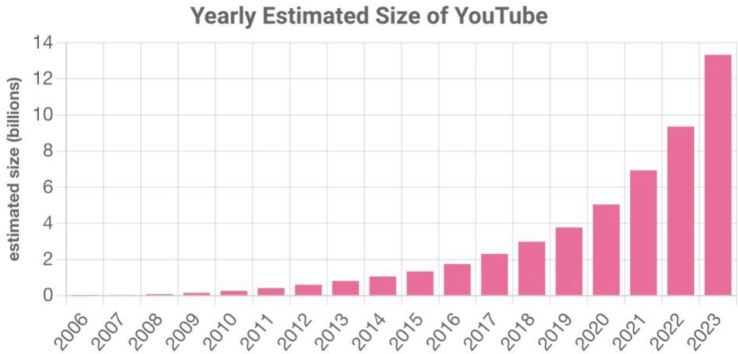
from: https://www.genomeweb.com/informatics/anvil-platform-makes-popular-nhgri-gtex-database-free-download

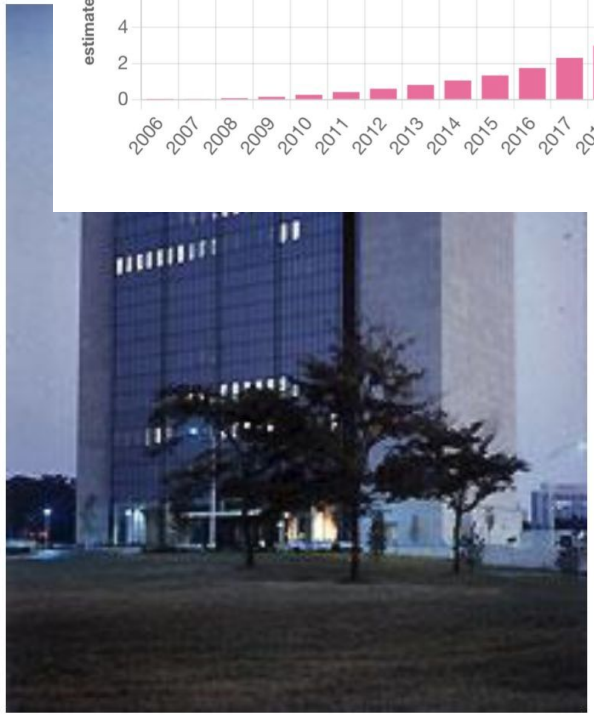
Type: reads*

* but many use just the expression data



(Youtube: 300 PB)





Institut Pasteur: 10 PB

Your laptop: 0.001 PB

State of Data Archives (2024):

With big data and big computers, one could perform wonderful, ground-breaking genomics

… But how?

People at the leading edge of a rapidly changing field "live in the future."

- Paul Buchheit (GMail creator)

“Living in the future” in biology?

- ❑ Have a lab technique only a few know
- ❑ Have data that will only be public later
- ❑ Hold a belief that isn’t established yet
- ❑ Discover for the first time that [*some phenomenon*] happens
- ❑ Work on “sci-fi” projects (e.g. create a cell from scratch, genome editing, ..)
- ❑ ...

“Living in the future” in ~~biology~~ bioinformatics

- ❑ Have a ~~lab~~ *computational* technique only a few know
- ❑ Have data that will only be public later
- ❑ Work on “sci-fi” projects (e.g. quantum computing, AI, big data, ..)

Some people living in the future

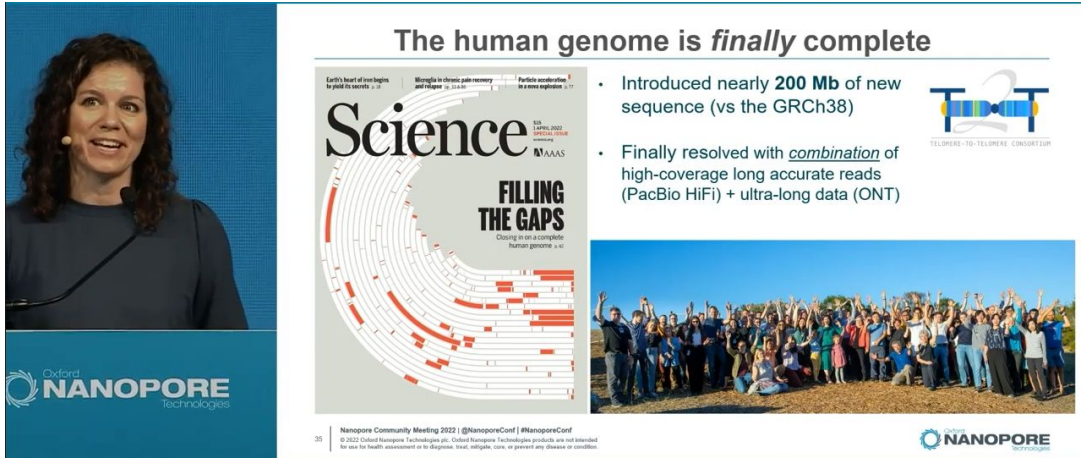
- George Church, Craig Venter
- Karen Miga & T2T team*
- Evan Eichler, Erik Garrison
- **All researchers****

MICROBIOLOGY

Scientists Create the Smallest-Ever Moving Cell

Just two genes get tiny synthetic cells moving, offering clues to life's evolution

By Saugat Bolakhe on April 1, 2023



* While the rest of the world still used GRCh38/hg19

** Generally ~months ahead, with your papers to be published

Part 2: Big Data Toolbox

<u>Computation</u>

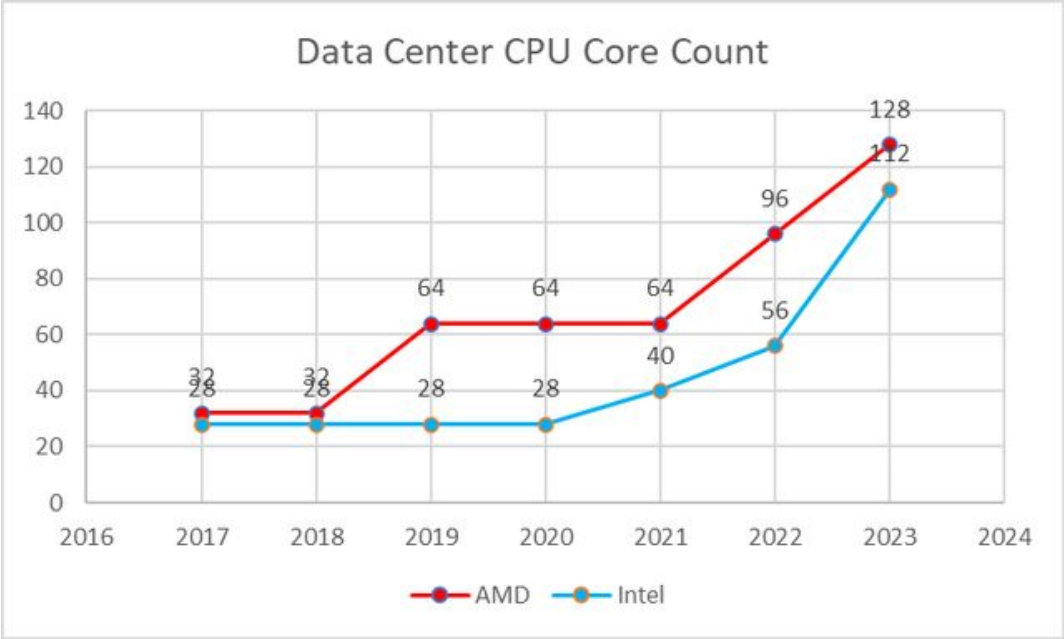
- Big computers, Cloud, Cluster
- Storage management
- Galaxy
- Knowledge of scaling limits
- Knowledge of cloud costs
- GNU parallel

<u>Data mining</u>

- Pebblescout, branchwater
- ORA
- deCOM
- SRA metadata

Future genomics, today?

No such thing as ‘big data’, only ‘small computers’



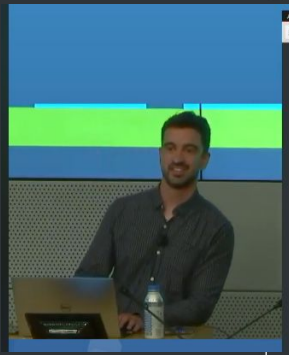
source
https://seekingalpha.com/article/4468119-advanced-micro-devices-amd-server-roadmap-not-strong-enough

Cloud

= *A collection of computers owned by a single organization and accessible from the Internet*

Recap of last year's talk


```
6a.48xlarge:~$ aws s3 cp s3://sra-pub-src-2/SRR11292120/m64062_190806_063919.fastq.1
  --no-sign-request
ompleted 4.6 GiB/39.1 GiB (278.0 MiB/s) with 1 file(s) remaining
```



```
96[||||||||||||||||||||||||||||      ] Tasks: 45, 263 thr          ; 192 runnin
97[||||||||||||||||||||||||          ] Load average: 84.83 25.05 8.99
98[||||||||||||||||||||||||          ] Uptime: 01:29:46
99[||||||||||||||||||||||||          ]
100[||||||||||||||||||||||||         ]
101[||||||||||||||||||||||||         ]
102[||||||||||||||||||||||||         ]
103[||||||||||||||||||||||||         ]
104[||||||||||||||||||||||||         ]
105[||||||||||||||||||||||||         ]
106[||||||||||||||||||||||||         ]
107[||||||||||||||||||||||||         ]
108[||||||||||||||||||||||||         ]
109[||||||||||||||||||||||||         ]
110[||||||||||||||||||||||||         ]
```

```
M::worker_pipeline::114.476*64.47] mapped 24103 sequences
M::worker_pipeline::116.688*64.99] mapped 24151 sequences
M::worker_pipeline::118.523*65.75] mapped 24157 sequences
M::worker_pipeline::119.921*66.67] mapped 24213 sequences
M::worker_pipeline::121.669*67.42] mapped 24135 sequences
M::worker_pipeline::123.241*68.13] mapped 24129 sequences
M::worker_pipeline::124.775*68.95] mapped 24160 sequences
M::worker_pipeline::126.667*69.41] mapped 22755 sequences
M::main] Version: 2.26-r1175
M::main] CMD: minimap2 -xmap-hifi -t192 chm13v2.0.fa m64062_190806_063919.fastq.1
M::main] Real time: 127.332 sec; CPU: 8792.248 sec; Peak RSS: 12.868 GB
547.99user 244.33system 2:07.42elapsed 6900%CPU (0avgtext+0avgdata 13493472maxresid
)k
inputs+344088outputs (0major+7316184minor)pagefaults 0swaps
6a.48xlarge:~$
1] 0:time* 1:htop-                    "ip-172-31-65-227.ec2." 23:39 17-Jul-
```

Needto Mapquik

Live: Demo of mapping human 10x coverage HiFi reads using mapquik in <20 seconds, including FASTA conversion using seqkit and chatgpt

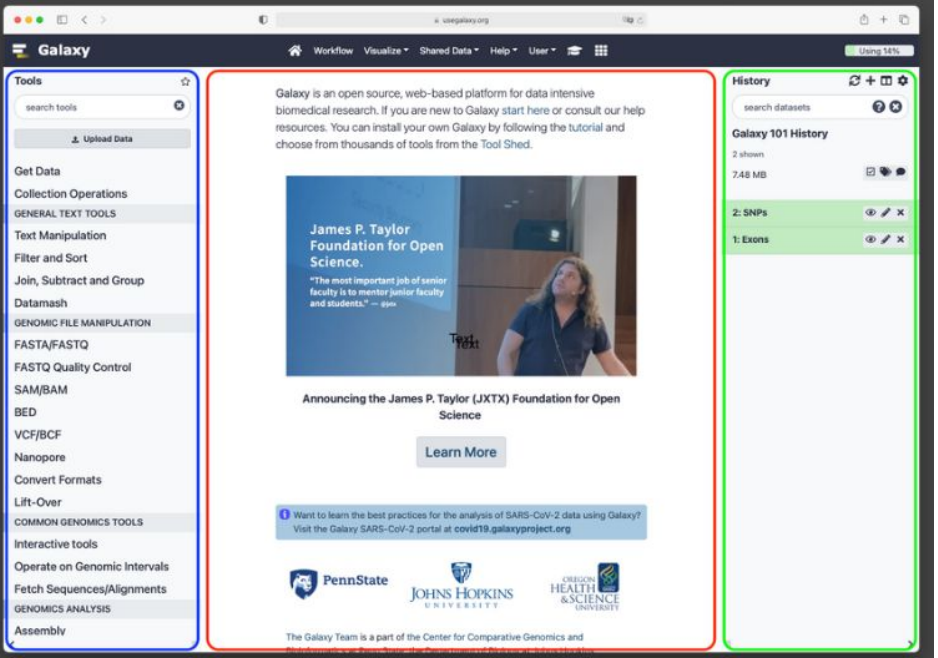
Galaxy Project

Galaxy

Data Intensive *analysis* for everyone

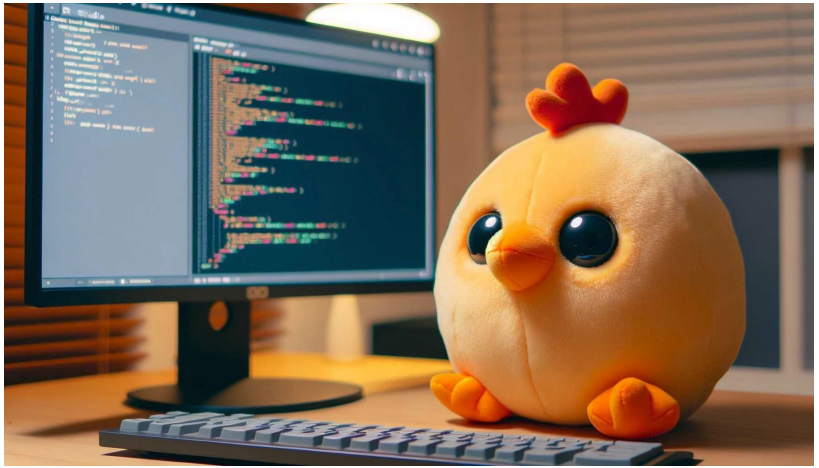
- Versatile and reproducible workflows
- **Web** platform
- **Open source** under Academic Free License

Main Galaxy interface



- If you do not have a cluster
- ..or the will to install tools..
- Galaxy offers free computation on pre-installed workflows

Cluster



Acquire knowledge about it:

- Queues:
  - How many CPUs/RAM per job, what timelimit
  - Can your group access any ✨*special* queue✨
- Storage:
  - Your quota
  - Is “scratch” quota-free? Do files expire?

My scripts:

```
srun -q seqbio -p seqbio --mem 100G -c 10 --pty bash
```

Quickly allocates a terminal on any machine

```
squeue -o "%.18i %.9P %.8j %.8u %.2t %.10M %.6D %R cores:%c mem:%m cmd:%o " | grep seqbio
```

See what machines are currently being used

Storage management

- How to never run out of storage space:
  - Have 2 folders:
    - `~/archive`
    - `~/scratch`
  - Rules:
    - Archive is backed up, contains command lines and final results
    - Scratch is fast, but may be deleted at any time
    - Keep the list of files for both, somewhere
  - Keep a dummy 100 GB file ready to be deleted
- Data compression
  - BAM => CRAM => delete it
  - FASTQ => gzip => delete it
  - VCF => BCF
  - GFF/GTF => `don't annotate`

Part 2: Big Data Toolbox

Computation

- Big computers, Cloud, Cluster
- Galaxy
- Storage management
- Knowledge of scaling limits
- Knowledge of cloud costs
- GNU parallel

Data mining

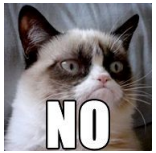
- Pebblescout, branchwater
- ORA
- deCOM
- SRA metadata

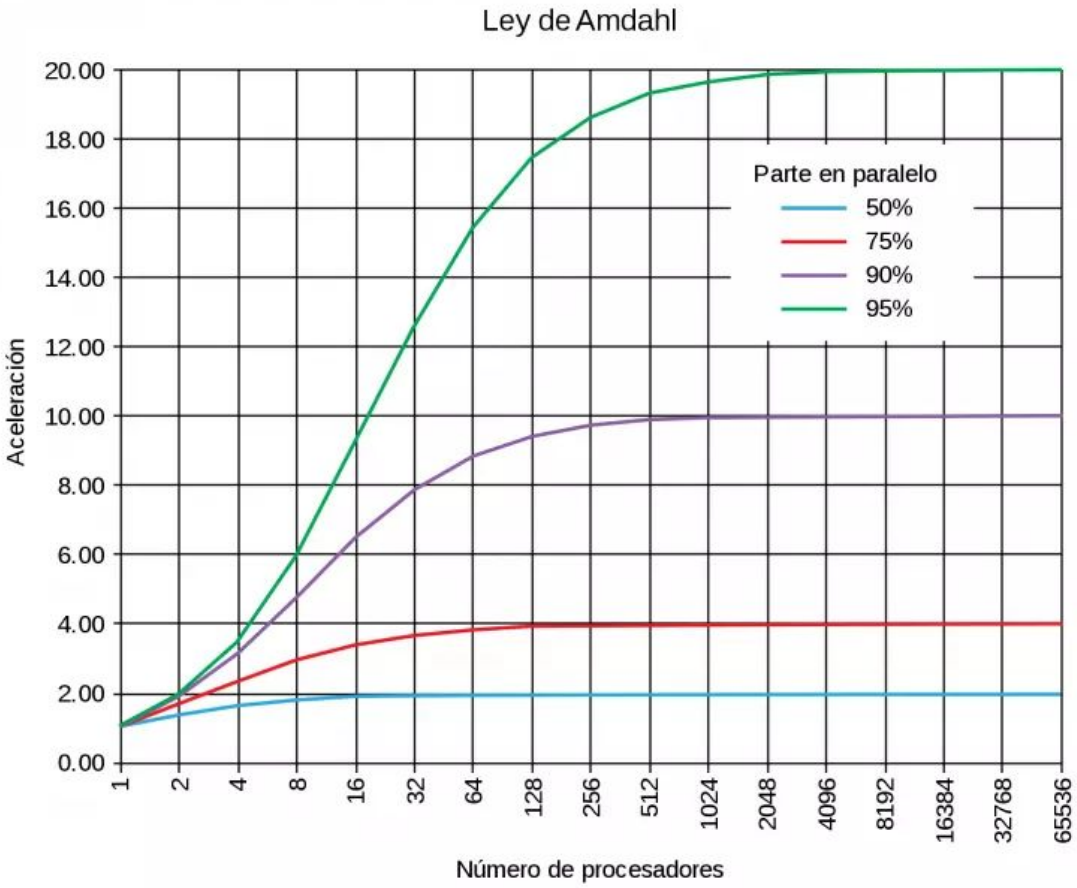
Knowledge of scaling limits

In order of difficulty:

1. **Estimate** how long an analysis will take
2. Reasons **why** some analyses are slower than expected
3. **How** to reduce that time

Do 200 CPUs always go 200x faster?

Amdahl's law: 



Connect the dots from left to right

1) Access data from a SSD disk

2) Access data in memory

3) Access http://www.evomics.org in Australia

4) Human cell cycle

5) Align 1 million short reads

- 100 nanoseconds
- 100 microseconds
- 200 milliseconds
- 10 seconds
- 24 hours

n	nano	10^{-9}
μ	micro	10^{-6}
m	milli	10^{-3}

Connect the dots from left to right

- Access data from a SSD disk
- Access data in memory
- Access http://www.evomics.org in Australia
- Human cell cycle
- Align 1 million short reads

- 100 nanoseconds
- 100 microseconds
- 200 milliseconds
- 10 seconds
- 24 hours

n	nano	10^{-9}
μ	micro	10^{-6}
m	milli	10^{-3}

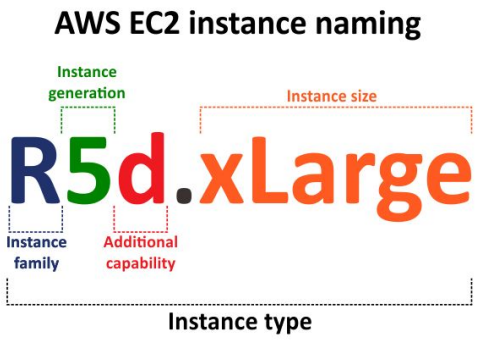
Knowledge of scaling limits

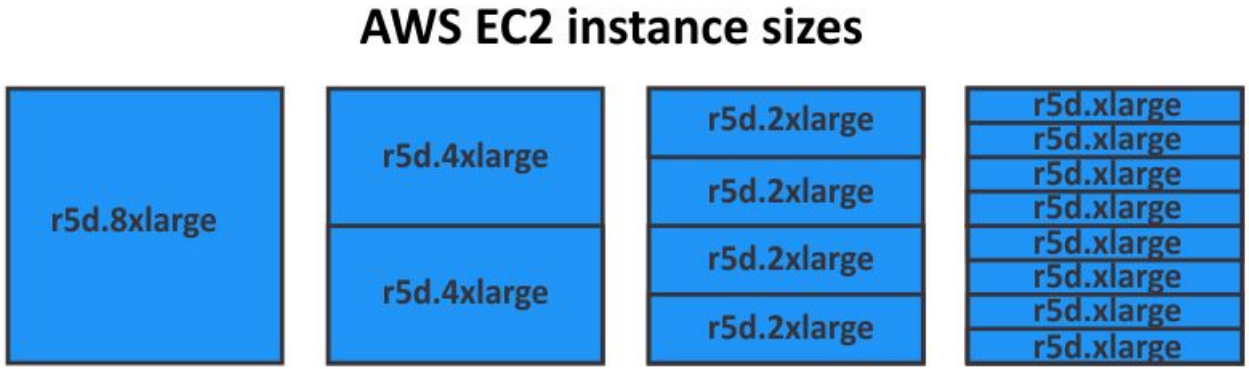
In order of difficulty:

1. **Estimate** how long an analysis will take
   - Look at performance table in tool paper
   - Try on smaller data and extrapolate
2. Reasons **why** some analyses are slower than expected
   - Limited number of CPUs
   - Limited RAM
   - Slow disk (HDD < Cluster network drives < SSD < NVMe)
3. **How** to reduce that time
   - Most analyses go fast enough on a big cloud/cluster and the right tools

Knowledge of cloud costs

Your workshop instance: `t3a.large` : 2 CPU cores, 8 GB memory
15 cents per hour, 3$/day





💕 c6a.48xlarge 💕 : 192 cores, 384 GB mem, 7$/hour

All costs: https://instances.vantage.sh/

Knowledge of cloud costs

Storage costs!

EBS (instances hard drive): $0.08/GB/month

S3 (“Dropbox”): $0.023/GB/month

- If an instance is stopped: EBS costs occur
- If you create an instance snapshot: EBS costs occur too

How to avoid these costs? Terminate instances, delete snapshots, don’t store too much on your S3

General scaling considerations

- Alignment
  - Highly parallel, low memory, scales well with number of CPUs
- Assembly
  - Moderately parallel, high memory, typically requires a single big machine
- Annotation
  - Don’t! (jk), but moderately parallel. Single machine too?
- Phylogenomics
  - Can be made parallel (RAxML, Iq-Tree)

GNU parallel

Allows to run the same job on multiple files, simultaneously. Circumvents SLURM.

To count number of lines across many FASTQ files:

```
find . -name *.fastq | parallel -j10 "wc -l {} > {}.nb_lines"
```

To run many jobs defined by CSV data:

```
cat data.csv | parallel --colsep ','  "./myprogram {1} {2}"
```

Part 2: Big Data Toolbox

Computation

- Big computers, Cloud, Cluster
- Storage management
- Galaxy
- Knowledge of scaling limits
- Knowledge of cloud costs
- GNU parallel

Data mining

- Pebblescout, branchwater
- ORA
- deCOM
- SRA metadata

Exploring metagenomes: Pebblescout and Branchwater

- Cutting-edge sequence database search tools
- Think BLAST, but the database is no longer “nr”; it’s all metagenomes.

National Library of Medicine
National Center for Biotechnology Information

PebbleScout BETA Search Documentation

Pebblescout pre-indexes nucleotide resources and searches them. The index contains at least one 25-mer from every 42-mer for all subjects in the database. Search has three modes: profile, summary, and detailed. Summary search ranks matching subjects using Pebblescout score. Search generates hashes from given user queries using the same scheme as used for indexing. This guarantees that every 42 bp match between the user query and any subject in the database is found.

Seven databases currently available are as follows:

1. **Metagenomic:** All metagenomic and metatranscriptomic runs released in public SRA before the end of 2021
2. **WGS:** All assemblies for the Whole Genome Shotgun sequencing projects available as of Feb 14, 2022
3. **RefSeq:** All assemblies available in the Reference Sequence collection as of April 22, 2022
4. **PH2HS_Runs:** Runs from Phase 3 of the 1000 Genomes project
5. **PH3HS_Biosample:** Runs from Phase 3 of the 1000 Genomes project where all runs for the same BioSample are considered as one subject
6. **Human RNAseq 2021:** All Human RNAseq runs released in public SRA in the year 2021
7. **Virus PacBio HiFi:** Viral samples sequenced with the PacBio SMRT technology defined in PMC9528980

Documentation provides additional information. A preprint for the Pebblescout manuscript is available at biorxiv.

Please provide nucleotide queries, choose database and type of search to be performed, change parameters, as needed, and click View or Download. Please re-click View or Download if you change inputs.

Type FASTA Lines or GenBank Accessions Separated by Commas

Type FASTA lines here (sequence length must be at least 42 bases) or comma separated list or GenBank accessions

or Upload FASTA File

- All metagenomes, all assemblies (WGS), all human RNAseq, RefSeq
- Search for any sequence > 42 nt

Pebblescout usage example

Collaborator needs to search SRA for all samples containing Wolbachia

PebbleScout BETA

We did exactly this in our paper!

- 36 host species were known for Wolbachia
  - Found by searching SRA metadata (2,545 runs)
- Pebblescout search for 3 genes (ftsZ, groE, wsp)
  - Found 16 more hosts (35 runs)

Branchwater Metagenome Query

Real-time search for a genome within metagenomes in the SRA.

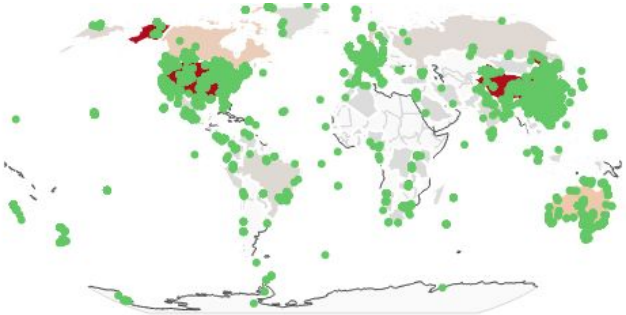
Your query returned 11100 accession IDs. The returned metadata can be pre-filtered prior to .CSV download and plotting with the table below. Your filtered table contains 11100 accession IDs

Download CSV

acc	assay_type	bioproject	biosample_link	cANI	collection_date_...	containment	geo_loc_name_c...	lat_lon	organism
				Min Max		Min Max			
SRR14986175	WGA	PRJNA742226	https://www.ncbi.nl...	0.9	2017-06-14	0.12	Germany	49.61,10.28	soil metagenome
SRR6958475	WGS	PRJNA444974	https://www.ncbi.nl...	0.95	2012-05-01	0.37	USA	33.5944,-109.1397	soil metagenome
SRR3501856	WGS	PRJNA320780	https://www.ncbi.nl...	0.9	2015-07-03	0.11	Singapore	1.33,103.75	activated sludge met...
SRR8925775	WGS	PRJNA681092	https://www.ncbi.nl...	0.9	2017-10-23	0.12	China	36.19,111.59	bioreactor metagen...

Compared to Pebblescout:

- Only support long queries (> 10 kbp)
- More verbose output/visualizations



OCEAN READ ATLAS

ONE CLICK MARINE K-MER BIOGEOGRAPHY

kmindex and ORA: indexing and real-time user-friendly queries in terabyte-sized complex genomic datasets

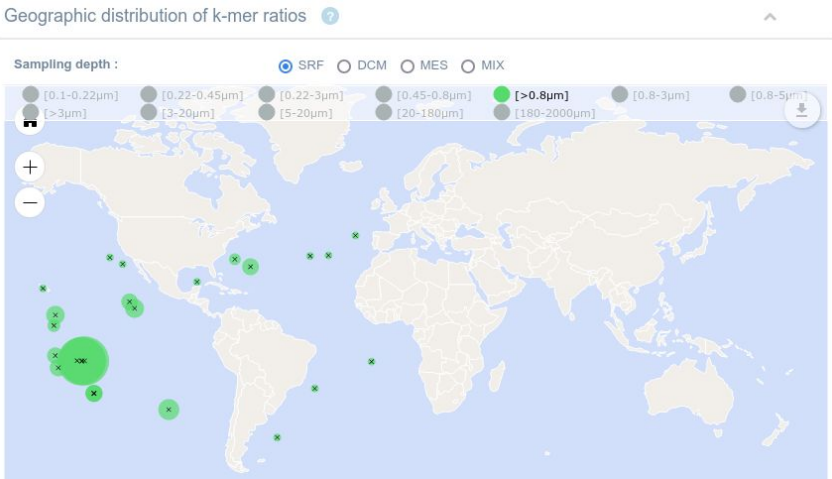
Lemane et al,
2023 (BioRxiv)
2024 (Nat Comp Biol)

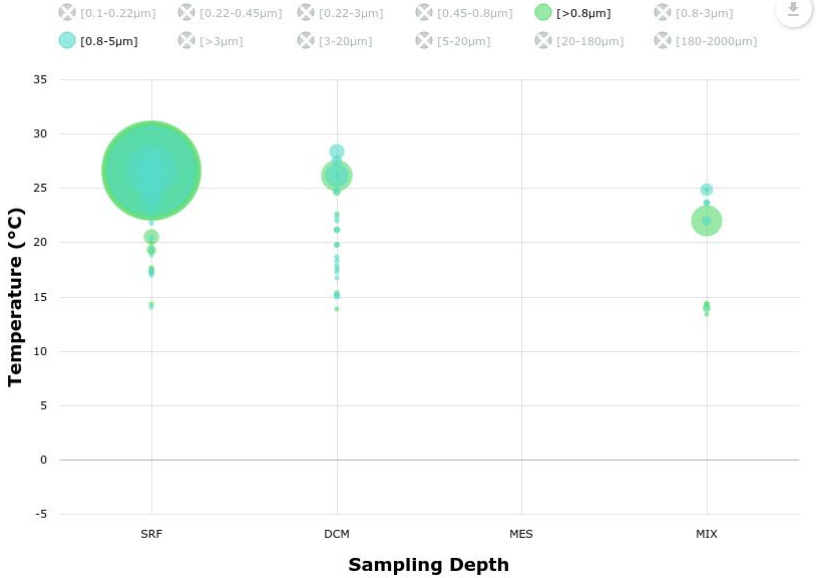
All TARA data,
Supports short queries,
Instant results

Dataset: TARA

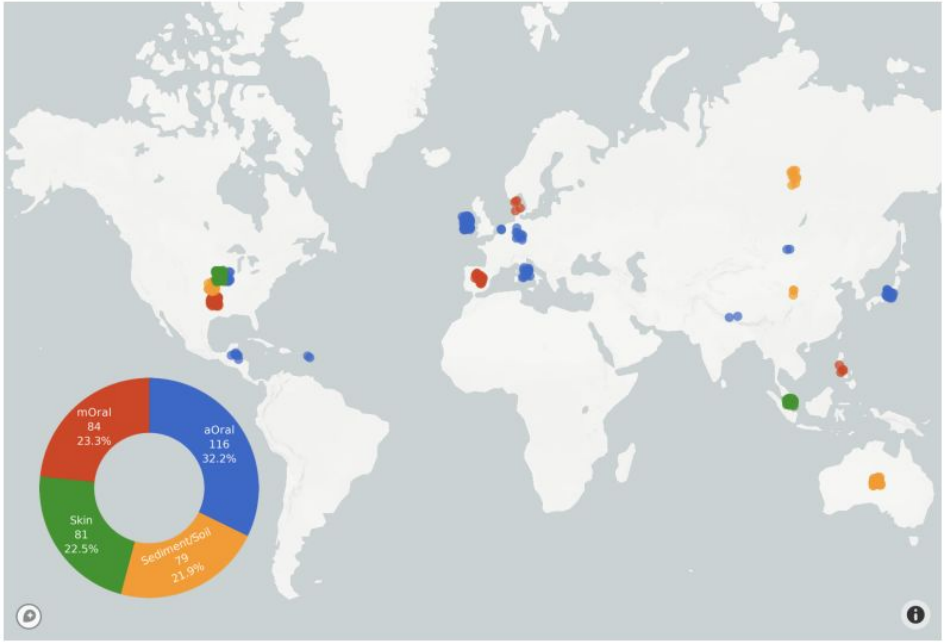
Job title: nifH_gene_example

Query sequence:
>nifH_gene LT907975.1:3538795..3539625 [Pseudodesulfovibrio profundus]
atgagaaaagtagcaatttacggaaaaggcggcattggaaaatccaccaccactcagaac
actgtcgccggtttggcggaaatgggccgca
gccgactccacccgcctgttgctcggtggtct
cgtgaagagggcgaggatgtggaactcga





deCOM: integrating all ancient oral metagenomes



We gathered a collection of 360 samples (including contaminants and non contaminants) and obtained a k-mer matrix

	S1	S2	S3	S4	S5	S6	S7	S8
AATCG	1	0	0	0	0	1	1	0
GGGCT	0	0	0	0	0	1	1	0
TTCGA	0	0	1	1	0	1	1	0
AAACG	0	0	0	0	0	1	1	0
GGGCT	0	0	0	1	1	0	0	1
AATTT	0	0	0	0	0	0	0	0
ATCCC	0	0	1	0	0	0	0	0
GGGGT	1	1	1	1	1	1	1	1

New Results

Follow this preprint

decOM: Similarity-based microbial source tracking of ancient oral samples using k-mer-based methods

Camila Duitama González, Riccardo Vicedomini, Téo Lemane, Nicolas Rascovan, Hugues Richard, Rayan Chikhi

doi: https://doi.org/10.1101/2023.01.26.525439

This article is a preprint and has not been certified by peer review [what does this mean?].

0 0 0 0 0 0 21

SRA metadata

.. will be presented in the next part

Wrapping up of Part 2: Big Data Toolbox

<u>Computation</u>

- Big computers, Cloud, Cluster
- Galaxy
- Storage management
- Knowledge of scaling limits
- Knowledge of cloud costs
- GNU parallel

<u>Data mining</u>

- Pebblescout, branchwater
- ORA
- deCOM
- SRA metadata

Part 3

SRA-scale sequence exploration

NCBI SRA

All public sequencing reads

Size: 47 PB
as of late 2023

SRA | SRA | Search
Advanced | Help

SRA

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 Heliscope®, Complete Genomics®, and Pacific Biosciences SMRT®.

Search results

Items: 1 to 20 of 19964

1. NextSeq 500 paire
   1 ILLUMINA (Illumina
   Accession: ERX34307
2. NextSeq 500 paire
   1 ILLUMINA (Illumina

NextSeq 500 paired end sequencing (ERR3407135)

Metadata | Analysis (alpha) | Reads | Download

Filter: Find | Filtered Download | What does it do?
What can the filter be applied to?

< 1 1 346553 >

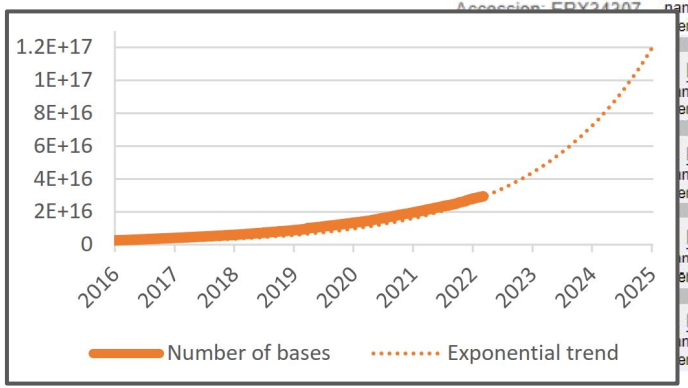
View: biological reads | technical reads

1. ERR3407135.1 ERS3549882
   name: NB551234:144:HL523AFXY:1:11101:5421:
   ember: default
2. ERR3407135.2 ERS3549882
   me: NB551234:144:HL523AFXY:1:11101:2248
   ember: default
3. ERR3407135.3 ERS3549882
   me: NB551234:144:HL523AFXY:1:11101:2566
   ember: default
4. ERR3407135.4 ERS3549882
   me: NB551234:144:HL523AFXY:1:11101:2119
   ember: default
5. ERR3407135.5 ERS3549882
   me: NB551234:144:HL523AFXY:1:11101:2350
   ember: default

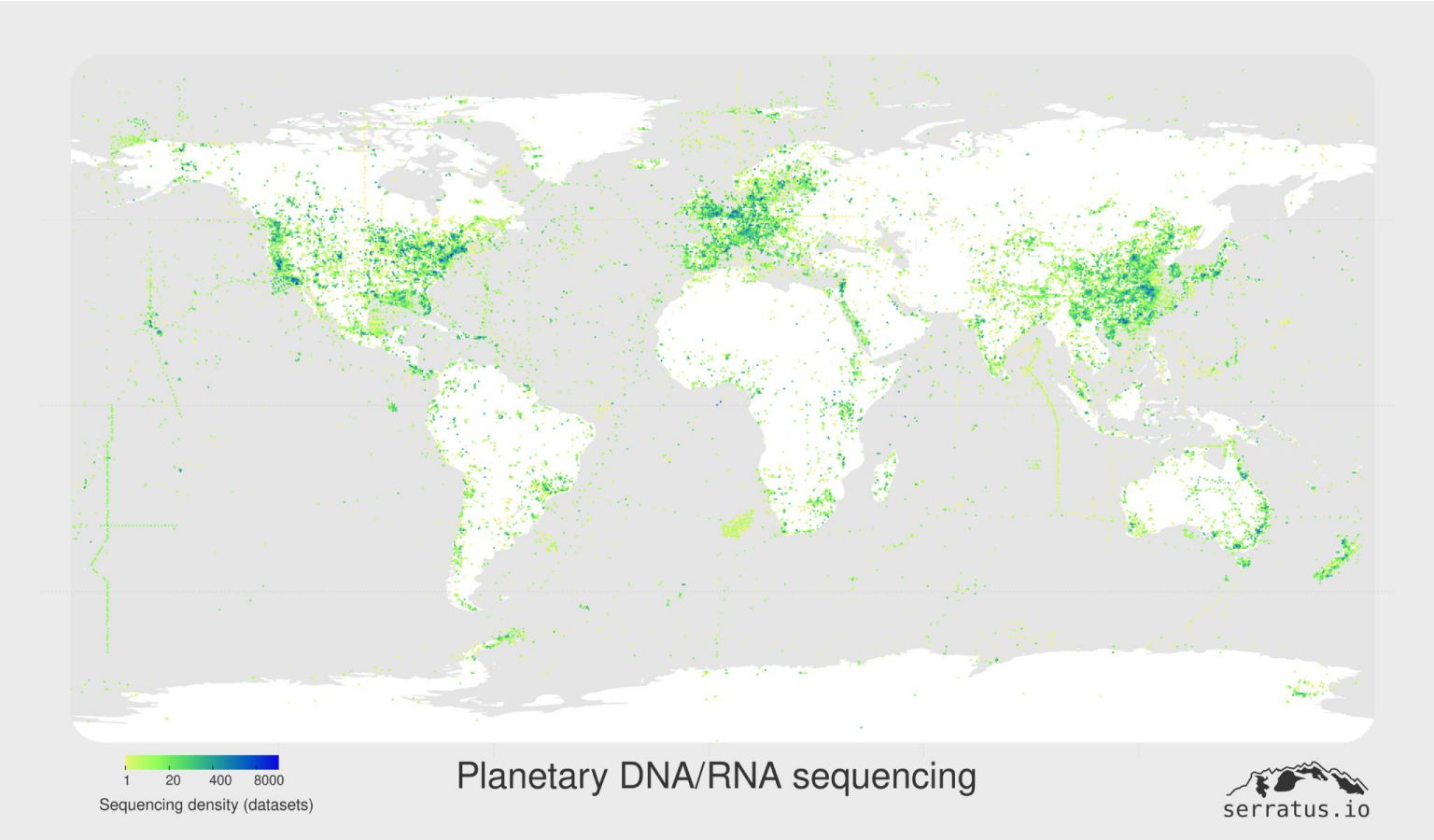
Reads (separated)

>gnl|SRA|ERR3407135.1.1 NB551234:144:HL523AFXY:1:11101:5421:1076 *F (Biological)*
ACCTGAGCGCGCAGCTCCAGTAAATCAAACGCGGCGCGGAATTTGGGATGTTCCATCAGT
TTCCAGGCGCGTTTGCCCTGACGTCGCGACATGCGTAACTGAAGCTGCCAAATATCACGG
GTAAGCGTGGTAAGGCGTTTCGGGATCGCCA

>gnl|SRA|ERR3407135.1.2 NB551234:144:HL523AFXY:1:11101:5421:1076 *R (Biological)*
ATCAACAACAGCGGGAATACCACCTCTTCCAGCCGTTGTTTCCAACCAATACGCGTTAAT
TCACCGAAACCGCGACAGCGCAATGGAACGCATCATTGCGCAGGTGTTGCAGAATACGGA
AAACCGCATCCGAAACGAGATGCGCGTTAAT



Geography of SRA samples

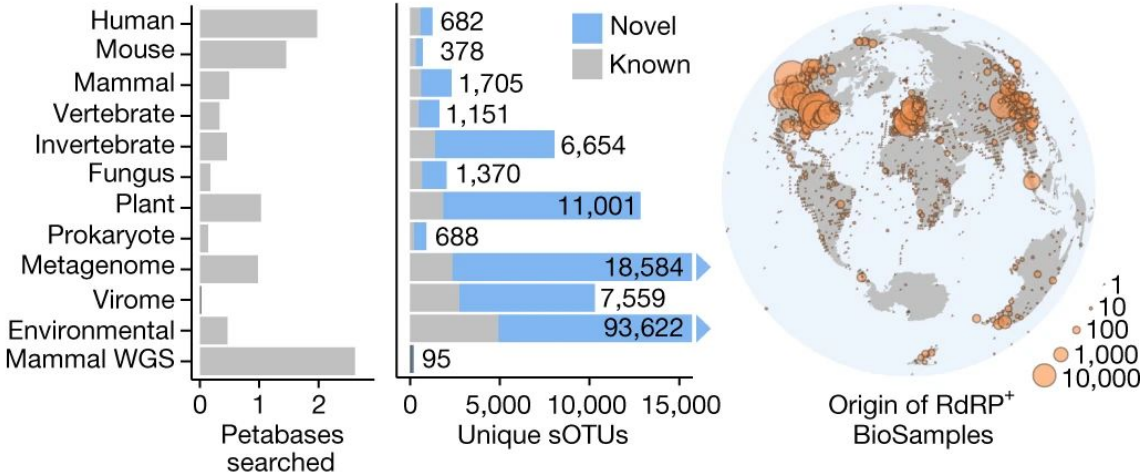


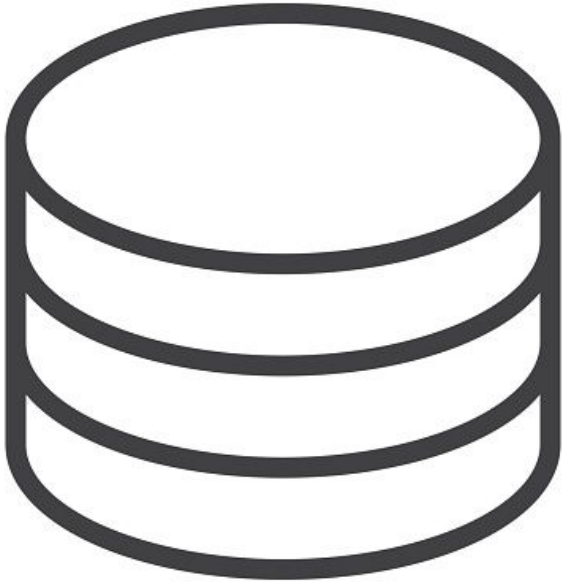
Credit: A. Babian

What to do with the entire SRA?

Serratus: all public RNA-seqs analyzed for viral discovery

Discovery of 130,000 new RNA viral species. One-off analysis, 20,000 CPUs (Nature, 2022)





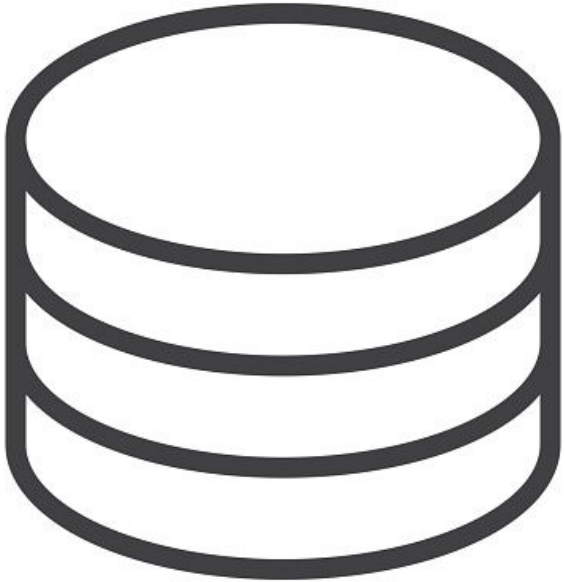
All RNA-seqs
pre-2020

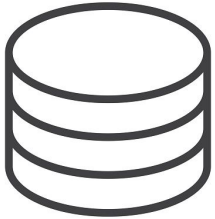
(10 petabases)

Serratus download & align (bowtie2**) to all viral reference genomes**

56,000 CoV+ samples
including 9 novel coronavirus species discovered

Serratus download &
sensitive align
(DIAMOND2**)**
to all known versions of
RNA virus universal gene





aligned reads
(.bam files)
130k novel species
discovered

All RNA-seqs
pre-2020

Toolbox used in Serratus

Part 2: Big Data Toolbox

Computation

- Big computers, Cloud
- Galaxy
- Knowledge of scaling limits
- Knowledge of cloud costs
- GNU parallel

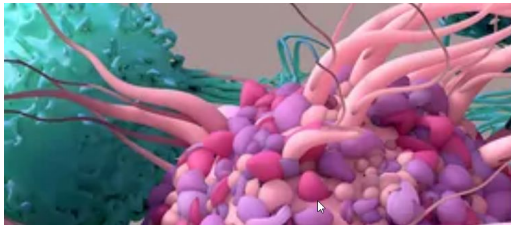
Data mining

- Pebblescout, branchwater
- ORA
- deCOM
- SRA metadata

Didn't exist

Some follow-ups to Serratus

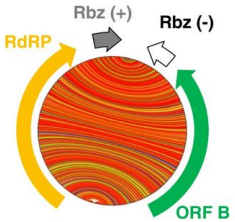
Viral reactivation (Nature 2023)



Discovered HHV-6 reactivation in CAR-T cells

Independent use of Serratus data

Ambiviruses (Nat Comm 2023)



50 known => 20,000 discovered viroids

Analysis of **circular contigs** in Serratus assemblies

Diving into SRA’s data

What's SRA metadata?

All this information

SRX8451857: Resequencing of Vicugna vicugna V_ss18
1 ILLUMINA (HiSeq X Ten) run: 111.2M spots, 33.4G bases, 11.8Gb downloads

Design: Resequencing

Submitted by: Universidad Austral de Chile

Study: Resequencing of Genomes of South American Camelids
PRJNA612032 • SRP265528 • All experiments • All runs

Sample: V_ss18
SAMN14360346 • SRS6753932 • All experiments • All runs
Organism: Vicugna vicugna mensalis

Library:
Name: Vss18
Instrument: HiSeq X Ten
Strategy: WGS
Source: GENOMIC
Selection: RANDOM
Layout: PAIRED

Runs: 1 run, 111.2M spots, 33.4G bases, 11.8Gb

Run	# of Spots	# of Bases	Size	Published
SRR11905265	111,191,160	33.4G	11.8Gb	2020-06-08

Accessing SRA metadata

~~0. NCBI website~~

1. NCBI FTP metadata

https://trace.ncbi.nlm.nih.gov/Traces/index.html?view=mirroring

2. SRA metadata on cloud SQL database (AWS Athena, GCP BigQuery)

```
SELECT acc, mbases, mbytes, avgspotlen, librarylayout, instrument
FROM sra.metadata as s
WHERE consent = 'public' and avgspotlen >= 31
```

SQL Ln 1, Col 1

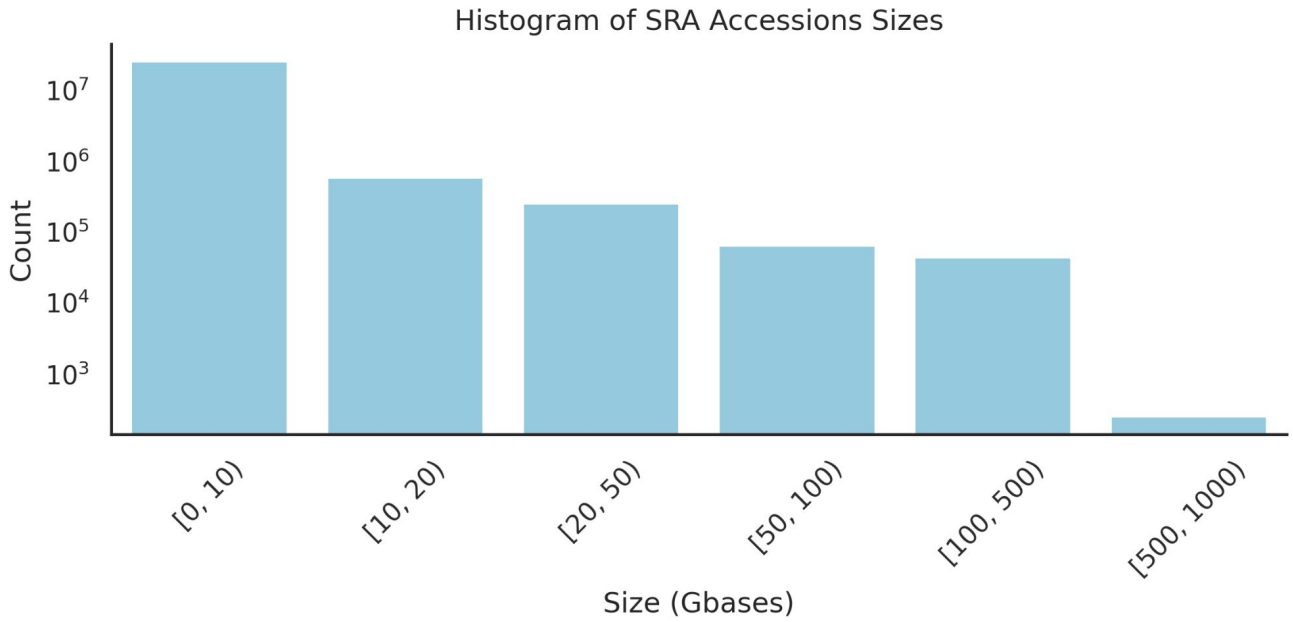
Run | Explain | Cancel | Clear | Create

SRA metadata

tax_analysis	
acc	string
tax_id	int
rank	string
name	string
total_count	bigint
self_count	bigint
ilevel	int
ileft	int
iright	int

metadata	
acc	string
assay_type	string
center_name	string
consent	string
experiment	string
sample_name	string
instrument	string
librarylayout	string
libraryselection	string
librarysource	string
platform	string
sample_acc	string
biosample	string
organism	string
sra_study	string
releasedate	date
bioproject	string
mbytes	int
loaddate	timestamp
avgspotlen	int
mbases	int
insertsize	int
library_name	string
biosamplemodel_sam	array<string>
collection_date_sam	array<string>
geo_loc_name_country_calc	string
geo_loc_name_country_continent_calc	

SRA accessions sizes (2023)



Jonathan Jacobs 🧬🦠🔬🌲
@bioinformer

Wondering: how many individual sequencing reads are in SRA?

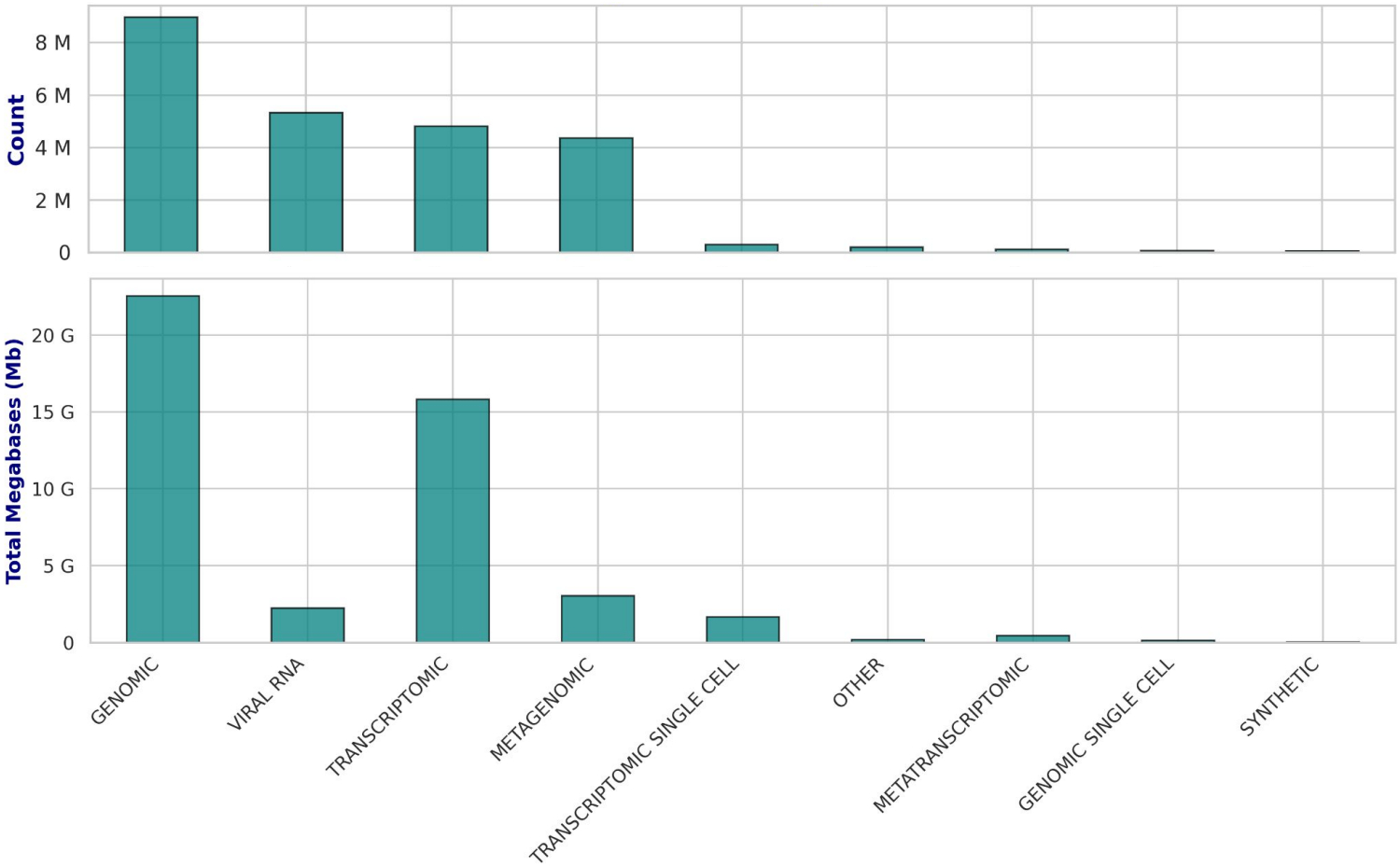
@chris_osulliva is there an estimate of this someplace? 🤔

7:46 PM · Nov 10, 2023

Rayan Chikhi @RayanChikhi · 3m
About 387 trillion as of today.

```
SELECT SUM(
            CASE
                WHEN avgspotlen > 0 THEN (CAST(mbases AS BIGINT) *1000000) / avgspotlen
                ELSE 0
            END
        ) AS total_number_of_reads
FROM metadata;
```

SRA accessions types (2023)



SRA taxonomy analysis

Method | Open Access | Published: 20 September 2021

STAT: a fast, scalable, MinHash-based *k*-mer tool to assess Sequence Read Archive next-generation sequence submissions

Kenneth S. Katz, Oleg Shutov, Richard Lapoint, Michael Kimelman, J. Rodney Brister & Christopher O'Sullivan

Genome Biology **22**, Article number: 270 (2021) | Cite this article

"we have processed more than 27.9 Peta base pairs from runs"

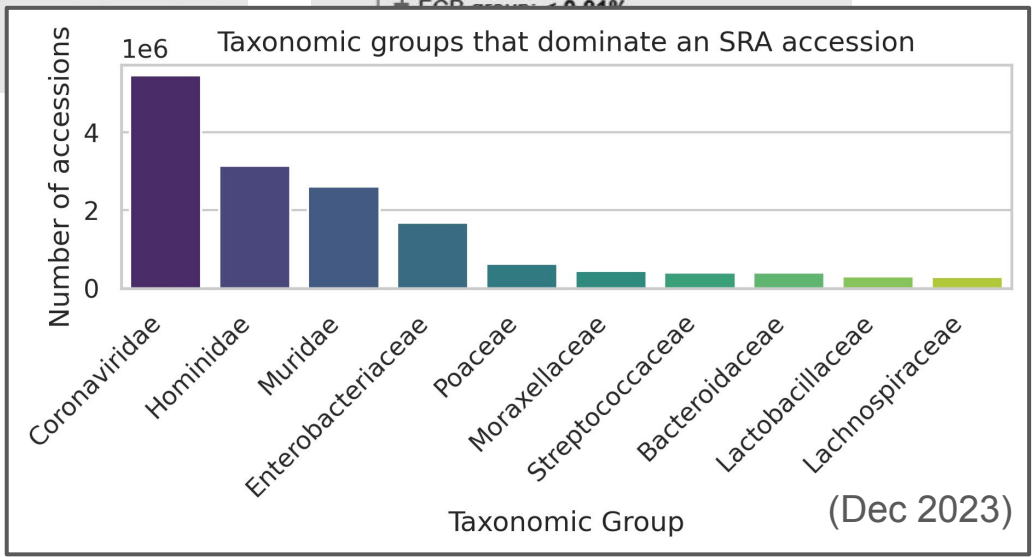
Example STAT output:

Taxonomy Analysis

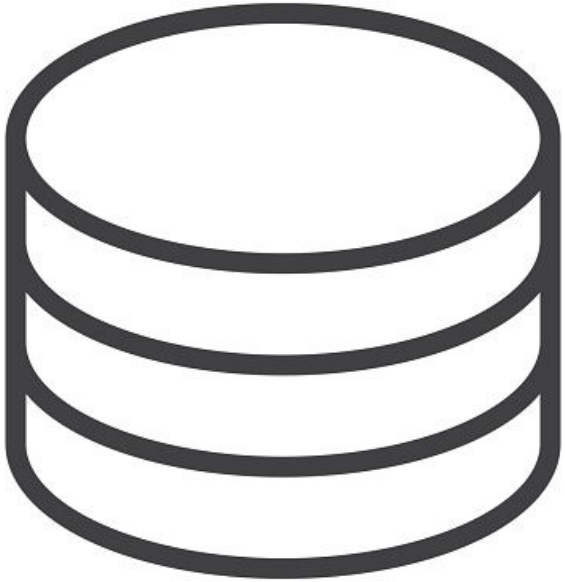
Unidentified reads: 40.04%

Identified reads: 59.96%

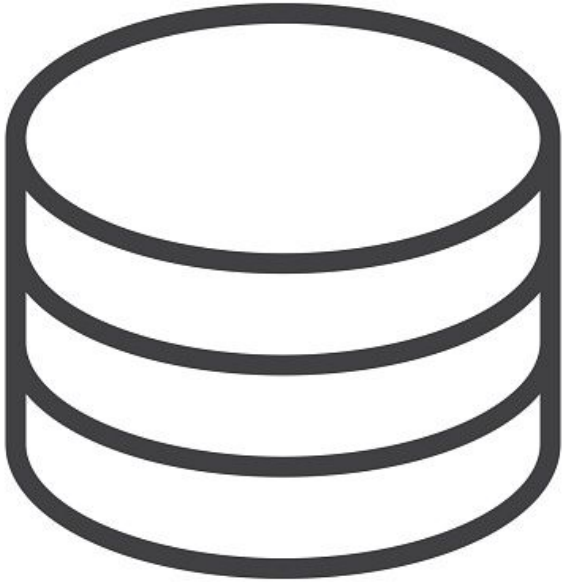
- Viruses: **50.55%**
  - ssRNA viruses: **50.55%**
    - Measles morbillivirus: **50.55%**
  - dsDNA viruses, no RNA stage: **< 0.01%**
  - ssDNA viruses: **< 0.01%**
  - Ortervirales: **< 0.01%**
- cellular organisms: **9.4%**
  - Bacteria: **6.44%**
    - Proteobacteria: **1.76%**
    - Terrabacteria group: **0.48%**



How to analyze the entire SRA?



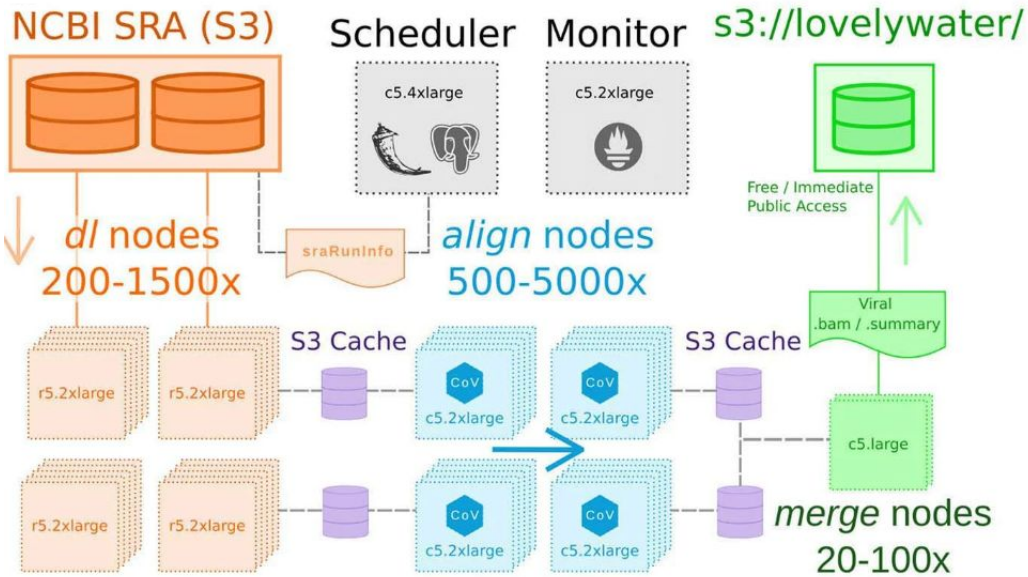
- How much time to download 40 petabytes at 200 MB/sec?



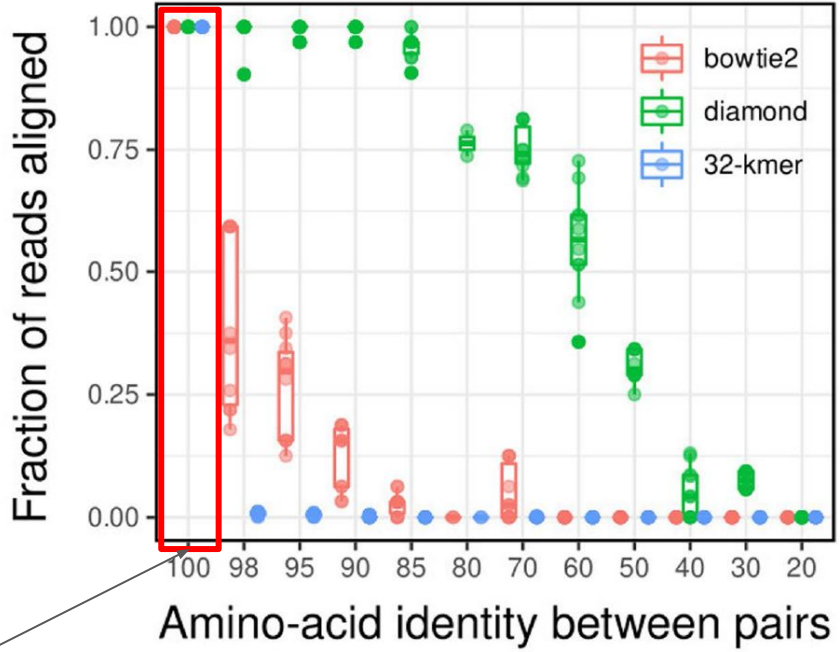
- How much time to download 40 petabytes at 200 MB/sec?

~ 6 years

Serratus infrastructure



Alignment: high **speed** or high **sensitivity**, choose one



Credit: RC Edgar

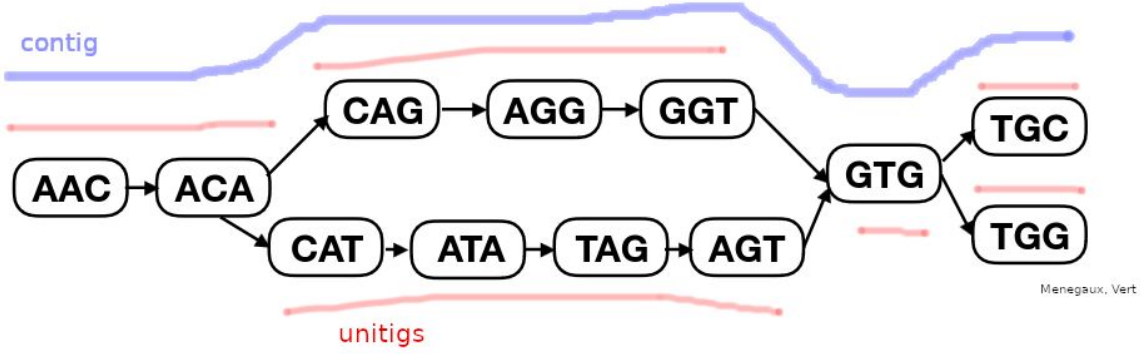
Human reads alignment

SRA-scale alignment

State of the art (ordered by sensitivity/speed):

1. **Sourmash branchwater** (sketches)
   - Metagenomes, long sequences
2. **NCBI Pebblescout** (k-mers, no alignment)
   - Metagenomes, > 42 bp sequences
3. **Bowtie2, STAR** (k-mers, alignment)
   - Serratus1 (all RNAseqs)
   - Recount3 (750k human/mouse RNAseqs)
4. **DIAMOND** (AA-mers)
   - Serratus1.5 (all RNAseqs)
5. ***HMMs?*** (profile)

An apparté on unitigs



Many dedicated construction methods:

- **BCALM** (2014), **BCALM2** (2016), .., **Cuttlefish2** (2022), **GGCAT** (2023)

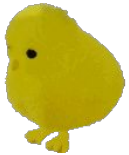
Summary

- Exploring all of Life’s sequencing data
- Tools:
  - SRA metadata
  - SRA data on cloud
  - Alignment algorithms (fast+sensitive)
  - Short read assembly (fast+lowmem+contiguous)
  - Indexing algorithms (fast+sensitive)

Part 4: pangenomics into the wild

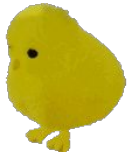
Species:
Gallus gallus

Vocabulary

Kmer: 

A “fun” experiment..

Let’s study why Kmer has this bright yellow color.



1) First, what is the gene responsible for feather color in Gallus gallus?
   (let’s ask Kmer itself)
2) Then we’ll gather sequencing data from chickens

MC1R

Where does this gene appear in the wild?

First we need to get its sequence, or a chunk of it:

https://www.ncbi.nlm.nih.gov/gene/427562

Gathering MC1R genes

I don't have a farm of chicken, but I have big public data

"If you don't have data, download it"

Bonus slide

Searching for MC1R

>MC1R
CTTCCCCATCACCGCCGCTGAGCCCTCTCTGAGCTCCGGATACGCCGGGCAGTGCCGGTGGGGAGGGCGGCCGAGACAGCGGAGTCCCCGCGCTGCTGCCCAGAGGGCTCCCGGGT
GGGGGACCGCTTCCCCATCCTTGTGCCTGGGGTGCAGAGGTGCCCACATCCCCTCTGCCTCGTGACCGCGTGCTGCGGGAGCACTGGTGGGGCTGGTTGGGCGCACGGGGGCTTTG
TAGGTGCTGCAGTTGTGCTCGGGGCCACGGCCTCCAGCCAGGGGGTCCCTGGGGGCTGAGGCCGGGGCCATGTCGATGCTGGCCCCCCTGCGCCTGCTGCGCGAGCCCTGGAACGC
CAGTGAGGGCAACCAGAGCAATGCCACGGCCGGGGCCGGAGGTGCCTGGTGCCAGGGGCTGGACATCCCCAATGAGCTCTTCCTGACGCTGGGGCTGGTGAGCCTGGTGGAGAACC
TGCTGGTGGTGGCCGCCATCCTCAAGAACAGGAATCTGCACTCGCCCACGTACTACTTCATCTGCTGCCTGGCCGTCTCCGACATGCTGGTGAGCGTCAGCAACCTGGCCAAGACGCTC
TTCATGCTGCTGATGGAGCACGGCGTGCTGGTGATCCGCGCCAGCATCGTCCGCCACATGGACAATGTCATCGACATGCTCATCTGCAGCTCCGTCGTGTCCTCCCTCTCCTTCCTGGG
GGTCATCGCCGTGGACCGCTACATCACCATCTTCTATGCGCTGCGCTACCACAGCATCATGACGCTGCAGCGCGCCGTGGTCACCATGGCCAGCGTCTGGCTGGCCAGCACCGTCTCC
AGCACCGTCTTAATCACCTACTACCGCAACAACGCCATCCTGCTCTGCCTCATTGGCTTCTTCCTCTTCATGCTGGTCCTCATGCTGGTGCTCTACATTCACATGTTCGCGCTGGCGTGCC
ACCACGTGCGCAGCATCTCCAGCCAGCAGAAGCAGCCCACCATCTACCGCACCAGCAGCCTGAAGGGAGCCGTCACGCTCACCATCCTGCTGGGAGTCTTCTTCATCTGCTGGGGGCC
CTTCTTCTTCCACCTCATCCTCATCGTCACCTGCCCCACCAACCCCTTCTGCACCTGCTTCTTCAGCTATTTCAACCTCTTCCTCATCCTCATCATCTGCAATTCAGTGGTCGATCCCCTGA
TCTATGCCTTCCGGAGCCAGGAGCTCCGGCGGACGCTGCGGGAGGTGGTGCTGTGCTCCTGGTAGGAGGCGGCACAGACAGGAGGATGGATGGATGGATGGACGGATGGACG
GATGGATGGATGGACAAACAGATGGGTGGATGGACAGATGGGTGGATGGACAGACAGACGCACCGCGGGGTGTCCCCTGGGTGCCCCAGTGCAGCTGGGGTTGGGCTGCCTGGCCT
CGCGCTCCCAAATAAAGGCTCTTTGCAGTGA

Three routes:

1) https://pebblescout.ncbi.nlm.nih.gov/

2) SRA metadata query

3) SRA taxonomy query

Pebblescout query

Finds 2000+ hits.

Some of doubtful quality. “chicken” stops appearing in titles after hit number 1750.

Metagenomic Summary

#	QueryID	SubjectID	RawScore	%coverage	PBSscore	BioSample	Title
1	MC1R	ERR2241657	149.21	100.00	100.00	SAMEA104467160	As part of the EFFORT project, we sampled feces from pig and poultry livestock in nine European countries (BE, BG, DK, FR, ES, GE, NL, PL, SP). More than 9000 animals were sampled, across 181 pig and 178 poultry herds to generate herd-level composite fecal samples. Using shotgun metagenomics, we have quantified and characterized the antimicrobial resistance gene pools (resistomes) in Europe's two most intensively raised livestock species.
2	MC1R	ERR3340873	149.21	100.00	100.00	SAMEA5662822	Trial B 2019
3	MC1R	ERR3340883	149.21	100.00	100.00	SAMEA5662832	Trial B 2019
4	MC1R	ERR4832135	149.21	100.00	100.00	SAMEA7556319	CP562
5	MC1R	ERR4832949	149.21	100.00	100.00	SAMEA7556328	CP562
6	MC1R	ERR4833354	149.21	100.00	100.00	SAMEA7556341	CP562
7	MC1R	SRR12730716	149.21	100.00	100.00	SAMN16282411	Chicken Cecal Metagenome Sequencing
8	MC1R	SRR12730718	149.21	100.00	100.00	SAMN16282445	Chicken Cecal Metagenome Sequencing
9	MC1R	SRR12730726	149.21	100.00	100.00	SAMN16282437	Chicken Cecal Metagenome Sequencing
10	MC1R	SRR12730731	149.21	100.00	100.00	SAMN16282433	Chicken Cecal Metagenome Sequencing

SRA metadata query

https://www.ncbi.nlm.nih.gov/sra/?term=%22yellow+chicken%22

SRA Run Selector

Select	Runs	Bytes	Bases	Download
Total	324	1.23 Tb	3.43 T	Metadata or Accession List

Overlap with Pebblescout: 0 😱

https://www.ncbi.nlm.nih.gov/sra/SRX4478521[accn]

SRX4478521: DNA-seq of Gallus gallus: Wuhua yellow chicken
1 ILLUMINA (HiSeq X Ten) run: 38M spots, 11G bases, 3.9Gb downloads

SRA Athena STAT query

```
SELECT *
FROM "sra"."tax_analysis"
WHERE name = 'Gallus gallus' AND total_count > 100
```

In retrospect this is probably way too low, many false hits

Results (317,949) 

Contains 72% of the Pebblescout hits

Contains 83% of the “yellow chicken” metadata hits

Getting data from the SRA

TL;DR: state of the art is **prefetch + fasterq-dump**

prefetch: downloads .sra file locally

fasterq-dump: transforms .sra to .fastq or .fasta

Example:

```
prefetch [accession] && fasterq-dump [accession].sra
```

Getting data from the SRA, easily

```
aws s3 cp s3://sra-pub-run-odp/sra/{accession}/{accession} \
         {accession}.sra \
         --no-sign-request

fasterq-dump --fasta-unsorted --stdout {accession}.sra
```

NIH NCBI Sequence Read Archive (SRA) on AWS

bam | cram | fastq | genetic | genomic | life sciences | STRIDES | transcriptomics | whole exome sequencing | whole genome sequencing

Description

The Sequence Read Archive (SRA), produced by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine (NLM) at the National Institutes of Health (NIH), stores raw DNA sequencing data and alignment information from high-throughput sequencing platforms. The SRA provides open access to these biological sequence data to support the research community's efforts to enhance reproducibility and make new discoveries by comparing data sets. Buckets in this registry contain public SRA data in the original (user submitted) format from select high value and newly-released studies as well as all public-access SRA formatted ETL+BQS data. Also included is all SRA metadata that can be leveraged for attribute-based data discovery.

Update Frequency

Daily

License

Resources on AWS

Description
.bam, .cram, and .fastq files in a public S3 bucket. This is the first of two S3 buckets for source submissions from sequencing methodologies such as PacBio, Oxford Nanopore Technologies, and 10X Genomics.

Resource type
S3 Bucket

Amazon Resource Name (ARN)
`arn:aws:s3:::sra-pub-src-1`

AWS Region
`us-east-1`

AWS CLI Access (No AWS account required)
`aws s3 ls --no-sign-request s3://sra-pub-src-1/`

Big data genomics:)

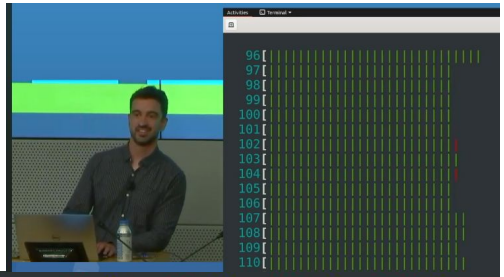
```
$ cat download_and_map_accession.sh

set -e
accession=$1

aws s3 cp s3://sra-pub-run-odp/sra/$accession/$accession \
          $accession.sra --no-sign-request

minimap2 -t20 -x sr mc1r.fa <(fasterq-dump --fasta-unsorted $accession.sra) \
         -o mapping/$accession.minimap2_output

rm -f $accession.sra
```

Parallelize processing:

```
cat accessions.txt | parallel -j 10 "./download_and_map_accession.sh {}"
```

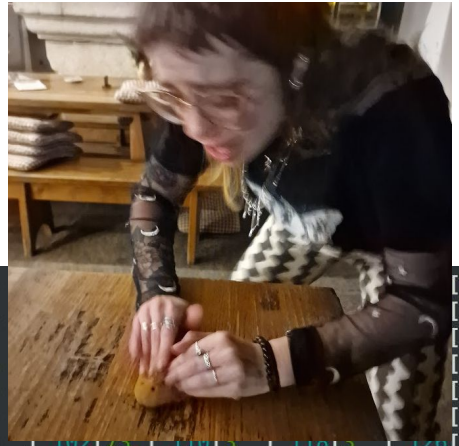
Running.. (htop)

On c6a.32xlarge (128 threads, 256 GB mem):



```
  0[89.]   8[92.]  16[95.]  24[94.]  32[83.]  40[91.]  48[95.] 56[90.]  64[92.]  72[91.]  80[92.]  88[94.]  96[89.] 104[92.] 112[92.] 120[90.]
  1[91.]   9[94.]  17[92.]  25[94.]  33[89.]  41[91.]  49[93.] 57[91.]  65[96.]  73[92.]  81[93.]  89[94.]  97[88.] 105[92.] 113[92.] 121[92.]
  2[90.]  10[91.]  18[92.]  26[91.]  34[89.]  42[94.]  50[92.] 58[91.]  66[94.]  74[91.]  82[94.]  90[91.]  98[91.] 106[93.] 114[91.] 122[91.]
  3[91.]  11[87.]  19[90.]  27[88.]  35[90.]  43[92.]  51[92.] 59[94.]  67[91.]  75[94.]  83[92.]  91[91.]  99[88.] 107[94.] 115[93.] 123[95.]
  4[90.]  12[91.]  20[92.]  28[92.]  36[91.]  44[92.]  52[90.] 60[93.]  68[91.]  76[94.]  84[93.]  92[91.] 100[87.] 108[92.] 116[92.] 124[93.]
  5[90.]  13[92.]  21[96.]  29[92.]  37[87.]  45[91.]  53[91.] 61[90.]  69[90.]  77[92.]  85[92.]  93[91.] 101[90.] 109[92.] 117[88.] 125[94.]
  6[91.]  14[94.]  22[92.]  30[89.]  38[92.]  46[94.]  54[90.] 62[94.]  70[91.]  78[91.]  86[91.]  94[92.] 102[92.] 110[92.] 118[91.] 126[93.]
  7[92.]  15[92.]  23[89.]  31[90.]  39[90.]  47[93.]  55[91.] 63[93.]  71[90.]  79[90.]  87[94.]  95[92.] 103[88.] 111[88.] 119[92.] 127[92.]
Mem[||||||||||||||                                                   ] Tasks: 92, 303 thr          ; 128 running
Swp[                                                                 ] Load average: 21.41 30.84 45.33
                                                                       Uptime: 01:39:13

 Main   I/O
   PID USER       PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
 70507 ec2-user    20      1672M  235M  2544    889.2  0.1  1:14.58 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
 71216 ec2-user    20   0  1906M  274M  2524 S 882.6  0.1  0:43.80 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
 70747 ec2-user    20      1866M  273M  2264    856.8  0.1  0:56.76 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
 70544 ec2-user    20      1842M  242M  2276    856.1  0.1  1:07.39 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
 70904 ec2-user    20      1736M  266M  2264    838.2  0.1  0:51.07 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
 71073 ec2-user    20      1672M  242M  2524    833.0  0.1  0:46.12 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
 73054 ec2-user    20      1842M  249M  2312    815.8  0.1  0:24.28 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
 70627 ec2-user    20      1906M  267M  2492    793.3  0.1  0:58.48 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
 70656 ec2-user    20      1906M  274M  2544    781.4  0.1  0:58.39 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
 71433 ec2-user    20      1736M  266M  2284    773.4  0.1  0:40.49 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
 71522 ec2-user    20      1900M  826M  4876    303.0  0.3  0:17.61 fasterq-dump --fasta-unsorted --stdout SRR10315070.sra
 70706 ec2-user    20      1900M  829M  4880    300.4  0.3  0:23.55 fasterq-dump --fasta-unsorted --stdout SRR10058581.sra
 70536 ec2-user    20      1900M  829M  4872    298.4  0.3  0:28.64 fasterq-dump --fasta-unsorted --stdout SRR10058584.sra
 71305 ec2-user    20      1900M  826M  4876    298.4  0.3  0:18.30 fasterq-dump --fasta-unsorted --stdout SRR10315066.sra
 70685 ec2-user    20      1900M  829M  4880    296.4  0.3  0:23.57 fasterq-dump --fasta-unsorted --stdout SRR10058582.sra
 70973 ec2-user    20      1900M  827M  4888    296.4  0.3  0:20.93 fasterq-dump --fasta-unsorted --stdout SRR10315069.sra
```

But then a bit later..



```
  0[25 ]   8[6  ]  16[   ]  24[1  ]  32[6. ]  40[2  ]  48[5. ] 56[   ]  64[6  ]  72[1  ]  80[6  ]  88[3                                    [2  ]
  1[4  ]   9[0  ]  17[3  ]  25[3  ]  33[41.]  41[48.]  49[3  ] 57[   ]  65[1  ]  73[1  ]  81[   ]  89[1                                    [   ]
  2[1  ]  10[   ]  18[   ]  26[1  ]  34[1  ]  42[9  ]  50[3  ] 58[   ]  66[6  ]  74[1  ]  82[2  ]  90[7                                    [   ]
  3[1  ]  11[3  ]  19[6  ]  27[1  ]  35[5  ]  43[9  ]  51[7  ] 59[1  ]  67[6  ]  75[3  ]  83[   ]  91[6                                    [   ]
  4[1  ]  12[8  ]  20[1  ]  28[3  ]  36[1  ]  44[   ]  52[   ] 60[   ]  68[6  ]  76[19 ]  84[   ]  92[6                                    [   ]
  5[7  ]  13[   ]  21[2  ]  29[3  ]  37[12 ]  45[2  ]  53[3  ] 61[   ]  69[1  ]  77[1  ]  85[   ]  93[3                                    [   ]
  6[3  ]  14[7  ]  22[11 ]  30[   ]  38[   ]  46[   ]  54[6  ] 62[35.]  70[   ]  78[1  ]  86[   ]  94[6  ] 102[75.] 110[3  ] 118[3  ] 126[   ]
  7[1  ]  15[7  ]  23[1  ]  31[6  ]  39[0  ]  47[3  ]  55[3  ] 63[2  ]  71[   ]  79[3  ]  87[3  ]  95[7  ] 103[58.] 111[3  ] 119[1  ] 127[9  ]
Mem[||||||||||||||||||||||||||||||||||||||||||||                      Tasks: 66, 185 thr          ; 15 running
Swp[                                                                  Load average: 24.14 21.70 29.99
                                                                      Uptime: 02:23:01

Main   I/O
  PID USER      PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
802888 ec2-user  20   0 1736M  259M  2508 S 944.7  0.1  2:46.23 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
802917 ec2-user  20     1900M  828M  4868   305.6  0.3  0:56.21 fasterq-dump --fasta-unsorted --stdout SRR6490215.sra
802914 ec2-user  20     1736M  259M  2508    73.6  0.1  0:11.68 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
802915 ec2-user  20     1736M  259M  2508 R  59.0  0.1  0:10.75 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
802916 ec2-user  20     1736M  259M  2508 R  57.0  0.1  0:11.10 minimap2 -t 20 -a -x sr mc1r.fa /dev/fd/63
802919 ec2-user  20     1900M  828M  4868    51.0  0.3  0:08.86 fasterq-dump --fasta-unsorted --stdout SRR6490215.sra
802924 ec2-user  20     1900M  828M  4868    51.0  0.3  0:08.85 fasterq-dump --fasta-unsorted --stdout SRR6490215.sra
802922 ec2-user  20     1900M  828M  4868 R  50.4  0.3  0:08.99 fasterq-dump --fasta-unsorted --stdout SRR6490215.sra
802889 ec2-user  20      5340  1312  1200 R  49.7       0:08.56 samtools view -hF4 -
802923 ec2-user  20     1900M  828M  4868 R  49.7  0.3  0:08.57 fasterq-dump --fasta-unsorted --stdout SRR6490215.sra
802920 ec2-user  20     1900M  828M  4868    45.1  0.3  0:08.77 fasterq-dump --fasta-unsorted --stdout SRR6490215.sra
802921 ec2-user  20     1900M  828M  4868 R  38.5  0.3  0:08.55 fasterq-dump --fasta-unsorted --stdout SRR6490215.sra
802918 ec2-user  20     1900M  828M  4868 R  14.6  0.3  0:02.51 fasterq-dump --fasta-unsorted --stdout SRR6490215.sra
686759 ec2-user  20     1338M  370M 16052     8.0  0.1  1:55.79 /usr/bin/python3 -s /usr/bin/aws s3 cp s3://sra-pub-run-odp/sra/SRR4897316/SRR4
```

Remember Part 2..

Knowledge of scaling limits

In order of difficulty:

1. **Estimate** how long an analysis will take
   - Look at performance table in tool paper
   - Try on smaller data and extrapolate
2. Reasons **why** some analyses are slower than expected
   - Limited number of CPUs
   - Limited RAM
   - Slow disk (HDD < Cluster network drives < SSD < NVMe)
3. **How** to reduce that time
   - Most analyses go fast enough on a big cloud/cluster and the right tools

What’s happening? see `iotop`

```
Total DISK READ:         0.00 B/s | Total DISK WRITE:       121.48 M/s
Current DISK READ:       0.00 B/s | Current DISK WRITE:     114.90 M/s
    TID  PRIO  USER     DISK READ  DISK WRITE  SWAPIN     IO>    COMMAND
 618212 be/4 ec2-user    0.00 B/s   12.02 M/s  ?unavailable?  python3 -s /usr/bi
 618393 be/4 ec2-user    0.00 B/s   11.79 M/s  ?unavailable?  python3 -s /usr/bi
 687135 be/4 ec2-user    0.00 B/s   12.48 M/s  ?unavailable?  python3 -s /usr/bi
 688754 be/4 ec2-user    0.00 B/s   12.59 M/s  ?unavailable?  python3 -s /usr/bi
 778885 be/4 ec2-user    0.00 B/s   11.81 M/s  ?unavailable?  python3 -s /usr/bi
 791289 be/4 ec2-user    0.00 B/s   12.40 M/s  ?unavailable?  python3 -s /usr/bi
 798802 be/4 ec2-user    0.00 B/s   11.40 M/s  ?unavailable?  python3 -s /usr/bi
 802884 be/4 ec2-user    0.00 B/s   12.67 M/s  ?unavailable?  python3 -s /usr/bi
 807479 be/4 ec2-user    0.00 B/s   11.98 M/s  ?unavailable?  python3 -s /usr/bi
 809481 be/4 ec2-user    0.00 B/s   12.34 M/s  ?unavailable?  python3 -s /usr/bi
      1 be/4 root        0.00 B/s    0.00 B/s  ?unavailable?  systemd --switched
```

Disk speed limit around 125 MB/sec

Workaround

-> Use more machines, smaller ones.

```
# setup 10 machines, make a list of different accessions on each, then run:
cat ips.txt | parallel -j 10 \
              ssh ec2-user@{} ./run_previous_parallel_cmd.sh
```

Didn’t do that here 😬 (lazy instructor)

Analyzed so far

3 terabases from “yellow chicken” SRA accessions downloaded and mapped to MC1R

```
-rw-r--r--. 1 ec2-user ec2-user 154700 Jan 11 18:22 SRR11521907.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 174639 Jan 11 18:24 SRR11521908.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 150667 Jan 11 18:25 SRR11521909.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 135759 Jan 11 18:25 SRR11521910.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 194411 Jan 11 18:23 SRR11521911.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 149717 Jan 11 18:24 SRR11521912.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 149674 Jan 11 18:25 SRR11521913.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 204873 Jan 11 18:26 SRR11521914.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 180067 Jan 11 18:26 SRR11521915.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 139216 Jan 11 18:26 SRR11521916.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 113860 Jan 11 18:26 SRR11521917.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user 157065 Jan 11 18:27 SRR11521918.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user   6240 Jan 11 18:25 SRR11678145.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user  11665 Jan 11 18:25 SRR11678146.minimap2_output
-rw-r--r--. 1 ec2-user ec2-user  15025 Jan 11 18:25 SRR11678147.minimap2_output
```

Took around 1.5 hours, 6$/hour

```
1:36:09elapsed 2026%CPU (0avgtext+0avgdata 1182952maxresident)k
```

Making mini-assemblies of MC1R

$ sgpt --model gpt-4 "setup a variant calling script that takes a SAM file as input, mapped to a reference, and outputs a FASTA consensus of the reference"

```
samtools view -S -b $SAM_FILE > $BAM_FILE

samtools sort $BAM_FILE -o $SORTED_BAM_FILE

samtools index $SORTED_BAM_FILE

bcftools mpileup -O b -o $BCF_FILE -f reference.fasta $SORTED_BAM_FILE

bcftools call --ploidy 1 -m -v -o $VCF_FILE $BCF_FILE

cat reference.fasta | bcftools consensus $VCF_FILE > $FASTA_FILE
```

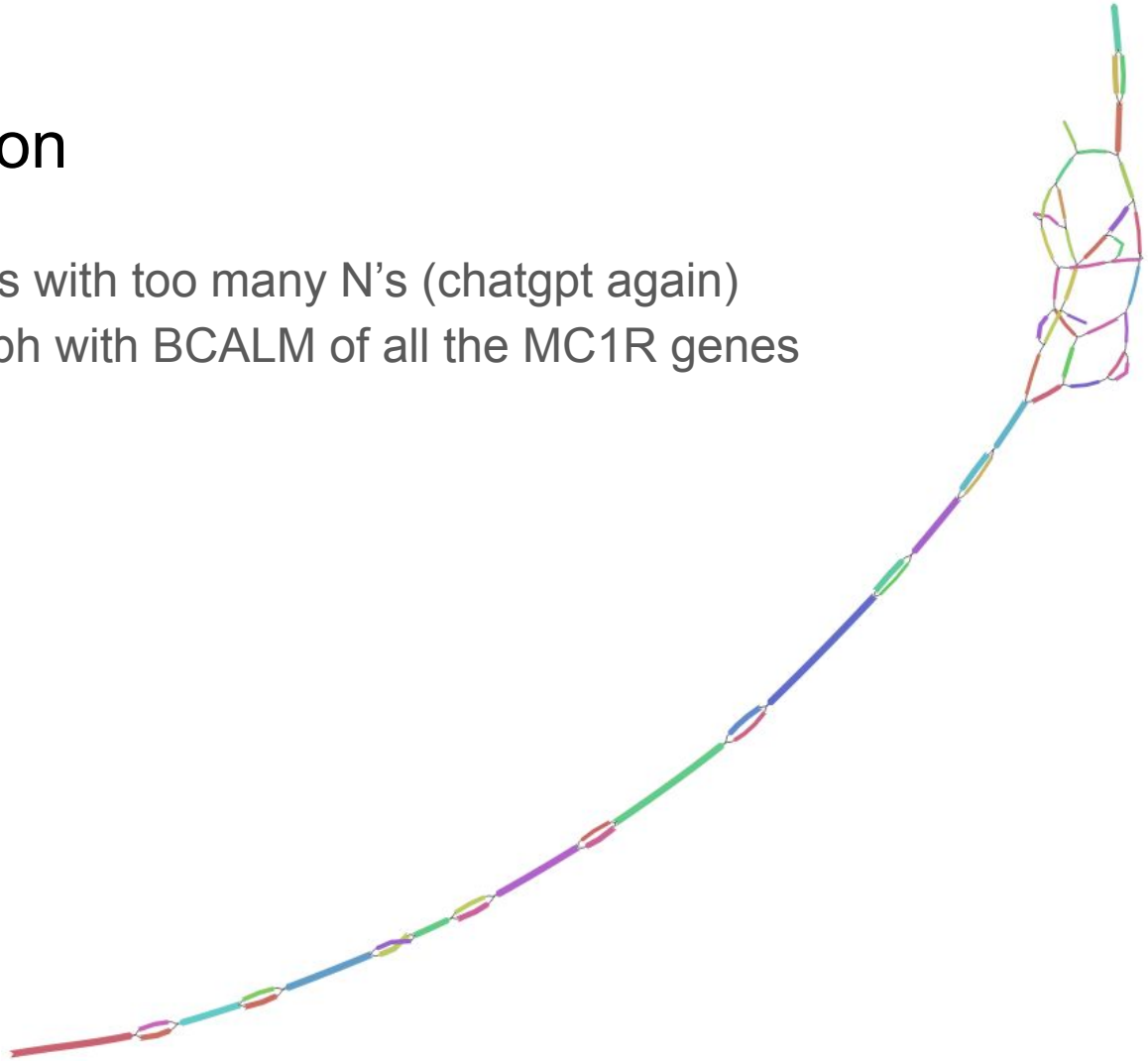
Making mini-assemblies of MC1R

$ sgpt --model gpt-4 "can bcftools consensus mask regions where it doesn't have confidence in the base call? how to get mask.bed?"

```
samtools depth -a alignment.bam | awk '$3 < 5' | awk '{print $1"\t"$2"\t"$2+1}' > mask.bed
```

Analysis of variation

- Filtered consensuses with too many N’s (chatgpt again)
- Constructed DB graph with BCALM of all the MC1R genes



So, where’s the “yellow chicken” allele?

- Remember the Wuhua yellow chicken accession?
- BLASTed the consensus gene against the pangenome graph

Several hits to low-frequency SNPs, could be any/some of those..


We need more data

Need color metadata

Back to SRA metadata

"SRR13606998","WGS","ZHEJIANG ACADEMY OF AGRICULTURAL SCIENCES","public","SRX10001135","ATC
LLUMINA","SRS8170736","SAMN17734173","Gallus gallus","SRP304191","2022-02-01","PRJNA698651"
al]",,,,,,,"[fastq, run.zq, sra]","[gs, ncbi, s3]","[gs.US, ncbi.public, s3.us-east-1]","[{
es, v=3591984874}, {k=run_file_create_date, v=2021-02-01T21:25:00.000Z}, {k=breed_sam, v=Ti
umber_sam_s_dpl45, v=Tibetan chicken ex situ in vivo conservation 1}, {k=tissue_sam_ss_dpl1
y_search, v=698651}, {k=primary_search, v=ATC1}, {k=primary_search, v=ATC1_1.fq.gz}, {k=pri
7734173}, {k=primary_search, v=SRP304191}, {k=primary_search, v=SRR13606998}, {k=primary_se
{k=primary_search, v=bp0}]","{""sex_calc"": ""female"", ""bases"": 11206718290, ""bytes"":
25:00.000Z"", ""breed_sam"": [""Tibetan chicken""], ""dev_stage_sam"": [""adult""], ""sampl
vo conservation 1"", ""tissue_sam_ss_dpl145"": [""blood""], ""primary_search"": ""17734173"

Breed information given for *some* of the chicken. How to extract?

Python script calling chatGPT

```
from openai import OpenAI

def determine_chicken_color(line):

    query = f"Determine the color of the chicken based on the
following data: {line}. You may reply only: yellow, orange, or
other. [..]. Do not guess."

    response = client.completions.create(

        model="gpt-3.5-turbo-instruct",

        prompt=query, max_tokens=50)

    return response.choices[0].text.strip()
```

Result of chicken coloring

```
$ tail chicken_color.txt
SRR2917304,other
SRR8490109,other
SRR25338401,Other
SRR13193600,other
ERR5036744,yellow
SRR24605477,other
SRR12228200,other
ERR4351384,other
ERR3505973,other
```

Those chatGPT color predictions..

```
$ tail chicken_color.txt

[..]

ERR5036744,yellow

[..]

$ grep ERR5036744 *.csv

[..], {k=common_name_sam, v=chicken}, [..],
{k=insdc_center_alias_sam, v=QUEEN MARY UNIVERSITY OF LONDON}, [..]
```

No breed information! ChatGPT hallucinated that yellow color.

This was a failed analysis

- This is OK
- Not all analyses are successes
- Move on to the next one
- We learned along the way, right?

Outro

What we've seen today

- Some elements of big data bioinformatics
- Toolbox for Big Data
  - Cloud, parallelism, storage handling, knowledge of limitations
- SRA primer
  - Mining metadata
  - Mining sequences
  - Serratus
- Chicken Pop-Pan
  - Mining 1 gene for 1 species across the SRA
  - Using metadata search and taxonomy search


bigger data
NCC-1701-C
big data

WE'RE GONNA NEED

A BIGGER INSTANCE TYPE

❤️ This talk was first dedicated to the

Workshop on Genomics 2023 in Cesky Krumlov:

Guy, Janina, Milos, Kartik, Alena, Madee, Joan, Mercè, and Josie

.. and secondly dedicated to JOBIM'23 organizers..

Thirdly to **CGSI**'23

Fourthly to Workshop on Genomics 2024!

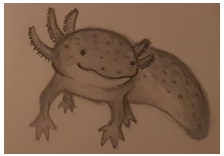
Side note: all microbes can fit onto an SD, carrier pigeons are faster than Internet

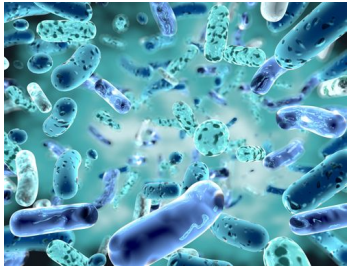
Testing this still on our todo list

K. Brinda

Sequence Bioinformatics

Institut Pasteur
Computational Biology Department

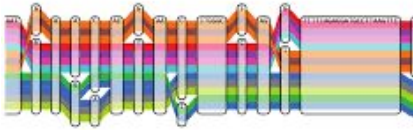




Genomes & metagenomes assembly

Algorithms and data structures on k-mers

Sequence search in very large datasets



Pangenomics

cnrs PR[AI]RIE PaRis Artificial Intelligence Research InstitutE INSTITUT PASTEUR erc

Thank you for your attention!