

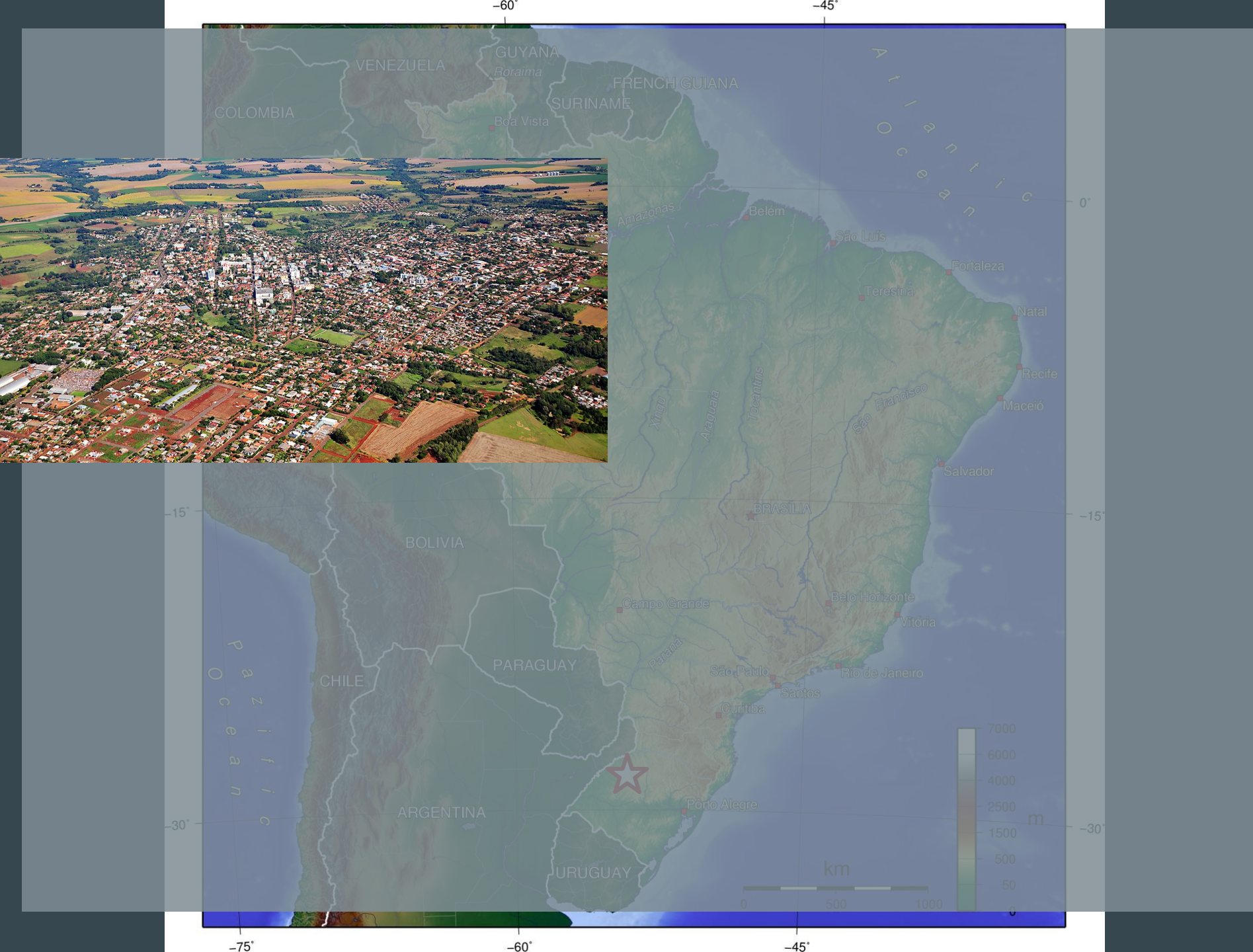
# Decentralized indices for genomic data



Luiz Irber  
CS PhD @ DIB Lab  
UC Davis  
Qualifying Exam - 2019/04/17









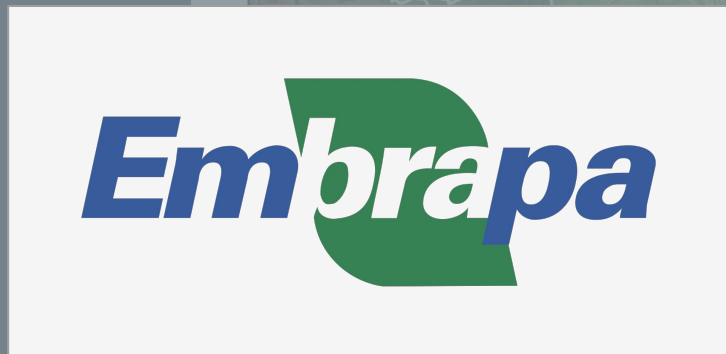


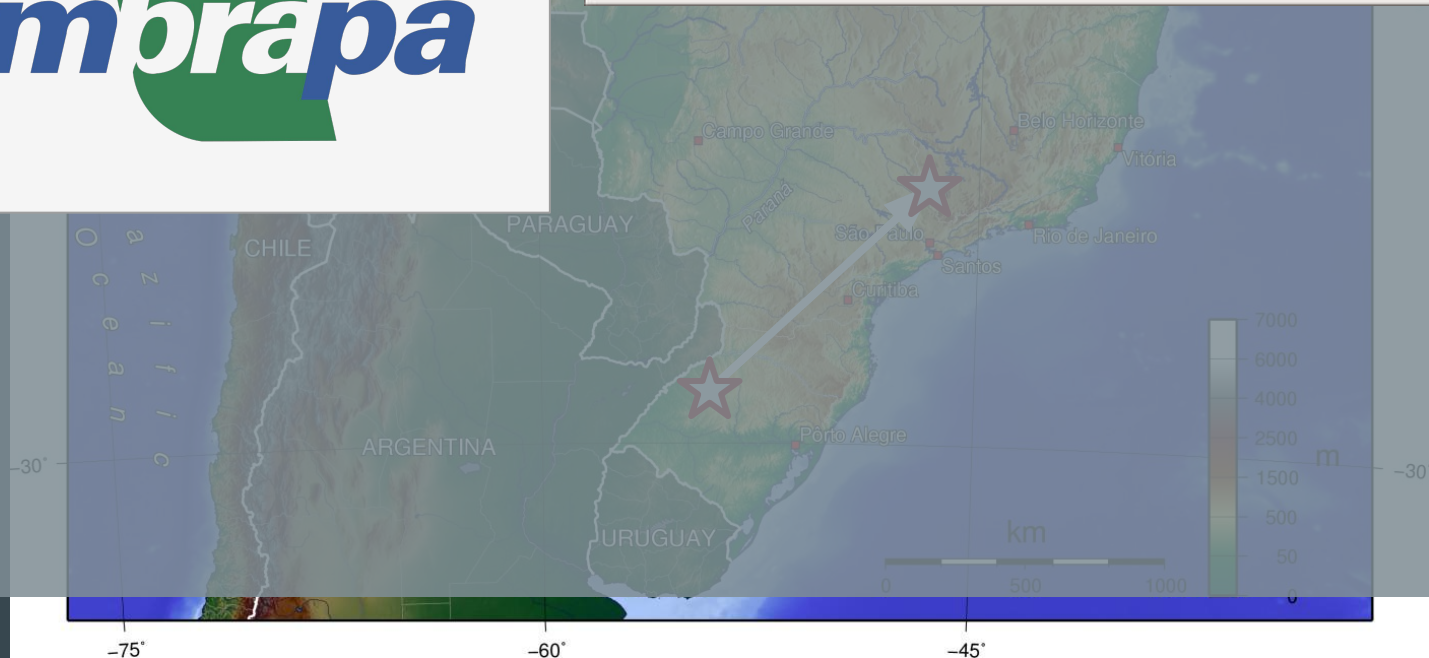
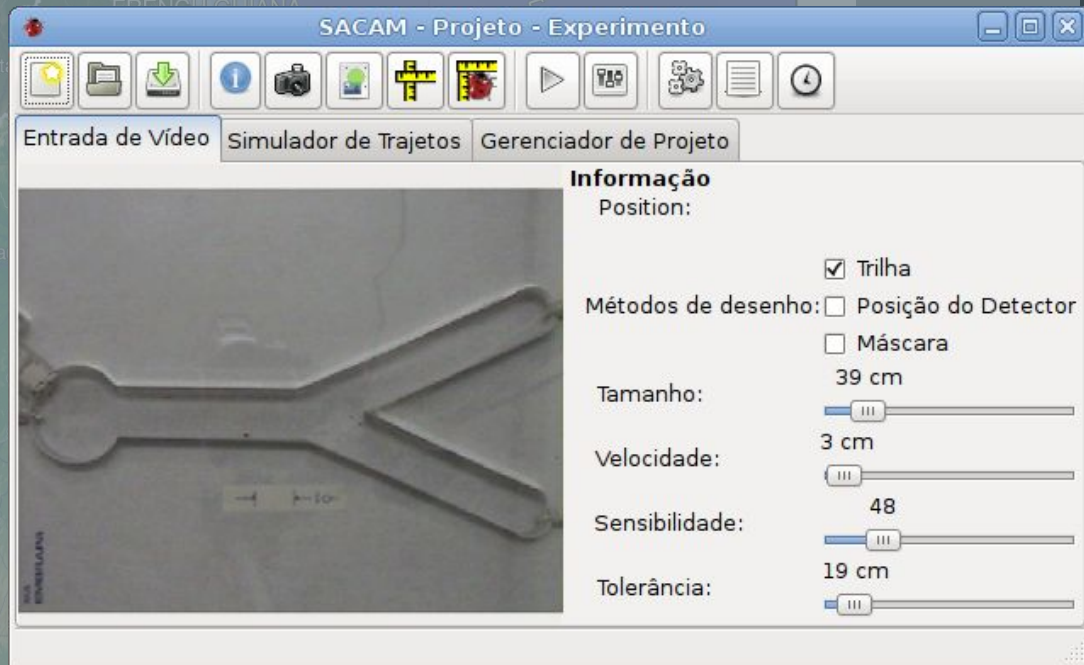












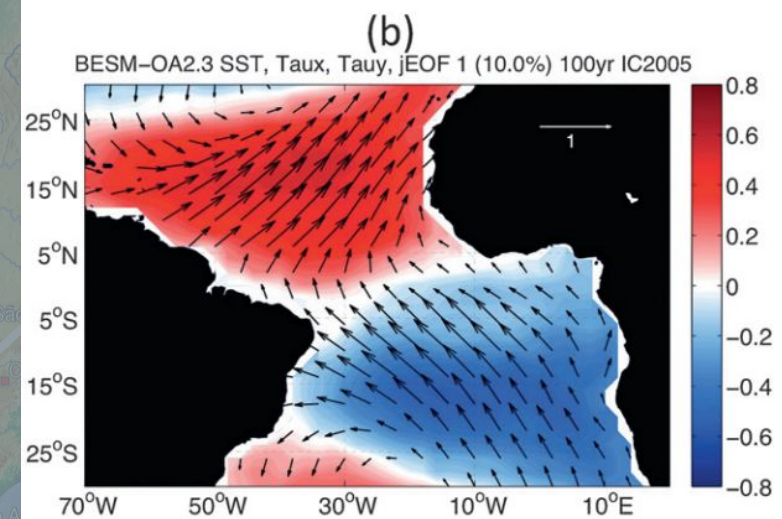
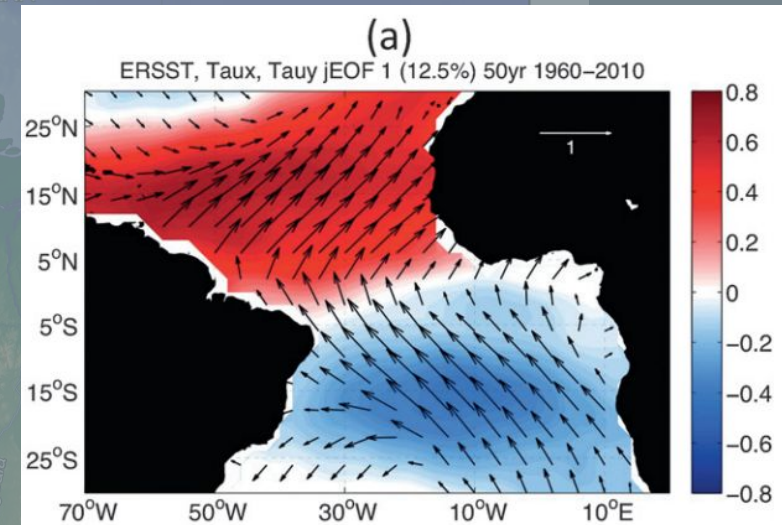
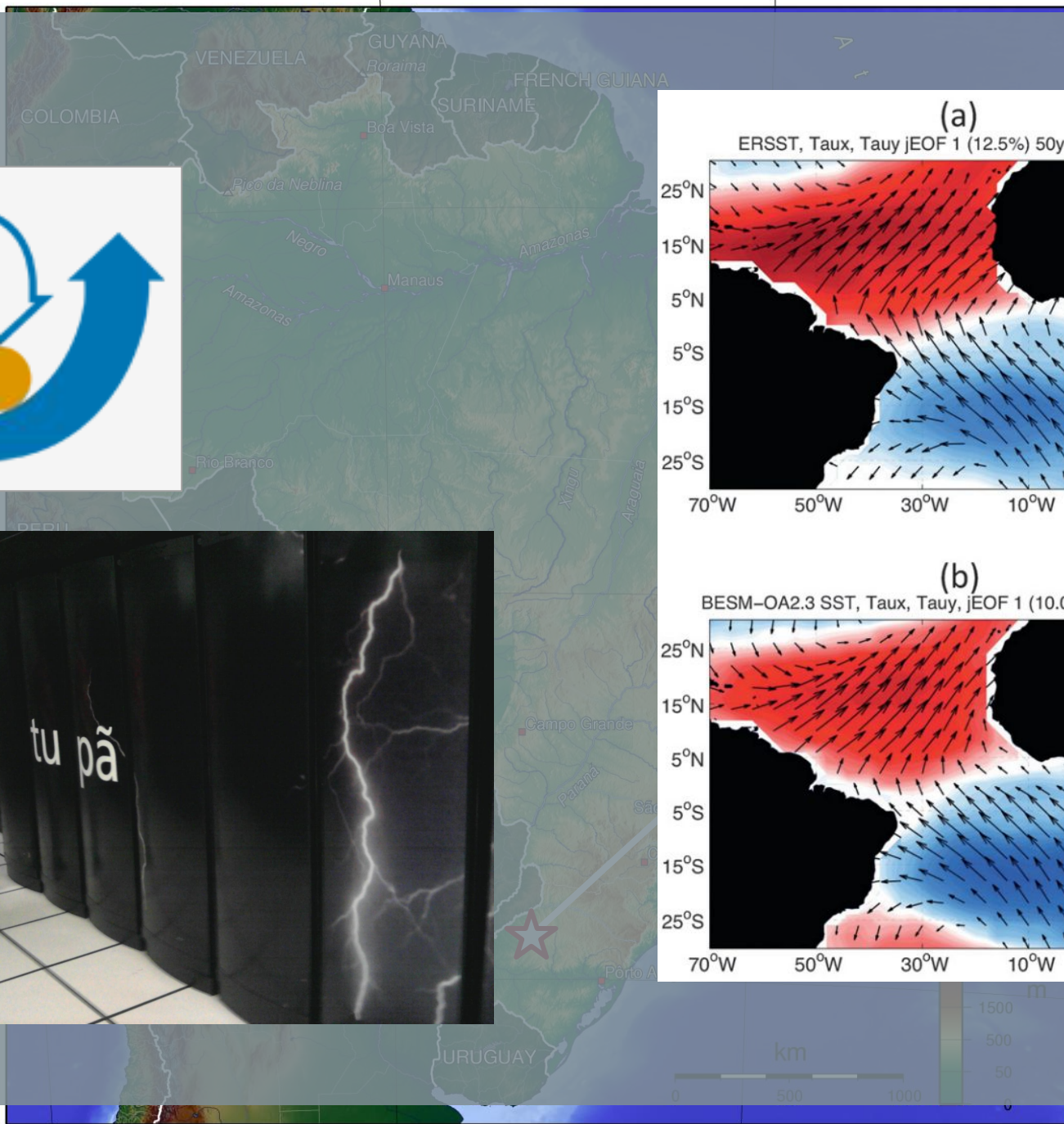














# Before grad school

- Developing software for other scientists to use
- Working together with domain experts to translate their knowledge into software
- Open source was a great influence...

# Before grad school

- Developing software
- Working together
- Open source was

## Applying for grad school

luiz.irber x

**Luiz Irber** <luiz.irber@gmail.com>

Jan 30, 2013, 6:09 PM



to ctb ▾

Hello Dr. Brown,

my name is Luiz Irber, I'm a computer engineer working with climate models at the Brazilian National Institute for Space Research.

Well, that isn't quite biological, is it?

I developed tools to help research and make it more efficient. Before I joined the group all the steps were done by hand and just a handful of people knew how to do all of them (prepare inputs, compile the model, run it, archive the results and analyze them). So I wrote a library that helped document and automate the process, and also served as a runtime environment. With this library/runtime working I could set up a CI server and test the model. I also set up the group repository and gave Mercurial and Python workshops.

So, why am I sending this email? I follow your blog and always tried to reproduce some ideas here in our workplace. Now I'm planning to go to grad school and was thinking about which problems I would like to solve, and in which areas. Making science easier and more efficient is something that I like to do. While searching for nice places to study I found your lab page and read about the "Software engineering methodologies" line of research, and I think that it is a pretty good fit.

But I don't know if you take foreign students, or if you are interested.

My C.V. is in [1], but I'm also sending it attached.

knowledge into software



# Before grad school

- Developing software
- Working together
- Open source was

## Applying for grad school

luiz.irber x

**Luiz Irber** <luiz.irber@gmail.com>

Jan 30, 2013, 6:09 PM



to ctb

Hello Dr. Brown,

my name is Luiz Irber, I'm a computer engineer working with climate models at the Brazilian National Institute for Space Research.

Well, that isn't quite biological, is it?

I developed tools to help research and make it more efficient. Before I joined the group all the steps were done by hand and just a handful of people knew how to do all of them (prepare inputs, compile the model, run it, archive the results and analyze them). So I wrote a library that helped document and automate the process, and also served as a runtime environment. With this library/runtime working I could set up a CI server and test the model. I also set up the group repository and gave Mercurial and Python workshops.

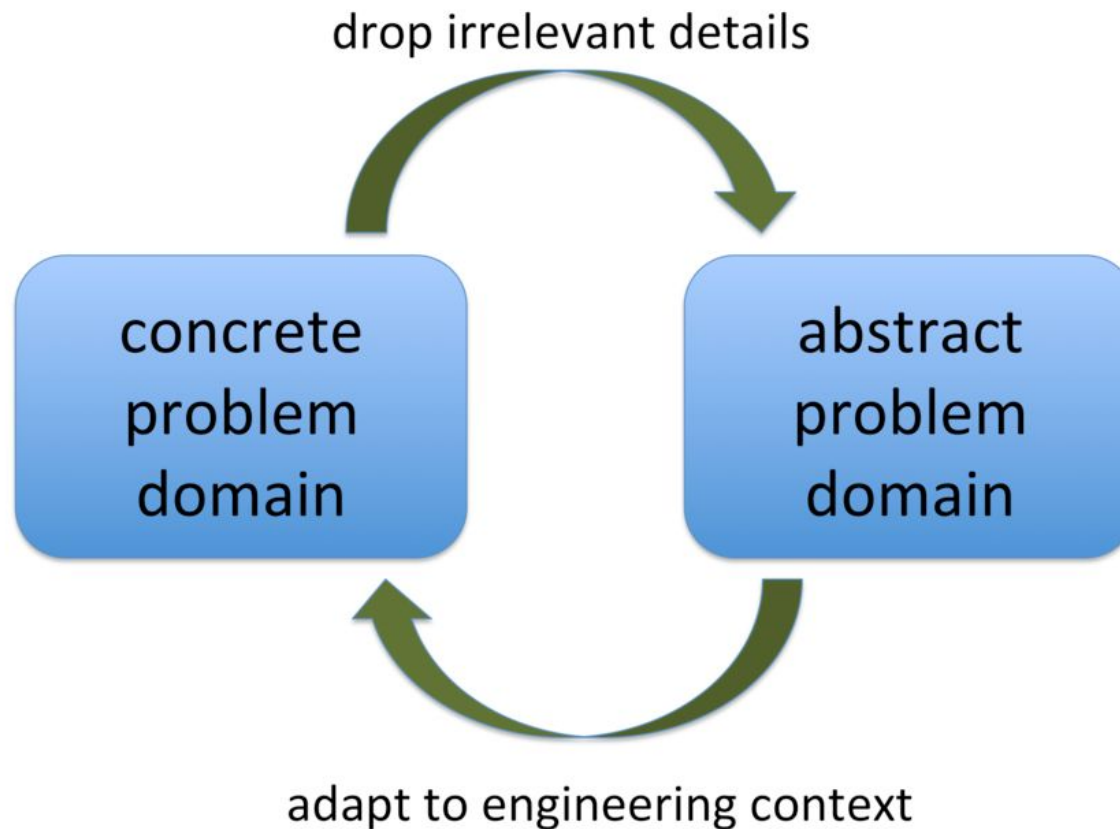
to reproduce some ideas here in our workplace. Now I'm planning to go to grad school and was thinking about which problems I would like to solve, and in which areas. Making science easier and more efficient is something that I like to do. While searching for nice places to study

I found your lab page and read about the "Software engineering methodologies" line of research, and I think that it is a pretty good fit.

But I don't know if you take foreign students, or if you are interested.

My C.V. is in [1], but I'm also sending it attached.

knowledge into software



Closing the Loop: The Importance of External Engagement in Computer Science Research

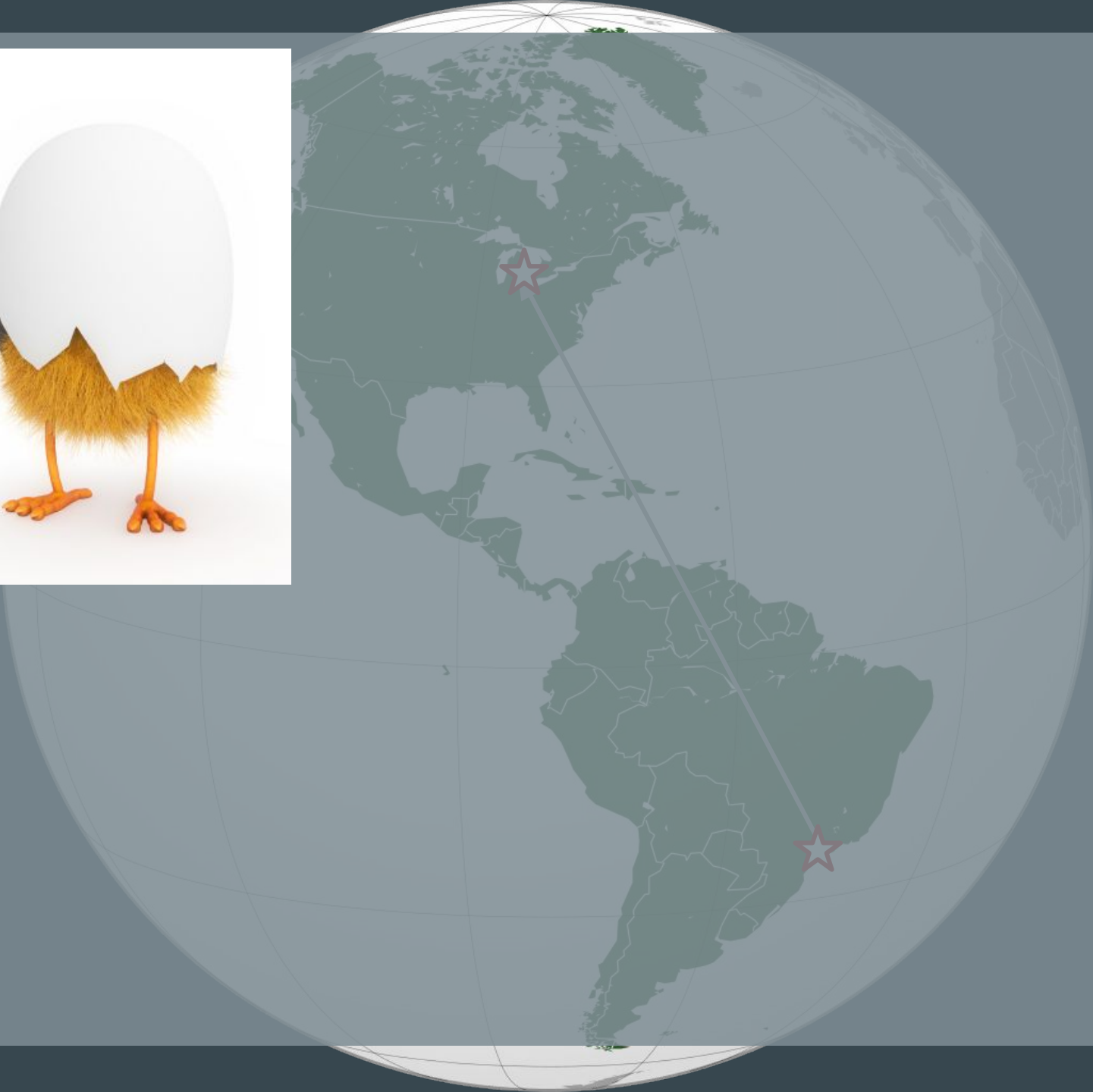
<https://blog.regehr.org/archives/1582>

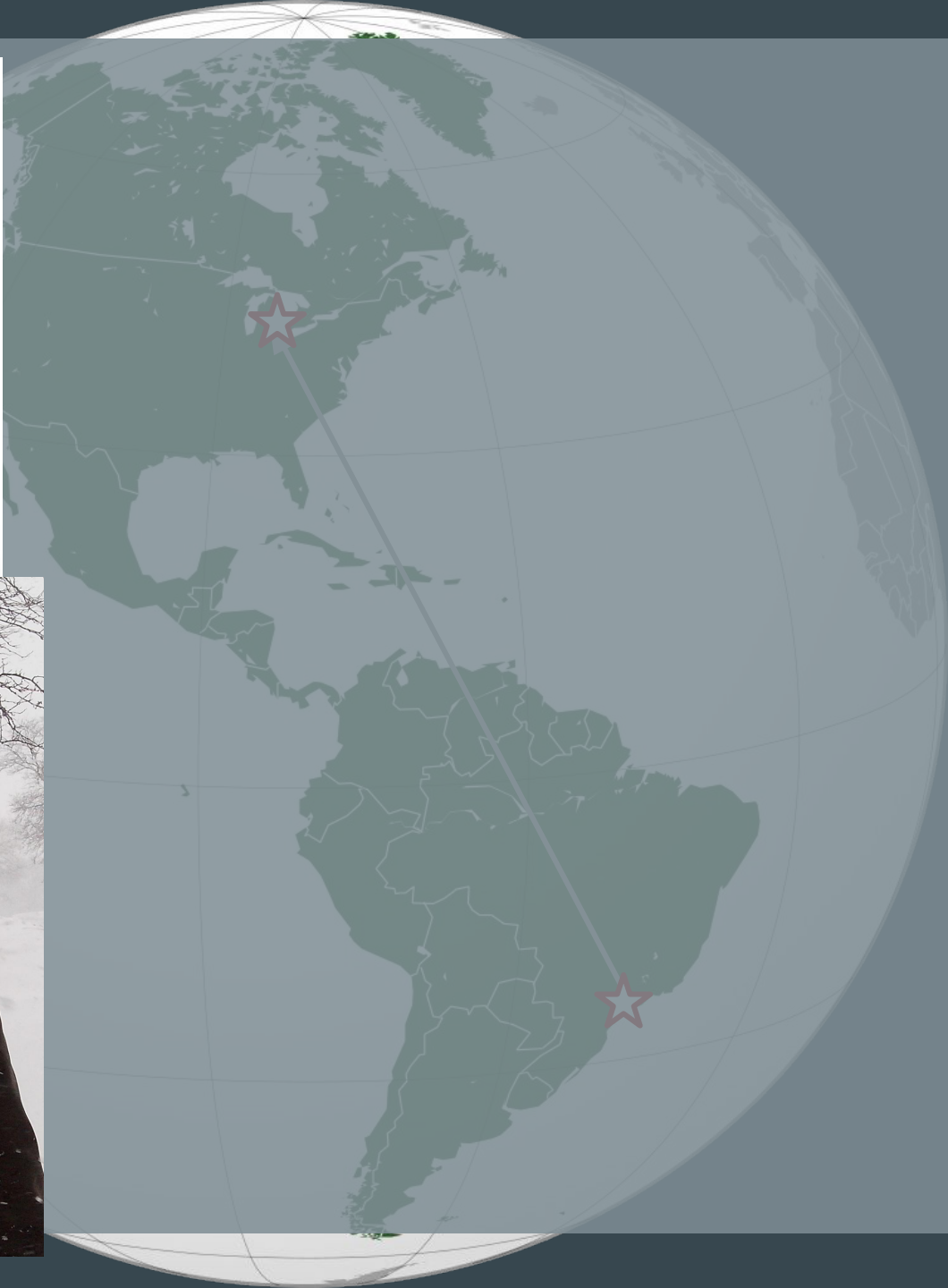




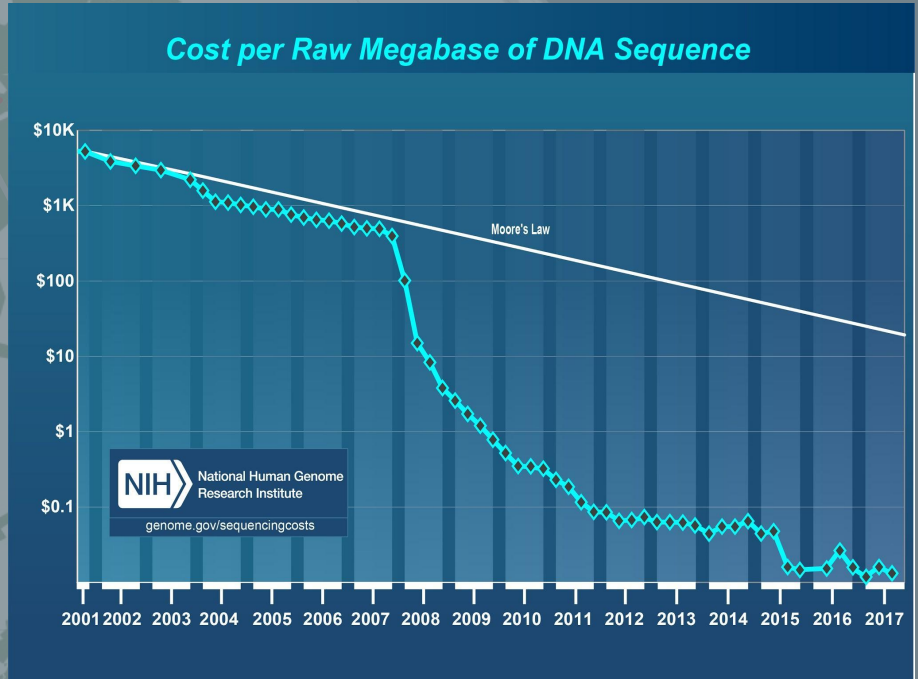




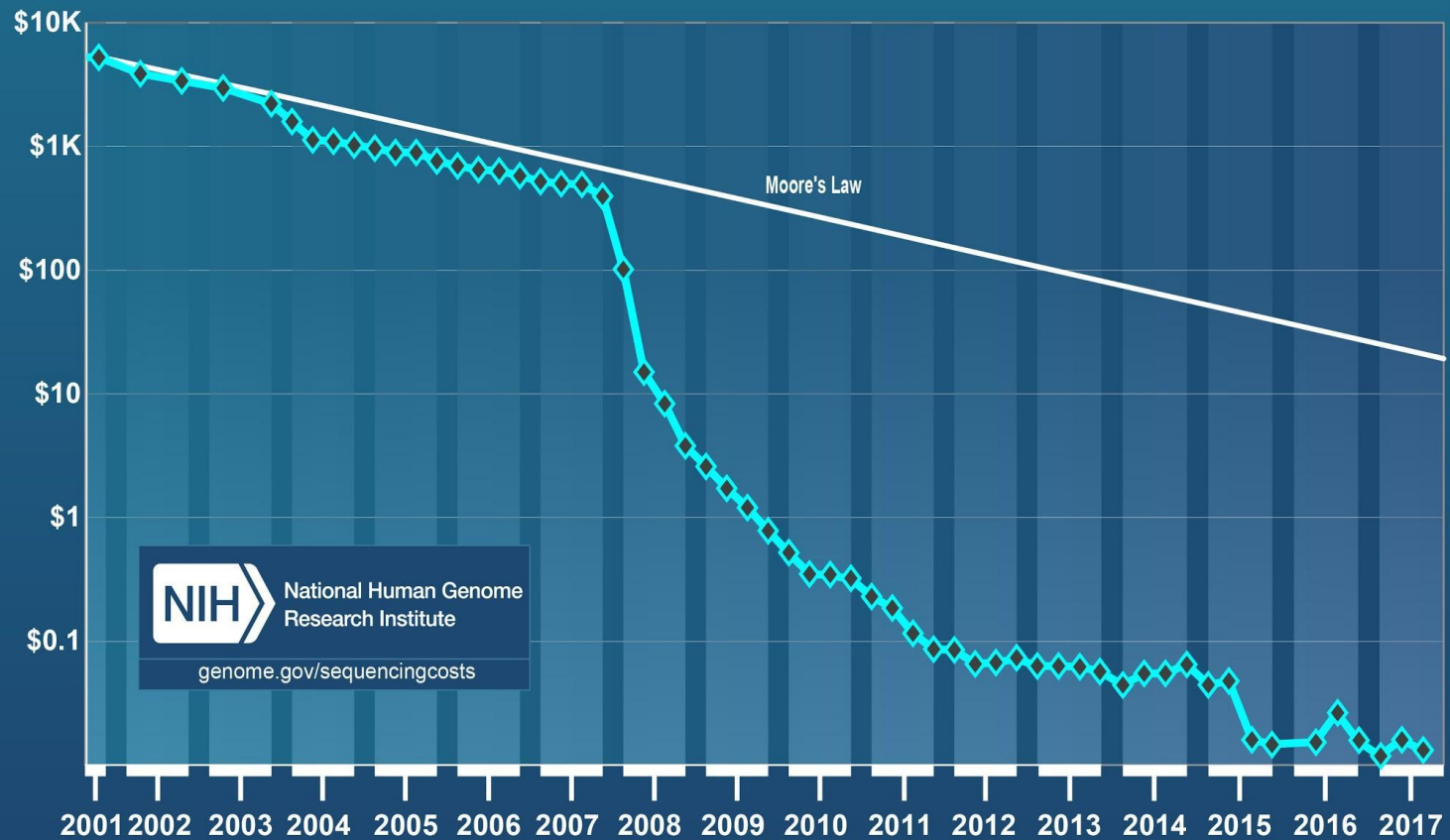








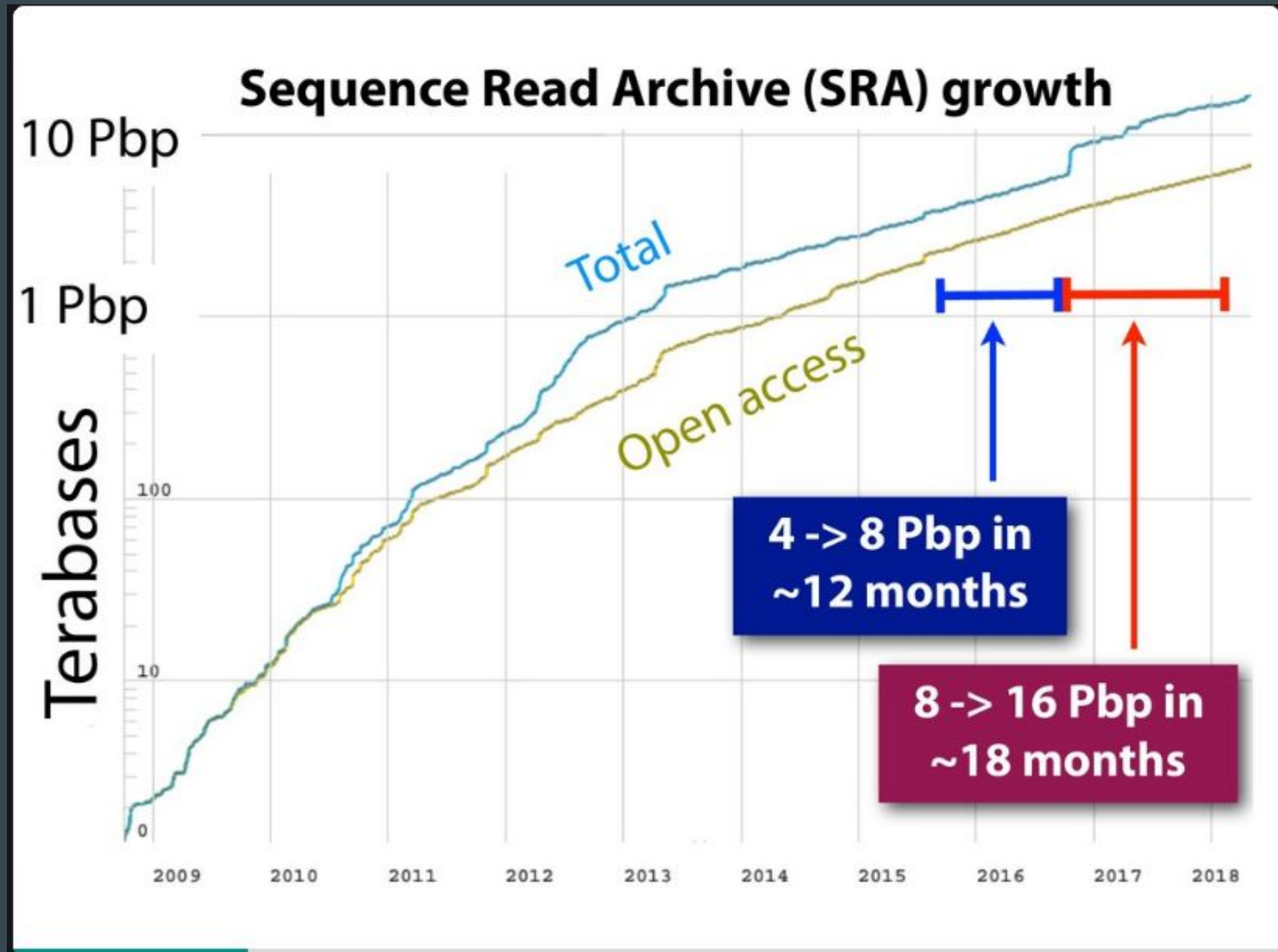
## Cost per Raw Megabase of DNA Sequence



DNA Sequencing Costs: Data

<https://www.genome.gov/sequencingcostsdata/>





Using huge public sequencing datasets to answer scientific questions

<https://speakerdeck.com/benlangmead/using-huge-public-sequencing-datasets-to-answer-scientific-questions-1?slide=6>

# These are good challenges!

- Cheaper sequencing -> more scientists able to use it to answer their questions
- More data -> unveil understudied...
  - Diseases
  - Organisms
  - Environments
- Exploratory experiments -> better hypothesis formulation



# These are terrifying challenges!

- Transferring data becomes a bottleneck
- How to find anything in this mountain of data?
- Is the data trustable?
  - Quality control and assurance
- Current algorithms don't scale
  - “Throw more computing” is not enough



<https://commons.wikimedia.org/wiki/File:Genbank100CD.jpg>

# Experiment discovery problem

- Question: given a query sequence, what datasets contain it (within a threshold)?
- Example query: transcripts
- Example datasets: transcriptomes
- Traditional approach: alignment
  - Global alignment (Dynamic programming)
  - Seed and extend
- Don't scale well to PB of data



# Current solutions - databases and services

- Prepared databases (for tools)
  - Large
  - “University server”
- Sequence databases
  - Public funding
  - Central server
  - Independent mirrors
  - Services accept small queries

# What I want to enable

- Reduce computational resources required...
  - From cluster to laptop
  - From theoretical to practical
- ...to allow efficient search of very large databases...
  - by Similarity and Containment
- ... while building resilient services that
  - distribute the load to users
  - won't disappear when
    - Funding for maintaining servers runs out
    - I graduate? =]
  - “Planning for poverty”

# Aims

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data

1. Using scaled MinHash for containment and cardinality estimation
  - Big dataset -> small sketch
2. Fast queries on many MinHash sketches using MinHash Bloom Trees
  - Indexing N sketches
3. Decentralized indices for genomic data.
  - Keeping in sync and sharing data



# Aims

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data

1. Using scaled MinHash for containment and cardinality estimation
  - Big dataset -> small sketch
2. Fast queries on many MinHash sketches using MinHash Bloom Trees
  - Indexing N sketches
3. Decentralized indices for genomic data.
  - Keeping in sync and sharing data

# MinHash

Big dataset -> small sketch

Indexing N sketches

Keeping in sync and sharing data

- From big dataset to small signature
- Question: how similar are two datasets?
- Question: how much of this dataset we see in another one?

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

$$C(A, B) = \frac{|A \cap B|}{|A|}$$

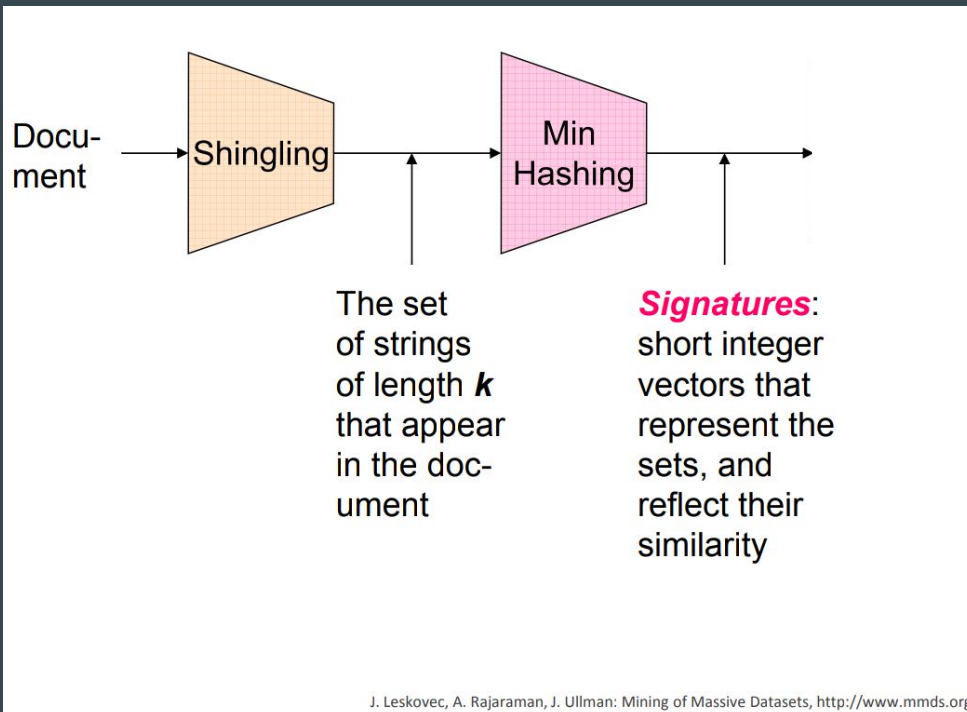
# MinHash

Big dataset -> small sketch

Indexing  $N$  sketches

Keeping in sync and sharing data

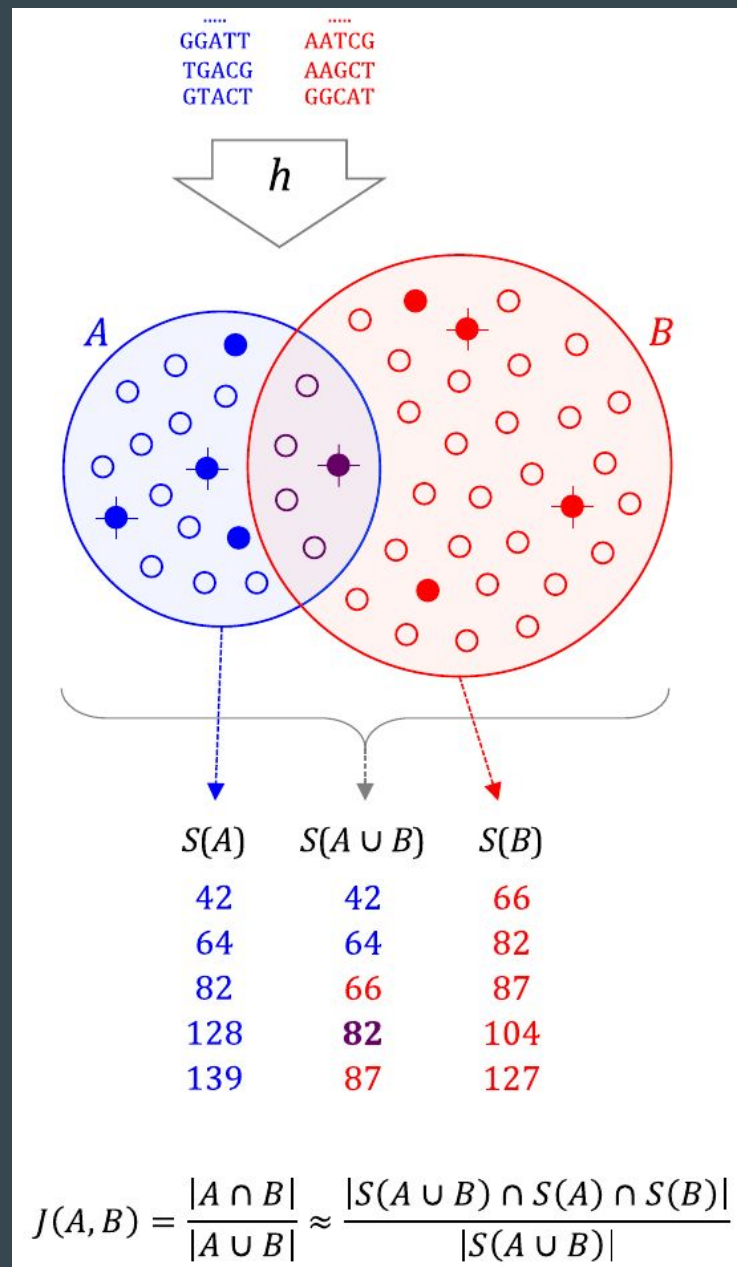
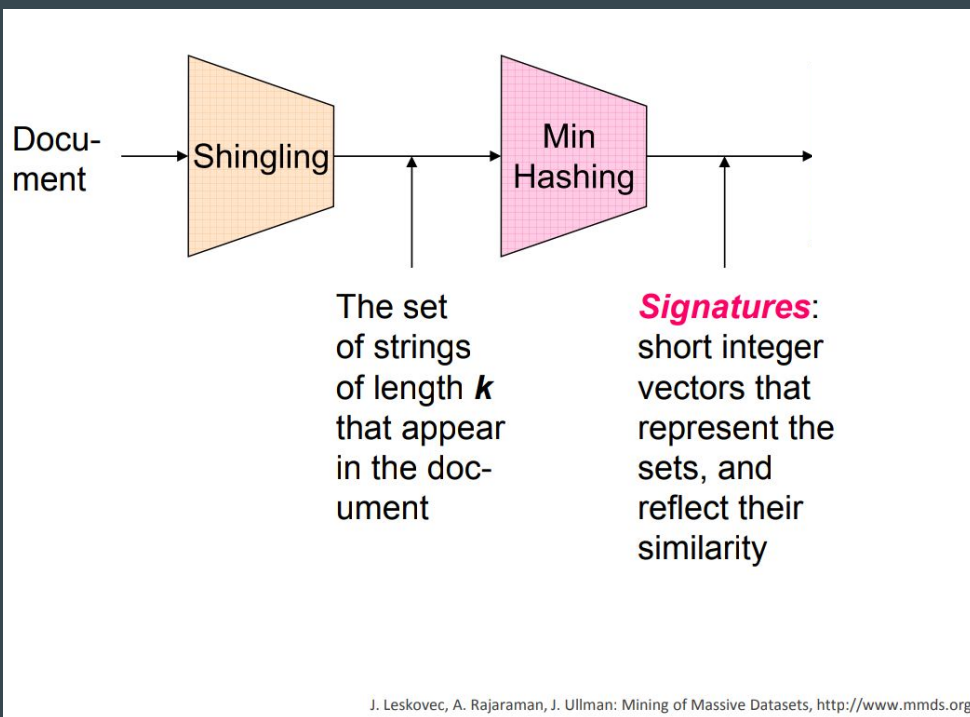
- Broder (1997)
- Comparing web pages (Altavista)
- Estimators
  - Resemblance (Jaccard similarity)
  - Containment





# MinHash

- Broder (1997)
- Comparing web pages (Altavista)
- Estimators
  - Resemblance (Jaccard similarity)
  - Containment



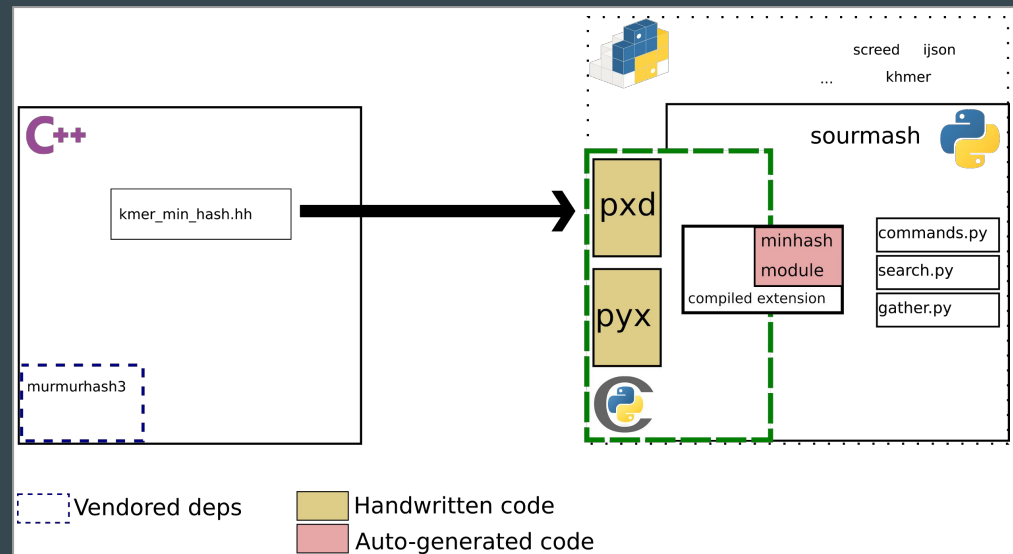
# sourmash 1.0

Big dataset -> small sketch

Indexing N sketches

Keeping in sync and sharing data

- Python library and CLI
  - Core implemented in C++
- Commands
  - Compute
  - Compare
  - Plot
- Compatible with Mash
  - JSON export/import
- Documentation!
- Tutorials!
- Tests!
- Enabled new use cases...



Brown and **Irber**, (2016), sourmash: a library for MinHash sketching of DNA, Journal of Open Source Software

# Scaled MinHash

Big dataset -> small sketch

Indexing  $N$  sketches

Keeping in sync and sharing data

The set  $M(D)$  has the advantage that it has a fixed size, but it allows only the estimation of resemblance. The size of  $L(D)$  grows as  $D$  grows, but allows the estimation of both resemblance and containment.

Broder 1997

- Mash:  $M(D)$
- sourmash:  $L(D)$ 
  - But can convert back to  $M(D)$  for compatibility
- Variable size length, bound by `max_hash/scaled`
- More complexity -> more hashes
  - `minhash(virus) <<< minhash(metagenome)`



# Scaled MinHash

Big dataset -> small sketch

Indexing N sketches

Keeping in sync and sharing data

|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 1  | 0  | 0  | 0  |

$$\text{max\_hash} = \frac{2^{32}}{\text{scaled}}$$

# Scaled MinHash

Big dataset -> small sketch

Indexing N sketches

Keeping in sync and sharing data

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 1  | 0  | 0  | 0  |

$$\text{max\_hash} = \frac{2^{32}}{\text{scaled}}$$

$$\text{max\_hash} = \frac{2^{32}}{10000} \approx 429496 \approx 2^{18.71} \approx 2^{19}$$

# Scaled MinHash

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data



| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 1  | 0  | 0  | 0  |

$$\text{max\_hash} = \frac{2^{32}}{\text{scaled}}$$

$$\text{max\_hash} = \frac{2^{32}}{10000} \approx 429496 \approx 2^{18.71} \approx 2^{19}$$



# Scaled MinHash

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data



|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 1  | 0  | 0  | 0  |



$$\text{max\_hash} = \frac{2^{32}}{\text{scaled}}$$

$$\text{max\_hash} = \frac{2^{32}}{10000} \approx 429496 \approx 2^{18.71} \approx 2^{19}$$

# Scaled MinHash

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data



$$\text{max\_hash} = \frac{2^{32}}{\text{scaled}}$$

$$\text{max\_hash} = \frac{2^{32}}{10000} \approx 429496 \approx 2^{18.71} \approx 2^{19}$$

# Scaled MinHash

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data



$$\text{max\_hash} = \frac{2^{32}}{\text{scaled}}$$

$$\text{max\_hash} = \frac{2^{32}}{10000} \approx 429496 \approx 2^{18.71} \approx 2^{19}$$



# Supporting more operations in MinHash sketches

- Mash: similarity
- Scaled MinHash: similarity and containment
- Extensions:
  - Abundance tracking (set to multiset)
  - Cardinality estimation

PARTITION #2

$$p(x) = 3$$


01000101...



Juan P Lopes - A Language for Real Time Stream Processing

<https://speakerdeck.com/juanplobes/nubank-machine-learning-meetup>

HARMONIC  
MEAN


$$\alpha_m m^2 \left( \sum_{j=0}^{m-1} 2^{-M[j]} \right)^{-1}$$

|   |   |   |    |   |   |   |   |
|---|---|---|----|---|---|---|---|
| 1 | 8 | 3 | 12 | 9 | 1 | 0 | 7 |
| 0 | 1 | 2 | 3  | 4 | 5 | 6 | 7 |

Juan P Lopes - A Language for Real Time Stream Processing

<https://speakerdeck.com/juanplobes/nubank-machine-learning-meetup>

# HyperLogLog sketch

- linear transformation of the original data maintaining desirable properties
- Memory/accuracy tradeoffs -> approximate answer
- Exact answer is too expensive!
- Fast, parallel, composable

Counting the number of unique DNA substrings (k=32)

700k basepairs

| Implementation | Time (seconds) | Memory (MB) | Cardinality | Error |
|----------------|----------------|-------------|-------------|-------|
| HLL            | 0.193          | 13.44       | 670,328     | ~0    |
| C++/sparsehash | 3.888          | 77.64       | 670,487     | 0     |
| Python         | 4.306          | 83.48       | 670,487     | 0     |

| Implementation | Time (seconds) | Memory (MB) | Cardinality | Error |
|----------------|----------------|-------------|-------------|-------|
| HLL            | 12.08          | 13.38       | 17,686,322  | 1.01  |
| C++/sparsehash | 344.04         | 2,018.50    | 17,510,301  | 0     |
| Python         | 891.17         | 2,056.05    | 17,510,301  | 0     |

169M basepairs

Irber and Brown (2016). *Efficient cardinality estimation for k-mers in large DNA sequencing data sets*, Biorxiv. <https://doi.org/10.1101/056846>



# What if...

- Big dataset -> small sketch
- Indexing N sketches
- Keeping in sync and sharing data



# What if...

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data



# Aims

Big dataset -> small sketch

Indexing N sketches

Keeping in sync and sharing data

1. Using scaled MinHash for containment and cardinality estimation
  - Big dataset -> small sketch
2. Fast queries on many MinHash sketches using MinHash Bloom Trees
  - Indexing N sketches
3. Decentralized indices for genomic data.
  - Keeping in sync and sharing data

Big dataset -> small sketch

Indexing  $N$  sketches

Keeping in sync and sharing data

# Experiment discovery problem

- Question: given a query sequence, what datasets contain it (within a threshold)?
- Example query: transcripts
- Example datasets: transcriptomes
- Traditional approach: alignment
  - Global alignment (Dynamic programming)
  - Seed and extend
- Don't scale well to PB of data



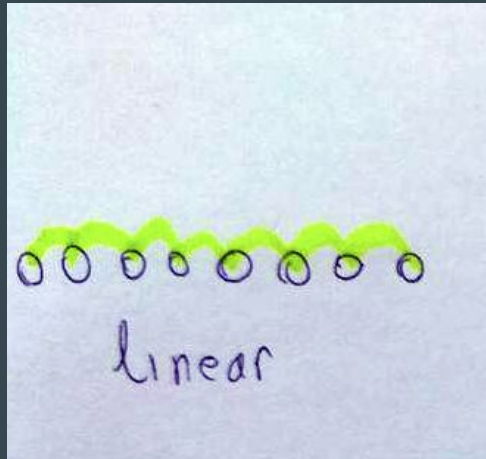
# Indexing and search

Big dataset -> small sketch

Indexing  $N$  sketches

Keeping in sync and sharing data

- From one signature to many signatures
- Once we have many signatures, how to search them quickly?
- Linear -  $O(n)$ 
  - 1 million signatures -> 1 million comparisons



# Sequence Bloom Trees

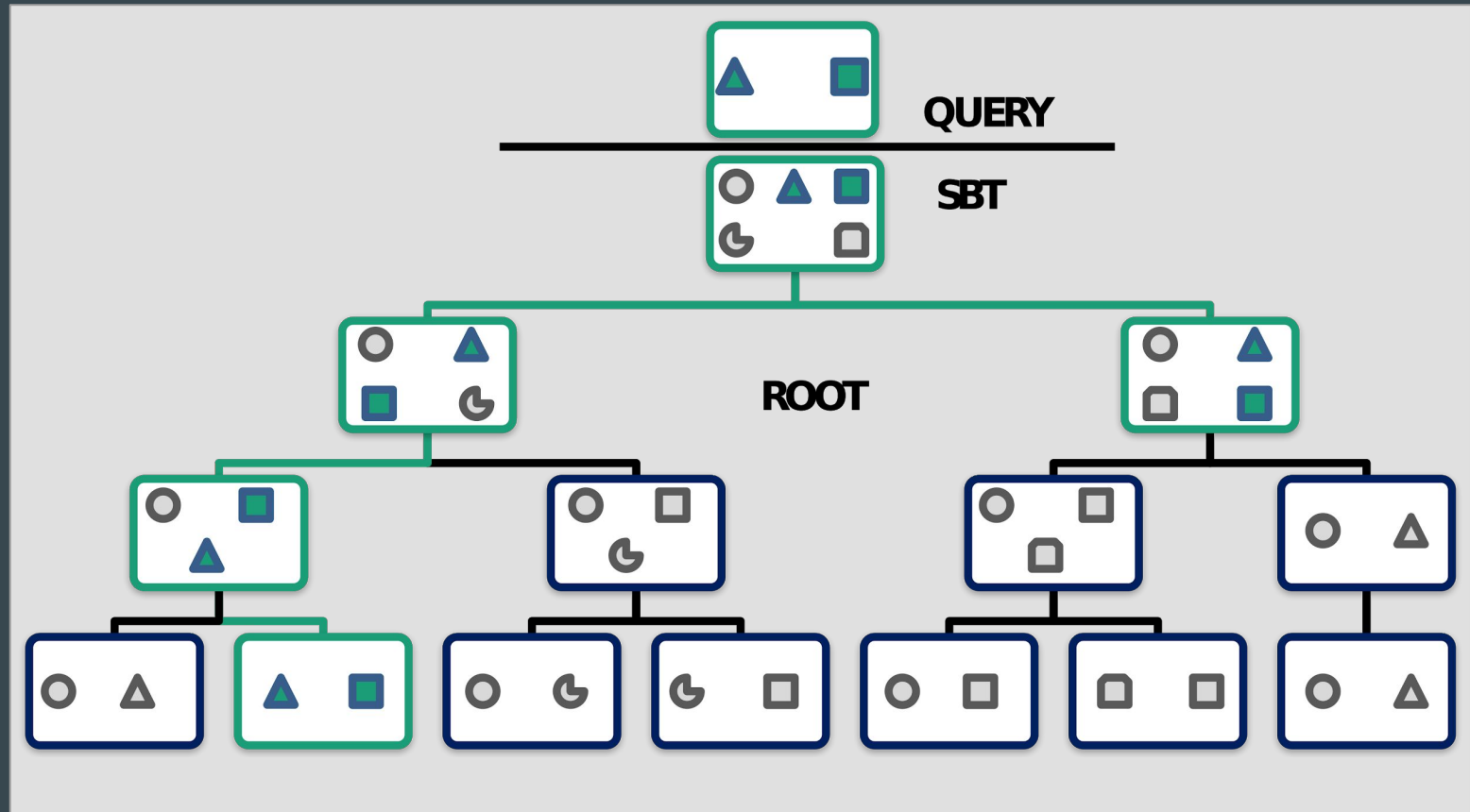
(Solomon 2016)

Big dataset -> small sketch

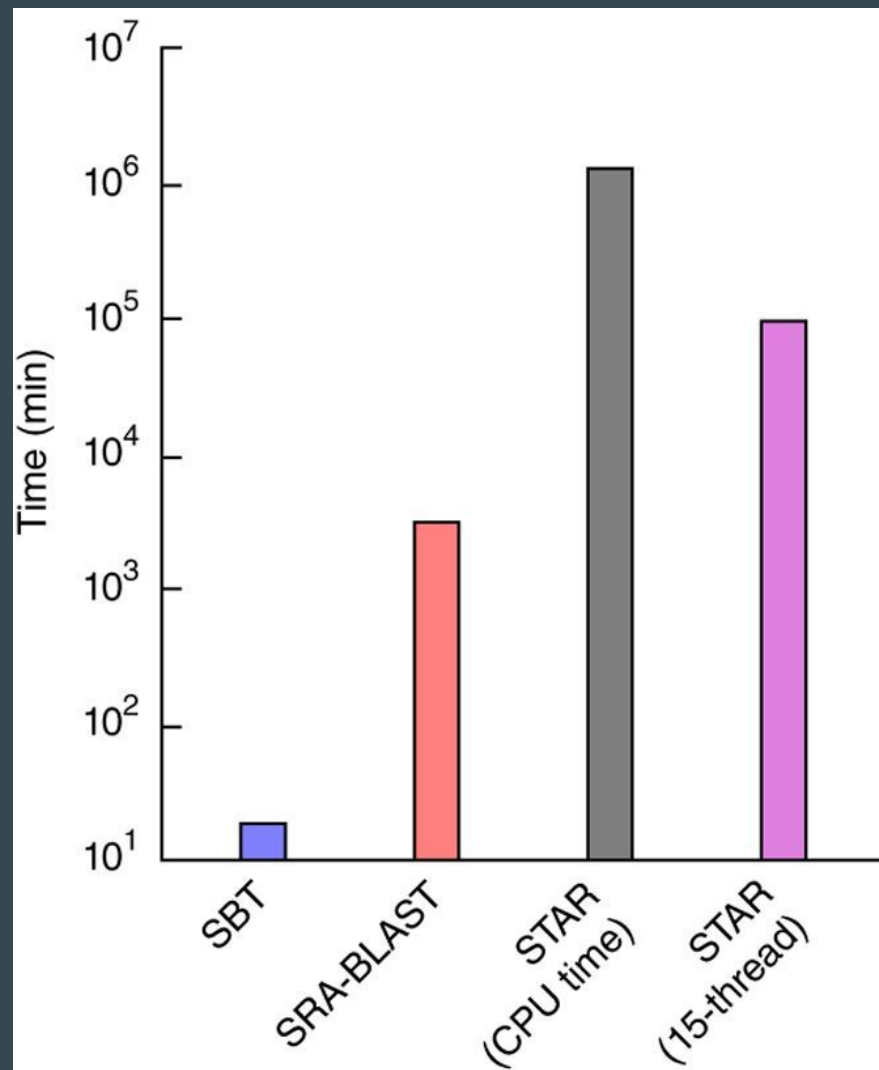
Indexing  $N$  sketches

Keeping in sync and sharing data

- For each dataset, compute k-mer set and add to a Bloom Filter
- Hierarchical index: Internal nodes contain all k-mer below it



Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data



Solomon and Kingsford (2016). “Fast Search of Thousands of Short-Read Sequencing Experiments.”  
Nature Biotechnology 34 (3): 300–302. <https://doi.org/10.1038/nbt.3442>.

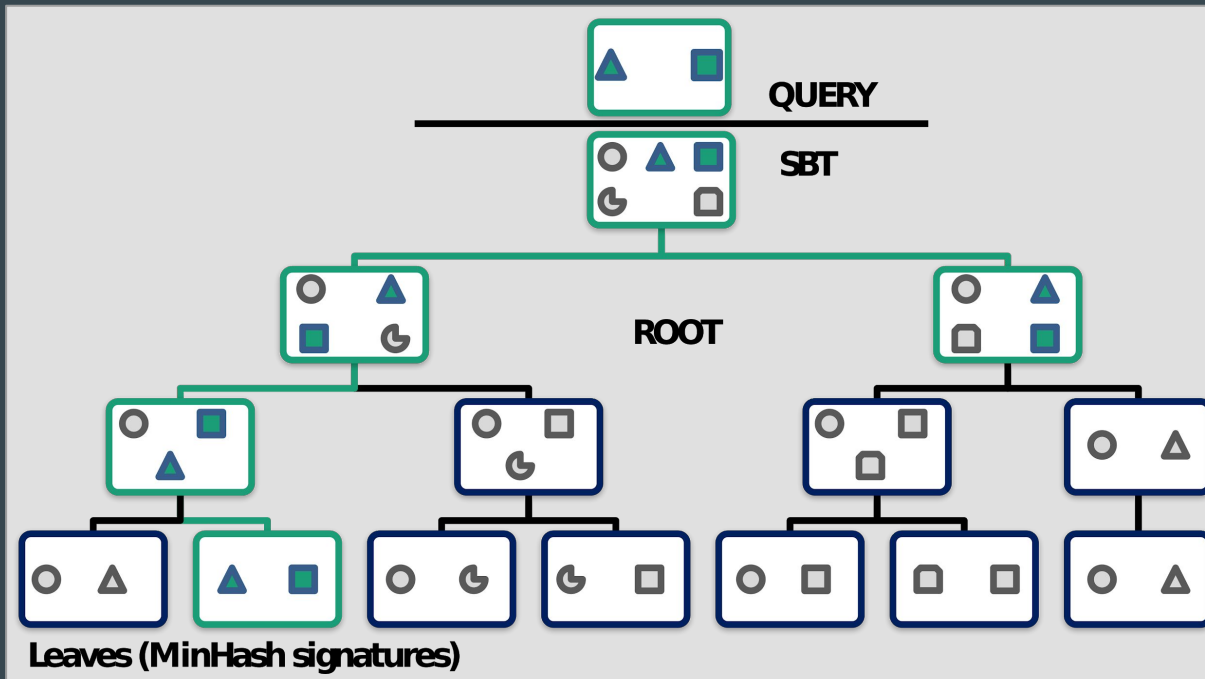
# SBT + MH: MHBT

Big dataset -> small sketch

Indexing  $N$  sketches

Keeping in sync and sharing data

- Different question: given an experiment, what other similar experiments are present in a collection?
- Internal nodes are still Bloom Filters, but leaves are Minhash sketches





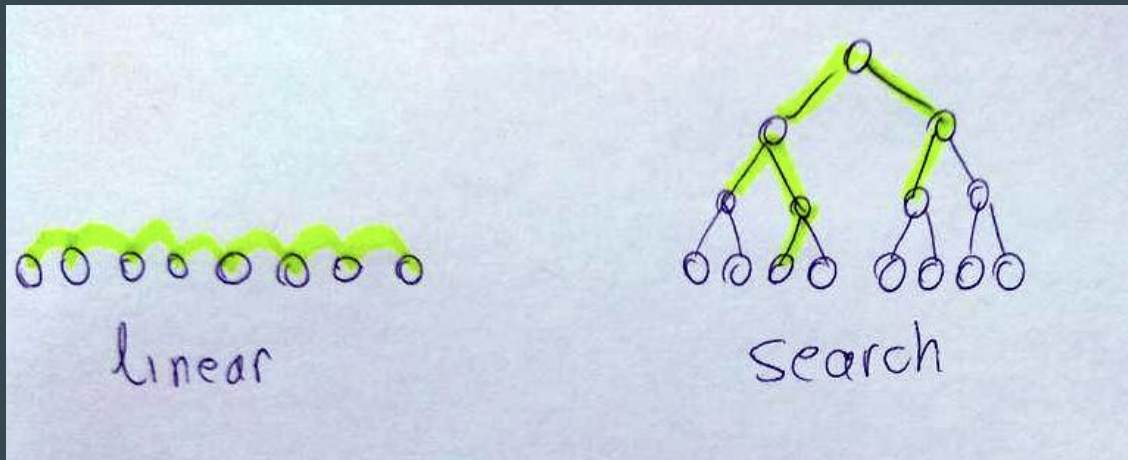
# sourmash search

Big dataset -> small sketch

Indexing  $N$  sketches

Keeping in sync and sharing data

- From one signature to many signatures
- Once we have many signatures, how to search them quickly?
- Linear
  - 1 million signatures -> 1 million comparisons
- Search - Breadth-first search, early pruning
- Question: which are all the datasets similar to my query?

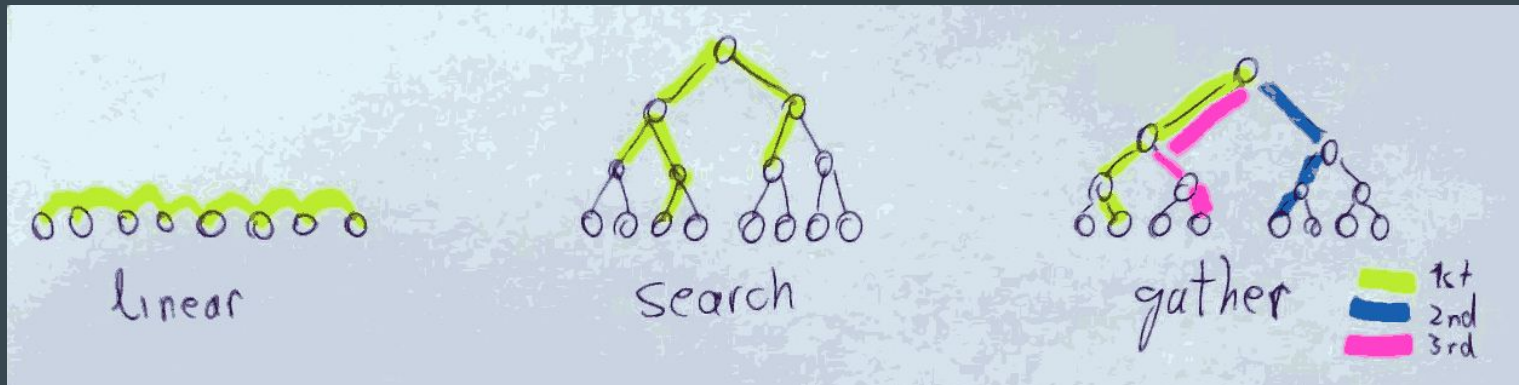


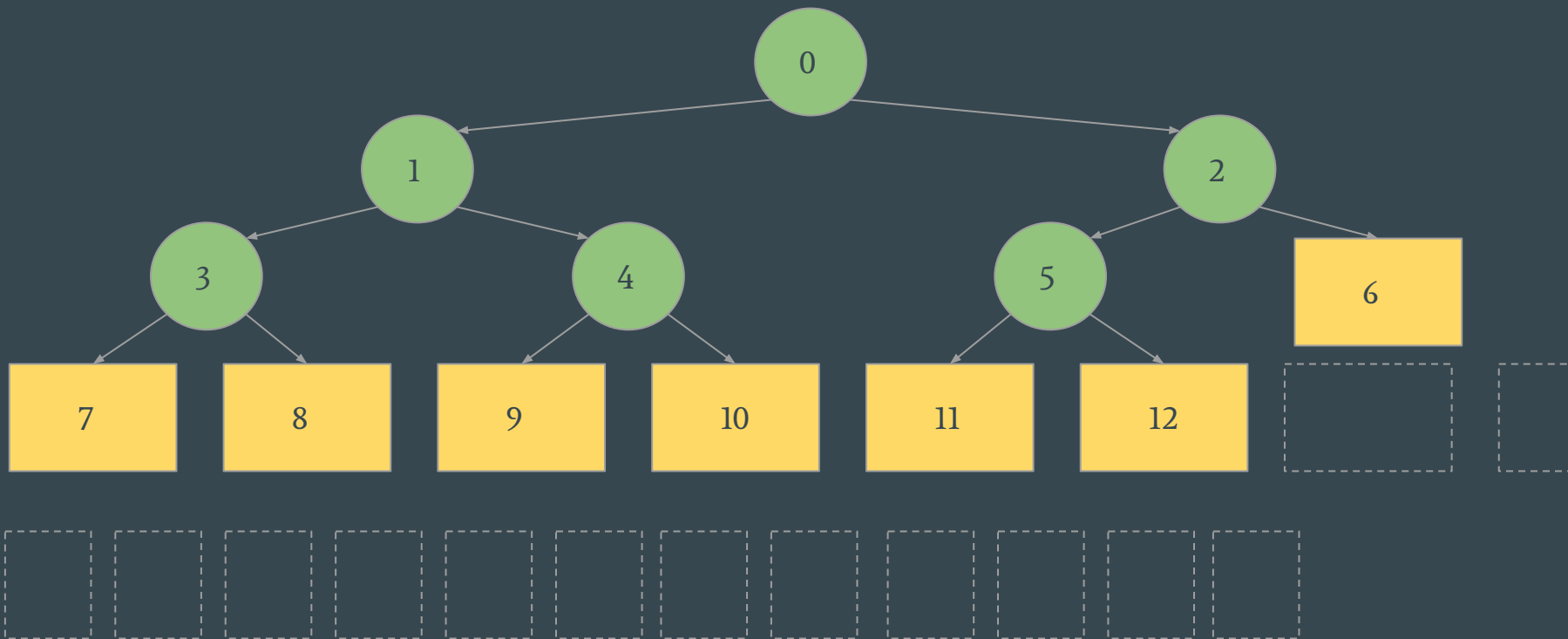
# Another use case: Taxonomic classification

- What are the species present in a sample?
- And what are their abundances?
- Current methods
  - k-mer composition (Kraken)
  - Alignment (Diamond-MEGAN)
  - Markers (PhyloSift)

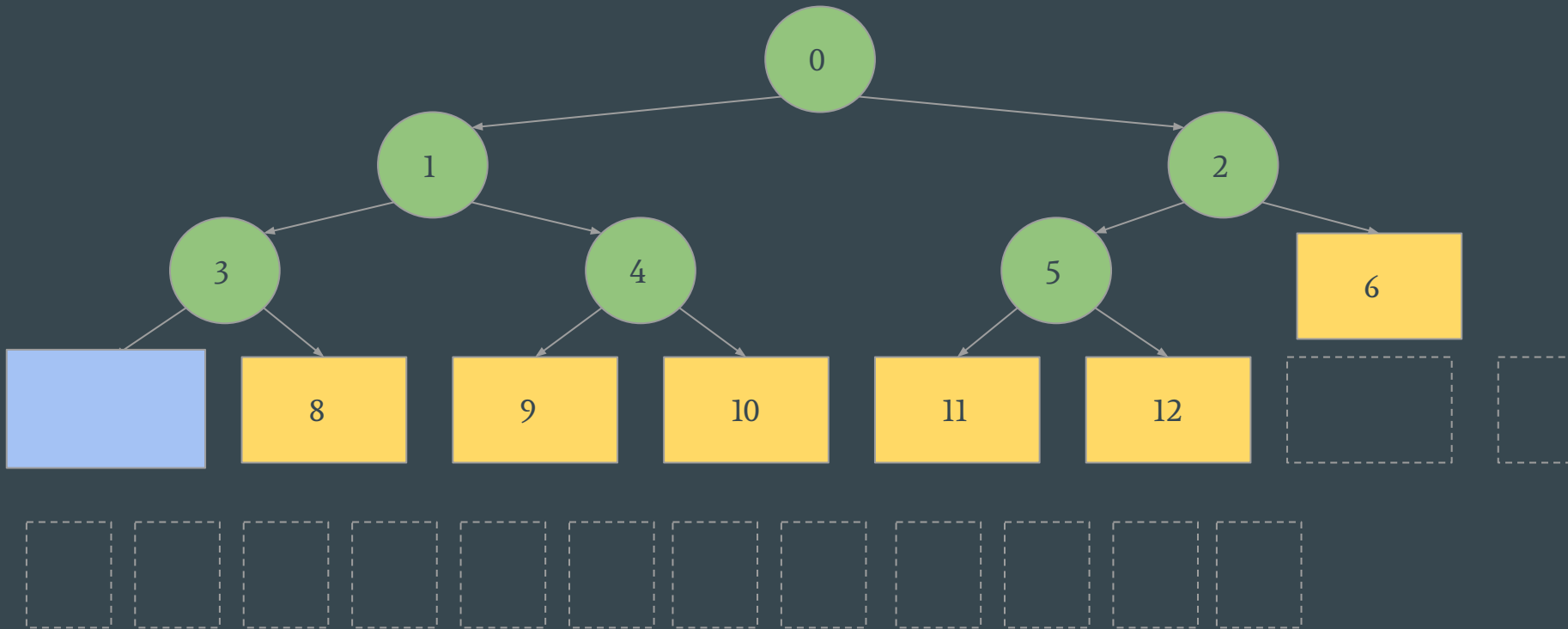
# sourmash gather

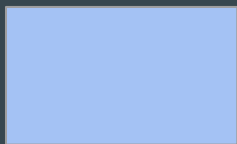
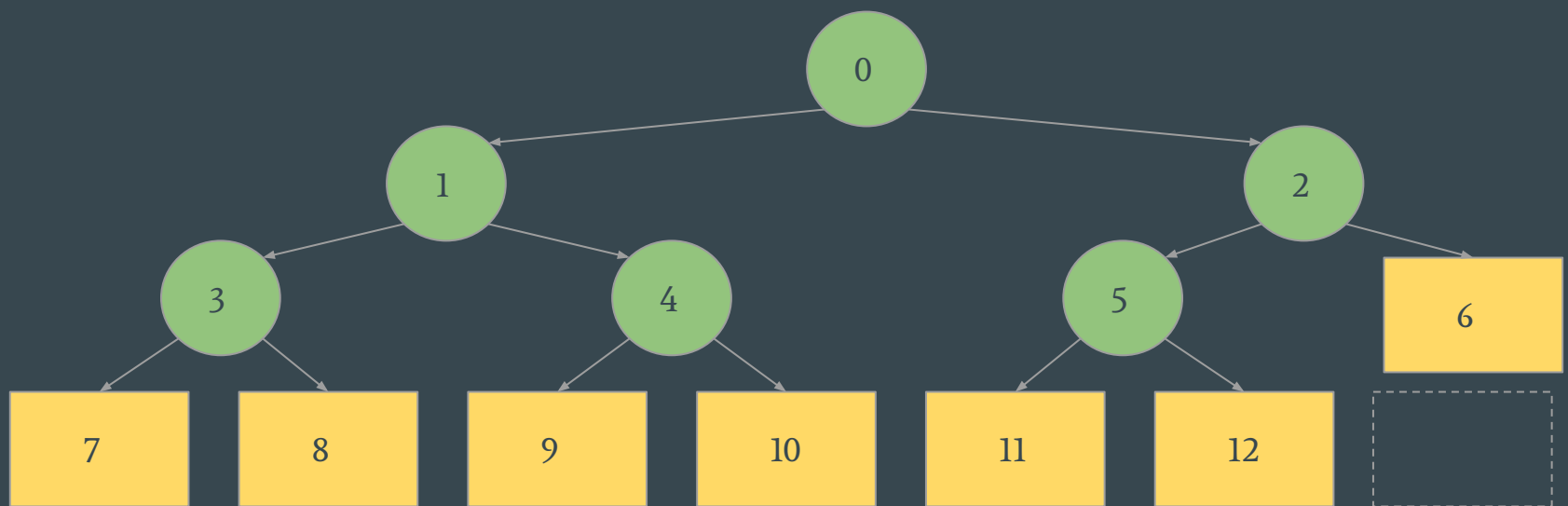
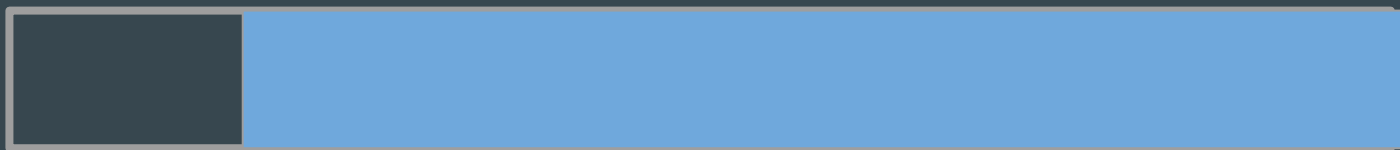
- Same search index
- Distinct search method
  - search: breadth-first with early pruning
  - gather: multiple best-first searches
- A mix of k-mer composition and markers
  - Uses hashed k-mers...
  - ... but only a systematically consistent subset

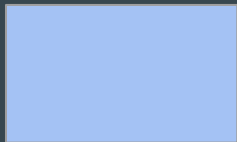
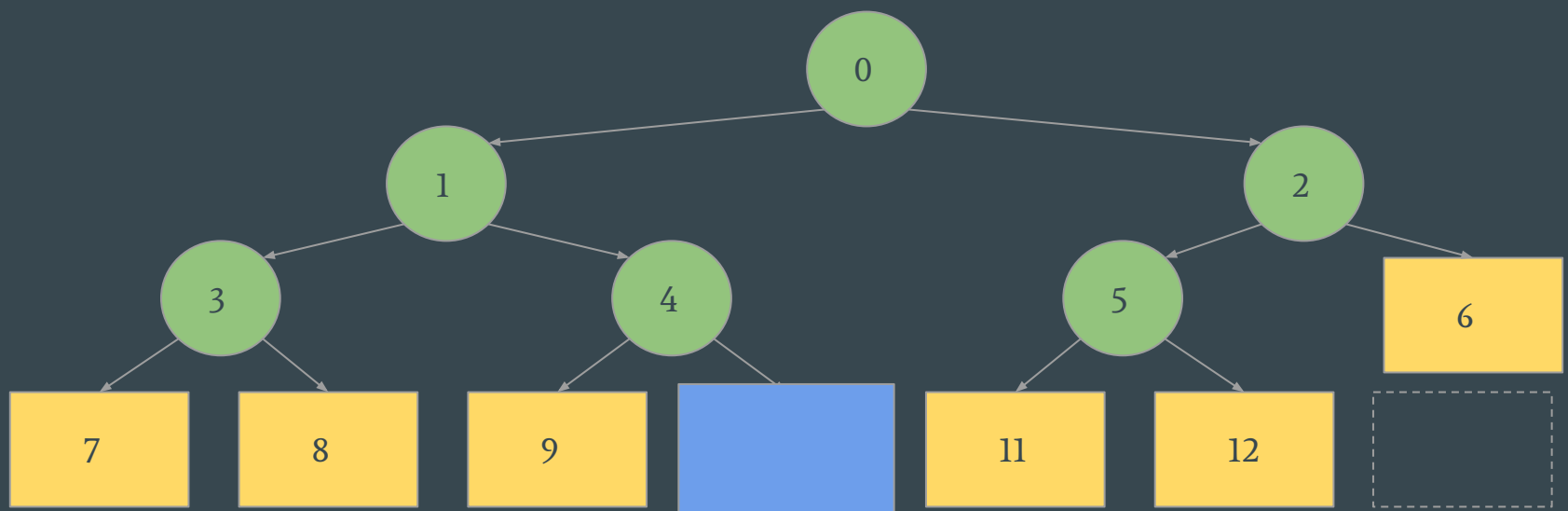
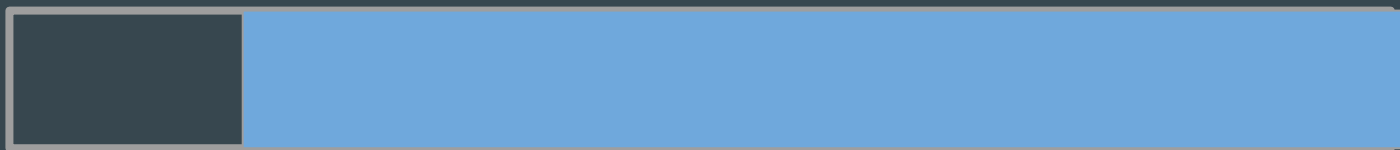


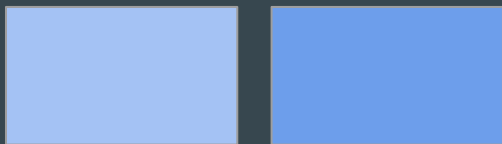
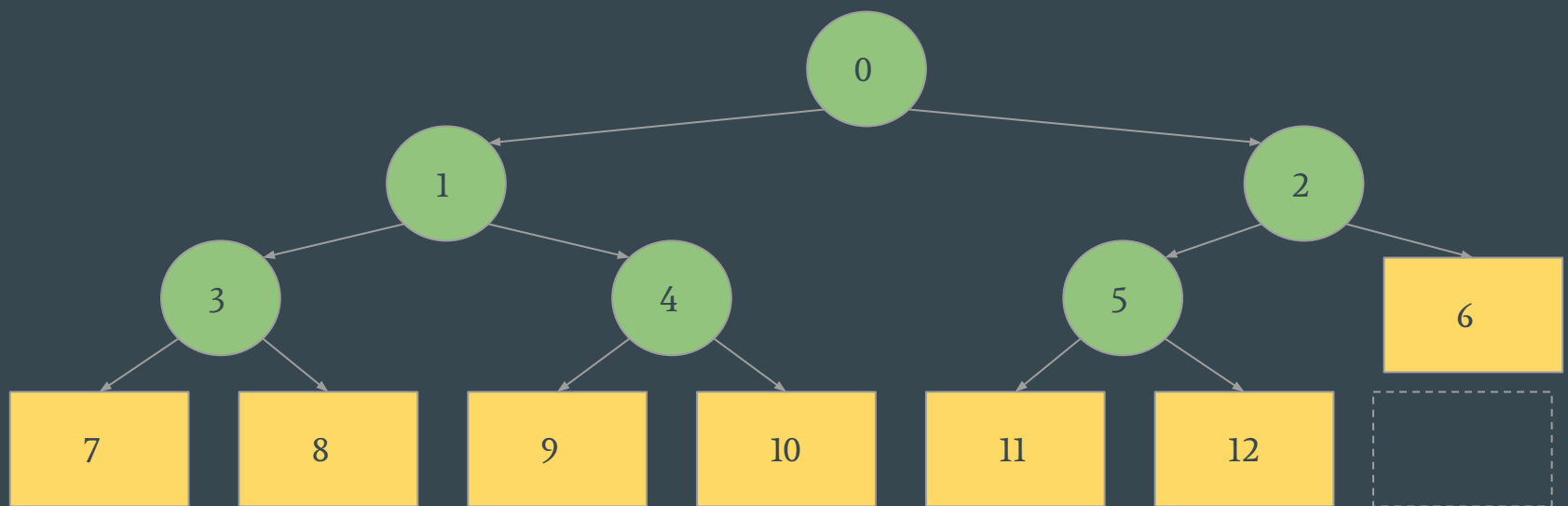
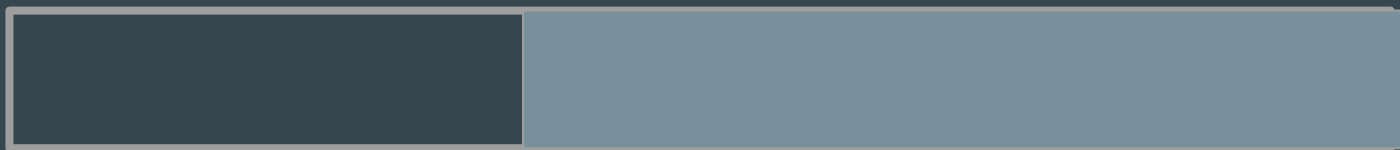


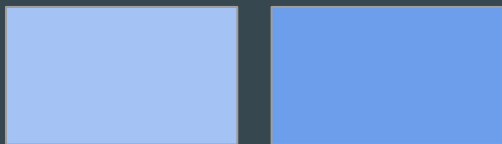
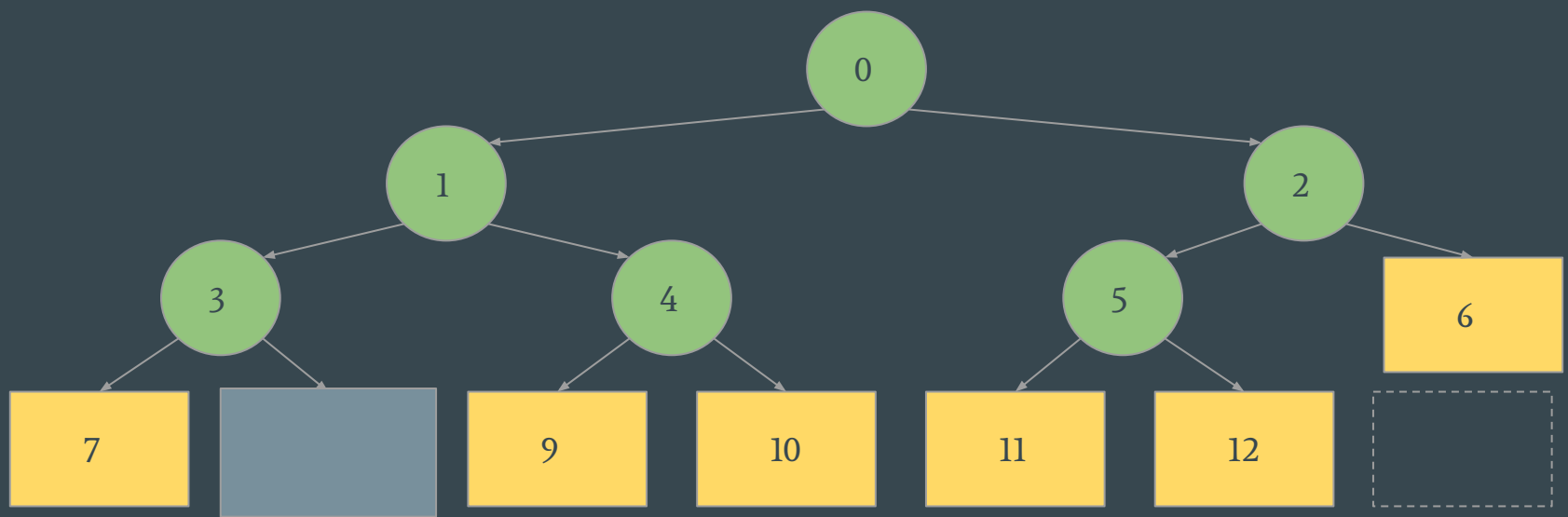
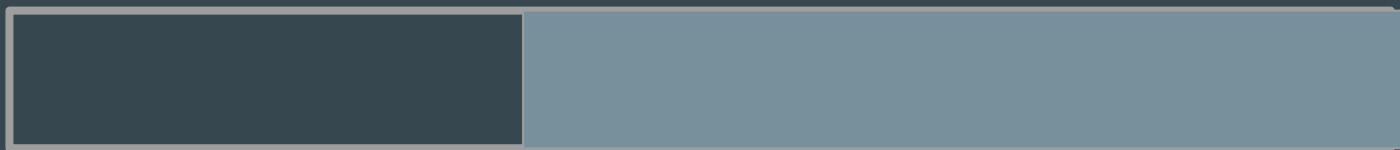




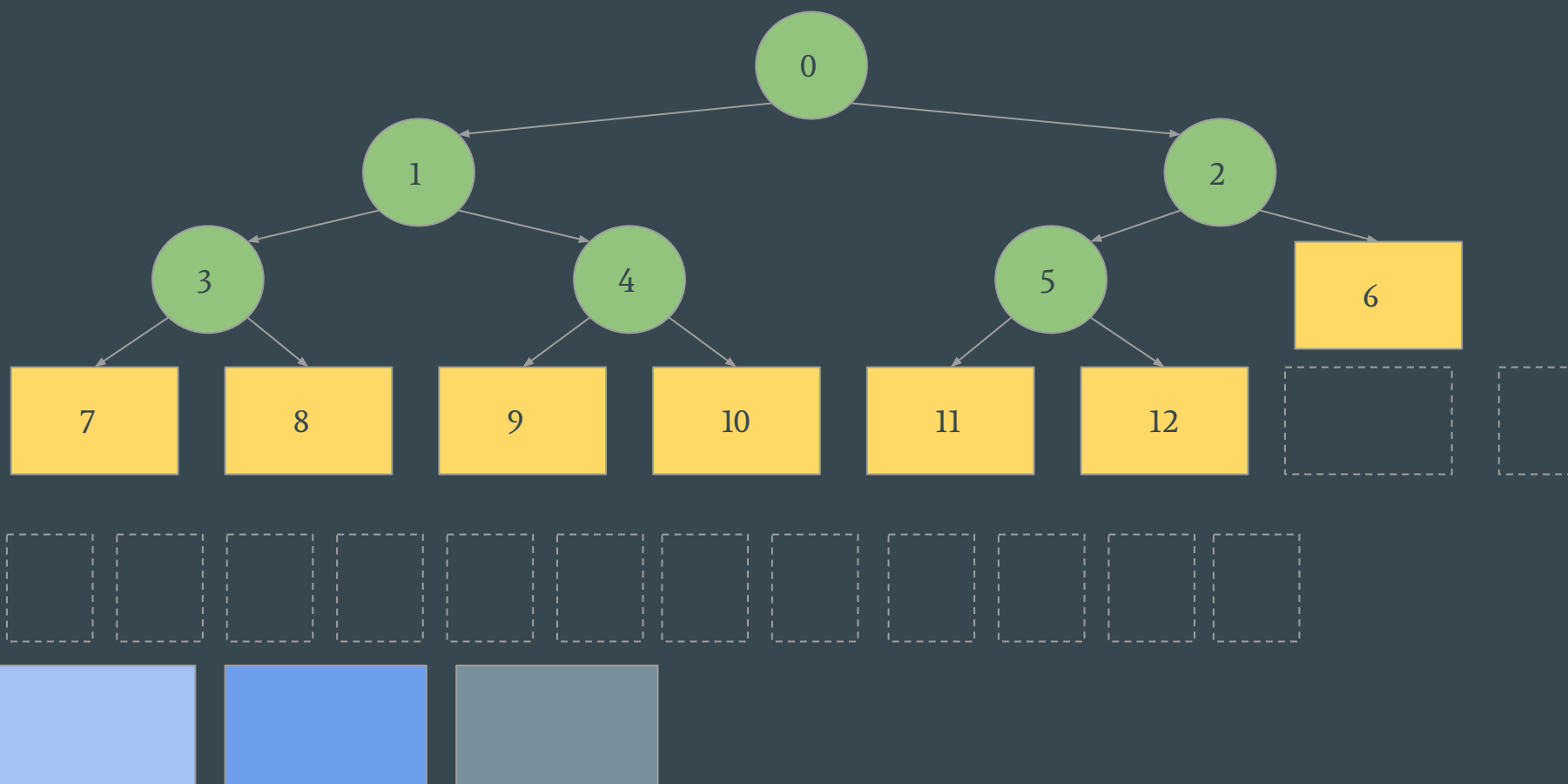












| Dataset_Name         | Paired-End | Species_Present | Total_Reads | Unique_51-mers | Measure   | Sourmash | GOTTCHA | DiamondMegan_filtered | LMAT   | BlastMegan_filtered | Kraken_filtered | BlastMegan_filtered_liberal | CLARK-S | Kraken | CLARK  | MetaPhlAn | MetaFlow | PhyloSift_filtered | PhyloSift | NBC   |
|----------------------|------------|-----------------|-------------|----------------|-----------|----------|---------|-----------------------|--------|---------------------|-----------------|-----------------------------|---------|--------|--------|-----------|----------|--------------------|-----------|-------|
| ds.7                 | no         | 523             | 5727654     | 144727105      | precision | 99.58    | 98.62   | 100.00                | 99.79  | 99.43               | 98.20           | 97.55                       | 96.50   | 96.86  | 96.70  | 96.48     | 98.44    | 66.41              | 67.94     | 42.79 |
| ds.7                 | no         | 523             | 5727654     | 144727105      | recall    | 90.06    | 97.47   | 26.46                 | 91.05  | 67.51               | 95.72           | 93.00                       | 96.69   | 95.91  | 96.89  | 79.96     | 61.28    | 84.63              | 86.19     | 51.36 |
| ds.buccal            | no         | 12              | 600000      | 6193231        | precision | 100.00   | 83.33   | 100.00                | 70.59  | 62.50               | 50.00           | 43.48                       | 41.67   | 32.26  | 33.33  | 90.91     | 100.00   | 34.38              | 5.39      | 1.74  |
| ds.buccal            | no         | 12              | 600000      | 6193231        | recall    | 100.00   | 83.33   | 100.00                | 100.00 | 83.33               | 83.33           | 83.33                       | 83.33   | 83.33  | 83.33  | 83.33     | 25.00    | 91.67              | 91.67     | 58.33 |
| ds.cityparks         | no         | 48              | 1200000     | 41614294       | precision | 100.00   | 100.00  | 100.00                | 94.00  | 100.00              | 94.12           | 92.31                       | 90.57   | 87.27  | 85.71  | 100.00    | 100.00   | 20.87              | 14.38     | 5.60  |
| ds.cityparks         | no         | 48              | 1200000     | 41614294       | recall    | 100.00   | 95.83   | 100.00                | 97.92  | 100.00              | 100.00          | 100.00                      | 100.00  | 100.00 | 100.00 | 79.17     | 62.50    | 89.58              | 93.75     | 58.33 |
| ds.gut               | no         | 20              | 500000      | 10904560       | precision | 100.00   | 100.00  | 100.00                | 90.91  | 95.00               | 79.17           | 79.17                       | 61.29   | 65.52  | 65.52  | 86.67     | 100.00   | 14.29              | 5.99      | 9.18  |
| ds.gut               | no         | 20              | 500000      | 10904560       | recall    | 100.00   | 95.00   | 100.00                | 100.00 | 95.00               | 95.00           | 95.00                       | 95.00   | 95.00  | 95.00  | 65.00     | 35.00    | 95.00              | 95.00     | 95.00 |
| ds.hous1             | no         | 30              | 750000      | 20736401       | precision | 100.00   | 100.00  | 100.00                | 100.00 | 96.77               | 93.75           | 83.33                       | 78.95   | 76.92  | 76.92  | 100.00    | 100.00   | 27.18              | 9.27      | 4.20  |
| ds.hous1             | no         | 30              | 750000      | 20736401       | recall    | 100.00   | 93.33   | 100.00                | 100.00 | 100.00              | 100.00          | 100.00                      | 100.00  | 100.00 | 100.00 | 80.00     | 46.67    | 93.33              | 93.33     | 66.67 |
| ds.hous2             | no         | 20              | 500000      | 12867127       | precision | 100.00   | 100.00  | 95.00                 | 70.37  | 90.48               | 90.91           | 74.07                       | 80.00   | 80.00  | 76.92  | 86.67     | 100.00   | 20.45              | 6.71      | 2.74  |
| ds.hous2             | no         | 20              | 500000      | 12867127       | recall    | 95.00    | 90.00   | 95.00                 | 95.00  | 95.00               | 100.00          | 100.00                      | 100.00  | 100.00 | 100.00 | 65.00     | 60.00    | 90.00              | 95.00     | 65.00 |
| ds.nyccsm            | no         | 20              | 500000      | 11515743       | precision | 100.00   | 100.00  | 100.00                | 95.24  | 100.00              | 83.33           | 83.33                       | 71.43   | 76.92  | 76.92  | 93.75     | 100.00   | 12.32              | 5.01      | 3.33  |
| ds.nyccsm            | no         | 20              | 500000      | 11515743       | recall    | 100.00   | 95.00   | 100.00                | 100.00 | 100.00              | 100.00          | 100.00                      | 100.00  | 100.00 | 100.00 | 75.00     | 45.00    | 85.00              | 90.00     | 75.00 |
| ds.soil              | no         | 50              | 2500000     | 77422158       | precision | 100.00   | 100.00  | 100.00                | 100.00 | 100.00              | 98.00           | 90.91                       | 84.75   | 85.96  | 86.21  | 95.65     | 100.00   | 15.02              | 13.95     | 5.69  |
| ds.soil              | no         | 50              | 2500000     | 77422158       | recall    | 100.00   | 96.00   | 100.00                | 96.00  | 98.00               | 98.00           | 100.00                      | 100.00  | 98.00  | 100.00 | 88.00     | 64.00    | 94.00              | 94.00     | 58.00 |
| ds_Average_Precision |            |                 |             |                |           | 99.95    | 97.74   | 99.38                 | 90.11  | 93.02               | 85.94           | 80.52                       | 75.64   | 75.21  | 74.78  | 93.77     | 99.80    | 26.37              | 16.08     | 9.41  |
| ds_Average_Recall    |            |                 |             |                |           | 98.13    | 93.25   | 90.18                 | 97.50  | 92.36               | 96.51           | 96.42                       | 96.88   | 96.53  | 96.90  | 76.93     | 49.93    | 90.40              | 92.37     | 65.96 |
| ds_Average_All       |            |                 |             |                |           | 99.04    | 95.50   | 94.78                 | 93.80  | 92.69               | 91.22           | 88.47                       | 86.26   | 85.87  | 85.84  | 85.35     | 74.87    | 58.38              | 54.22     | 37.68 |

# sourmash gather: precision and recall

McIntyre et al. 2017 Additional File 4: Table S3, modified with an extra column for sourmash gather (Phillip Brooks and Krista Ternus)

# Aims

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data

1. Using scaled MinHash for containment and cardinality estimation
  - Big dataset -> small sketch
2. Fast queries on many MinHash sketches using MinHash Bloom Trees
  - Indexing N sketches
3. Decentralized indices for genomic data.
  - Keeping in sync and sharing data

# Index resilience

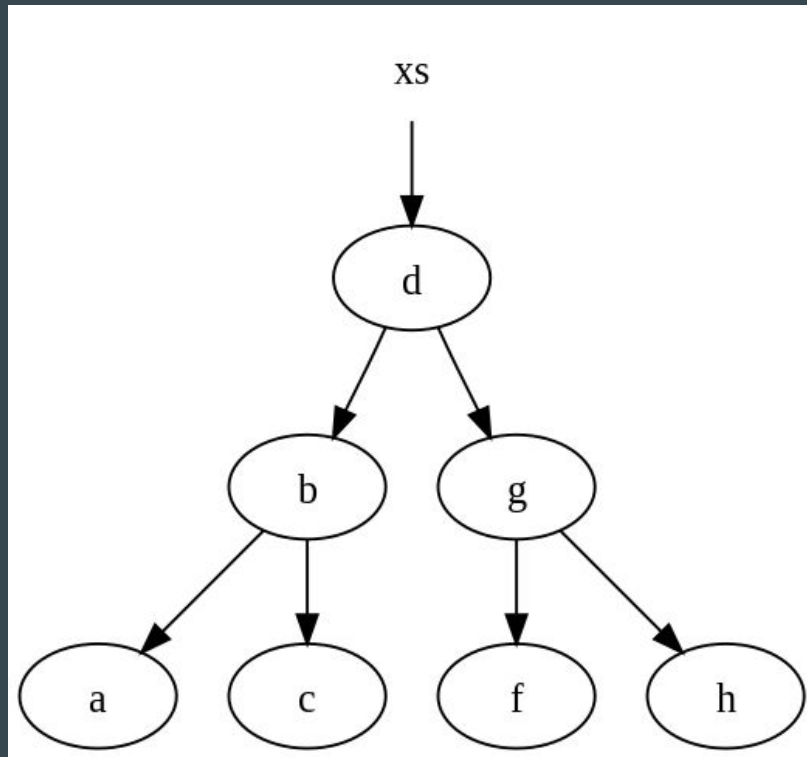
Big dataset -> small sketch

Indexing N sketches

Keeping in sync and sharing data

- Is it possible to rebuild if some information is missing?
- If I update the index, does it invalidate all data from previous version?

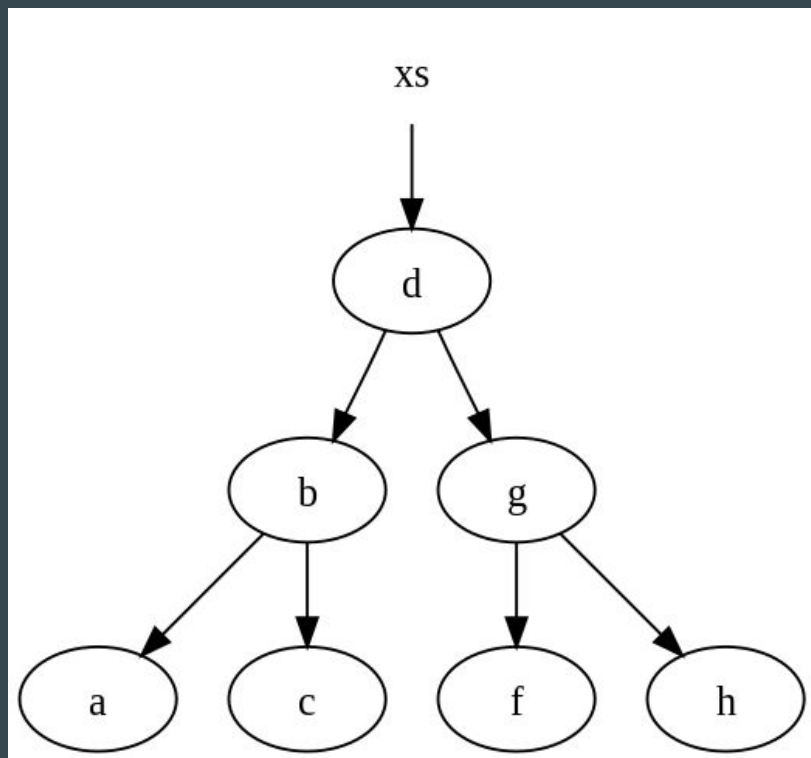
`xs = [a, b, c, d, f, g, h]`



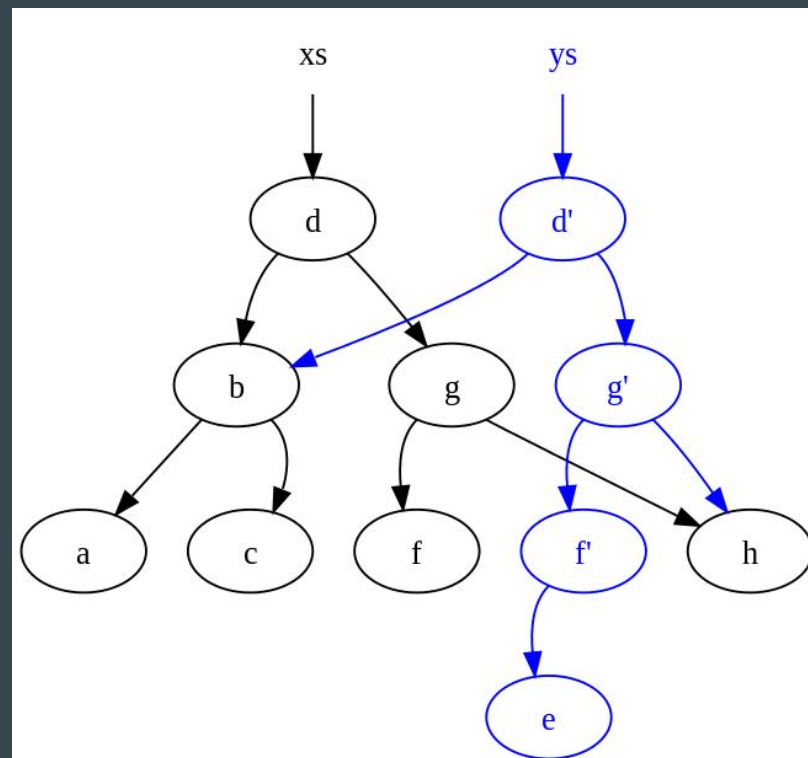
Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data

Driscoll, James R., Neil Sarnak, Daniel D. Sleator, and Robert E. Tarjan. 1989. "Making Data Structures Persistent." *Journal of Computer and System Sciences* 38 (1): 86–124.  
[https://dx.doi.org/10.1016/0022-0000\(89\)90034-2](https://dx.doi.org/10.1016/0022-0000(89)90034-2)

```
xs = [a, b, c, d, f, g, h]
```



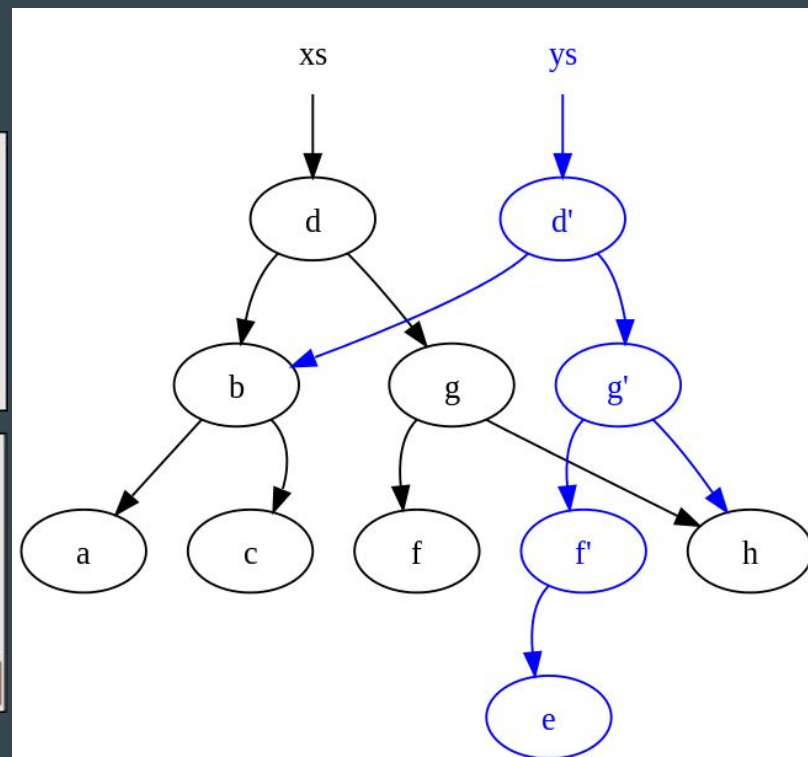
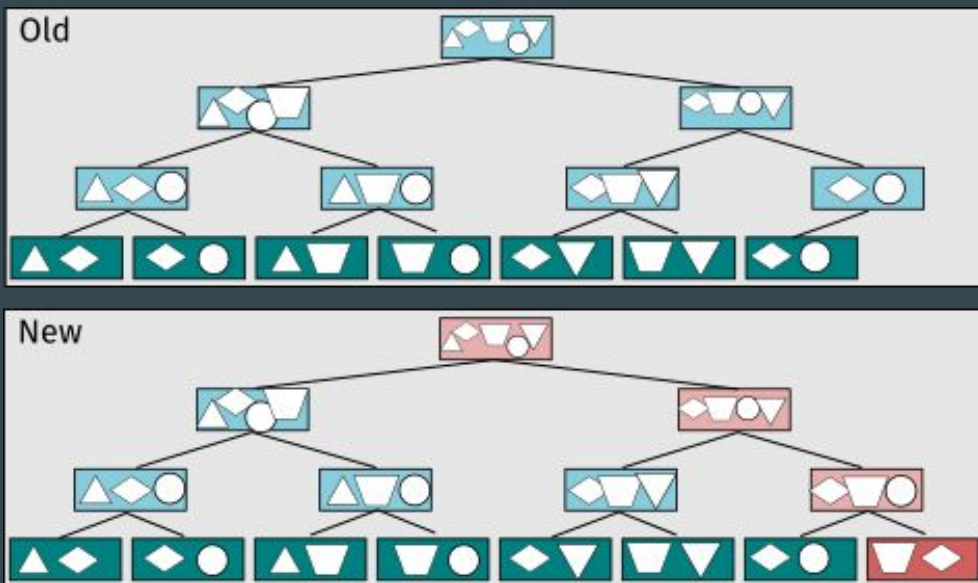
```
ys = insert ("e", xs)
```



Driscoll, James R., Neil Sarnak, Daniel D. Sleator, and Robert E. Tarjan. 1989. "Making Data Structures Persistent." *Journal of Computer and System Sciences* 38 (1): 86–124.

[https://dx.doi.org/10.1016/0022-0000\(89\)90034-2](https://dx.doi.org/10.1016/0022-0000(89)90034-2)





Irber, 2017. Decentralized indexes for public genomic data  
 Poster presented at Recomb 2017.  
<https://github.com/luizirber/2017-recomb>



Each file and all of the **blocks within it** are given a **unique fingerprint** called a **cryptographic hash**.



IPFS **removes duplications** across the network and tracks **version history** for every file.



Each **network node** stores only content it is interested in, and some indexing information that helps figure out who is storing what.



When **looking up files**, you're asking the network to find nodes storing the content behind a unique hash.



Every file can be found by **human-readable names** using a decentralized naming system called **IPNS**.

# IPFS

DHT + BitTorrent + Blockchain + git

# Sharing an SBT

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data

## Current

- JSON + .sbt directory
- One file for each node in the SBT
- 32 signatures + 31 internal nodes + JSON = 64 files

## Distributed

- One JSON file
- Each node content is available on IPFS
- Download content on demand



Lab for Data Intensive Biology



# Decentralized indexes for public genomic data

Luiz Carlos Irber Júnior<sup>1</sup>, C. Titus Brown<sup>1</sup>, Tim Head<sup>2</sup>

lcirberjr@ucdavis.edu, ctbrown@ucdavis.edu, tim@wildtreetech.com

<sup>1</sup>Department of Population Health and Reproduction, University of California, Davis, USA

<sup>2</sup>Head's Wild Tree Tech, Switzerland



@luizirber @ctitusbrown @betatim

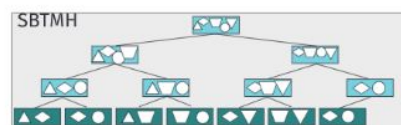


luizirber/2017-recomb

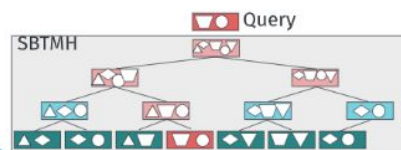


## Introduction

**MinHash** [Broder, 1997] is a technique for **estimating the similarity** of two or more datasets. Expanding on the work pioneered by **Mash** [Ondov et al, 2016] and extended in our library **sourmash** [Brown and Irber, 2016], we calculated signatures for **412 thousand microbial reads datasets** on the **Sequence Read Archive**. To be able to **efficiently search for matches** of these signatures in the **RefSeq microbial genomes database** we developed a new data structure based on **Sequence Bloom Trees** [Solomon and Kingsford, 2016] adapted for **searching MinHash signatures** (named **SBTMH**) to **index signatures** and made it available publicly.



The SBTMH is a **binary tree** where **leaf nodes** are **MinHash signatures** and **internal nodes** are **Bloom Filters**. Each Bloom Filter can be queried for approximate membership of **all the values from its children**, and so the **root node** roughly represents **all the values from all signatures** in the tree.



Searching for **similarity to a query signature** involves checking for **query elements** present in **each internal node**, and if it **doesn't reach the threshold** the subtree is **pruned**. If a **leaf is reached**, it is **returned as a match** to the query signature.

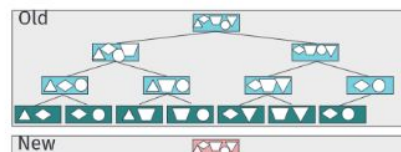
## Saving a SBTMH in IPFS



The SBTMH structure can be encoded as **nodes in a MerkleDAG** and stored in **IPFS** (InterPlanetary File System) [Benet, 2014]. Typical **data archive systems**, like **Amazon S3** or the **NCBI SRA**, stop working when the **central service is down**. **IPFS nodes** can communicate and synchronize data **without requiring a central source**, and they can also **serve data requests among them**, which benefits from local networks and **increases the bandwidth** available for **data transfers**.

The SBTMH behaves like a **persistent data structure** [Driscoll et al, 1989], where new versions of a SBTMH (after new nodes are added or removed) **can share parts of the structure of previous versions**. While this is usually used to avoid duplicating data on pure functional

Signatures calculated from **public datasets** can also be **shared**: by indexing **RefSeq** and **GenBank** and sharing the signatures on IPFS, users can become **curators** by **selecting organisms** of interest and creating **SBTMH indexes** that fit their needs or the needs of a specific area.



Adding a new signature to SBTMH causes **parent nodes to be updated**, but other nodes

Irber, 2017. Decentralized indexes for public genomic data  
Poster presented at Recomb 2017.  
<https://github.com/luizirber/2017-recomb>



Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data

GB Editorial Team *Genome Biology* 2011, 12:402  
<http://genomebiology.com/2011/12/3/402>



## EDITORIAL

# Closure of the NCBI SRA and implications for the long-term future of genomics data storage

GB Editorial Team\*

The National Center for Biotechnology Information (NCBI) in the US recently announced that, as a result of budgetary constraints, it would no longer be accepting submissions to its Sequence Read Archive (SRA) and that over the course of the next year or so it would slowly phase out support for this database (<http://www.ncbi.nlm.nih.gov/sra>). There seems to be a certain amount of confusion in the community about what effect this decision will have. At *Genome Biology* we feel that the free availability of data is an important concept for science, so we asked the views of various interested people on what the short-term implications of this announcement will be, and also how they envisaged the future of data storage in the long term. These people include those involved in the running of the databases (David Lipman (DL) from the NCBI and Paul Flicek (PF) from the European Bioinformatics Institute (EBI)) and users of the data stored in the database as well as data producers (Steven Salzberg (SS) from the University of Maryland, Mark Gerstein (MG) from Yale University and Rob Knight (RK) from the University of Colorado).

accept submissions doesn't mean that the SRA is closing, merely changing and the European Nucleotide Archive (ENA) at EMBL-EBI will remain. The NCBI's decision was based on budgetary constraints. It should be noted that most people don't realize that storage space is only a minor fraction of the budget of the database; the bulk of the cost is associated with the staff who maintain the database, process the submissions, develop the software and so on.

**SS:** From the outside, it appears that the SRA is closing because of NIH budgetary considerations. One problem is that the amount of sequence being generated is growing at an extraordinary rate, probably faster than increases to the budget. My group uses the SRA a lot. Due to the nature of our work, we rely on it maybe more than others. We download data reasonably frequently, but because of the size of the datasets we try not to do it too often.

**RK:** The SRA was widely disliked by a lot of users, in particular because it was hard to get data. Partly that was because of poor standards for metadata associated with the data entries. This makes it hard to find the

Closure of the NCBI SRA and Implications for the Long-Term Future of Genomics Data Storage.  
<https://doi.org/10.1186/gb-2011-12-3-402>

# wort - public indices

Big dataset -> small sketch

Indexing N sketches

Keeping in sync and sharing data

- From one index to many indices
- Once we have many indices, how to search them?
  - And maintain them updated?
- wort: Infrastructure for
  - Calculating signatures
  - Submitting signatures
  - Building indices
  - Searching indices
- Why?
  - We started calculating signatures for multiple public databases
    - GenBank, RefSeq, SRA, IMG...
  - It's pretty much spread into 3-4 different machines
    - ... that only a few people have access
  - The public databases are constantly being updated
    - Put a new index on S3 every once in a while is clunky





**Ole K. Tørresen** @Tierhon · Feb 19

Is sourmash named after the process in the distilling industry or in beer brewing?



1



**Titus Brown** @ctitusbrown · Feb 19

whiskey. @luizirber now has a wort repo too :)



2



1



**Luiz Irber** @luizirber · 19h

seems like I'm learning the important things in this lab 😊



1



3



**Ole K. Tørresen**

@Tierhon

Follow

Replying to @luizirber @ctitusbrown

There is more to life than whiskey and beer.  
Wine for instance.

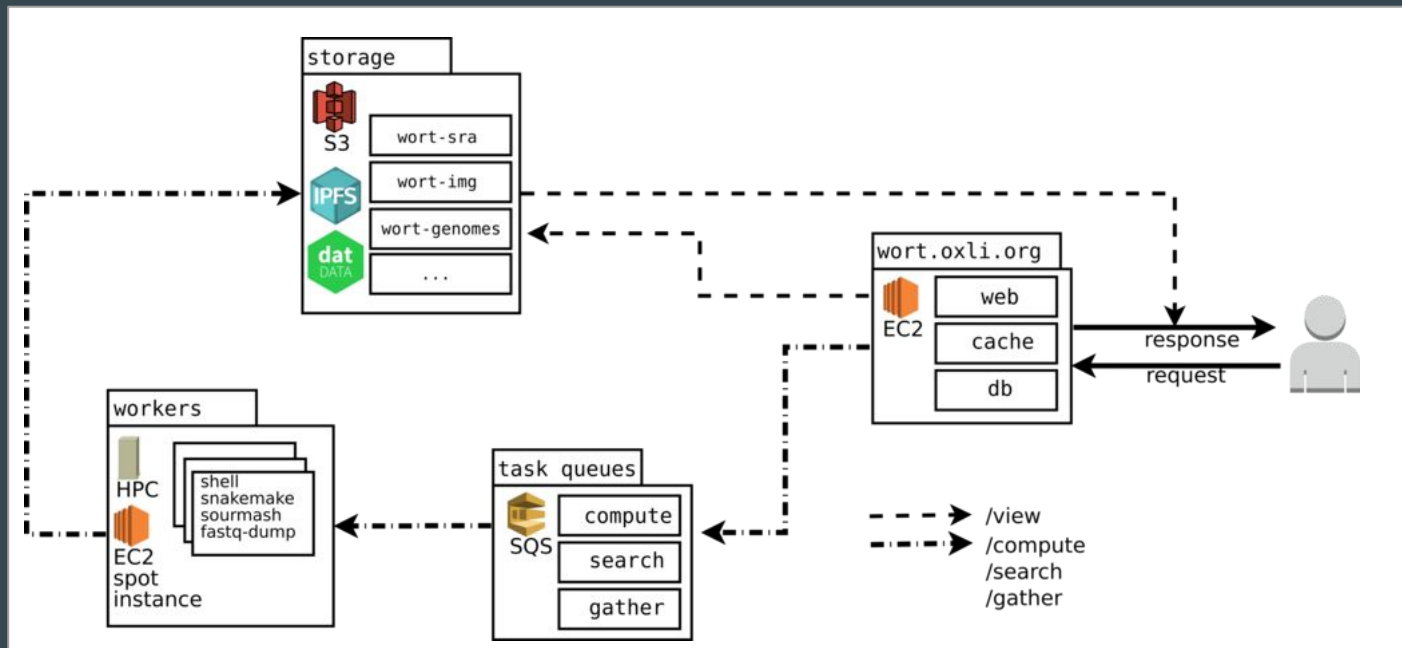
8:52 AM - 20 Feb 2018

# wort

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data

<https://wort.oxli.org>

- /view/
- /compute/
- /submit/



Irber, 2018. Building decentralized indexes for public genomic data  
Poster presented at Biological Data Science 2018.

<https://github.com/luizirber/2018-biods>

# wort CLI

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data

- `wort view -d <database> <dataset_id>`
  - `wort view -d sra ERR660673`
  - `wort view -d img 2522125045`
- `wort submit <signature> -d <database> -i <dataset_id> -t <token>`
  - `wort submit -d sra -i ERR660673 -t xxxxxx`
- Access token needed for ‘write’ operations (incur more \$\$\$)
- it’s Rust! <https://crates.io/crates/wort>
  - Single binary is convenient. Download one file, run it.
  - No need for conda, but it is conda-compatible

# wort CLI - why?

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data

- Currently just a thin wrapper over the HTTP API
- Why not use cURL/Requests directly?
  - More flexible, but more complicated
- End user unlikely to care, as long as it works...
- ... which allows me to change how the CLI works internally without them noticing!
  - Like, using IPFS underneath 🐱
- The biggest roadblock to adoption of IPFS is how “alien” they are
  - Users don’t care about the technological details
  - Engineers try to sell system based on technological details
    - (yes, I see the irony)
- But, if you have an useful tool already...
  - (let’s hope I can pull the transition without anyone noticing!)

# wort - future plans?

Big dataset -> small sketch  
Indexing N sketches  
Keeping in sync and sharing data

## Future Work

The current architecture is a proof of concept, with a **concrete, then abstract** approach: have something working first with a public API, then **refactor** and **generalize**. While it is deployed on AWS it can also be **run in other cloud providers**, and the next goal is to replace most of the **task queue and communication** with **P2P technologies**, using more of IPFS and dat (and not only as file storage).

The **WebAssembly** support in **sourmash** also allows doing more data processing in the browser, instead of transferring large datasets to the server. Currently wort is more focused on the API and command line usage, but more functionality will be added to the web frontend.

NCBI provides an **alpha feature** based on **STAT** to report the **taxonomic composition of reads** within a sequencing run. This analysis can also be done with **sourmash gather**, and a **browser extension** can overlay these results in the SRA Run Browser. This extension idea is similar to what the **BioJupies** project [Torres, 2018] does for RNA-Seq datasets.

Any **public database** can be store and queried using **wort**, and we intend to add more over time.

# Schedule

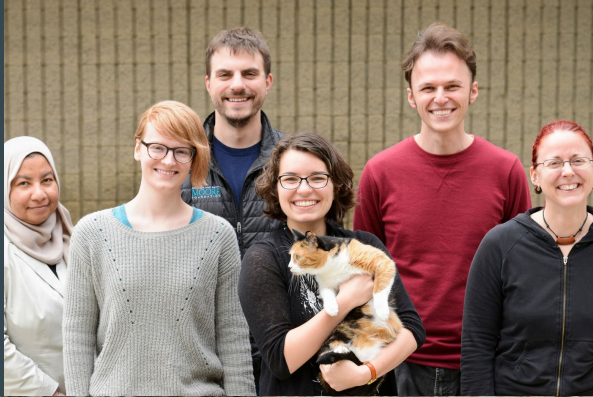
- Early May: thesis committee formed
- Committee meetings:
  - June - Aim 1 chapter
  - September - Aim 2 chapter
  - December - Aim 3 chapter
- December: First final draft

## Products:

- sourmash 2.0 release article (F1000)
- Gather paper
  - Benchmarks
- Scaled minhash paper
- The Great Oxidation



# Thanks!



Lab for Data Intensive Biology



Thanks!





# Thanks!

