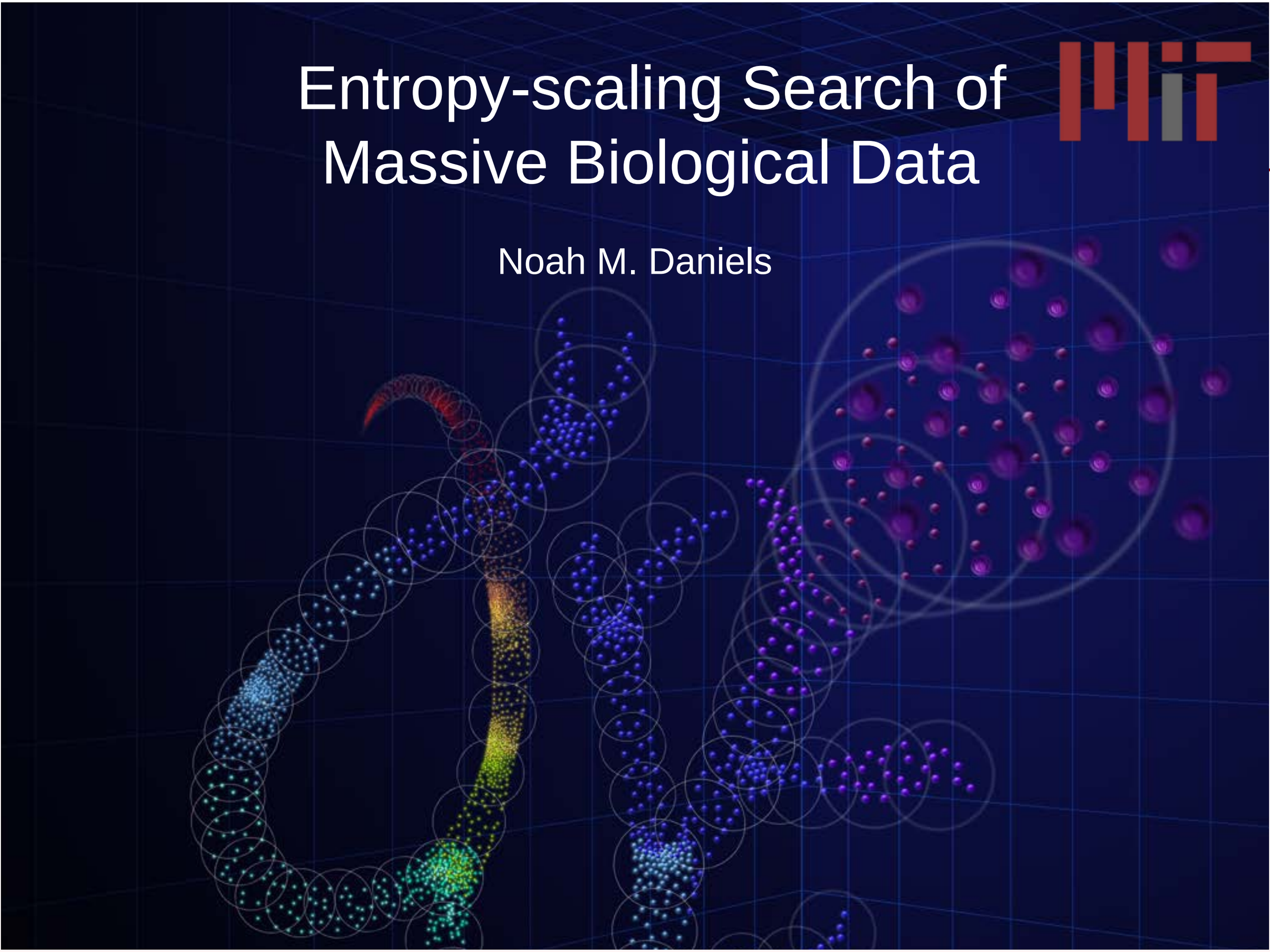
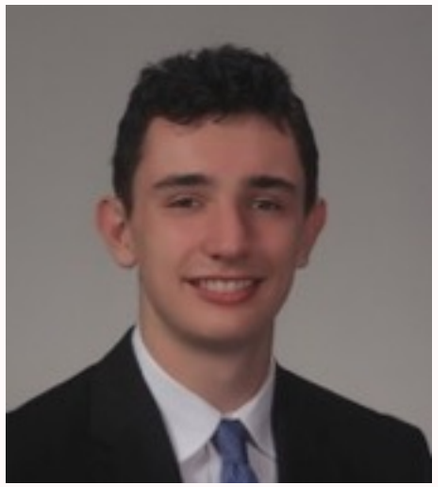
Entropy-scaling Search of Massive Biological Data

Noah M. Daniels



Bonnie Berger

Y. William Yu

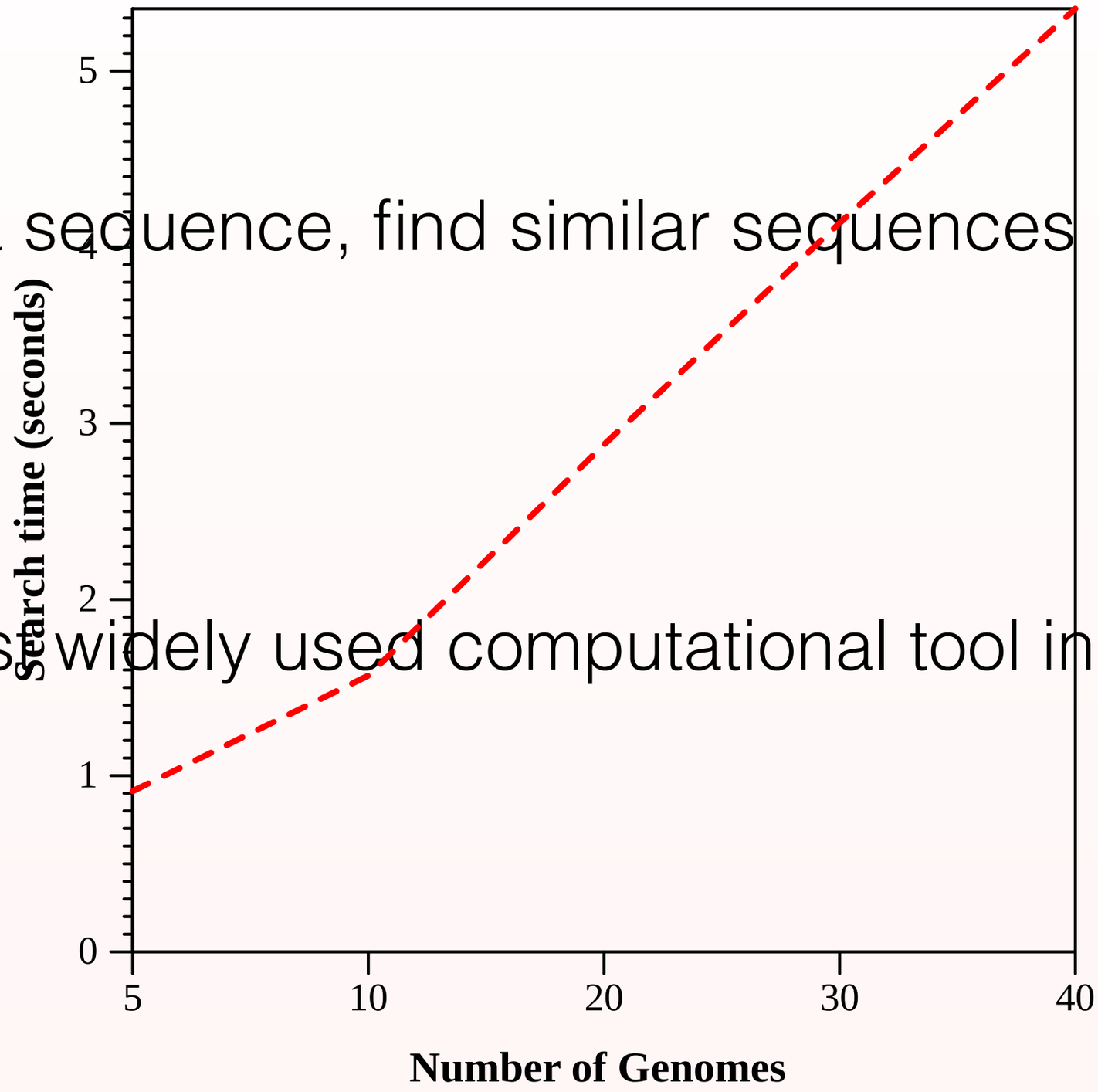


David Christian Danko

Andrew Gallant, Tufts
Michael Baym, Harvard
Deniz Yorukoglu, MIT
Jian Peng, UIUC

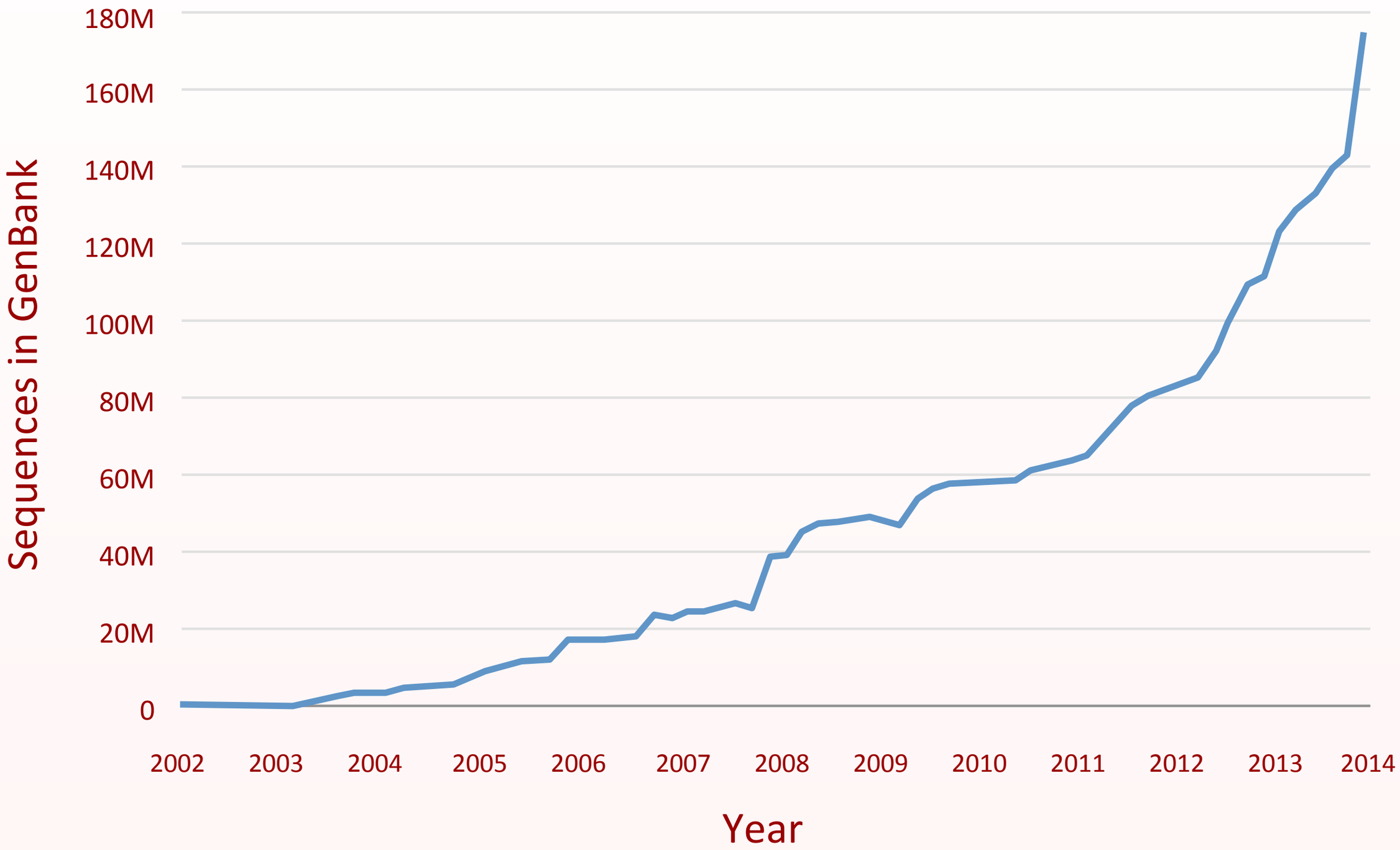
BLAST: biological sequence search

Given a sequence, find similar sequences in a database



The most widely used computational tool in all of biology

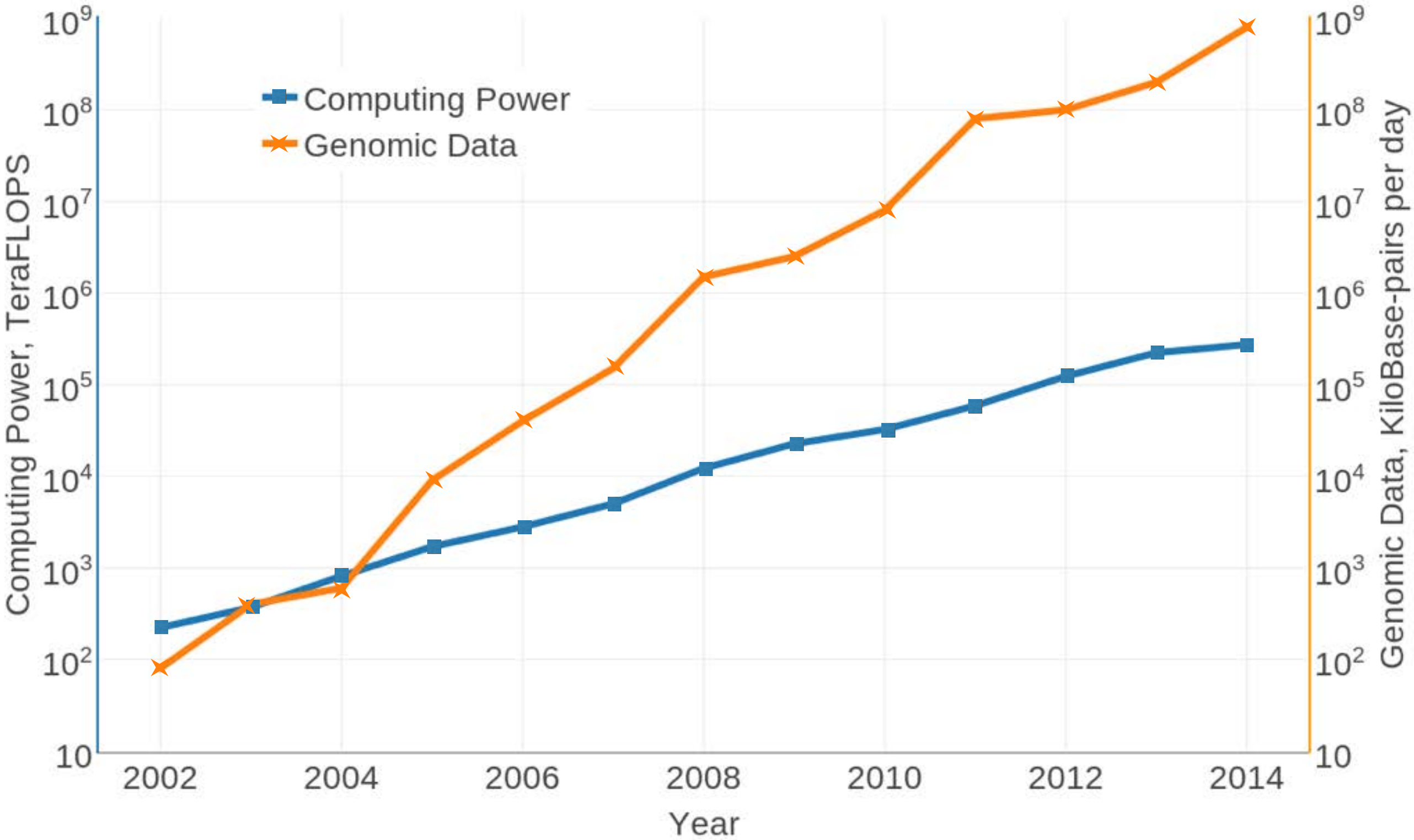
Genomic data are growing exponentially



A bigger cloud?

Still constrained by Moore

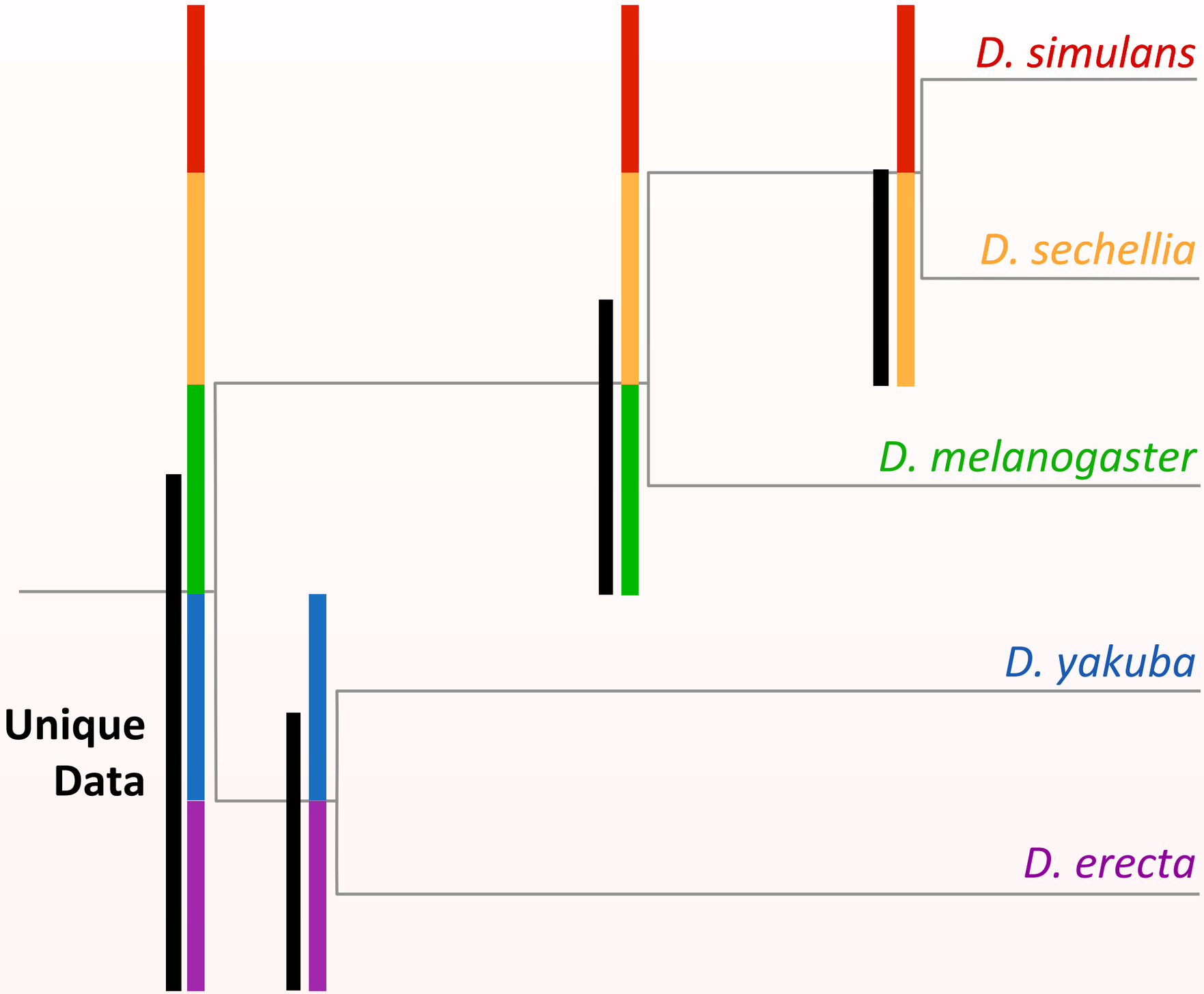
Genomic data are growing exponentially



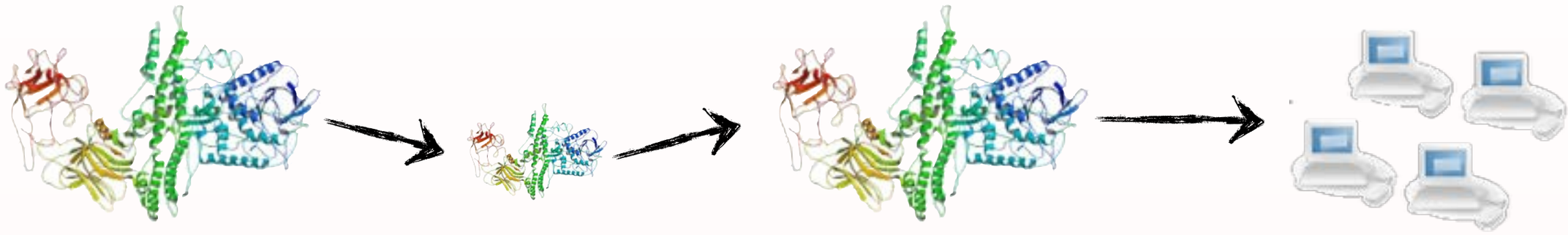
Strategy: exploit structure

- Dimensionality reduction (e.g. PCA, DCA)
- Indexing (e.g., FM-index)
- K-mer based approaches (e.g., Kingsford)
- Compressed sensing (e.g., M. Linial)
- Low-dimensional polytopes (U. Alon)
- Tree decomposition (J. Xu & Berger)
- Isorank for network alignment (Berger lab)

Redundancy in data



Compressive Genomics



Compute in compressed space

Loh, Baym, Berger 2012
Daniels, Gallant, Peng, Baym, Cowen, Berger 2013

Our contributions

Formalization of compressive genomics

Application to metagenomics

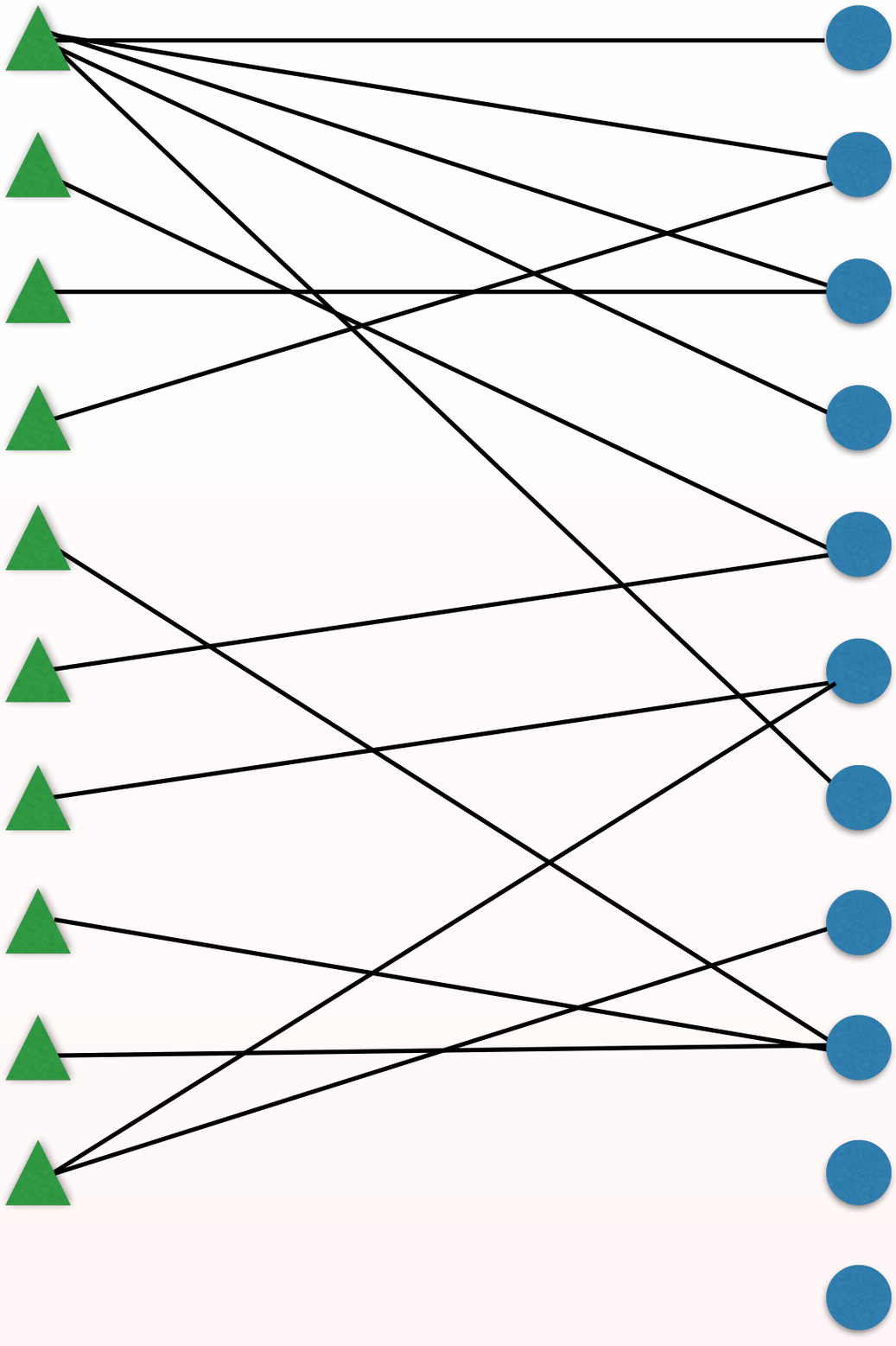
Application to chemogenomics

Relationship to genome compression

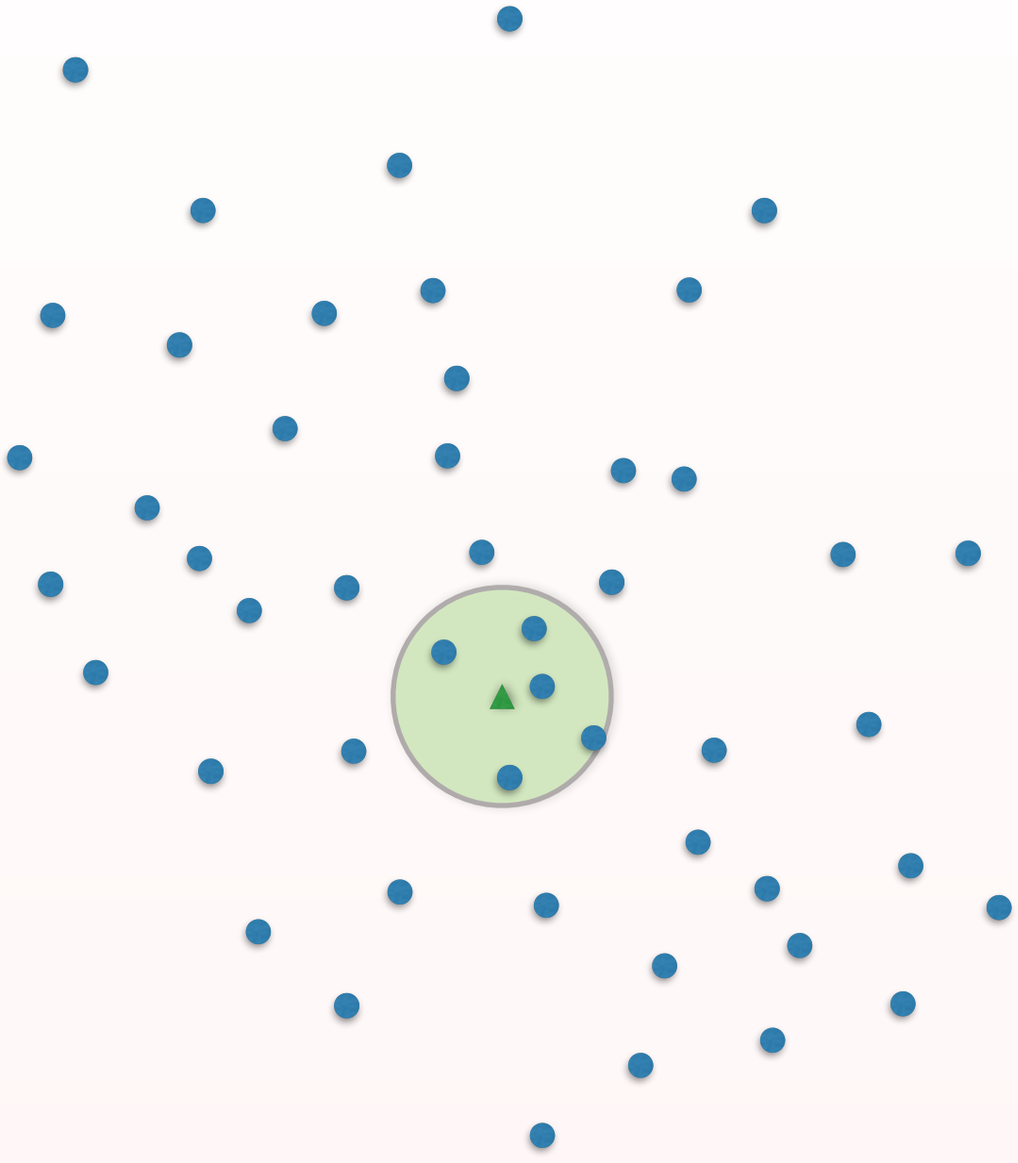
Quality score compression

Generalization: approximate search

Given a database, for any new observation of the same type, which are the most similar?



Queries



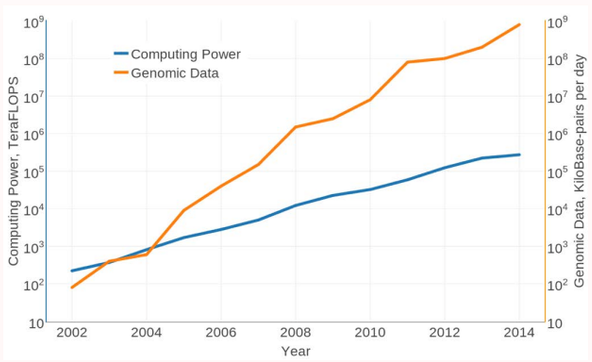
Database

Naïve complexity bounds

m queries, database of size n

Approximate matching $O(mn)$

Approximate search $O(n)$



…but n is growing exponentially!

$$O(\text{metric entropy} + \text{output size} \times \text{scaling factor})$$

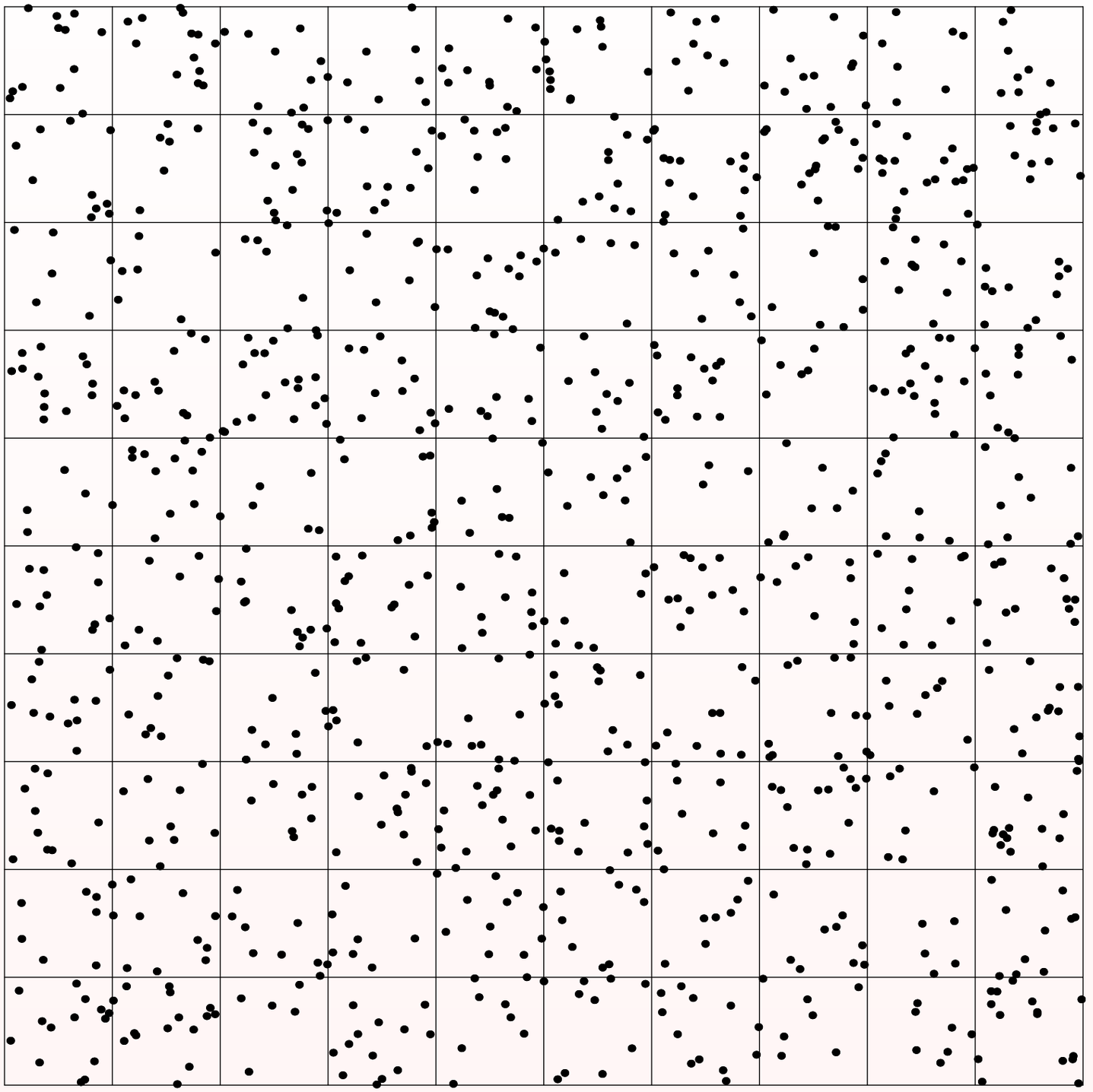
much smaller than n

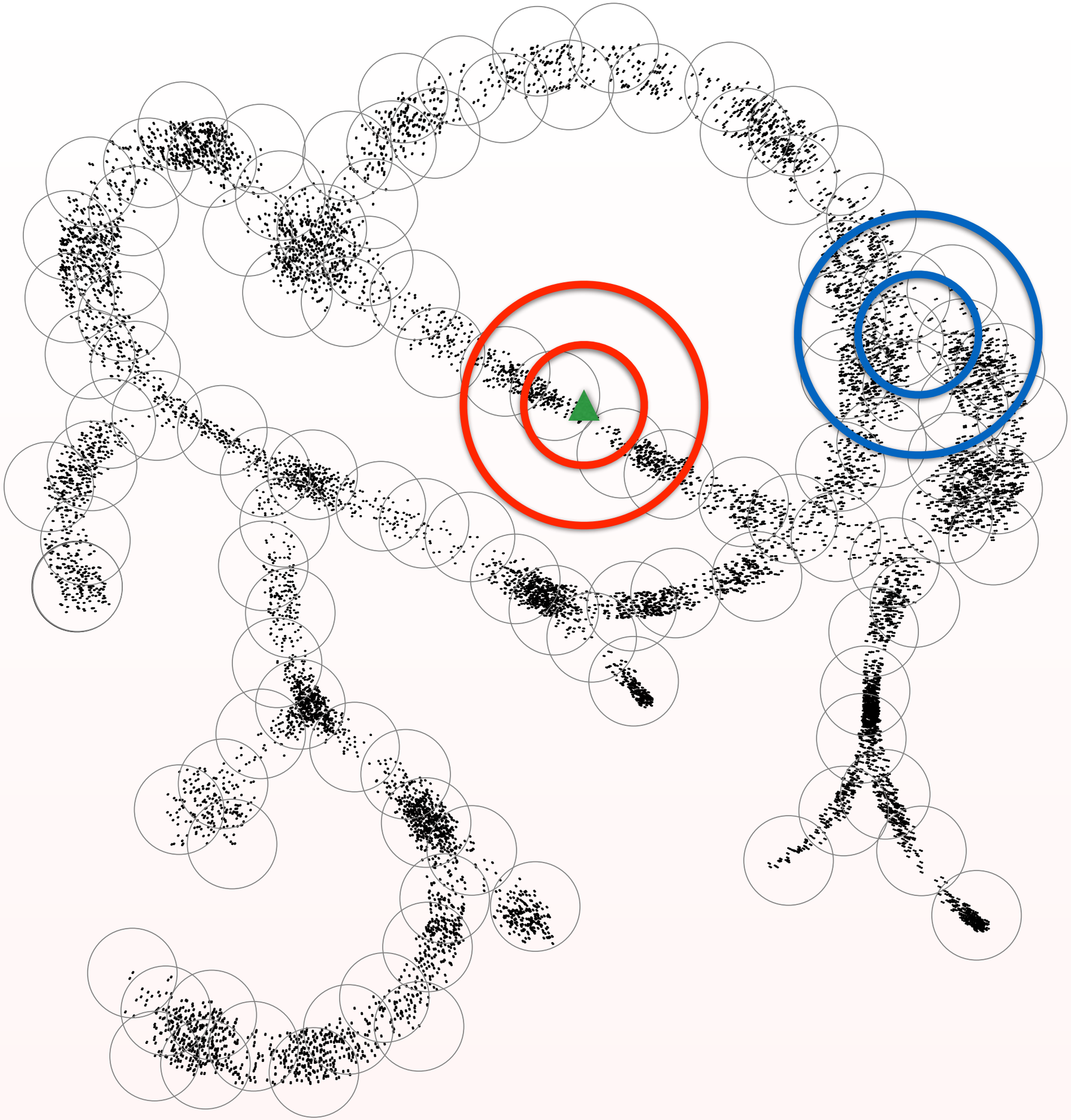
Yu, Daniels, Danko, Berger - Cell Systems 2015

Queries

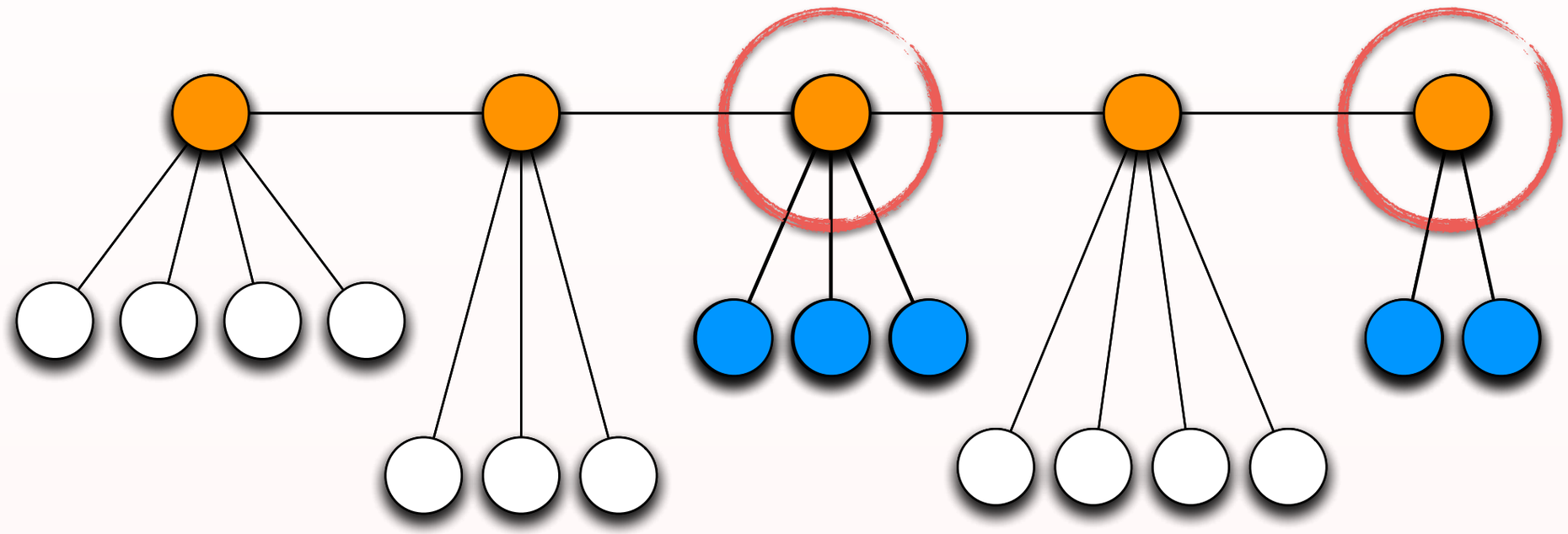
Database

Grid search





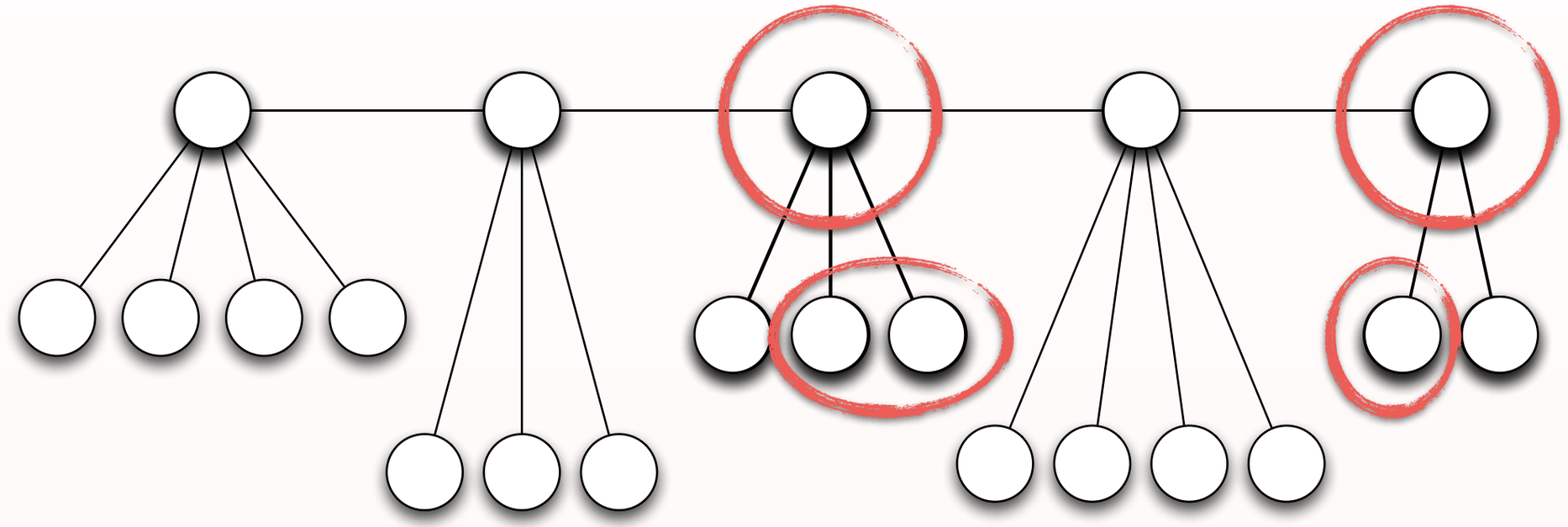
The search tree

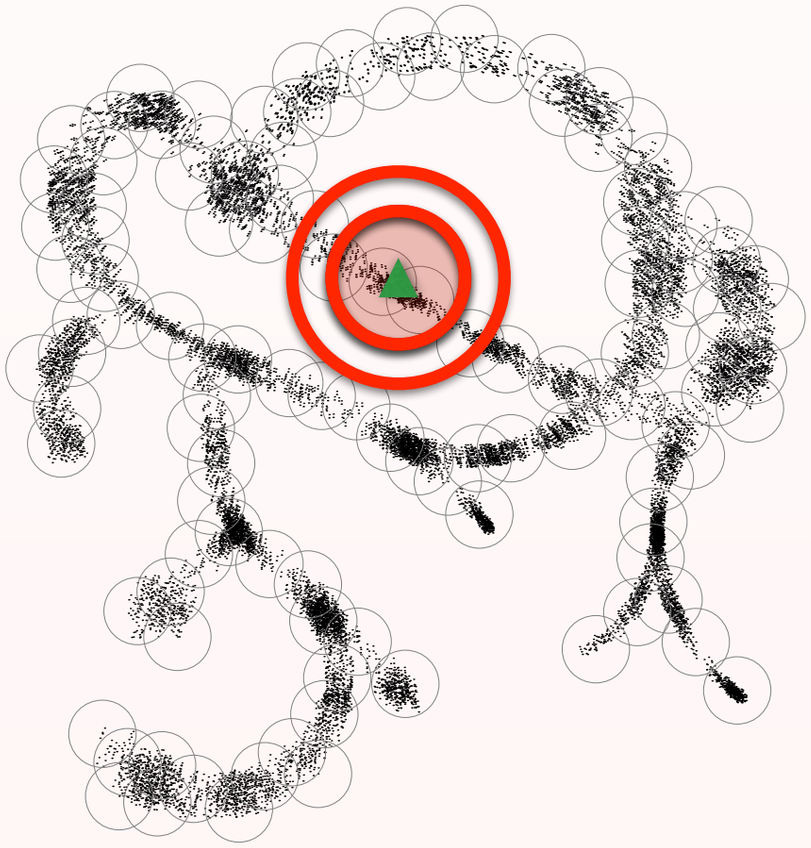


coarse search

$$O(coarse + fine)$$

Entropy-scaling search





$$O\left(\underbrace{N_{r_c}(D)}_{\text{\# clusters}} + \underbrace{|B_D(q,r)|}_{\text{output}} \underbrace{\left(\frac{r+2r_c}{r}\right)^d}_{\text{scaling factor}}\right)$$

Metagenomics

Metagenomic Reads

```
>read1
ACGTGGCTATCAACTCGCTAACTAA
>read2
ACGTGGCTATCAACTCGCTAACTAA
>read3
ACGTGGCTATCAACTCGCTAACTAT

...

>readk
TCGTCGAACTACATTACATTTACAG
>readk+1
TCGTCGAACTACATTACAAATACAG

...

>readm
GCTCGGACTATATATAGGCCTAGAA
...
```

Translated ORFs

```
>read1-1
TWLSTR
>read1-2
RGYQLAN
>read1-3
VAINSLT
>read1-r1
LVSELIAT
>read1-r2
LAS
>read1-r3
RVDSH
>read2-1
SSNYITFT
>read2-2
RRTTLHLQ
>read2-3
VELHYIY
>read2-r1
CSST
>read2-r2
CKCNVVRR
>read2-r3
VNV
...
```

Protein Database (NR)

```
>protein 1
MRVLVINSGSSSIKYQLIEM
>protein 2
EILGKKLEELKIITCHIGNGASVAAVKY
>protein 3
LKKLLESSGCRLVRYGNILIG
...
```

roblem: Giv ds from a microbiome,
match each read with similar proteins.

BLASTX [Altschul, et al. 1990]
RapSearch2 [Zhao, et al. 2012]
DIAMOND [Buchfink, et al. 2015]

Metagenomic Inquiry Compressive Acceleration

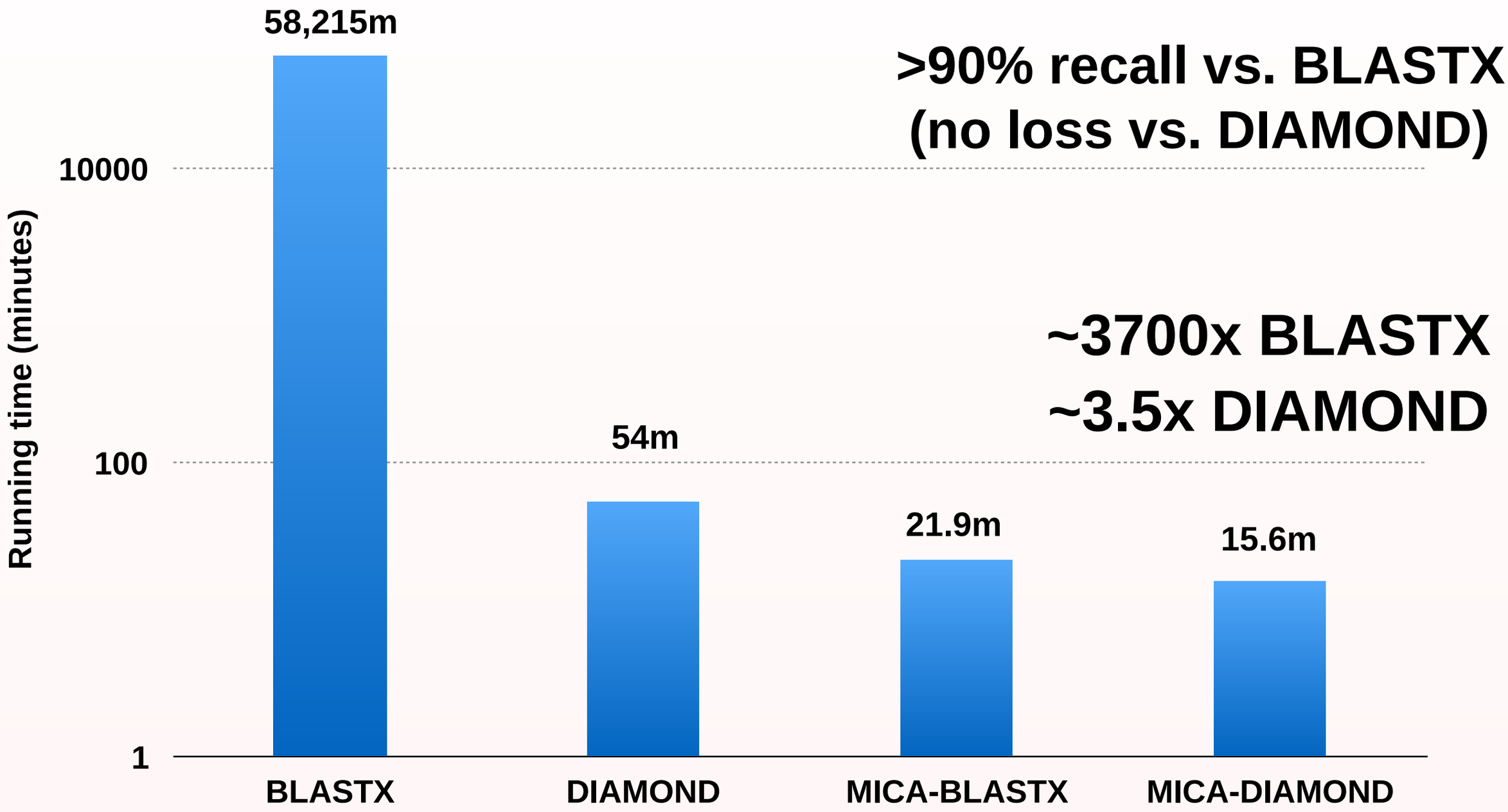
gems.csail.mit.edu

Compressive acceleration applied to metagenomics

Coarse search: DIAMOND

Fine search: DIAMOND or BLASTX

Metagenomic Inquiry Compressive Acceleration



>90% recall vs. BLASTX
(no loss vs. DIAMOND)

~3700x BLASTX
~3.5x DIAMOND

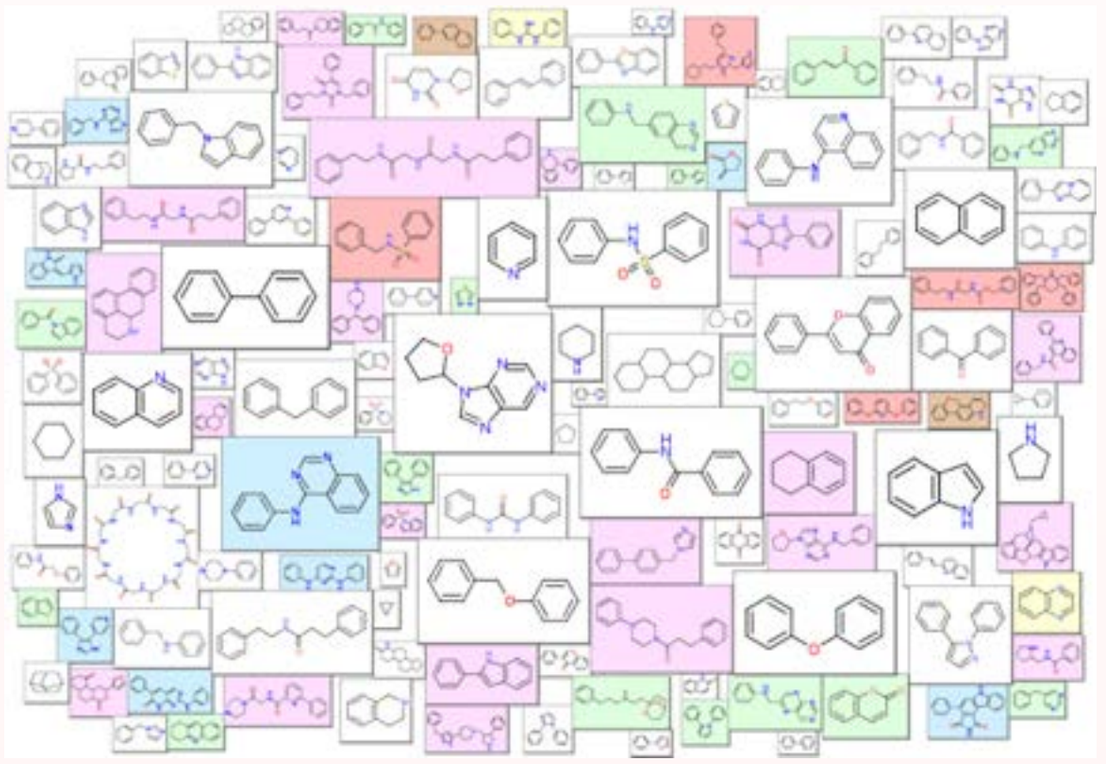
American gut microbiome project NGS reads, searching NCBI NR database

Rational drug design

Design molecules to fit targets ($$$)

Repurpose old drugs for new targets

High-throughput screening

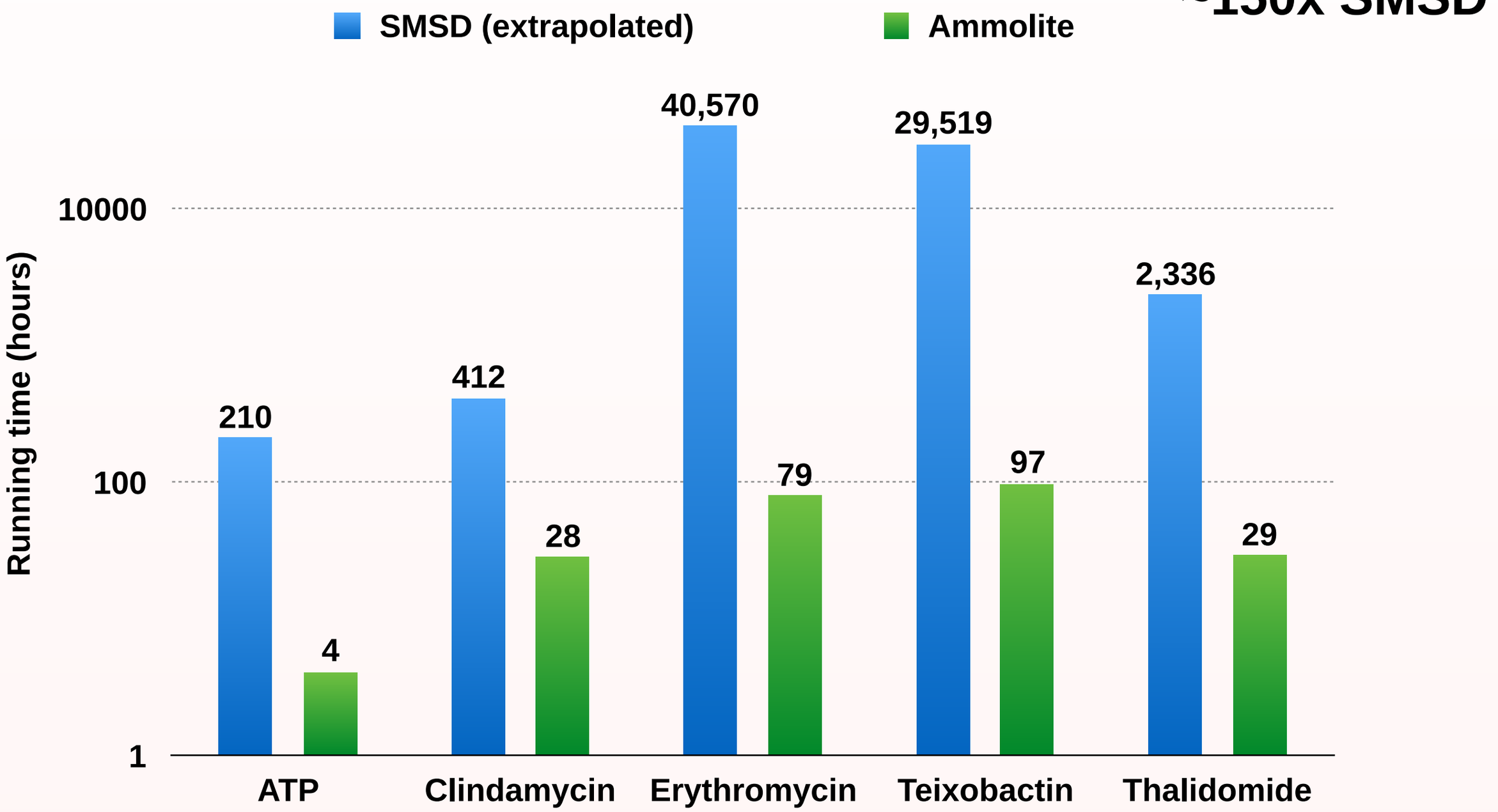


Ammolite

Accelerated molecule search

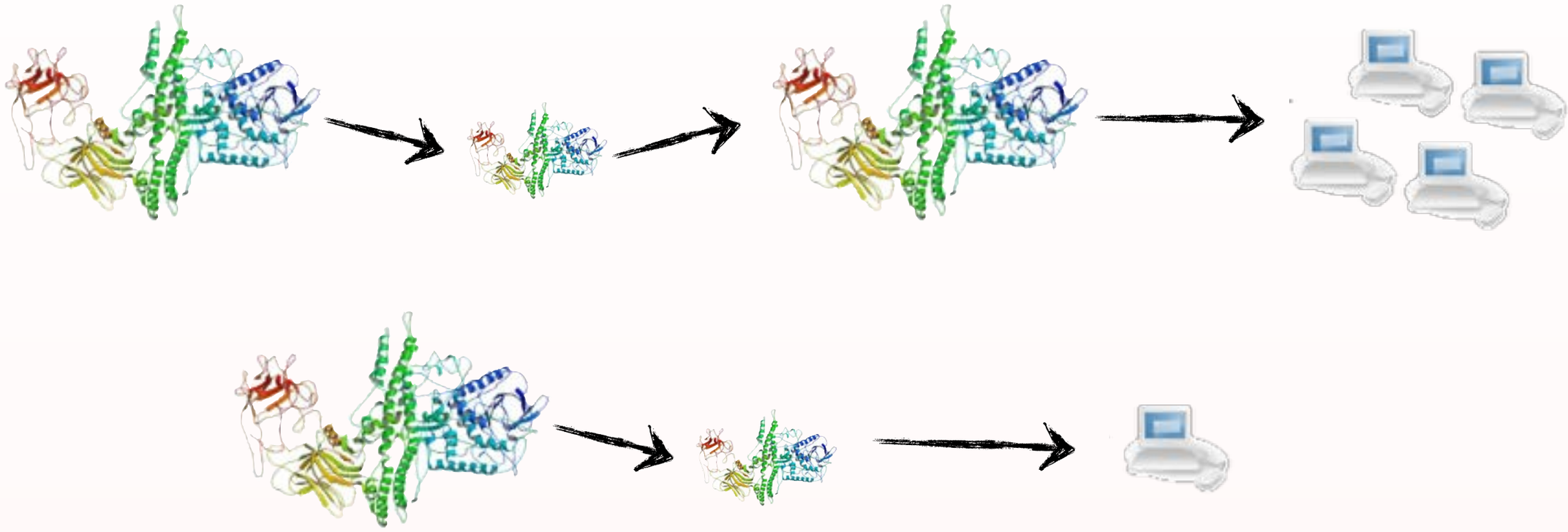
>90% recall vs. SMSD

~150x SMSD



Searching 10/2013 PubChem (47 million compounds)

Compressive Genomics



Compute in compressed space

Standard compression doesn’t respect the data structures!

Search needs inexact matches

How does our compression work?

Seed Table

AAAA AAAC
AAAD AAAE
…
YWWW WWWW

STAQEPKSAEDSLRARD

--DSLRARD

QRDSVNARD

LQSTAQEPKSAEQRDSVNARDRQRNVIIAQE

LQ QR VN

Coarse Database

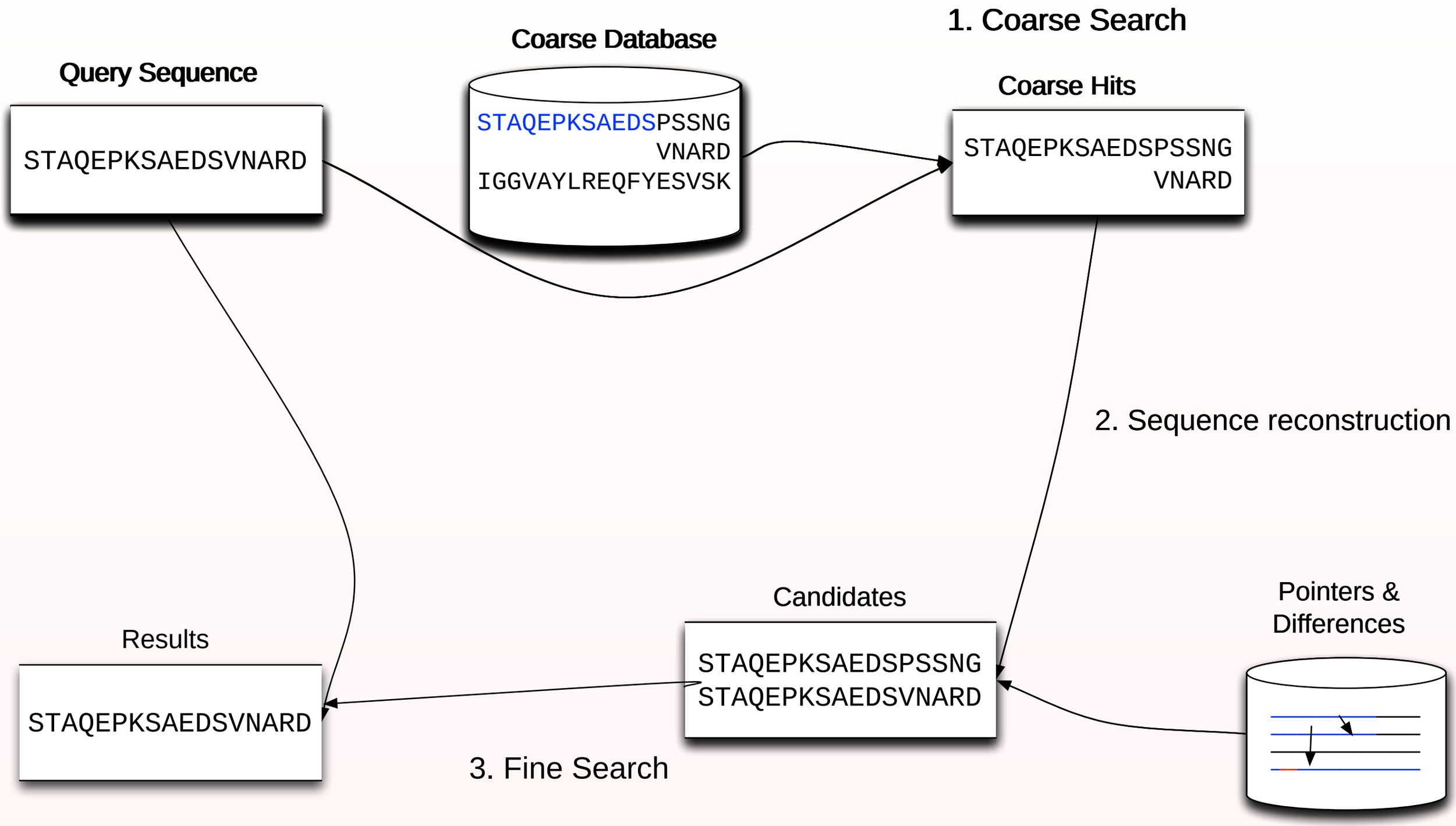
STAQEPKSAEDSLRARD

RQRNVIIAQE

Link index

Lossless compression!

How does our search work?



Why should this be faster?

Sequences in the coarse database are:

- long enough to not be degenerate
- dissimilar to each other
- similar to the sequences they represent

BLAST-searchable, but small

Having our cake & eating it, too

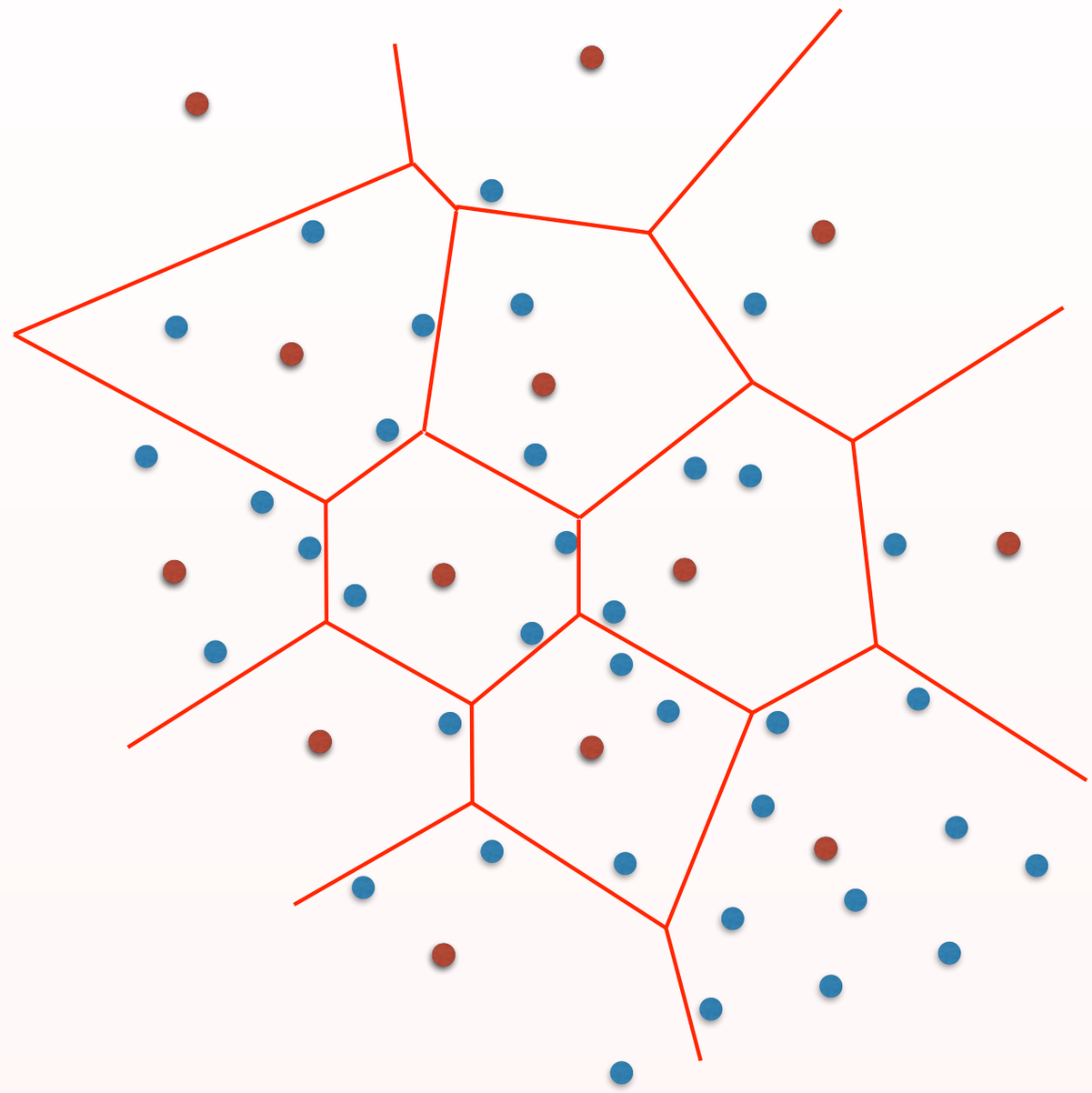
Small, if any, storage-compression tradeoff

Data structures that allow certain computations to be carried out in compressed space

Avoids:

- Intermediate storage of decompressed data for analysis
- Time spent decompressing
- Time spent computing on redundant data

Database



How?

Representative data is available to applications

Not Huffman coded

or

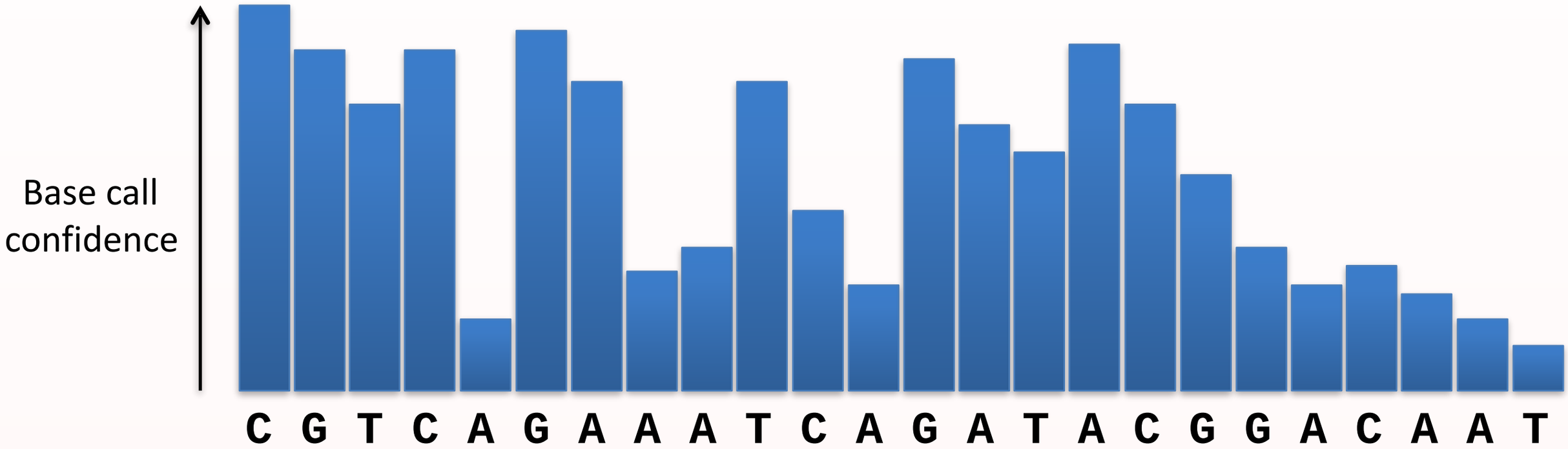
Indexed Huffman-coded format (e.g. BZGF) indexed appropriately (e.g. k-mers)

or…. your ideas?

Rest of data available for post-processing (decompress it)

A detour into quality score compression

The Quality Score Problem



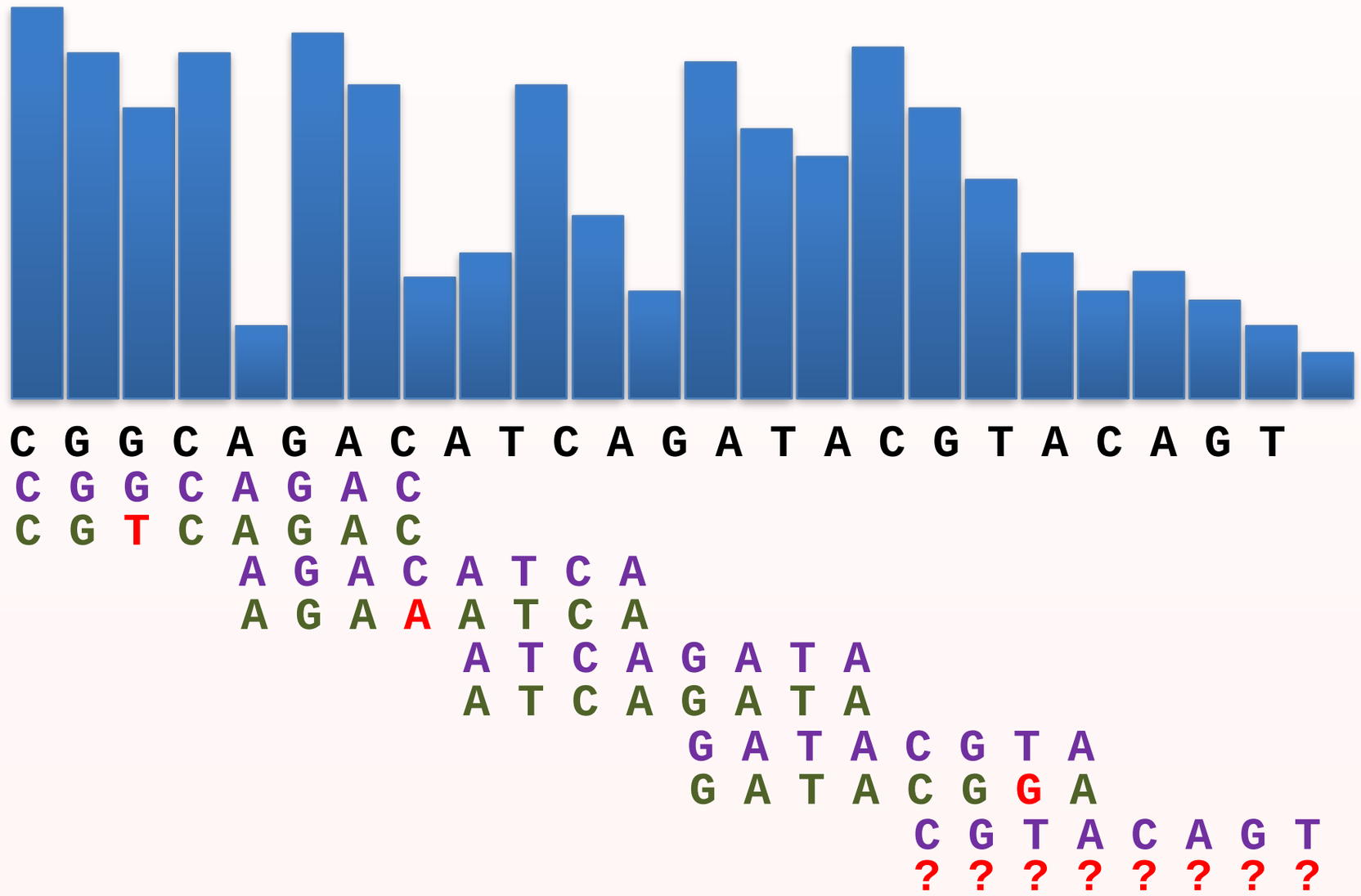
Given reads from a genome, can we **efficiently** compress quality scores while maintaining or even improving accuracy?

(Yu, Yorukoglou, Peng, Berger; Nature Biotechnology, 2015; RECOMB, 2014)

Quartz algorithm

Preprocessing

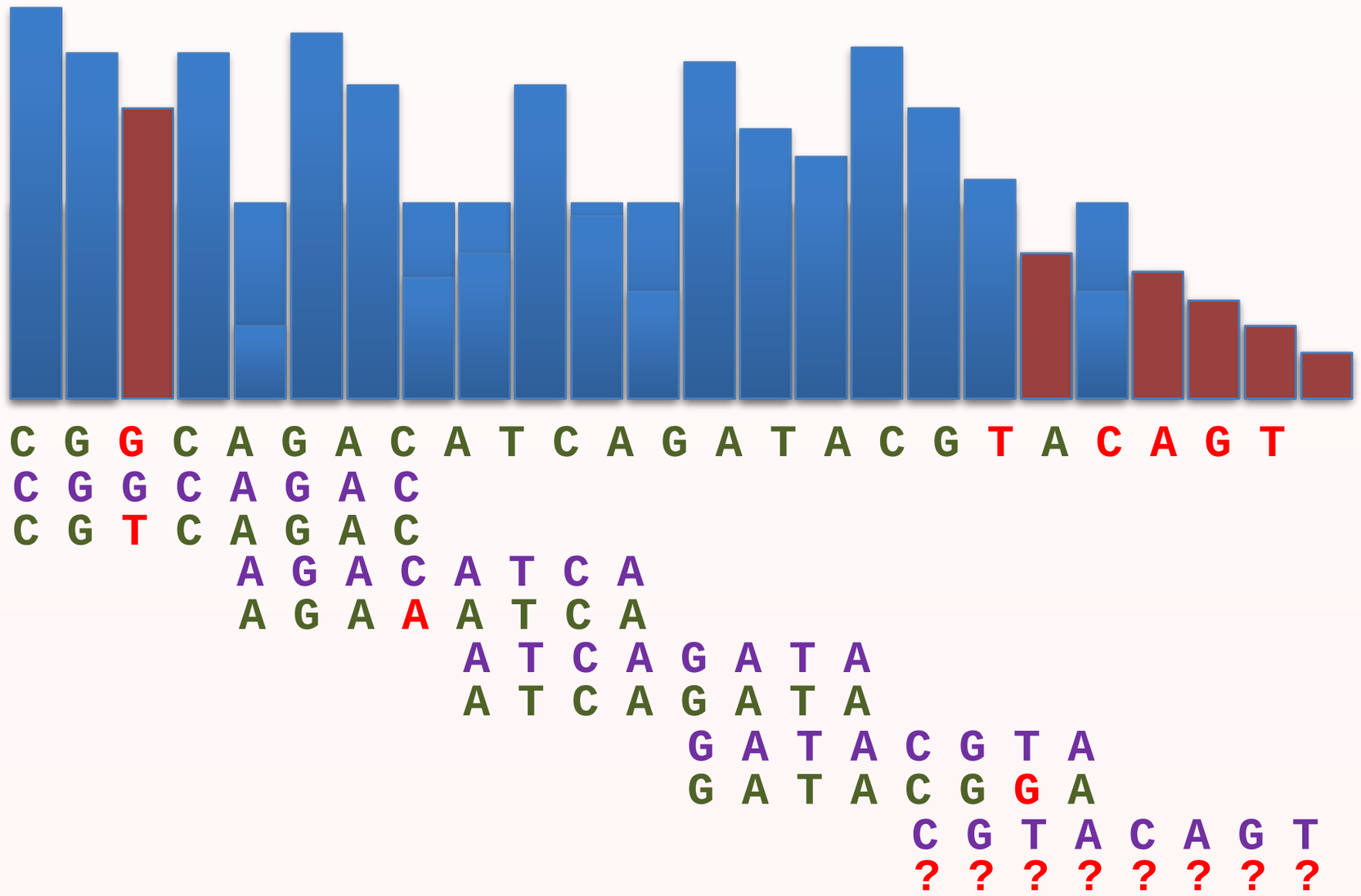
Compression



Quartz algorithm

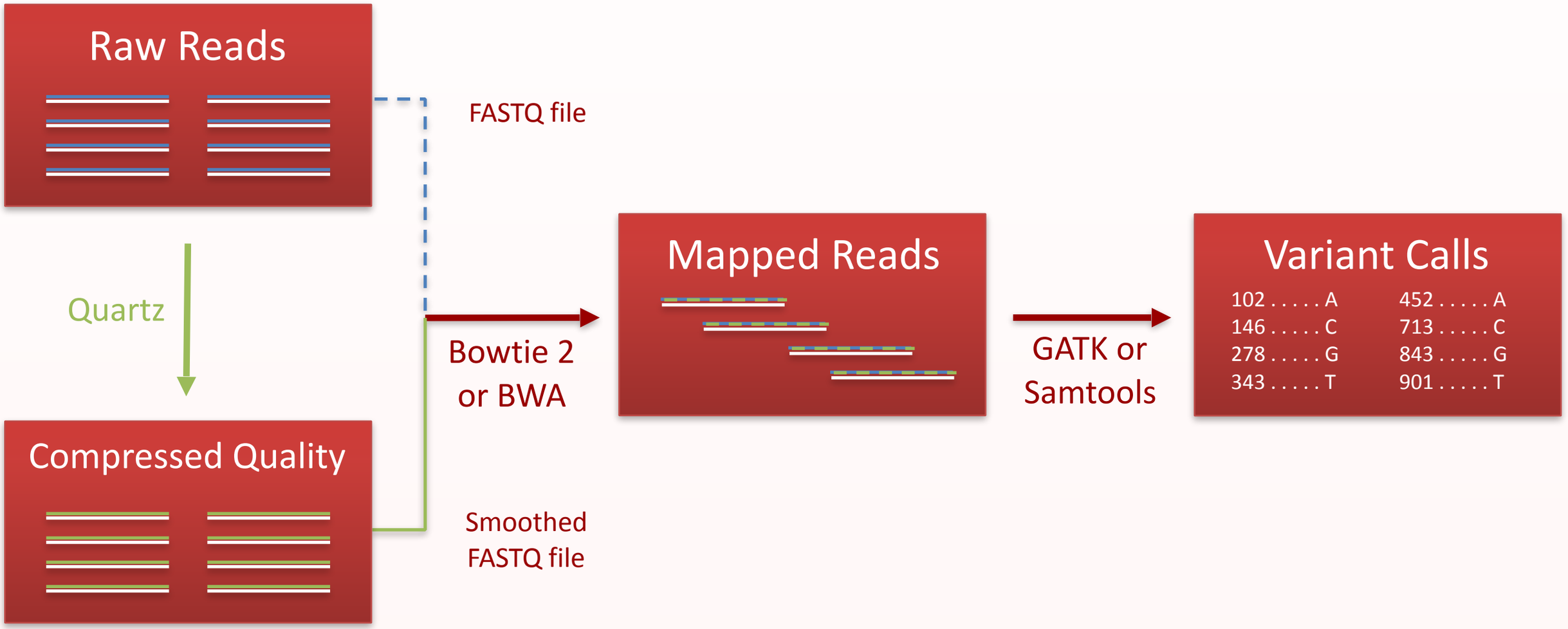
Preprocessing

Compression



Quartz Genotyping Pipeline

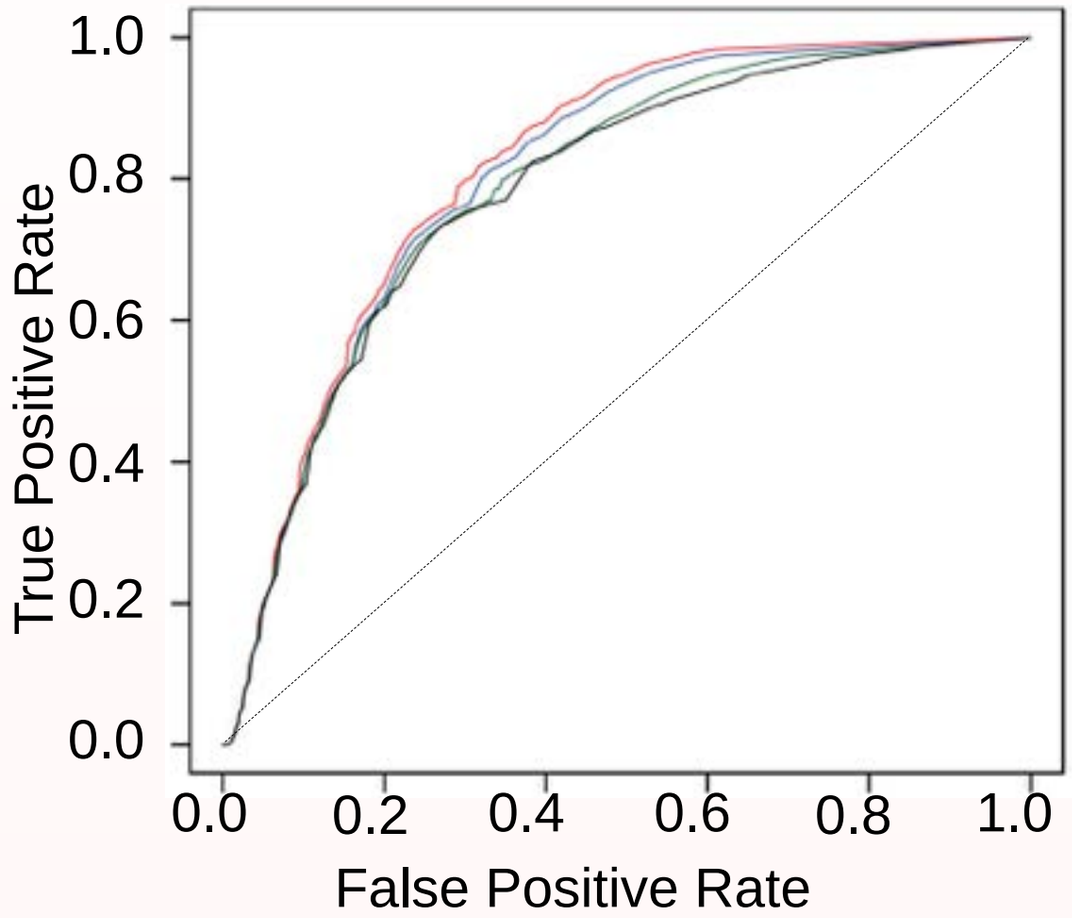
K-mer Based Quality Score Compression



Result Highlights

Comparison with other methods

Method	Bits/Q	Time (s)	Area Under ROC Curve
Uncompressed	8	N/A	0.8254
Quartz	0.3564	**2,696**	0.8288
QualComp	0.5940	33,316	0.8053
Janin et al.	0.5376	164,702	0.8019



Quartz is orders of magnitude faster

Summary

- Lossless compression respecting data structures
- Accelerate operations such as search
- No longer a strict space/time tradeoff
- Formal bounds based on entropy
- Lossy compression of quality scores