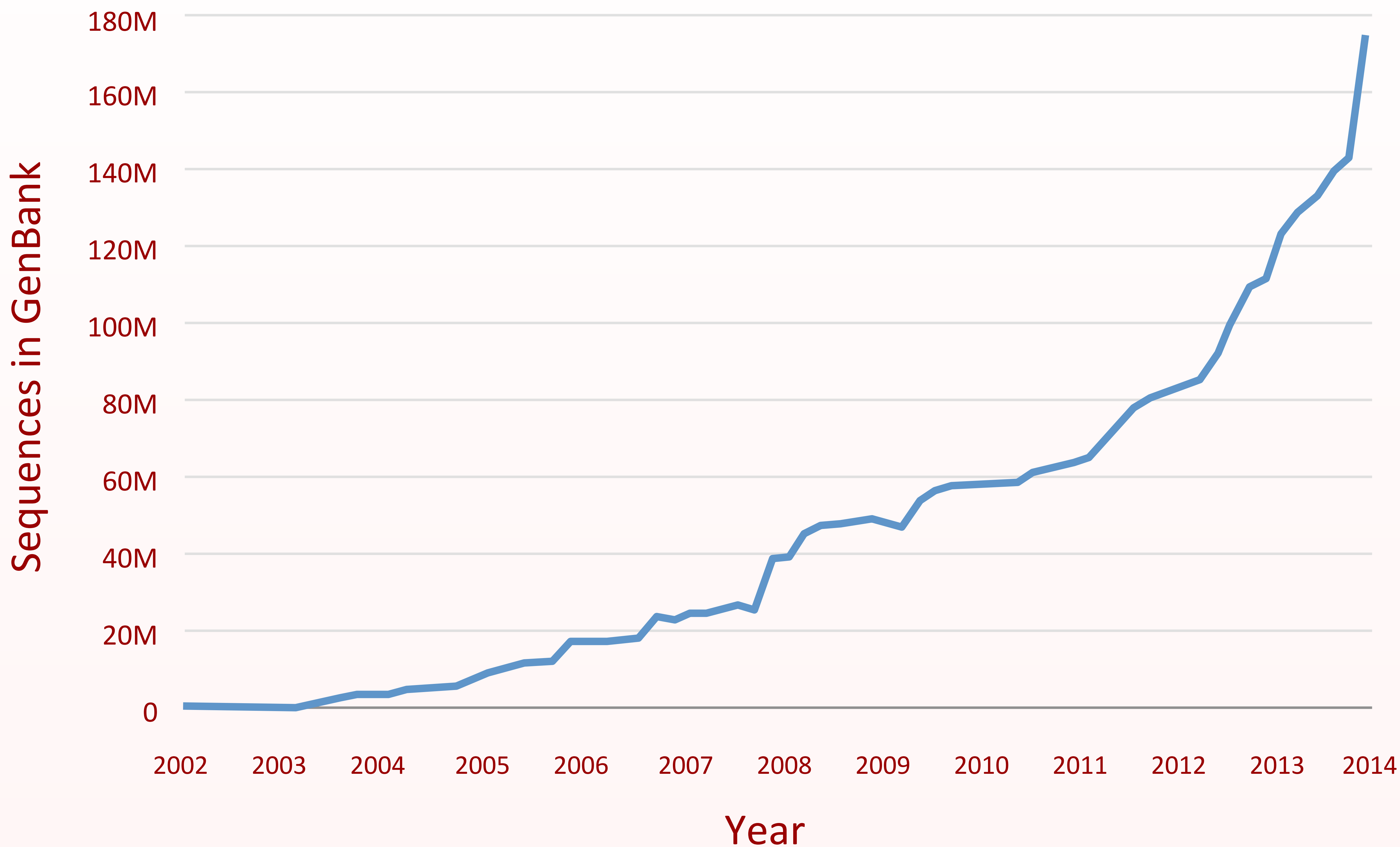
Genomic Compression: Storage, Transmission, and Analytics

Noah M. Daniels
Bonnie Berger

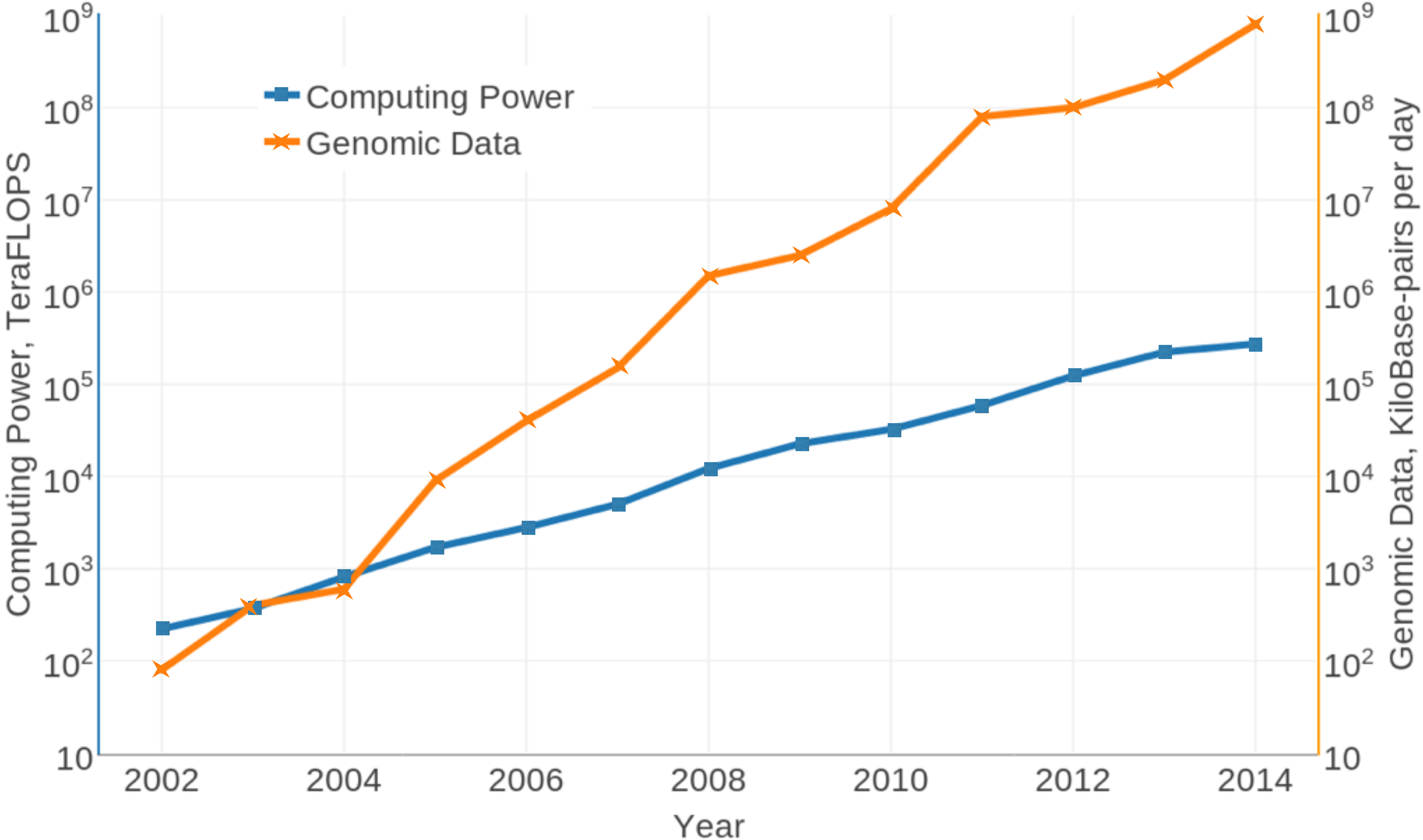
Genomic data are growing exponentially



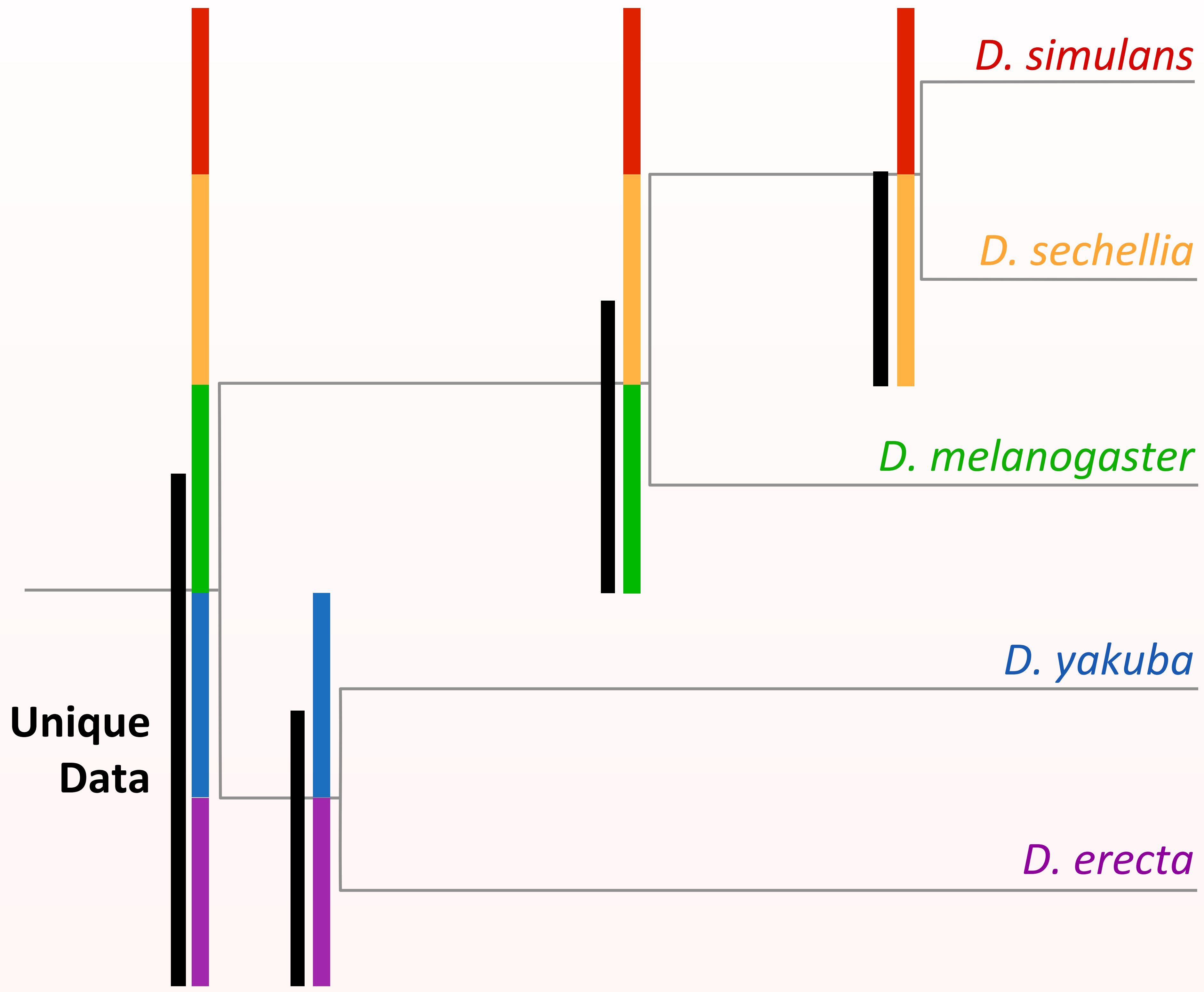

A bigger cloud?


A bigger cloud?
Still constrained by Moore

Genomic data are growing exponentially



Redundancy in data



Outline

Compression background

Storage and transmission

Analysis

Limits to compression

Source coding theorem [Shannon, 1948]

N i.i.d. random variables from source X

each with entropy $H(X)$

cannot be compressed into fewer than $NH(X)$ bits without loss of information

$$\mathrm{H}(X) = -\sum_{i} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)$$

What can we do?

Approach Shannon limit as efficiently as possible (fast, lossless compression)

Be willing to throw away information (*lossy* compression)

Approaching Shannon limit

Huffman coding [1952]

Arithmetic coding [Rissanen 1976]

Lempel-Ziv [1977]

Burrows-Wheeler Transform [1994]

Huffman coding

Huffman coding

Variable-length prefix code

Huffman coding

Variable-length prefix code

More frequent symbols take fewer bits

Huffman coding

Variable-length prefix code

More frequent symbols take fewer bits

DNA (A,C,T,G)

Huffman coding

Variable-length prefix code

More frequent symbols take fewer bits

DNA (A,C,T,G)

2 bits/symbol: 00, 01, 10, 11

Huffman coding

Variable-length prefix code

More frequent symbols take fewer bits

DNA (A,C,T,G)

2 bits/symbol: 00, 01, 10, 11

Suppose AT bias

Huffman coding

Variable-length prefix code

More frequent symbols take fewer bits

DNA (A,C,T,G)

2 bits/symbol: 00, 01, 10, 11

Suppose AT bias

P. falciparum 20% GC

A

T

C

G

A	0.4
T	0.4
C	0.1
G	0.1

A 0.4

T 0.4

C 0.1

G 0.1

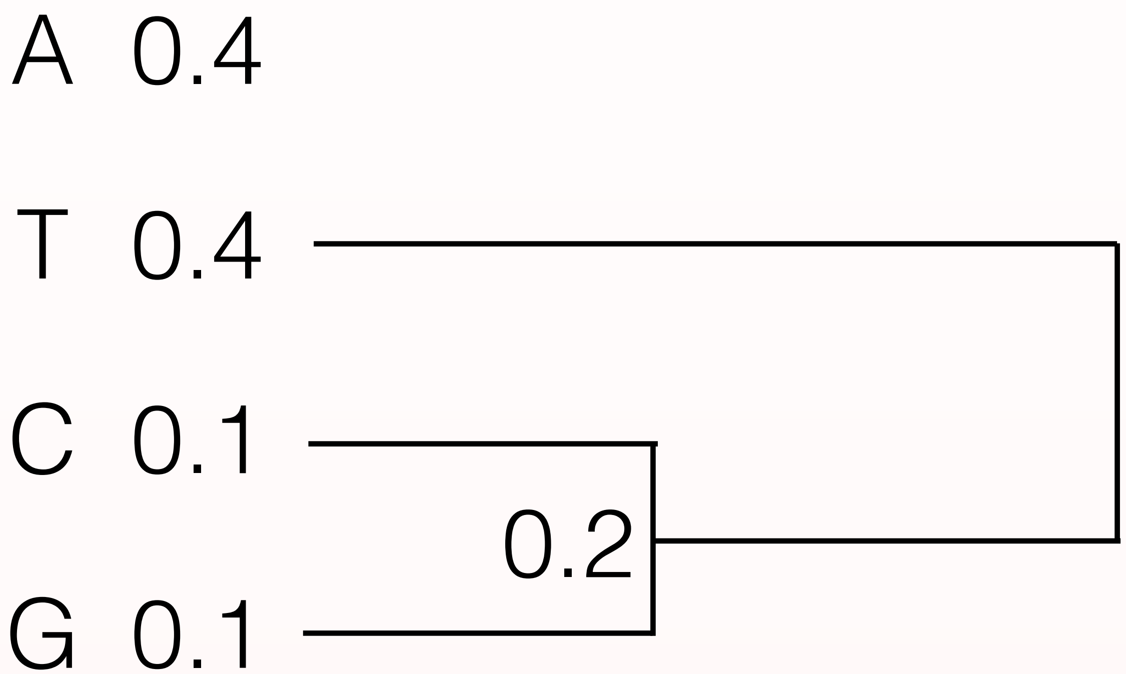
A 0.4

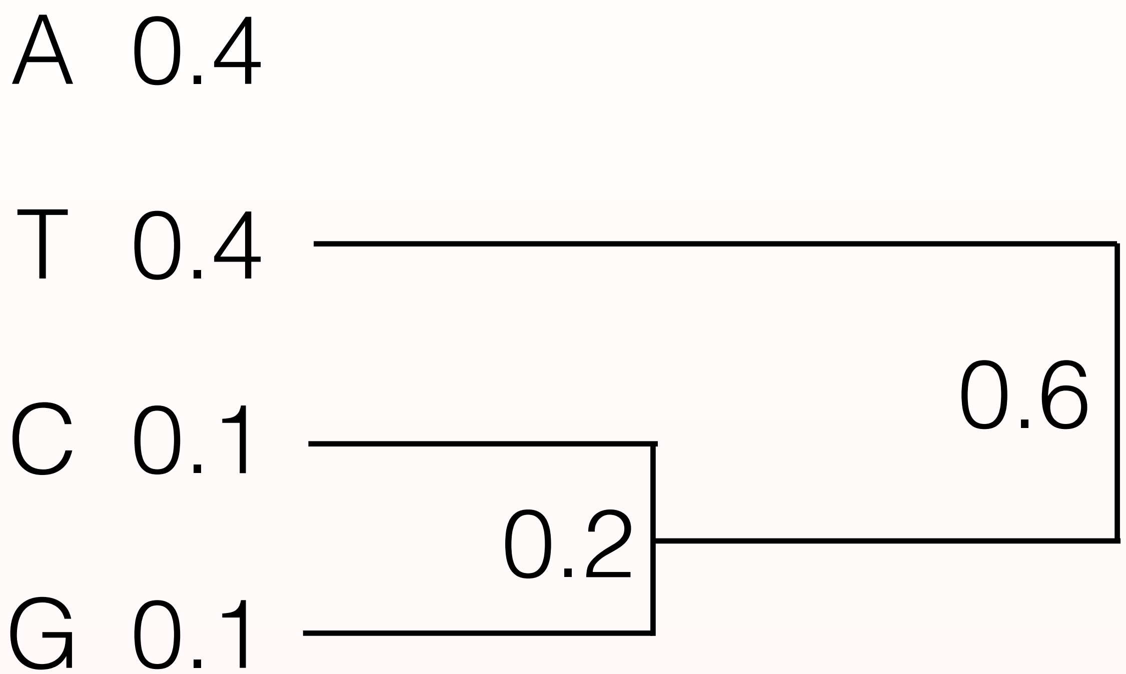
T 0.4

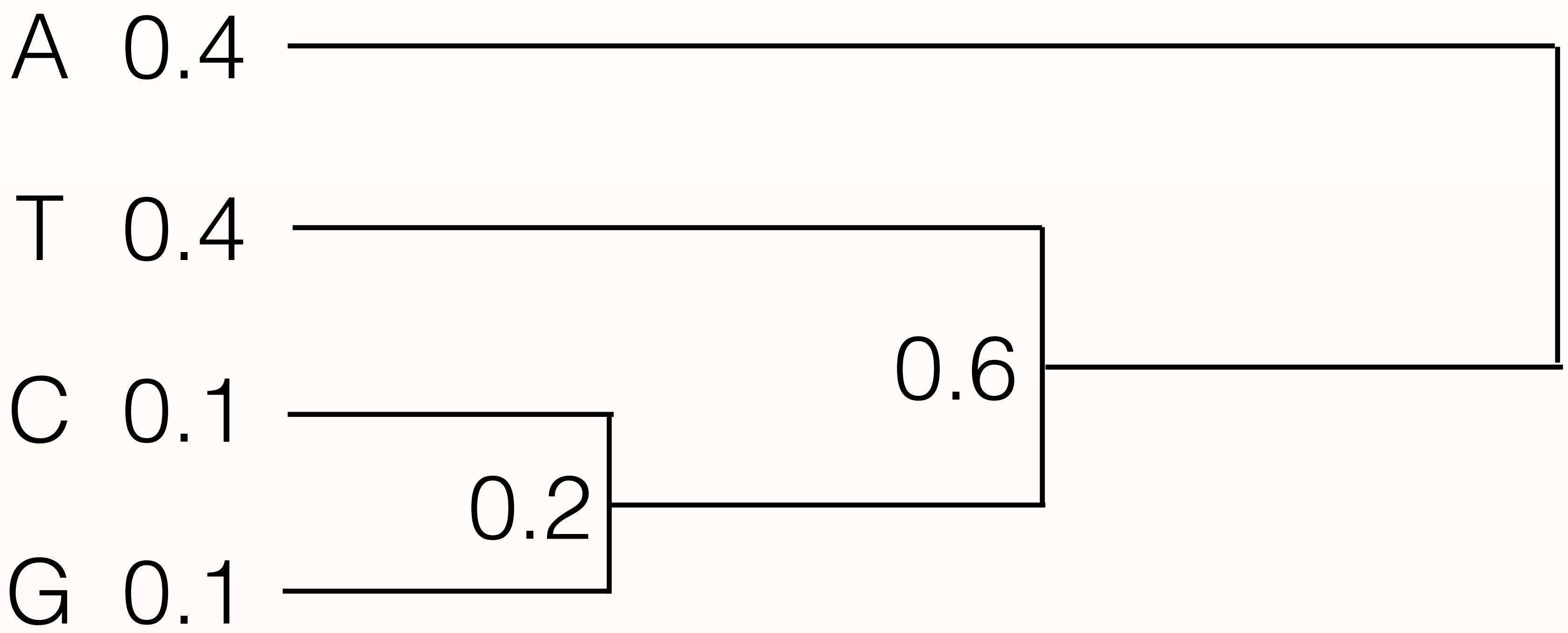
C 0.1

0.2

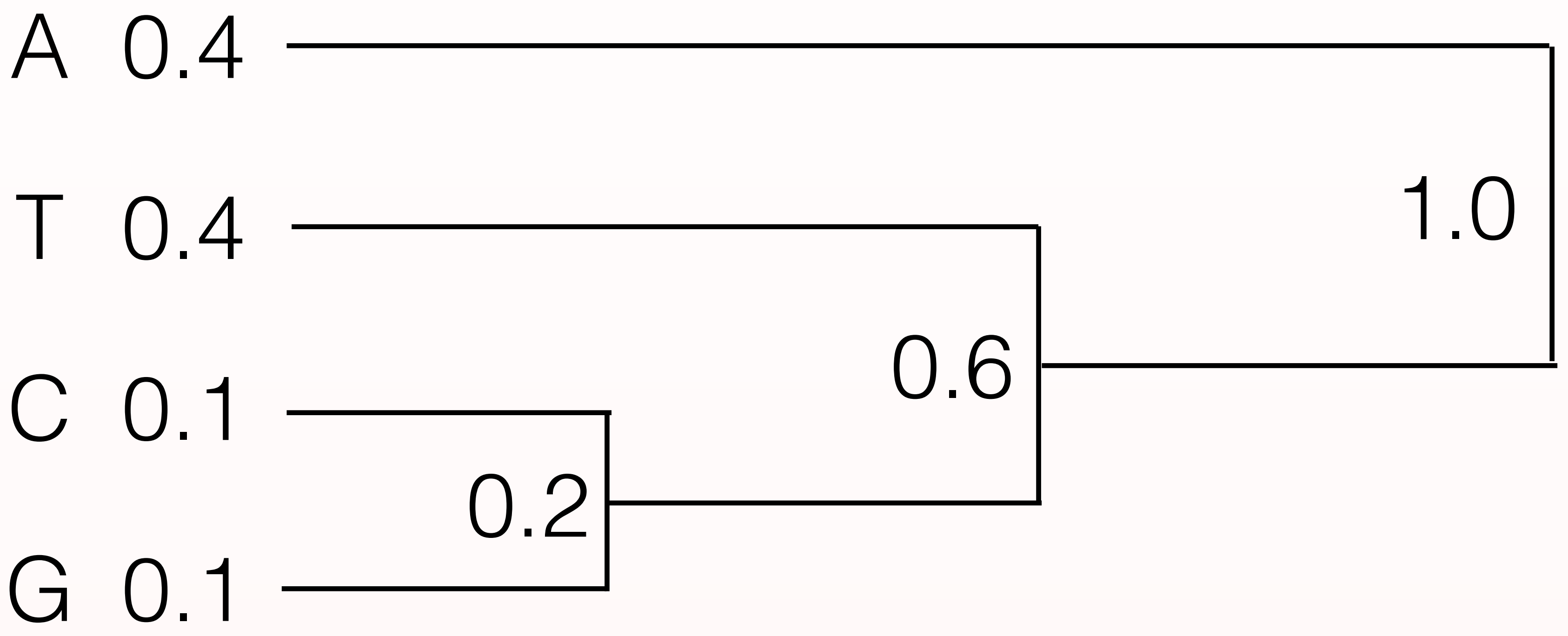
G 0.1

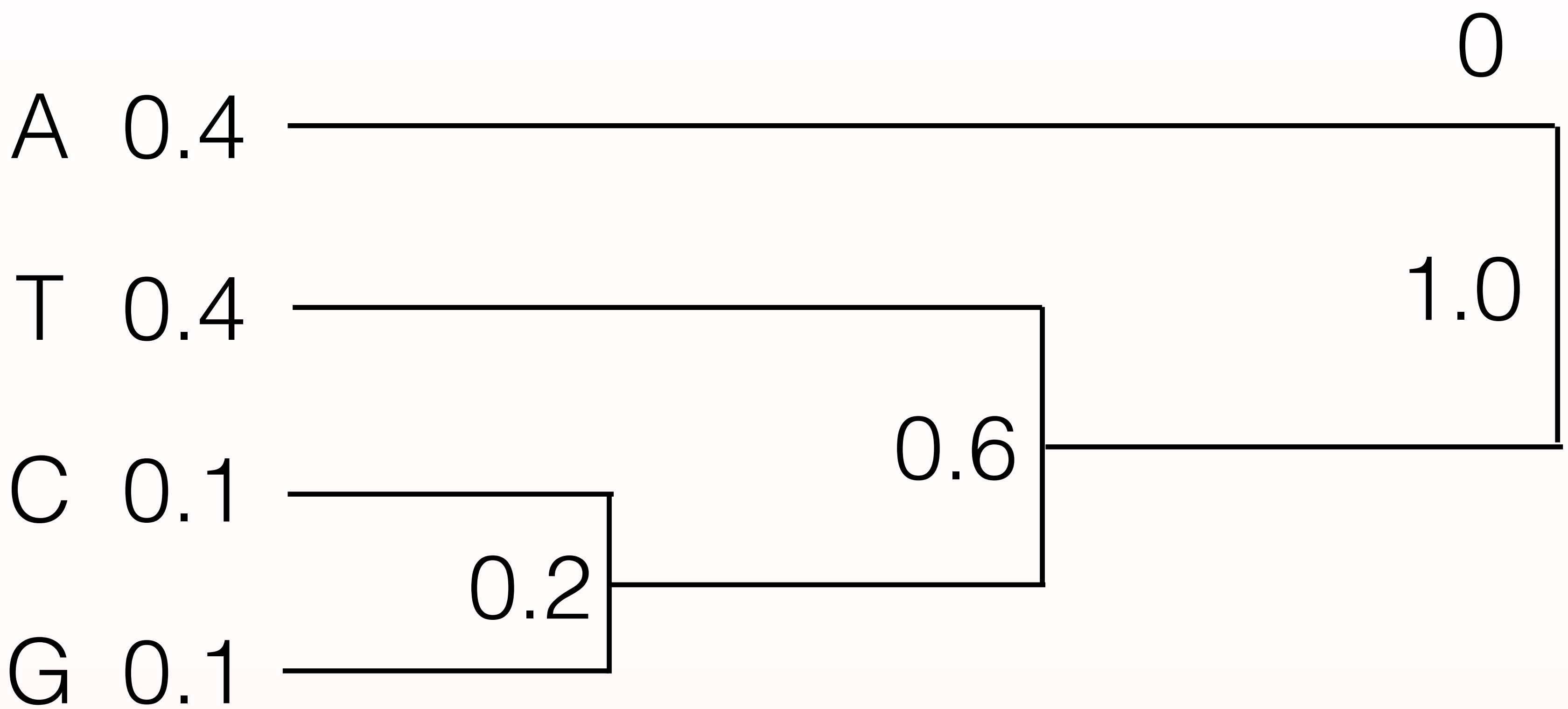

A 0.4
T 0.4
C 0.1
G 0.1
0.2

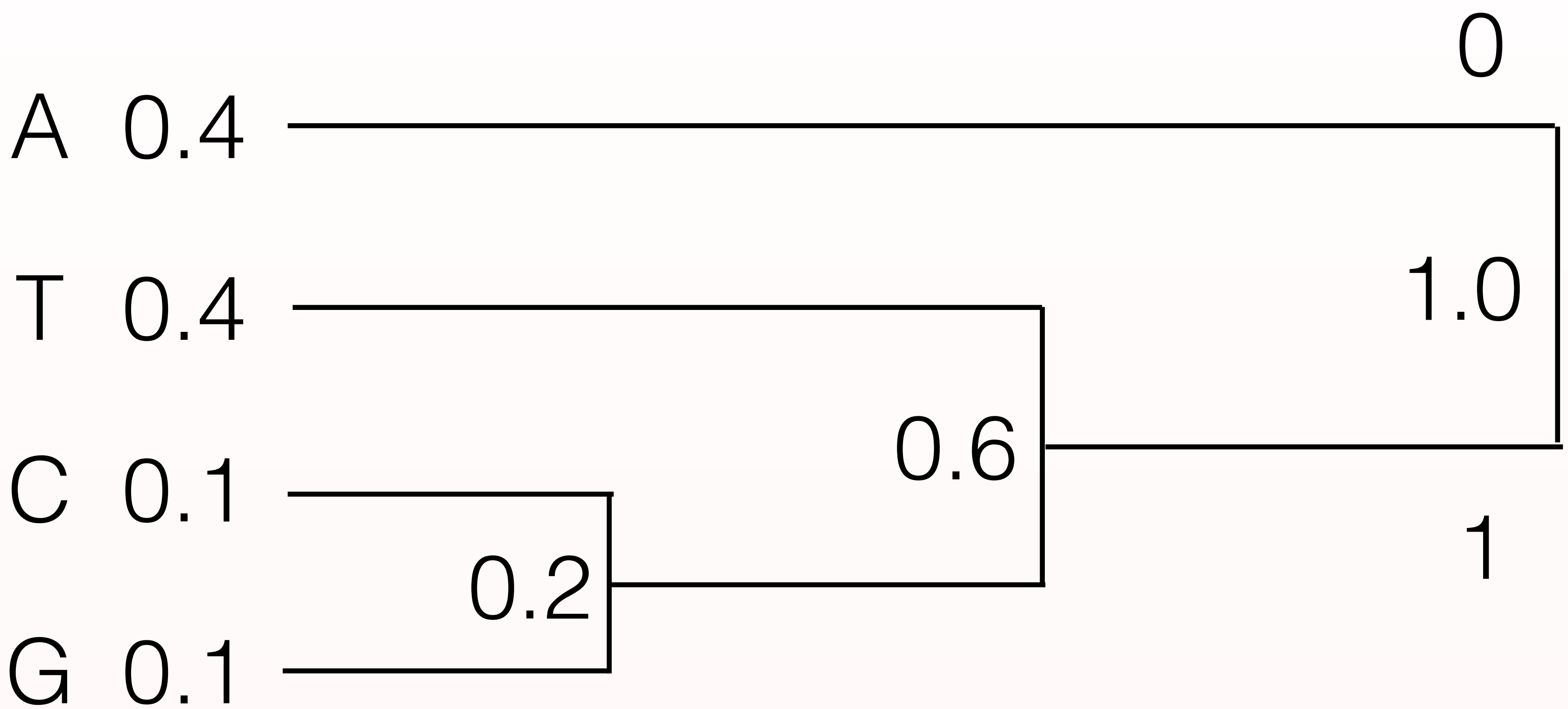

A 0.4
T 0.4
C 0.1
G 0.1
0.2
0.6

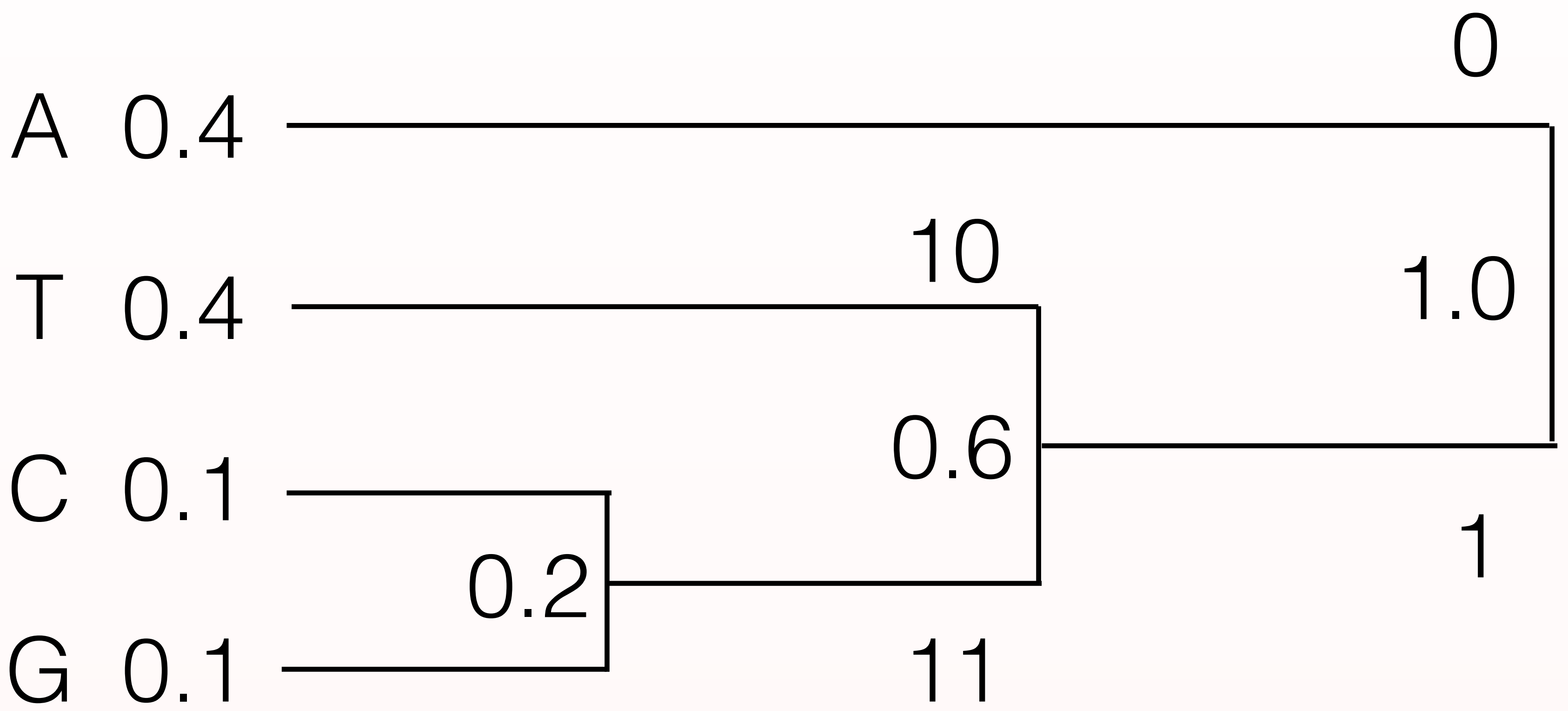


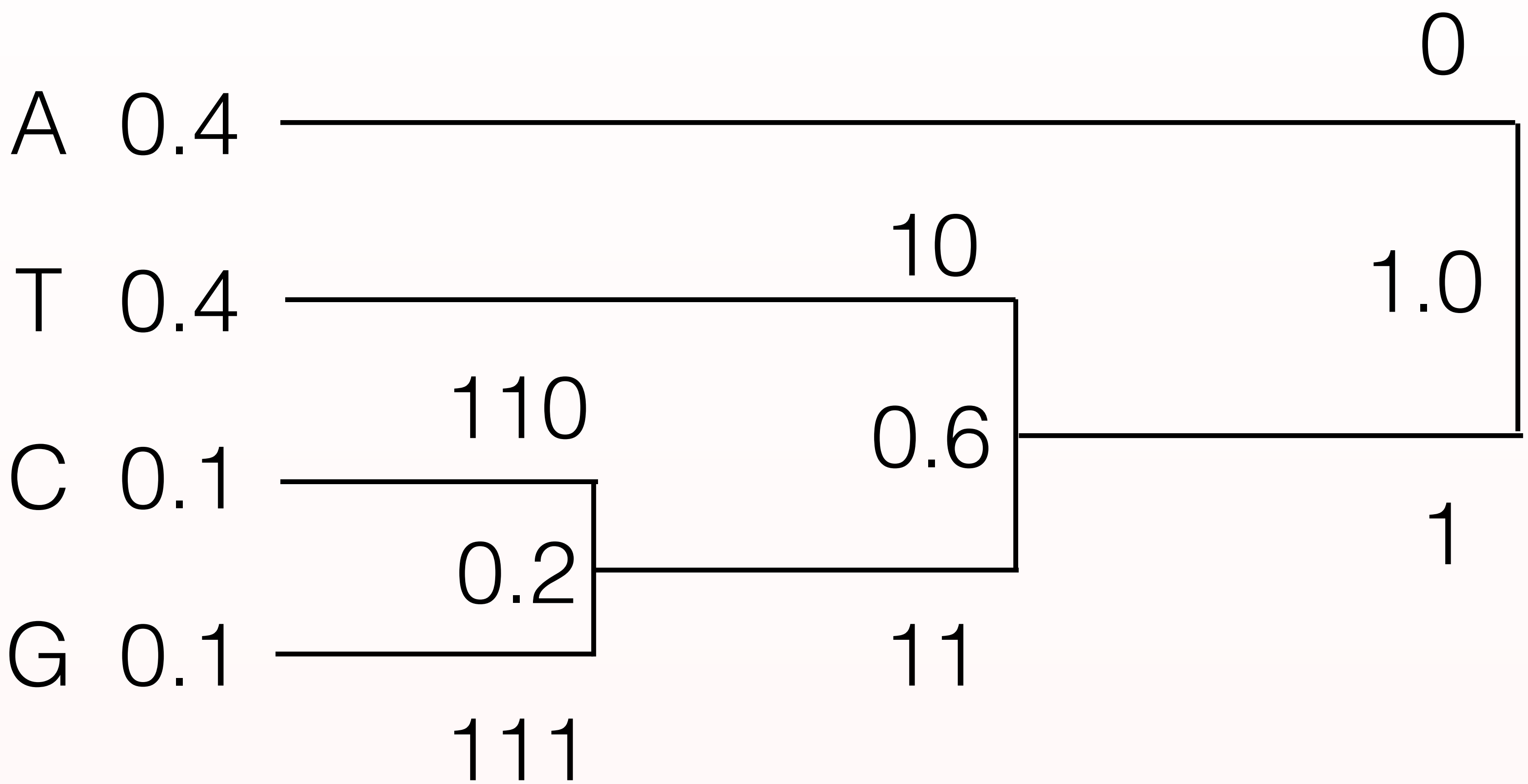
A 0.4
T 0.4
C 0.1
G 0.1
0.2
0.6


A 0.4
T 0.4
C 0.1
G 0.1
0.2
0.6
1.0


A 0.4
T 0.4
C 0.1
G 0.1
0.2
0.6
1.0
0


A 0.4
T 0.4
C 0.1
G 0.1
0.2
0.6
1.0
0
1


A 0.4
T 0.4
C 0.1
G 0.1
0.2
0.6
1.0
0
10
11
1



A 0.4
T 0.4
C 0.1
G 0.1
0
10
1.0
110
0.6
0.2
1
11
111

Huffman coding

Naive encoding: 2 bits/base

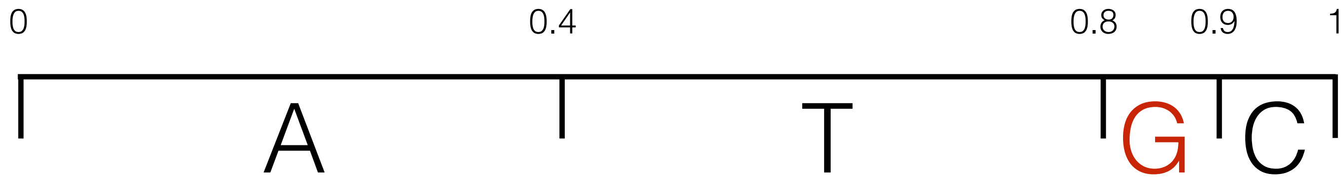
Huffman coding: 1.8 bits/base

	A	T	G	C
F	0.4	0.4	0.1	0.1
Codeword	0	10	110	111
Bits	1	2	3	3
Avg Bits	0.4	0.8	0.3	0.3

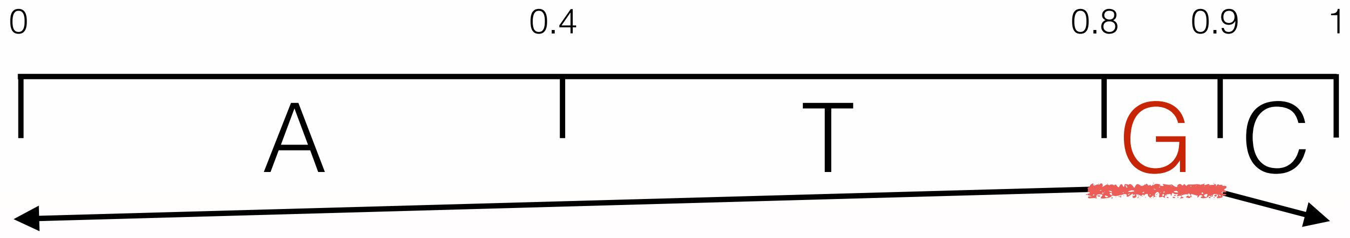
Arithmetic coding

Encode entire message as a number

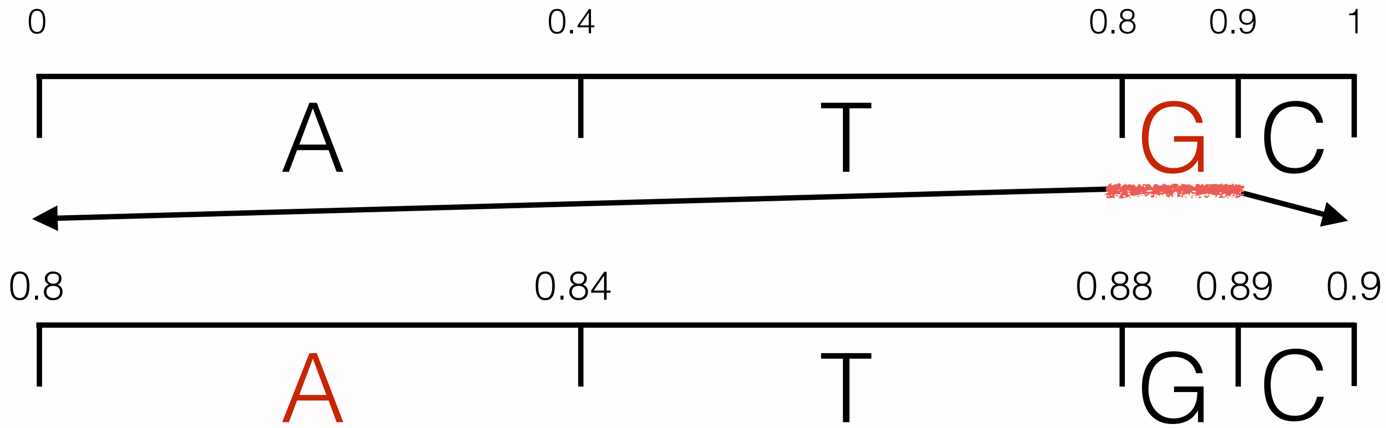
	A	T	G	C
F	0.4	0.4	0.1	0.1
Interval	[0,0.4)	[0.4,0.8)	[0.8,0.9)	[0.9,1)



GATTACA

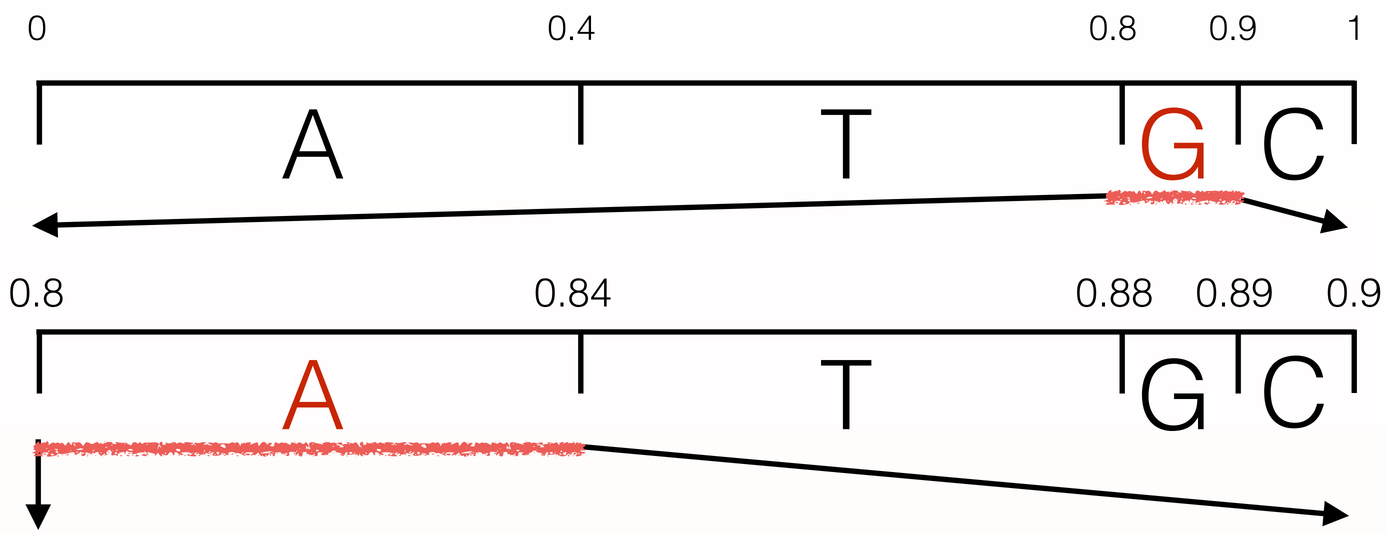


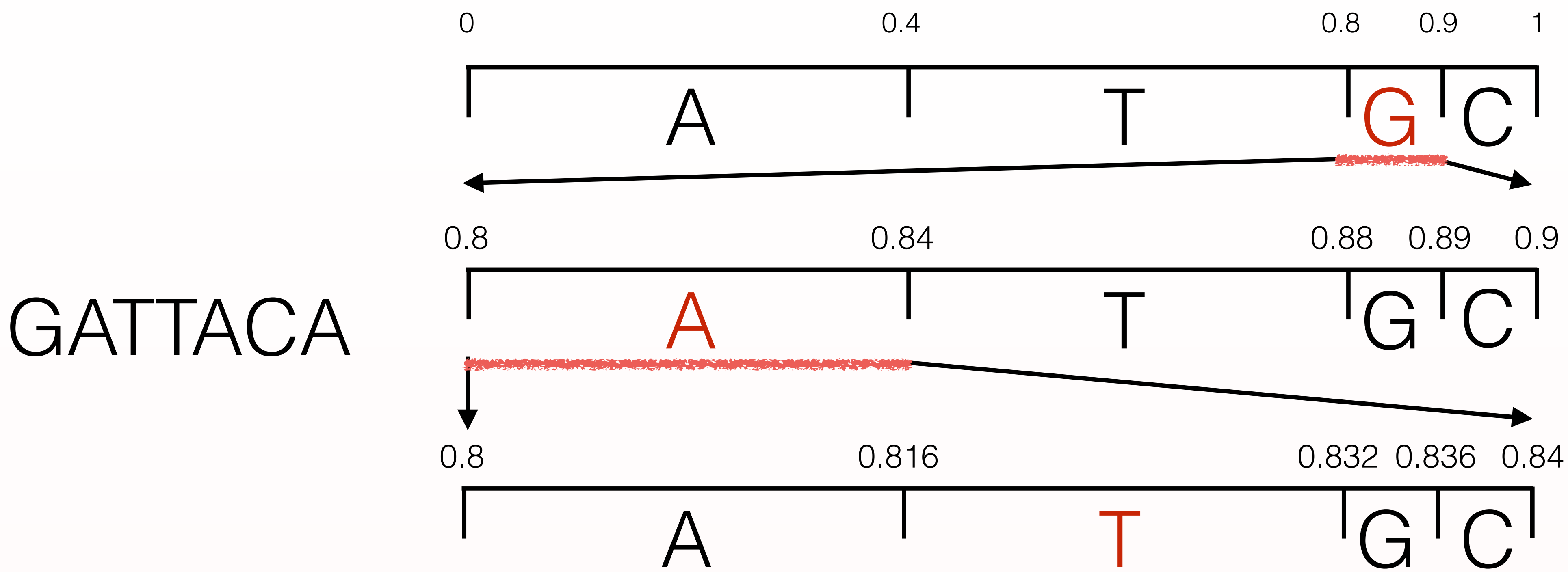
GATTACA



GATTACA

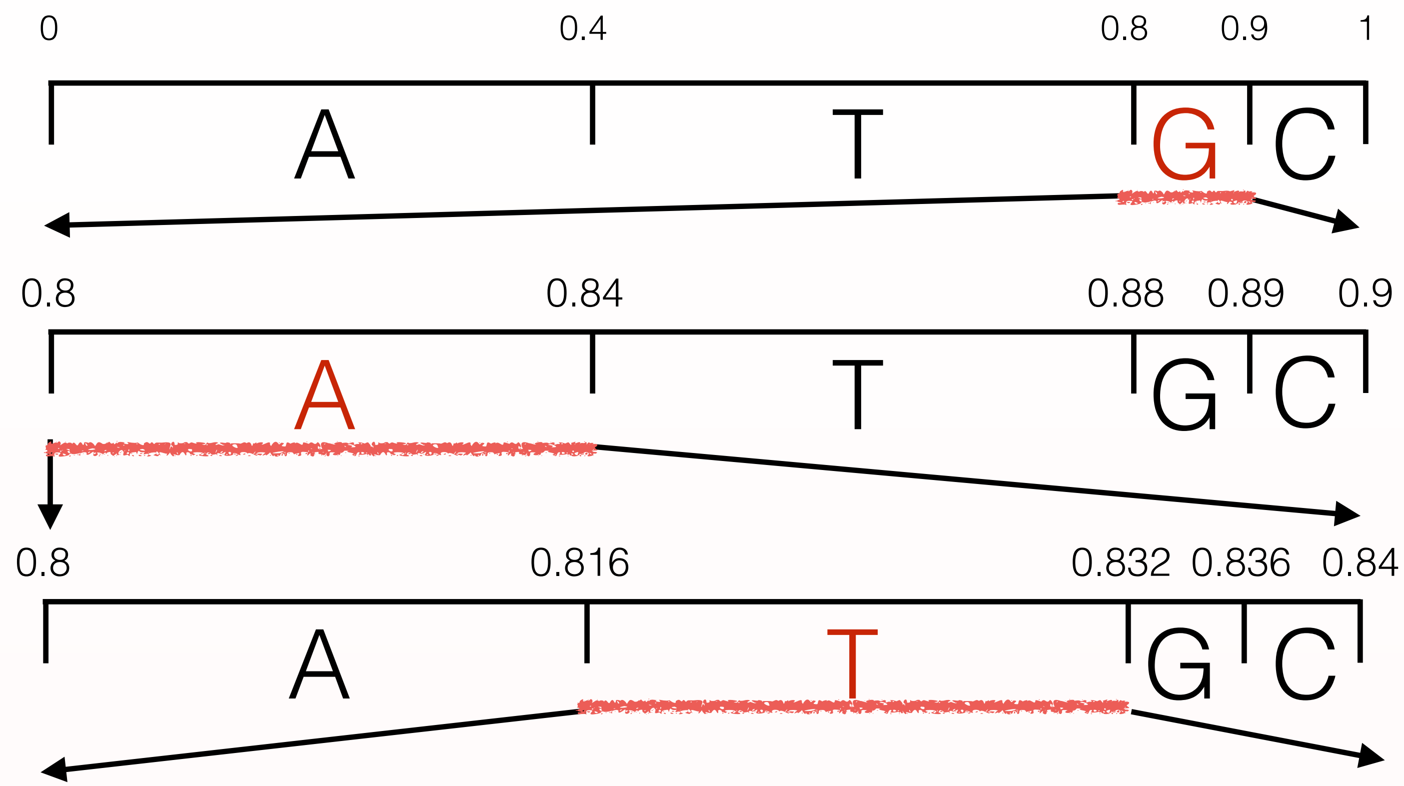
GATTACA



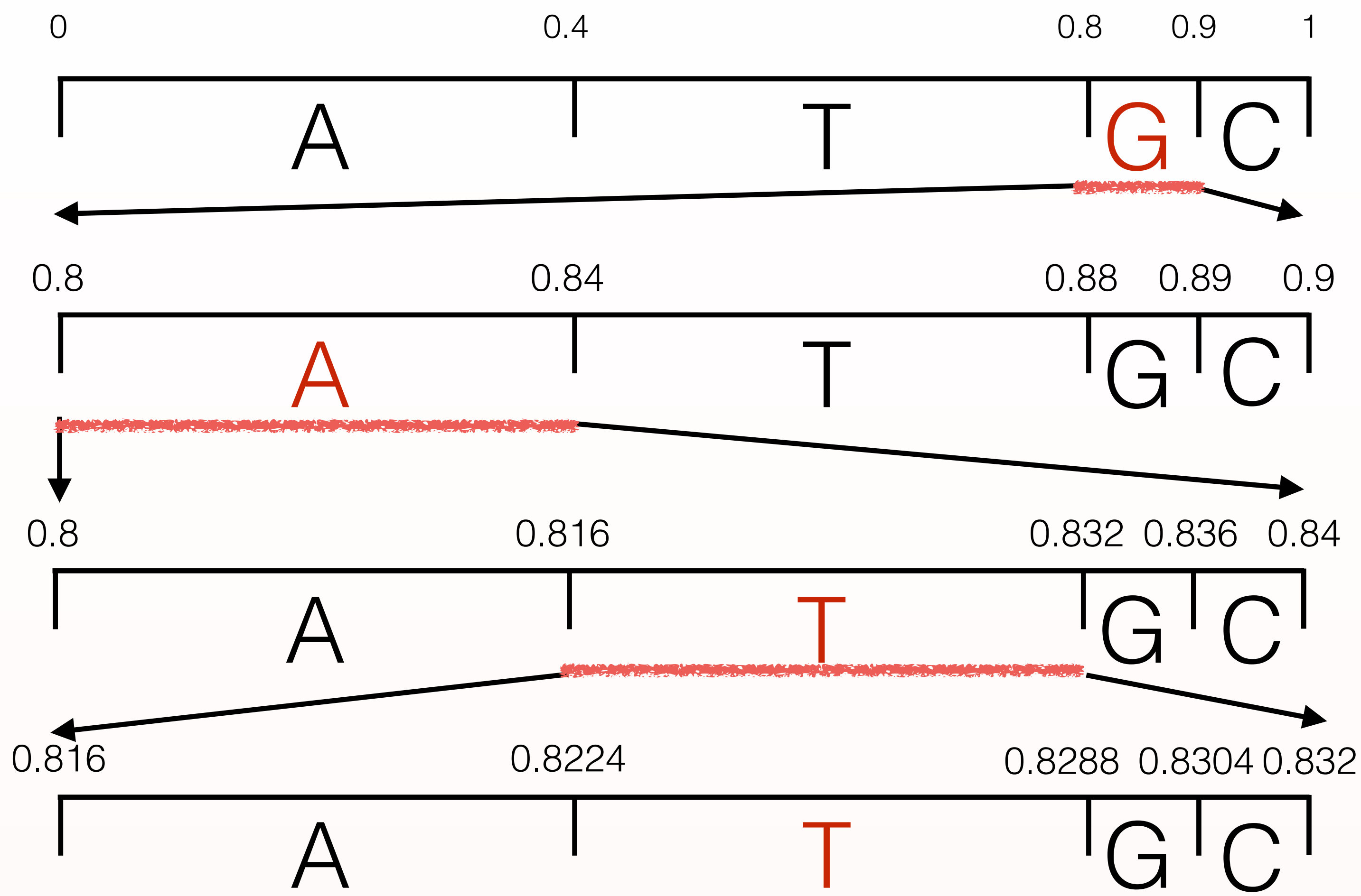


GATTACA
0
0.4
0.8
0.9
1
A
T
G
C
0.8
0.84
0.88
0.89
0.9
A
T
G
C
0.8
0.816
0.832
0.836
0.84
A
T
G
C

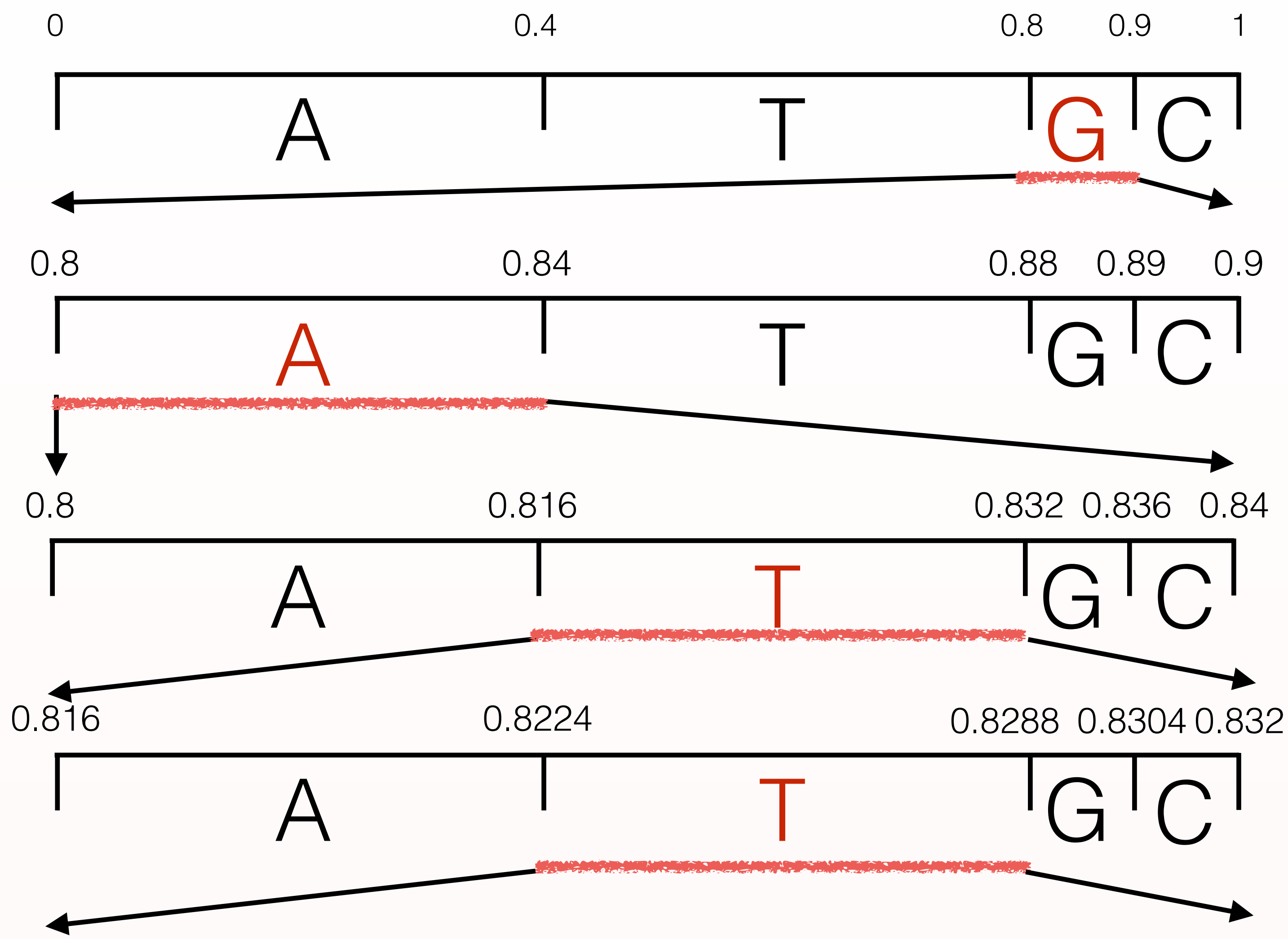
GATTACA


0
0.4
0.8
0.9
1
A
T
G
C
0.8
0.84
0.88
0.89
0.9
A
T
G
C
0.8
0.816
0.832
0.836
0.84
A
T
G
C

GATTACA

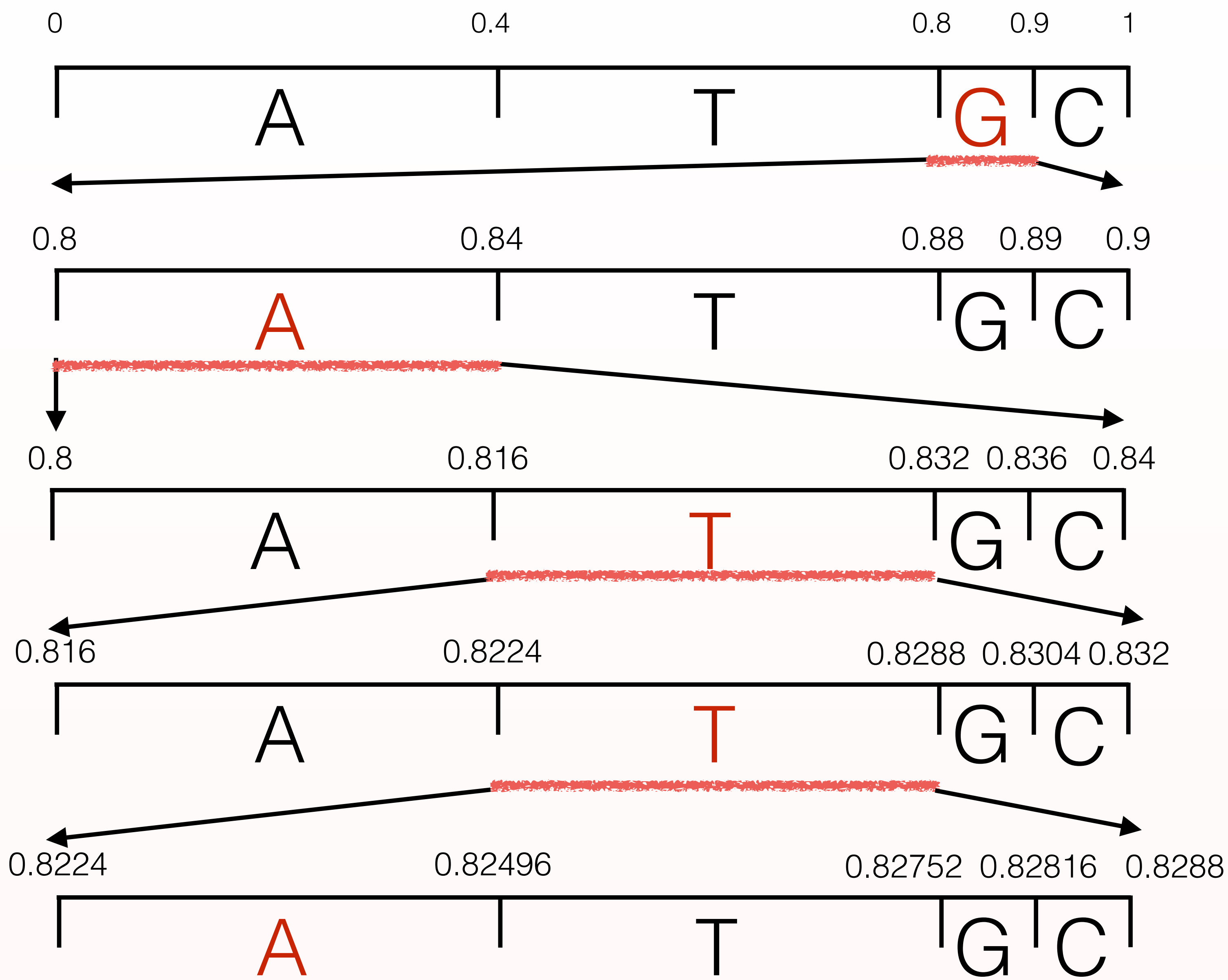

0
0.4
0.8
0.9
1
A
T
G
C
0.8
0.84
0.88
0.89
0.9
A
T
G
C
0.8
0.816
0.832
0.836
0.84
A
T
G
C
0.816
0.8224
0.8288
0.8304
0.832
A
T
G
C

GATTACA



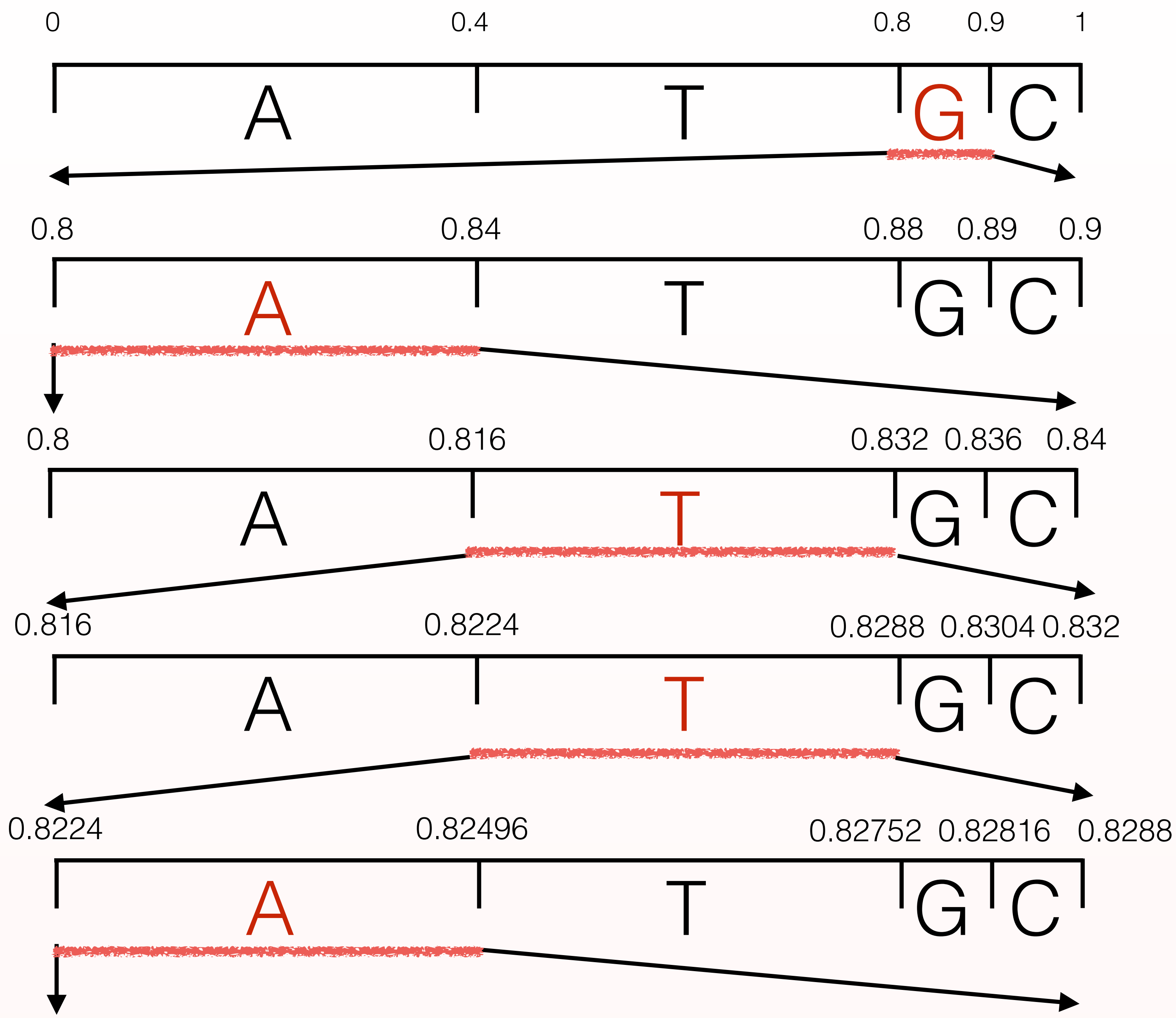
0
0.4
0.8
0.9
1
A
T
G
C
0.8
0.84
0.88
0.89
0.9
A
T
G
C
0.8
0.816
0.832
0.836
0.84
A
T
G
C
0.816
0.8224
0.8288
0.8304
0.832
A
T
G
C

GATTACA



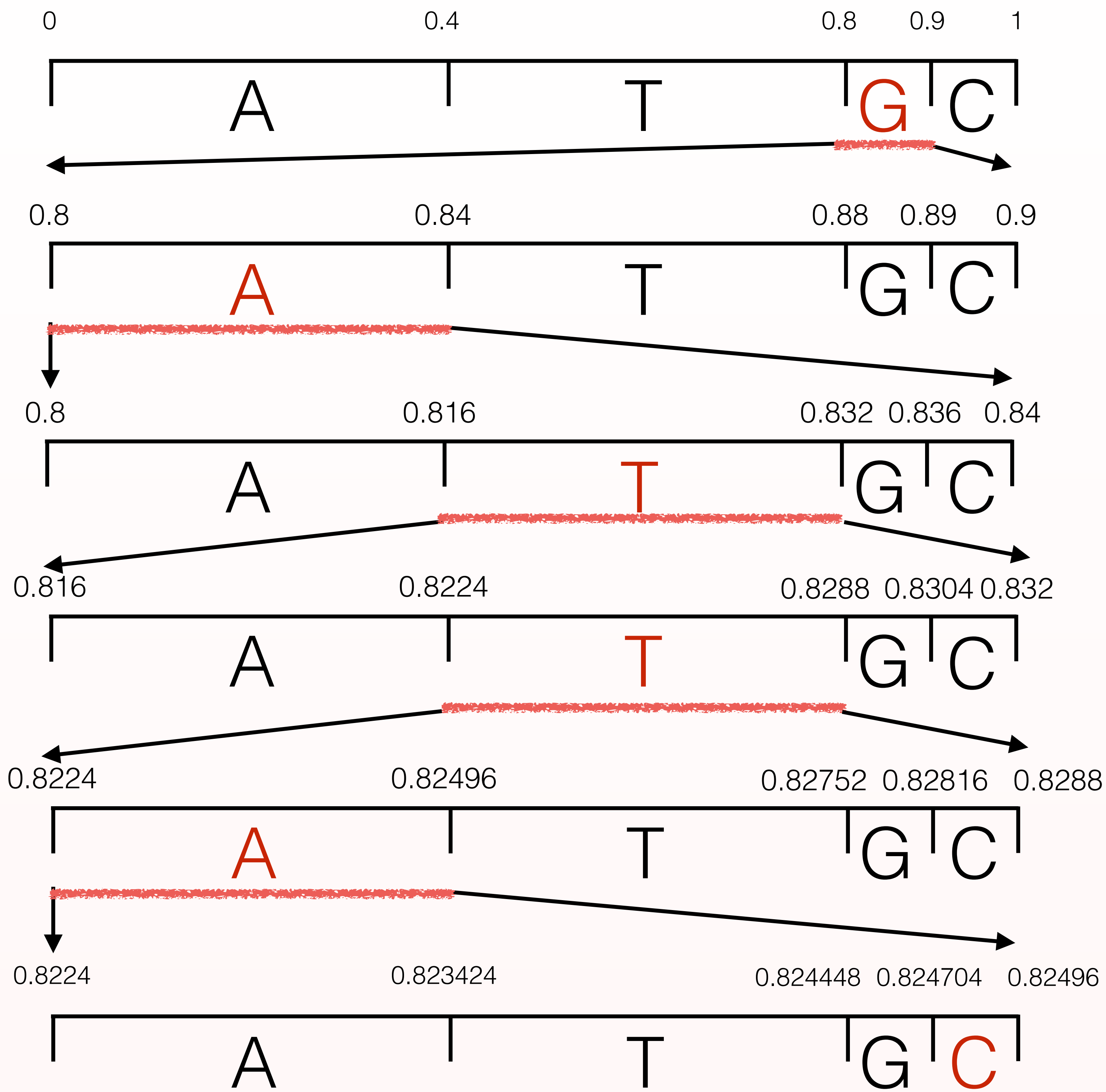
0
0.4
0.8
0.9
1
A
T
G
C
0.8
0.84
0.88
0.89
0.9
A
T
G
C
0.8
0.816
0.832
0.836
0.84
A
T
G
C
0.816
0.8224
0.8288
0.8304
0.832
A
T
G
C
0.8224
0.82496
0.82752
0.82816
0.8288
A
T
G
C

GATTACA

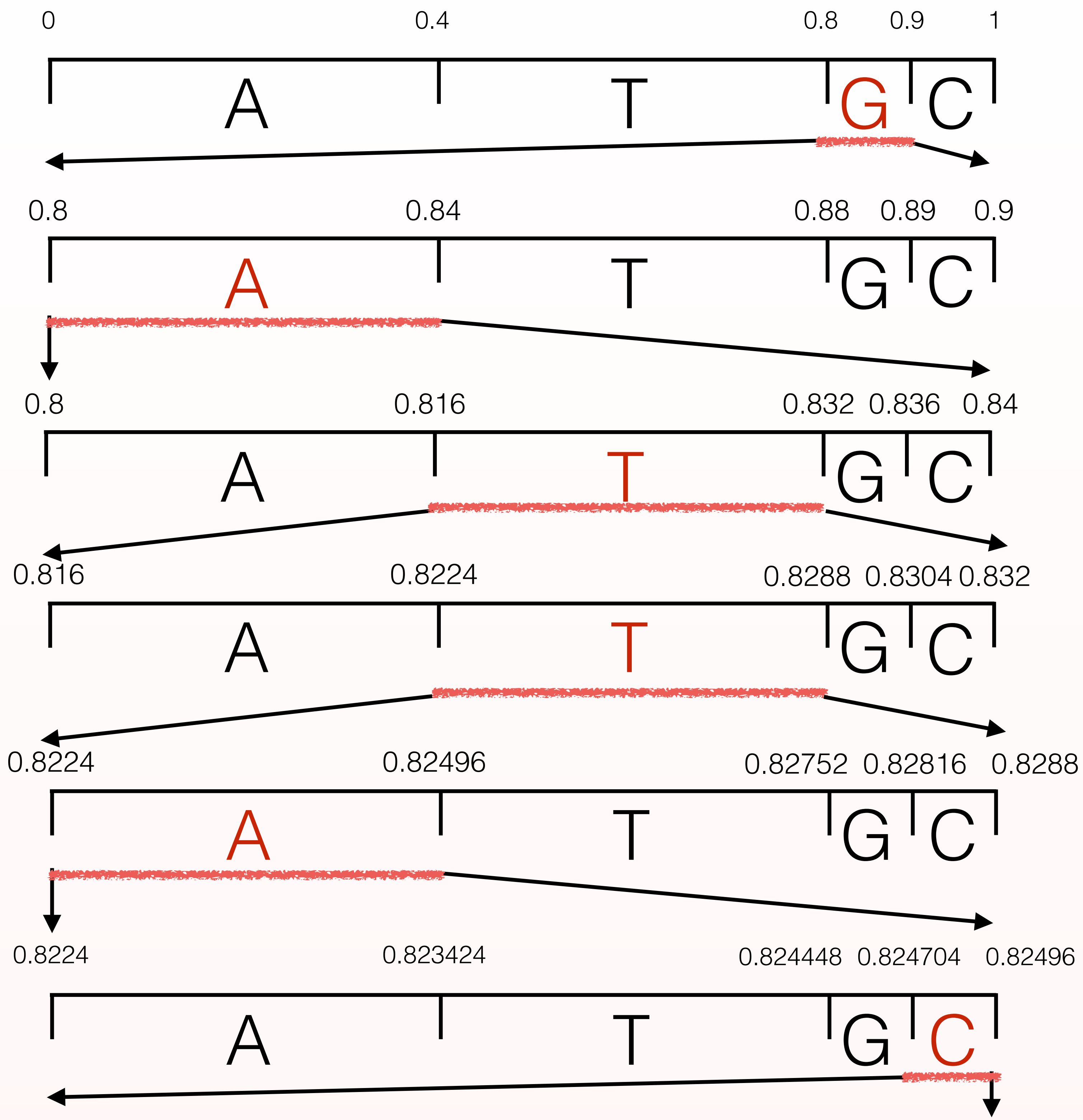


0
0.4
0.8
0.9
1
A
T
G
C
0.8
0.84
0.88
0.89
0.9
A
T
G
C
0.8
0.816
0.832
0.836
0.84
A
T
G
C
0.816
0.8224
0.8288
0.8304
0.832
A
T
G
C
0.8224
0.82496
0.82752
0.82816
0.8288
A
T
G
C

GATTACA

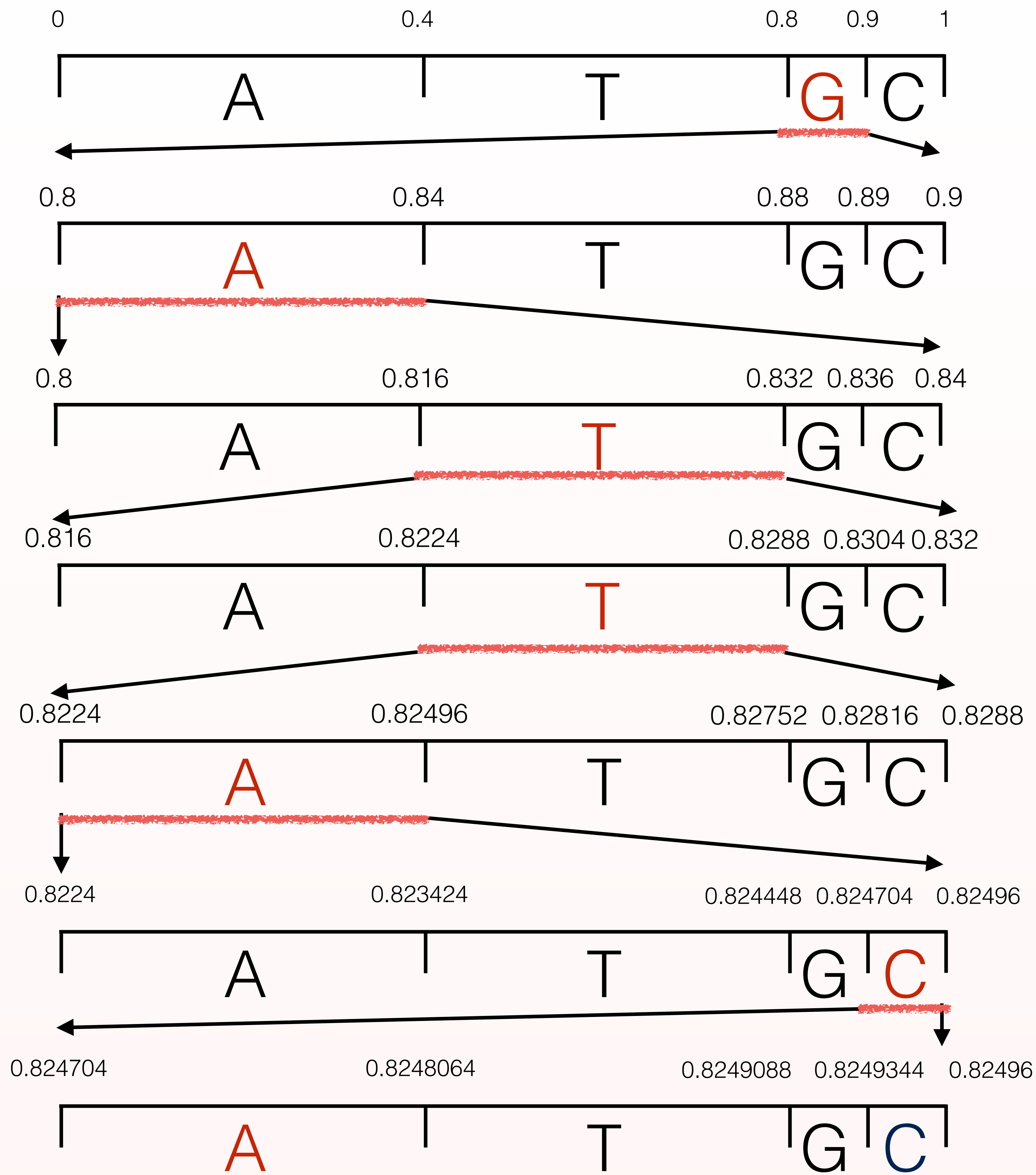

0
0.4
0.8
0.9
1
A
T
G
C
0.8
0.84
0.88
0.89
0.9
A
T
G
C
0.8
0.816
0.832
0.836
0.84
A
T
G
C
0.816
0.8224
0.8288
0.8304
0.832
A
T
G
C
0.8224
0.82496
0.82752
0.82816
0.8288
A
T
G
C
0.8224
0.823424
0.824448
0.824704
0.82496
A
T
G
C

GATTACA



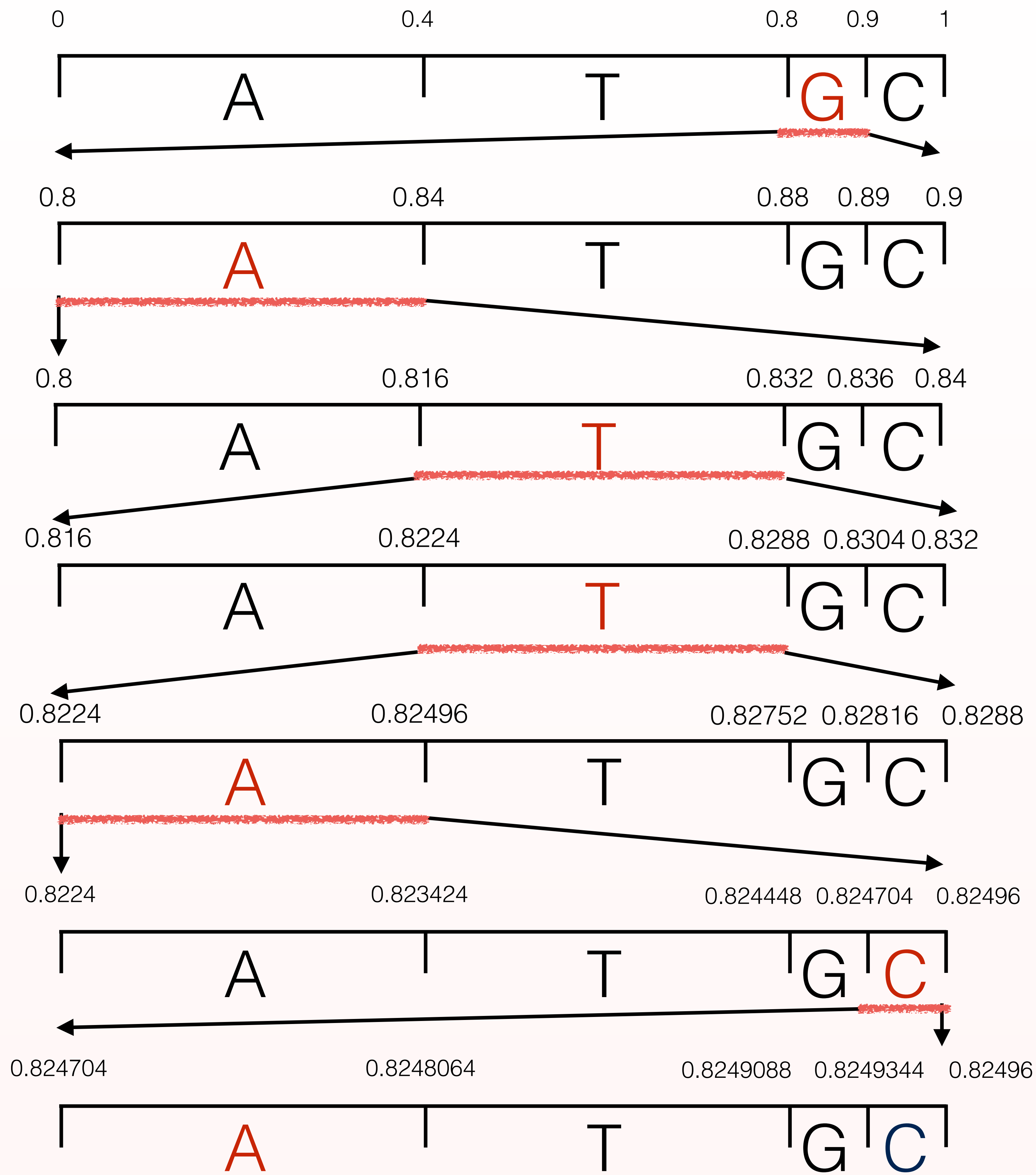
0
0.4
0.8
0.9
1
A
T
G
C
0.8
0.84
0.88
0.89
0.9
A
T
G
C
0.8
0.816
0.832
0.836
0.84
A
T
G
C
0.816
0.8224
0.8288
0.8304
0.832
A
T
G
C
0.8224
0.82496
0.82752
0.82816
0.8288
A
T
G
C
0.8224
0.823424
0.824448
0.824704
0.82496
A
T
G
C

GATTACA



0
0.4
0.8
0.9
1
A
T
G
C
0.8
0.84
0.88
0.89
0.9
A
T
G
C
0.8
0.816
0.832
0.836
0.84
A
T
G
C
0.816
0.8224
0.8288
0.8304
0.832
A
T
G
C
0.8224
0.82496
0.82752
0.82816
0.8288
A
T
G
C
0.8224
0.823424
0.824448
0.824704
0.82496
A
T
G
C
0.824704
0.8248064
0.8249088
0.8249344
0.82496
A
T
G
C

GATTACA



0.824704

Arithmetic coding

Arithmetic coding

How do we know when the message ends?

Arithmetic coding

How do we know when the message ends?

I lied; we need a stop symbol

Arithmetic coding

How do we know when the message ends?

I lied; we need a stop symbol

Bigger problem: floating point precision

Arithmetic coding

How do we know when the message ends?

I lied; we need a stop symbol

Bigger problem: floating point precision

Integer version

Arithmetic coding

How do we know when the message ends?

I lied; we need a stop symbol

Bigger problem: floating point precision

Integer version

Rodionov & Volkov, 2007 & 2010 (p-adic arithmetic)

Arithmetic coding

How do we know when the message ends?

I lied; we need a stop symbol

Bigger problem: floating point precision

Integer version

Rodionov & Volkov, 2007 & 2010 (p-adic arithmetic)

Huffman coding is a specialized case, less likely to be optimal

Lempel-Ziv

Slide a fixed-length window along sequence

Replace already-seen patterns (of some maximum length)

Used by gzip (& others)

`GATTACATTA`

Lempel-Ziv

Slide a fixed-length window along sequence

Replace already-seen patterns (of some maximum length)

Used by gzip (& others)

GATTACATTA

Lempel-Ziv

Slide a fixed-length window along sequence

Replace already-seen patterns (of some maximum length)

Used by gzip (& others)

GATTACATTA

Lempel-Ziv

Slide a fixed-length window along sequence

Replace already-seen patterns (of some maximum length)

Used by gzip (& others)

`GATTACATTA`

Lempel-Ziv

Slide a fixed-length window along sequence

Replace already-seen patterns (of some maximum length)

Used by gzip (& others)

`GATTACATTA`

Lempel-Ziv

Slide a fixed-length window along sequence

Replace already-seen patterns (of some maximum length)

Used by gzip (& others)

GATTACATTA

Lempel-Ziv

Slide a fixed-length window along sequence

Replace already-seen patterns (of some maximum length)

Used by gzip (& others)

GATTACATTA

Lempel-Ziv

Slide a fixed-length window along sequence

Replace already-seen patterns (of some maximum length)

Used by gzip (& others)

```
GATTACATTA

GATTAC<1,4>
```

Lempel-Ziv

Slide a fixed-length window along sequence

Replace already-seen patterns (of some maximum length)

Used by gzip (& others)

GATTACATTA

GATTAC<1,4>

Pointer

Lempel-Ziv

Slide a fixed-length window along sequence

Replace already-seen patterns (of some maximum length)

Used by gzip (& others)

GATTACATTA

GATTAC<1,4>

Pointer Length

Burrows-Wheeler Transform (BWT)

Burrows-Wheeler Transform (BWT)

block-sorting compression

Burrows-Wheeler Transform (BWT)

block-sorting compression

not itself a compressor, but a way to sort input, reversibly

Burrows-Wheeler Transform (BWT)

block-sorting compression

not itself a compressor, but a way to sort input, reversibly

Lends itself to run-length encoding

Burrows-Wheeler Transform (BWT)

block-sorting compression

not itself a compressor, but a way to sort input, reversibly

Lends itself to run-length encoding

Genomic compressors

Genomic compressors

Why not just gzip it?

Genomic compressors

Why not just gzip it?

sequence

Genomic compressors

Why not just gzip it?

sequence

quality scores (one per base)

Genomic compressors

Why not just gzip it?

sequence

quality scores (one per base)

sequence: 4-letter alphabet

Genomic compressors

Why not just gzip it?

sequence

quality scores (one per base)

sequence: 4-letter alphabet

quality scores: 40-value range

Genomic compressors

Why not just gzip it?

sequence

quality scores (one per base)

sequence: 4-letter alphabet

quality scores: 40-value range

sequence: must be lossless

Genomic compressors

Why not just gzip it?

sequence

quality scores (one per base)

sequence: 4-letter alphabet

quality scores: 40-value range

sequence: must be lossless

quality scores: lossy might be ok

Genomic compressors

Genomic compressors

Take advantage of structure

Genomic compressors

Take advantage of structure

Store differences from a **reference**

Genomic compressors

Take advantage of structure

Store differences from a **reference**

Cluster similar reads (**reference-free**)

Genomic compressors

Take advantage of structure

Store differences from a **reference**

Cluster similar reads (**reference-free**)

don’t have references for all species

SCALCE [Hach et al. 2012]

SCALCE [Hach et al. 2012]

Reference-free

SCALCE [Hach et al. 2012]

Reference-free

Locally Consistent Encoding

SCALCE [Hach et al. 2012]

Reference-free

Locally Consistent Encoding

Buckets reads by substrings

SCALCE [Hach et al. 2012]

Reference-free

Locally Consistent Encoding

Buckets reads by substrings

improves LZ77 runtime AND compression

SCALCE [Hach et al. 2012]

SCALCE [Hach et al. 2012]

Identify local maxima

SCALCE [Hach et al. 2012]

Identify local maxima
Identify local minima

SCALCE [Hach et al. 2012]

Identify local maxima
Identify local minima
Partition at each side

SCALCE [Hach et al. 2012]

Identify local maxima
Identify local minima
Partition at each side
Extend L and R to “core blocks”

SCALCE [Hach et al. 2012]

Identify local maxima
Identify local minima
Partition at each side
Extend L and R to “core blocks”

X0X

SCALCE [Hach et al. 2012]

Identify local maxima
Identify local minima
Partition at each side
Extend L and R to “core blocks”

X0X

2 1 3 1 2 0 3 2 1 0 2 0 2 1 3 1 2 0 3 2 1 0 2

SCALCE [Hach et al. 2012]

Identify local maxima
Identify local minima
Partition at each side
Extend L and R to “core blocks”

XOX

2 1 **3** 1 **2** 0 **3** 2 1 0 **2** 0 **2** 1 **3** 1 **2** 0 **3** 2 1 0 2

SCALCE [Hach et al. 2012]

Identify local maxima

Identify local minima

Partition at each side

Extend L and R to “core blocks”

X0X

2 1 **3** 1 **2** 0 **3** 2 1 0 **2** 0 **2** 1 **3** 1 **2** 0 **3** 2 1 0 2

| 2 1 **3** | 1 **2** | 0 **3** | 2 1 0 **2** | 0 **2** | 1 **3** | 1 **2** | 0 **3** | **2** 1 0 2 |

SCALCE [Hach et al. 2012]

Identify local maxima
Identify local minima
Partition at each side
Extend L and R to “core blocks”

X0X

21**3**1**2**0**3**210**2**0**2**1**3**1**2**0**3**2102

|21**3**|1**2**|0**3**|210**2**|0**2**|1**3**|1**2**|0**3**|**2**102|

2131,3**12**0,2**032**,3**2102**0,2**02**1,2**13**1,
3**12**0,2**03**2,3**2102**

SCALCE [Hach et al. 2012]

Core blocks between 8 & 20

99% of all HTS reads of length ≥50

include at least one core of length ≤14

Identify overlapping reads by core blocks

Reorder reads to favor LZ77

DeeZ [Hach, et al. 2014]

DeeZ [Hach, et al. 2014]

Given a mapping (reads to reference)

DeeZ [Hach, et al. 2014]

Given a mapping (reads to reference)

Partition reads into blocks according to locus

DeeZ [Hach, et al. 2014]

Given a mapping (reads to reference)

Partition reads into blocks according to locus

Build contig covering all reads within a block

DeeZ [Hach, et al. 2014]

Given a mapping (reads to reference)

Partition reads into blocks according to locus

Build contig covering all reads within a block

Encode locus for each read within contig

DeeZ [Hach, et al. 2014]

Given a mapping (reads to reference)

Partition reads into blocks according to locus

Build contig covering all reads within a block

Encode locus for each read within contig

Encode rare differences (sequencing errors)

DeeZ [Hach, et al. 2014]

```
           GTCGTCTACA T
          CGTCGTCTACA
        AACCTCGTCTAC
              GTCTACA TCTA
ACGTGCTAAACGTCGTTTACAGTCTACAGA
```

DeeZ [Hach, et al. 2014]

```
           GTCGTCTACA T
          CGTCGTCTACA
        AACCTCGTCTAC
              GTCTACA TCTA
ACGTGCTAAACGTCGTTTACAGTCTACAGA

ACGTGCTAAACGTCGTCTACA TCTACAGA
```

DeeZ [Hach, et al. 2014]

```
           GTCGTCTACA T
          CGTCGTCTACA
        AACCTCGTCTAC
              GTCTACA TCTA
ACGTGCTAAACGTCGTTTACAGTCTACAGA

ACGTGCTAAACGTCGTCTACA TCTACAGA
```

DeeZ [Hach, et al. 2014]

DeeZ [Hach, et al. 2014]

Tokenization of read names

DeeZ [Hach, et al. 2014]

Tokenization of read names

LZ77

DeeZ [Hach, et al. 2014]

Tokenization of read names

LZ77

Lossy QS compression from SCALCE

DeeZ [Hach, et al. 2014]

Tokenization of read names

LZ77

Lossy QS compression from SCALCE

Random access

PathEnc [Kingsford & Patro 2015]

Reference-based

- but no aligning
- statistical, generative model of reads

For RNA-seq data (but need not be)

PathEnc [Kingsford & Patro 2015]

GAUU

GAUUAGAUUG

PathEnc [Kingsford & Patro 2015]

GAUU

$\downarrow$

AUUA

GAUUAGAUUG

PathEnc [Kingsford & Patro 2015]

$$\begin{tikzcd} \text{GAUU} \arrow[d] \\ \text{AUUA} \arrow[d] \\ \text{UUAG} \end{tikzcd}$$

GAUUAGAUUG

PathEnc [Kingsford & Patro 2015]

GAUU

↓

AUUA

↓

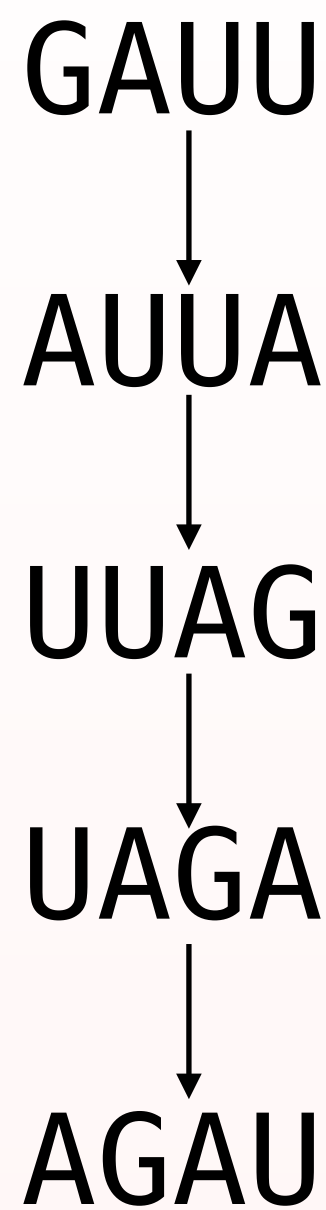
UUAG

↓

UAGA

GAUUAGAUUG

PathEnc [Kingsford & Patro 2015]



GAUUAGAUUG

PathEnc [Kingsford & Patro 2015]

GAUU

↓

AUUA

↓

UUAG

↓

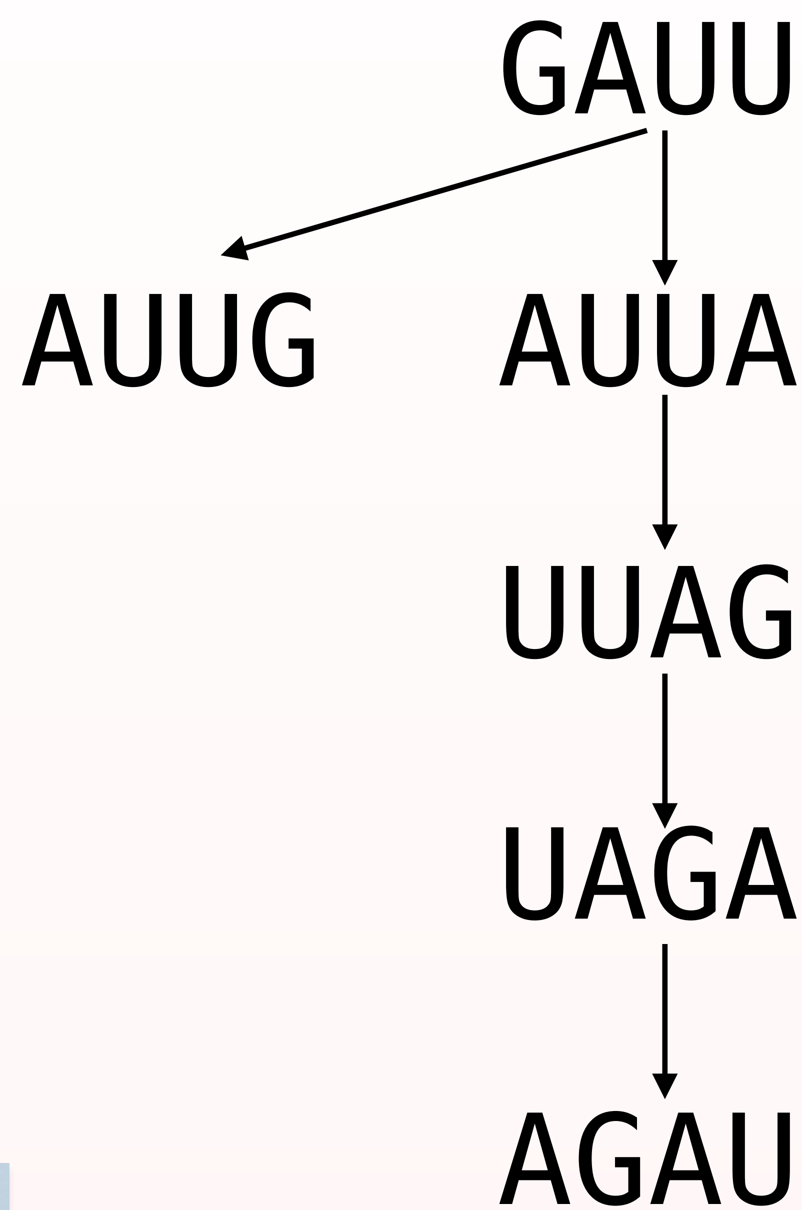
UAGA

↓

AGAU

GAUUAGAUUG

PathEnc [Kingsford & Patro 2015]



GAUUAGAUUG

PathEnc [Kingsford & Patro 2015]

PathEnc [Kingsford & Patro 2015]

Path encoding in graph G

PathEnc [Kingsford & Patro 2015]

Path encoding in graph G

1 node per k-mer

PathEnc [Kingsford & Patro 2015]

Path encoding in graph G

1 node per k-mer

edge between k-mers *(u,v)* if *v* follows *u*

PathEnc [Kingsford & Patro 2015]

Path encoding in graph G

1 node per k-mer

edge between k-mers *(u,v)* if *v* follows *u*

Each read encoded as a path in G

PathEnc [Kingsford & Patro 2015]

PathEnc [Kingsford & Patro 2015]

1st node of each read path (read head)

PathEnc [Kingsford & Patro 2015]

1st node of each read path (read head)

4-ary tree of depth k

PathEnc [Kingsford & Patro 2015]

1st node of each read path (read head)

4-ary tree of depth k

edges removed for nonexistent k-mers

PathEnc [Kingsford & Patro 2015]

1st node of each read path (read head)

4-ary tree of depth k

edges removed for nonexistent k-mers

traverse in fixed order, emitting 1 for edge

PathEnc [Kingsford & Patro 2015]

1st node of each read path (read head)

4-ary tree of depth k

edges removed for nonexistent k-mers

traverse in fixed order, emitting 1 for edge

resulting bit string gzipped

PathEnc [Kingsford & Patro 2015]

PathEnc [Kingsford & Patro 2015]

remaining nodes (read tails)

PathEnc [Kingsford & Patro 2015]

remaining nodes (read tails)

arithmetic coding

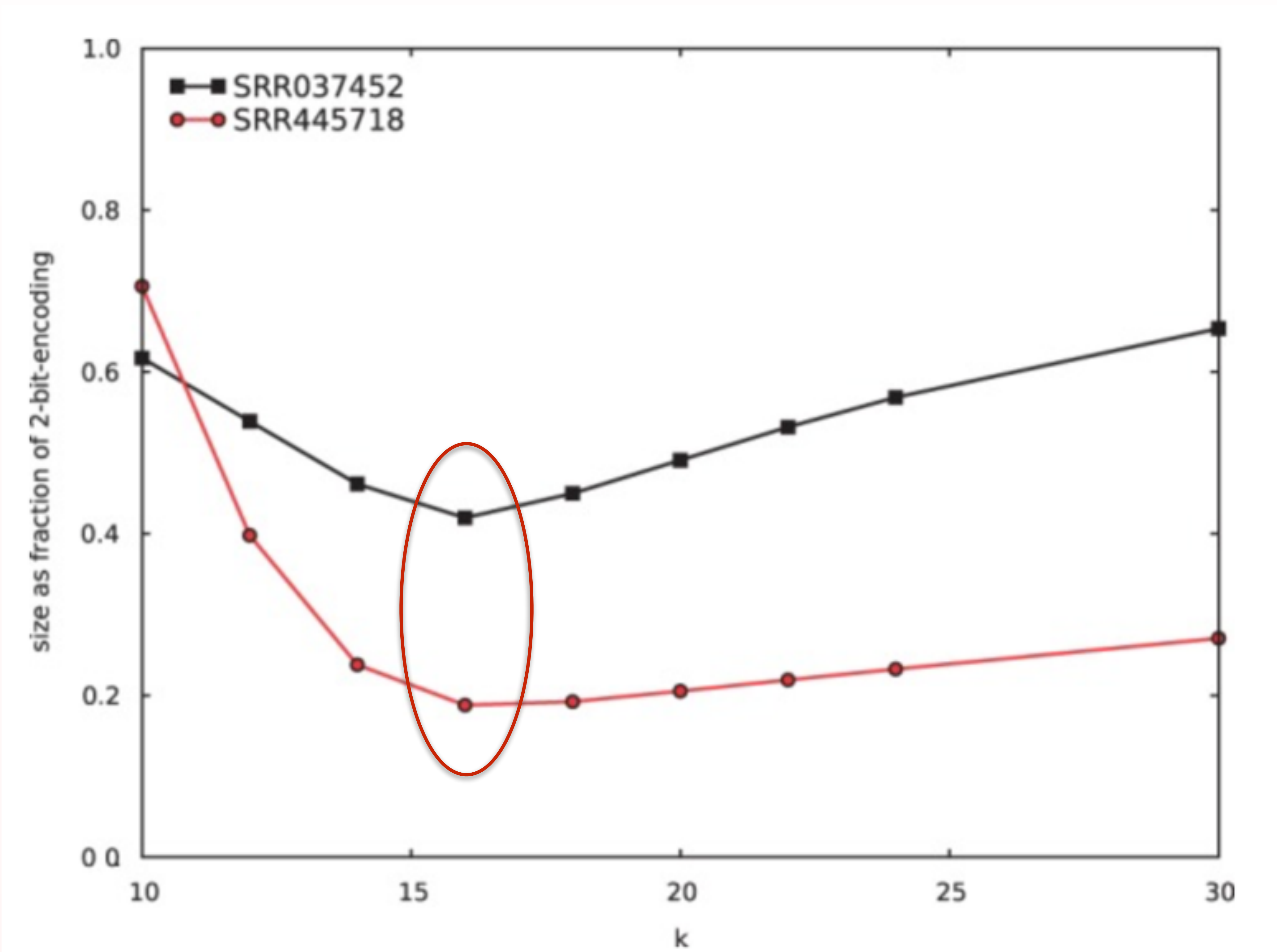
PathEnc [Kingsford & Patro 2015]

remaining nodes (read tails)

arithmetic coding

probability distribution per node in G

PathEnc [Kingsford & Patro 2015]



MINCE [Patro & Kingsford 2015]

MINCE [Patro & Kingsford 2015]

Reference-free

MINCE [Patro & Kingsford 2015]

Reference-free

Buckets reads based on *k*-mers (k=15)

MINCE [Patro & Kingsford 2015]

Reference-free

Buckets reads based on *k*-mers (*k*=15)

Replace common *k*-mer with pointer

MINCE [Patro & Kingsford 2015]

Reference-free

Buckets reads based on *k*-mers (*k*=15)

Replace common *k*-mer with pointer

Reorder reads within each bucket

MINCE [Patro & Kingsford 2015]

Reference-free

Buckets reads based on *k*-mers (*k*=15)

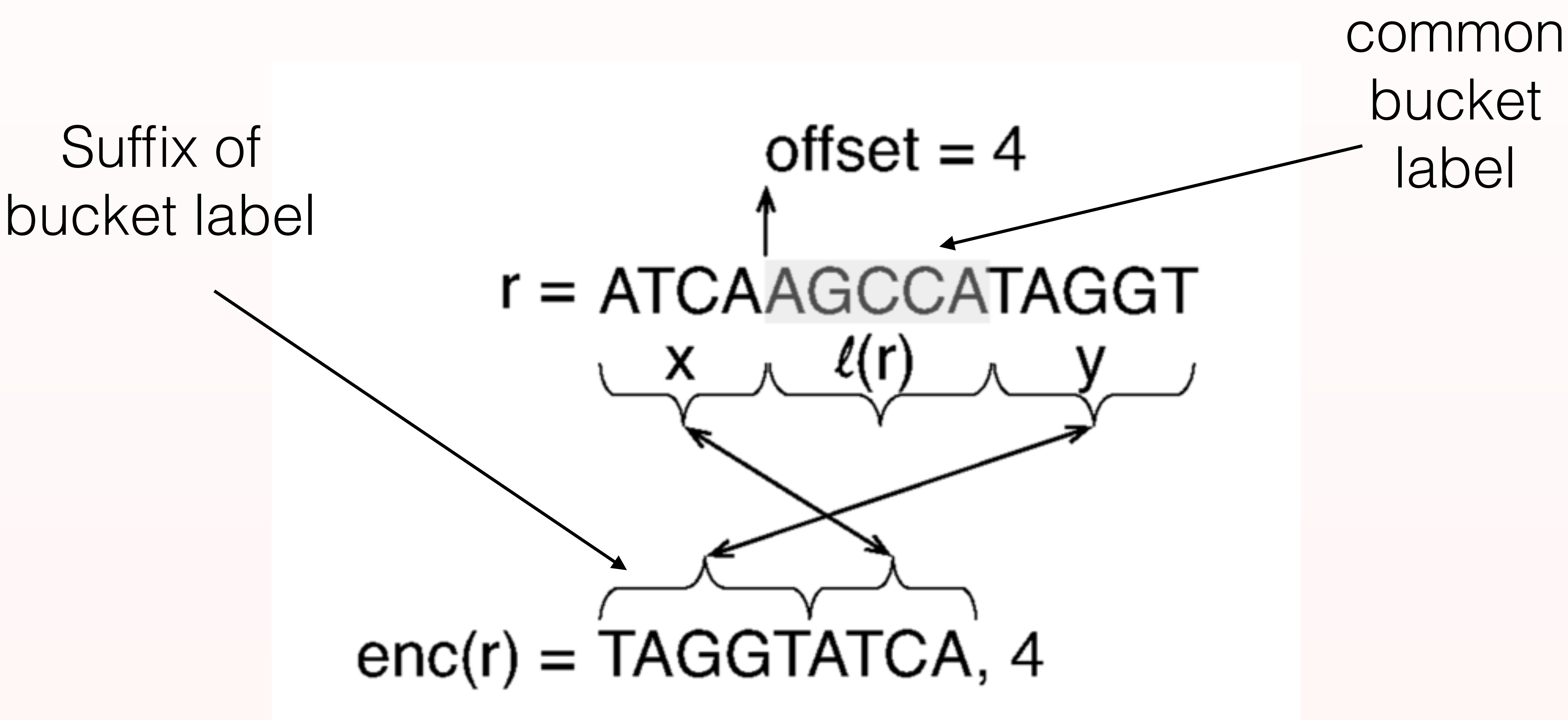
Replace common *k*-mer with pointer

Reorder reads within each bucket

This boosts lzip performance

MINCE [Patro & Kingsford 2015]

Split-swap read transformation

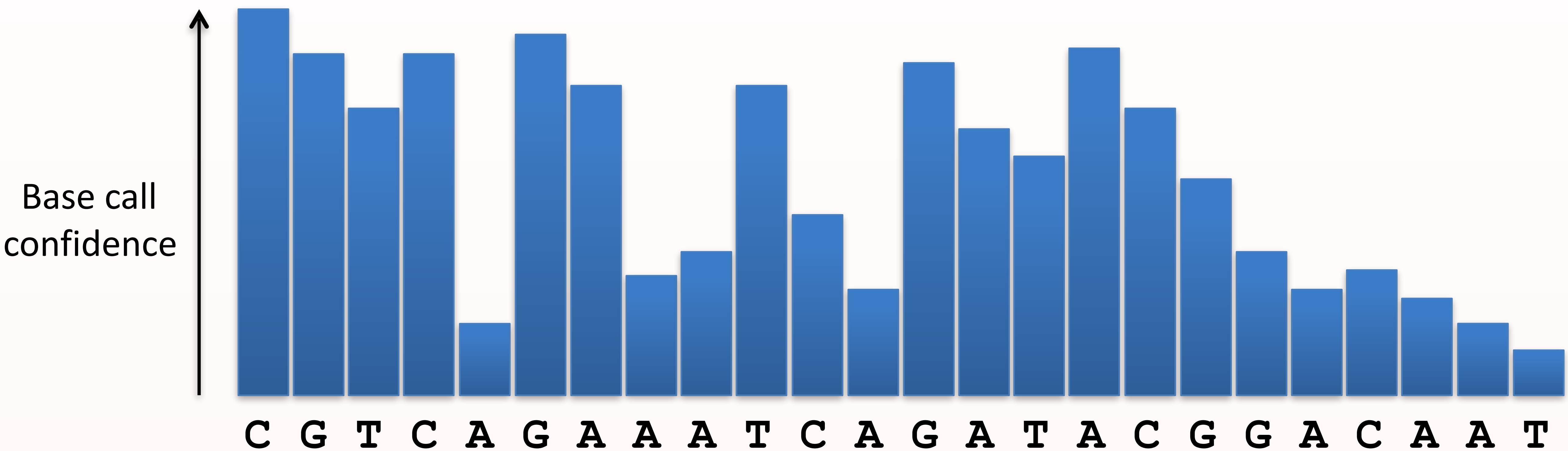


The Quality Score Problem

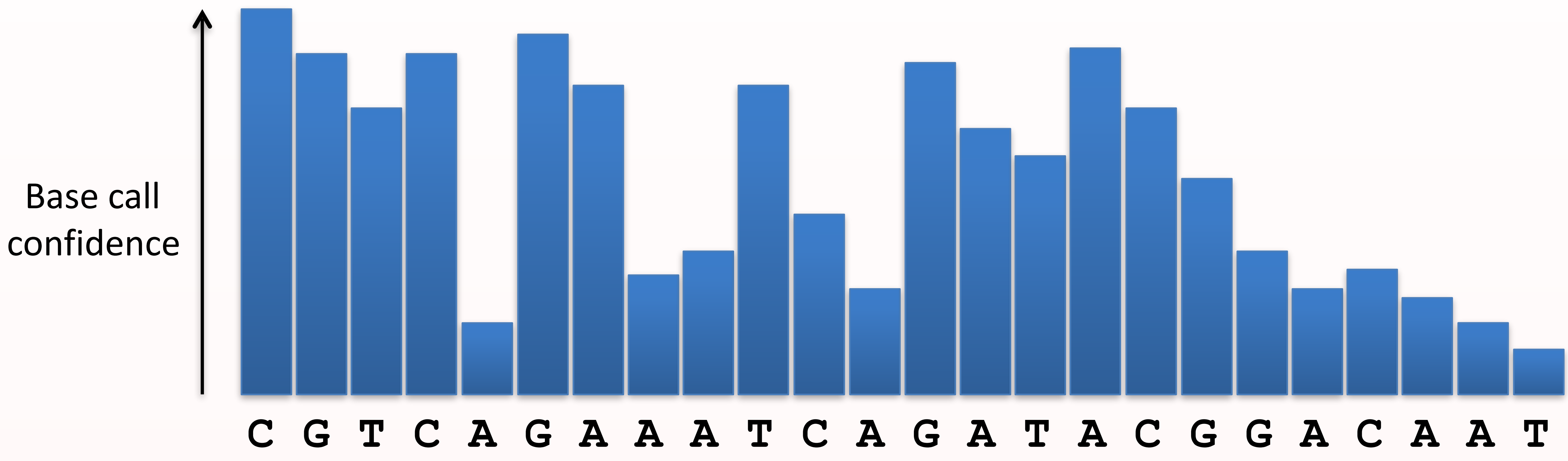
The Quality Score Problem

C G T C A G A A A T C A G A T A C G G A C A A T

The Quality Score Problem



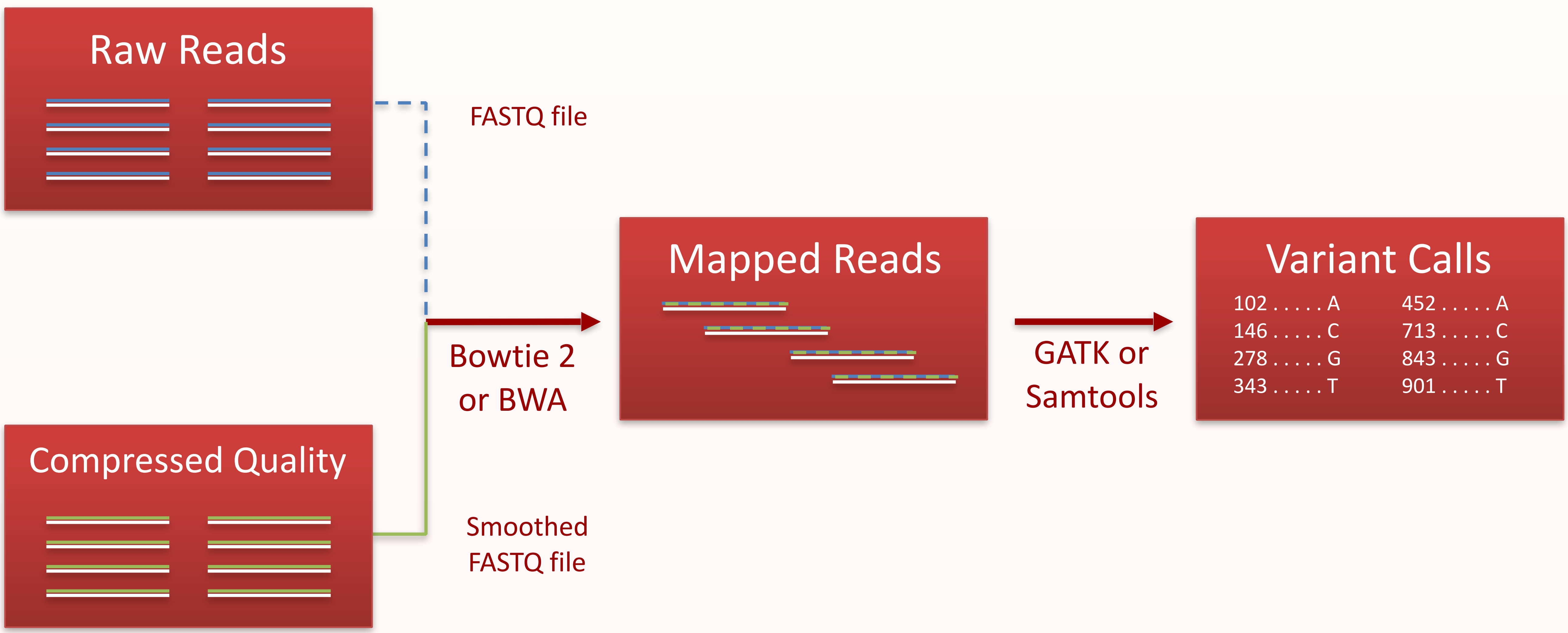
The Quality Score Problem



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

Quality scores in SNP-calling

Quality scores in SNP-calling



Quality score compressors

Quality score compressors

Greater dynamic range than sequence

Quality score compressors

Greater dynamic range than sequence

Can be lossless or lossy

Quality score compressors

Greater dynamic range than sequence

Can be lossless or lossy

Focus here on lossy

8-binning [Illumina]

Universally collapse dynamic range 40 => 8

Small effect on SNP calling error

Available in CRAM

QS bins	New value
N (no call)	N (no call)
2-9	6
10-19	15
20-24	22
25-29	27
30-34	33
35-39	37
≥40	40

SCALCE [Hach et al. 2012]

SCALCE [Hach et al. 2012]

Lossy compression of quality scores

SCALCE [Hach et al. 2012]

Lossy compression of quality scores

Neighboring QSs often similar

SCALCE [Hach et al. 2012]

Lossy compression of quality scores

Neighboring QSs often similar

Reduce dynamic range

SCALCE [Hach et al. 2012]

Lossy compression of quality scores

Neighboring QSs often similar

Reduce dynamic range

Arithmetic coding

SCALCE [Hach et al. 2012]

Lossy compression of quality scores

Neighboring QSs often similar

Reduce dynamic range

Arithmetic coding

Small loss (<0.1%) of SNP calling accuracy

QVZ [Malysa, et al. 2015]

QVZ [Malysa, et al. 2015]

Based on rate-distortion theory

Discard as little information as possible for a desired bit rate

Key insight: neighboring QSs are likely to be correlated

Illumina reads often have lower QSs at end

QVZ [Malysa, et al. 2015]

QVZ [Malysa, et al. 2015]

1. Compute the empirical transition probabilities of a Markov-1 model

QVZ [Malysa, et al. 2015]

1. Compute the empirical transition probabilities of a Markov-1 model

2. Construct a codebook using the Lloyd-Max algorithm

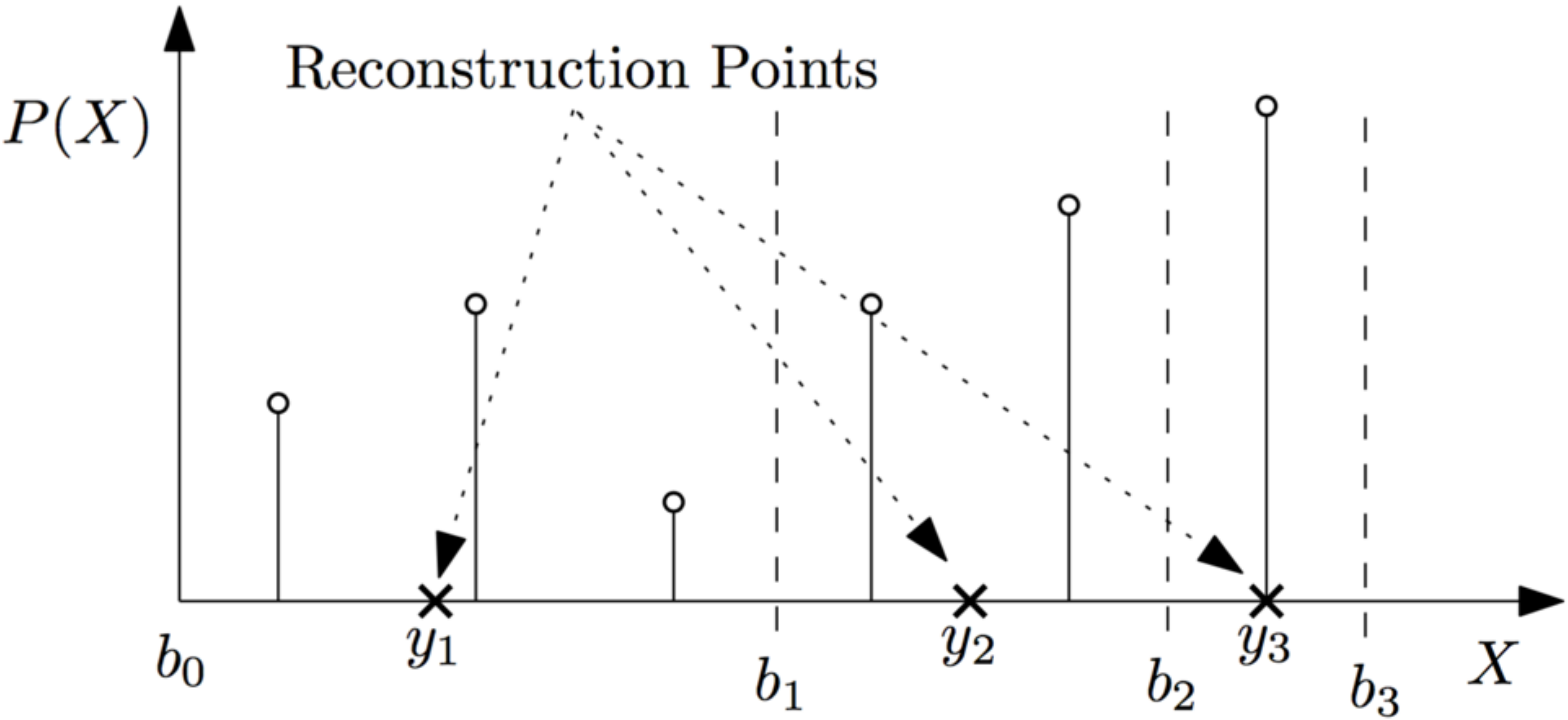
QVZ [Malysa, et al. 2015]

1. Compute the empirical transition probabilities of a Markov-1 model

2. Construct a codebook using the Lloyd-Max algorithm

3. Quantize the input using the codebook, use arithmetic encoder

QVZ [Malysa, et al. 2015]



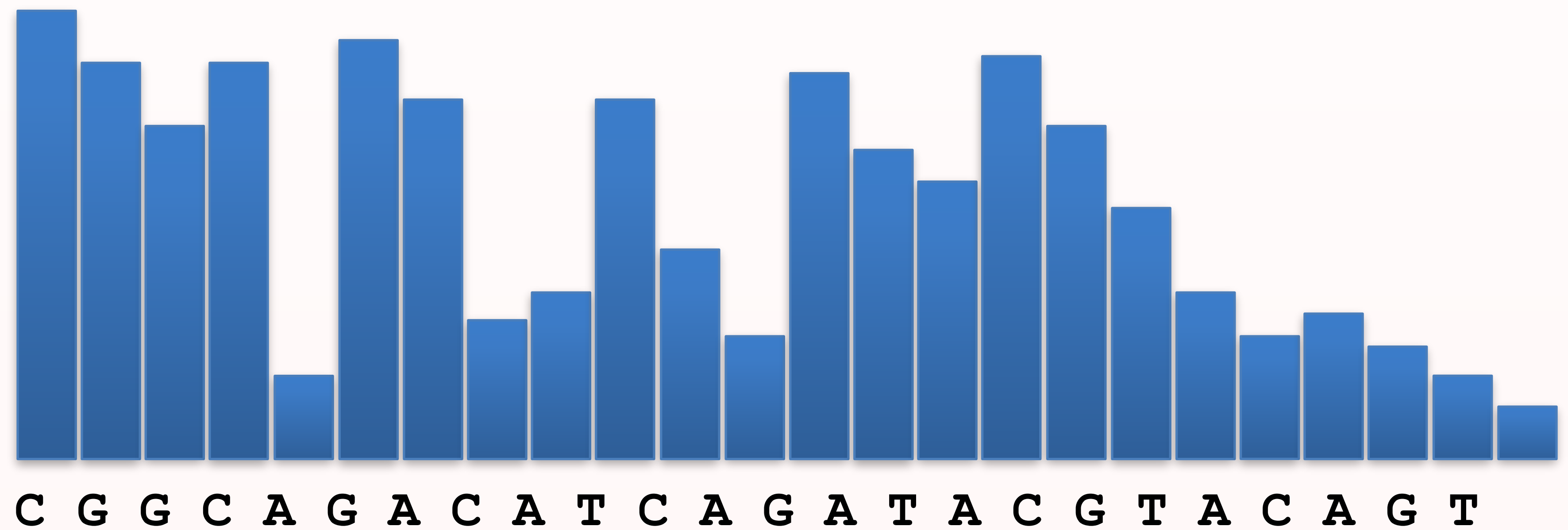
Quartz [Yu, et al. 2015]

Preprocessing

Quartz [Yu, et al. 2015]

Preprocessing

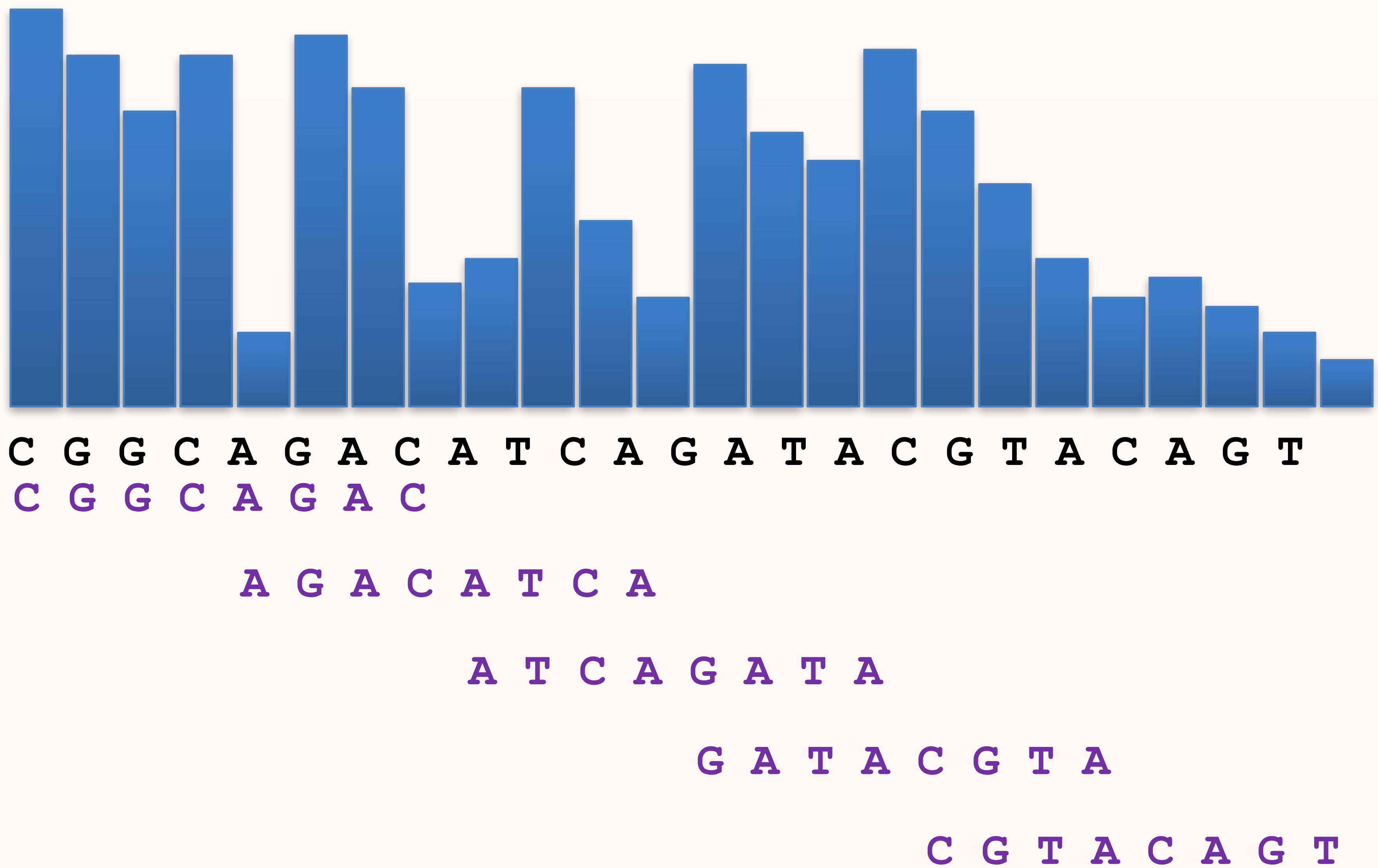
Compression



Quartz [Yu, et al. 2015]

Preprocessing

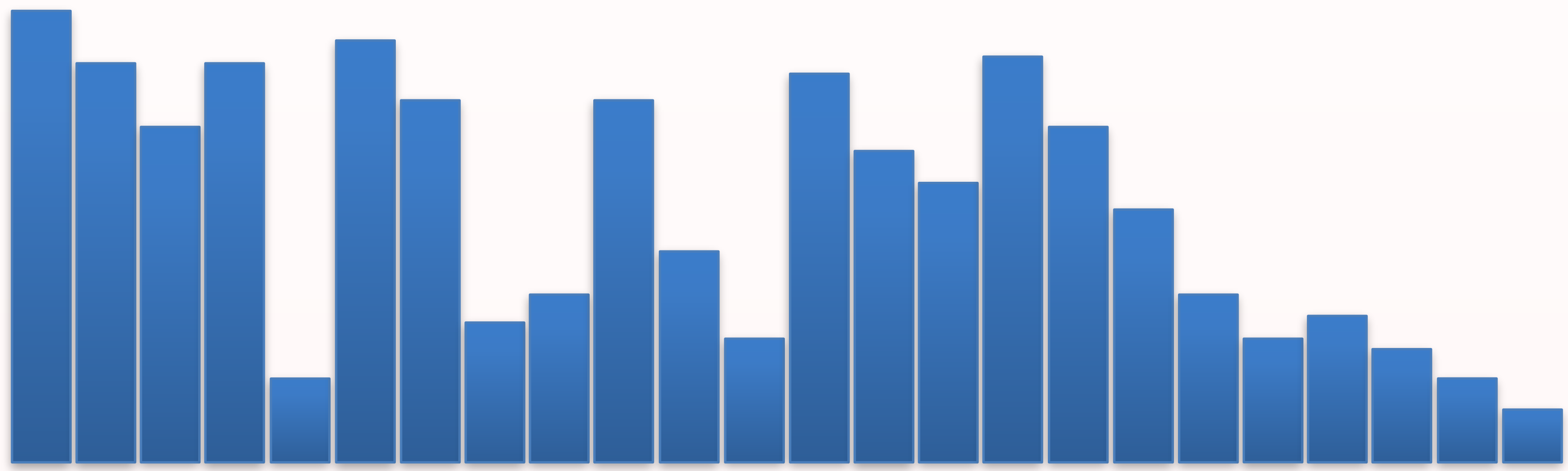
Compression



Quartz [Yu, et al. 2015]

Preprocessing

Compression



C G G C A G A C A T C A G A T A C G T A C A G T

C G G C A G A C

C G T C A G A C

A G A C A T C A

A G A A A T C A

A T C A G A T A

A T C A G A T A

G A T A C G T A

G A T A C G G A

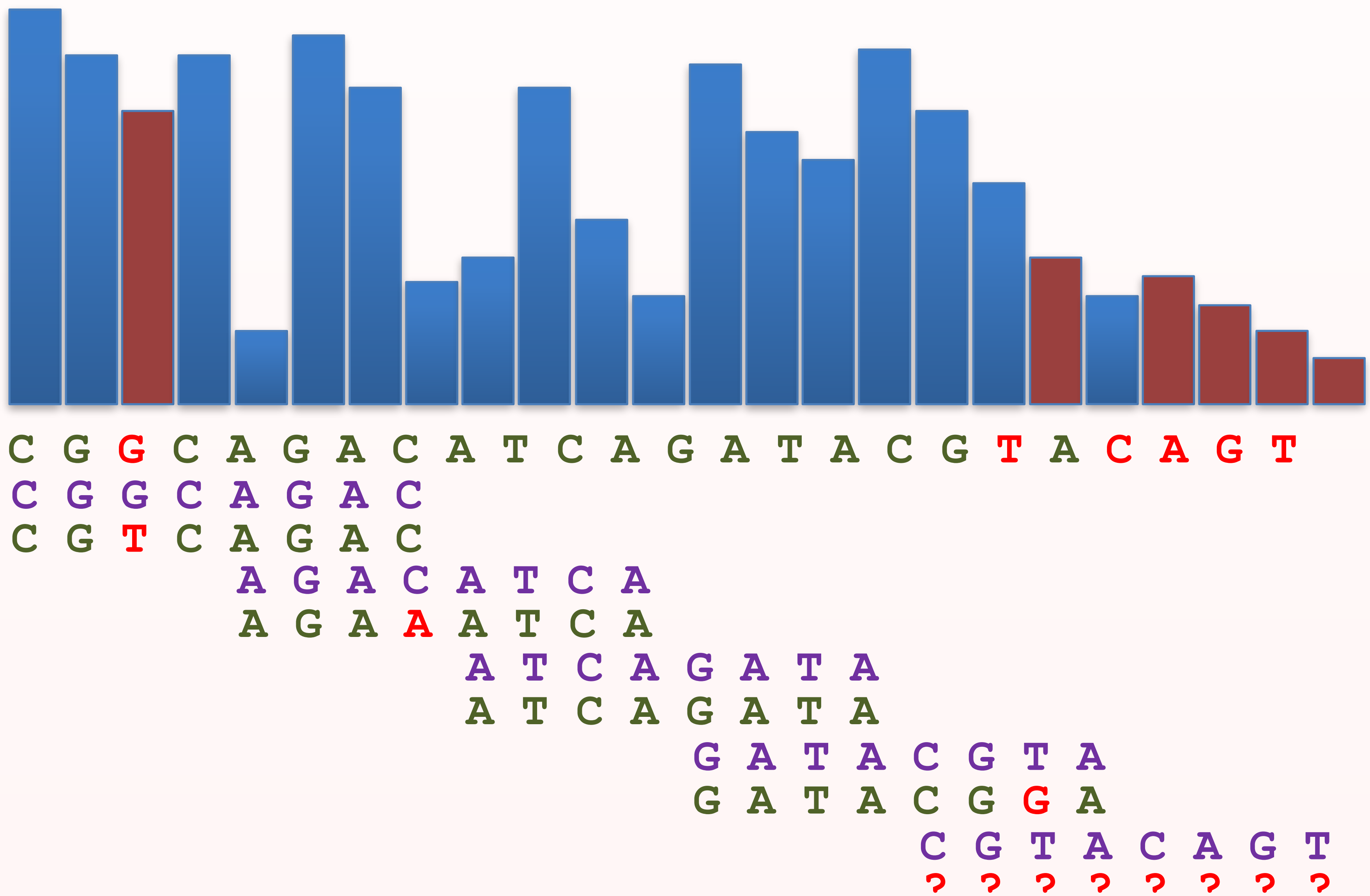
C G T A C A G T

? ? ? ? ? ? ? ?

Quartz [Yu, et al. 2015]

Preprocessing

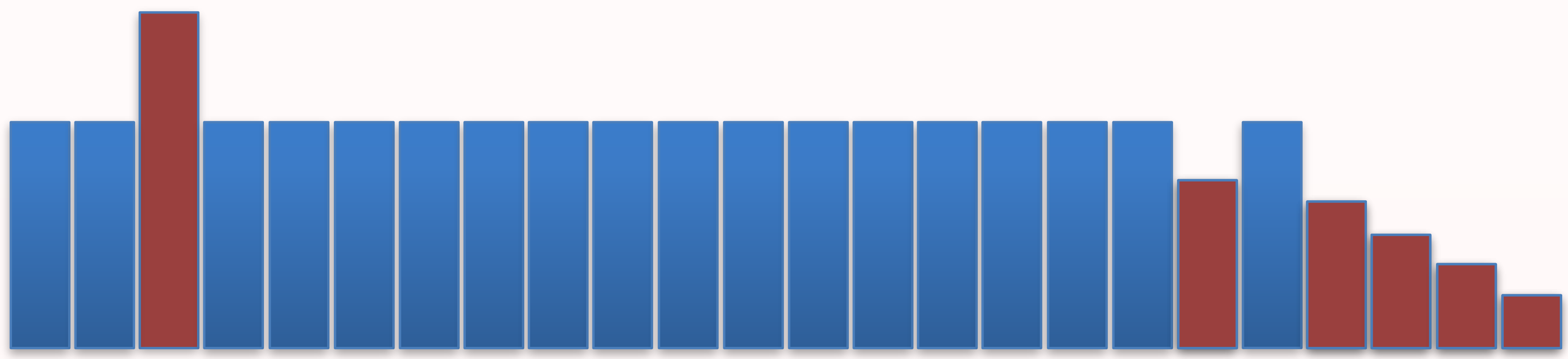
Compression



Quartz [Yu, et al. 2015]

Preprocessing

Compression



Approximate k-mer search

Naïve approaches

Approximate k-mer search
Naïve approaches

- Need to quickly find all *3k* Hamming neighbors of each k-mer from the read in the dictionary to identify mis-matched bases.

Approximate k-mer search
Naïve approaches

- Need to quickly find all *3k* Hamming neighbors of each k-mer from the read in the dictionary to identify mis-matched bases.
- Naïve approaches
  - Sorted list
    - Memory efficient but binary search is CPU and cache inefficient
  - Hash tables
    - Faster CPU-wise, but memory and cache inefficient

Approximate k-mer search

Locality sensitive hashing

An (R, cR, P_1, P_2) -sensitive LSH family F of hash functions $h : M \to S$ is defined if

$\forall p, q \in M$, a uniformly random $h \in F$ satisfies:

if $||p - q|| \leq R$ then $\mathbb{P}\left(h(p) = h(q)\right) \geq P_1$

If $||p - q|| \geq cR$ then $\mathbb{P}\left(h(p) = h(q)\right) \leq P_2$

Project k-mers onto random $\frac{k}{2}$-mers, forming a $(1, c, \frac{1}{2}, 2^{c-1})$-sensitive family of hash functions under the Hamming metric

Approximate k-mer search

Double hashing for fun and profit

Approximate k-mer search

Double hashing for fun and profit

- Notice that each $h : M \rightarrow S$ comes with an orthogonal projection $h' : M \rightarrow S$

Approximate k-mer search

Double hashing for fun and profit

- Notice that each $h : M \to S$ comes with an orthogonal projection $h' : M \to S$
- Also, if $||p, q|| \leq 1$ then by counting, at least one of $h(p) = h(q)$ or $h'(p) = h'(q)$ must hold.

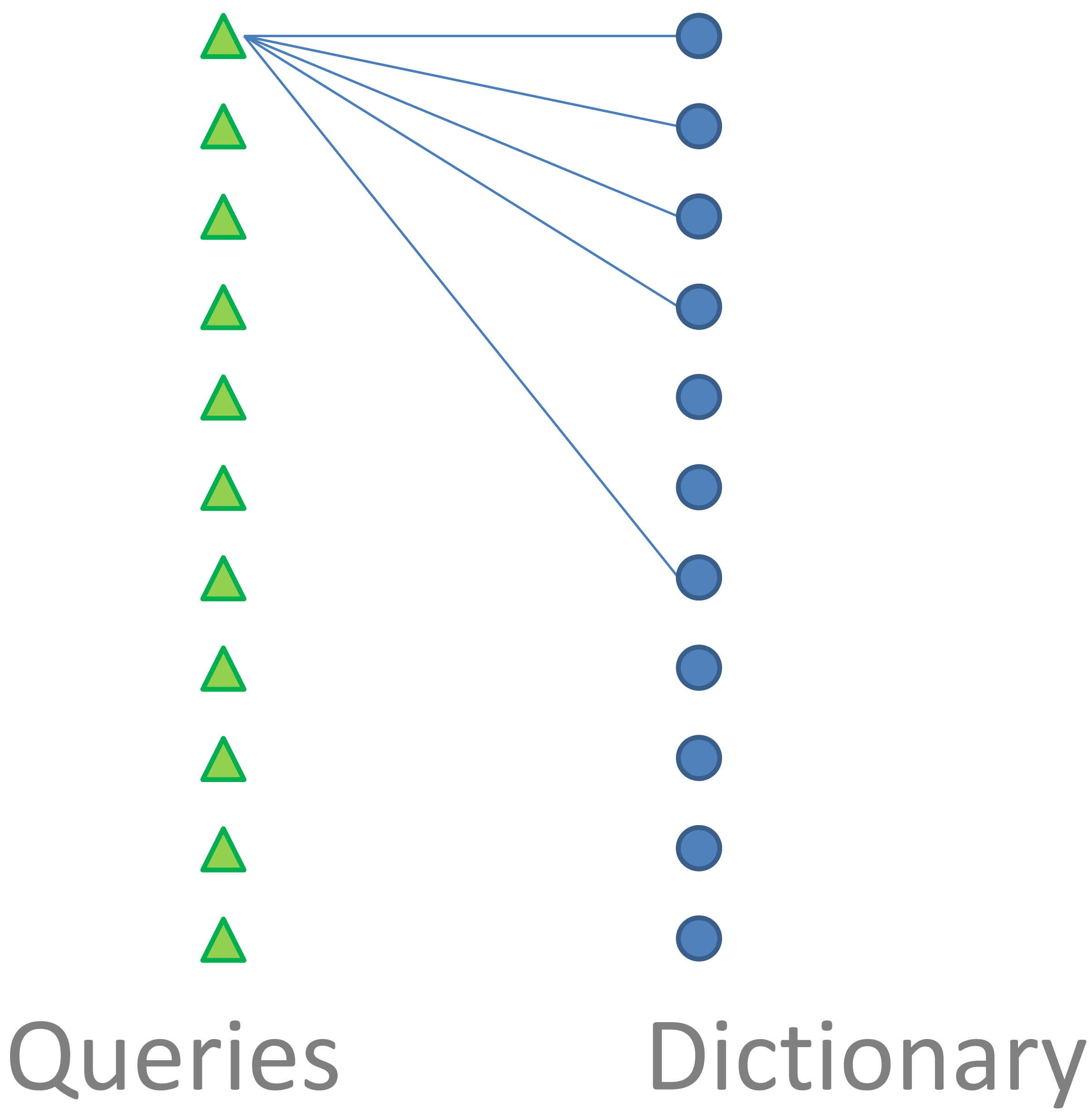
Approximate k-mer search

Double hashing for fun and profit

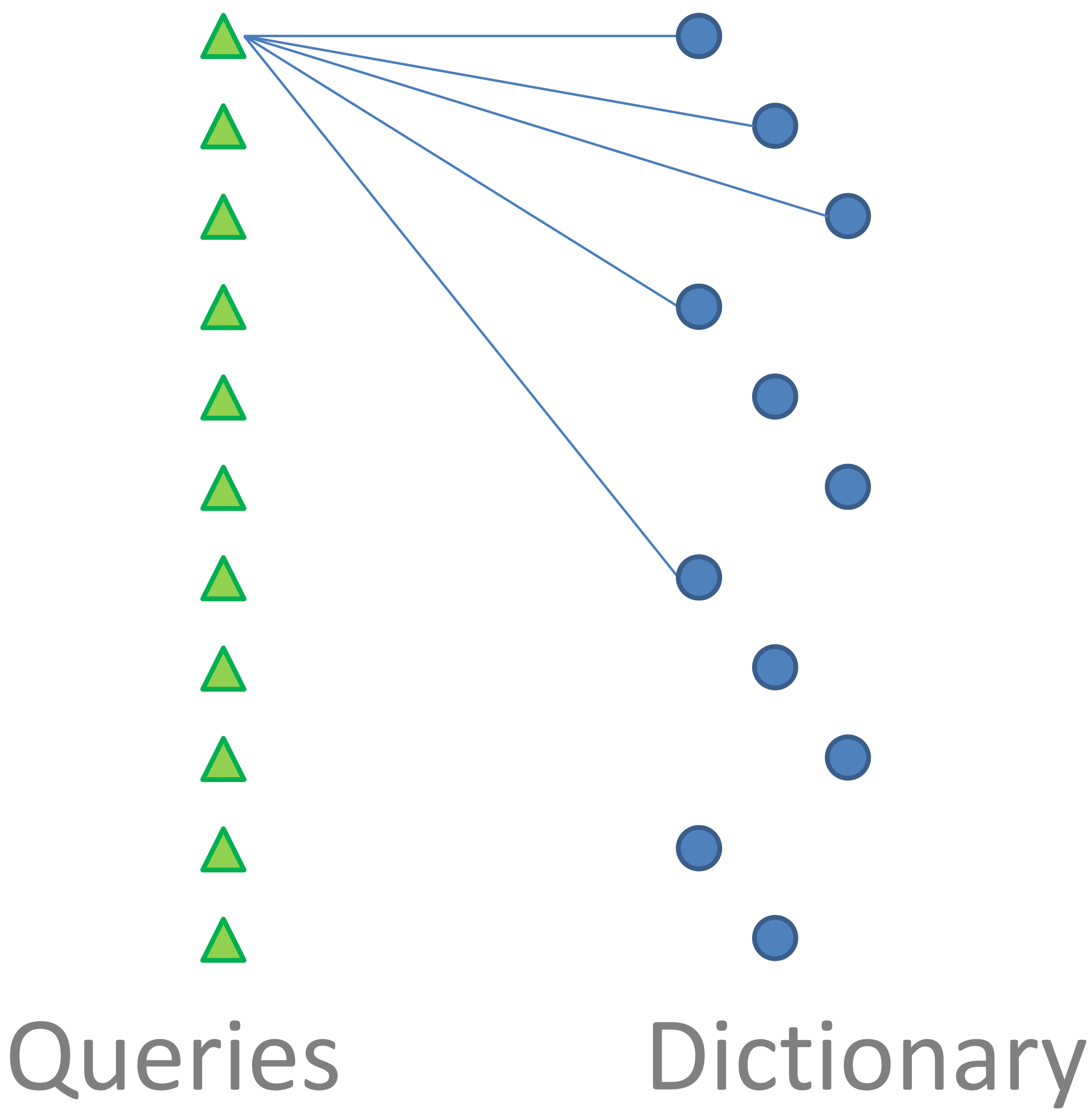
- Notice that each $h : M \to S$ comes with an orthogonal projection $h' : M \to S$
- Also, if $||p, q|| \leq 1$ then by counting, at least one of $h(p) = h(q)$ or $h'(p) = h'(q)$ must hold.
- Thus, by double hashing, all Hamming neighbors of a *k*-mer can be found by looking in just two hash buckets.
  - Better cache-efficiency
  - Cheating by carefully choosing the projection and sorting the buckets gives also processor-efficiency.

Approximate k-mer search

Fast retrieval of Hamming neighbors

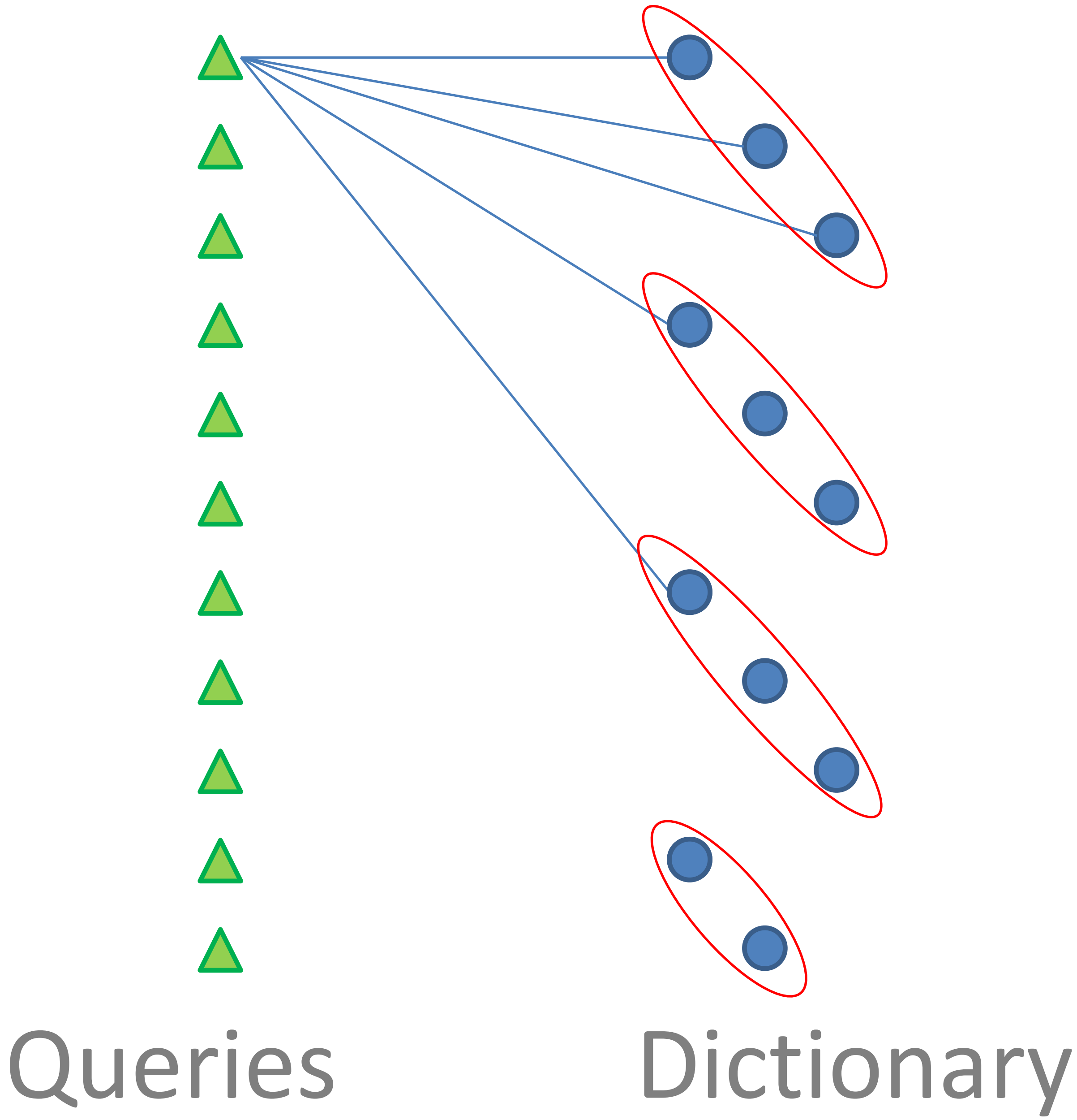


Approximate k-mer search
Fast retrieval of Hamming neighbors



Approximate k-mer search

Fast retrieval of Hamming neighbors

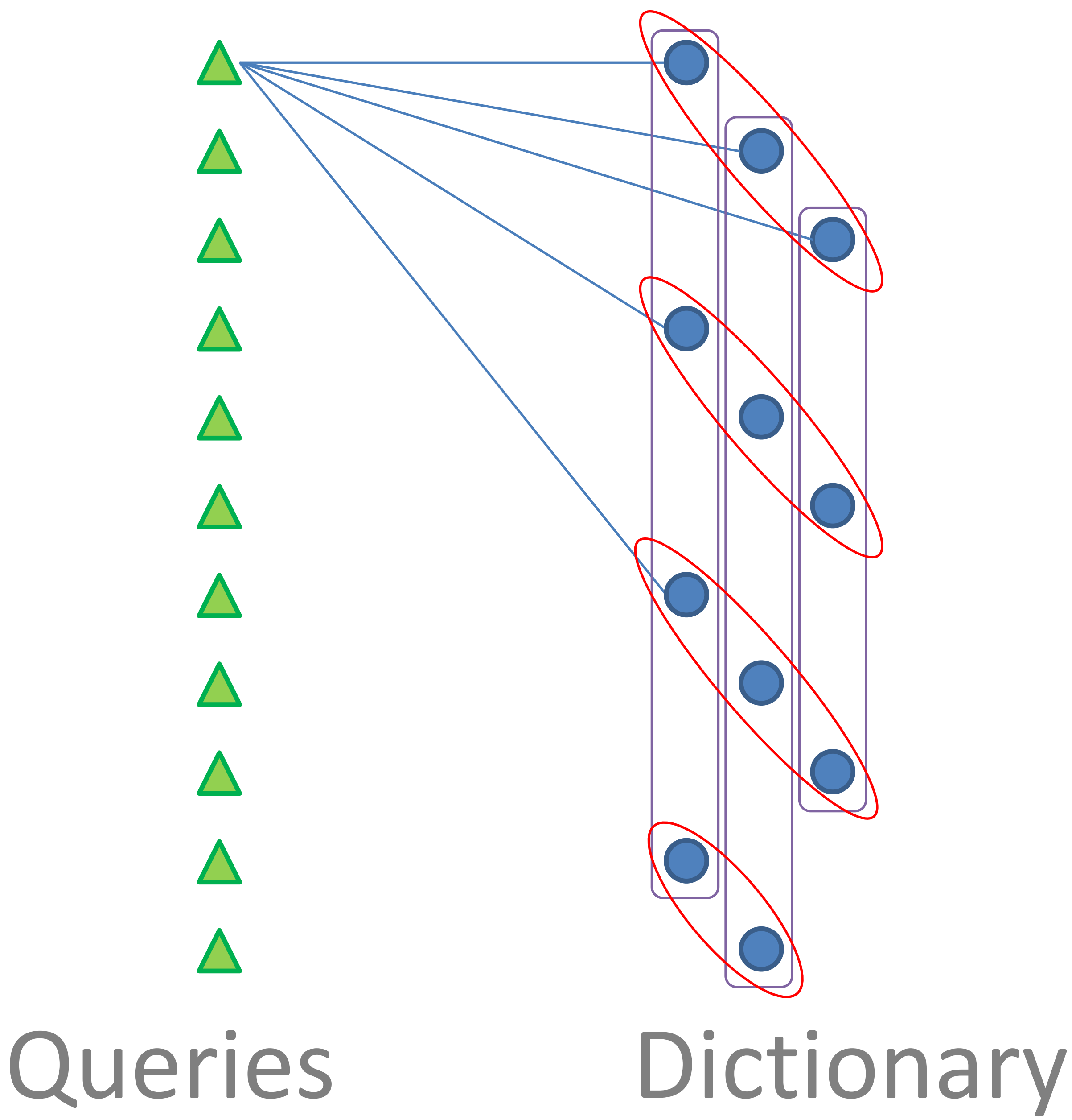


Clustered by front half of k-mer

```
C G G C A G A C
C G G C C G A C
C G G C A G T C
```

Approximate k-mer search

Fast retrieval of Hamming neighbors



Clustered by front half of k-mer

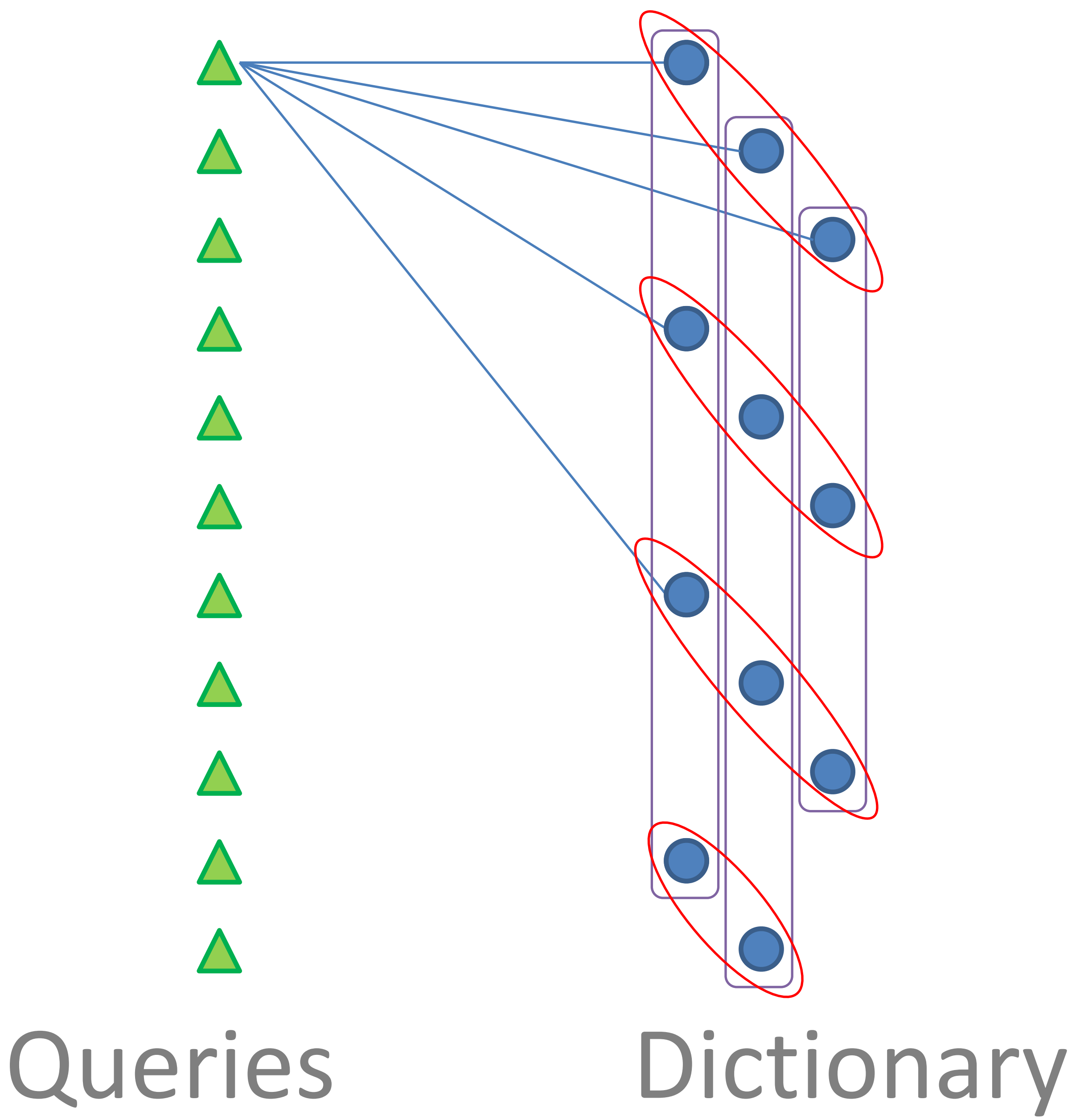
```
C G G C A G A C
C G G C C G A C
C G G C A G T C
```

Clustered by back half of k-mer

```
A G G C A G A C
G G G C A G A C
C C G C A G A C
T A T A A G A C
```

Approximate k-mer search

Fast retrieval of Hamming neighbors



Clustered by front half of k-mer

C G G C A G A C
C G G C C G A C
C G G C A G T C

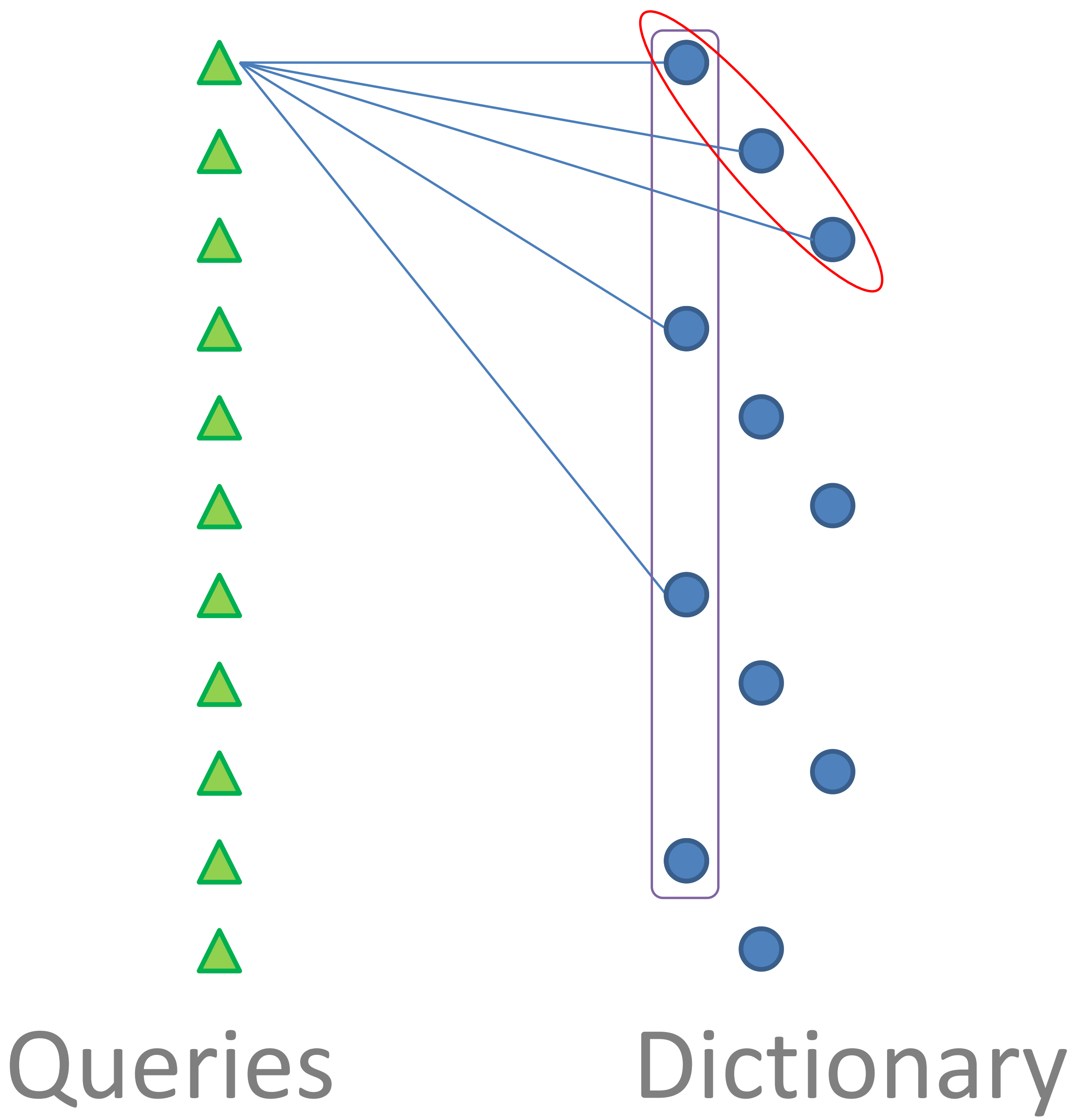
Clustered by back half of k-mer

A G G C A G A C
G G G C A G A C
C C G C A G A C
T A T A A G A C

= C G G C A G A C

Approximate k-mer search

Fast retrieval of Hamming neighbors



Clustered by front half of k-mer

C G G C A G A C
C G G C C G A C
C G G C A G T C

Clustered by back half of k-mer

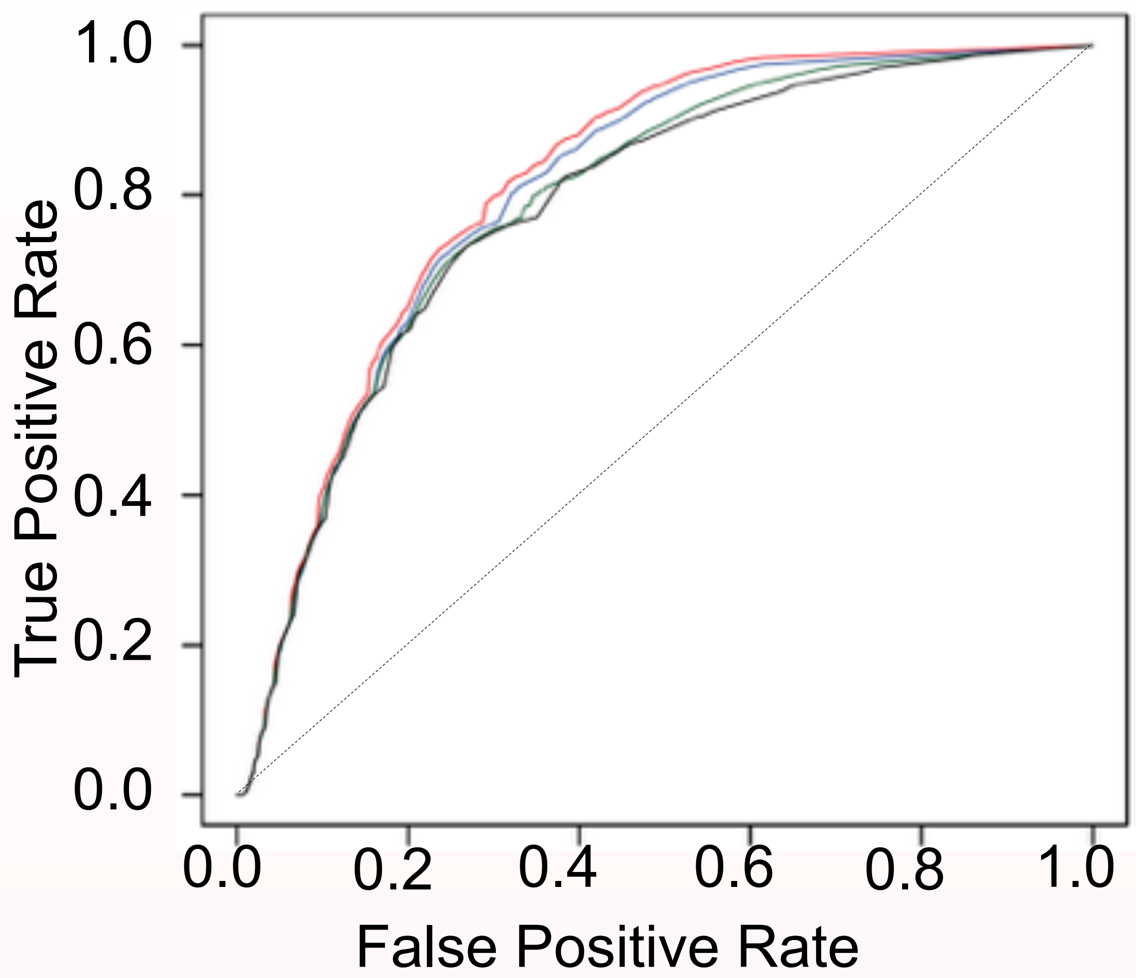
A G G C A G A C
G G G C A G A C
C C G C A G A C
T A T A A G A C

 = C G G C A G A C

Result Highlights

Comparison with other methods

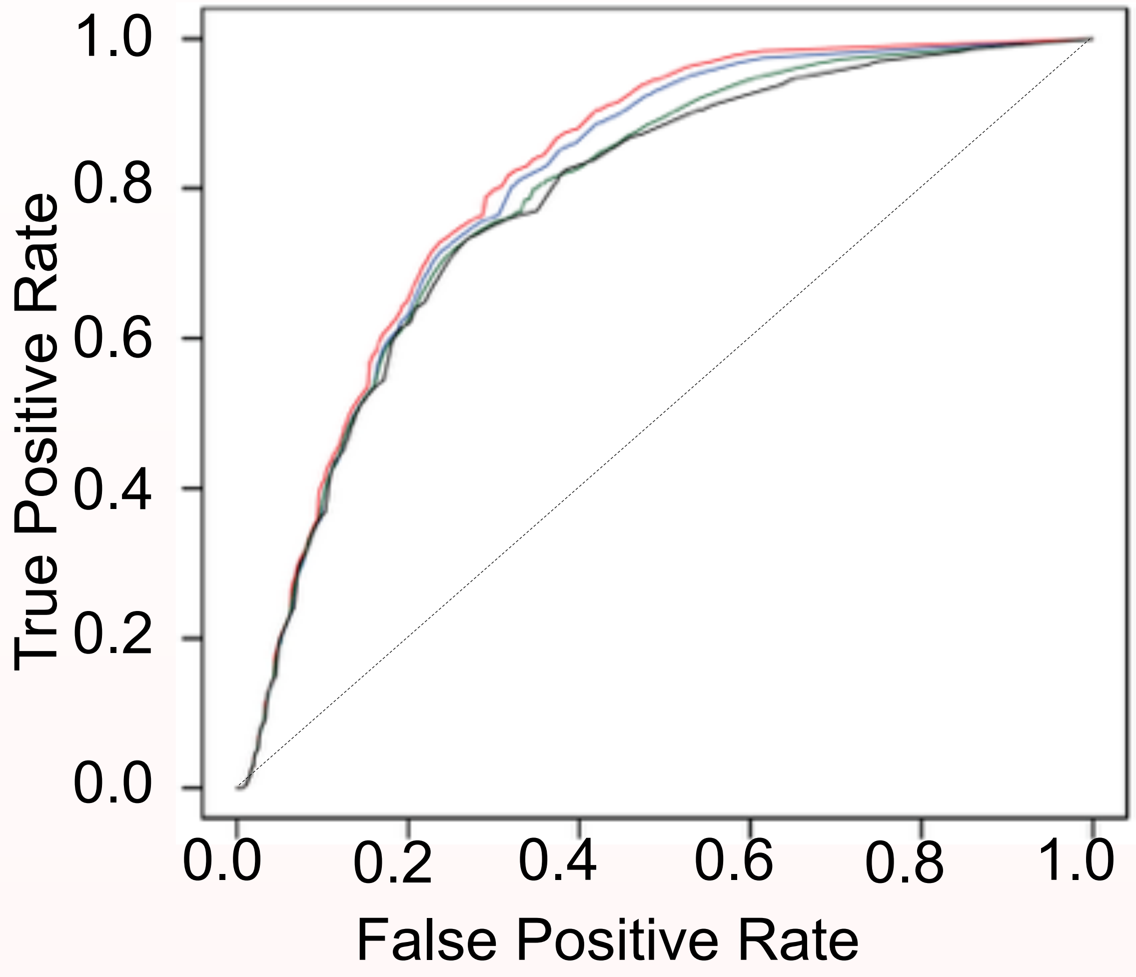
Method	Bits/Q	Time (s)	Area Under ROC Curve
Uncompressed	8	N/A	0.8254
Quartz	0.3564	**2,696**	0.8288
QualComp	0.5940	33,316	0.8053
Janin et al.	0.5376	164,702	0.8019



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

A word on k-mers

MINCE: 15-mers optimal (8-mer labels!)

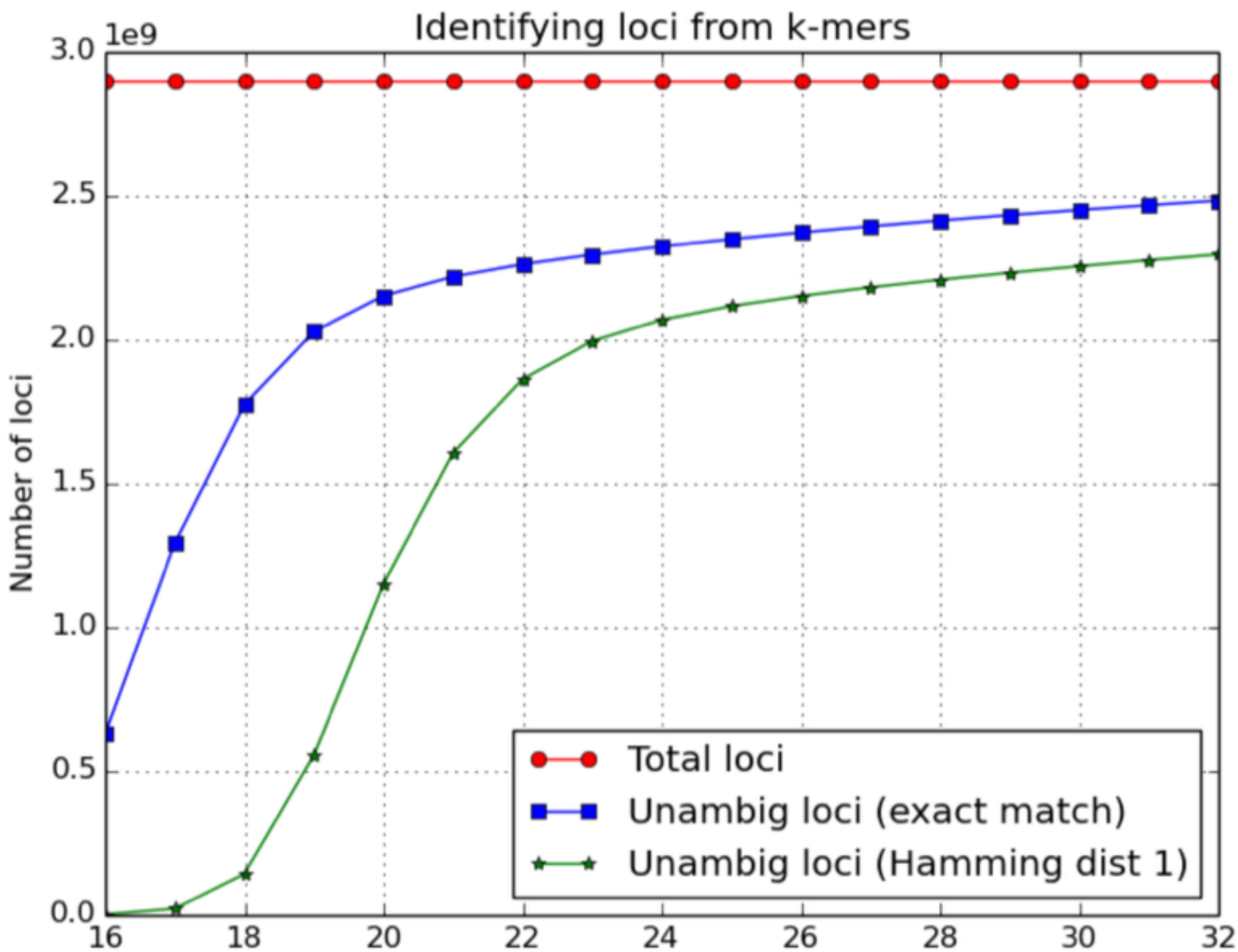
- using labels as search heuristic; don't want too many

Quartz: 32-mers optimal

- in a genome, most 8-mers may exist; most 32-mers will not
- and they tend to be unique

	k=16	k=32
unique	21.8%	85.7%
unique at Hamming 1	0.0008%	79.3%

Uniqueness of k-mers (hg19)



Why not longer k-mers?

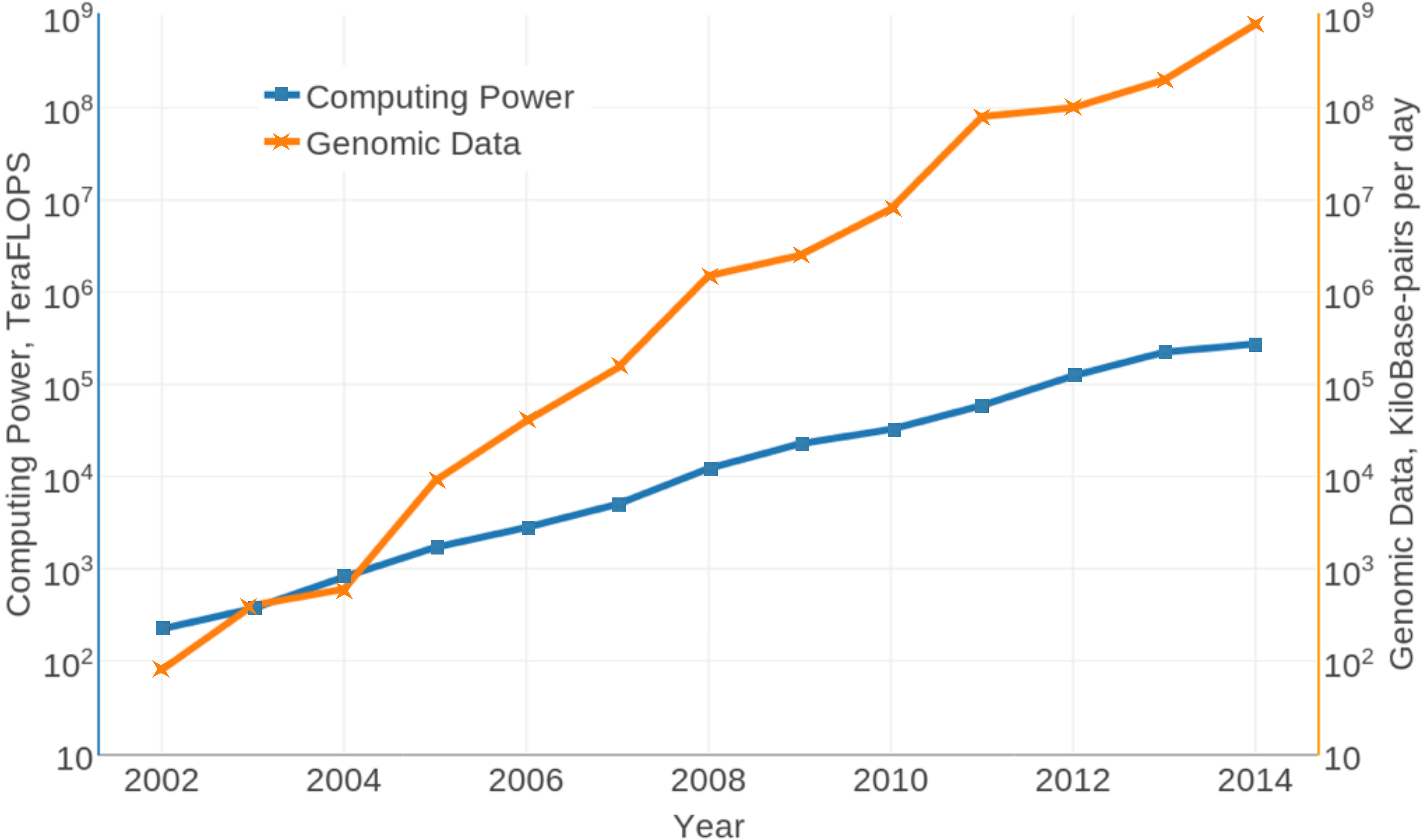
Want to ensure at most 1 sequencing error per k-mer

Assume independence of errors
Error rate of p

Likelihood of l errors $\binom{k}{l}(1-p)^{k-l}p^l$

1% error rate			2% error rate		
k=32	l≥2	2%	k=32	l≥2	13%
k=64	l≥2	13%	k=64	l≥2	36%

Compression for speed



Compressive genomics

caBLAST [Loh, et al. 2012]

caBLASTP [Daniels, et al. 2013]

BLAST uses seed-and-extend

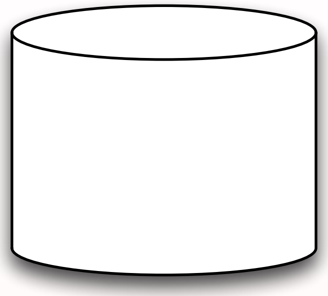
but must extend on many fruitless seeds

Use compression to reduce the search space

How does this compression work?

How does this compression work?

Coarse Database

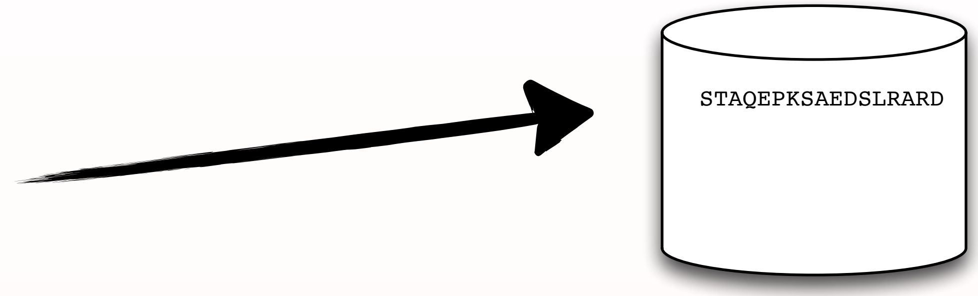


STAQEPKSAEDSLRARD

How does this compression work?

Coarse Database

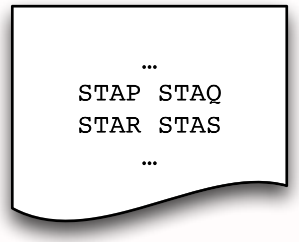
STAQEPKSAEDSLRARD



How does this compression work?

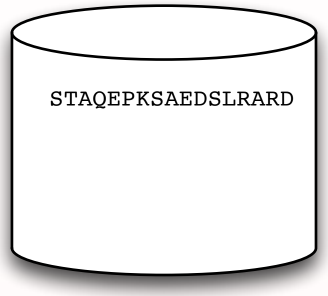
How does this compression work?

Seed Table



STAQEPKSAEDSLRARD

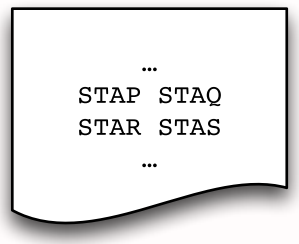
Coarse Database



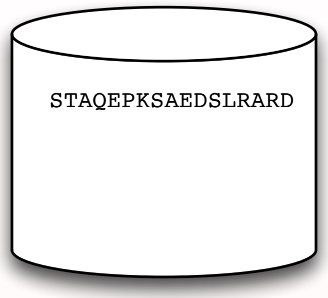
LQSTAQEPKSAEQRDSVNARDRQRNVIIAQE

How does this compression work?

Seed Table



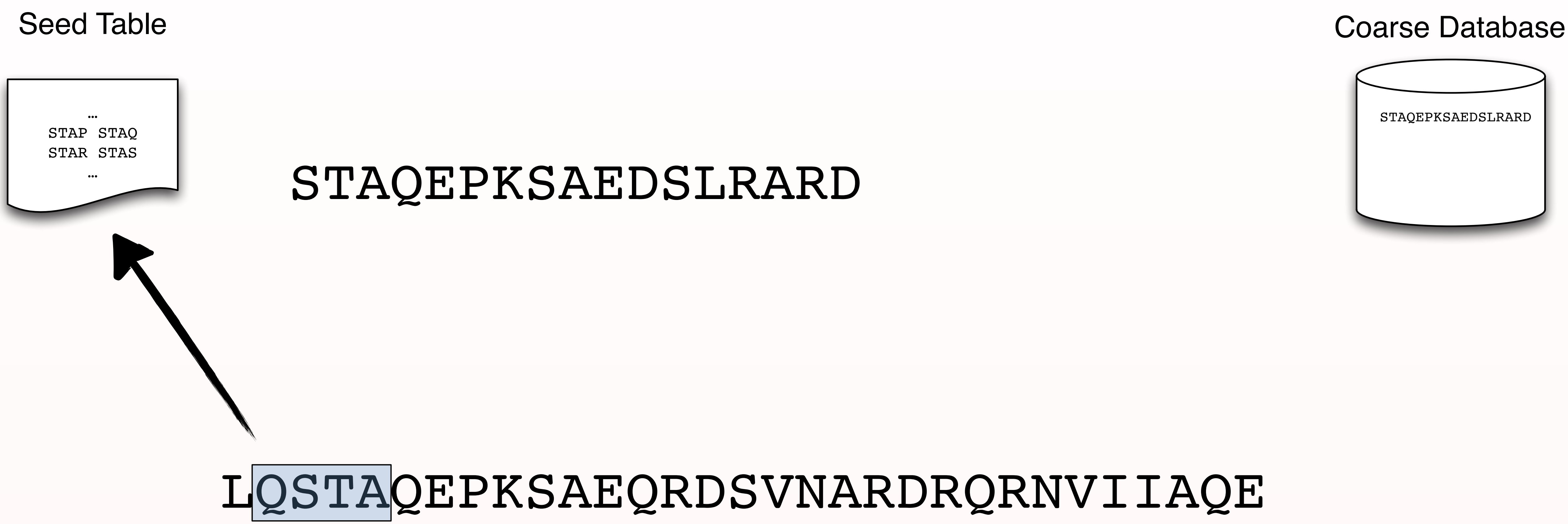
Coarse Database



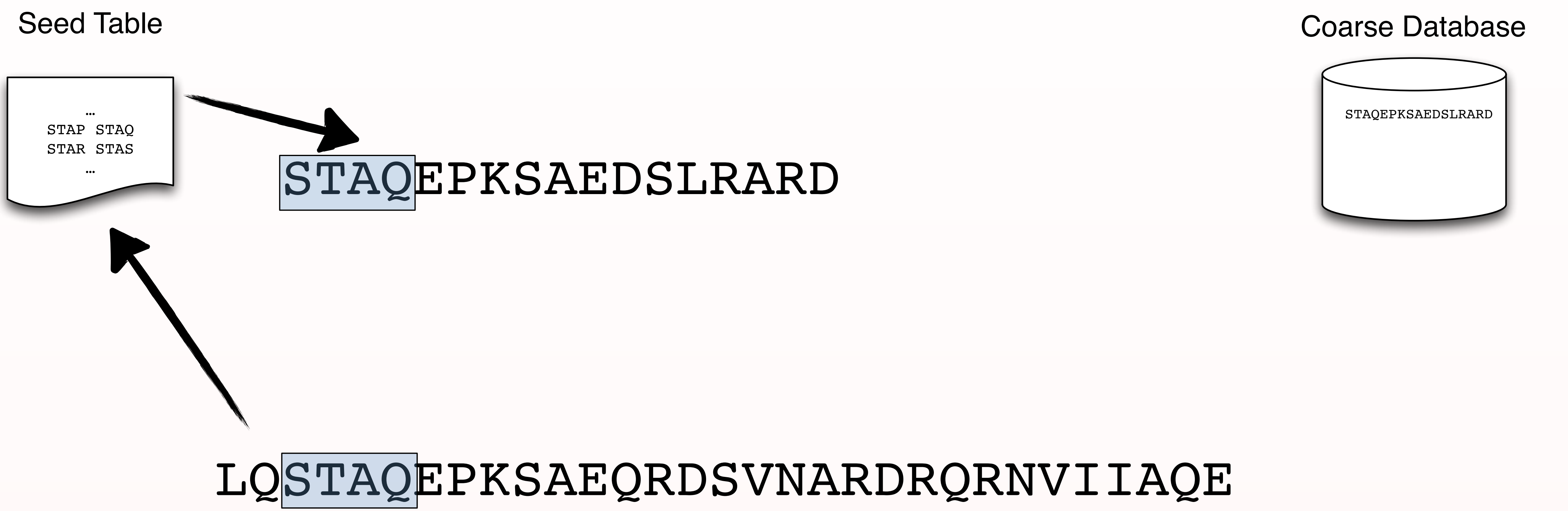
STAQEPKSAEDSLRARD

LQSTAQEPKSAEQRDSVNARDRQRNVIIAQE

How does this compression work?



How does this compression work?



How does this compression work?

Seed Table

…
STAP STAQ
STAR STAS
…

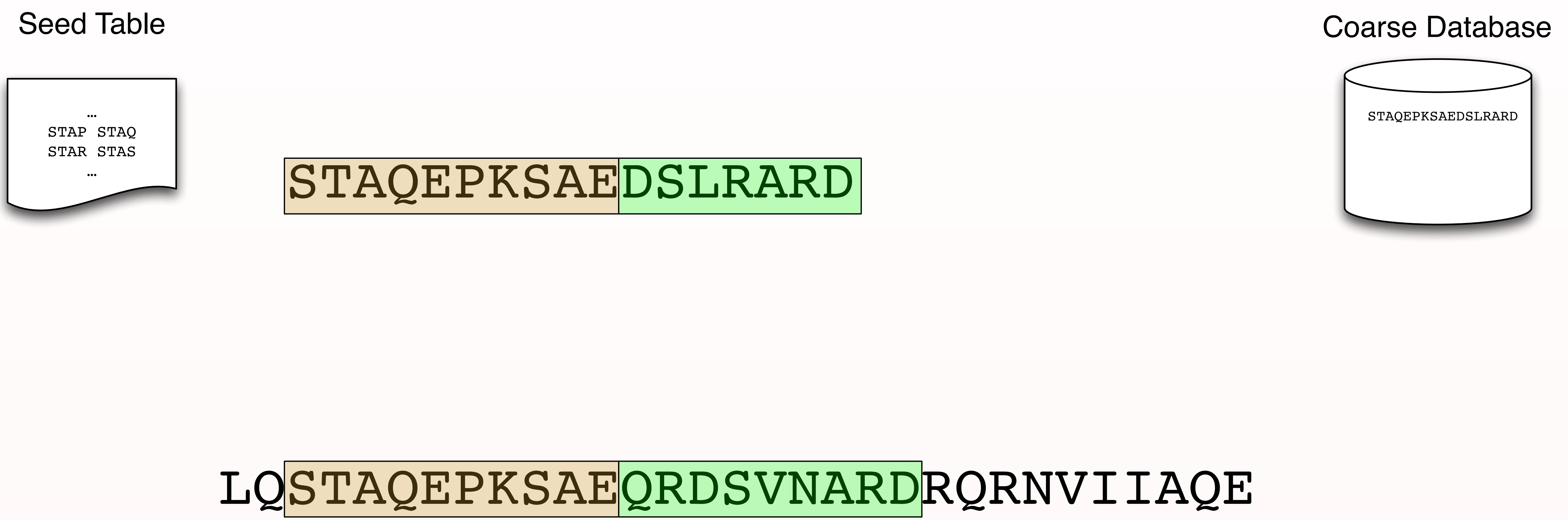
Coarse Database

STAQEPKSAEDSLRARD

$\boxed{\texttt{STAQEPKSAE}}\texttt{DSLRARD}$

$\texttt{LQ}\boxed{\texttt{STAQEPKSAE}}\texttt{QRDSVNARDRQRNVIIAQE}$

How does this compression work?



How does this compression work?

Seed Table

…
STAP STAQ
STAR STAS
…

Coarse Database

STAQEPKSAEDSLRARD

STAQEPKSAEDSLRARD

--DSLRARD

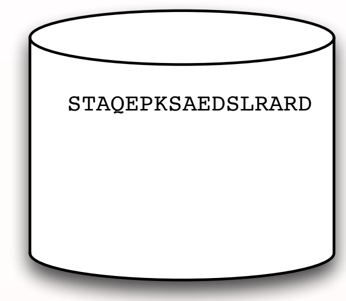
QRDSVNARD

LQSTAQEPKSAEQRDSVNARDRQRNVIIAQE

How does this compression work?

Seed Table

Coarse Database



STAQEPKSAEDSLRARD

--DSLRARD

QRDSVNARD

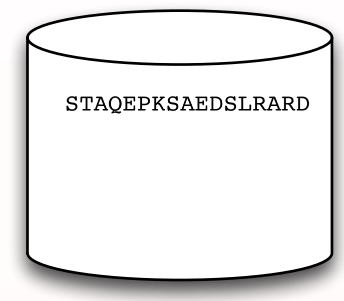
LQSTAQEPKSAEQRDSVNARDRQRNVIIAQE

LQ QR VN

How does this compression work?

Seed Table

Coarse Database



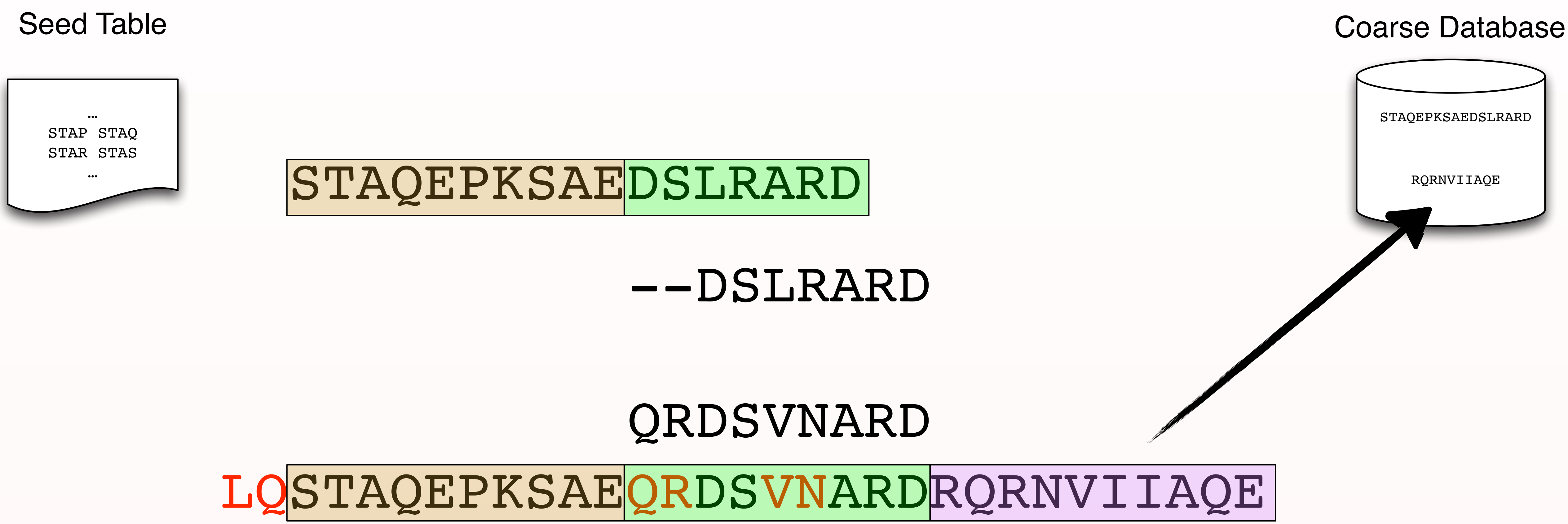
STAQEPKSAEDSLRARD

--DSLRARD

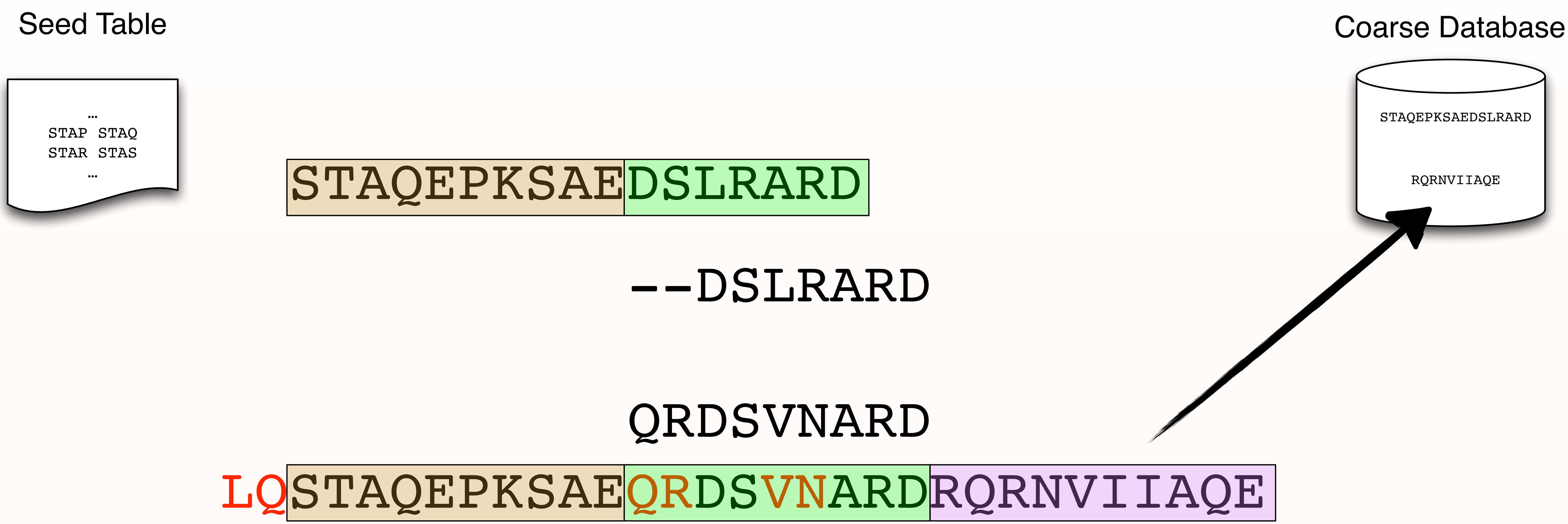
QRDSVNARD

LQSTAQEPKSAEQRDSVNARDRQRNVIIAQE

How does this compression work?



How does this compression work?



Lossless compression!

How does compressed search work?

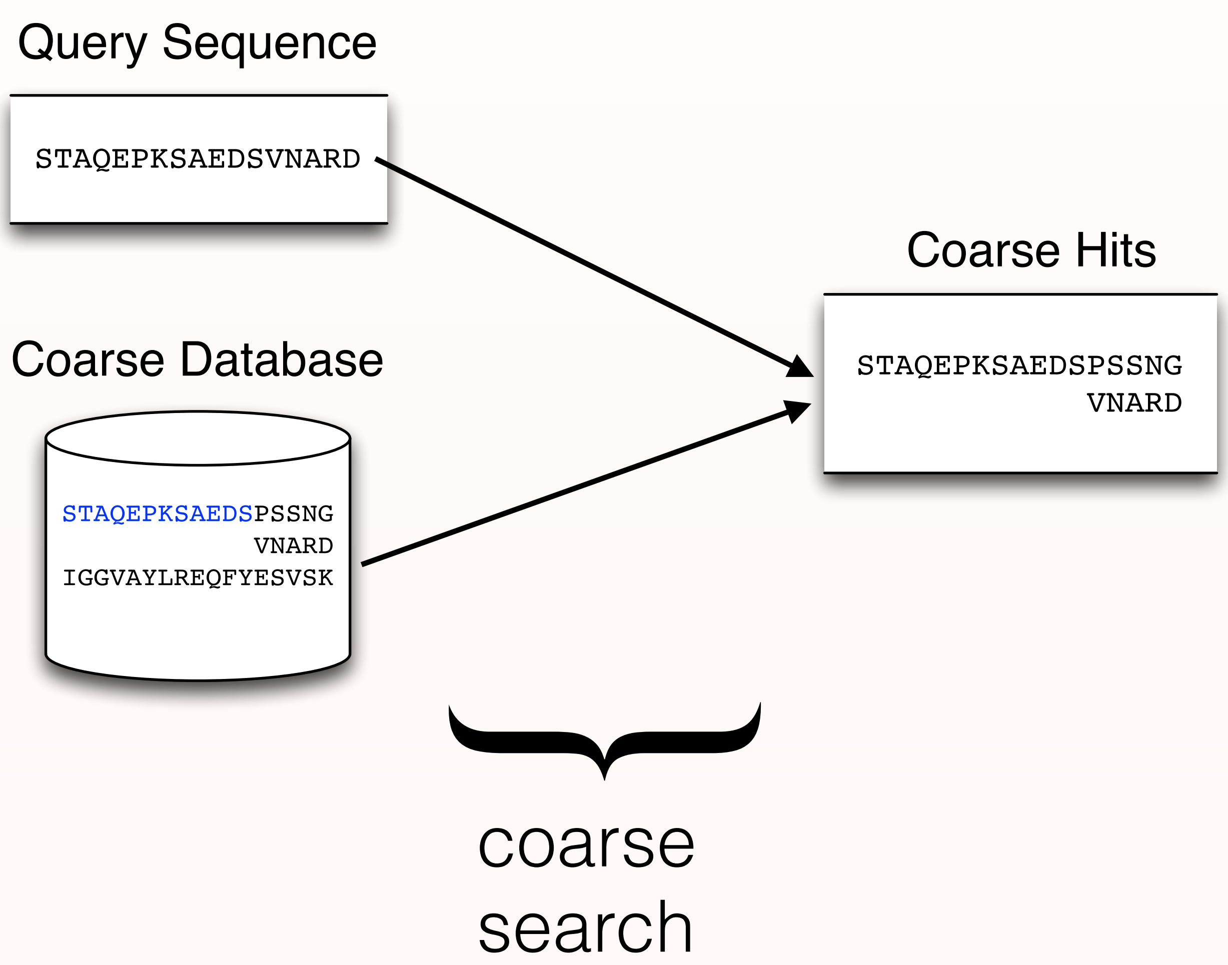
How does compressed search work?

Query Sequence

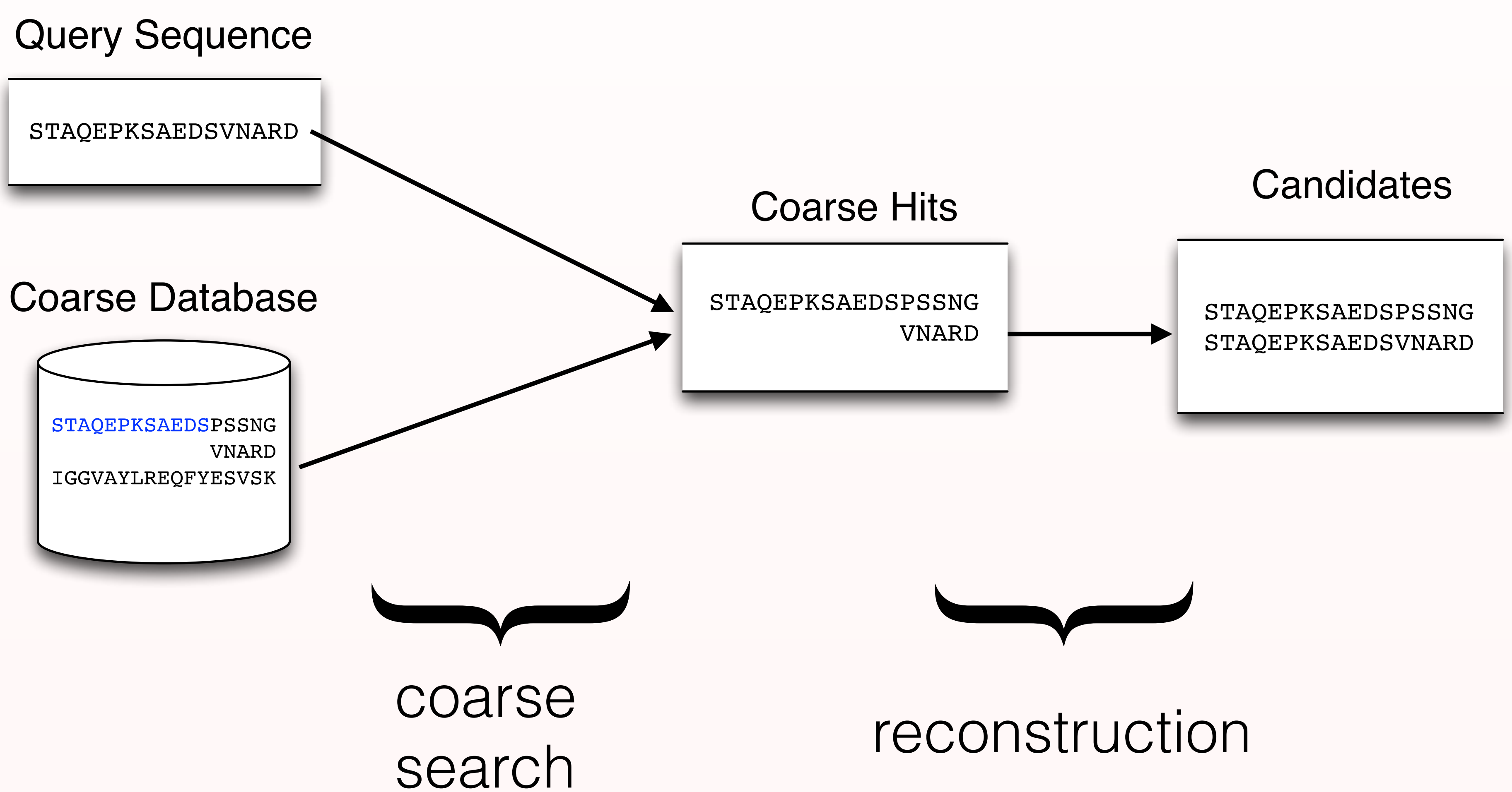
STAQEPKSAEDSVNARD

Coarse Database

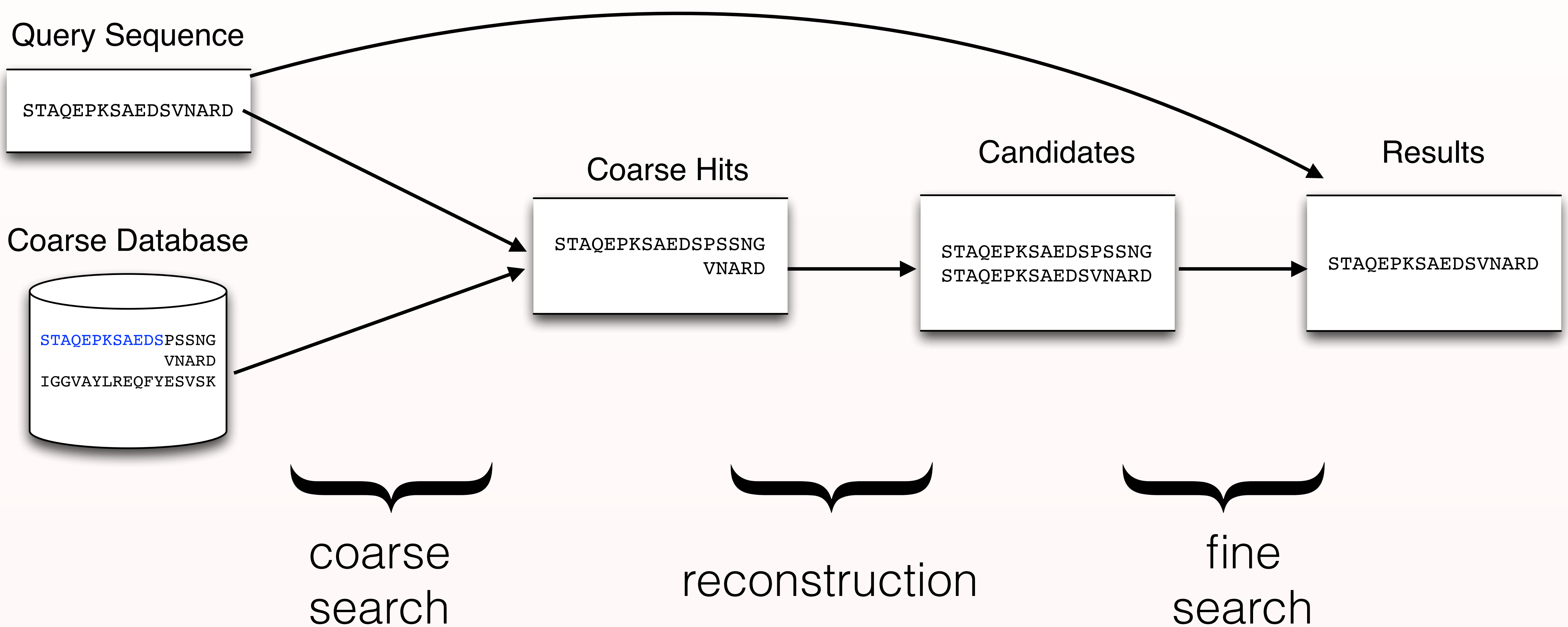
How does compressed search work?



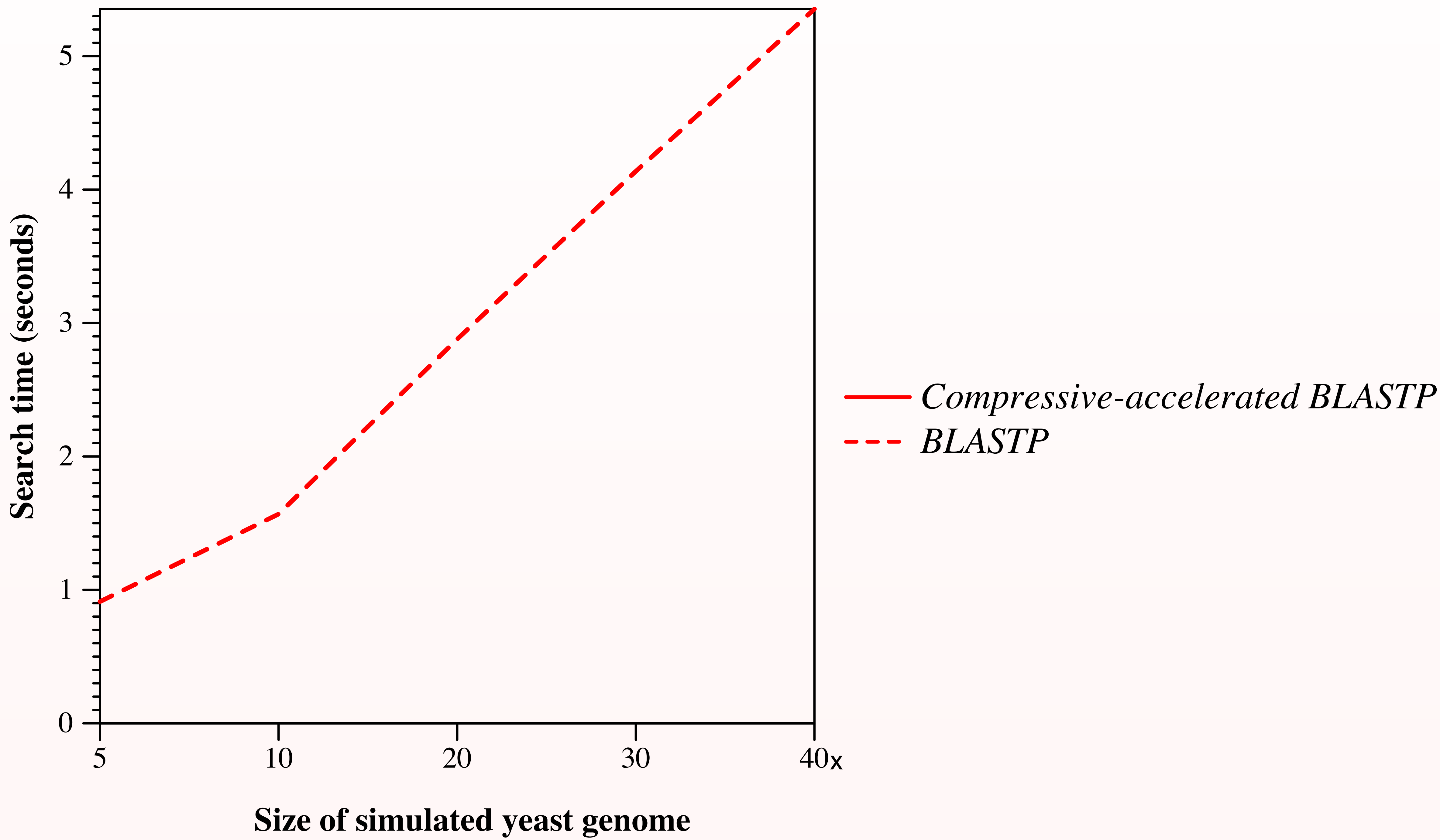
How does compressed search work?



How does compressed search work?

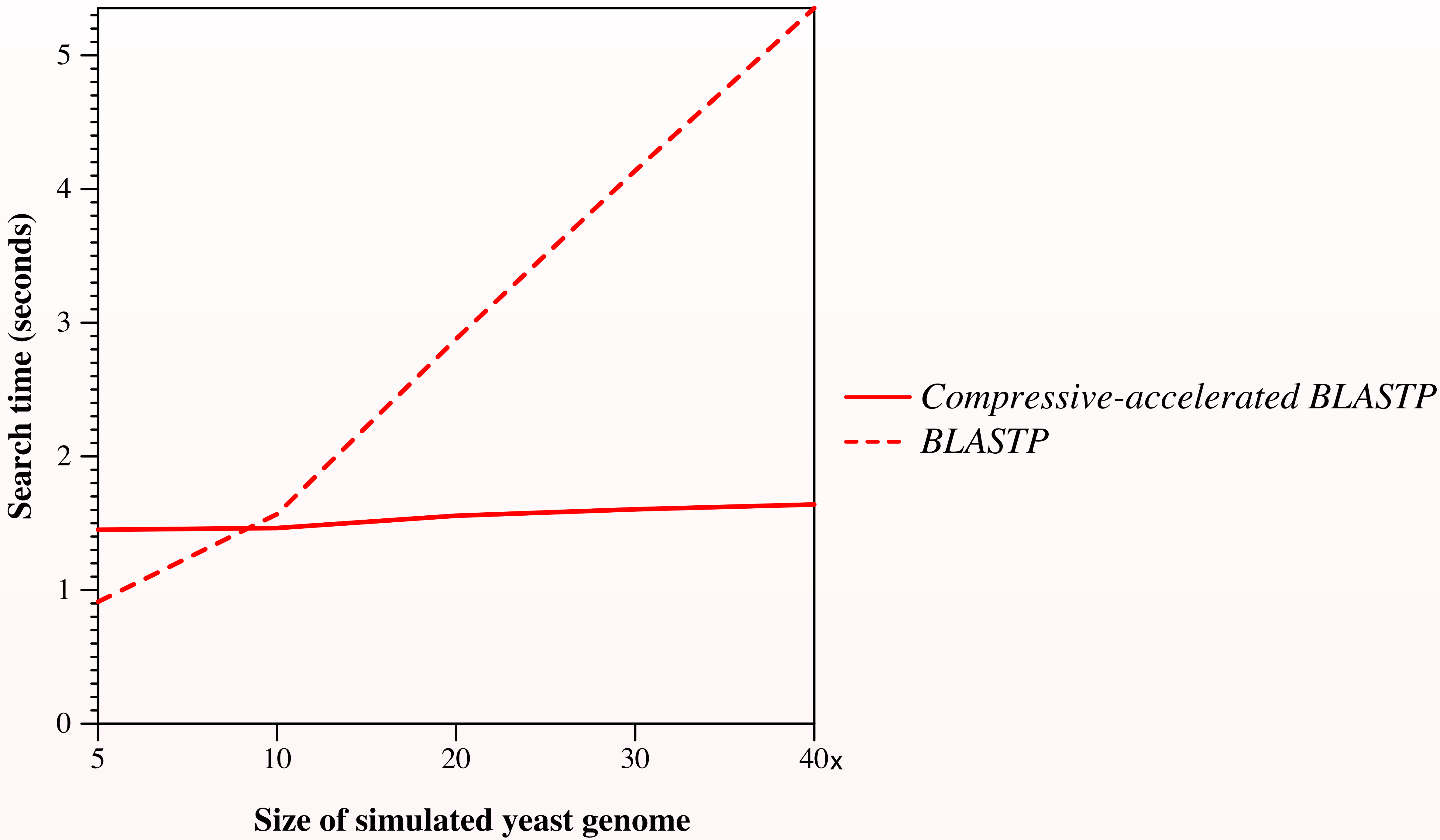


Simulated data growth



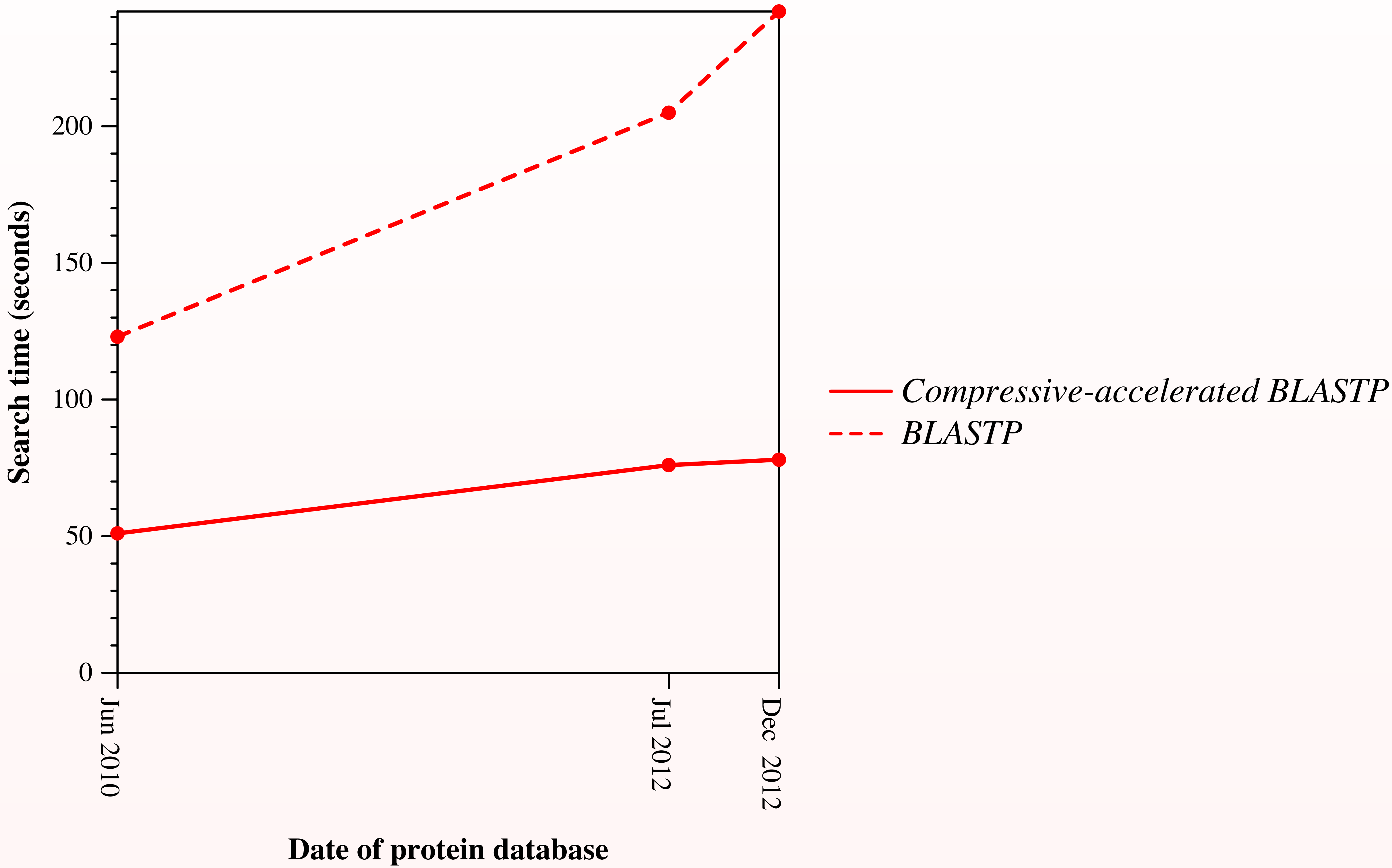
Daniels, Gallant, Peng, Baym, Cowen, Berger 2013

Simulated data growth

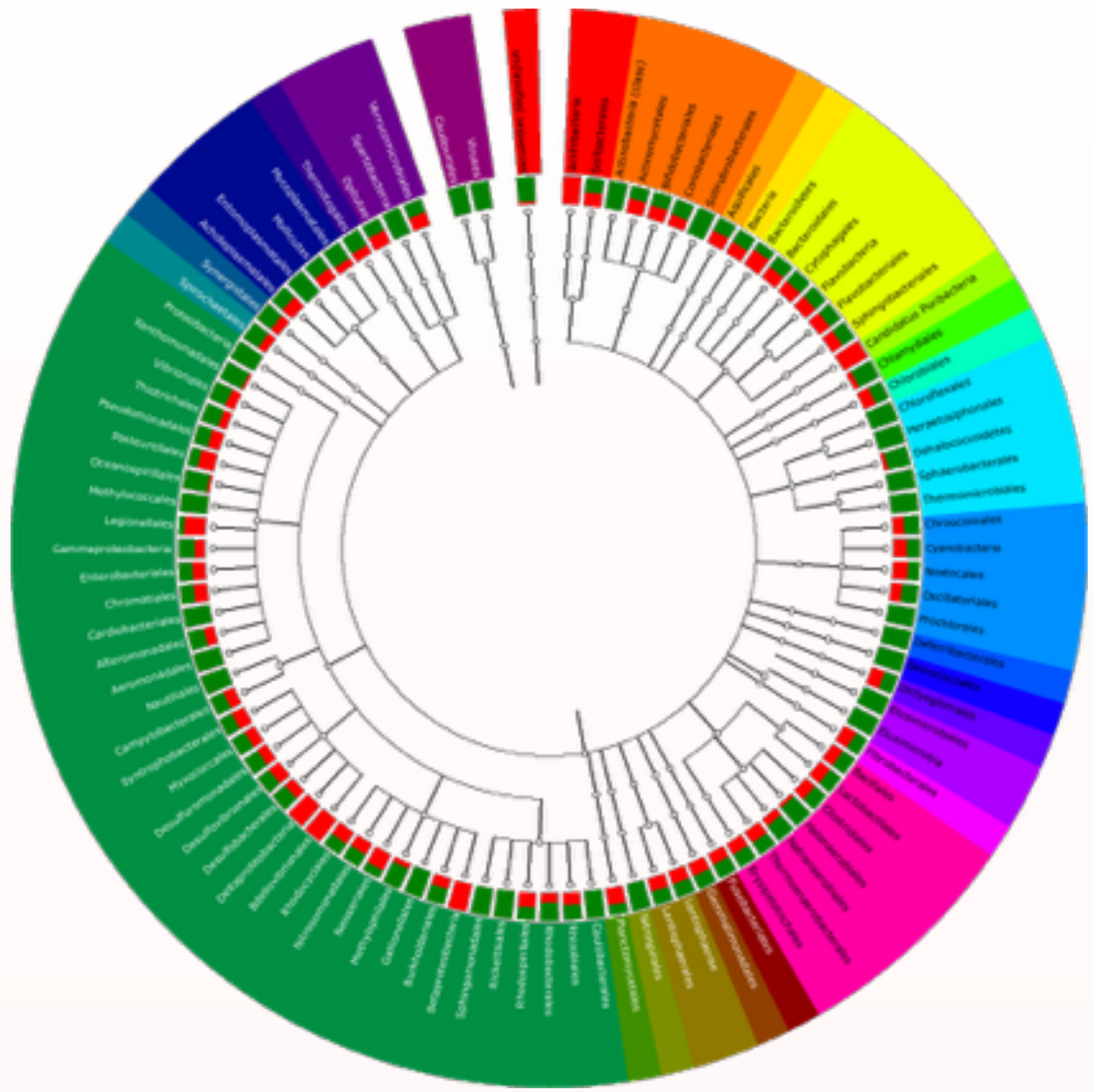


Daniels, Gallant, Peng, Baym, Cowen, Berger 2013

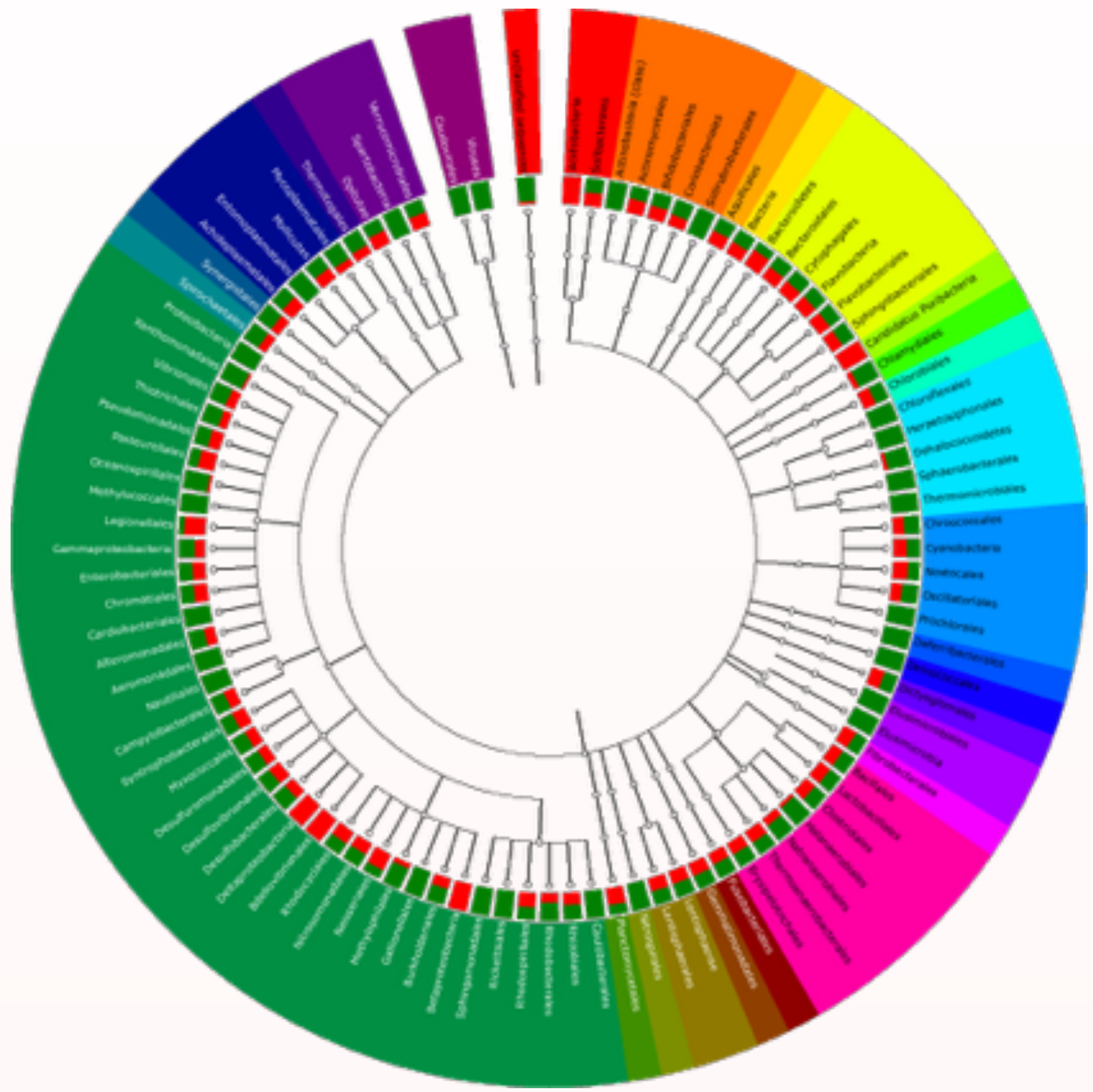
Real data growth on a protein database



Metagenomics



Metagenomics



Problem: Given reads from a microbiome, match each read with similar proteins.

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

Metagenomics

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

Metagenomics

Metagenomic Reads

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

...

>readk
TCGTCGAACTACATTACATTTACAG
>readk+1
TCGTCGAACTACATTACAAATACAG

...

>readm
GCTCGGACTATATATAGGCCTAGAA
...
```

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

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
...
```

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

Metagenomics

Metagenomic Reads

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

...

>readk
TCGTCGAACTACATTACATTTACAG
>readk+1
TCGTCGAACTACATTACAAATACAG

...

>readm
GCTCGGACTATATATAGGCCTAGAA
...
```

Translated ORFs

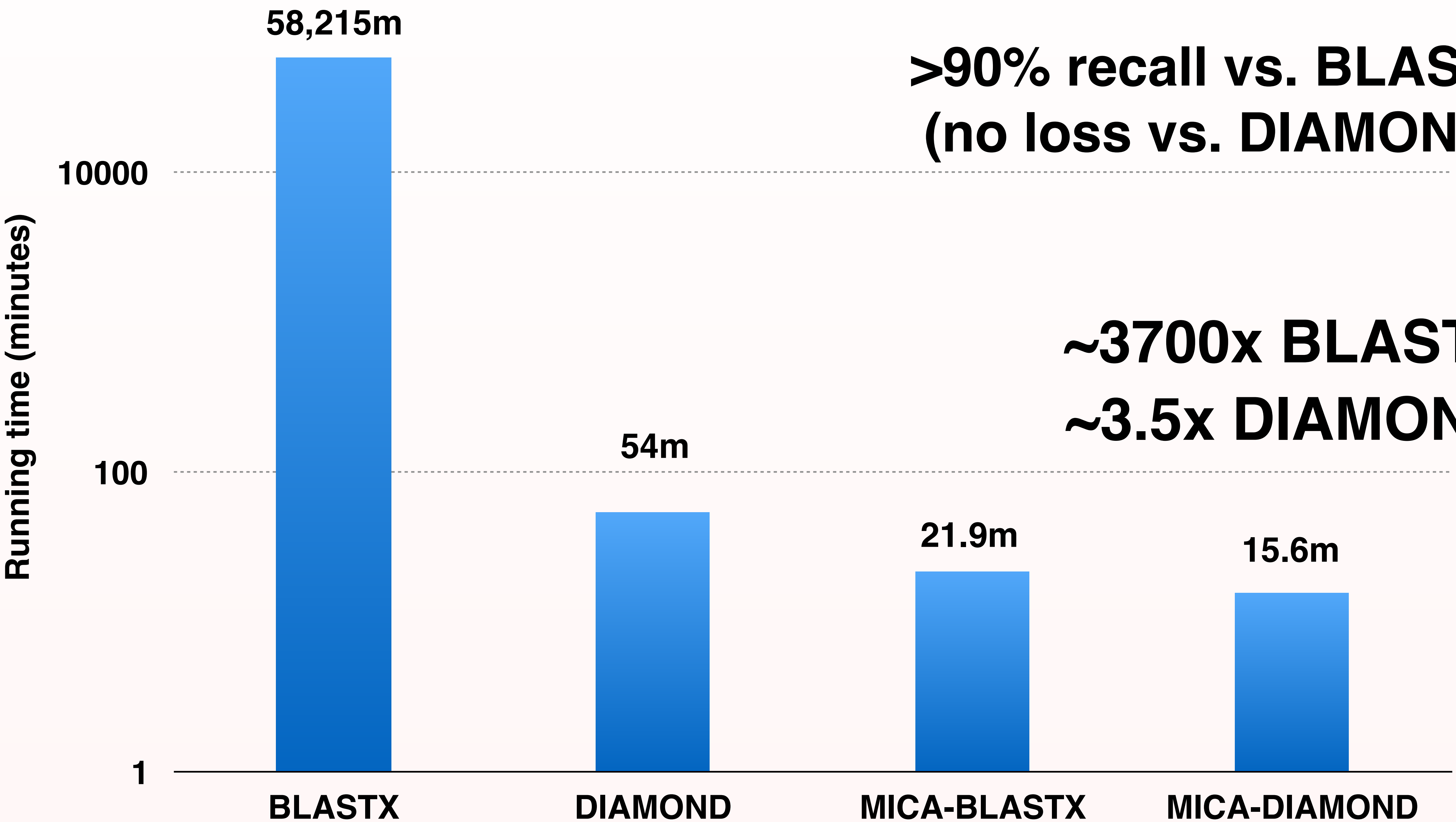
```
>read1-1
TWLSTR
>read1-2
RGYQLAN
>read1-3
VAINSLT
>read1-r1
LVSELIAT
>read1-r2
LAS
>read1-r3
RVDSH
>read2-1
SSNYITFT
>read2-2
RRTTLHLQ
>read2-3
VELHYIY
>read2-r1
CSST
>read2-r2
CKCNVVRR
>read2-r3
VNV
...
```

Protein Database (NR)

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

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

MICA [Yu, Daniels, et al. 2015]



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

~3700x BLASTX
~3.5x DIAMOND

American gut microbiome project NGS reads, searching NCBI NR database

Compression = clustering

Compression = clustering

Cluster similar entities

Compression = clustering

Cluster similar entities

or sub-entities

Compression = clustering

Cluster similar entities

or sub-entities

Only store the common parts once

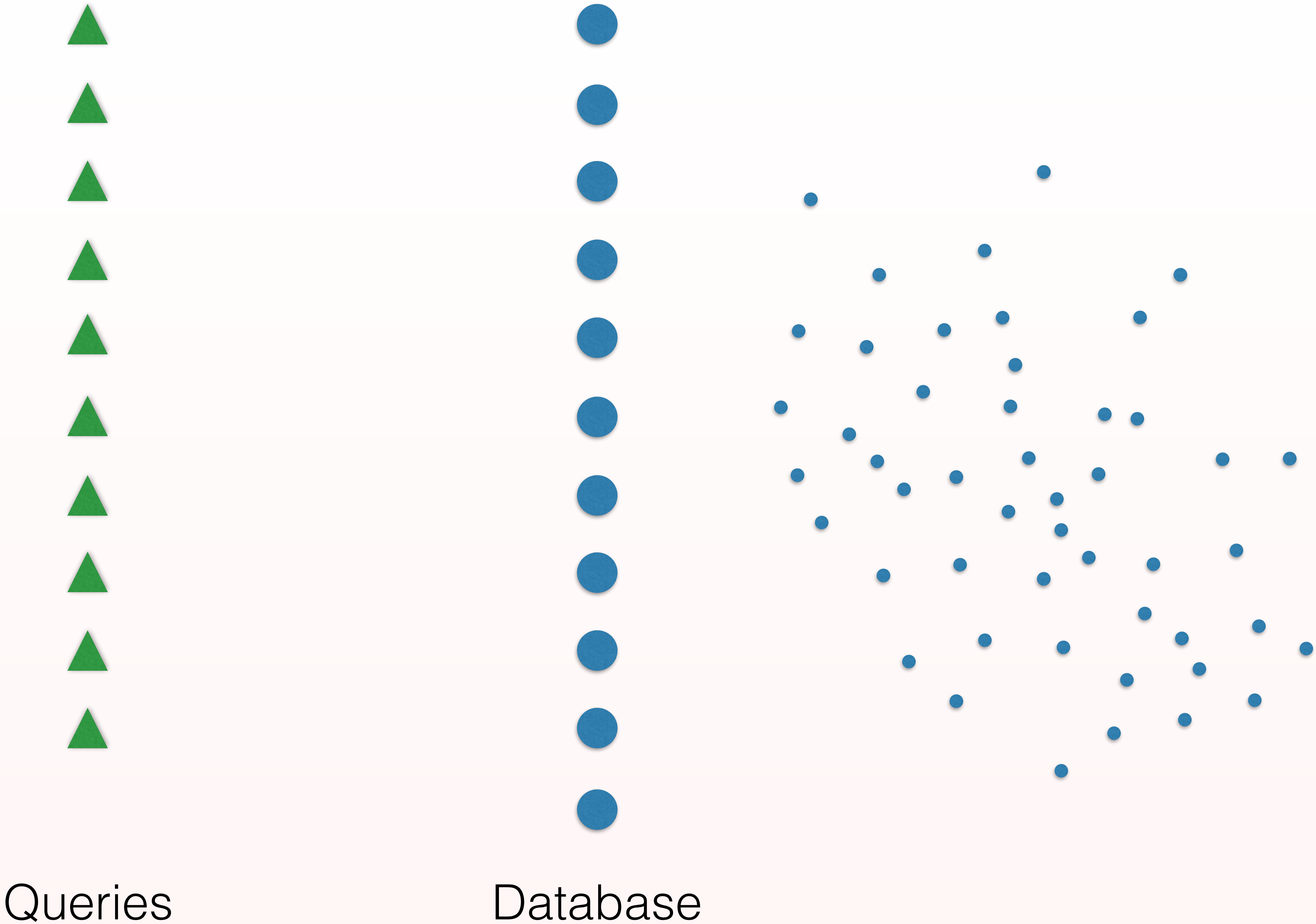
Compression = clustering

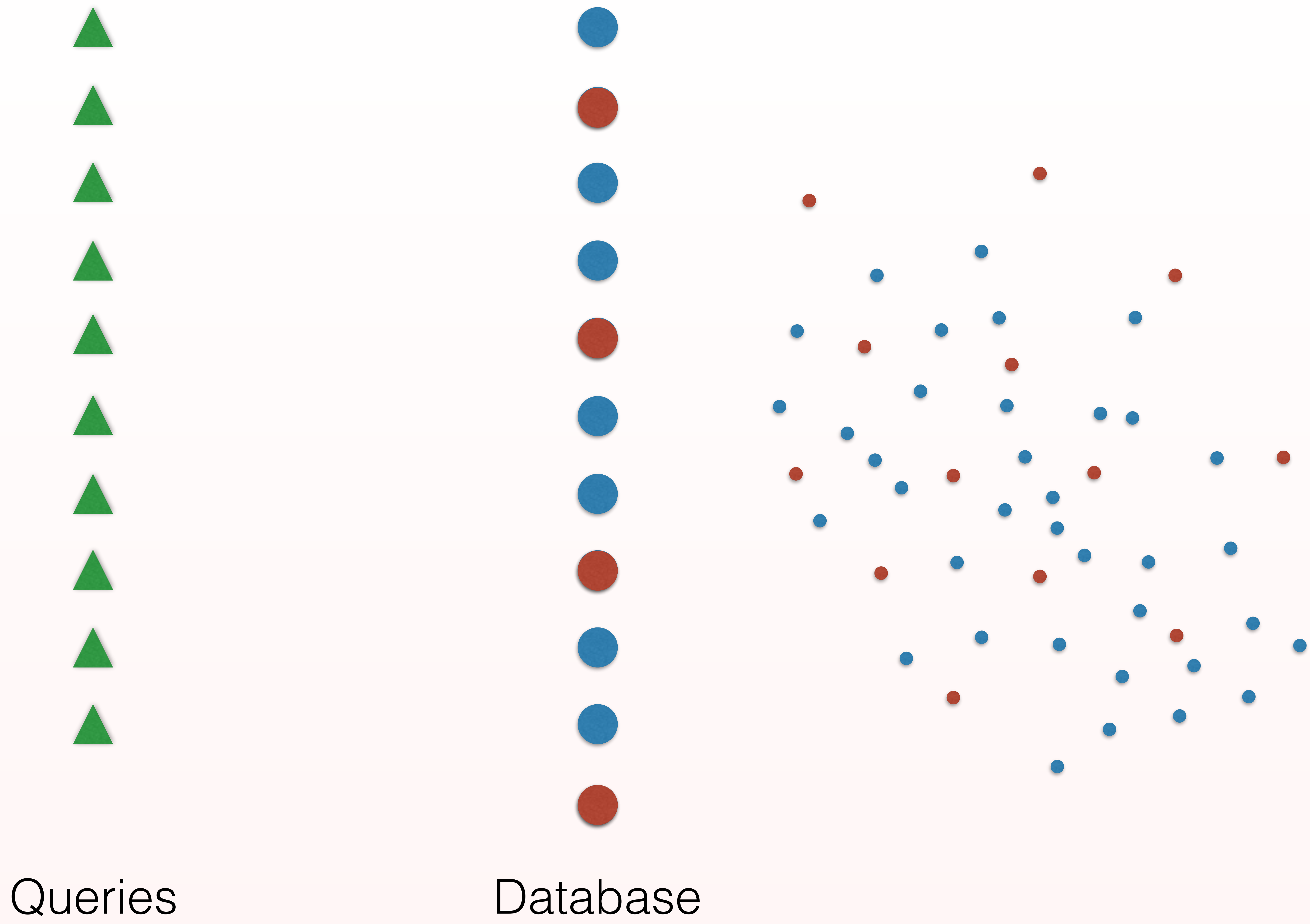
Cluster similar entities

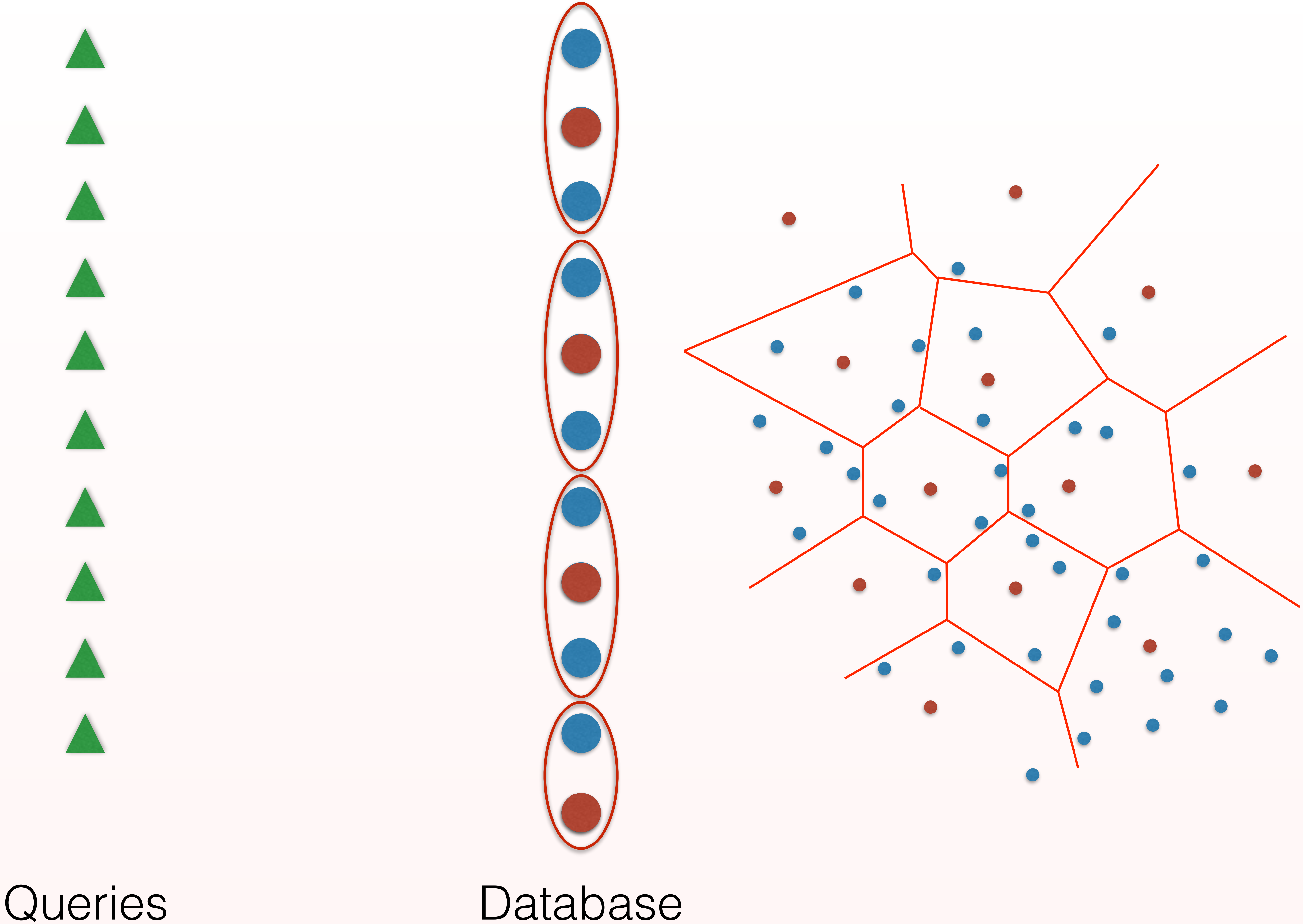
or sub-entities

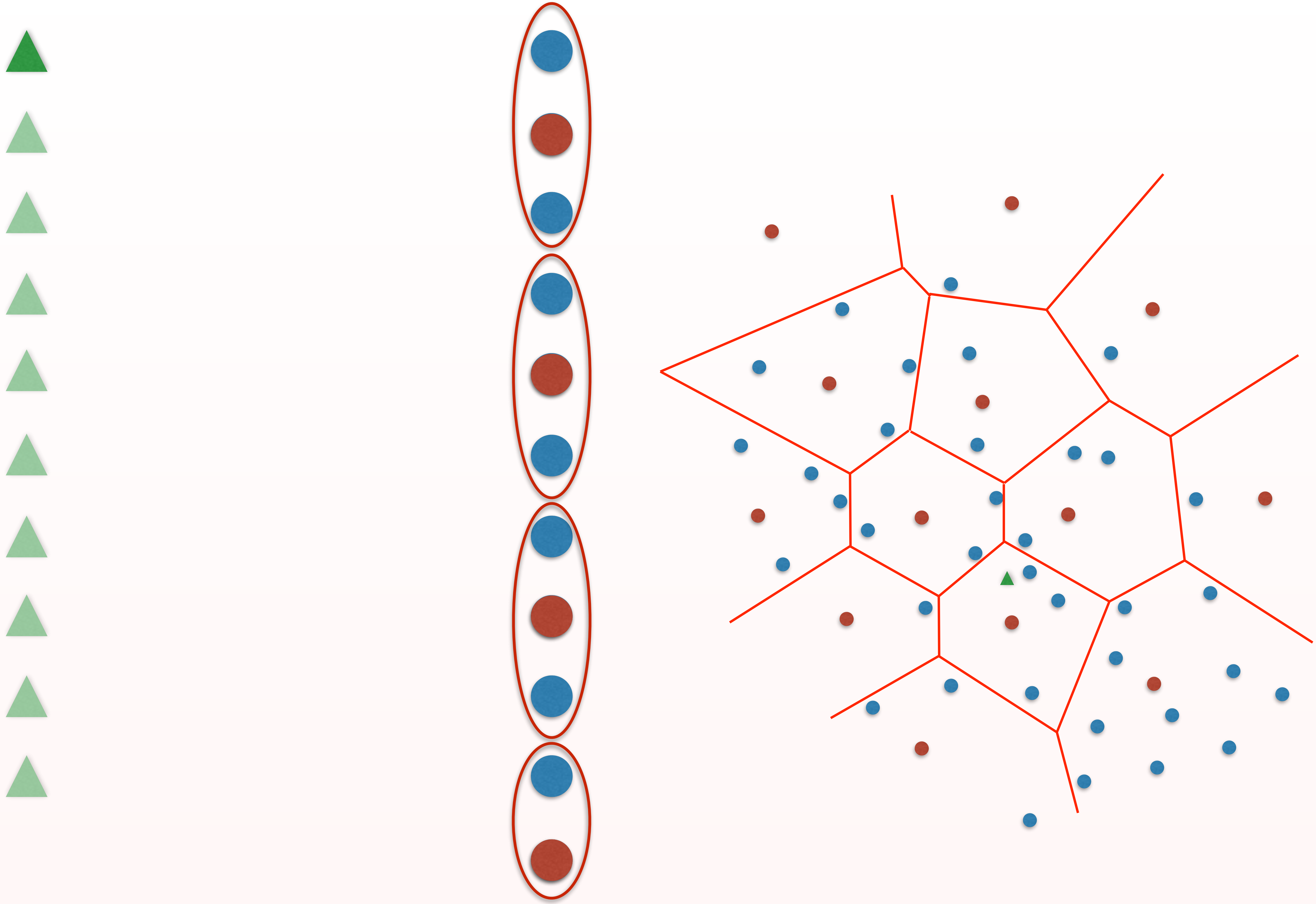
Only store the common parts once

Only analyze the common parts once


Queries
Database

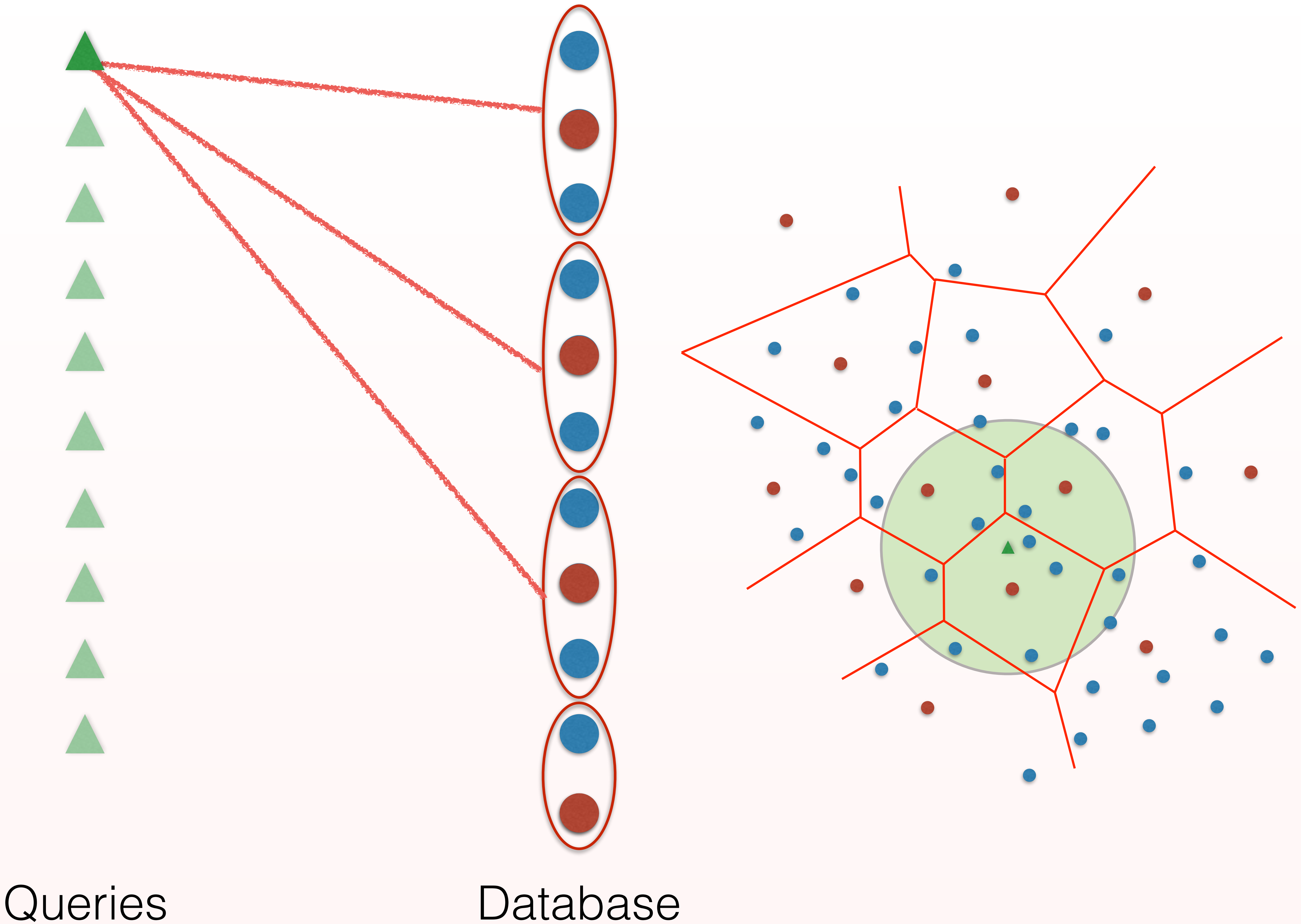

Queries
Database

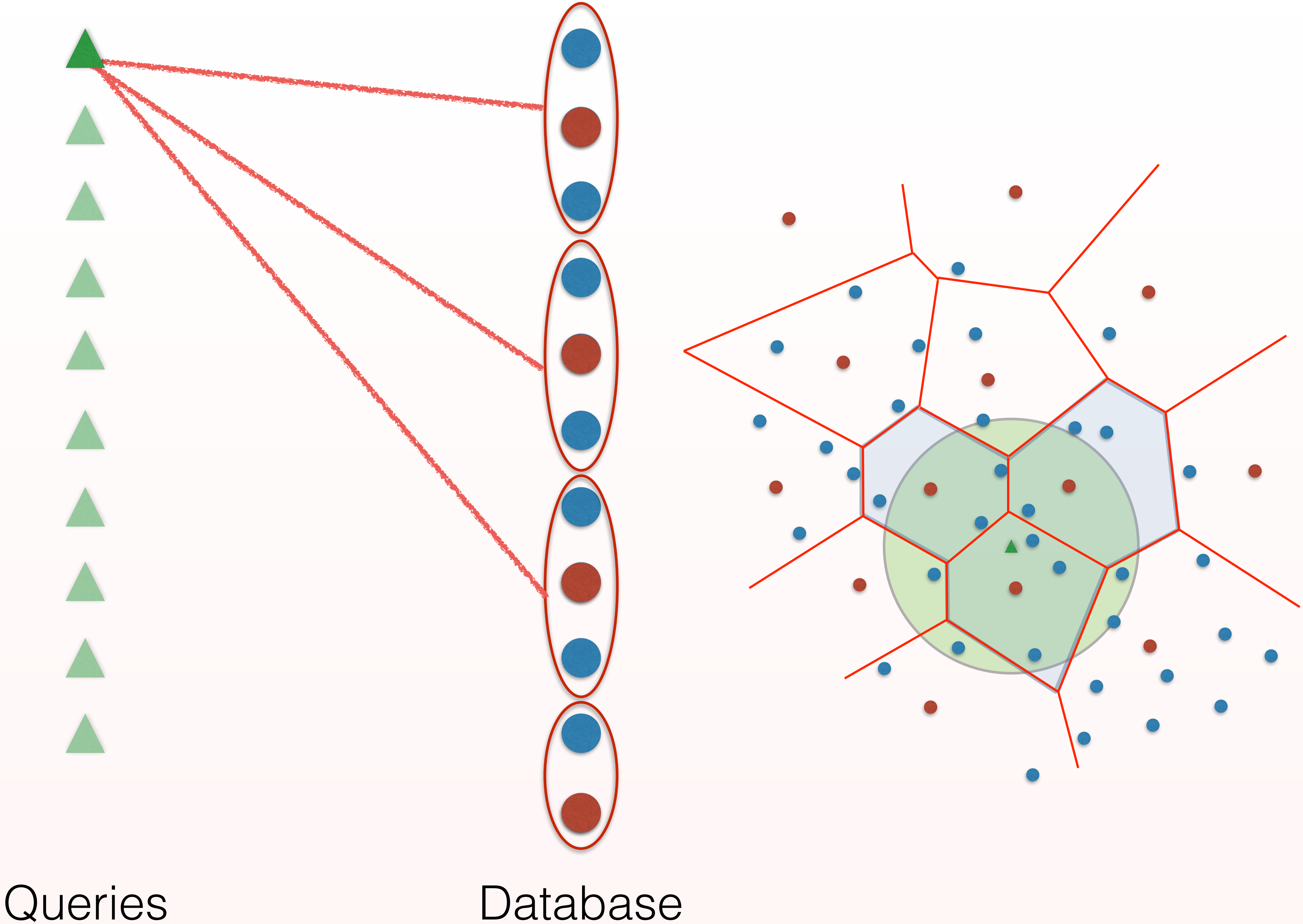

Queries
Database



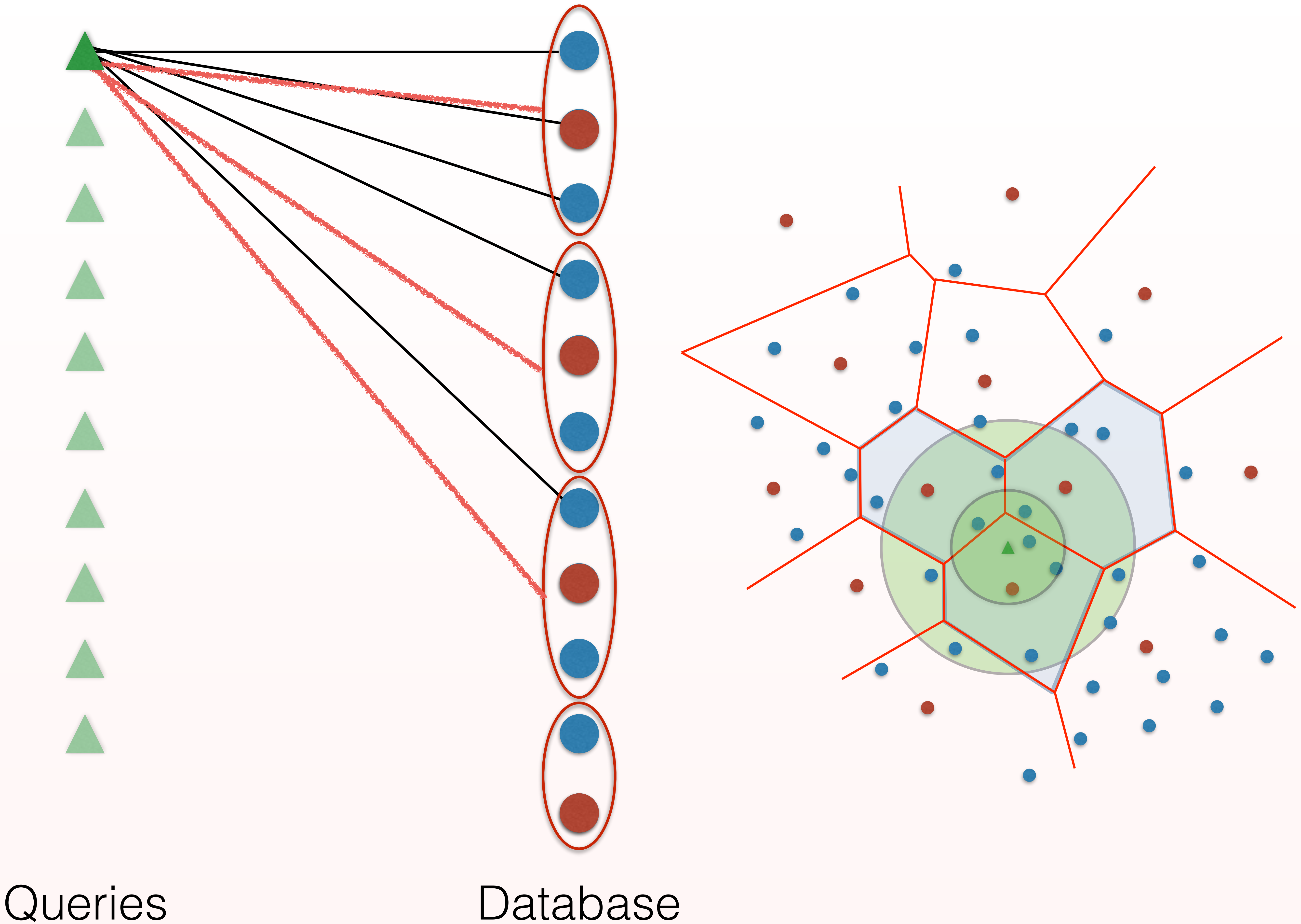
Queries

Database

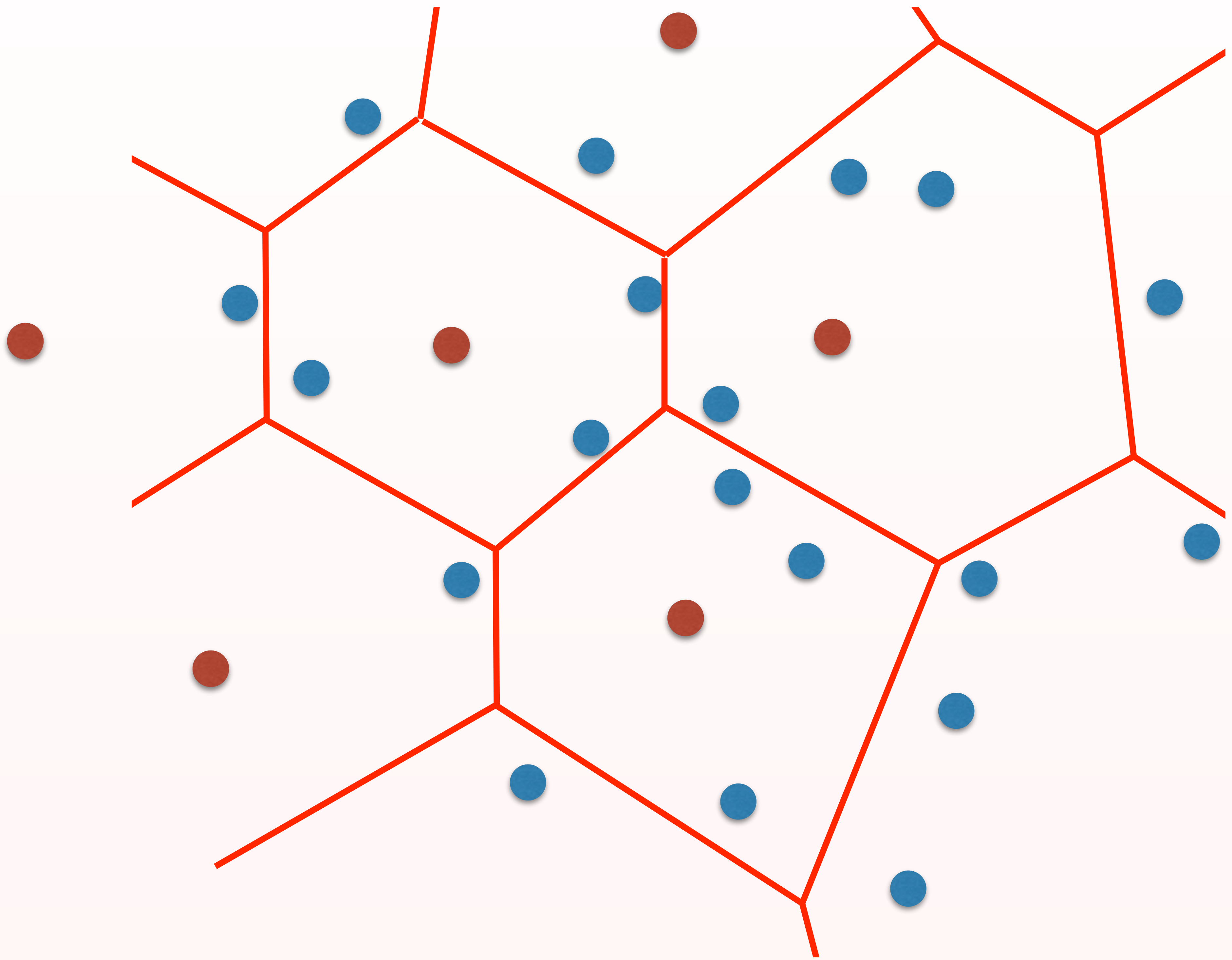

Queries
Database



Queries
Database

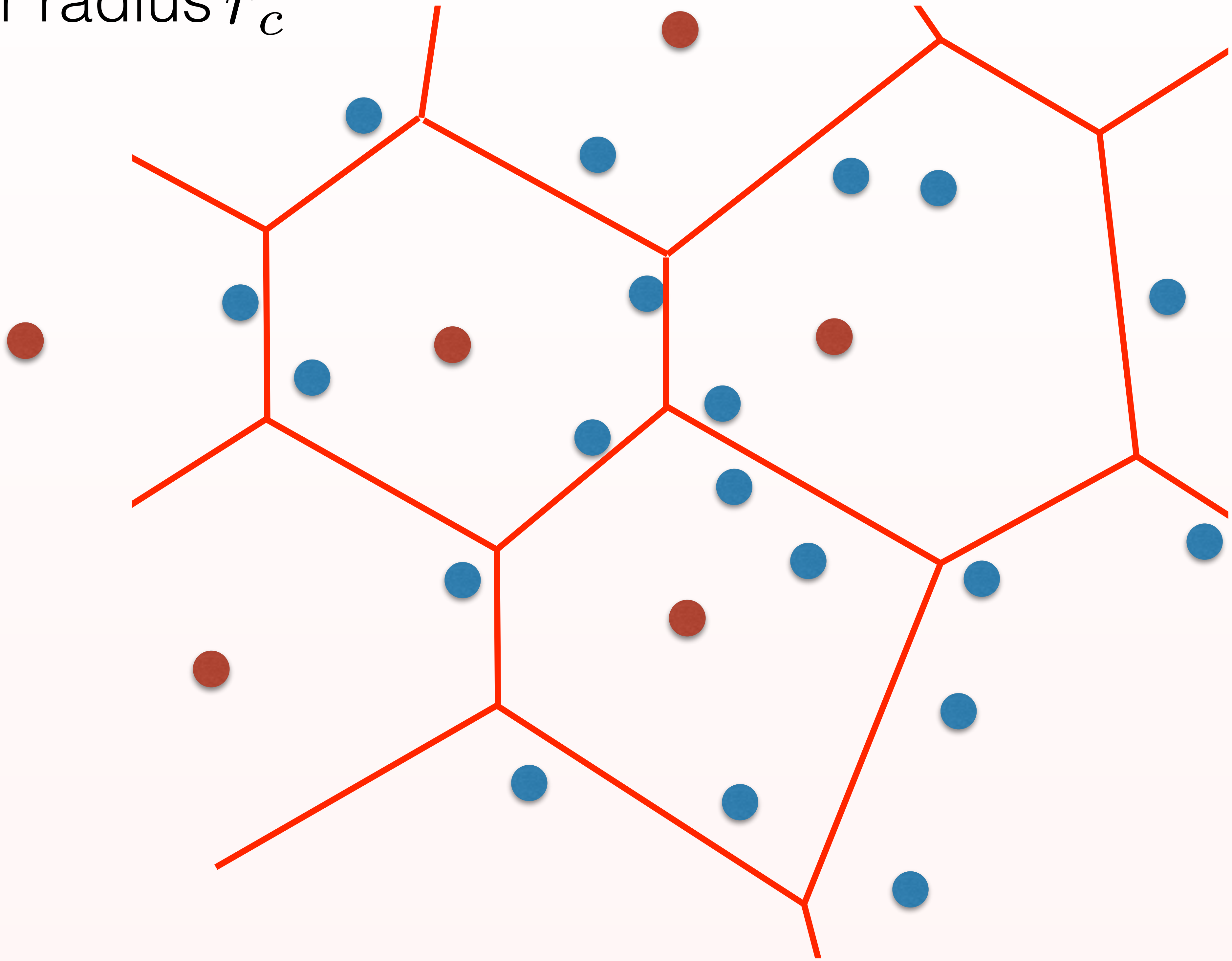

Queries
Database

Triangle inequality allows this



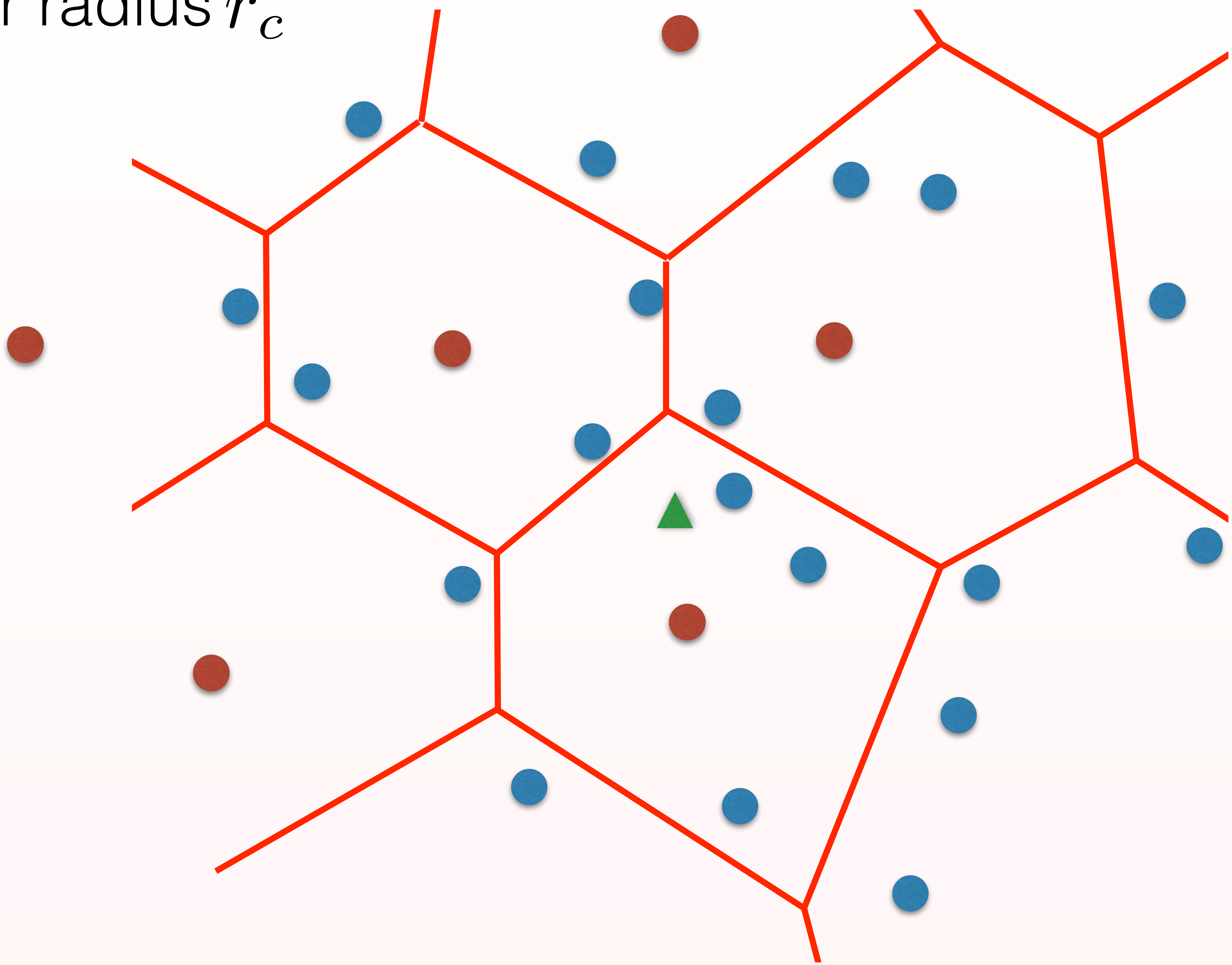
Triangle inequality allows this

cluster radius r_c



Triangle inequality allows this

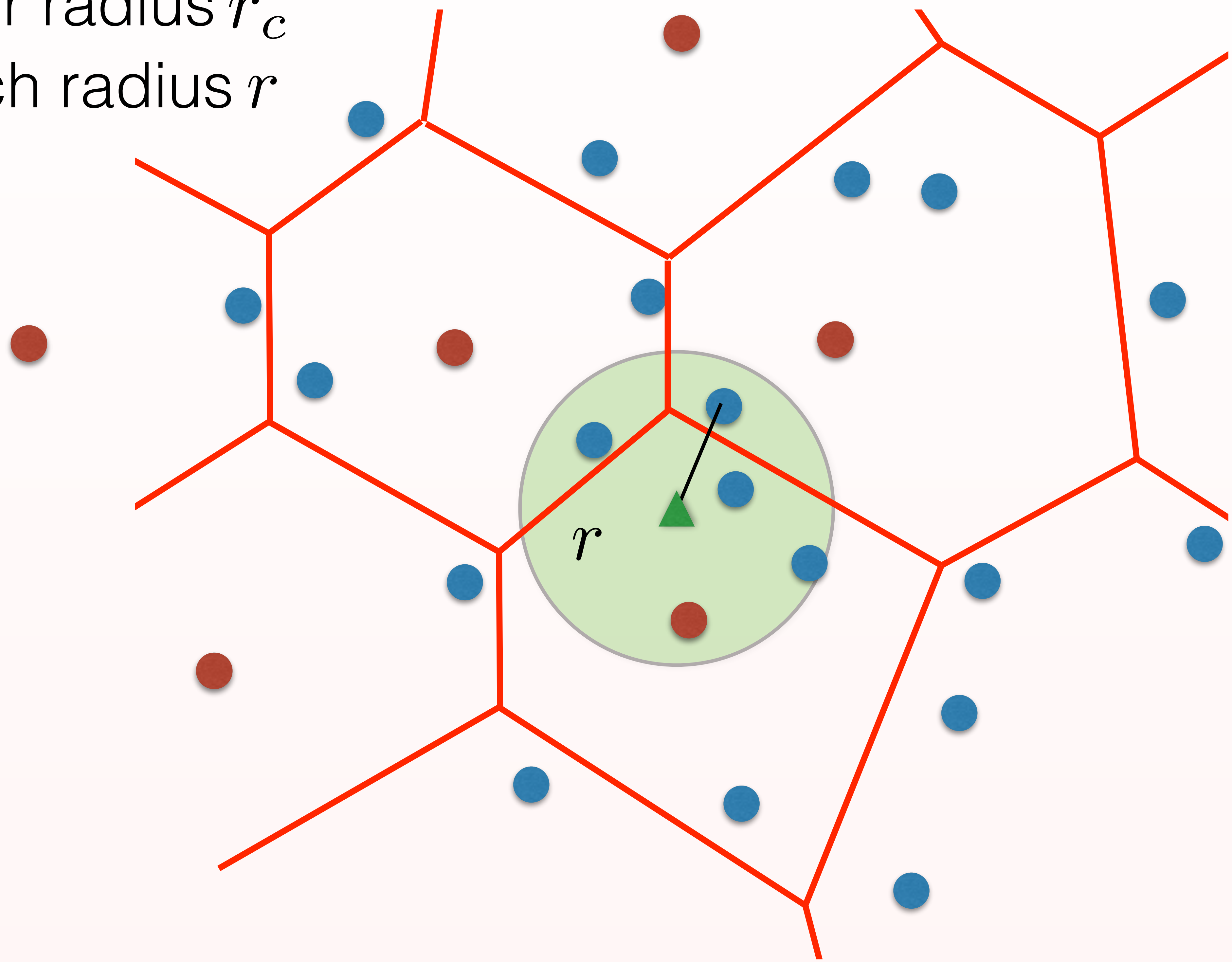
cluster radius r_c



Triangle inequality allows this

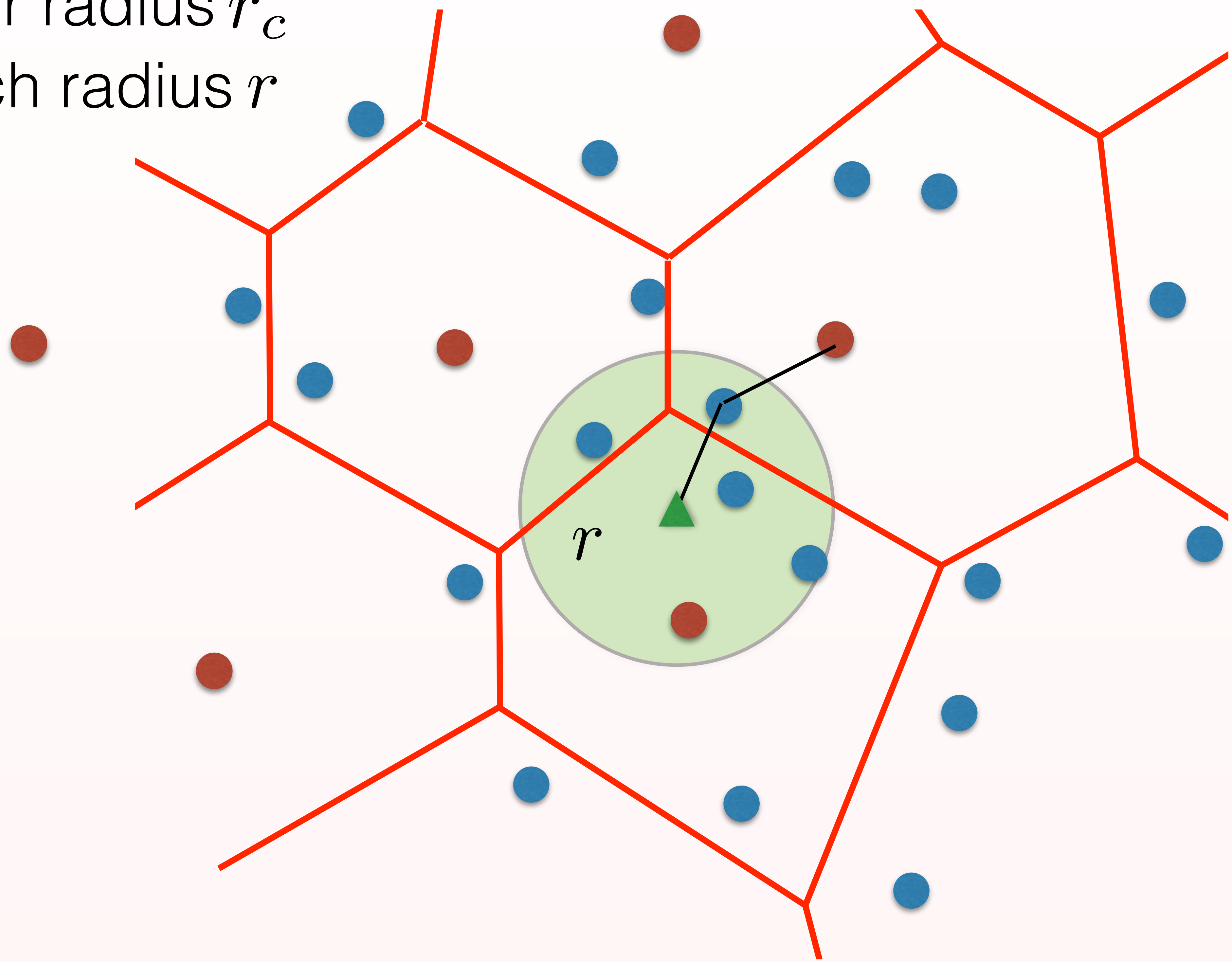
cluster radius r_c

search radius r



Triangle inequality allows this

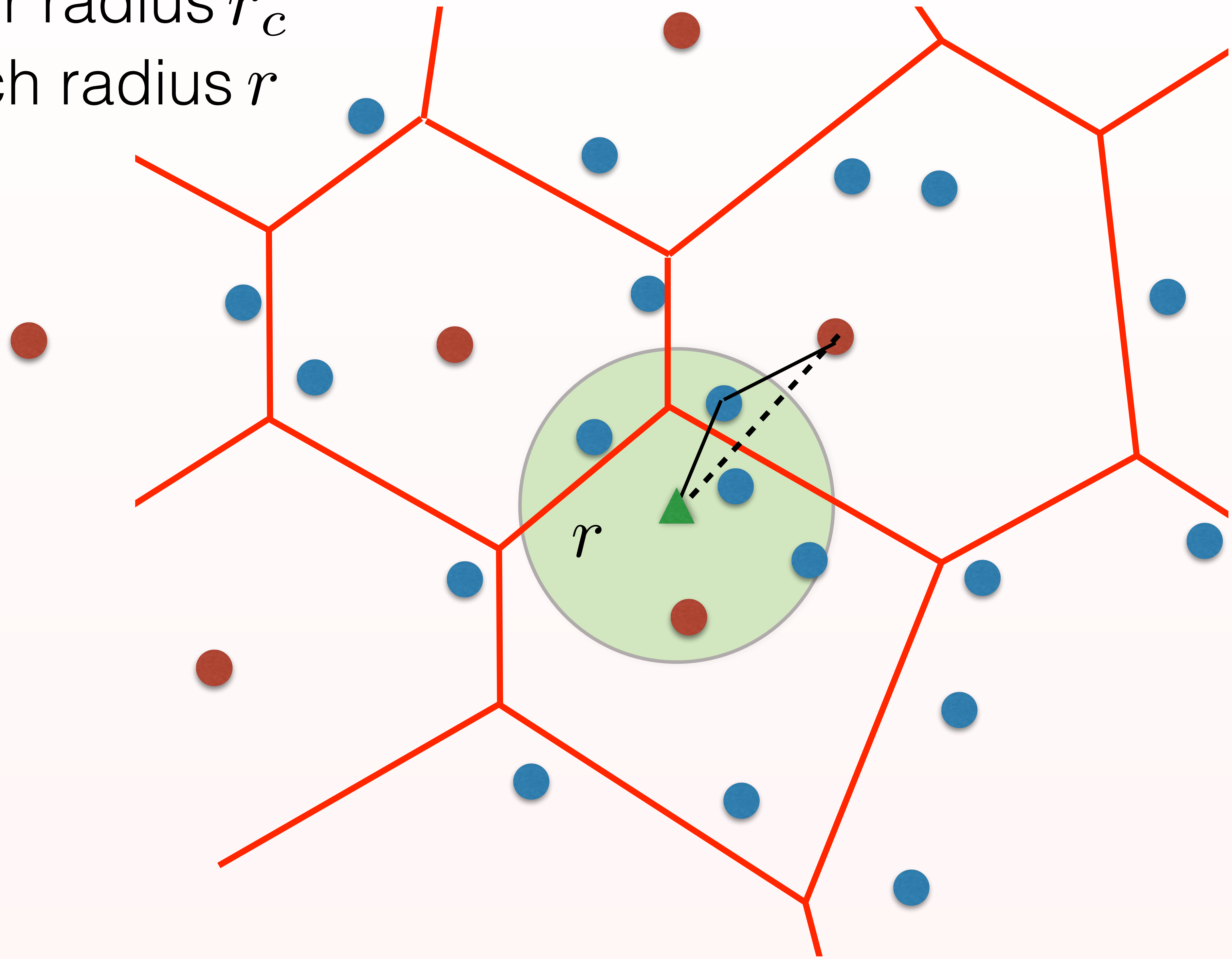
cluster radius r_c
search radius r



Triangle inequality allows this

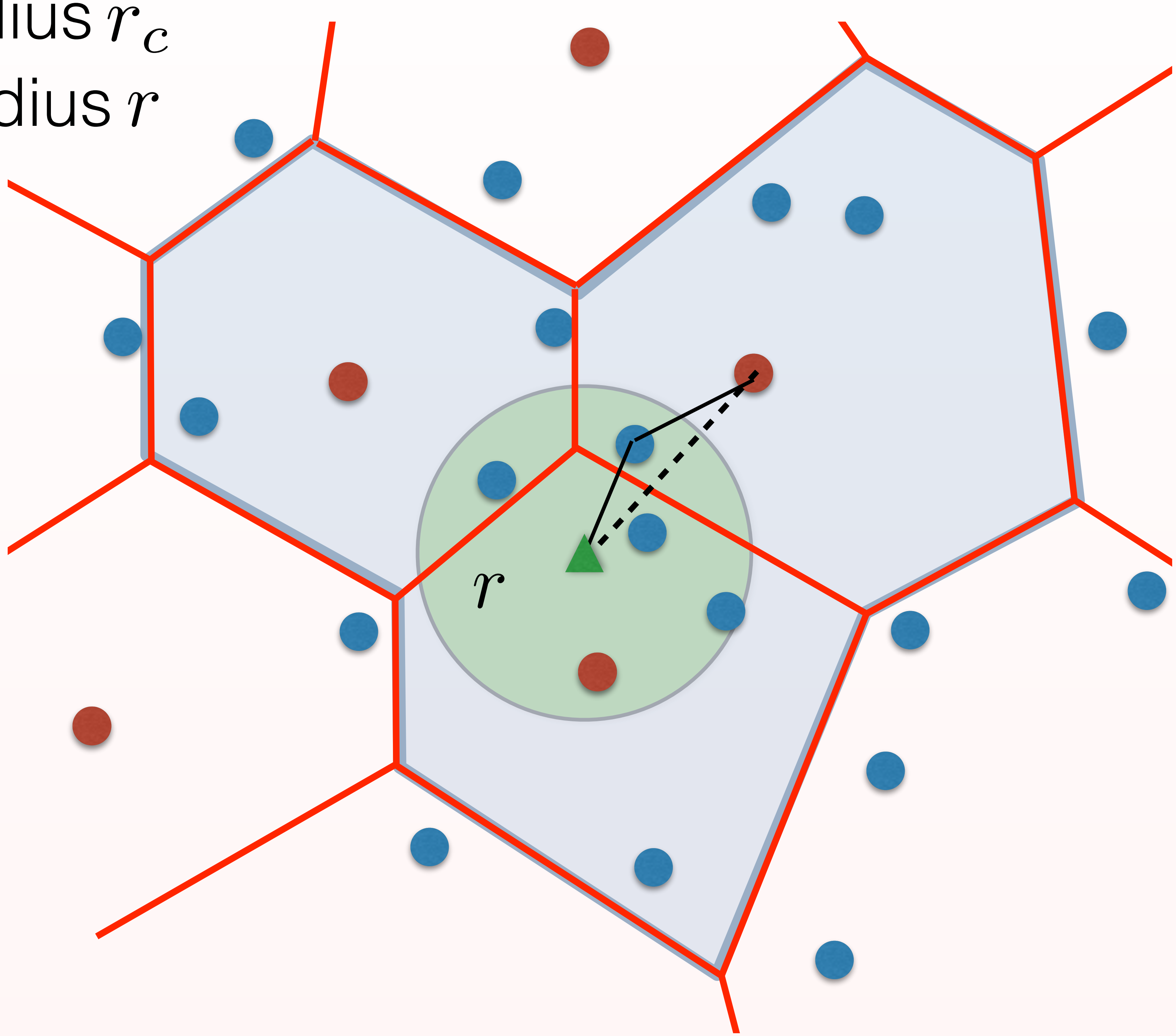
cluster radius r_c

search radius r

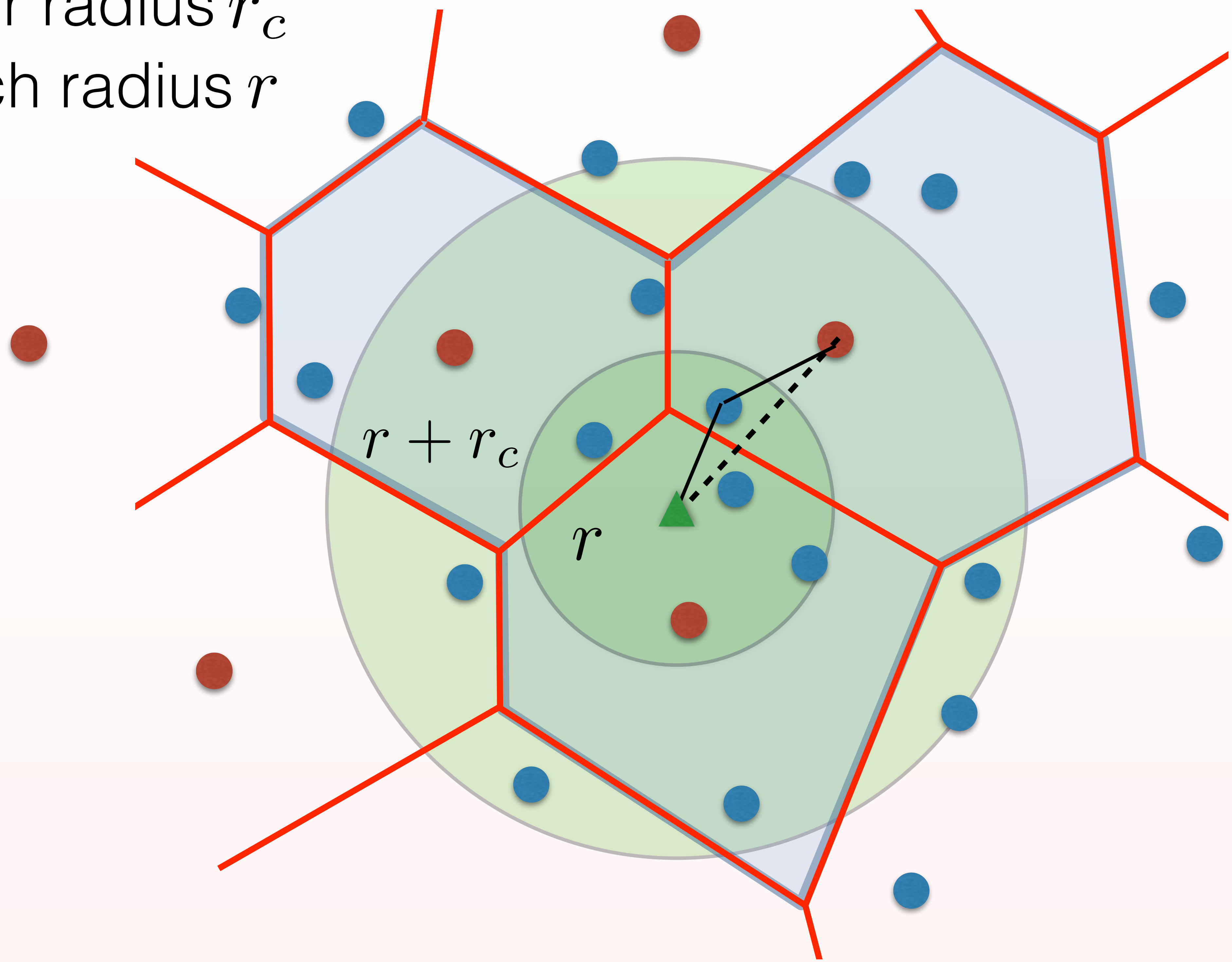


Triangle inequality allows this

cluster radius r_c
search radius r

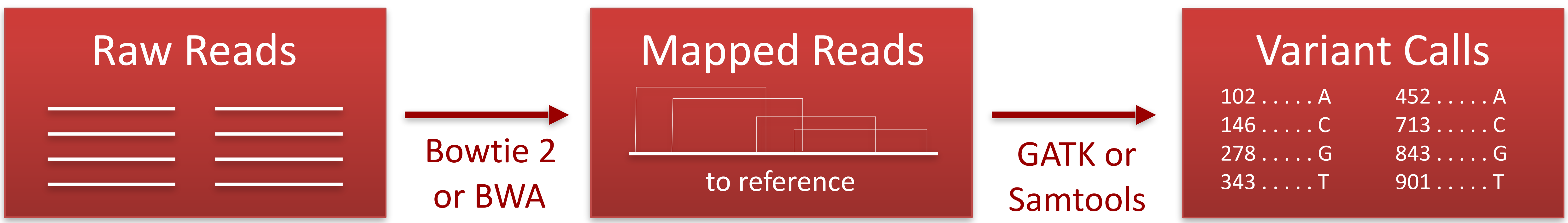


Triangle inequality allows this



Large-scale NGS Analysis

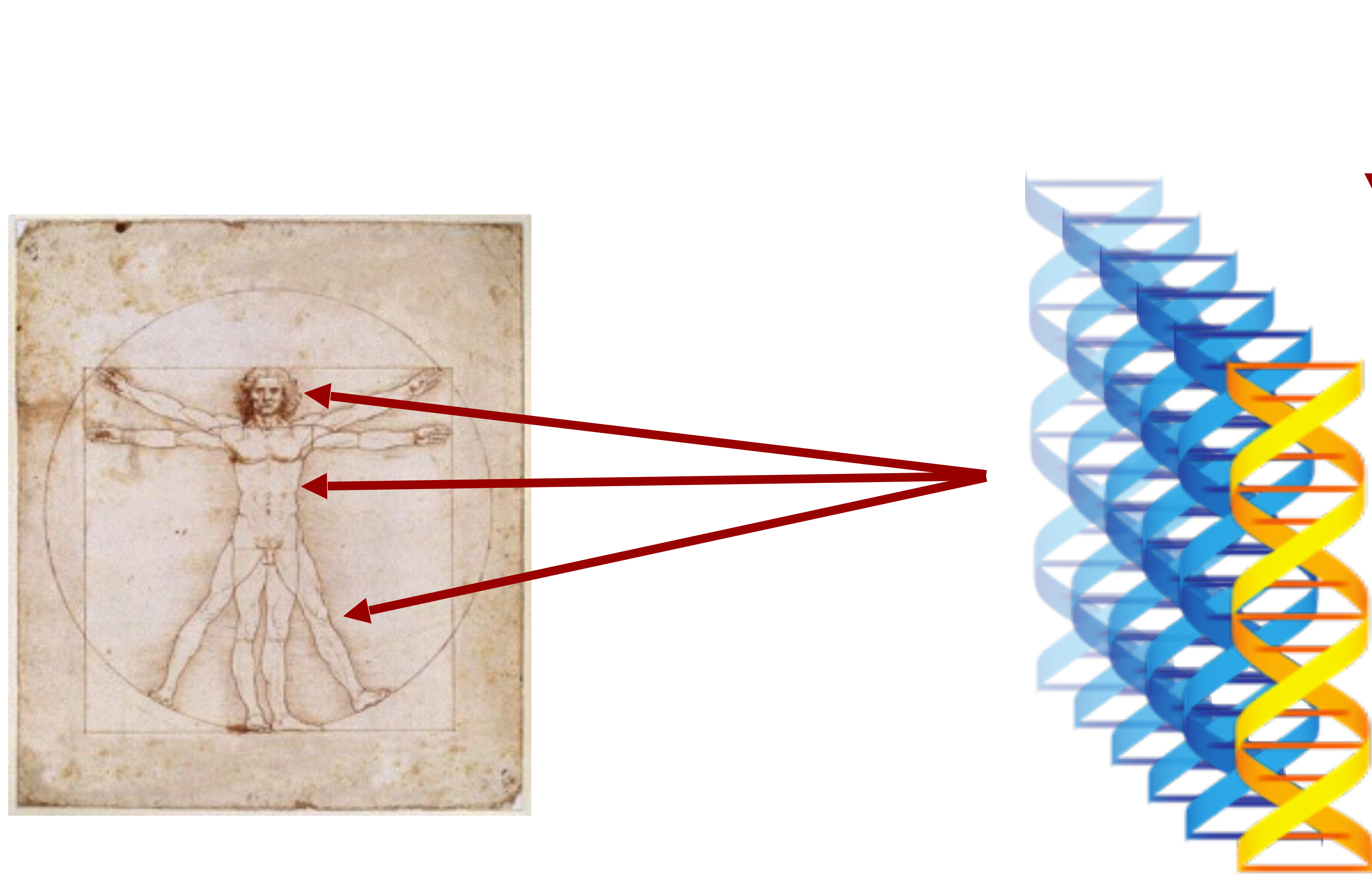
Large-scale NGS Analysis



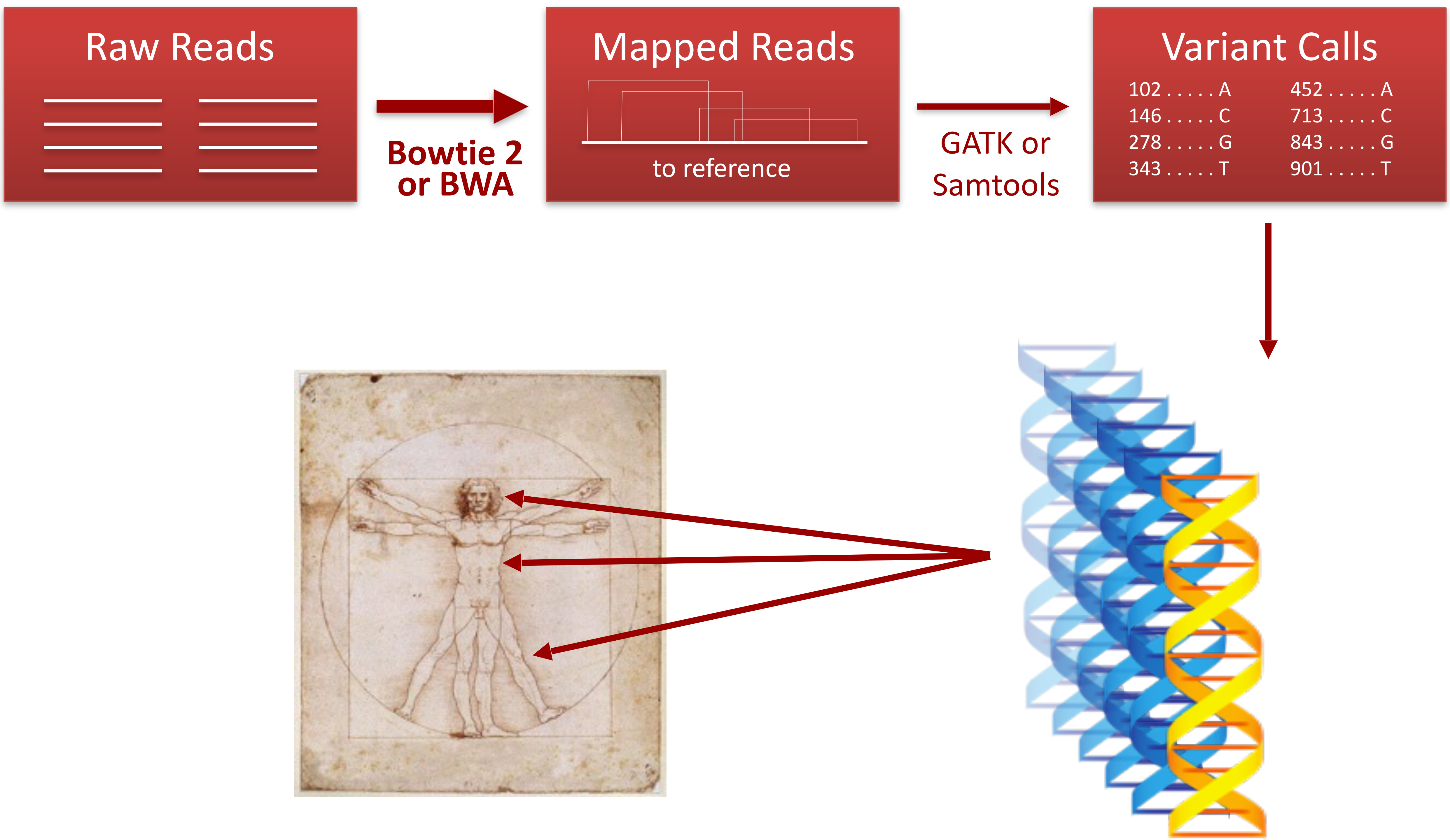
Large-scale NGS Analysis

Raw Reads → Bowtie 2 or BWA → **Mapped Reads** (to reference) → GATK or Samtools → **Variant Calls**

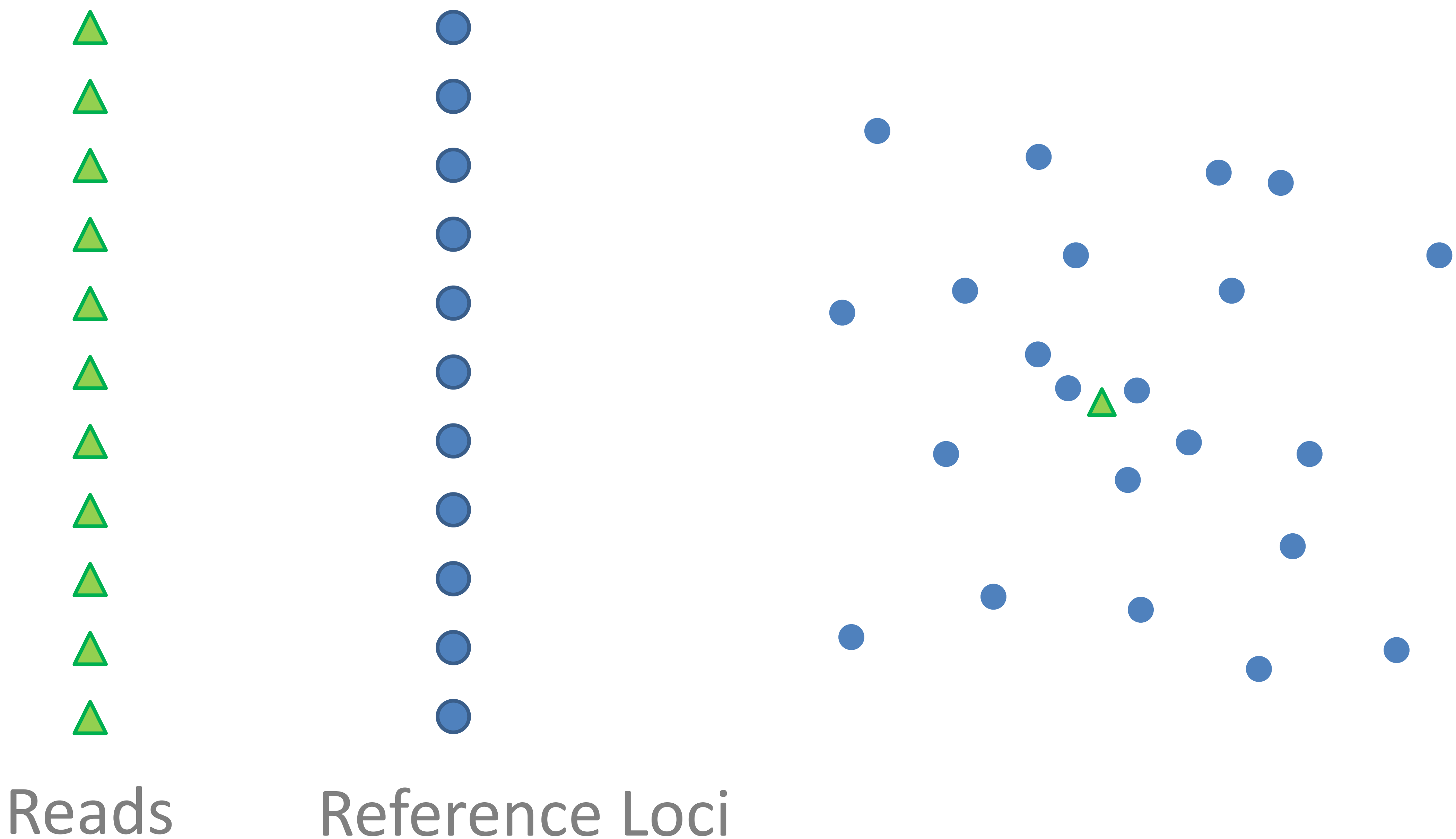
102 . . . . . A	452 . . . . . A
146 C	713 C
278 . . . . . G	843 . . . . . G
343 . . . . . T	901 . . . . . T



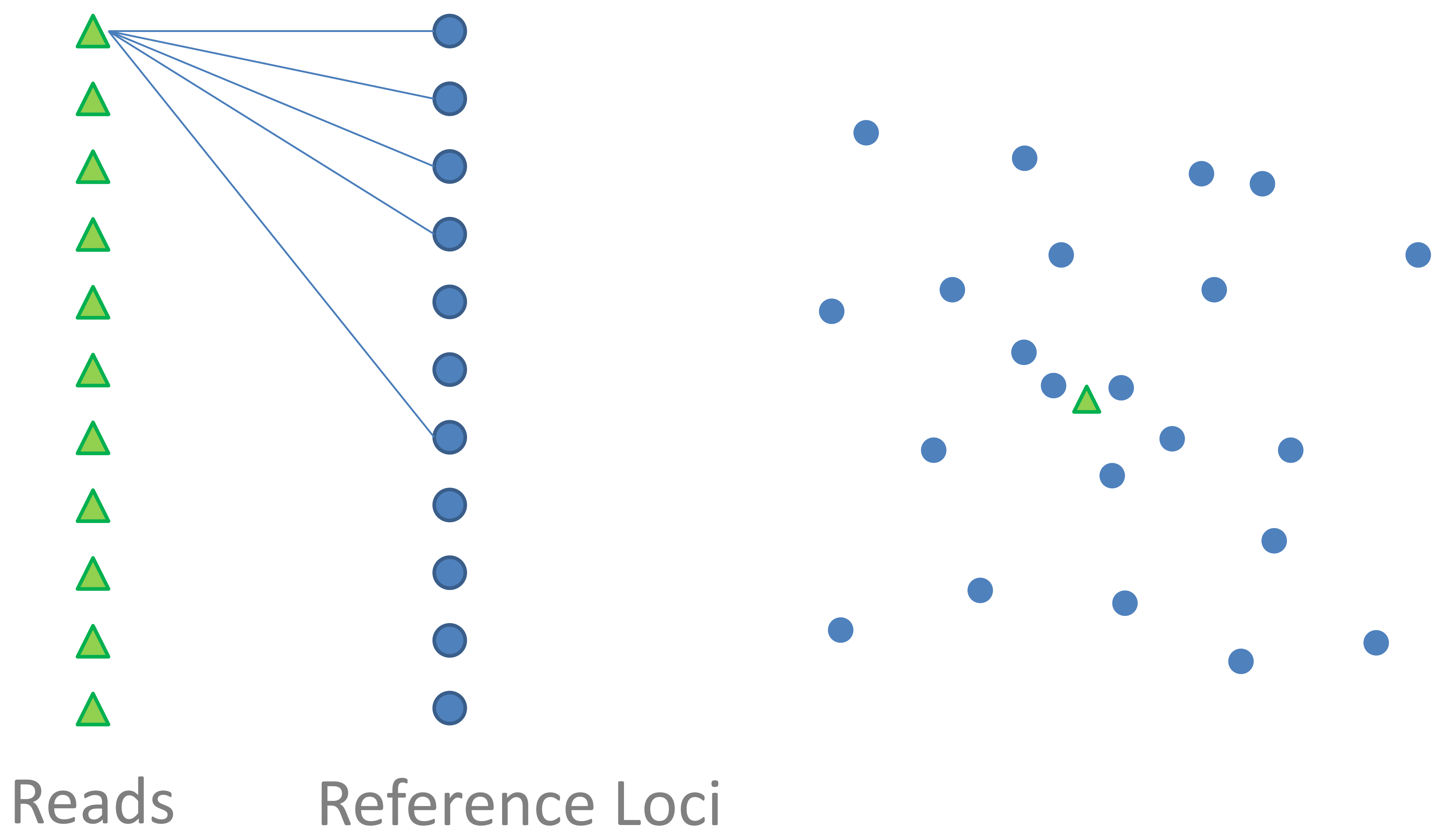
Large-scale NGS Analysis



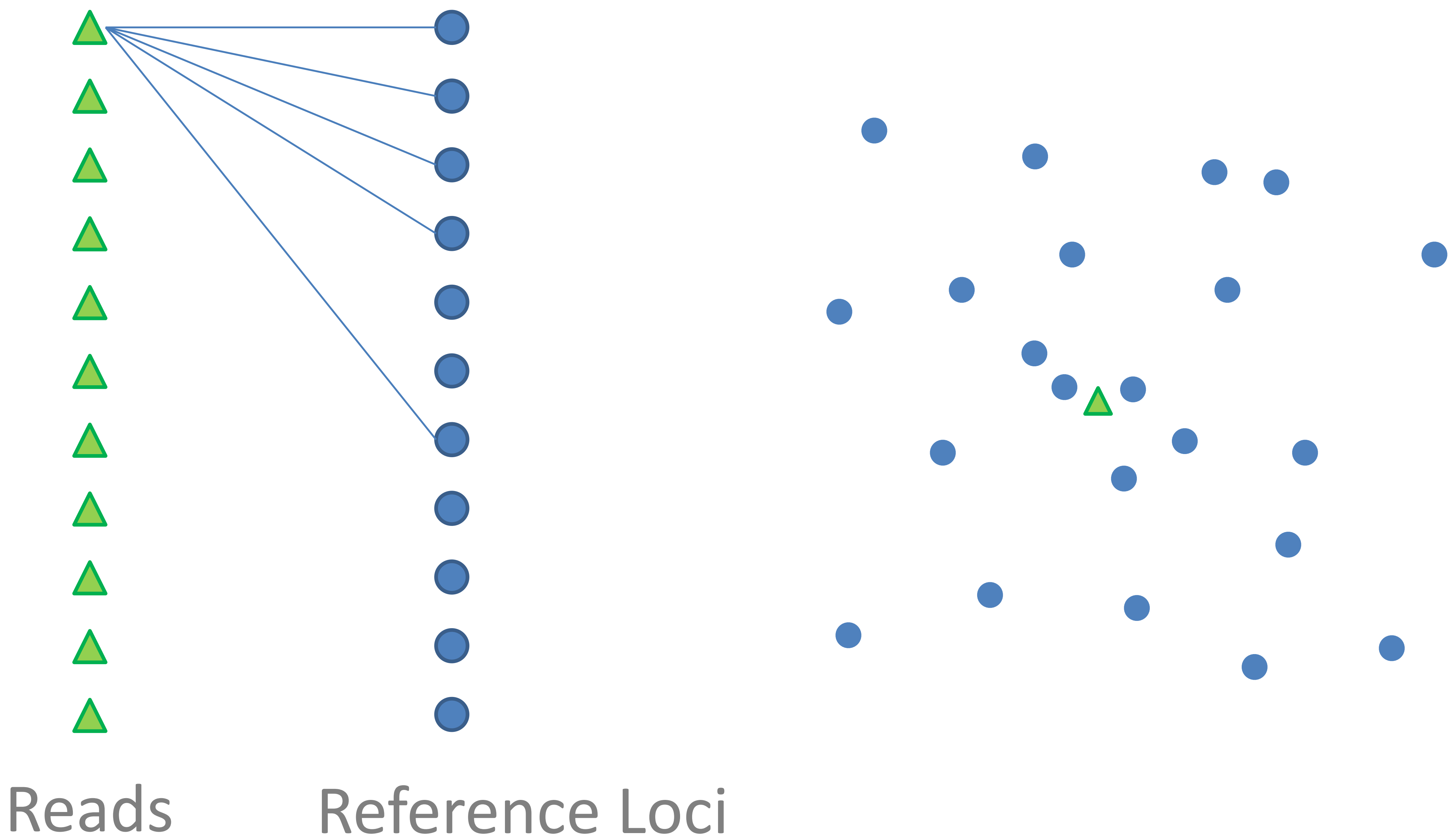
NGS-Mapping in Personal Genomics Era



NGS-Mapping in Personal Genomics Era

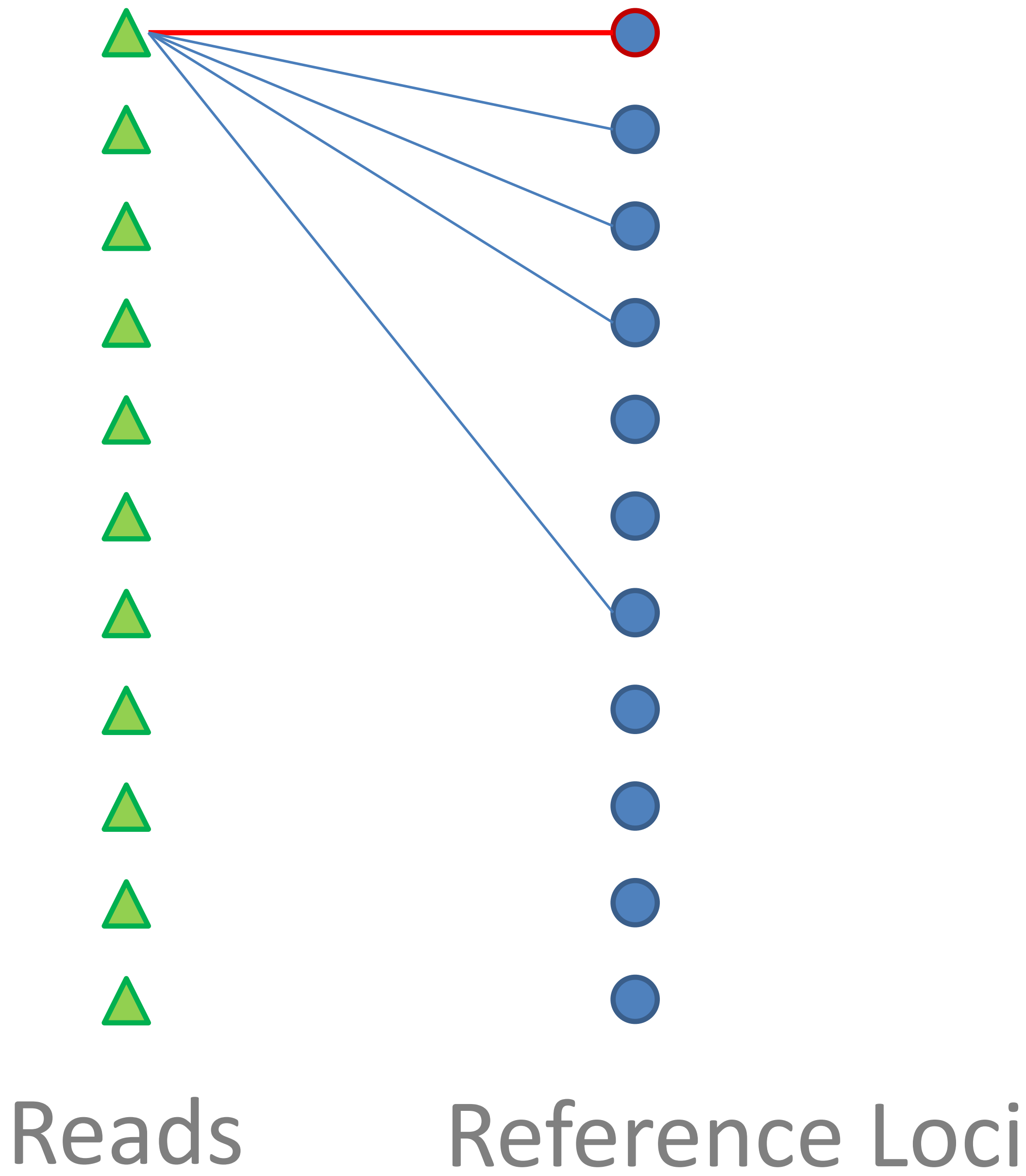


NGS-Mapping in Personal Genomics Era

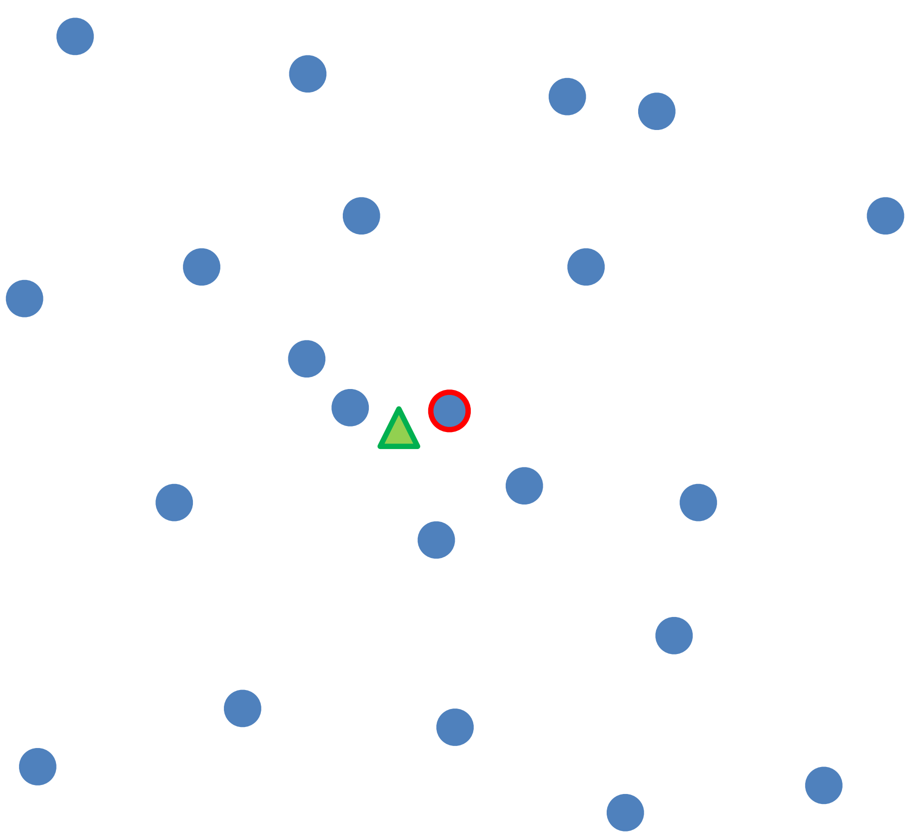


Key Intuition: Reads have lots of redundancy!

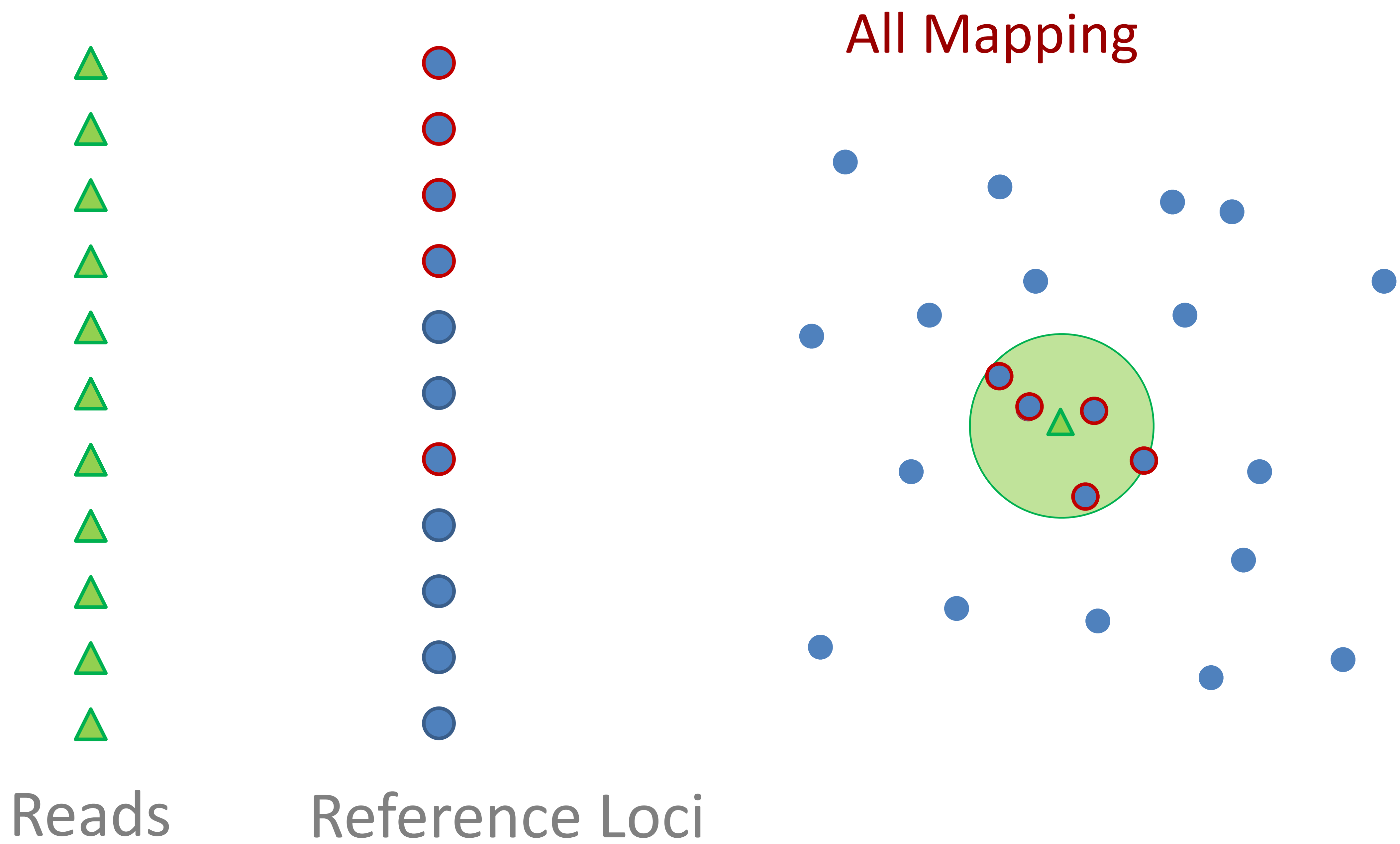
NGS-Mapping in Personal Genomics Era



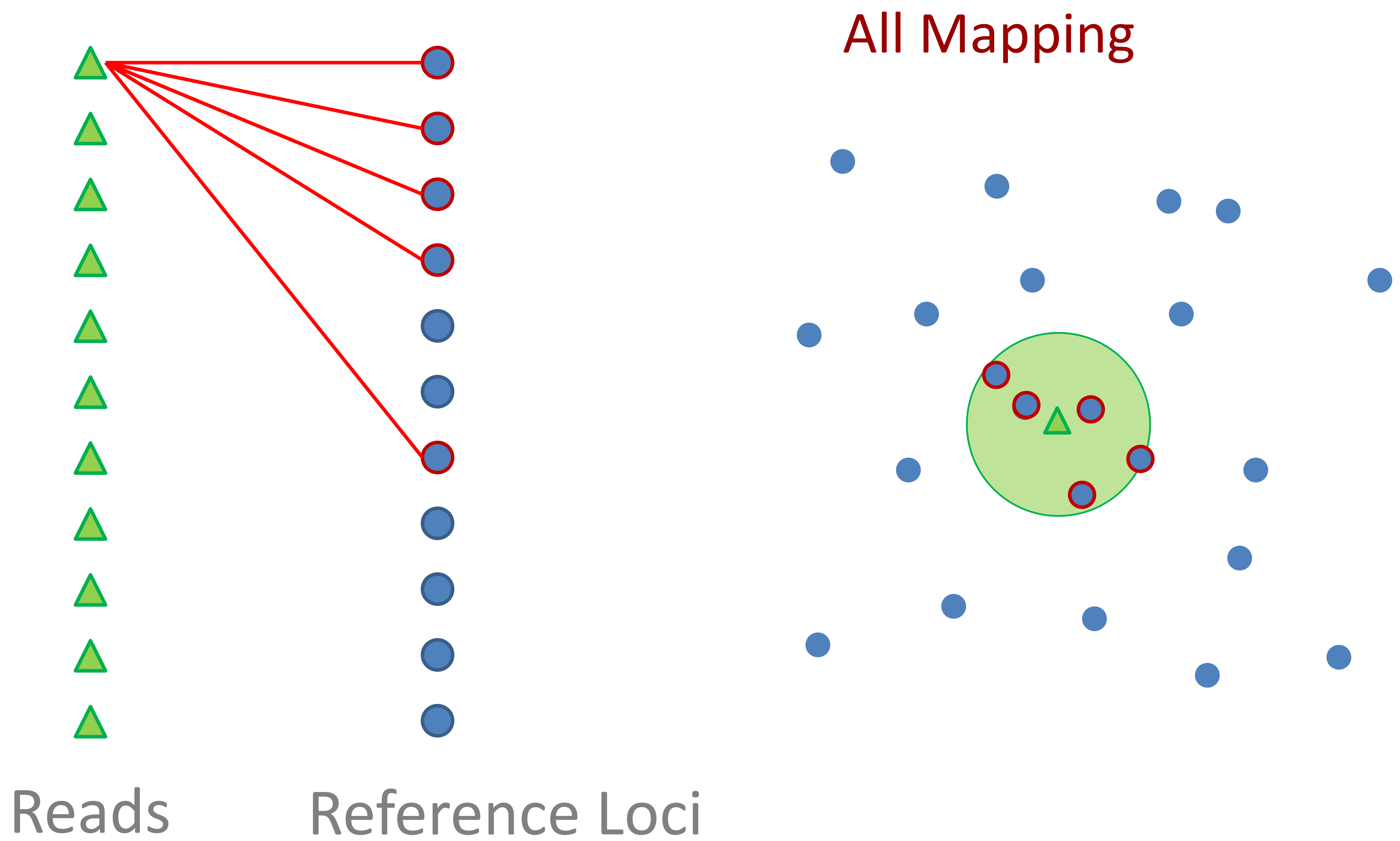
Best Mapping



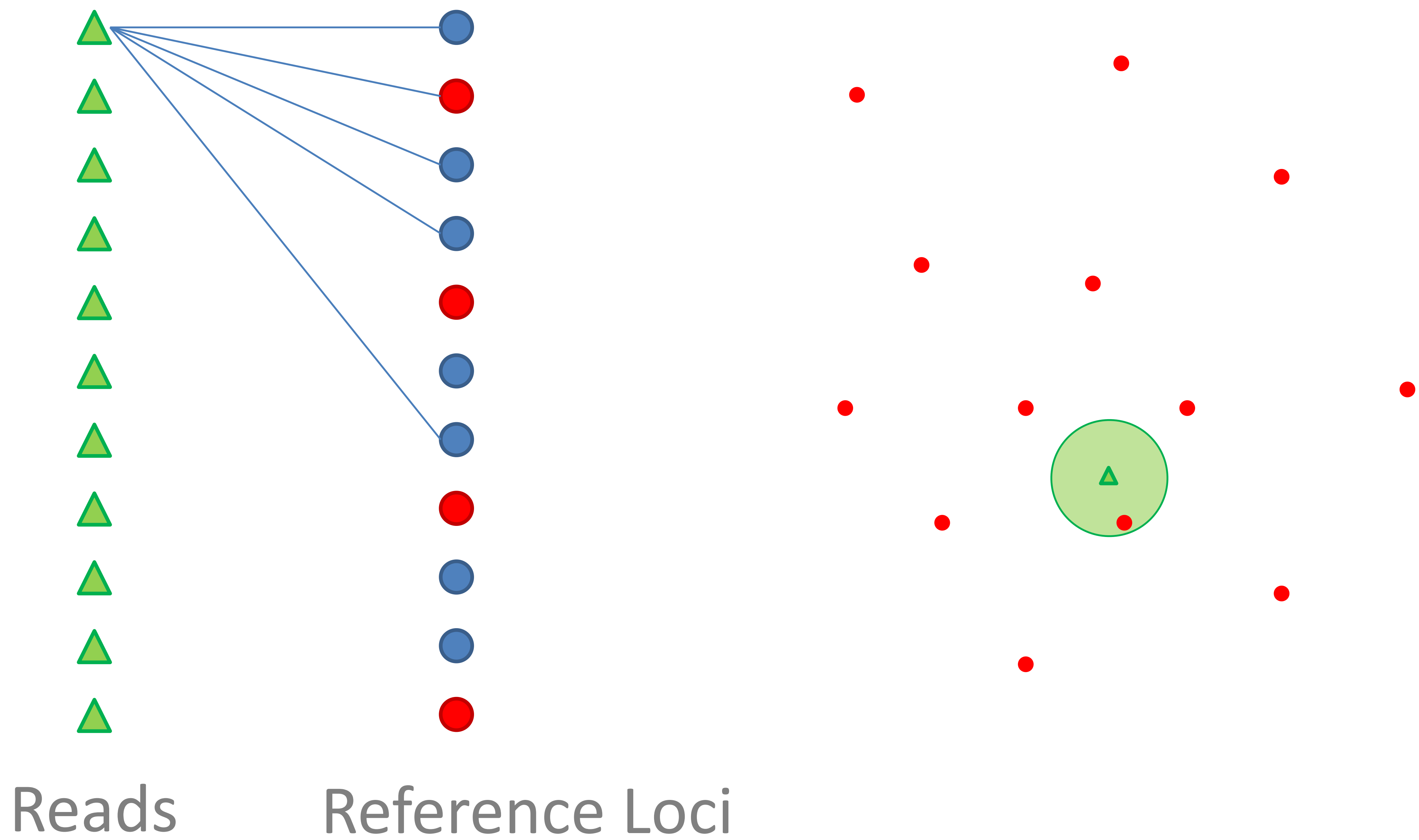
NGS-Mapping in Personal Genomics Era



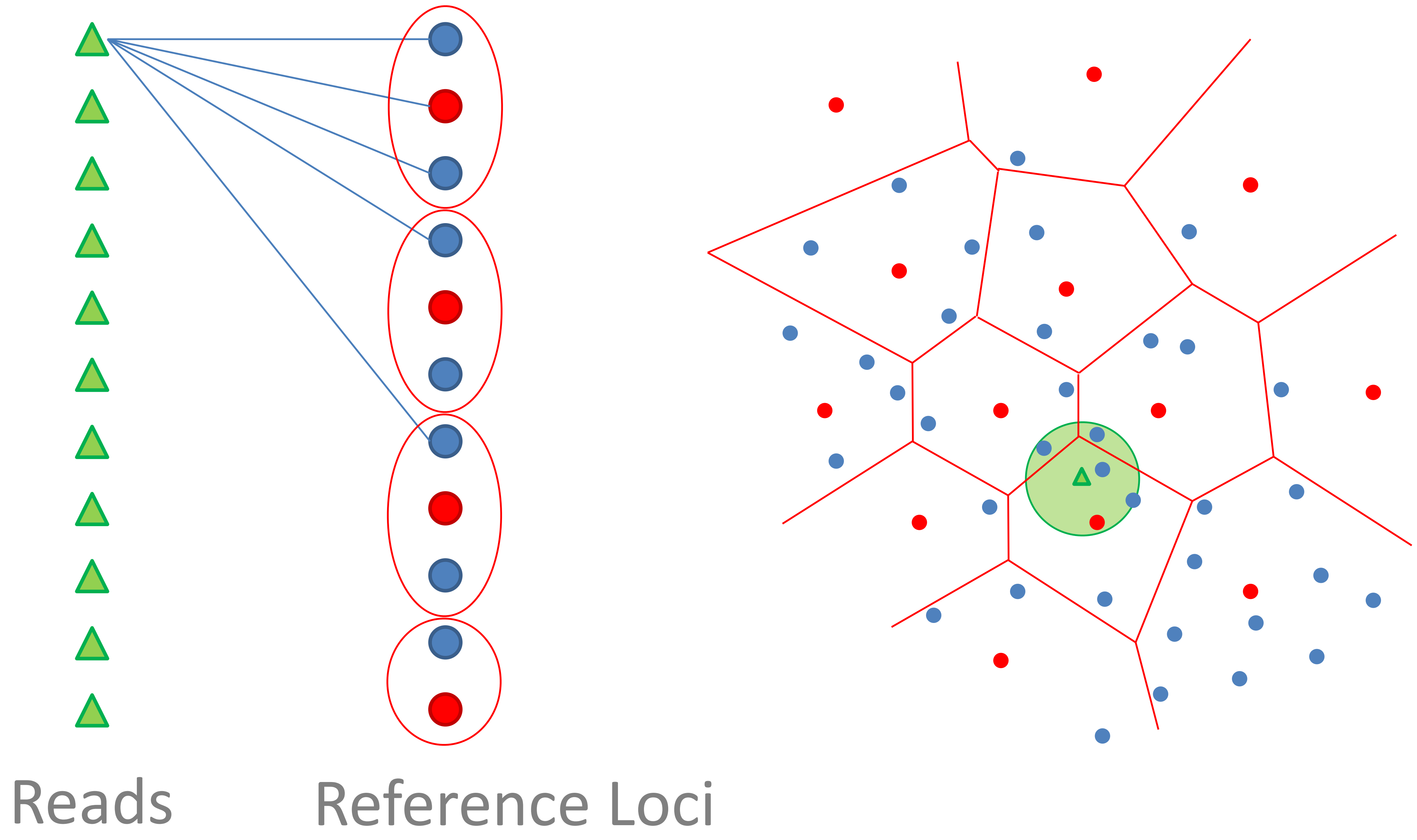
NGS-Mapping in Personal Genomics Era



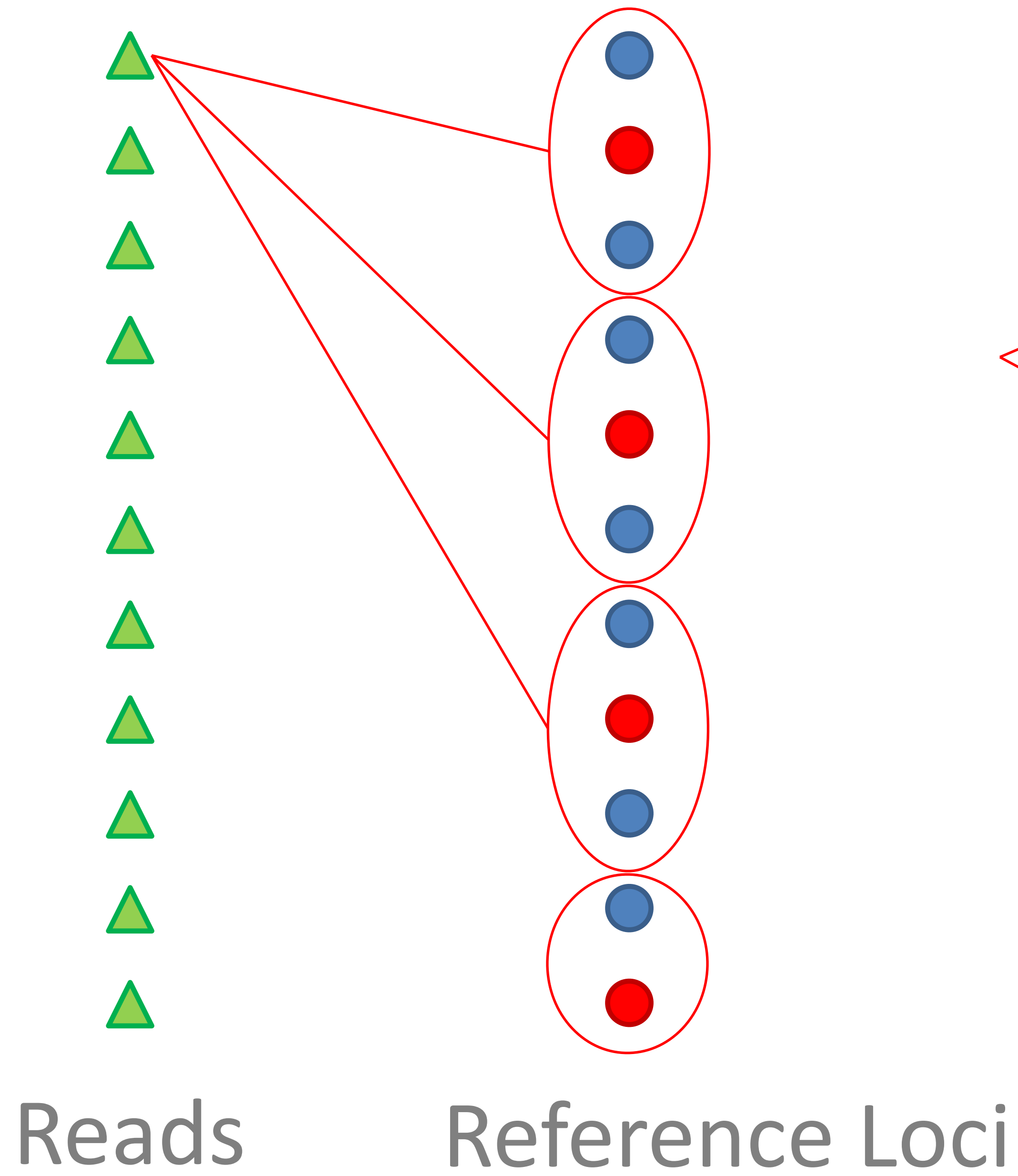
CORA [Yorukoglu, et al. accepted for publication]

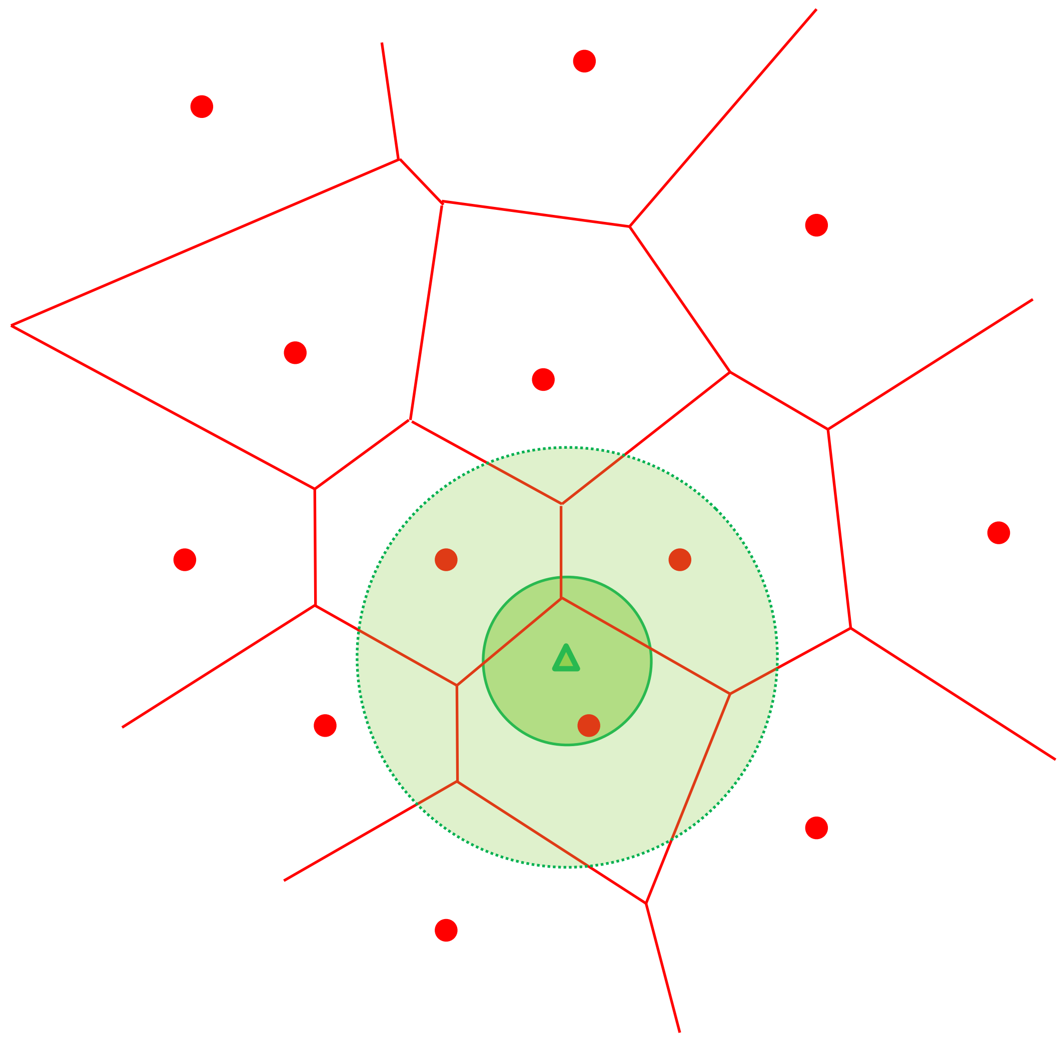


CORA [Yorukoglu, et al. accepted for publication]

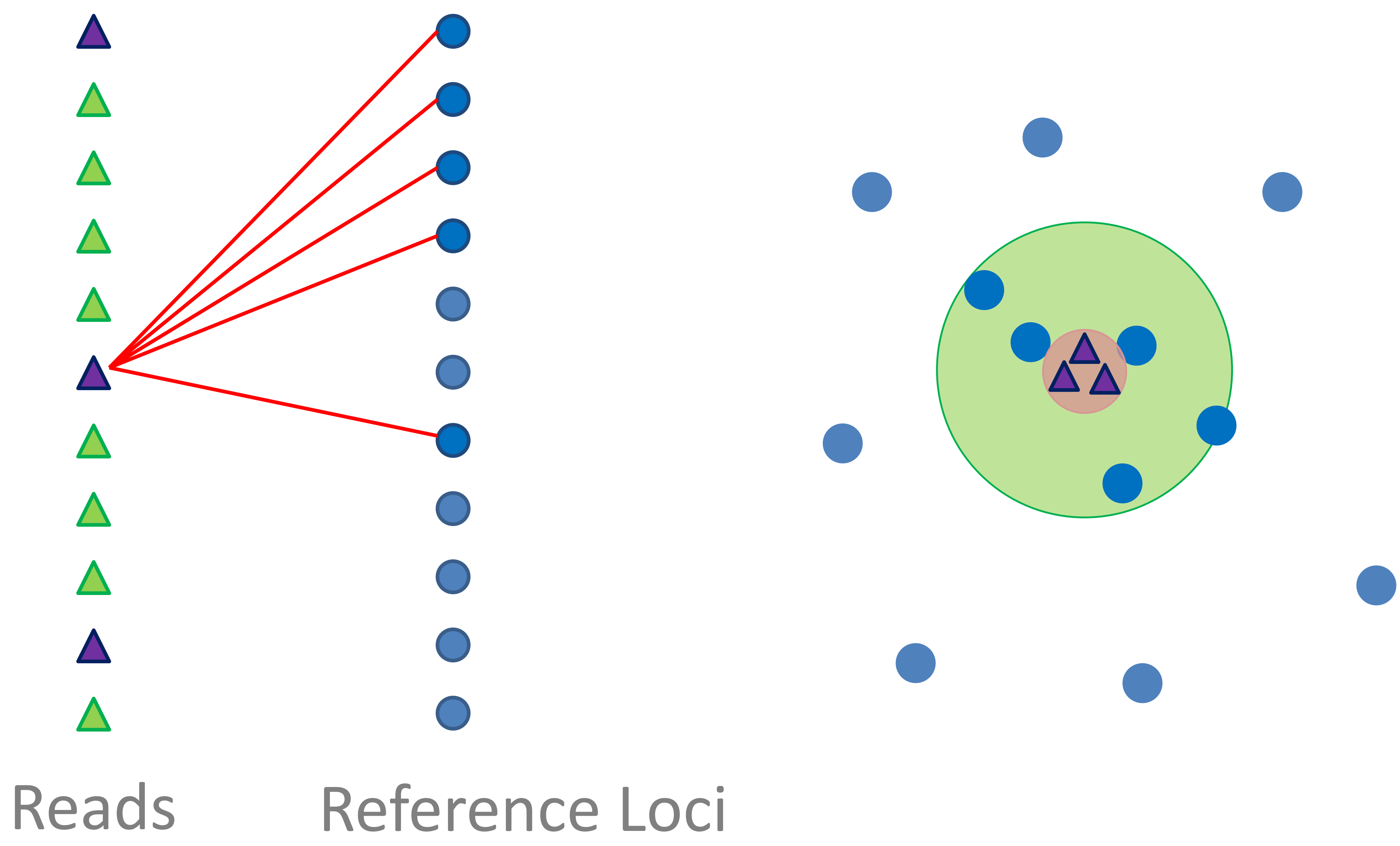


CORA [Yorukoglu, et al. accepted for publication]

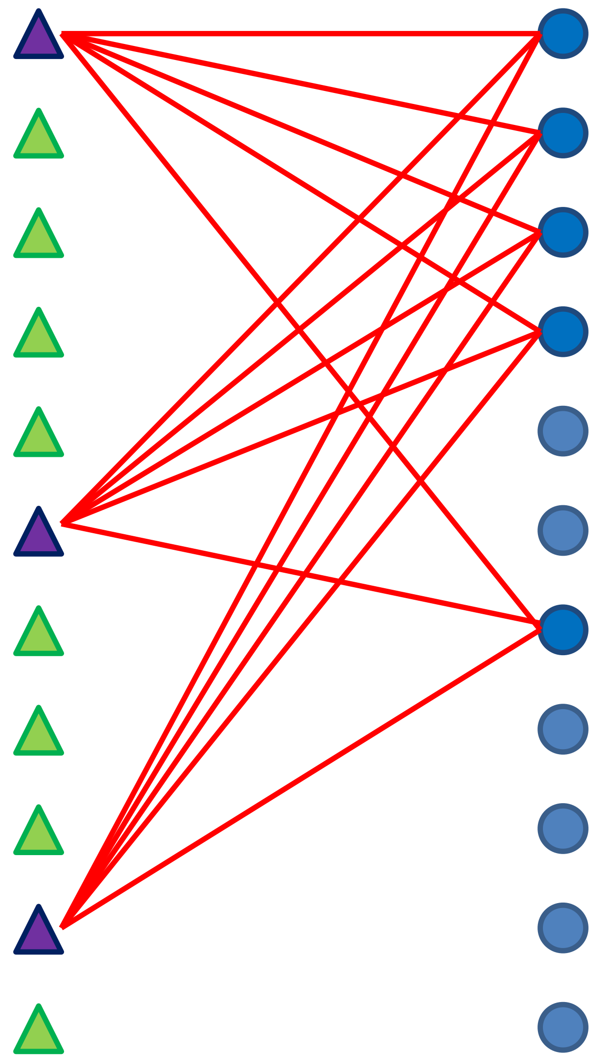




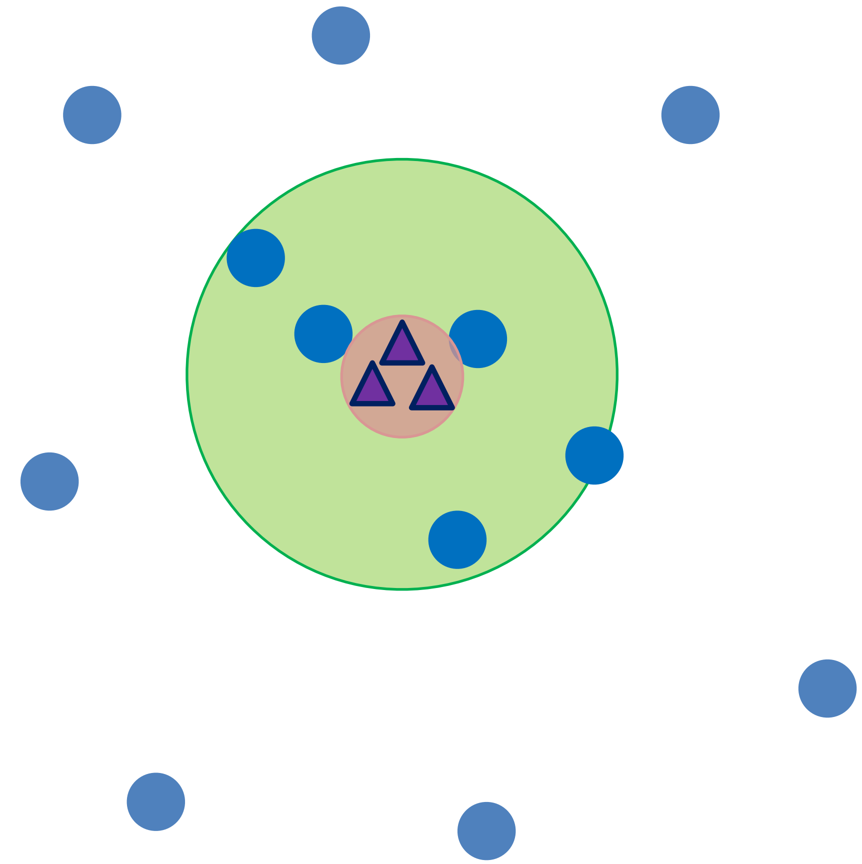
CORA [Yorukoglu, et al. accepted for publication]



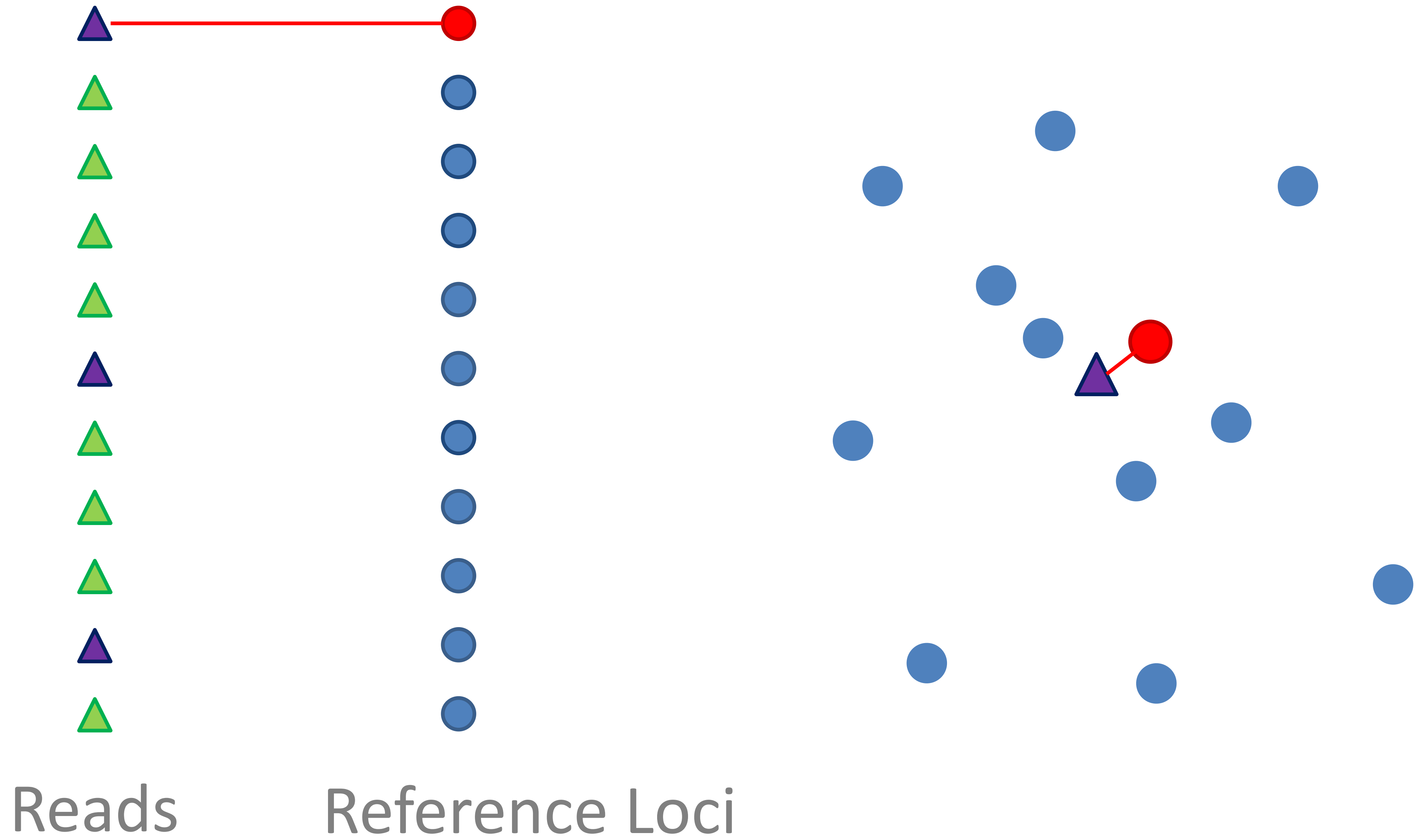
CORA [Yorukoglu, et al. accepted for publication]



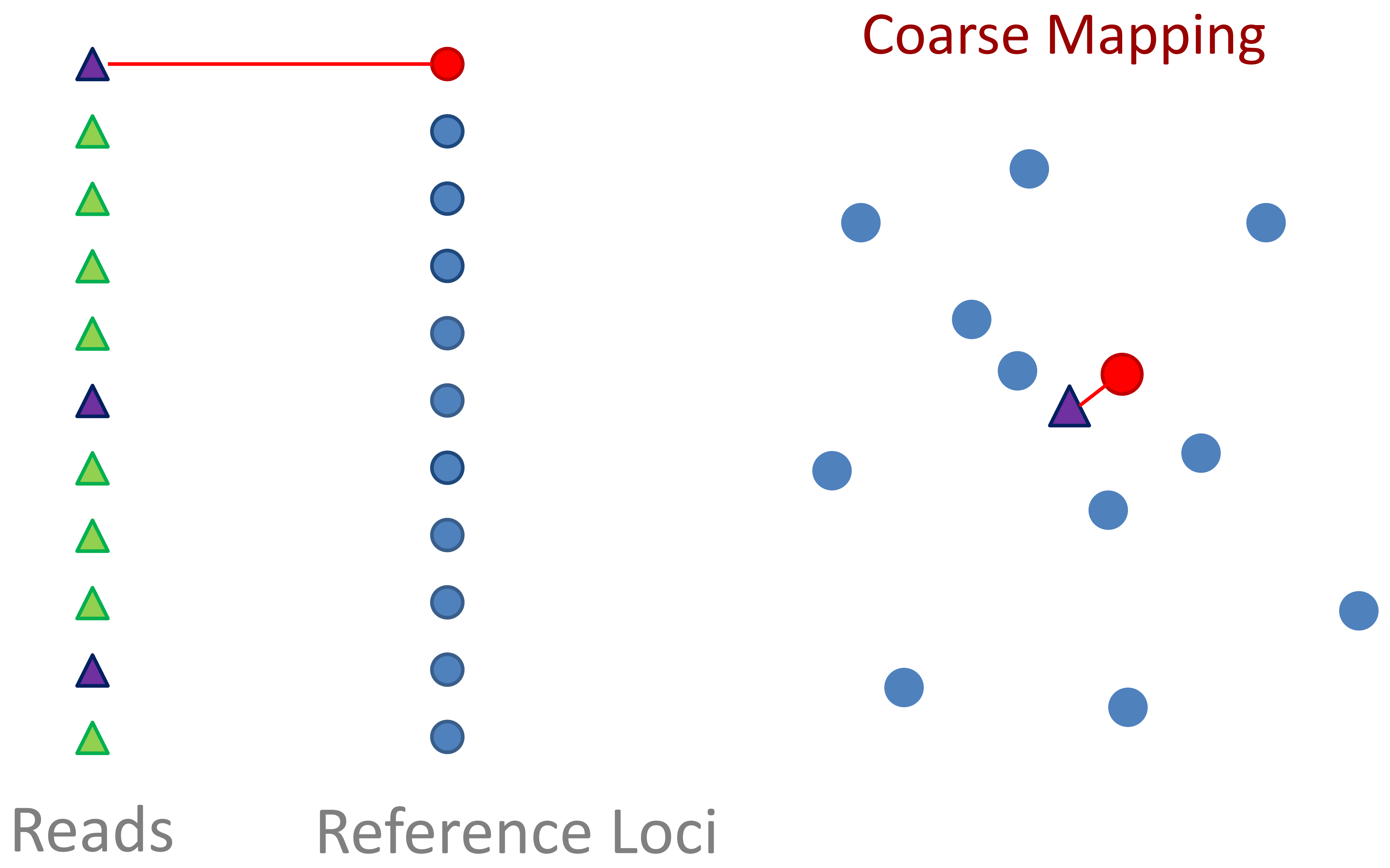
Reads Reference Loci



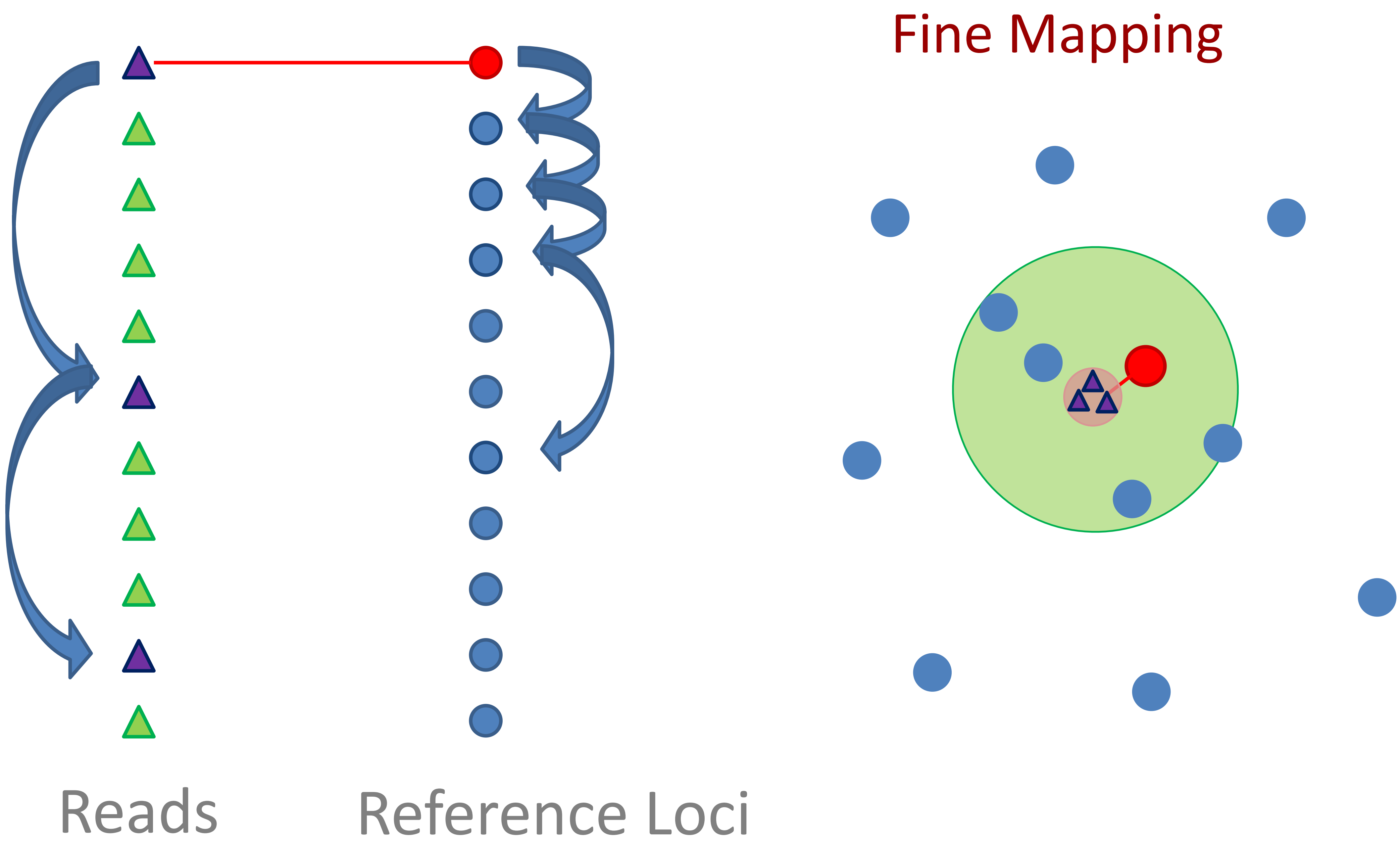
CORA [Yorukoglu, et al. accepted for publication]



CORA [Yorukoglu, et al. accepted for publication]

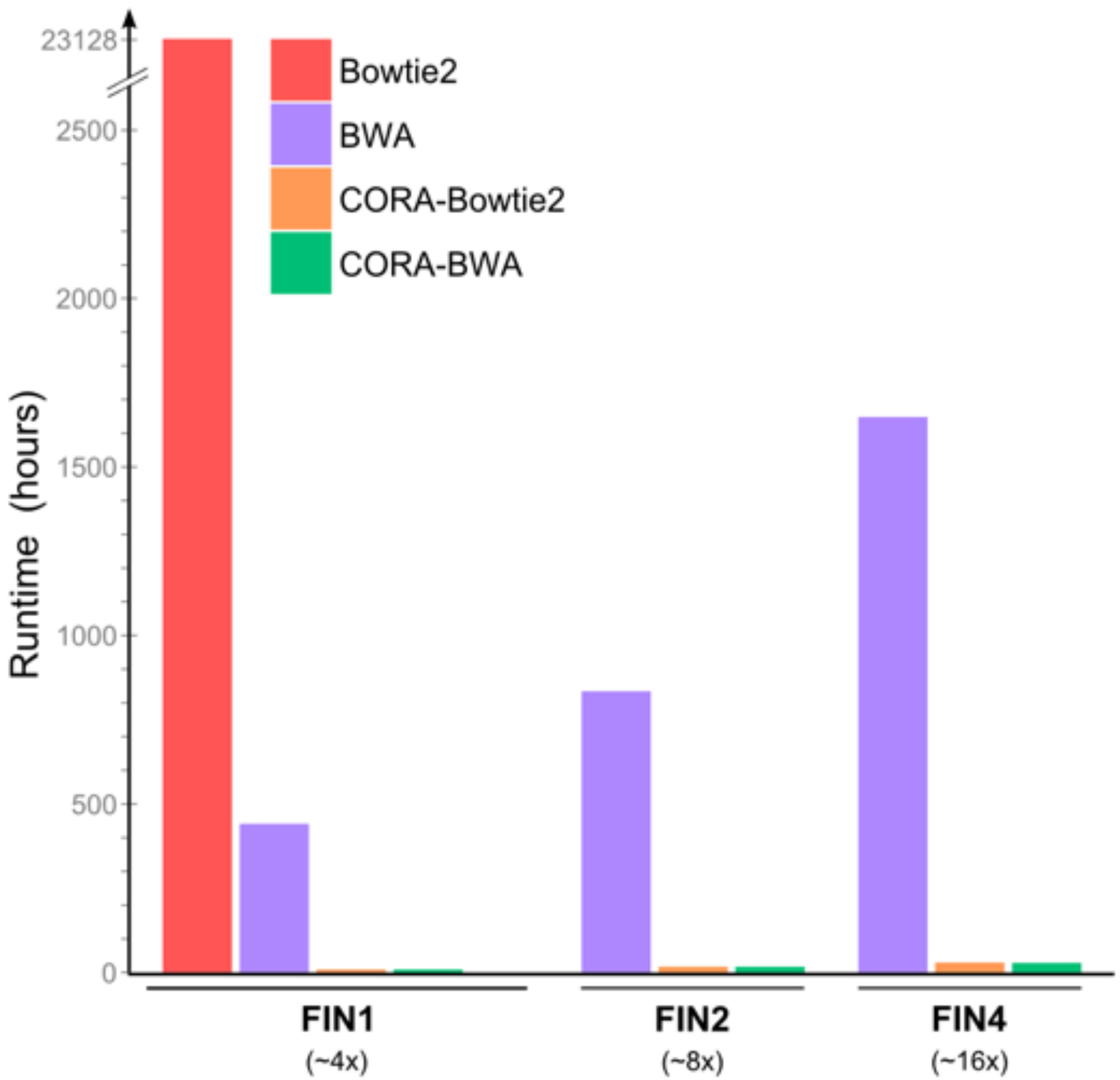


CORA [Yorukoglu, et al. accepted for publication]



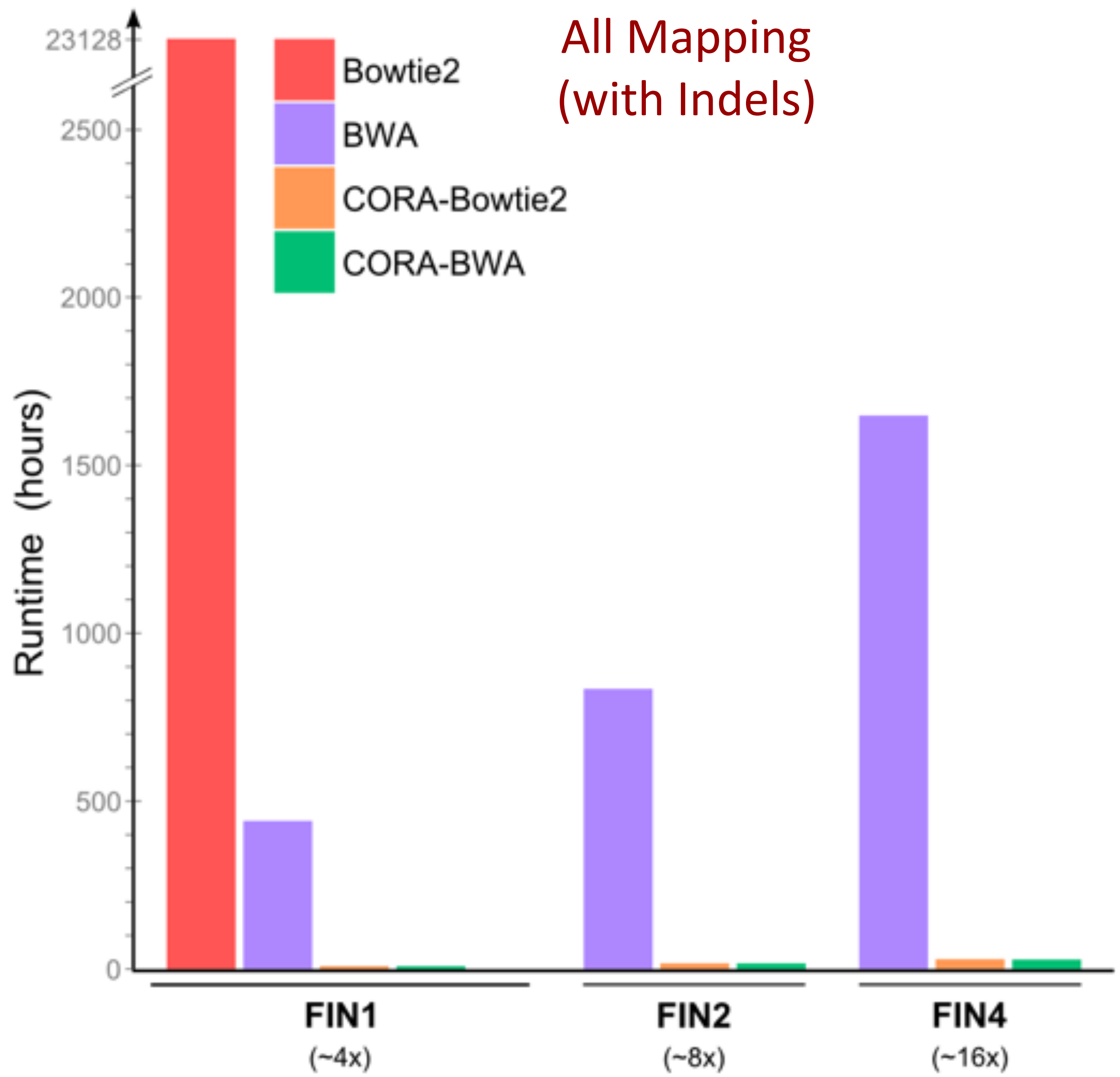
Result Highlights

CORA: 1000 Genomes Project



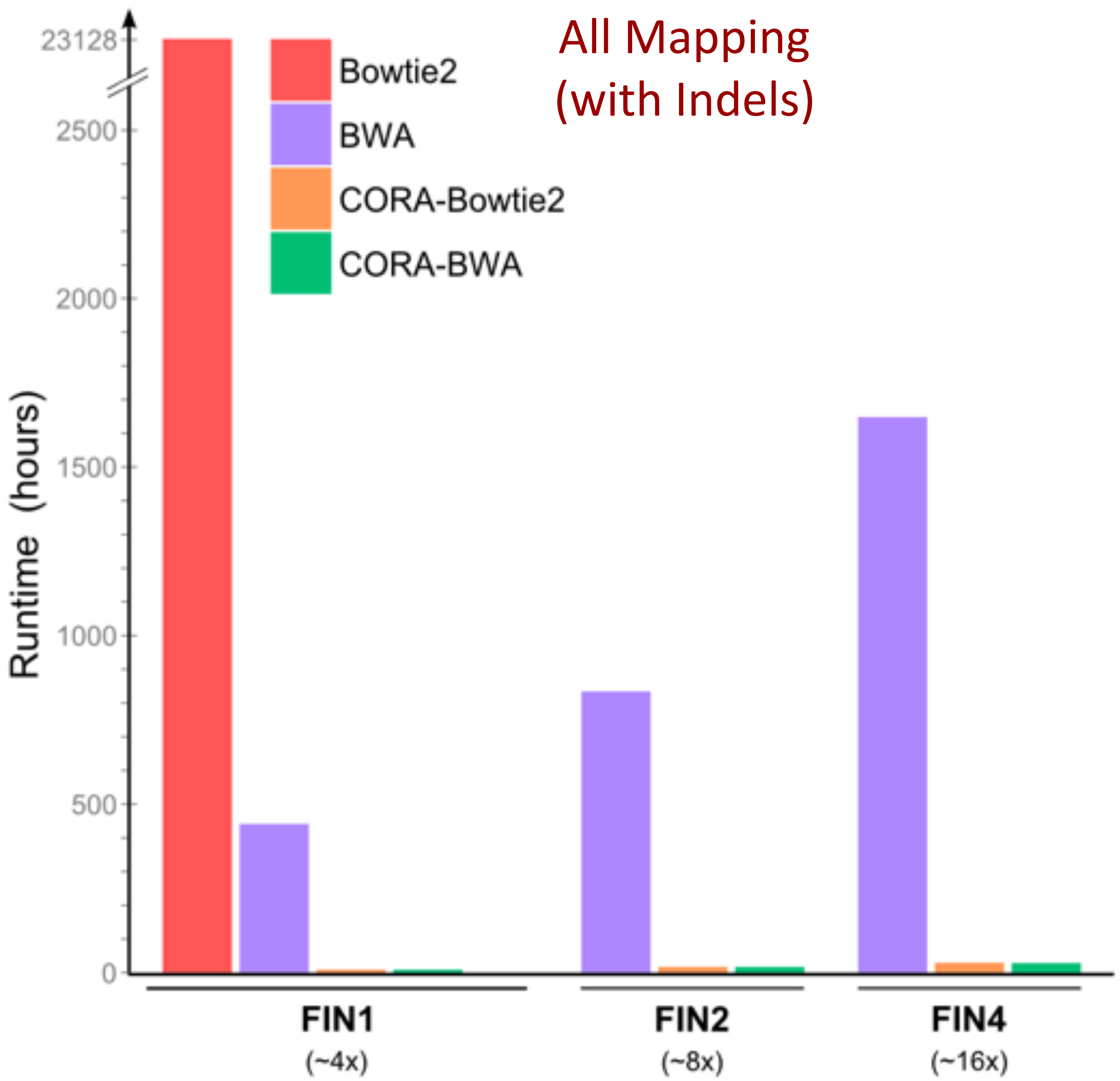
Result Highlights

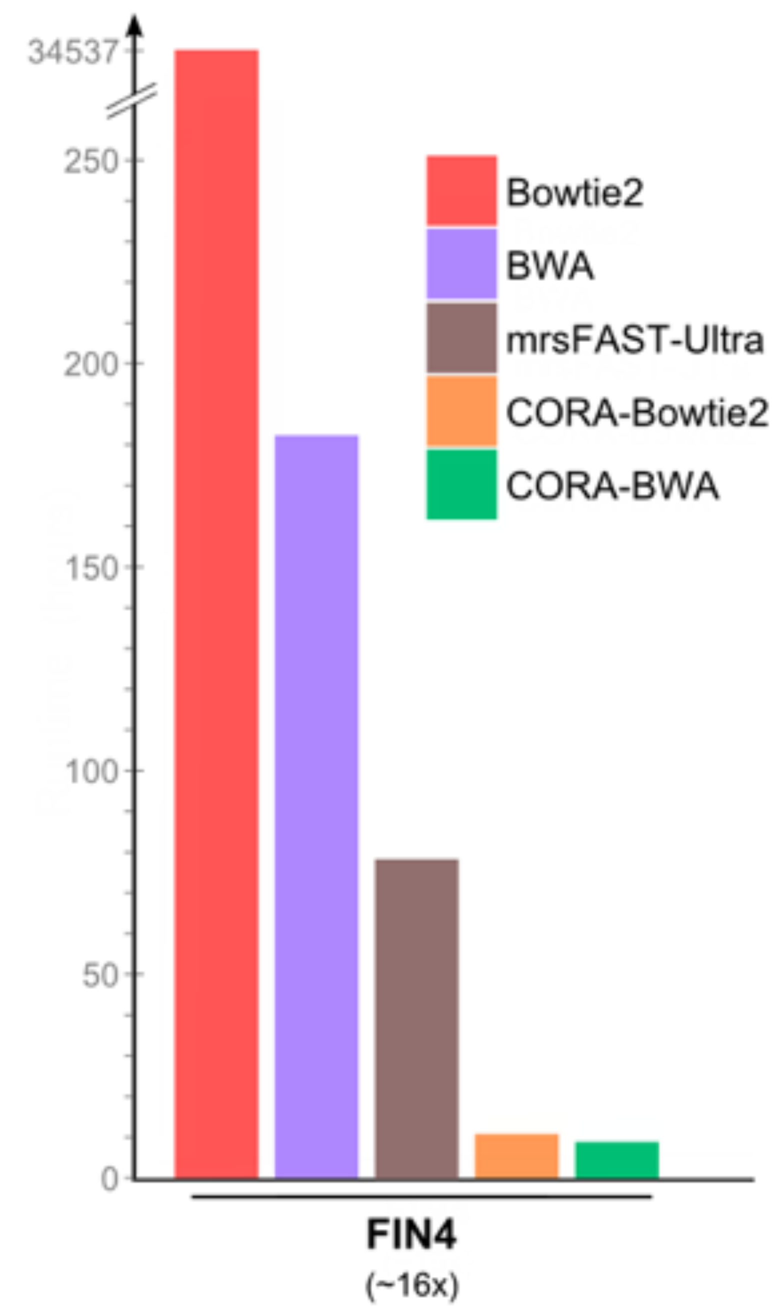
CORA: 1000 Genomes Project



Result Highlights

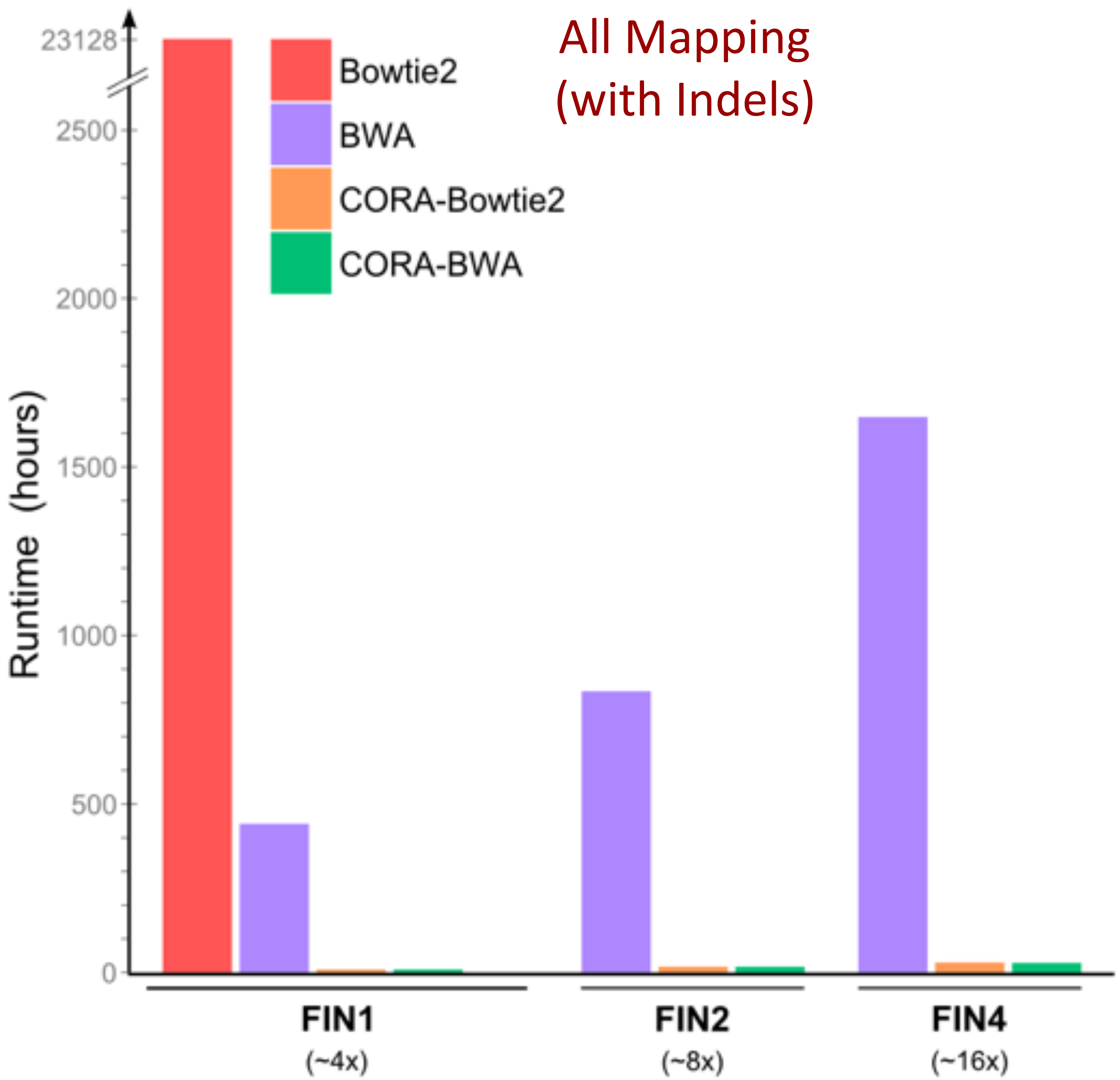
CORA: 1000 Genomes Project

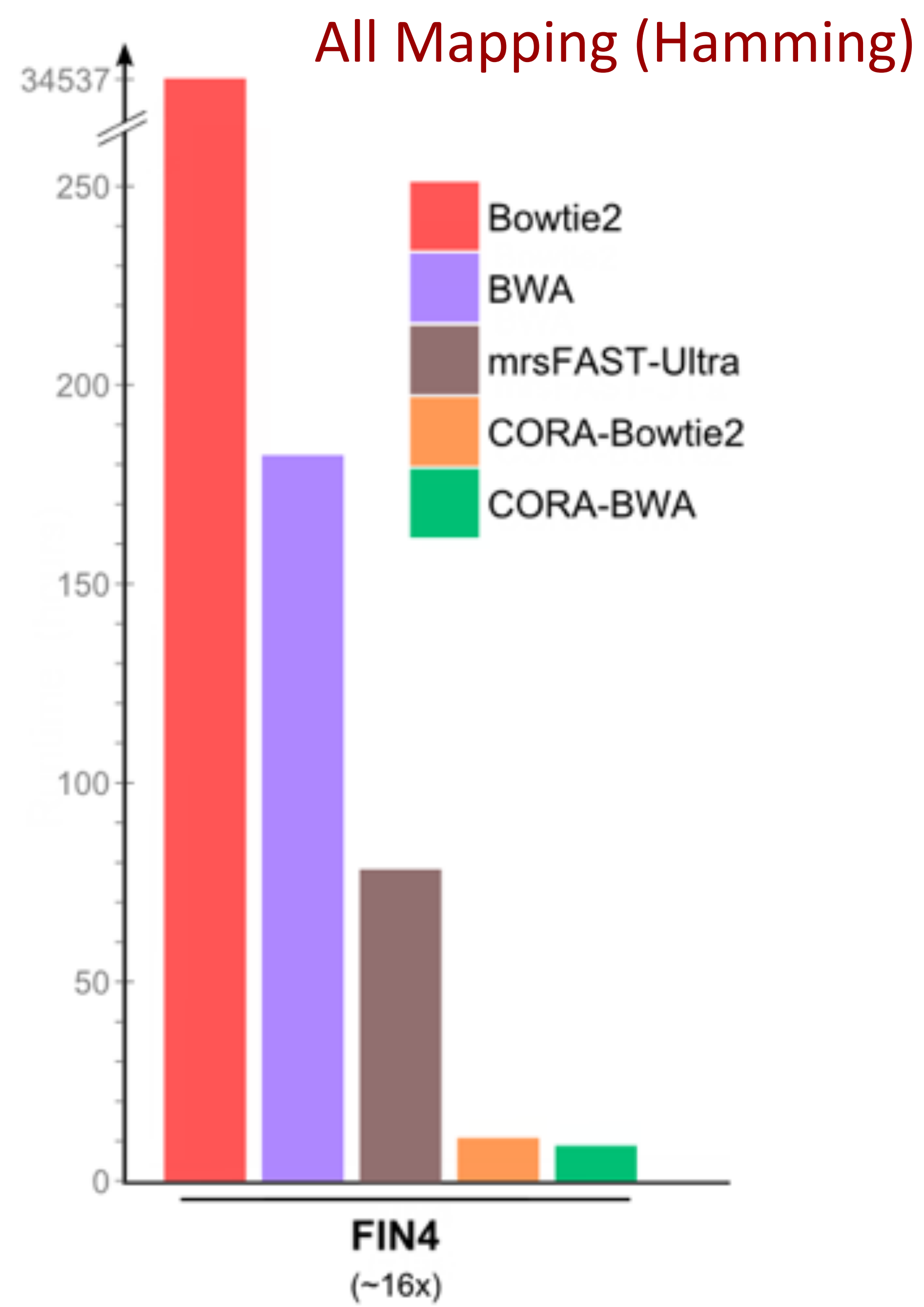




Result Highlights

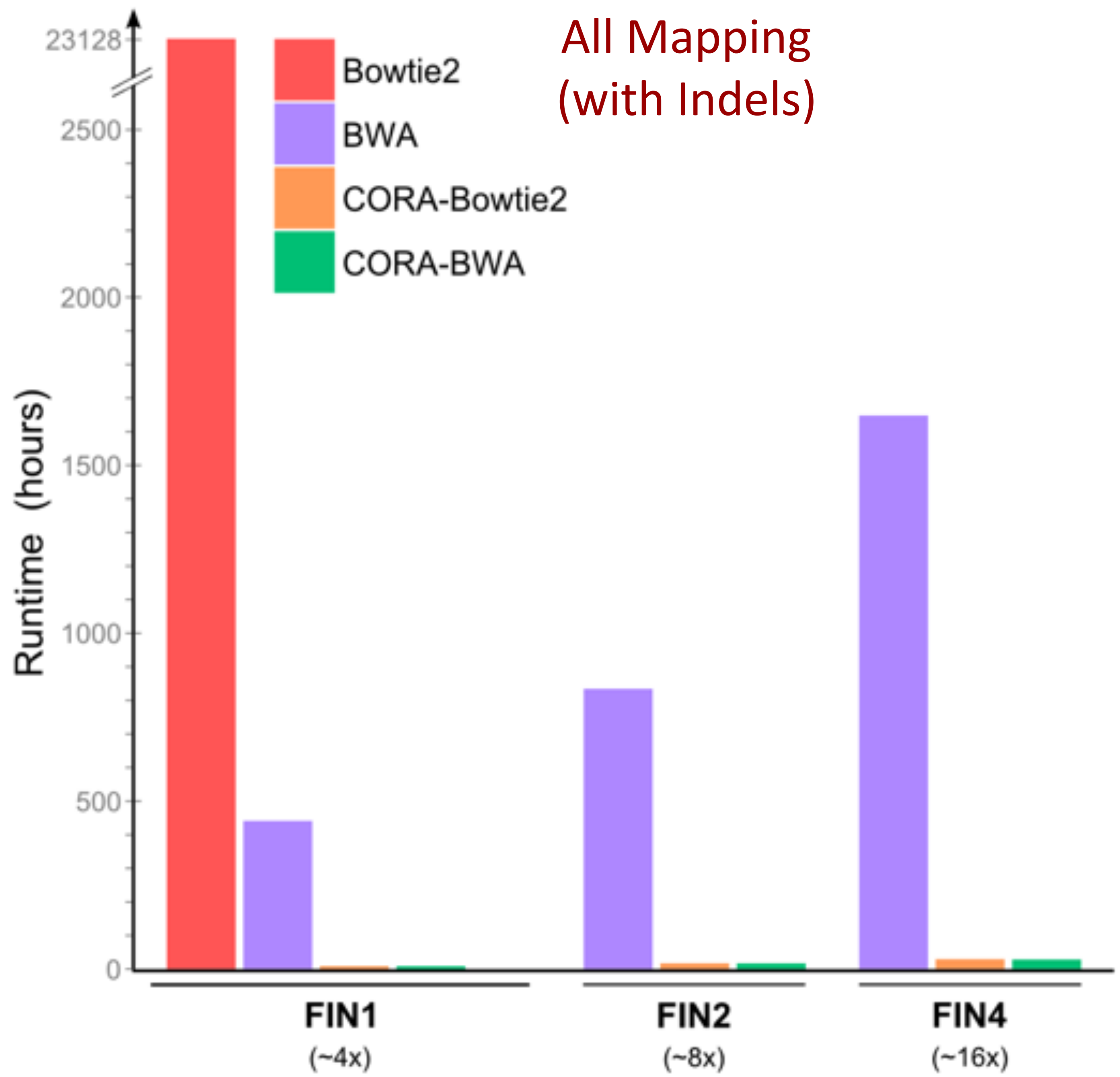
CORA: 1000 Genomes Project

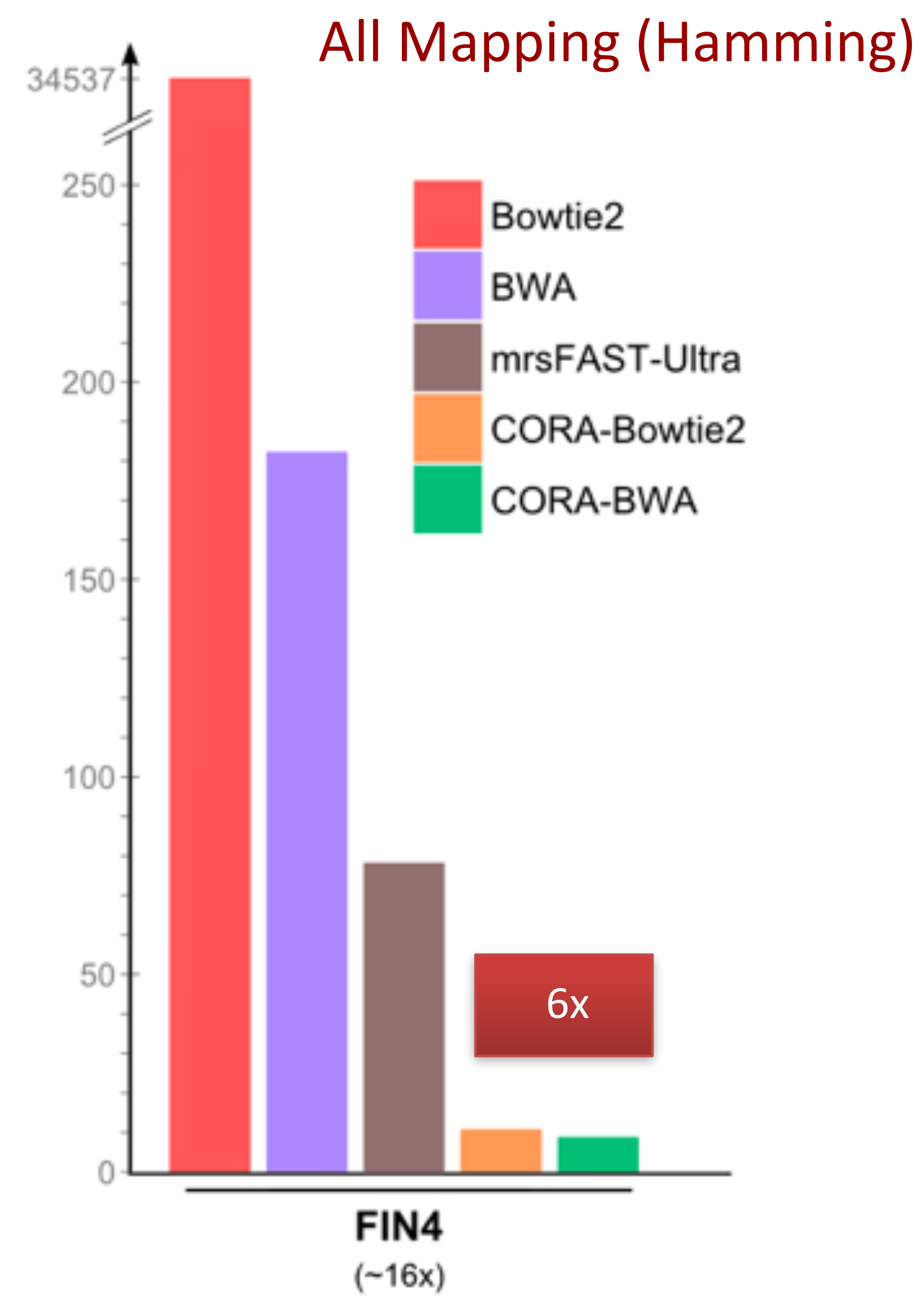




Result Highlights

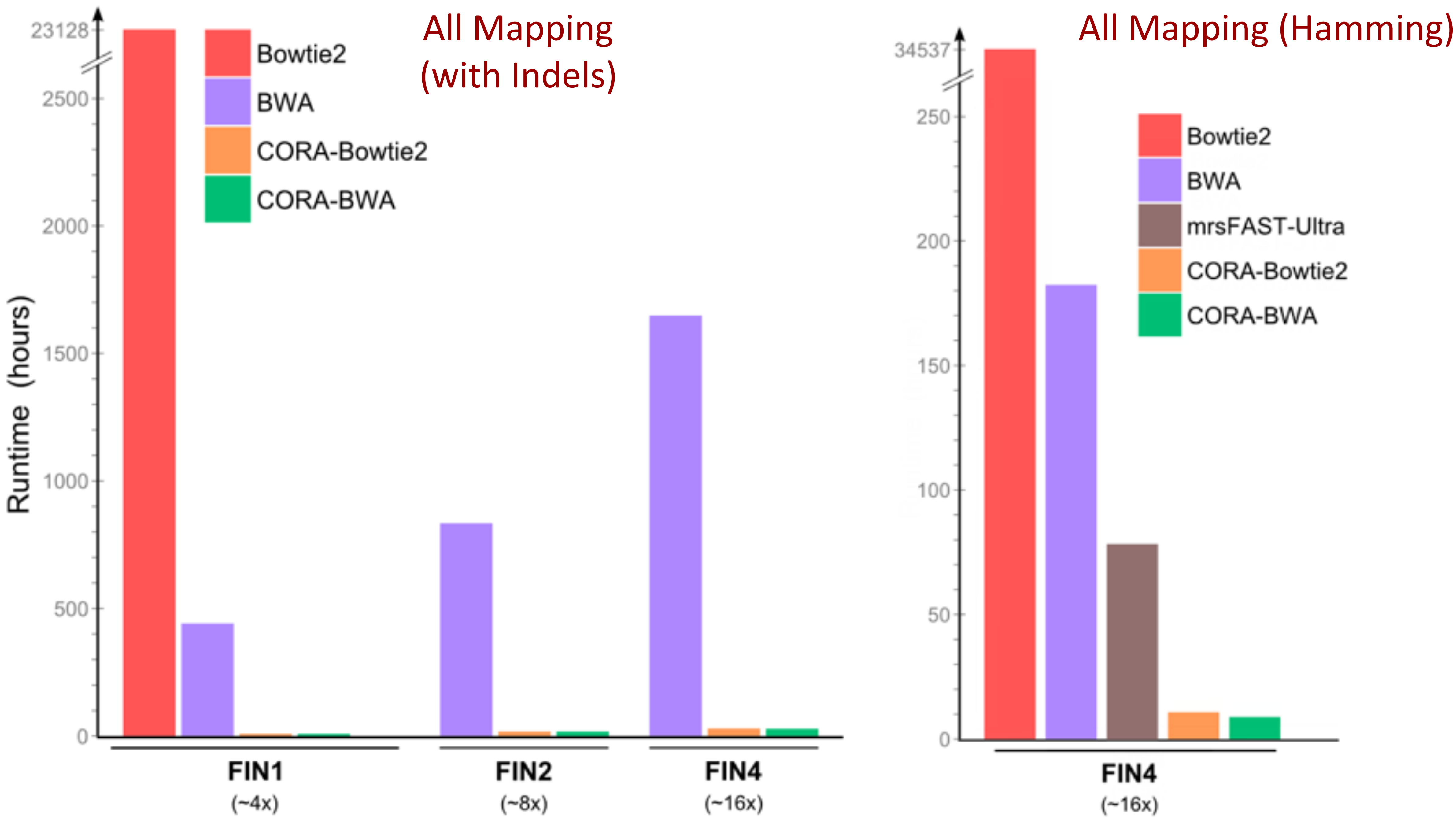
CORA: 1000 Genomes Project





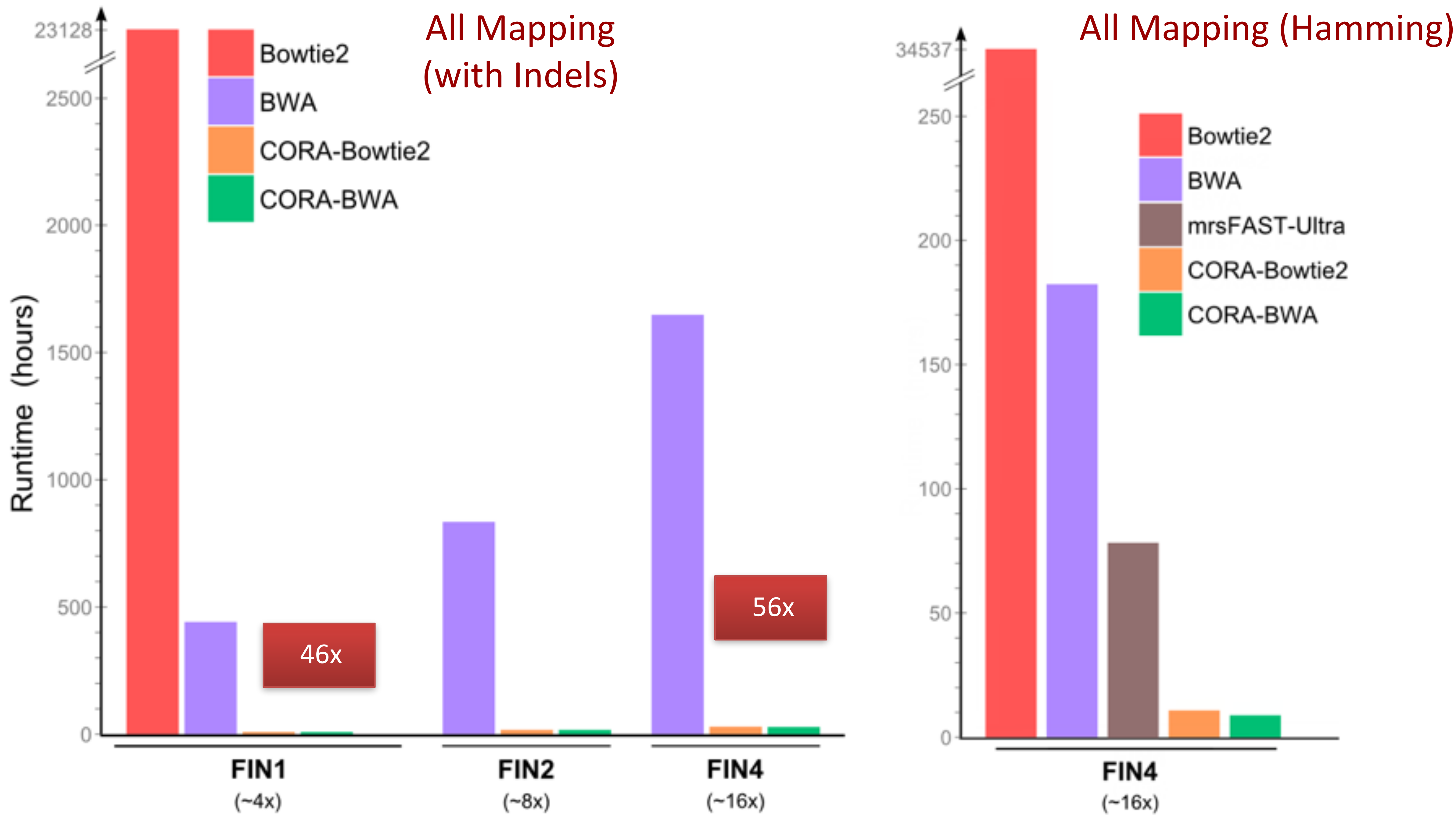
Result Highlights

CORA: 1000 Genomes Project



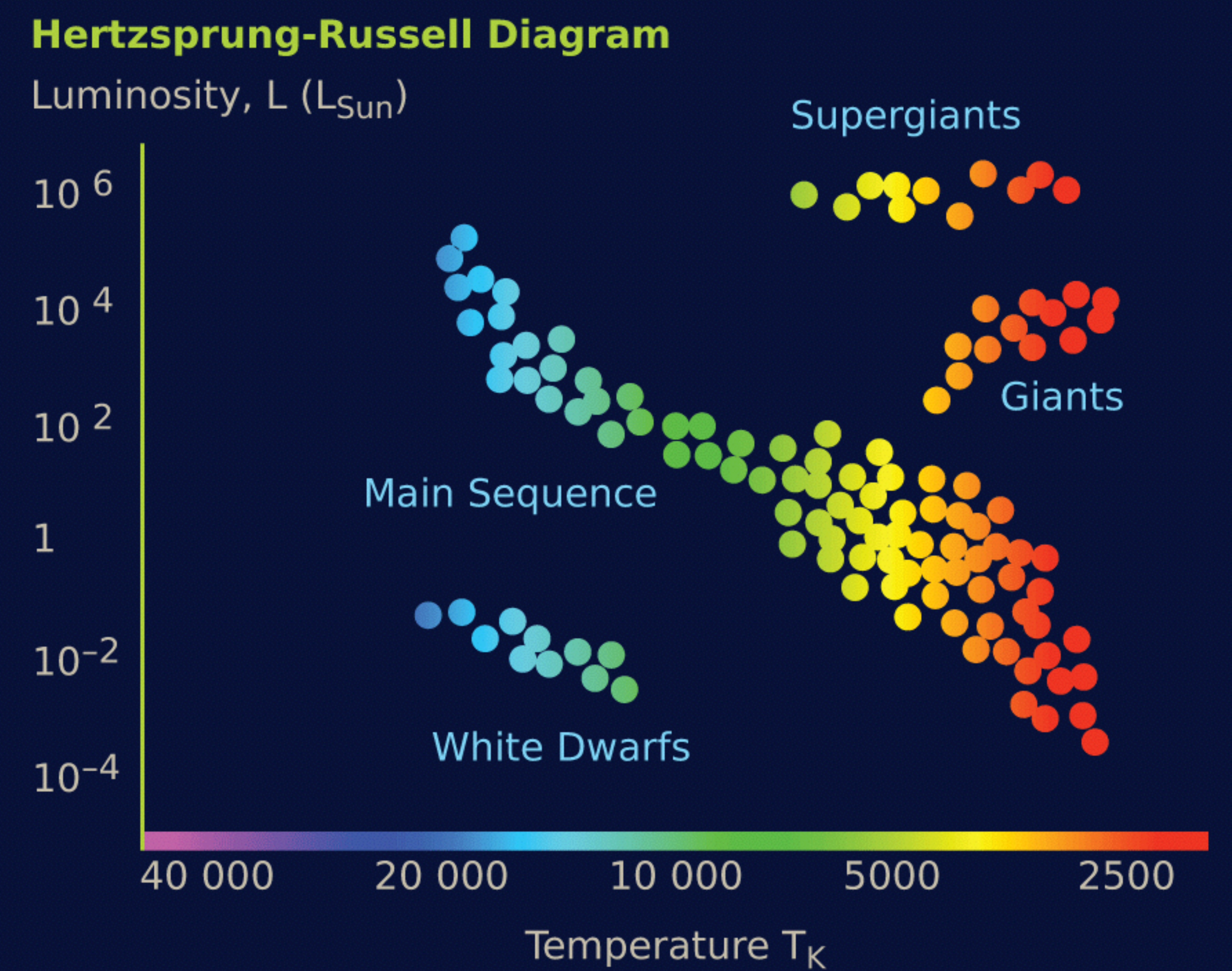
Result Highlights

CORA: 1000 Genomes Project



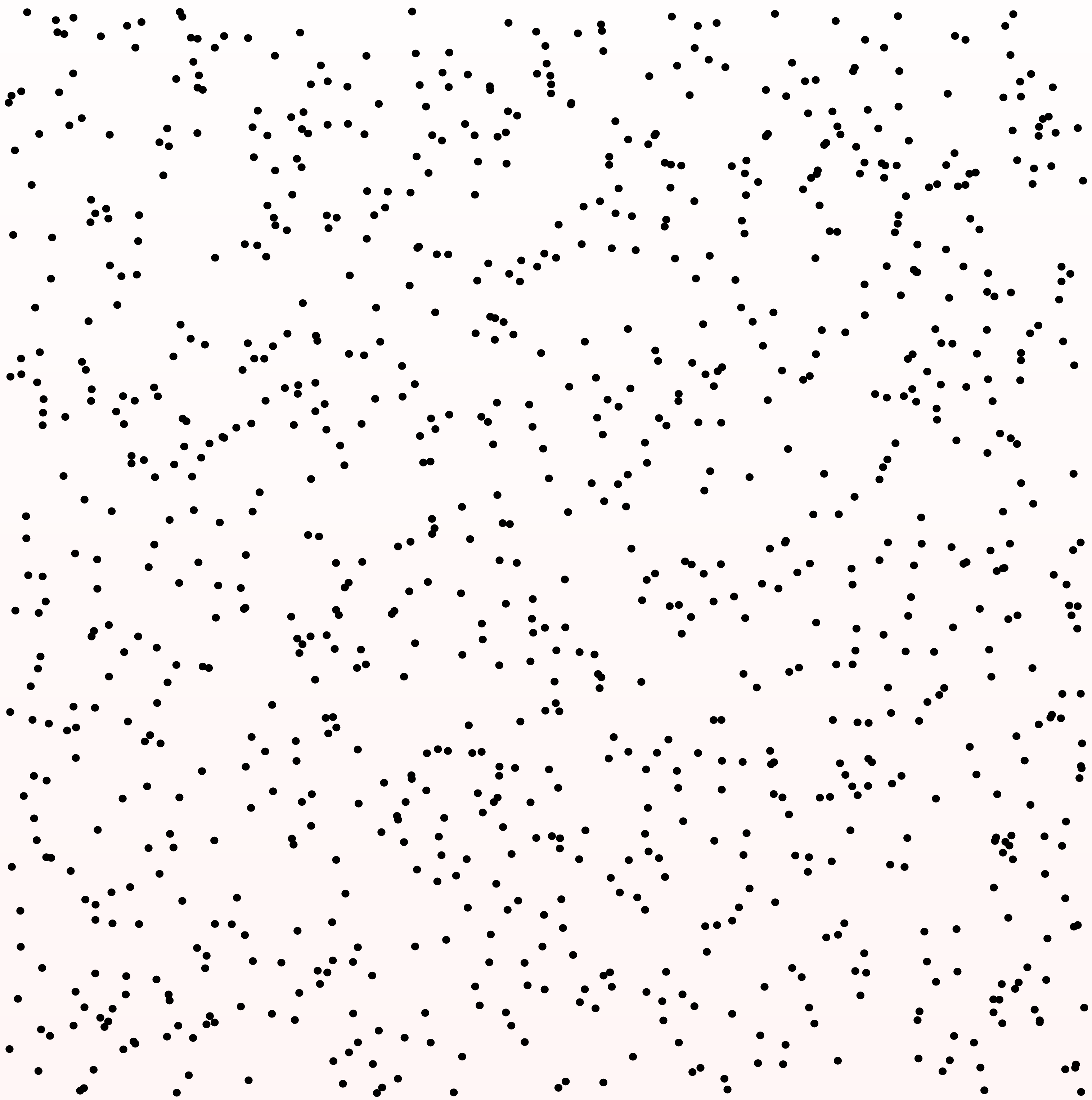
Scales sublinearly with very high sensitivity on real genomic data

Data constrained by physical process

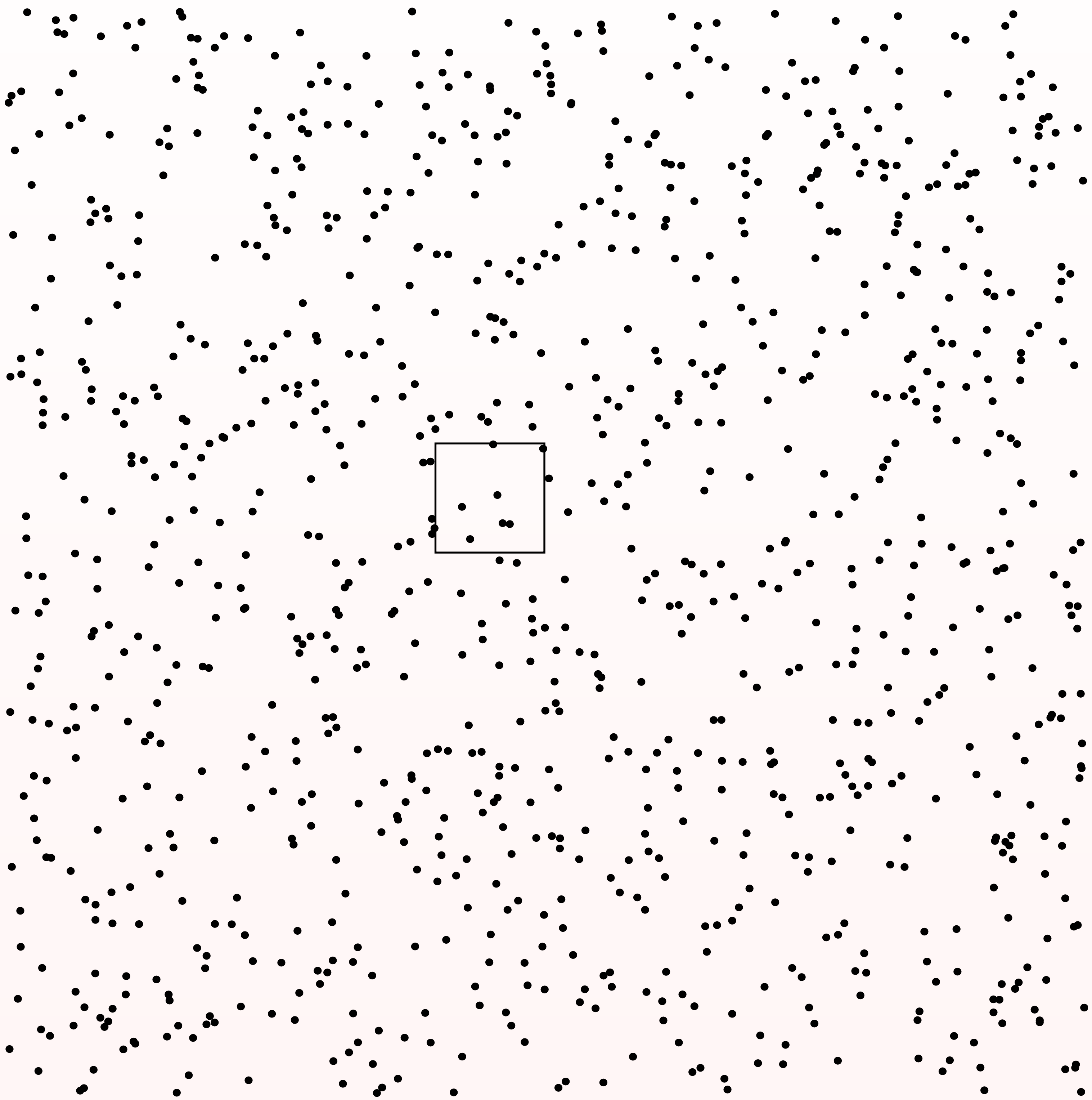


Hertzsprung-Russell Diagram
Luminosity, L (L_{Sun})
Supergiants
Giants
Main Sequence
White Dwarfs
10^6
10^4
10^2
1
10^{-2}
10^{-4}
40 000
20 000
10 000
5000
2500
Temperature T_K

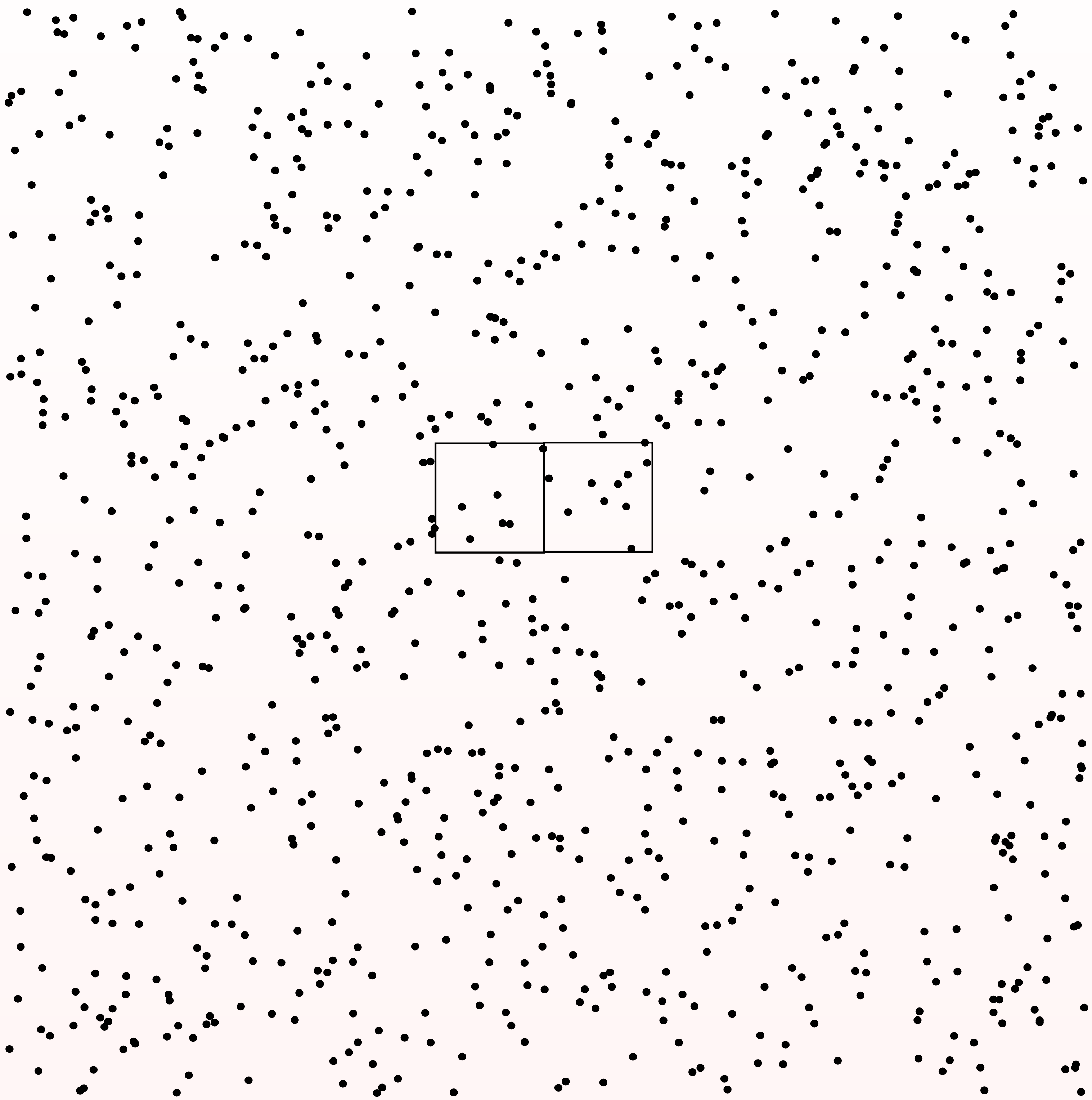
Grid search



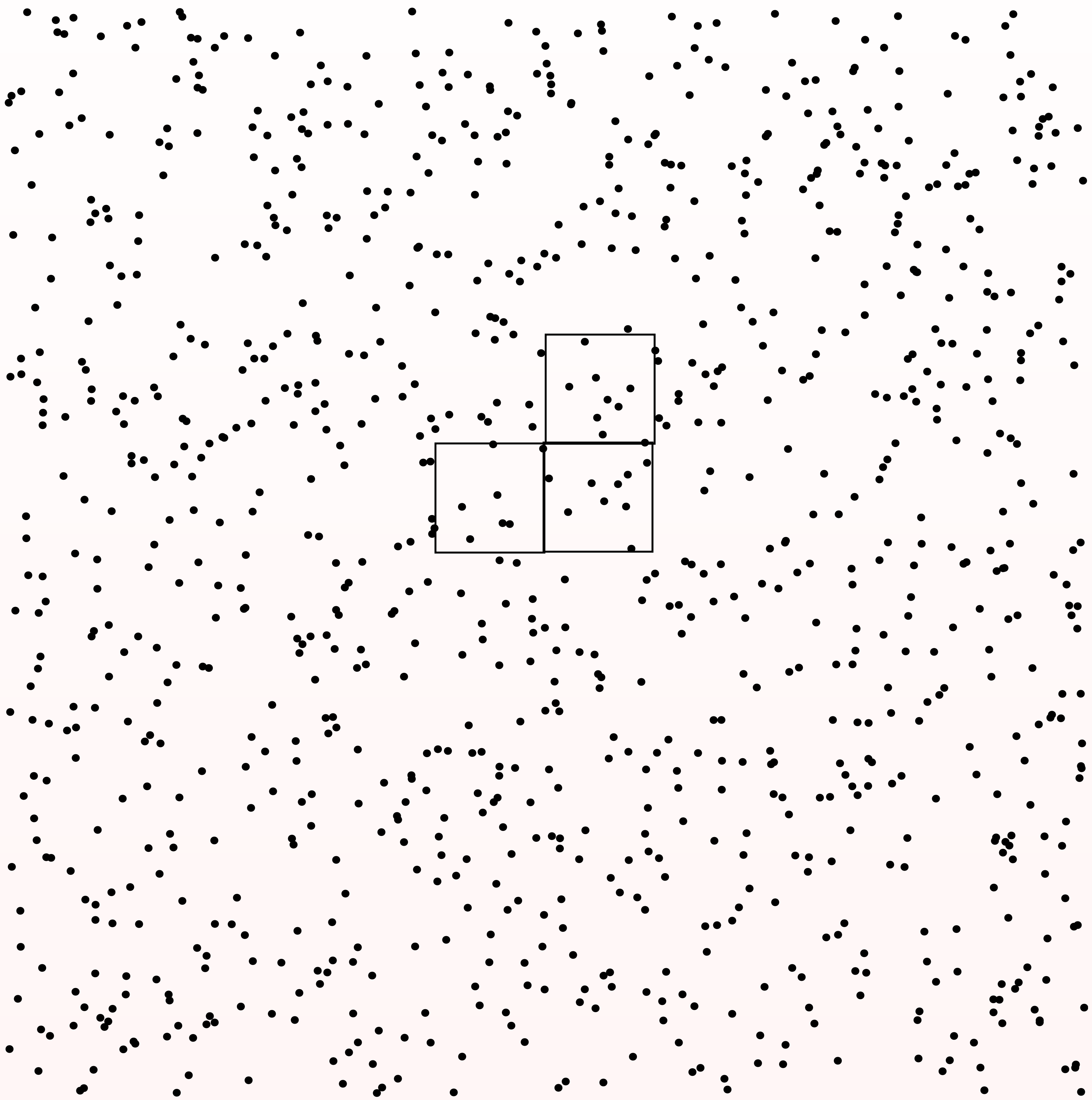
Grid search



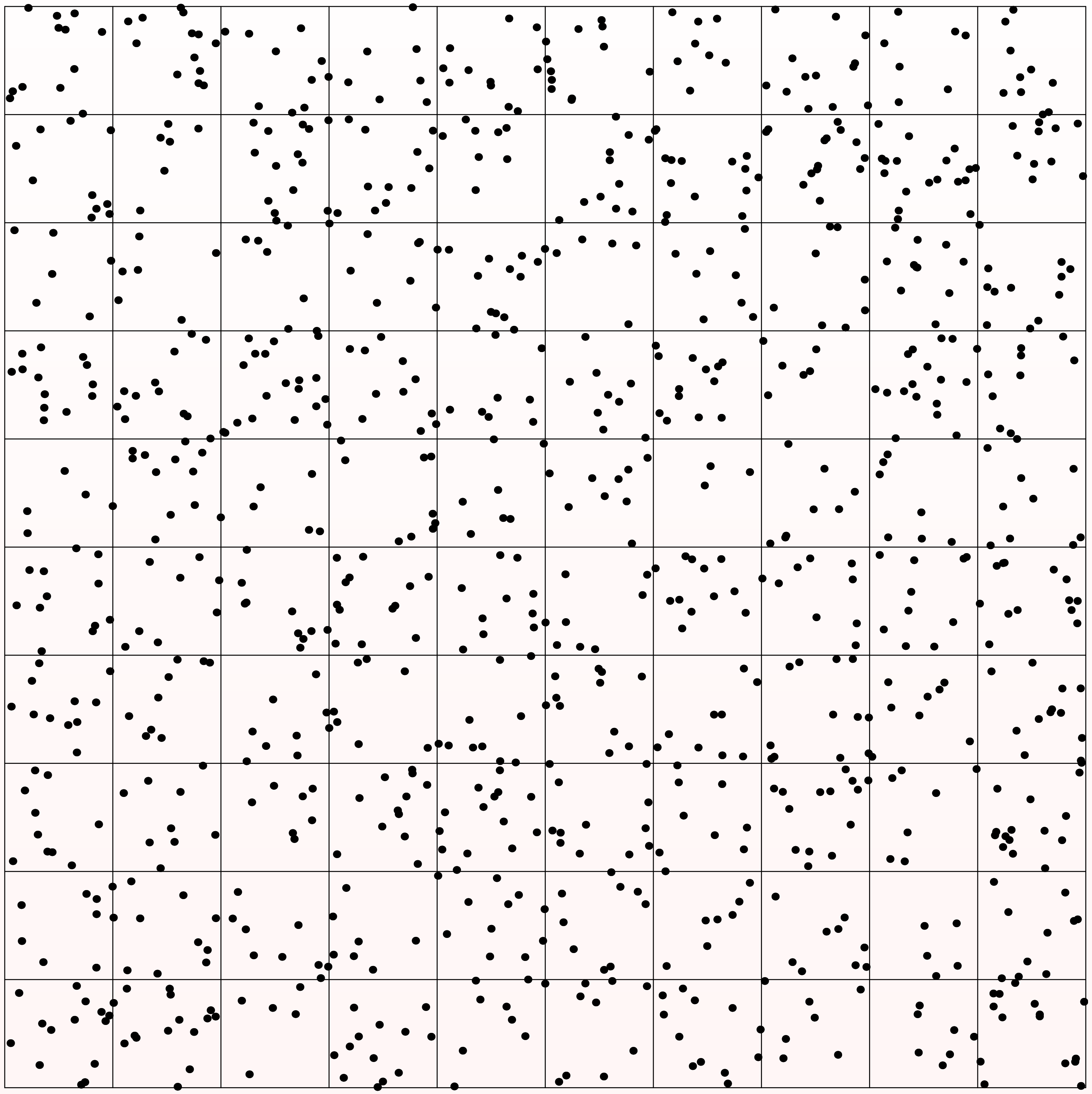
Grid search

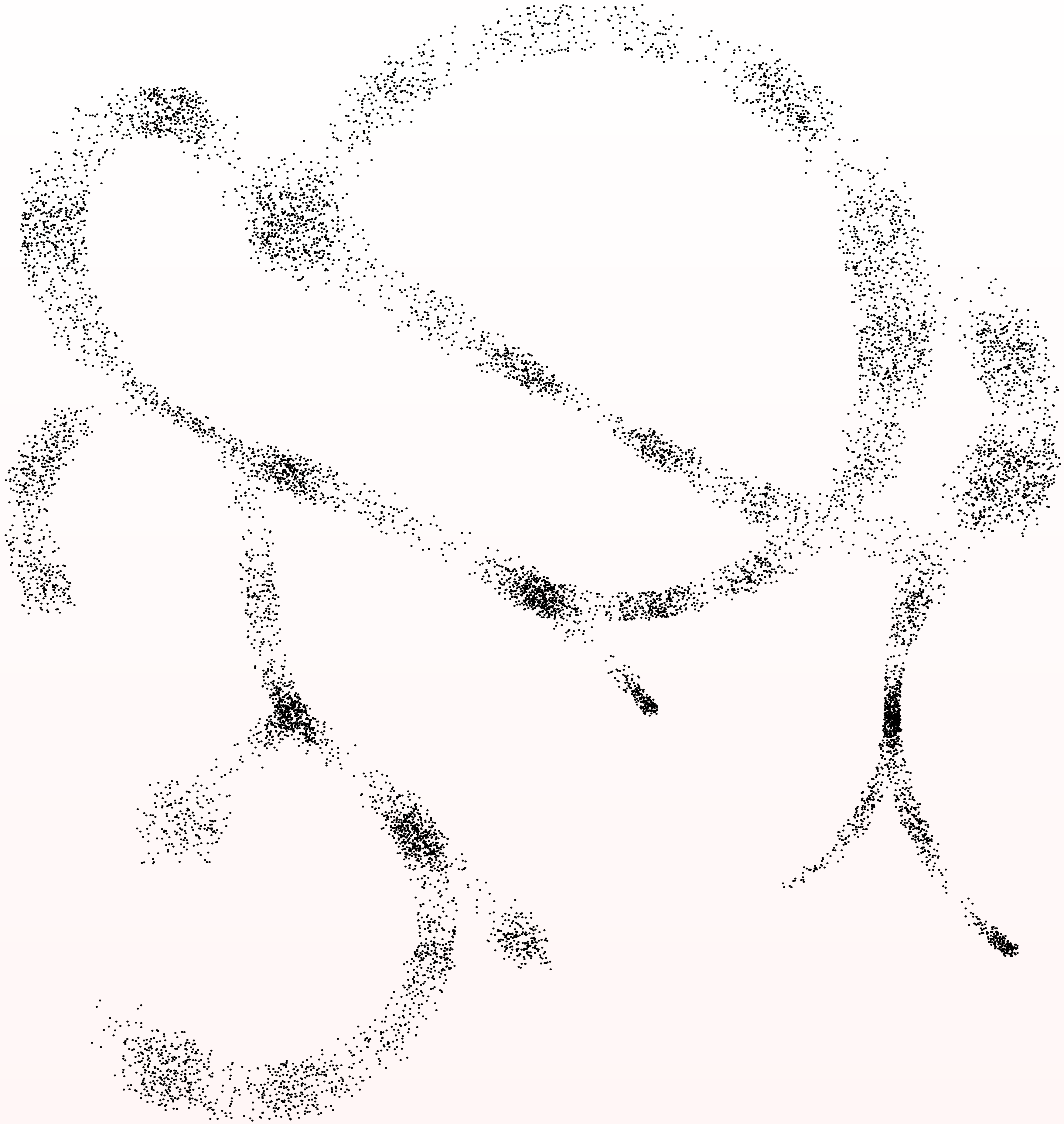


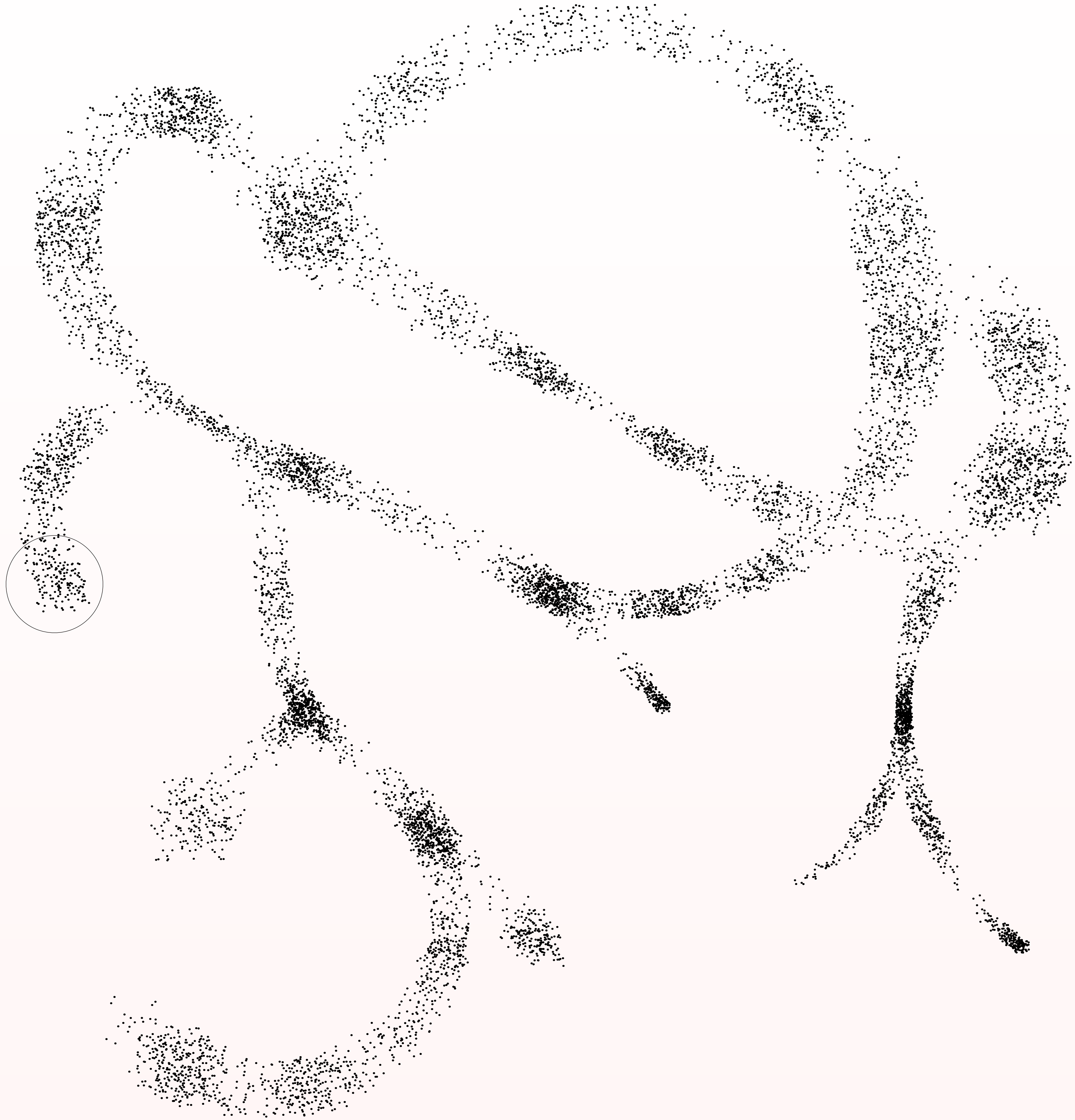
Grid search

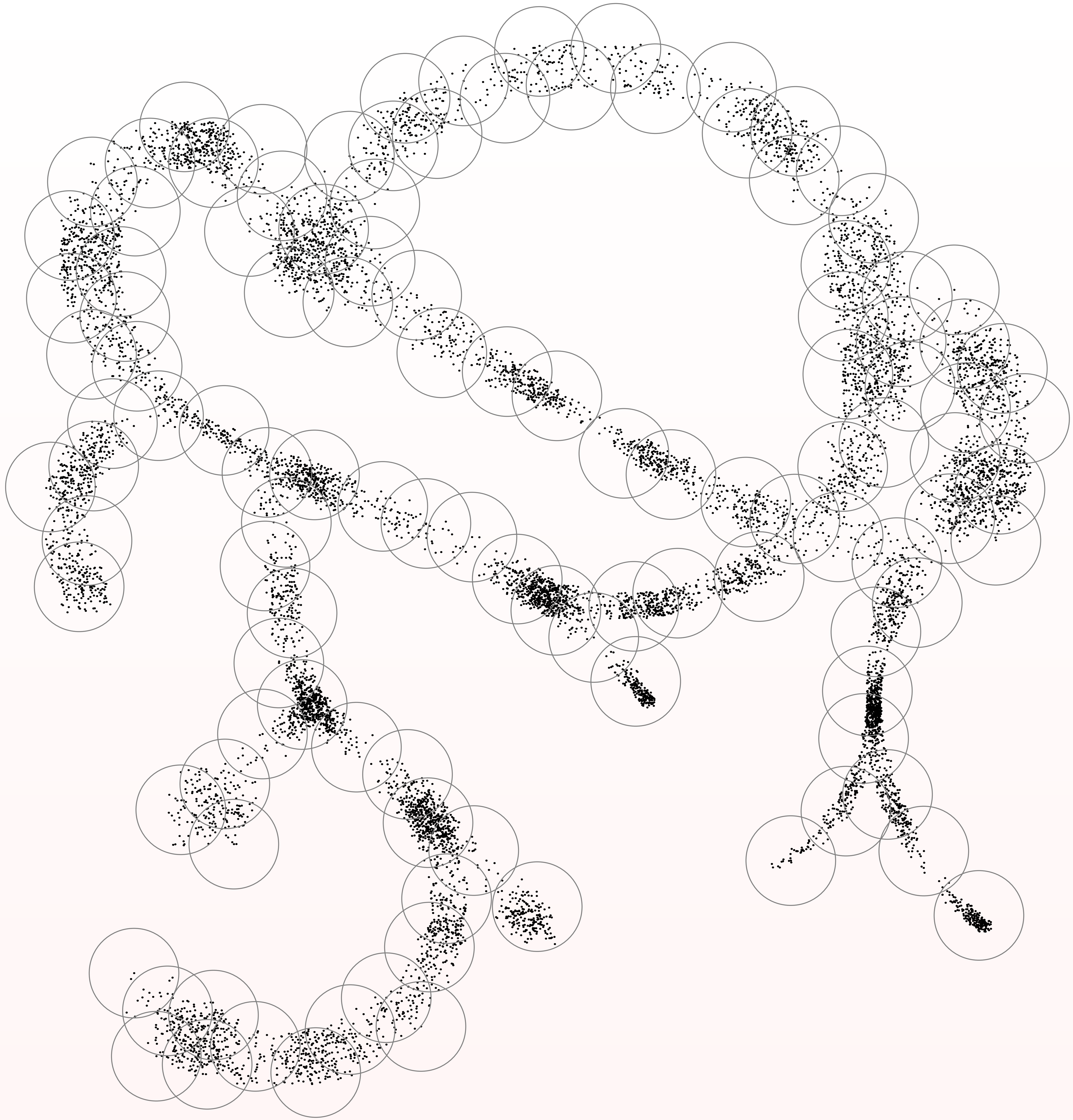


Grid search

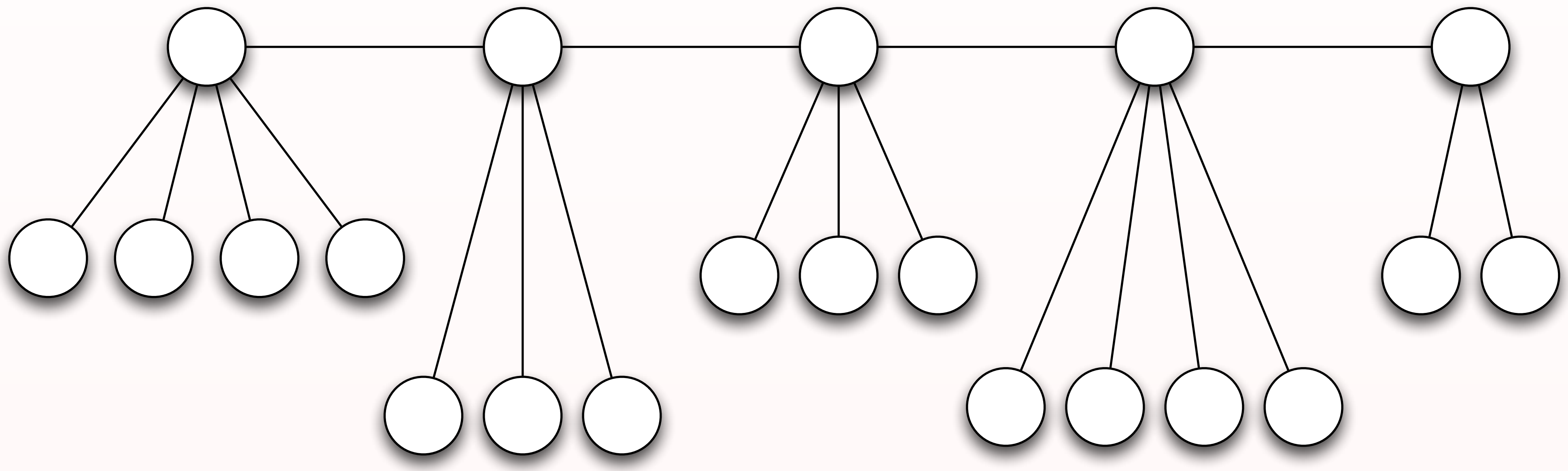




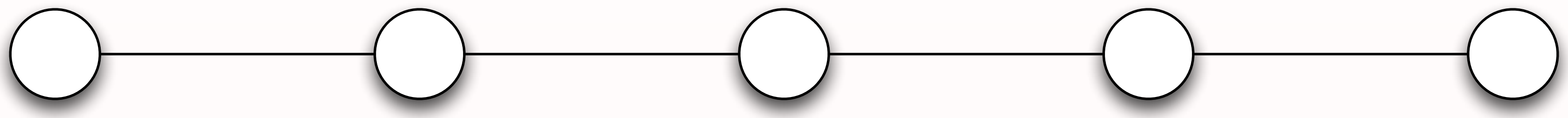




The search tree

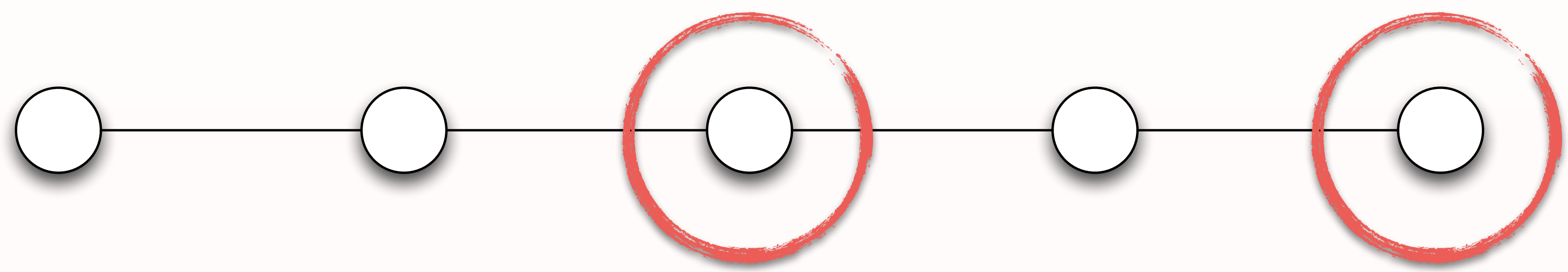


The search tree



coarse search

The search tree

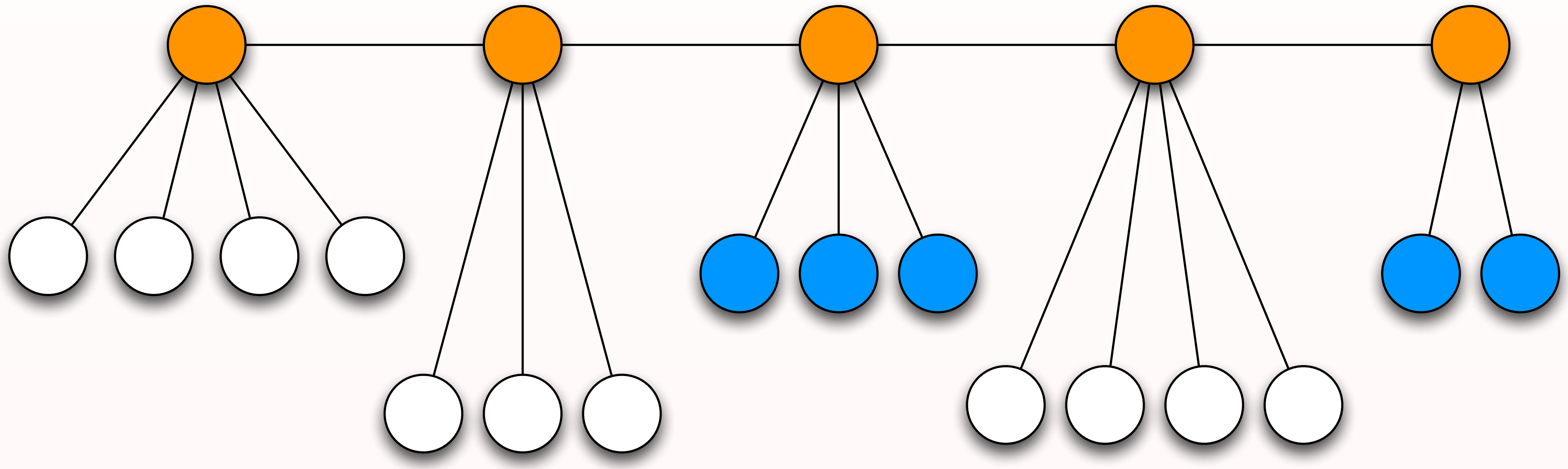


coarse search

The search tree

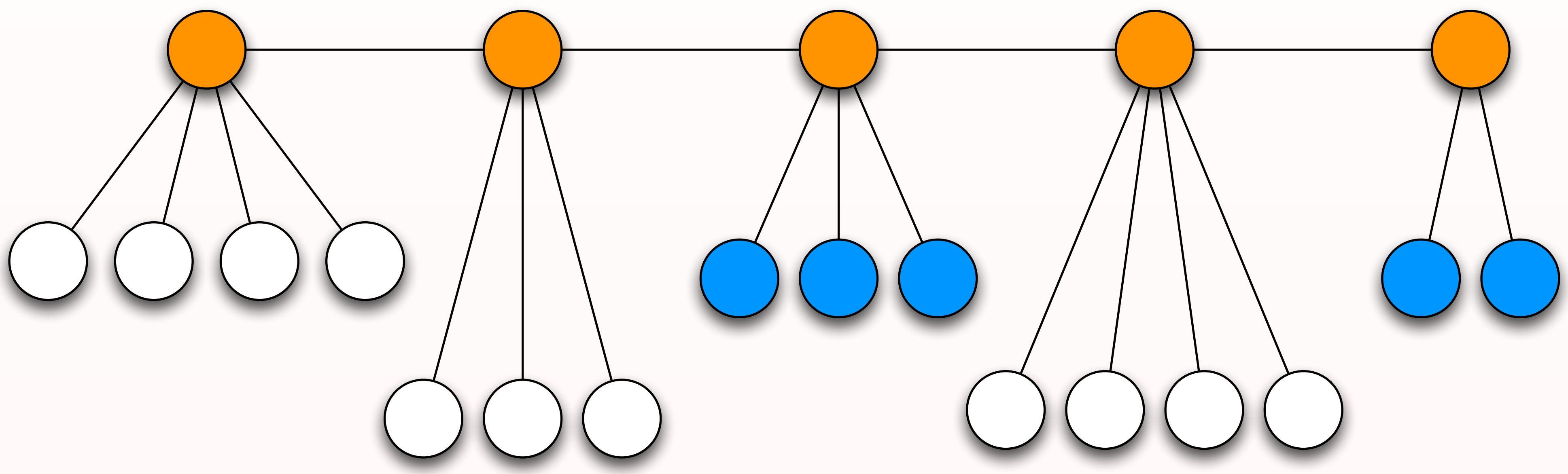
fine search

The search tree



$$O(\textit{coarse} + \textit{fine})$$

The search tree



$$O(\quad k \quad + fine)$$

Metric entropy

Metric entropy

$$N_{r_c}^{ent}(D) :=$$

Metric entropy

$$N_{r_c}^{ent}(D) := \text{largest number of points } x_1, \ldots, x_n \in D$$

Metric entropy

$$N_{r_c}^{ent}(D) := \text{largest number of points } x_1, \ldots, x_n \in D$$

s.t. $||x_i - x_j|| \geq \rho, \forall i \neq j$

Metric entropy

$$N^{ent}_{r_c}(D) := \text{largest number of points } x_1, \ldots, x_n \in D$$

s.t. $||x_i - x_j|| \geq \rho, \forall i \neq j$ for some radius ρ

Metric entropy

$$N_{r_c}^{ent}(D) := \text{largest number of points } x_1, \ldots, x_n \in D$$

$$\text{s.t. } ||x_i - x_j|| \geq \rho, \forall i \neq j \quad \text{for some radius } \rho$$

$$\dim_{\text{Minkowski}}(D) := \lim_{\rho \to 0} \frac{\log N_\rho(D)}{\log 1/\rho}$$

Discrete fractal dimension

Discrete fractal dimension

$$d := \arg\max_{d^*} \{ N_\rho(D) \propto \rho^{d^*} | \rho \in [\rho_1, \rho_2] \}$$

d is the dimension that maximizes the metric entropy

Discrete fractal dimension

$$d := \arg\max_{d^*} \{N_\rho(D) \propto \rho^{d^*} | \rho \in [\rho_1, \rho_2]\}$$

d is the dimension that maximizes the metric entropy

$$\rho = r_c$$

Discrete fractal dimension

$$d := \arg\max_{d^*} \{N_\rho(D) \propto \rho^{d^*} | \rho \in [\rho_1, \rho_2]\}$$

d is the dimension that maximizes the metric entropy

$$\rho = r_c$$

k clusters

Discrete fractal dimension

$$d := \arg\max_{d^*} \{N_\rho(D) \propto \rho^{d^*} | \rho \in [\rho_1, \rho_2]\}$$

d is the dimension that maximizes the metric entropy

$$\rho = r_c$$

k clusters $\qquad k \leq N_{r_c}^{ent}(D)$

Discrete fractal dimension

$$d := \underset{d^*}{\arg\max}\{N_\rho(D) \propto \rho^{d^*} | \rho \in [\rho_1, \rho_2]\}$$

d is the dimension that maximizes the metric entropy

$$\rho = r_c$$

k clusters $\quad k \leq N_{r_c}^{ent}(D)$

$$O\left(k + fine\right)$$

Discrete fractal dimension

$$d := \arg\max_{d^*} \{N_\rho(D) \propto \rho^{d^*} | \rho \in [\rho_1, \rho_2]\}$$

d is the dimension that maximizes the metric entropy

$$\rho = r_c$$

k clusters $\qquad k \leq N_{r_c}^{ent}(D)$

$$O(N_{r_c}^{ent}(D) + fine)$$

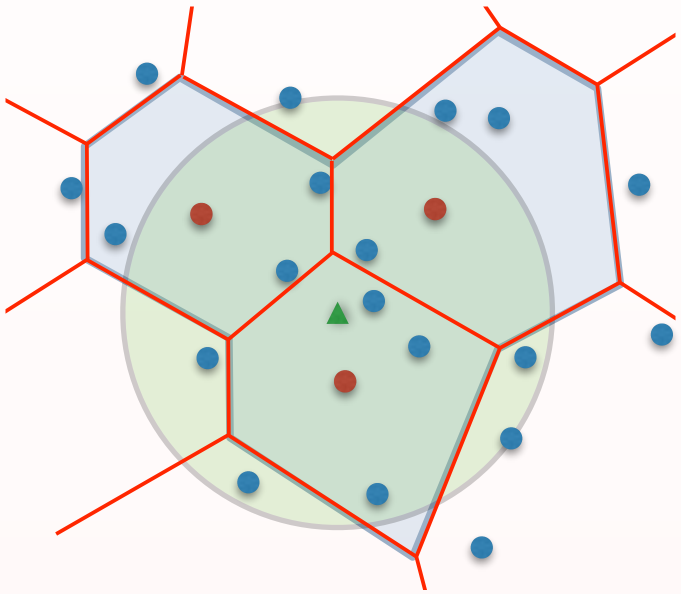
Fine search bounds

$$O(N_{r_c}^{ent}(D) + |F|)$$

Fine search bounds

$O(N_{r_c}^{ent}(D) + |F|)$

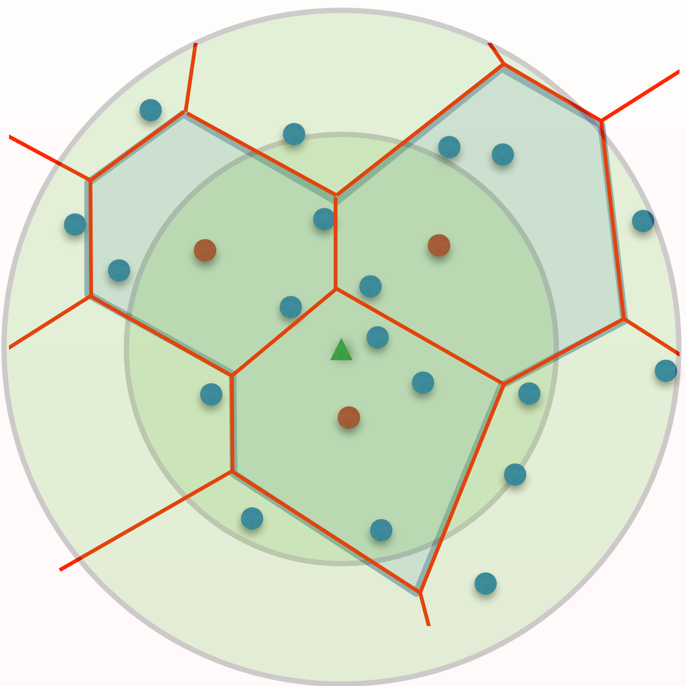
$$F = \bigcup_{c \in B_C(q, r + r_c)} c$$



Fine search bounds

$$O(N_{r_c}^{ent}(D) + |F|)$$

$$F = \bigcup_{c \in B_C(q, r+r_c)} c$$

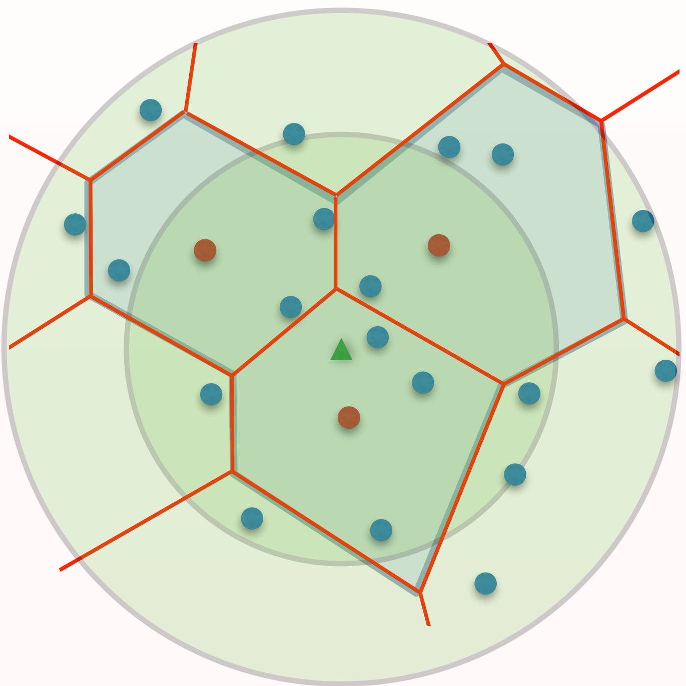


$$F \subset B_D(q, r + 2r_c)$$

Fine search bounds

$O(N_{r_c}^{ent}(D) + |F|)$

$$F = \bigcup_{c \in B_C(q, r + r_c)} c$$

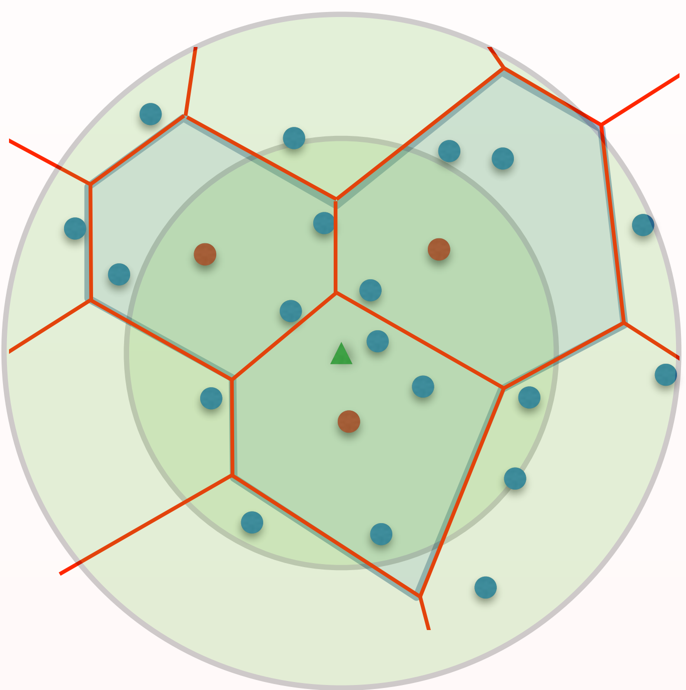


$$F \subset B_D(q, r + 2r_c)$$

$$|F| \leq |B_D(q, r + 2r_c)|$$

Fine search bounds

$$O(N_{r_c}^{ent}(D) + |F|) \qquad F = \bigcup_{c \in B_C(q, r+r_c)} c$$

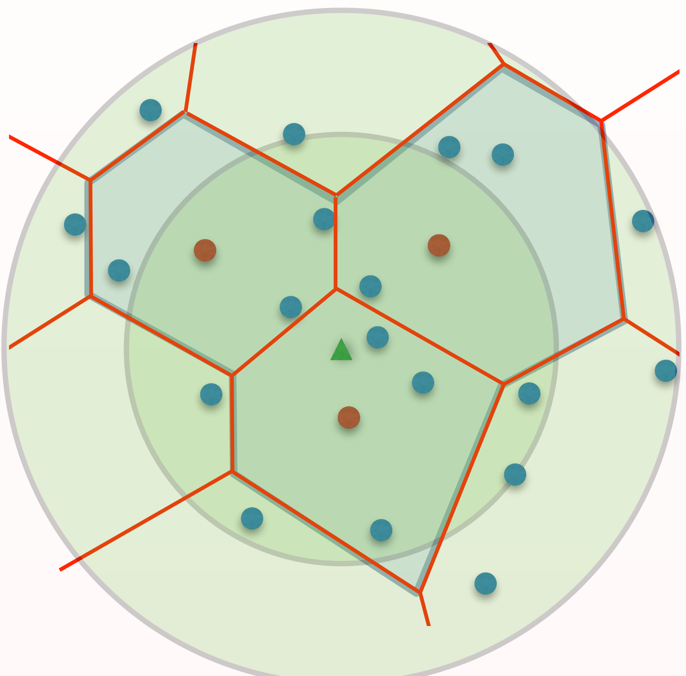


$$F \subset B_D(q, r + 2r_c)$$

$$|F| \leq |B_D(q, r + 2r_c)| \sim |B_D(q, r)| \left(\frac{r + 2r_c}{r} \right)^d$$

Fine search bounds

$$O(N^{ent}_{r_c}(D) + |F|) \qquad F = \bigcup_{c \in B_C(q, r+r_c)} c$$



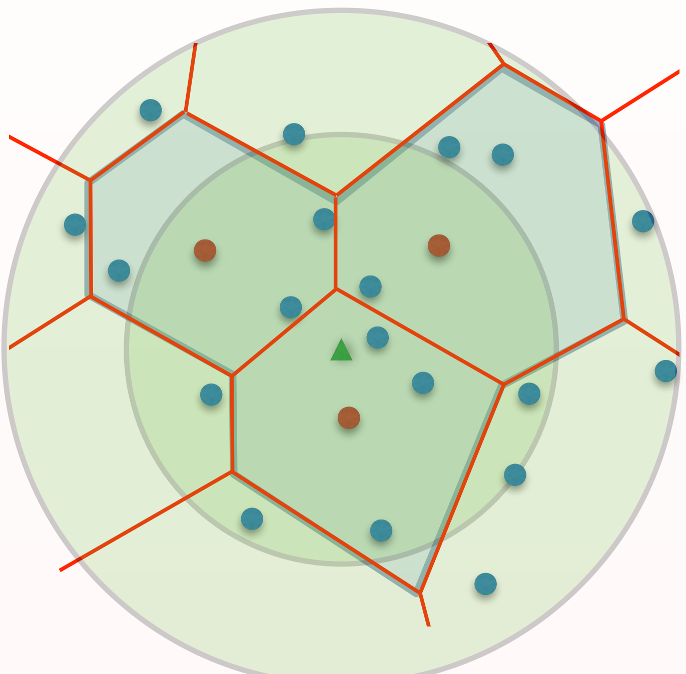
$$F \subset B_D(q, r + 2r_c)$$

$$|F| \leq |B_D(q, r + 2r_c)| \sim |B_D(q, r)| \left(\frac{r + 2r_c}{r}\right)^d$$

Doubling search radius increases hits by at most 2^d

Fine search bounds

$$O(N_{r_c}^{ent}(D) + |F|) \qquad F = \bigcup_{c \in B_C(q, r+r_c)} c$$



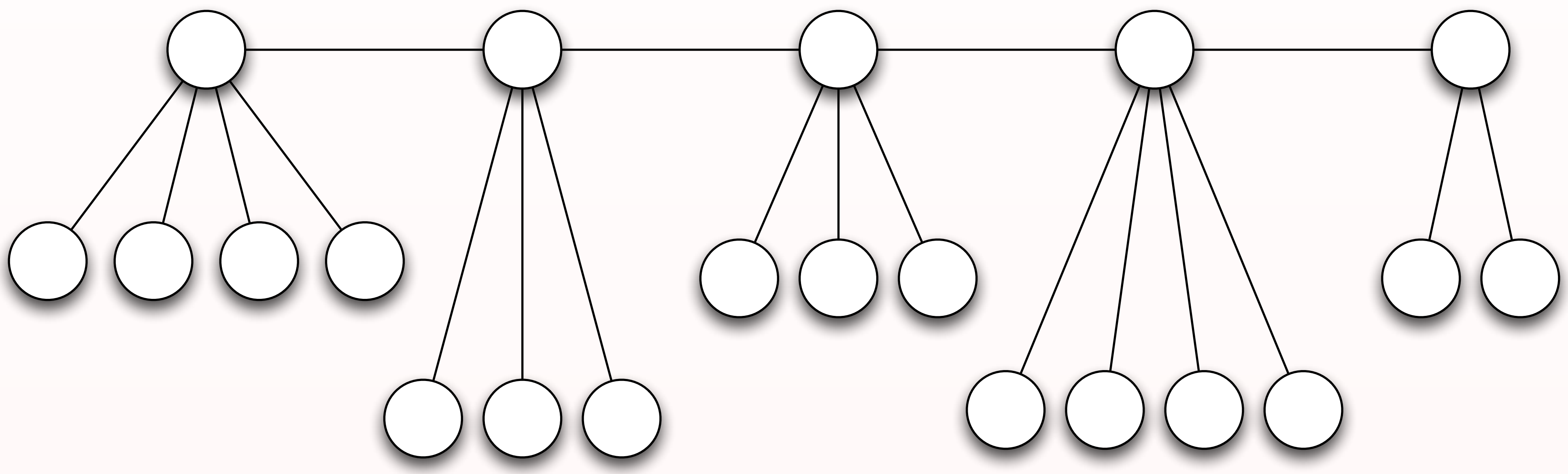
$$F \subset B_D(q, r + 2r_c)$$

$$|F| \leq |B_D(q, r + 2r_c)| \sim |B_D(q, r)| \left(\frac{r + 2r_c}{r} \right)^d$$

Doubling search radius increases hits by at most 2^d

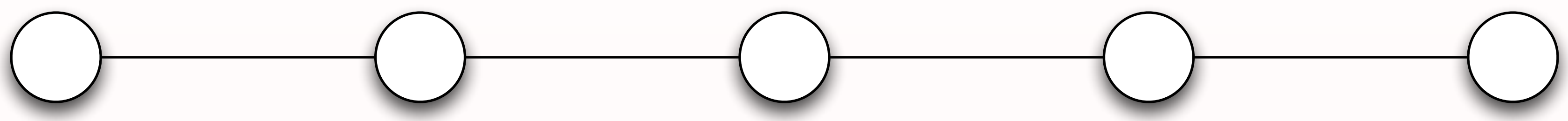
$$O\left(N_{r_c}^{ent}(D) + |B_D(q, r)| \left(\frac{r + 2r_c}{r} \right)^d \right)$$

Entropy-scaling data structure



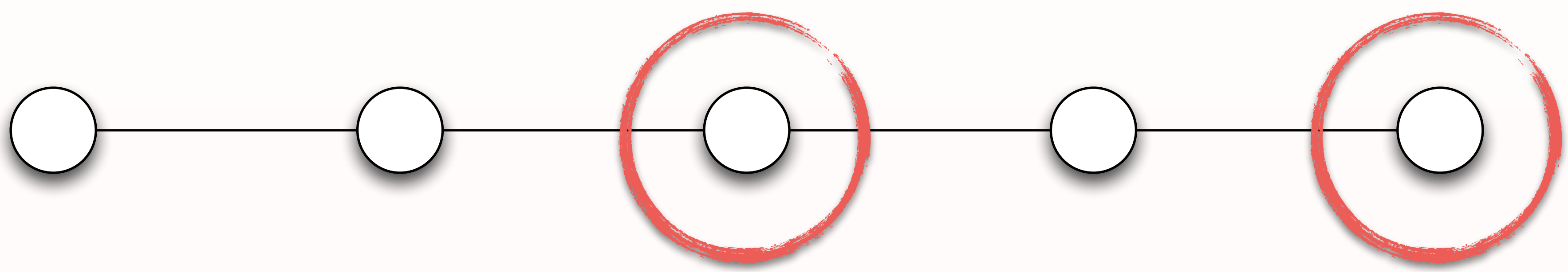
$$O\left(N_{r_c}^{ent}(D) + |B_D(q,r)| \left(\frac{r+2r_c}{r}\right)^d\right)$$

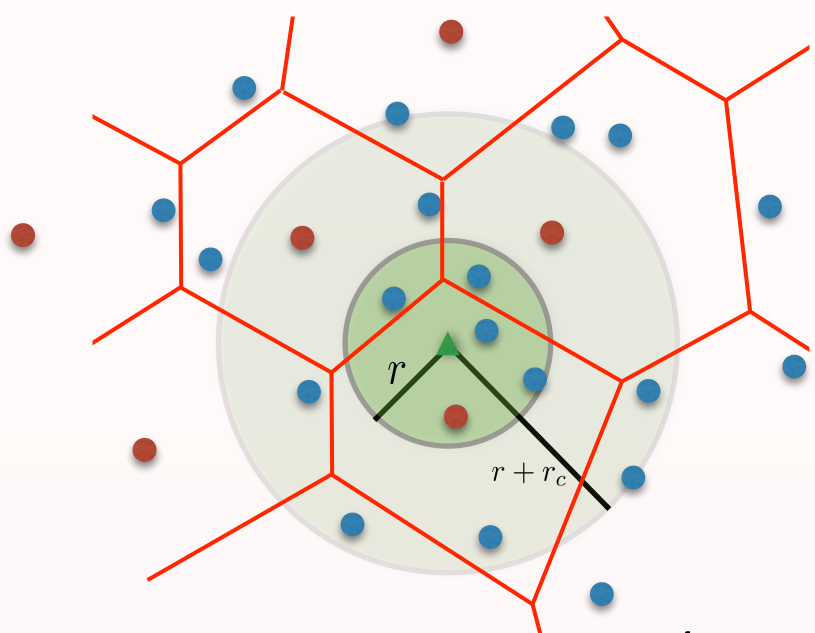
Entropy-scaling data structure



$$O\left(N_{r_c}^{ent}(D) + |B_D(q,r)|\left(\frac{r+2r_c}{r}\right)^d\right)$$

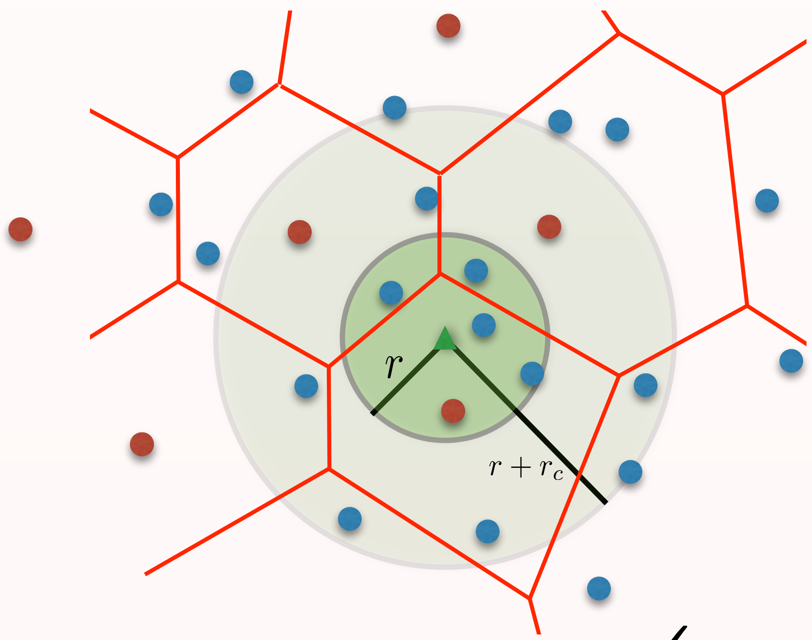
Entropy-scaling data structure

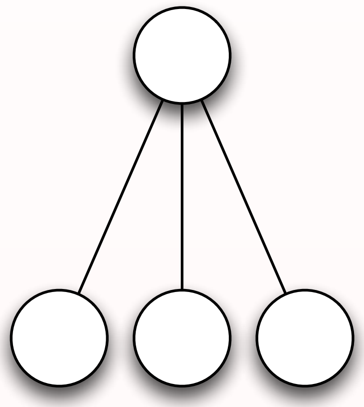


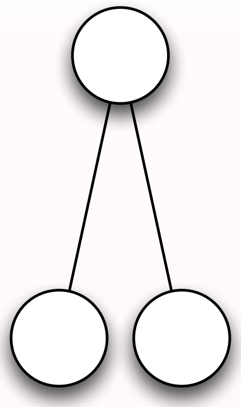


$$O\left(N^{ent}_{r_c}(D) + |B_D(q,r)| \left(\frac{r+2r_c}{r}\right)^d\right)$$

Entropy-scaling data structure

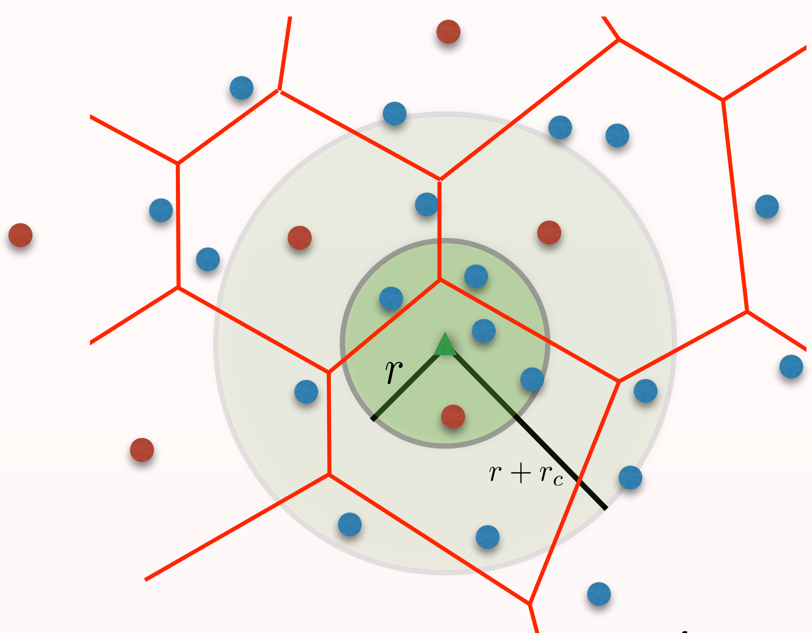


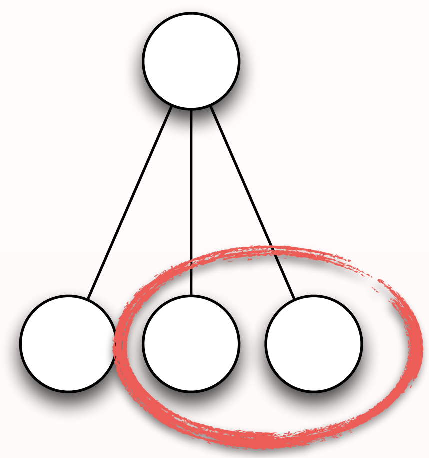


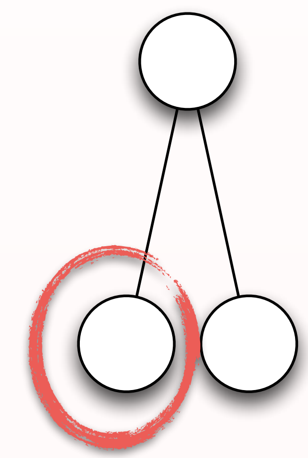


$$O\left(N_{r_c}^{ent}(D) + |B_D(q,r)|\left(\frac{r+2r_c}{r}\right)^d\right)$$

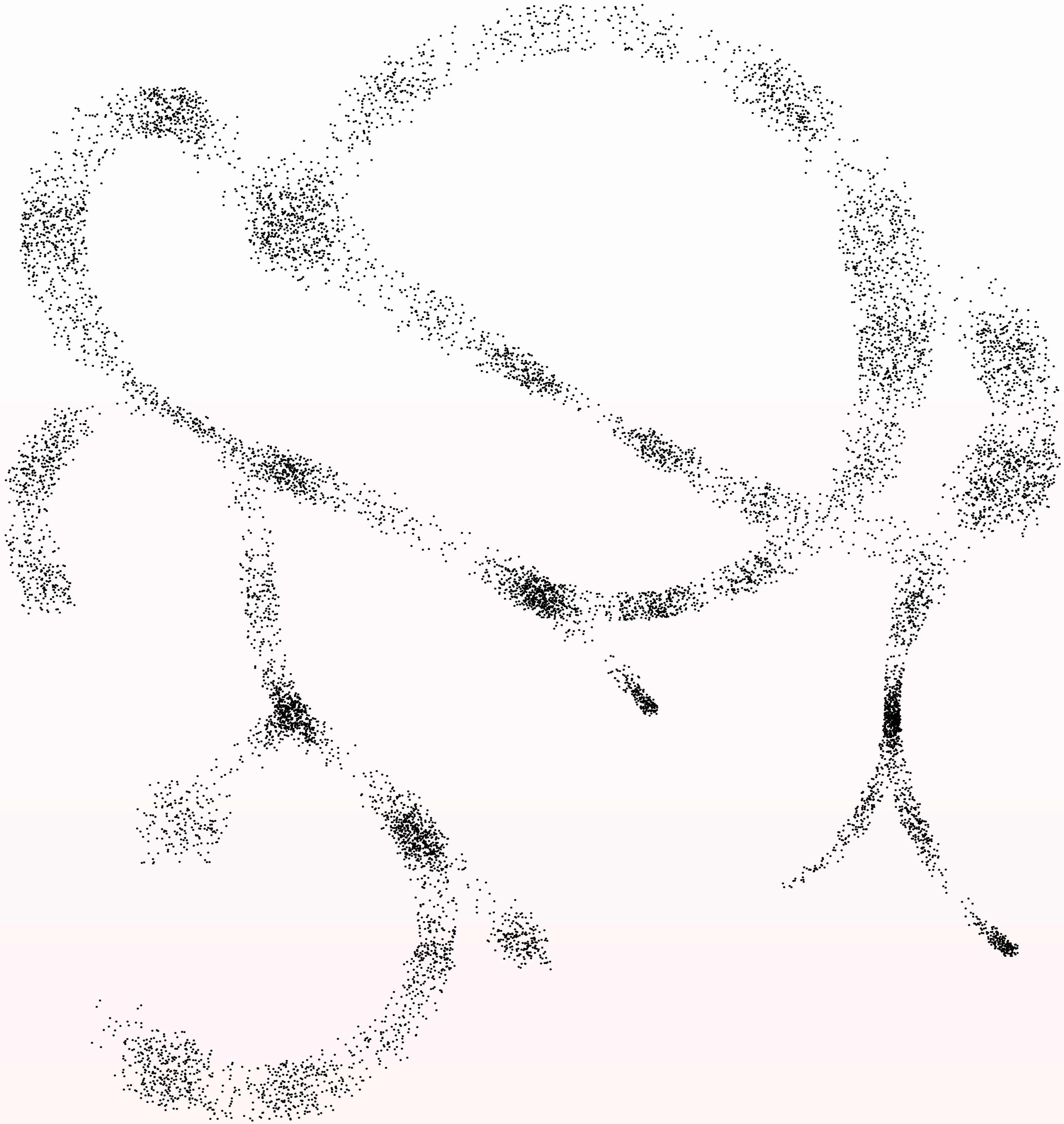
Entropy-scaling data structure

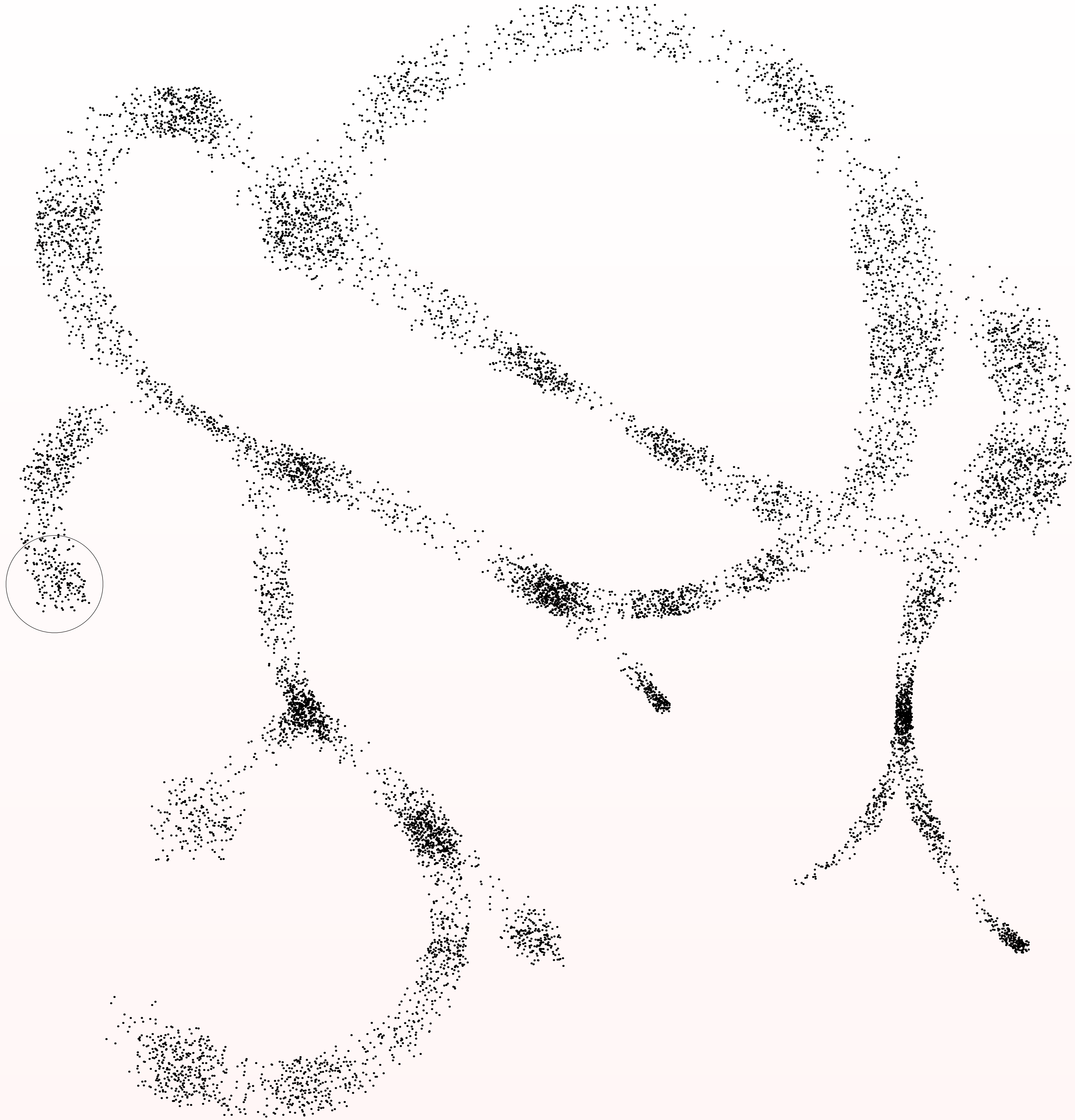


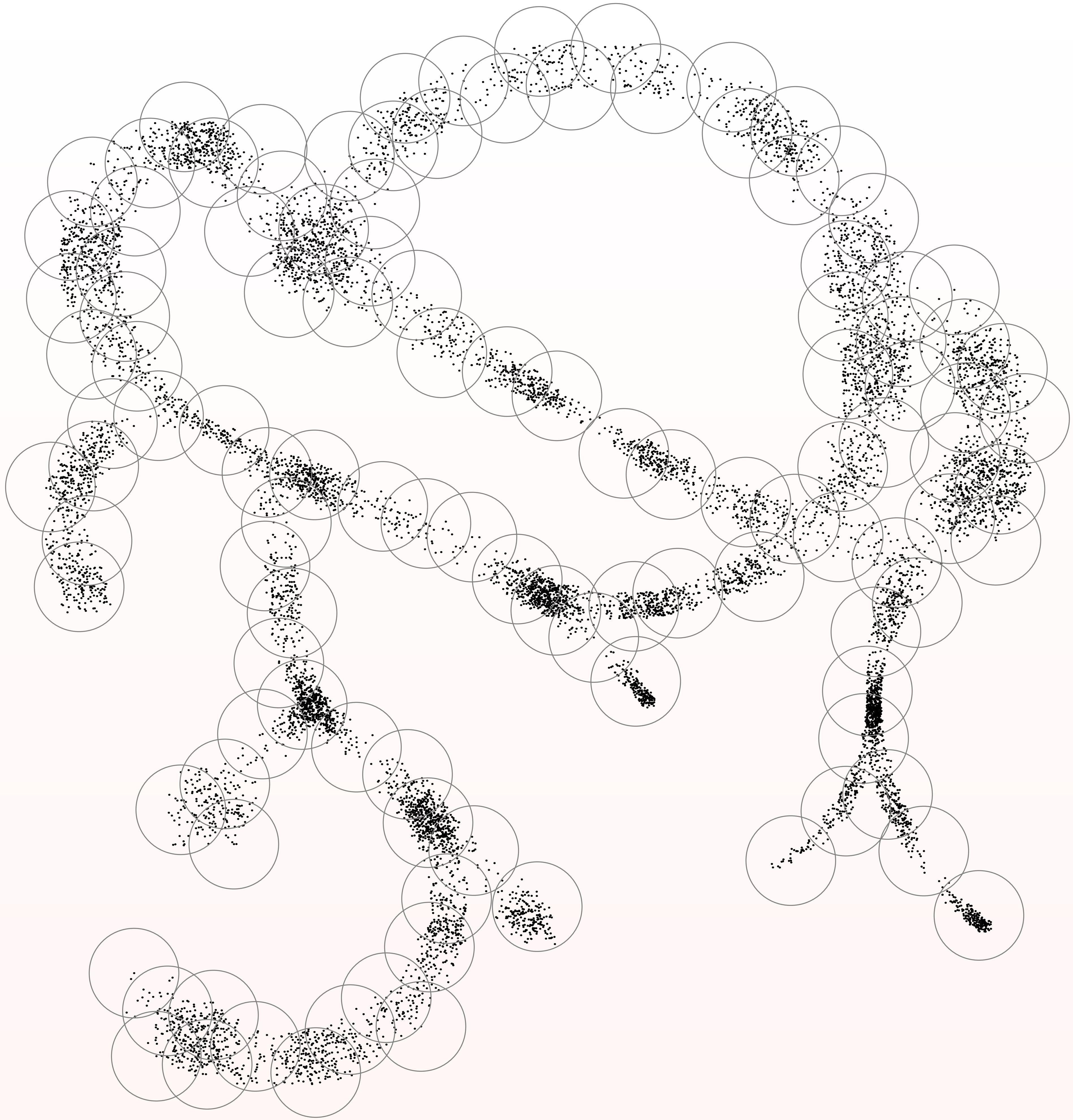


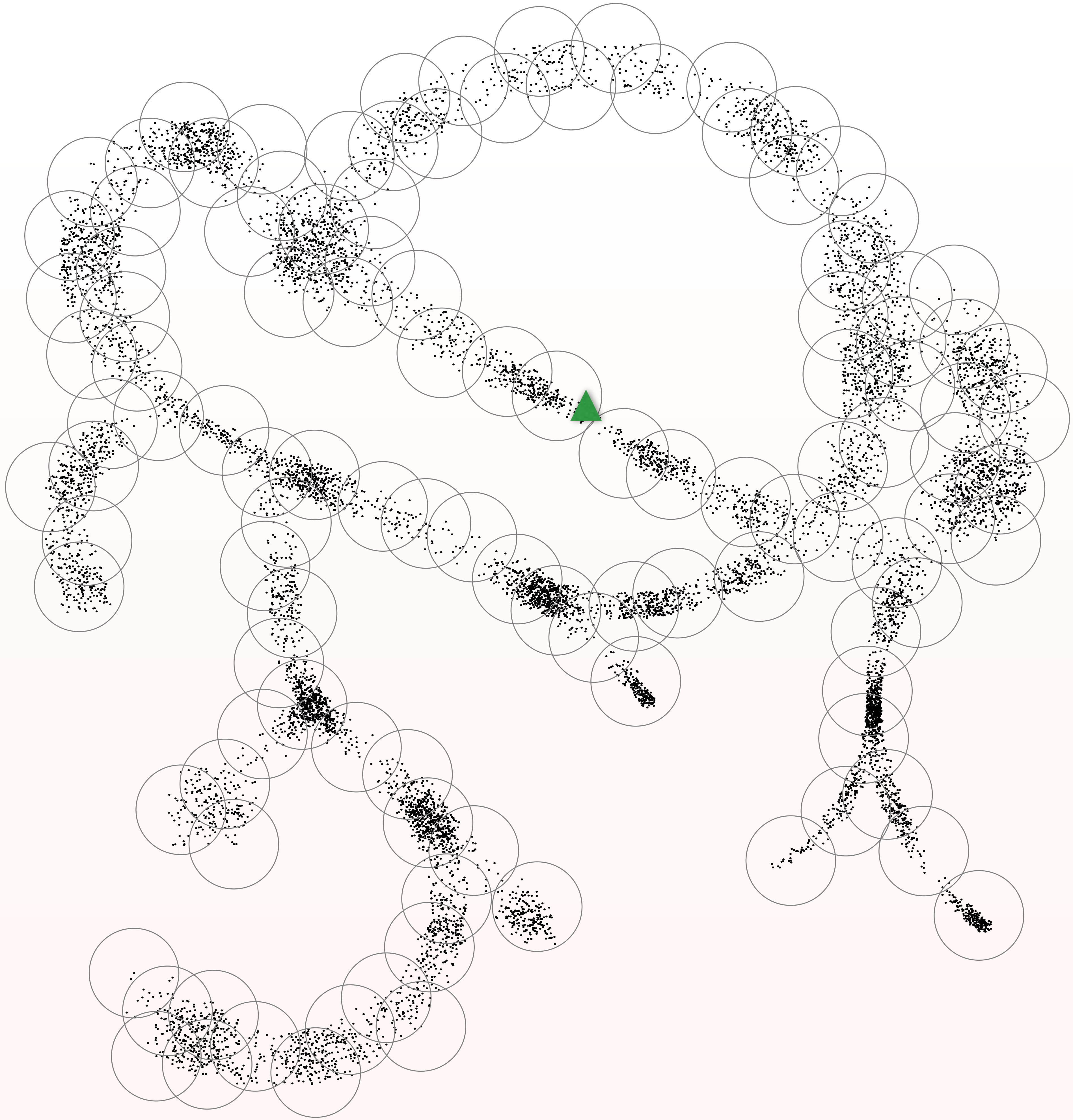


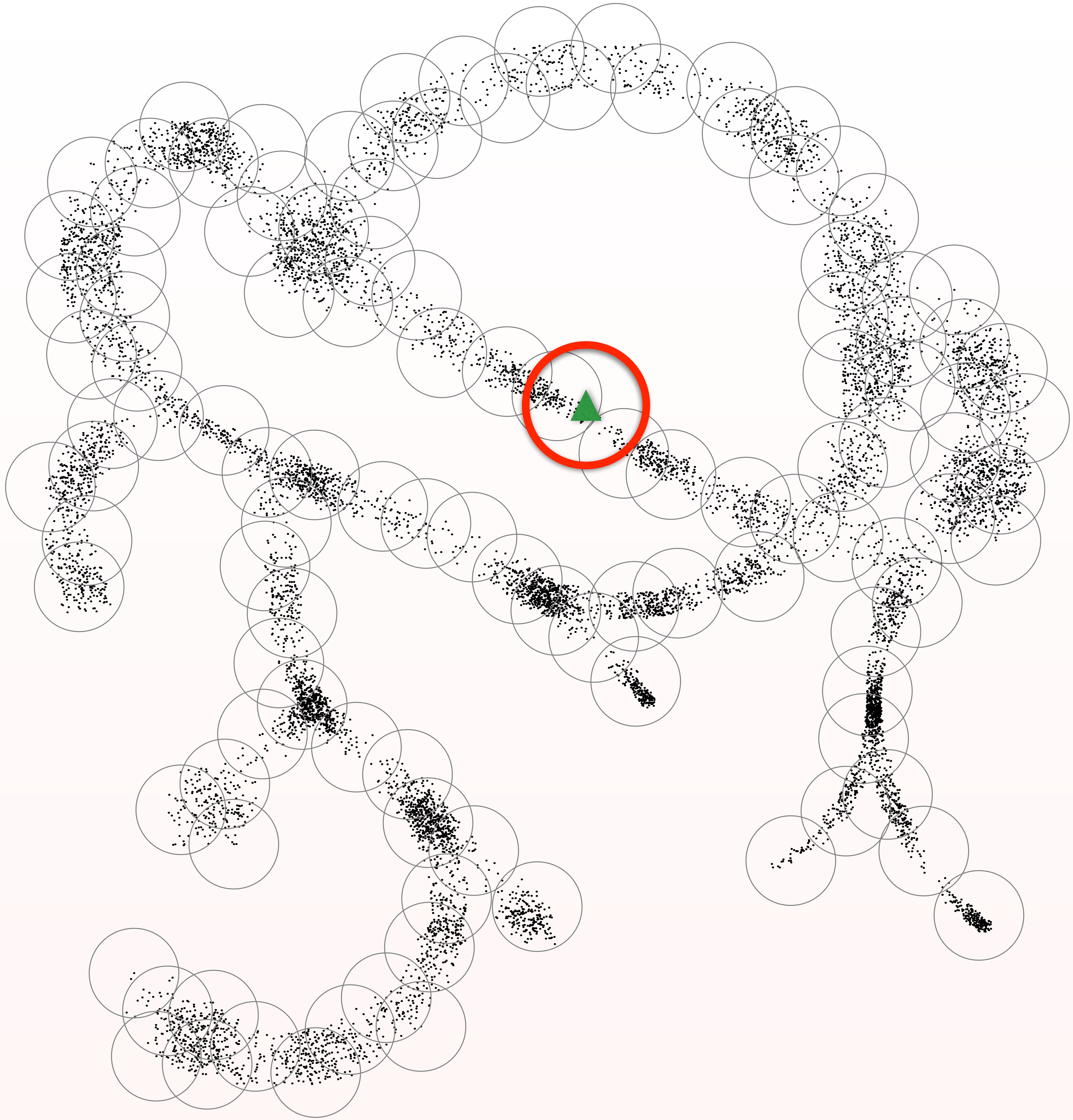
$$O\left(N^{ent}_{r_c}(D) + |B_D(q,r)|\left(\frac{r+2r_c}{r}\right)^d\right)$$

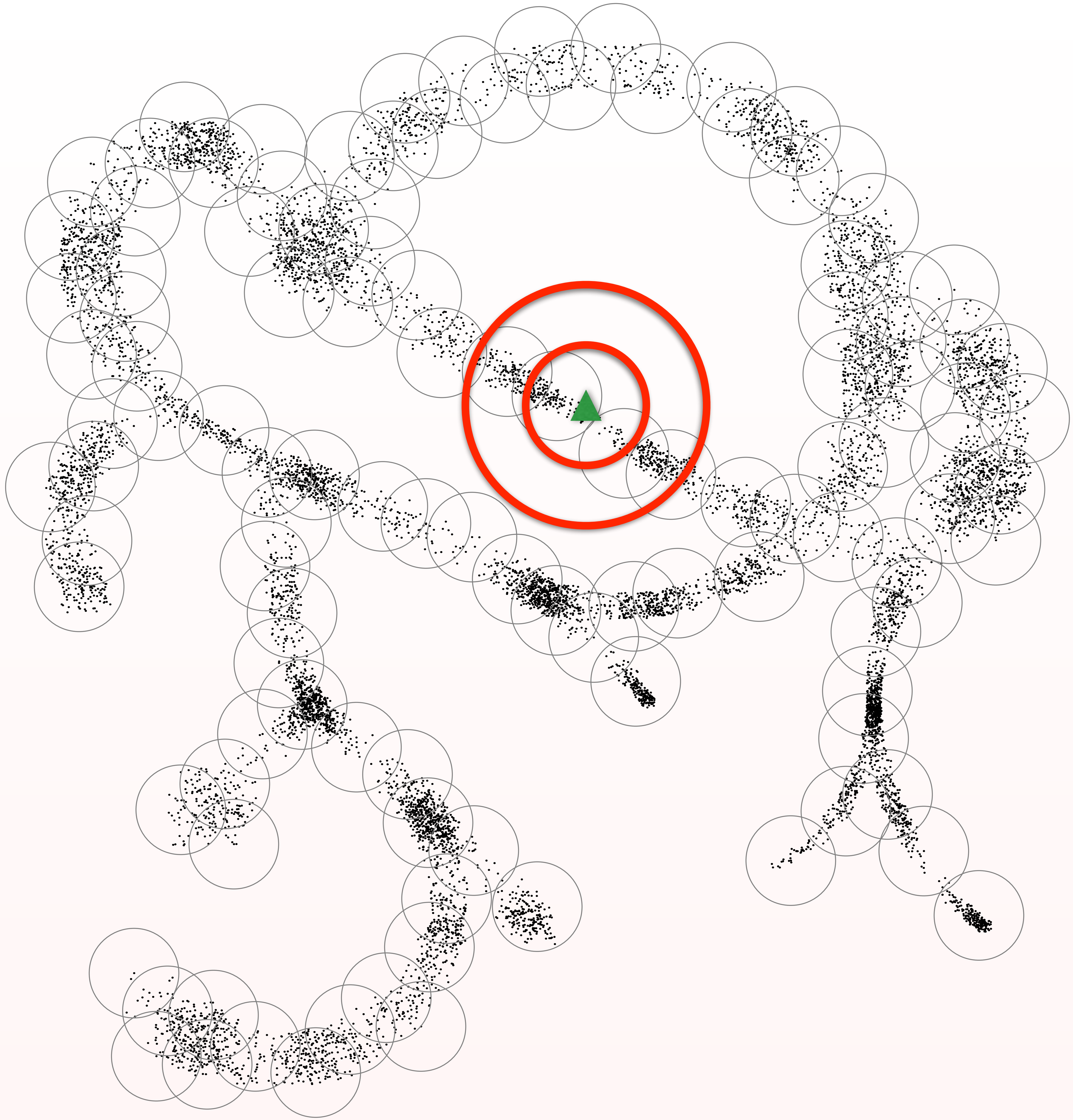


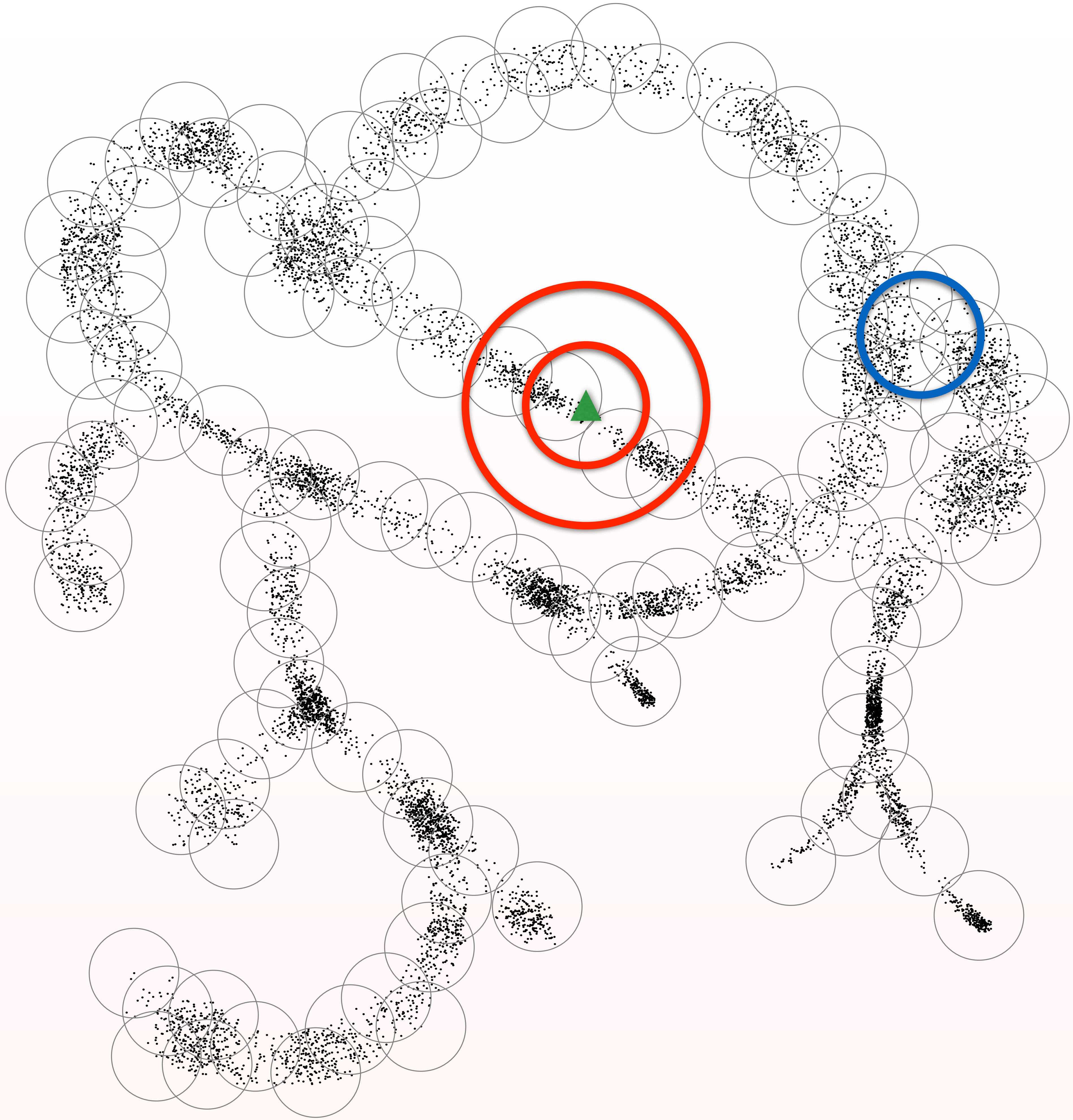


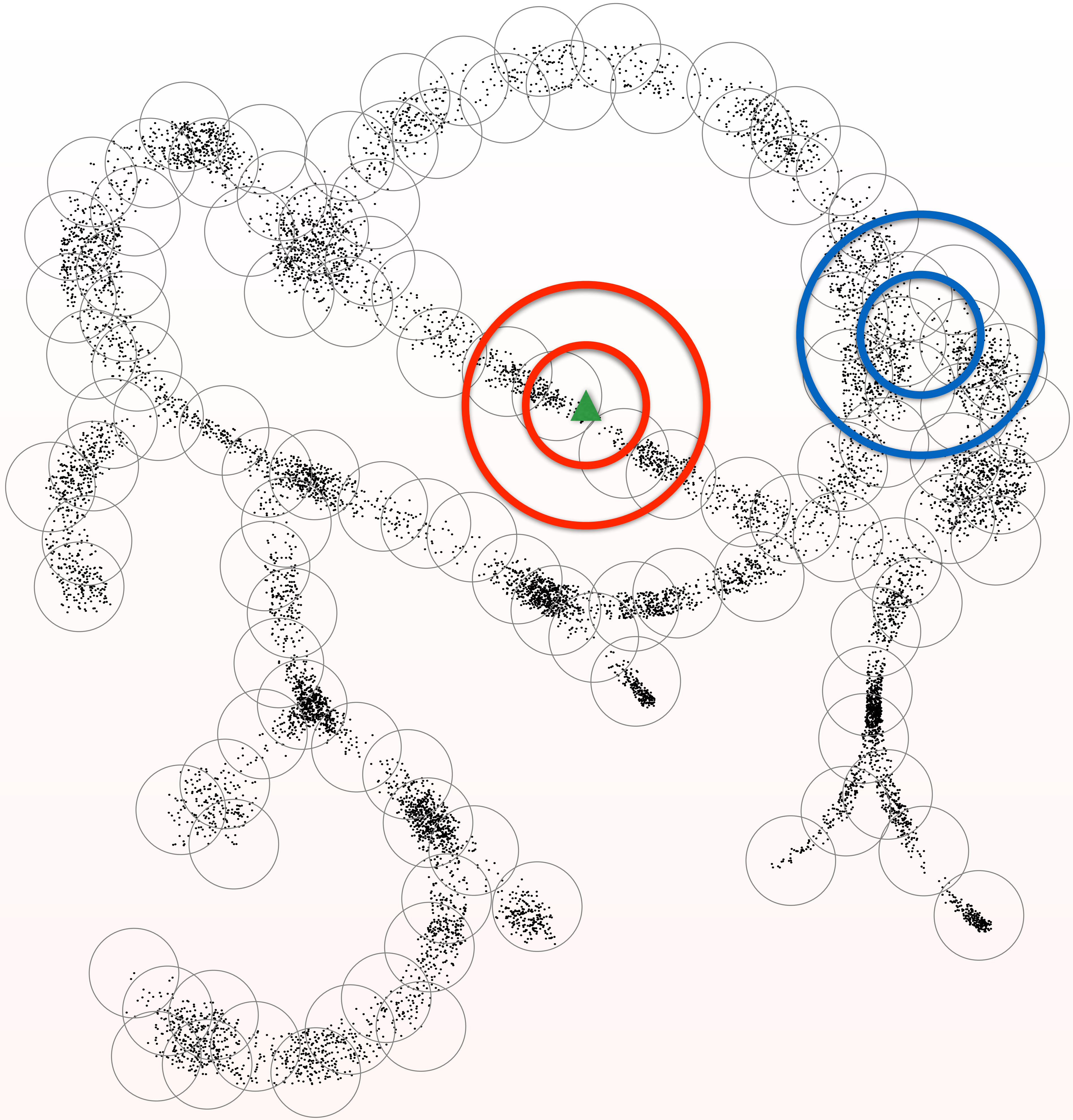












Summary

Summary

Lossless compression bounded by Shannon

Summary

Lossless compression bounded by Shannon

Lossy compression can go further

Summary

Lossless compression bounded by Shannon

Lossy compression can go further

Clever preprocessing of reads can boost standard compressors

Summary

Lossless compression bounded by Shannon

Lossy compression can go further

Clever preprocessing of reads can boost standard compressors

All comes down to *structure of the data:*

Summary

Lossless compression bounded by Shannon

Lossy compression can go further

Clever preprocessing of reads can boost standard compressors

All comes down to *structure of the data:*

minimize bits/base

Summary

Lossless compression bounded by Shannon

Lossy compression can go further

Clever preprocessing of reads can boost standard compressors

All comes down to *structure of the data:*

minimize bits/base

identify common substrings

Summary

Lossless compression bounded by Shannon

Lossy compression can go further

Clever preprocessing of reads can boost standard compressors

All comes down to *structure of the data:*

minimize bits/base

identify common substrings

find correlations among quality scores

Going further

Going further

Many more challenges:

Going further

Many more challenges:

New sequencing technologies

Going further

Many more challenges:

New sequencing technologies

longer, more error-prone reads

Going further

Many more challenges:

New sequencing technologies

longer, more error-prone reads

different error models

Going further

Many more challenges:

New sequencing technologies

longer, more error-prone reads

different error models

Can we get the benefits of compressive acceleration with a more general model