Approximating Sequence Similarity

Brad Solomon

March 27, 2019

Lecture 16: Scalable Methods for Genomics

Approximating Sequence Similarity

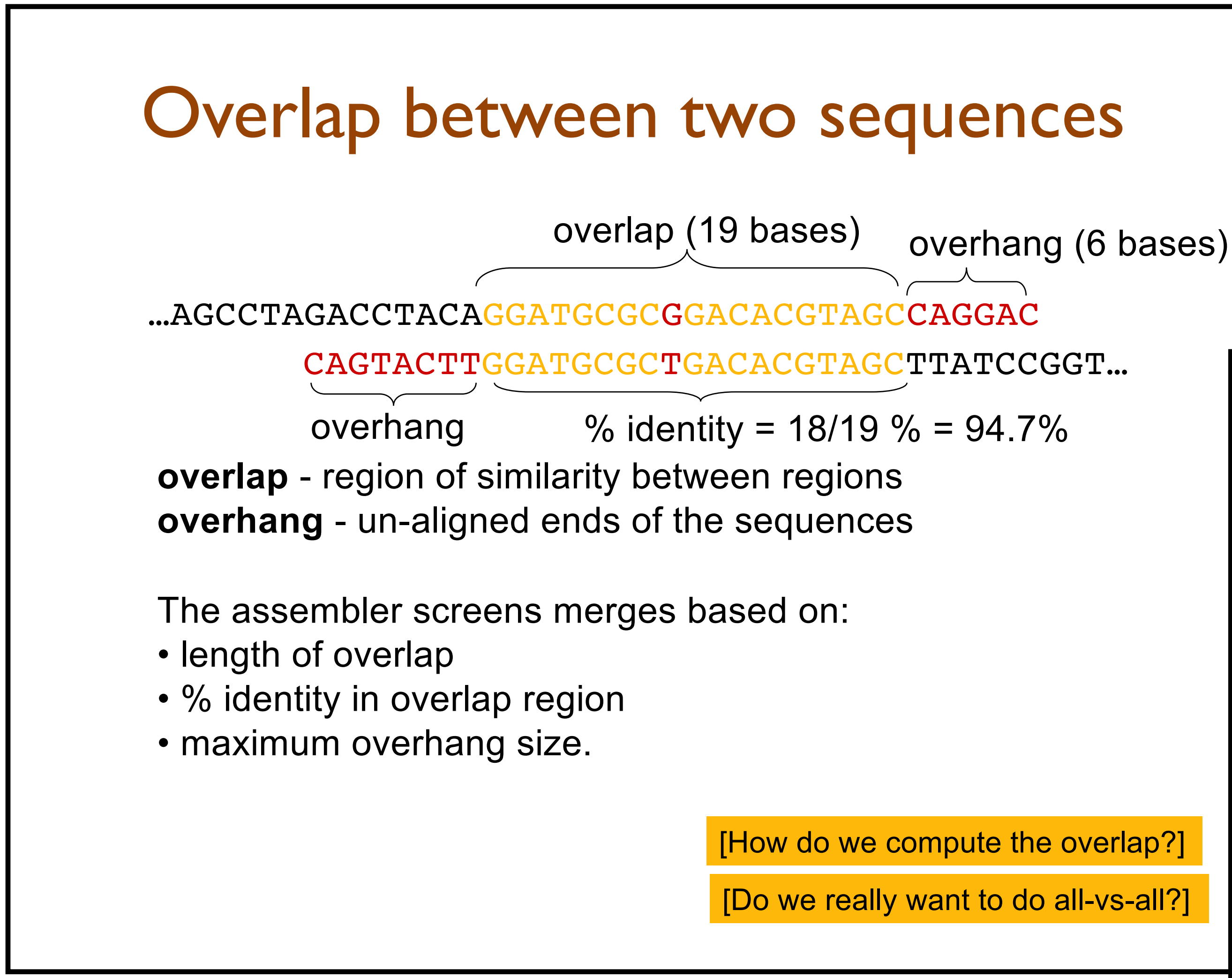
With Special Focus on RNA-Seq

Brad Solomon

March 27, 2019

Lecture 16: Scalable Methods for Genomics

A Quick Recap



Overlap between two sequences

```
                  overlap (19 bases)     overhang (6 bases)
...AGCCTAGACCTACAGGATGCGCGGACACGTAGCCAGGAC
        CAGTACTTGGATGCGCTGACACGTAGCTTATCCGGT...
         overhang        % identity = 18/19 % = 94.7%
```

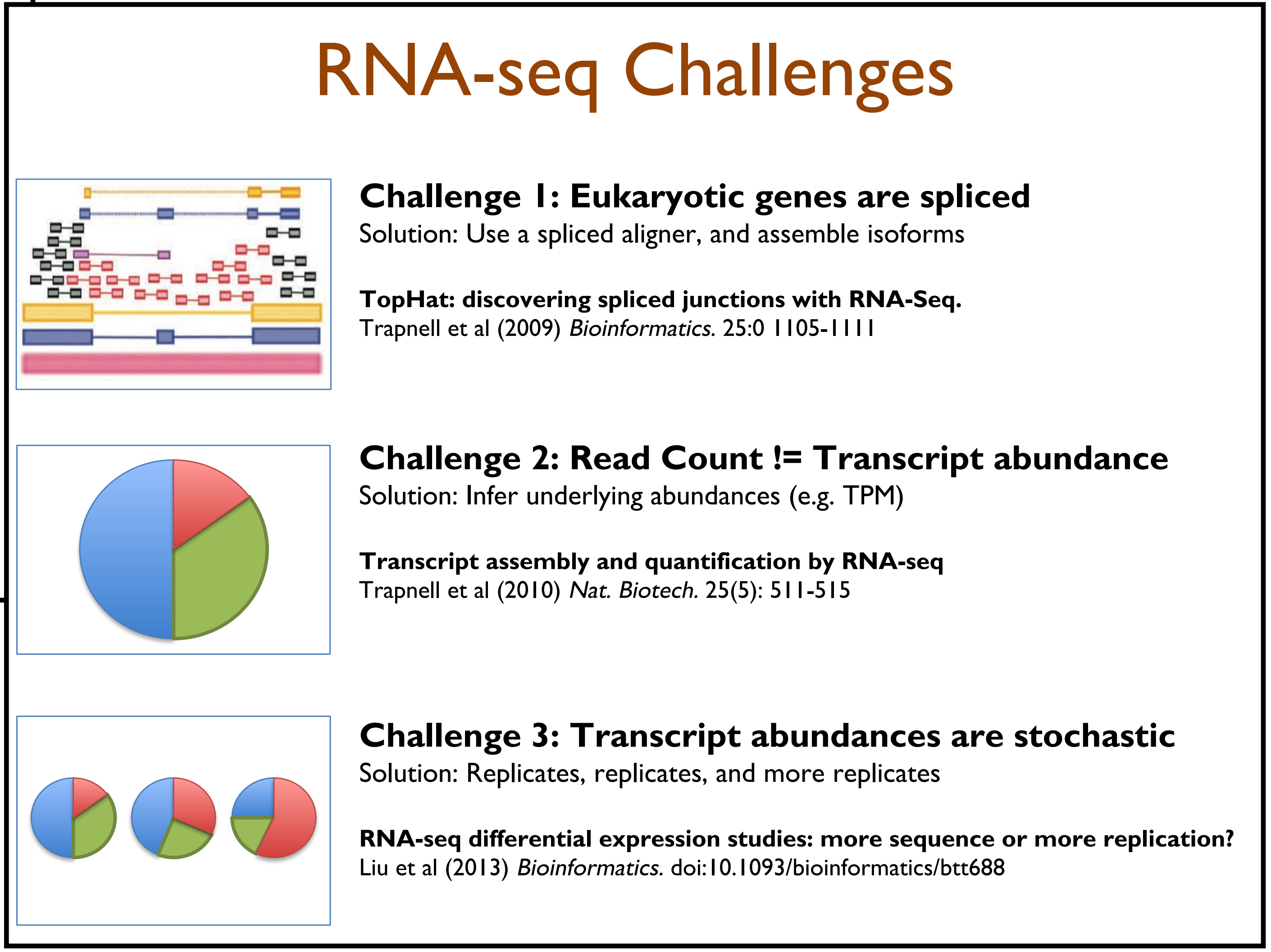
overlap - region of similarity between regions
overhang - un-aligned ends of the sequences

The assembler screens merges based on:
- length of overlap
- % identity in overlap region
- maximum overhang size.

[How do we compute the overlap?]

[Do we really want to do all-vs-all?]

See Lecture 4 Assembly & Whole Genome Alignment



RNA-seq Challenges

Challenge 1: Eukaryotic genes are spliced
Solution: Use a spliced aligner, and assemble isoforms

TopHat: discovering spliced junctions with RNA-Seq.
Trapnell et al (2009) *Bioinformatics.* 25:0 1105-1111

Challenge 2: Read Count != Transcript abundance
Solution: Infer underlying abundances (e.g. TPM)

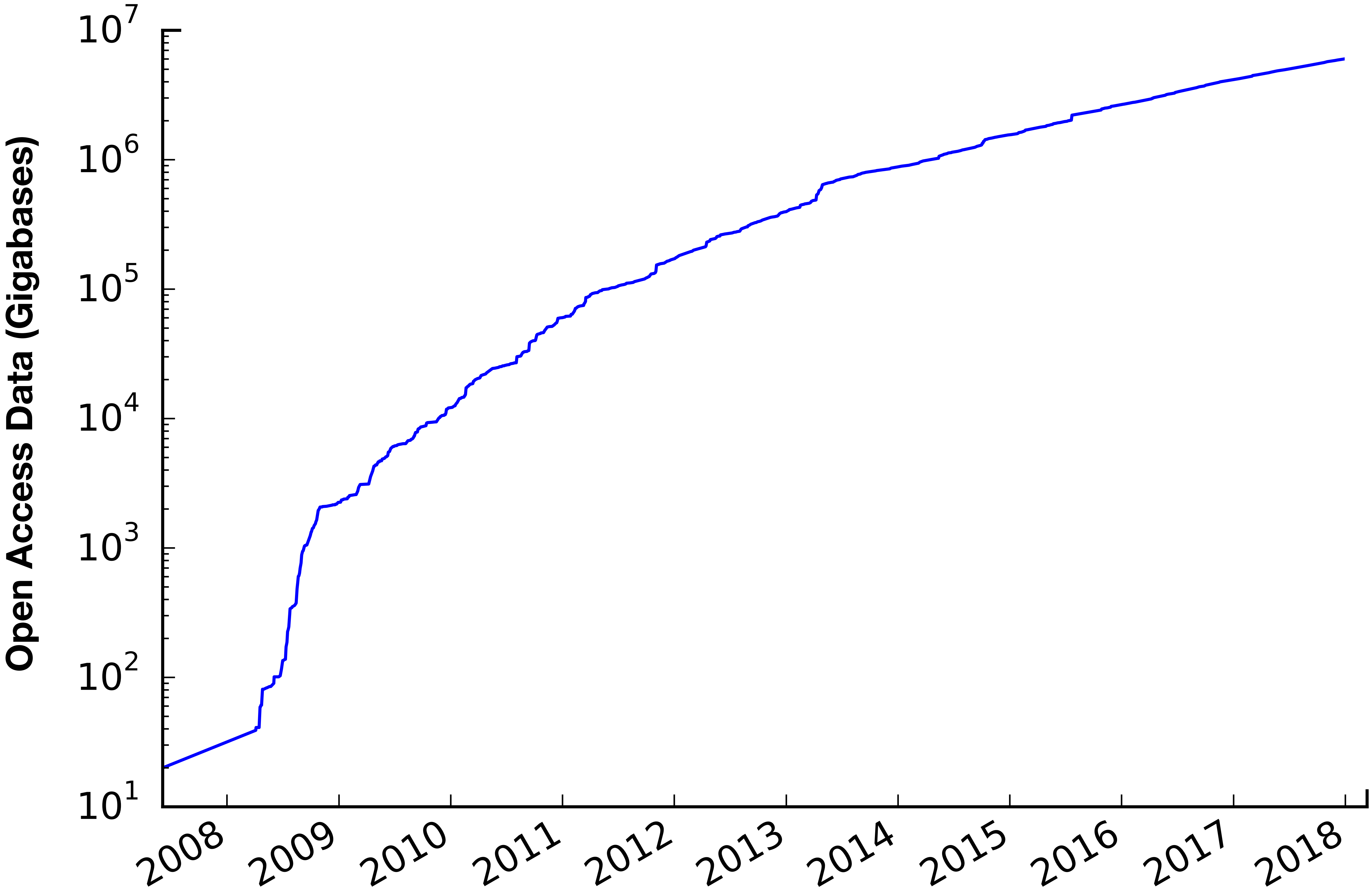
Transcript assembly and quantification by RNA-seq
Trapnell et al (2010) *Nat. Biotech.* 25(5): 511-515

Challenge 3: Transcript abundances are stochastic
Solution: Replicates, replicates, and more replicates

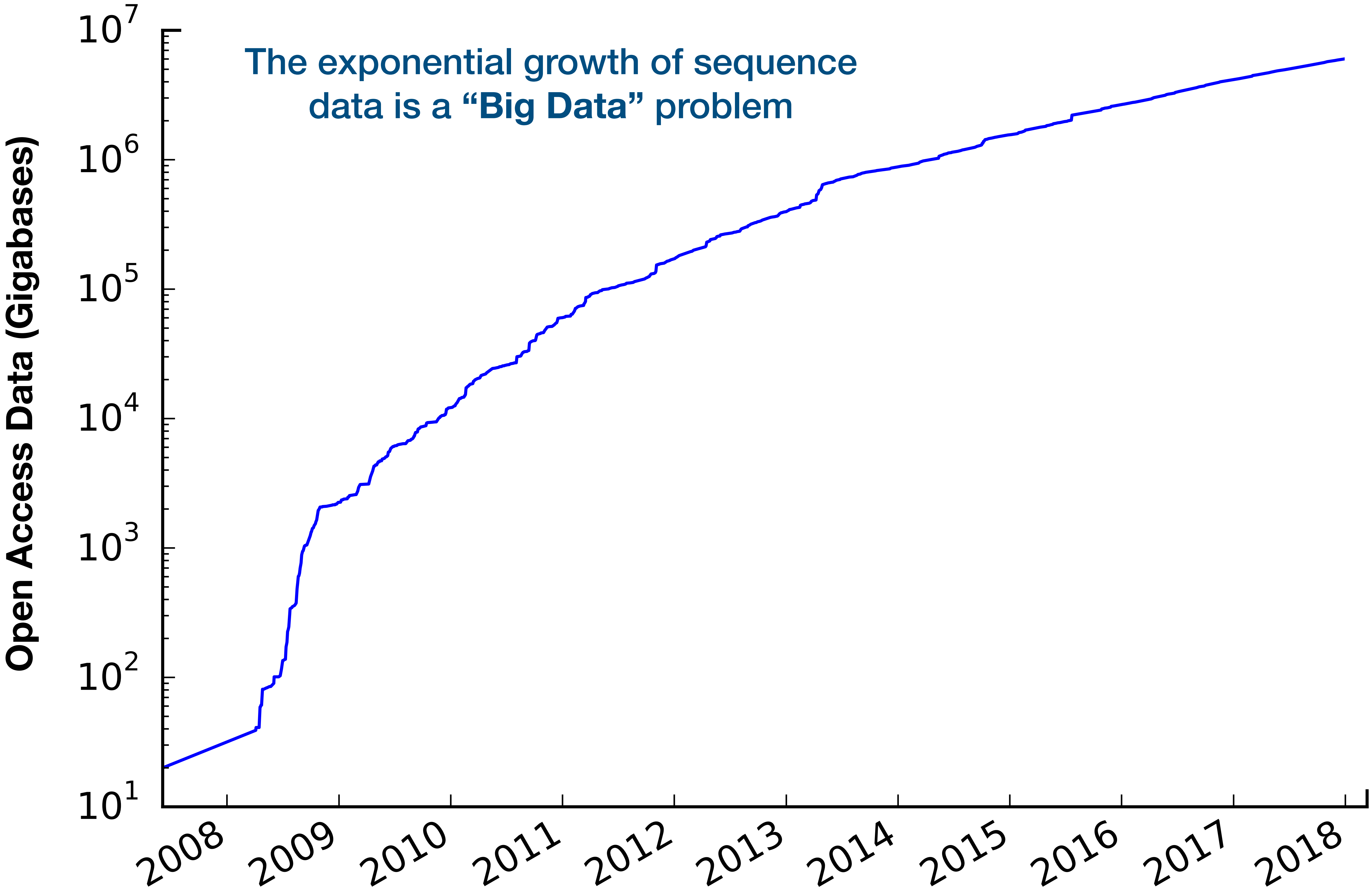
RNA-seq differential expression studies: more sequence or more replication?
Liu et al (2013) *Bioinformatics.* doi:10.1093/bioinformatics/btt688

See Lecture 10 RNA Sequencing

Sequence similarity at scale is a ‘big’ problem



Sequence similarity at scale is a ‘big’ problem


The exponential growth of sequence data is a **“Big Data”** problem

Biological questions at scale

Biological questions at scale

Experiment Discovery

I have sample X, find me related samples

Biological questions at scale

Experiment Discovery

I have sample X, find me related samples

Containment Query

I am interested in transcript X, find me studies expressing it

Biological questions at scale

Experiment Discovery

I have sample X, find me related samples

Containment Query

I am interested in transcript X, find me studies expressing it

Cardinality Estimation

I have N studies, how many unique sequences are present?

Biological questions at scale

Experiment Discovery

I have sample X, find me related samples

Containment Query

I am interested in transcript X, find me studies expressing it

Cardinality Estimation

I have N studies, how many unique sequences are present?

Expression Estimation

I am interested in the global expression patterns of transcriptome X on N studies

Biological questions at scale

Experiment Discovery

I have sample X, find me related samples

Containment Query

I am interested in transcript X, find me studies expressing it

Cardinality Estimation

I have N studies, how many unique sequences are present?

Expression Estimation

I am interested in the global expression patterns of transcriptome X on N studies

How can we solve these problems?

Biological questions at scale

Experiment Discovery

I have sample X, find me related samples

Containment Query

I am interested in transcript X, find me studies expressing it

Cardinality Estimation

I have N studies, how many unique sequences are present?

Expression Estimation

I am interested in the global expression patterns
of transcriptome X on N studies

How can we solve these problems?

Biological questions at scale

Experiment Discovery Measure similarity as **alignment** overlap

I have sample X, find me related samples

Containment Query

I am interested in transcript X, find me studies expressing it

Cardinality Estimation

I have N studies, how many unique sequences are present?

Expression Estimation

I am interested in the global expression patterns
of transcriptome X on N studies

How can we solve these problems?

Biological questions at scale

Experiment Discovery Measure similarity as **alignment** overlap

I have sample X, find me related samples

Containment Query **Align** each study to a reference, observe X

I am interested in transcript X, find me studies expressing it

Cardinality Estimation

I have N studies, how many unique sequences are present?

Expression Estimation

I am interested in the global expression patterns
of transcriptome X on N studies

How can we solve these problems?

Biological questions at scale

Experiment Discovery Measure similarity as **alignment** overlap

I have sample X, find me related samples

Containment Query **Align** each study to a reference, observe X

I am interested in transcript X, find me studies expressing it

Align each study to a reference, observe **Cardinality Estimation**

I have N studies, how many unique sequences are present?

Expression Estimation

I am interested in the global expression patterns
of transcriptome X on N studies

How can we solve these problems?

Biological questions at scale

Experiment Discovery Measure similarity as **alignment** overlap

I have sample X, find me related samples

Containment Query **Align** each study to a reference, observe X

I am interested in transcript X, find me studies expressing it

Align each study to a reference, observe **Cardinality Estimation**

I have N studies, how many unique sequences are present?

Align each study to a reference, normalize and observe **Expression Estimation**

I am interested in the global expression patterns
of transcriptome X on N studies

How can we solve these problems?

Methods for alignment at scale

ExAC Browser Beta

Rail-RNA

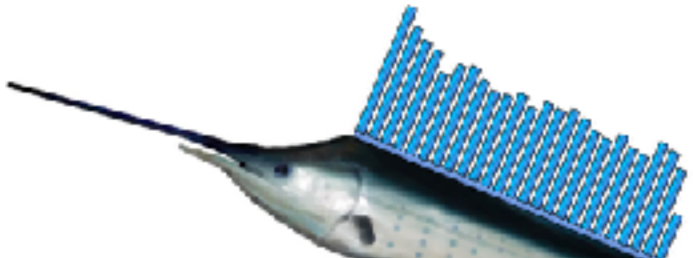
recount2

Methods for alignment at scale

ExAC Browser Beta

∀ Rail-RNA

recount2

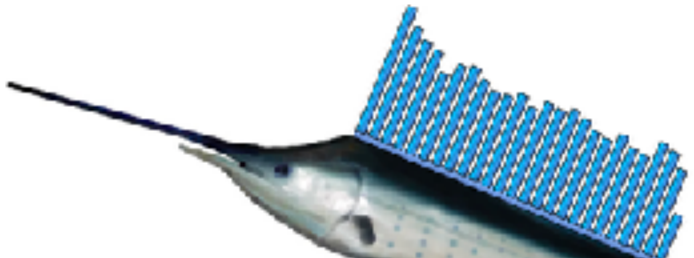


Methods for alignment at scale

ExAC Browser Beta

Rail-RNA

recount2



Rail-RNA: Bulk Alignment

Given a set of N sequencing studies,

1) **Aggregate** the reads into sets of overlapping **reads** and **readlets** (a subsequence of a set of reads with partial overlap)

2) Perform **parallel alignment** where each thread is given distinct **reads** based on nucleotide similarity

Rail-RNA: scalable analysis of RNA-seq splicing and coverage
Nellore et al (2017) *Bioinformatics*

Rail-RNA: Bulk Alignment

Given a set of N sequencing studies,

1) **Aggregate** the reads into sets of overlapping **reads** and **readlets** (a subsequence of a set of reads with partial overlap)

2) Perform **parallel alignment** where each thread is given distinct **reads** based on nucleotide similarity

3) **Second pass alignment** on all non-perfect or tied score alignments adds quality scores to break ties. Additional **readlets** are generated for poor alignments

4) **Readlets** are **parallel aligned** based on nucleotide similarity

Rail-RNA: scalable analysis of RNA-seq splicing and coverage
Nellore et al (2017) *Bioinformatics*

Rail-RNA: Bulk Alignment

Given a set of N sequencing studies,

1) **Aggregate** the reads into sets of overlapping **reads** and **readlets** (a subsequence of a set of reads with partial overlap)

2) Perform **parallel alignment** where each thread is given distinct **reads** based on nucleotide similarity

3) **Second pass alignment** on all non-perfect or tied score alignments adds quality scores to break ties. Additional **readlets** are generated for poor alignments

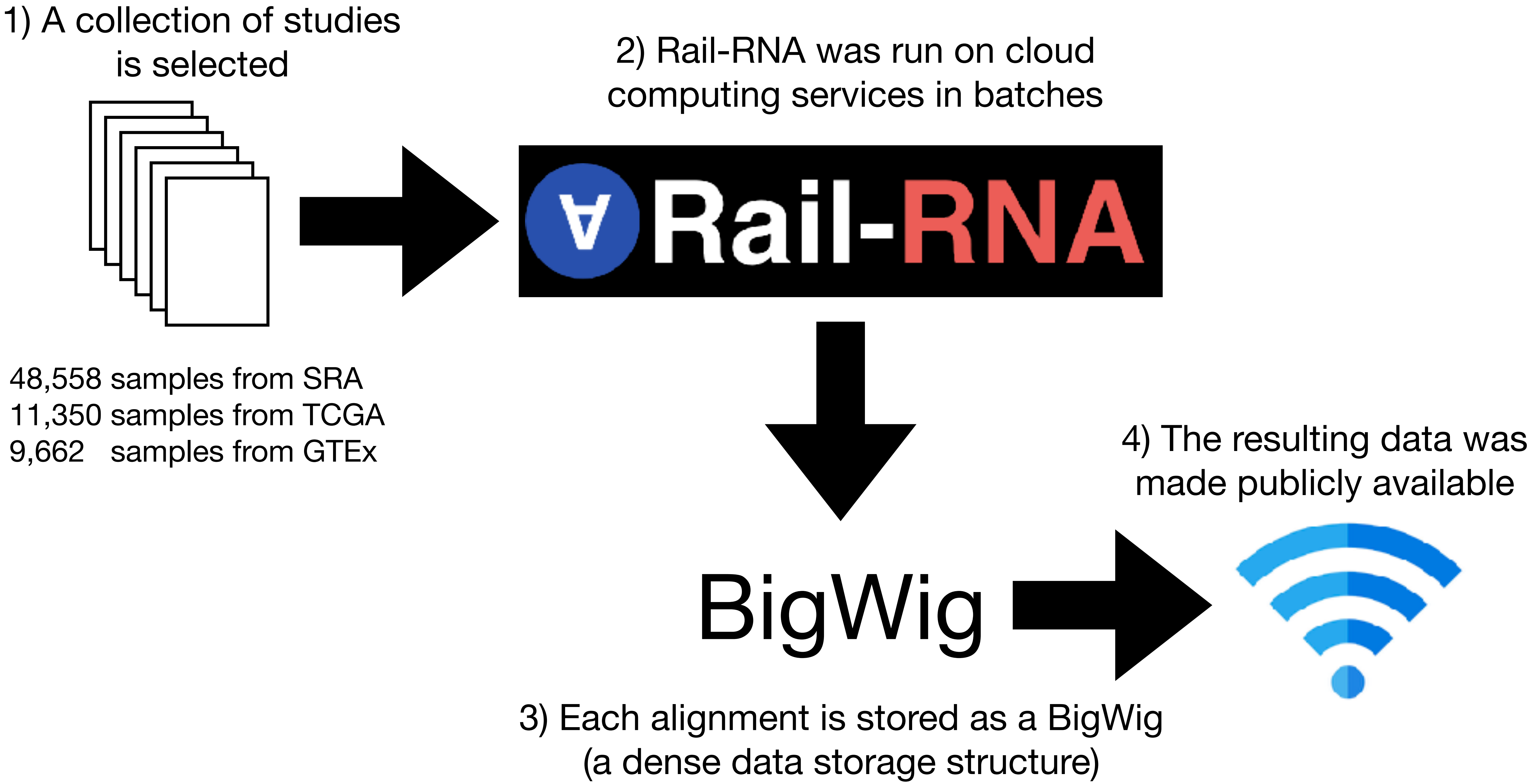
4) **Readlets** are **parallel aligned** based on nucleotide similarity

5) Further steps are taken to identify **exon-exon** junctions and one final alignment is performed using the **bulk exon data** identified from the studies

6) All individual alignments are output, with a single primary alignment compiled for each individual study

Rail-RNA: scalable analysis of RNA-seq splicing and coverage
Nellore et al (2017) *Bioinformatics*

Recount2: Combining methods



1) A collection of studies is selected

48,558 samples from SRA
11,350 samples from TCGA
9,662 samples from GTEx

2) Rail-RNA was run on cloud computing services in batches

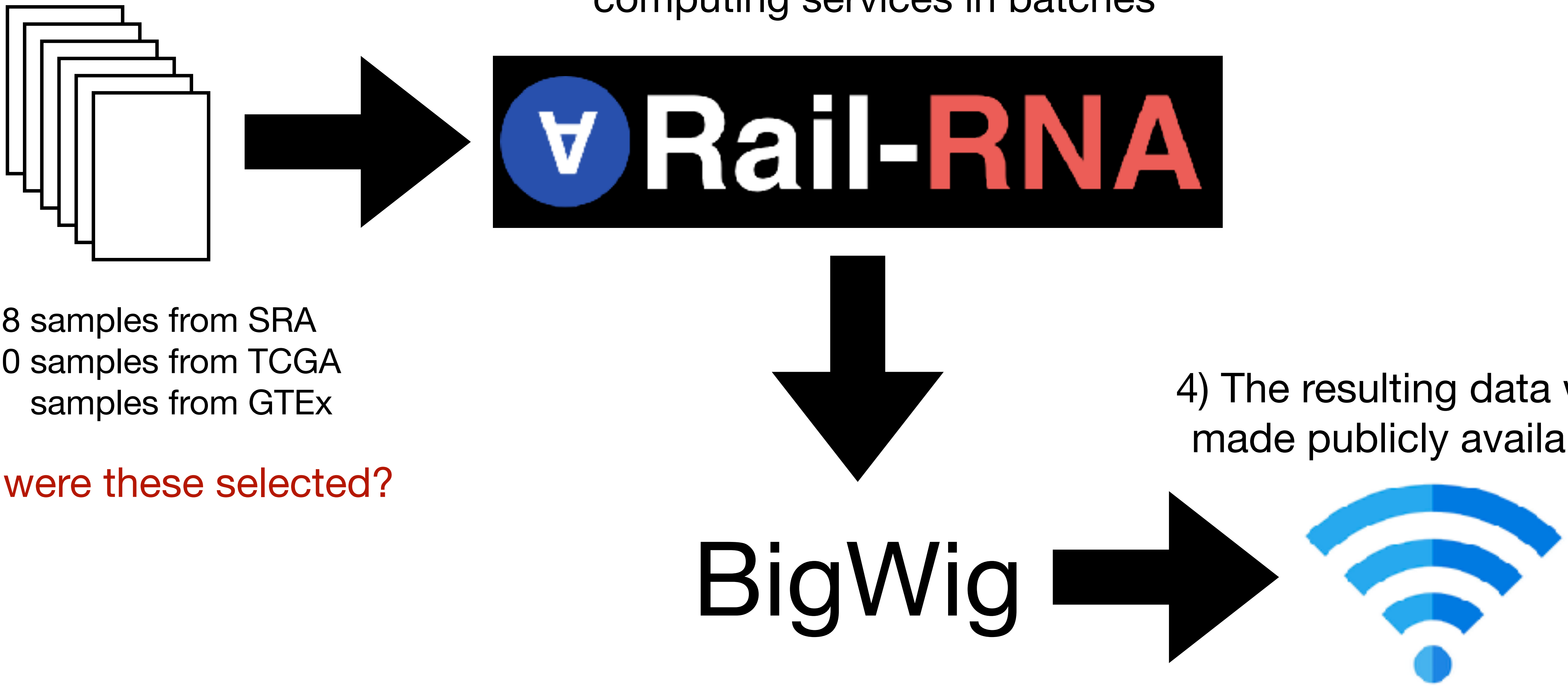
BigWig

3) Each alignment is stored as a BigWig (a dense data storage structure)

4) The resulting data was made publicly available

Reproducible RNA-seq analysis using *recount2*
Collado-Torres et al (2017) *Nature Biotechnology*

Recount2: Combining methods



1) A collection of studies is selected

48,558 samples from SRA
11,350 samples from TCGA
9,662 samples from GTEx

How were these selected?

2) Rail-RNA was run on cloud computing services in batches

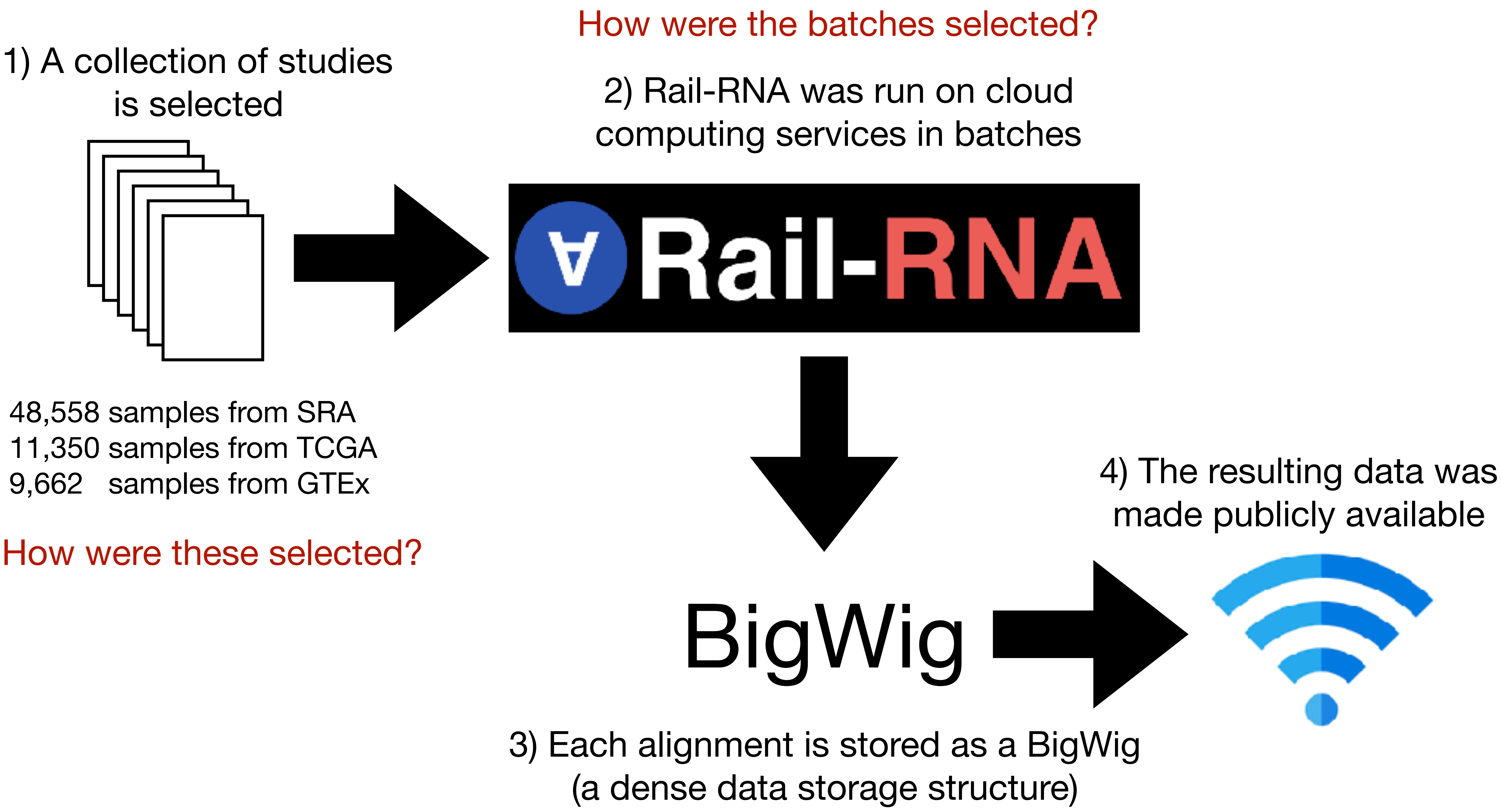
BigWig

3) Each alignment is stored as a BigWig (a dense data storage structure)

4) The resulting data was made publicly available

Reproducible RNA-seq analysis using *recount2*
Collado-Torres et al (2017) *Nature Biotechnology*

Recount2: Combining methods



How were the batches selected?

1) A collection of studies is selected

48,558 samples from SRA
11,350 samples from TCGA
9,662 samples from GTEx

How were these selected?

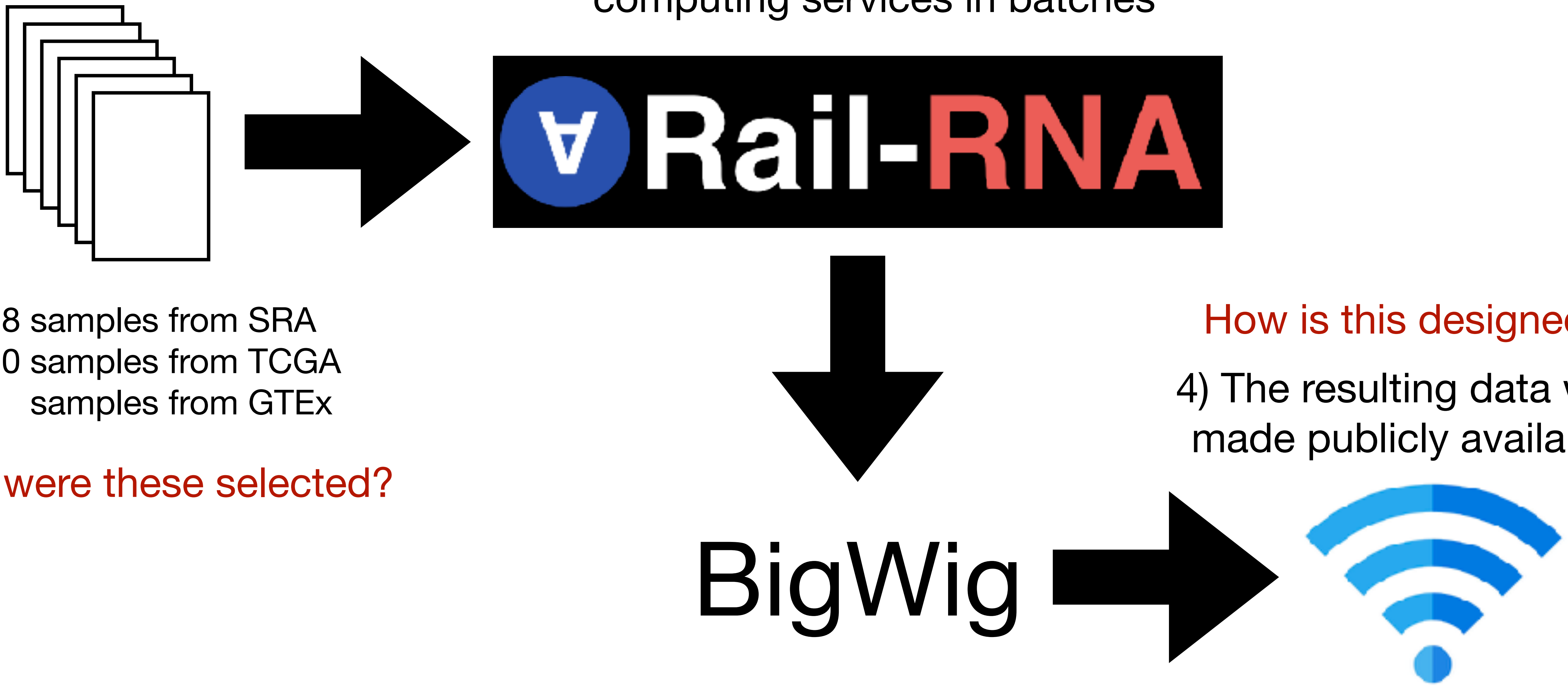
2) Rail-RNA was run on cloud computing services in batches

3) Each alignment is stored as a BigWig (a dense data storage structure)

4) The resulting data was made publicly available

Reproducible RNA-seq analysis using *recount2*
Collado-Torres et al (2017) *Nature Biotechnology*

Recount2: Combining methods



1) A collection of studies is selected

48,558 samples from SRA
11,350 samples from TCGA
9,662 samples from GTEx

How were these selected?

How were the batches selected?

2) Rail-RNA was run on cloud computing services in batches

BigWig

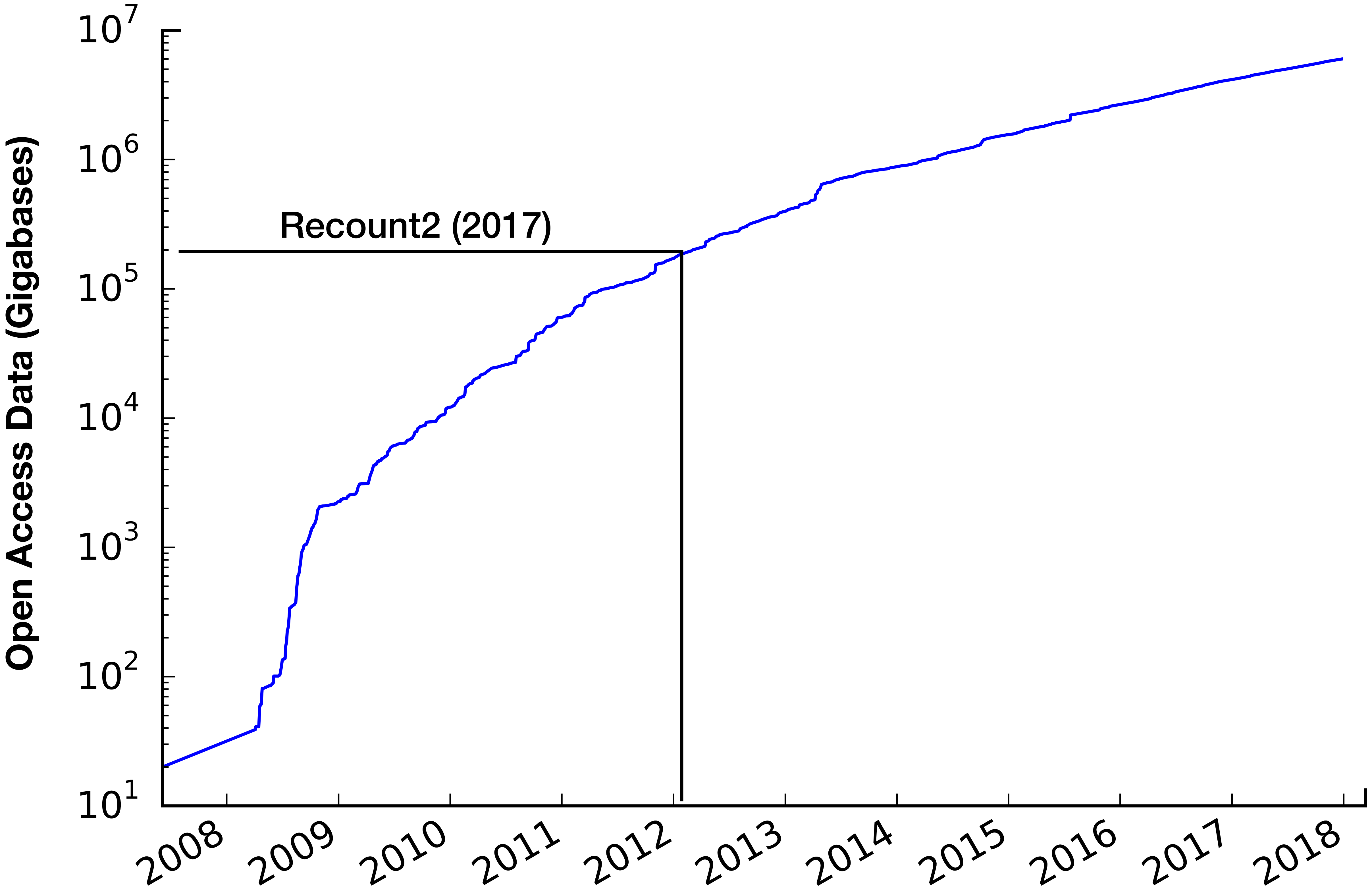
3) Each alignment is stored as a BigWig (a dense data storage structure)

How is this designed?

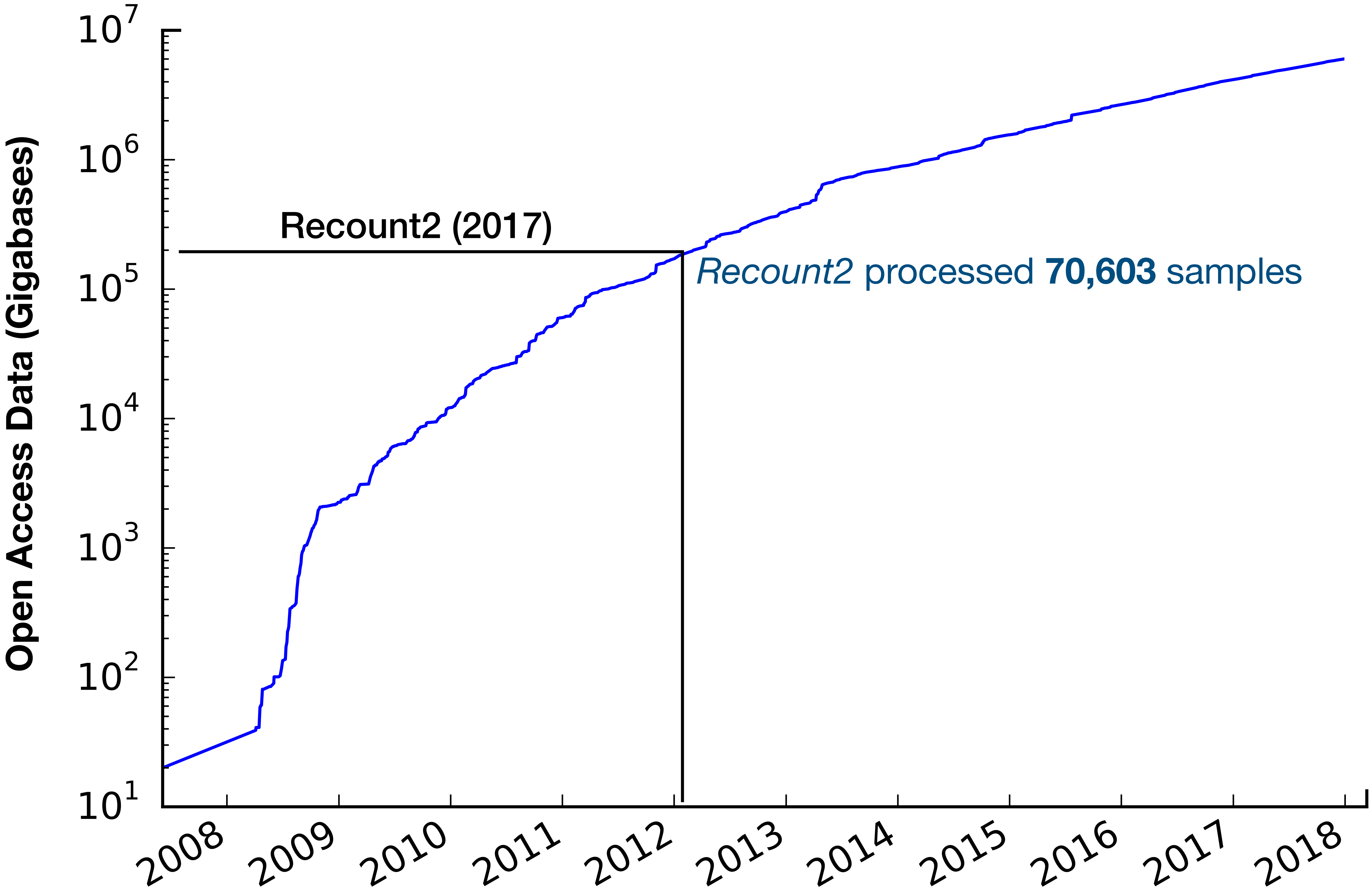
4) The resulting data was made publicly available

Reproducible RNA-seq analysis using *recount2*
Collado-Torres et al (2017) *Nature Biotechnology*

Alignment can't scale to Big Data

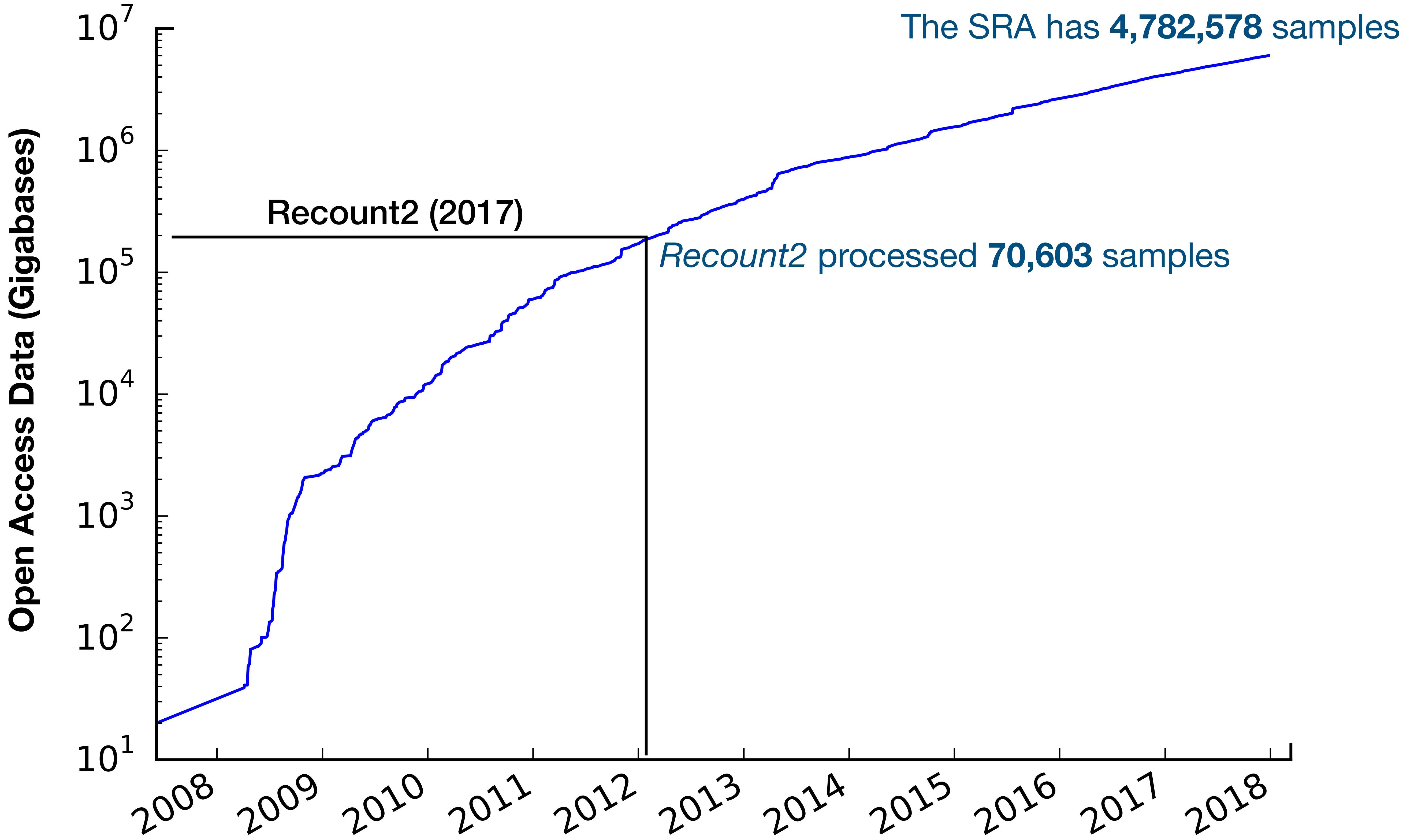


Alignment can't scale to Big Data



Recount2 processed **70,603** samples

Alignment can’t scale to Big Data



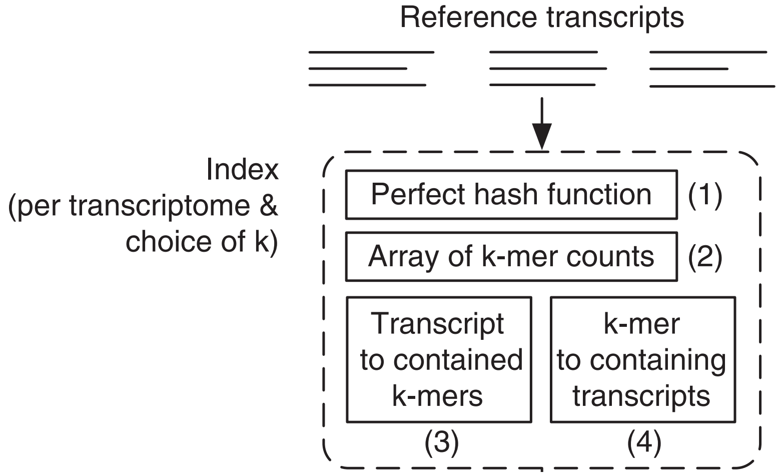
The SRA has **4,782,578** samples

Recount2 processed **70,603** samples

Recap: Sailfish

1) Indexing

- Parse reference transcriptome into kmers
- All unique kmers are hashed and counted
- Two indices for bi-directional mapping of transcripts and kmers



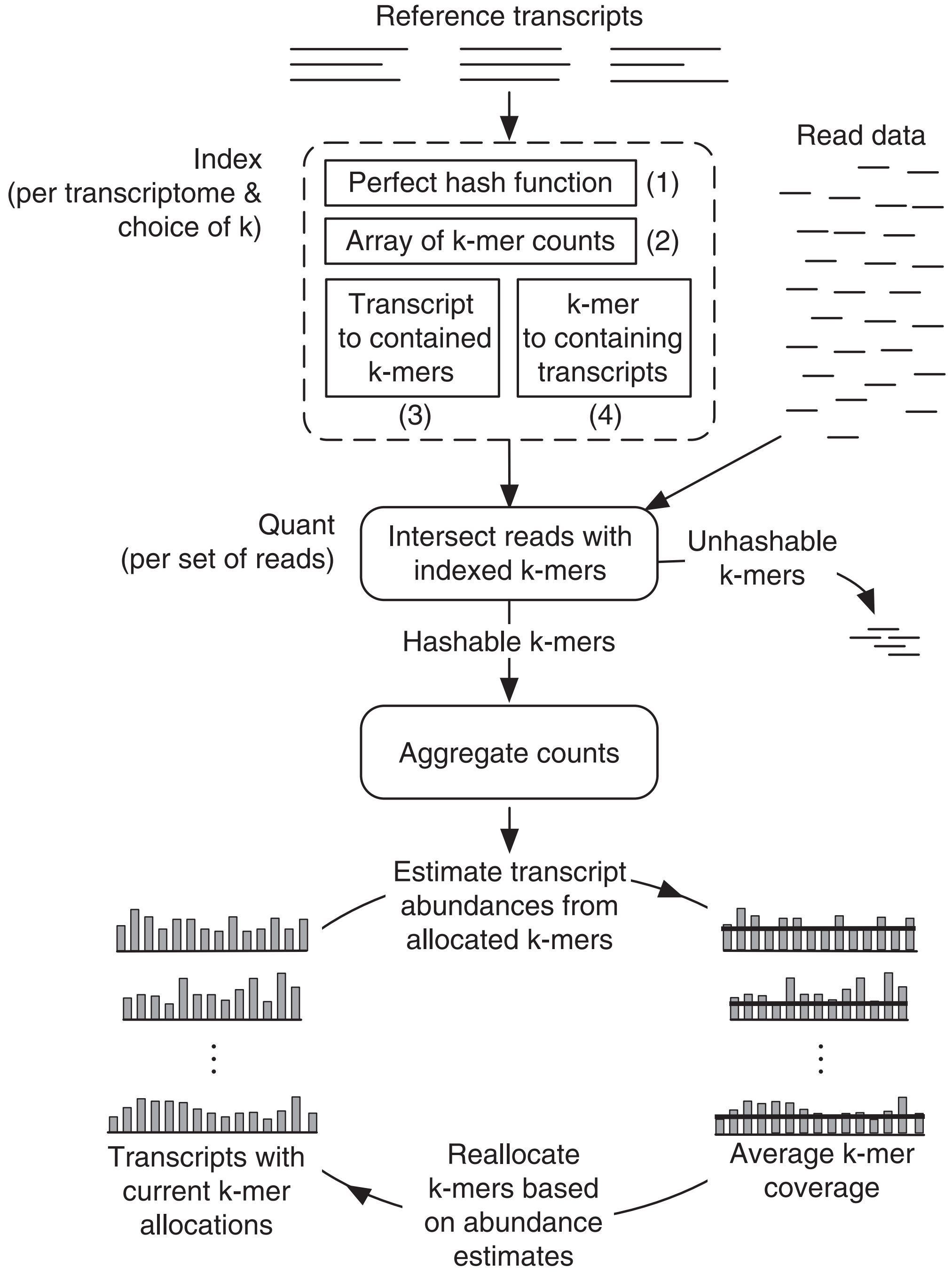
Recap: Sailfish

1) Indexing

- Parse reference transcriptome into kmers
- All unique kmers are hashed and counted
- Two indices for bi-directional mapping of transcripts and kmers

2) Quantification

- Count kmers in read set
- Use EM procedure to estimate transcript abundances, repeating as necessary


Reference transcripts
Index
(per transcriptome & choice of k)
Perfect hash function
(1)
Array of k-mer counts
(2)
Transcript to contained k-mers
(3)
k-mer to containing transcripts
(4)
Read data
Quant
(per set of reads)
Intersect reads with indexed k-mers
Unhashable k-mers
Hashable k-mers
Aggregate counts
Estimate transcript abundances from allocated k-mers
Transcripts with current k-mer allocations
Reallocate k-mers based on abundance estimates
Average k-mer coverage

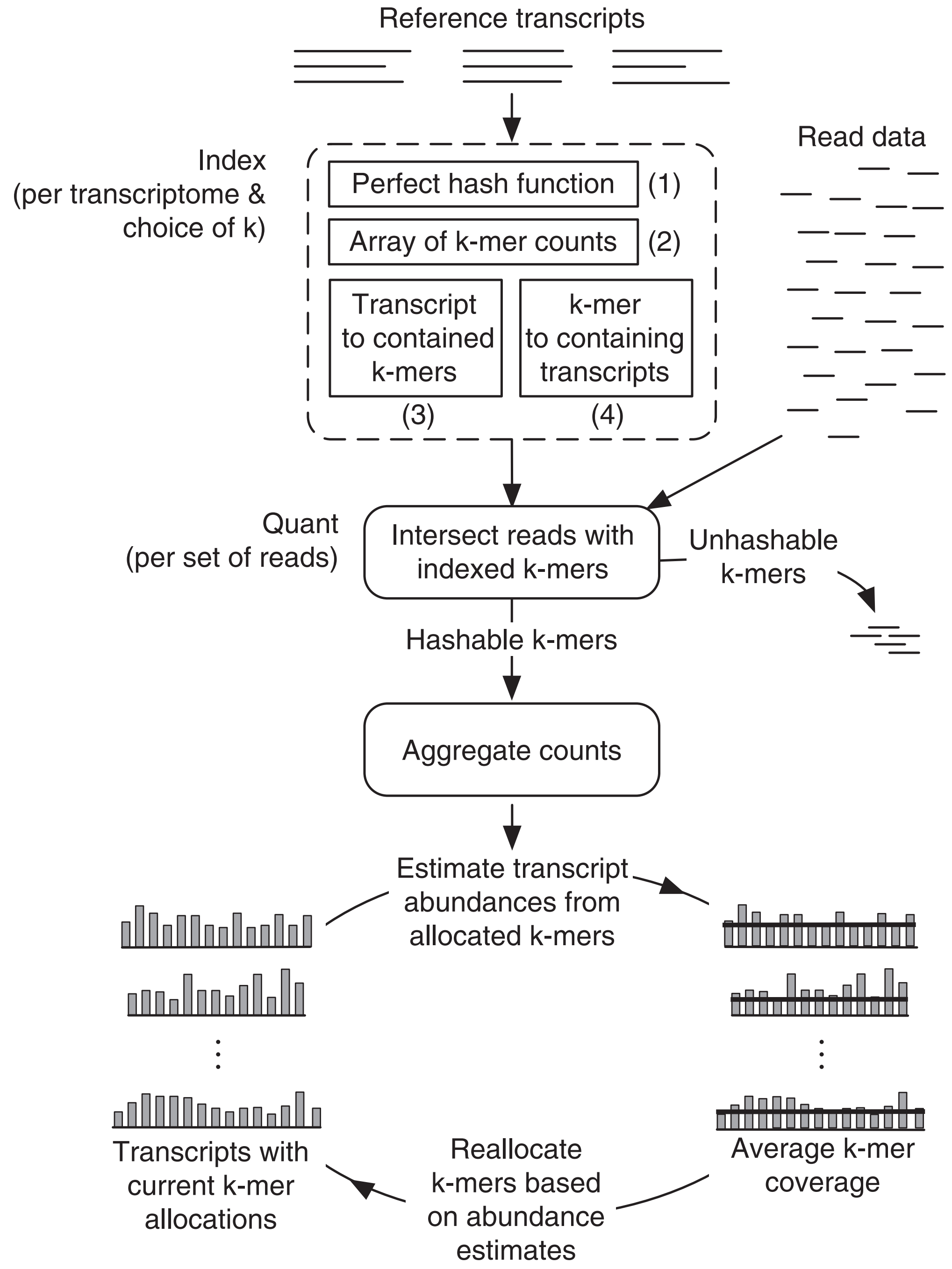
Recap: Sailfish

1) Indexing

- Parse reference transcriptome into kmers
- All unique kmers are hashed and counted
- Two indices for bi-directional mapping of transcripts and kmers

2) Quantification

- Count kmers in read set
- Use EM procedure to estimate transcript abundances, repeating as necessary



Sailfish, Salmon, and Kallisto all operate on a per-sample basis

The Big Data Problem

- There’s an abundance of underutilized sequence data
- Alignment can solve a number of important biological questions but doesn’t scale
- Alignment-free methods are more efficient but are not *several orders of magnitude* more efficient

The Big Data Problem

- There’s an abundance of underutilized sequence data
- Alignment can solve a number of important biological questions but doesn’t scale
- Alignment-free methods are more efficient but are not *several orders of magnitude* more efficient

How can we address big data?

The Big Data Problem

- There’s an abundance of underutilized sequence data
- Alignment can solve a number of important biological questions but doesn’t scale
- Alignment-free methods are more efficient but are not *several orders of magnitude* more efficient

How can we address big data?

Sketching!

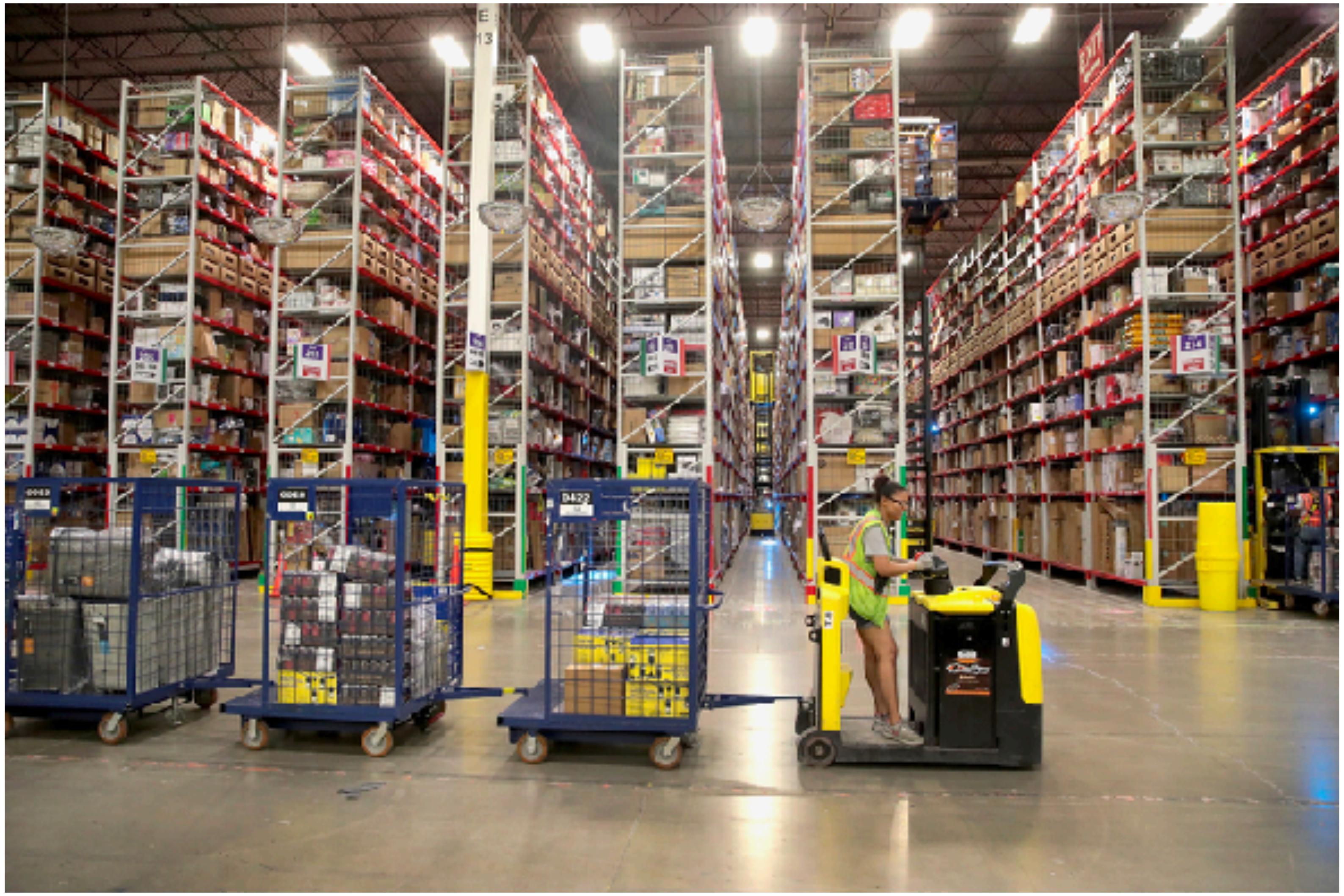
Sketching algorithms trade accuracy for speed

Given a box, it's easy to discover what's inside

Sketching algorithms trade accuracy for speed

Given a box, it's easy to discover what's inside

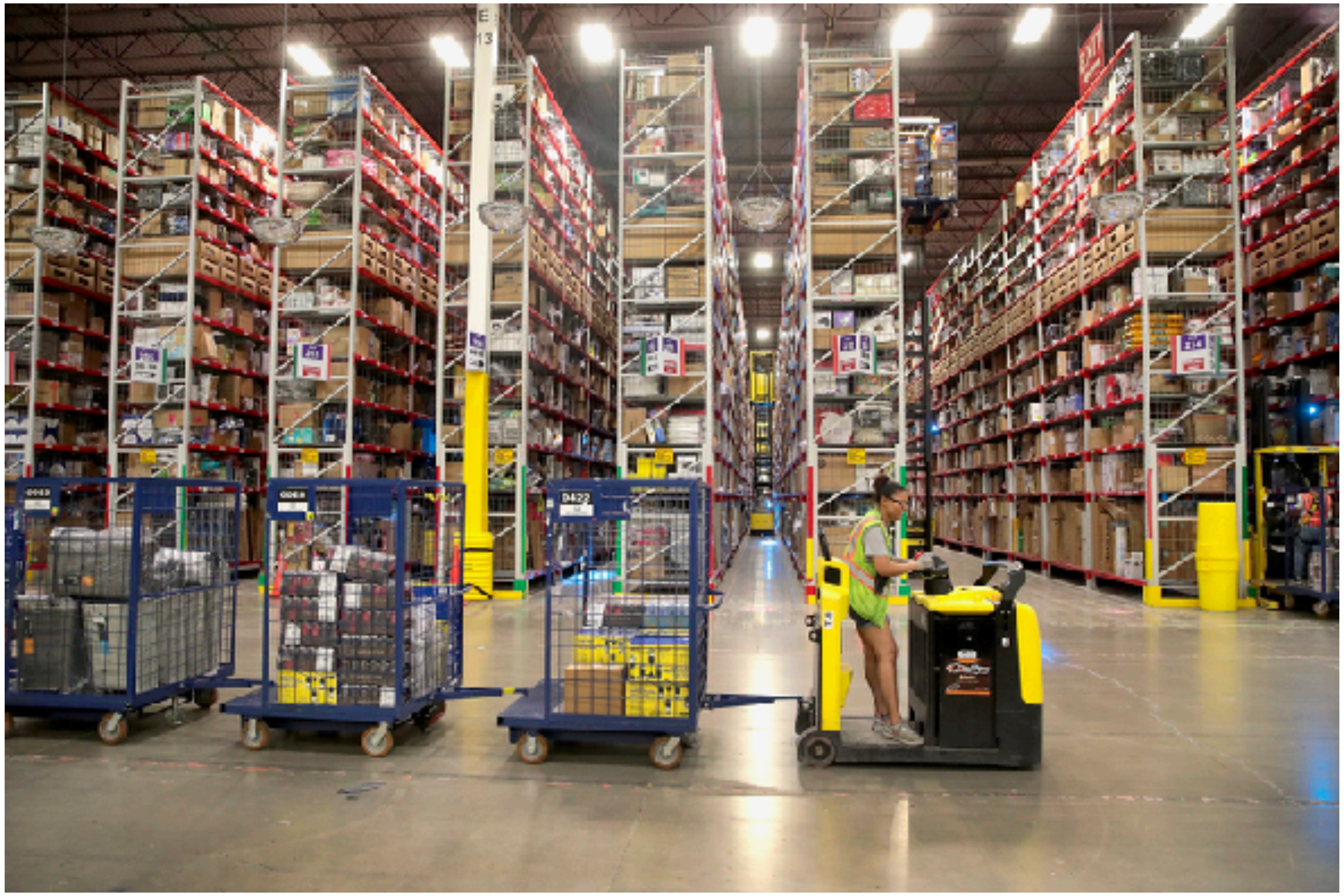
But what is there's too many boxes?



Sketching algorithms trade accuracy for speed

Given a box, it's easy to discover what's inside

But what is there's too many boxes?



A sketch solution would organize boxes based on labels, simpler observations such as size or weight, or sub-sample boxes to learn about its neighbors

Tradeoffs in Similarity Metrics

- Hamming distance
  - Count the number of substitutions to transform one string into another

```
MIKESCHATZ
||X||XXXX|
MICESHATZZ
    5
```

- Edit distance
  - The minimum number of substitutions, insertions, or deletions to transform one string into another

```
MIKESCHAT-Z
||X||X|||X|
MICES-HATZZ
     3
```

See Lecture 10: RNA Sequencing

Tradeoffs in Similarity Metrics

- Hamming distance
  - Count the number of substitutions to transform one string into another

```
MIKESCHATZ
||x||xxxx|
MICESHATZZ
     5
```

Much faster to calculate

- Edit distance
  - The minimum number of substitutions, insertions, or deletions to transform one string into another

```
MIKESCHAT-Z
||x||x|||x|
MICES-HATZZ
     3
```

See Lecture 10: RNA Sequencing

Tradeoffs in Similarity Metrics

- Hamming distance
  - Count the number of substitutions to transform one string into another

```
MIKESCHATZ
||X||XXXX|
MICESHATZZ
    5
```

Much faster to calculate

- Edit distance
  - The minimum number of substitutions, insertions, or deletions to transform one string into another

```
MIKESCHAT-Z
||X||X|||X|
MICES-HATZZ
     3
```

More biologically meaningful

See Lecture 10: RNA Sequencing

Tradeoffs in Similarity Metrics

- Hamming distance
  - Count the number of substitutions to transform one string into another

```
MIKESCHATZ
||X||XXXX|
MICESHATZZ
    5
```

Much faster to calculate

- Edit distance
  - The minimum number of substitutions, insertions, or deletions to transform one string into another

```
MIKESCHAT-Z
||X||X|||X|
MICES-HATZZ
     3
```

More biologically meaningful

Can you think of other tradeoffs in alignment methods?

See Lecture 10: RNA Sequencing

Solving Big Data through Sketching

Experiment Discovery

I have sample X, find me related samples

Containment Query

I am interested in transcript X, find me experiments expressing it

Cardinality Estimation

I have N studies, how many unique sequences are present?

Expression Estimation

I am interested in the global expression patterns
of transcriptome X in N studies

Solving Big Data through Sketching

Experiment Discovery Mash [Minhash]

I have sample X, find me related samples

Containment Query

I am interested in transcript X, find me experiments expressing it

Cardinality Estimation

I have N studies, how many unique sequences are present?

Expression Estimation

I am interested in the global expression patterns
of transcriptome X in N studies

Solving Big Data through Sketching

Experiment Discovery Mash [Minhash]

I have sample X, find me related samples

Containment Query Sequence Bloom Trees [Bloom Filter]

I am interested in transcript X, find me experiments expressing it

Cardinality Estimation

I have N studies, how many unique sequences are present?

Expression Estimation

I am interested in the global expression patterns
of transcriptome X in N studies

Solving Big Data through Sketching

Experiment Discovery Mash [Minhash]

I have sample X, find me related samples

Containment Query Sequence Bloom Trees [Bloom Filter]

I am interested in transcript X, find me experiments expressing it

Dashing [HyperLogLog] **Cardinality Estimation**

I have N studies, how many unique sequences are present?

Expression Estimation

I am interested in the global expression patterns
of transcriptome X in N studies

Solving Big Data through Sketching

Experiment Discovery Mash [Minhash]

I have sample X, find me related samples

Containment Query Sequence Bloom Trees [Bloom Filter]

I am interested in transcript X, find me experiments expressing it

Dashing [HyperLogLog] **Cardinality Estimation**

I have N studies, how many unique sequences are present?

Salmon / Kallisto **Expression Estimation**

I am interested in the global expression patterns
of transcriptome X in N studies

Solving Big Data through Sketching

Experiment Discovery Mash [Minhash]

I have sample X, find me related samples

Containment Query Sequence Bloom Trees [Bloom Filter]

I am interested in transcript X, find me experiments expressing it

Dashing [HyperLogLog] **Cardinality Estimation**

I have N studies, how many unique sequences are present?

Salmon / Kallisto **Expression Estimation**

I am interested in the global expression patterns
of transcriptome X in N studies

Mantis [Counting Quotient Filter]

The Minhash Sketch

1) Sequence decomposed into **kmers**

S_1 : CATGGACCGACCAG   GCAGTACCGATCGT : S_2

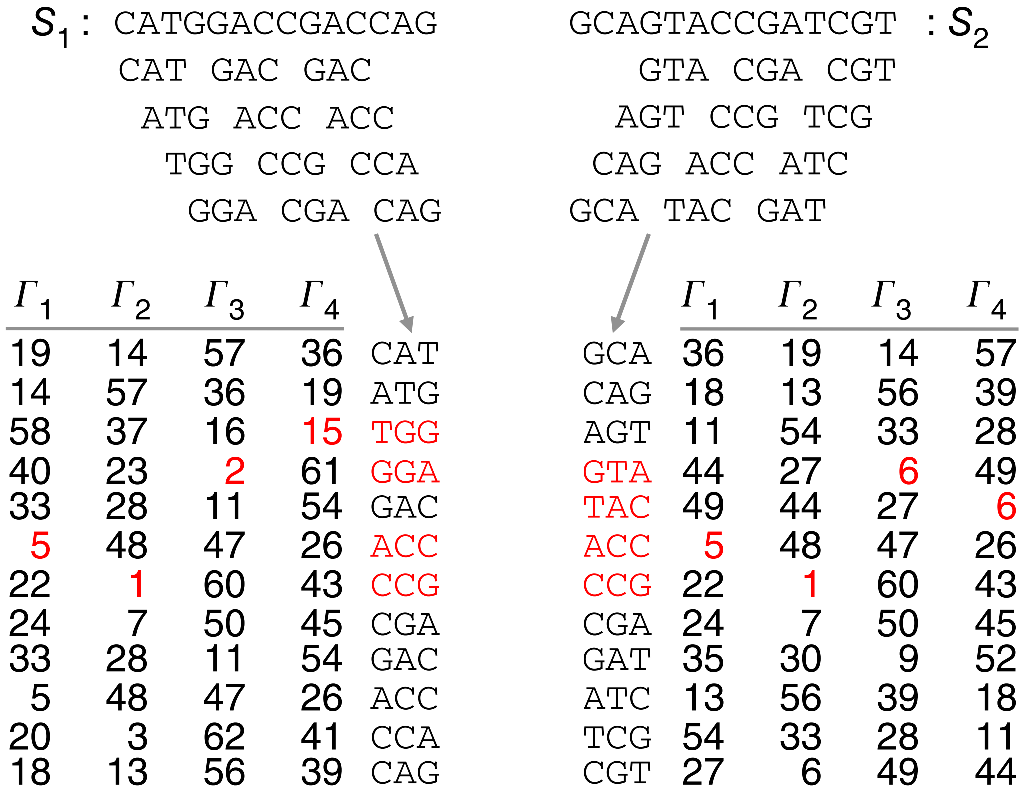
```
CAT GAC GAC             GTA CGA CGT
 ATG ACC ACC           AGT CCG TCG
  TGG CCG CCA         CAG ACC ATC
   GGA CGA CAG       GCA TAC GAT
```

Assembling large genomes with single-molecule sequencing and locality-sensitive hashing
Berlin et al (2015) *Nature Biotechnology*

The Minhash Sketch

1) Sequence decomposed into **kmers**

2) Multiple hash functions (**Γ**) map kmers to values.



S_1 : CATGGACCGACCAG
CAT GAC GAC
ATG ACC ACC
TGG CCG CCA
GGA CGA CAG

GCAGTACCGATCGT : S_2
GTA CGA CGT
AGT CCG TCG
CAG ACC ATC
GCA TAC GAT

Γ_1	Γ_2	Γ_3	Γ_4	
19	14	57	36	CAT
14	57	36	19	ATG
58	37	16	15	TGG
40	23	2	61	GGA
33	28	11	54	GAC
5	48	47	26	ACC
22	1	60	43	CCG
24	7	50	45	CGA
33	28	11	54	GAC
5	48	47	26	ACC
20	3	62	41	CCA
18	13	56	39	CAG

	Γ_1	Γ_2	Γ_3	Γ_4
GCA	36	19	14	57
CAG	18	13	56	39
AGT	11	54	33	28
GTA	44	27	6	49
TAC	49	44	27	6
ACC	5	48	47	26
CCG	22	1	60	43
CGA	24	7	50	45
GAT	35	30	9	52
ATC	13	56	39	18
TCG	54	33	28	11
CGT	27	6	49	44

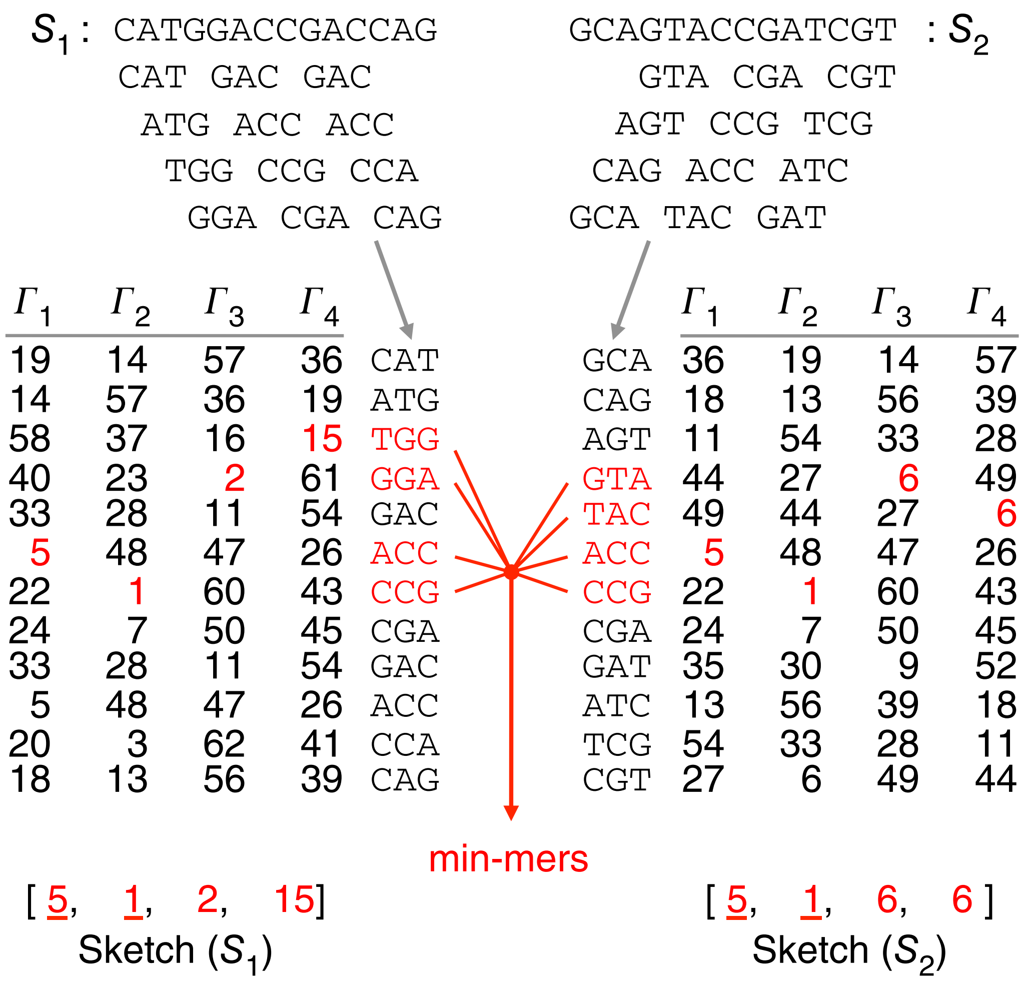
Assembling large genomes with single-molecule sequencing and locality-sensitive hashing
Berlin et al (2015) *Nature Biotechnology*

The Minhash Sketch

1) Sequence decomposed into **kmers**

2) Multiple hash functions (**Γ**) map kmers to values.

3) The smallest values for each hash function is chosen



S_1 : CATGGACCGACCAG

```
CAT  GAC  GAC
 ATG  ACC  ACC
  TGG  CCG  CCA
   GGA  CGA  CAG
```

GCAGTACCGATCGT : S_2

```
   GTA  CGA  CGT
  AGT  CCG  TCG
 CAG  ACC  ATC
GCA  TAC  GAT
```

Γ_1	Γ_2	Γ_3	Γ_4	
19	14	57	36	CAT
14	57	36	19	ATG
58	37	16	15	TGG
40	23	2	61	GGA
33	28	11	54	GAC
5	48	47	26	ACC
22	1	60	43	CCG
24	7	50	45	CGA
33	28	11	54	GAC
5	48	47	26	ACC
20	3	62	41	CCA
18	13	56	39	CAG

	Γ_1	Γ_2	Γ_3	Γ_4
GCA	36	19	14	57
CAG	18	13	56	39
AGT	11	54	33	28
GTA	44	27	6	49
TAC	49	44	27	6
ACC	5	48	47	26
CCG	22	1	60	43
CGA	24	7	50	45
GAT	35	30	9	52
ATC	13	56	39	18
TCG	54	33	28	11
CGT	27	6	49	44

[5, 1, 2, 15]
Sketch (S_1)

[5, 1, 6, 6]
Sketch (S_2)

Assembling large genomes with single-molecule sequencing and locality-sensitive hashing
Berlin et al (2015) *Nature Biotechnology*

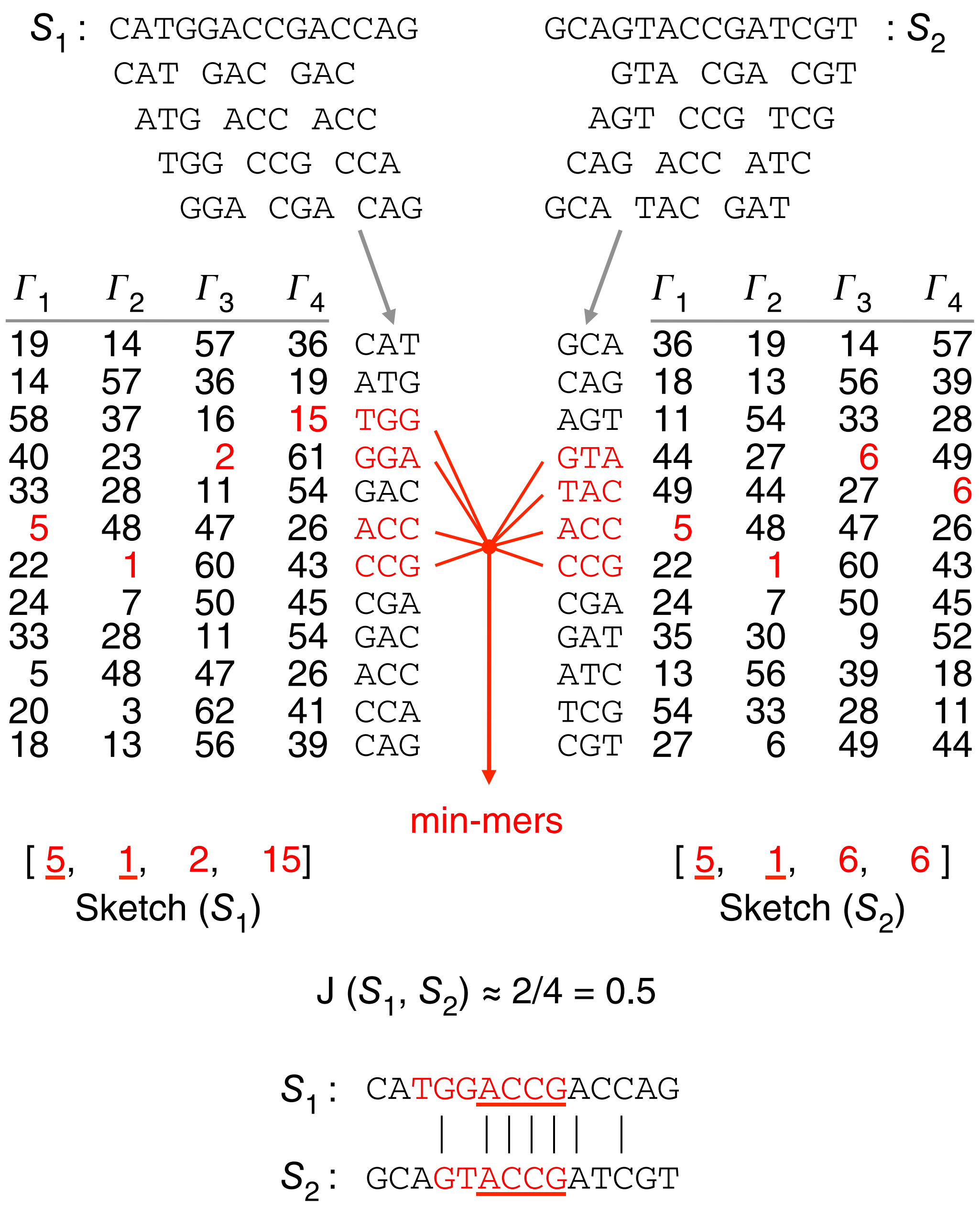
The Minhash Sketch

1) Sequence decomposed into **kmers**

2) Multiple hash functions (**Γ**) map kmers to values.

3) The smallest values for each hash function is chosen

4) The Jaccard similarity can be estimated by the overlap in the **Min**imum **Hash**es (**Minhash)**



S_1 : CATGGACCGACCAG

CAT GAC GAC
ATG ACC ACC
TGG CCG CCA
GGA CGA CAG

GCAGTACCGATCGT : S_2

GTA CGA CGT
AGT CCG TCG
CAG ACC ATC
GCA TAC GAT

Γ_1	Γ_2	Γ_3	Γ_4	
19	14	57	36	CAT
14	57	36	19	ATG
58	37	16	15	TGG
40	23	2	61	GGA
33	28	11	54	GAC
5	48	47	26	ACC
22	1	60	43	CCG
24	7	50	45	CGA
33	28	11	54	GAC
5	48	47	26	ACC
20	3	62	41	CCA
18	13	56	39	CAG

	Γ_1	Γ_2	Γ_3	Γ_4
GCA	36	19	14	57
CAG	18	13	56	39
AGT	11	54	33	28
GTA	44	27	6	49
TAC	49	44	27	6
ACC	5	48	47	26
CCG	22	1	60	43
CGA	24	7	50	45
GAT	35	30	9	52
ATC	13	56	39	18
TCG	54	33	28	11
CGT	27	6	49	44

min-mers

[5, 1, 2, 15]
Sketch (S_1)

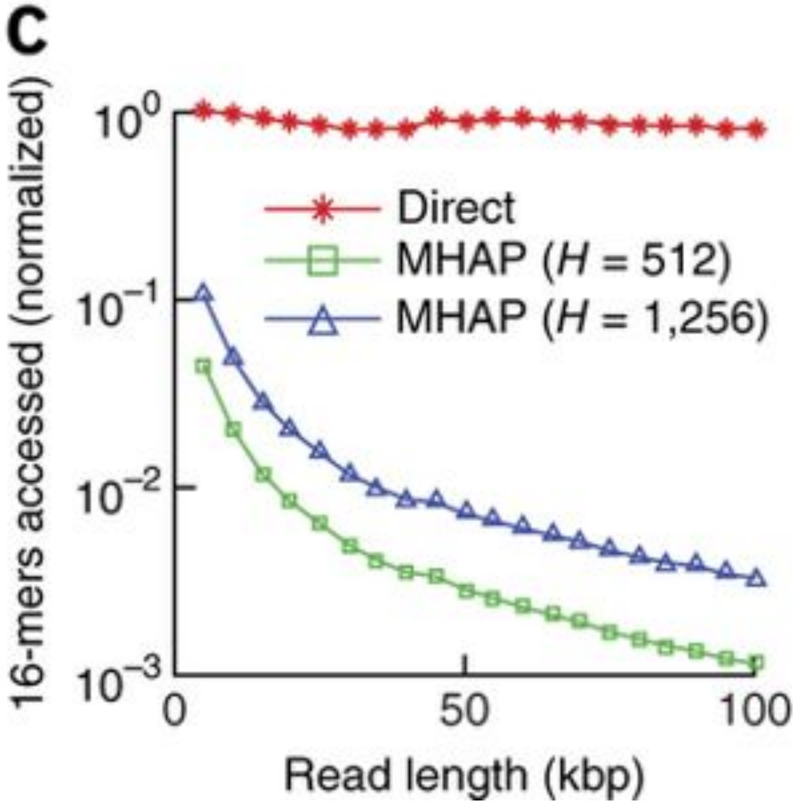
[5, 1, 6, 6]
Sketch (S_2)

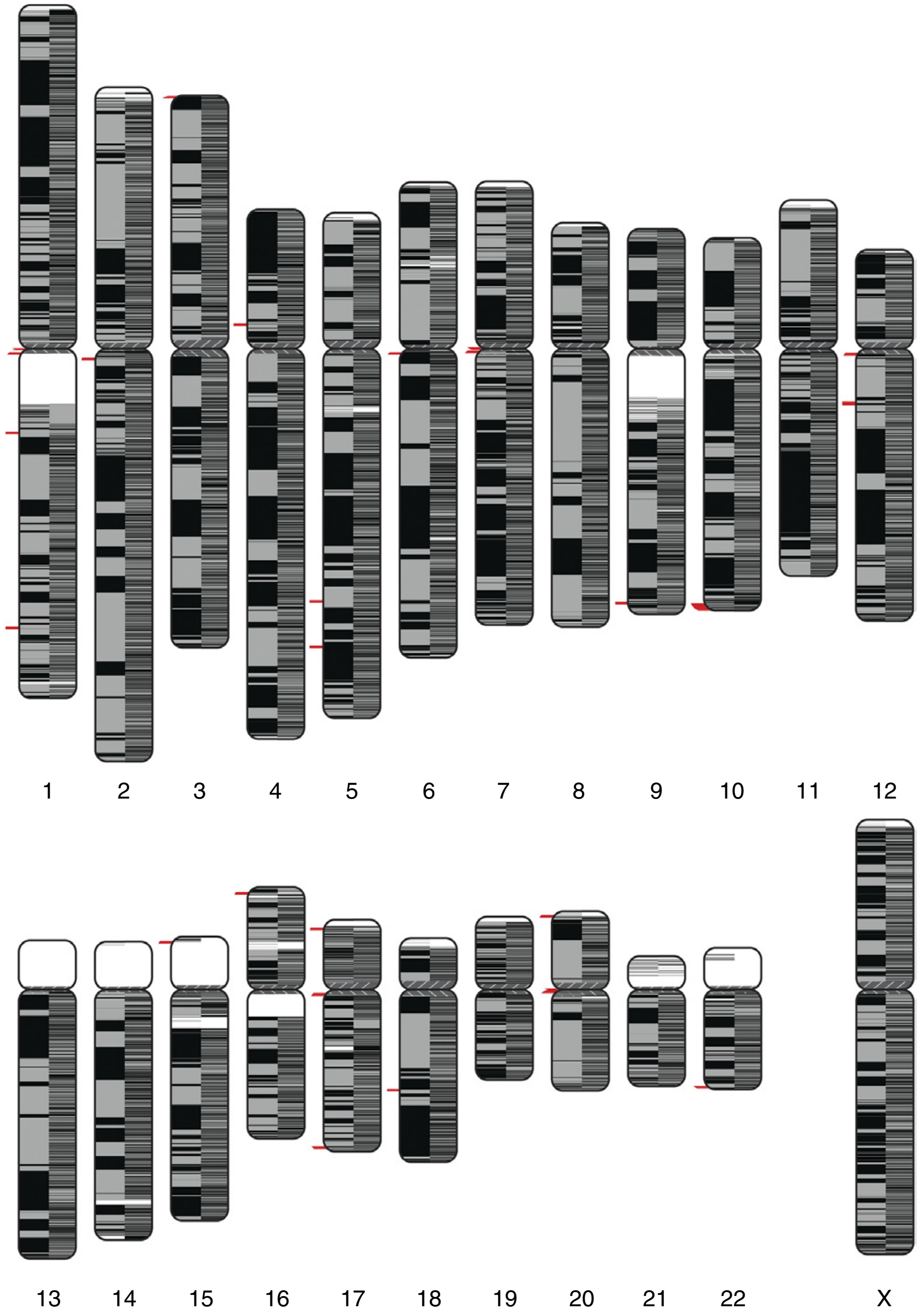
J (S_1, S_2) ≈ 2/4 = 0.5

S_1 : CATGGACCGACCAG
| ||||| |
S_2 : GCAGTACCGATCGT

Assembling large genomes with single-molecule sequencing and locality-sensitive hashing
Berlin et al (2015) *Nature Biotechnology*

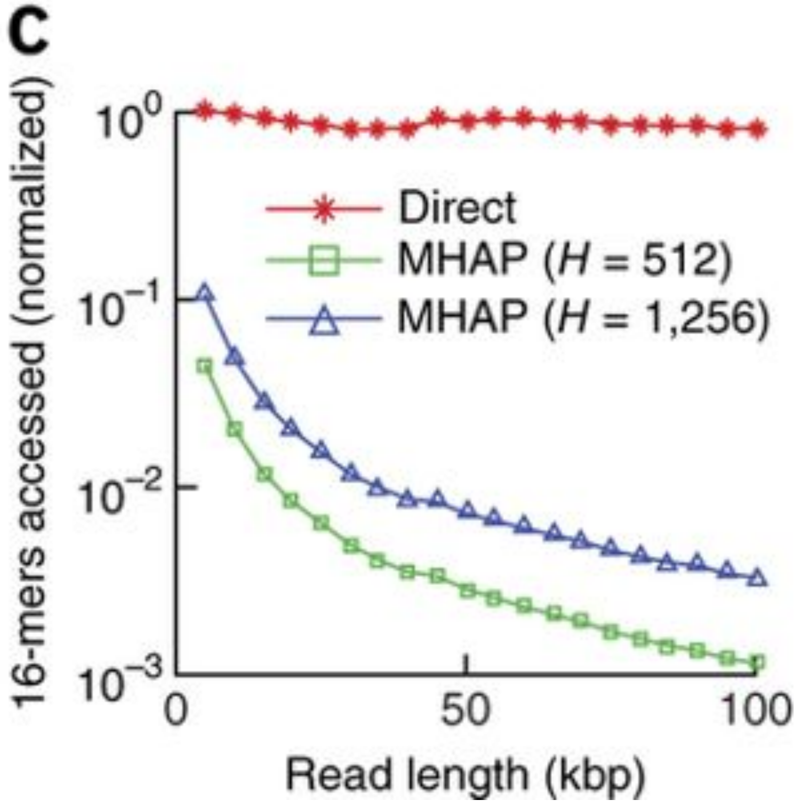
MHAP uses Minhash to approximate read overlap

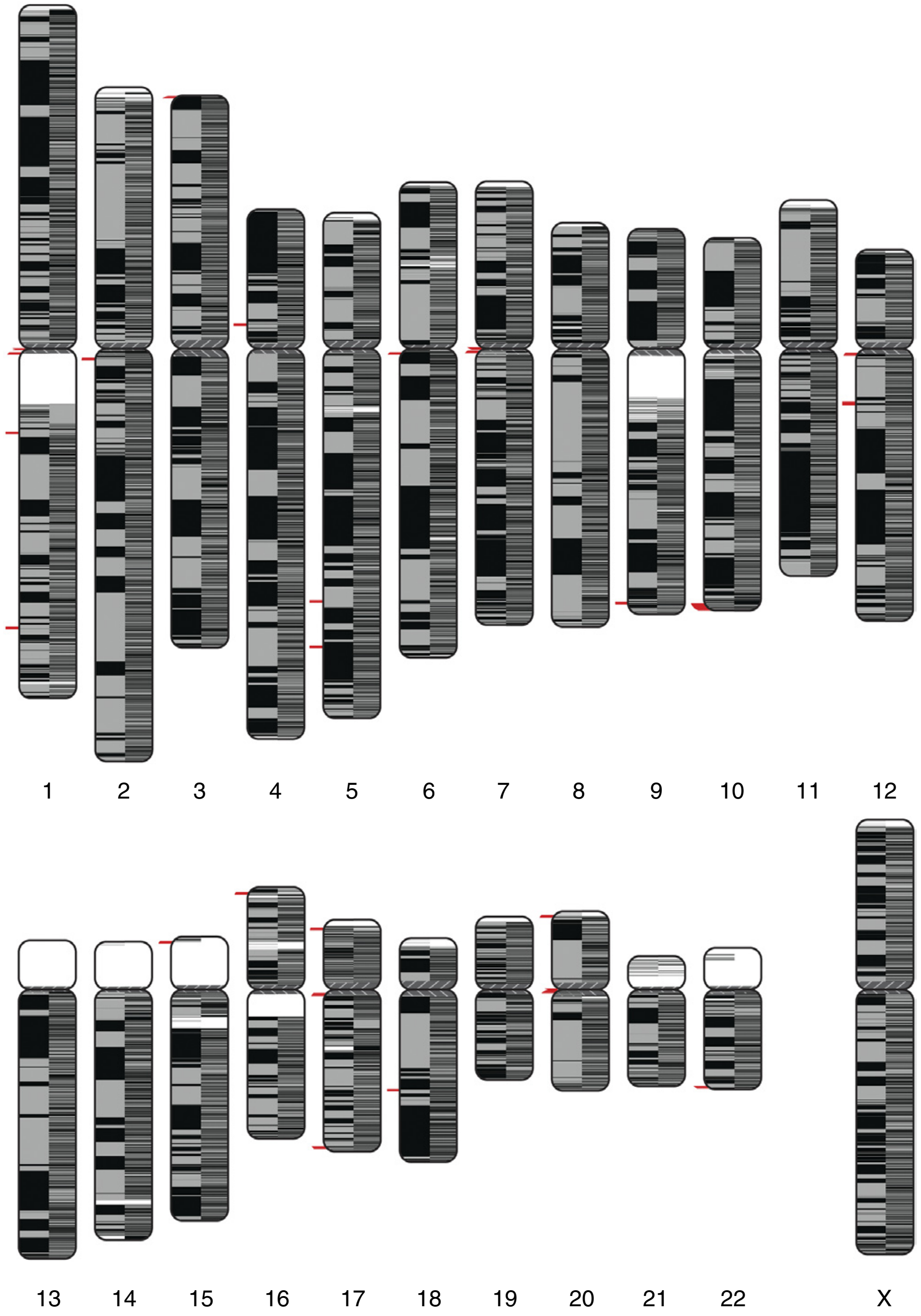




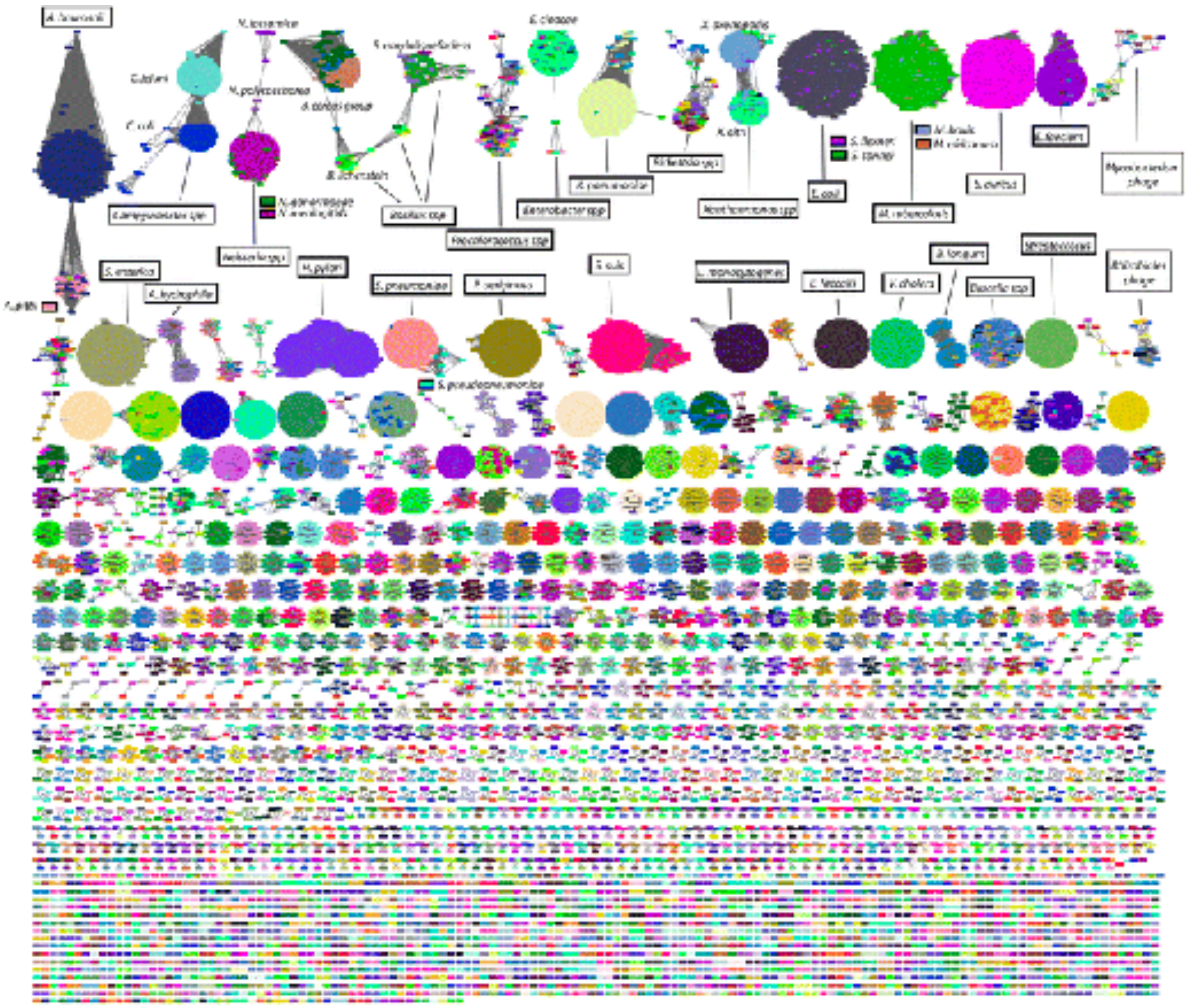
MHAP uses Minhash to approximate read overlap

An improvement in heuristic efficiency leads to an improvement in accuracy

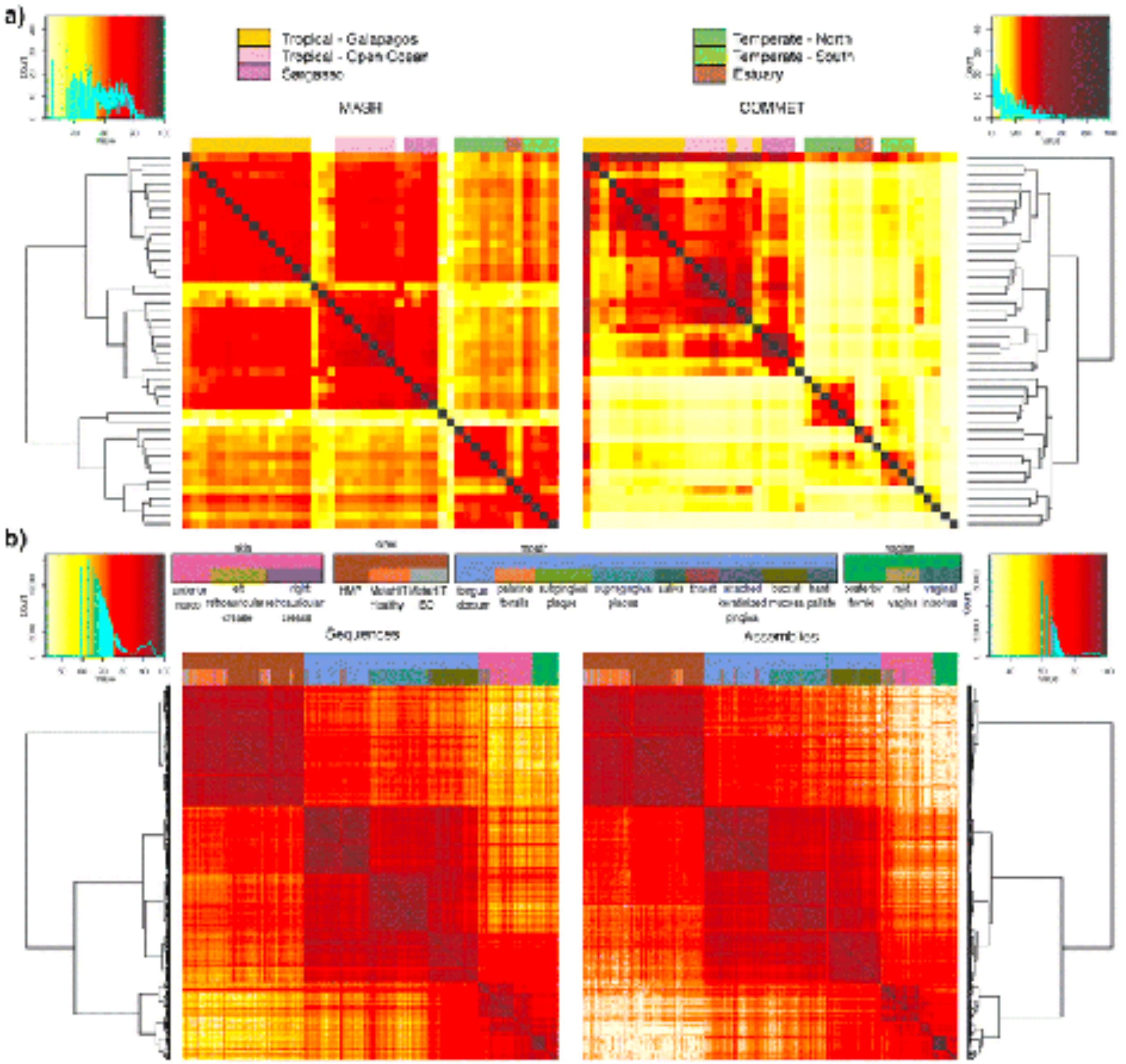




Mash uses Minhash to approximate genome and read overlap



Ref Seq Genome Clusterings

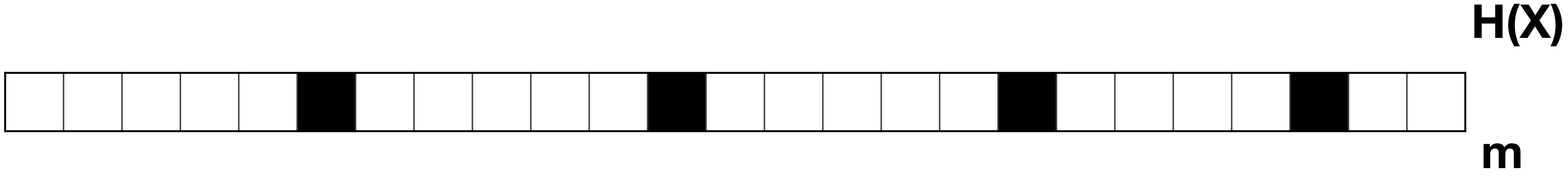


Read set overlap

Mash: fast genome and metagenome distance estimation using MinHash
Ondov et al (2016) *Genome Biology*

Bloom Filters efficiently encode sets

A Bloom Filter is a length *m* bit-vector and associated hash function(s) H(X)



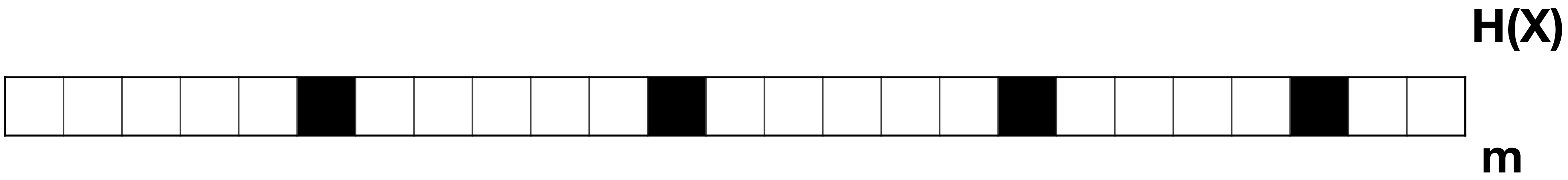
Space/Time Trade-offs in Hash Coding with Allowable Errors.
Burton Bloom. (1970) *Communications of the ACM.*

Bloom Filters efficiently encode sets

A Bloom Filter is a length *m* bit-vector and associated hash function(s) H(X)

Hash Function H(X) takes in an arbitrary element and returns an integer

Ex: H(🟦) = 12



Space/Time Trade-offs in Hash Coding with Allowable Errors.
Burton Bloom. (1970) *Communications of the ACM.*

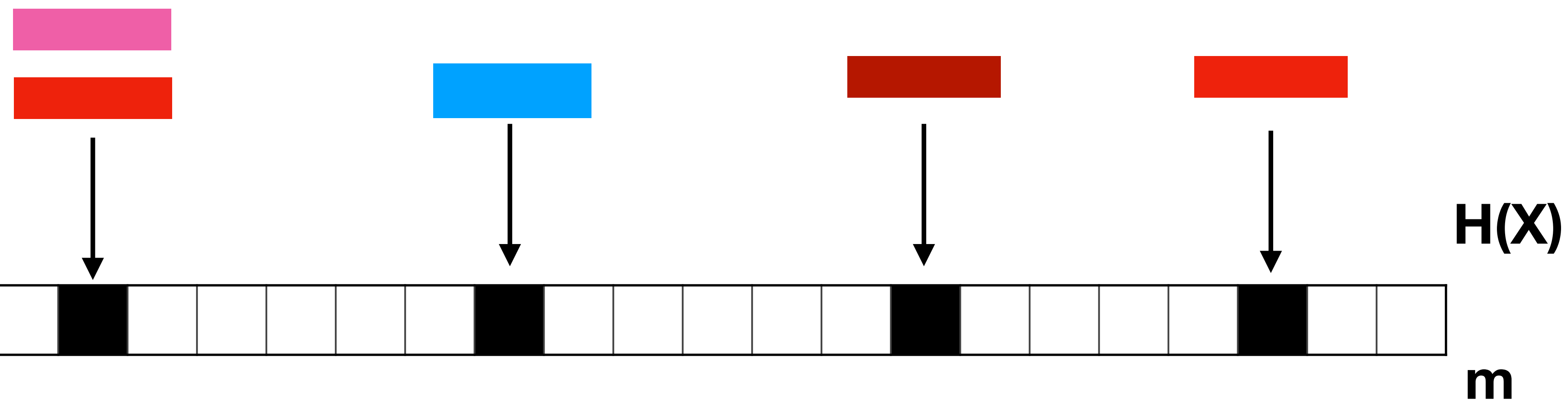
Bloom Filters efficiently encode sets

A Bloom Filter is a length *m* bit-vector and associated hash function(s) H(X)

Hash Function H(X) takes in an arbitrary element and returns an integer

Ex: H(🟦) = 12

Set elements are inserted by setting an associated bit to 1



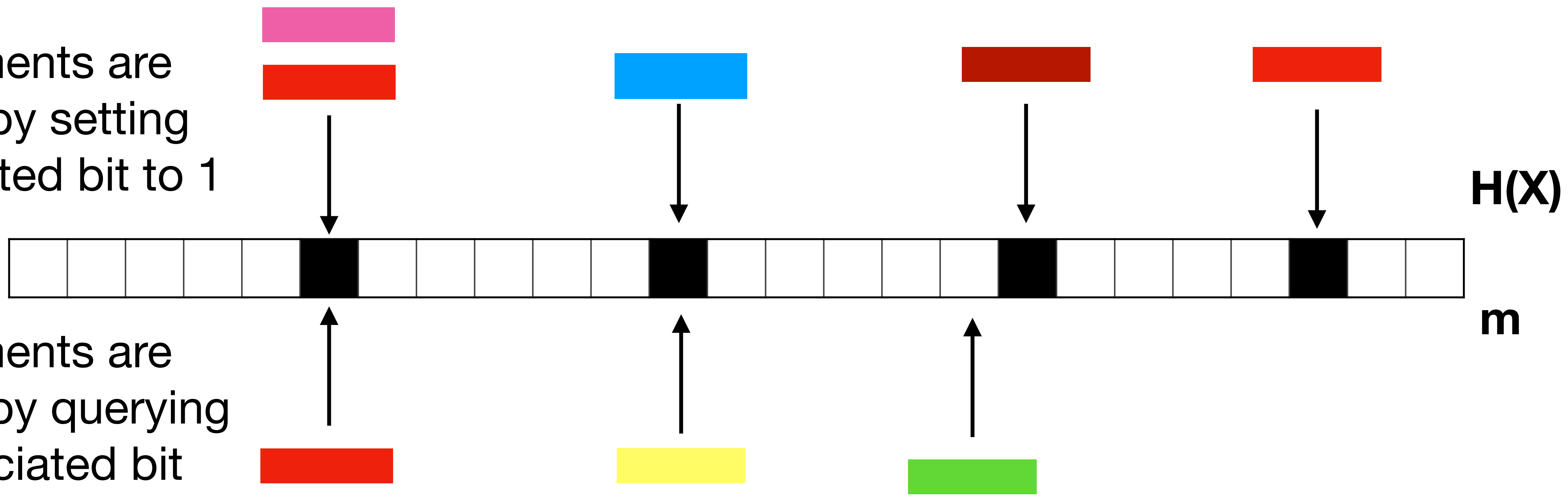
Space/Time Trade-offs in Hash Coding with Allowable Errors.
Burton Bloom. (1970) *Communications of the ACM.*

Bloom Filters efficiently encode sets

A Bloom Filter is a length *m* bit-vector and associated hash function(s) H(X)

Hash Function H(X) takes in an arbitrary element and returns an integer

Ex: H(🟦) = 12



Set elements are inserted by setting an associated bit to 1

Set elements are looked up by querying the associated bit

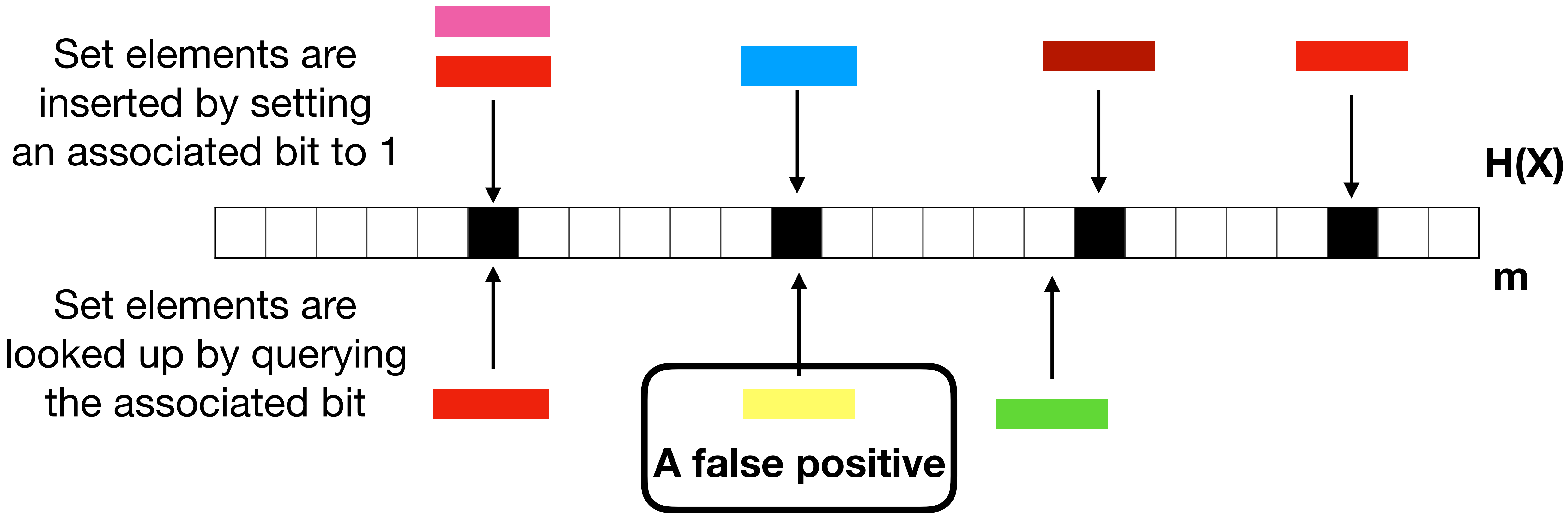
Space/Time Trade-offs in Hash Coding with Allowable Errors.
Burton Bloom. (1970) *Communications of the ACM.*

Bloom Filters efficiently encode sets

A Bloom Filter is a length *m* bit-vector and associated hash function(s) H(X)

Hash Function H(X) takes in an arbitrary element and returns an integer

Ex: H(🟦) = 12



Set elements are inserted by setting an associated bit to 1

Set elements are looked up by querying the associated bit

A Bloom Filter is probabilistic data structure!

Space/Time Trade-offs in Hash Coding with Allowable Errors.
Burton Bloom. (1970) *Communications of the ACM.*

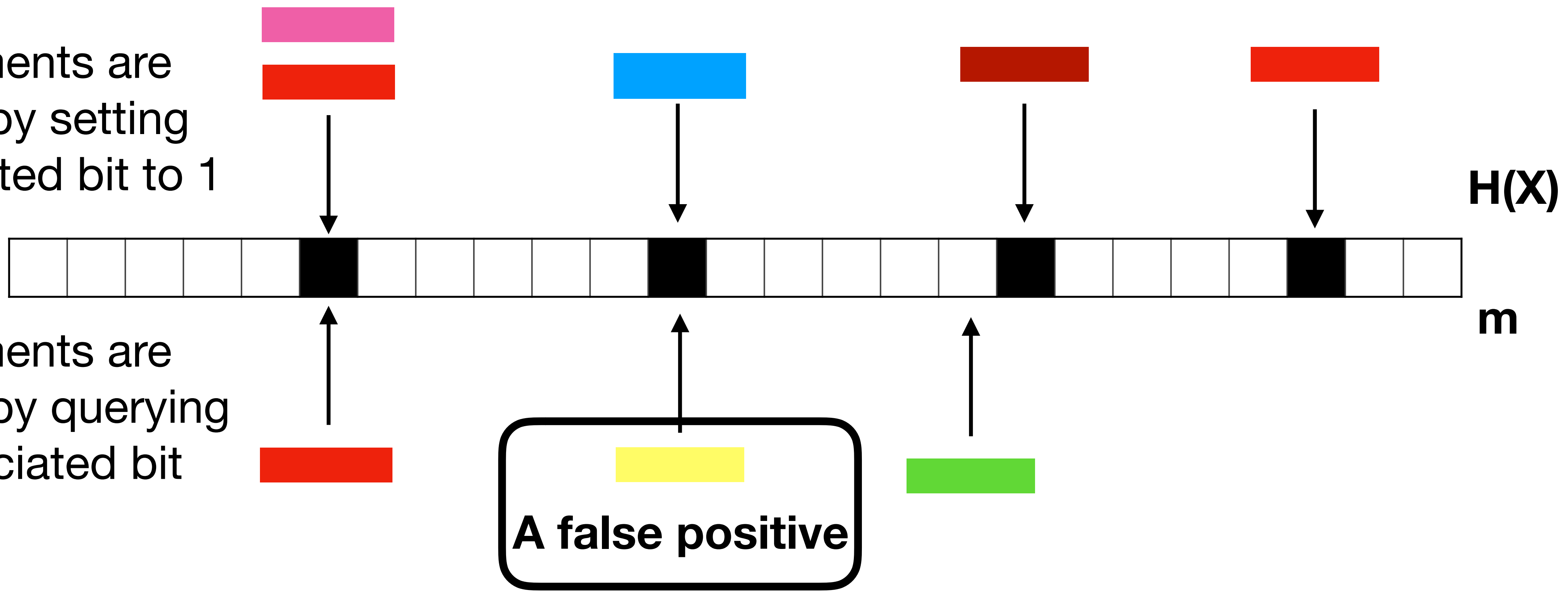
Bloom Filters efficiently encode sets

A Bloom Filter is a length *m* bit-vector and associated hash function(s) H(X)

Hash Function H(X) takes in an arbitrary element and returns an integer

Ex: H(🟦) = 12

Set elements are inserted by setting an associated bit to 1



Set elements are looked up by querying the associated bit

A Bloom Filter is probabilistic data structure!

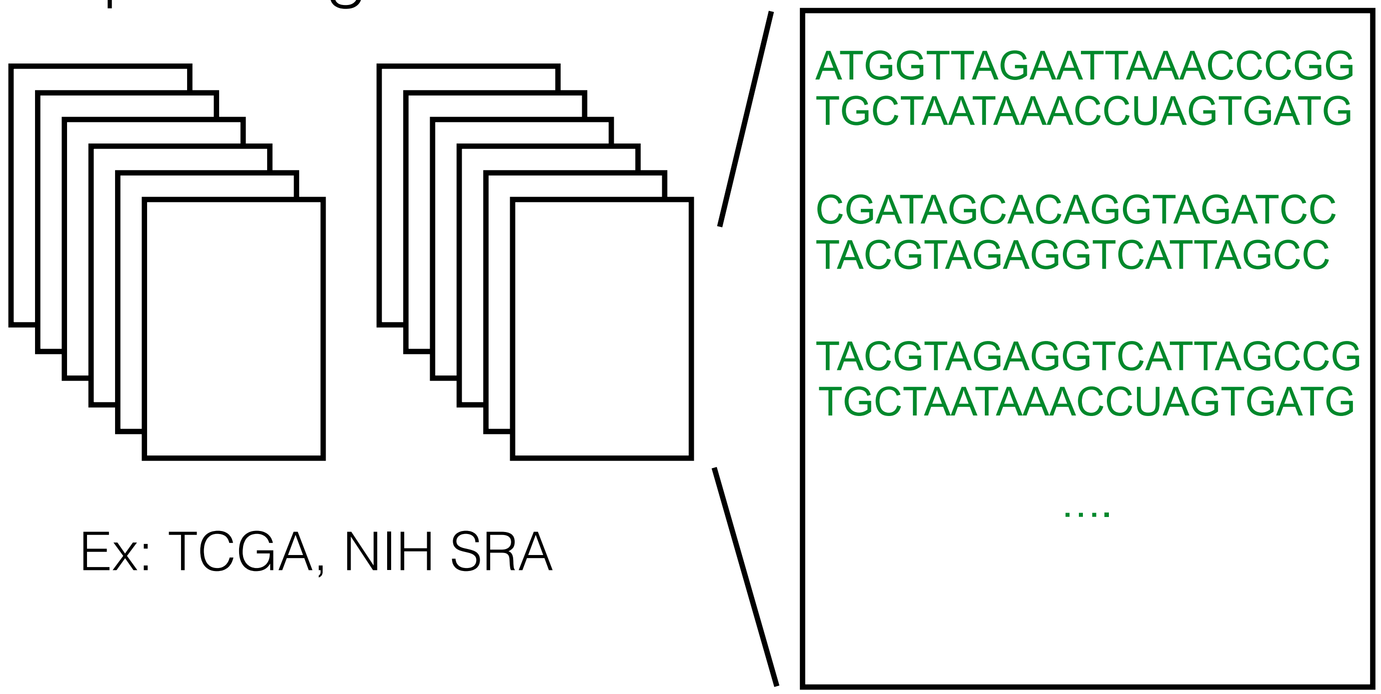
Stores arbitrary sets and supports O(1) insertion and membership testing

Space/Time Trade-offs in Hash Coding with Allowable Errors.
Burton Bloom. (1970) *Communications of the ACM.*

The “Containment” Query

Input:

- Set of individual sequencing studies



ATGGTTAGAATTAAACCCGG
TGCTAATAAACCUAGTGATG

CGATAGCACAGGTAGATCC
TACGTAGAGGTCATTAGCC

TACGTAGAGGTCATTAGCCG
TGCTAATAAACCUAGTGATG

....

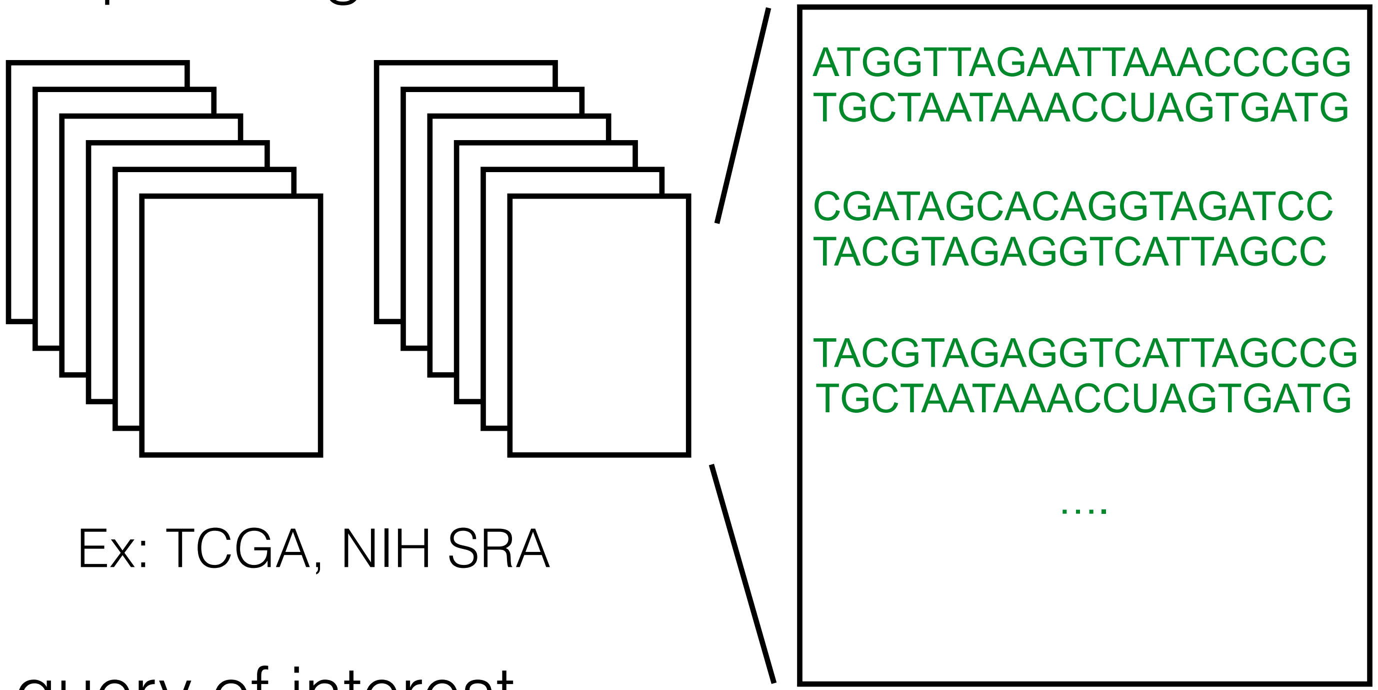
Ex: TCGA, NIH SRA

Each study contains a set of raw reads

The "Containment" Query

Input:

- Set of individual sequencing studies



ATGGTTAGAATTAAACCCGG
TGCTAATAAACCUAGTGATG

CGATAGCACAGGTAGATCC
TACGTAGAGGTCATTAGCC

TACGTAGAGGTCATTAGCCG
TGCTAATAAACCUAGTGATG

....

Ex: TCGA, NIH SRA

Each study contains a set of raw reads

- A query of interest

ATGGTTAGAATTAAACCTGGATC
TGCTAATAAACCUAGTGATGATG
CGATAGCACAGGTAGATCCAGT
TACGTAGAGGTCATTAGCCGTAT
TGCTAATAAACCTAGTGATGATT
CGATAGCGTAGAGGTCATTAGC
CTTGTGCTAATAAACAGGTAGA
TCCGTATACGTAGAGGTCATTA
CCTTGTGCTAATAAACCTAGTG

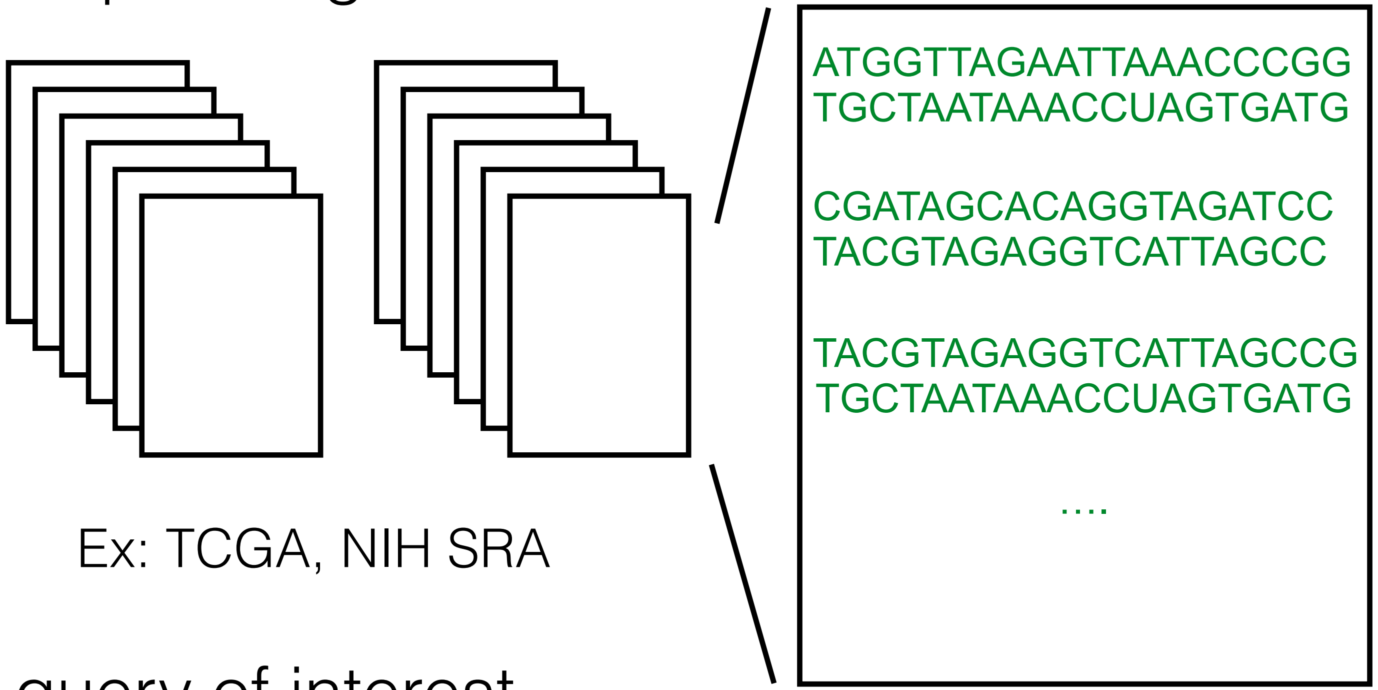
Ex: A novel transcript

- θ, the definition of containment

The “Containment” Query

Input:

- Set of individual sequencing studies



ATGGTTAGAATTAAACCCGG
TGCTAATAAACCUAGTGATG

CGATAGCACAGGTAGATCC
TACGTAGAGGTCATTAGCC

TACGTAGAGGTCATTAGCCG
TGCTAATAAACCUAGTGATG

....

Ex: TCGA, NIH SRA

Each study contains a set of raw reads

- A query of interest

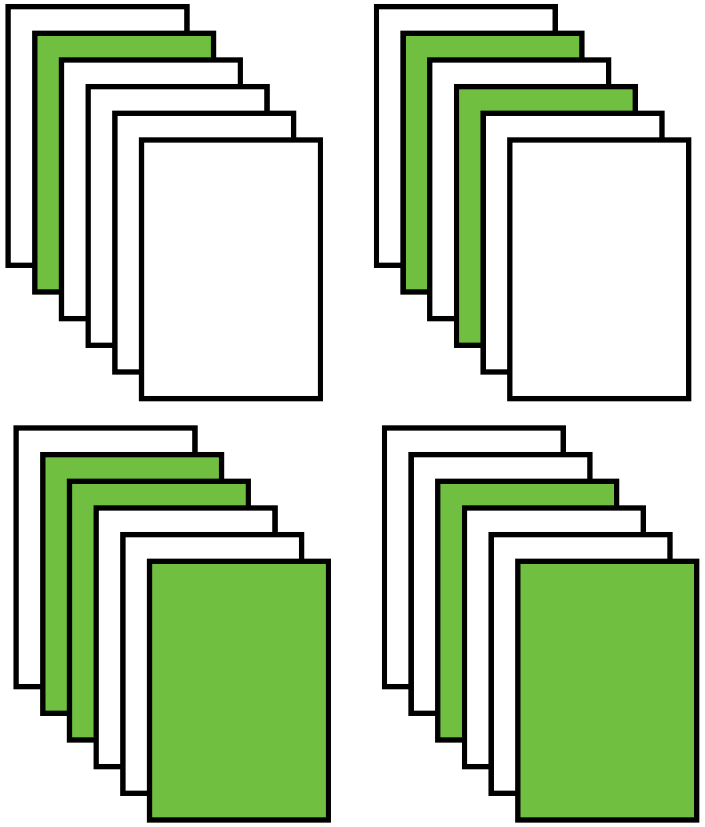
ATGGTTAGAATTAAACCTGGATC
TGCTAATAAACCUAGTGATGATG
CGATAGCACAGGTAGATCCAGT
TACGTAGAGGTCATTAGCCGTAT
TGCTAATAAACCTAGTGATGATT
CGATAGCGTAGAGGTCATTAGC
CTTGTGCTAATAAACAGGTAGA
TCCGTATACGTAGAGGTCATTA
CCTTGTGCTAATAAACCTAGTG

Ex: A novel transcript

- θ, the definition of containment

Output:

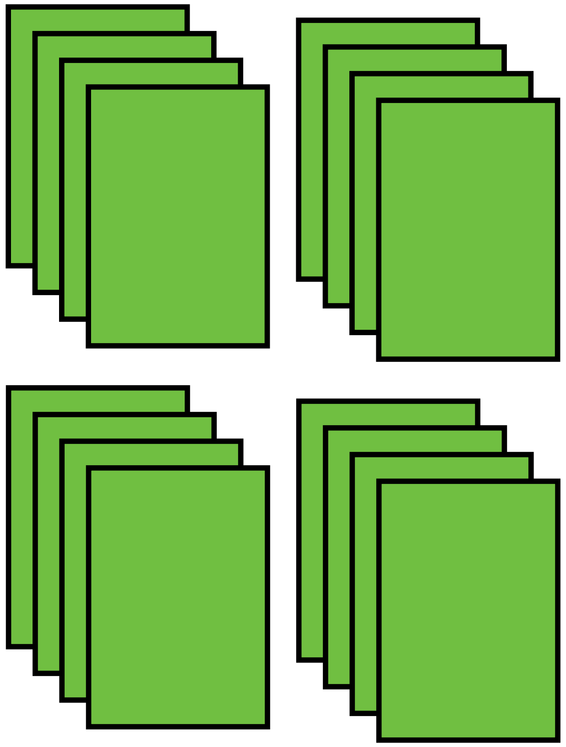
All studies whose **read sets can cover θ fraction of the query**



Finding **relevant experiments** is the first step in many large-scale analyses.

All studies whose reads cover the query

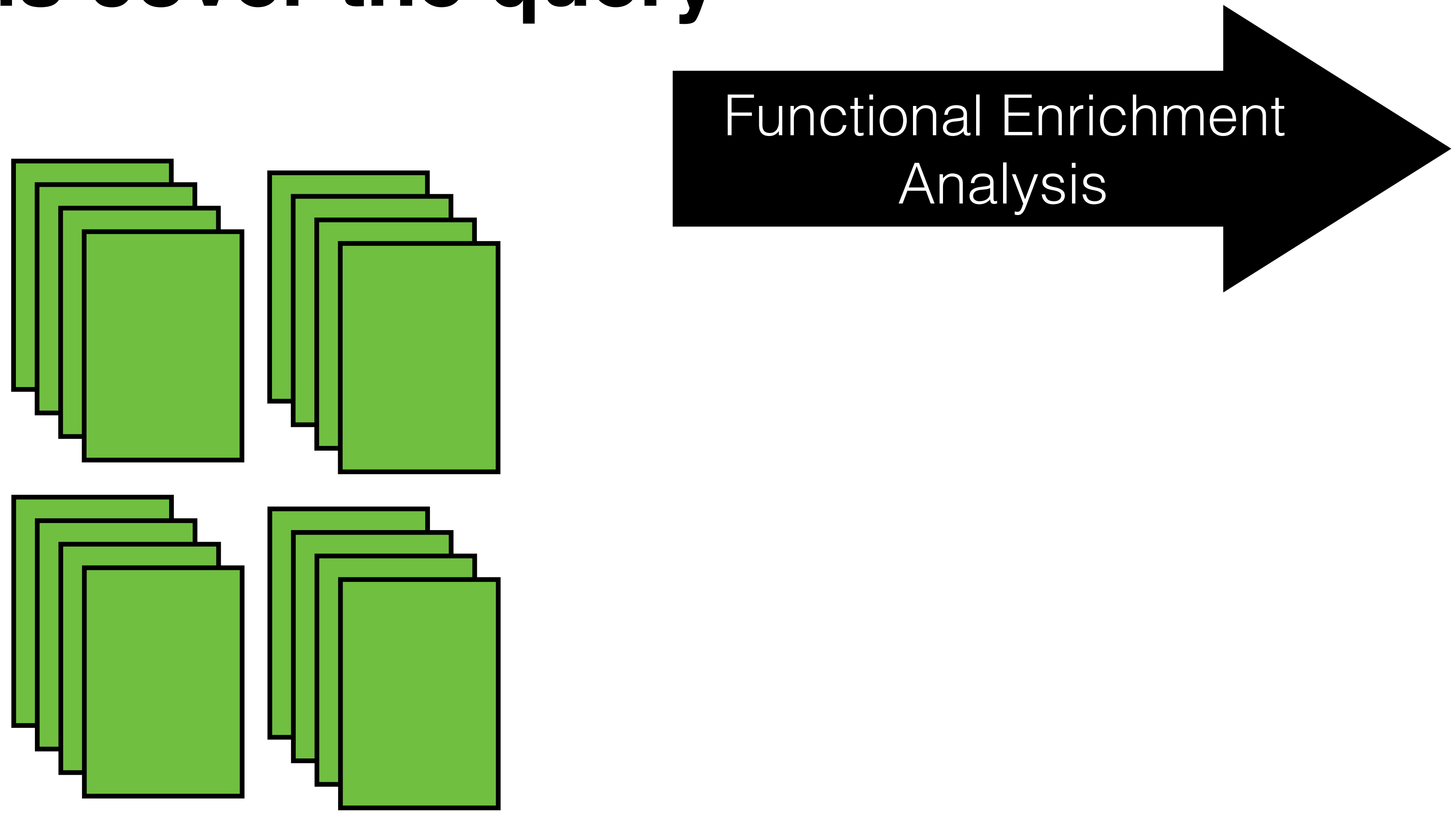
Output



Finding **relevant experiments** is the first step in many large-scale analyses.

All studies whose reads cover the query

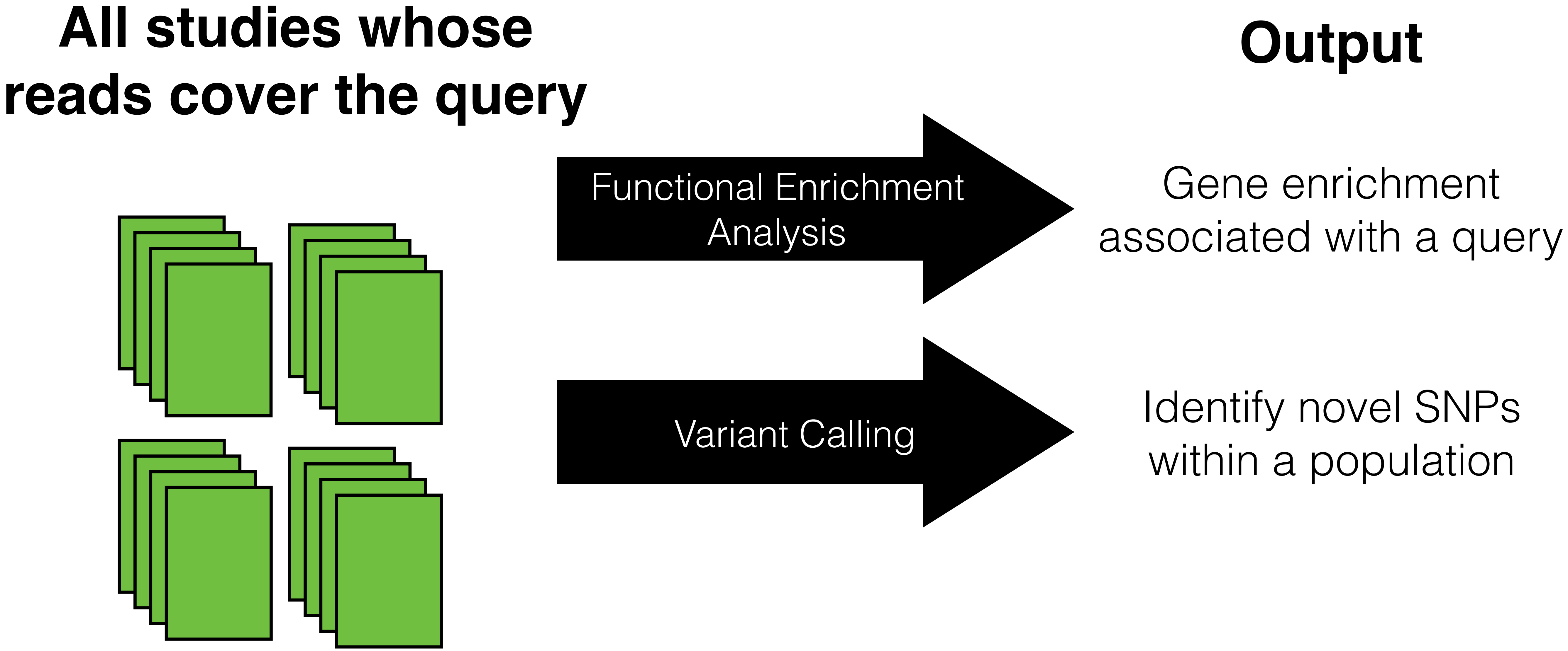
Output



Functional Enrichment Analysis

Gene enrichment associated with a query

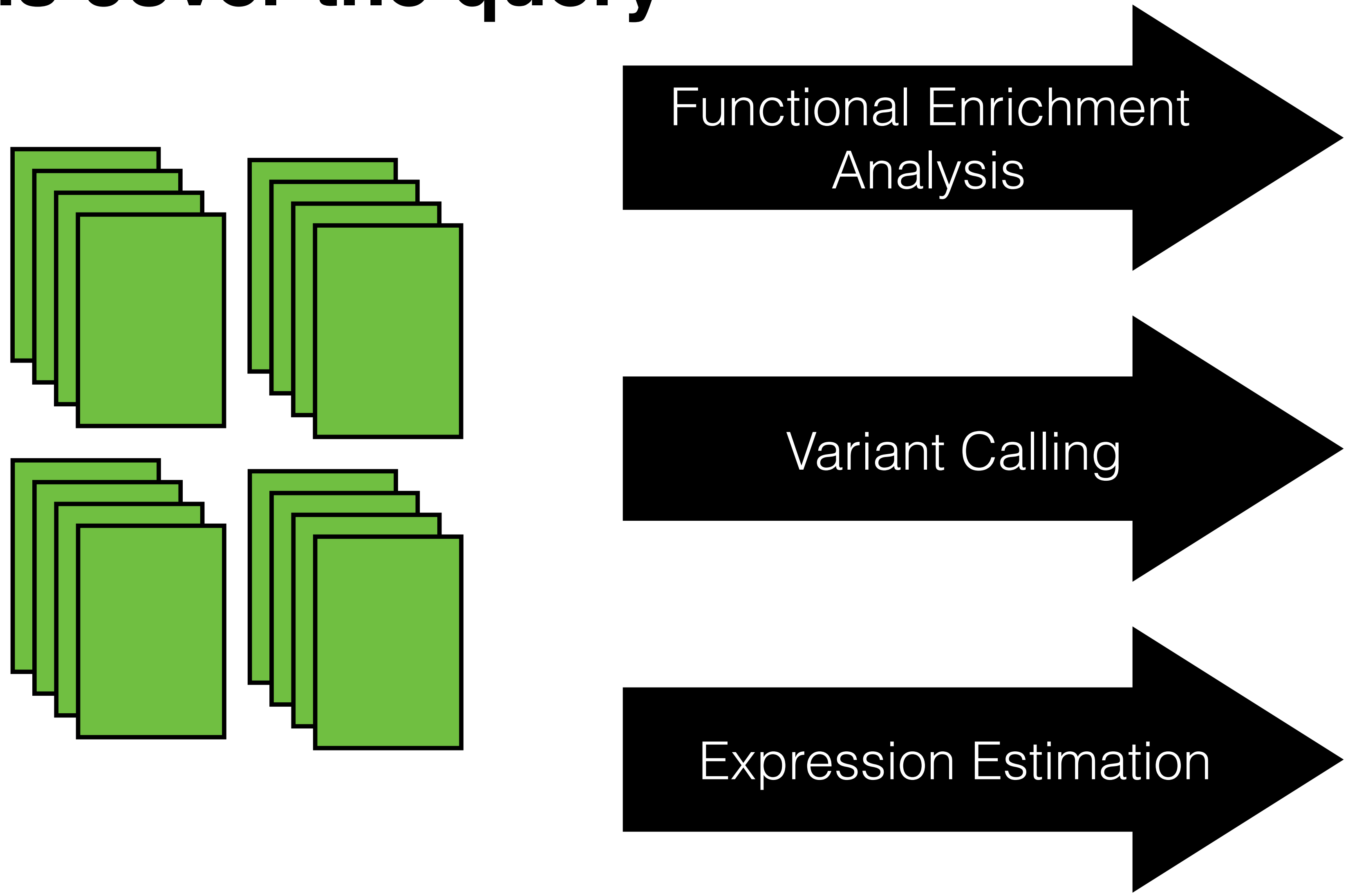
Finding **relevant experiments** is the first step in many large-scale analyses.


All studies whose reads cover the query
Output
Functional Enrichment Analysis
Gene enrichment associated with a query
Variant Calling
Identify novel SNPs within a population

Finding **relevant experiments** is the first step in many large-scale analyses.

All studies whose reads cover the query

Output

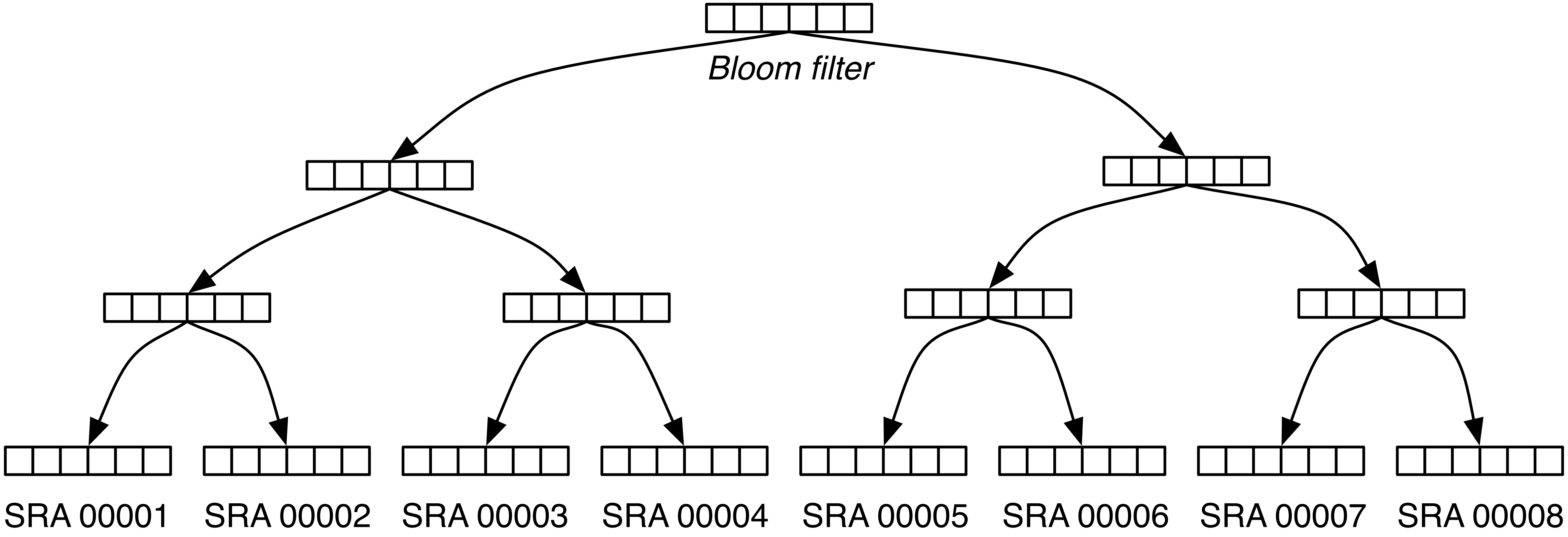


Functional Enrichment Analysis → Gene enrichment associated with a query

Variant Calling → Identify novel SNPs within a population

Expression Estimation → Expression levels in a population

The Sequence Bloom Tree



Fast search of thousands of short-read sequencing experiments.
Brad Solomon and Carl Kingsford. (2016) *Nature Biotechnology.*

The Sequence Bloom Tree

1) Sequence reads are broken down into kmers

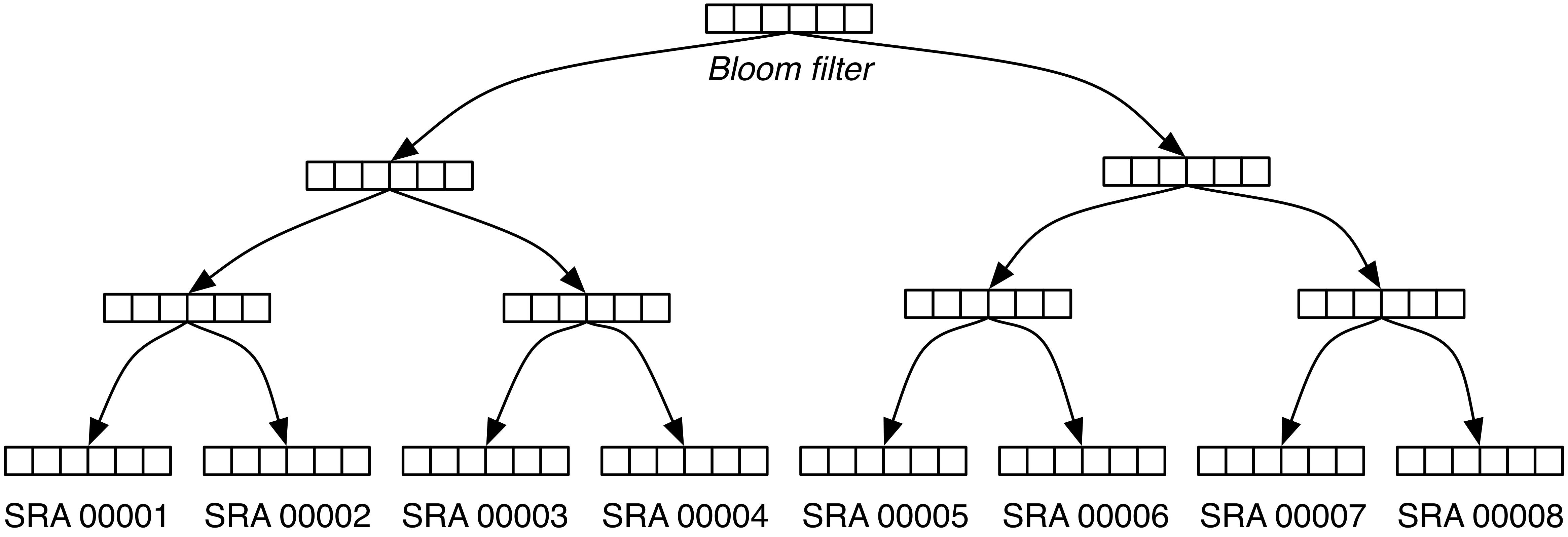
ATGGTTAGAATTAAA

ATG TTA AAT AAA

TGG TAG ATT

GGT AGA TTA

GTT GAA TAA



Fast search of thousands of short-read sequencing experiments.
Brad Solomon and Carl Kingsford. (2016) *Nature Biotechnology.*

The Sequence Bloom Tree

1) Sequence reads are broken down into kmers

ATGGTTAGAATTAAA

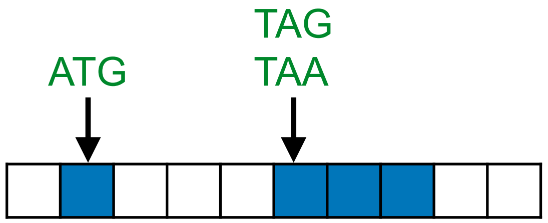
ATG TTA AAT AAA

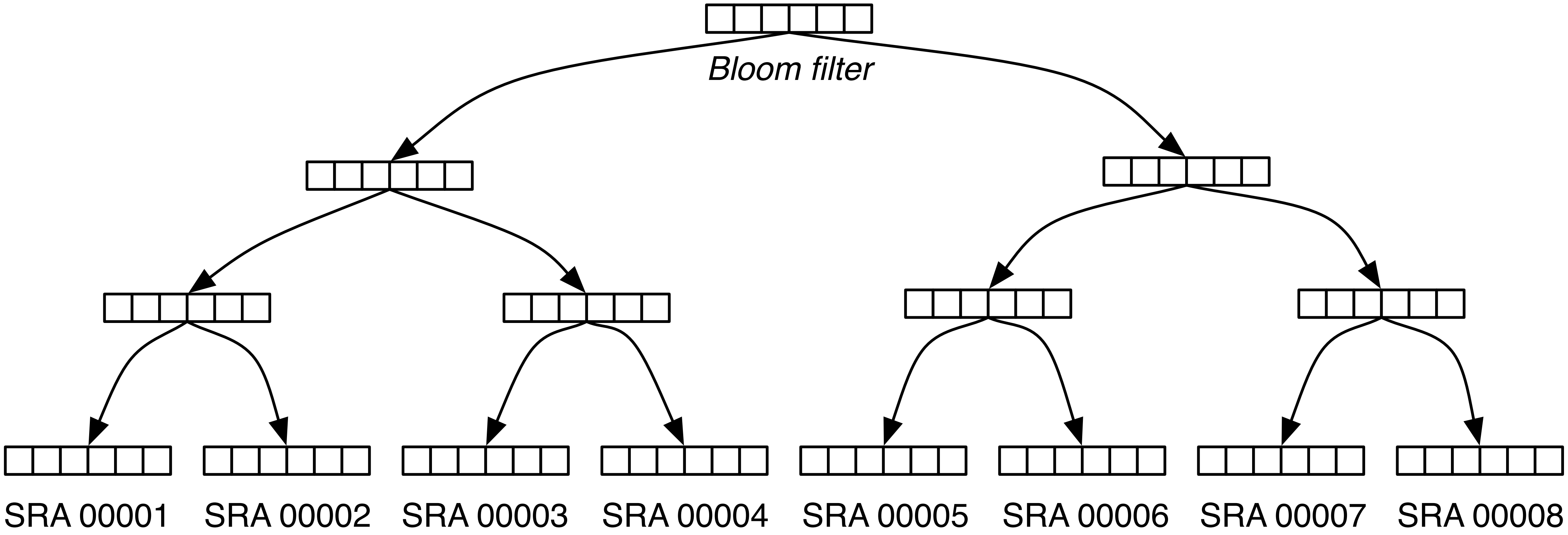
TGG TAG ATT

GGT AGA TTA

GTT GAA TAA

2) The set of all kmers in a study are stored in a **bloom filter**





Fast search of thousands of short-read sequencing experiments.
Brad Solomon and Carl Kingsford. (2016) *Nature Biotechnology.*

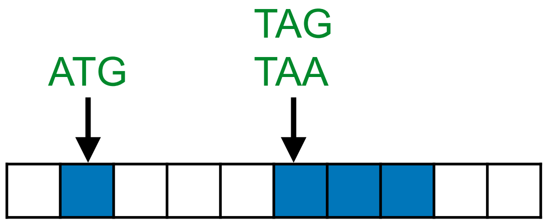
The Sequence Bloom Tree

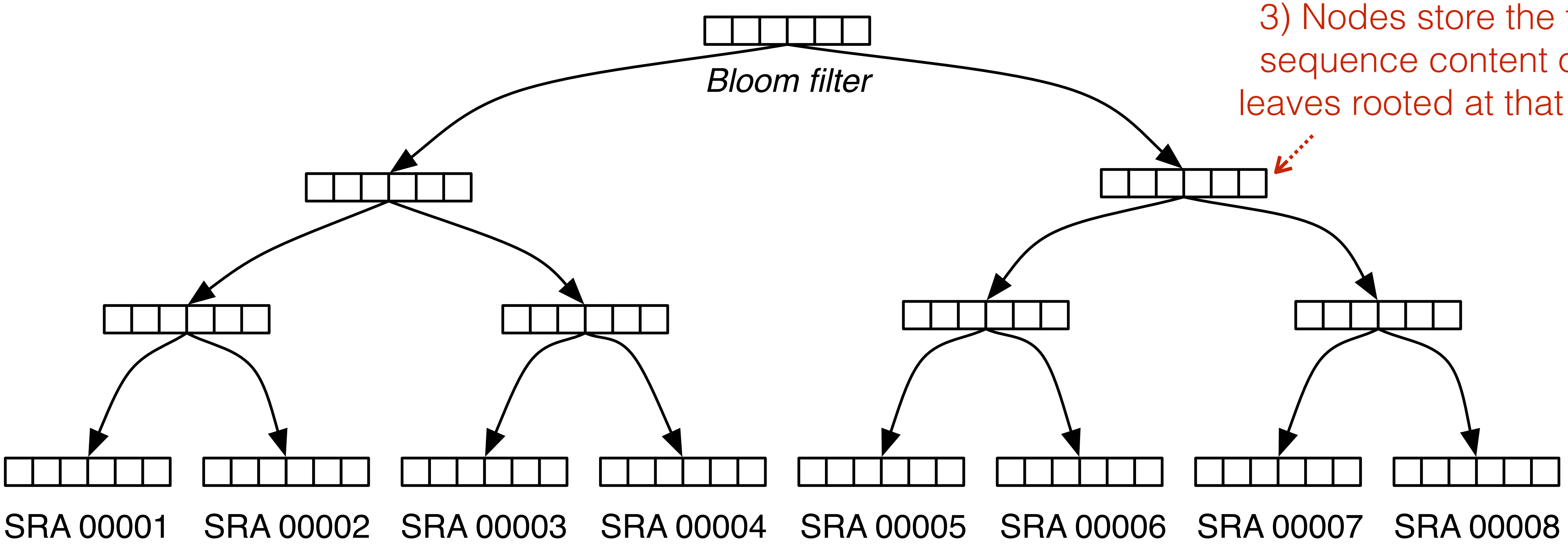
1) Sequence reads are broken down into kmers

ATGGTTAGAATTAAA

ATG TTA AAT AAA
TGG TAG ATT
GGT AGA TTA
GTT GAA TAA

2) The set of all kmers in a study are stored in a **bloom filter**





3) Nodes store the total sequence content of all leaves rooted at that node

Fast search of thousands of short-read sequencing experiments.
Brad Solomon and Carl Kingsford. (2016) *Nature Biotechnology.*

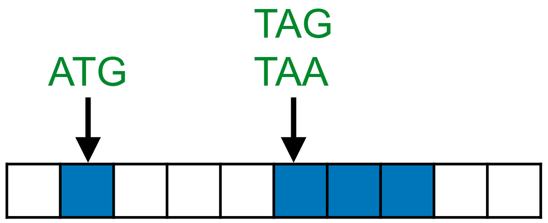
The Sequence Bloom Tree

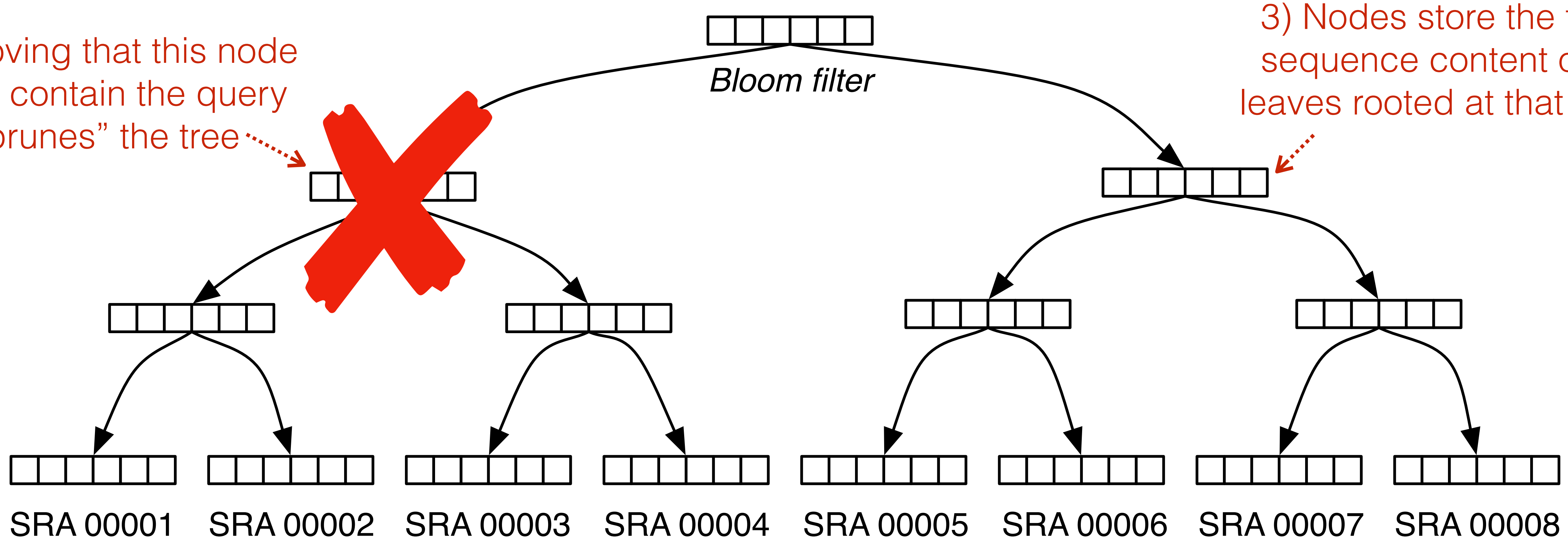
1) Sequence reads are broken down into kmers

ATGGTTAGAATTAAA

ATG TTA AAT AAA
TGG TAG ATT
GGT AGA TTA
GTT GAA TAA

2) The set of all kmers in a study are stored in a **bloom filter**





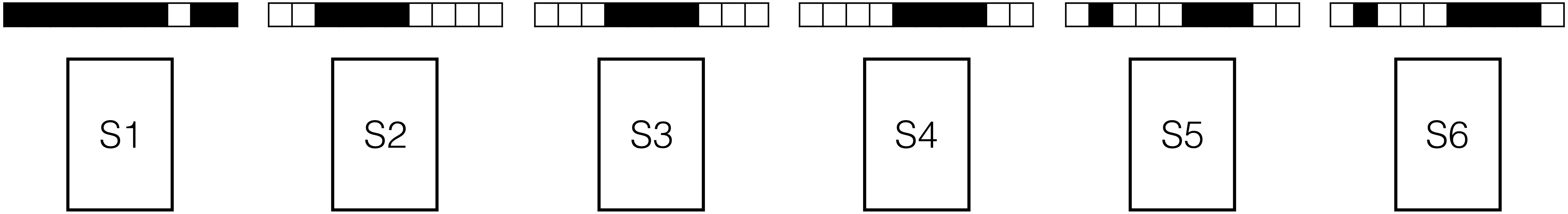
4) Proving that this node can't contain the query "prunes" the tree

3) Nodes store the total sequence content of all leaves rooted at that node

Fast search of thousands of short-read sequencing experiments.
Brad Solomon and Carl Kingsford. (2016) *Nature Biotechnology.*

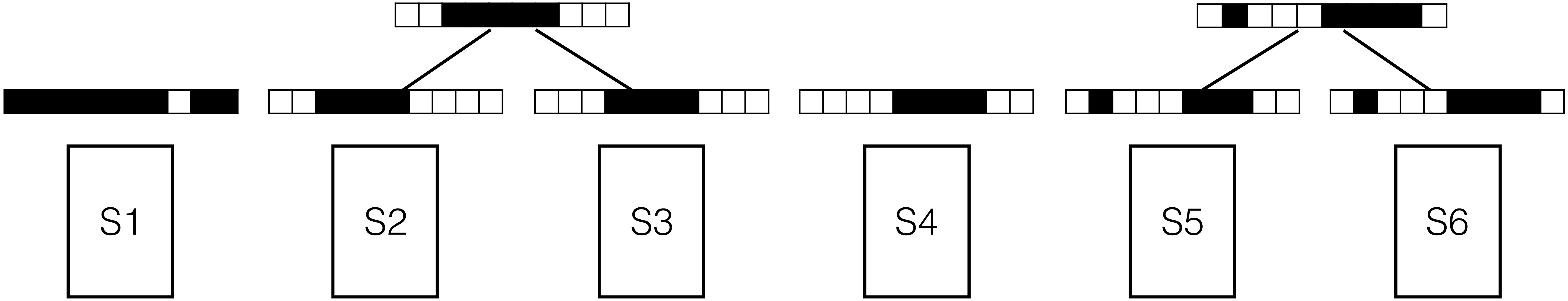
Conceptual Construction

Bit-wise union of bloom filter produces bloom filter with total kmer content of each file



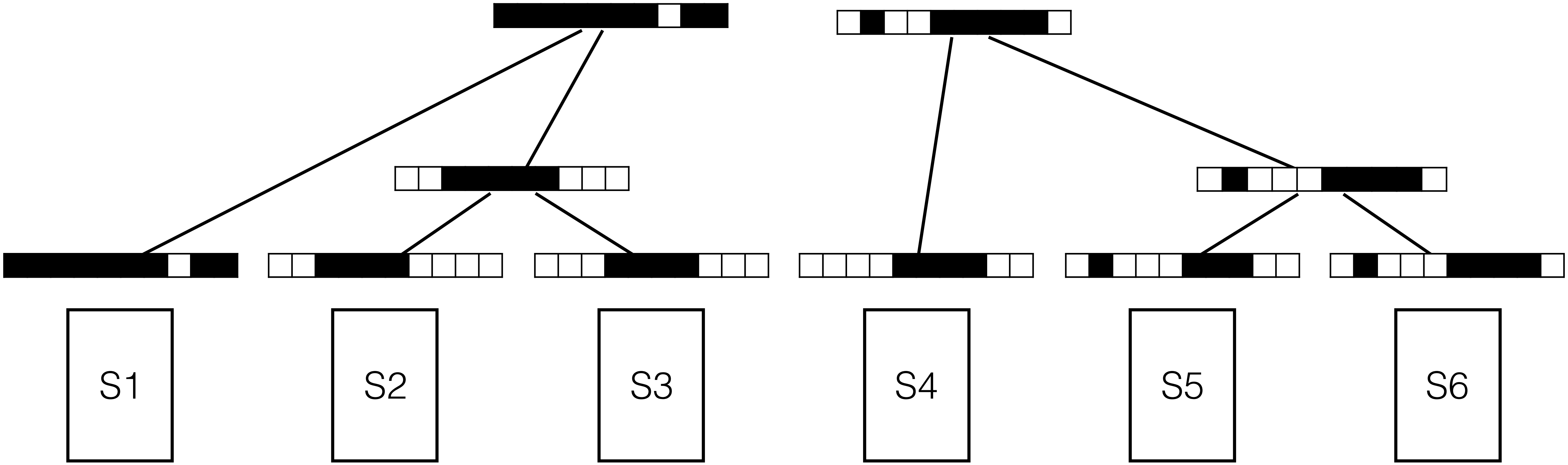
Conceptual Construction

Bit-wise union of bloom filter produces bloom filter with total kmer content of each file



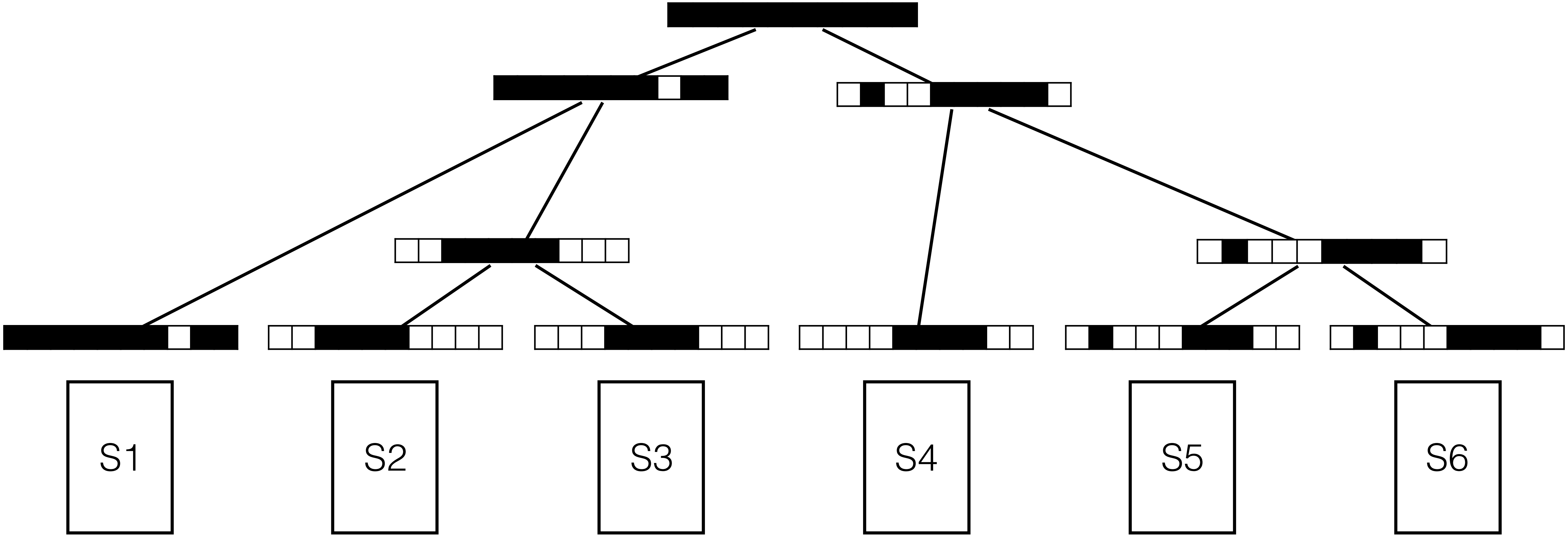
Conceptual Construction

Bit-wise union of bloom filter produces bloom filter with total kmer content of each file



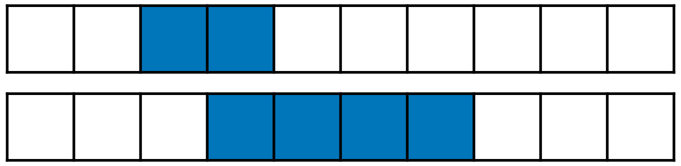
Conceptual Construction

Bit-wise union of bloom filter produces bloom filter with total kmer content of each file

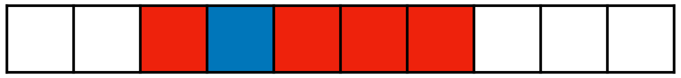


Similarity Metrics for Bloom Filters

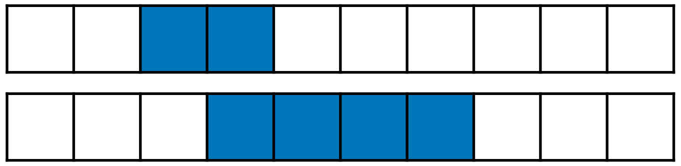
Hamming distance: Number of ‘substitutions’ to transform one vector to another



Hamming Distance = 4

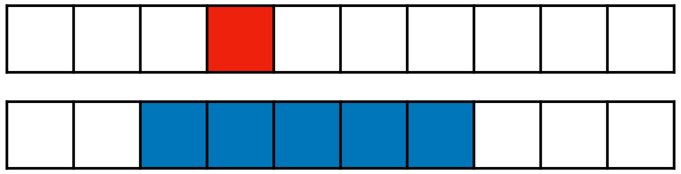


Jaccard similarity: Intersection of 1-bits over **union** of 1-bits



Intersection = 1

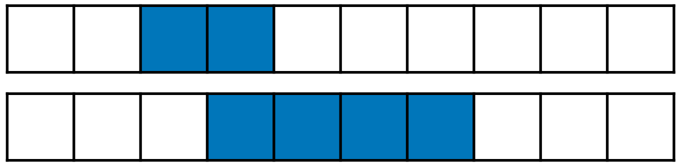
Union = 5



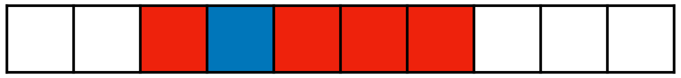
Similarity = 1/5

Similarity Metrics for Bloom Filters

Hamming distance: Number of ‘substitutions’ to transform one vector to another

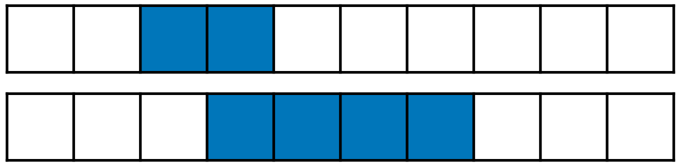


Hamming Distance = 4



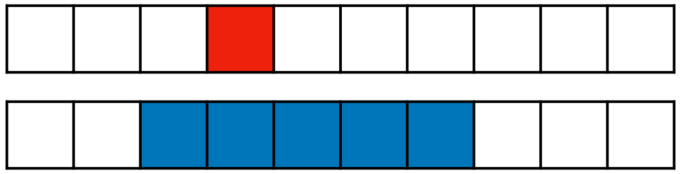
Compare these metrics by **normalizing** (and **inverting**)

Jaccard similarity: **Intersection** of 1-bits over **union** of 1-bits



Intersection = 1

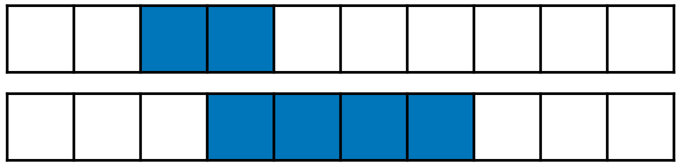
Union = 5



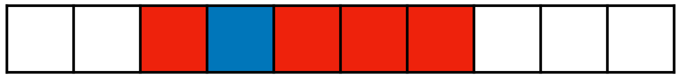
Similarity = 1/5

Similarity Metrics for Bloom Filters

Hamming distance: Number of ‘substitutions’ to transform one vector to another



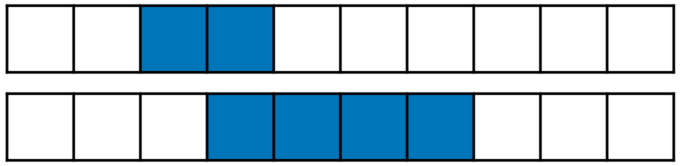
Hamming Distance = 4



Compare these metrics by **normalizing** (and **inverting**)

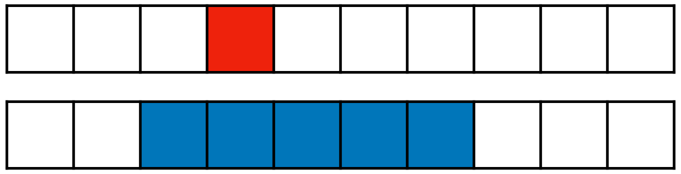
Distance / Length = 4 / 10

Jaccard similarity: **Intersection** of 1-bits over **union** of 1-bits



Intersection = 1

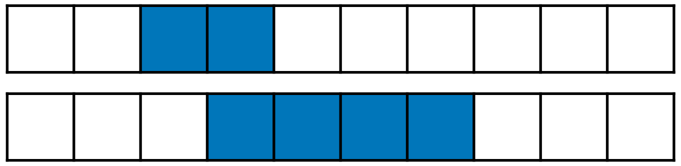
Union = 5



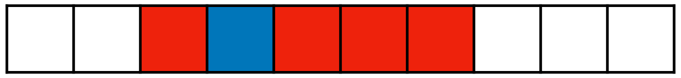
Similarity = 1/5

Similarity Metrics for Bloom Filters

Hamming distance: Number of ‘substitutions’ to transform one vector to another



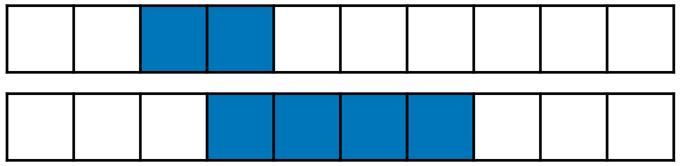
Hamming Distance = 4



Compare these metrics by **normalizing** (and **inverting**)

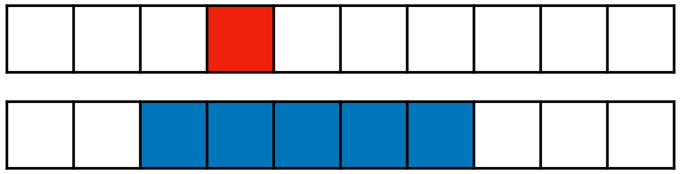
Distance / Length = 4 / 10 (Similarity = 6/10)

Jaccard similarity: Intersection of 1-bits over **union** of 1-bits



Intersection = 1

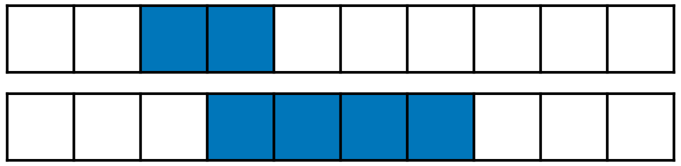
Union = 5



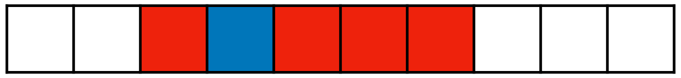
Similarity = 1/5

Similarity Metrics for Bloom Filters

Hamming distance: Number of ‘substitutions’ to transform one vector to another



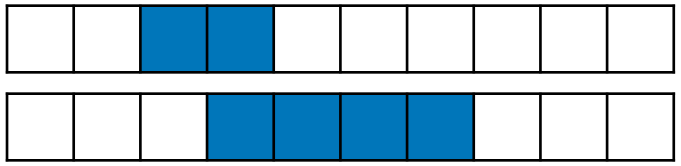
Hamming Distance = 4



Compare these metrics by **normalizing** (and **inverting**)

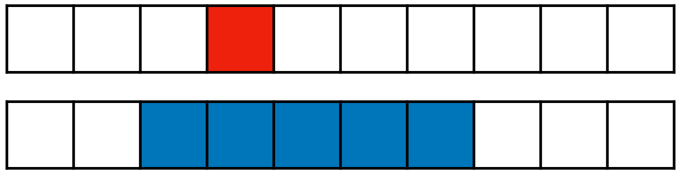
Distance / Length = 4 / 10 (Similarity = 6/10)

Jaccard similarity: Intersection of 1-bits over **union** of 1-bits



Intersection = 1

Union = 5

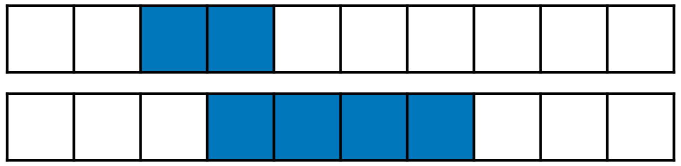


Similarity = 1/5

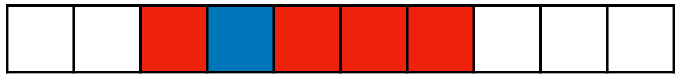
The **Hamming** similarity does not correlate with sequence content
The **Jaccard** similarity correlates with sequence content

Similarity Metrics for Bloom Filters

Hamming distance: Number of ‘substitutions’ to transform one vector to another



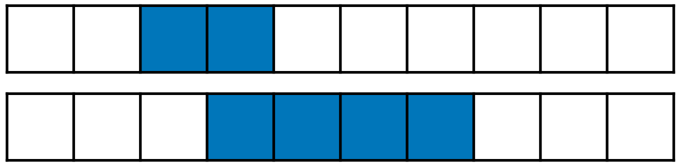
Hamming Distance = 4



Compare these metrics by **normalizing** (and **inverting**)

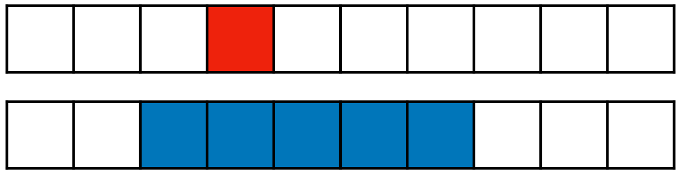
Distance / Length = 4 / 10 (Similarity = 6/10)

Jaccard similarity: Intersection of 1-bits over **union** of 1-bits



Intersection = 1

Union = 5



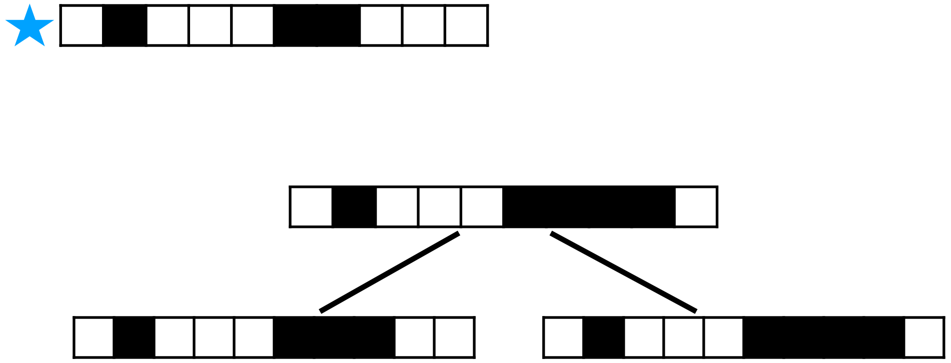
Similarity = 1/5

The **Hamming** similarity does not correlate with sequence content
The **Jaccard** similarity correlates with sequence content

Why?

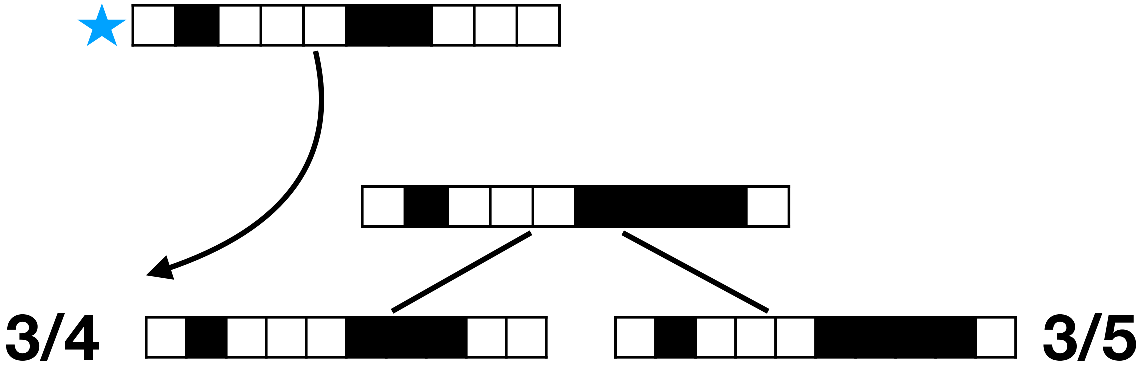
Building an SBT

Top Down Streaming: Given an existing tree (which may be empty) and a new bloom filter, how can we build a tree?



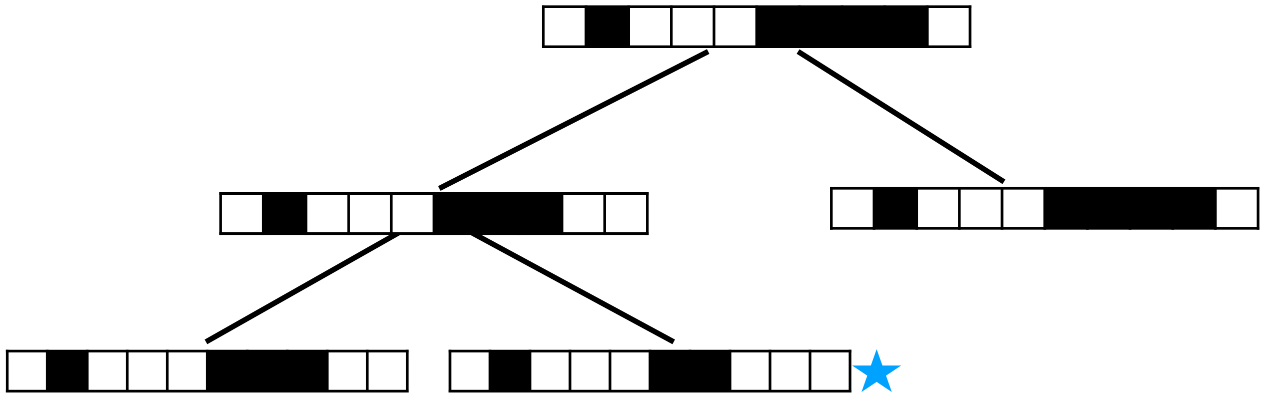
Building an SBT

Top Down Streaming: Given an existing tree (which may be empty) and a new bloom filter, how can we build a tree?



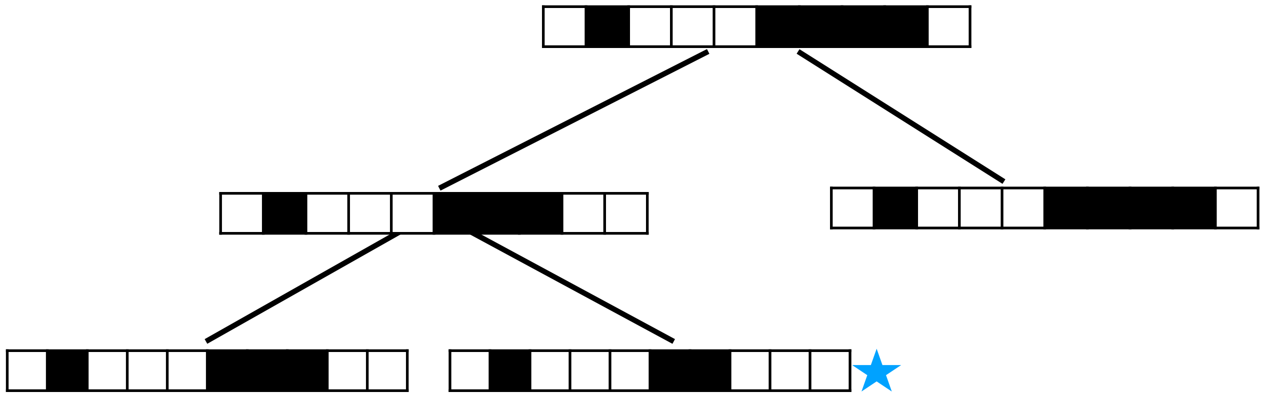
Building an SBT

Top Down Streaming: Given an existing tree (which may be empty) and a new bloom filter, how can we build a tree?



Building an SBT

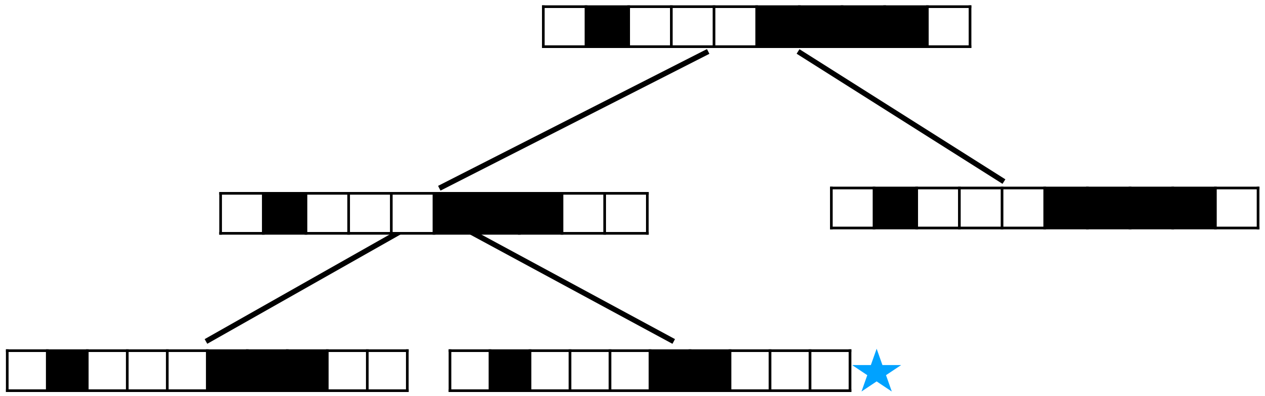
Top Down Streaming: Given an existing tree (which may be empty) and a new bloom filter, how can we build a tree?



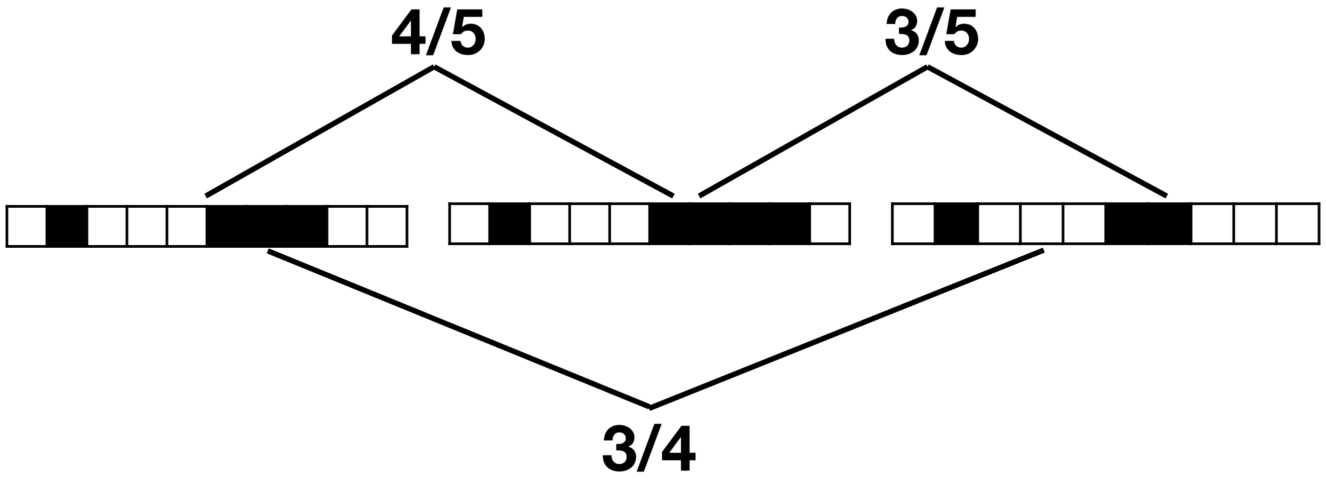
Global Pairwise Construction: Given all bloom filters, how can we build a tree?

Building an SBT

Top Down Streaming: Given an existing tree (which may be empty) and a new bloom filter, how can we build a tree?

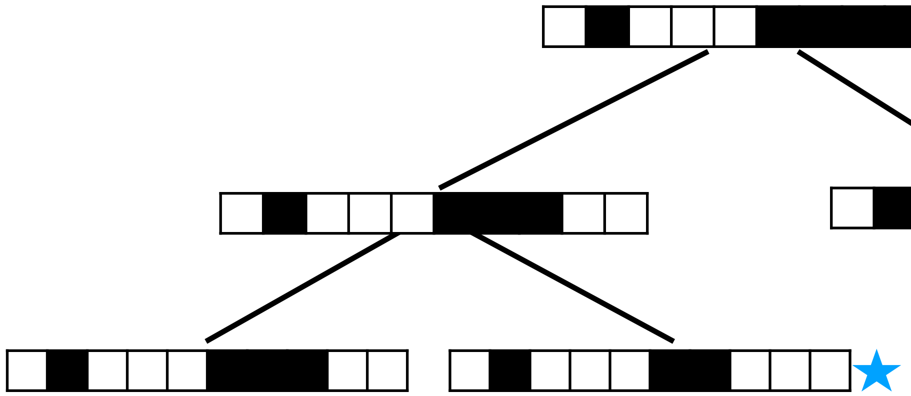


Global Pairwise Construction: Given all bloom filters, how can we build a tree?

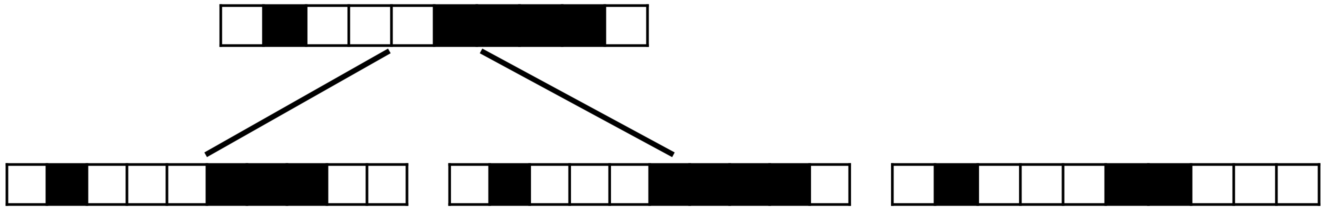


Building an SBT

Top Down Streaming: Given an existing tree (which may be empty) and a new bloom filter, how can we build a tree?

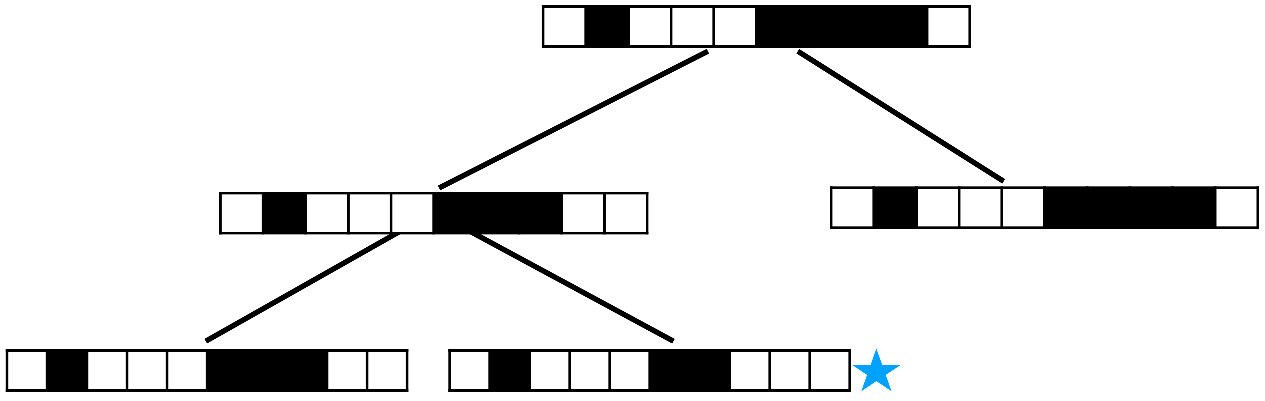


Global Pairwise Construction: Given all bloom filters, how can we build a tree?

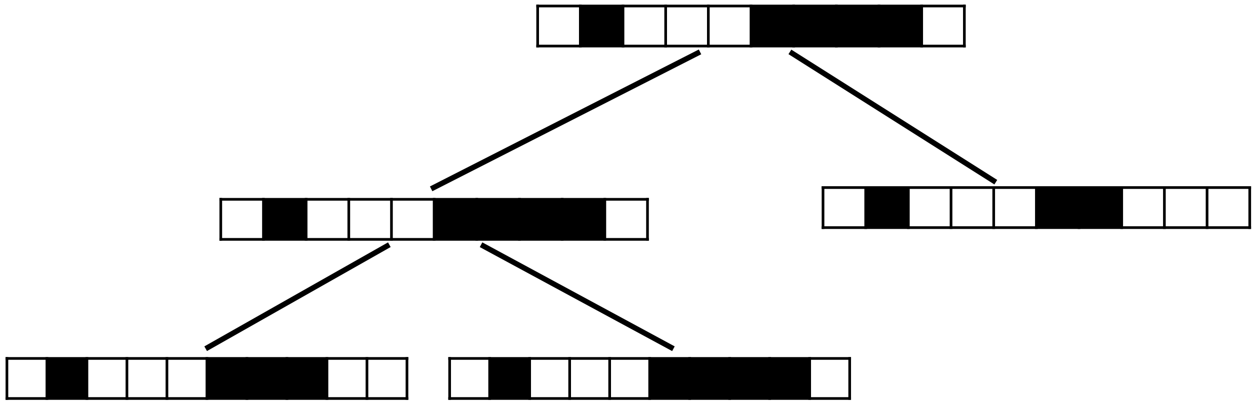


Building an SBT

Top Down Streaming: Given an existing tree (which may be empty) and a new bloom filter, how can we build a tree?

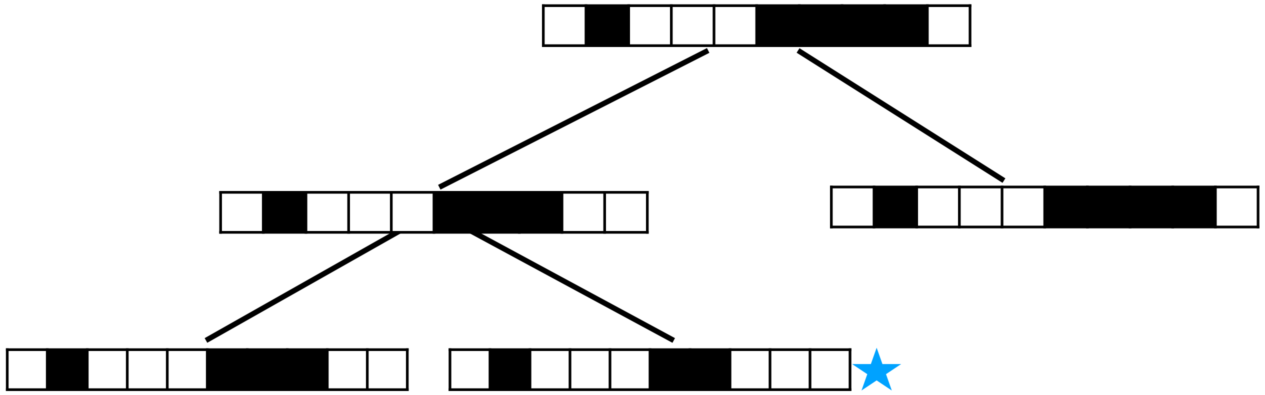


Global Pairwise Construction: Given all bloom filters, how can we build a tree?

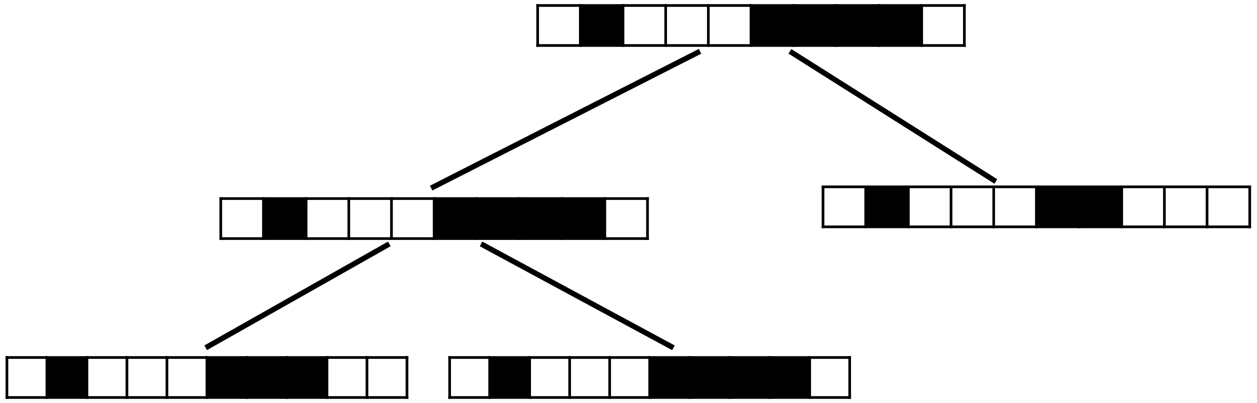


Building an SBT

Top Down Streaming: Given an existing tree (which may be empty) and a new bloom filter, how can we build a tree?



Global Pairwise Construction: Given all bloom filters, how can we build a tree?

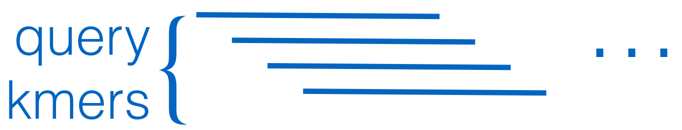


What are the benefits?
What are the weaknesses?

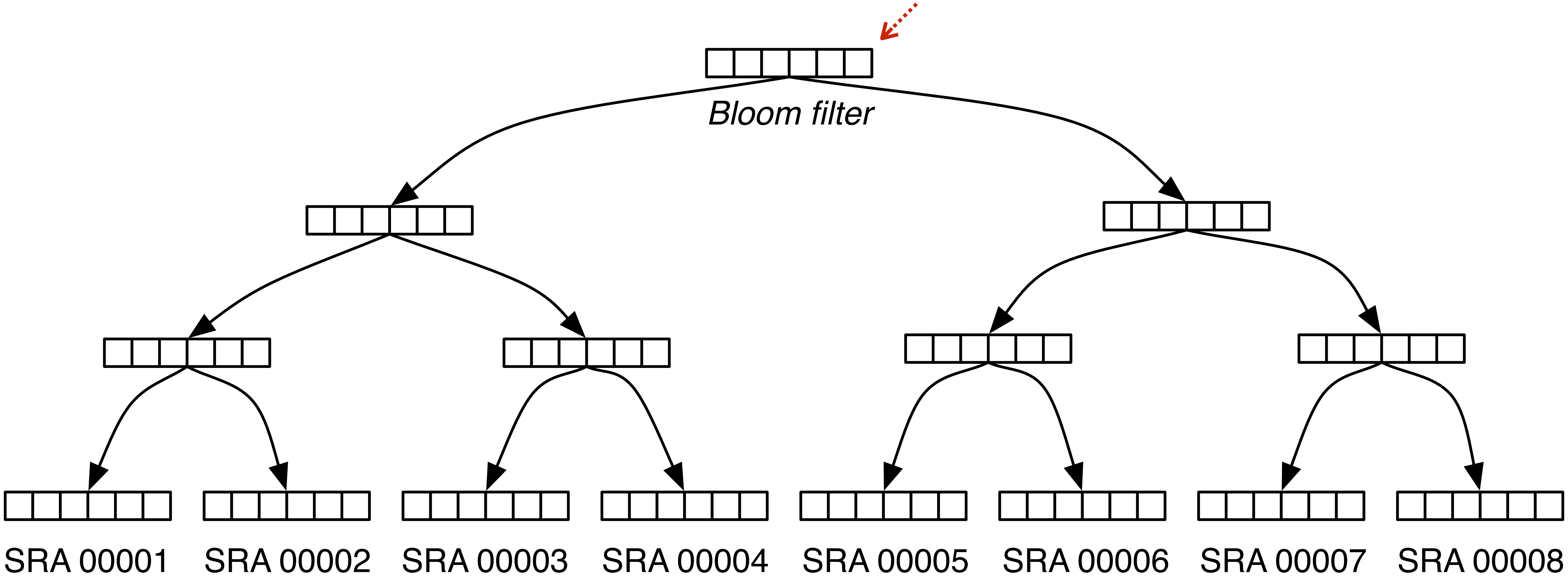
Querying a Sequence Bloom Tree

Good approximate sequence match == lots of shared kmers:

ATGGTTAGAATTAAACCTGGTTCTGCTAATAAACCTAGTGATGAT



Are ≥ θ fraction of query kmers ∊ this Bloom filter?

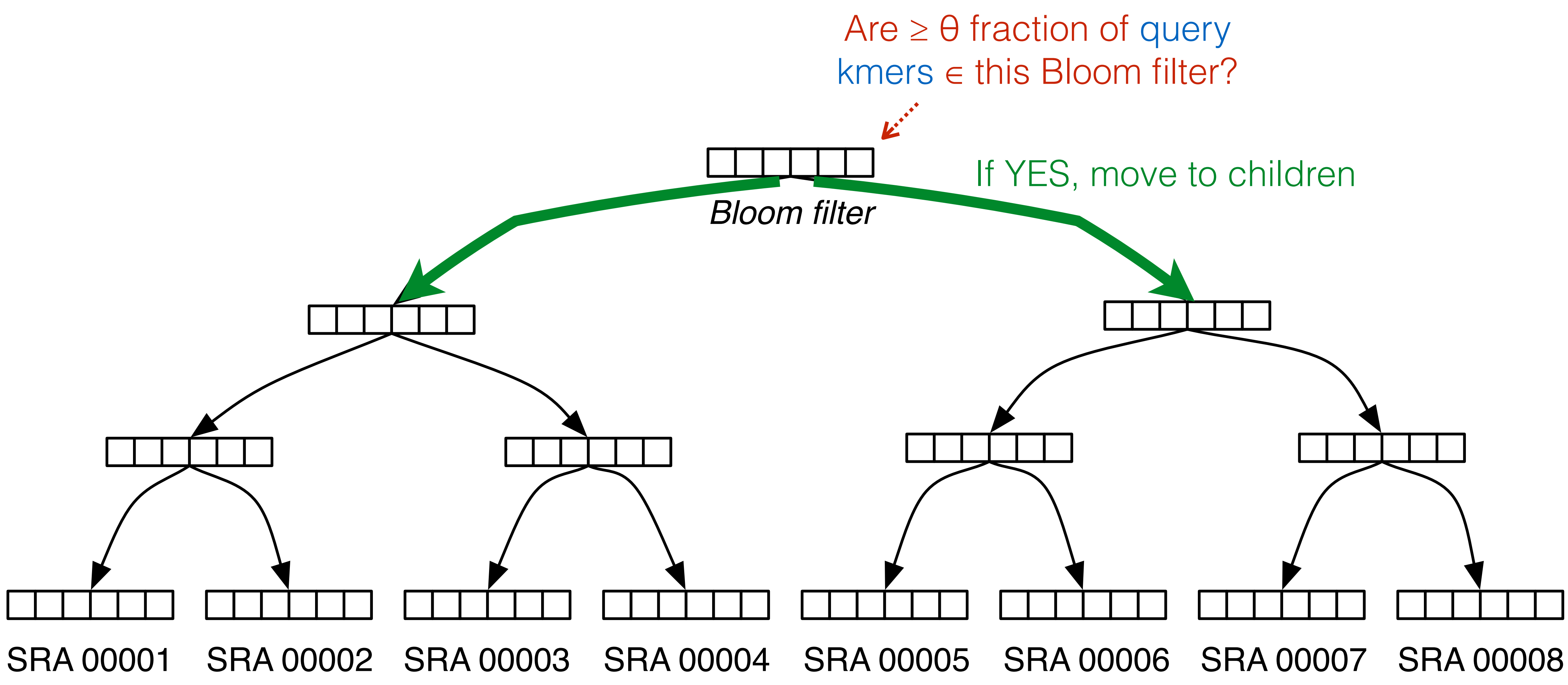


Querying a Sequence Bloom Tree

Good approximate sequence match == lots of shared kmers:

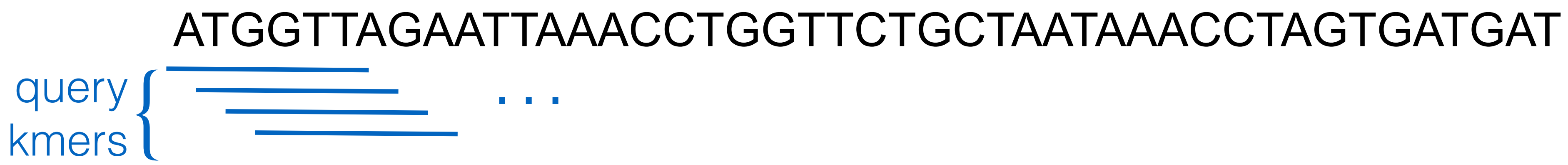
ATGGTTAGAATTAAACCTGGTTCTGCTAATAAACCTAGTGATGAT

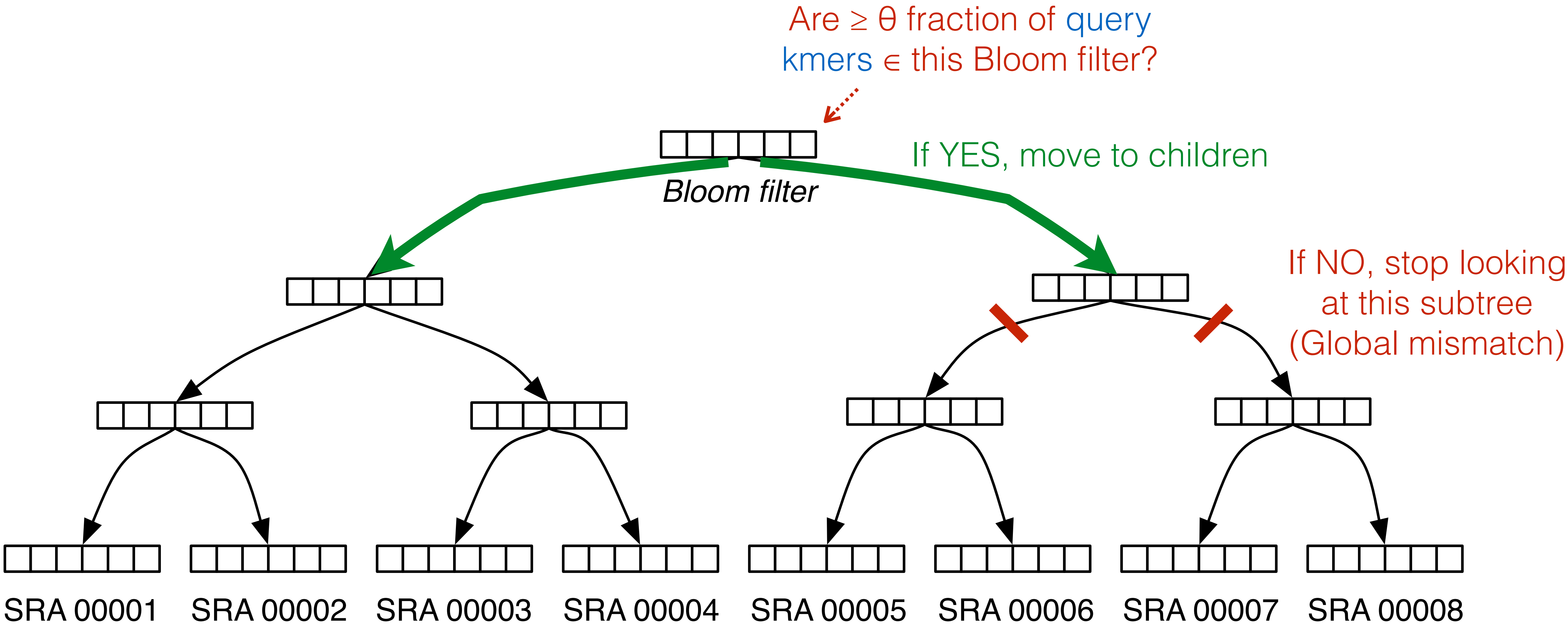
query kmers . . .



Querying a Sequence Bloom Tree

Good approximate sequence match == lots of shared kmers:

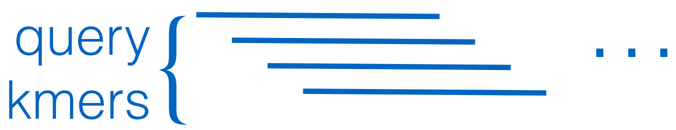


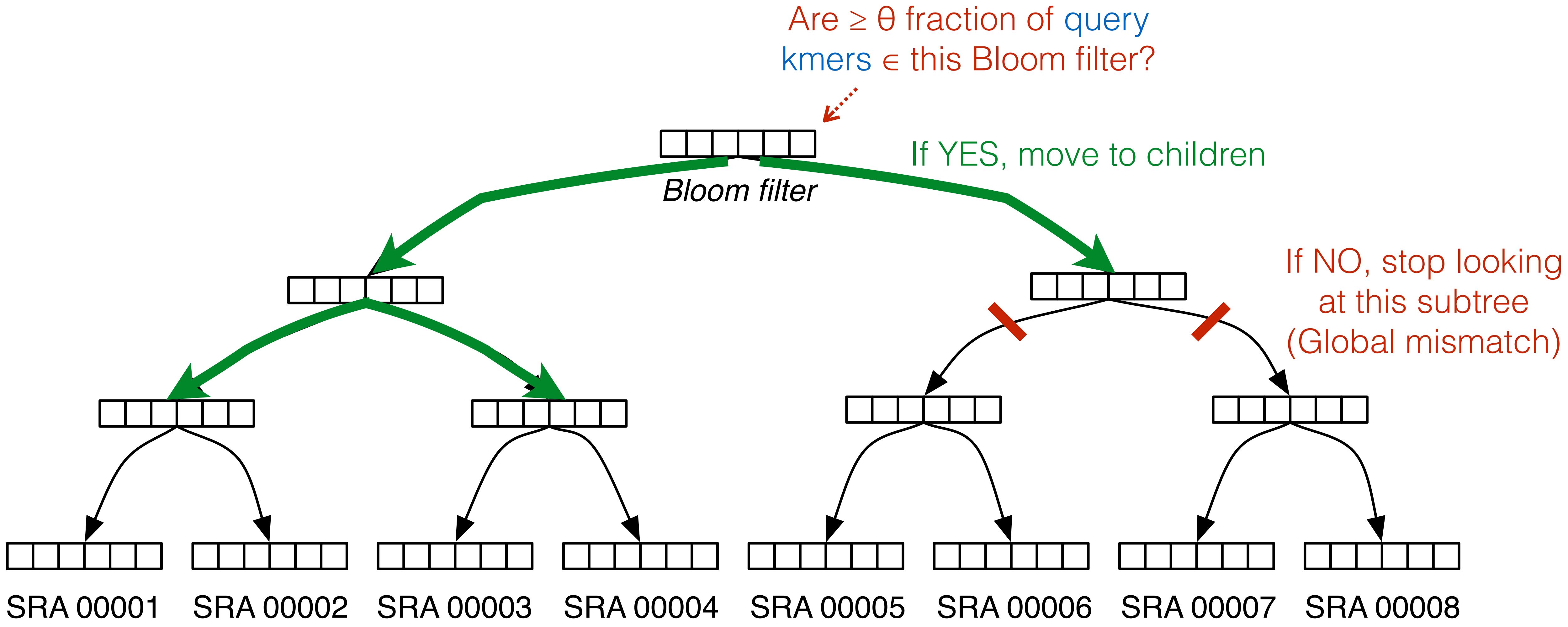


Querying a Sequence Bloom Tree

Good approximate sequence match == lots of shared kmers:

ATGGTTAGAATTAAACCTGGTTCTGCTAATAAACCTAGTGATGAT



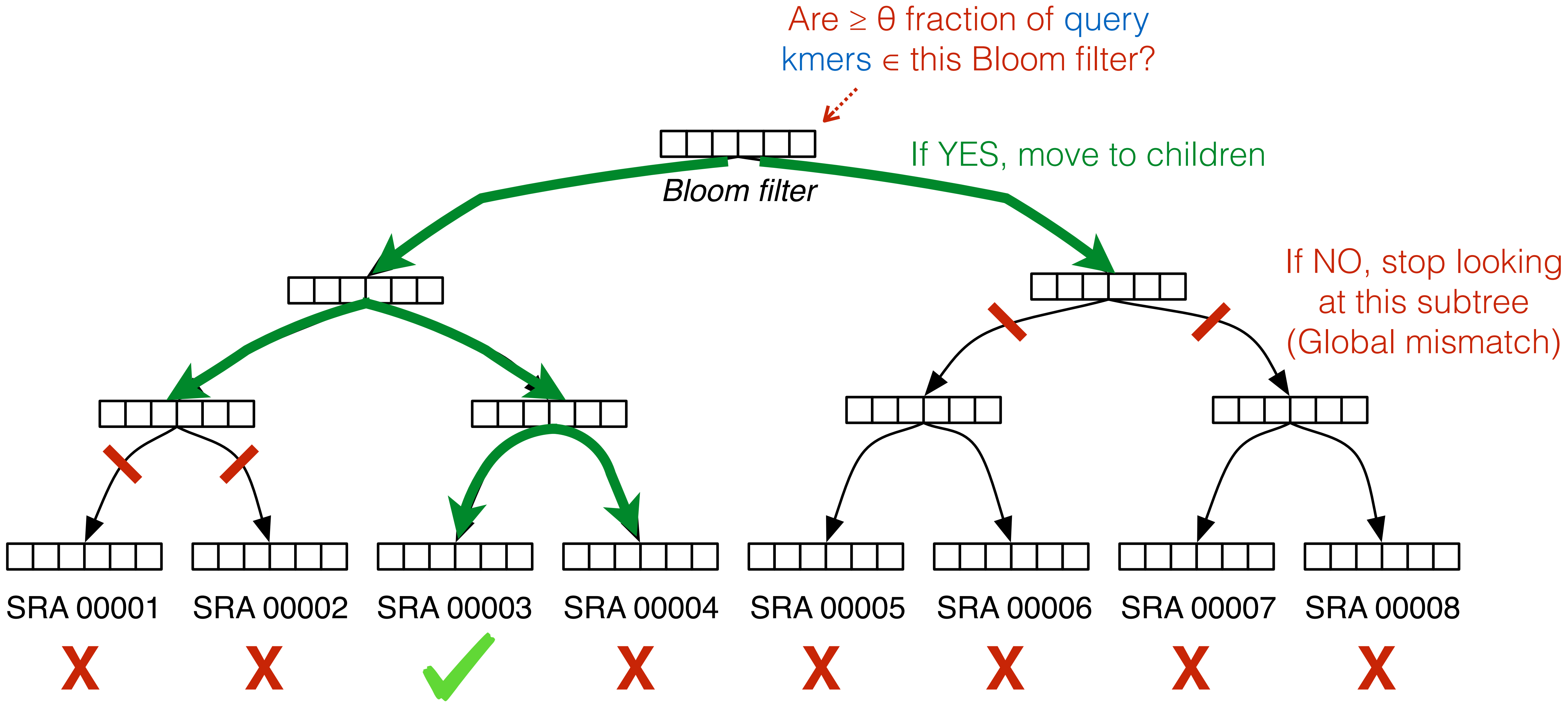


Querying a Sequence Bloom Tree

Good approximate sequence match == lots of shared kmers:

ATGGTTAGAATTAAACCTGGTTCTGCTAATAAACCTAGTGATGAT

query kmers { . . .

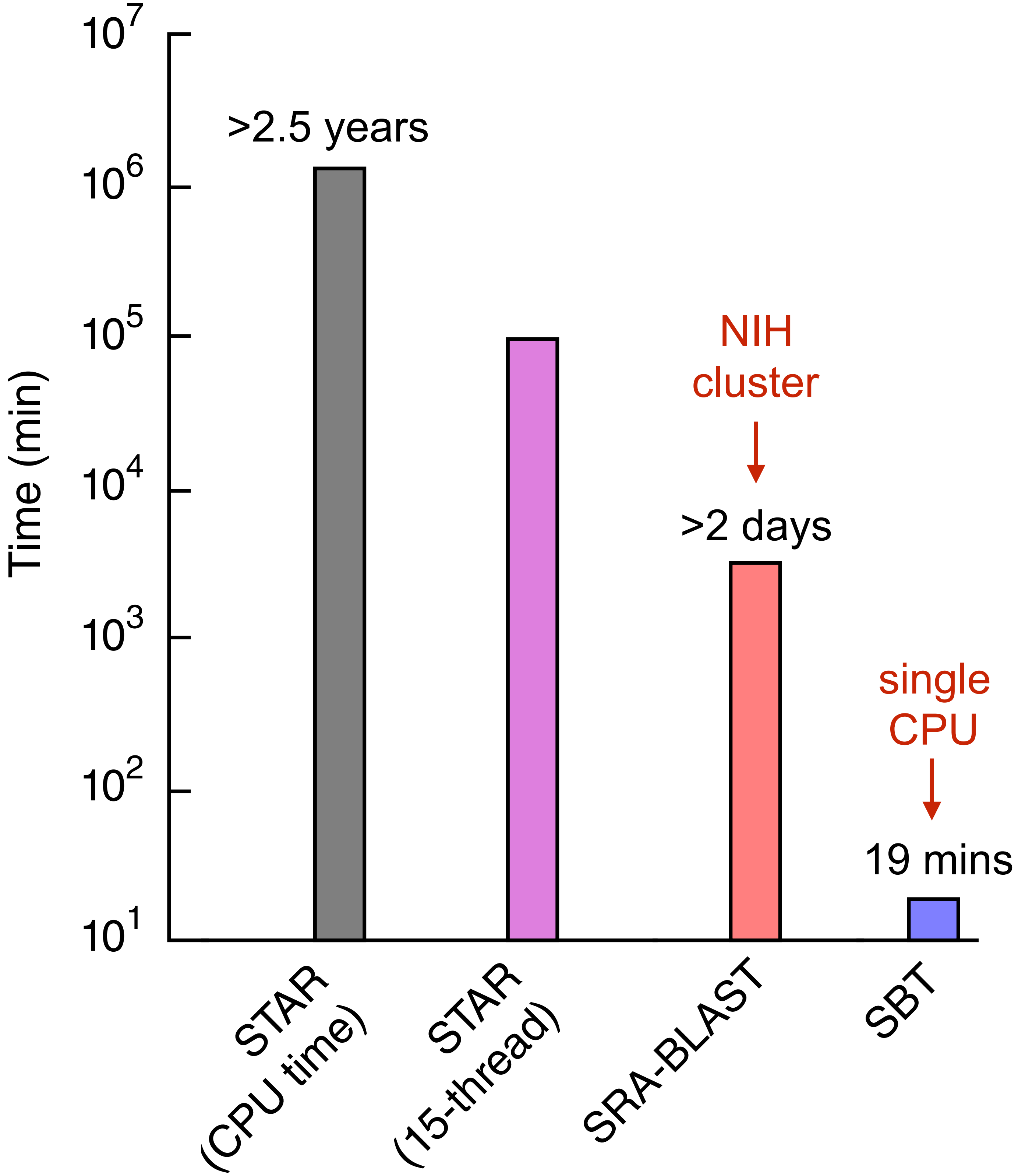


Are ≥ θ fraction of query kmers ∊ this Bloom filter?

If YES, move to children

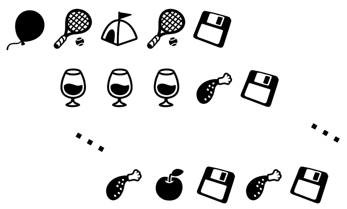
If NO, stop looking at this subtree (Global mismatch)

Sequence Bloom Tree is a very fast solution for containment queries



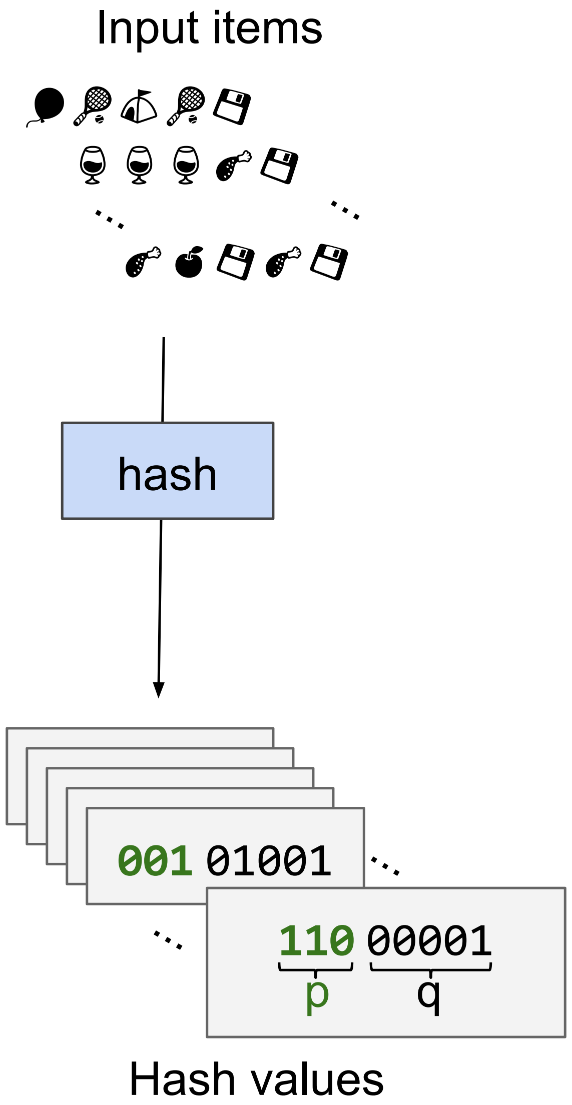
The HyperLogLog Sketch

Input items



Dashing: Fast and Accurate Genomic Distances with HyperLogLog
Daniel Baker and Ben Langmead (2018) *bioRxiv*

The HyperLogLog Sketch

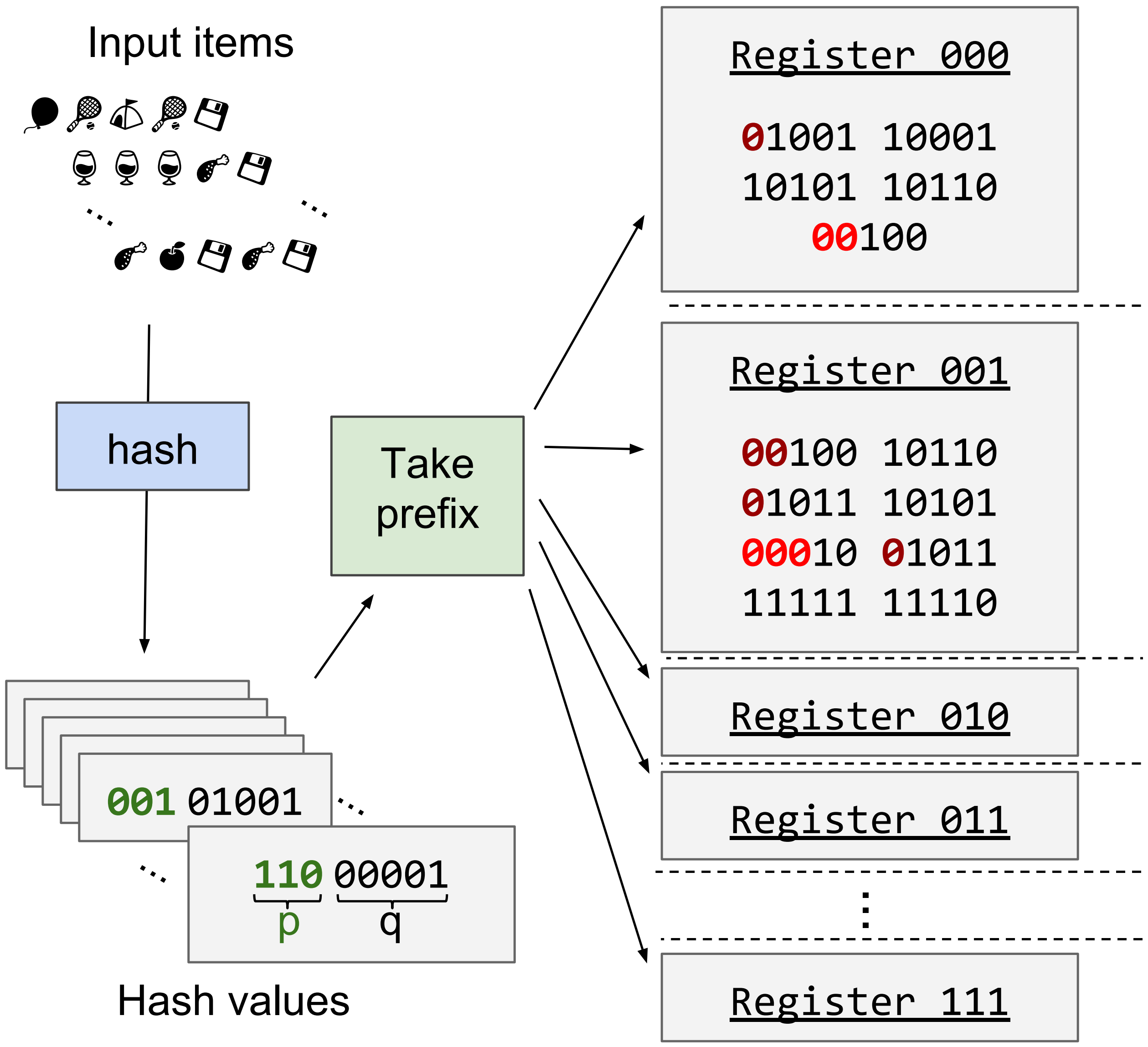


1) Input is hashed into prefix and suffix

Dashing: Fast and Accurate Genomic Distances with HyperLogLog
Daniel Baker and Ben Langmead (2018) *bioRxiv*

The HyperLogLog Sketch

2) Prefix is used to bucket hash values



Input items

hash

Take prefix

001 01001

110 00001

p q

Hash values

Register 000

01001 10001
10101 10110
00100

Register 001

00100 10110
01011 10101
00010 01011
11111 11110

Register 010

Register 011

⋮

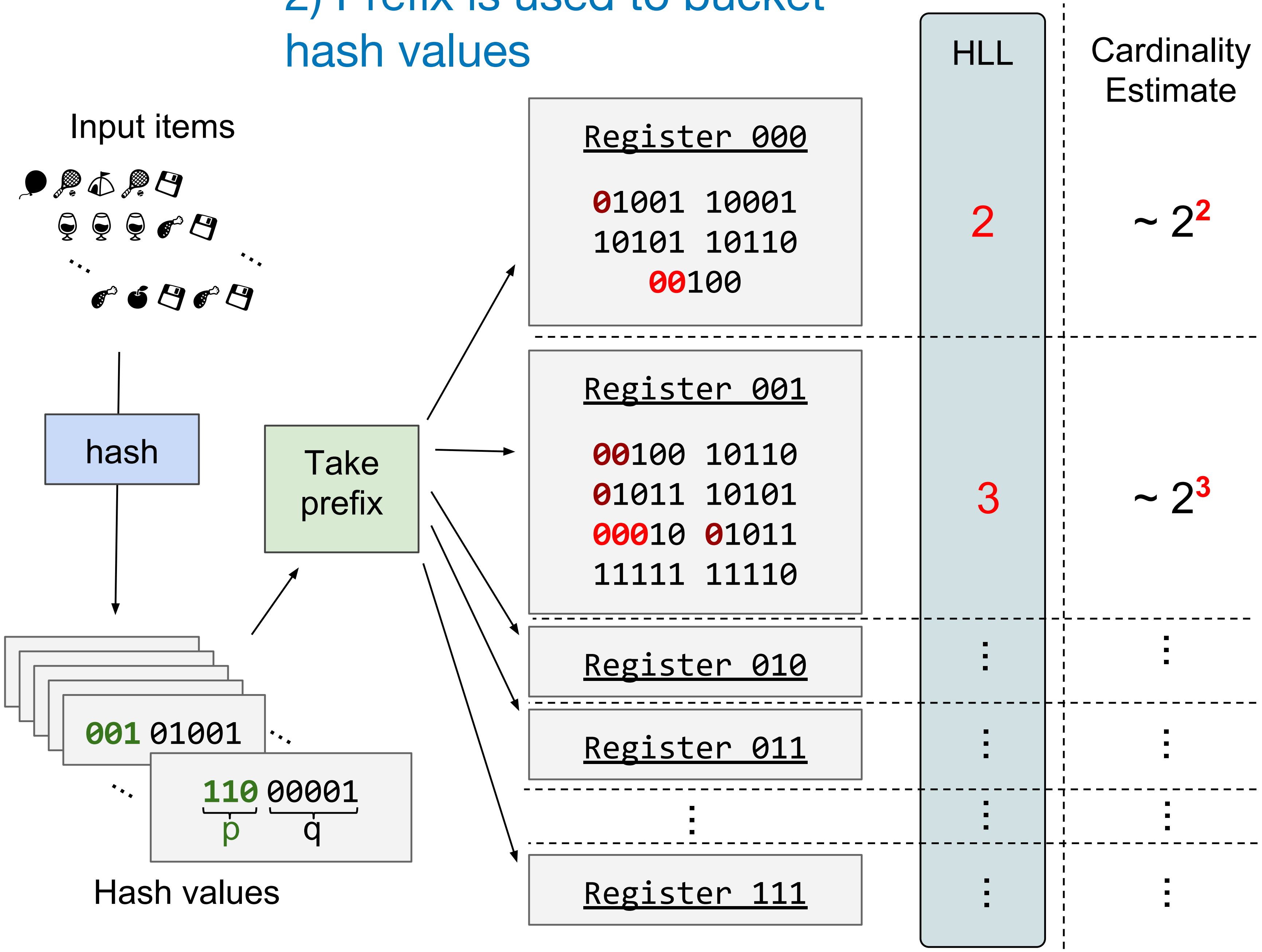
Register 111

1) Input is hashed into prefix and suffix

Dashing: Fast and Accurate Genomic Distances with HyperLogLog
Daniel Baker and Ben Langmead (2018) *bioRxiv*

The HyperLogLog Sketch

2) Prefix is used to bucket hash values



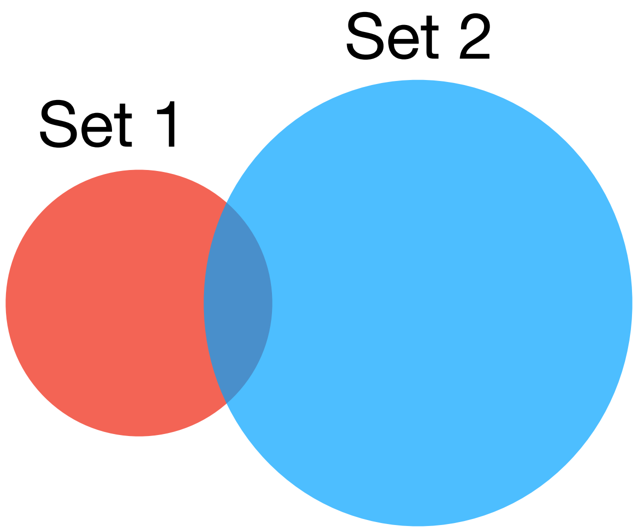
1) Input is hashed into prefix and suffix

3) The maximum leading zero is recorded. Cardinality is estimated based on this value

Dashing: Fast and Accurate Genomic Distances with HyperLogLog
Daniel Baker and Ben Langmead (2018) *bioRxiv*

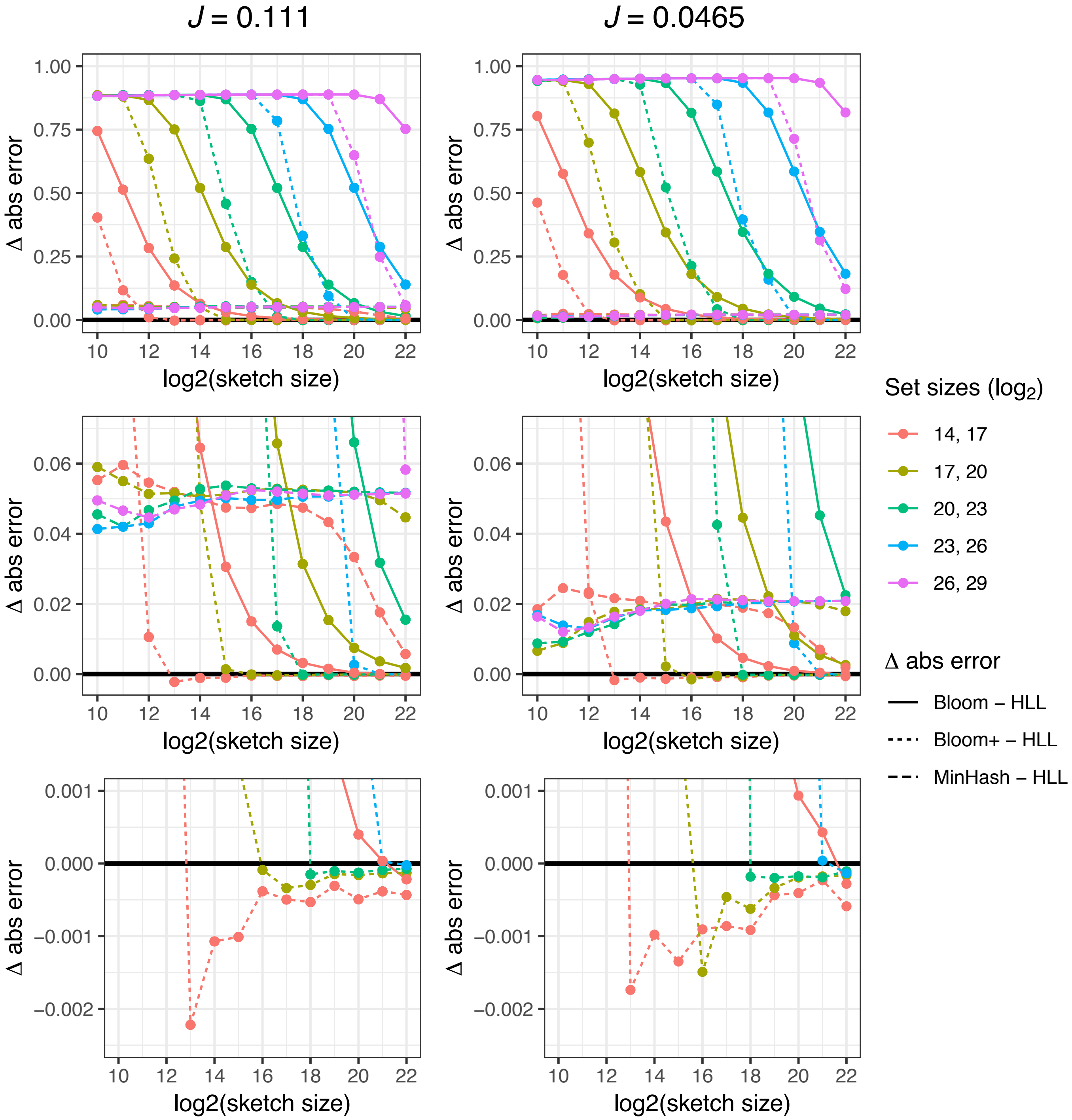
Dashing demonstrates distance difficulties for divergent data

Given an order of magnitude difference in data set size



HyperLogLog outperforms other sketch methods (most of the time)

Estimation error for computing Jaccard



The sketch size and underlying similarity can affect accuracy

The Main Take-away:

- **“Big Data”** in genomics makes conventional analysis difficult
  - Methods like Rail-RNA and recount2 try to improve efficiency through **bulk analysis**
- Sketch techniques trade accuracy for speed — and can often improve both
  - **Minhash** and **HyperLogLog** provide rapid similarity approximations
  - **Bloom Filters** provide efficient set lookup

The Main Take-away:

- **“Big Data”** in genomics makes conventional analysis difficult
    - Methods like Rail-RNA and recount2 try to improve efficiency through **bulk analysis**
- Sketch techniques trade accuracy for speed — and can often improve both
    - **Minhash** and **HyperLogLog** provide rapid similarity approximations
    - **Bloom Filters** provide efficient set lookup

Questions?

Class Projects

- Keep working on them!
- Feel free to come see me if you need help or want advice