Communication-Efficient Jaccard Similarity for High-Performance Distributed Genome Comparisons

MACIEJ BESTA, RAGHAVENDRA KANAKAGIRI, HARUN MUSTAFA, MIKHAIL KARASIKOV, GUNNAR RÄTSCH, TORSTEN HOEFLER, EDGAR SOLOMONIK

भारतीय प्रौद्योगिकी संस्थान तिरुपति
IIT
TIRUPATI

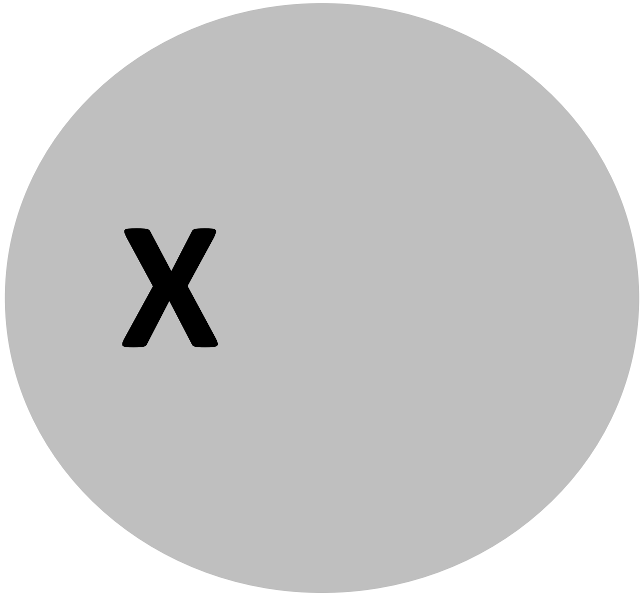
L P N A @CS@Illinois

SPCL

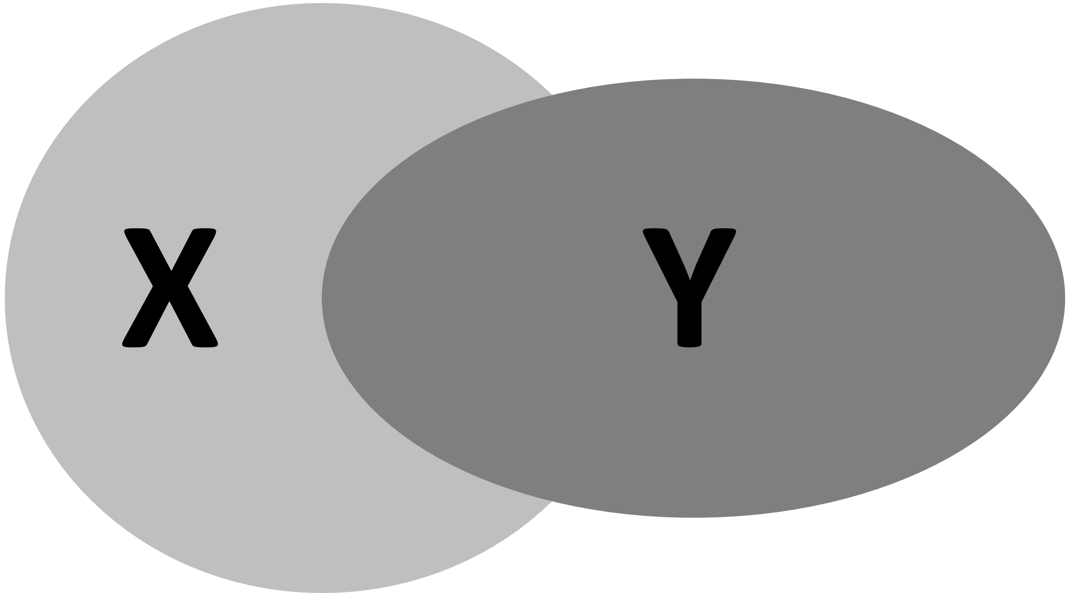
BIOMEDICAL INFORMATICS

SET SIMILARITY

Set similarity

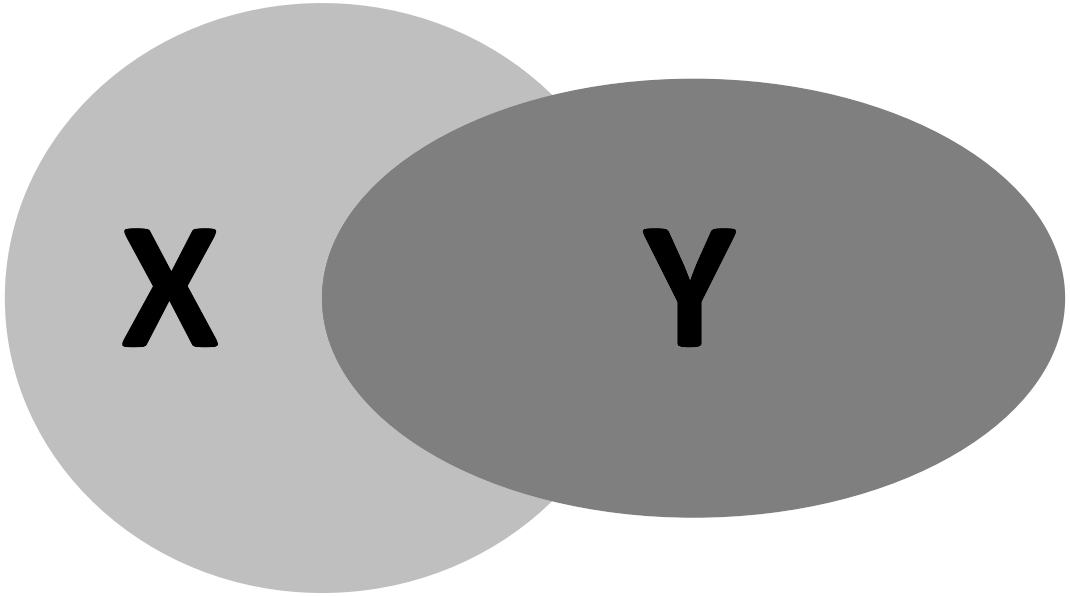


SET SIMILARITY



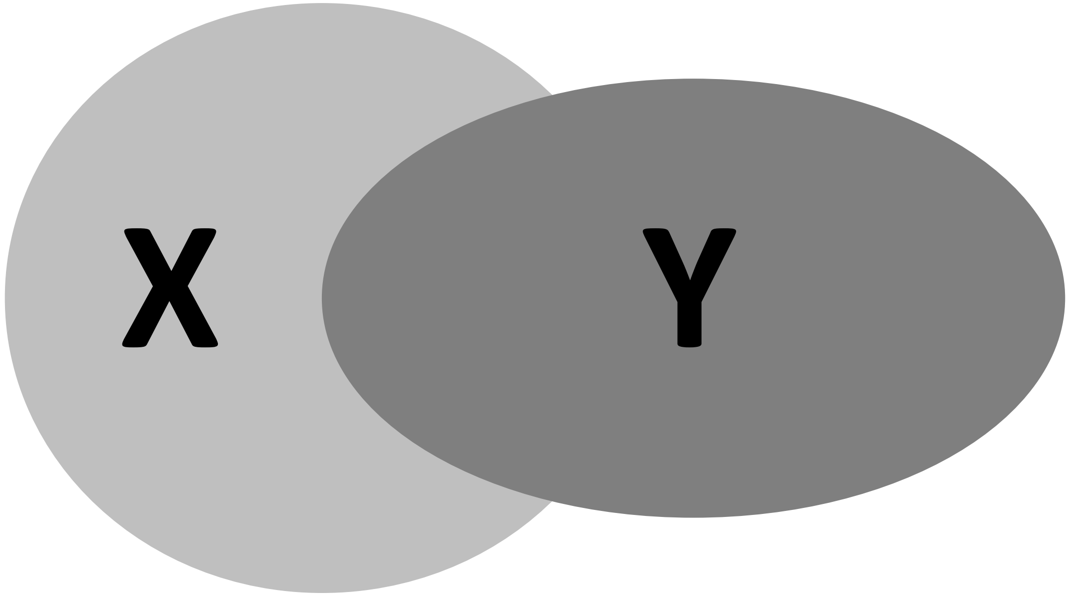
SET SIMILARITY

How can we measure the „similarity” of X and Y?



SET SIMILARITY

How can we measure the „similarity" of X and Y?

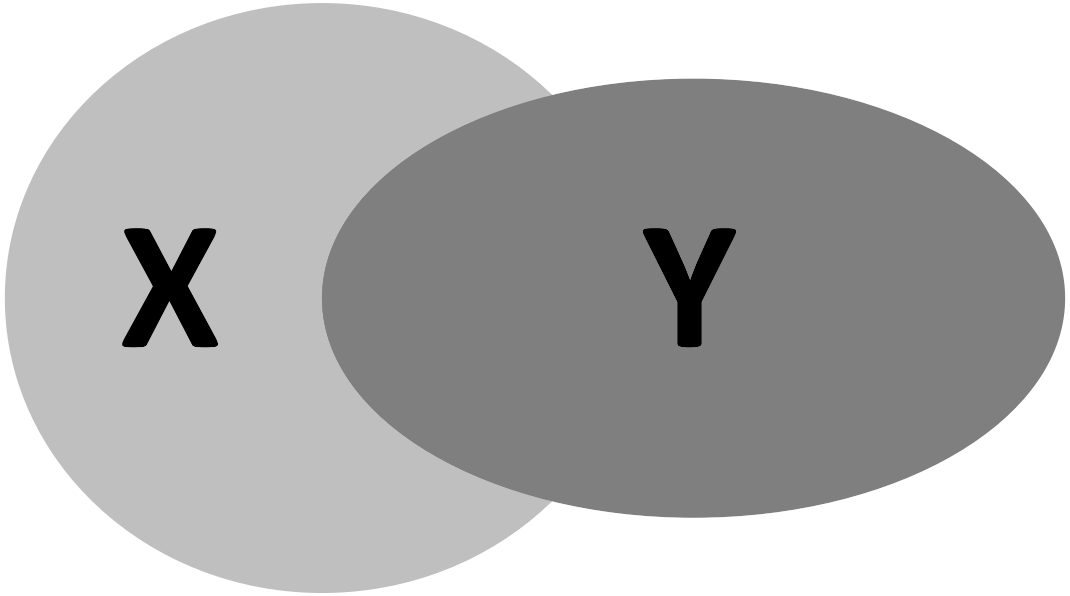


Jaccard Index:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}$$

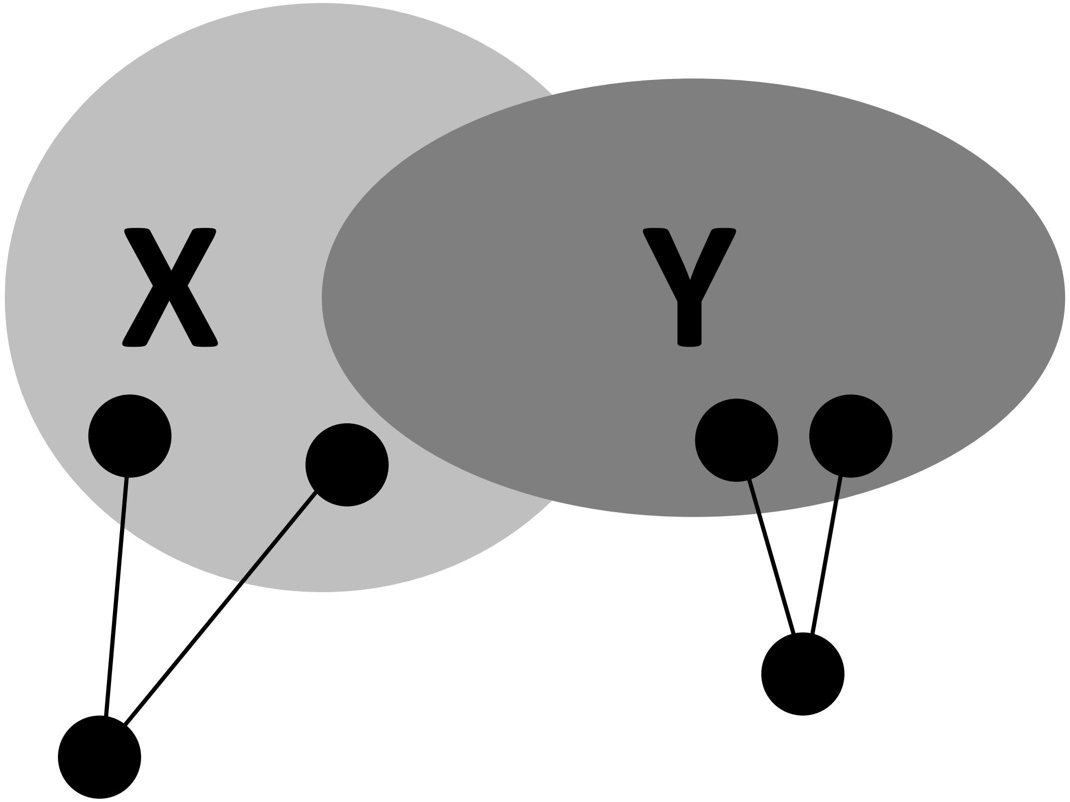
Set similarity

What are X and Y in practice?



SET SIMILARITY

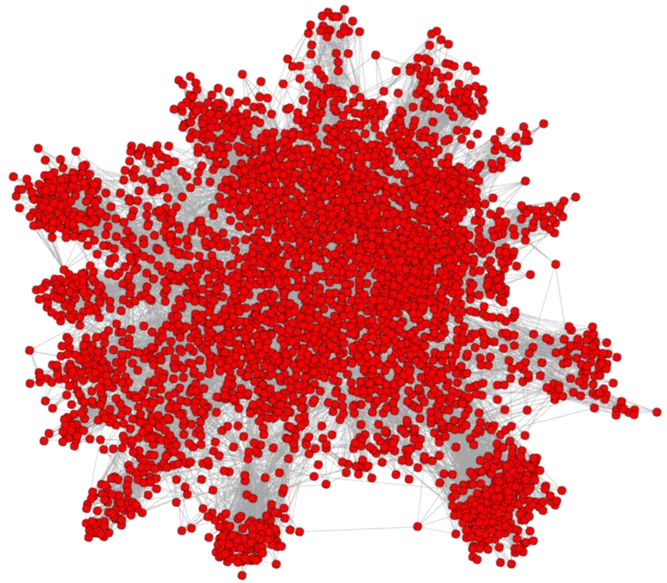
What are X and Y in practice?

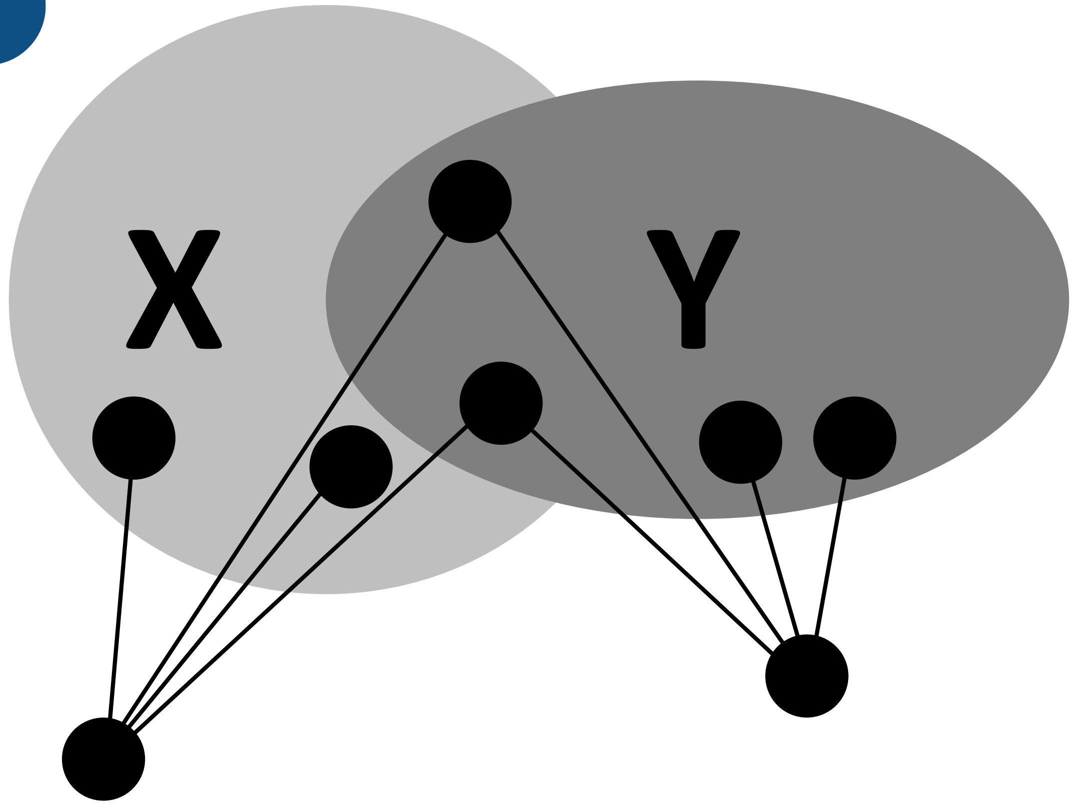


SET SIMILARITY

Neighborhoods of vertices in a graph

What are X and Y in practice?

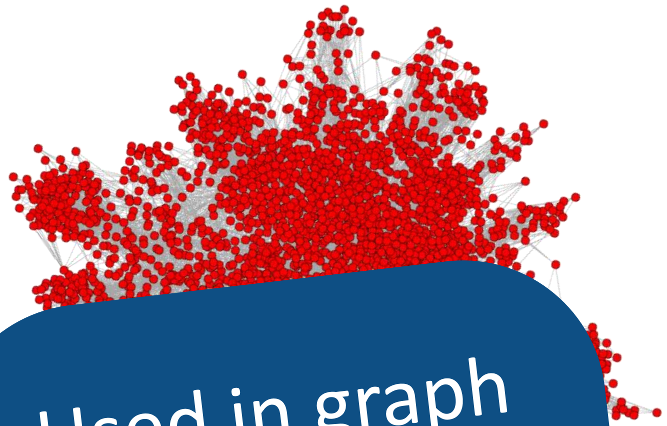


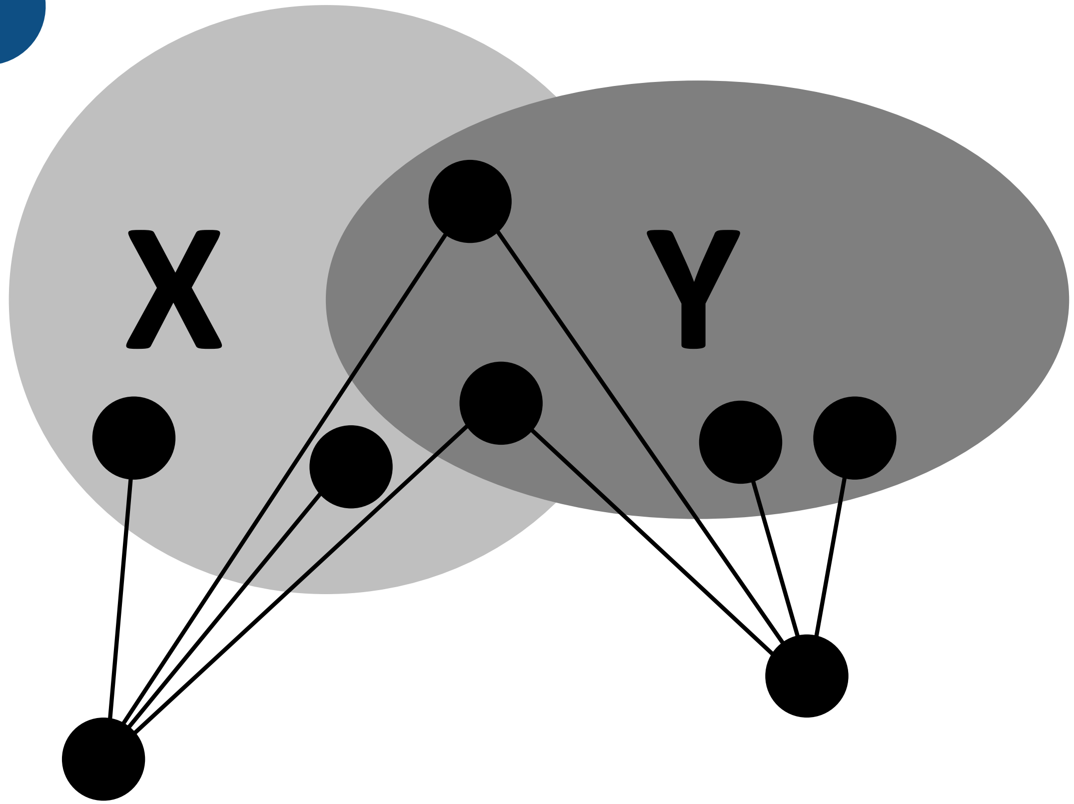


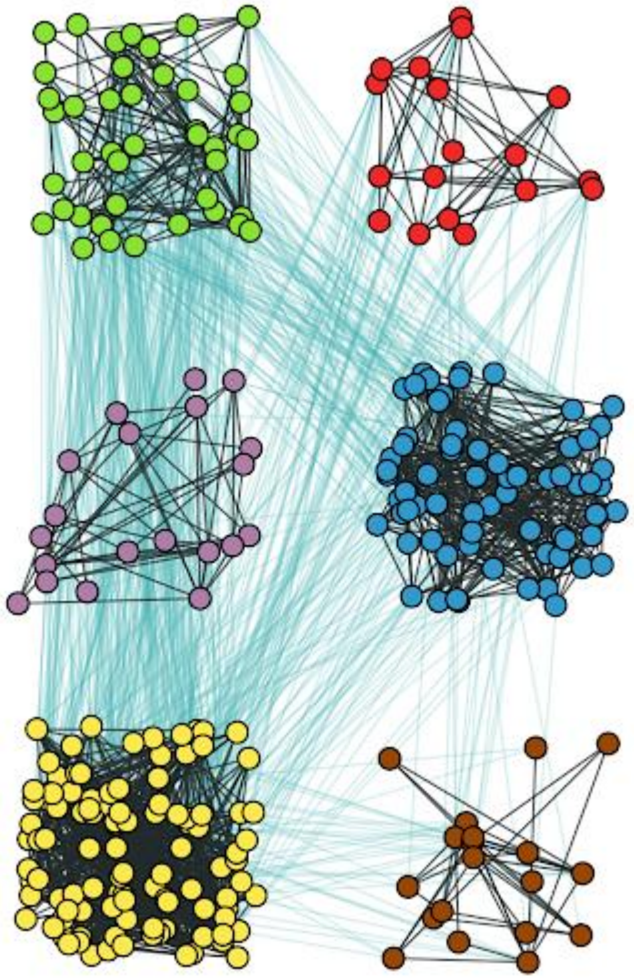
Set similarity

Neighborhoods of vertices in a graph

What are X and Y in practice?



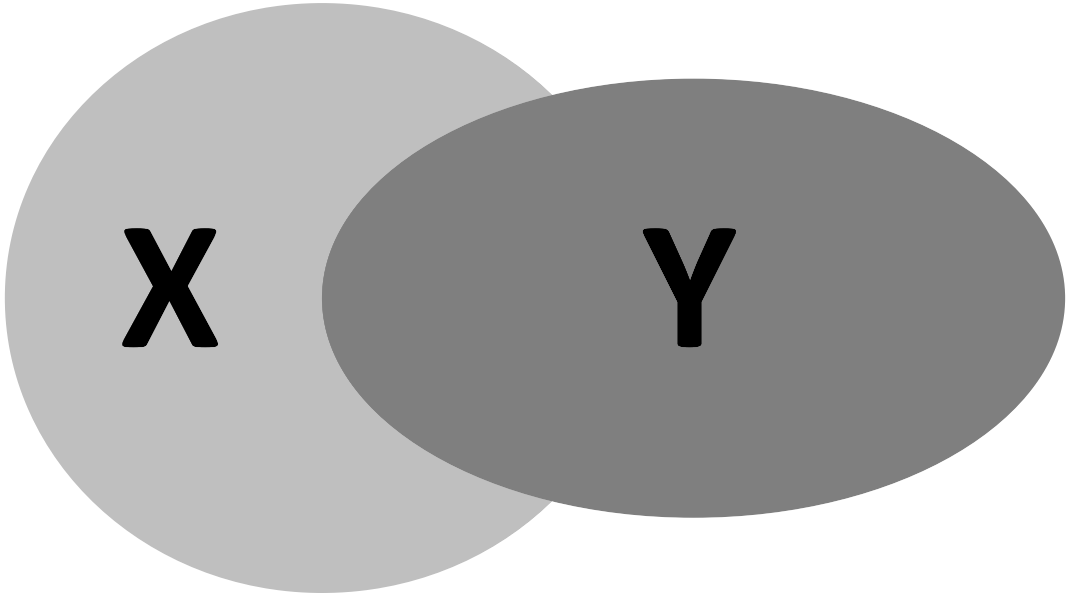




Used in graph analytics (clustering, link prediction, ...)

Set similarity

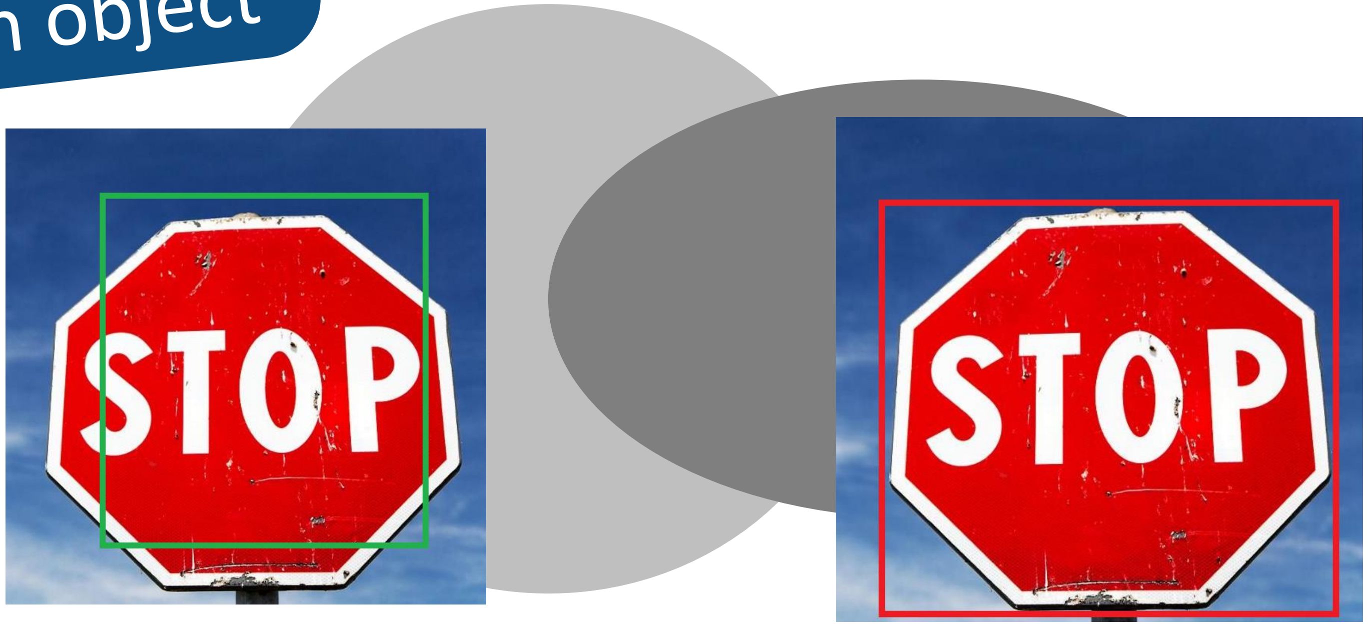
What are X and Y in practice?



Set similarity

Detected object & ground-truth object

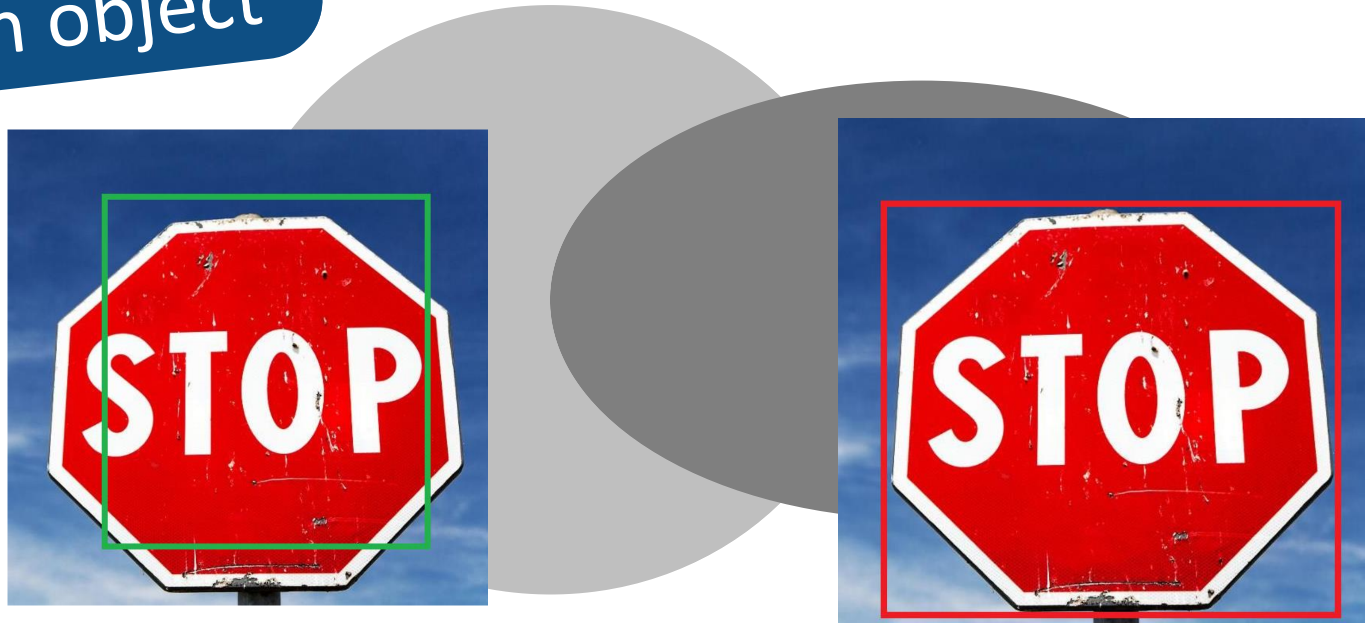
What are X and Y in practice?



Set similarity

What are X and Y in practice?

Detected object & ground-truth object



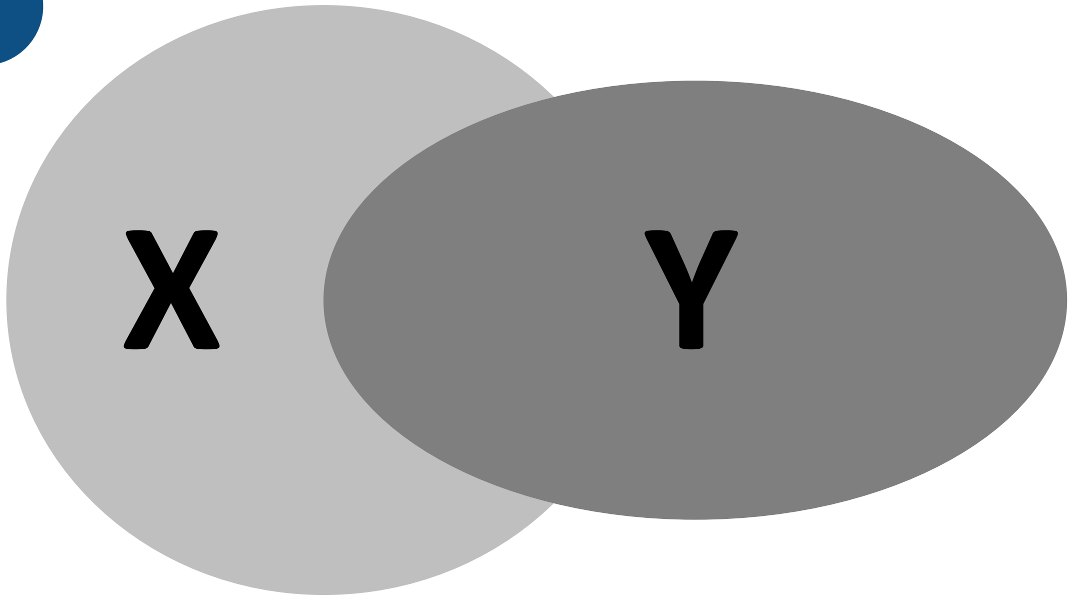
Used in image recognition

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

SET SIMILARITY

Sequences of genomes

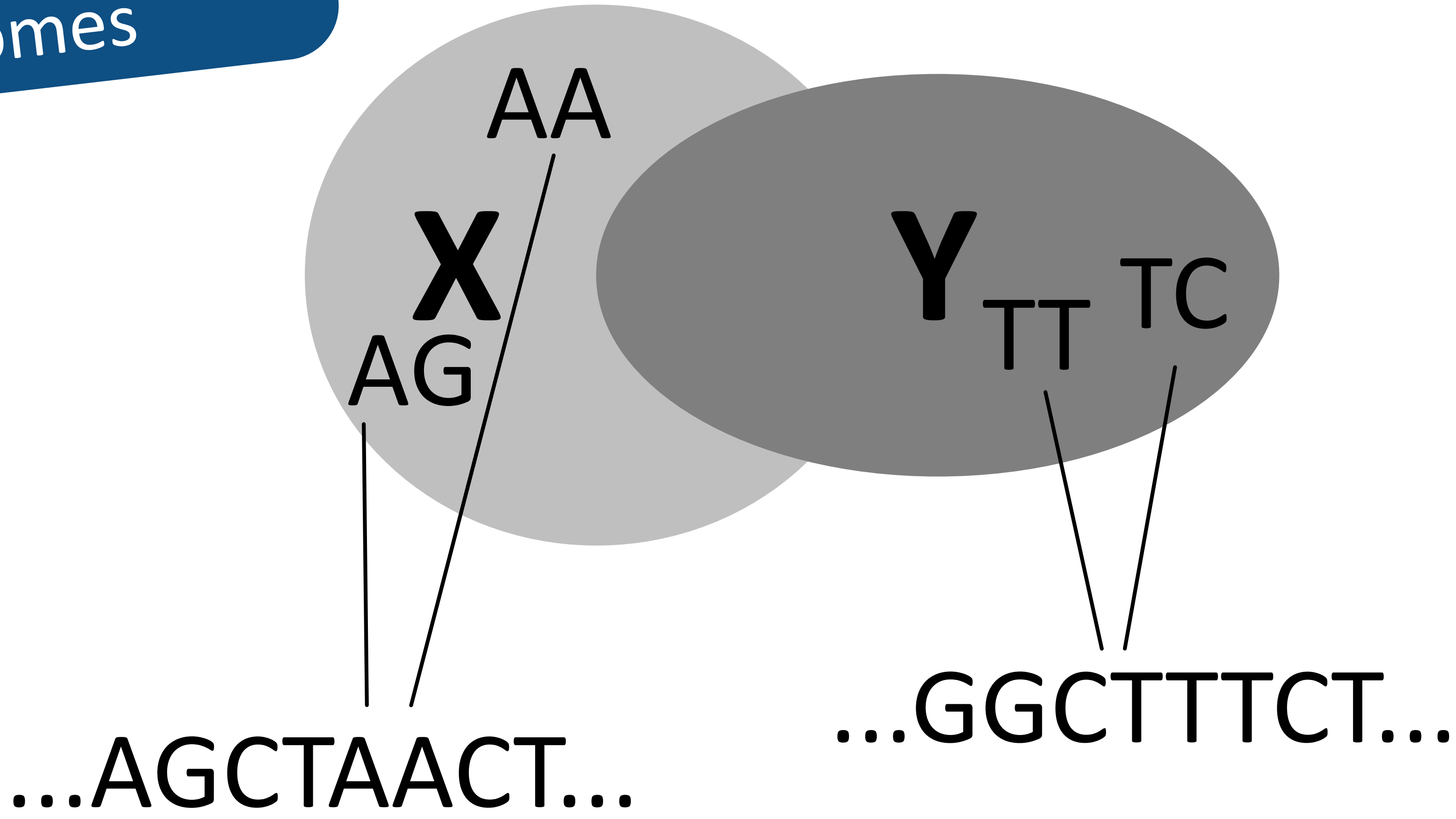
What are X and Y in practice?



SET SIMILARITY

Sequences of genomes

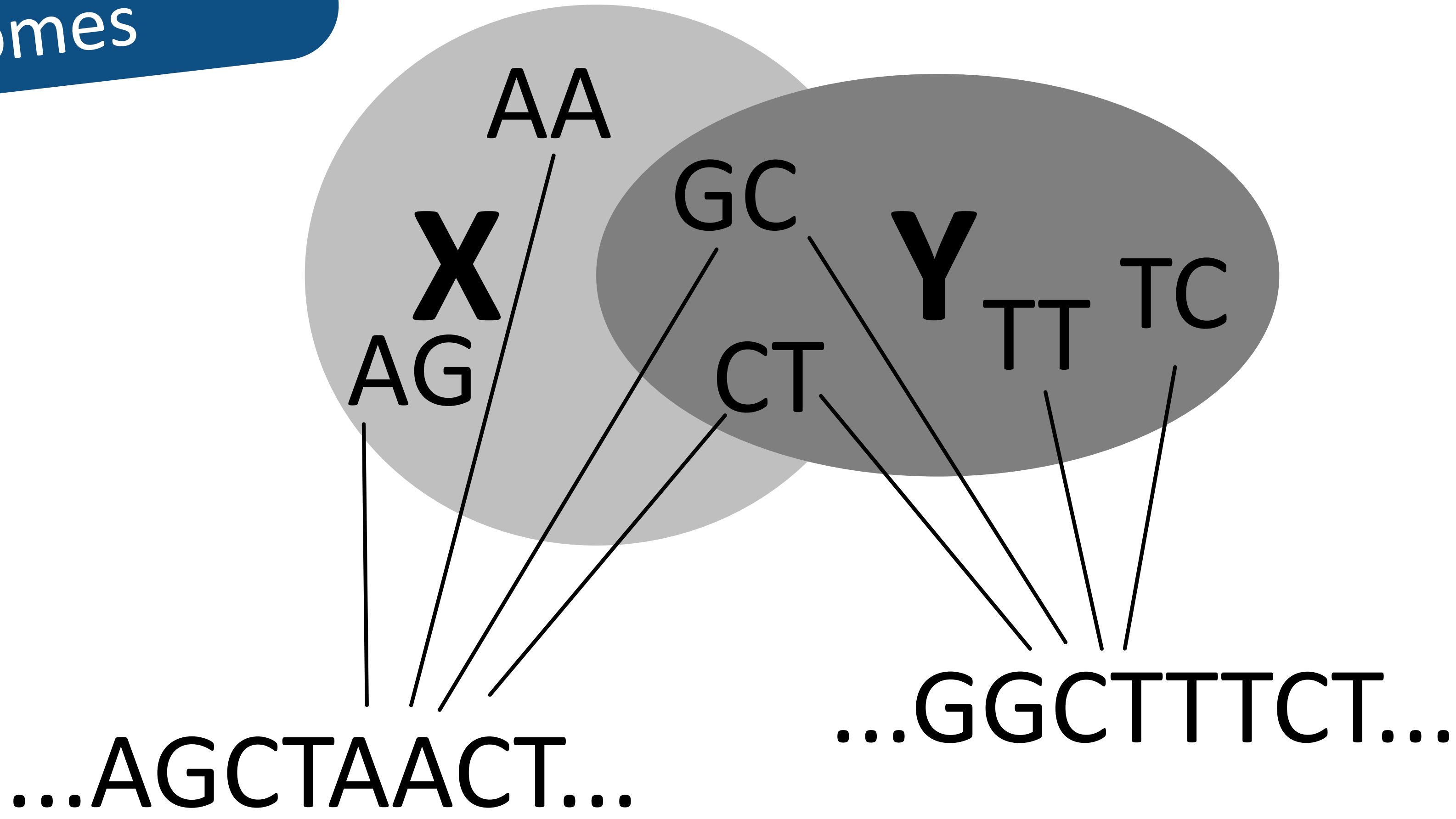
What are X and Y in practice?



Set similarity

Sequences of genomes

What are X and Y in practice?

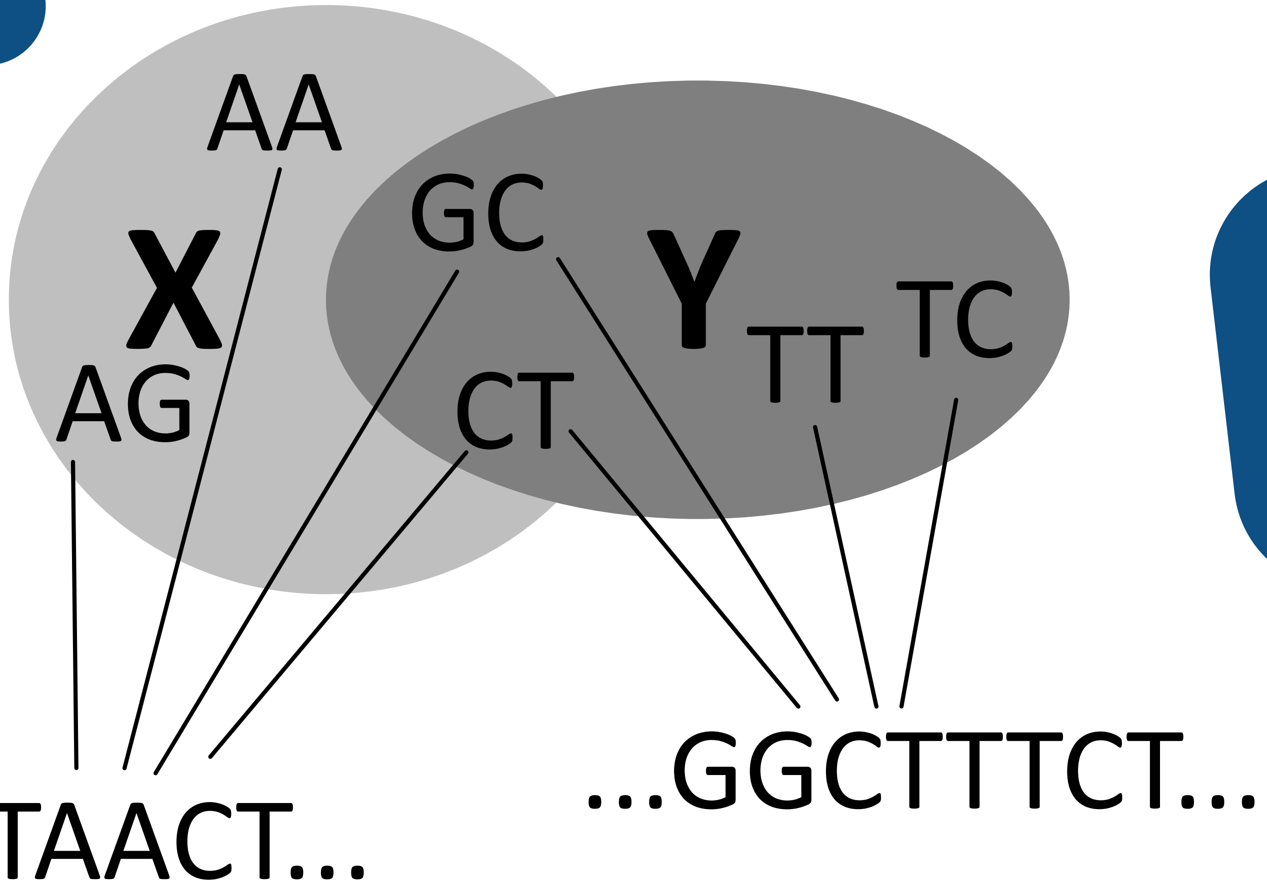


Set similarity

Sequences of genomes

What are X and Y in practice?

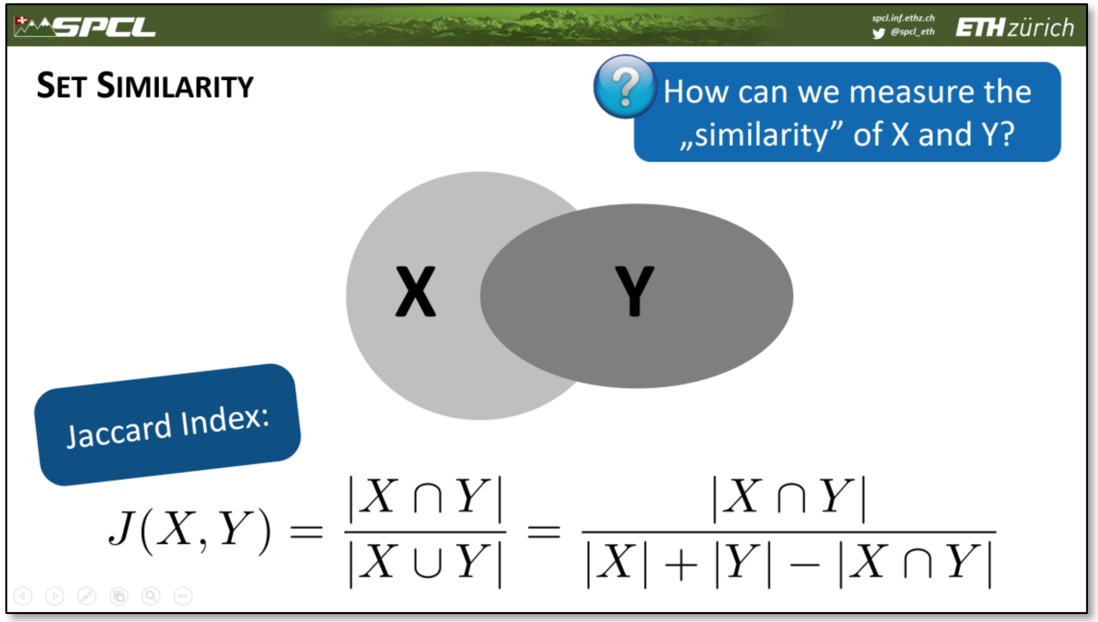
A: adenine,
T: thymine,
G: guanine,
C: cytosine



Used in genome assembly

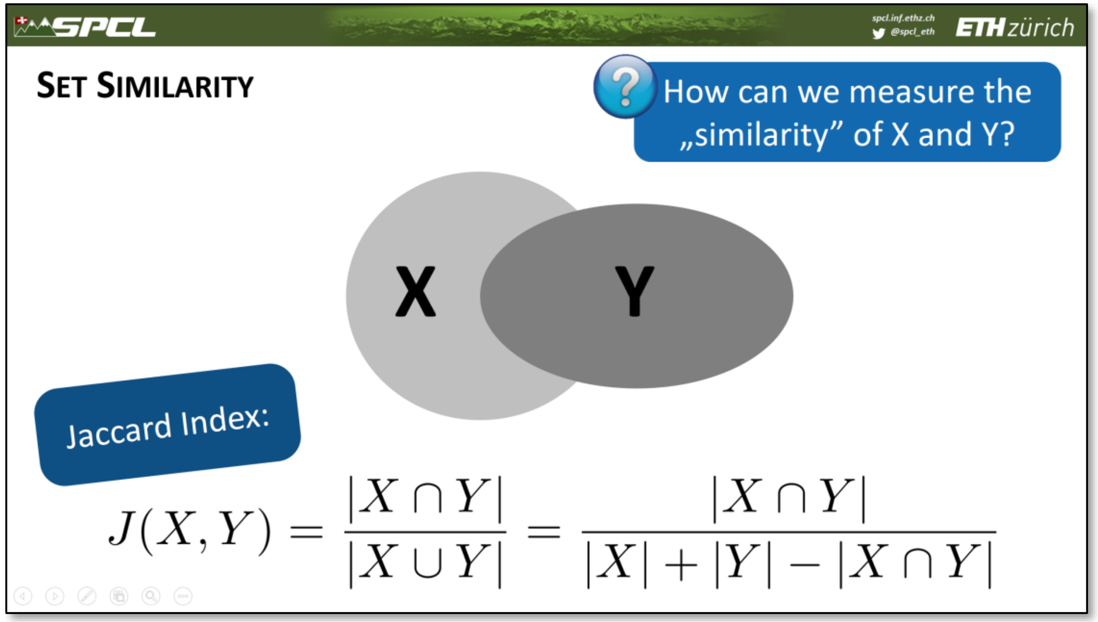
Scope of work

Scope of work



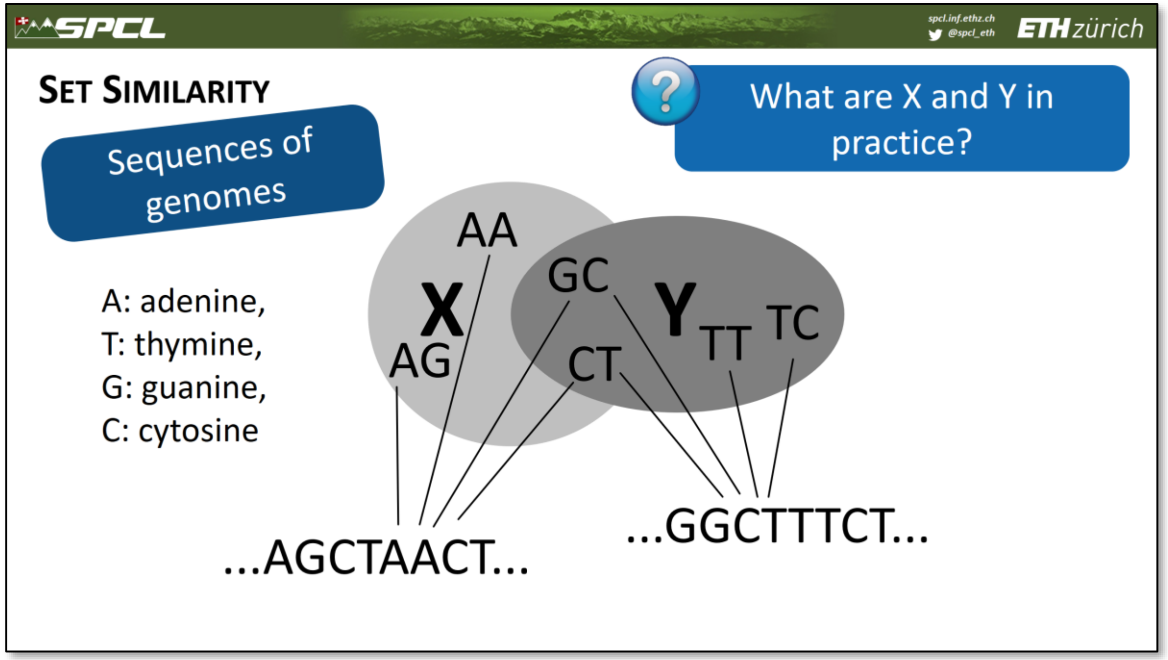
Part 1 „SimilarityAtScale": the first communication-efficient distributed algorithm to compute the general Jaccard similarity index and distance

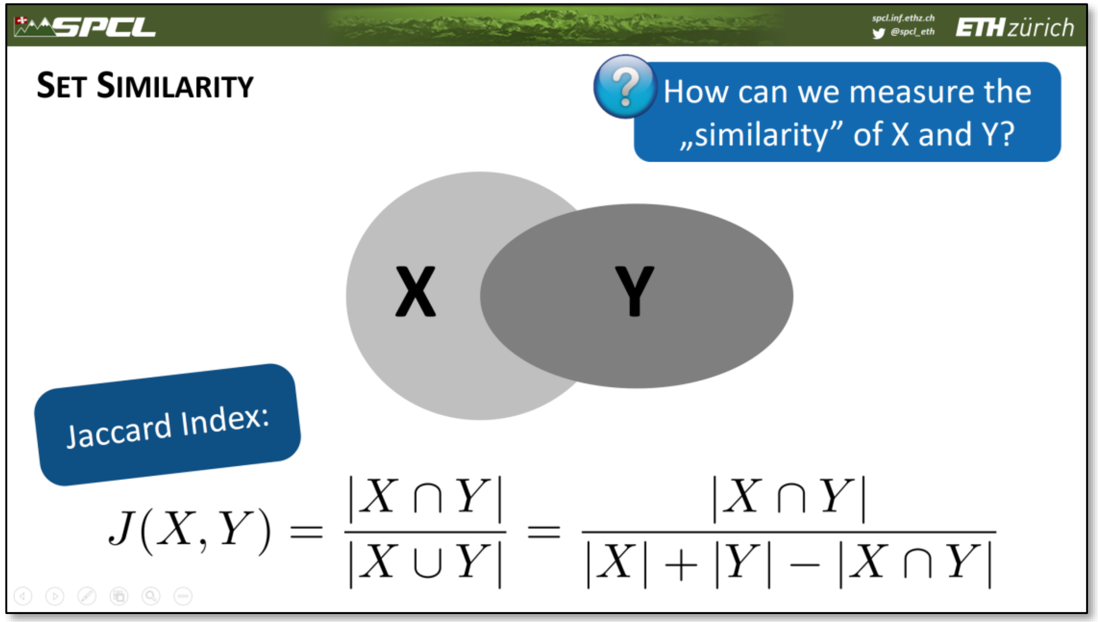
Scope of work



Part 1 „SimilarityAtScale": the first communication-efficient distributed algorithm to compute the general Jaccard similarity index and distance

Part 2 „GenomeAtScale": the first tool for fast, scalable, accurate, and large-scale derivations of Jaccard index between genome sequences





Part 1 „SimilarityAtScale": the first communication-efficient distributed algorithm to compute the general Jaccard similarity index and distance

SimilarityAtScale

SimilarityAtScale

(1) Sets $X_i \quad i = 1, \ldots, n$

SimilarityAtScale

(1) Sets X_i $i = 1, \ldots, n$

(2) Indicator matrix $\boldsymbol{A} \in \mathbb{B}^{m \times n}$ $a_{ij} = \begin{cases} 1 & : i \in X_j \\ 0 & : \text{otherwise} \end{cases}$

SimilarityAtScale

Describes which element (out of all *m* possible ones) belongs to which set (out of all *n* possible ones)

(1) Sets $X_i \quad i = 1, \dots, n$

(2) Indicator matrix $\boldsymbol{A} \in \mathbb{B}^{m \times n} \quad a_{ij} = \begin{cases} 1 & : i \in X_j \\ 0 & : \text{otherwise} \end{cases}$

SimilarityAtScale

Describes which element (out of all *m* possible ones) belongs to which set (out of all *n* possible ones)

(1) Sets $X_i \quad i = 1, \ldots, n$

(2) Indicator matrix $\boldsymbol{A} \in \mathbb{B}^{m \times n} \quad a_{ij} = \begin{cases} 1 & : i \in X_j \\ 0 & : \text{otherwise} \end{cases}$

(3) Similarity matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$

$$s_{ij} = J(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|}$$

SimilarityAtScale

Describes which element (out of all *m* possible ones) belongs to which set (out of all *n* possible ones)

(1) Sets $X_i \quad i = 1, \ldots, n$

(2) Indicator matrix $\boldsymbol{A} \in \mathbb{B}^{m \times n} \quad a_{ij} = \begin{cases} 1 & : i \in X_j \\ 0 & : \text{otherwise} \end{cases}$

(3) Similarity matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$

$$s_{ij} = J(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|}$$

Describes Jaccard similarity between each set

SIMILARITYATSCALE: COMPUTE SIMILARITY MATRIX $\boldsymbol{S} \in \mathbb{R}^{n \times n}$

$$s_{ij} = J(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|}$$

SIMILARITYATSCALE: COMPUTE SIMILARITY MATRIX $\boldsymbol{S} \in \mathbb{R}^{n \times n}$

Cardinalities of intersections of each pair of data samples

$$s_{ij} = J(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|}$$

SIMILARITYATSCALE: COMPUTE SIMILARITY MATRIX $S \in \mathbb{R}^{n \times n}$

Cardinalities of intersections of each pair of data samples

$$s_{ij} = J(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|}$$

Cardinalities of unions of each pair of data samples

SimilarityAtScale: Compute similarity matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$

Cardinalities of intersections of each pair of data samples

$\boldsymbol{B} \in \mathbb{N}^{n \times n}$

$$s_{ij} = J(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|}$$

Cardinalities of unions of each pair of data samples

SimilarityAtScale: Compute similarity matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$

Cardinalities of intersections of each pair of data samples

$$s_{ij} = J(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|}$$

$\boldsymbol{B} \in \mathbb{N}^{n \times n}$

$\boldsymbol{C} \in \mathbb{N}^{n \times n}$

Cardinalities of unions of each pair of data samples

SimilarityAtScale: Compute similarity matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$

Cardinalities of intersections of each pair of data samples

$$s_{ij} = J(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|}$$

$\rightarrow \boldsymbol{B} \in \mathbb{N}^{n \times n}$

Can be obtained from ***B*** with simple set theory

$\rightarrow \boldsymbol{C} \in \mathbb{N}^{n \times n}$

Cardinalities of unions of each pair of data samples

$$|X_i \cup X_j| = |X_i| + |X_j| - |X_i \cap X_j|$$

SimilarityAtScale: Compute similarity matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$

Cardinalities of intersections of each pair of data samples

Critical element!

$$s_{ij} = J(X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|}$$

$|X_i \cap X_j| \rightarrow \boldsymbol{B} \in \mathbb{N}^{n \times n}$

Can be obtained from ***B*** with simple set theory

$|X_i \cup X_j| \rightarrow \boldsymbol{C} \in \mathbb{N}^{n \times n}$

Cardinalities of unions of each pair of data samples

$$|X_i \cup X_j| \\ = |X_i| + |X_j| - |X_i \cap X_j|$$

SimilarityAtScale: Compute intersection matrix $\boldsymbol{B} \in \mathbb{N}^{n \times n}$

$$b_{ij} = |X_i \cap X_j| = \sum_k a_{ki} a_{kj}, \text{ so } \boldsymbol{B} = \boldsymbol{A}^T \boldsymbol{A}$$

SimilarityAtScale: Compute intersection matrix $\boldsymbol{B} \in \mathbb{N}^{n \times n}$

$$b_{ij} = |X_i \cap X_j| = \sum_k a_{ki} a_{kj}, \text{ so } \boldsymbol{B} = \boldsymbol{A}^T \boldsymbol{A}$$

Sparse-matrix product

SimilarityAtScale: Compute intersection matrix $B \in \mathbb{N}^{n \times n}$

$$b_{ij} = |X_i \cap X_j| = \sum_k a_{ki} a_{kj}, \text{ so } \boldsymbol{B} = \boldsymbol{A}^T \boldsymbol{A}$$

Sparse-matrix product

Largest computational challenge: for the considered workloads, the indicator matrix *A* is incredibly sparse

SimilarityAtScale: Compute intersection matrix $B \in \mathbb{N}^{n \times n}$

$$b_{ij} = |X_i \cap X_j| = \sum_k a_{ki} a_{kj}, \text{ so } \boldsymbol{B} = \boldsymbol{A}^T \boldsymbol{A}$$

Sparse-matrix product

<u>**Largest computational challenge:**</u> for the considered workloads, the indicator matrix *A* is <u>incredibly sparse</u>

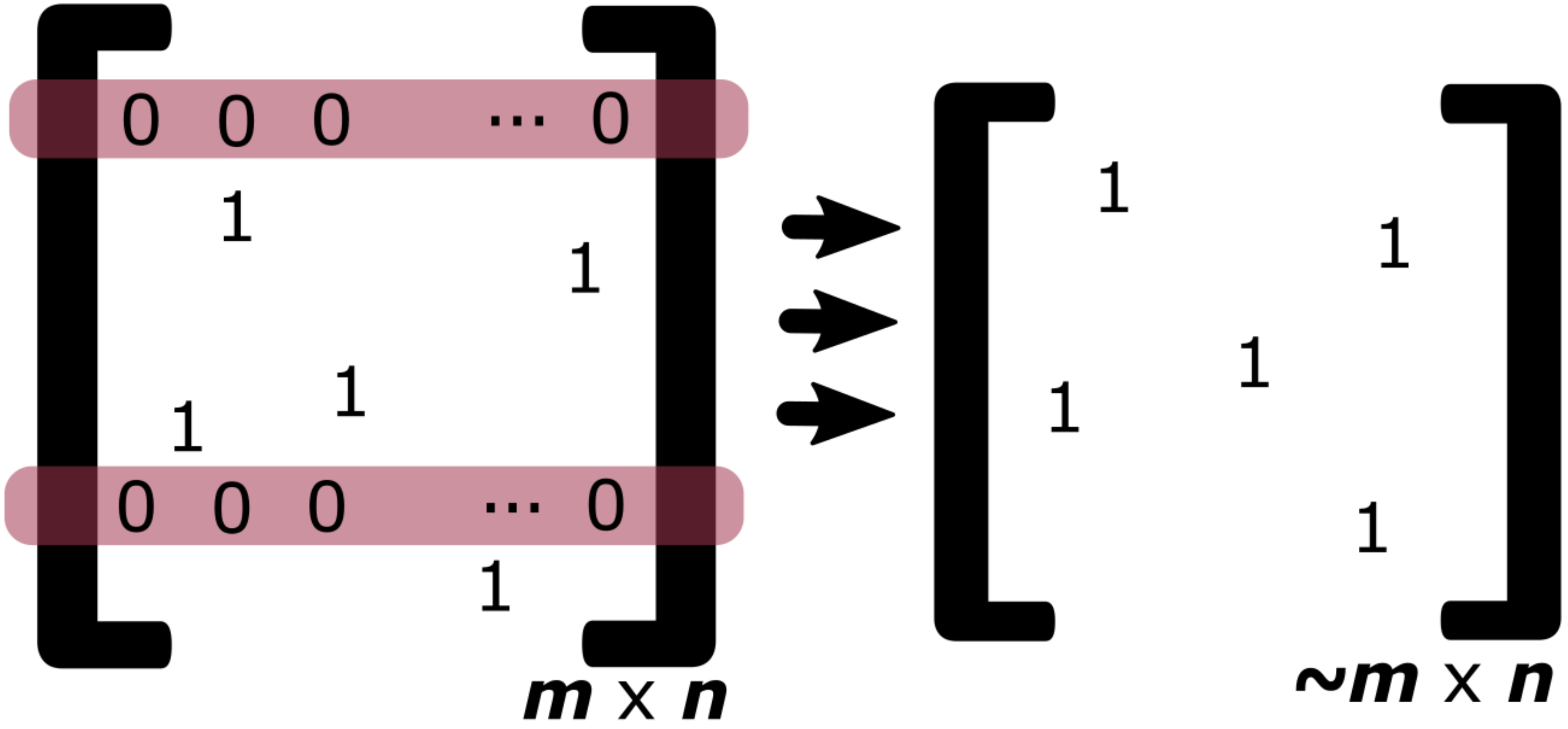
We use three techniques to tackle it

SimilarityAtScale: Compute intersection matrix $B = A^T A$

Technique 1: remove zero rows from *A*

SimilarityAtScale: Compute intersection matrix $B = A^T A$

Technique 1: remove zero rows from *A*

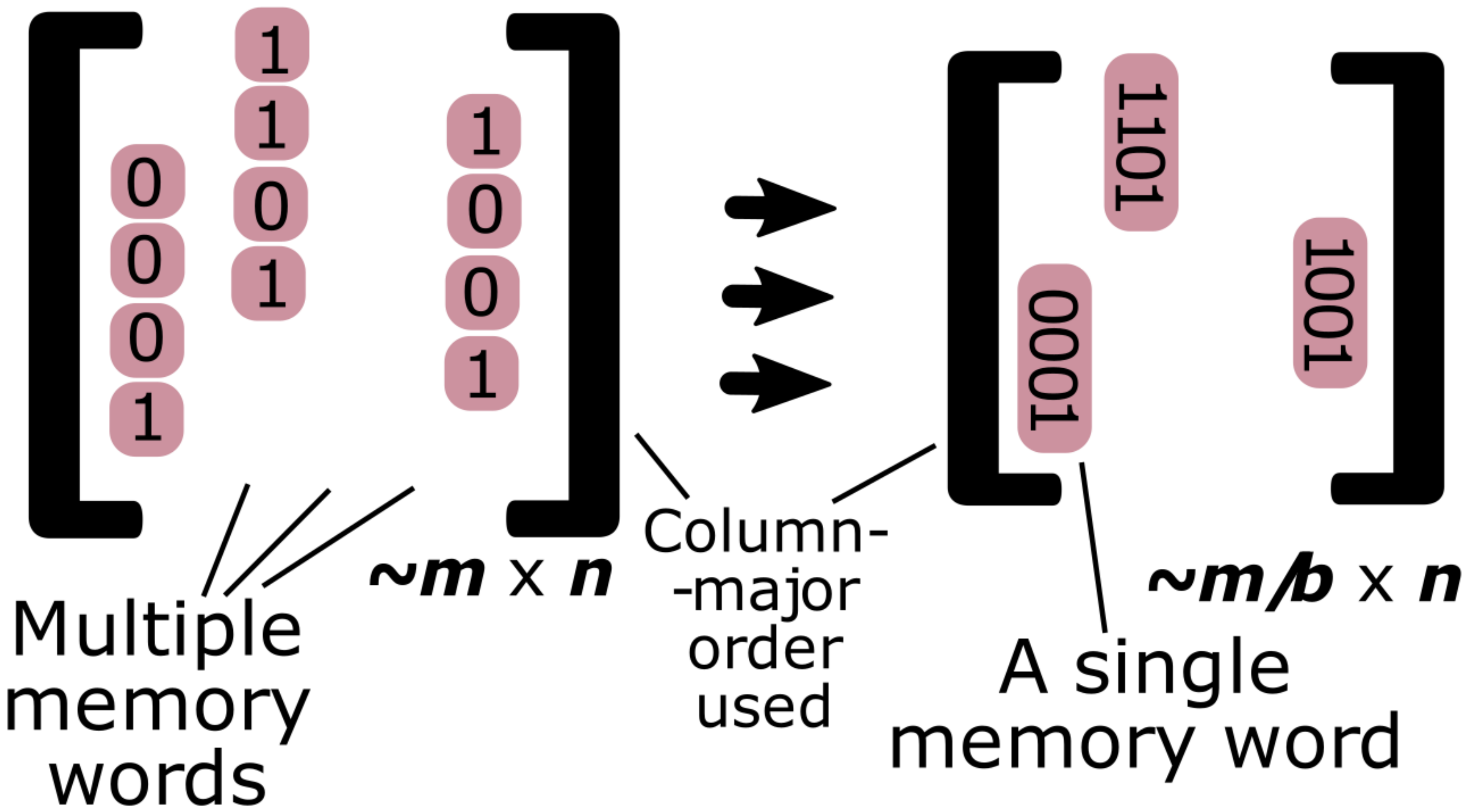


SimilarityAtScale: Compute intersection matrix $B = A^T A$

Technique 2: Compress (mask row segments into bit vectors)

SimilarityAtScale: Compute intersection matrix $B = A^T A$

Technique 2: Compress (mask row segments into bit vectors)

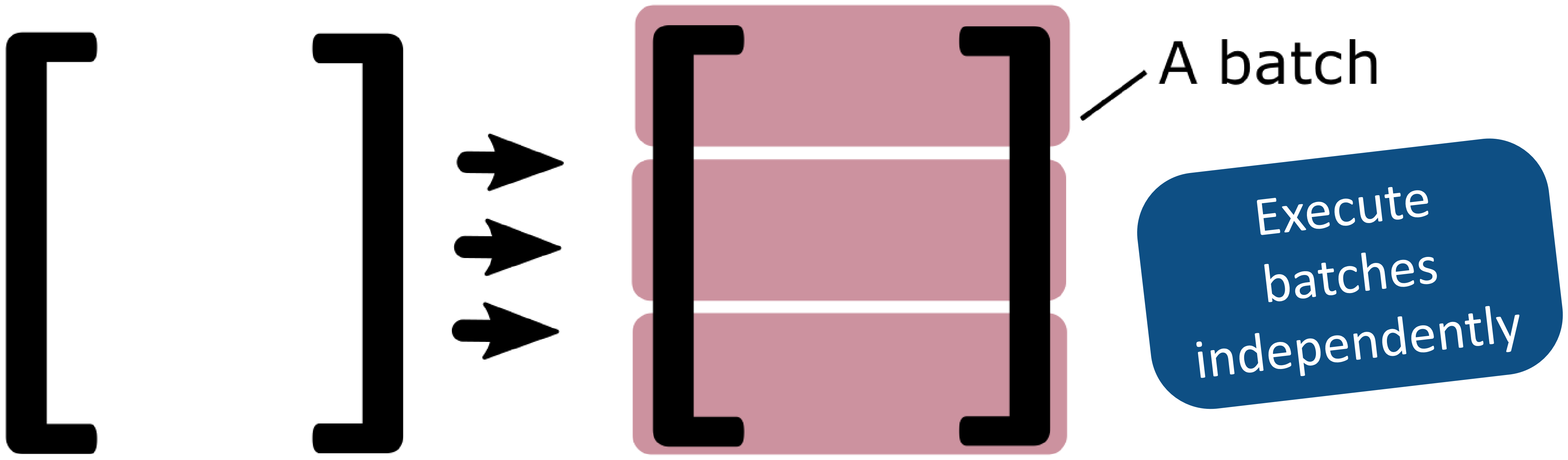


SimilarityAtScale: Compute intersection matrix $B = A^T A$

Technique 3: Divide into batches to further alleviate the large size of ***A***

SimilarityAtScale: Compute intersection matrix $B = A^T A$

Technique 3: Divide into batches to further alleviate the large size of ***A***



SimilarityAtScale: Compute intersection matrix $B = A^T A$

SimilarityAtScale: Compute intersection matrix $B = A^T A$

Full algebraic formulations
of all the techniques

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}^{(1)} \\ \vdots \\ \boldsymbol{A}^{(r)} \end{bmatrix}, \text{ where } \boldsymbol{A}^{(l)} \in \mathbb{B}^{\tilde{m}\times n}, \forall l \in \{1,\ldots,r\}.$$

$$\boldsymbol{B} = \sum_{l=1}^{r} \boldsymbol{A}^{(l)T}\boldsymbol{A}^{(l)}, \quad \hat{\boldsymbol{a}} = \sum_{l=1}^{r} \hat{\boldsymbol{a}}^{(l)}, \text{ where } \hat{a}_i^{(l)} = \sum_k a_{ki}^{(l)}$$

SimilarityAtScale: Compute intersection matrix $\boldsymbol{B} = \boldsymbol{A}^T\boldsymbol{A}$

Full algebraic formulations of all the techniques

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}^{(1)} \\ \vdots \\ \boldsymbol{A}^{(r)} \end{bmatrix}, \text{ where } \boldsymbol{A}^{(l)} \in \mathbb{B}^{\tilde{m} \times n}, \forall l \in \{1, \dots, r\}.$$

Theoretical analysis of communication cost

$$\boldsymbol{B} = \sum_{l=1}^{r} \boldsymbol{A}^{(l)T} \boldsymbol{A}^{(l)}, \quad \hat{\boldsymbol{a}} = \sum_{l=1}^{r} \hat{\boldsymbol{a}}^{(l)}, \text{ where } \hat{a}_i^{(l)} = \sum_k a_{ki}^{(l)}$$

$$O\left(\left(1 + \frac{z}{M\sqrt{cp}}\right) \cdot \alpha + \left(\frac{z}{\sqrt{cp}} + \frac{cn^2}{p}\right) \cdot \beta\right)$$

SimilarityAtScale: Compute intersection matrix $B = A^T A$

Full algebraic formulations of all the techniques

$$A = \begin{bmatrix} A^{(1)} \\ \vdots \\ A^{(r)} \end{bmatrix}, \text{ where } A^{(l)} \in \mathbb{B}^{\tilde{m} \times n}, \forall l \in \{1, \dots, r\}.$$

Theoretical analysis of communication cost

$$B = \sum_{l=1}^{r} A^{(l)T} A^{(l)}, \quad \hat{a} = \sum_{l=1}^{r} \hat{a}^{(l)}, \text{ where } \hat{a}_i^{(l)} = \sum_k a_{ki}^{(l)}$$

$$O\left(\left(1 + \frac{z}{M\sqrt{cp}}\right) \cdot \alpha + \left(\frac{z}{\sqrt{cp}} + \frac{cn^2}{p}\right) \cdot \beta\right)$$

Public implementation based on CTF [1]

[1] E. Solomonik, D. Matthews, J. R. Hammond, J. F. Stanton, and J. Demmel. A massively parallel tensor contraction framework for coupledcluster computations. JPDC, 2014

SIMILARITYATSCALE: COMPUTE INTERSECTION MATRIX $B = A^T A$

Full algebraic formulations of all the techniques

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}^{(1)} \\ \vdots \\ \boldsymbol{A}^{(r)} \end{bmatrix}, \text{ where } \boldsymbol{A}^{(l)} \in \mathbb{B}^{\tilde{m} \times n}, \forall l \in \{1, \ldots, r\}.$$

Theoretical analysis of communication cost

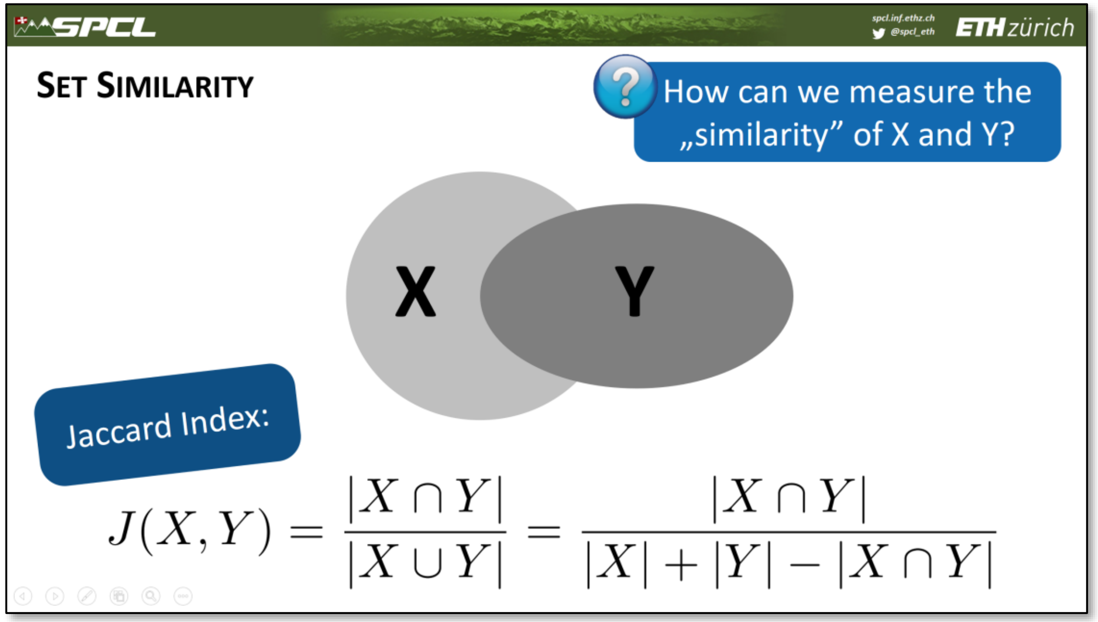
$$\boldsymbol{B} = \sum^{r} \boldsymbol{A}^{(l)T} \boldsymbol{A}^{(l)} \quad \ldots \quad \sum^{r} \quad \ldots \quad \text{e } \hat{a}_i^{(l)} = \sum_k a_{ki}^{(l)}$$

Check the paper for details ☺

$$O\left(\left(1 + \frac{z}{M\sqrt{cp}}\right) \cdot \alpha + \left(\frac{\quad}{\sqrt{cp}} + \frac{\quad n}{p}\right) \cdot \beta\right)$$

Public implementation based on CTF [1]

[1] E. Solomonik, D. Matthews, J. R. Hammond, J. F. Stanton, and J. Demmel. A massively parallel tensor contraction framework for coupledcluster computations. JPDC, 2014

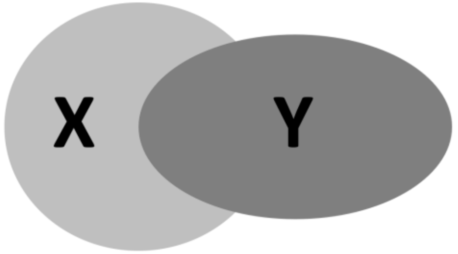


Part 1 „SimilarityAtScale": the first communication-efficient distributed algorithm to compute the general Jaccard similarity index and distance

SPCL spcl.inf.ethz.ch @spcl_eth ETH zürich

Set Similarity

How can we measure the „similarity" of X and Y?



Jaccard Index:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}$$

Part 1 „SimilarityAtScale": the first communication-efficient distributed algorithm to compute the general Jaccard similarity index and distance

Part 2 „GenomeAtScale": the first tool for fast, scalable, accurate, and large-scale derivations of Jaccard index between genome sequences

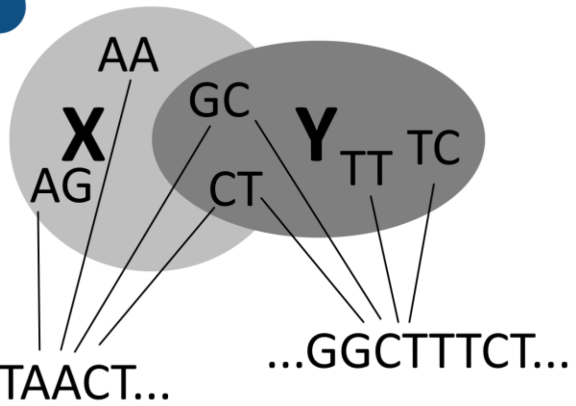
SPCL spcl.inf.ethz.ch @spcl_eth ETH zürich

Set Similarity

Sequences of genomes

What are X and Y in practice?

A: adenine,
T: thymine,
G: guanine,
C: cytosine



Genome sequence comparison: k-mers

Genome sequence comparison: k-mers

A: adenine,
T: thymine,
G: guanine,
C: cytosine

Example sequence

AGCTAACT

Genome sequence comparison: k-mers

A: adenine,
T: thymine,
G: guanine,
C: cytosine

Example sequence

AGCTAACT

2-mers: AG, GC, CT, TA, AA, AC, CT

GENOME SEQUENCE COMPARISON: K-MERS

A: adenine,
T: thymine,
G: guanine,
C: cytosine

Example sequence

AGCTAACT

2-mers: AG, GC, CT, TA, AA, AC, CT

3-mers: AGC, GCT, CTA, TAA, AAC, ACT

GENOME SEQUENCE COMPARISON: K-MERS

A: adenine,
T: thymine,
G: guanine,
C: cytosine

Example sequence

AGCTAACT

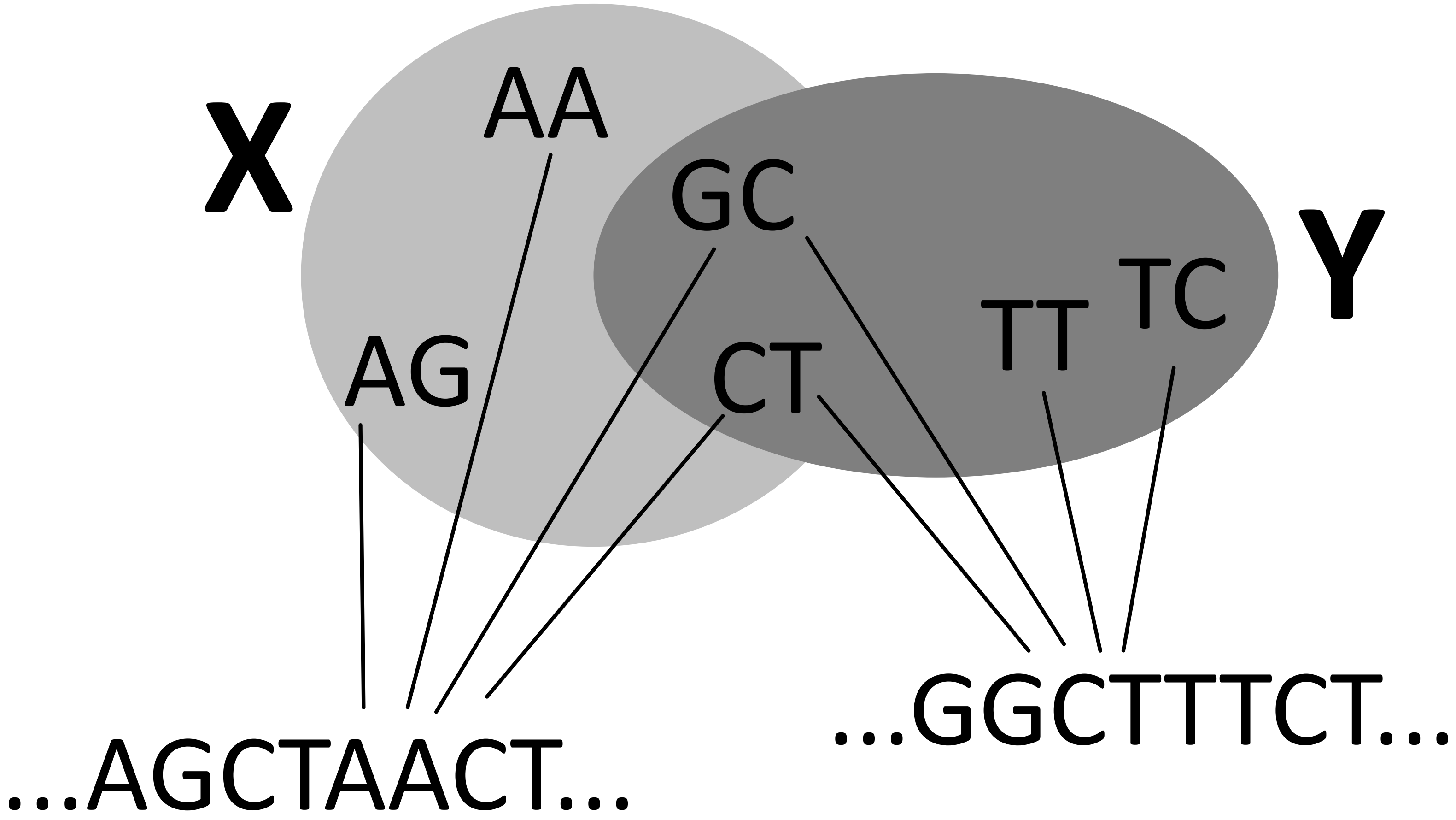
2-mers: AG, GC, CT, TA, AA, AC, CT

3-mers: AGC, GCT, CTA, TAA, AAC, ACT

⋮ ⋮

Genome sequence comparison based on k-mers

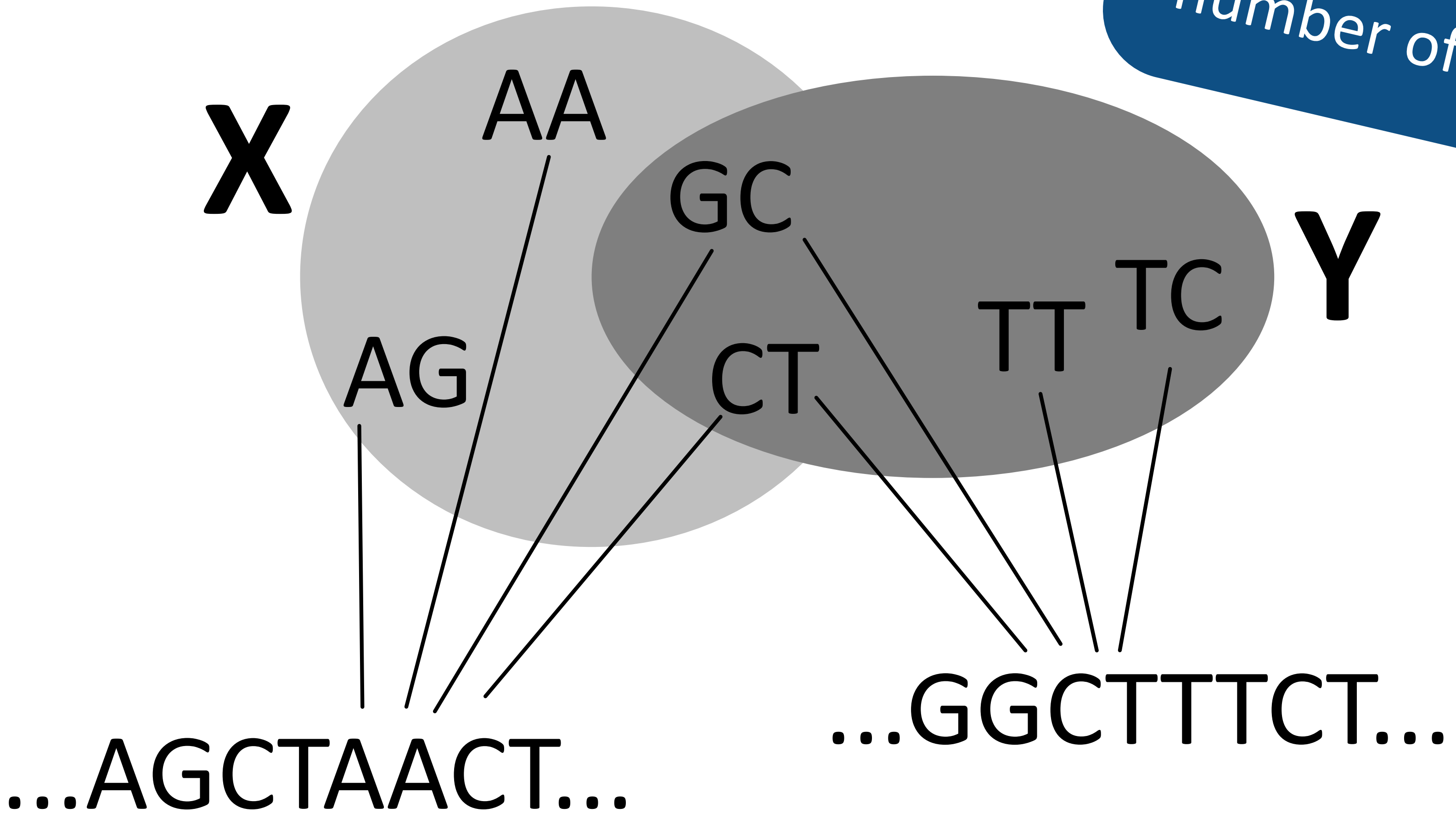
2-mers:



Genome sequence comparison based on k-mers

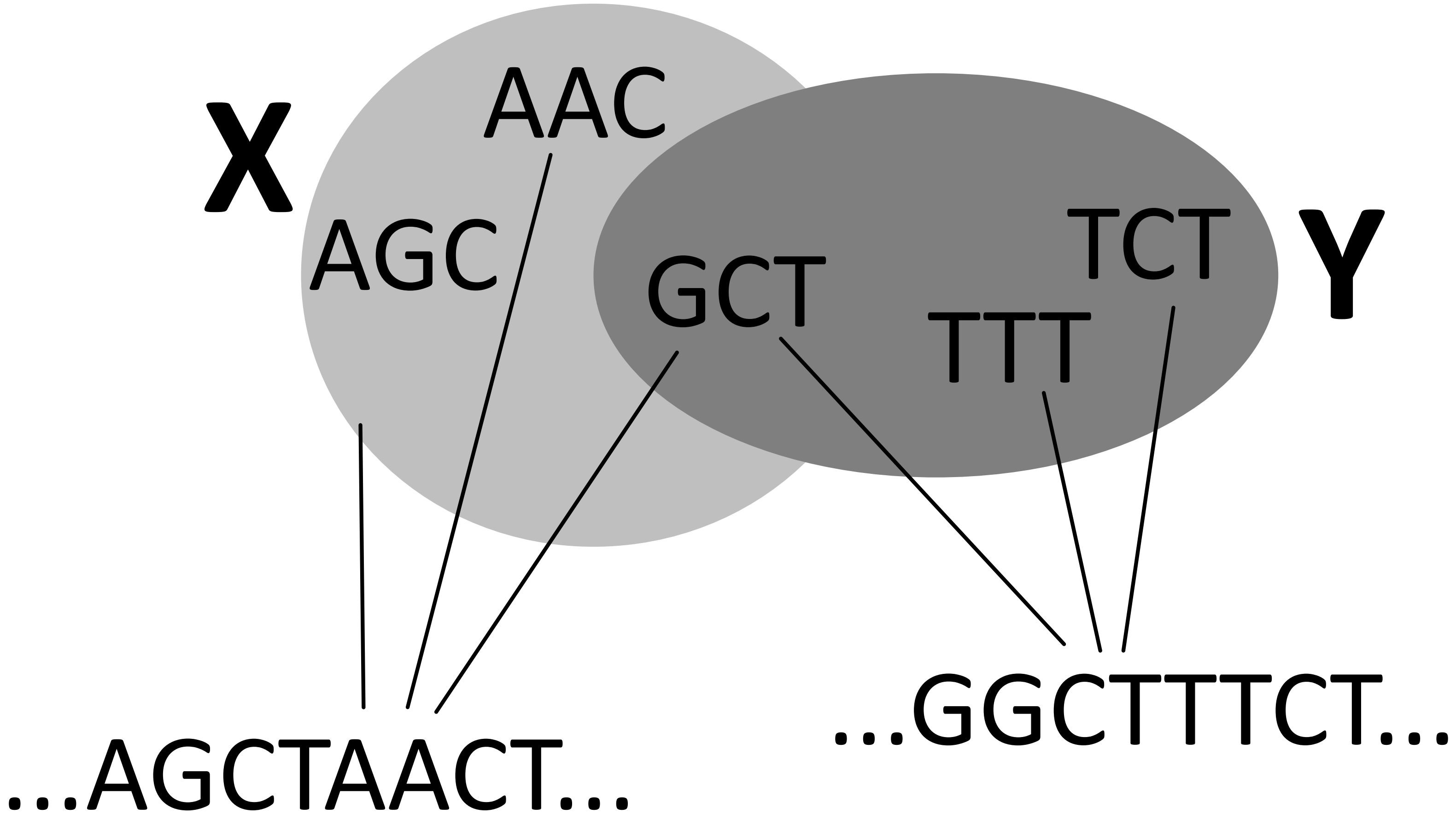
2-mers:

Similarity based on the number of common 2-mers



Genome sequence comparison based on k-mers

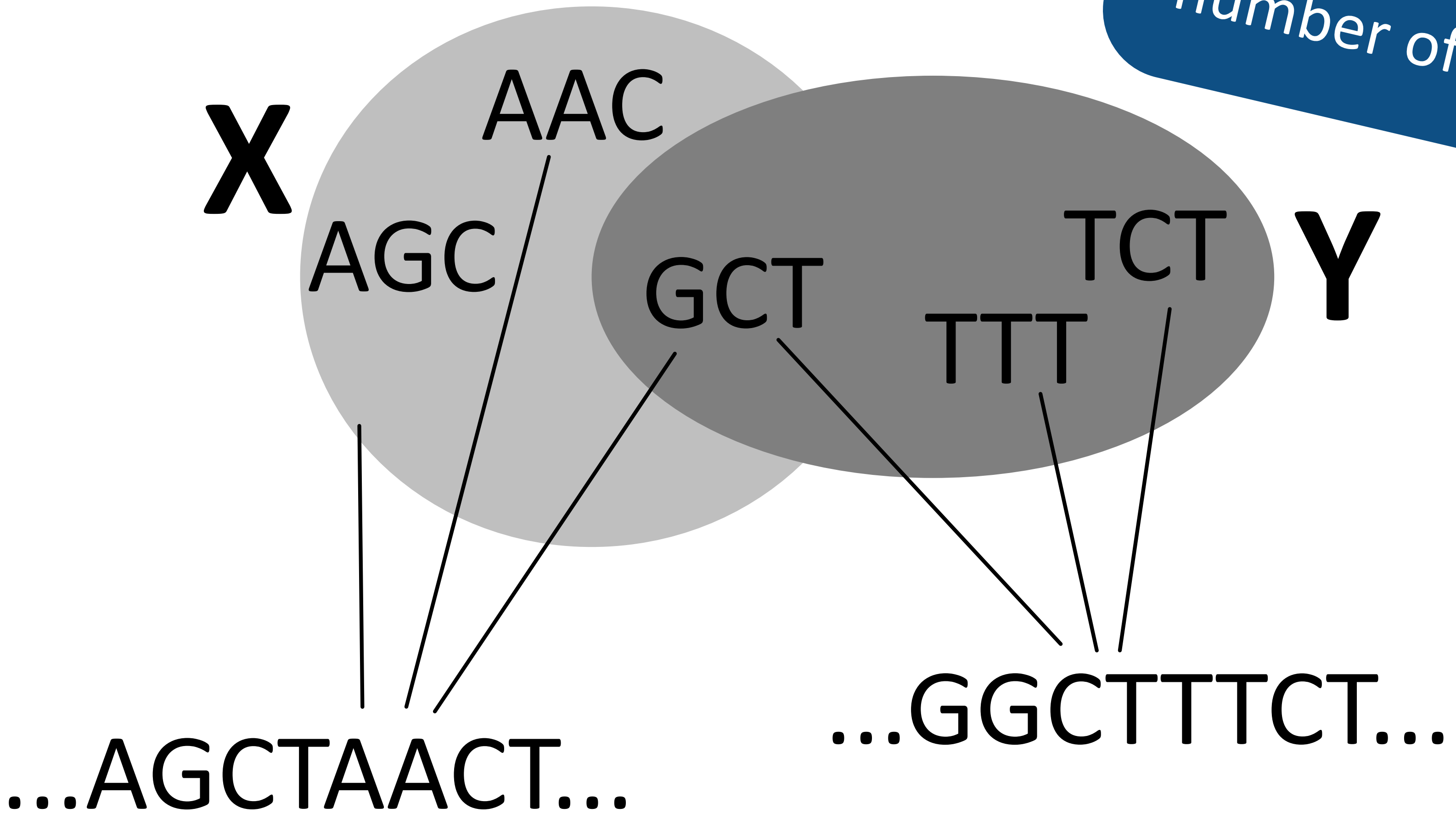
3-mers:



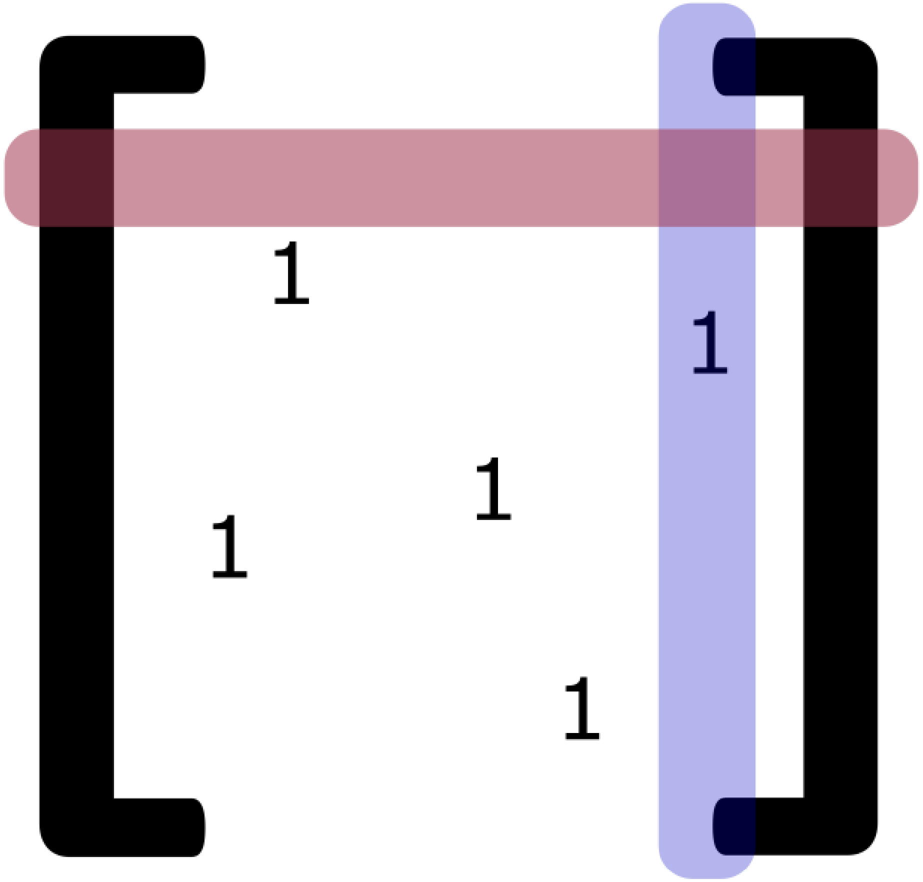
Genome sequence comparison based on k-mers

3-mers:

Similarity based on the number of common 3-mers

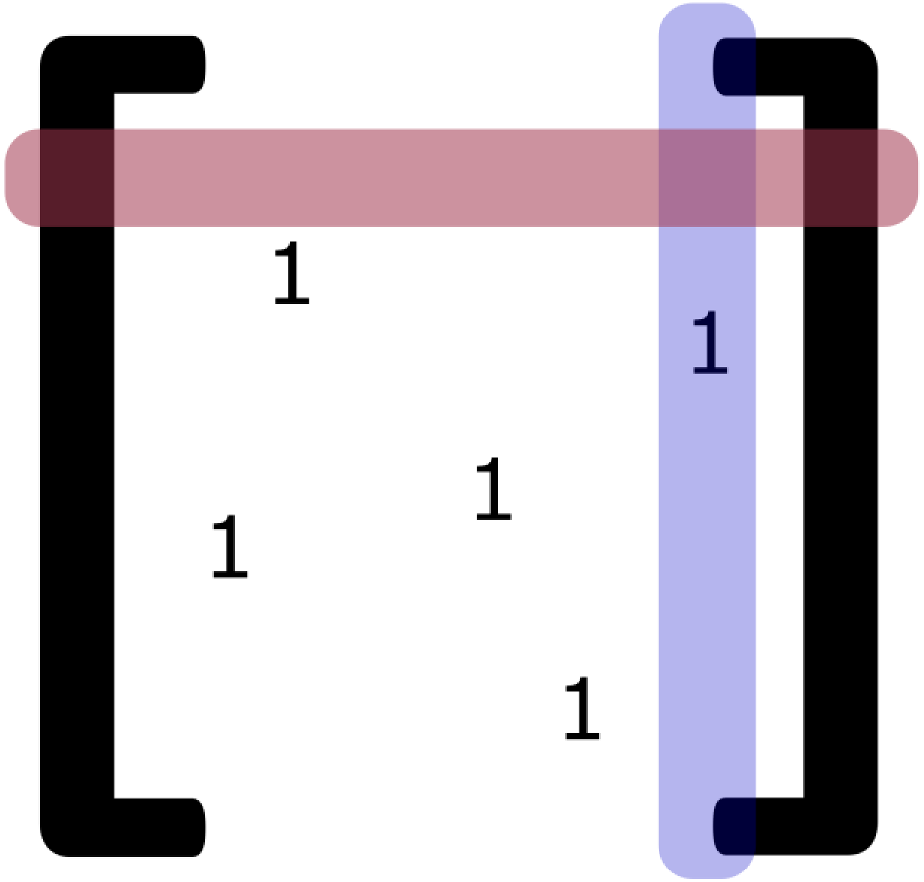


GenomeAtScale: algebraic representation of matrix A



GENOMEATSCALE: ALGEBRAIC REPRESENTATION OF MATRIX A

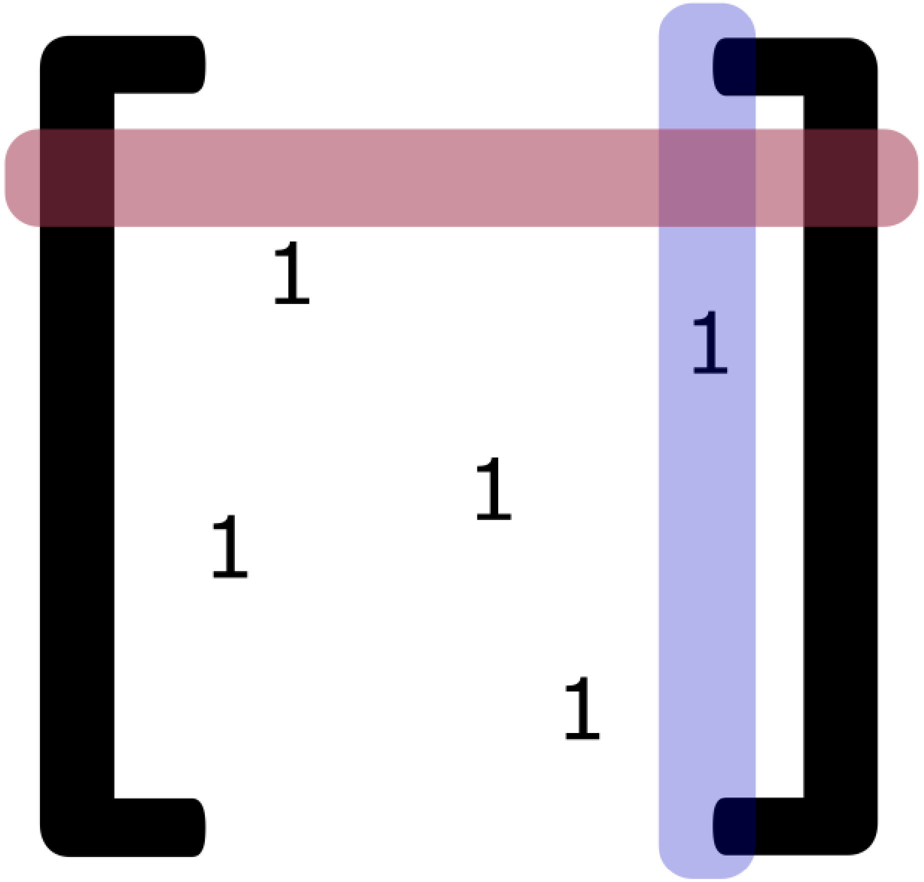
One **row** describes one sequence ***k*-mer**



GenomeAtScale: algebraic representation of matrix A

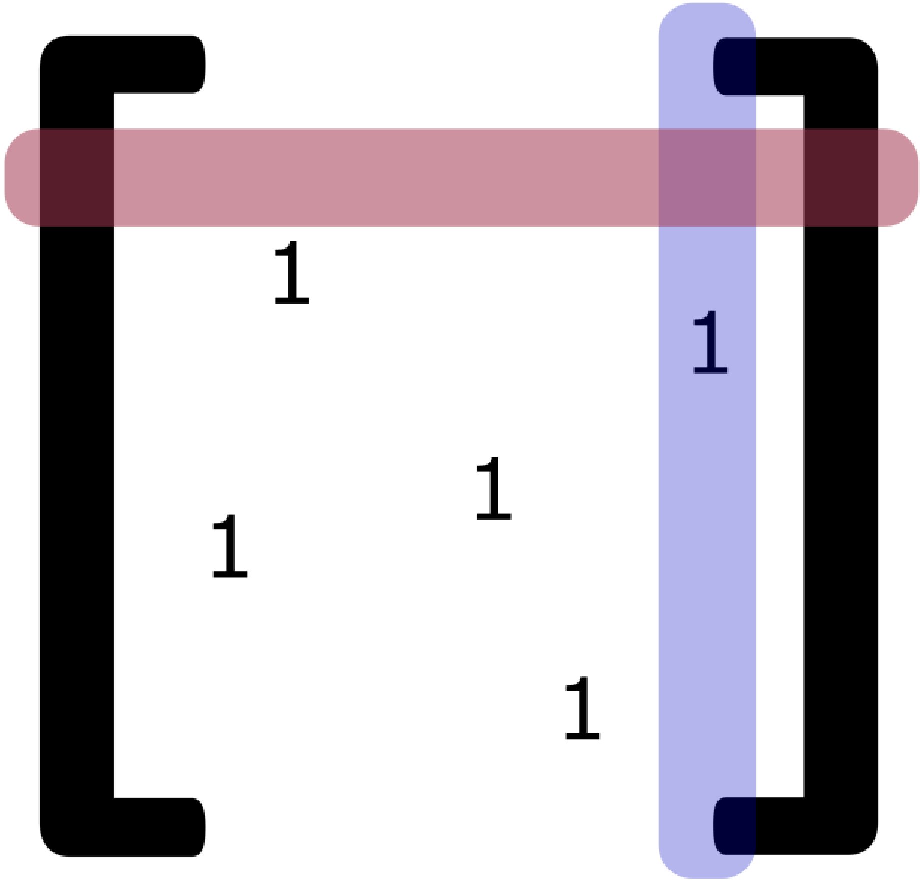
One **row** describes one sequence ***k*-mer**

One **column** describes one **data sample**



GenomeAtScale: algebraic representation of matrix A

One **row** describes one sequence ***k*-mer**



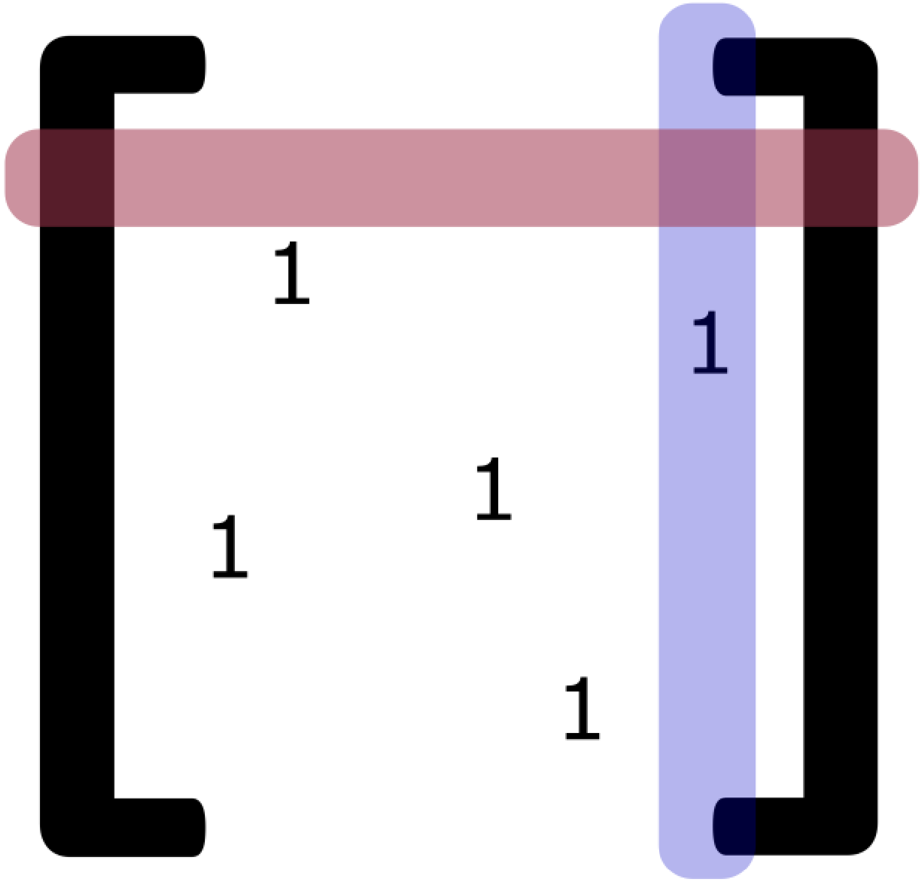
One **column** describes one **data sample**

"0": a given data sample does not contain a given *k*-mer

GenomeAtScale: algebraic representation of matrix A

One **row** describes one sequence ***k*-mer**

One **column** describes one **data sample**



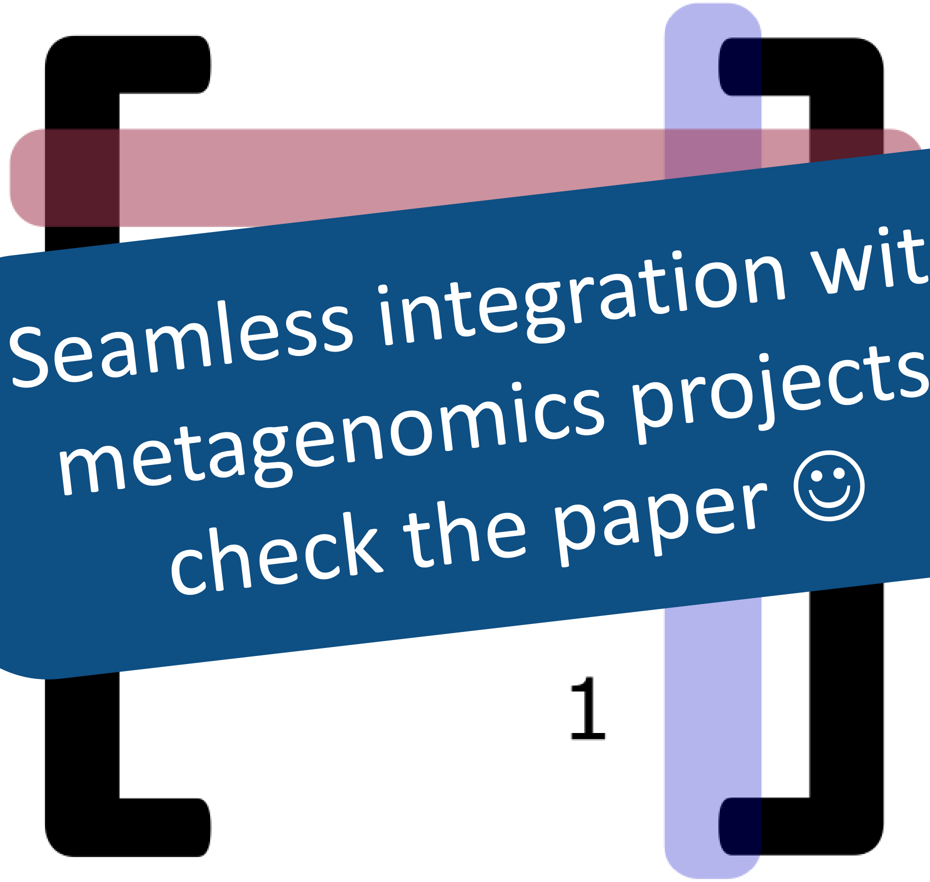
"0": a given data sample does not contain a given *k*-mer

"1": a given data sample contains a given *k*-mer

GenomeAtScale: Algebraic representation of matrix A

One **row** describes one sequence ***k*-mer**

One **column** describes one **data sample**



Seamless integration with metagenomics projects: check the paper ☺

"0": a given data sample does not contain a given *k*-mer

"1": a given data sample contains a given *k*-mer

GenomeAtScale: scale comparison

GenomeAtScale: scale comparison

Tool	# compute nodes	# samples	Raw input data size	Preprocessed data size	Similarity
DSM [71]	1	435	3.3TB	N/A‡	Jaccard
Mash [63]	1	54,118	N/A†	674GB	Jaccard (MinHash)
Libra [29]	10	40	372GB	N/A‡	Cosine
GenomeAtScale	1024	446,506	170TB	1.8TB	Jaccard

GENOMEATSCALE: SCALE COMPARISON

Tool	# compute nodes	# samples	Raw input data size	Preprocessed data size	Similarity
DSM [71]	1	435	3.3TB	N/A‡	Jaccard
Mash [63]	1	54, 118	N/A†	674GB	Jaccard (MinHash)
Libra [29]	10	40	372GB	N/A‡	Cosine
GenomeAtScale	1024	446, 506	170TB	1.8TB	Jaccard

GenomeAtScale: scale comparison

Tool	# compute nodes	# samples	Raw input data size	Preprocessed data size	Similarity
DSM [71]	1	435	3.3TB	N/A‡	Jaccard
Mash [63]	1	54, 118	N/A†	674GB	Jaccard (MinHash)
Libra [29]	10	40	372GB	N/A‡	Cosine
GenomeAtScale	1024	446, 506	170TB	1.8TB	Jaccard

GenomeAtScale: scale comparison

Tool	# compute nodes	# samples	Raw input data size	Preprocessed data size	Similarity
DSM [71]	1	435	3.3TB	N/A‡	Jaccard
Mash [63]	1	54, 118	N/A†	674GB	Jaccard (MinHash)
Libra [29]	10	40	372GB	N/A‡	Cosine
GenomeAtScale	1024	446, 506	170TB	1.8TB	Jaccard

GenomeAtScale: scale comparison

Tool	# compute nodes	# samples	Raw input data size	Preprocessed data size	Similarity
DSM [71]	1	435	3.3TB	N/A‡	Jaccard
Mash [63]	1	54, 118	N/A†	674GB	Jaccard (MinHash)
Libra [29]	10	40	372GB	N/A‡	Cosine
GenomeAtScale	1024	446, 506	170TB	1.8TB	Jaccard

GenomeAtScale: scale comparison

„GenomeAtScale" achieves larger problem size and parallelism scales than past approaches

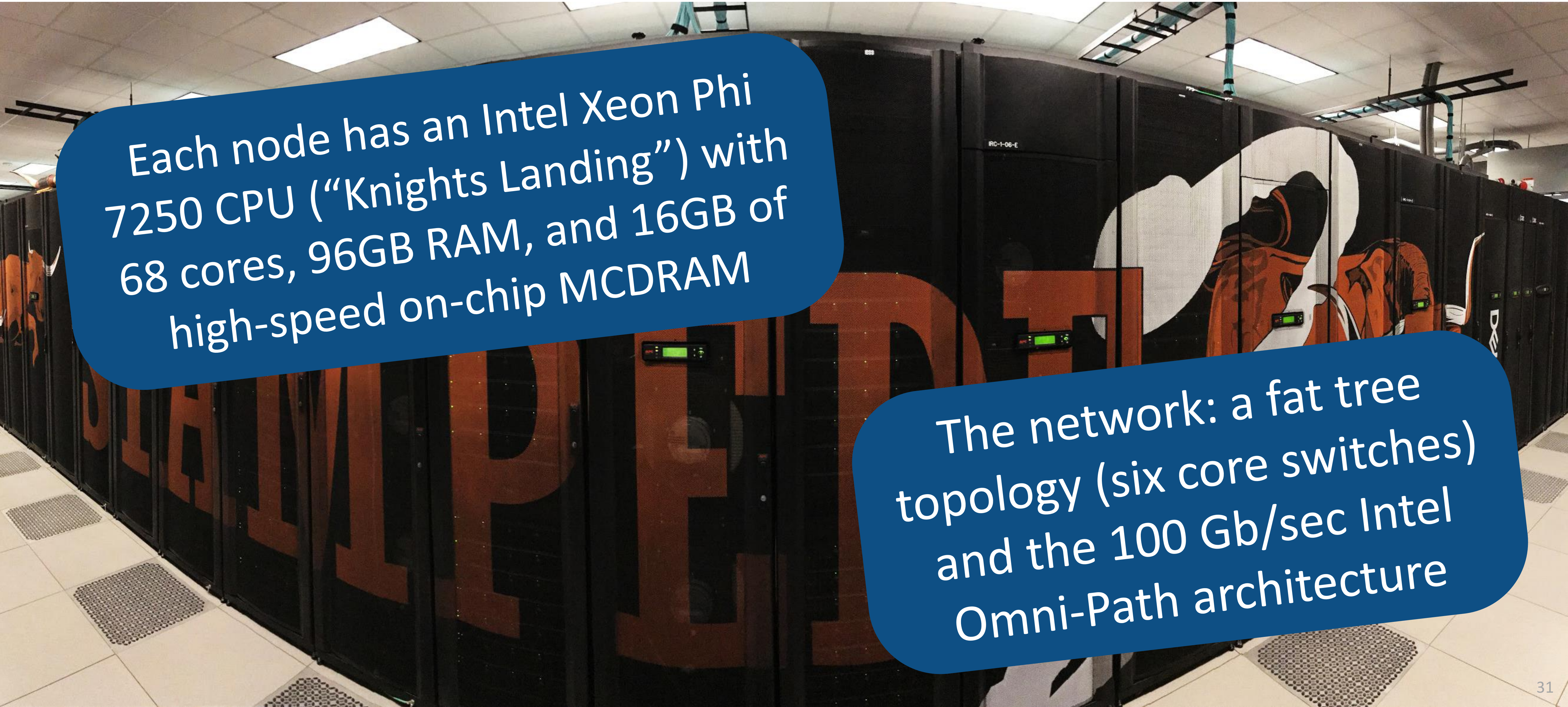
Tool	# compute nodes	# samples	Raw input data size	Preprocessed data size	Similarity
DSM [71]	1	435	3.3TB	N/A‡	Jaccard
Mash [63]	1	54,118	N/A†	674GB	Jaccard (MinHash)
Libra [29]	10	40	372GB	N/A‡	Cosine
GenomeAtScale	1024	446,506	170TB	1.8TB	Jaccard

PERFORMANCE ANALYSIS: STAMPEDE2 SUPERCOMPUTER

Performance analysis: Stampede2 supercomputer

Each node has an Intel Xeon Phi 7250 CPU (“Knights Landing”) with 68 cores, 96GB RAM, and 16GB of high-speed on-chip MCDRAM

Performance analysis: Stampede2 supercomputer



Each node has an Intel Xeon Phi 7250 CPU (“Knights Landing”) with 68 cores, 96GB RAM, and 16GB of high-speed on-chip MCDRAM

The network: a fat tree topology (six core switches) and the 100 Gb/sec Intel Omni-Path architecture

PERFORMANCE ANALYSIS: STATISTICAL ANALYSIS

PERFORMANCE ANALYSIS: STATISTICAL ANALYSIS

We calculate 95% confidence intervals for the reported mean values by assuming the batch times are normally distributed samples.

Performance analysis: statistical analysis

We calculate 95% confidence intervals for the reported mean values by assuming the batch times are normally distributed samples.

The derived confidence intervals are very tight around the means, and we exclude them from the plots to ensure clarity

Performance analysis: used datasets

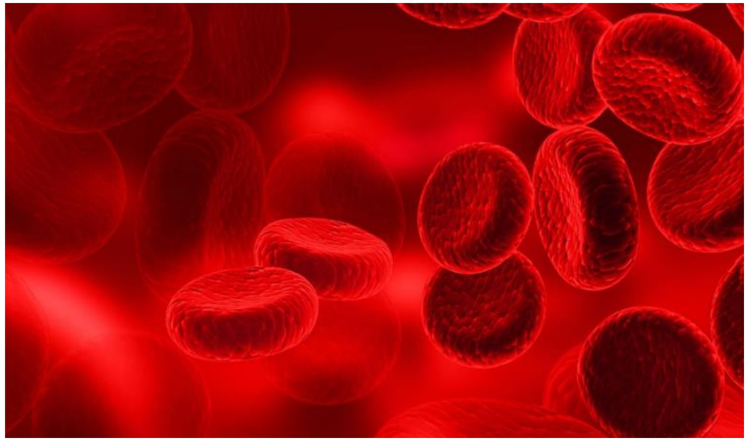
PERFORMANCE ANALYSIS: USED DATASETS

BBB/Kingsford [1]

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.

PERFORMANCE ANALYSIS: USED DATASETS

BBB/Kingsford [1]



[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.

PERFORMANCE ANALYSIS: USED DATASETS

BBB/Kingsford [1]

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.

PERFORMANCE ANALYSIS: USED DATASETS

BBB/Kingsford [1]

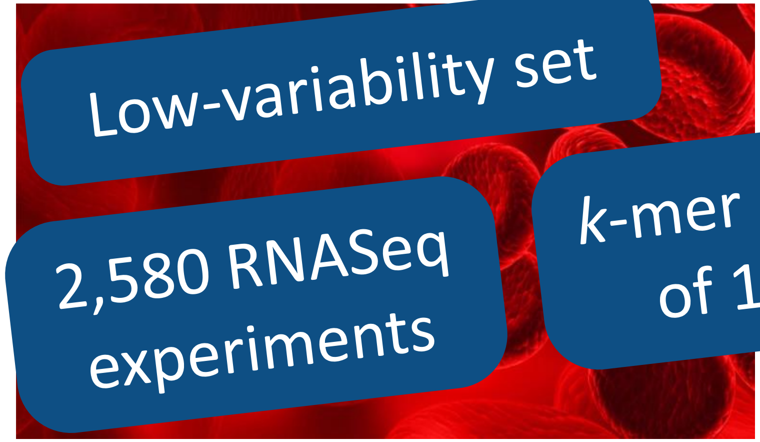
Low-variability set

2,580 RNASeq experiments

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.

Performance analysis: used datasets

BBB/Kingsford [1]



Low-variability set

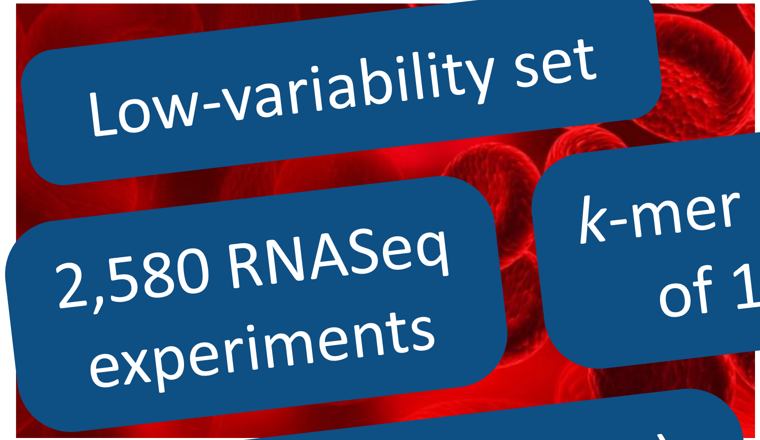
2,580 RNASeq experiments

k-mer size of 19

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.

PERFORMANCE ANALYSIS: USED DATASETS

BBB/Kingsford [1]



Low-variability set

2,580 RNASeq experiments

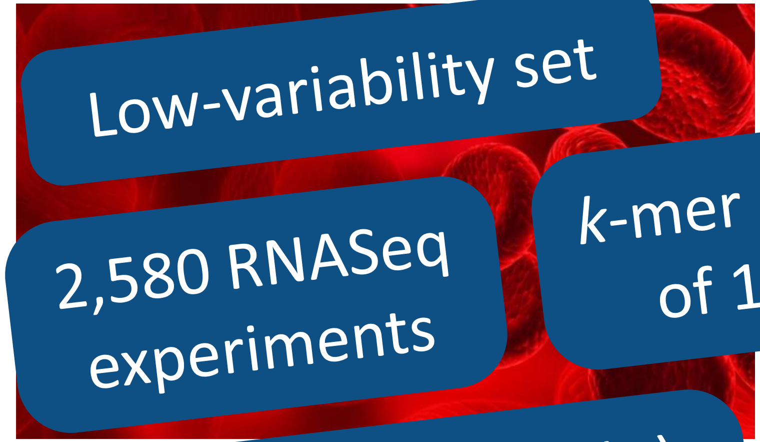
k-mer size of 19

Indicator matrix (A) sparsity: $1.5 \cdot 10^{-4}$

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.

PERFORMANCE ANALYSIS: USED DATASETS

BBB/Kingsford [1]



Low-variability set

2,580 RNASeq experiments

k-mer size of 19

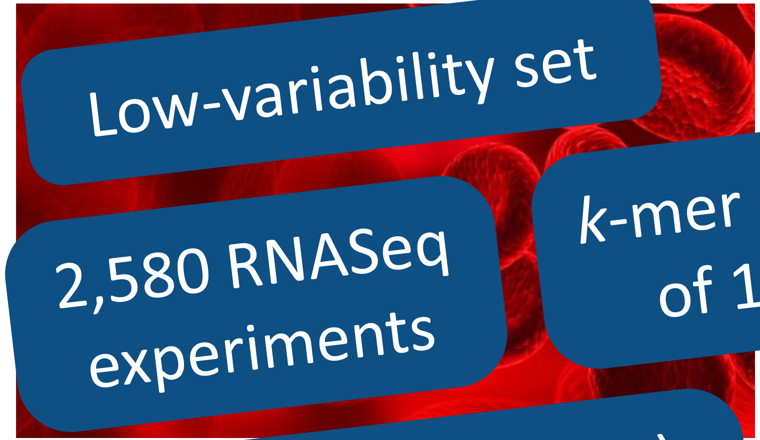
Indicator matrix (A) sparsity: $1.5 \cdot 10^{-4}$

BIGSI [2]

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.
[2] P. Bradley, H. C. den Bakker, E. P. Rocha, G. McVean, and Z. Iqbal. Ultrafast search of all deposited bacterial and viral genomic data. Nature biotechnology, 2019.

PERFORMANCE ANALYSIS: USED DATASETS

BBB/Kingsford [1]

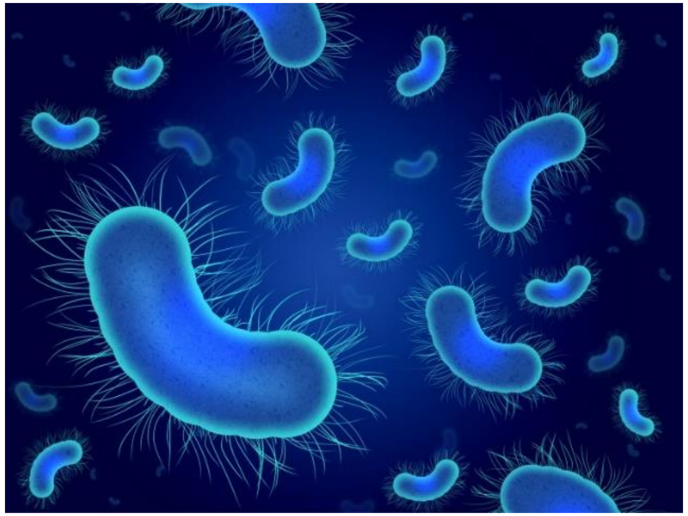


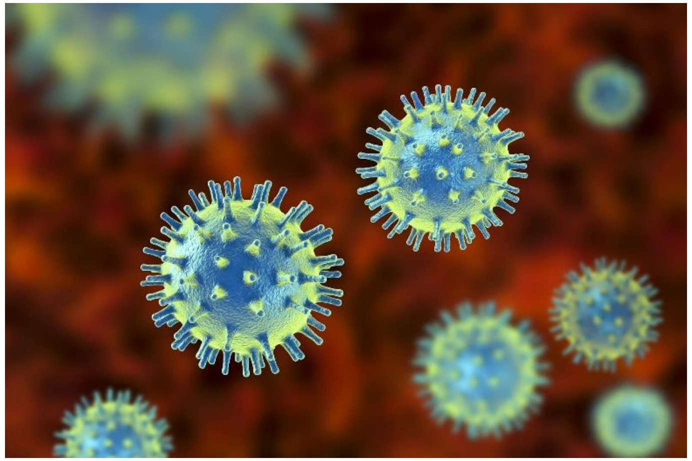
Low-variability set

2,580 RNASeq experiments

k-mer size of 19

Indicator matrix (A) sparsity: $1.5 \cdot 10^{-4}$

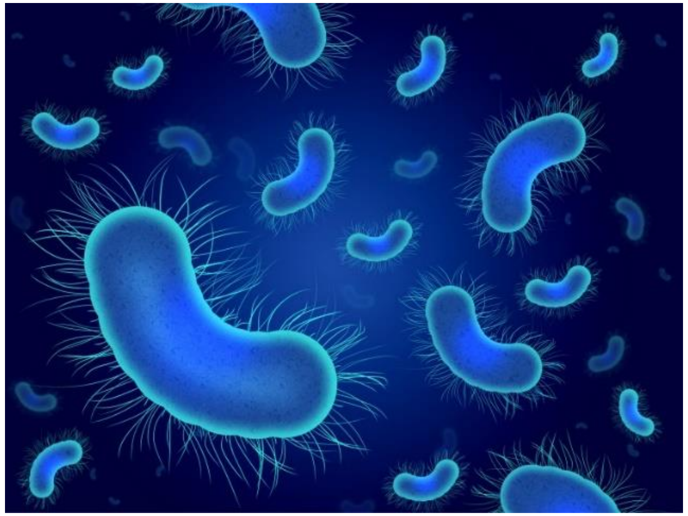




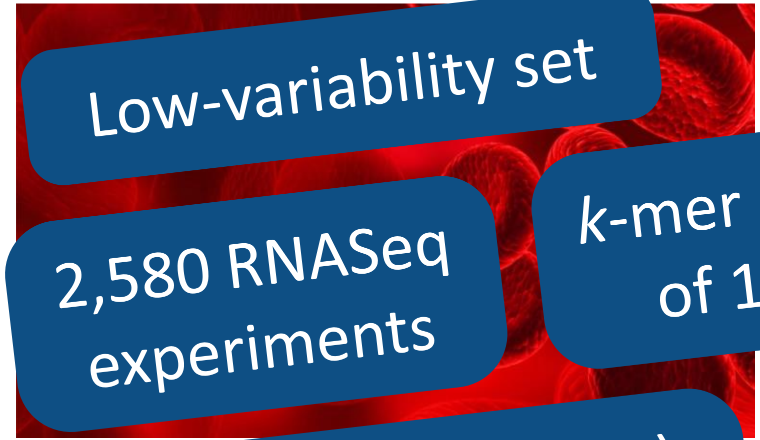
BIGSI [2]

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.
[2] P. Bradley, H. C. den Bakker, E. P. Rocha, G. McVean, and Z. Iqbal. Ultrafast search of all deposited bacterial and viral genomic data. Nature biotechnology, 2019.

PERFORMANCE ANALYSIS: USED DATASETS



BBB/Kingsford [1]

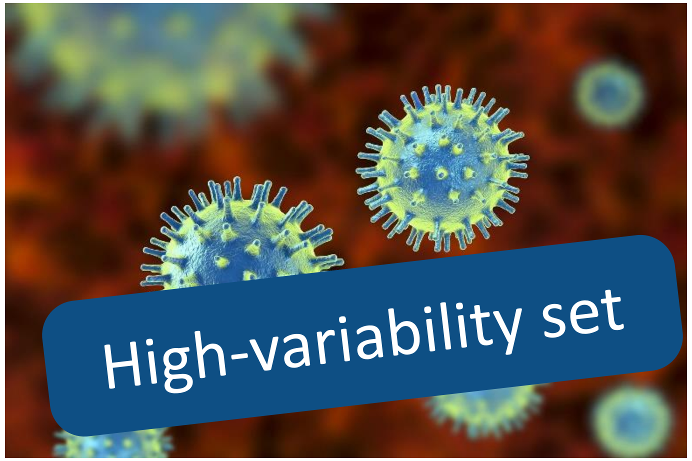


Low-variability set

2,580 RNASeq experiments

k-mer size of 19

Indicator matrix (A) sparsity: $1.5 \cdot 10^{-4}$

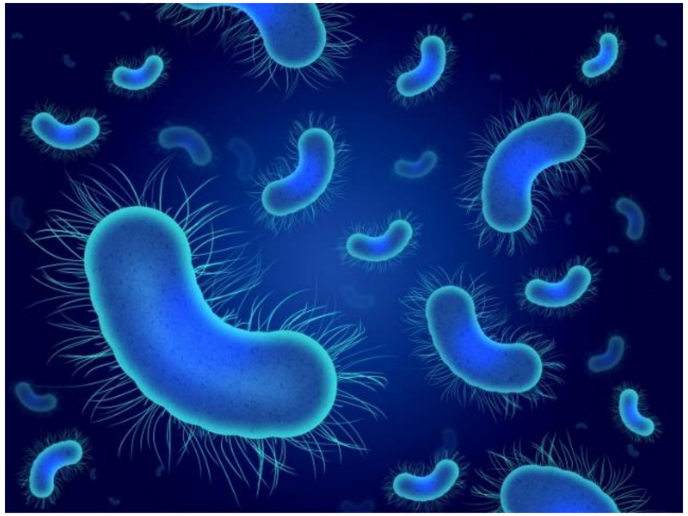


High-variability set

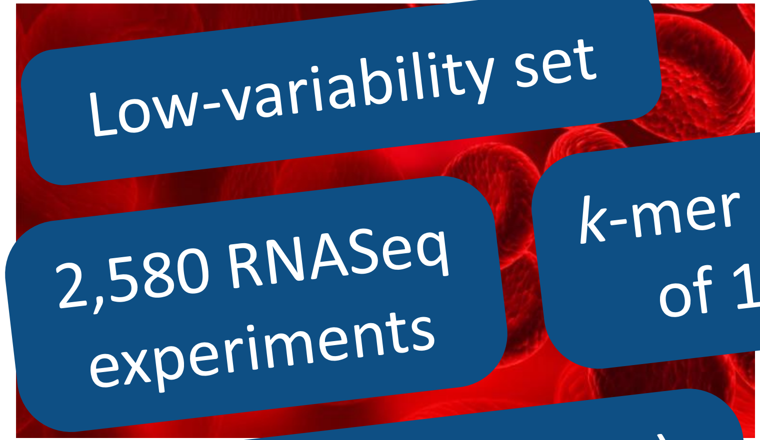
BIGSI [2]

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.
[2] P. Bradley, H. C. den Bakker, E. P. Rocha, G. McVean, and Z. Iqbal. Ultrafast search of all deposited bacterial and viral genomic data. Nature biotechnology, 2019.

PERFORMANCE ANALYSIS: USED DATASETS



BBB/Kingsford [1]

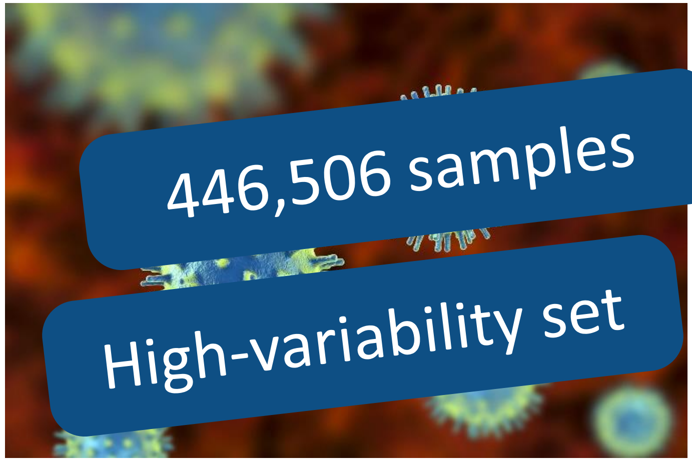


Low-variability set

2,580 RNASeq experiments

k-mer size of 19

Indicator matrix (A) sparsity: $1.5 \cdot 10^{-4}$



446,506 samples

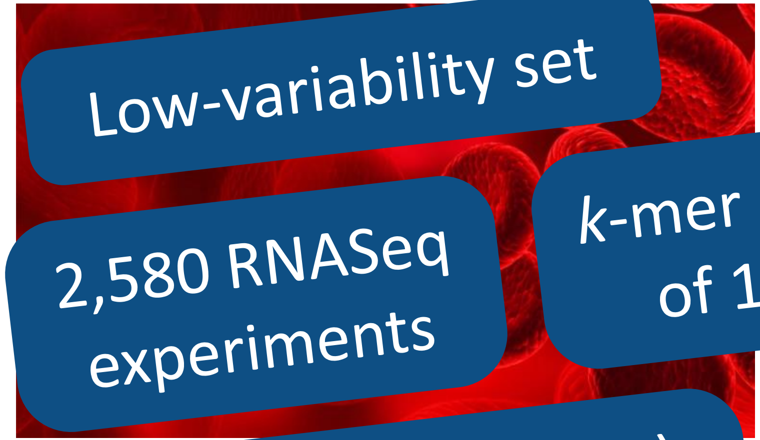
High-variability set

BIGSI [2]

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.
[2] P. Bradley, H. C. den Bakker, E. P. Rocha, G. McVean, and Z. Iqbal. Ultrafast search of all deposited bacterial and viral genomic data. Nature biotechnology, 2019.

PERFORMANCE ANALYSIS: USED DATASETS

BBB/Kingsford [1]

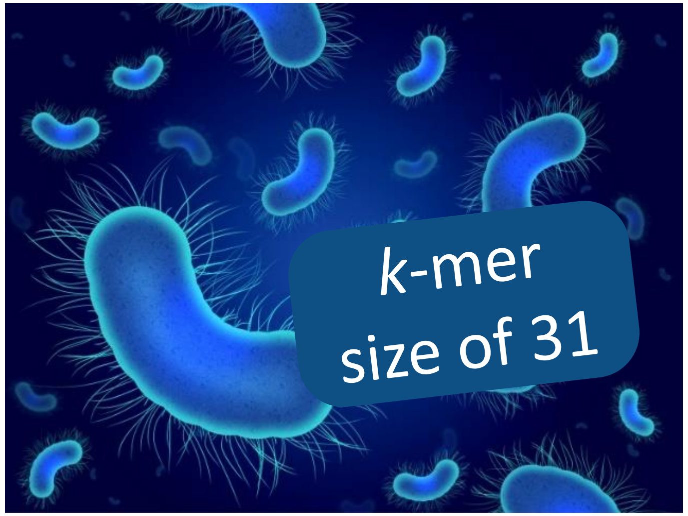


Low-variability set

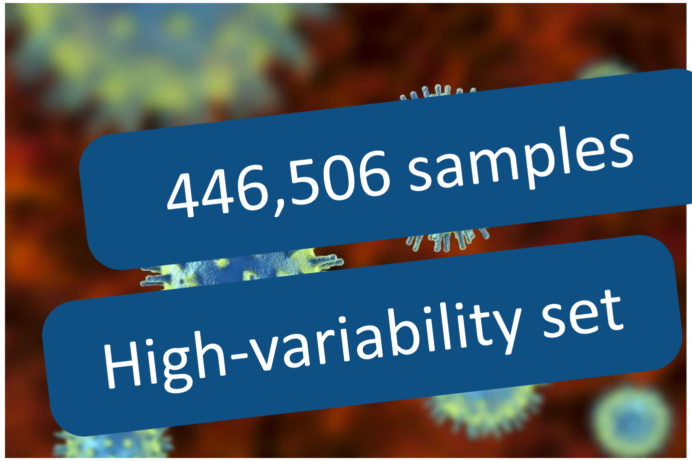
2,580 RNASeq experiments

k-mer size of 19

Indicator matrix (A) sparsity: $1.5 \cdot 10^{-4}$



k-mer size of 31



446,506 samples

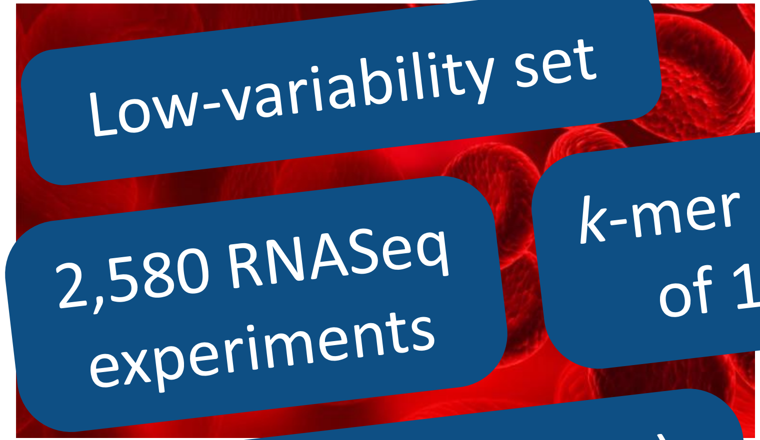
High-variability set

BIGSI [2]

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.
[2] P. Bradley, H. C. den Bakker, E. P. Rocha, G. McVean, and Z. Iqbal. Ultrafast search of all deposited bacterial and viral genomic data. Nature biotechnology, 2019.

PERFORMANCE ANALYSIS: USED DATASETS

BBB/Kingsford [1]

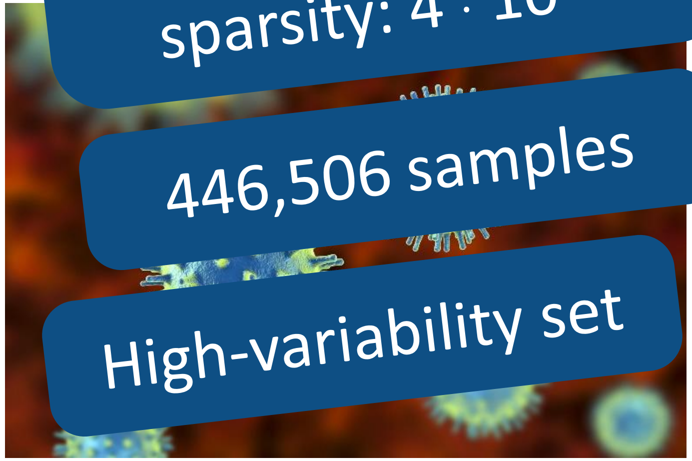


Low-variability set

2,580 RNASeq experiments

k-mer size of 19

Indicator matrix (A) sparsity: $1.5 \cdot 10^{-4}$



k-mer size of 31

Indicator matrix (A) sparsity: $4 \cdot 10^{-12}$

446,506 samples

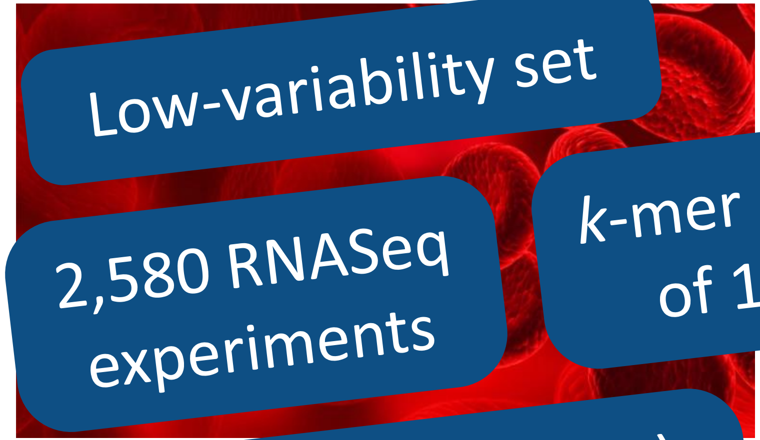
High-variability set

BIGSI [2]

[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.
[2] P. Bradley, H. C. den Bakker, E. P. Rocha, G. McVean, and Z. Iqbal. Ultrafast search of all deposited bacterial and viral genomic data. Nature biotechnology, 2019.

Performance analysis: used datasets

BBB/Kingsford [1]



Low-variability set

2,580 RNASeq experiments

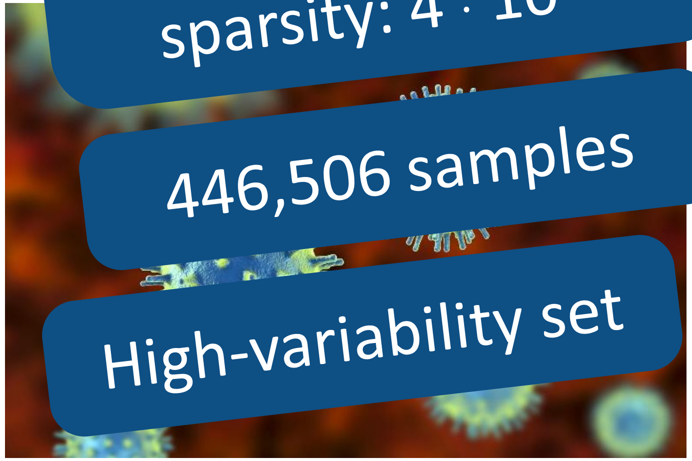
k-mer size of 19

Indicator matrix (A) sparsity: $1.5 \cdot 10^{-4}$

Synthetic data

Indicator matrix (A) with a pre-determined sparsity p

BIGSI [2]



k-mer size of 31

Indicator matrix (A) sparsity: $4 \cdot 10^{-12}$

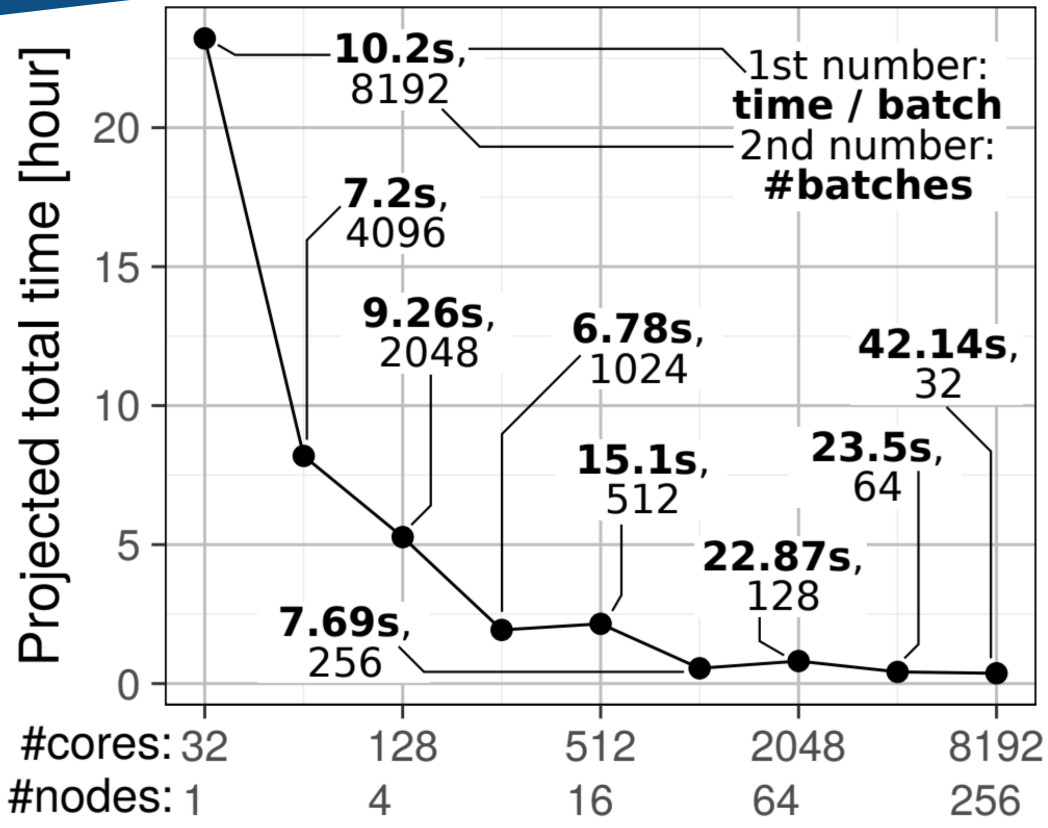
446,506 samples

High-variability set

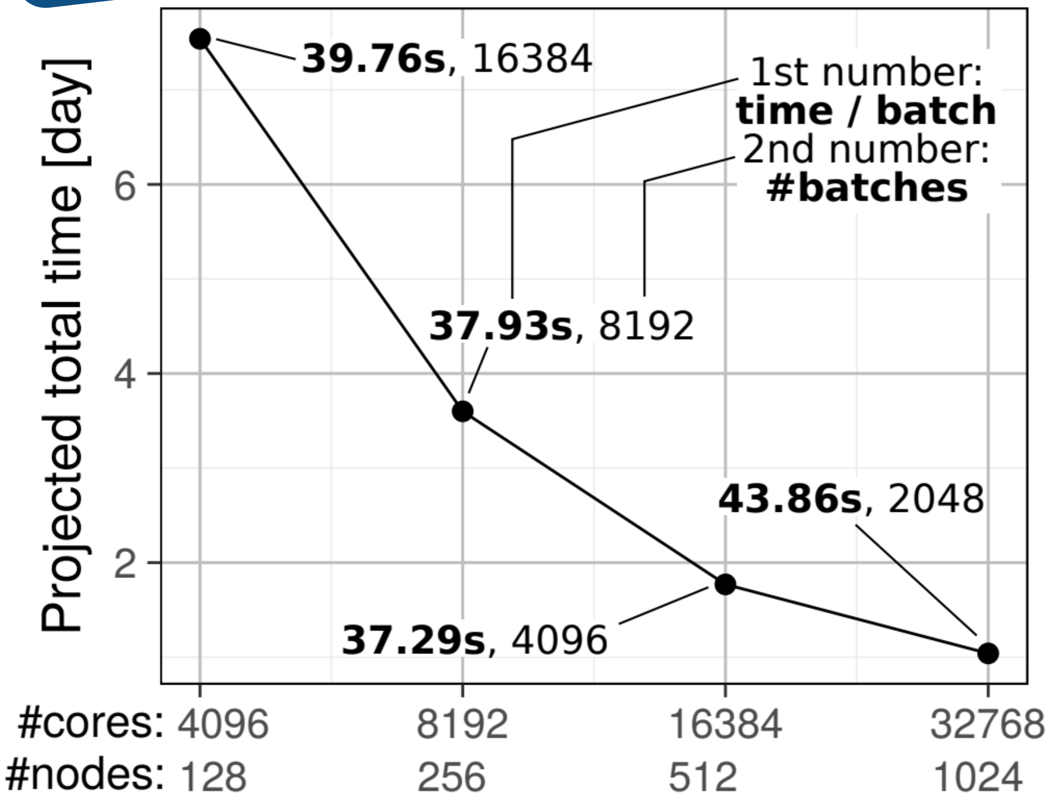
[1] B. Solomon and C. Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 2016.
[2] P. Bradley, H. C. den Bakker, E. P. Rocha, G. McVean, and Z. Iqbal. Ultrafast search of all deposited bacterial and viral genomic data. Nature biotechnology, 2019.

Performance analysis: real data, strong scaling

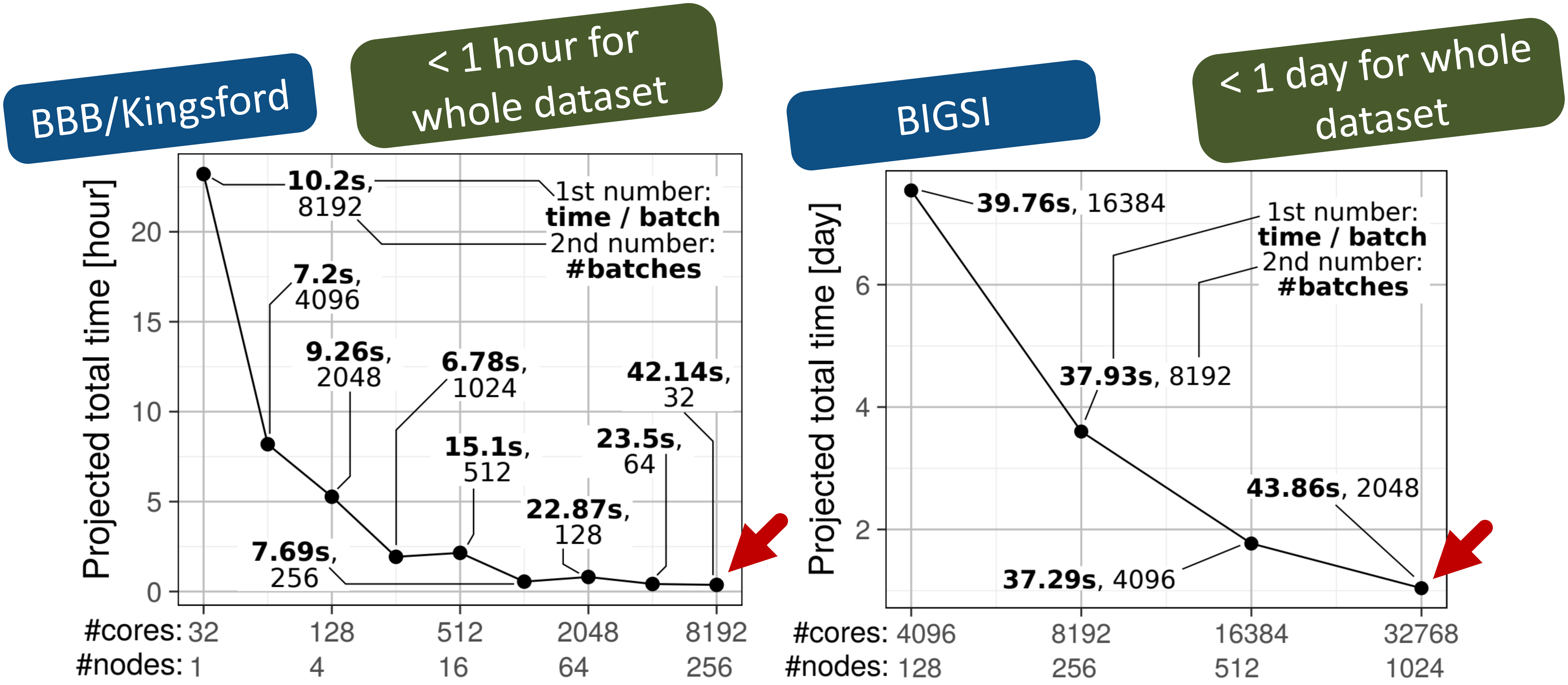
BBB/Kingsford



BIGSI



Performance analysis: real data, strong scaling



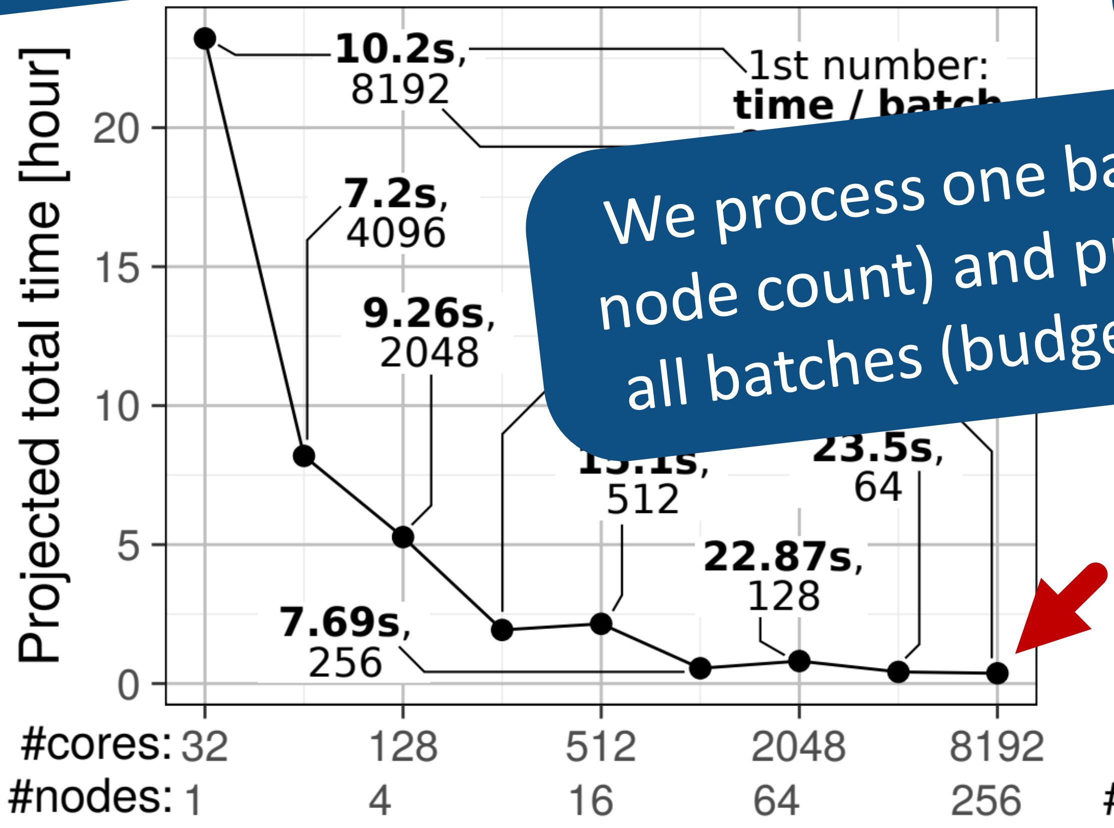
PERFORMANCE ANALYSIS: REAL DATA, STRONG SCALING

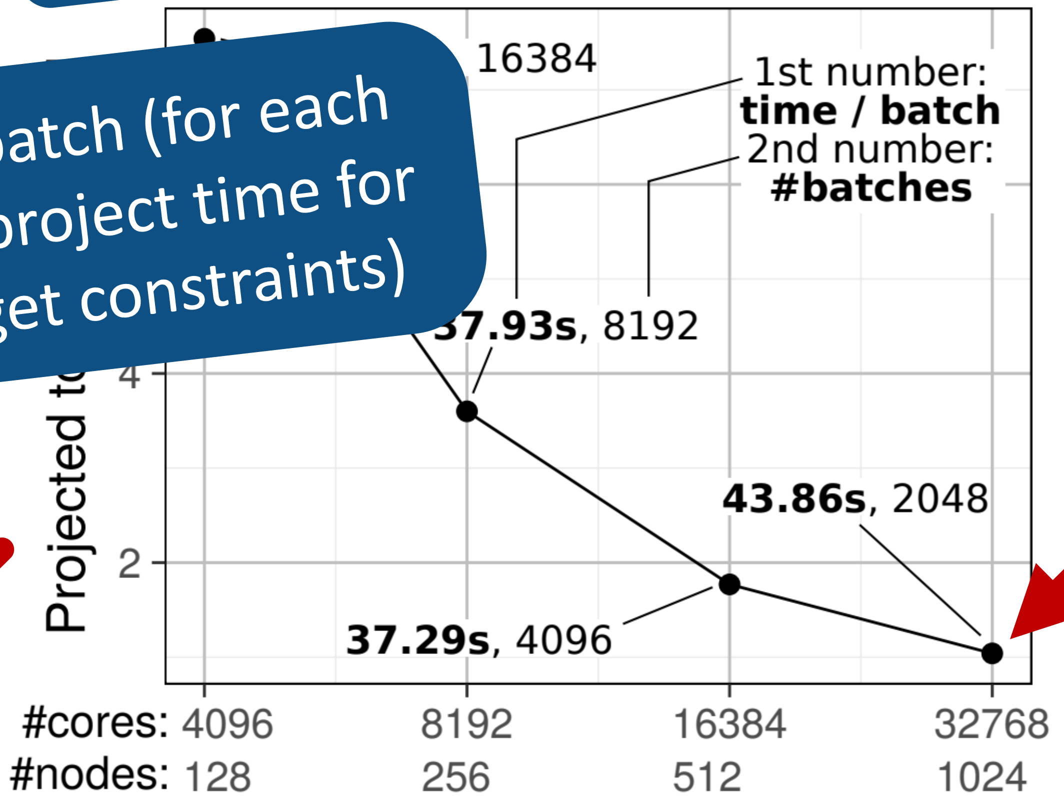
BBB/Kingsford

< 1 hour for whole dataset

BIGSI

< 1 day for whole dataset





We process one batch (for each node count) and project time for all batches (budget constraints)

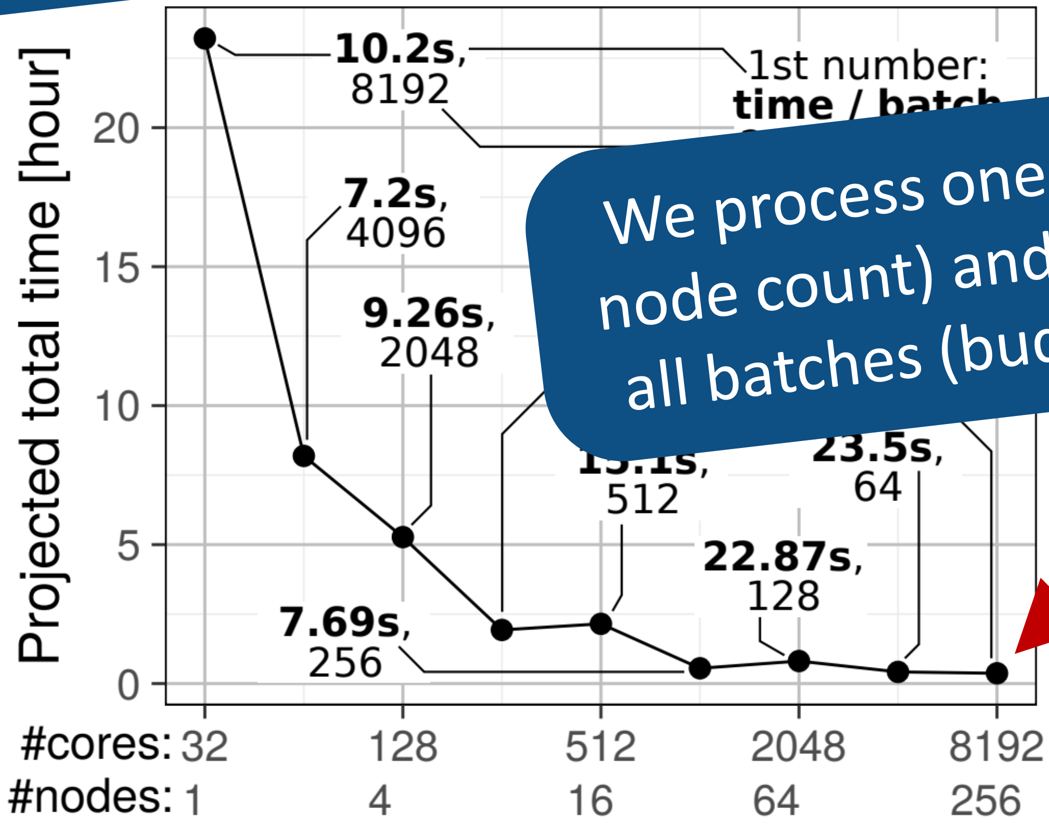
Performance analysis: real data, strong scaling

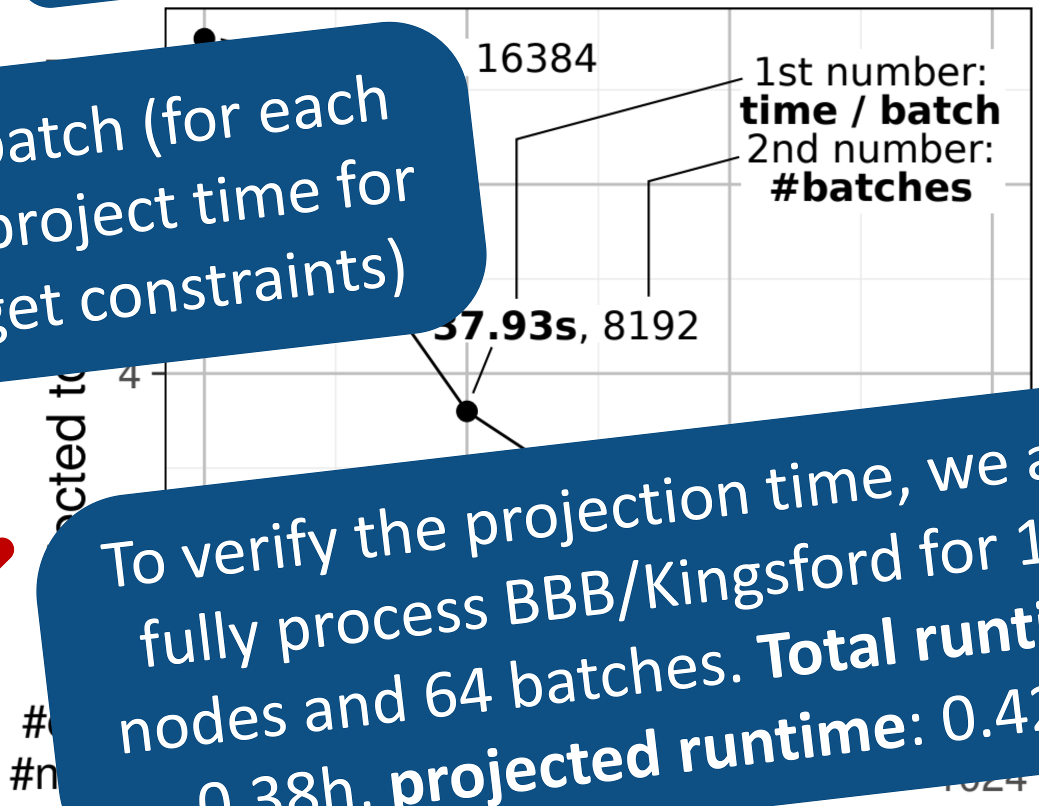
BBB/Kingsford

< 1 hour for whole dataset

BIGSI

< 1 day for whole dataset



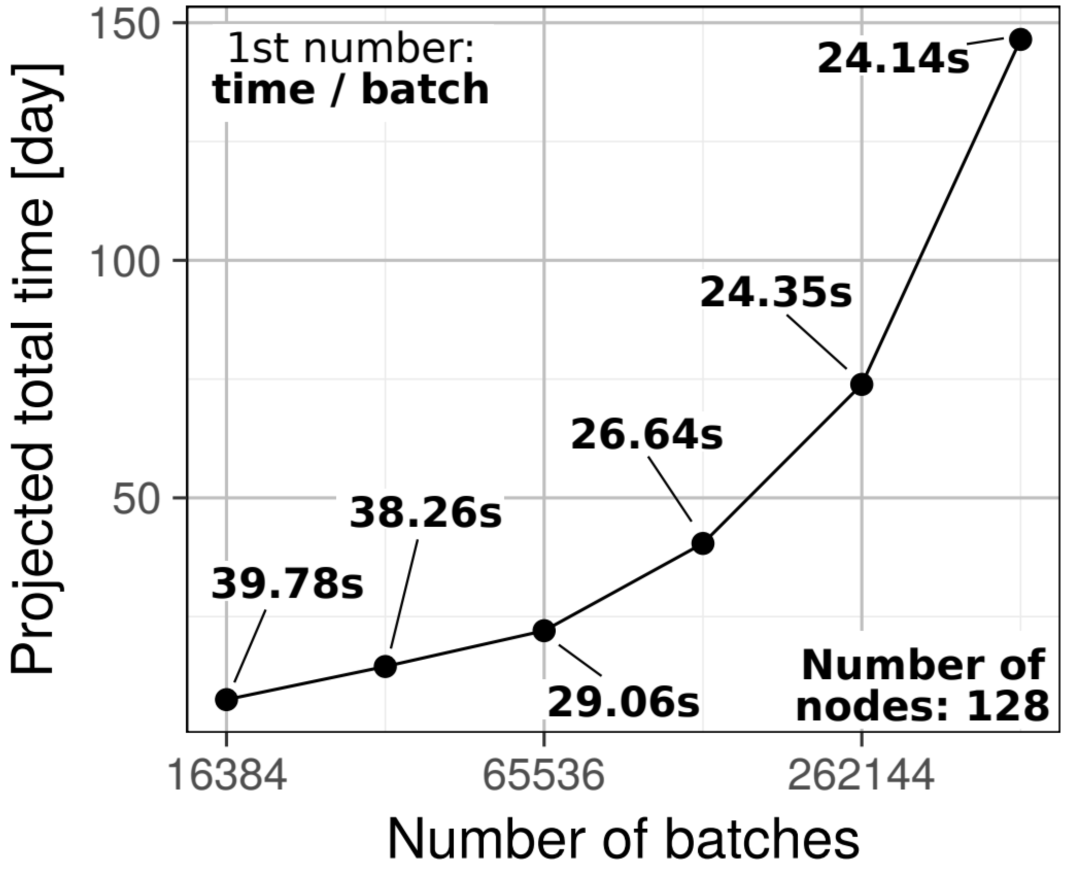


We process one batch (for each node count) and project time for all batches (budget constraints)

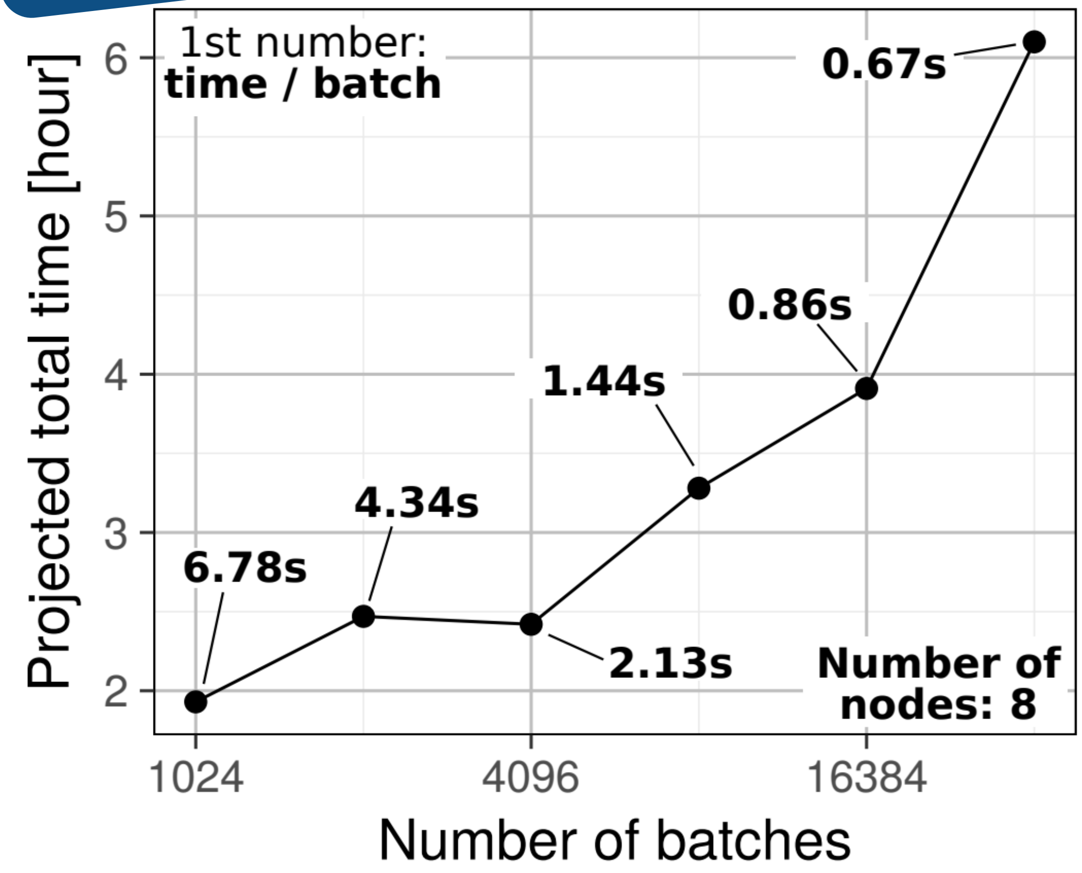
To verify the projection time, we also fully process BBB/Kingsford for 128 nodes and 64 batches. **Total runtime**: 0.38h, **projected runtime**: 0.42h

PERFORMANCE ANALYSIS: REAL DATA, SENSITIVITY ANALYSIS (#BATCHES)

BBB/Kingsford

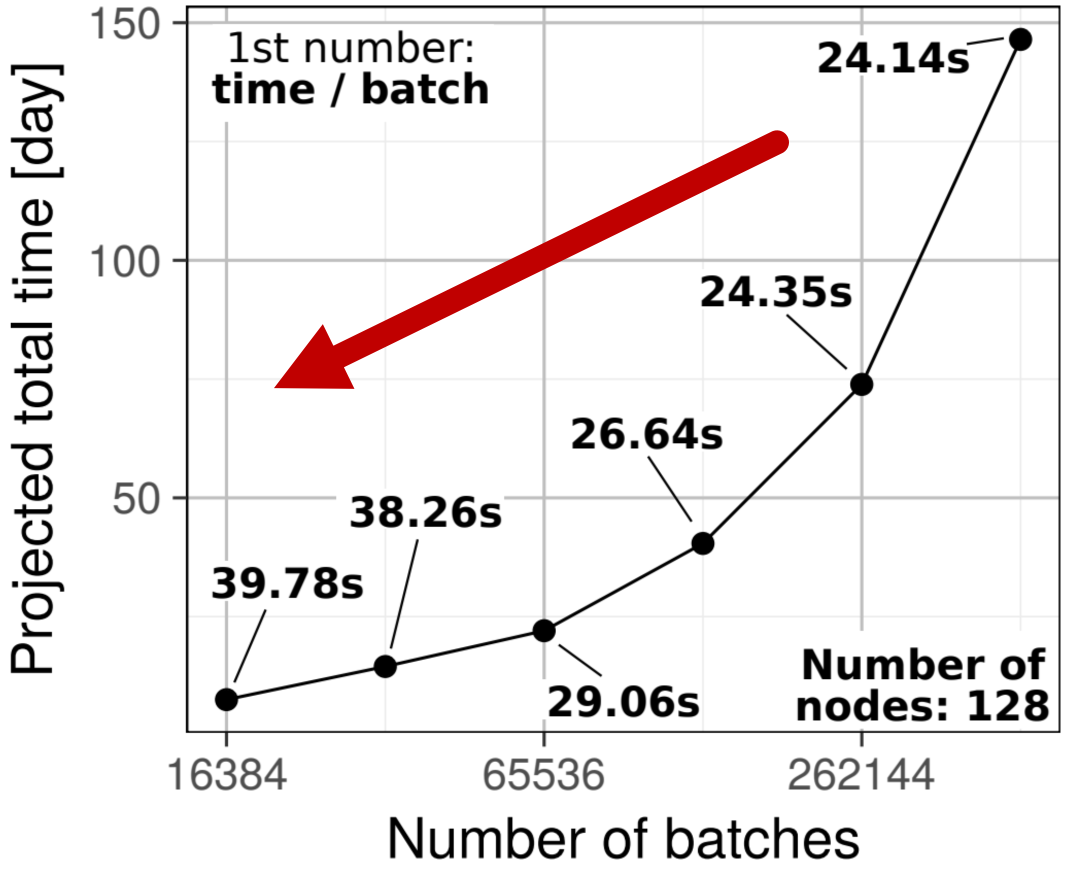


BIGSI

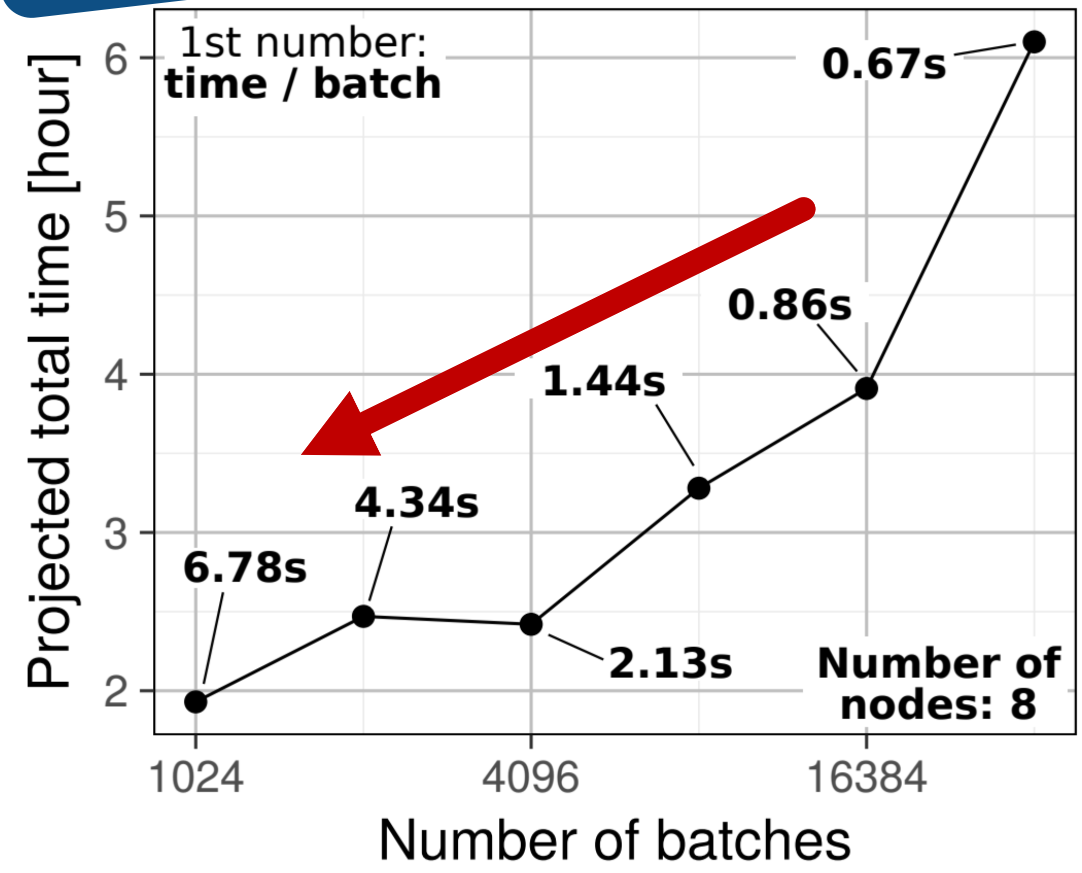


PERFORMANCE ANALYSIS: REAL DATA, SENSITIVITY ANALYSIS (#BATCHES)

BBB/Kingsford

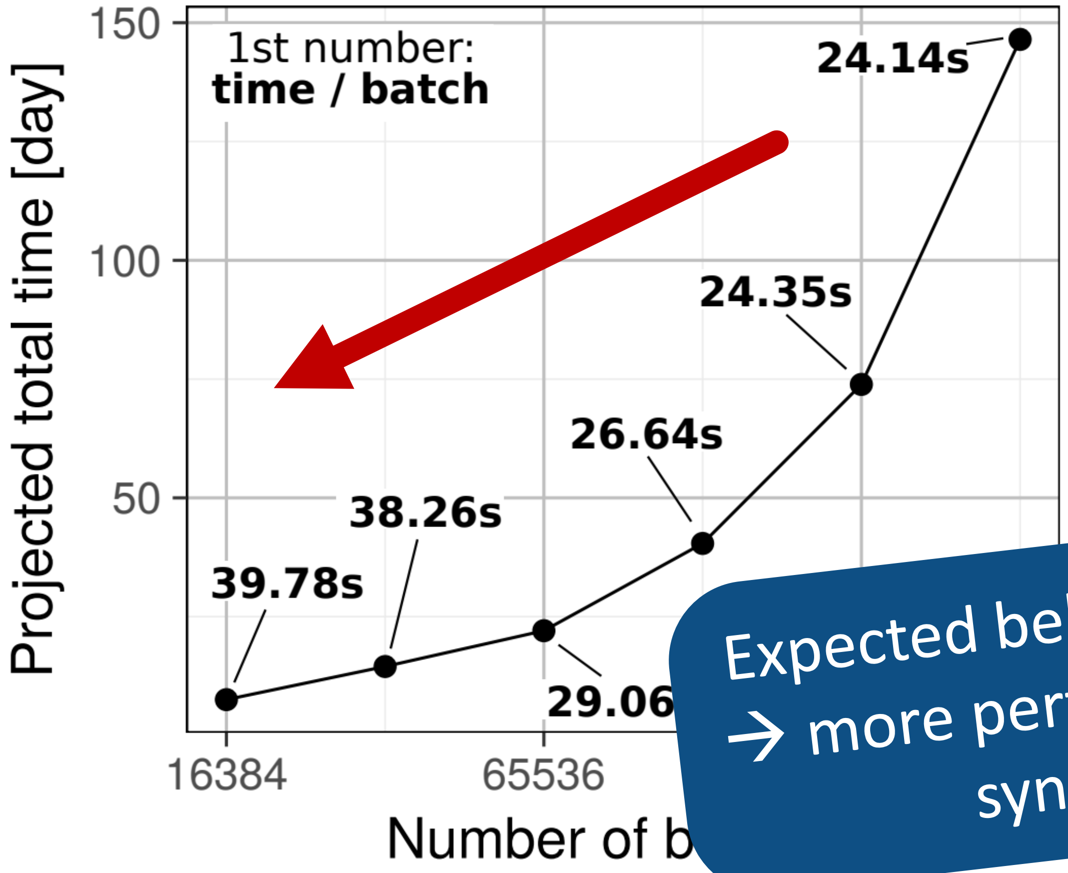


BIGSI

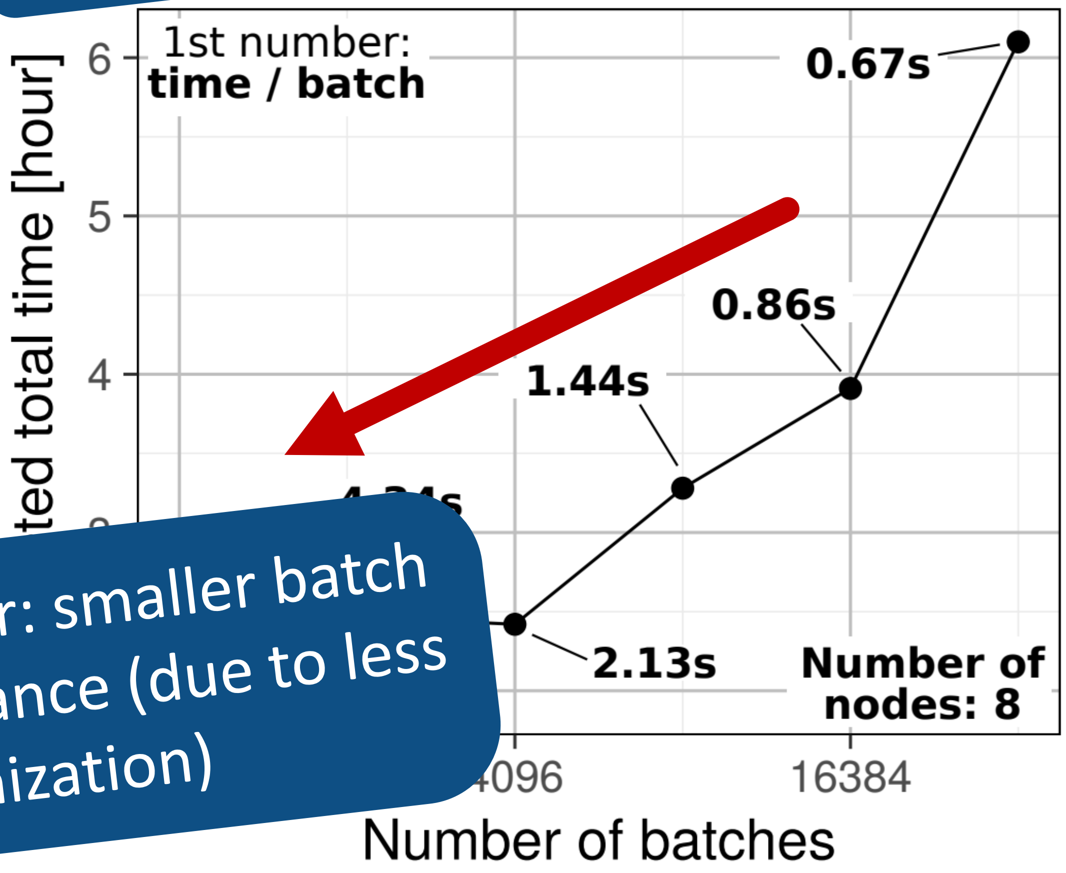


PERFORMANCE ANALYSIS: REAL DATA, SENSITIVITY ANALYSIS (#BATCHES)

BBB/Kingsford



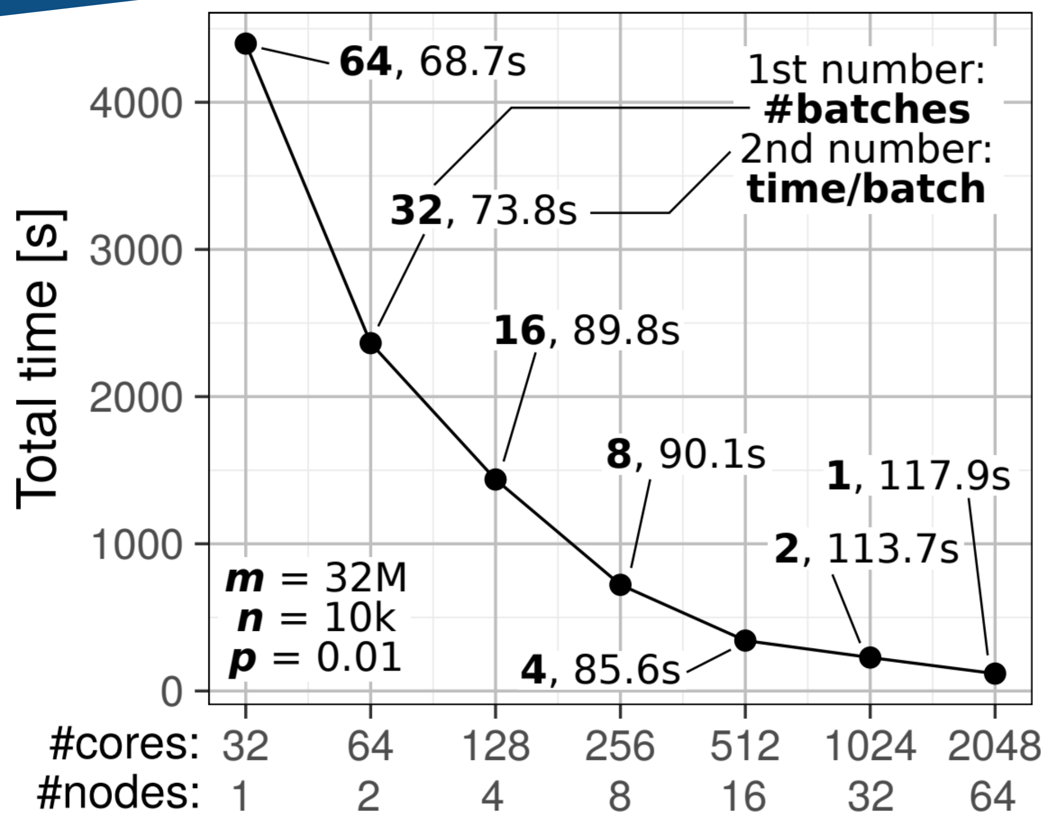
BIGSI



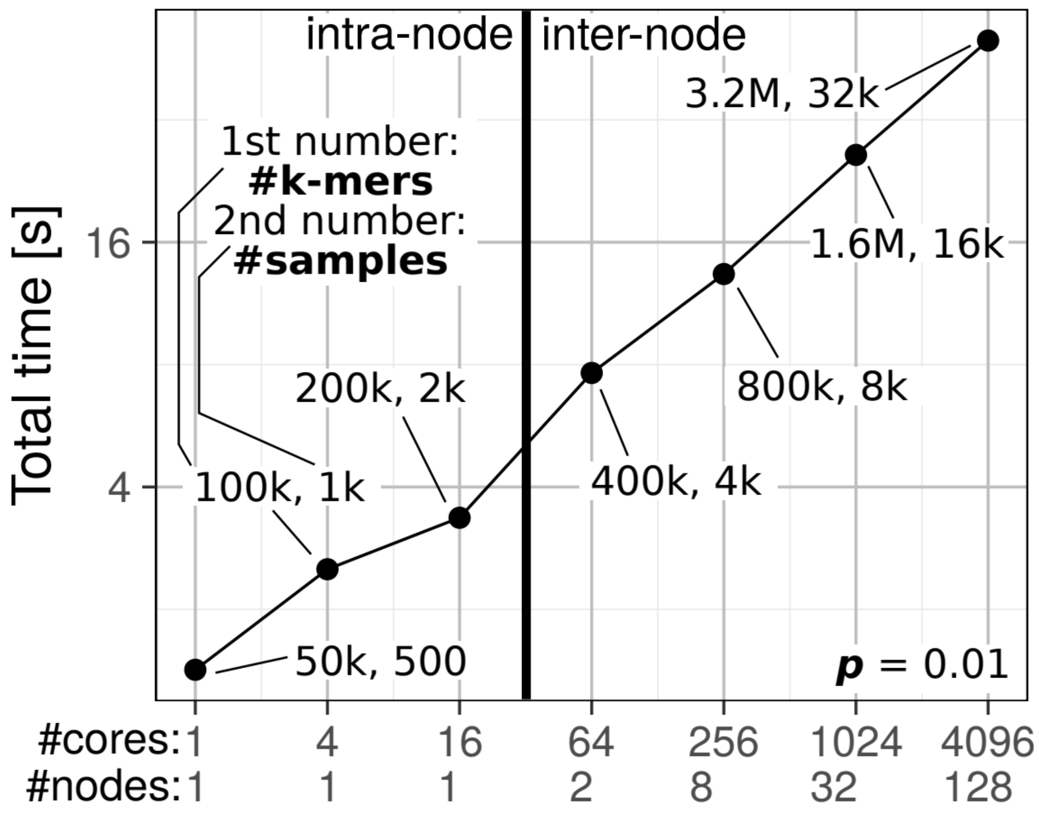
Expected behavior: smaller batch → more performance (due to less synchronization)

Performance analysis: synthetic data

Strong scaling

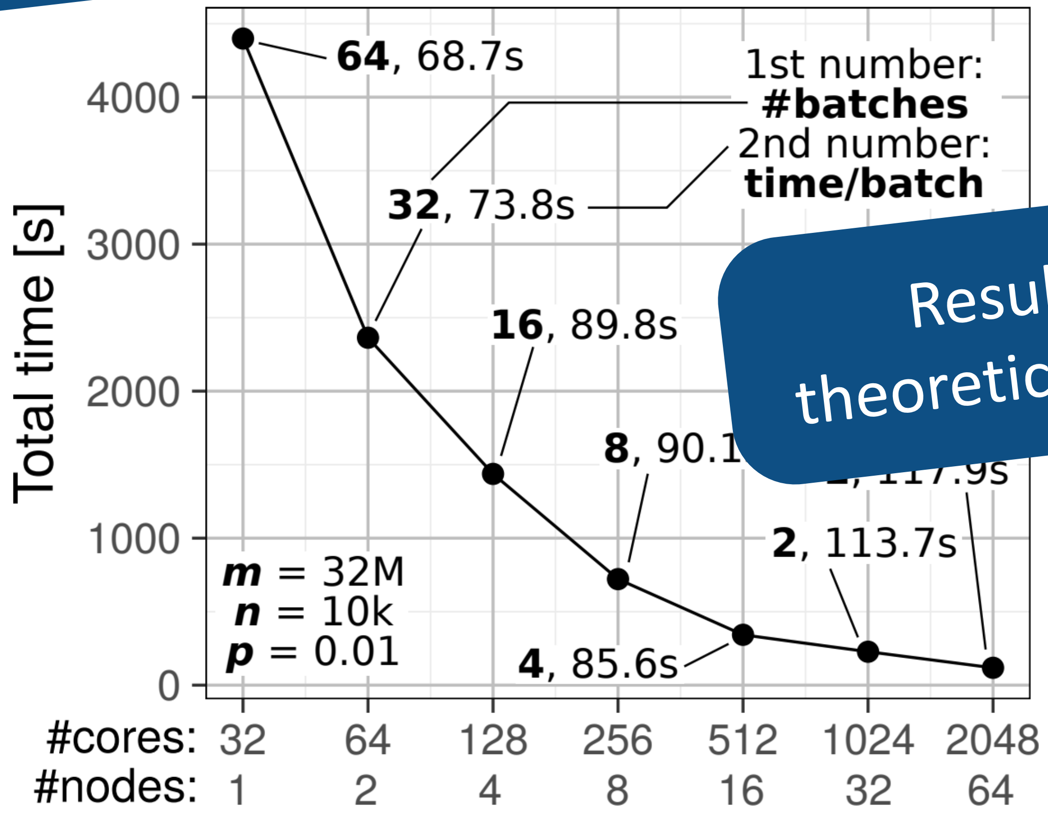


Weak scaling

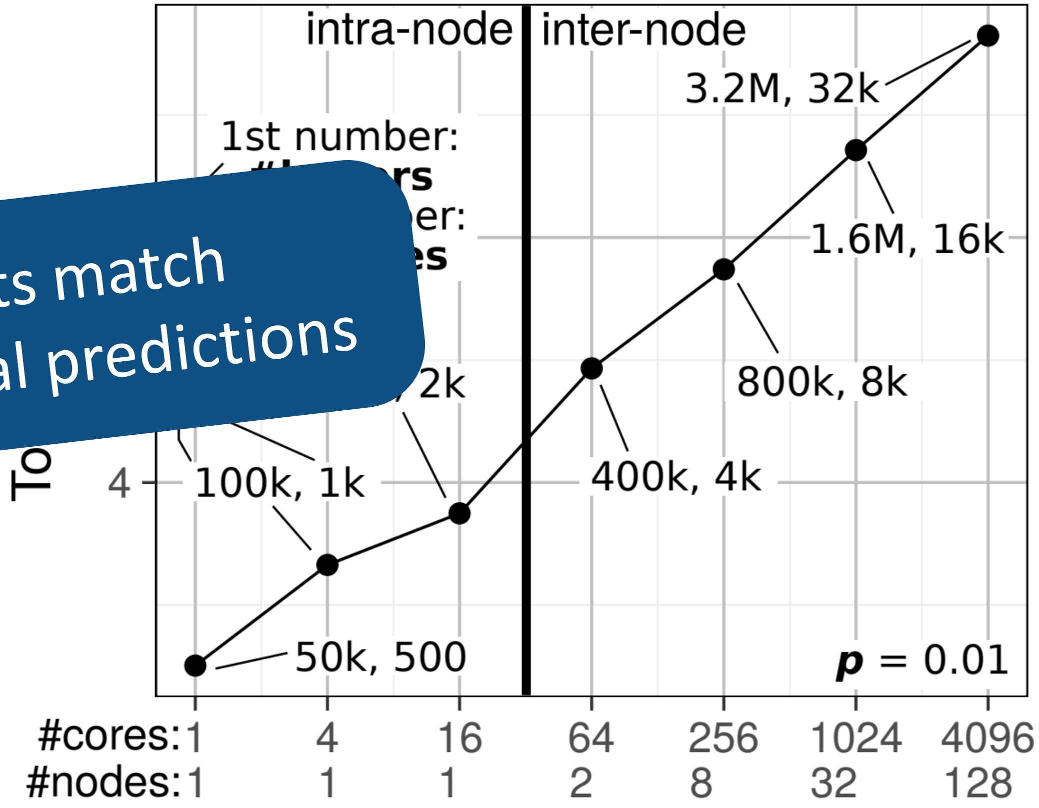


Performance analysis: synthetic data

Strong scaling

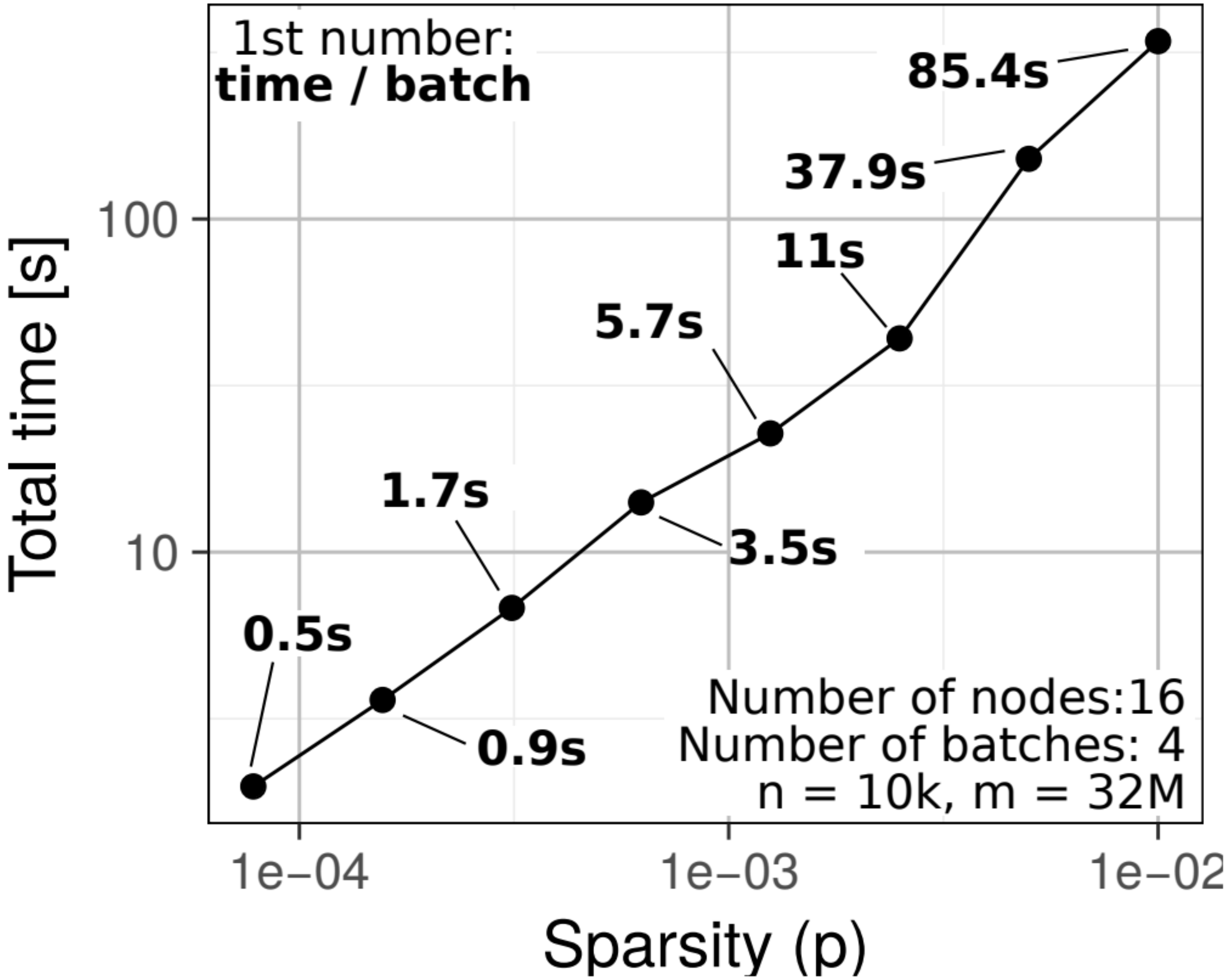


Weak scaling

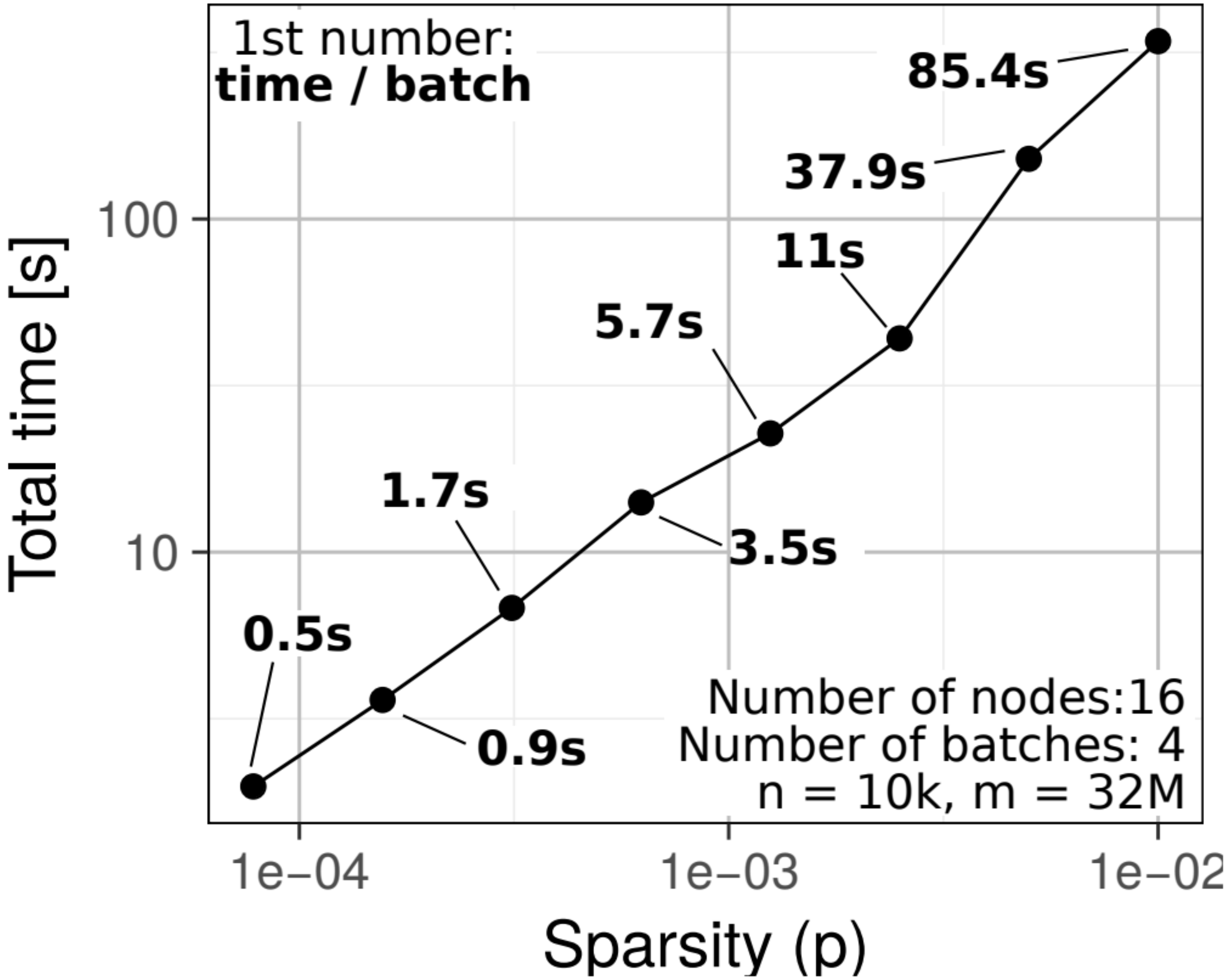


Results match theoretical predictions

PERFORMANCE ANALYSIS: SYNTHETIC DATA, SPARSITY ANALYSIS

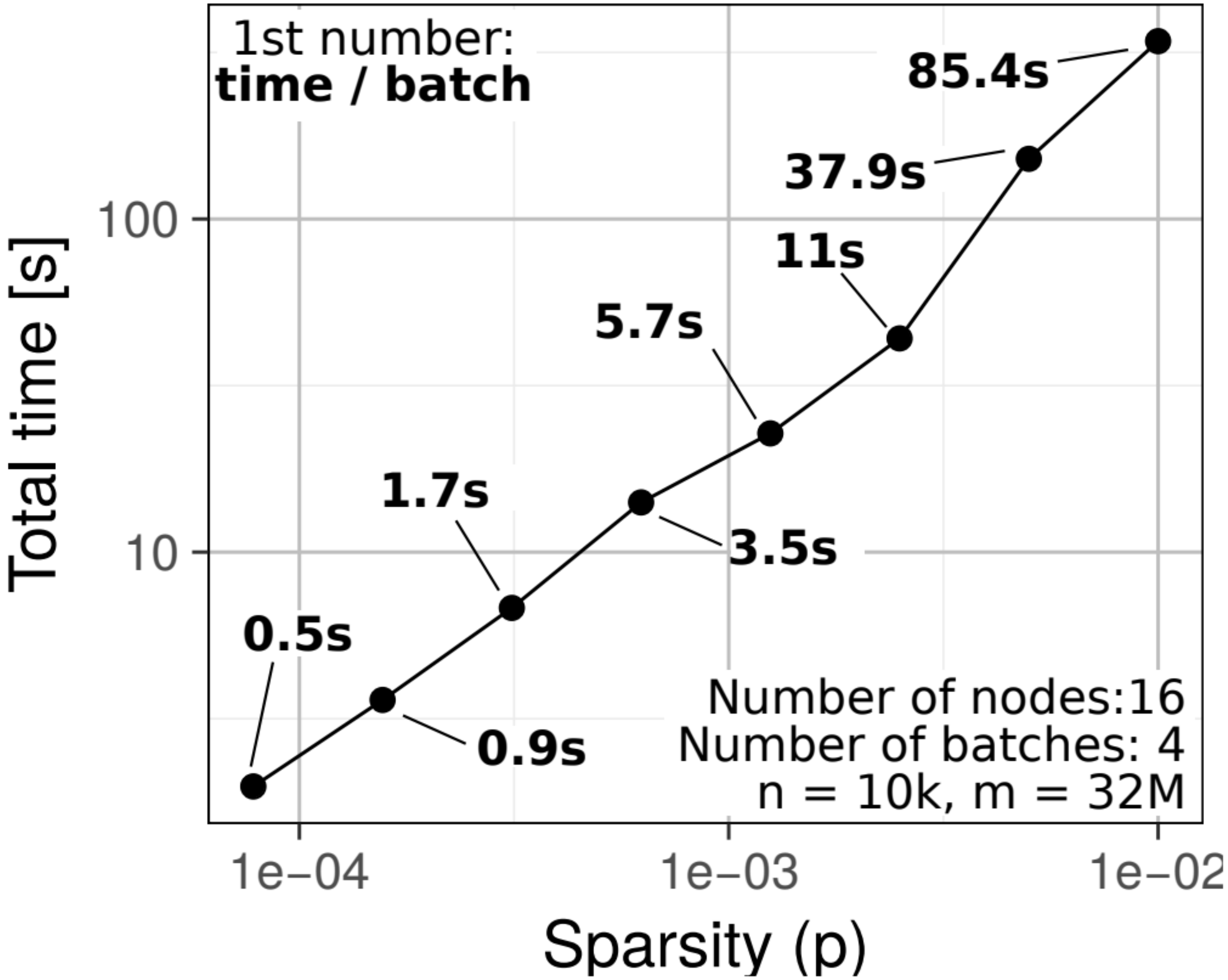


PERFORMANCE ANALYSIS: SYNTHETIC DATA, SPARSITY ANALYSIS



Sparsity (p) corresponds to the probability of the occurrence of a particular *k*-mer

Performance analysis: synthetic data, sparsity analysis



Sparsity (p) corresponds to the probability of the occurrence of a particular *k*-mer

Nearly ideal scaling of the total runtime with the decreasing data sparsity