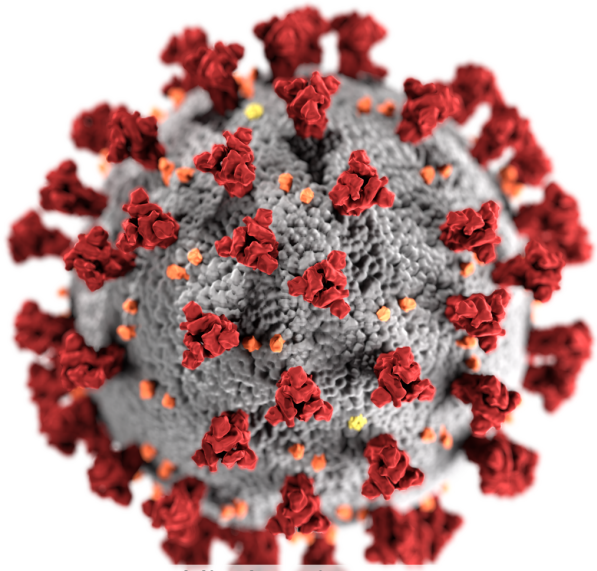
Compressed indexing of (ultra) large viral alignments

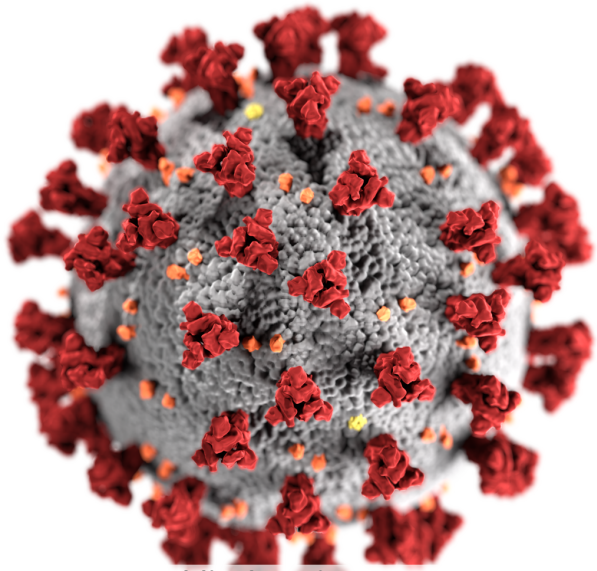
M. Salson, T. Baudeau, A. Boddaert, A. B. Gueye, L. Bulteau,
Y. Hernandez--Courbevoie, C. Marchet, N. Pan, S. Will, Y. Ponty

Characterising SARS-CoV-2 RNA structure



Public domain CDC

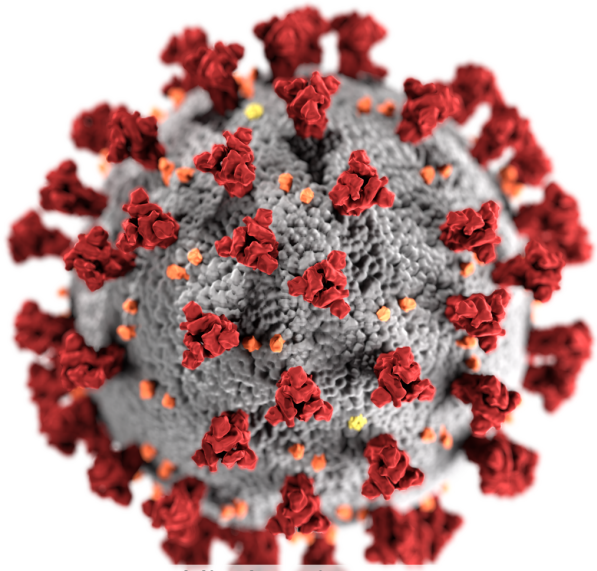
Characterising SARS-CoV-2 RNA structure



Public domain CDC

RNA virus
30 kbp genome

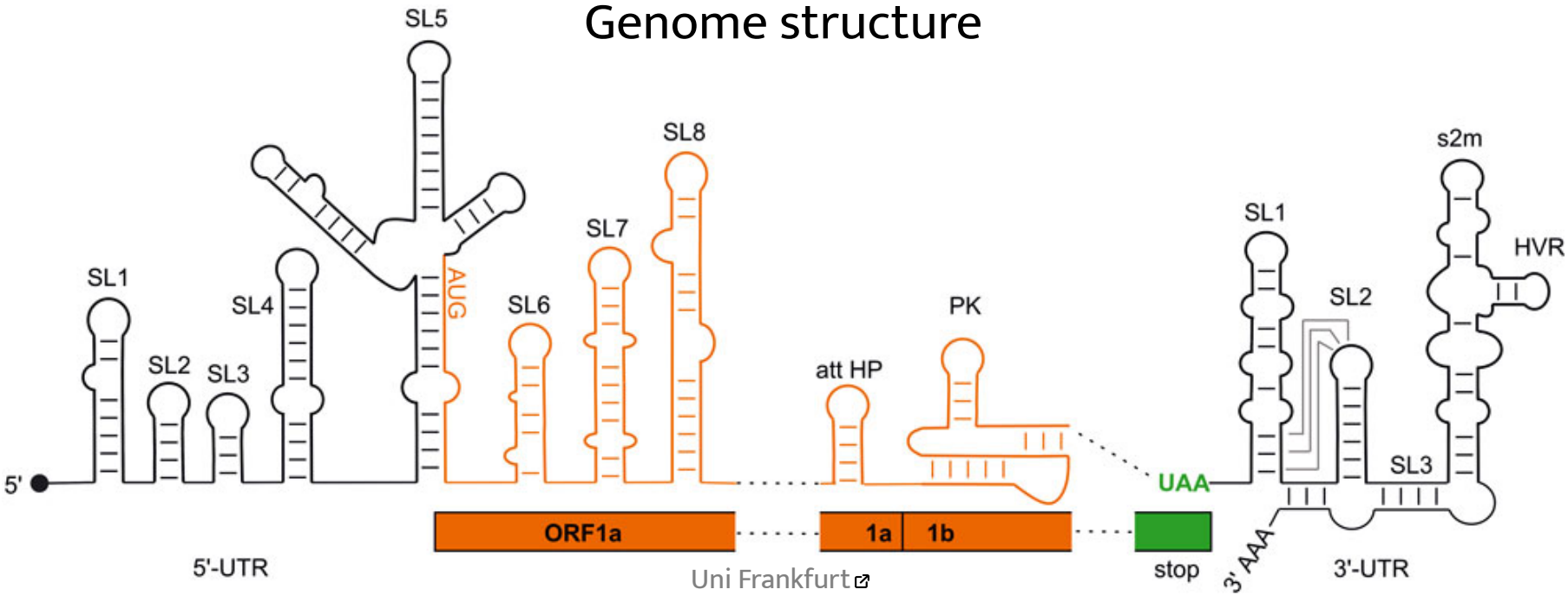
Characterising SARS-CoV-2 RNA structure



Public domain CDC

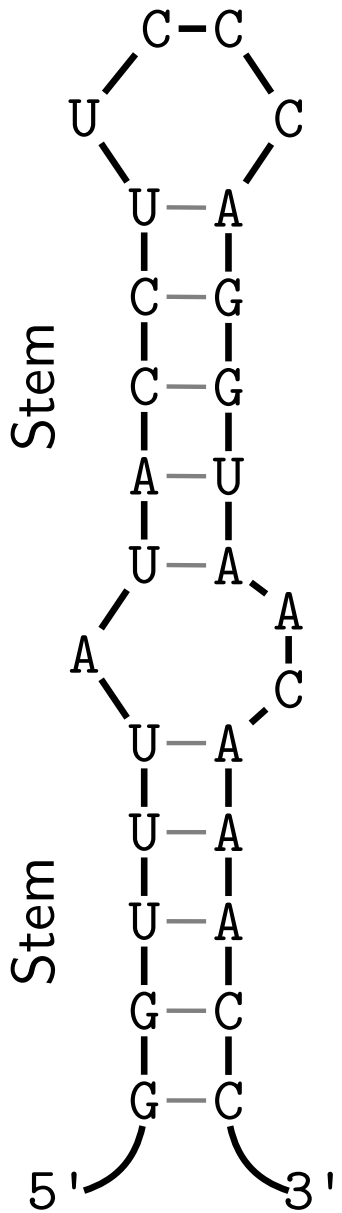
RNA virus
30 kbp genome

Genome structure

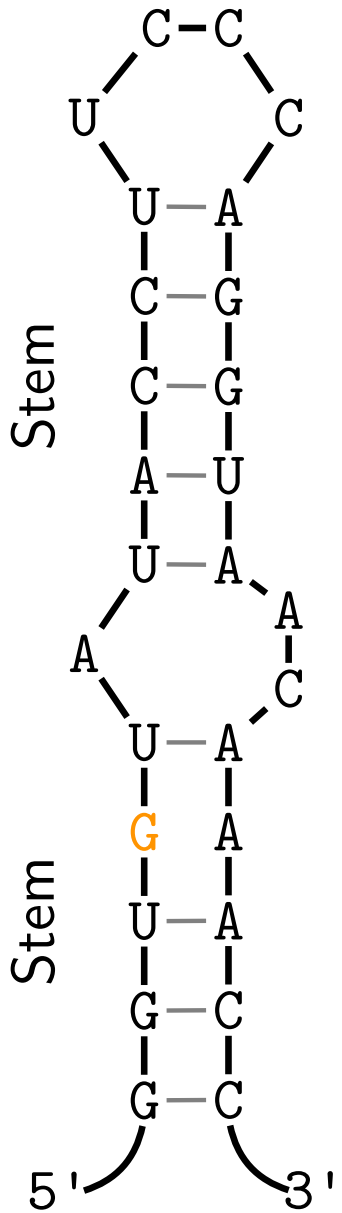


Uni Frankfurt

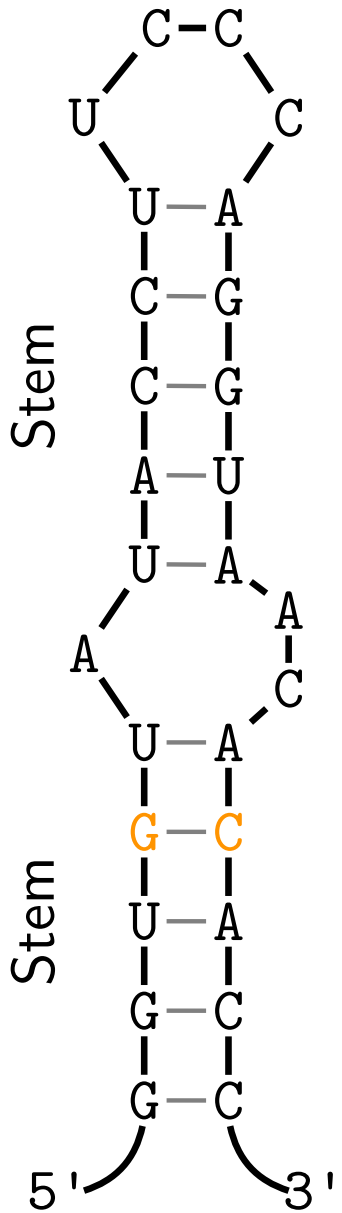
Covariations are a clue to identify stems



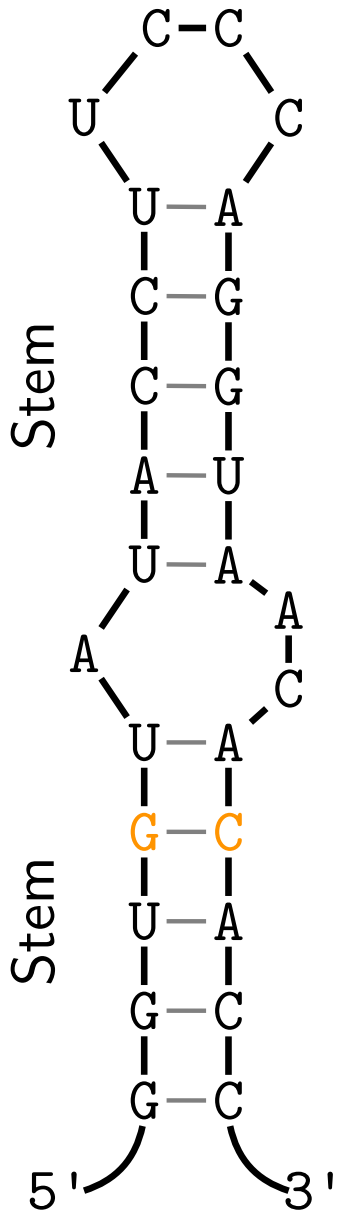
Covariations are a clue to identify stems



Covariations are a clue to identify stems

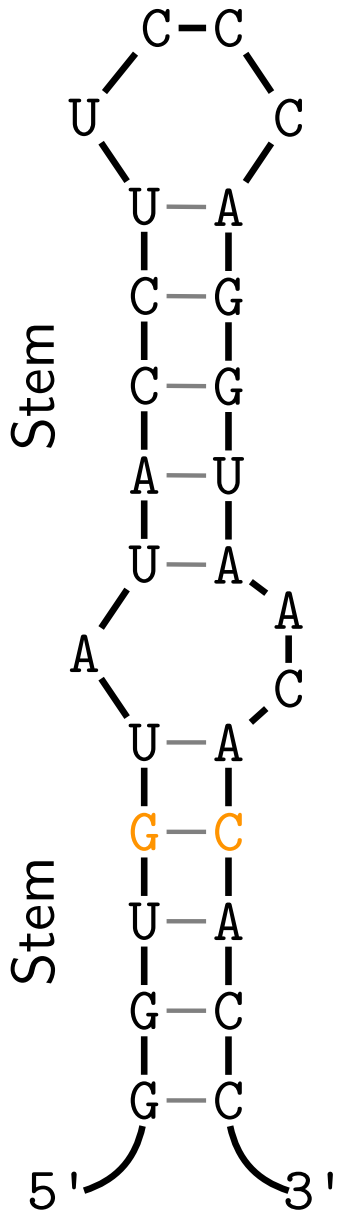


Covariations are a clue to identify stems



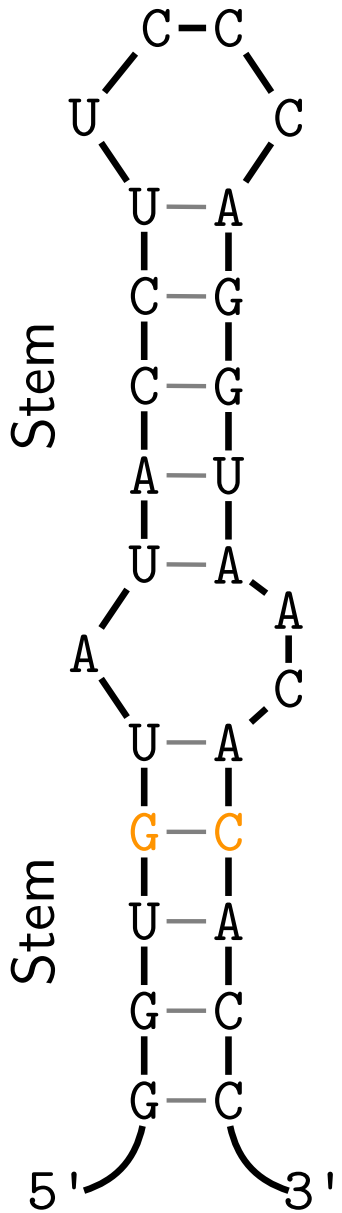
GGUUUAUACCUUCCCAGGUAACAAACC

Covariations are a clue to identify stems



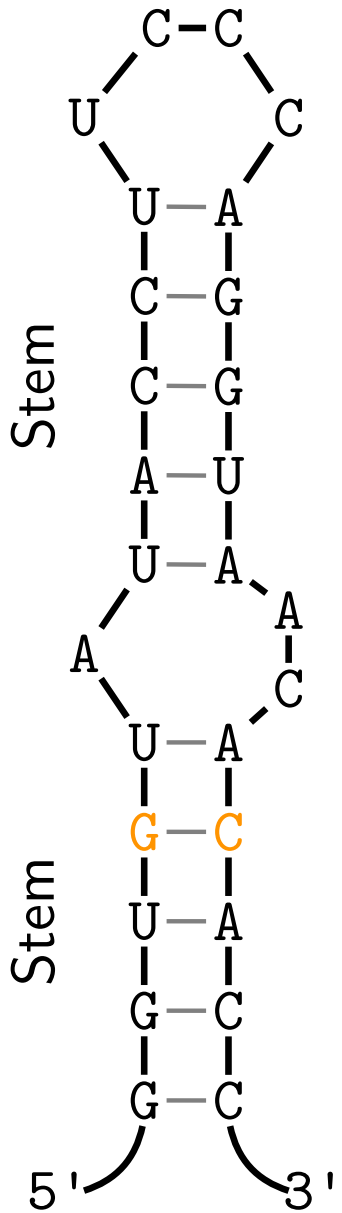
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCCCAGGUAACAAACC

Covariations are a clue to identify stems



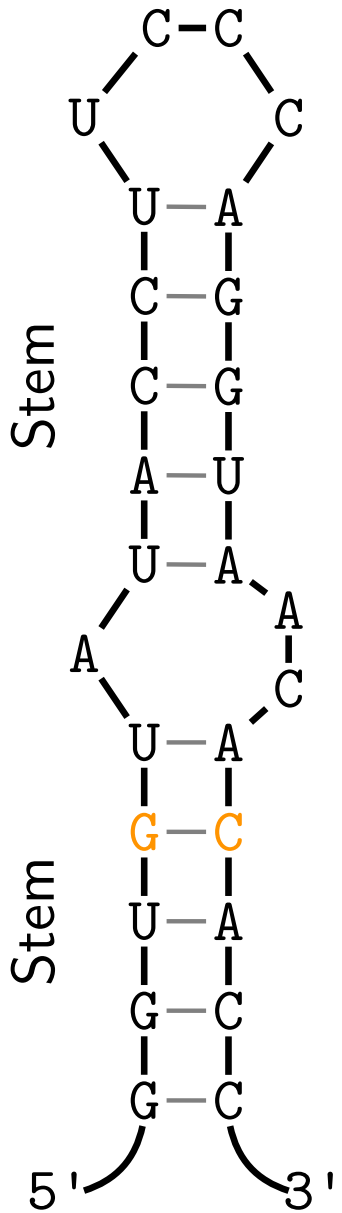
GGUUUAUACCUUCCCAGGUAACAAACC
GGUGUAUACCUUCCCAGGUAACACACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUGUAUACCUUCCCAGGUAACACACC
GGUGUAUACCUUCCCAGGUAACACACC

Covariations are a clue to identify stems



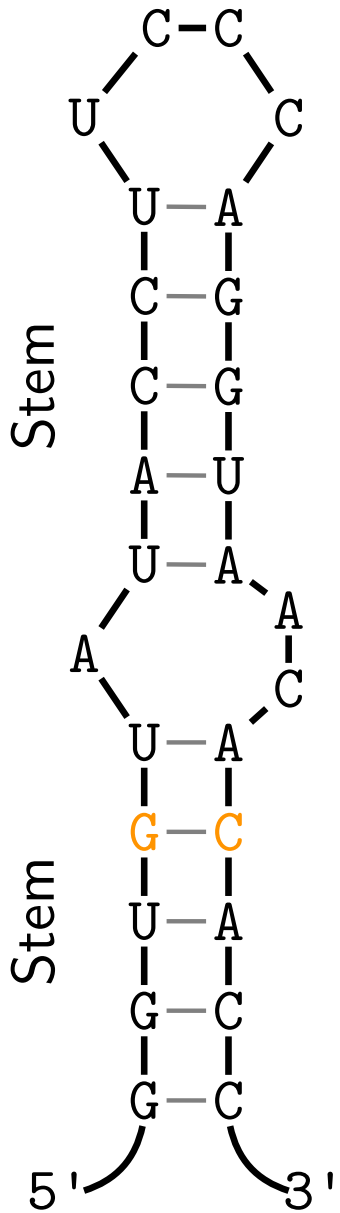
GGUUUAUACCUUCGCAGGUAACAAACC
GGUGUAUACCUUCCCAGGUAACACACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCGCAGGUAACAAACC
GGUGUAUACCUUCGCAGGUAACACACC
GGUGUAUACCUUCCCAGGUAACACACC

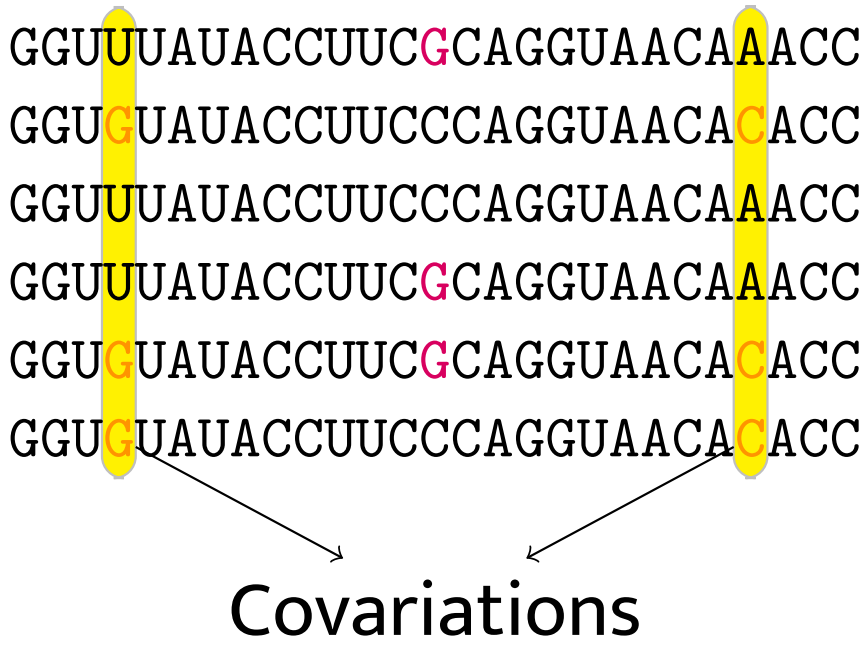
Covariations are a clue to identify stems




```
GGUUUAUACCUUCGCAGGUAACAAACC
GGUGUAUACCUUCCCAGGUAACACACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCGCAGGUAACAAACC
GGUGUAUACCUUCGCAGGUAACACACC
GGUGUAUACCUUCCCAGGUAACACACC
```

Covariations are a clue to identify stems





Identifying covariations through conservation scores

U	A
G	C
U	A
U	A
G	C
G	C

6 compatible pairs

U	A
G	A
G	A
U	C
G	C
U	C

2 compatible pairs

The conservation score of a pair of columns depends on the number of compatible pairs

See RNAalifold (Bernhart et al, 2008)

With n columns of size s, this could be computed in $\Theta(sn^2)$

Lots of data to identify covariations in SARS-CoV-2

3,079,138

genomes

On NCBI Virus, as of 26 Nov 2024

Awesome! But that's quite a MSA...

Assuming we can build it...

Computing conservations scores on millions of genomes

Let's assume we have a MSA of 1M SARS-CoV-2

Computing conservations scores on millions of genomes

Let's assume we have a MSA of 1M SARS-CoV-2

- That's 30k columns of 1Mbp

Computing conservations scores on millions of genomes

Let's assume we have a MSA of 1M SARS-CoV-2

- That's 30k columns of 1Mbp
- We need to consider every pair of columns
  $\rightarrow$ 450 M pairs of columns

Computing conservations scores on millions of genomes

Let's assume we have a MSA of 1M SARS-CoV-2

- That's 30k columns of 1Mbp
- We need to consider every pair of columns
  $\rightarrow$ 450 M pairs of columns
- 10^{14} pairs of nucleotides must be considered

Computing conservations scores on millions of genomes

Let's assume we have a MSA of 1M SARS-CoV-2

- That's 30k columns of 1Mbp
- We need to consider every pair of columns
  $\rightarrow$ 450 M pairs of columns
- 10^{14} pairs of nucleotides must be considered

More generally, getting any kind of information from a 1Mbp column is inefficient

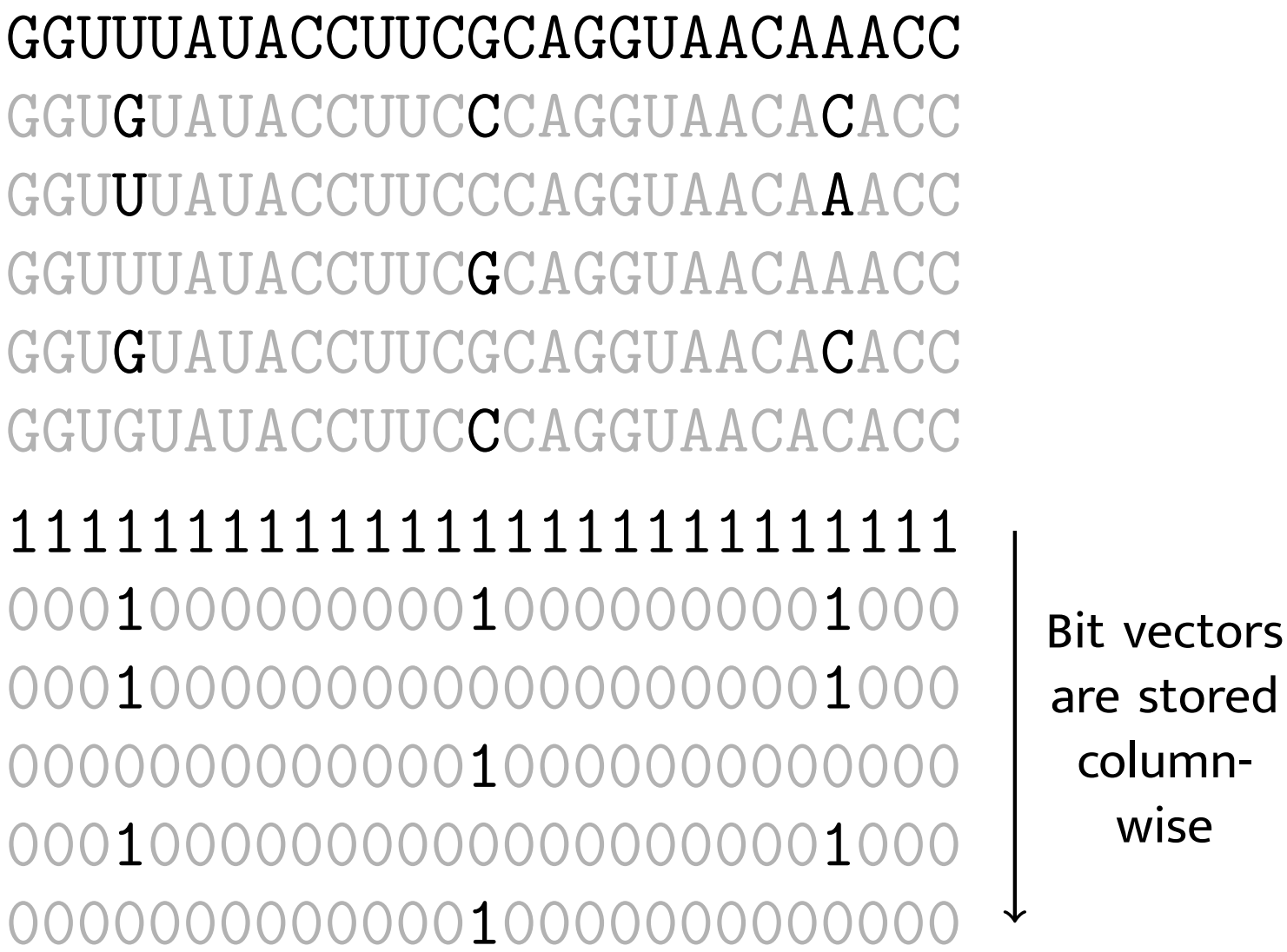
A self-index for multiple sequence alignement (MSA)

GGUUUAUACCUUCGCAGGUAACAAACC
GGUGUAUACCUUCCCAGGUAACACACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCGCAGGUAACAAACC
GGUGUAUACCUUCGCAGGUAACACACC
GGUGUAUACCUUCCCAGGUAACACACC

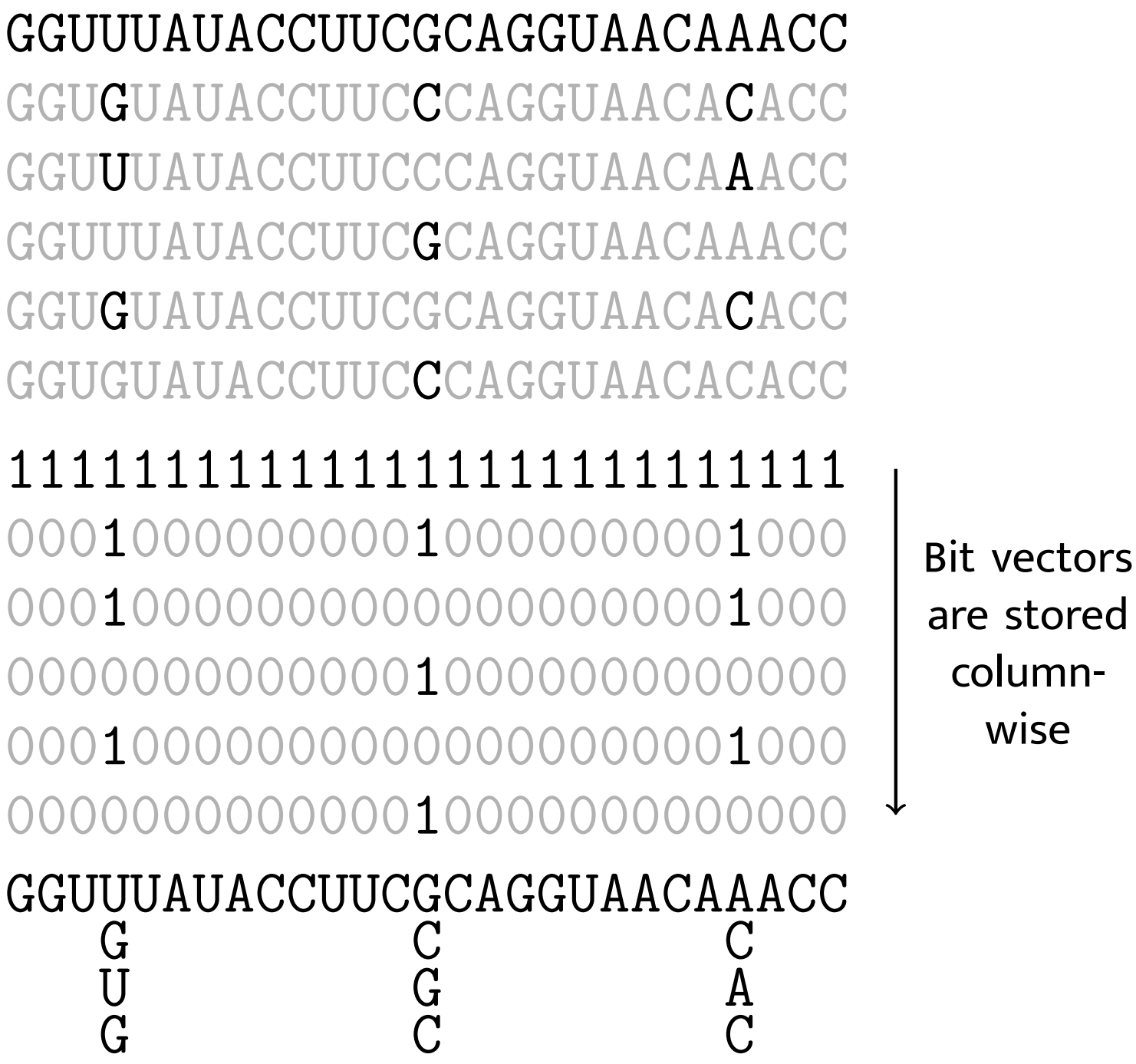
A self-index for multiple sequence alignement (MSA)

GGUUUAUACCUUCGCAGGUAACAAACC
GGUGUAUACCUUCCCAGGUAACACACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCGCAGGUAACAAACC
GGUGUAUACCUUCGCAGGUAACACACC
GGUGUAUACCUUCCCAGGUAACACACC

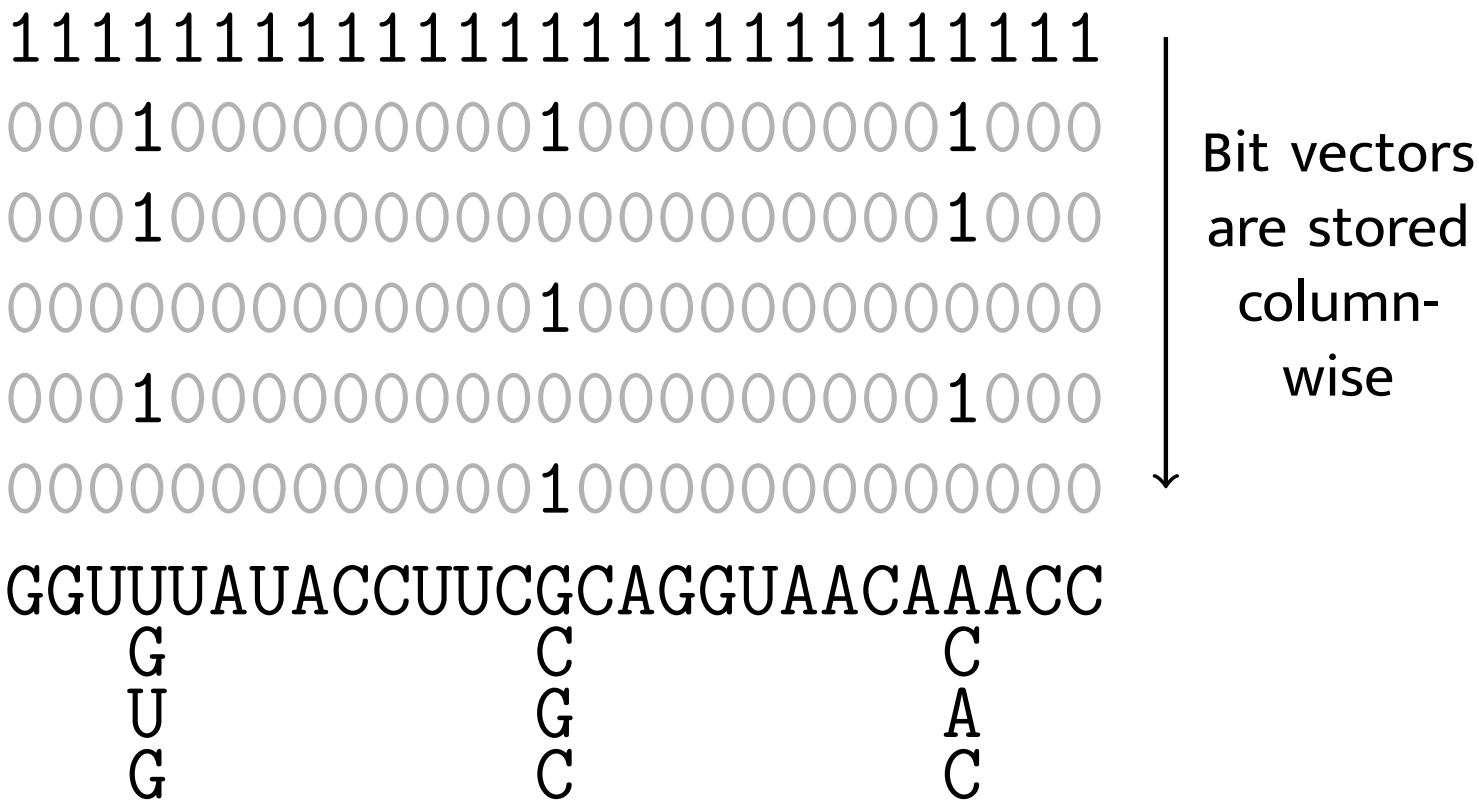
A self-index for multiple sequence alignement (MSA)



A self-index for multiple sequence alignement (MSA)



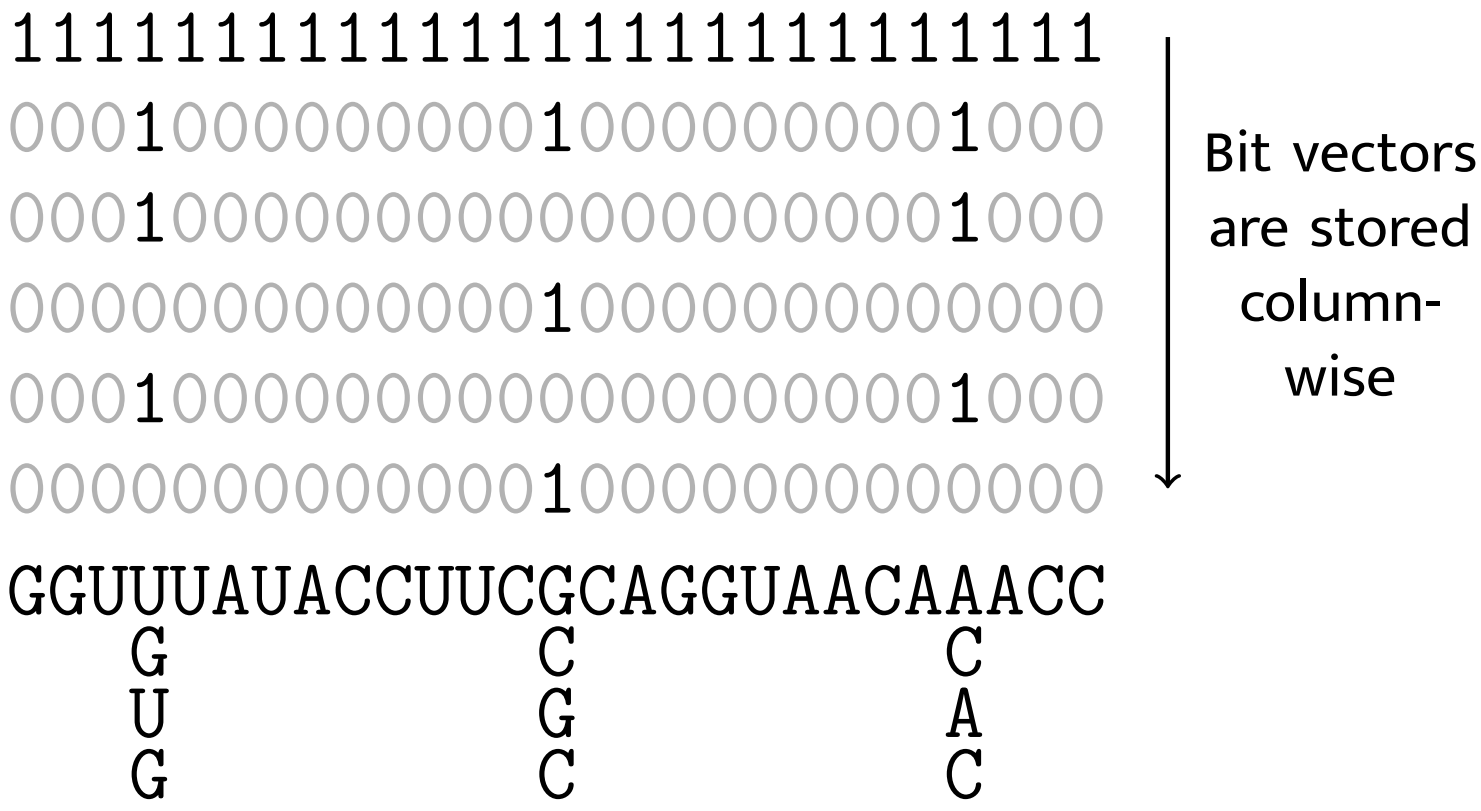
A self-index for multiple sequence alignement (MSA)



A self-index for multiple sequence alignement (MSA)

r_i : number of runs of an identical letter in column i of the MSA

r_i: number of 1s in the bit vector of column i

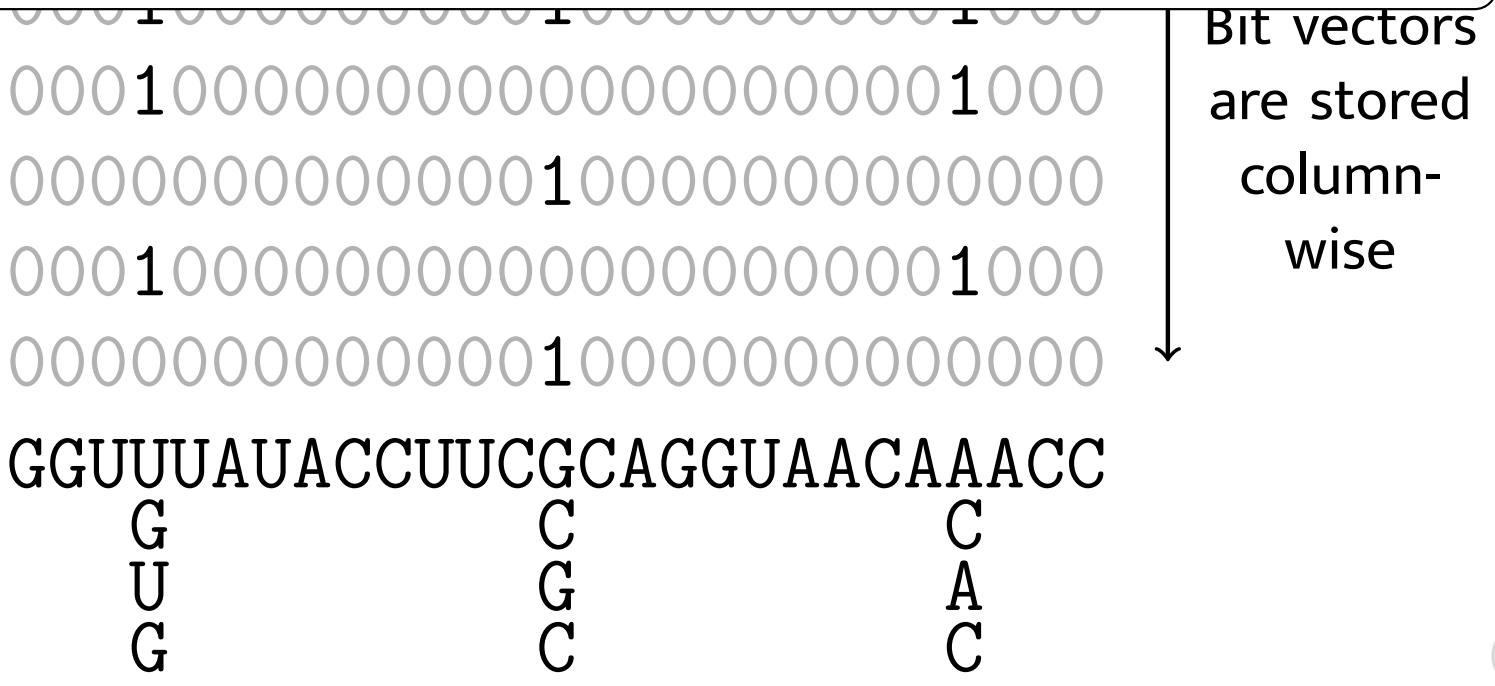


A self-index for multiple sequence alignement (MSA)

r_i : number of runs of an identical letter in column i of the MSA

r_i: number of 1s in the bit vector of column i

The fewer runs, the more we compress



`rank` and `select` for querying our self-index for MSA

`rank`/`select` are fundamental operations on bit vectors

`rank`(B, i): nb of 1s in $B[O \ldots i]$
`select`(B, i): position of the i-th 1 in B

`rank` and `select` for querying our self-index for MSA

`rank`/`select` are fundamental operations on bit vectors

`rank`(B, i): nb of 1s in $B[O \ldots i]$

`select`(B, i): position of the i-th 1 in B

1

0

0

0

1

0

1

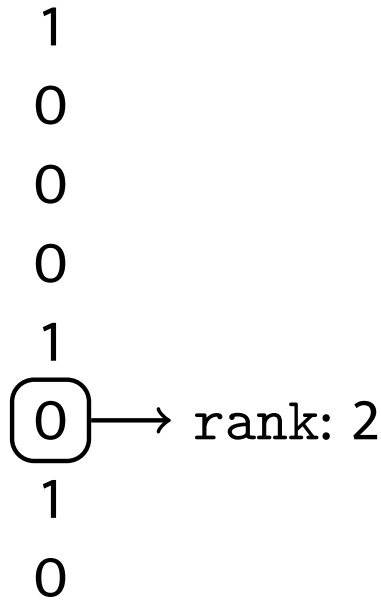
0

`rank` and `select` for querying our self-index for MSA

`rank`/`select` are fundamental operations on bit vectors

`rank`(B, i): nb of 1s in $B[0 \ldots i]$

`select`(B, i): position of the i-th 1 in B

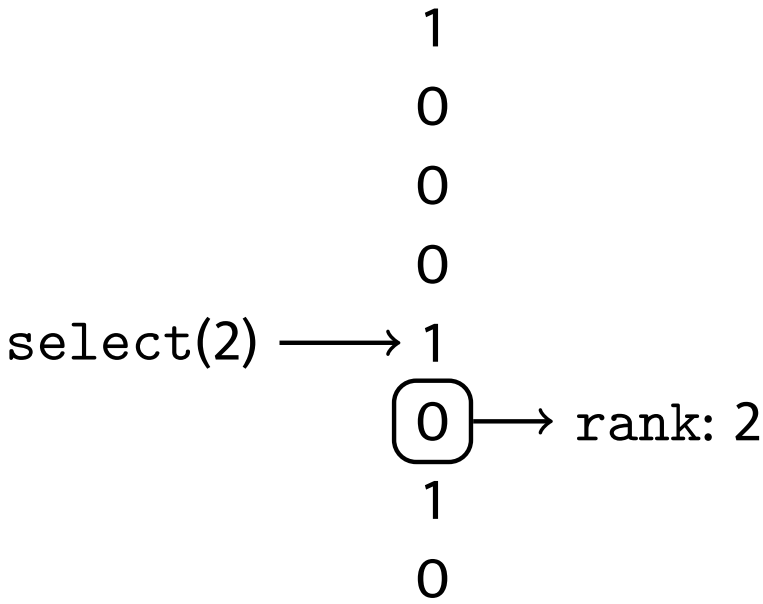


`rank` and `select` for querying our self-index for MSA

`rank`/`select` are fundamental operations on bit vectors

`rank`(B, i): nb of 1s in $B[0 \ldots i]$

`select`(B, i): position of the i-th 1 in B



`rank` and `select` for querying our self-index for MSA

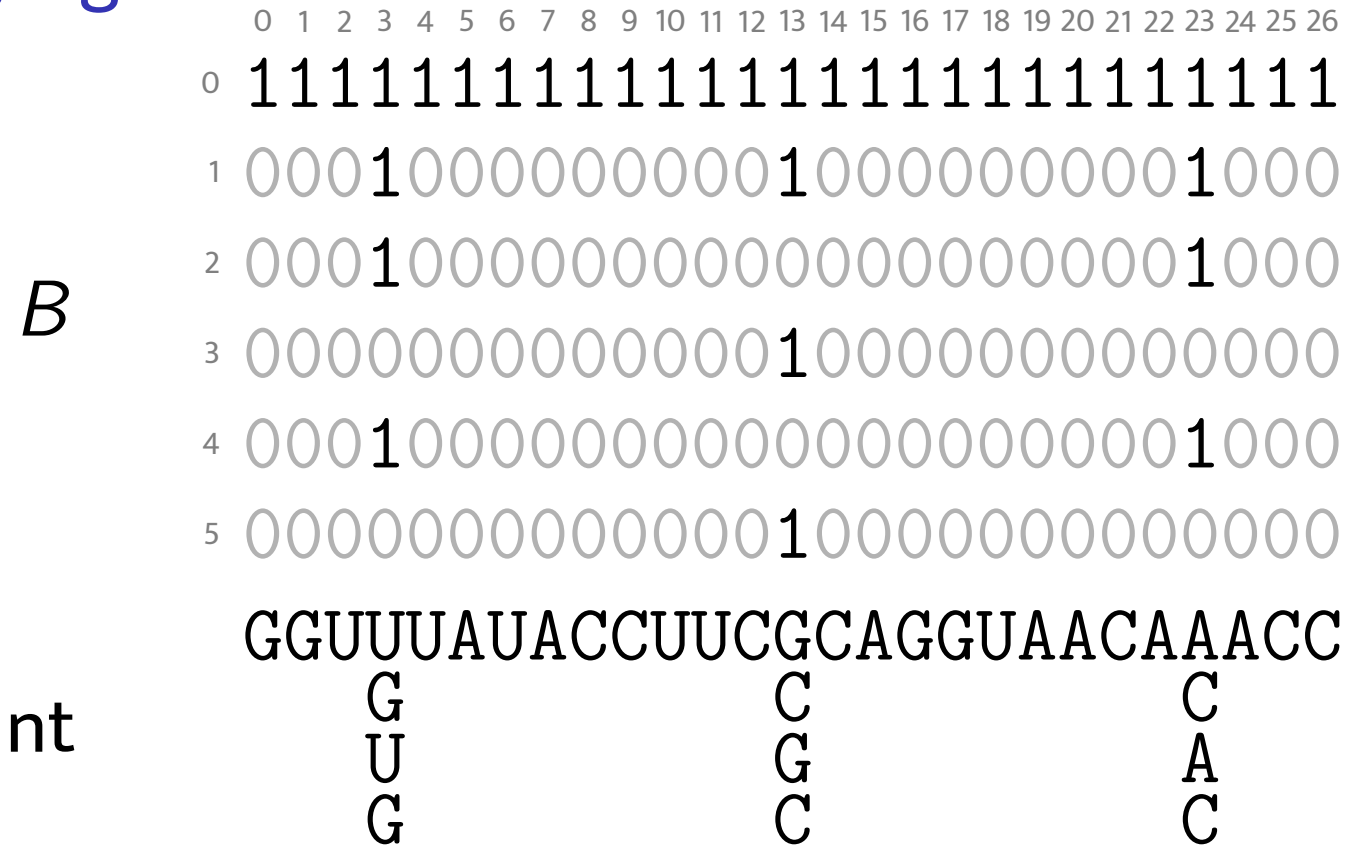
`rank`/`select` are fundamental operations on bit vectors

`rank`(B, i): nb of 1s in $B[O \ldots i]$
`select`(B, i): position of the i-th 1 in B

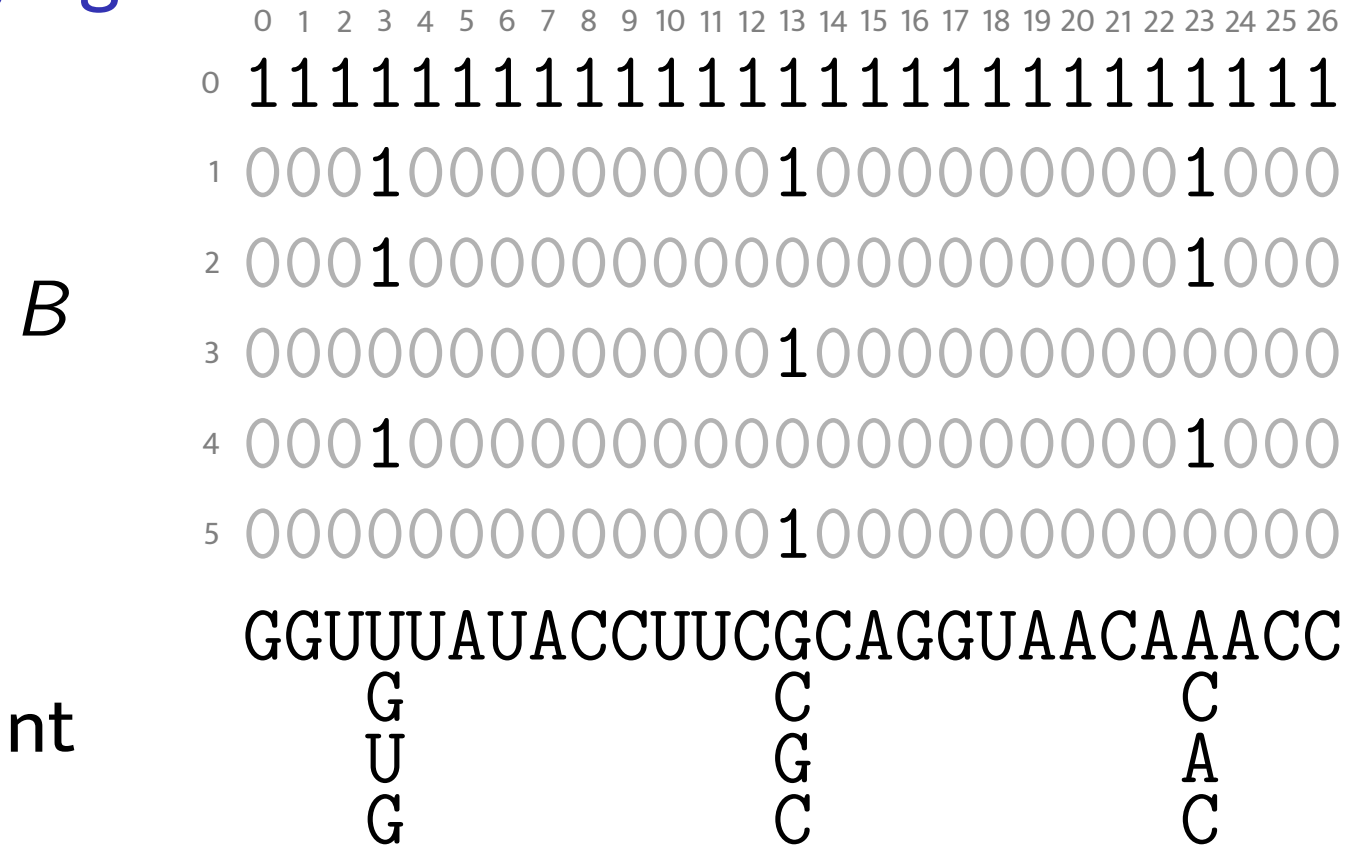
Compressed bit-vectors can support constant-time `rank`/`select` operations

See eg. Gog et al, 2014

Querying CREMSA

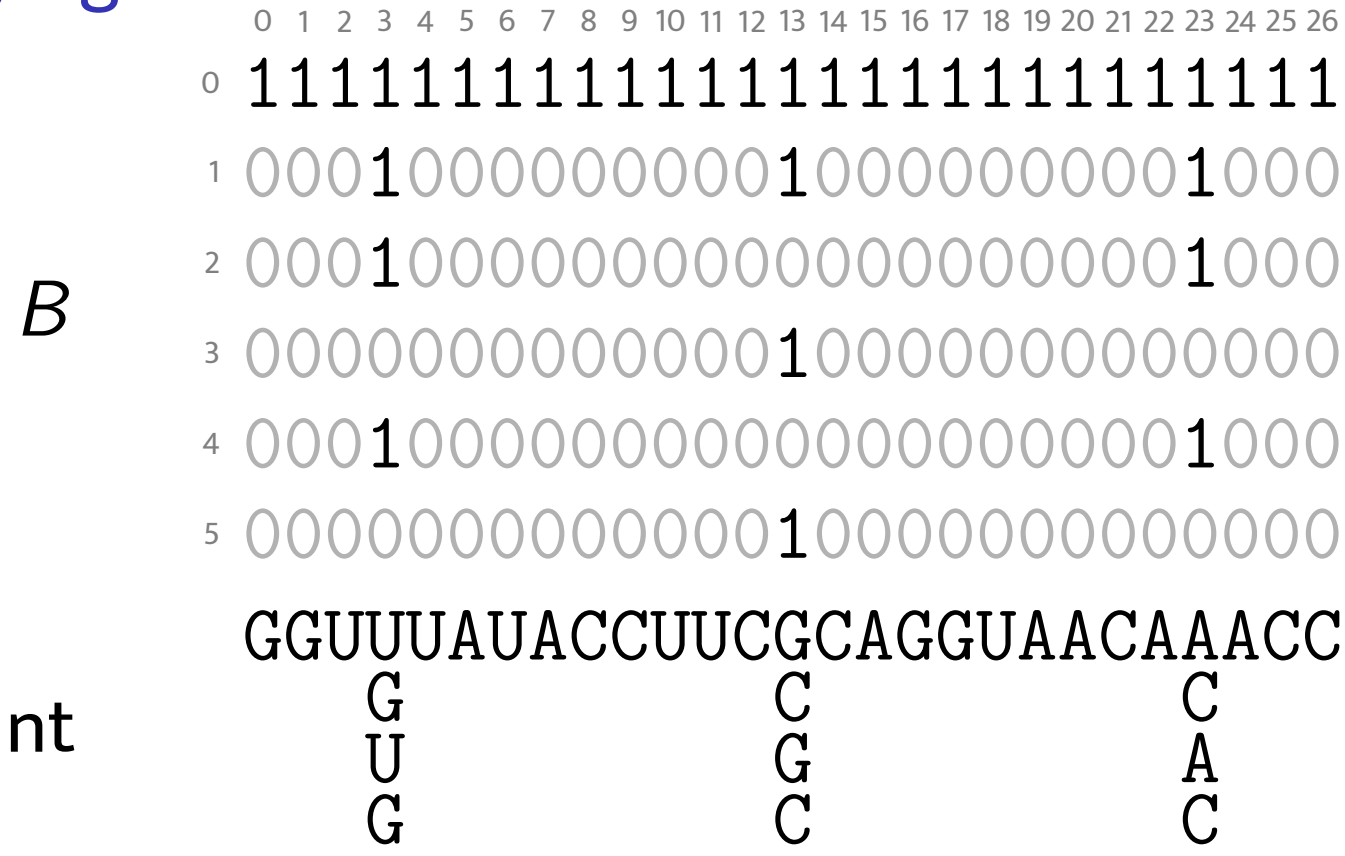


Querying CREMSA



Accessing $s_i[j]$

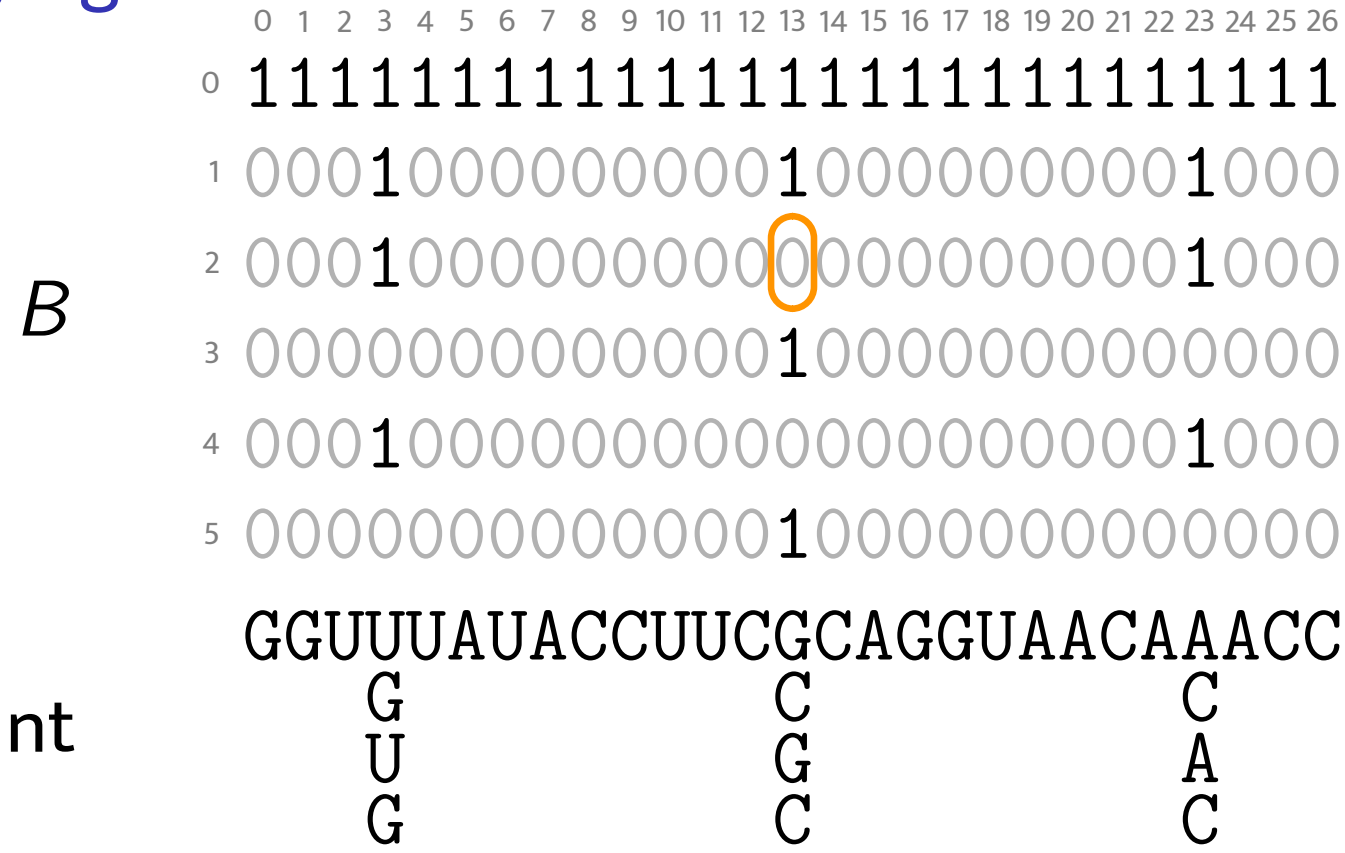
Querying CREMSA



Accessing $s_i[j]$

$s_2[13]$?

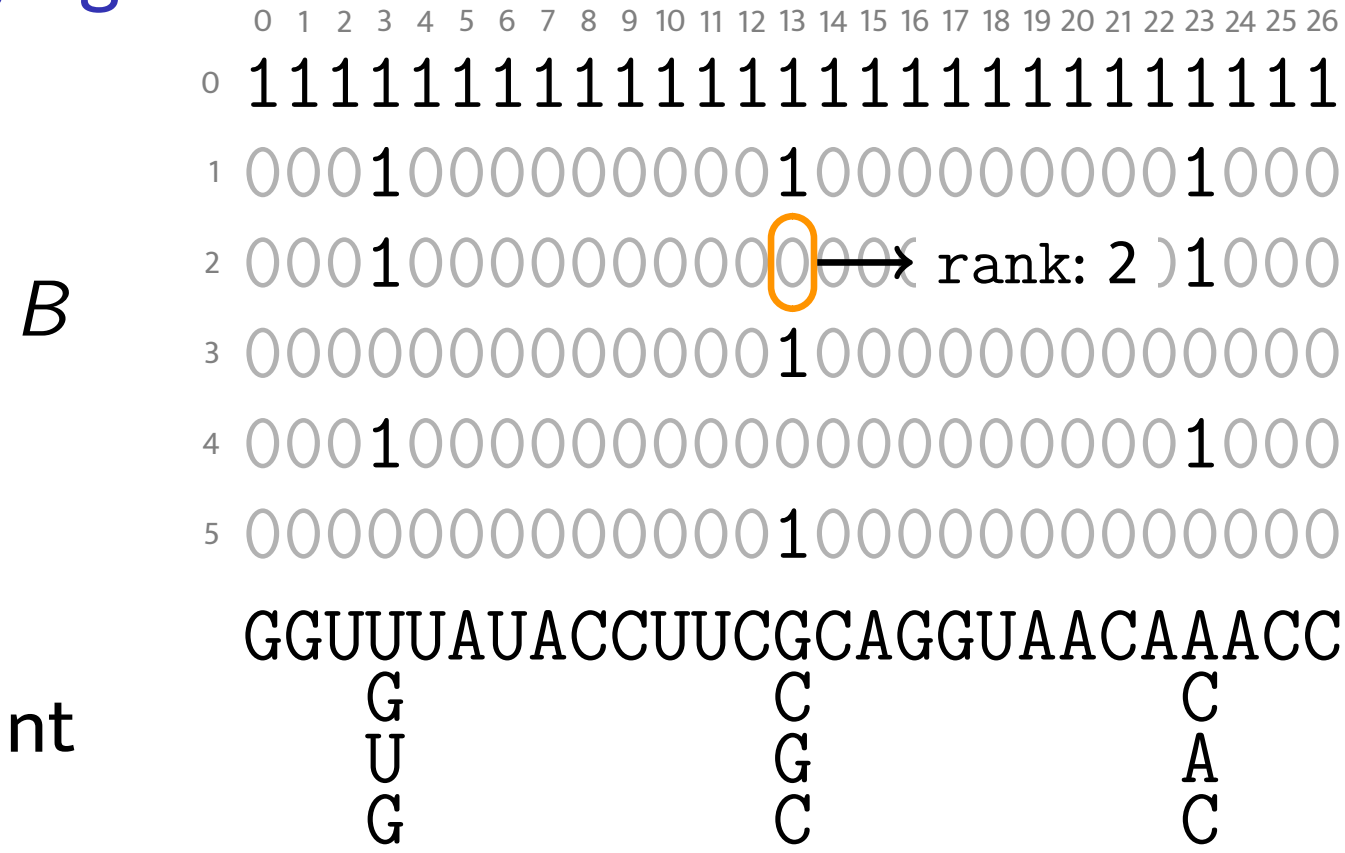
Querying CREMSA



Accessing $s_i[j]$

$s_2[13]$?

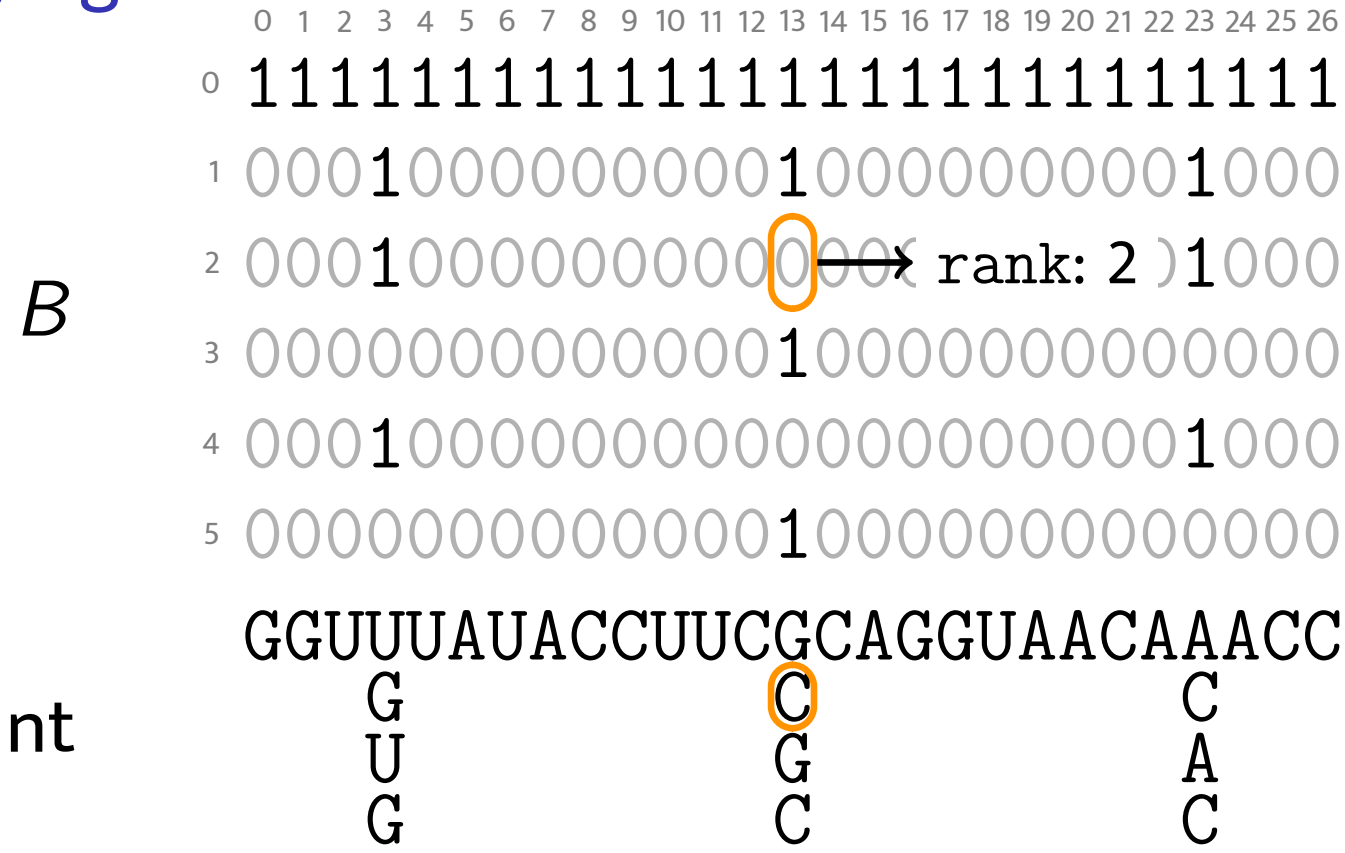
Querying CREMSA



Accessing $s_i[j]$

$s_2[13]$?

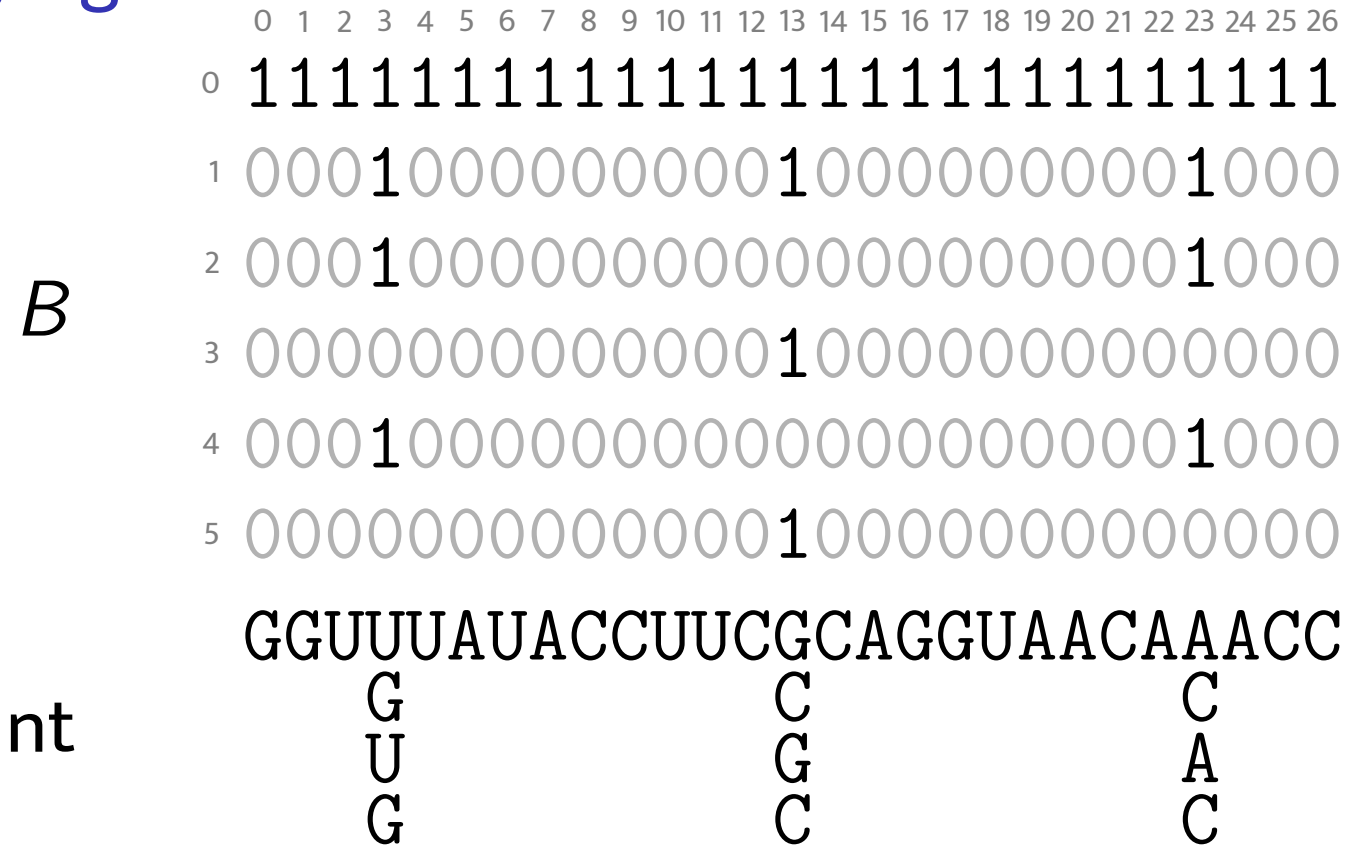
Querying CREMSA



Accessing $s_i[j]$

$s_2[13]$?

Querying CREMSA

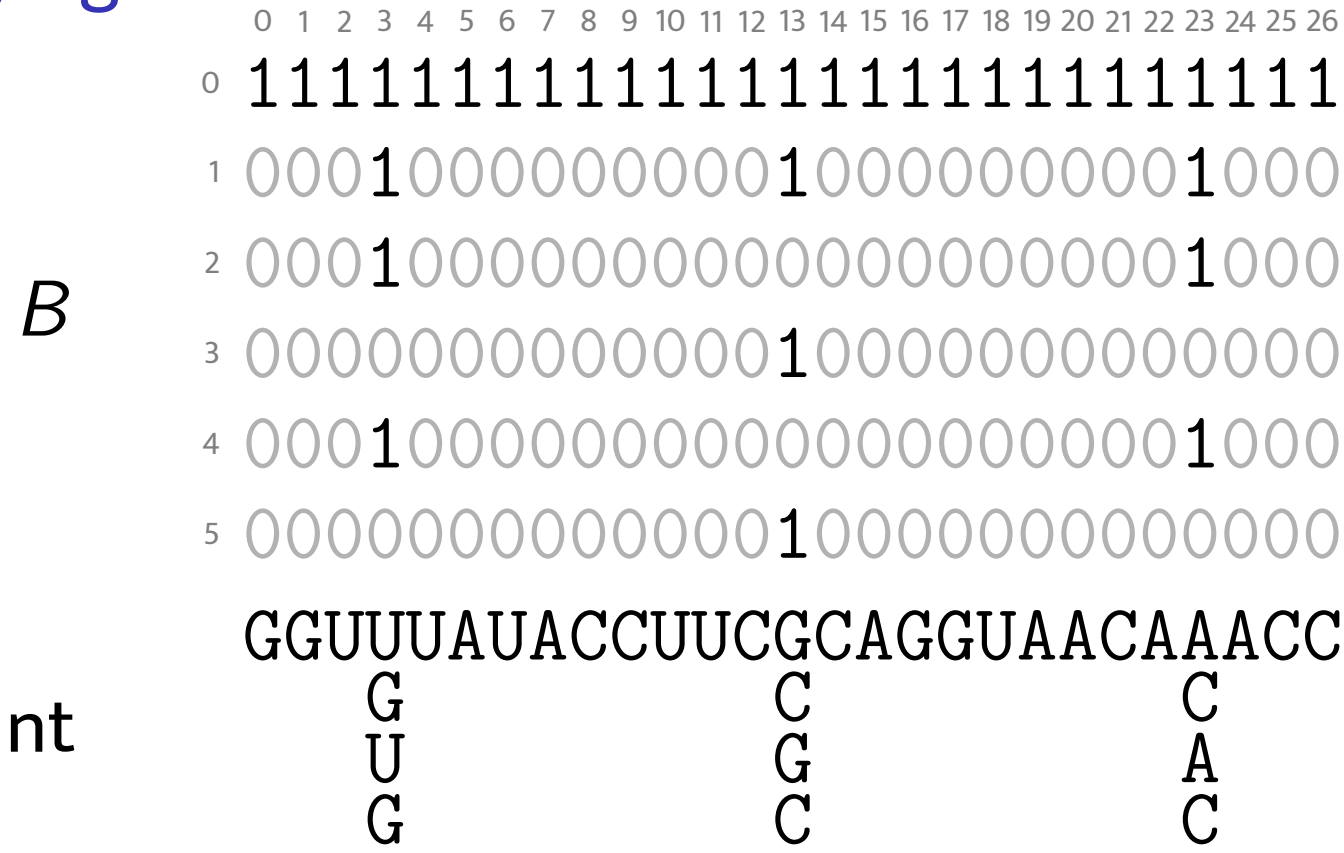


Accessing $s_i[j]$

$$s_i[j] = \mathsf{nt}_j[\mathtt{rank}(B_j, i)]$$

$$\Theta(1)$$

Querying CREMSA

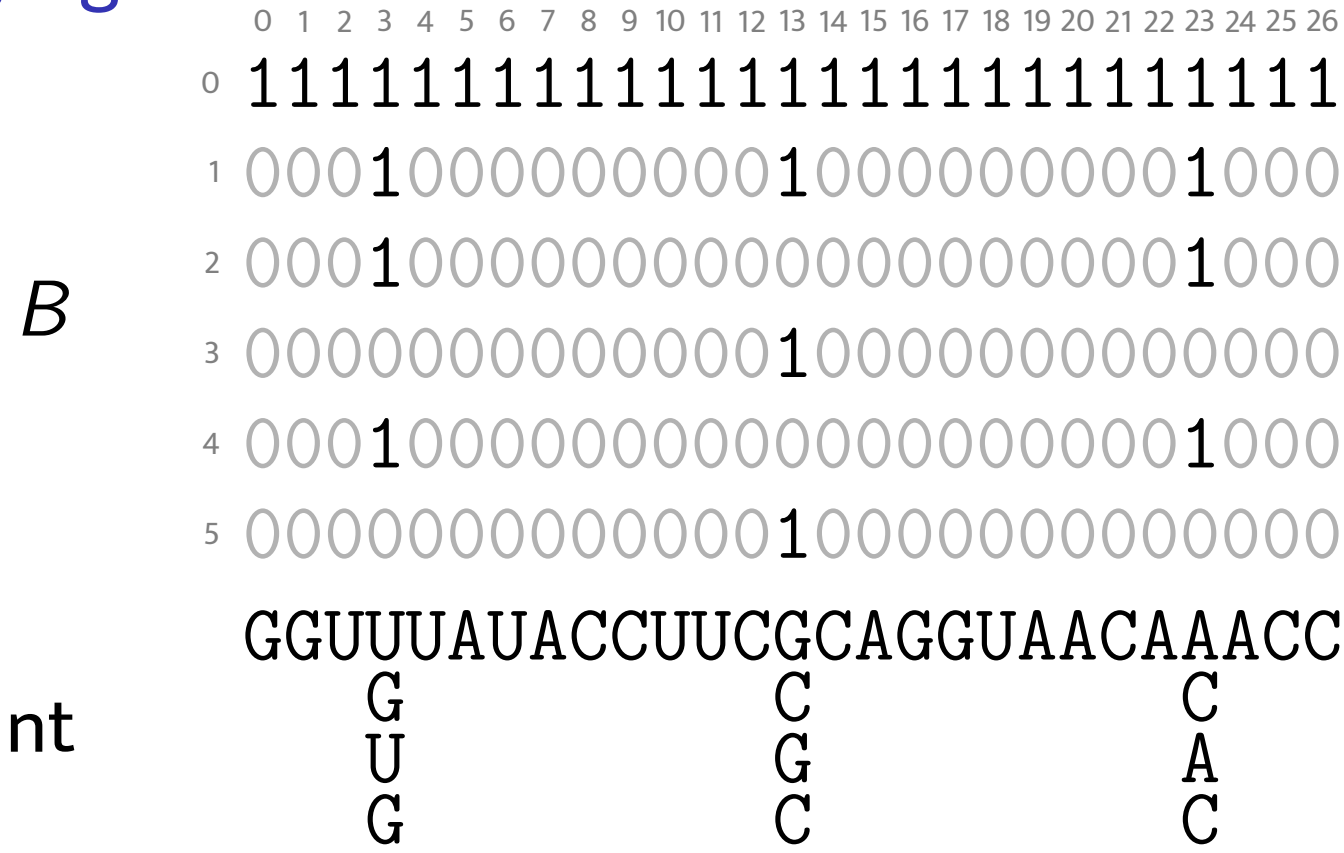


Accessing $s_i[j]$ Counting nt in col. j

$$s_i[j] = \mathrm{nt}_j[\mathtt{rank}(B_j, i)]$$

$$\Theta(1)$$

Querying CREMSA



Accessing $s_i[j]$

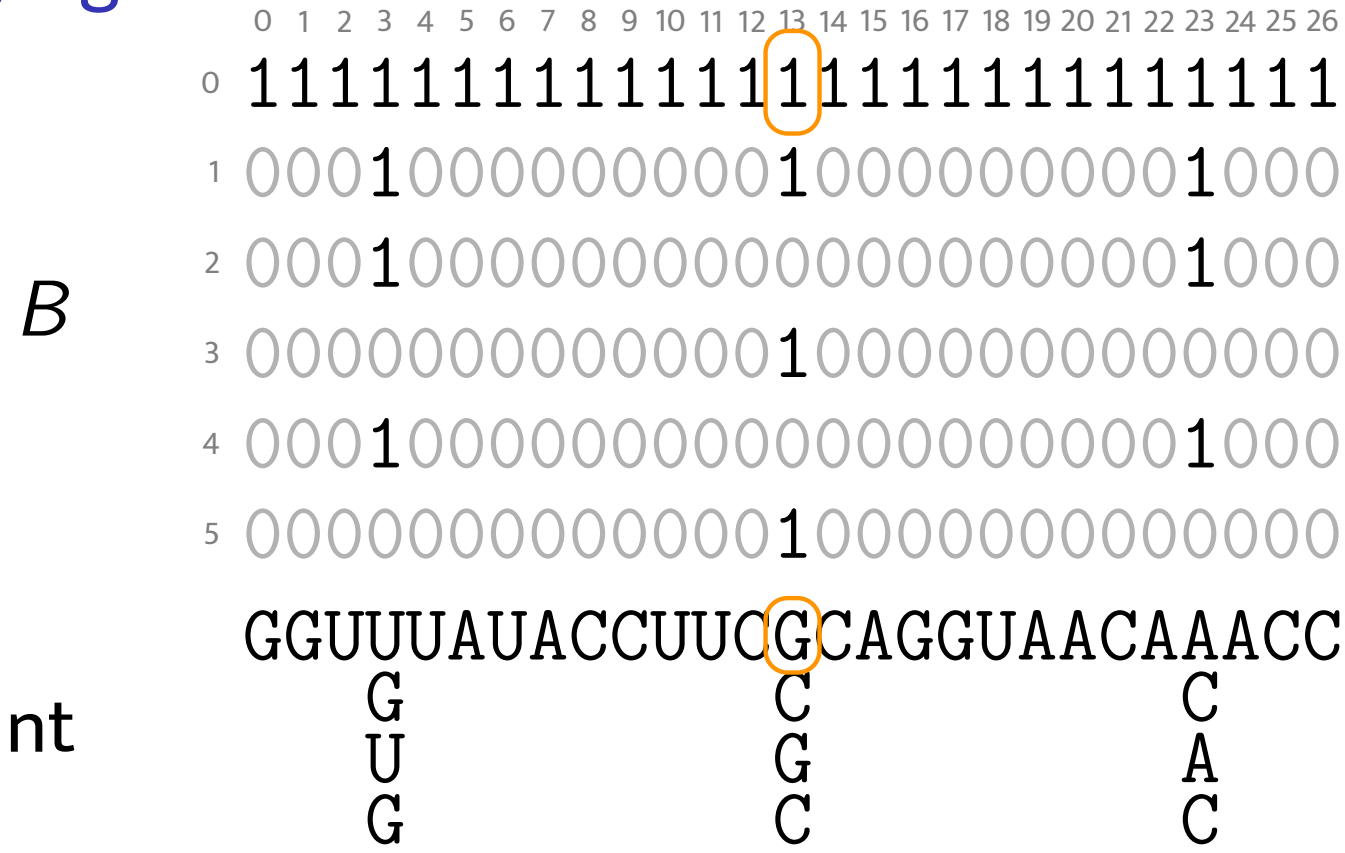
$$s_i[j] = \mathrm{nt}_j[\mathtt{rank}(B_j, i)]$$

$$\Theta(1)$$

Counting nt in col. j

Iterating on 1s (with `select`) to compute the length of each run

Querying CREMSA



Accessing $s_i[j]$

$$s_i[j] = \mathrm{nt}_j[\mathtt{rank}(B_j, i)]$$

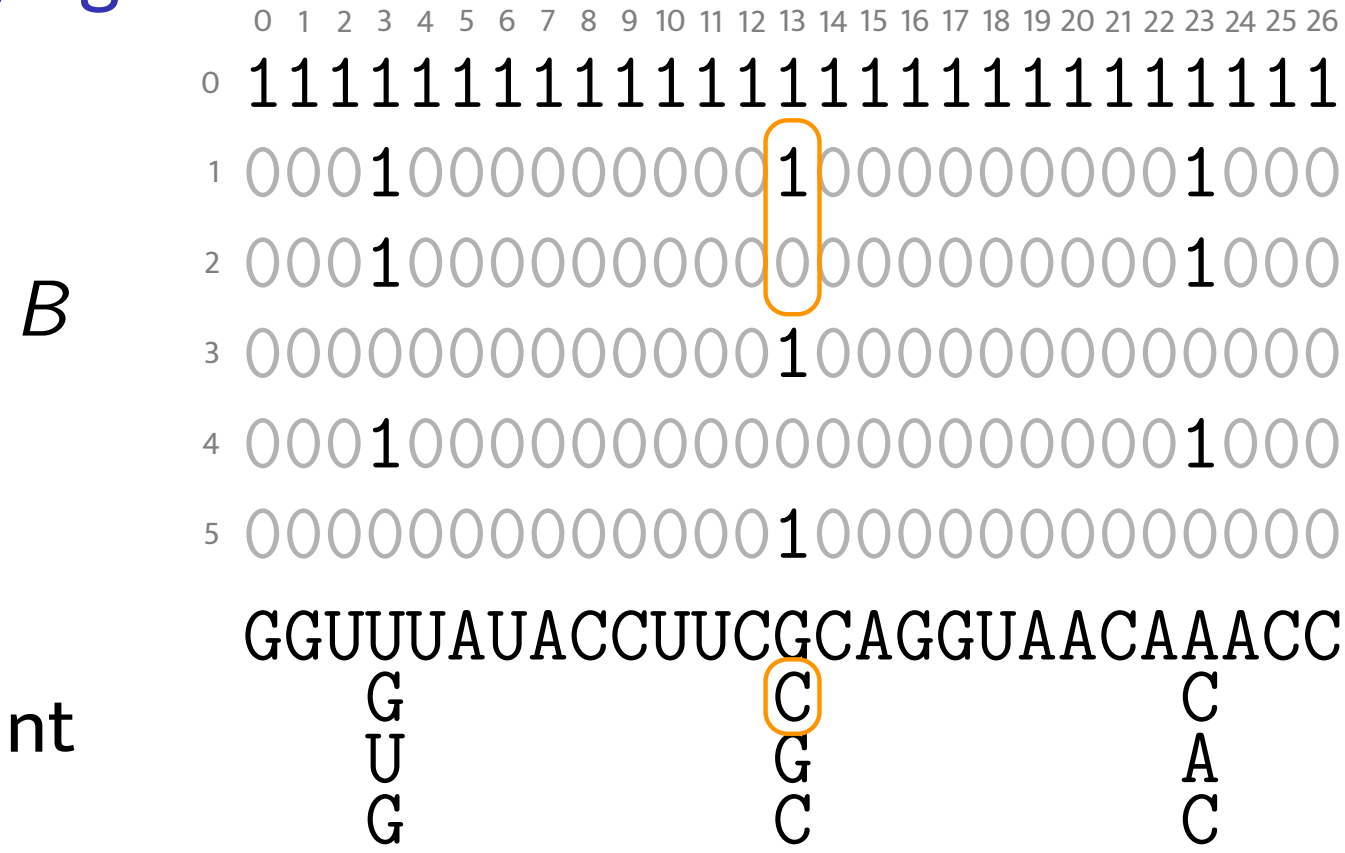
$$\Theta(1)$$

Counting nt in col. j

Iterating on 1s (with `select`) to compute the length of each run

A	C	G	T
0	0	1	0

Querying CREMSA



Accessing $s_i[j]$

$$s_i[j] = \mathrm{nt}_j[\mathtt{rank}(B_j, i)]$$

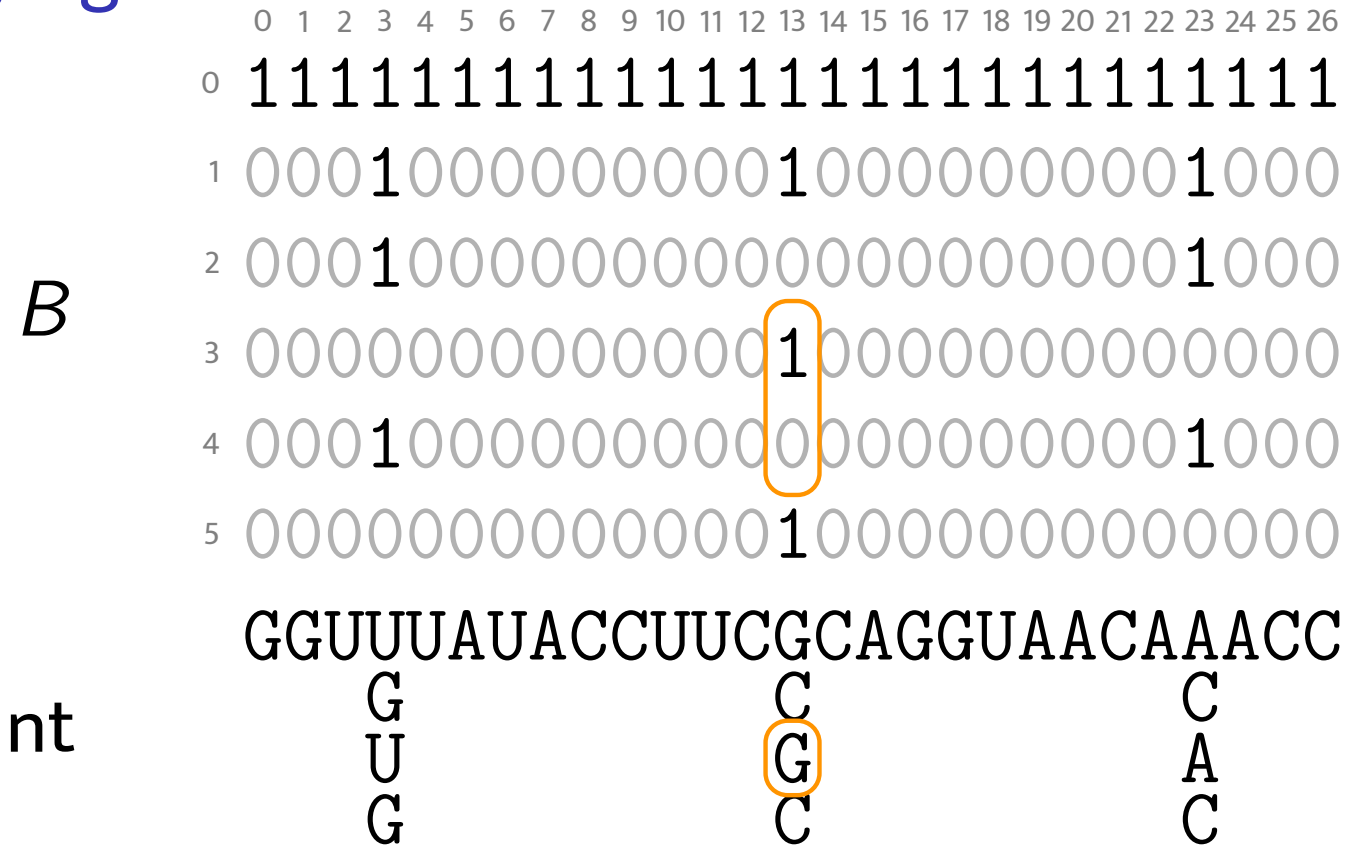
$$\Theta(1)$$

Counting nt in col. j

Iterating on 1s (with `select`) to compute the length of each run

A	C	G	T
0	2	1	0

Querying CREMSA



Accessing $s_i[j]$

$$s_i[j] = \mathrm{nt}_j[\mathtt{rank}(B_j, i)]$$

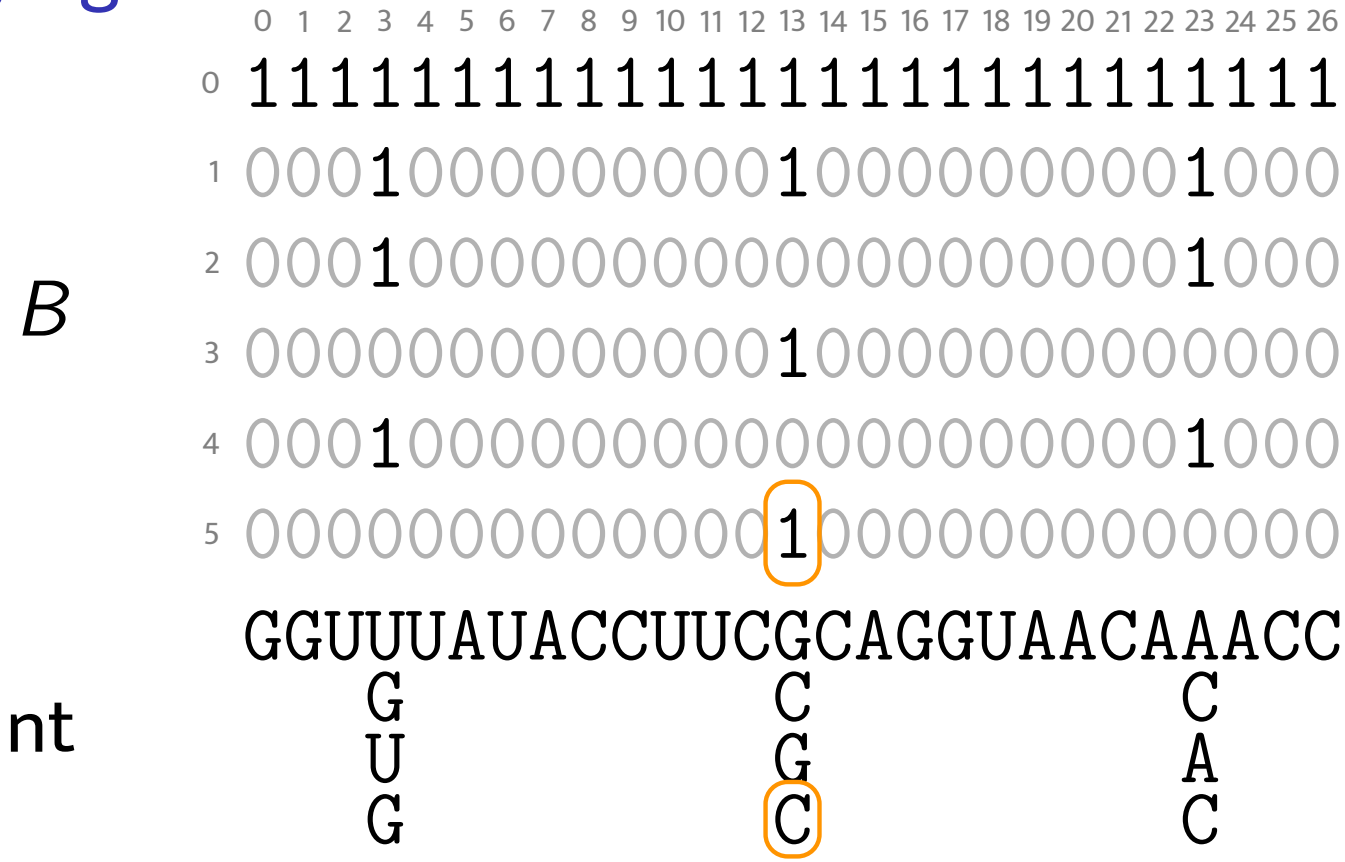
$$\Theta(1)$$

Counting nt in col. j

Iterating on 1s (with `select`) to compute the length of each run

A	C	G	T
0	2	3	0

Querying CREMSA



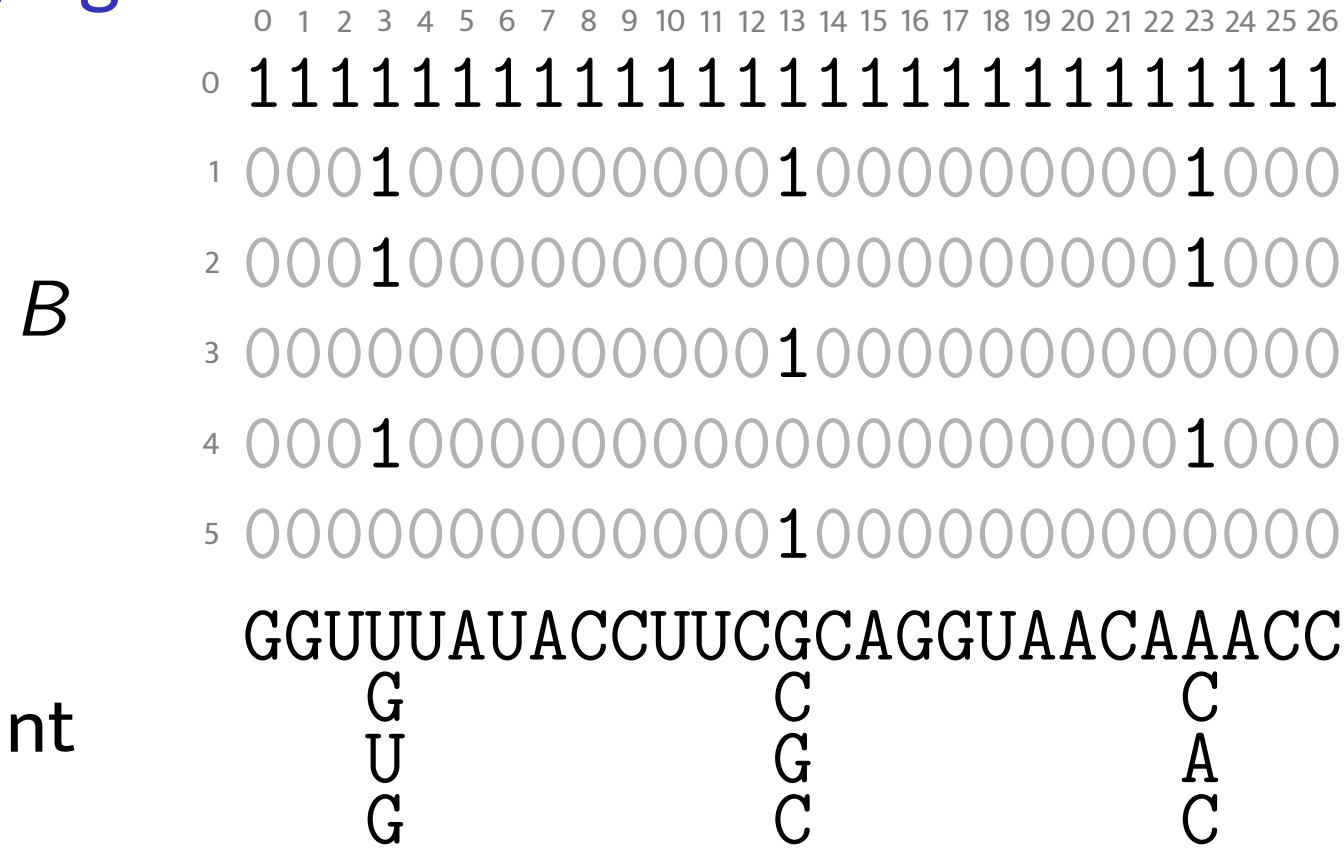
Accessing $s_i[j]$ Counting nt in col. j

$$s_i[j] = \mathrm{nt}_j[\mathtt{rank}(B_j, i)]$$

$\Theta(1)$

A	C	G	T
0	3	3	0

Querying CREMSA

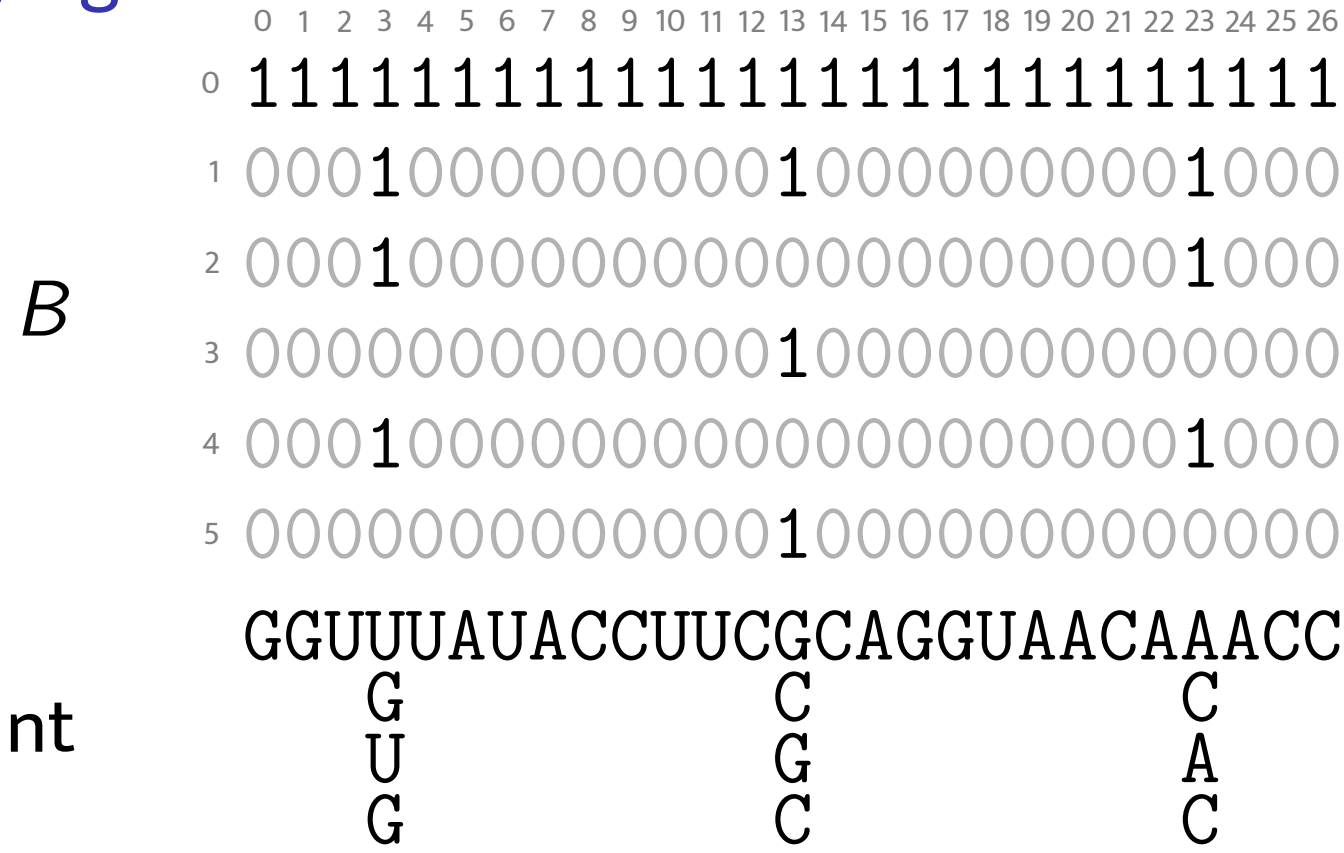


Accessing $s_i[j]$ Counting nt in col. j

$$s_i[j] = \mathrm{nt}_j[\mathtt{rank}(B_j, i)]$$

$\Theta(1)$ $\Theta(r_j)$

Querying CREMSA

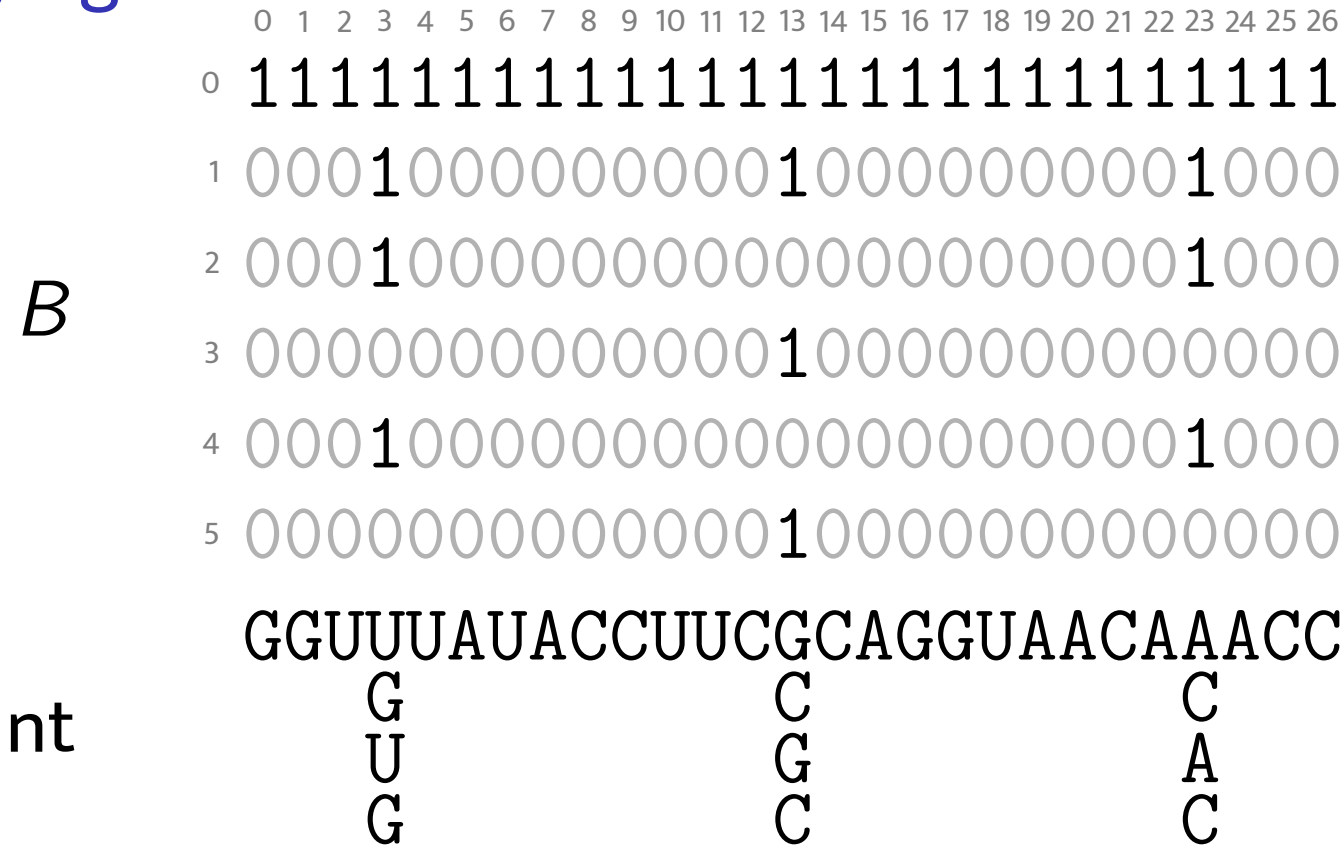


Accessing $s_i[j]$ Counting nt in col. j Counting in j_1, j_2

$$s_i[j] = \mathsf{nt}_j[\mathtt{rank}(B_j, i)]$$

$\Theta(1)$ $\Theta(r_j)$

Querying CREMSA

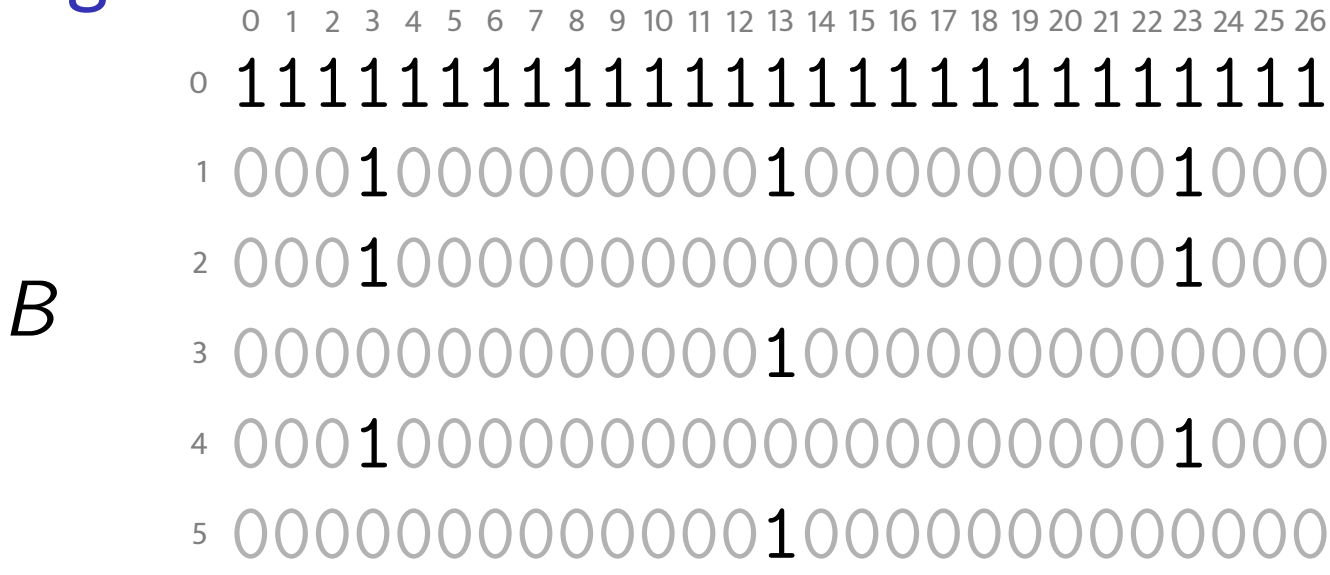


Accessing $s_i[j]$ Counting nt in col. j Counting in j_1, j_2

$s_i[j] = \mathsf{nt}_j[\mathtt{rank}(B_j, i)]$

$\Theta(1)$ $\Theta(r_j)$ $\Theta(\max(r_{j_1}, r_{j_2}))$

Querying CREMSA



The fewer runs, the faster the queries

U G A
G C C

Accessing $s_i[j]$ Counting nt in col. j Counting in j_1, j_2

$s_i[j] = \mathsf{nt}_j[\mathtt{rank}(B_j, i)]$

$\Theta(1)$ $\Theta(r_j)$ $\Theta(\max(r_{j_1}, r_{j_2}))$

Order of the sequences in the MSA matters

Fewer/longer runs lead to a lighter **and** quicker index

Order of the sequences in the MSA matters

Fewer/longer runs lead to a lighter **and** quicker index

GGUUUAUACCUUCGCAGGUAACAAACC
GGUGUAUACCUUCCCAGGUAACACACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUUUAUACCUUCGCAGGUAACAAACC
GGUGUAUACCUUCGCAGGUAACACACC
GGUGUAUACCUUCCCAGGUAACACACC

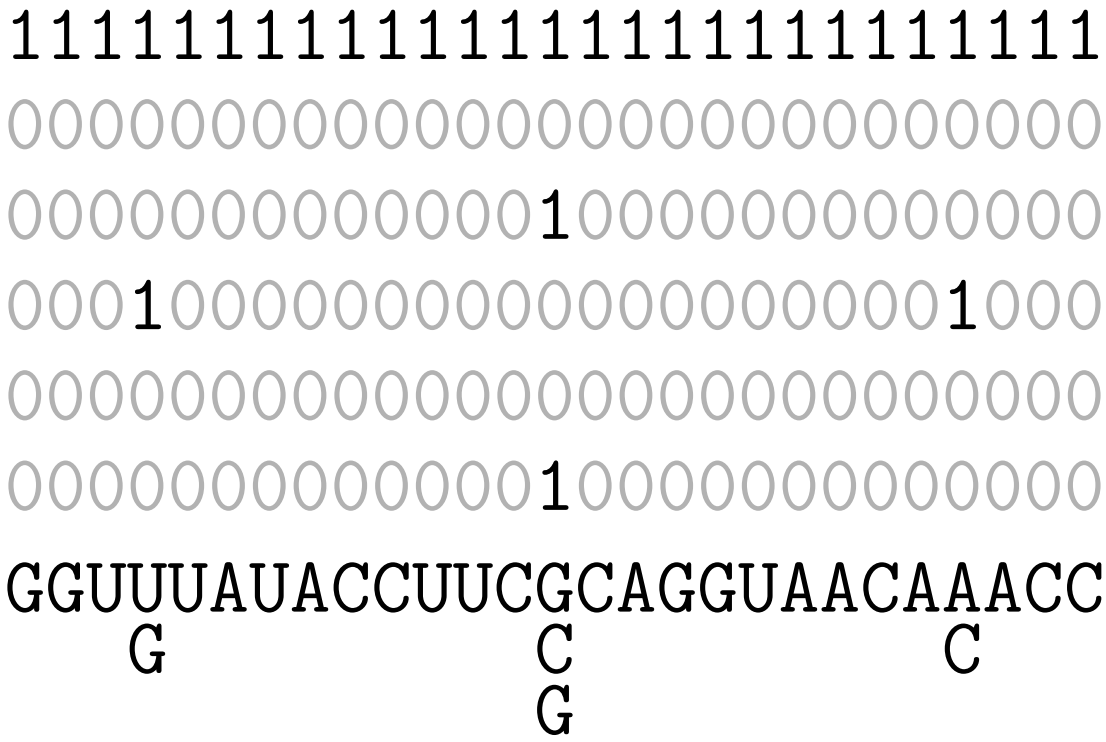
Order of the sequences in the MSA matters

Fewer/longer runs lead to a lighter **and** quicker index

GGUUUAUACCUUCGCAGGUAACAAACC
GGUUUAUACCUUCGCAGGUAACAAACC
GGUUUAUACCUUCCCAGGUAACAAACC
GGUGUAUACCUUCCCAGGUAACACACC
GGUGUAUACCUUCCCAGGUAACACACC
GGUGUAUACCUUCGCAGGUAACACACC

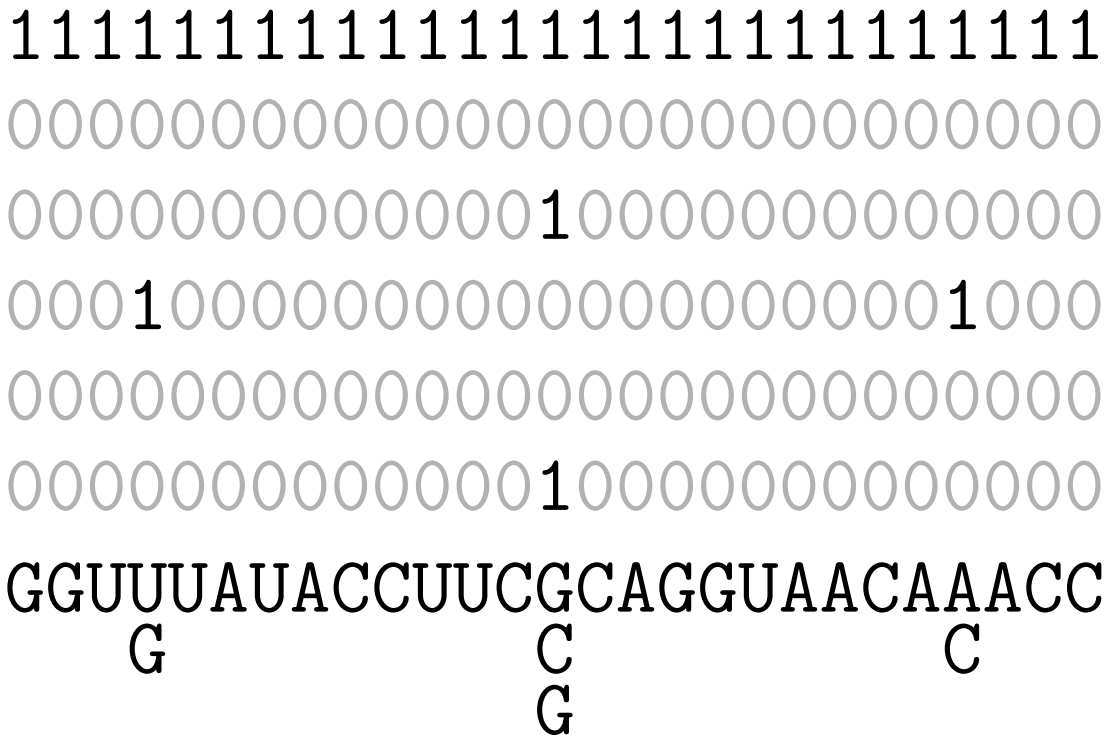
Order of the sequences in the MSA matters

Fewer/longer runs lead to a lighter **and** quicker index



Order of the sequences in the MSA matters

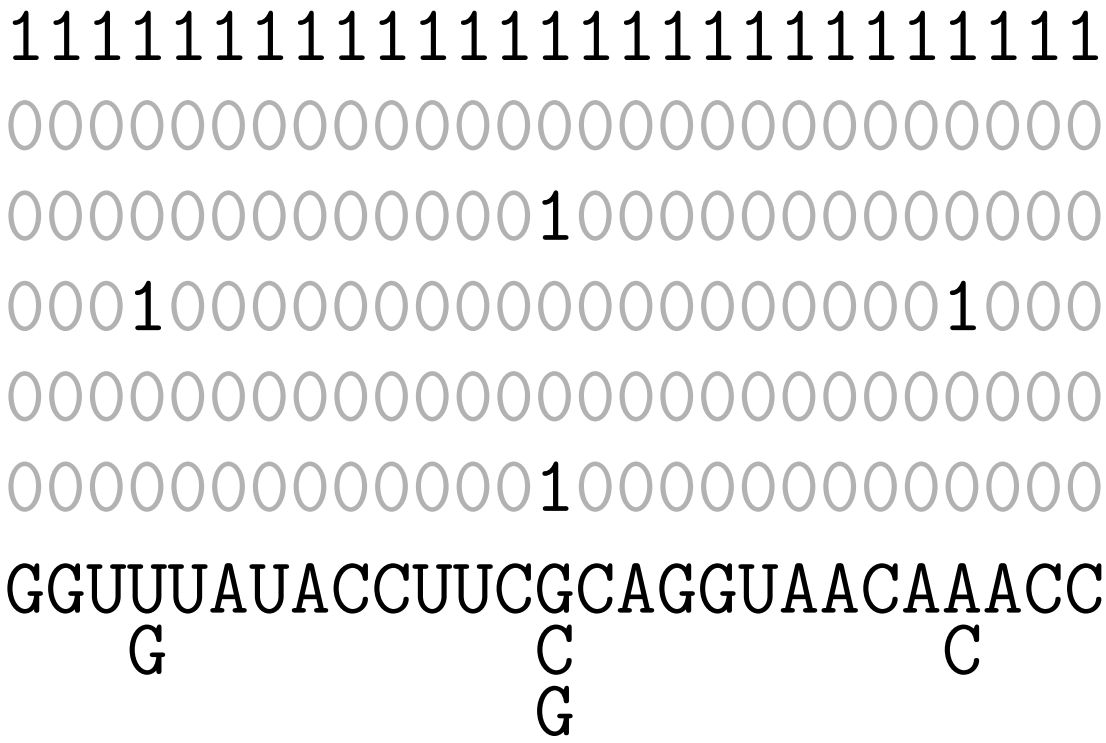
Fewer/longer runs lead to a lighter **and** quicker index



But finding the optimal order is NP-complete

Order of the sequences in the MSA matters

Fewer/longer runs lead to a lighter **and** quicker index



But finding the optimal order is NP-complete

Using known lineages offer good results in practice

CREMSA on 1M SARS-CoV-2 sequences

1,031,437 genomes

Downloaded in Dec 2023, deduplicated, removed those with $\geq$ 20 uncertain nucleotides

CREMSA on 1M SARS-CoV-2 sequences

1,031,437 genomes

Downloaded in Dec 2023, deduplicated, removed those with $\geq$ 20 uncertain nucleotides

Multiple sequence alignment

Using HAlign3, with 1.5 h CPU and **600 GB** of RAM. 36,000 columns

CREMSA on 1M SARS-CoV-2 sequences

1,031,437 genomes

Downloaded in Dec 2023, deduplicated, removed those with $\geq$ 20 uncertain nucleotides

Multiple sequence alignment

Using HAlign3, with 1.5 h CPU and **600 GB** of RAM. 36,000 columns

	CREMSA
Time	10 min
RAM	245 MB
Size on disk	53 MB

CREMSA on 1M SARS-CoV-2 sequences

1,031,437 genomes

Downloaded in Dec 2023, deduplicated, removed those with $\geq$ 20 uncertain nucleotides

Multiple sequence alignment

Using HAlign3, with 1.5 h CPU and **600 GB** of RAM. 36,000 columns

	CREMSA	gzip	xz
Time	10 min	90 min	70 min
RAM	245 MB	2 MB	97 MB
Size on disk	53 MB	11,873 MB	33 MB

CREMSA on 1M SARS-CoV-2 sequences

1,031,437 genomes

Downloaded in Dec 2023, deduplicated, removed those with $\geq$ 20 uncertain nucleotides

Multiple sequence alignment

Using HAlign3, with 1.5 h CPU and **600 GB** of RAM. 36,000 columns

	CREMSA	gzip	xz	CoMSA
Time	10 min	90 min	70 min	16 min
RAM	245 MB	2 MB	97 MB	63,266 MB
Size on disk	53 MB	11,873 MB	33 MB	9 MB

CREMSA on 1M SARS-CoV-2 sequences

1,031,437 genomes

Downloaded in Dec 2023, deduplicated, removed those with $\geq$ 20 uncertain nucleotides

Multiple sequence alignment

Using HAlign3, with 1.5 h CPU and **600 GB** of RAM. 36,000 columns

	CREMSA	gzip	xz	CoMSA
Time	10 min	90 min	70 min	16 min
RAM	245 MB	2 MB	97 MB	63,266 MB
Sorted input	28 MB		24 MB	
Size on disk	53 MB	11,873 MB	33 MB	9 MB

CREMSA on 1M SARS-CoV-2 sequences

1,031,437 genomes

Downloaded in Dec 2023, deduplicated, removed those with $\geq$ 20 uncertain nucleotides

Multiple sequence alignment

Using HAlign3, with 1.5 h CPU and **600 GB** of RAM. 36,000 columns

	Index		Compressors		
	CREMSA		gzip	xz	CoMSA
Time	10 min		90 min	70 min	16 min
RAM	245 MB		2 MB	97 MB	63,266 MB
Sorted input	28 MB			24 MB	
Size on disk	53 MB		11,873 MB	33 MB	9 MB

CREMSA on 1M SARS-CoV-2 sequences

1,031,437 genomes

Downloaded in Dec 2023, deduplicated, removed those with $\geq$ 20 uncertain nucleotides

Multiple sequence alignment

Using HAlign3, with 1.5 h CPU and **600 GB** of RAM. 36,000 columns

Index

	CREMSA	gzip	xz	CoMSA
Time	10 min	90 min	70 min	16 min
RAM	245 MB	2 MB	97 MB	63,266 MB
Sorted input	28 MB		24 MB	
Size on disk	53 MB	11,873 MB	33 MB	9 MB
Access	0.1 μs			
Count	30 μs			
Cons. score	40 μs			

Compressors

CREMSA on 1M SARS-CoV-2 sequences

1,031,437 genomes

Downloaded in Dec 2023, deduplicated, removed those with $\geq$ 20 uncertain nucleotides

Multiple sequence alignment

Using HAlign3, with 1.5 h CPU and **600 GB** of RAM. 36,000 columns

Index

	CREMSA	gzip	xz	CoMSA
Time	10 min	90 min	70 min	16 min
RAM	245 MB	2 MB	97 MB	63,266 MB
Sorted input	28 MB		24 MB	
Size on disk	53 MB	11,873 MB	33 MB	9 MB
Access	0.1 μs			
Count	30 μs			
Cons. score	40 μs			

Compressors

$\rightarrow \sim$ 6h for all the pairs of columns

What conservation scores on 1M genomes tell us?

What conservation scores on 1M genomes tell us?

Not much (yet)

Only 251 pairs have a good conservation score

Computing conservation scores on 1M genomes is great but...

What conservation scores on 1M genomes tell us?

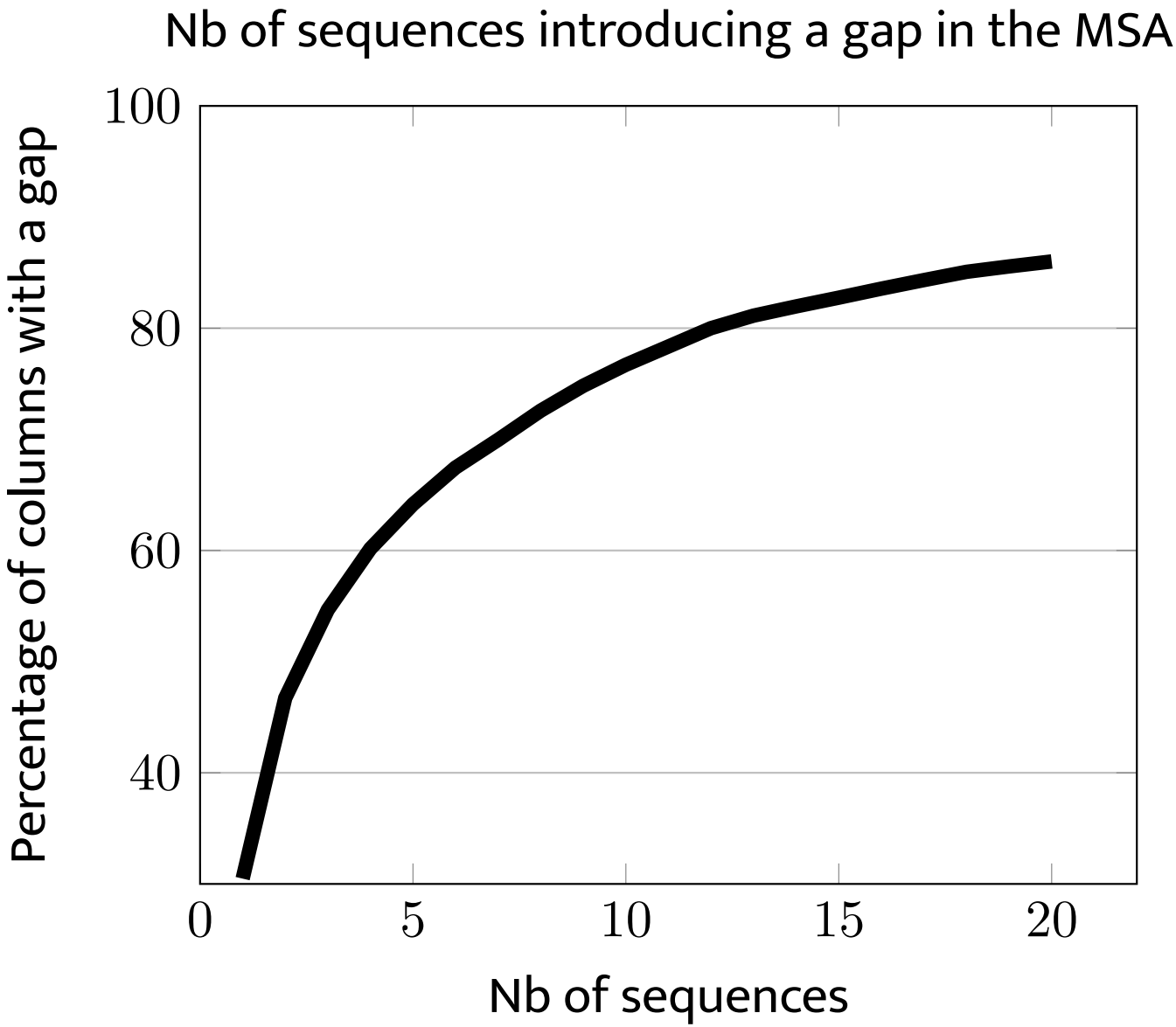
Not much (yet)

Only 251 pairs have a good conservation score

Computing conservation scores on 1M genomes is great but...

More data $\rightarrow$ more signal? **or** More data $\rightarrow$ more noise?

A few sequences introduce gaps in a column



CREMSA : Compressed indexing of (ultra) large viral alignments

https://gitlab.univ-lille.fr/mikael.salson/cremsa

CREMSA is fast and lightweight

CREMSA : Compressed indexing of (ultra) large viral alignments

https://gitlab.univ-lille.fr/mikael.salson/cremsa

CREMSA is fast and lightweight

Reordering sequences save time and space

CREMSA : Compressed indexing of (ultra) large viral alignments

https://gitlab.univ-lille.fr/mikael.salson/cremsa

CREMSA is fast and lightweight

Reordering sequences save time and space

Finding the optimal order is NP-hard

CREMSA : Compressed indexing of (ultra) large viral alignments

https://gitlab.univ-lille.fr/mikael.salson/cremsa

CREMSA is fast and lightweight

Reordering sequences save time and space

Finding the optimal order is NP-hard

Conservation scores can be computed for 1M SARS-CoV-2

But we identified little interesting candidates

CREMSA : Compressed indexing of (ultra) large viral alignments

https://gitlab.univ-lille.fr/mikael.salson/cremsa

CREMSA is fast and lightweight

Reordering sequences save time and space

Finding the optimal order is NP-hard

Conservation scores can be computed for 1M SARS-CoV-2

But we identified little interesting candidates

Perspectives: Identify with CREMSA the sequences messing up the MSA

CREMSA : Compressed indexing of (ultra) large viral alignments

https://gitlab.univ-lille.fr/mikael.salson/cremsa

CREMSA is fast and lightweight

Reordering sequences save time and space

Finding the optimal order is NP-hard

Conservation scores can be computed for 1M SARS-CoV-2

But we identified little interesting candidates

Perspectives: Identify with CREMSA the sequences messing up the MSA

Side note: anyone interested to design a **lightweight** method for multiple sequence aligments of highly similar sequences?

Announcement: Open Maître de conférences position in Lille

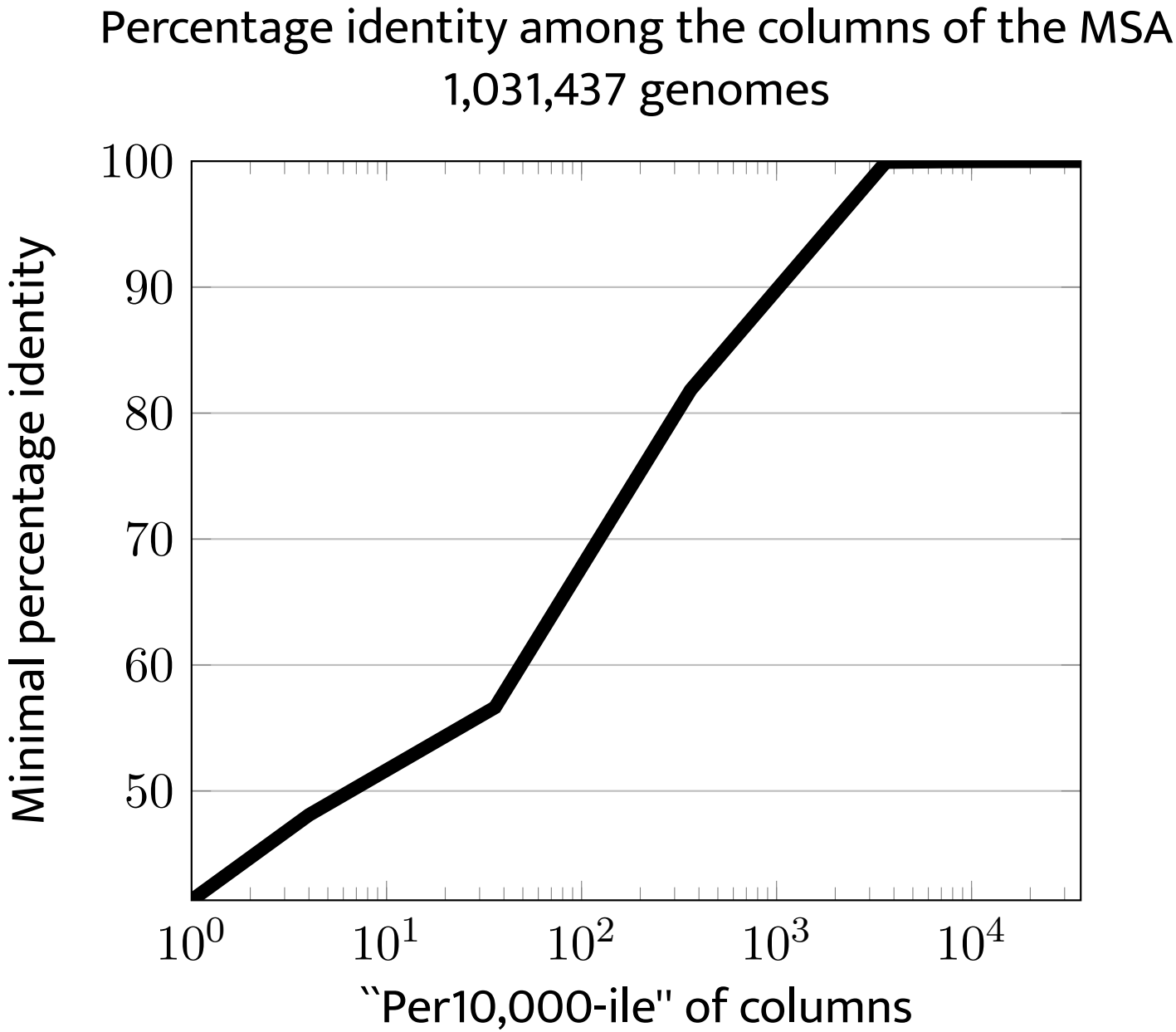
Maître de conférences position open in Bonsai (CRIStAL), Lille

Research themes in Bonsai

data structures, sequence algorithms, hash functions
oncohematology
non-ribosomal peptides
paleoproteomics
...

Contact me if interested: `mikael.salson@univ-lille.fr`

The 1M genomes are highly conserved



In practice *r* is tiny