

Algorithms in bioinformatics

Anton Bankevich
Center for Algorithmic Biotechnology
SPbSU

Bioinformatics vs Algorithmic Biology

Algorithmic biology:

development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems.

Bioinformatics:

field that develops methods and software tools for understanding biological data

Bioinformatics vs Algorithmic Biology

Algorithmic biology:



Bioinformatics:



Bioinformatics vs Algorithmic Biology

Algorithmic biology:



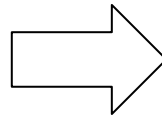
Bioinformatics:



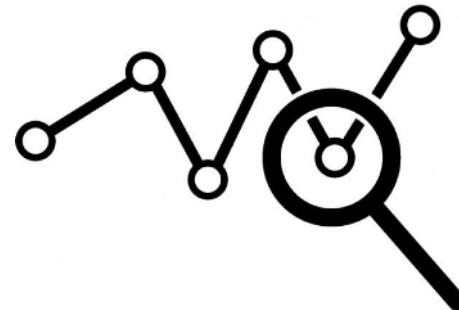
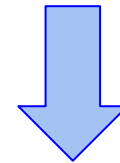
Discovery



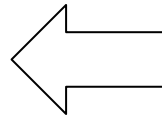
Technology



Data



Analysis



Discovery

Data

What is biological data?

- Sequencing reads
- Mass spectrometry
- Expression measurements

Questions about sequences

- Are they the same?
- Is one of them a part of another?
- What is common between sequences?

Questions about sequences

- Are they similar?
- Is one of them similar to a part of another?
- What is similar in the sequences?

Alignment

AACGCTAACGGTAA

AACCGCGAACTAA

Alignment

AACGCTAACGGTAA

AACCGCGAACTAA



AAC - GCTAACGGTAA

AACCGCGAAC - - TAA

Needleman–Wunsch algorithm

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

Score(x, x) = 1 (match)

Score(x, y) = -1 (mismatch)

Score(x, -) = Score(-, x) = -1 (Indel)

Score(ACGTTAG, ACCTAG) = 3

Needleman–Wunsch algorithm

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

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1							
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1							
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1							
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1						
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1						
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1/-2						
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1/-2						
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1/-2/-2						
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1						
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1						
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	-2					
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	-2/0					
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	-2/0/-3					
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0					
C	-2							
C	-3							
T	-4							
A	-5							
G	-6							

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0						
C	-3							
T	-4							
A	-5							
G	-6							

Diagram illustrating the Needleman–Wunsch algorithm. The table shows the dynamic programming table for sequence alignment between two sequences: A (top row) and ACGCTAG (left column). The top row and left column contain the sequence characters. The table cells contain the scores for each alignment. Red arrows indicate the path of the optimal alignment, starting from the bottom-left cell (A, G) and moving up and right to the top-right cell (A, A). The cell (A, A) is highlighted in green, and the cell (A, G) is highlighted in orange.

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0						
C	-3							
T	-4							
A	-5							
G	-6							

The image shows a dynamic programming table for the Needleman-Wunsch algorithm. The table is 8x8, with the first row and column representing the sequence "ACCTAAG". The top-left cell (0,0) is 0. The rest of the first row and first column contain values from -1 to -7. The second row (A) contains values from -1 to -5. The second column (C) contains values from -2 to 0. The cell at row 3, column 3 (C-C) is highlighted in orange. Red arrows indicate backpointers: from (1,1) to (0,0), from (1,2) to (1,1), from (1,3) to (1,2), from (1,4) to (1,3), from (1,5) to (1,4), from (1,6) to (1,5), from (1,7) to (1,6), from (1,8) to (1,7), from (2,2) to (1,1), and from (2,3) to (2,2).

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2					
C	-3							
T	-4							
A	-5							
G	-6							

The image shows a dynamic programming table for the Needleman–Wunsch algorithm. The table is a grid with 8 columns and 8 rows. The columns are labeled with nucleotides: A, C, G, T, T, A, G. The rows are labeled with nucleotides: A, C, C, T, A, G. The top-left cell (0,0) is 0. The first row contains values from -1 to -7. The first column contains values from -1 to -6. The cell at row A, column C (1,2) is highlighted in green and contains the value 1. The cell at row C, column G (2,3) is highlighted in orange and contains the value 2. Red arrows indicate the path of the algorithm: from (0,0) to (1,1), from (1,1) to (1,2), from (1,2) to (2,3), from (2,3) to (2,4), from (2,4) to (2,5), from (2,5) to (2,6), from (2,6) to (2,7), from (2,7) to (2,8), from (2,8) to (1,8), from (1,8) to (1,7), from (1,7) to (1,6), from (1,6) to (1,5), from (1,5) to (1,4), from (1,4) to (1,3), from (1,3) to (1,2), and from (1,2) to (0,1).

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2/-1					
C	-3							
T	-4							
A	-5							
G	-6							

Diagram illustrating the Needleman–Wunsch algorithm. The table shows the dynamic programming table for sequence alignment. The top row and left column contain the sequence indices (0 to -7). The top row labels are A, C, G, T, T, A, G. The left column labels are A, C, C, T, A, G. The cell (A, C) is highlighted in green (value 0). The cell (C, G) is highlighted in orange (value 2/-1). Red arrows indicate the path of the alignment, starting from (A, G) and moving left to (A, C), then up to (C, C), and finally up to (A, A).

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2/-1/-1					
C	-3							
T	-4							
A	-5							
G	-6							

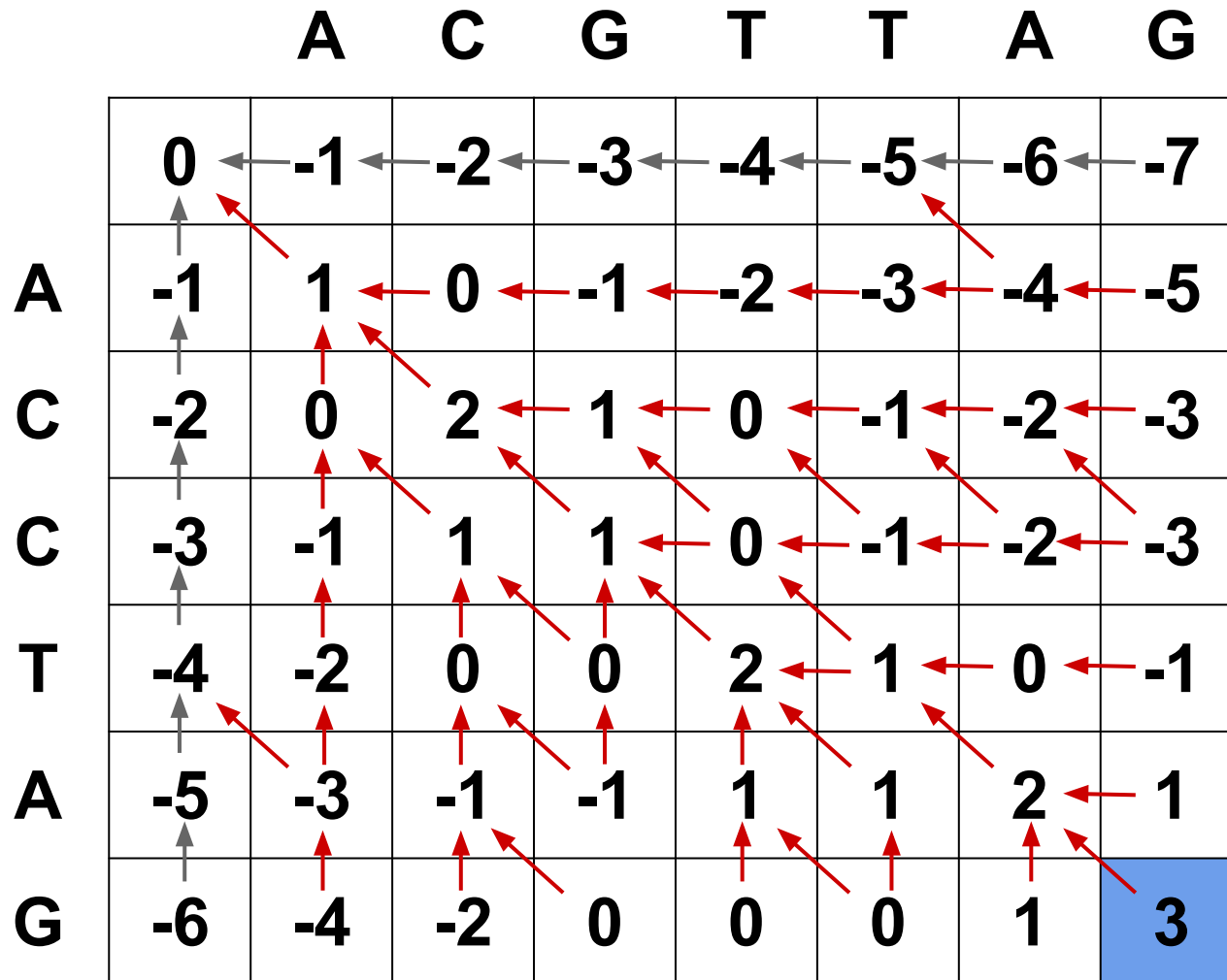
Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2					
C	-3							
T	-4							
A	-5							
G	-6							

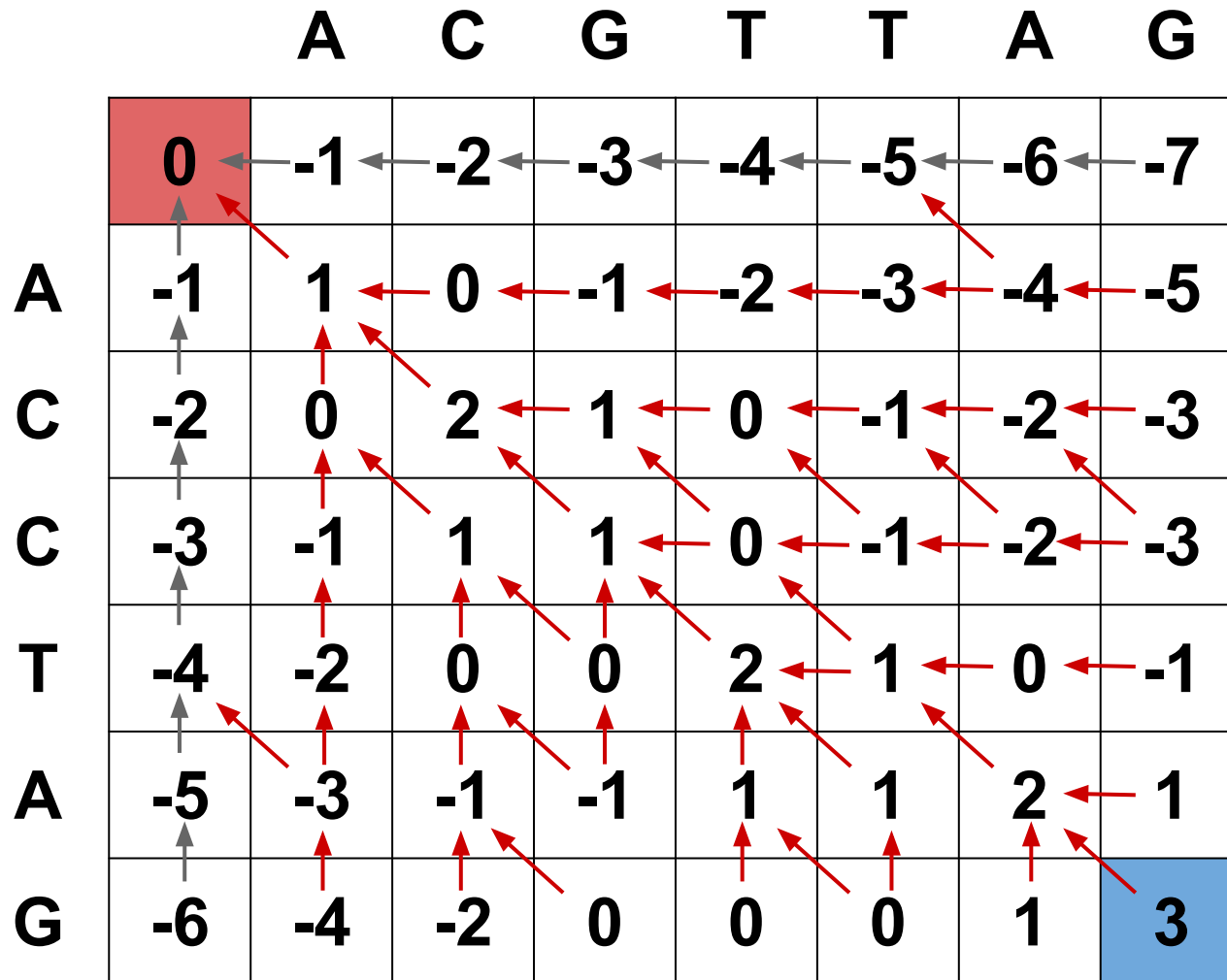
Needleman–Wunsch algorithm

$$M(i, j) = \text{MAX} \begin{cases} M(i - 1, j - 1) + \text{Score}(X(i), Y(j)) \\ M(i - 1, j) + \text{Score}(X(i), '-') \\ M(i, j - 1) + \text{Score}('-', Y(j)) \end{cases}$$

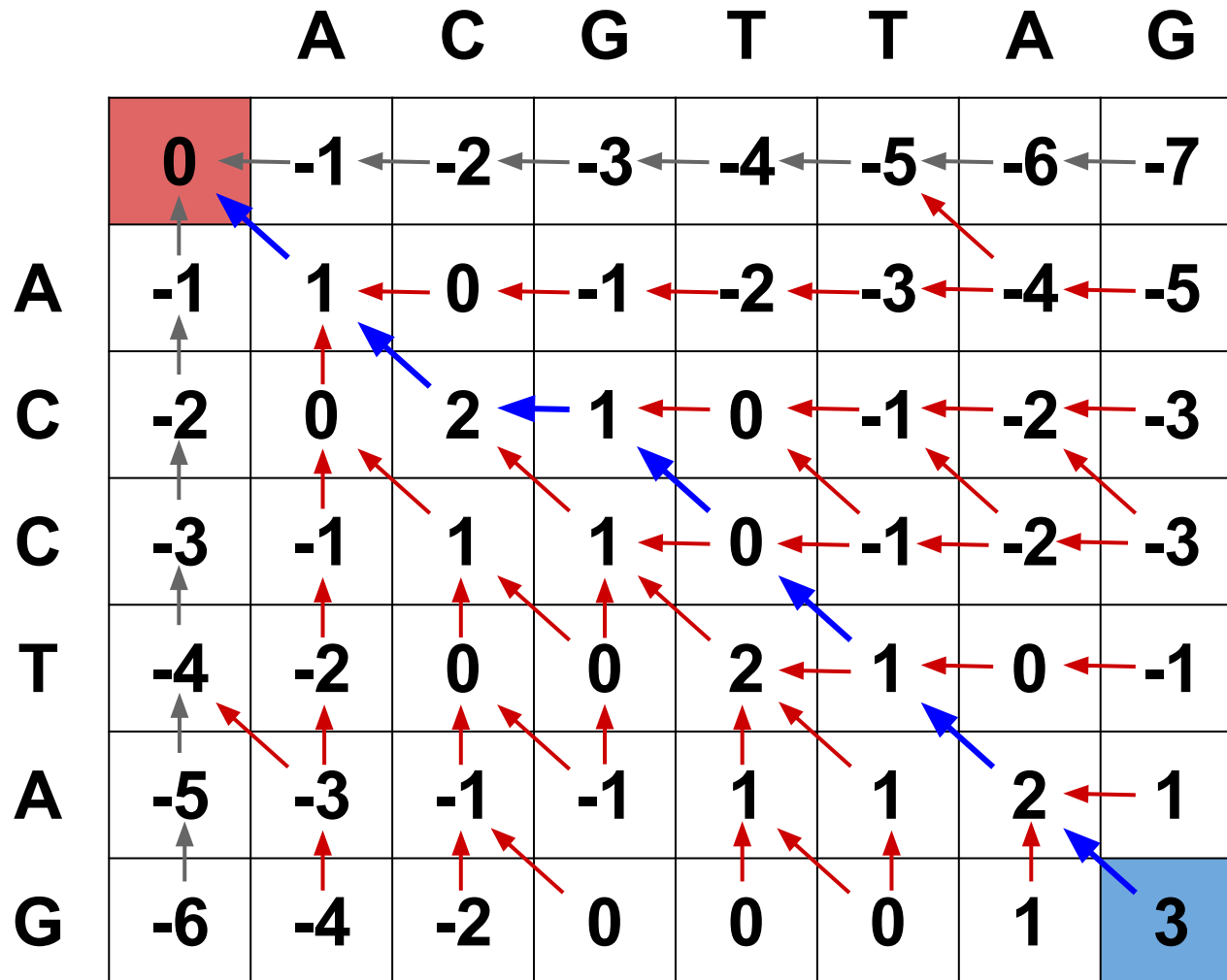
Needleman–Wunsch algorithm



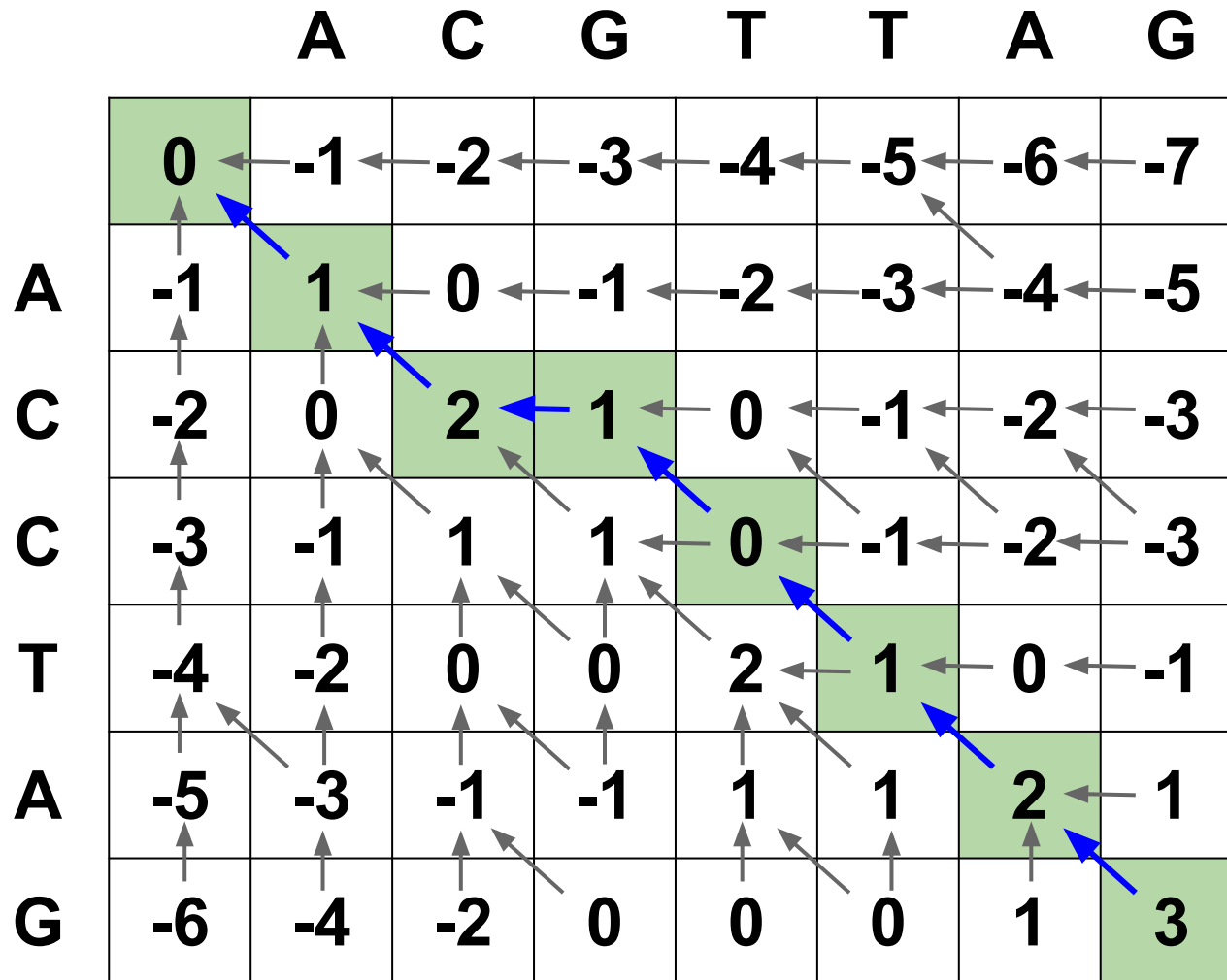
Needleman–Wunsch algorithm



Needleman–Wunsch algorithm



Needleman–Wunsch algorithm



Needleman–Wunsch algorithm

G	G

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2	1	0	-1	-2	-3
C	-3	-1	1	1	0	-1	-2	-3
T	-4	-2	0	0	2	1	0	-1
A	-5	-3	-1	-1	1	1	2	1
G	-6	-4	-2	0	0	0	1	3

Needleman–Wunsch algorithm

A	A
G	G

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2	1	0	-1	-2	-3
C	-3	-1	1	1	0	-1	-2	-3
T	-4	-2	0	0	2	1	0	-1
A	-5	-3	-1	-1	1	1	2	1
G	-6	-4	-2	0	0	0	1	3

Needleman–Wunsch algorithm

T	T
A	A
G	G

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2	1	0	-1	-2	-3
C	-3	-1	1	1	0	-1	-2	-3
T	-4	-2	0	0	2	1	0	-1
A	-5	-3	-1	-1	1	1	2	1
G	-6	-4	-2	0	0	0	1	3

Needleman–Wunsch algorithm

C	T
T	T
A	A
G	G

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2	1	0	-1	-2	-3
C	-3	-1	1	1	0	-1	-2	-3
T	-4	-2	0	0	2	1	0	-1
A	-5	-3	-1	-1	1	1	2	1
G	-6	-4	-2	0	0	0	1	3

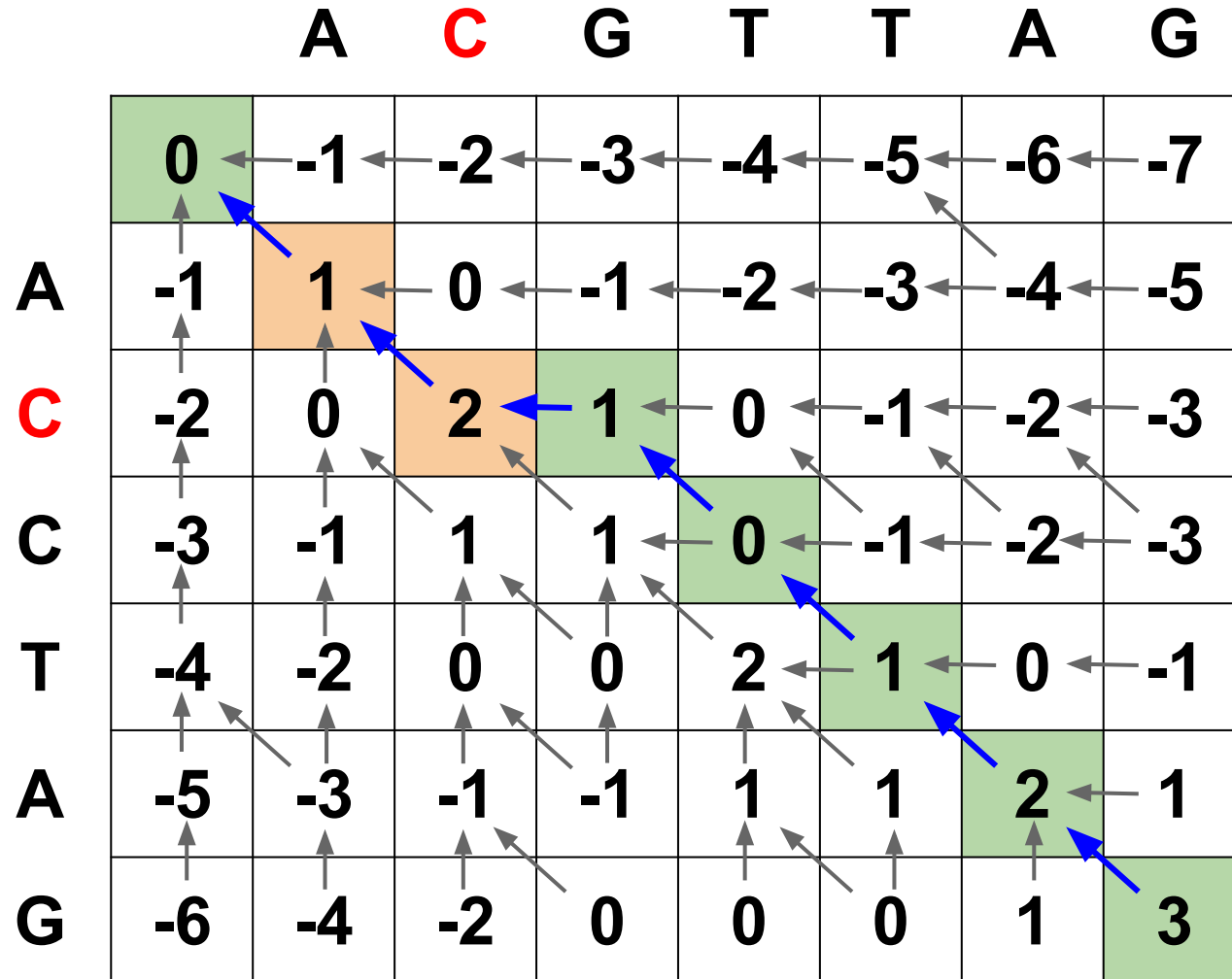
Needleman–Wunsch algorithm

-	G
C	T
T	T
A	A
G	G

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2	1	0	-1	-2	-3
C	-3	-1	1	1	0	-1	-2	-3
T	-4	-2	0	0	2	1	0	-1
A	-5	-3	-1	-1	1	1	2	1
G	-6	-4	-2	0	0	0	1	3

Needleman–Wunsch algorithm

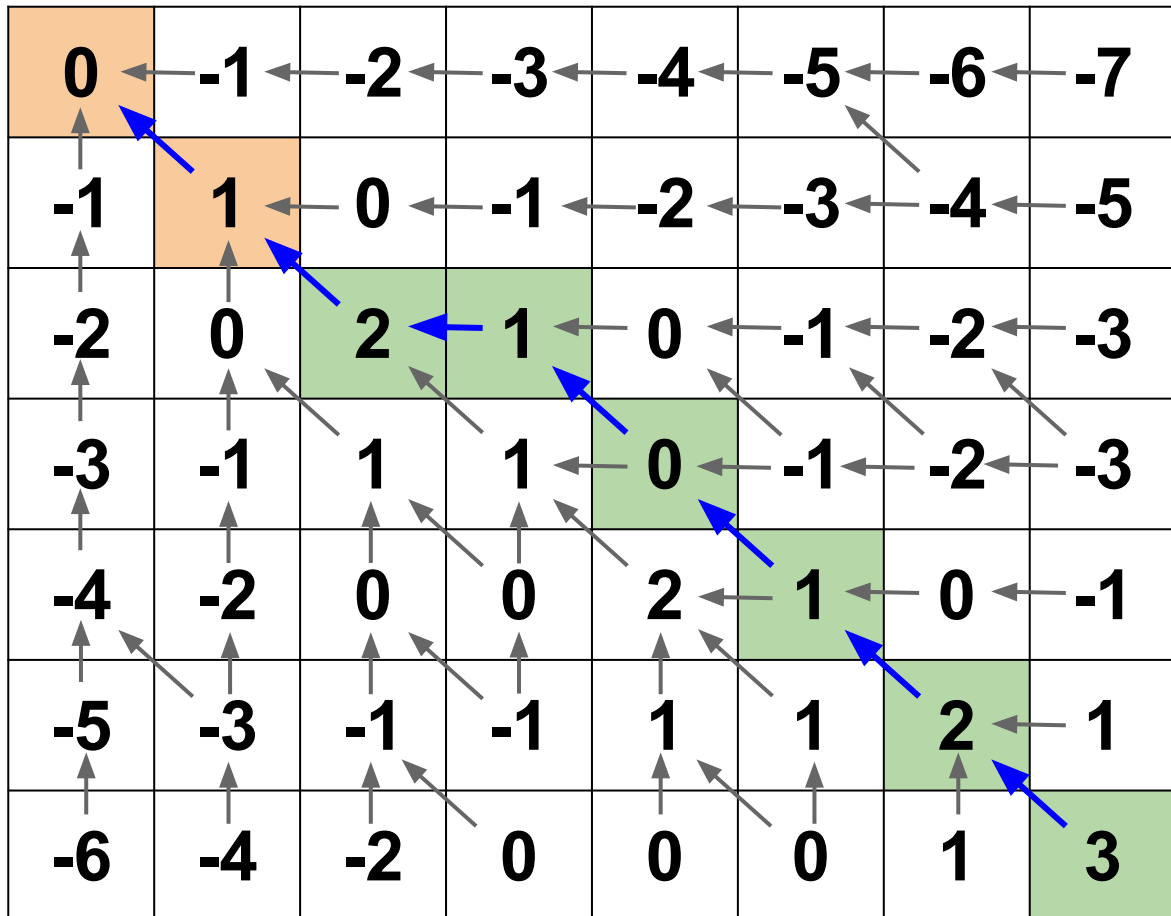
C	C
-	G
C	T
T	T
A	A
G	G



Needleman–Wunsch algorithm

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

	A	C	G	T	T	A	G	
A	0	-1	-2	-3	-4	-5	-6	-7
C	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2	1	0	-1	-2	-3
C	-3	-1	1	1	0	-1	-2	-3
T	-4	-2	0	0	2	1	0	-1
A	-5	-3	-1	-1	1	1	2	1
G	-6	-4	-2	0	0	0	1	3



Needleman–Wunsch algorithm

A C G T T A G

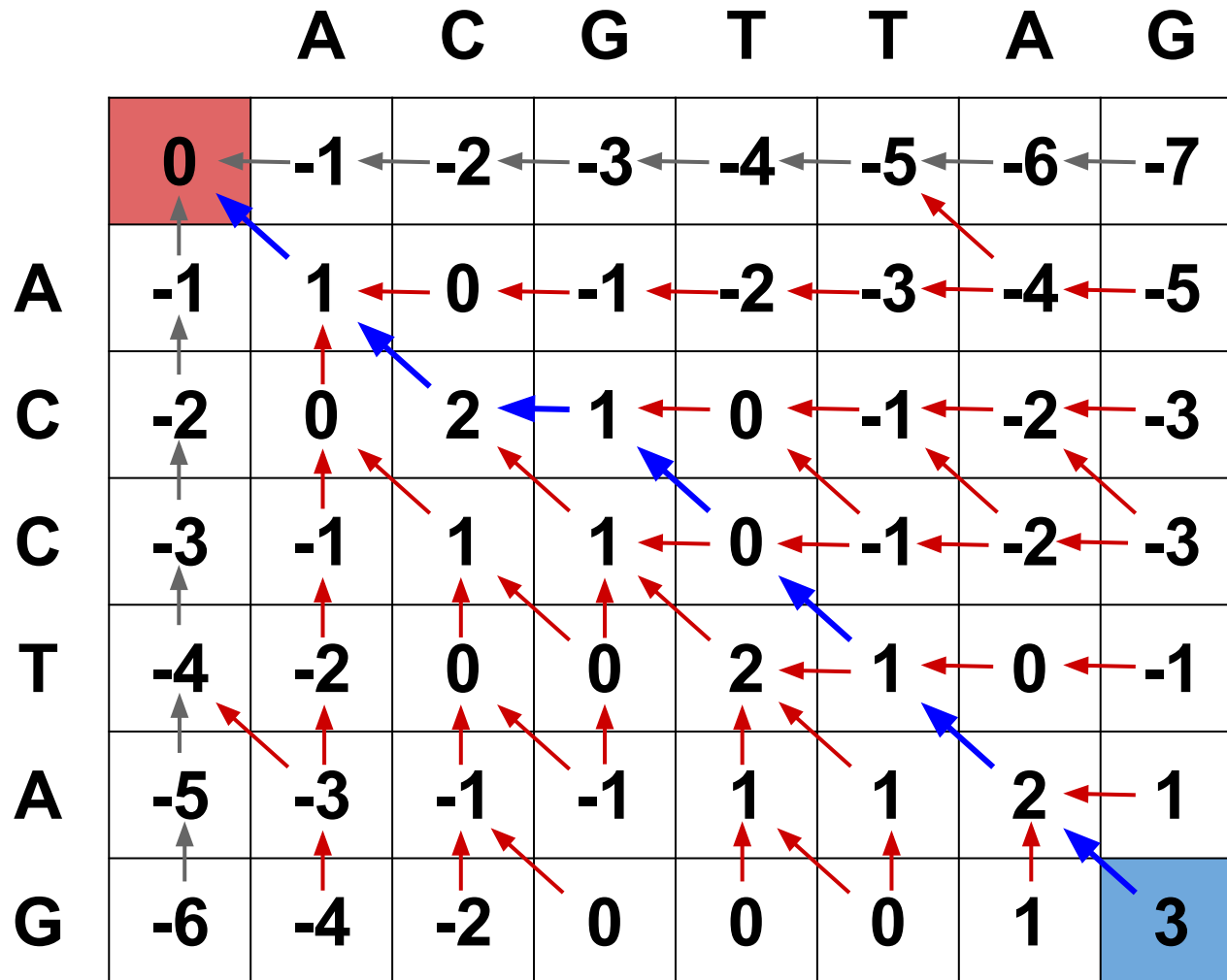
A C C T - A G

A C G T T A G

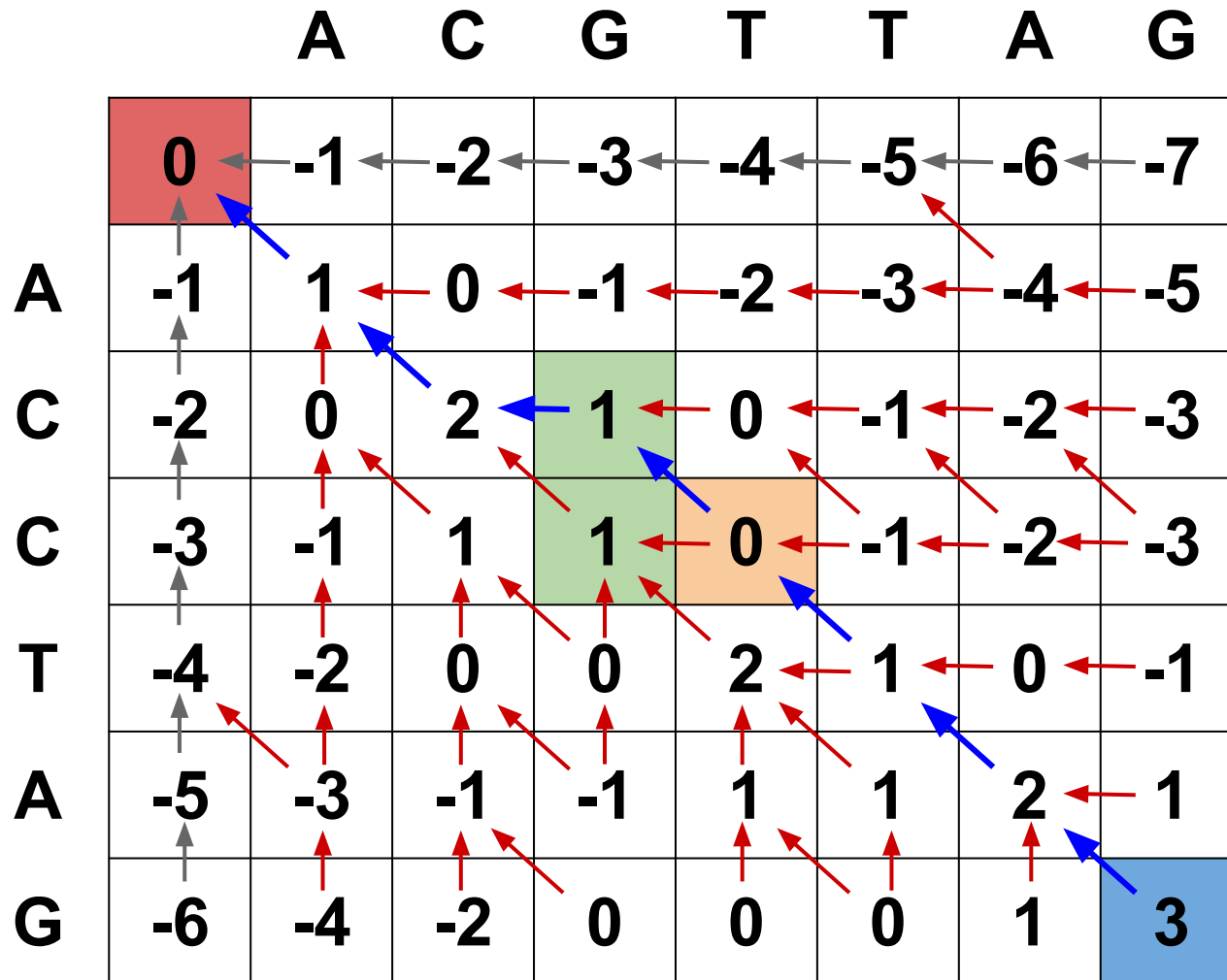
A C - C T A G

$$\text{Score}(\text{ACGTTAG}, \text{ACCTAG}) = 3$$

Needleman–Wunsch algorithm



Needleman–Wunsch algorithm



Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2	1	0	-1	-2	-3
C	-3	-1	1	1	0	-1	-2	-3
T	-4	-2	0	0	2	1	0	-1
A	-5	-3	-1	-1	1	1	2	1
G	-6	-4	-2	0	0	0	1	3

Needleman–Wunsch algorithm

	A	C	G	T	T	A	G	
	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
C	-2	0	2	1	0	-1	-2	-3
C	-3	-1	1	1	0	-1	-2	-3
T	-4	-2	0	0	2	1	0	-1
A	-5	-3	-1	-1	1	1	2	1
G	-6	-4	-2	0	0	0	1	3

Needleman–Wunsch algorithm

Time =

Memory =

Needleman–Wunsch algorithm

$$\text{Time} = O(n^2)$$

$$\text{Memory} = O(n^2)$$

How about the remaining questions?

- Are they similar?
- Is one of them similar to a part of another?
- What is similar in the sequences?

Needleman–Wunsch algorithm

	A	C	G	C	G	A	G	
	0	0	0	0	0	0	0	
G	-1	-1	-1	1	0	1	0	1
C	-2	-2	0	0	2	1	0	-1
G	-3	-3	-1	-1	1	3	2	1

- Is one of them similar to a part of another?

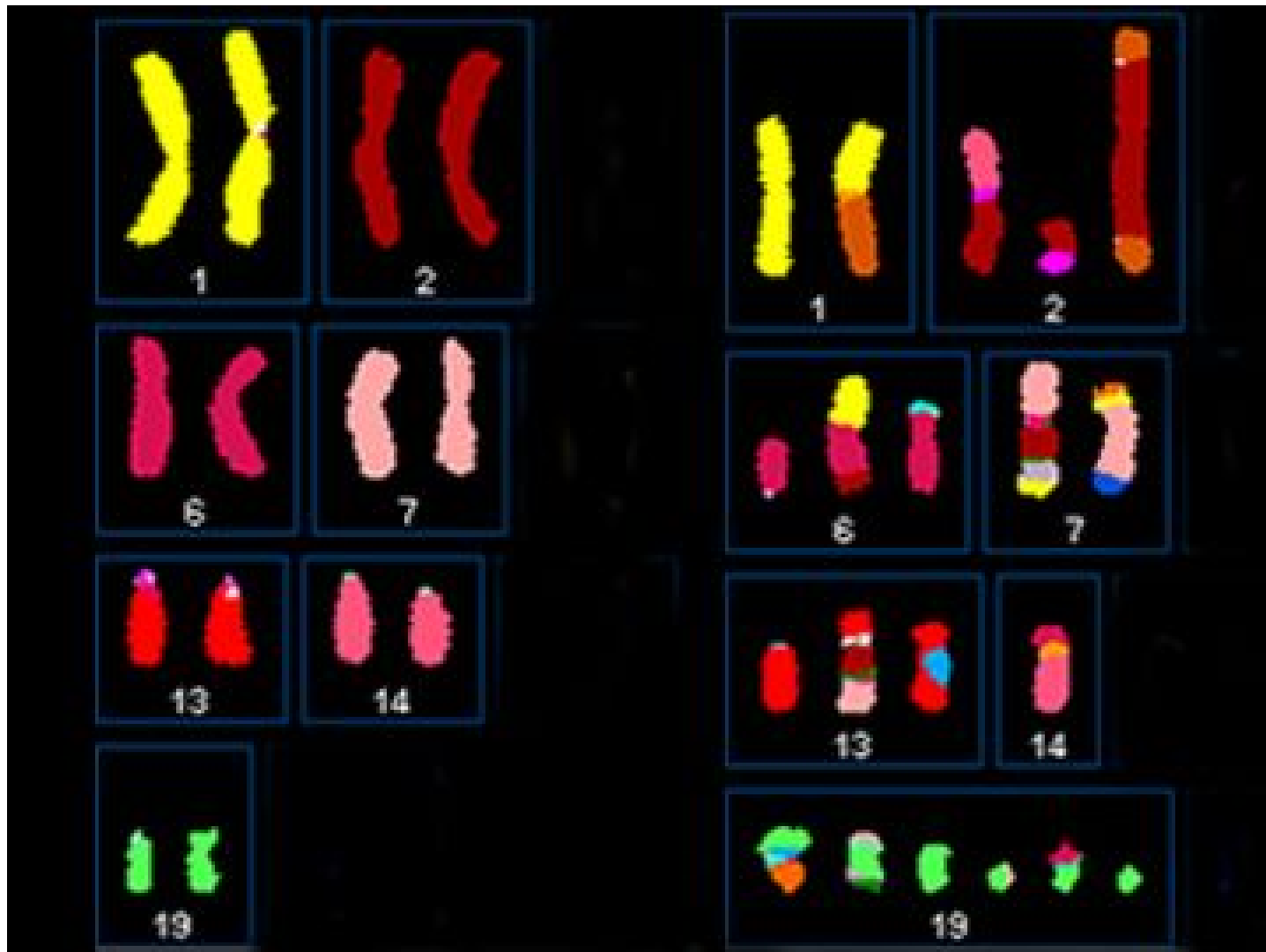
How about the remaining questions?

- Are they similar?
- Is one of them similar to a part of another?
- What is similar in the sequences?

Why is this not enough?

- We deal with genomes of extreme size
- We have extreme number of sequencing reads
- We need more careful alignment that can handle structural rearrangements

Rearrangements



Burrows-Wheeler transform

a c a a c g

Burrows-Wheeler transform

a c a a c g \$

Burrows-Wheeler transform

a c a a c g \$
\$ a c a a c g

Burrows-Wheeler transform

a c a a c g \$
\$ a c a a c g
g \$ a c a a c

Burrows-Wheeler transform

a c a a c g \$
\$ a c a a c g
g \$ a c a a c
c g \$ a c a a
a c g \$ a c a
a a c g \$ a c
c a a c g \$ a

Burrows-Wheeler transform

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

a c a a c g \$



g c \$ a a a c

Burrows-Wheeler transform

a c a a c g \$
↑ ?
g c \$ a a a c

Burrows-Wheeler transform

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

g \$ a c a a c
c a a c g \$ a
\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c g \$ a c a a

Burrows-Wheeler transform

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

g	\$ a c a a c g
c	a a c g \$ a c
\$	a c a a c g \$
a	a c g \$ a c a
a	c a a c g \$ a
a	c g \$ a c a a
c	g \$ a c a a c

Burrows-Wheeler transform

\$	\$ a c a a c g
a	a a c g \$ a c
a	a c a a c g \$
a	a c g \$ a c a
c	c a a c g \$ a
c	c g \$ a c a a
g	g \$ a c a a c

Burrows-Wheeler transform

\$
a
a
a
c
c
g

\$	a	c	a	a	c	g
a	a	c	g	\$	a	c
a	c	a	a	c	g	\$
a	c	g	\$	a	c	a
c	a	a	c	g	\$	a
c	g	\$	a	c	a	a
g	\$	a	c	a	a	c

Burrows-Wheeler transform

g \$	\$ a c a a c g
c a	a a c g \$ a c
\$ a	a c a a c g \$
a a	a c g \$ a c a
a c	c a a c g \$ a
a c	c g \$ a c a a
c g	g \$ a c a a c

Burrows-Wheeler transform

\$ a

a a

a c

a c

c a

c g

g \$

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

Burrows-Wheeler transform

\$ a

a a

a c

a c

c a

c g

g \$

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

Burrows-Wheeler transform

g \$ a

c a a

\$ a c

a a c

a c a

a c g

c g \$

\$ a c a a c **g**

a a c g \$ a **c**

a c a a c g **\$**

a c g \$ a c **a**

c a a c g \$ **a**

c g \$ a c a **a**

g \$ a c a a **c**

Burrows-Wheeler transform

\$ a c

a a c

a c a

a c g

c a a

c g \$

g \$ a

\$ a c a a c **g**

a a c g **\$** a **c**

a c a a c g **\$**

a c g **\$** a c **a**

c a a c g **\$** **a**

c g **\$** a c a **a**

g **\$** a c a a **c**

Burrows-Wheeler transform

\$ a c

a a c

a c a

a c g

c a a

c g \$

g \$ a

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

Burrows-Wheeler transform

g \$ a c

c a a c

\$ a c a

a a c g

a c a a

a c g \$

c g \$ a

\$ a c a a c **g**

a a c g \$ a **c**

a c a a c g **\$**

a c g \$ a c **a**

c a a c g \$ **a**

c g \$ a c a **a**

g \$ a c a a **c**

Burrows-Wheeler transform

\$ a c a

a a c g

a c a a

a c g \$

c a a c

c g \$ a

g \$ a c

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

Burrows-Wheeler transform

g \$ a c a

c a a c g

\$ a c a a

a a c g \$

a c a a c

a c g \$ a

c g \$ a c

\$ a c a a c **g**

a a c g \$ a **c**

a c a a c g **\$**

a c g \$ a c **a**

c a a c g \$ **a**

c g \$ a c a **a**

g \$ a c a a **c**

Burrows-Wheeler transform

\$ a c a a
a a c g \$
a c a a c
a c g \$ a
c a a c g
c g \$ a c
g \$ a c a

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

g \$ a c a a

c a a c g \$

\$ a c a a c

a a c g \$ a

a c a a c g

a c g \$ a c

c g \$ a c a

\$ a c a a c **g**

a a c g \$ a **c**

a c a a c g **\$**

a c g \$ a c **a**

c a a c g \$ **a**

c g \$ a c a **a**

g \$ a c a a **c**

Burrows-Wheeler transform

\$ a c a a c
a a c g \$ a
a c a a c g
a c g \$ a c
c a a c g \$
c g \$ a c a
g \$ a c a a

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

g \$ a c a a c

c a a c g \$ a

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c g \$ a c a a

\$ a c a a c **g**

a a c g \$ a **c**

a c a a c g **\$**

a c g \$ a c **a**

c a a c g \$ **a**

c g \$ a c a **a**

g \$ a c a a **c**

Burrows-Wheeler transform

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

First-last property

$\$$ ₁	a	c	a	a	c	g
a ₁	a	c	g	$\$$	a	c
a ₂	c	a	a	c	g	$\\$
a ₃	c	g	$\$$	a	c	a
c ₁	a	a	c	g	$\$$	a
c ₂	g	$\$$	a	c	a	a
g ₁	$\$$	a	c	a	a	c

First-last property

$\$$ ₁ a c a a c **g**₁
a₁ a c g \$ a **c**
a₂ c a a c g **\$**₁
a₃ c g \$ a c **a**
c₁ a a c g \$ **a**
c₂ g \$ a c a **a**
g₁ \$ a c a a **c**

First-last property

$\$$ ₁ a c a a c **g**₁

a₁ a c g \$ a **c**

a₂ c a a c g **\$**₁

a₃ c g \$ a c **a**

c₁ a a c g \$ **a**

c₂ g \$ a c a **a**

g₁ \$ a c a a **c**

First-last property

a_3 c g \$ a c **a**
 c_1 a a c g \$ **a**
 c_2 g \$ a c a **a**

$\$$ ₁ a c a a c **g**₁
 a ₁ a c g \$ a **c**
 a ₂ c a a c g **\$**₁
a₃ c g \$ a c **a**
c₁ a a c g \$ **a**
c₂ g \$ a c a **a**
g₁ \$ a c a a **c**

First-last property

a a₃ c g \$ a c
a c₁ a a c g \$
a c₂ g \$ a c a

\$₁ a c a a c **g**₁
a₁ a c g \$ a **c**
a₂ c a a c g \$₁
a₃ c g \$ a c **a**
c₁ a a c g \$ **a**
c₂ g \$ a c a **a**
g₁ \$ a c a a **c**

First-last property

a a₃ c g \$ a c
a c₁ a a c g \$
a c₂ g \$ a c a

\$₁ a c a a c **g**₁
a₁ a c g \$ a **c**
a₂ c a a c g \$₁
a₃ c g \$ a c **a**
c₁ a a c g \$ **a**
c₂ g \$ a c a **a**
g₁ \$ a c a a **c**

First-last property

a_1 a_3 c g \$ a c
 a_2 c_1 a a c g \$
 a_3 c_2 g \$ a c a

$\$1$ a c a a c g_1
 a_1 a c g \$ a c
 a_2 c a a c g $\$1$
 a_3 c g \$ a c a
 c_1 a a c g \$ a
 c_2 g \$ a c a a
 g_1 \$ a c a a c

First-last property

a₃ c g \$ a c **a₁**
c₁ a a c g \$ **a₂**
c₂ g \$ a c a **a₃**

\$₁ a c a a c **g₁**
a₁ a c g \$ a **c**
a₂ c a a c g **\$₁**
a₃ c g \$ a c **a**
c₁ a a c g \$ **a**
c₂ g \$ a c a **a**
g₁ \$ a c a a **c**

First-last property

a_3 c g \$ a c a_1
 c_1 a a c g \$ a_2
 c_2 g \$ a c a a_3

$\$1$ a c a a c g_1
 a_1 a c g \$ a c
 a_2 c a a c g $\$1$
 a_3 c g \$ a c a
 c_1 a a c g \$ a
 c_2 g \$ a c a a
 g_1 \$ a c a a c

First-last property

a_3 c g \$ a c a_1
 c_1 a a c g \$ a_2
 c_2 g \$ a c a a_3

$\$1$ a c a a c g_1
 a_1 a c g \$ a c
 a_2 c a a c g $\$1$
 a_3 c g \$ a c a_1
 c_1 a a c g \$ a_2
 c_2 g \$ a c a a_3
 g_1 \$ a c a a c

First-last property

$\$$ ₁	a	c	a	a	c	g	g ₁
a ₁	a	c	g	$\$$	a	c	c ₁
a ₂	c	a	a	c	g	$\$$	$\$$ ₁
a ₃	c	g	$\$$	a	c	a	a ₁
c ₁	a	a	c	g	$\$$	a	a ₂
c ₂	g	$\$$	a	c	a	a	a ₃
g ₁	$\$$	a	c	a	a	a	c ₂

Pattern search

$\$$ ₁	a	c	a	a	c	g	g ₁
a ₁	a	c	g	$\$$	a	c ₁	
a ₂	c	a	a	c	g	$\$$ ₁	
a ₃	c	g	$\$$	a	c	a ₁	
c ₁	a	a	c	g	$\$$	a ₂	
c ₂	g	$\$$	a	c	a	a ₃	
g ₁	$\$$	a	c	a	a	c ₂	

a a c

Pattern search

$\$$ ₁	a	c	a	a	c	g	g ₁	a	a	c
a ₁	a	c	g	\$	a	c	c ₁			
a ₂	c	a	a	c	g	\$	$\$$ ₁			
a ₃	c	g	\$	a	c	a	a ₁			
c ₁	a	a	c	g	\$	a	a ₂			
c ₂	g	\$	a	c	a	a	a ₃			
g ₁	\$	a	c	a	a	c	c ₂			

Pattern search

$\$$ ₁	a	c	a	a	c	g	g ₁	a	a	c
a ₁	a	c	g	\$	a	c	c ₁			
a ₂	c	a	a	c	g	\$	$\$$ ₁			
a ₃	c	g	\$	a	c	a	a ₁			
c ₁	a	a	c	g	\$	a	a ₂			
c ₂	g	\$	a	c	a	a	a ₃			
g ₁	\$	a	c	a	a	c	c ₂			

Pattern search

$\$$ ₁ a c a a c **g**₁

a a c

a₁ a c g \$ a **c**₁

a₂ c a a c g **\$**₁

a₃ c g \$ a c **a**₁

c₁ a a c g \$ **a**₂

c₂ g \$ a c a **a**₃

g₁ \$ a c a a **c**₂

Pattern search

$\$$ ₁ a c a a c **g**₁

a a c

a₁ a c g \$ a **c**₁

a₂ c a a c g **\$**₁

a₃ c g \$ a c **a**₁

c₁ a a c g \$ **a**₂

c₂ g \$ a c a **a**₃

g₁ \$ a c a a **c**₂

Pattern search

$\$$ ₁	a	c	a	a	c	g	g ₁
a	a	c	g	$\$$	a	c	c ₁
a	c	a	a	c	g	$\$$	$\$$ ₁
a	c	g	$\$$	a	c	a	a ₁
c	a	a	c	g	$\$$	a	a ₂
c	g	$\$$	a	c	a	a	a ₃
g	$\$$	a	c	a	a	a	c ₂

a a c

Pattern search

$\$$ ₁	a	c	a	a	c	g	g ₁
a ₁	a	c	g	$\$$	a	c ₁	
a ₂	c	a	a	c	g	$\$$ ₁	
a ₃	c	g	$\$$	a	c	a ₁	
c ₁	a	a	c	g	$\$$	a ₂	
c ₂	g	$\$$	a	c	a	a ₃	
g ₁	$\$$	a	c	a	a	c ₂	

a a c

Pattern search

$\$$ ₁	a	c	a	a	c	g	g ₁
a ₁	a	c	g	$\$$	a	c ₁	
a ₂	c	a	a	c	g	$\$$ ₁	
a ₃	c	g	$\$$	a	c	a ₁	
c ₁	a	a	c	g	$\$$	a ₂	
c ₂	g	$\$$	a	c	a	a ₃	
g ₁	$\$$	a	c	a	a	c ₂	

a a c

Pattern search

$\$$ ₁	a	c	a	a	c	g	g ₁		a	a	c
a	a ₁	c	g	$\$$	a	c	c ₁				
a	a ₂	c	a	a	c	g	$\$$ ₁				
a	a ₃	c	g	$\$$	a	c	a ₁				
c	c ₁	a	a	c	g	$\$$	a ₂				
c	c ₂	g	$\$$	a	c	a	a ₃				
g	g ₁	$\$$	a	c	a	a	c ₂				

Pattern search

$\$$ ₁	a	c	a	a	c	g	g ₁
a ₁	a	c	g	$\$$	a	c ₁	
a ₂	c	a	a	c	g	$\$$ ₁	
a ₃	c	g	$\$$	a	c	a ₁	
c ₁	a	a	c	g	$\$$	a ₂	
c ₂	g	$\$$	a	c	a	a ₃	
g ₁	$\$$	a	c	a	a	c ₂	

a a c

Pattern search

$\$$ ₁ a c a a c g ₁

a ₁ a c g \$ a c ₁

a ₂ c a a c g $\$$ ₁

a ₃ c g \$ a c a ₁

c ₁ a a c g \$ a ₂

c ₂ g \$ a c a a ₃

g ₁ \$ a c a a c ₂

a a c

Pattern search

$\$$ ₁	a	c	a	a	c	g	g ₁
a ₁	a	c	g	$\$$	a	c ₁	
a ₂	c	a	a	c	g	$\$$ ₁	
a ₃	c	g	$\$$	a	c	a ₁	
c ₁	a	a	c	g	$\$$	a ₂	
c ₂	g	$\$$	a	c	a	a ₃	
g ₁	$\$$	a	c	a	a	c ₂	

a a c



Pattern search

For how long does it work?

Pattern search

In first column find index of a letter with given rank

$\$$ ₁ a c a a c **g**₁

a₁ a c g \$ a **c**₁

a₂ c a a c g **\$**₁

a₃ c g \$ a c **a**₁

c₁ a a c g \$ **a**₂

c₂ g \$ a c a **a**₃

g₁ \$ a c a a **c**₂

Pattern search

In first column find index of a letter with given rank

→	\$₁	a	c	a	a	c	g₁
→	a₁	a	c	g	\$	a	c₁
	a₂	c	a	a	c	g	\$₁
	a₃	c	g	\$	a	c	a₁
→	c₁	a	a	c	g	\$	a₂
	c₂	g	\$	a	c	a	a₃
→	g₁	\$	a	c	a	a	c₂

Pattern search

In first column find index of a letter with given rank — $O(1)$

$\$$ ₁ a c a a c **g**₁

a₁ a c g \$ a **c**₁

a₂ c a a c g **\$**₁

a₃ c g \$ a c **a**₁

c₁ a a c g \$ **a**₂

c₂ g \$ a c a **a**₃

g₁ \$ a c a a **c**₂

Pattern search

In BWT find all given letters in specified range

$\$$ ₁ a c a a c **g**₁

a₁ a c g \$ a **c**₁

a₂ c a a c g **\$**₁

a₃ c g \$ a c **a**₁

c₁ a a c g \$ **a**₂

c₂ g \$ a c a **a**₃

g₁ \$ a c a a **c**₂

Pattern search

In BWT find a number of given letter above a certain row

$\$$ ₁ a c a a c **g**₁

a₁ a c g \$ a **c**₁

a₂ c a a c g **\$**₁

a₃ c g \$ a c **a**₁

c₁ a a c g \$ **a**₂

c₂ g \$ a c a **a**₃

g₁ \$ a c a a **c**₂

Pattern search

In BWT find a number of given letter above a certain row

$\$$ ₁	a	c	a	a	c	g	₁	0
a	a	c	g	\$	a	c	₁	0
a	c	a	a	c	g	\$	₁	0
a	c	g	\$	a	c	a	₁	1
c	a	a	c	g	\$	a	₂	2
c	g	\$	a	c	a	a	₃	3
g	\$	a	c	a	a	c	₂	3

Pattern search

In BWT find a number of given letter above a certain row

$\$$ ₁ a c a a c **g**₁ 0

a₁ a c g \$ a **c**₁ 0

a₂ c a a c g **\$**₁ 0

a₃ c g \$ a c **a**₁ 1

c₁ a a c g \$ **a**₂ 2

c₂ g \$ a c a **a**₃ 3

g₁ \$ a c a a **c**₂ 3

Pattern search

In BWT find a number of given letter above a certain row

$\$$ ₁	a	c	a	a	c	g	₁	0
a	a	c	g	\$	a	c	₁	0
a	c	a	a	c	g	\$	₁	0
a	c	g	\$	a	c	a	₁	1
c	a	a	c	g	\$	a	₂	2
c	g	\$	a	c	a	a	₃	3
g	\$	a	c	a	a	c	₂	3

Pattern search

In BWT find a number of given letter above a certain row

$\$$ ₁	a	c	a	a	c	g	₁	0
a	a	c	g	\$	a	c	₁	1
a	c	a	a	c	g	\$	₁	1
a	c	g	\$	a	c	a	₁	1
c	a	a	c	g	\$	a	₂	1
c	g	\$	a	c	a	a	₃	1
g	\$	a	c	a	a	c	₂	2

Pattern search

In BWT find a number of given letter above a certain row

$\$$ ₁	a	c	a	a	c	g	₁	0
a	a	c	g	\$	a	c	₁	1
a	c	a	a	c	g	\$	₁	1
a	c	g	\$	a	c	a	₁	1
c	a	a	c	g	\$	a	₂	1
c	g	\$	a	c	a	a	₃	1
g	\$	a	c	a	a	c	₂	2

Pattern search

In BWT find a number of given letter above a certain row

$\$$ ₁	a	c	a	a	c	g	₁	0
a	a	c	g	\$	a	c	₁	1
a	c	a	a	c	g	\$	₁	1
a	c	g	\$	a	c	a	₁	1
c	a	a	c	g	\$	a	₂	1
c	g	\$	a	c	a	a	₃	1
g	\$	a	c	a	a	c	₂	2

Pattern search

In BWT find a number of given letter above a certain row — $O(1)$

$\$$ ₁ a c a a c **g**₁

a₁ a c g \$ a **c**₁

a₂ c a a c g **\$**₁

a₃ c g \$ a c **a**₁

c₁ a a c g \$ **a**₂

c₂ g \$ a c a **a**₃

g₁ \$ a c a a **c**₂

Pattern search

In BWT find all given letters in specified range — $O(1)$

$\$$ ₁ a c a a c **g**₁

a₁ a c g \$ a **c**₁

a₂ c a a c g **\$**₁

a₃ c g \$ a c **a**₁

c₁ a a c g \$ **a**₂

c₂ g \$ a c a **a**₃

g₁ \$ a c a a **c**₂

Pattern search

Pattern search works in linear time

Thank you!

Questions?