Sequence alignment

Eva Stukenbrock & Julien Dutheil

eva.stukenbrock@mpi-marburg.mpg.de

julien.dutheil@mpi-marburg.mpg.de

Max Planck Institute for Terrestrial Microbiology – Marburg

19 February 2013

Sequence similarity

The concept of sequence similarity is used for:

- Predicting gene functions, by “homology”
- Find particular motives in a sequence
- Find particular sequences in a genome / collection
- Classify sequences into families
- Compute evolutionary distances and build phylogenies
- Assemble genomes
- *etc.*

An old problem

Definition: Edit distance

$\Rightarrow$ *a.k.a* Levenshtein distance • Comes for Information Theory • Is defined as:

"The minimal number of operations to turn a string into another"

An old problem

Definition: Edit distance

$\Rightarrow$ *a.k.a* Levenshtein distance • Comes for Information Theory • Is defined as:

"The minimal number of operations to turn a string into another"

where "operations" can be:

- A substitution of one letter with another
- An insertion of one or several letters
- A deletion of one or several letters

Application (to biology)

- Computing the edit distance between two sequences is equivalent to computing their alignment:

```
-I--NTERESTINGLY
<.<<.**.**..**>>
BIOINFORMATICS--
```

  ☞ 3 deletions (<), 5 substitutions (.) and 2 insertions (>).

Application (to biology)

- Computing the edit distance between two sequences is equivalent to computing their alignment:

```
-I--NTERESTINGLY
<.<<.**.**..**>>
BIOINFORMATICS--
```

  ☞ 3 deletions (<), 5 substitutions (.) and 2 insertions (>).

- We need to give weights to each operation to obtain a numerical value

Application (to biology)

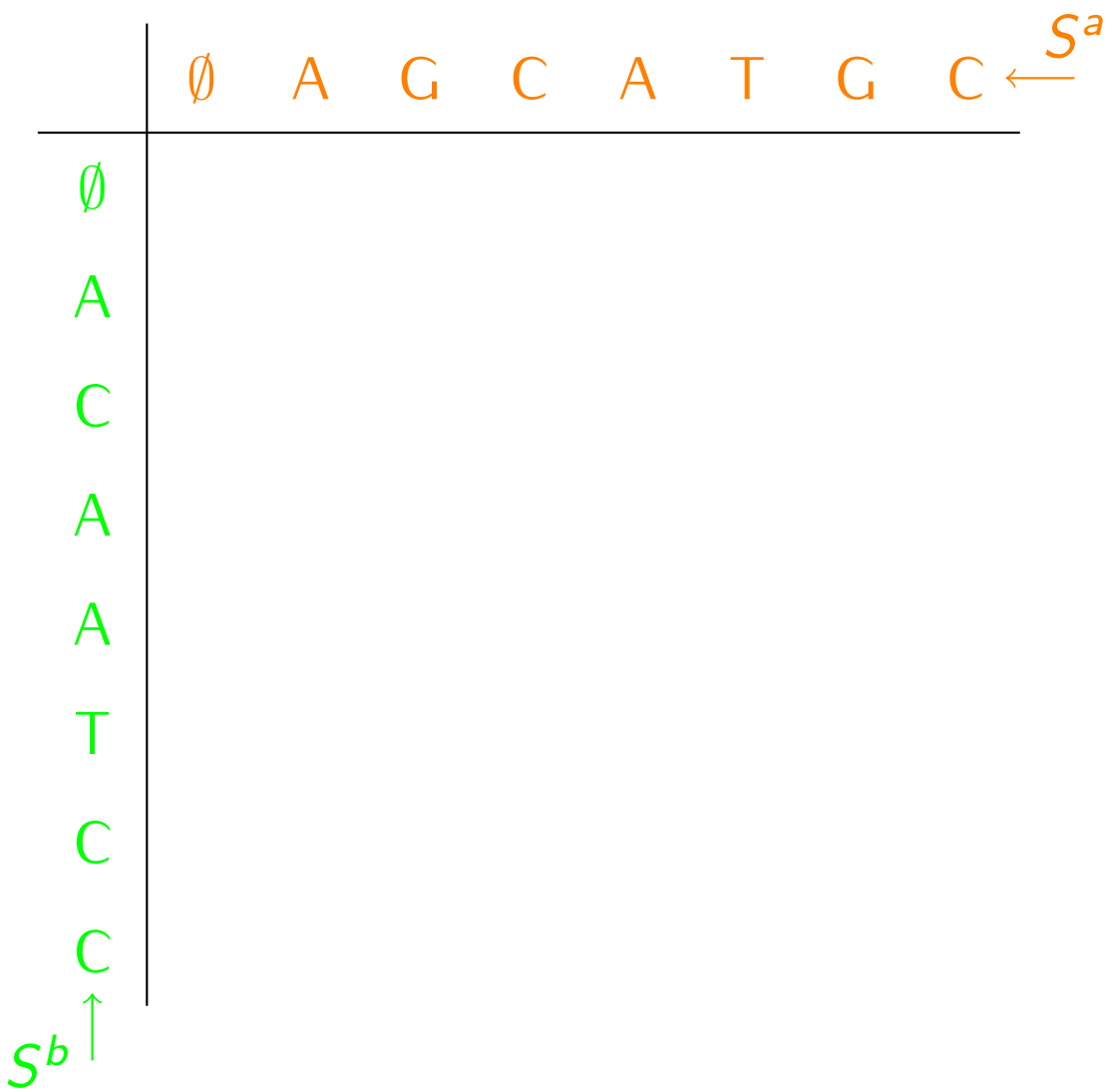
- Computing the edit distance between two sequences is equivalent to computing their alignment:

```
-I--NTERESTINGLY
<.<<.**.**..**>>
BIOINFORMATICS--
```

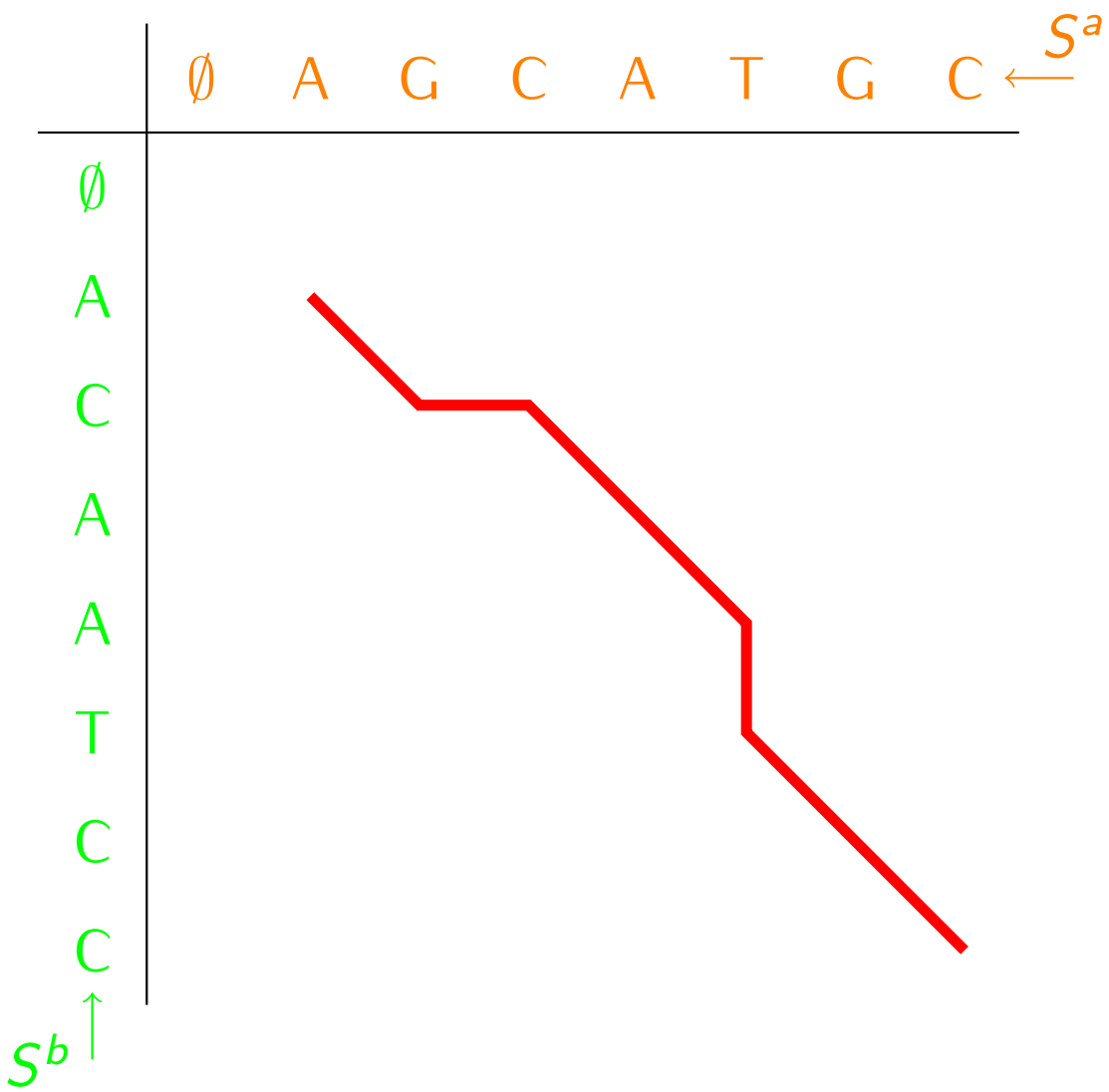
  ☞ 3 deletions (<), 5 substitutions (.) and 2 insertions (>).

- We need to give weights to each operation to obtain a numerical value
- These operations have biological interpretation, they correspond to distinct mutational events

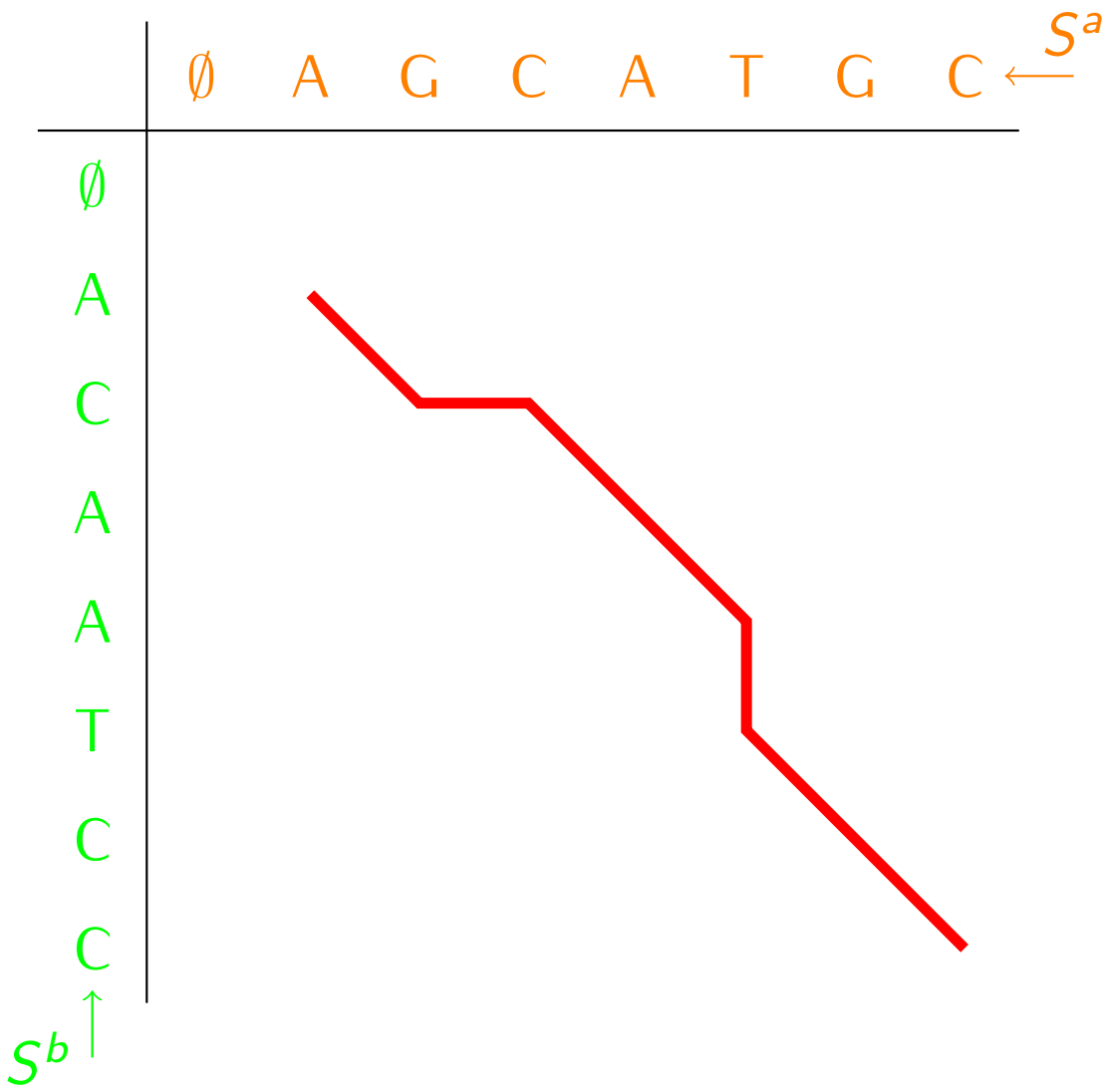
Another way to represent alignments



Another way to represent alignments

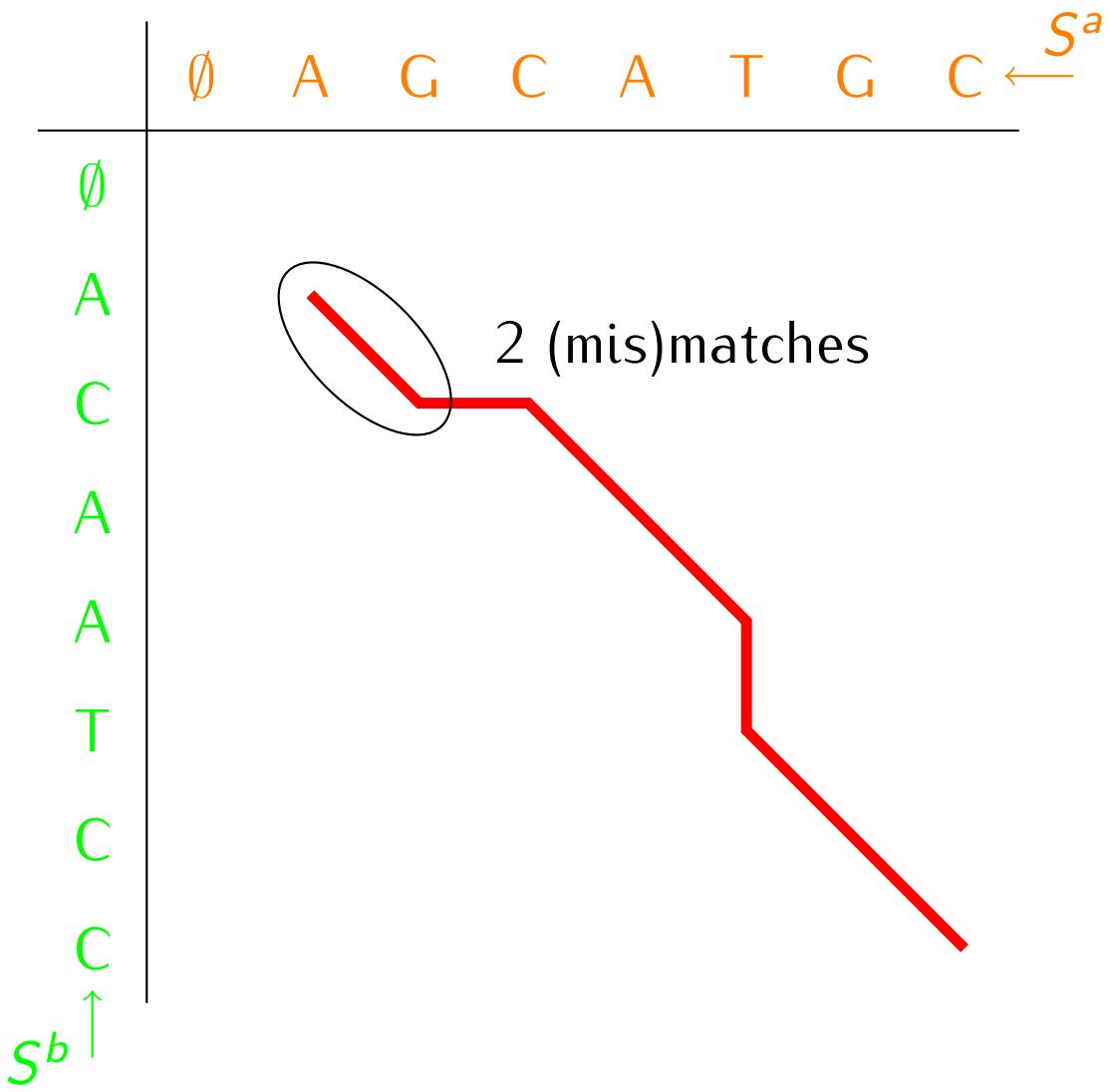


Another way to represent alignments



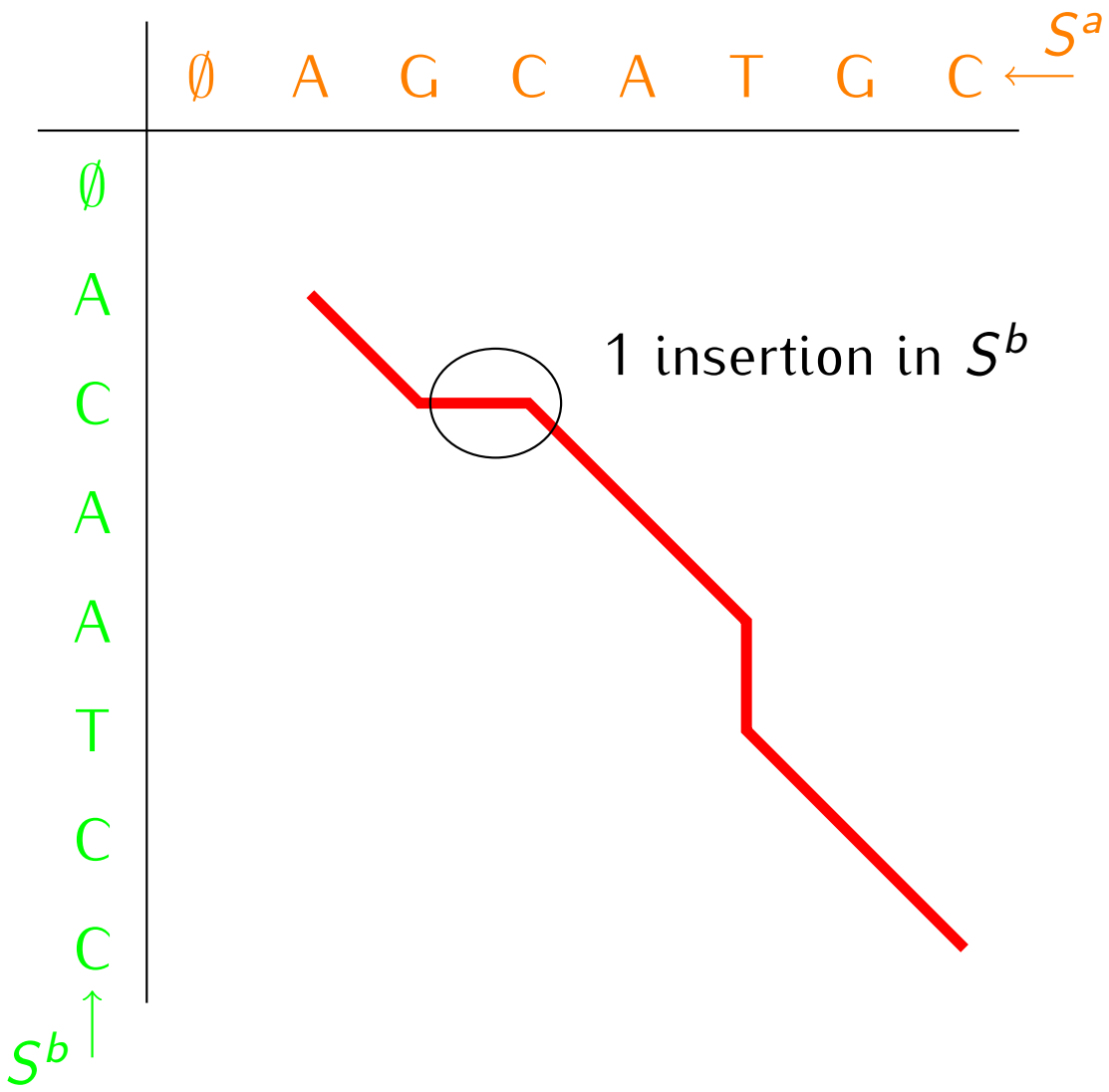
S^a = AGCAT-GC
S^b = AC-AATCC

Another way to represent alignments



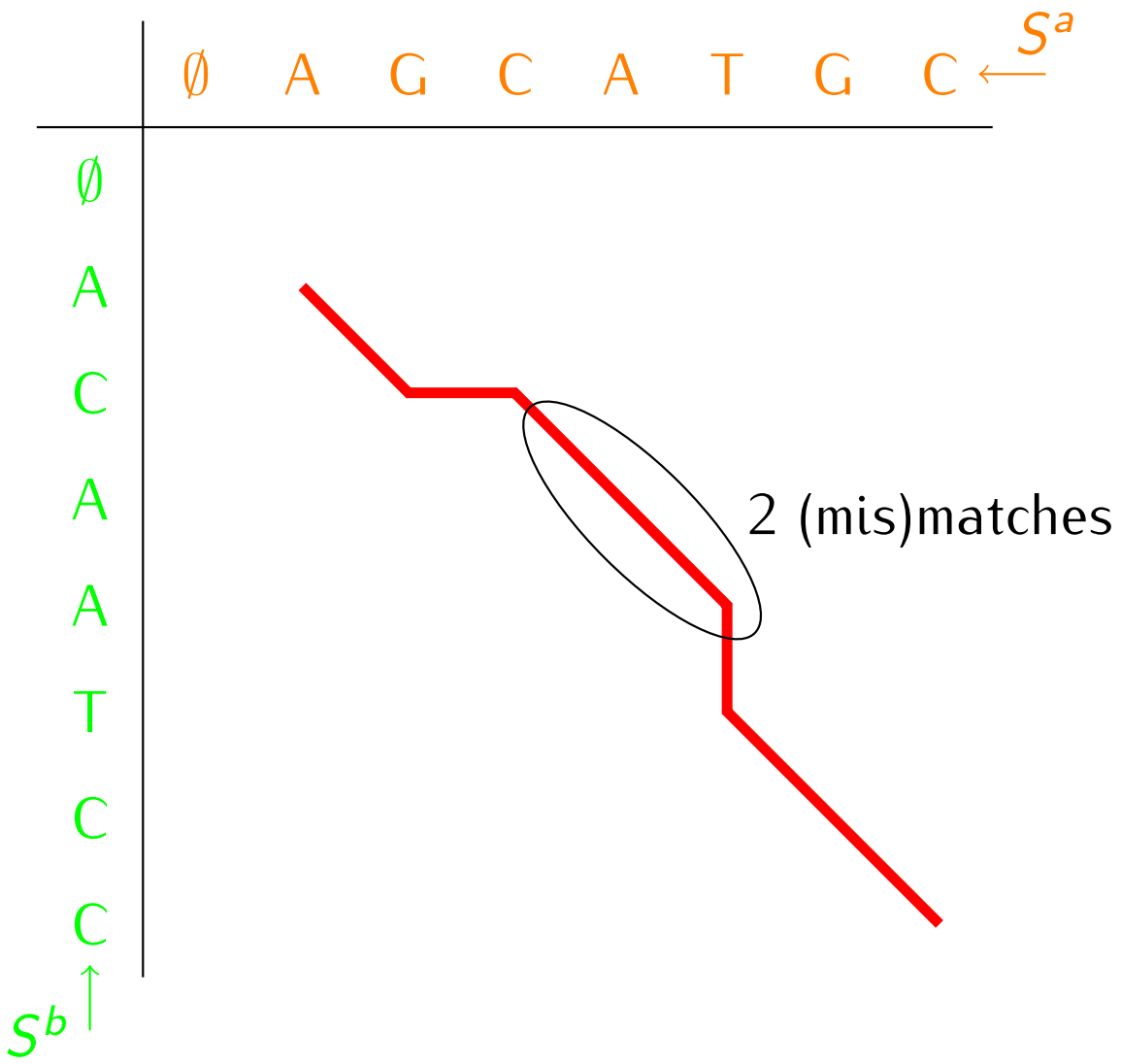
S^a = AGCAT-GC
S^b = AC-AATCC

Another way to represent alignments



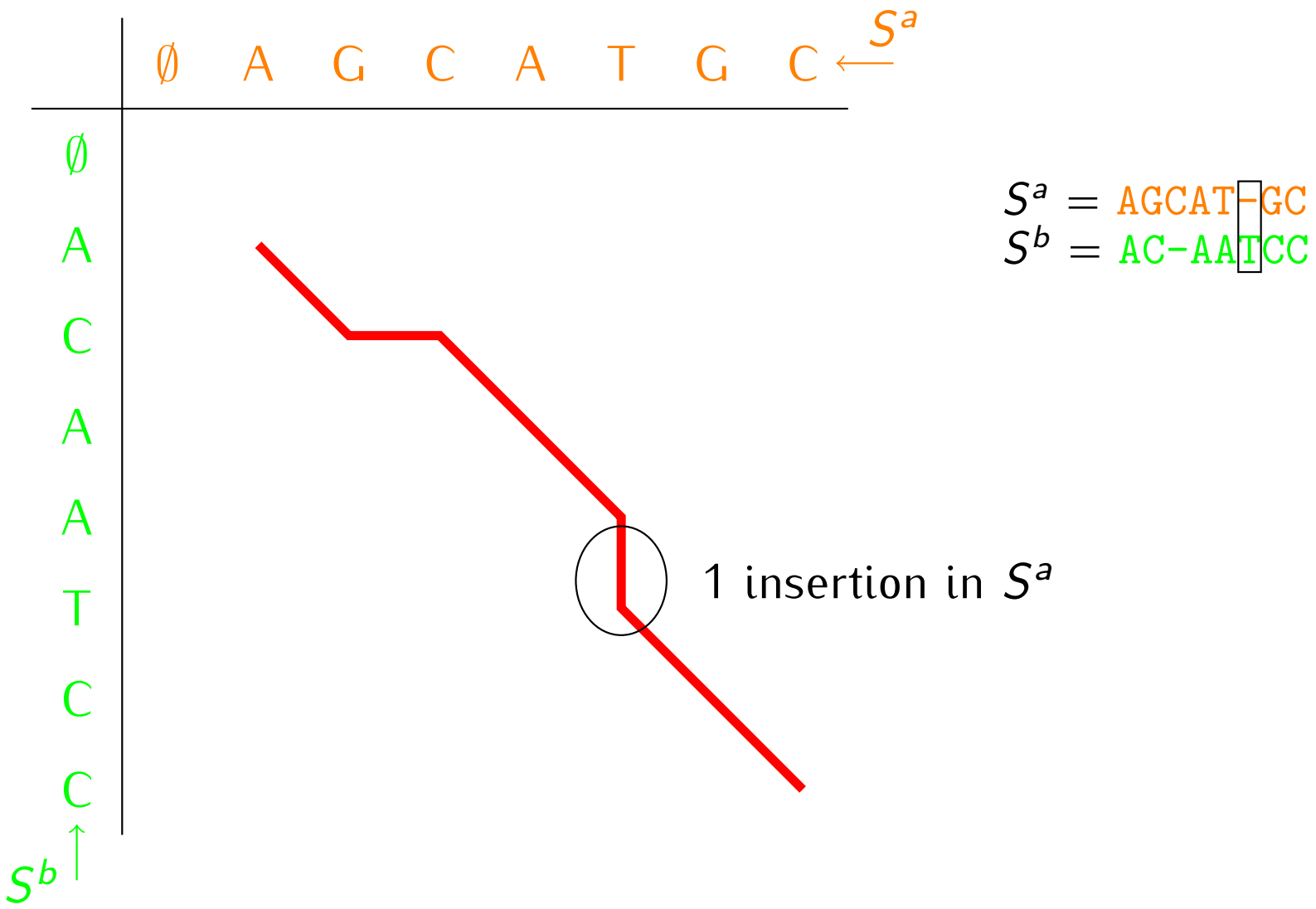
S^a = AGCAT-GC
S^b = AC-AATCC

Another way to represent alignments

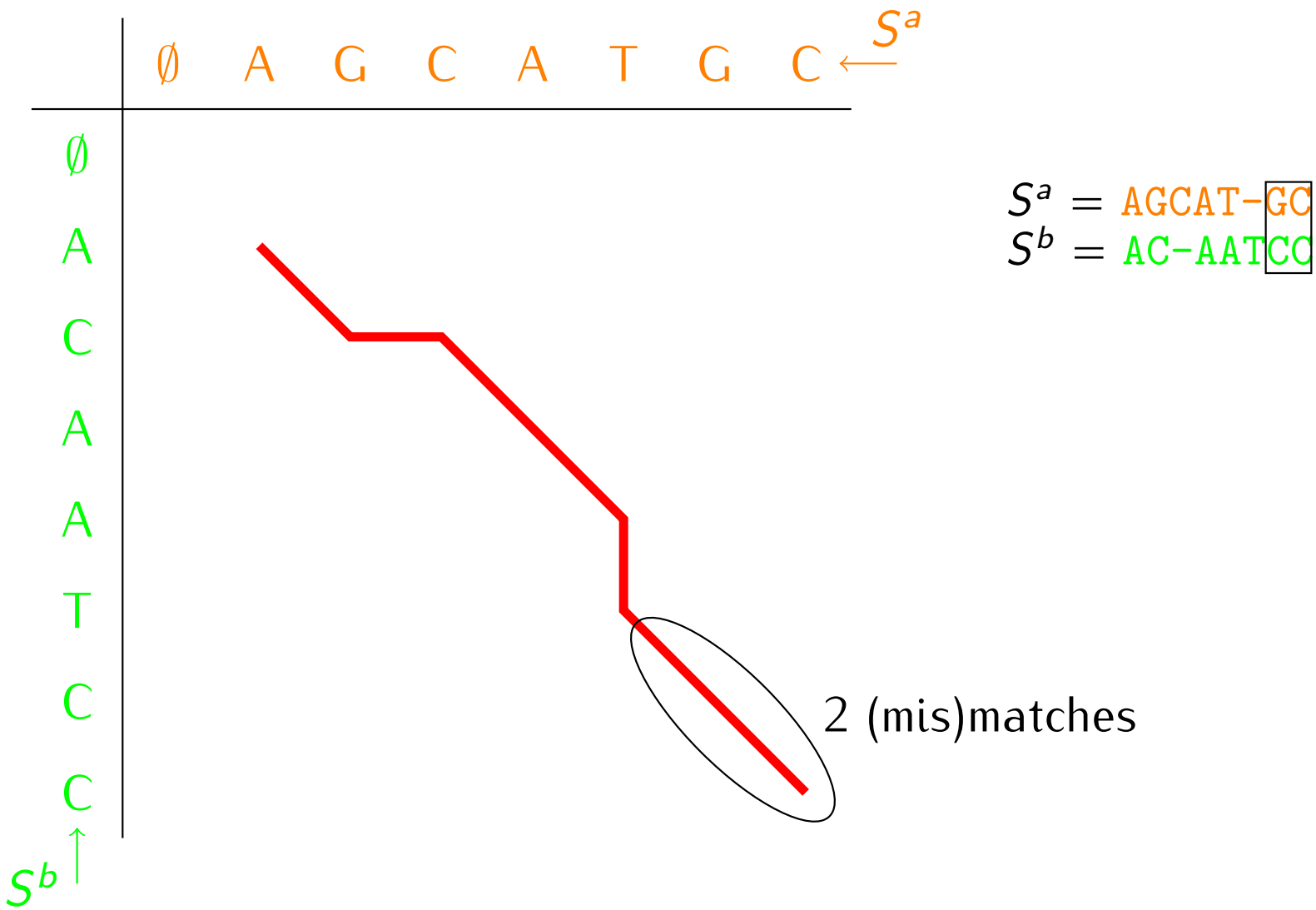


S^a = AGCAT-GC
S^b = AC-AATCC

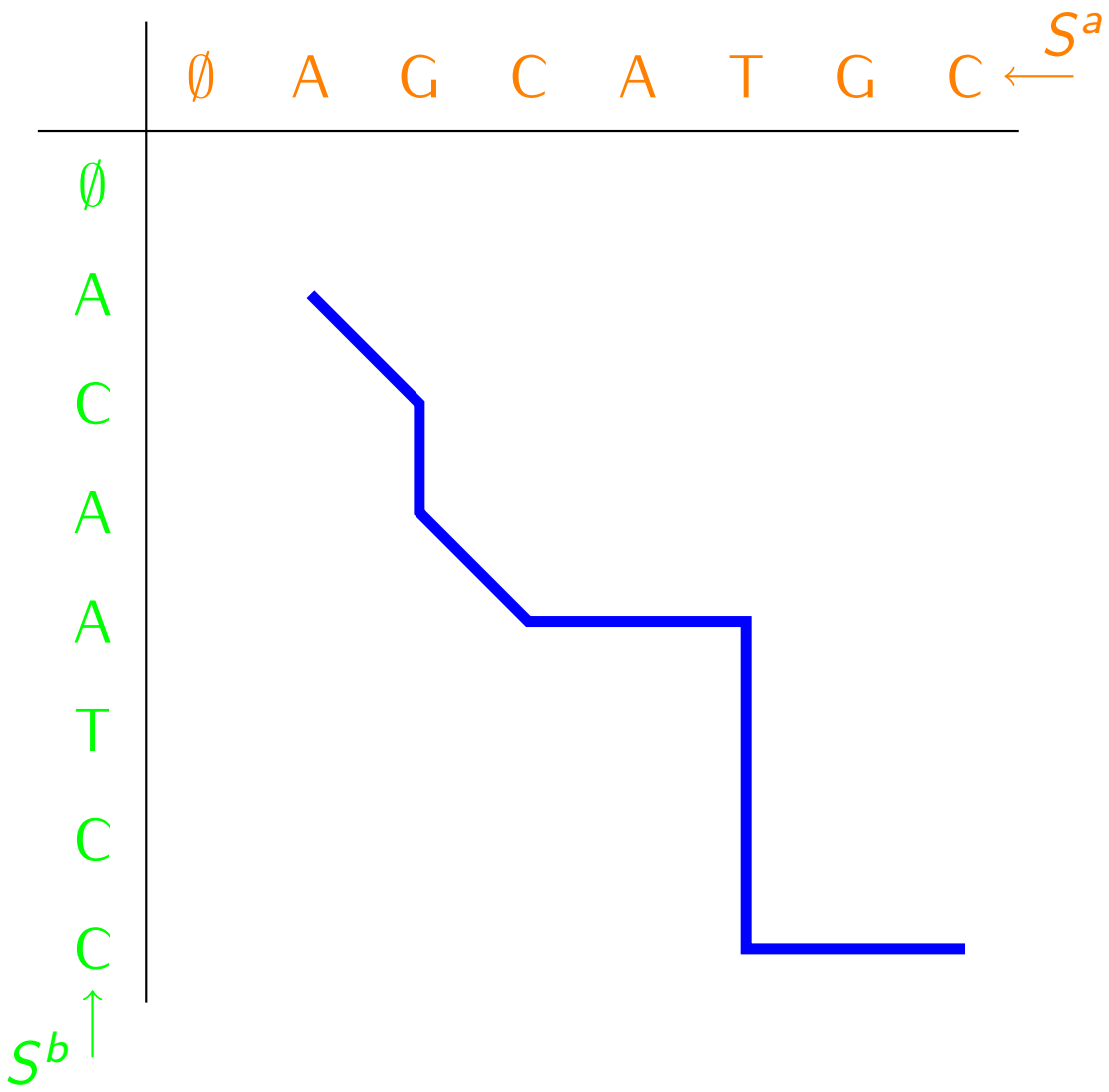
Another way to represent alignments



Another way to represent alignments

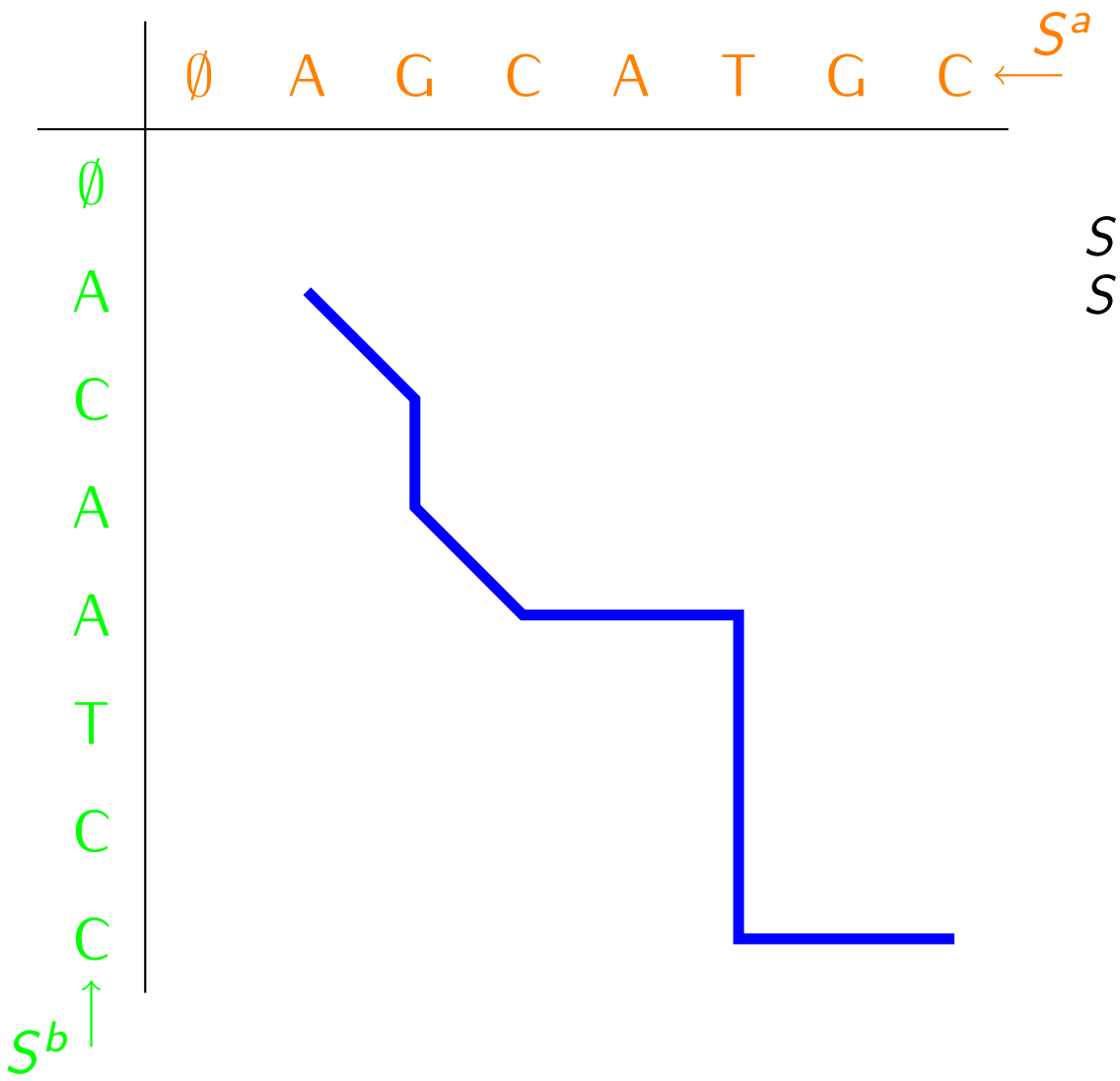


Another way to represent alignments



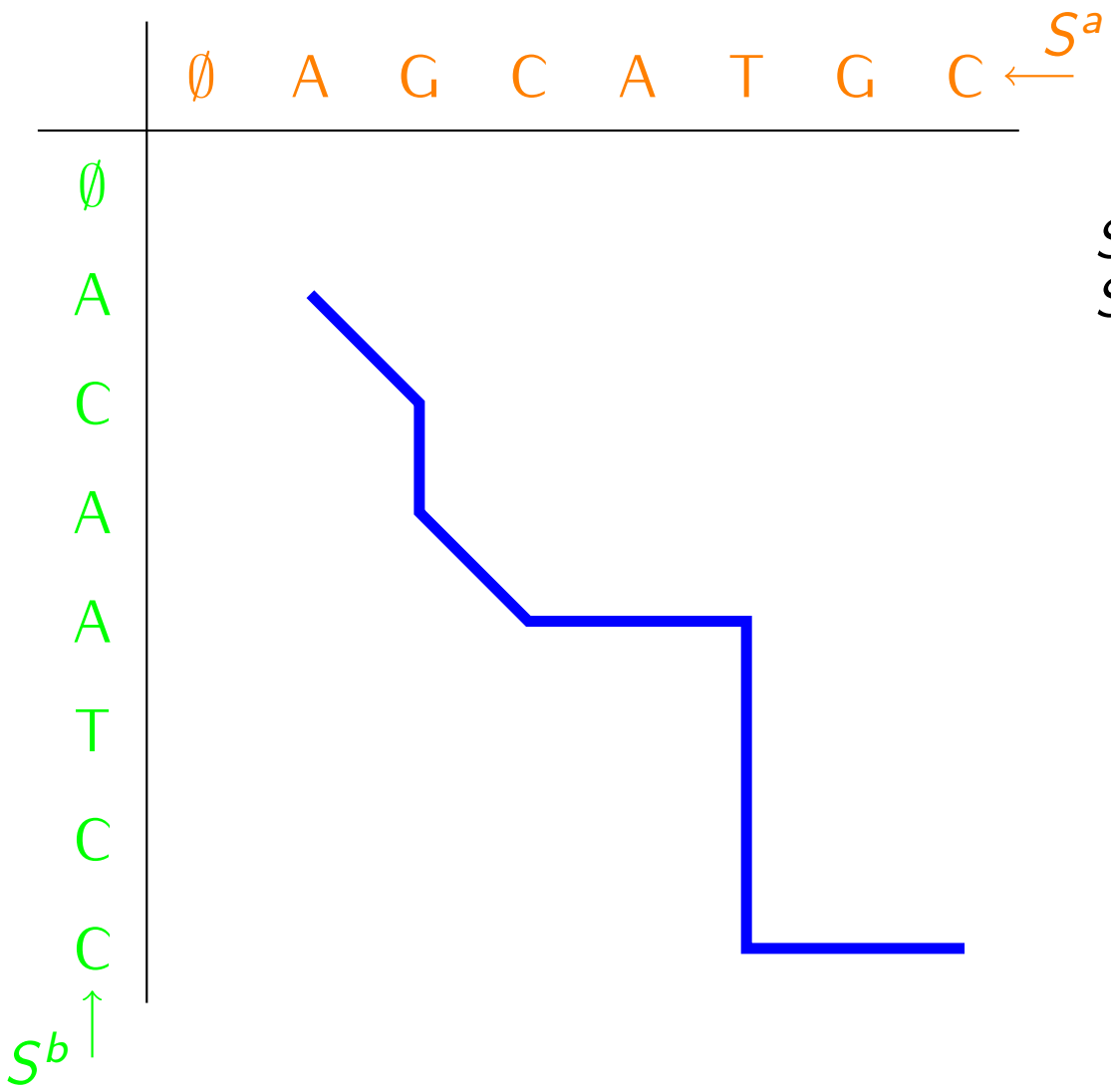
What is this alignment?

Another way to represent alignments



S^a = AG-CAT---GC
S^b = ACAA--TCC--

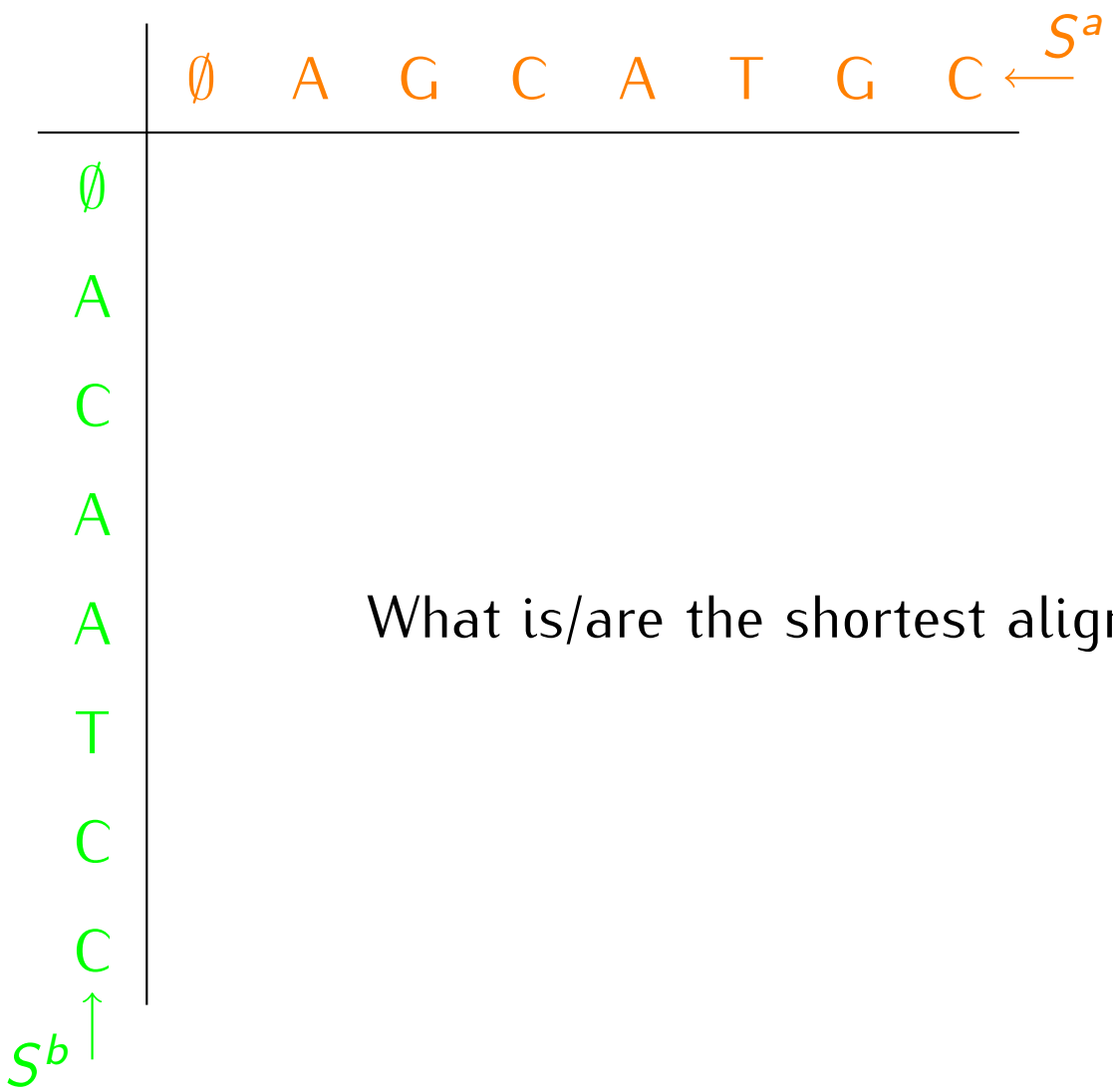
Another way to represent alignments



Alignment length: 11 sites

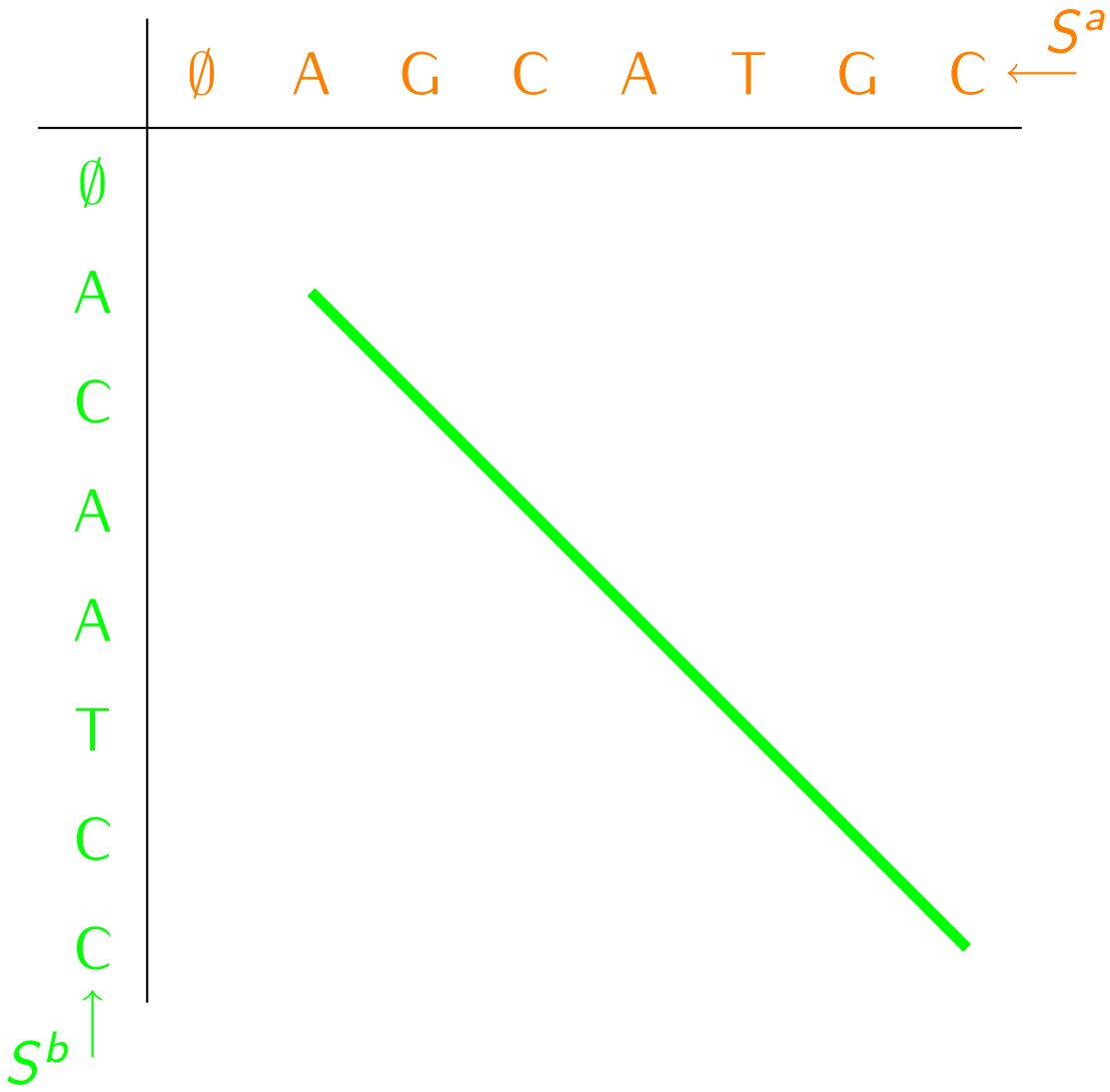
S^a = AG-CAT---GC
S^b = ACAA--TCC--

Another way to represent alignments



What is/are the shortest alignment(s)?

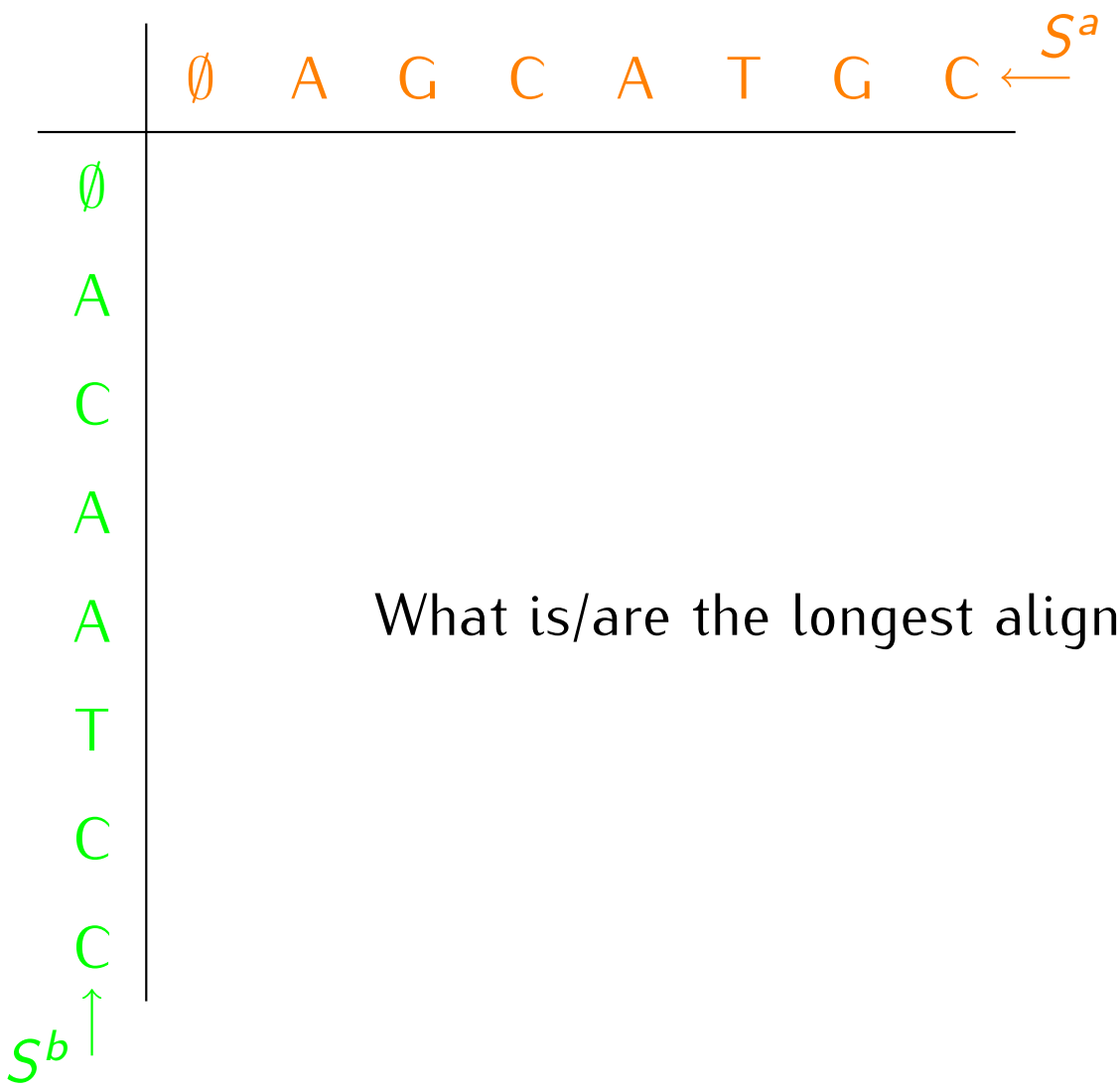
Another way to represent alignments



Alignment length: 7 sites

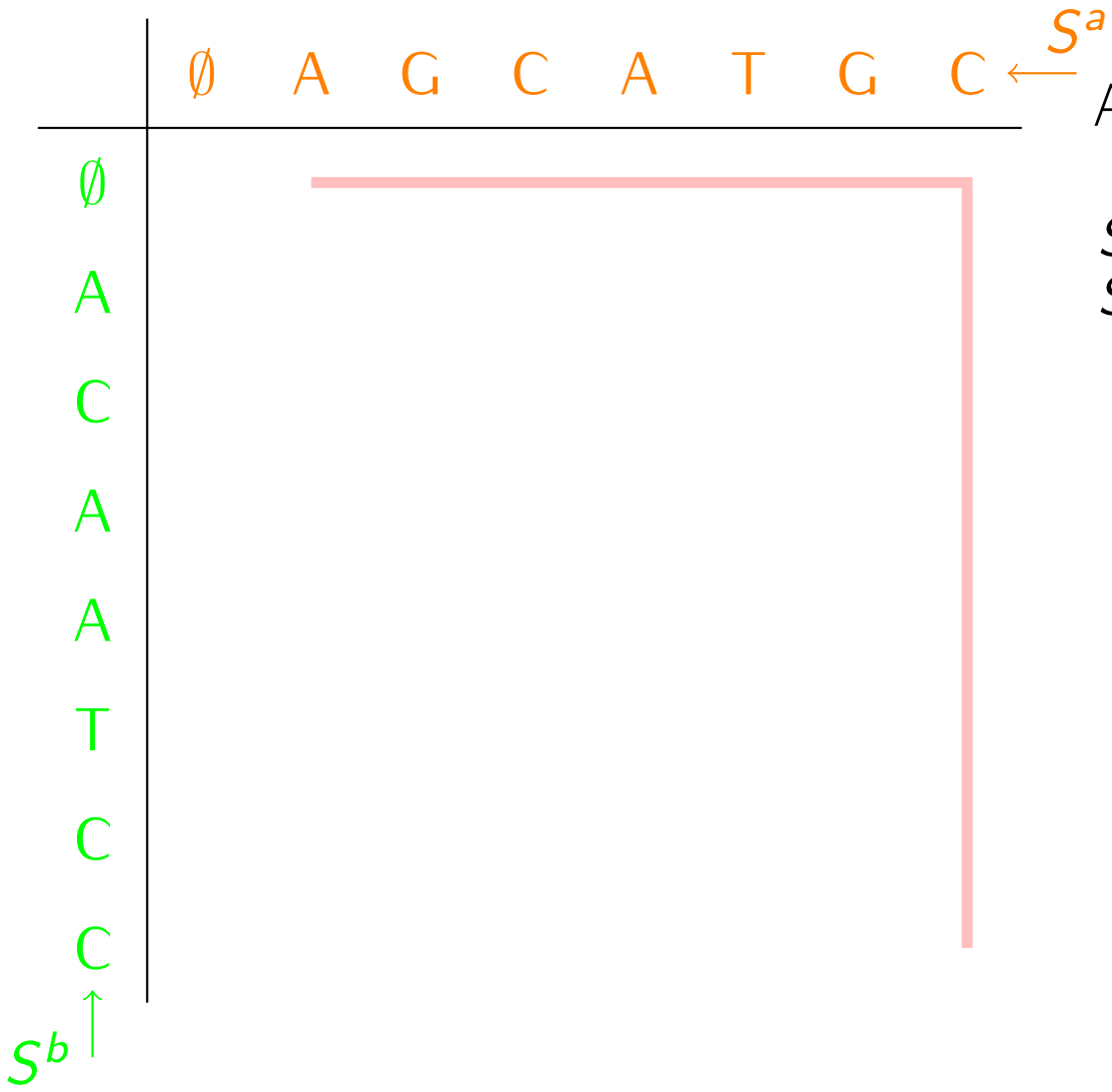
S^a = AGCATGC
S^b = ACAATCC

Another way to represent alignments



What is/are the longest alignment(s)?

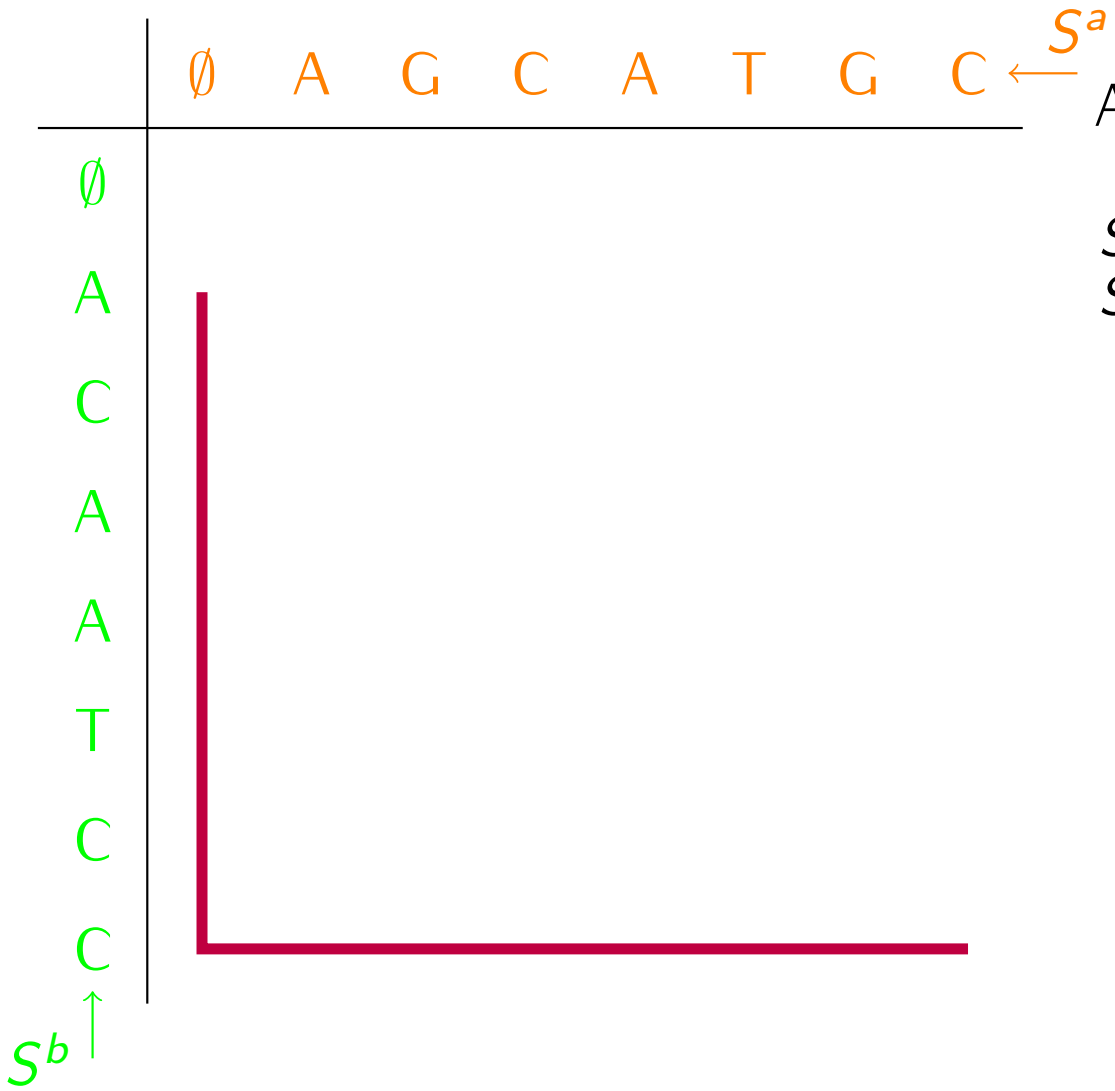
Another way to represent alignments



Alignment length: 14 sites

S^a = AGCATGC-------
S^b = -------ACAATCC

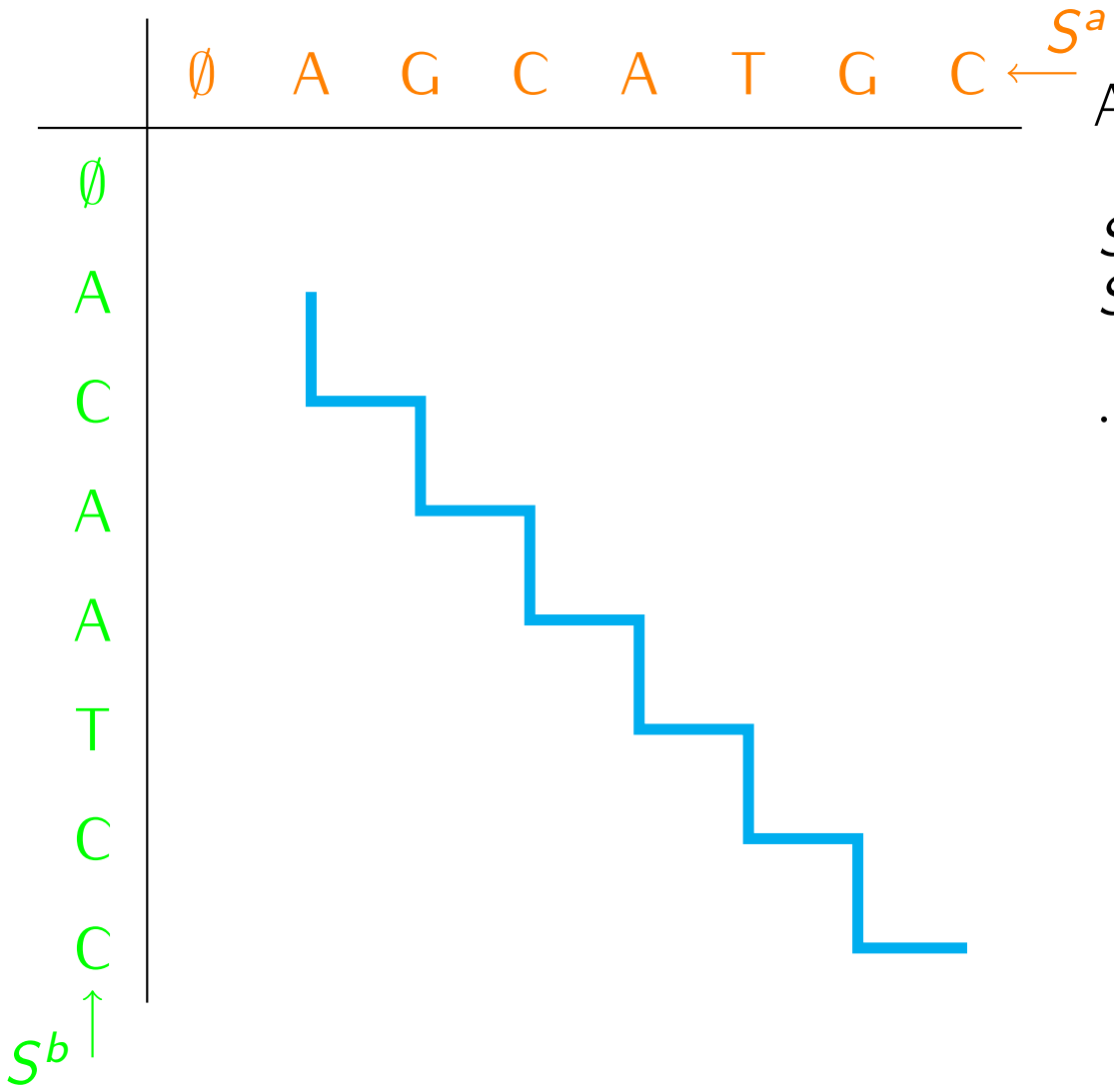
Another way to represent alignments



Alignment length: 14 sites

S^a = -------AGCATGC
S^b = ACAATCC-------

Another way to represent alignments



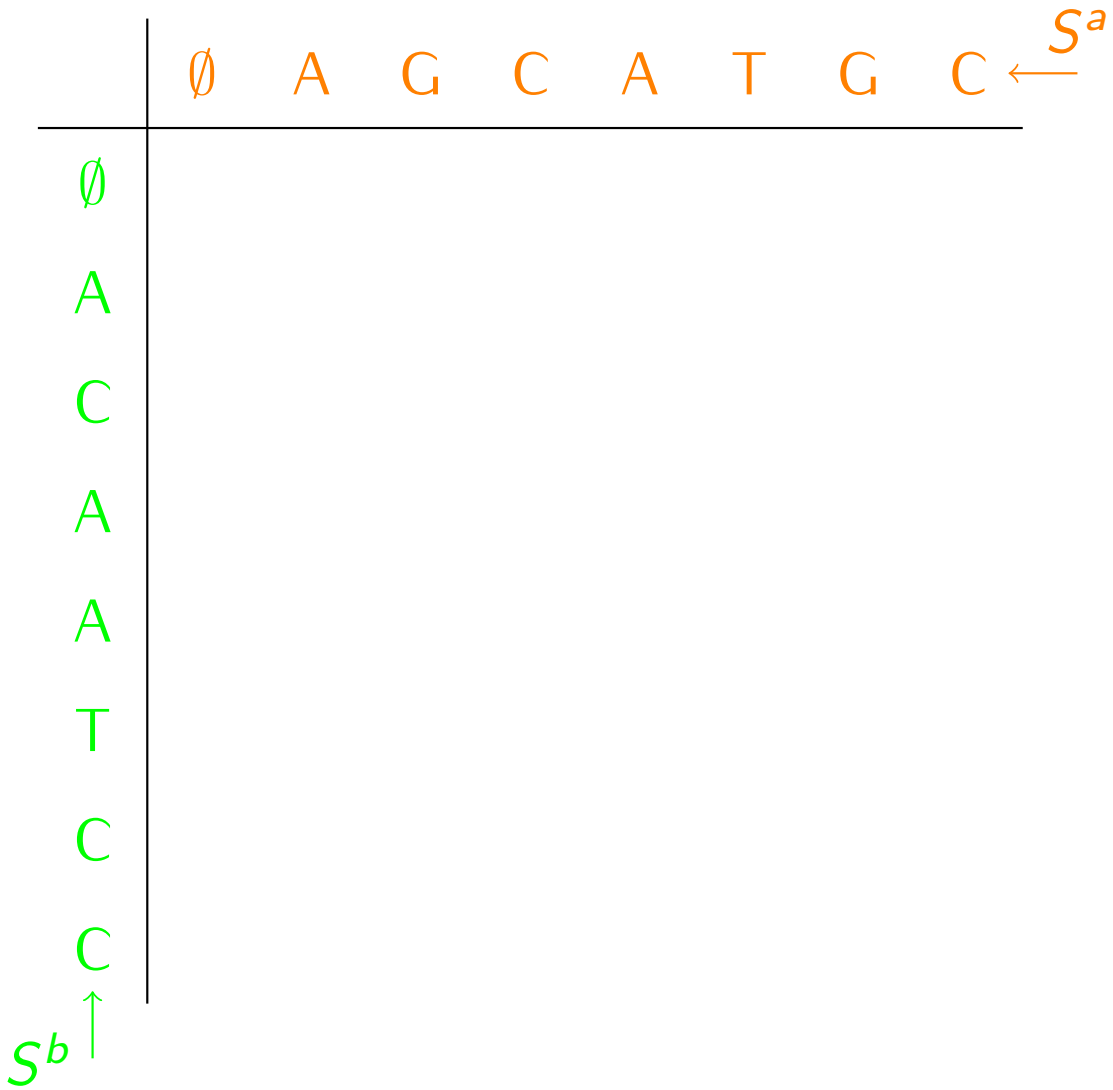
Alignment length: 14 sites

S^a = -A-G-C-A-T-G-C
S^b = A-C-A-A-T-C-C-

...and all intermediates!

What is the best alignment?

Part1: alignment score

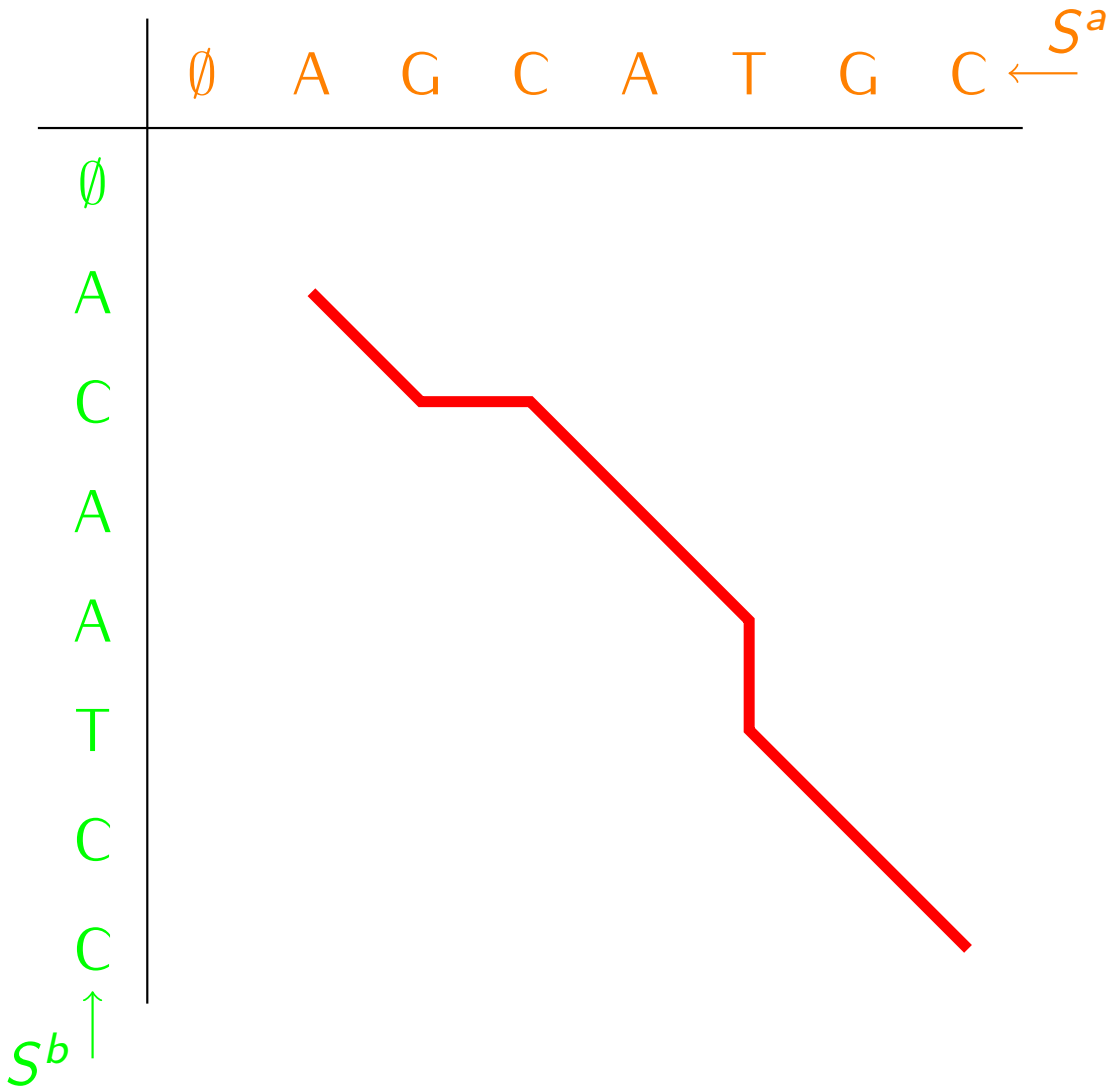


Scores:

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

What is the best alignment?

Part1: alignment score



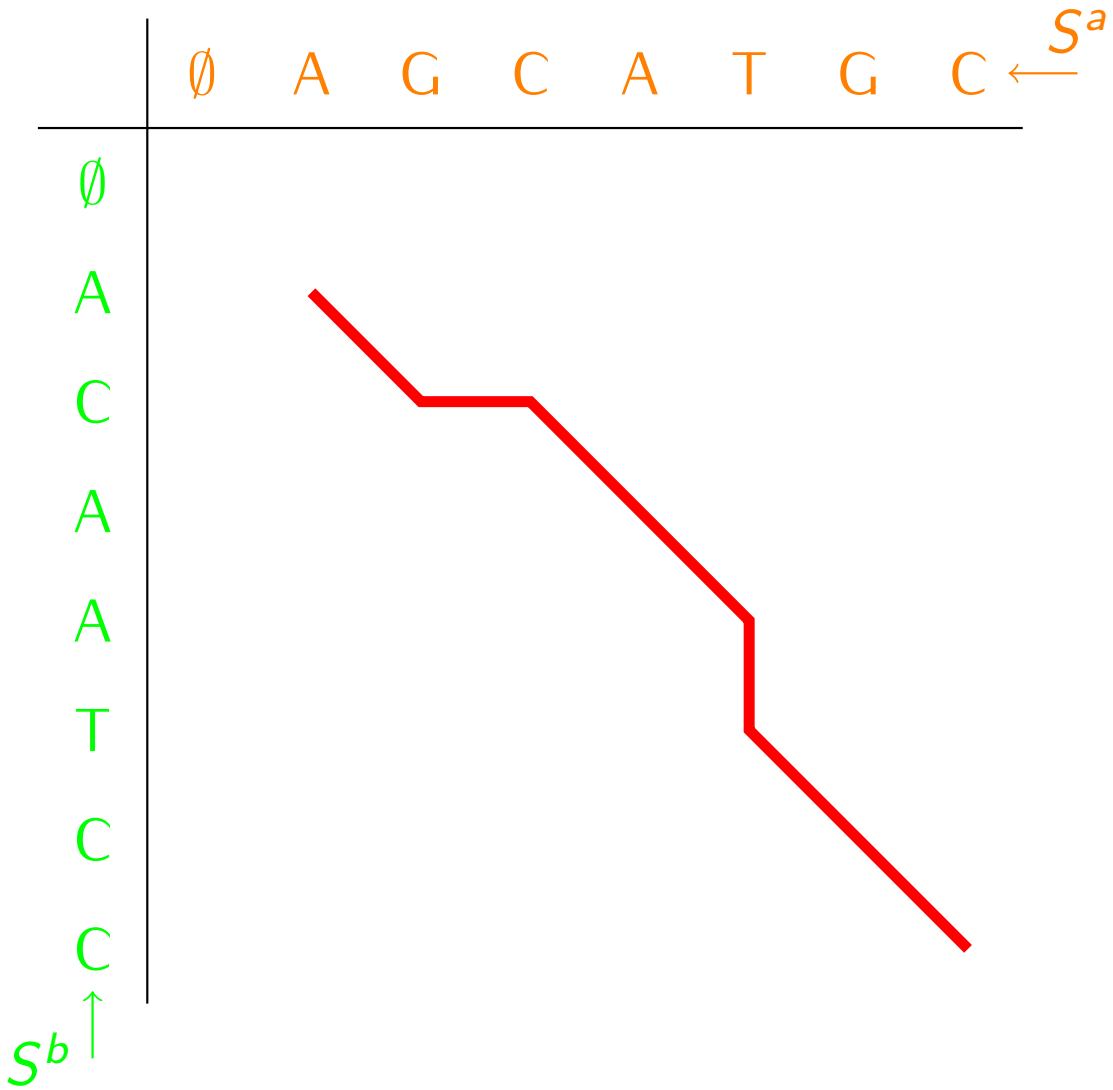
Scores:

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

$S^a = \mathtt{A\ G\ C\ A\ T\ -\ G\ C}$
$S^b = \mathtt{A\ C\ -\ A\ A\ T\ C\ C}$

What is the best alignment?

Part1: alignment score



Scores:

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

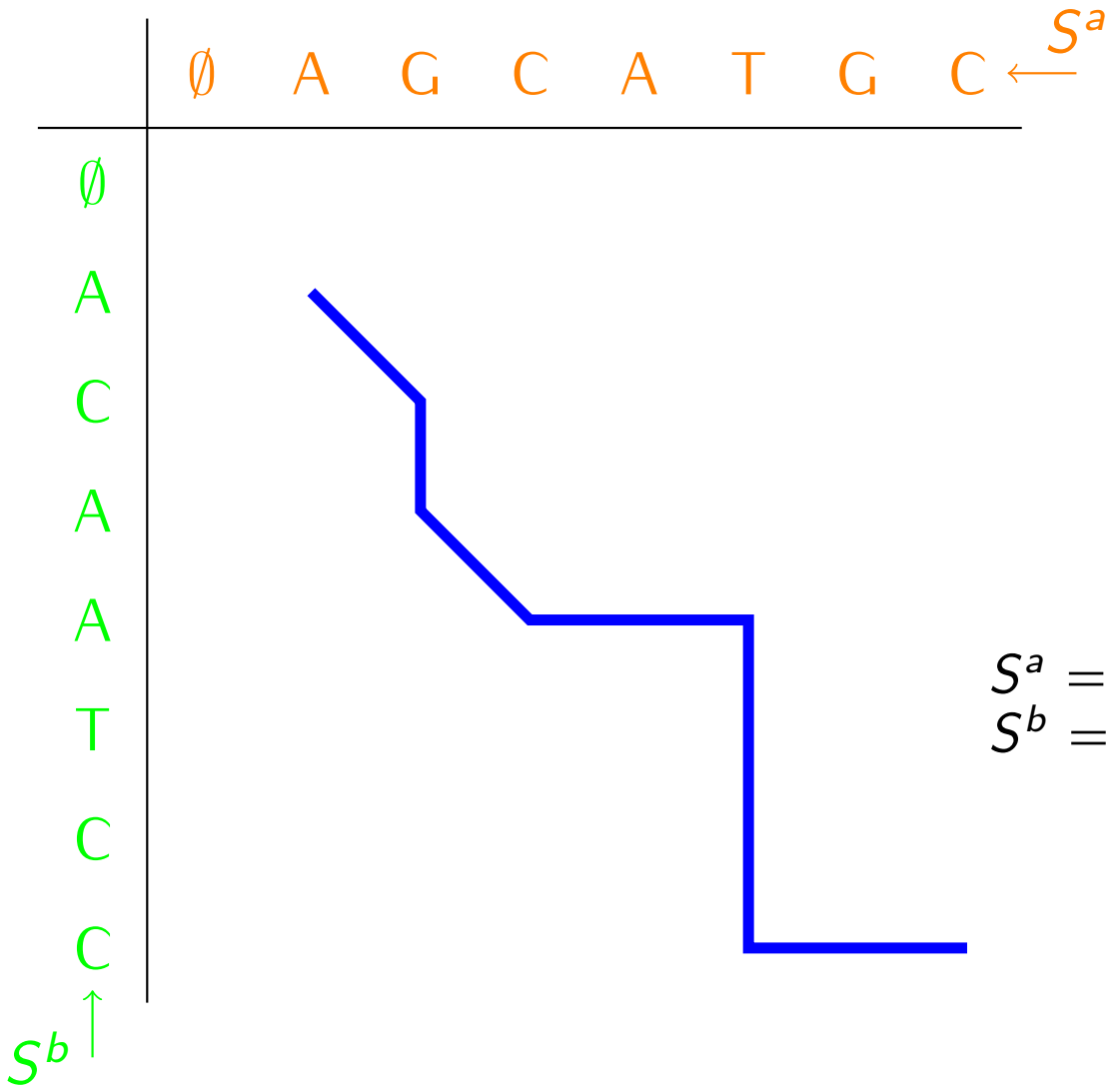
S^a = A G C A T - G C
S^b = A C - A A T C C

+2−1−1+2−1−1−1+2

Score = 1

What is the best alignment?

Part1: alignment score



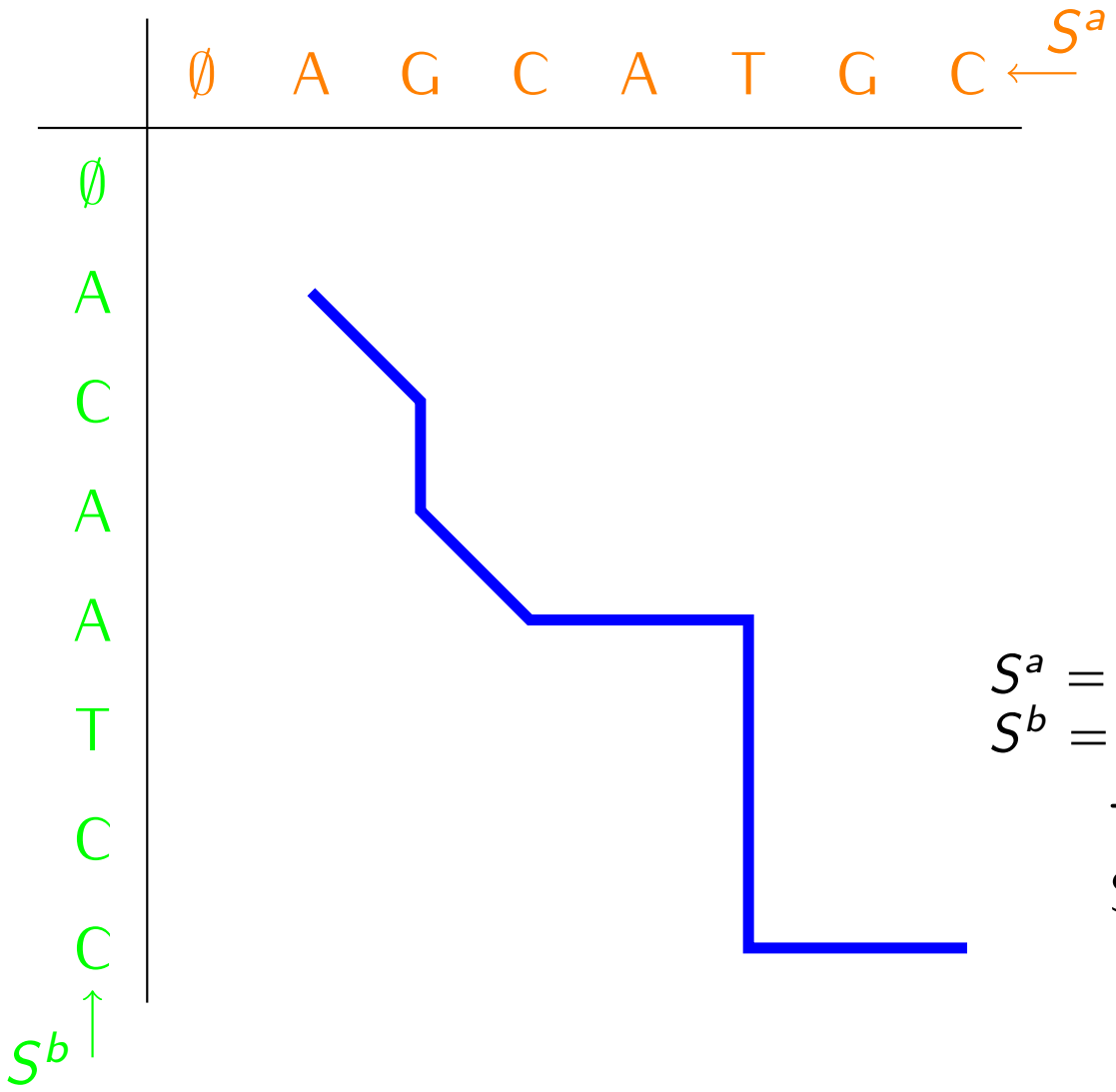
Scores:

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

S^a = A G - C A T - - - G C
S^b = A C A A - - T C C - -

What is the best alignment?

Part1: alignment score



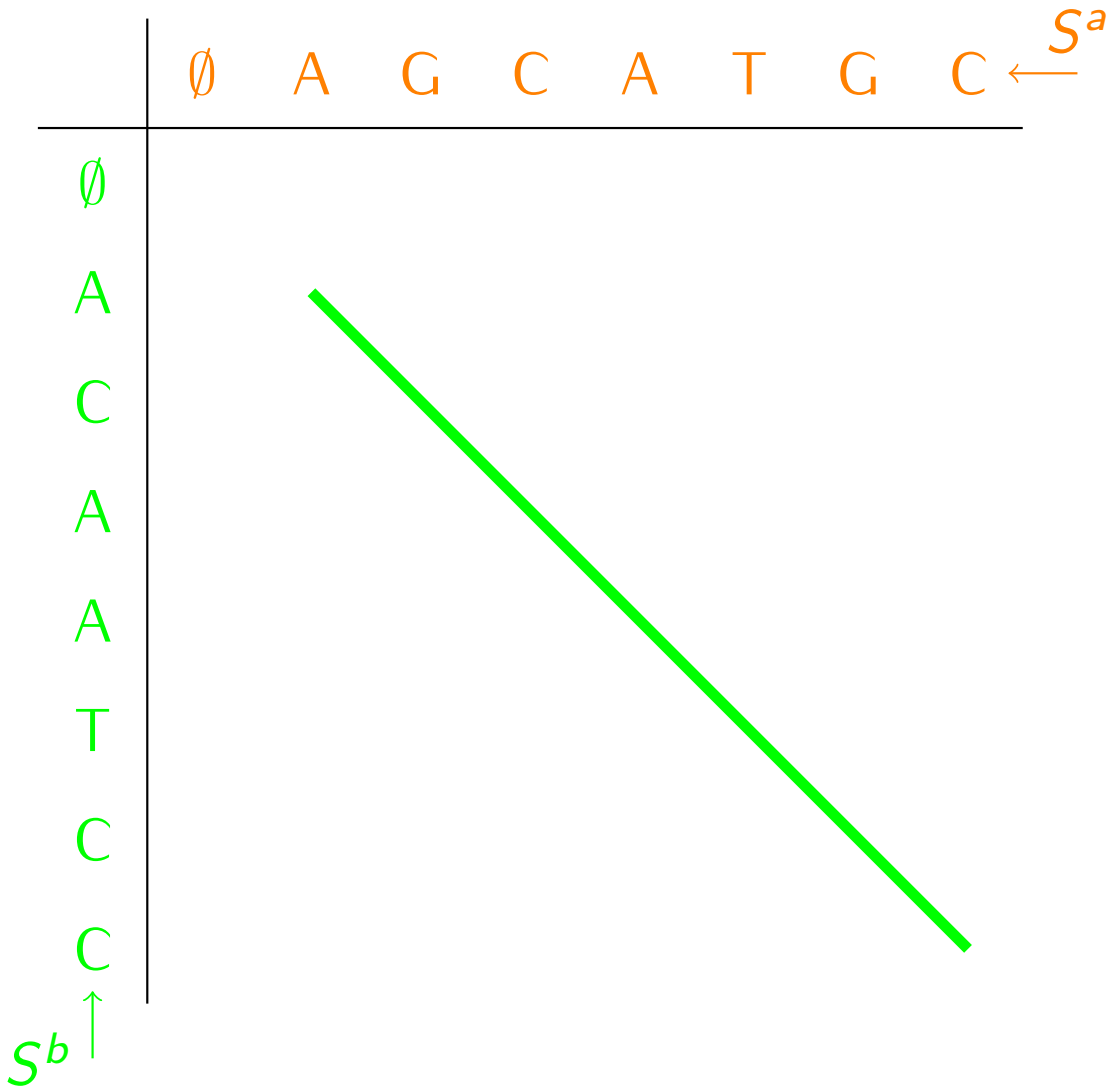
Scores:

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

S^a = A G - C A T - - - G C
S^b = A C A A - - T C C - -
+2-1-1-1-1-1-1-1-1-1-1

Score = -9

What is the best alignment?
Part1: alignment score

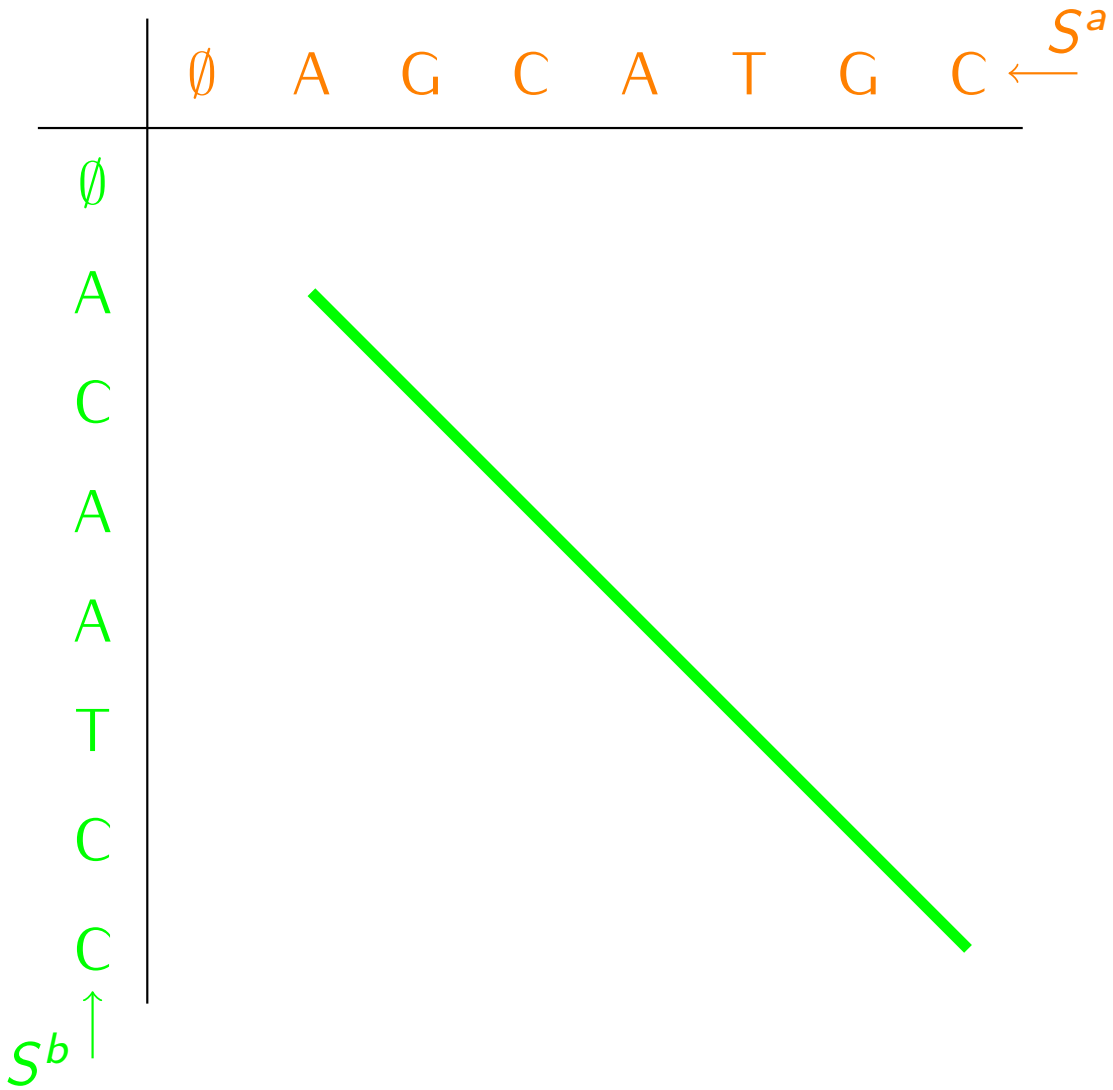


Scores:

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

$S^a = \mathtt{A\ G\ C\ A\ T\ G\ C}$
$S^b = \mathtt{A\ C\ A\ A\ T\ C\ C}$

What is the best alignment?
Part1: alignment score



Scores:

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

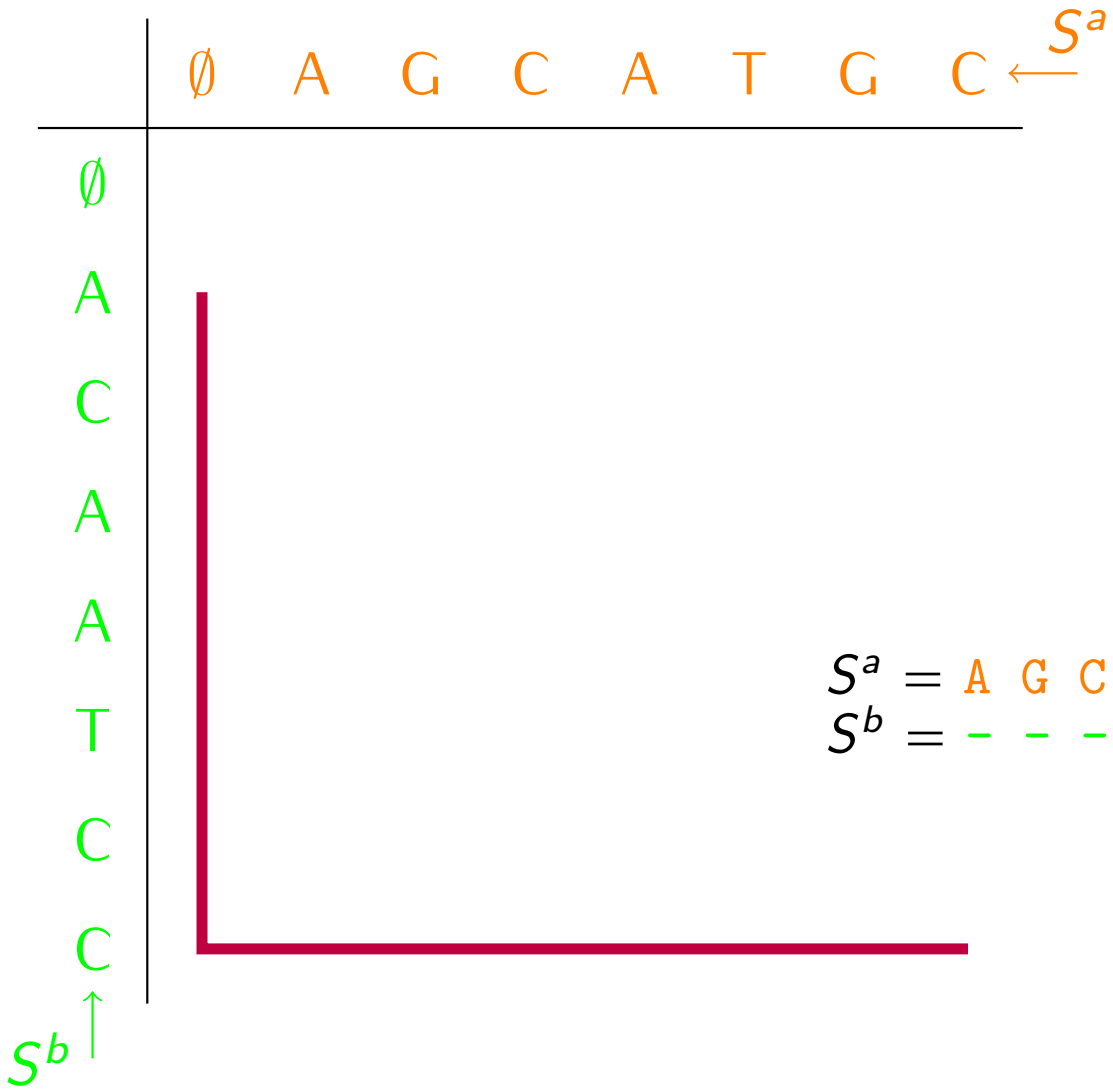
$S^a = \mathtt{A\ G\ C\ A\ T\ G\ C}$
$S^b = \mathtt{A\ C\ A\ A\ T\ C\ C}$

+2−1−1+2+2−1+2

Score = 5

What is the best alignment?

Part1: alignment score



Scores:

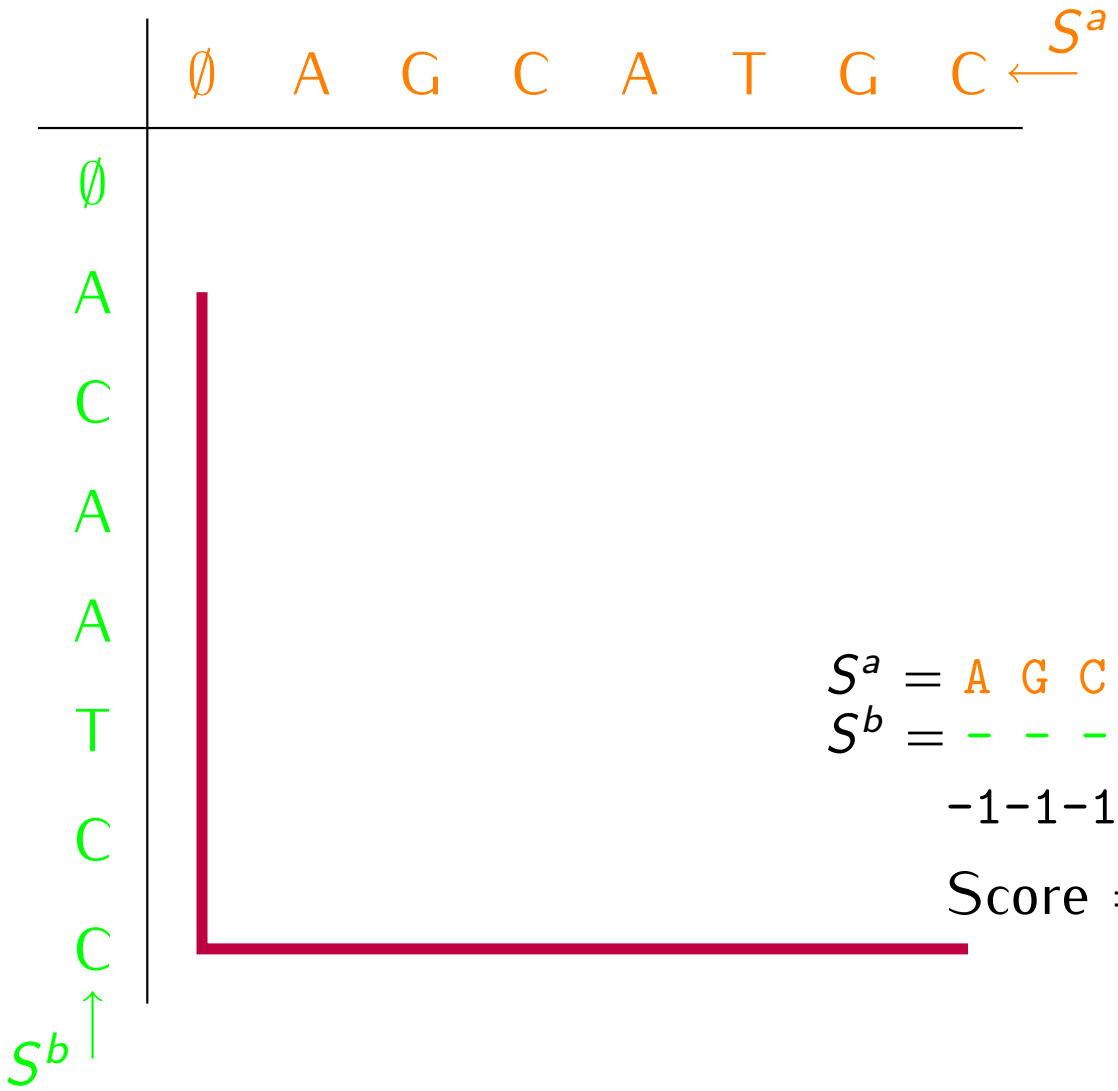
	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

$S^a =$ A G C A T G C - - - - - - - -

$S^b =$ - - - - - - - - A C A A T C C

What is the best alignment?

Part1: alignment score



Scores:

	-	A	C	G	T
-		-1	-1	-1	-1
A	-1	2	-1	-1	-1
C	-1	-1	2	-1	-1
G	-1	-1	-1	2	-1
T	-1	-1	-1	-1	2

S^a = A G C A T G C - - - - - - - -

S^b = - - - - - - - - A C A A T C C

-1-1-1-1-1-1-1-1-1-1-1-1-1-1

Score = -14

What is the best alignment?
Part2: How to find the best score

The brute force

- Define scores for insertions, deletions and substitutions

What is the best alignment?
Part2: How to find the best score

The brute force

- Define scores for insertions, deletions and substitutions
- Generate a possible alignment of the two sequences

What is the best alignment?
Part2: How to find the best score

The brute force

- Define scores for insertions, deletions and substitutions
- Generate a possible alignment of the two sequences
- Computes the edit distance for this alignment, and record it

What is the best alignment?
Part2: How to find the best score

The brute force

- Define scores for insertions, deletions and substitutions
- Generate a possible alignment of the two sequences
- Computes the edit distance for this alignment, and record it
- Generate another alignment, and compute its edit distance, *etc*

What is the best alignment?
Part2: How to find the best score

The brute force

- Define scores for insertions, deletions and substitutions
- Generate a possible alignment of the two sequences
- Computes the edit distance for this alignment, and record it
- Generate another alignment, and compute its edit distance, *etc*
- Do that for all possible alignments, and keep the one with the highest score (=minimal edit distance)!

What is the best alignment?
Part2: How to find the best score

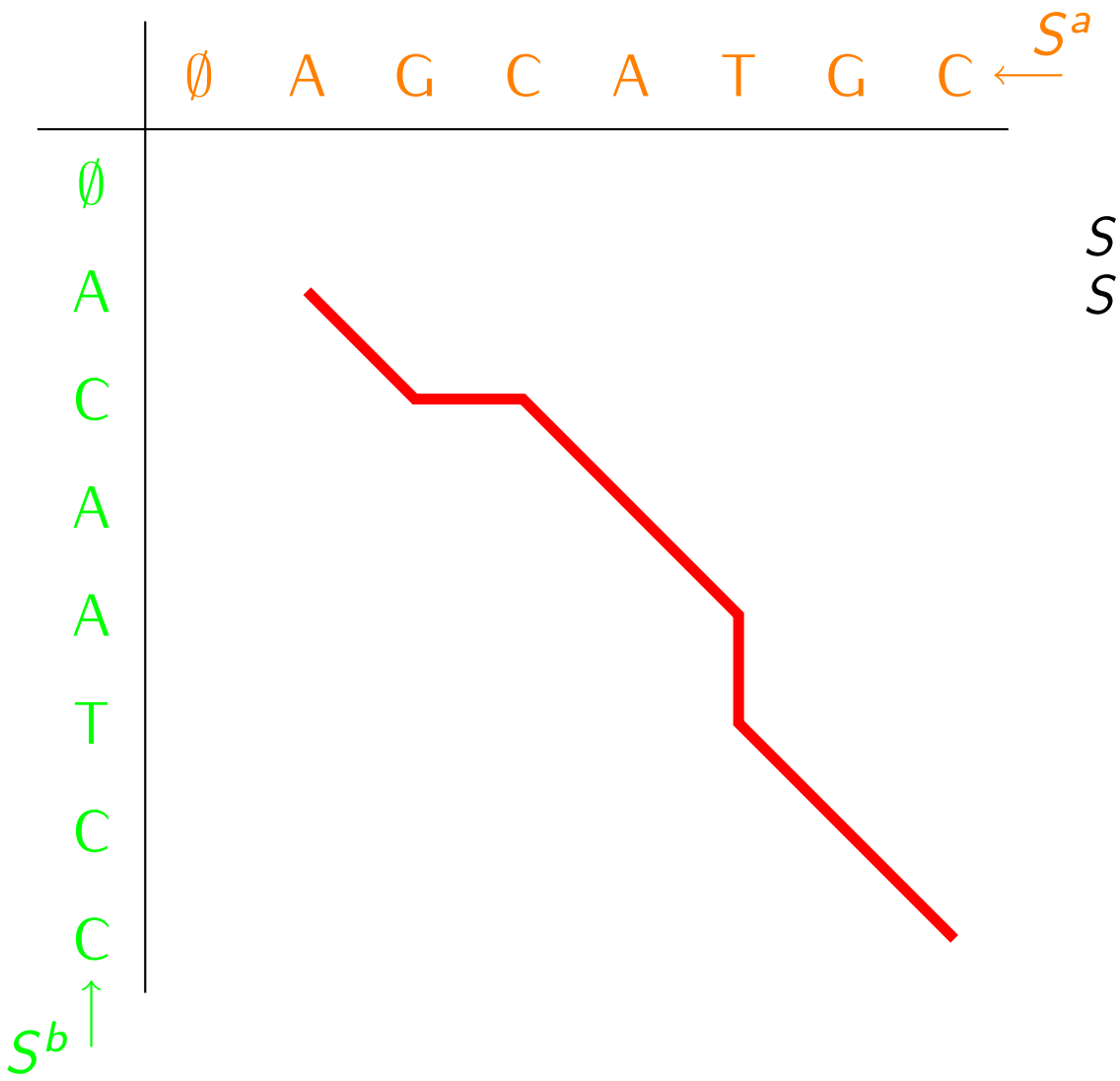
The brute force

- Define scores for insertions, deletions and substitutions
- Generate a possible alignment of the two sequences
- Computes the edit distance for this alignment, and record it
- Generate another alignment, and compute its edit distance, *etc*
- Do that for all possible alignments, and keep the one with the highest score (=minimal edit distance)!

☞ extremely inefficient, as the number for possible alignments is huge!

What is the best alignment?

Part2: Trivial non-optimal alignments

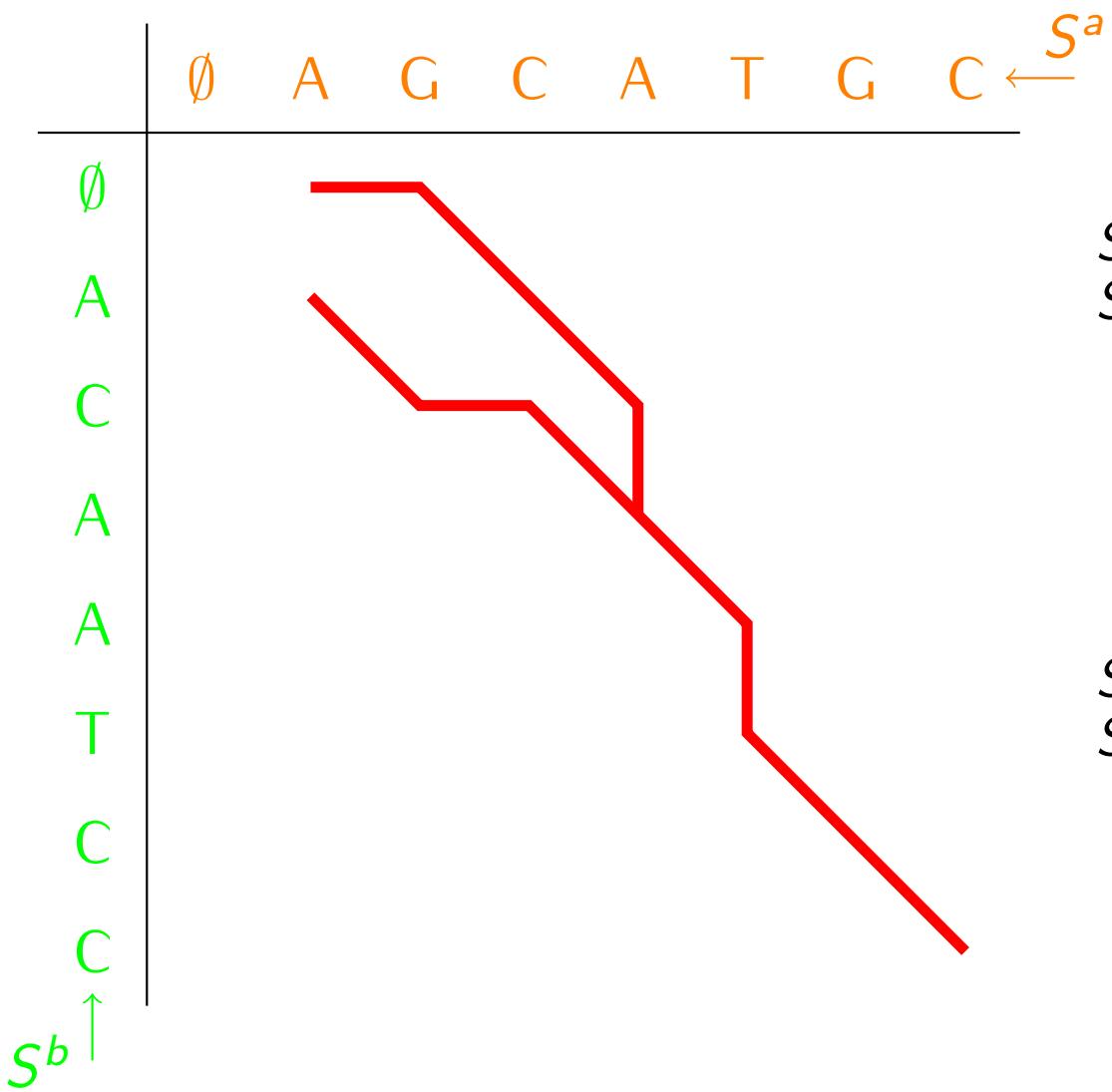


$$S^a = \texttt{A G C A T - G C}$$
$$S^b = \texttt{A C - A A T C C}$$
$$\texttt{+2-1-1+2-1-1-1+2}$$

Score $= 1$

What is the best alignment?

Part2: Trivial non-optimal alignments



S^a = A G C A T - G C
S^b = A C - A A T C C

+2−1−1+2−1−1−1+2

Score = 1

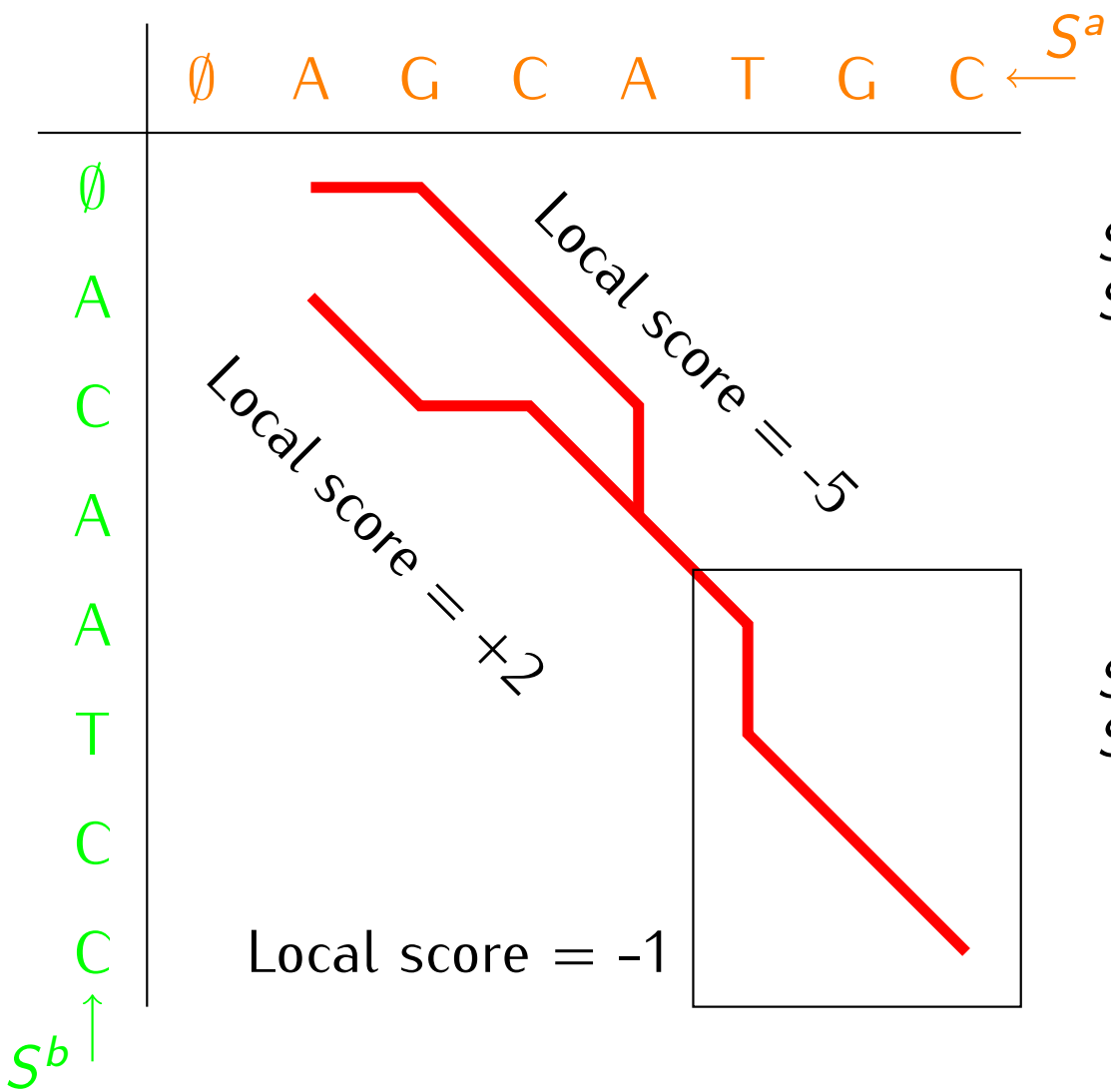
S^a = A G C A - T - G C
S^b = - - A C A A T C C

−1−1−1−1−1−1−1−1+2

Score = -6

What is the best alignment?

Part2: Trivial non-optimal alignments



S^a = A G C A T - G C
S^b = A C - A A T C C
+2−1−1+2−1−1−1+2

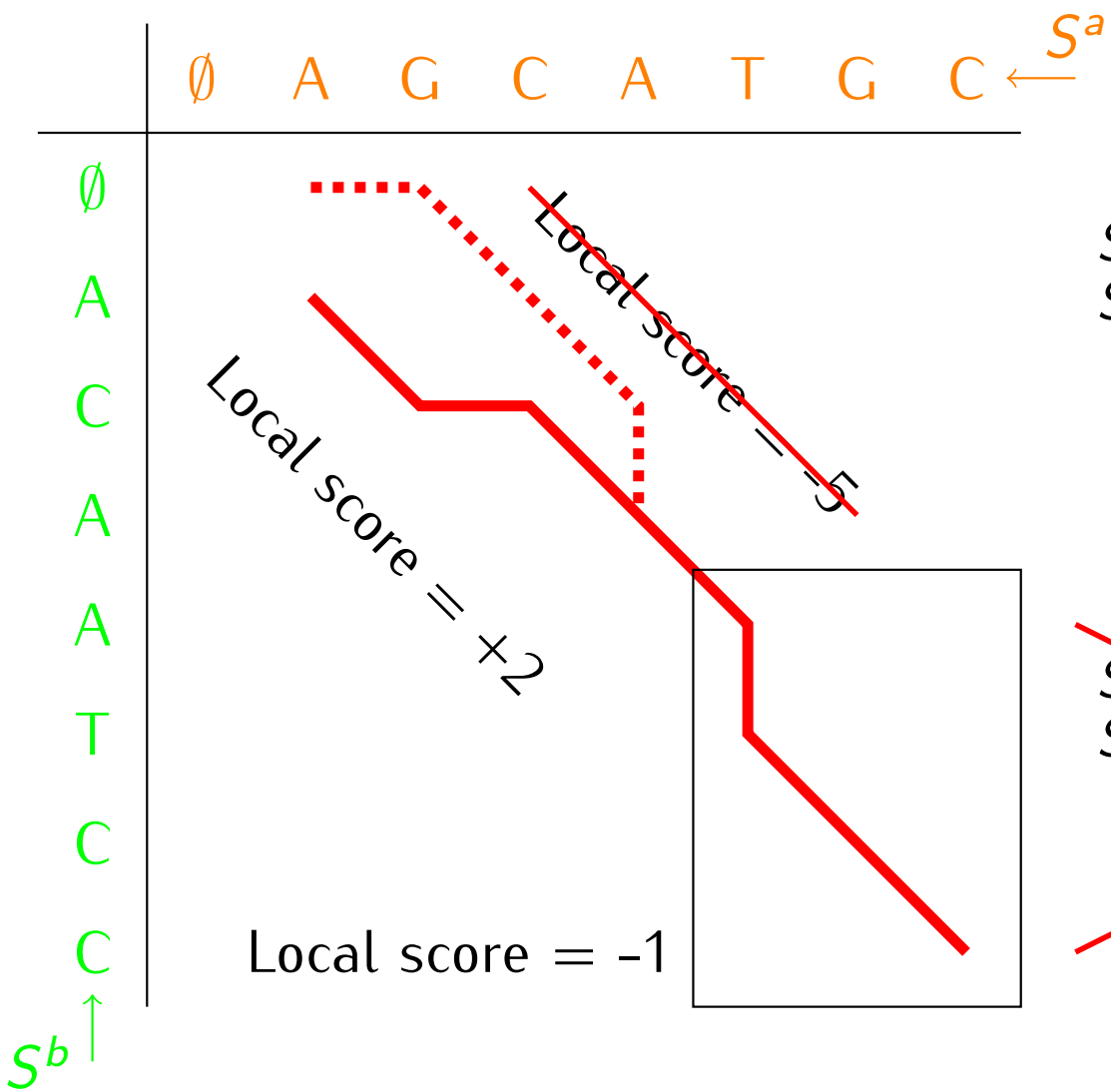
Score = 1

S^a = A G C A - T - G C
S^b = - - A C A A T C C
−1−1−1−1−1−1−1−1+2

Score = -6

What is the best alignment?

Part2: Trivial non-optimal alignments



S^a = A G C A T - G C
S^b = A C - A A T C C
+2−1−1+2−1−1−1+2

Score = 1

S^a = A G C A - T - G C
S^b = - - A C A A T C C
−1−1−1−1−1−1−1−1+2

Score = -6

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0							
A								
C								
A								
A								
T								
C								
C								

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0 ← -1							
A								
C								
A								
A								
T								
C								
C								

$S_1^a = \mathtt{A}$

$S_0^b = \mathtt{-}$

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2					
A								
C								
A								
A								
T								
C								
C								

$$S_2^a = \texttt{AG}$$
$$S_0^b = \texttt{--}$$

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3				
A								
C								
A								
A								
T								
C								
C								

$$S_3^a = \texttt{AGC}$$
$$S_0^b = \texttt{---}$$

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

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

$S_7^a = \texttt{AGCATGC}$

$S_0^b = \texttt{-------}$

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

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

$S_0^a =$ -------

$S_7^b =$ ACAATCC

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0 ←	-1 ←	-2 ←	-3 ←	-4 ←	-5 ←	-6 ←	-7
	↑							
A	-1	?						
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖						
A	-1	2						
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

$S_1^a = \mathtt{A}$

$S_1^b = \mathtt{A}$

$0 + 2 = 2$

Insertion in S^a:

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

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

Match / Mismatch:

$S_1^a =$ A

$S_1^b =$ A

$0 + 2 = 2$

Insertion in S^a:

$S_1^a =$ A-

$S_1^b =$ -A

-1 - 1 = -2

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

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

Match / Mismatch:

$S_1^a = \mathtt{A}$

$S_1^b = \mathtt{A}$

$0 + 2 = 2$

Insertion in S^a:

$S_1^a = \mathtt{A-}$

$S_1^b = \mathtt{-A}$

-1 - 1 = -2

Insertion in S^b:

$S_1^a = \mathtt{-A}$

$S_1^b = \mathtt{A-}$

-1 - 1 = -2

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ -1	↖ 2						
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

$S_1^a = \mathtt{A}$

$S_1^b = \mathtt{A}$

$0 + 2 = 2$

Insertion in S^a:

$S_1^a = \mathtt{A-}$

$S_1^b = \mathtt{-A}$

-1 - 1 = -2

Insertion in S^b:

$S_1^a = \mathtt{-A}$

$S_1^b = \mathtt{A-}$

-1 - 1 = -2

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↖				
A	-1	2		-2				
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

$S_2^a = \texttt{AG}$

$S_1^b = \texttt{-A}$

-1 - 1 = -2

Insertion in S^a:

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖	↑					
A	-1	2	-3					
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

$S_2^a =$ AG

$S_1^b =$ -A

-1 - 1 = -2

Insertion in S^a:

$S_2^a =$ AG-

$S_1^b =$ --A

-2 - 1 = -3

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ -1	↖ 2	← 1					
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

$S_2^a =$ AG

$S_1^b =$ -A

-1 - 1 = -2

Insertion in S^a:

$S_2^a =$ AG-

$S_1^b =$ --A

-2 - 1 = -3

Insertion in S^b:

$S_2^a =$ AG

$S_1^b =$ A-

2 - 1 = 1

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ -1	↖ 2	← 1					
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

$S_2^a =$ AG

$S_1^b =$ -A

-1 - 1 = -2

Insertion in S^a:

$S_2^a =$ AG-

$S_1^b =$ --A

-2 - 1 = -3

Insertion in S^b:

$S_2^a =$ AG

$S_1^b =$ A-

2 - 1 = 1

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ -1	↖ 2	← 1	↖ -3				
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

$S_3^a = \texttt{AGC}$

$S_1^b = \texttt{--A}$

-2 - 1 = -3

Insertion in S^a:

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↑				
A	-1	2 ←	1	-4				
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

S_3^a = AGC

S_1^b = --A

-2 - 1 = -3

Insertion in S^a:

S_3^a = AGC-

S_1^b = ---A

-3 - 1 = -4

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ -1	↖ 2	← 1	← 0				
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

$S_3^a = \texttt{AGC}$

$S_1^b = \texttt{--A}$

-2 - 1 = -3

Insertion in S^a:

$S_3^a = \texttt{AGC-}$

$S_1^b = \texttt{---A}$

-3 - 1 = -4

Insertion in S^b:

$S_3^a = \texttt{AGC}$

$S_1^b = \texttt{A--}$

1 - 1 = 0

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ -1	↖ 2	← 1	← 0				
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

$S_3^a = \texttt{AGC}$

$S_1^b = \texttt{--A}$

-2 - 1 = -3

Insertion in S^a:

$S_3^a = \texttt{AGC-}$

$S_1^b = \texttt{---A}$

-3 - 1 = -4

Insertion in S^b:

$S_3^a = \texttt{AGC}$

$S_1^b = \texttt{A--}$

1 - 1 = 0

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↖				
A	-1	2	← 1	← 0	-1			
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

S_4^a = AGCA

S_1^b = ---A

$-3 + 2 = -1$

Insertion in S^a:

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖			↑			
A	-1	2 ←	1 ←	0	-5			
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

$S_4^a =$ AGCA

$S_1^b =$ ---A

-3 + 2 = -1

Insertion in S^a:

$S_4^a =$ AGCA-

$S_1^b =$ ----A

-4 - 1 = -5

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑	↖						
	-1	2	← 1	← 0	← -1			
C	↑							
	-2							
A	↑							
	-3							
A	↑							
	-4							
T	↑							
	-5							
C	↑							
	-6							
C	↑							
	-7							

Match / Mismatch:

S_4^a = AGCA

S_1^b = ---A

-3 + 2 = -1

Insertion in S^a:

S_4^a = AGCA-

S_1^b = ----A

-4 - 1 = -5

Insertion in S^b:

S_4^a = AGCA

S_1^b = A---

0 - 1 = -1

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↖				
A	-1	2	← 1	← 0	← -1			
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

$S_4^a =$ AGCA

$S_1^b =$ ---A

-3 + 2 = -1

Insertion in S^a:

$S_4^a =$ AGCA-

$S_1^b =$ ----A

-4 - 1 = -5

Insertion in S^b:

$S_4^a =$ AGCA

$S_1^b =$ A---

0 - 1 = -1

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ ↖ -1	2 ←	1	← 0	← -1	↖ ↖ -5		
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

S_5^a = AGCAT

S_1^b = ----A

-4 - 1 = -5

Insertion in S^a:

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↖		↑		
A	-1	2	← 1	← 0	← -1	-6		
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

S_5^a = AGCAT

S_1^b = ----A

-4 - 1 = -5

Insertion in S^a:

S_5^a = AGCAT-

S_1^b = -----A

-5 - 1 = -6

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↖				
A	-1	2	← 1	← 0	← -1	← -2		
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

S_5^a = AGCAT

S_1^b = ----A

-4 - 1 = -5

Insertion in S^a:

S_5^a = AGCAT-

S_1^b = -----A

-5 - 1 = -6

Insertion in S^b:

S_5^a = AGCAT

S_1^b = A----

-1 - 1 = -2

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↖				
A	-1	2	← 1	← 0	← -1	← -2		
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

$S_5^a =$ AGCAT

$S_1^b =$ ----A

-4 - 1 = -5

Insertion in S^a:

$S_5^a =$ AGCAT-

$S_1^b =$ -----A

-5 - 1 = -6

Insertion in S^b:

$S_5^a =$ AGCAT

$S_1^b =$ A----

-1 - 1 = -2

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ ↖ -1	2 ←	1 ←	0 ←	-1 ←	-2 ↖	-6	
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

S_6^a = AGCATG

S_1^b = ------A

-5 - 1 = -6

Insertion in S^a:

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ -1	↖ 2	← 1	← 0	← -1	← -2	↑ -7	
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

S_6^a = AGCATG

S_1^b = -----A

-5 - 1 = -6

Insertion in S^a:

S_6^a = AGCATG-

S_1^b = ------A

-6 - 1 = -7

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ ↖ -1	2 ←	1 ←	0 ←	-1 ←	-2 ←	-3	
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

S_6^a = AGCATG

S_1^b = -----A

-5 - 1 = -6

Insertion in S^a:

S_6^a = AGCATG-

S_1^b = ------A

-6 - 1 = -7

Insertion in S^b:

S_6^a = AGCATG

S_1^b = A-----

-2 - 1 = -3

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ ↖ -1	2 ←	1 ←	0 ←	-1 ←	-2 ←	-3	
C	↑ -2							
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

S_6^a = AGCATG

S_1^b = -----A

-5 - 1 = -6

Insertion in S^a:

S_6^a = AGCATG-

S_1^b = ------A

-6 - 1 = -7

Insertion in S^b:

S_6^a = AGCATG

S_1^b = A-----

-2 - 1 = -3

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

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

Match / Mismatch:

S_7^a = AGCATGC

S_1^b = ------A

-6 - 1 = -7

Insertion in S^a:

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↖				↑
A	-1	2 ←	1 ←	0 ←	-1 ←	-2 ←	-3	-8
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

S_7^a = AGCATGC

S_1^b = ------A

-6 - 1 = -7

Insertion in S^a:

S_7^a = AGCATGC-

S_1^b = -------A

-7 - 1 = -8

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↖				
A	-1	2	← 1	← 0	← -1	← -2	← -3	← -4
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

S_7^a = AGCATGC

S_1^b = ------A

-6 - 1 = -7

Insertion in S^a:

S_7^a = AGCATGC-

S_1^b = -------A

-7 - 1 = -8

Insertion in S^b:

S_7^a = AGCATGC

S_1^b = A------

-3 - 1 = -4

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↖				
A	-1	2	← 1	← 0	← -1	← -2	← -3	← -4
	↑							
C	-2							
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

Match / Mismatch:

S_7^a = AGCATGC

S_1^b = ------A

-6 - 1 = -7

Insertion in S^a:

S_7^a = AGCATGC-

S_1^b = -------A

-7 - 1 = -8

Insertion in S^b:

S_7^a = AGCATGC

S_1^b = A------

-3 - 1 = -4

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ ↖ -1	2 ←	1 ←	0 ←	-1 ←	-2 ←	-3 ←	-4
C	↑ ↖ -2	-2						
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

$S_1^a = -\mathtt{A}$

$S_2^b = \mathtt{AC}$

-1 - 1 = -2

Insertion in S^a:

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

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

Match / Mismatch:

$S_1^a = -\mathtt{A}$

$S_2^b = \mathtt{AC}$

-1 - 1 = -2

Insertion in S^a:

$S_1^a = \mathtt{A}-$

$S_2^b = \mathtt{AC}$

2 - 1 = 1

Insertion in S^b:

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
A	↑ ↖ -1	2 ←	1	← 0	← -1	← -2	← -3	← -4
C	↑ -2	← -3						
A	↑ -3							
A	↑ -4							
T	↑ -5							
C	↑ -6							
C	↑ -7							

Match / Mismatch:

$S_1^a =$ -A

$S_2^b =$ AC

-1 - 1 = -2

Insertion in S^a:

$S_1^a =$ A-

$S_2^b =$ AC

2 - 1 = 1

Insertion in S^b:

$S_1^a =$ --A

$S_2^b =$ AC-

-2 - 1 = -3

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

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

Match / Mismatch:

$S_1^a =$ -A

$S_2^b =$ AC

-1 - 1 = -2

Insertion in S^a:

$S_1^a =$ A-

$S_2^b =$ AC

2 - 1 = 1

Insertion in S^b:

$S_1^a =$ --A

$S_2^b =$ AC-

-2 - 1 = -3

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

	∅	A	G	C	A	T	G	C
∅	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7
	↑	↖		↖				
A	-1	2	← 1	← 0	← -1	← -2	← -3	← -4
	↑	↑	↖	↖			↖	
C	-2	1	1	3	← 2	← 1	← 0	← -1
	↑							
A	-3							
	↑							
A	-4							
	↑							
T	-5							
	↑							
C	-6							
	↑							
C	-7							

The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

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

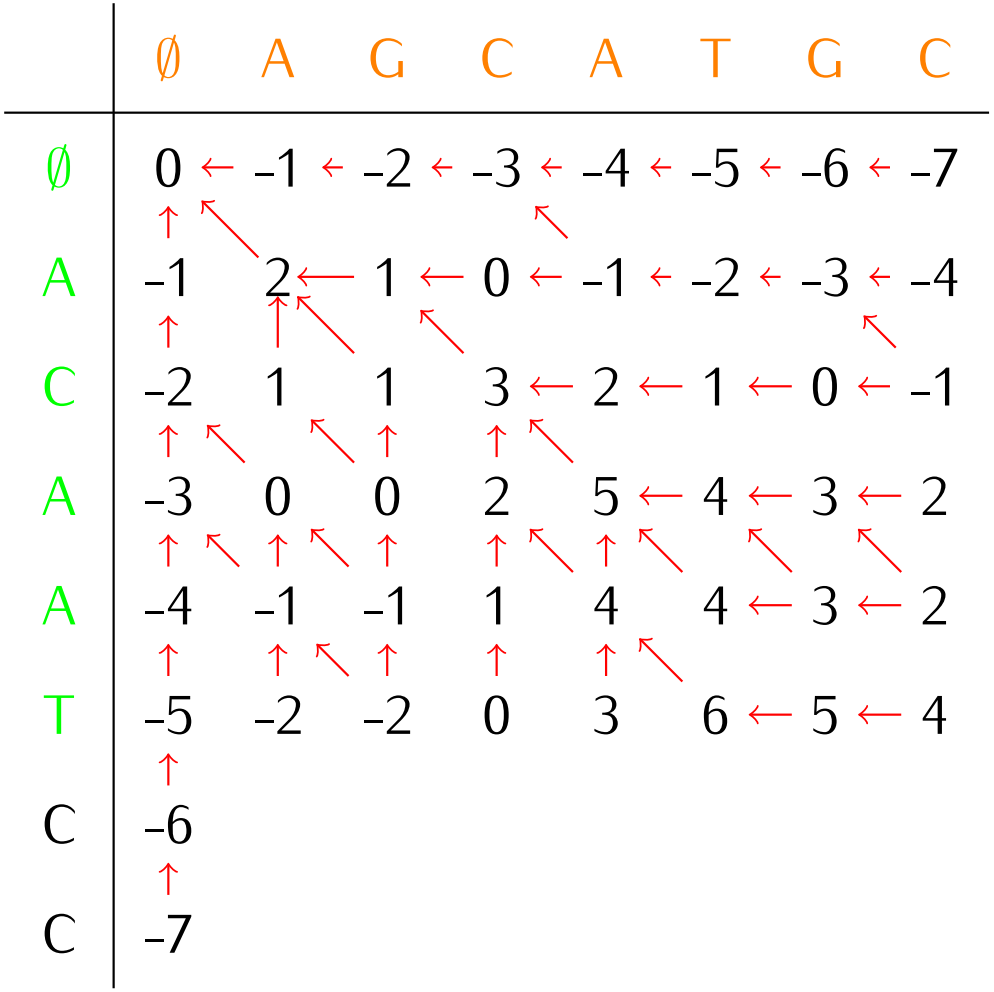
The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.

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

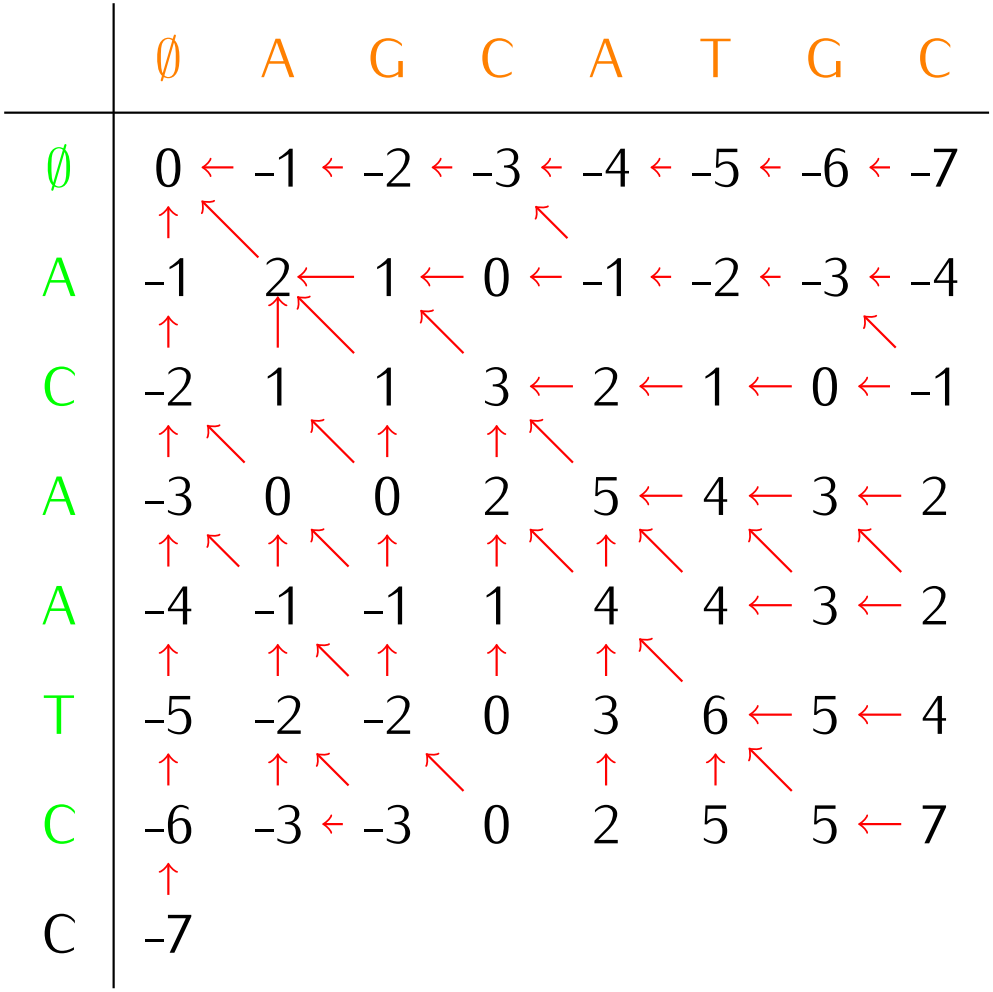
The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.



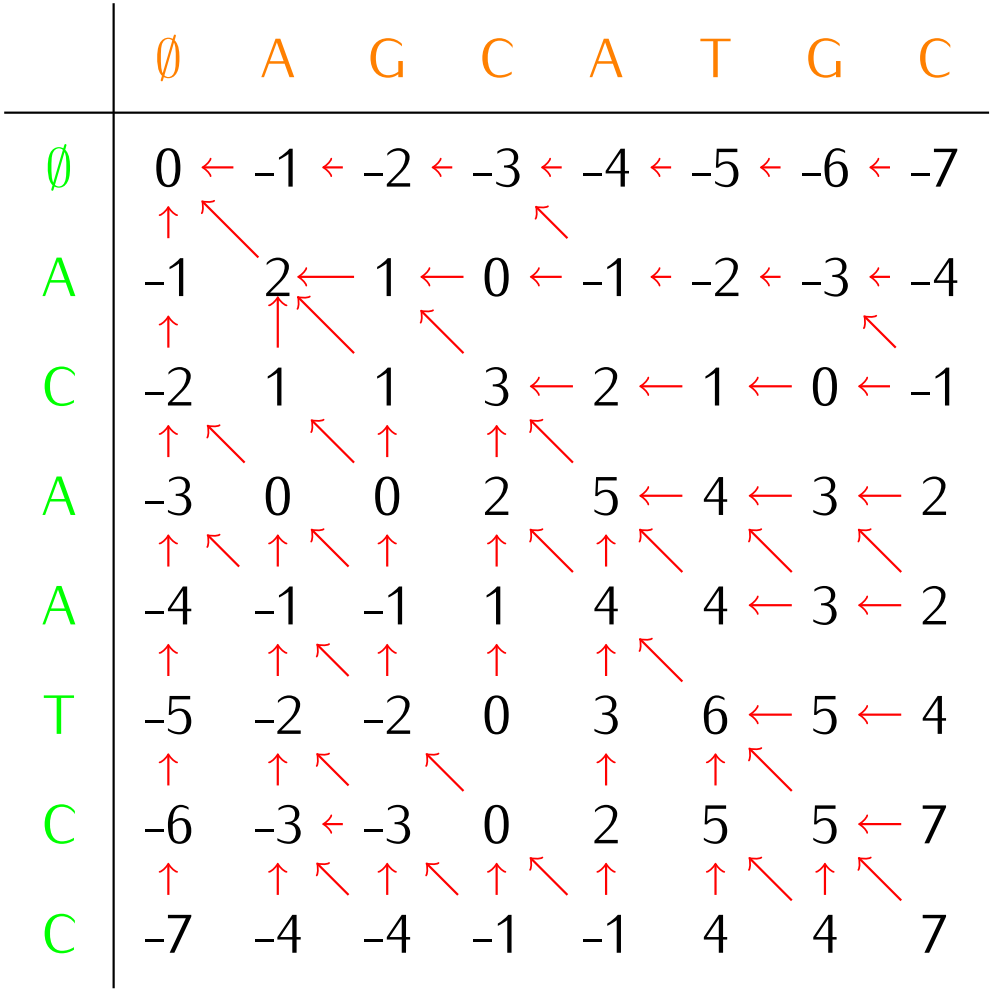
The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.



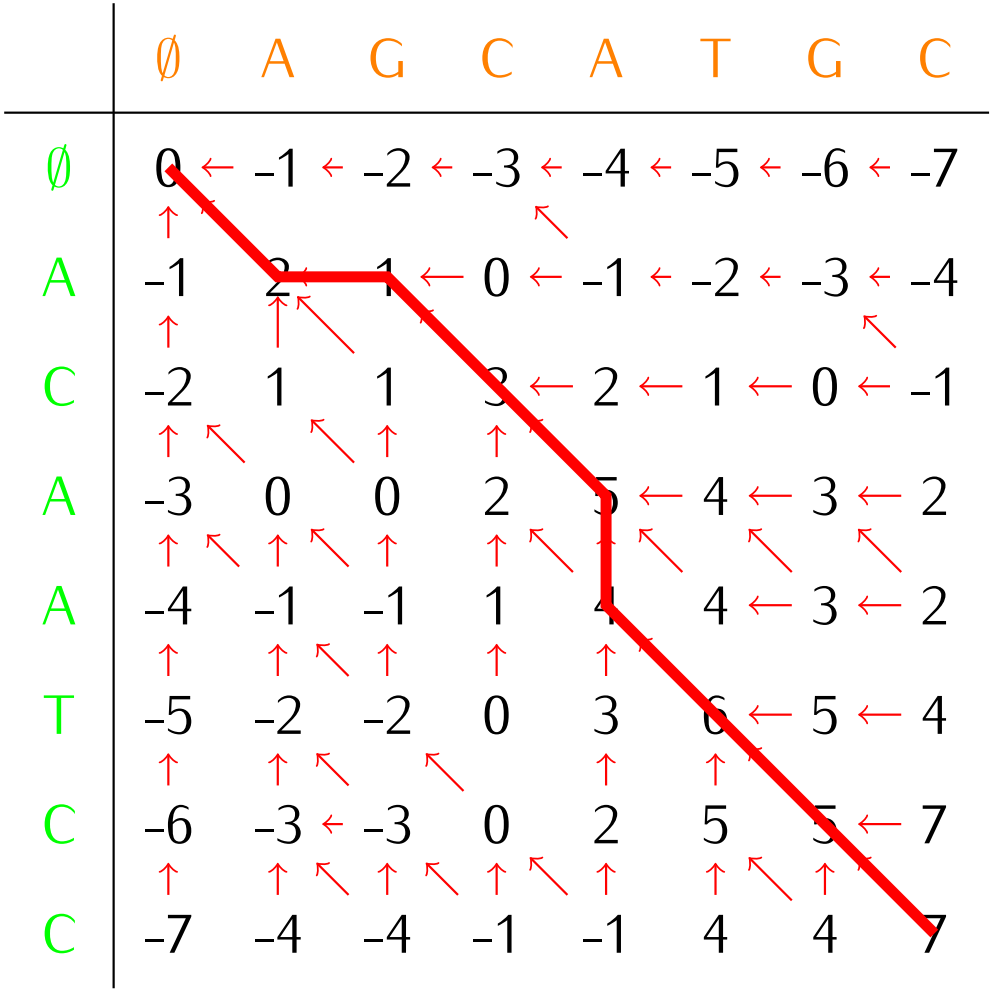
The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.



The Needleman-Wunsch algorithm

Needleman SB and Wunsch CD (1970). Journal of Molecular Biology, 48:443-453.



S_7^a = A G C A - T G C

S_7^b = A - C A A T C C

+2−1+2+2−1+2−1+2

Score = 7

Variations on a theme
Scoring

Variations on a theme
Scoring

Match / Mismatch

Not all substitutions are as likely:

- Transitions / transversions for nucleic acids
- PAM, BLOSUM matrices for proteins

Variations on a theme
Scoring

Match / Mismatch

Not all substitutions are as likely:

- Transitions / transversions for nucleic acids
- PAM, BLOSUM matrices for proteins

Gap penalty

It is biologically more relevant to give a higher cost for gap opening than for gap extension. The algorithm can be generalized to account for a gap penalty, as a function of the gap length (the longer the gap, the easier it is to extend it).

Variations on a theme
Semi-global alignment

☞ Some issues arise when aligning a small sequence on a large one:

$$\begin{aligned} S^a &= \texttt{ATCCGAACATCCAATCGAAGC} \\ S^b &= \texttt{AGCATGCAAT} \end{aligned}$$

are aligned as

$$\begin{aligned} S^a &= \texttt{ATCCGAACATCCAATCGAAGC} \\ S^b &= \texttt{A---G--CATGCAAT------} \end{aligned}$$

Semi-global alignment

- Initialize first row and first column with 0 instead of negative scores
- Back-tracing: start from highest score in the last row and the last column

Semi-global alignment

- Initialize first row and first column with 0 instead of negative scores
- Back-tracing: start from highest score in the last row and the last column

$$
\begin{aligned}
S^a &= \texttt{ATCCGAACATCCAATCGAAGC} \\
S^b &= \texttt{-----AGCATGCAAT------}
\end{aligned}
$$

Variations on a theme
Local alignment

- (Semi)global alignment: align two complete sequences
- Local alignment: find the best alignment of two sub-sequences

☞ A local alignment starts and ends with a match

Variations on a theme
Local alignment

- (Semi)global alignment: align two complete sequences
- Local alignment: find the best alignment of two sub-sequences

☞ A local alignment starts and ends with a match

Local alignment

☞ Smith TF and Waterman MS (1981). Journal of Molecular Biology, 147:195-197.

- Alignment with empty sequence costs 0 (as in semi-global)
- No negative score! If a sub-alignment has negative score, then its score is set to 0
- Back-tracing: we start from the highest value in the full table

Multiple alignment

☞ An alignment of at least 3 sequences!

```
KGVHSDLNET YFVGDINDVP KEGKELVETC YFSLMEAIKK CKPGMFYKNI GTLIDAYVSK KN-------F SVVRSYSGHG VGKLFHSNP
KGVHADLNET YFVGENIS-- NEAKQLVETC YFSLMEAIKK CKPGMLYKNL GNIIDAYVSK KH-------F SVIRTYSGHG VGKLFHSNP
QGMHADLNET YYVGDNIS-- KEALNTVETA RECLKIATKM CKPGVRFQDL GDAIEKHAKQ NK-------C SVVKTYCGHG VGKFFHCSP
QGYHADLNET YYVGENIS-- KEALNTTETS RECLKLAIKM CKPGTTFQEL GDHIEKHATE NK-------C SVVRTYCGHG VGEFFHCSP
LGFHADLNET YYVGDKAKCN PELVNLVETT RECLDLAIKH VKPGIAFREL GNIIEKHASE NN-------C SVVRTYCGHG CGALFHCQP
NGFHGDLNET YYVGDKAKAN PDLVCLVENT RIALDKAIAA VKPGVLFQEF GNIIEKHTNS IT----EKQI SVVRTYCGHG INQLFHCSP
EGFHGDINET YYVGEKARSN PDAVRVVETA RECLDKSIEI VKPGMLFRDP GNVIEKHAKS RN-------C SVVKSYCGHG INQLFHCAP
EGYHADLNET YYIGDKAKAD PDTVRVVETA RQCLDESIKA VKPGTLIREF GNIIEKHAKQ HN-------C SVIRTYCGHG VGKLFHCPP
EGYHGDLNET YYVGDKAKAD ADSVRVVETA RECLEEAIKL VKPGTLFRDF GNVIEAHAKS RG-------C SVIRTYVGHG INKTFHCPP
EGYHGDLNET YYVGDKALAD PDVVRLVETT RECLDEAIKL VKPGTLFREF GNVIEKHAKA NN-------C SVIRTYVGHG INSVFHCPP
KGYHGDLNAT FPVGKKAEDD AESMKLIRVA RECLDAAINI CGPGVPYGEI GRVIQPLAES QG-------C AVVKNYTGHG ISNCFHAAP
KGCHGDLNET YFVGNVD--- EASRQLVKCT YECLEKAIAI VKPGVRFREI GEIVNRHATM SG-------L SVVRSYCGHG IGDLFHCAP
RGFHGDLNET FFVGNVS--- EKHKKLVQVT HEALSKAIEF VRPGEKYRDI GNVIQKYVAP HG-------F SVVRSYCGHG IHRVFHTAP
DSYHGDTSRT FFVGTPS--- PLAKRLVEVT EKCLMEAIKT VKPGSRIGDI GAIIQECAEP QG-------F SVVRDFVGHG VSKVFHTAP
DGYHGDTSKT FIVGNAA--- PKTKKLVEVT QECLNLGIAE VKPGAKIGDI GAAIQEYAEA QG-------F SVVRDFVGHG ISNIFHTAP
EGYHGDTSKT FFVGTPS--- PKTRKLVEVT EECLRLGIAE VKPGGRIGDI GAAIQEYAEG QG-------F SVVRDFVGHG ISNIFHTAP
NGYYSDASRM FIIGEAS--- ENAKRLVKVA KECLEKGIEA VKPWGFLGDI GAAIQEHAEK NG-------Y SVVRDFGGHG VGLKFHEDP
KGYFSDASRM FMIGDVS--- PEMRKLVQVT KECMEIGIAA AQPWKQLGDV GAAIQEHAEK NG-------F NVVRDLCGHG VGMQFHEAP
DGYHGDSSRT FIVGGNTS-- SEVKTLVQDT EKAMFIGIEQ VRPGNRVHDI ANAIDDFLTP KG-------Y GIVRDLMGHG IGRGFHEDP
DGYFGDNSKM YIVGGETN-- IRSKKLVEAA QEALYVGIRT VKPDIRLNEI GKAVQKYTES QT-------F SVVREYCGHG VGTEFHCEP
DGYFGDNSKM YIVGGETN-- IRSKKLVEAA QEALYVGIRT VKPDIRLNEI GKAVQKYTES QT-------F SVVREYCGHG VGTEFHCEP
DGYHGDTSKM FLIGDVS--- IEDKRLCHVA QECLYLALKQ VKPGVQLGEI GTTIEKHIKT NNKNNPRFKF SIVRDYCGHG IGAEFHEEP
QGYHGDTSKM FLVGDVS--- PANKRLCMVA QEALYVGMRT VKPGSTVGDI GTAIEKYIKE NNKNNPRNKF SIVKDFCGHG IGDEFHEEP
QGFHGDTSKM FLVGDVS--- PANKRLCMVA QEALYIGMRQ VKPGATVGDI GTAIEKYIKD NNKNNPRNKF SIVKDFCGHG IGDEFHEEP
DGYHGDTSAM FIVGETT--- PLRRQLCKVA QESLYAAIKQ VRPGMCITEI GAVIQPIVEK AG-------F SVVRDYCGHG IGAEFHEEP
DGYHGDTSKM FYIGKTS--- ILSKRLCQVA RESLYLSLKL VKPGIPLYKI GEIIQNYVES NN-------F SVVKEYCGHG IGRNFHEEP
DKYYSDASKM FFVGKPT--- ELGKKLCYVA KKSLYLALYT IRPGINLQKL GKVIQNYVKK QN-------F SIVKEYCGHG IGRSFHEPP
```

Finding the best multiple alignment

- Score of a multiple alignment: sum of the scores of all pairwise alignments

Finding the best multiple alignment

- Score of a multiple alignment: sum of the scores of all pairwise alignments
- Several approaches to maximize the score:
  - Global maximization approaches
  - Agglomerative approaches
  - Iterative approaches

Finding the best multiple alignment
Global maximize approaches

Very slow as the number of sequences increases • several approximations exist • yet not usable for medium or large data sets

☞ Example methods: MSA, DCS, SAGA

Finding the best multiple alignment
Agglomerative approaches

Heuristic: use a series of pairwise alignments to build a multiple alignment • first proposed by Feng DF and Doolittle RF (1987). Journal of Molecular Evolution, 25:351-360.

Finding the best multiple alignment
Agglomerative approaches

Heuristic: use a series of pairwise alignments to build a multiple alignment • first proposed by Feng DF and Doolittle RF (1987). Journal of Molecular Evolution, 25:351-360.

1. Compute scores for all pairwise alignments
2. Build a clustering tree from the scores, to identify the closest sequences (guide tree)
3. Align sequences following the guide tree

Finding the best multiple alignment
Agglomerative approaches

Heuristic: use a series of pairwise alignments to build a multiple alignment • first proposed by Feng DF and Doolittle RF (1987). Journal of Molecular Evolution, 25:351-360.

1. Compute scores for all pairwise alignments
2. Build a clustering tree from the scores, to identify the closest sequences (guide tree)
3. Align sequences following the guide tree

Profile-profile alignment

Gather two alignments by inserting gap columns in each of them, without modifying the original alignments

☞ Can be done using a modification of the global alignment procedure for two sequences

Finding the best multiple alignment
Agglomerative approaches

Heuristic: use a series of pairwise alignments to build a multiple alignment • first proposed by Feng DF and Doolittle RF (1987). Journal of Molecular Evolution, 25:351-360.

1. Compute scores for all pairwise alignments
2. Build a clustering tree from the scores, to identify the closest sequences (guide tree)
3. Align sequences following the guide tree

Profile-profile alignment

Gather two alignments by inserting gap columns in each of them, without modifying the original alignments

☞ Can be done using a modification of the global alignment procedure for two sequences

☞ Example programs: ClustalW

Finding the best multiple alignment
Iterative approaches

Problems with the agglomerative approach:

- The existing sub alignments are never modified, only fused
- Highly depends on the guiding tree, any early error is reported to all further steps

Finding the best multiple alignment
Iterative approaches

Problems with the agglomerative approach:

- The existing sub alignments are never modified, only fused
- Highly depends on the guiding tree, any early error is reported to all further steps

1. Build a multiple alignment using an agglomerative approach
2. Use the resulting alignment to build a new guiding tree
3. Build a new multiple alignment
4. etc.

☞ Example software: MAFFT, MUSCLE

Other approaches to multiple alignment

Multiple Local Alignment:

- DIALIGN

Probabilistic methods:

- Joint estimation of phylogeny and alignment: StatAlign, BAli-Phy
- Use models to tune the alignment: PRANK, PROBCONS, ProAlign, MUMMALS, etc.

Meta-alignments:

- Take a consensus of several methods, example: M-Coffee

Evaluating the confidence of alignments

- ☞ All methods assume that sequences are homologous! There will always be an output, even from random input sequences!
- ☞ Apart from methodological artifacts, some regions are more difficult to align than others, and are therefore more prone to errors

Evaluating the confidence of alignments

- ☞ All methods assume that sequences are homologous! There will always be an output, even from random input sequences!
- ☞ Apart from methodological artifacts, some regions are more difficult to align than others, and are therefore more prone to errors

  - Methods based on block conservation: GBlocks
  - The Heads-or-Tails method
  - Methods assessing the uncertainty in the guide tree: GUIDANCE

Genome alignment

Genome alignment

- Complexity issue (memory, CPU)
- Caryotype
- Genome rearrangements
- Duplicates/Repeated regions
- Recombination
- etc.

Genome alignment
General strategy

1. Find anchors = highly conserved regions between two genomes. For example: shared $k-$mers,

Genome alignment
General strategy

1. Find anchors = highly conserved regions between two genomes. For example: shared $k-$mers,

Maximum Unique Match (MUM)

A MUM is a substring of a given minimum size (usually 20) that is common to two genomes, so that

- it cannot be extend on either side without introducing a mismatch,
- it is unique to both sequences.

Genome alignment
General strategy

1. Find anchors = highly conserved regions between two genomes. For example: shared $k-$mers,
2. Define conserved regions: sets of collinear, non-overlapping anchors, which form the basis of the alignment,

Maximum Unique Match (MUM)

A MUM is a substring of a given minimum size (usually 20) that is common to two genomes, so that

- it cannot be extend on either side without introducing a mismatch,
- it is unique to both sequences.

Genome alignment
General strategy

1. Find anchors = highly conserved regions between two genomes. For example: shared $k-$mers,
2. Define conserved regions: sets of collinear, non-overlapping anchors, which form the basis of the alignment,
3. Close the gaps between the anchors to build the alignment.

Maximum Unique Match (MUM)

A MUM is a substring of a given minimum size (usually 20) that is common to two genomes, so that

- it cannot be extend on either side without introducing a mismatch,
- it is unique to both sequences.