



# Algorithms for pairwise sequence alignments

Lukas Käll  
lukask@kth.se





## What is needed to define an optimal pairwise alignment?

- ▶ A **scoring function**,  $d(x, y)$ , giving the score of a column of any letter  $x$  and  $y$ . A typical scoring function could be

$$d(x, y) = \begin{cases} p & \text{if } x = y \\ g & \text{if } x = - \text{ or } y = - . \\ n & \text{otherwise} \end{cases}$$

Here,  $p$ , is called a match score,  $n$ , a mismatch score, and  $g$  a gap penalty.



## What is needed to define an optimal pairwise alignment?

- ▶ A **scoring function**,  $d(x, y)$ , giving the score of a column of any letter  $x$  and  $y$ . A typical scoring function could be

$$d(x, y) = \begin{cases} p & \text{if } x = y \\ g & \text{if } x = - \text{ or } y = - \\ n & \text{otherwise} \end{cases}$$

Here,  $p$ , is called a match score,  $n$ , a mismatch score, and  $g$  a gap penalty.

- ▶ An **alignment approach**. If we want to find an optimal alignment of the full length sequences, we are searching a *global* alignment approach. If we search the highest scoring stretch of an alignment, you should use a *local* alignment approach. You can also use a semi-global alignment, searching for an optimal alignment, with the exception for any overshooting sequence terminals.



## Needleman-Wunsch (global alignment)

Given two sequences  $a_1, \dots, a_N$  and  $b_1, \dots, b_M$ , a scoring function  $d(x,y)$ , we can find an optimal *global* alignment by investigating the dynamic programming matrix of size  $(N+1, M+1)$ , defined by

$$S_{0,0} = 0,$$

$$S_{i,0} = d(x, -) \cdot i \text{ for all } i,$$

$$S_{0,j} = d(-, y) \cdot j \text{ for all } j$$

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} & +d(a_i, b_j) \\ S_{i-1,j} & +d(a_i, -) \\ S_{i,j-1} & +d(-, b_j) \end{cases}$$

The score of an optimal alignment is  $S_{N,M}$ .

## Needleman-Wunsch (global alignment)

Given two sequences  $a_1, \dots, a_N$  and  $b_1, \dots, b_M$ , a scoring function  $d(x,y)$ , we can find an optimal *global* alignment by investigating the dynamic programming matrix of size  $(N+1, M+1)$ , defined by

$$S_{0,0} = 0,$$

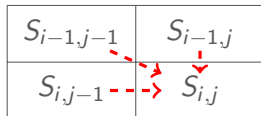
$$S_{i,0} = d(x, -) \cdot i \text{ for all } i,$$

$$S_{0,j} = d(-, y) \cdot j \text{ for all } j$$

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} & +d(a_i, b_j) \\ S_{i-1,j} & +d(a_i, -) \\ S_{i,j-1} & +d(-, b_j) \end{cases}$$

The score of an optimal alignment is  $S_{N,M}$ .

$S_{i-1,j-1}$	$S_{i-1,j}$
$S_{i,j-1}$	$S_{i,j}$



## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red arrows indicate the path from (0,0) to (1,0), (2,0), and (3,0).

$$S_{0,0} = 0,$$

$$S_{i,0} = -1 \cdot i \text{ for all } i,$$

$$S_{0,j} = -1 \cdot j \text{ for all } j$$

## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red dashed arrows indicate the path from (0,0) to (1,1): (0,0) to (0,1) down, (0,1) to (1,1) down-right.

$$S_{1,1} = \max \begin{cases} S_{0,0} + d(G, A) & = 0 + -1 = -1 \\ S_{0,1} + d(G, -) & = -1 + -1 = -2 \\ S_{1,0} + d(-, A) & = -1 + -1 = -2 \end{cases}$$

## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red dashed arrows indicate the optimal alignment path: (-,0) → (G,-1) → (G,A) → (G,C) → (A,C) → (A,G).

$$S_{1,2} = \max \begin{cases} S_{0,1} + d(G, C) & = -1 + -1 = -2 \\ S_{0,2} + d(G, -) & = -2 + -1 = -3 \\ S_{1,1} + d(-, C) & = -1 + -1 = -2 \end{cases}$$



## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red dashed arrows indicate the alignment path: from (0,0) to (1,1) to (2,2) to (3,3).

$$S_{1,3} = \max \begin{cases} S_{0,2} + d(G, G) & = -2 + 1 = -1 \\ S_{0,3} + d(G, -) & = -3 + -1 = -4 \\ S_{1,2} + d(-, G) & = -2 + -1 = -3 \end{cases}$$

## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red dashed arrows indicate the alignment path: (-, -) to (G, A) to (A, C) to (C, G).

$$S_{2,1} = \max \begin{cases} S_{1,0} + d(A, A) & = -1 + 1 = 0 \\ S_{1,1} + d(A, -) & = -1 + -1 = -2 \\ S_{2,0} + d(-, A) & = -2 + -1 = -3 \end{cases}$$

## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red dashed arrows indicate the alignment path: (-, -) to (G, A) to (A, C) to (C, G).

$$S_{2,2} = \max \begin{cases} S_{1,1} + d(A, C) & = -1 + -1 = -2 \\ S_{1,2} + d(A, -) & = -2 + -1 = -3 \\ S_{2,1} + d(-, C) & = 0 + -1 = -1 \end{cases}$$

## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red dashed arrows indicate the optimal alignment path: (-,0) → (G,-) → (G,A) → (A,A) → (A,C) → (C,C).

$$S_{2,3} = \max \begin{cases} S_{1,2} + d(A, G) & = -2 + -1 = -3 \\ S_{1,3} + d(A, -) & = -1 + -1 = -2 \\ S_{2,2} + d(-, G) & = -1 + -1 = -2 \end{cases}$$

## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red dashed arrows indicate the alignment path: (-, -) to (-, A) to (G, A) to (A, A) to (C, A).

$$S_{3,1} = \max \begin{cases} S_{2,0} + d(C, A) & = -2 + -1 = -3 \\ S_{2,1} + d(C, -) & = 0 + -1 = -1 \\ S_{3,0} + d(-, A) & = -3 + -1 = -4 \end{cases}$$

## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red dashed arrows indicate the optimal alignment path: (-, -) → (G, A) → (A, C) → (C, G).

$$S_{3,2} = \max \begin{cases} S_{2,1} + d(C, C) & = 0 + 1 = +1 \\ S_{2,2} + d(C, -) & = -1 + -1 = -2 \\ S_{3,1} + d(-, C) & = -1 + -1 = -2 \end{cases}$$

## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red dashed arrows indicate the optimal alignment path: (-, -) → (G, A) → (A, C) → (C, G).

$$S_{3,3} = \max \begin{cases} S_{2,2} + d(C, G) & = -1 + -1 = -2 \\ S_{2,3} + d(C, -) & = -2 + -1 = -3 \\ S_{3,2} + d(-, G) & = 1 + -1 = 0 \end{cases}$$

## Example of Needleman-Wunsch (global alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Diagram illustrating the Needleman-Wunsch algorithm for global alignment. The table shows the dynamic programming table with scores for aligning sequences  $a = \text{GAC}$  and  $b = \text{ACG}$ . Red dashed arrows indicate the path of the optimal alignment, and blue solid arrows indicate the backtracing path.

Optimal score given by  $S_{3,3} = 0$ .

An optimal alignment can be found by **back tracing**  $(-,G)$ ,  $(C,C)$ ,  $(A,A)$ ,  $(G,-)$  i.e.

GAC-  
-ACG





## Smith-Waterman (local alignment)

Given two sequences  $a_1, \dots, a_N$  and  $b_1, \dots, b_M$ , a scoring function  $d(x,y)$ , we can find an optimal *local* alignment by investigating the dynamic programming matrix of size  $(N+1, M+1)$ , defined by

$$\begin{aligned} S_{0,0} &= 0, \\ S_{i,0} &= 0 \text{ for all } i, \\ S_{0,j} &= 0 \text{ for all } j \end{aligned}$$

The score of an optimal alignment is  $\max_{i,j} S_{i,j}$

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} & +d(a_i, b_j) \\ S_{i-1,j} & +d(a_i, -) \\ S_{i,j-1} & +d(-, b_j) \\ 0 \end{cases}$$

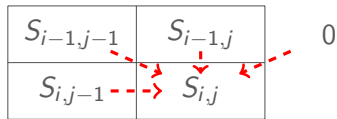
## Smith-Waterman (local alignment)

Given two sequences  $a_1, \dots, a_N$  and  $b_1, \dots, b_M$ , a scoring function  $d(x,y)$ , we can find an optimal *local* alignment by investigating the dynamic programming matrix of size  $(N+1, M+1)$ , defined by

$$\begin{aligned} S_{0,0} &= 0, \\ S_{i,0} &= 0 \text{ for all } i, \\ S_{0,j} &= 0 \text{ for all } j \end{aligned}$$

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} & +d(a_i, b_j) \\ S_{i-1,j} & +d(a_i, -) \\ S_{i,j-1} & +d(-, b_j) \\ 0 \end{cases}$$

The score of an optimal alignment is  $\max_{i,j} S_{i,j}$



## Example of Smith-Waterman (local alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

	-	A	C	G
-	0	0	0	0
G	0			
A	0			
C	0			

$$S_{0,0} = 0,$$

$$S_{i,0} = 0 \text{ for all } i,$$

$$S_{0,j} = 0 \cdot j \text{ for all } j$$

## Example of Smith-Waterman (local alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Red dashed arrows indicate the path of the local alignment: from (3,3) to (2,3), (2,3) to (2,2), (2,2) to (1,2), (1,2) to (1,1), (1,1) to (1,2), (1,2) to (2,2), (2,2) to (3,2), (3,2) to (3,3).

$$S_{3,3} = \max \begin{cases} S_{2,2} + d(C, G) & = 0 + -1 = -1 \\ S_{2,3} + d(C, -) & = 0 + -1 = -1 \\ S_{3,2} + d(-, G) & = 2 + -1 = 1 \\ 0 & \end{cases}$$

## Example of Smith-Waterman (local alignment)

Align  $a = \text{GAC}$ ,  $b = \text{ACG}$ , using  $d(x, y) = \begin{cases} 1 & \text{if } x = y \\ -1 & \text{otherwise} \end{cases}$ .

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

Diagram illustrating the Smith-Waterman local alignment matrix for sequences  $a = \text{GAC}$  and  $b = \text{ACG}$ . The matrix shows scores for each pair of characters. Red dashed arrows indicate the path of the optimal alignment: from (C,C) to (A,A) to (G,G). Blue solid arrows indicate the path of the optimal alignment: from (C,C) to (A,A) to (G,G).

Optimal score given by  $\max_{i,j} S_{i,j} = 2$ .

An optimal alignment can be found by **back tracing** (C,C), (A,A) i.e.

AC

AC



## Semi-global alignment

Given two sequences  $a_1, \dots, a_N$  and  $b_1, \dots, b_M$ , a scoring function  $d(x,y)$ , we can find an optimal *semi-global* alignment by investigating the dynamic programming matrix of size  $(N+1, M+1)$ , defined by

$$S_{0,0} = 0,$$

$$S_{i,0} = 0 \text{ for all } i,$$

$$S_{0,j} = 0 \text{ for all } j$$

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} & +d(a_i, b_j) \\ S_{i-1,j} & +d(a_i, -) \\ S_{i,j-1} & +d(-, b_j) \end{cases}$$

The score of an optimal alignment is  
 $\max_i (\max_j S_{i,M})$



Thanks!