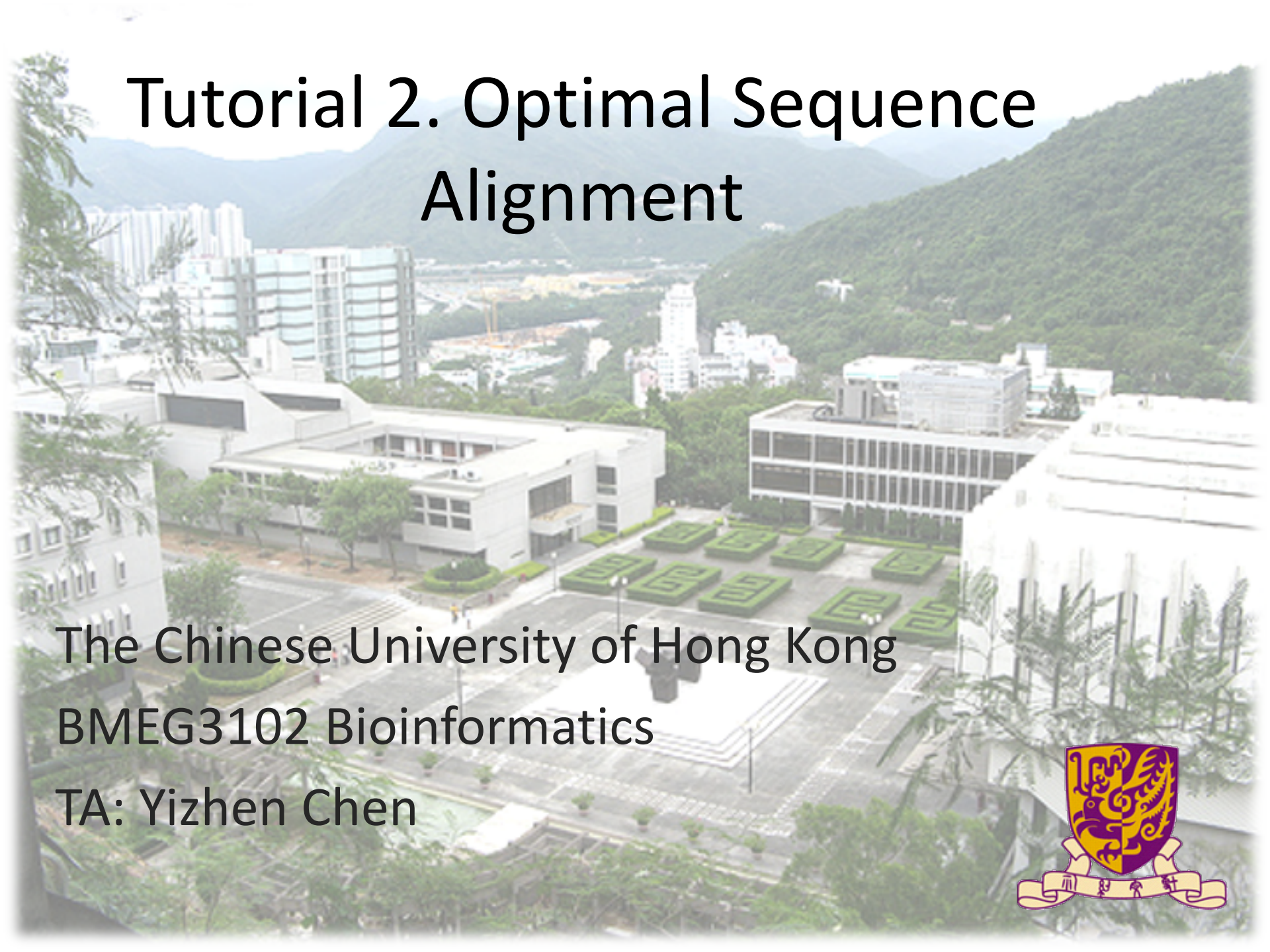
Tutorial 2. Optimal Sequence Alignment



The Chinese University of Hong Kong

BMEG3102 Bioinformatics

TA: Yizhen Chen

Agenda

- What is sequence alignment?
- Sequence data and databases
- Global alignment
- Local alignment
- Scoring matrix

What is sequence alignment?

- Biological problem:
  - **Identify** consistent **patterns** in DNA, RNA or protein sequences associated with genetic diseases or important biological functions (e.g. TF binding)
  - **Evaluate** the **similarity** of the sequences of different individuals or different species

- Computational problem:
  - Given 2 or more DNA, RNA or protein sequences, find the best alignment according to some scoring matrix, σ.

Sequence data and databases

- Sequence data
  - DNA: {A, C, G, T}
  - RNA: {A, C, G, U}
  - Protein: 20 amino acids
- Nucleotide databases
  - GenBank
  - EMBL-EBI
  - DDBJ
- Protein sequence database
  - UniProt

Dynamic Programming (DP)

- **Dynamic programming** is a method for simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner, solving each of those subproblems just once, and storing their solutions.

Key Idea: Store the results of overlapping smaller sub-problems to avoid repetitive computations.

Global alignment

- Find the optimal alignment for the whole two sequences, *r* and *s*.
- Needleman-Wunsch algorithm
- Initialization: depend on the gap penalty
- DP Table Construction:

$$\max\begin{cases} V(i+1,j+1)+\sigma(r[i],s[j]) \\ V(i,j+1)+\sigma('\_',s[j]) \\ V(i+1,j)+\sigma(r[i],'\_') \end{cases}$$

- Optimal alignment score: *V*(1, 1)

Global alignment: An example

- *r*: GCTTAG
- *s*: ACTTCGG

Match: +1; otherwise: -1

r \ s	A	C	T	T	C	G	G	ϕ
G								-6
C								-5
T								-4
T								-3
A								-2
G								-1
ϕ	-7	-6	-5	-4	-3	-2	-1	0

Recall: $V(i, j)$ equals the optimal (i.e., highest) alignment score of the suffixes $r[i..m]$ and $s[j..n]$

Initialization:
Depend on the gap penalty

Global alignment: An example

- *r*: GCTTAG
- *s*: ACTTCGG

$$V(i,j) = \max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \end{cases}$$

r \ s	A	C	T	T	C	G	G	ϕ
G								-6
C								-5
T								-4
T								-3
A								-2
G							1	-1
ϕ	-7	-6	-5	-4	-3	-2	-1	0

Align *r*[6..6] with *s*[7..7]:
max(0+1, -1-1, -1-1) =
max(1, -2, -2) = 1

Global alignment: An example

- ***r***: GCTTAG
- ***s***: ACTTCGG

$$V(i,j) = \max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \end{cases}$$

r \ s	A	C	T	T	C	G	G	ϕ
G	1	1	-1	-2	-2	-2	-4	-6
C	1	2	0	-1	-1	-3	-3	-5
T	-1	0	1	0	-2	-2	-2	-4
T	-3	-2	-1	0	-1	-1	-1	-3
A	-3	-4	-3	-2	-1	0	0	-2
G	-5	-4	-3	-2	-1	0	1	-1
ϕ	-7	-6	-5	-4	-3	-2	-1	0

Global alignment: An example

- ***r***: GCTTAG
- ***s***: ACTTCGG

$$V(i,j) = \max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \end{cases}$$

r \ s	A	C	T	T	C	G	G	ϕ
G	1	1	-1	-2	-2	-2	-4	-6
C	1	2	0	-1	-1	-3	-3	-5
T	-1	0	1	0	-2	-2	-2	-4
T	-3	-2	-1	0	-1	-1	-1	-3
A	-3	-4	-3	-2	-1	0	0	-2
G	-5	-4	-3	-2	-1	0	1	-1
ϕ	-7	-6	-5	-4	-3	-2	-1	0

Align *r*[4..6] with *s*[6..7]:
max(0-1, -1-1, 0-1) =
max(-1, -2, -1) = -1

Global alignment: An example

- ***r***: GCTTAG
- ***s***: ACTTCGG

$$V(i,j) = \max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \end{cases}$$

r \ s	A	C	T	T	C	G	G	ϕ
G	1	1	-1	-2	-2	-2	-4	-6
C	1	2	0	-1	-1	-3	-3	-5
T	-1	0	1	0	-2	-2	-2	-4
T	-3	-2	-1	0	-1	-1	-1	-3
A	-3	-4	-3	-2	-1	0	0	-2
G	-5	-4	-3	-2	-1	0	1	-1
ϕ	-7	-6	-5	-4	-3	-2	-1	0

Align *r*[3..6] with *s*[3..7]:
max(0+1, 0-1, -1-1) =
max(1, -1, -2) = 1

Local alignment

- Find the optimal alignment for the subsequences of two sequences, *r* and *s*.
- Smith-Waterman Algorithm
- Initialization: 0 (usually)
- DP Table Construction:

$$\max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \\ \mathbf{0} \end{cases}$$

- Optimal alignment score: $max_{i,j} V(i, j)$

Local alignment: An example

- *r*: GCTTAG
- *s*: ACTTCGG

Match: +1; otherwise: -1

r \ s	A	C	T	T	C	G	G	ϕ
G								0
C								0
T								0
T								0
A								0
G								0
ϕ	0	0	0	0	0	0	0	0

Recall: $V(i, j)$ equals the optimal (i.e., highest) alignment score of all the prefixes of suffixes $r[i..m]$ and $s[j..n]$

Initialization:
zeros

Local alignment: An example

- *r*: GCTTAG
- *s*: ACTTCGG

$$\max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \\ \mathbf{0} \end{cases}$$

r \ s	A	C	T	T	C	G	G	ϕ
G								0
C								0
T								0
T								0
A								0
G							1	0
ϕ	0	0	0	0	0	0	0	0

Align *r*[6..6] with *s*[7..7]:
max(0+1, 0-1, 0-1, 0) =
max(1, -1, -1, 0) = 1

Local alignment: An example

- ***r***: GCTTAG
- ***s***: ACTTCGG

$$\max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \\ \mathbf{0} \end{cases}$$

r \ s	A	C	T	T	C	G	G	ϕ
G	2	2	0	0	0	1	1	0
C	2	3	1	0	1	0	0	0
T	0	1	2	1	0	0	0	0
T	0	0	1	1	0	0	0	0
A	1	0	0	0	0	0	0	0
G	0	0	0	0	0	1	1	0
ϕ	0	0	0	0	0	0	0	0

Local alignment: An example

- ***r***: GCTTAG
- ***s***: ACTTCGG

$$\max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \\ \mathbf{0} \end{cases}$$

r \ s	A	C	T	T	C	G	G	ϕ
G	2	2	0	0	0	1	1	0
C	2	3	1	0	1	0	0	0
T	0	1	2	1	0	0	0	0
T	0	0	1	1	0	0	0	0
A	1	0	0	0	0	0	0	0
G	0	0	0	0	0	1	1	0
ϕ	0	0	0	0	0	0	0	0

Align *r*[4..6] with *s*[6..7]:
max(0-1, 0-1, 0-1, 0) =
max(-1, -1, -1, 0) = 0

Local alignment: An example

- *r*: GCTTAG
- *s*: ACTTCGG

$$\max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \\ \mathbf{0} \end{cases}$$

r \ s	A	C	T	T	C	G	G	ϕ
G	2	2	0	0	0	1	1	0
C	2	3	1	0	1	0	0	0
T	0	1	2	1	0	0	0	0
T	0	0	1	1	0	0	0	0
A	1	0	0	0	0	0	0	0
G	0	0	0	0	0	1	1	0
ϕ	0	0	0	0	0	0	0	0

Align *r*[3..6] with *s*[3..7]:
max(1+1, 1-1, 1-1, 0) =
max(2, 0, 0, 0) = 2

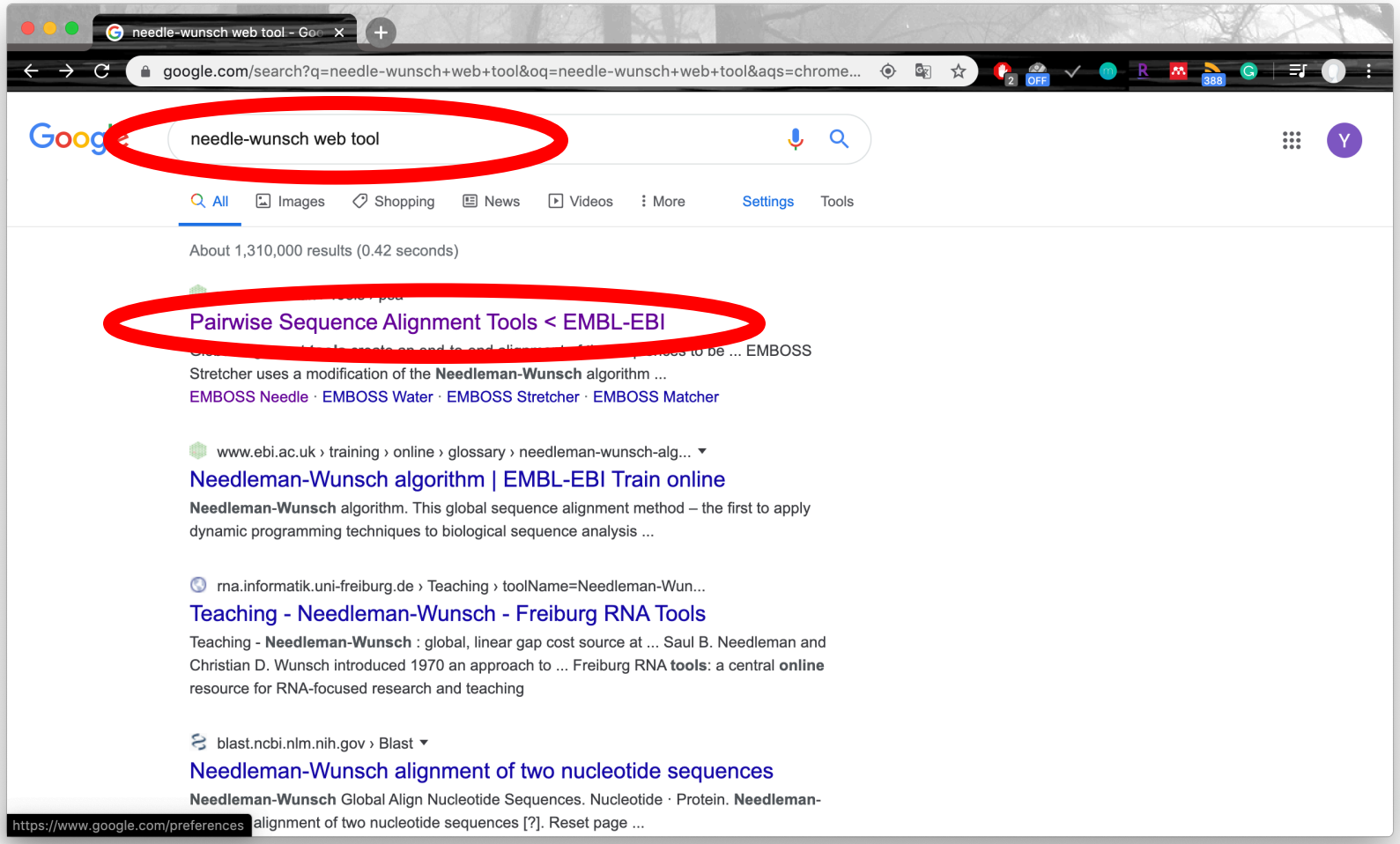
Comparison between global and local alignment

	Global Alignment	Local Alignment
Description	Alignment of 2 or more sequences	Alignment of **subsequences** of 2 or more sequences
Application	Whole genome sequence alignment	Find motifs, domains
Initialization	According to the gap penalty	Zeros (usually)
DP table, V(*i*, *j*) =	$\max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \end{cases}$	$\max \begin{cases} V(i+1, j+1) + \sigma(r[i], s[j]) \\ V(i, j+1) + \sigma('\_', s[j]) \\ V(i+1, j) + \sigma(r[i], '\_') \\ \mathbf{0} \end{cases}$
Optimal alignment score	Top left corner**,** i.e., ***V*(1, 1)**	The **largest** value

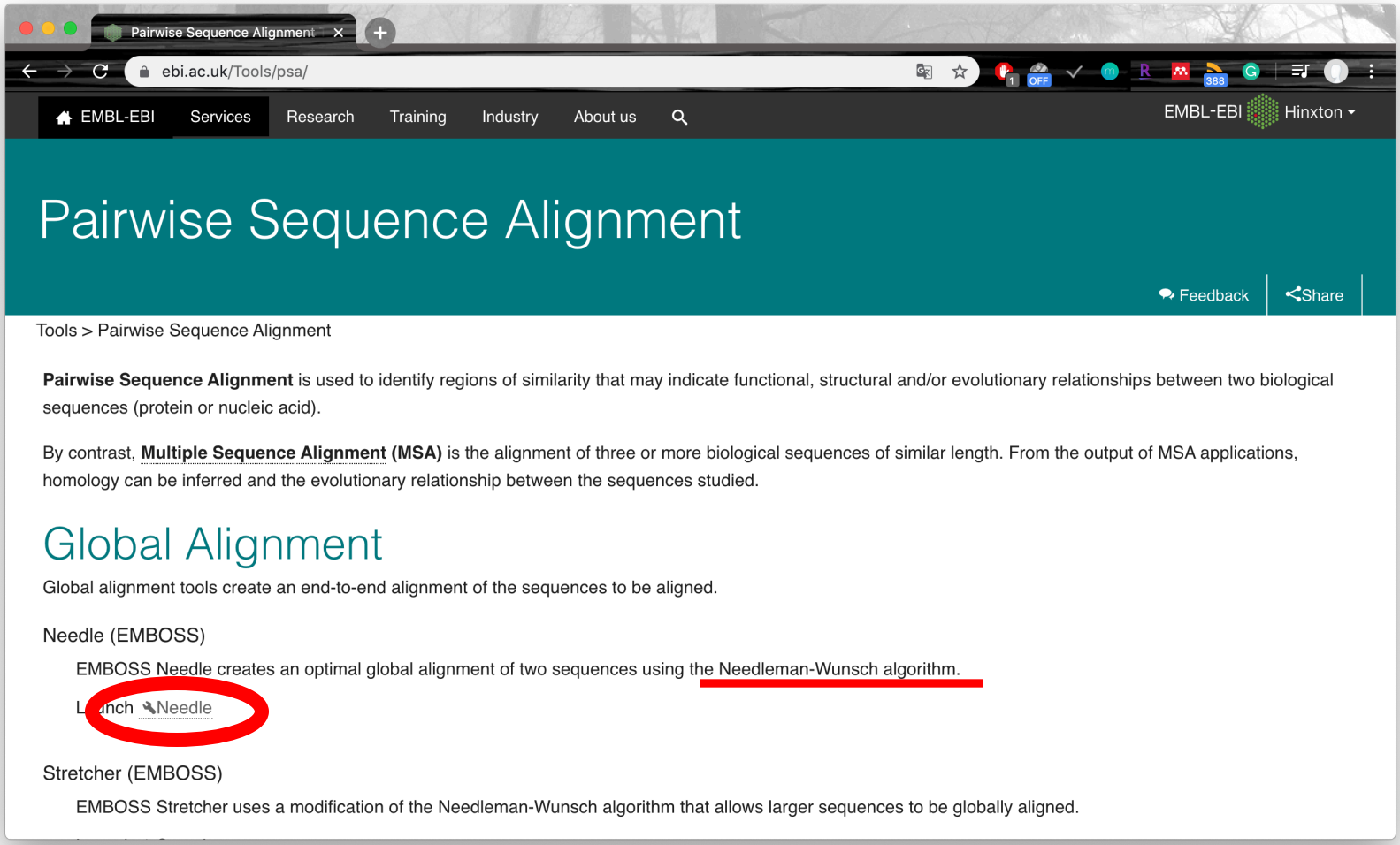
Demonstration: Web tools for alignment

- Find a Web tool to perform Needleman-Wunsch optimal global alignment on the sequences below:
  - `GCTTAG`
  - `ACTTCGG`

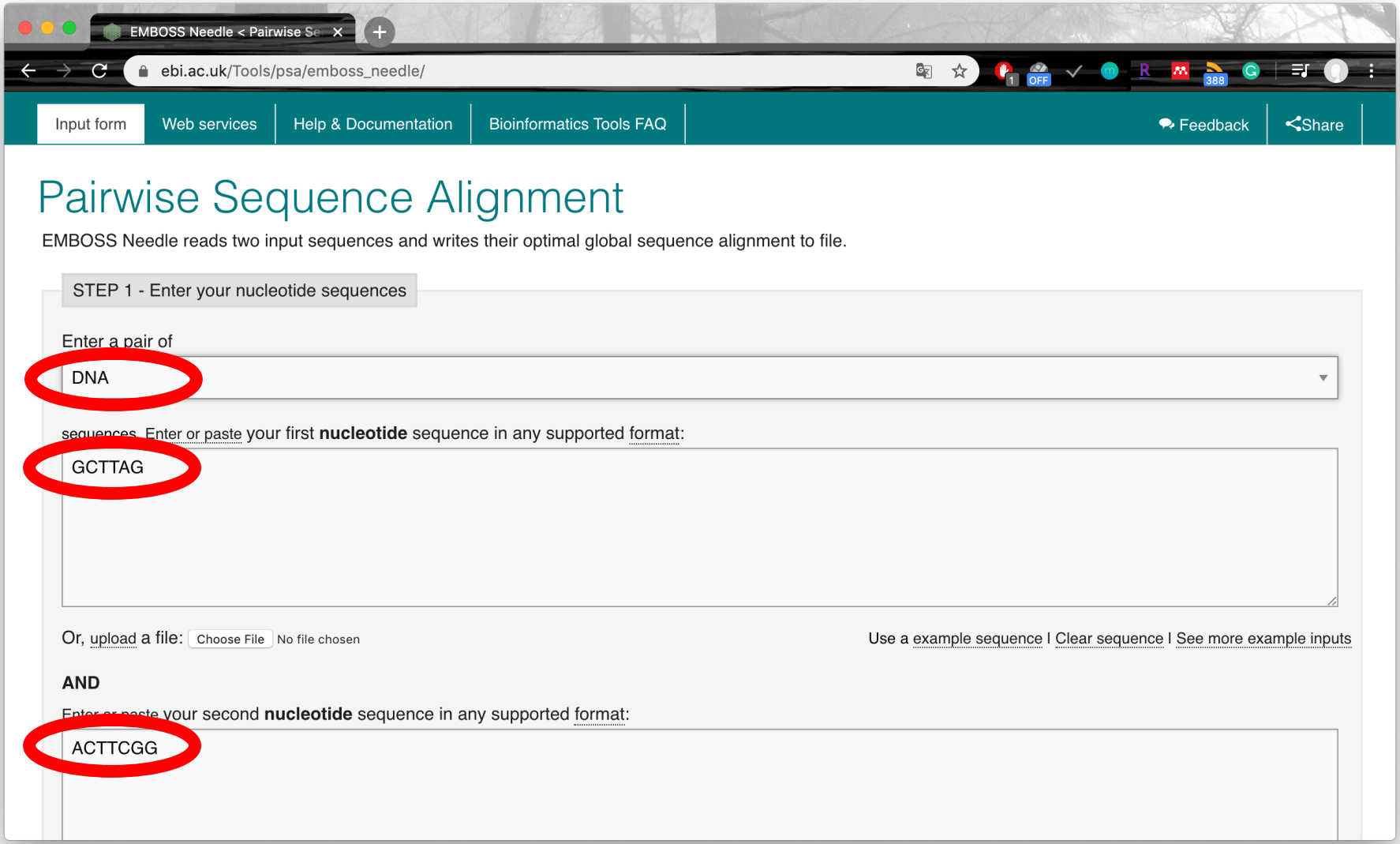
Step 1: Find tools on Google



Step 2: Select the tool

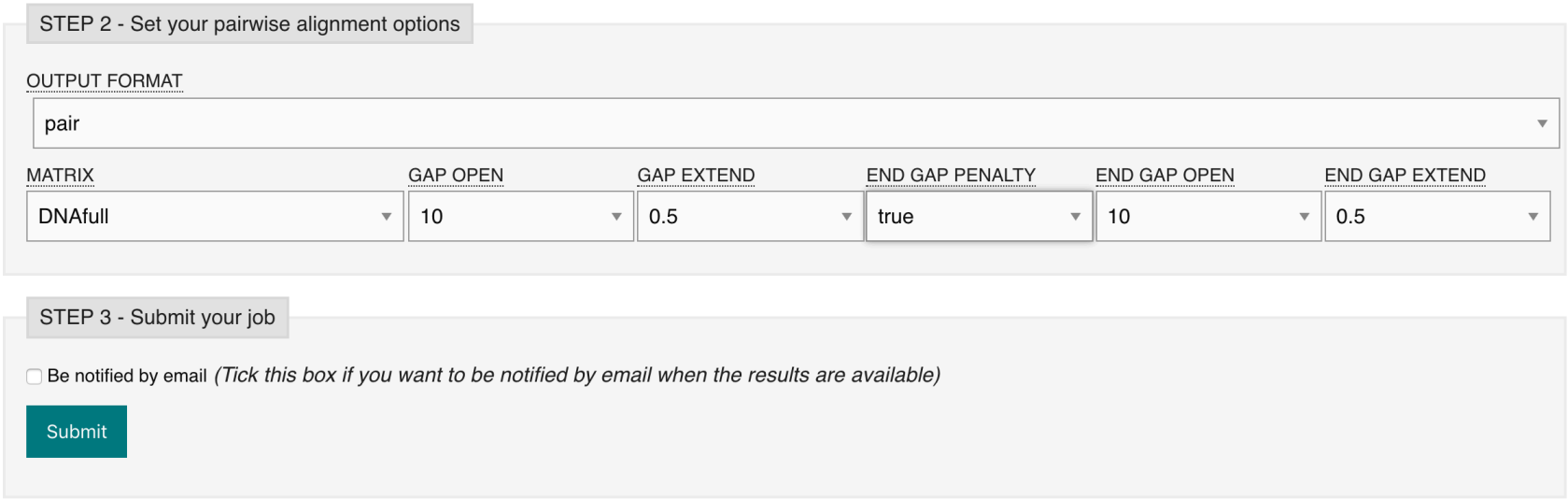


Step 3: Inputs

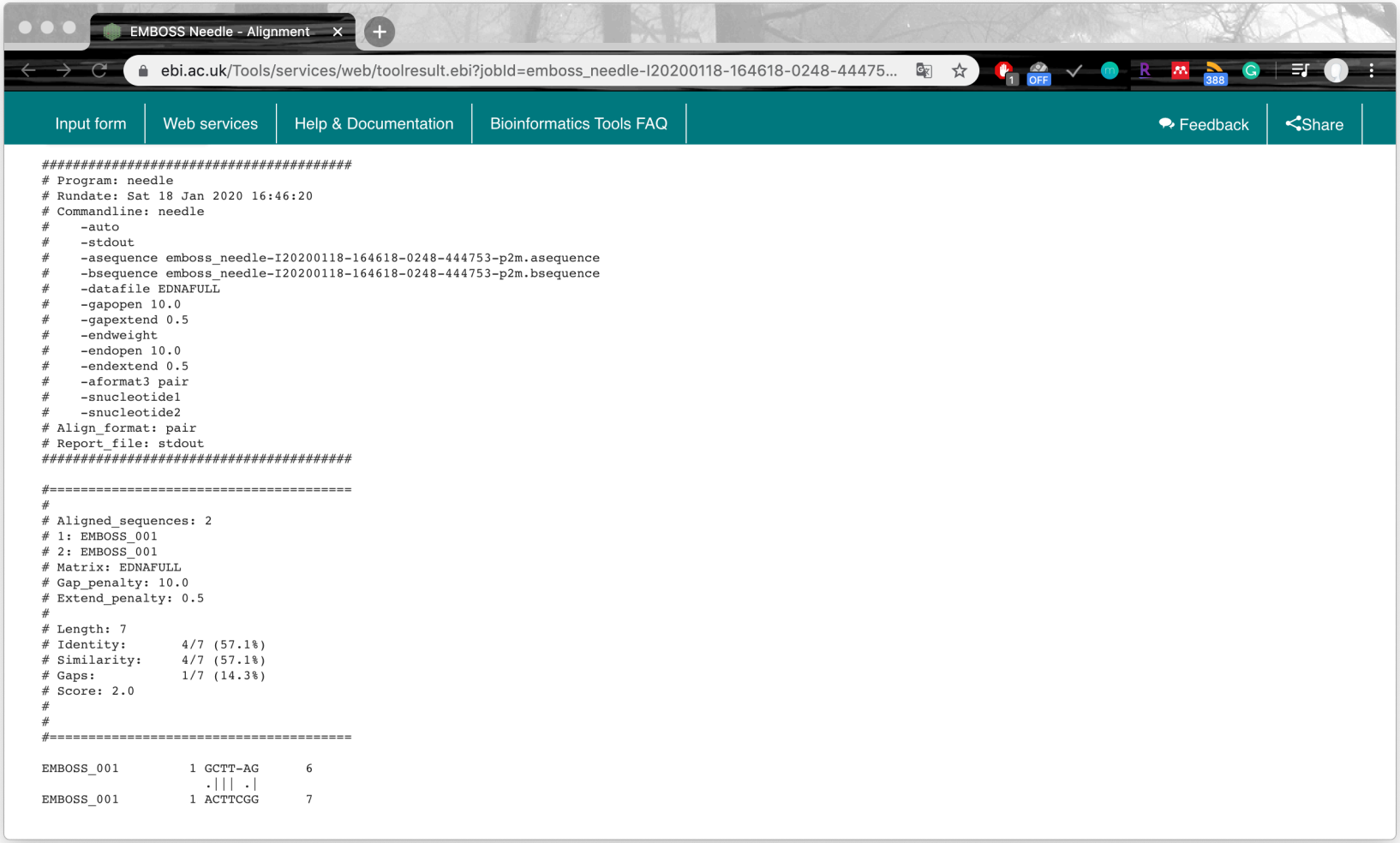


Step 4: Parameters

- End gap penalty: true
- Others: default settings
- Note: match: +5; mismatch: -4



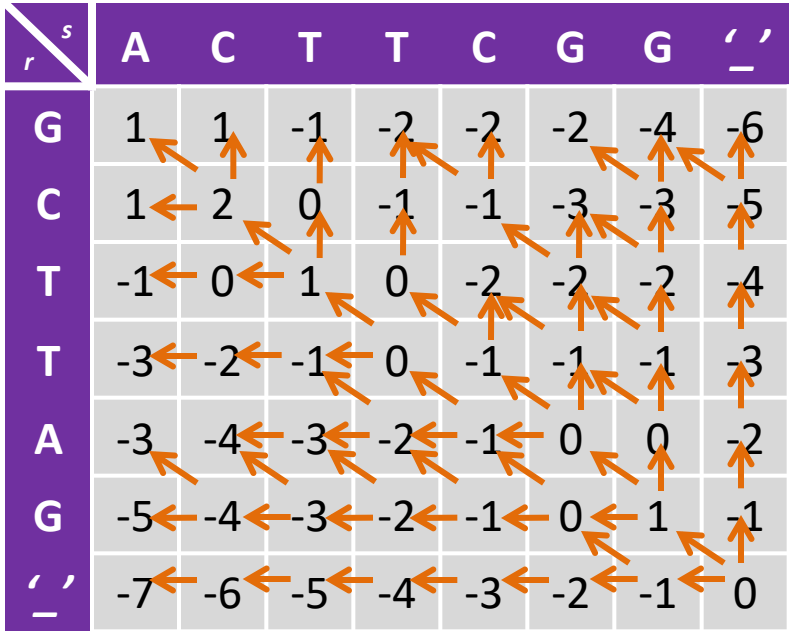
Step 5: Show the results



Exercise 1: Pairwise sequence alignment

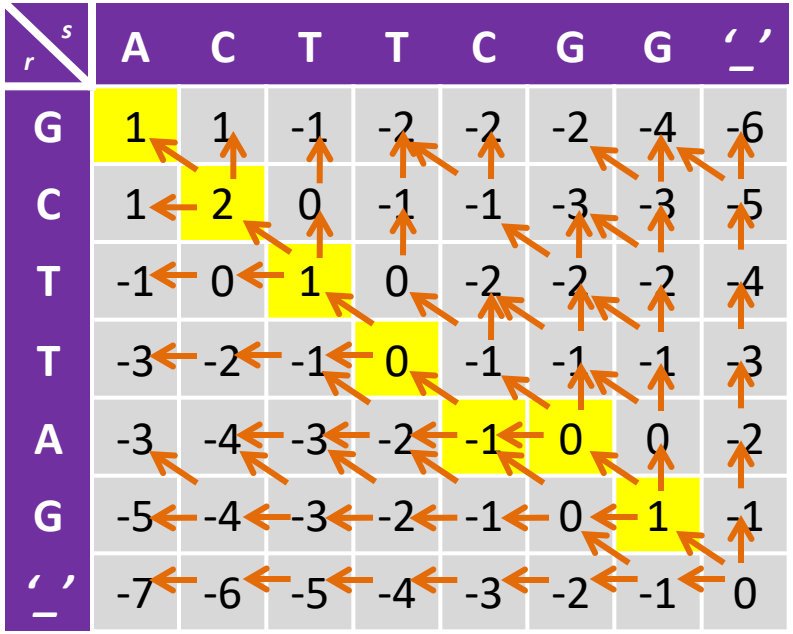
- Find the optimal **global** alignment(s) and **local** alignment(s) for the following sequences and find the alignment score for each optimal alignment.
  - *r*: `GCTTAG`
  - *s*: `ACTTCGG`
- Match: +1; otherwise: -1

Answer: optimal global alignment



r \ s	A	C	T	T	C	G	G	'_'
G	1	1	-1	-2	-2	-2	-4	-6
C	1	2	0	-1	-1	-3	-3	-5
T	-1	0	1	0	-2	-2	-2	-4
T	-3	-2	-1	0	-1	-1	-1	-3
A	-3	-4	-3	-2	-1	0	0	-2
G	-5	-4	-3	-2	-1	0	1	-1
'_'	-7	-6	-5	-4	-3	-2	-1	0

Answer: optimal global alignment



r \ s	A	C	T	T	C	G	G	'_'
G	1	1	-1	-2	-2	-2	-4	-6
C	1	2	0	-1	-1	-3	-3	-5
T	-1	0	1	0	-2	-2	-2	-4
T	-3	-2	-1	0	-1	-1	-1	-3
A	-3	-4	-3	-2	-1	0	0	-2
G	-5	-4	-3	-2	-1	0	1	-1
'_'	-7	-6	-5	-4	-3	-2	-1	0

Best alignment score: 1

Best alignment 1:

r: `GCTT_AG`

s: `ACTTCGG`

Answer: optimal global alignment

r \ s	A	C	T	T	C	G	G	'_'
G	1	1	-1	-2	-2	-2	-4	-6
C	1	2	0	-1	-1	-3	-3	-5
T	-1	0	1	0	-2	-2	-2	-4
T	-3	-2	-1	0	-1	-1	-1	-3
A	-3	-4	-3	-2	-1	0	0	-2
G	-5	-4	-3	-2	-1	0	1	-1
'_'	-7	-6	-5	-4	-3	-2	-1	0

Best alignment score: 1

Best alignment 2:

r: GCTTA_G

s: ACTTCGG

Answer: optimal global alignment

r \ s	A	C	T	T	C	G	G	'_'
G	1	1	-1	-2	-2	-2	-4	-6
C	1	2	0	-1	-1	-3	-3	-5
T	-1	0	1	0	-2	-2	-2	-4
T	-3	-2	-1	0	-1	-1	-1	-3
A	-3	-4	-3	-2	-1	0	0	-2
G	-5	-4	-3	-2	-1	0	1	-1
'_'	-7	-6	-5	-4	-3	-2	-1	0

Best alignment score: 1

Best alignment 3:

r: GCTTAG_

s: ACTTCGG

Answer: optimal local alignment

r \ s	A	C	T	T	C	G	G	'_'
G	2	2	0	0	0	1	1	0
C	2	3	1	0	1	0	0	0
T	0	1	2	1	0	0	0	0
T	0	0	1	1	0	0	0	0
A	1	0	0	0	0	0	0	0
G	0	0	0	0	0	1	1	0
'_'	0	0	0	0	0	0	0	0

Best alignment score: 3

Best alignment 1:

r: `CTT`

s: `CTT`

Best alignment 2:

r: `CTTAG`

s: `CTTCG`

Scoring matrix

- Give higher score for more similar characters
- Handle gaps more properly
- Always be **symmetric**

	A	C	G	T	φ
A	1	-1	-1	-1	-1
C	-1	1	-1	-1	-1
G	-1	-1	1	-1	-1
T	-1	-1	-1	1	-1
φ	-1	-1	-1	-1	NA

	A	C	G	T	φ
A	5	-3	-2	-3	-7
C	-3	6	-3	-2	-5
G	-2	-3	6	-3	-7
T	-3	-2	-3	5	-5
φ	-7	-5	-7	-5	NA

More about scoring matrix

- Different matrices for DNA sequence alignment and protein sequence alignment
- For similarity of DNA, consider *transition* and *transversion*
- For similarity of amino acid, consider *charge, hydrophobicity* and *size* of amino acid
- Affine gap penalty (separate costs for gap opening and gap size)

$$y = -a - bx$$

$-a$: Gap opening penalty
$-b$: Gap size penalty

Exercise 2: Scoring matrix and alignment

- A) Suppose the mismatch penalties for transition and transversion are -1 and -2 respectively, the indel penalty is -3, and the score of a match is 2. Write down the scoring matrix.

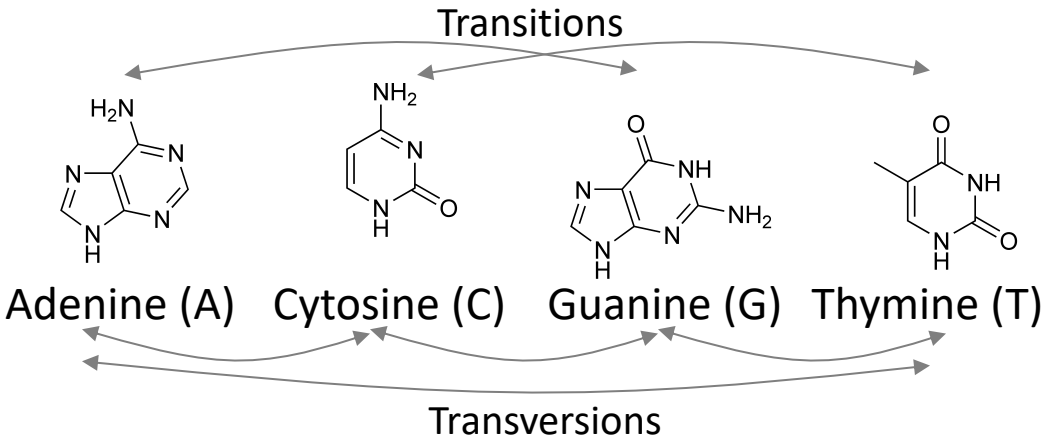


Image credit: Wikipedia

Answer: Scoring matrix

	A	C	G	T	φ
A	2	-2	-1	-2	-3
C	-2	2	-2	-1	-3
G	-1	-2	2	-2	-3
T	-2	-1	-2	2	-3
φ	-3	-3	-3	-3	NA

Exercise 2: Scoring matrix and alignment

- B) Using the previous scoring matrix, find the best global alignment of the following sequences:
  - TGCATAC
  - TACTGATG

	A	C	G	T	φ
A	2	-2	-1	-2	-3
C	-2	2	-2	-1	-3
G	-1	-2	2	-2	-3
T	-2	-1	-2	2	-3
φ	-3	-3	-3	-3	NA

Exercise 2: Scoring matrix and alignment

r \ s	T	A	C	T	G	A	T	G	φ
T	-3	-5	-2	1	-4	-9	-11	-16	-21
G	-8	-5	-6	-4	-1	-6	-11	-13	-18
C	-5	-7	-4	-4	-2	-3	-8	-13	-15
A	-7	-4	-5	-6	-3	0	-5	-10	-12
T	-9	-9	-6	-3	-4	-5	-2	-7	-9
A	-14	-11	-11	-8	-5	-2	-4	-4	-6
C	-19	-16	-13	-13	-10	-7	-4	-2	-3
φ	-24	-21	-18	-15	-12	-9	-6	-3	0

	A	C	G	T	φ
A	2	-2	-1	-2	-3
C	-2	2	-2	-1	-3
G	-1	-2	2	-2	-3
T	-2	-1	-2	2	-3
φ	-3	-3	-3	-3	NA

Best alignment score: 3

r: TGCATAC_
s: TACTGATG

r: TGC__ATAC
s: TACTGATG_

r: TGCAT_AC_
s: TAC_TGATG

Exercise 3: Affine gap penalty

- You are provided with a scoring matrix. Given that gap opening penalty = -5 and gap size penalty = -1, calculate the alignment score for the following alignments:

```
AACCTT
AAG_TT
```
4*(5)+1*(-3)+1*(-5)+1*(-1)
=11

```
ACGGCTTCGGCA
AC__CTTC_ACA
```
8*(5)+1*(-2)+2*(-5)+3*(-1)
=25

```
CTCTCGGGGGGAACC
CCCTC_______AACC
```
8*(5)+1*(-2)+1*(-5)+6*(-1)
=27

	A	C	G	T
A	5	-3	-2	-3
C	-3	5	-3	-2
G	-2	-3	5	-3
T	-3	-2	-3	5

Check list

- What are two main ideas behind dynamic programing?
- What are the considerations in proposing the scoring rules in sequence alignment problem?
- What is affine gap penalty in solving sequence alignment problem?
- Are there any negative numbers in the D.P. table of the local alignment? Why?
- (Optional) Deduce the number of possible alignments of a pair of sequences with length m and n.
- (Optional) How can you perform optimal alignment for three sequences?