# International Bioinformatics Workshop on Molecular Epidemiology and Phylodynamics

**Yu-Nong Gong, Ph.D.**

Assistant Professor

2024-04-26

**RCEVI**

RESEARCH CENTER FOR EMERGING VIRAL INFECTIONS

長庚大學
Chang Gung University

# Multiple Sequence Alignment

**Learning Objectives**

- Understand the algorithms behind multiple sequence alignment.

- Perform pairwise alignment and multiple sequence alignments using tools

  - EMBOSS (needle and water)

  - MAFFT

# Outline

1. Homology v.s. Similarity

2. Global Alignment v.s. Local Alignment

3. Pairwise Alignment v.s. Multiple Sequence Alignment

4. Alignment methods

   - Longest common subsequence

   - Time Complexity

5. Sum of pairs for evaluating an alignment

6. Hands-on session

# Homology v.s. Similarity



Homology vs Similarity in Bioinformatics

More Information Online    WWW.DIFFERENCEBETWEEN.COM

| | Homology in Bioinformatics | Similarity in Bioinformatics |
|---|---|---|
| DEFINITION | Homology refers to a statement about common evolutionary ancestry of two sequences | Similarity refers to the degree of likeness between two sequences |
| SIGNIFICANCE | Cannot be calculated since it could be true or false and usually depend on the hypothesis | Can be easily calculated as a percentage of similar residues over a given length of the alignment |

https://pin.it/76TSRgW

# Global Alignment v.s. local Alignment

| Global Alignment | Local Alignmennt |
| --- | --- |
| Purpose: Align the entire sequences | Find local regions with high similarity |
| Contains all nucletoides (or amino acids) | Align a substring of query to a substring of target |
| If two sequences are quite similar | Find stretches of sequences without considering the rest of sequence |
| Suitalbe for aligning two closely related sequences | Suitable for aligning more divergent sequences |
| Needleman-Wunsch algorithm (EMBOSS neelde) | Smith-Waterman algorithm (EMBOSS water) |

https://www.majordifferences.com/2016/05/difference-between-global-and-local.html#.VzizhtJ96M8

# Local Alignment

Target Sequence

```
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
             ||||  |||||||  ||||||||||||||||||
Query Sequence 5' TACTCACGGATGAGGTACTTTAGAGGC 3'
```

# Global Alignment

Target Sequence

```
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
   |||||||||||||        |||||||   |||||||||||||||| ||||||||
5' ACTACTAGATT----ACGGATC--GTACTTTAGAGGCTAGCAACCA 3'
```

Query Sequence

# Multiple Sequence Alignment



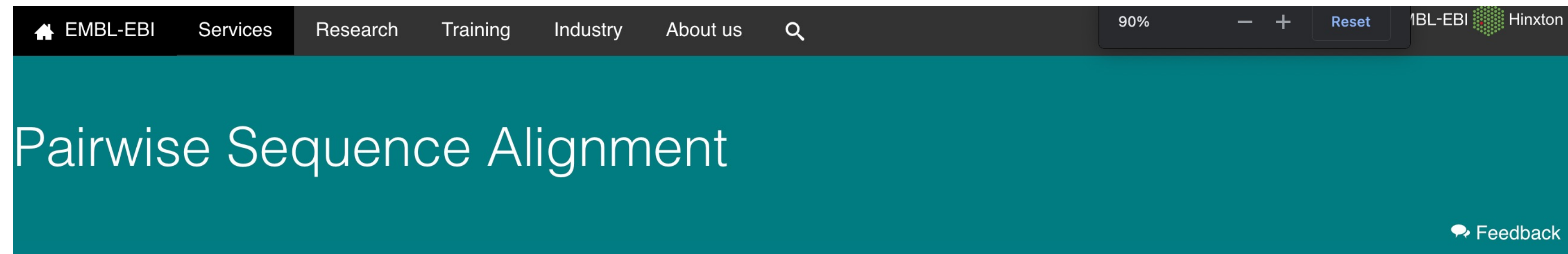https://en.wikipedia.org/wiki/File:RPLP0_90_ClustalW_aln.gif
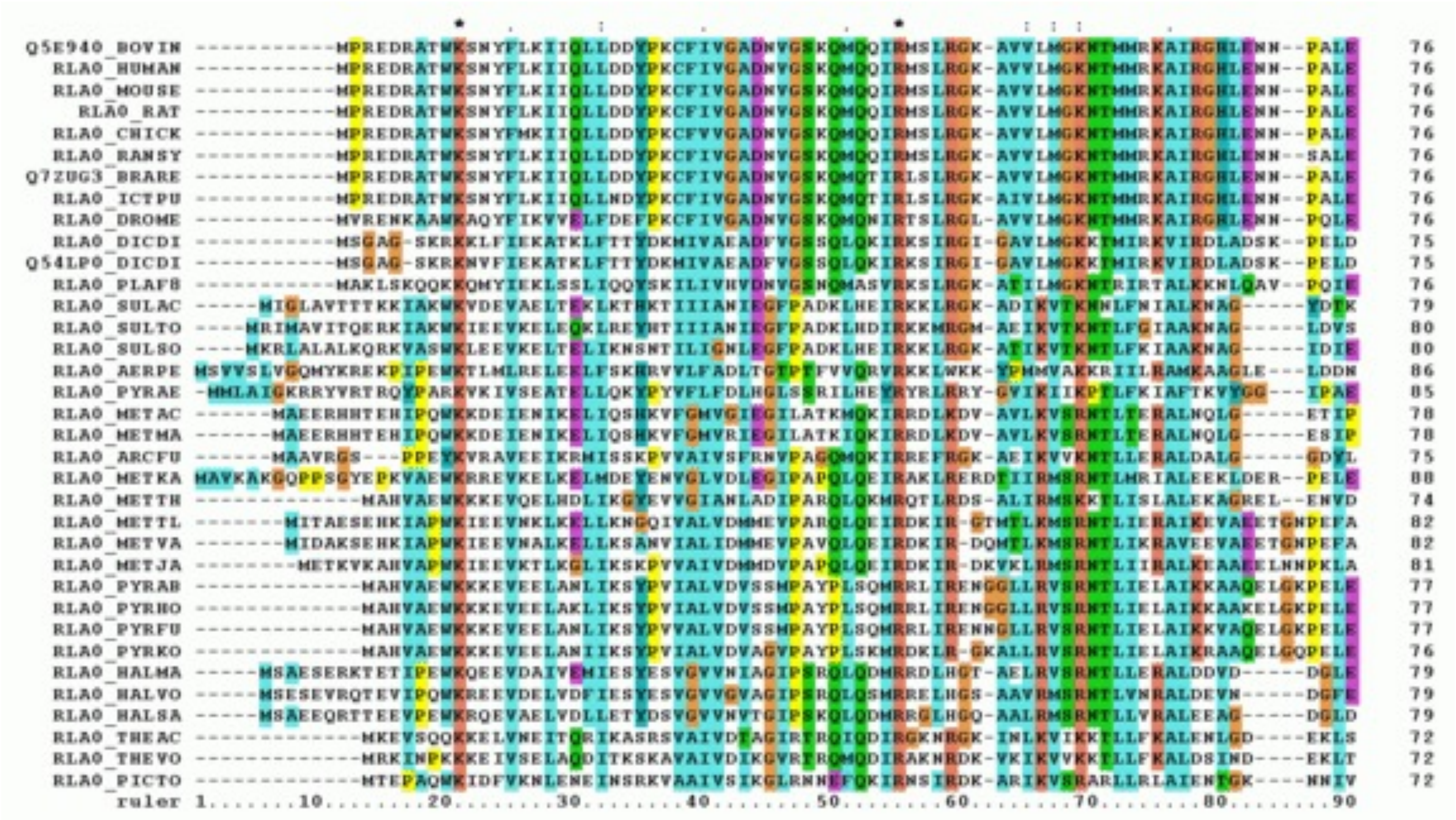
# Pairwise Alignment v.s. Multiple Sequence Alignment

| Pairwise Alignment | Multiple Sequence Alignmennt |
|---|---|
| two sequences | Three or more |
| Global or local alignment | Global |
| Comparatively simple algorithm | Complex sophisticated algorithm |
| Needleman-Wunsch algorithm (EMBOSS neelde), Smith-Waterman algorithm (EMBOSS water), and so on | Progressive alignment construction, Iterative methods, and so on |
| Applications: (1) find out conserved regions between two sequences; (2) similarity searches in a database | Applications: (1) find out regions of variability or conservation in a family; (2) similarity searches between a newly sequence and an existing gene family; (3) phylogenetic analysis |

# Online tools

## https://www.ebi.ac.uk/Tools/psa/; https://www.ebi.ac.uk/Tools/msa/

EMBL-EBI    Services    Research    Training    Industry    About us    🔍    90%    —  +    Reset    ...BL-EBI 🔷 Hinxton ▾

# Pairwise Sequence Alignment

💬 Feedback

Tools > Pairwise Sequence Alignment

### Service Announcement

The new Job Dispatcher Services beta website is now available at https://wwwdev.ebi.ac.uk/Tools/jdispatcher. We'd love to hear your feedback about the new webpages!

**Pairwise Sequence Alignment** is used to identify regions of similarity that may indicate functional, structural and/or evolutionary relationships between two biological sequences (protein or nucleic acid).

By contrast, **Multiple Sequence Alignment (MSA)** is the alignment of three or more biological sequences of similar length. From the output of MSA applications, homology can be inferred and the evolutionary relationship between the sequences studied.

# Global Alignment

Global alignment tools create an end-to-end alignment of the sequences to be aligned.

## Needle (EMBOSS)

EMBOSS Needle creates an optimal global alignment of two sequences using the Needleman-Wunsch algorithm.

Launch ✎Needle

## Stretcher (EMBOSS)

EMBOSS Stretcher uses a modification of the Needleman-Wunsch algorithm that allows larger sequences to be globally aligned.

Launch ✎Stretcher

## GGSEARCH2SEQ

GGSEARCH2SEQ finds an optimal global alignment using the Needleman-Wunsch algorithm.

Launch ✎ggsearch2seq

# Local Alignment

Local alignment tools find one, or more, alignments describing the most similar region(s) within the sequences to be aligned. They are can align protein and nucleotide sequences.

## Water (EMBOSS)

EMBOSS Water uses the Smith-Waterman algorithm (modified for speed enhancements) to calculate the local alignment of two sequences.

Launch ⚒Water

## Matcher (EMBOSS)

EMBOSS Matcher identifies local similarities between two sequences using a rigorous algorithm based on the LALIGN application.

Launch ⚒Matcher

## LALIGN

LALIGN finds internal duplications by calculating non-intersecting local alignments of protein or DNA sequences.

Launch ⚒LALIGN

## SSEARCH2SEQ

SSEARCH2SEQ finds an optimal local alignment using the Smith-Waterman algorithm.

Launch ⚒ssearch2seq

# Genomic Alignment

Genomic alignment tools concentrate on DNA (or to DNA) alignments while accounting for characteristics present in genomic data.

GeneWise

GeneWise compares a protein sequence to a genomic DNA sequence, allowing for introns and frameshifting errors.

🔧Launch GeneWise

The tools described on this page are provided using **Search and sequence analysis tools services from EMBL-EBI in 2022**

Please read the provided Help & Documentation and FAQs before seeking help from our support staff. If you have any feedback or encountered any issues please let us know via EMBL-EBI Support. If you plan to use these services during a course please contact us. Read our Privacy Notice if you are concerned with your privacy and how we handle personal information.

# Multiple Sequence Alignment

Tools > Multiple Sequence Alignment

## Service Announcement

The new Job Dispatcher Services beta website is now available at https://wwwdev.ebi.ac.uk/Tools/jdispatcher. We'd love to hear your feedback about the new webpages!

**Multiple Sequence Alignment (MSA)** is generally the alignment of three or more biological sequences (protein or nucleic acid) of similar length. From the output, homology can be inferred and the evolutionary relationships between the sequences studied.

By contrast, **Pairwise Sequence Alignment** tools are used to identify regions of similarity that may indicate functional, structural and/or evolutionary relationships between two biological sequences.

## Clustal Omega

New MSA tool that uses seeded guide trees and HMM profile-profile techniques to generate alignments. Suitable for medium-large alignments.

🔧Launch Clustal Omega

## Cons (EMBOSS)

EMBOSS Cons creates a consensus sequence from a protein or nucleotide multiple alignment.

🔧Launch EMBOSS Cons

## Clustal Omega

New MSA tool that uses seeded guide trees and HMM profile-profile techniques to generate alignments. Suitable for medium-large alignments.

🔧Launch Clustal Omega

## Cons (EMBOSS)

EMBOSS Cons creates a consensus sequence from a protein or nucleotide multiple alignment.

🔧Launch EMBOSS Cons

## Kalign

Very fast MSA tool that concentrates on local regions. Suitable for large alignments.

🔧Launch Kalign

## MAFFT

MSA tool that uses Fast Fourier Transforms. Suitable for medium-large alignments.

🔧Launch MAFFT

## MUSCLE

Accurate MSA tool, especially good with proteins. Suitable for medium alignments.

🔧Launch MUSCLE

## MView

Transform a Sequence Similarity Search result into a Multiple Sequence Alignment or reformat a Multiple Sequence Alignment using the MView program.

🔧Launch MView

## T-Coffee

Consistency-based MSA tool that attempts to mitigate the pitfalls of progressive alignment methods. Suitable for small alignments.

🔧Launch T-Coffee

## WebPRANK

The EBI has a new phylogeny-aware multiple sequence alignment program which makes use of evolutionary information to help place insertions and deletions.
Try it out at WebPRANK.

# Alignment methods

# Longest Common Subsequence

To find a longest common subsequence between two strings.

```
S1:  TAGTCACG

S2:  AGACTGTC

LCS: AGACG
```

# Dynamic Programming

a computer programming technique where an algorithmic problem is **first broken down into sub-problems, the results are saved, and then the sub-problems are optimized to find the overall solution** — which usually has to do with **finding the maximum and minimum range of the algorithmic query.**

https://www.spiceworks.com/tech/devops/articles/what-is-dynamic-programming/

## Dynamic programming:

$$c_{i,j} = \max \begin{cases} c_{i-1,j-1} + 1 & if \ a_i = b_j \\ c_{i-1,j} + 0 & if \ a_i \neq b_j \\ c_{i,j-1} + 0 & if \ a_i \neq b_j \end{cases}$$

## Step 1

|  | S2 | A | G | A | C | T | G | T | C |
|---|---|---|---|---|---|---|---|---|---|
| S1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T | 0 |  |  |  |  |  |  |  |  |
| A | 0 |  |  |  |  |  |  |  |  |
| G | 0 |  |  |  |  |  |  |  |  |
| T | 0 |  |  |  |  |  |  |  |  |
| C | 0 |  |  |  |  |  |  |  |  |
| A | 0 |  |  |  |  |  |  |  |  |
| C | 0 |  |  |  |  |  |  |  |  |
| G | 0 |  |  |  |  |  |  |  |  |

## Step 2

|  | S2 | A | G | A | C | T | G | T | C |
|---|---|---|---|---|---|---|---|---|---|
| S1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| A | 0 |  |  |  |  |  |  |  |  |
| G | 0 |  |  |  |  |  |  |  |  |
| T | 0 |  |  |  |  |  |  |  |  |
| C | 0 |  |  |  |  |  |  |  |  |
| A | 0 |  |  |  |  |  |  |  |  |
| C | 0 |  |  |  |  |  |  |  |  |
| G | 0 |  |  |  |  |  |  |  |  |

<u>Dynamic programming</u>:

$$c_{i,j} = \max \begin{cases} c_{i-1,j-1} + 1 & if \ a_i = b_j \\ c_{i-1,j} + 0 & if \ a_i \neq b_j \\ c_{i,j-1} + 0 & if \ a_i \neq b_j \end{cases}$$

<u>Step 3</u>

| | S2 | A | G | A | C | T | G | T | C |
|---|---|---|---|---|---|---|---|---|---|
| S1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| A | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G | 0 | | | | | | | | |
| T | 0 | | | | | | | | |
| C | 0 | | | | | | | | |
| A | 0 | | | | | | | | |
| C | 0 | | | | | | | | |
| G | 0 | | | | | | | | |

<u>Step 4</u>

| | S2 | A | G | A | C | T | G | T | C |
|---|---|---|---|---|---|---|---|---|---|
| S1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| A | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| T | 0 | | | | | | | | |
| C | 0 | | | | | | | | |
| A | 0 | | | | | | | | |
| C | 0 | | | | | | | | |
| G | 0 | | | | | | | | |

$$c_{i,j} = \max \begin{cases} c_{i-1,j-1} + 1 & if \ a_i = b_j \\ c_{i-1,j} + 0 & if \ a_i \neq b_j \\ c_{i,j-1} + 0 & if \ a_i \neq b_j \end{cases}$$

## Step 5

|    | S2 | A | G | A | C | T | G | T | C |
|----|----|----|----|----|----|----|----|----|----|
| S1 | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T  | 0  | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| A  | 0  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G  | 0  | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| T  | 0  | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| C  | 0  |   |   |   |   |   |   |   |   |
| A  | 0  |   |   |   |   |   |   |   |   |
| C  | 0  |   |   |   |   |   |   |   |   |
| G  | 0  |   |   |   |   |   |   |   |   |

...

$$c_{i,j} = \max \begin{cases} c_{i-1,j-1} + 1 & if \ a_i = b_j \\ c_{i-1,j} + 0 & if \ a_i \neq b_j \\ c_{i,j-1} + 0 & if \ a_i \neq b_j \end{cases}$$

S1:T**AG**TC**ACG**

S2:**AGAC**T**G**TC

**LCS:AGACG**

**Alignment**

T**AG**TC**AC-G--**

**-AG--AC**T**G**TC
...

Final

| | S2 | A | G | A | C | T | G | T | C |
|---|---|---|---|---|---|---|---|---|---|
| S1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T | **0** | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| A | 0 | **1** | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G | 0 | 1 | **2** | 2 | 2 | 2 | 2 | 2 | 2 |
| T | 0 | 1 | **2** | 2 | 2 | 3 | 3 | 3 | 3 |
| C | 0 | 1 | **2** | 2 | 3 | 3 | 3 | 3 | 4 |
| A | 0 | 1 | 2 | **3** | 3 | 3 | 3 | 3 | 4 |
| C | 0 | 1 | 2 | 3 | **4** | **4** | 4 | 4 | 4 |
| G | 0 | 1 | 2 | 3 | 4 | 4 | **5** | **5** | **5** |

# Equation

$$c_{i,j} = \max \begin{cases} c_{i-1,j-1} + 1 & \text{if } a_i = b_j \\ c_{i-1,j} + 0 & \text{if } a_i \neq b_j \\ c_{i,j-1} + 0 & \text{if } a_i \neq b_j \end{cases}$$

# Go to this website and input sequences

```cpp
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int **table;
6  string s1,s2;
7
8  void find_LCS()
9  {
10     int i,j,n,m;
11     char x;
12     n = s1.length();
13     m = s2.length();
14     // 宣告 2D-array by size of 2 strings
15     table = new int*[n+1];
16     for (int i=0;i<=n;i++) table[i] = new int[m+1];
17     // 將第一排第一列都先歸零
18     for (int i=0;i<=n;i++) table[i][0] = 0;
19     cout<<"str2: ";
20     for (int j=0;j<=m;j++)
21     {
22         table[0][j] = 0;
23         if (j==0) cout<<"___"; else cout<<setw(3)<<s2[j-1];    //*****顯示比較字
24     }
25     cout<<endl<<"str1 |"; for (int j=0;j<=m;j++) cout<<"  0";   //*****顯示比較字
26     cout<<endl;                                                //*****顯示比較字
27
```

```
Enter String1: TAGTCACG
Enter String2: AGACTGTC
str2:  ___   A  G  A  C  T  G  T  C
str1 |   0   0  0  0  0  0  0  0  0
   T   0  ^0 ^0 ^0 ^0 +1 <1 +1 <1
   A   0  +1 <1 +1 <1 ^1 ^1 ^1 ^1
   G   0  ^1 +2 <2 <2 <2 +2 <2 <2
   T   0  ^1 ^2 ^2 ^2 +3 <3 +3 <3
   C   0  ^1 ^2 ^2 +3 ^3 ^3 ^3 +4
   A   0  +1 ^2 +3 ^3 ^3 ^3 ^3 ^4
   C   0  ^1 ^2 ^3 +4 <4 <4 <4 +4
   G   0  ^1 +2 ^3 ^4 ^4 +5 <5 <5
```

**https://www.onlinegdb.com/Hkx1yiw2v**

# Time Complexity

The number of sequences is *k*

K-dimensional table

Each dimension is sequence length

Each entry depends on $2^k$-1 adjacent entries

Complexity: $O(2^k n^k)$

NP (nondeterministic polynomial time)-complete problem

$$c_{i,j} = \max \begin{cases} c_{i-1,j-1} + 1 & if\ a_i = b_j \\ c_{i-1,j} + 0 & if\ a_i \neq b_j \\ c_{i,j-1} + 0 & if\ a_i \neq b_j \end{cases}$$

# Evaluating an alignment: Sum-of-pairs

## Alignment #1

$$\sum_{i<j} scoring(S_i, S_j)$$

$s_1 = \text{ATTCGA}\textcolor{red}{\text{T}}$

$s_2 = \text{-TT-GA}\textcolor{red}{\text{G}}$

$s_3 = \text{AT--GC}\textcolor{red}{\text{T}}$

- $s(S_i$ and $S_j) = 1$ when match
- $s(S_i$ and $S_j) = -1$ when mismatch
- $s(S_i$ and $S_j) = -2$ when gap

For the alignment, the pairwise alignment score of position 7:

score($s_1$,$s_2$) = -1

score($s_2$,$s_3$) = -1

score($s_1$,$s_3$) = 1 $\Rightarrow$ SP score = -1

## Alignment #2

$$\sum_{i<j} scoring(S_i, S_j)$$

$s_1 = $ ATTCGAT-

$s_2 = $ -TT-GA-G

$s_3 = $ AT--GCT-

- $s(S_i$ and $S_j) = 1$ when match
- $s(S_i$ and $S_j) = -1$ when mismatch
- $s(S_i$ and $S_j) = -2$ when gap

For the alignment, the pairwise alignment score of position 7 and 8:

score($s_1$,$s_2$) = -2-2

score($s_2$,$s_3$) = -2+1

score($s_1$,$s_3$) = 1+1 $\Rightarrow$ SP score = -3

# Hands-on session

>1

ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA


>2

TACTCACGGATGAGGTACTTTAGAGGC


1. Copy and save as two sequence files (1.fas and 2.fas) (using Sublime app)
2. Install EMBOSS by using conda (https://anaconda.org/bioconda/emboss)
3. Perfrom Needle and Water to compare global and local alignment
   $ water 1.fas 2.fas 1-2.water –auto
   $ needle 1.fas 2.fas 1-2.needle -auto