Pairwise and Multiple Sequence Alignment

Why sequence alignment?

Evolutionary aspects of bioinformatics deals with the information transfer and subsequent modification of the information in due course of time, rather than the transfer of information from one component of the cell to another

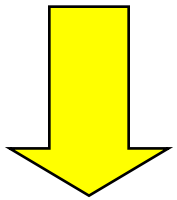
Why perform a pairwise sequence alignment?

Finding homology between two sequences

e.g., The wings of bats and arms of humans are homologous, *ie*. The structures were evolved from common ancestors.

In genetics, homology is used in reference to protein or DNA sequences, meaning that the given sequences shares the a common ancestry.

similar sequence (or structure)



similar function

What is sequence alignment?

Alignment: Comparing two (pairwise) or more (multiple) sequences.

Searching for a series of identical or similar characters in the sequences.

```
MVNLTSDEKTAVLALWNKVDVEDCGGE
||  ||   |||||  |||  ||  ||    ||
MVHLTPEEKTAVNALWGKVNVDAVGGE
```

Local vs. Global

- Local alignment – finds regions of high similarity in **parts** of the sequences

```
ADLGAVFALCDRYFQ
||||      |||| |
ADLGRTQN CDRYYQ
```

- Global alignment – finds the best alignment across the **entire** two sequences

```
ADLGAVFALCDRYFQ
||||      |||| |
ADLGRTQN-CDRYYQ
```

Evolutionary changes in sequences

Three types of nucleotide changes:

1. ***Substitution*** – a replacement of one (or more) sequence characters by another: **AAGA → AACA**
2. ***Insertion*** - an insertion of one (or more) sequence characters: **AAG A**
3. ***Deletion*** – a deletion of one (or more) sequence characters: **A A GA**

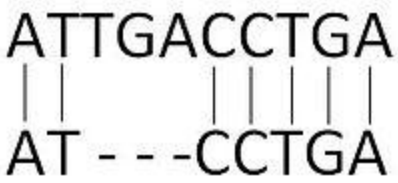
Insertion + Deletion → Indel

Gap penalty

- The **gap penalty** is a scoring system used in bioinformatics for aligning a small portion of genetic code, more accurately, fragmented genetic sequence.
  - **Constant**
  - **Linear**
  - **Affine**

Constant Gap penalty

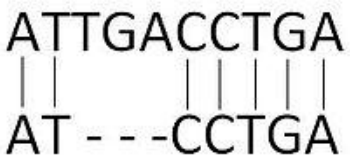
- This is the simplest type of gap penalty: a fixed negative score is given to every gap, regardless of the its length.



- Aligning two short DNA sequences, with '-' depicting a gap of one base pair.
- If each match was worth 1 point and the gap -1, the total score: 7 - 1 = 6

Linear Gap penalty

- Compared to the constant gap penalty, the linear gap penalty takes into account the length (L) of each insertion/deletion in the gap.
  - the penalty for each inserted/deleted element is B and the length of the gap L;
  - The total gap penalty would be the product of the two BL.



- Unlike constant gap penalty, the size of the gap is considered. With a match with score 1 and gap -1, the score here is (7-3 = 4).

Affine Gap penalty

- The most widely used gap penalty function is the affine gap penalty.
- The affine gap penalty combines the components in both the constant and linear gap penalty, taking the form $A+(B \cdot L)$.
  - This introduces new terms, A is known as the gap opening penalty, B the gap extension penalty and L the length of the gap.

If the size of the gap was important, a small A and large B (more costly to extend gap) is used and vice versa.

Toy exercise

Compute the scores of each of the following alignments using this naïve scoring scheme

Scoring scheme:

- Match: +1
- Mismatch: -2

} **Substitution matrix**

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

- Indel: -1 **Gap penalty (opening = extending)**

```
AAGCTGAATT-C-GAA
AGGCT-CATTTCTGA-
```

```
A-AGCTGAATTC--GAA
AG-GCTCA-TTTCTGA-
```

Choosing an alignment:

- Many different alignments between two sequences are possible:

```
AAGCTGAATTCGAA
AGGCTCATTTCTGA
```

```
AAGCTGAATT-C-GAA
AGGCT-CATTTCTGA-
```

...

```
A-AGCTGAATTC--GAA
AG-GCTCA-TTTCTGA-
```

How do we determine which is the best alignment?

Best alignment

- Similarity method
  - Based on the similarity property

- Distance Method
  - Based on the dissimilarity property

Best alignment

- Similarity method

$$S = x - \sum W_k Z_k$$

- S=similarity
- X= no of matched pairs
- Z=number of gaps of length k
- W=penalty for gap of length k

Best alignment

- Distance method

$$D = y + \sum W_k Z_k$$

- D=Distance
- Y= no of mismatches
- Z=number of gaps of length k
- W=penalty for gap of length k

Two kinds of sequence alignment: global and local

global alignment algorithm
--Needleman and Wunsch (1970)

local alignment algorithm
--Smith and Waterman (1981)

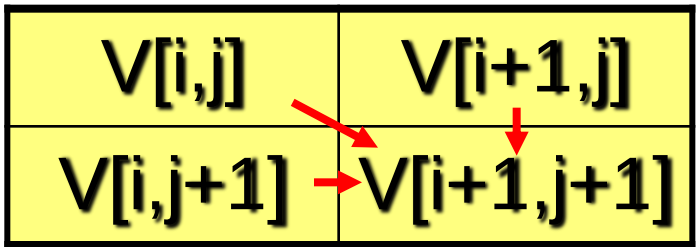
Pairwise alignment algorithm matrix representation: formulation

Needleman-Wunsch Algorithm(1970)

2 sequences: S1 and S2 and a Scoring scheme: match = 1, mismatch = -1, gap = -2

V[i,j] = value of the optimal alignment between S1[1…i] and S2[1…j]

$$V[i+1,j+1] = \max \begin{cases} V[i,j] + S(S1[i+1],S2[j+1]) \\ V[i+1,j] + S(gap) \\ V[i,j+1] + S(gap) \end{cases}$$



Pairwise alignment algorithm matrix representation: initialization

Scoring scheme:
- Match = 1
- Mismatch = -1
- Indel (gap) = -2

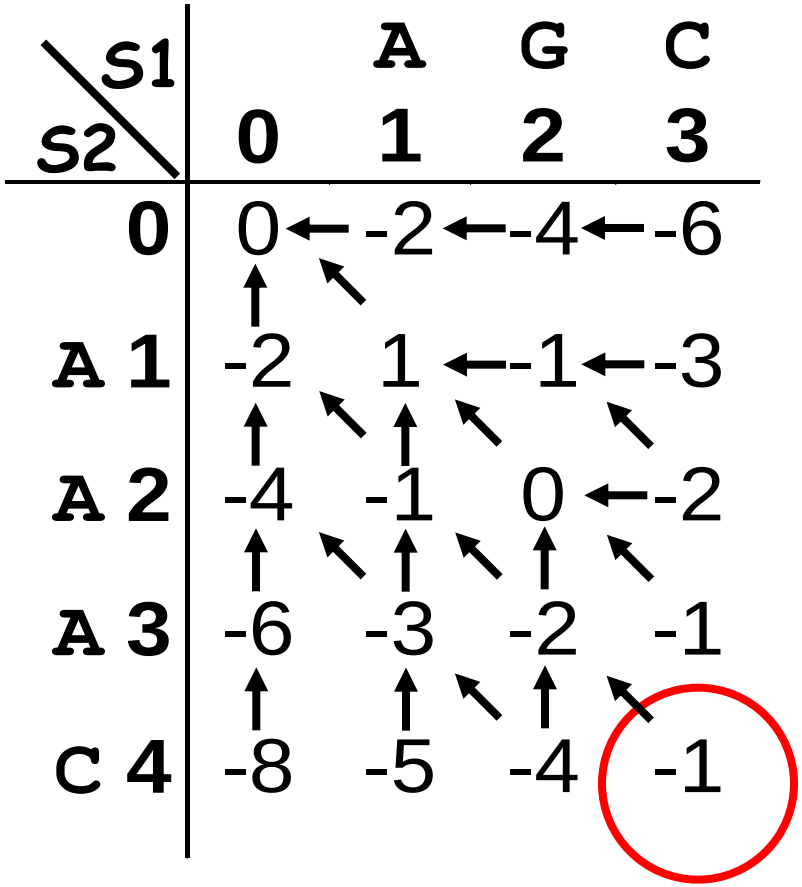
S1 / S2		A	G	C
	0	**1**	**2**	**3**
0	0	-2	-4	-6
A 1	-2			
A 2	-4			
A 3	-6			
C 4	-8			

Pairwise alignment algorithm matrix representation: filling the matrix

Scoring scheme:
- Match = 1
- Mismatch = -1
- Indel (gap) = -2

S1 / S2		A	G	C
	0	**1**	**2**	**3**
0	0	-2	-4	-6
A 1	-2	1	-1	-3
A 2	-4	-1	0	-2
A 3	-6	-3	-2	-1
C 4	-8	-5	-4	-1

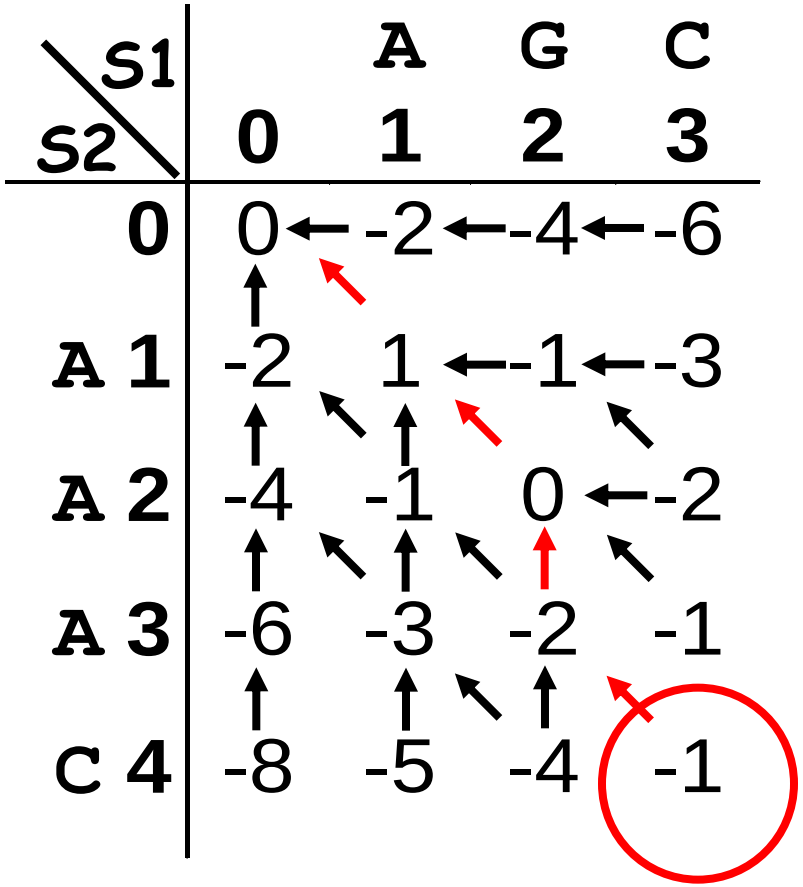
Pairwise alignment algorithm matrix representation: trace back



Pairwise alignment algorithm matrix representation: trace back

AAAC

AG-C



Global Pairwise alignment

Seq 1 : ATCA

Seq 2 : AGTA

Global Pairwise alignment

Seq 1 : ATCA

Seq 2 : AGTA

		A	T	C	A
	0	1	2	3	4
0	0	-1	-2	-3	-4
A 1	-1	1	0	-1	-2
G 2	-2	0	-1	-2	-3
T 3	-3	-1	1	0	-1
A 4	-4	-2	0	-1	1

A - T C A

A G T - A