# 8 Multiple sequence alignment

## 8.1 Multiple sequence alignment

A multiple seuqnece alignment is an effective tool to understand the characteristics of genes by comparing multiple sequences of different species at the same time.

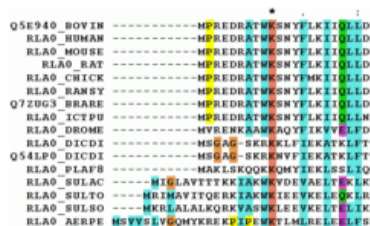**Multiple Sequence Alignment (MSA) for protein sequences**



**Figure 8.1:** An MSA of insulin proteins of seven sequences

**Notation of MSA**

- $\mathcal{A}$ : Alignment

- $m$ : Number of sequences in $\mathcal{A}$

- $s_j^i$ : An amino acid or a nucleotide of sequence $i$ and position $j$ (without gaps)

- $\bar{s}_j^i$ : An amino acid or a nucleotide of sequence $i$ and column $j$ (with gaps)

**Example of MSA notation**

```
HUMAN: TP-K
MOUSE: TLSK
RAT  : TPSK
```

- $m$ : 3

- $s_1^1$ : T (1st position of HUMAN)

- $s_2^2$ : L (2nd position of MOUSE)

- $s_4^3$ : K (4th position of RAT)

- $\bar{s}_3^1$ : - (3rd position of HUMAN)

**Making an optimal MSA**

- Insert gaps to the sequences in $\mathcal{A}$

- Maximize the score of $\mathcal{A}$

## All combinations of elements per column

The number of all possible combinations of elements per column can be calculated as follows.

$$\sum_{i=0}^{m-1} \binom{m}{i} = 2^m - 1$$

## Example of the number of combinations

$$
\begin{matrix}
s_1^1 & - & s_3^1 & s_4^1 & - & - & s_7^1 \\
s_1^2 & s_2^2 & - & s_4^2 & - & s_6^2 & - \\
s_1^3 & s_2^3 & s_3^3 & - & s_5^3 & - & -
\end{matrix}
$$

- $m$: 3

- $2 \times 2 \times 2 - 1 = 7$

## Alignment methods

- Dynamic programming with $m$-dimensional array (deterministic)

- Progressive alignment (heuristics)

## SP score

One of the common methods to calculate the score of an alignment is using SP (sum-of-pairs) scores. SP uses pair-wise scores on all possible paired sequences to obtain the final score for the alignment. SP is defined as below.

$$S(\mathcal{A}) = \sum_{i=0}^{m-1} \sum_{j=i+1}^{m} S(\bar{s}^i, \bar{s}^j)$$

**N.B.** The score of $S(\bar{s}^i, \bar{s}^j)$ is 0 when both elements are gaps.

## Example of SP score

Use the simple scoring scheme and calculate the SP score. Simple scoring scheme: Match: 1, Mismatch: 0, and Gap penalty: 1

```
Seq1 A-GC
Seq2 ACG-
Seq3 A-TC
```

$S(\bar{s}^1, \bar{s}^2) = 1 - 1 + 1 - 1 = 0$
$S(\bar{s}^1, \bar{s}^3) = 1 + 0 + 0 + 1 = 2$
$S(\bar{s}^2, \bar{s}^3) = 1 - 1 + 0 - 1 = -1$

$S(\mathcal{A}) = S(\bar{s}^1, \bar{s}^2) + S(\bar{s}^1, \bar{s}^3) + S(\bar{s}^2, \bar{s}^3) = 0 + 2 - 1 = 1$

**Exercise 8.1**

Use the simple scoring scheme and calculate the SP score.

```
Seq1 A-CC
Seq2 C-TC
Seq3 CAG-
```

## 8.2   Dynamic programming with $m$-dimensional array

Dynamic programming (DP) can be extended to handle multiple alignments.
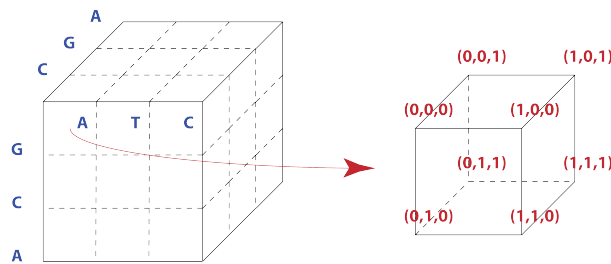
**Multi-dimensional array for dynamic programming**



**Figure 8.2:** A three-dimensional DP array

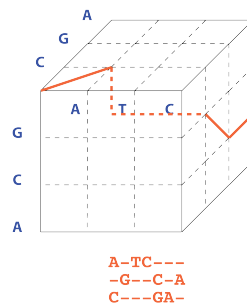**Example of alignment representation**



**Figure 8.3:** An alignment with a three-dimensional DP array

**The number of candidate scores for a vertex**

The number of the inbound neighboring vertices is defined as follows.

$$\sum_{i=0}^{m-1} \binom{m}{i} = 2^m - 1$$

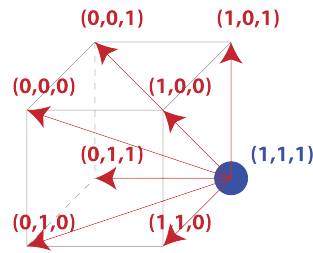**Example of edges of 3-dimensional cell**



**Figure 8.4:** An example of seven different edges to one vertex when m = 3

**A pruning method**

- $K$ : a score of an MSA (it does not need to be the optimal)

- $\nu$ : current vertex

- $S_\nu$ : best score from the start vertex to $\nu$ (by DP)

- $F_\nu$ : best score from the end vertex to $\nu$ (by non-DP)

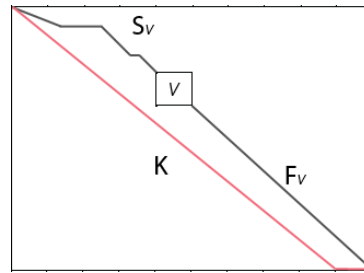- if $S_\nu + F_\nu < K$ then $\nu$ does not lie on the optimal path



**Figure 8.5:** Score estimation

**Forward-recursion DP for MSA**

Instead of looking up inbound neighboring vertices, the forward recursion DP sends the calculated score to all outbound neighboring vertices.
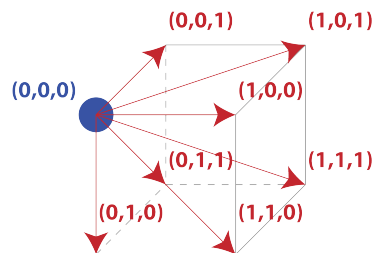


**Figure 8.6:** Values are forwarded to all outgoing neighbors