

Sequencing

1. A sequence of DNA is denoted by $S = x_1x_2 \dots x_n$ where x_i is a letter in $\{A, T, G, C\}$, $i = 1, 2, \dots, n$ and $S(i) = x_i$.
2. The first problem we study about DNA sequences is the comparison of two sequences and determine how similar they are.
3. In general, the sequences we compare may not be of the same length, that is, they may have different number of letters, but not too far from being the same.

For example, $S_1 = \text{AGACCTAG}$ and $S_2 = \text{CACCTGCAG}$.

Alignment

- In order to compare the similarity of two sequences, *alignment* is applied to adjust their length or the number of letters.
- There are three basic types of alignments:

Type 1: Global alignment

Align them by inserting space “X” to obtain two equal length sequences. e. g. $S_1 \rightarrow$ AGACCTXXAG

$S_2 \rightarrow$ XCACCTGCAG

Type 2: Local alignment

Align them according to sub-sequences (sub-strings). e.g.

$S_1 =$ AGACCTAG

$S_2 =$ CACCTGCAG

Alignment - Continued

- Type 3: Semi-global alignment

Align the **prefix** (字首) or **suffix** (字尾) of sequences (respectively) to make equal length sequences. e. g. AGACCTAG**XX**

and **X**CACCTGCAG (from S_1 and S_2)

- *Alignment is a very important tool in the comparison of two or more nucleotide or protein sequences (strings) to determine the degree of similarity.*
- *The similarity commonly used to deduct (discover) functional or evolutionary relationships between genes and proteins.*

Formal Definition of Alignment

- An alignment of two sequences S_1 and S_2 is a pair of sequences (S_1', S_2') obtained by insertion of spaces in S_1 and S_2 respectively such that
 - (1) S_1' and S_2' have the same length, and
 - (2) For each i , either $S_1'(i)$ or $S_2'(i)$ is not a space. (不可同時空白)
- Note that S_1 (resp. S_2) be viewed as a subsequence of S_1' (resp. S_2').
- Example: $S_1 = \text{AGAC}$ and $S_2 = \text{TACCC}$. Alignments:
 XAGAC with TACCC or XAGACX with TACCXC .

Score of Alignment

- Consider two sequences after alignment

$S = x_1x_2 \dots x_k$ and $T = y_1y_2 \dots y_k$. ($S(i) = x_i$, $T(i) = y_i$)

- For all i, j, l, l' , we have the following score assignment:

$x_i = y_i \rightarrow \text{score } \sigma(x_i, y_i) = p > 0,$

$x_j \neq y_j \rightarrow \text{score } \sigma(x_j, y_j) = q < 0,$

$x_l - X (= y_l) \rightarrow \text{score } \sigma(x_l, X) = r < 0,$ and

$X = (x_{l'}) - y_{l'} \rightarrow \text{score } \sigma(X, y_{l'}) = r < 0.$ (X represents space.)

- The score of (S, T) , $\sigma(S, T) = \sum \sigma(S(i), T(i))$ for $i = 1, 2, \dots, k$.

Similarity of Two Sequences

- Let S_1, S_2 be two sequences and S_1', S_2' be their alignments respectively.
- The similarity of S_1 and S_2 , $\text{Sim}(S_1, S_2)$, is defined as the maximum score of S_1' and S_2' , $\sigma(S_1', S_2')$, among all possible alignments of S_1 and S_2 .
- The pair (S_1', S_2') with largest score is called an optimal alignment.

Example: Let $p = 1$, $q = -1$ and $r = -2$. Then, for AGC and AAAC,

$\text{Sim}(\text{AGC}, \text{AAAC}) = -1$. (?)

XAGC	AXGC	AGXC
AAAC	AAAC	AAAC

Algorithm for Similarity

- Find *Sim*(S_1, S_2) for input sequences S_1 and S_2 .
 1. Consider all alignments (S_1', S_2') .
 2. Find $\sigma(S_1', S_2')$.
- *Theorem.* The time and space to accomplish “finding optimal alignment” is $O(|S_1| \cdot |S_2|)$.

Proof. In each step, it takes constant time and space.

Note: No algorithm is known that uses asymptotically less time and has the same generality. For some specific problems, use less time might be possible.

Difficulty Occurs

- The sequences we compare are of length in *thousands, say 10^4* .
- We are not able to store them as matrices of size $10^4 \times 10^4$.
(Comparison of all alignments)
- Better ideas are required for finding optimal alignments.

(*) Exercise 1. Give an algorithm to find an optimal alignment of two given sequences.

Note: You may refer to Bonnie Berger's idea or find one which you have better result.

Multiple Sequences Alignments (MSA)

- Definition. (Sum of pair-scores, SP-score)

Let Π be a collection of alignments of $S_1, S_2, \dots, S_n : S_1', S_2', \dots, S_n'$. Then, the SP-score of Π , denoted by $SP(\Pi)$ is the sum of $\sigma(S_i', S_j') = s(\Pi_{i,j})$ for all $1 \leq i < j \leq n$.

- MSA 's are applied in finding the similarity of protein sequences.

Example. RCTLEE RCLEE CTLEE CTEE

Alignments Π RCTLEE RCXLEE XCTLEE XCTXEE

$SP(\Pi) = 3 + 3 + 0 + 0 + (-3) + 2 = 5$ ($p = 1, q = -1, r = -2$) (倆倆比對)

MSA Problem

- Given k sequences, find the optimal alignment of these k sequences.
- Dynamic Programming

Time complexity = $O(k^2 2^k n^k)$ (Exponential time in k .)

Note. For each adjustment, $k-1$ comparisons are needed.

- Best known results
 - It was proved by Wang and Jiang (1994), & Bonizzoni and Vedova (2001) that MSA problem is NP-complete.
 - Only approximation algorithms were obtained so far.
- (*) Can you find one? (Project selection)