

Multiple Sequence Alignment

R.C.T. Lee, Chin Lung Lu

CS 5313 Algorithms for Molecular Biology

Two sequence alignment

1. **Dynamic programming algorithms:** guarantee to find an optimal solution
 - **Needleman-Wunsch algorithm:** global alignment (Needleman & Wunsch, 1970)
 - **Smith-Waterman algorithm:** local alignment (Smith & Waterman, 1981)
2. **Heuristic algorithms:** do not guarantee to find an optimal solution
 - **FastA:** (Pearson & Lipman, 1988)
 - **BLAST:** (Altschul et al., 1990)

Multiple sequence alignment

S_1 : RCTLEE

S_2 : RCLEE

S_3 : CTLEE

S_4 : CTEE

S_1 : R C T L E E

S_2 : R C - L E E

S_3 : - C T L E E

S_4 : - C T - E E

4 Sequences \Rightarrow A Multiple Sequence Alignment

SP-score of an MSA

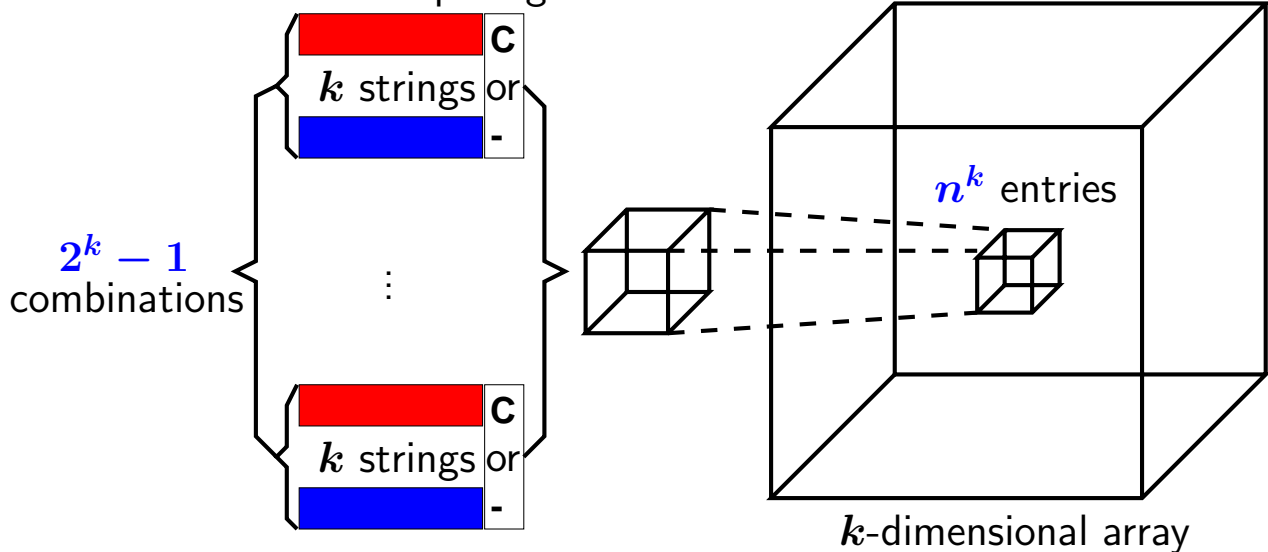
- **SP-score** (Sum-of-Pairs) of an MSA: the sum of the SP-scores of all columns
- **SP-score of a column**: the sum of pairwise scores of all pairs of characters

S_1 :	R	C	T	L	E	E
S_2 :	R	C	-	L	E	E
S_3 :	-	C	T	L	E	E
S_4 :	-	C	T	-	E	E

The SP-score of the first column is
 $score(R, R) + 4 \times score(R, -) + score(-, -)$,
where $score(-, -) = 0$

Sum-of-pairs MSA problem

$C(k, 2) = k^2$ pairs of symbols
for computing a column score



- The total time-complexity is $\mathcal{O}(k^2 2^k n^k)$.

Sum-of-pairs MSA problem

- **NP-complete** (Wang and Jiang, 1994; Bonizzoni and Della Vedova, 2001)
- **Approximate methods:** (k sequences)
 - $(2 - \frac{2}{k})$ -approximation (Gusfield, 1991)
 - $(2 - \frac{3}{k})$ -approximation (Pevzner, 1992)
 - $(2 - \frac{l}{k})$ -approximation, where $l < k$ (Bafna, Lawler and Pevzner, 1997)
- **Heuristic methods:** Progressive algorithms

2-Approximation algorithm ①

- Define the scoring function as follows:

$$\sigma(x, y) = 0 \begin{cases} 0, & \text{if } x = y, \\ 1, & \text{otherwise} \end{cases}$$

- $D(S_i, S_j)$: globally aligned distance between S_i and S_j
- Consider the set S of four sequences:

$$S_1 = \text{ATGCTC}$$

$$S_2 = \text{AGAGC}$$

$$S_3 = \text{TTCTG}$$

$$S_4 = \text{ATTGCATGC}$$

2-Approximation algorithm ②

$$S_1 = \text{ATGCTC}$$

$$S_2 = \text{AGAGC}$$

$$S_2 = \text{A-GAGC}$$

$$S_3 = \text{TTCTG}$$

$$D(S_1, S_2) = 3$$

$$D(S_2, S_3) = 5$$

$$S_1 = \text{ATGCTC}$$

$$S_2 = \text{A--G-A-GC}$$

$$S_3 = \text{TT-CTG}$$

$$S_4 = \text{ATTGCATGC}$$

$$D(S_1, S_3) = 3$$

$$D(S_2, S_4) = 4$$

$$S_1 = \text{AT-GC-T-C}$$

$$S_3 = \text{-TT-C-TG-}$$

$$S_4 = \text{ATTGCATGC}$$

$$S_4 = \text{ATTGCATGC}$$

$$D(S_1, S_4) = 3$$

$$D(S_3, S_4) = 4$$

2-Approximation algorithm ③

- Find the **center sequence** S_i of S which **minimizes** $\sum_{X \in S \setminus \{S_i\}} D(S_i, X)$.

$$D(S_1, S_2) + D(S_1, S_3) + D(S_1, S_4) = 9$$

$$D(S_2, S_1) + D(S_2, S_3) + D(S_2, S_4) = 12$$

$$D(S_3, S_1) + D(S_3, S_2) + D(S_3, S_4) = 12$$

$$D(S_4, S_1) + D(S_4, S_2) + D(S_4, S_3) = 11$$

Hence, S_1 is the center in this example

2-Approximation algorithm ④

- Align S_2 with S_1 :

$$S_1 = \text{ATGCTC}$$

$$S_2 = \text{A-GAGC}$$

$$S_2 = \text{A-GAGC}$$

- Add S_3 by aligning S_3 with S_1 :

$$S_1 = \text{ATGCTC}$$

$$S_3 = \text{-TTCTG}$$

- Add S_4 by aligning S_4 with S_1 :

$$S_3 = \text{- T - T C - T - G}$$

$$S_2 = \text{A - - G A - G - C}$$

$$S_1 = \text{AT - GC - T - C}$$

$$S_4 = \text{ATTGCATGC}$$

2-Approximation algorithm ⑤

- $d(S_i, S_j)$ ($d^*(S_i, S_j)$): the distance between S_i and S_j induced by this approximation (an optimal) algorithm
- $d(S_i, S_j) + d(S_i, S_k) \geq d(S_j, S_k)$ (triangular inequality)
- $App = \sum_{i=1}^k \sum_{\substack{j=1 \\ j \neq i}}^k d(S_i, S_j)$
- $Opt = \sum_{i=1}^k \sum_{\substack{j=1 \\ j \neq i}}^k d^*(S_i, S_j)$

2-Approximation algorithm ⑥

- Claim that $App \leq 2Opt$
- Since triangle inequality property of $d(S_i, S_j)$ and $d(S_1, S_i) = d(S_i, S_1)$, we have

$$\begin{aligned}
 App &= \sum_{i=1}^k \sum_{\substack{j=1 \\ j \neq i}}^k d(S_i, S_j) \\
 &\leq \sum_{i=1}^k \sum_{\substack{j=1 \\ j \neq i}}^k d(S_i, S_1) + d(S_1, S_j) \\
 &= 2(k-1) \sum_{i=2}^k d(S_1, S_i)
 \end{aligned}$$

2-Approximation algorithm ⑦

- Since $d(S_1, S_i) = D(S_1, S_i)$ for all i , we have

$$App \leq 2(k-1) \sum_{i=2}^k D(S_1, S_i)$$

- Note that $Opt = \sum_{i=1}^k \sum_{\substack{j=1 \\ j \neq i}}^k d^*(S_i, S_j)$
- Since $D(S_i, S_j) \leq d^*(S_i, S_j)$, we have

$$Opt = \sum_{i=1}^k \sum_{\substack{j=1 \\ j \neq i}}^k d^*(S_i, S_j) \geq \sum_{i=1}^k \sum_{\substack{j=1 \\ j \neq i}}^k D(S_i, S_j)$$

2-Approximation algorithm ⑧

- Since S_1 is the center, we have

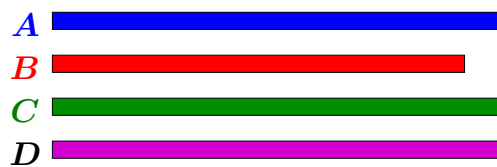
$$\begin{aligned} Opt &\geq \sum_{i=1}^k \sum_{\substack{j=1 \\ j \neq i}}^k D(S_i, S_j) \\ &\geq \sum_{i=1}^k \sum_{j=2}^k D(S_1, S_j) = k \sum_{j=2}^k D(S_1, S_j) \end{aligned}$$

- Hence, $\frac{App}{Opt} \leq \frac{2(k-1)}{k} \Rightarrow$

$$App \leq \left(2 - \frac{2}{k}\right) Opt < 2Opt$$

Progressive MSA

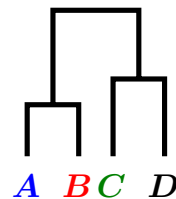
1. Construct the guide (evolutionary) tree



(1) Four sequences

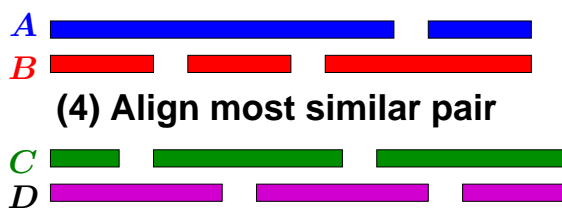
	A	B	C	D
A	0	3	5	5
B		0	5	5
C			0	4
D				0

(2) Distance matrix



(3) Evolutionary tree

2. Progressively align sequences by the tree



(4) Align most similar pair



(5) Align next similar pair

(once a gap, always a gap)



(6) Align two pre-aligned groups

Progressive MSA

1. Compute the distance matrix by aligning all pairs of sequences (using dynamic programming algorithm)
2. Compute the guide tree from the distance matrix:
 - **PILEUP** of GCG: UPGMA (Unweighted Pair-Group Method using Arithmetic mean)
 - **CLUSTAL W**: NJ (Neighbor-Joining)
 - **YAMA-MST** of Tang's lab: Kruskal MST
3. Progressively align the sequences according to the branching order in the guide tree (once a gap, always a gap)

1. Computation of distance matrix

S_1 : RCTLEE

S_2 : RCLEE

S_3 : CTLEE

S_4 : CTEE

	S_1	S_2	S_3	S_4
S_1	0	1	1	2
S_2		0	2	1
S_3			0	1
S_4				0

4 Sequences

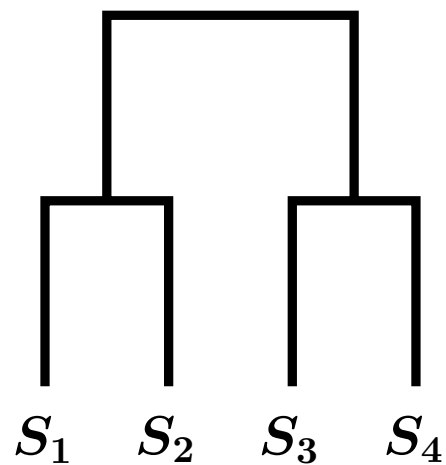


Distance Matrix

2. Construction of guide tree

	S_1	S_2	S_3	S_4
S_1	0	1	1	2
S_2		0	2	1
S_3			0	1
S_4				0

Distance Matrix



Evolutionary Tree

3. Progressive alignment

