String-searching Algorithms

Alignment algorithms

Sergio Peignier sergio.peignier@insa-lyon.fr

Associate Professor
INSA Lyon
Biosciences department

Alignment

- Arrange sequences and **identify regions of similarity**
- Conserved regions denote functional, structural, evolutionary relationships
- Mutations considered:
  substitution, insertions and deletions

Global Alignment

Goal:

- Finds the best **alignment** over the **entire length** of two sequences S and T
- Maximizes the **character-to-character matches.**

Needleman-Wunsch

Goal:

- Similar to **Wagner–Fischer** algorithm.
- Underlying **scoring system**
  - 1 for matches
  - -1 for mismatches
  - -1 for indels
- $F_{S,T}(i, j)$: Number of matches between $S[1, i]$ and $T[1, j]$

Needleman-Wunsch

Definition:

$F_{S,T}(i,j)$: Number of matches between $S[1,i]$ and $T[1,j]$

- If $i = 0$ or $j = 0$ $F_{S,T}(i,j) = -max(i,j)$
- Otherwise:

$$\mathcal{F}_{S,T}(i,j) = max \begin{cases} \mathcal{F}_{S,T}(i-1,j) - 1 \\ \mathcal{F}_{S,T}(i,j-1) - 1 \\ \mathcal{F}_{S,T}(i-1,j-1) - \mathbb{1}_{S[i] \neq T[j]} + \mathbb{1}_{S[i]=T[j]} \end{cases}$$

Needleman-Wunsch

- Record in a matrix values
  $F_{S,T}(i,j), \quad \forall i \in \{0,\dots,|S|\}$ and $\forall j \in \{0,\dots,|T|\}$

- For each element in the matrix, record the decision that was made to reach this value :
  - $F_{S,T}(i-1,j) \rightarrow S[i]$ is aligned with a gap
  - $F_{S,T}(i,j-1) \rightarrow T[j]$ is aligned with a gap
  - $F_{S,T}(i-1,j-1) \rightarrow S[i]$ and $T[j]$ are aligned

- **Traceback:** Start from the element at row $|S|$ and column $|T|$, and move backwards until row 0 and column 0 (retrieving the decision taken at each step).

Example

Local Alignment

- Finds alignments **shorter than the entire sequences**
- Compare two very different sequences sharing some **local regions with high similarity**

Smith-Waterman

Similar to the **Needleman-Wunsch** method.

$F_{S,T}(i,j)$: Local alignment score between $S[1,i]$ and $T[1,j]$

- If $i = 0$ or $j = 0$ $F_{S,T}(i,j) = 0$
- Otherwise:

$$\mathcal{F}_{S,T}(i,j) = max \begin{cases} \mathcal{F}_{S,T}(i-1,j) - 1 \\ \mathcal{F}_{S,T}(i,j-1) - 1 \\ \mathcal{F}_{S,T}(i-1,j-1) - \mathbb{1}_{S[i] \neq T[j]} + \mathbb{1}_{S[i]=T[j]} \\ 0 \end{cases}$$

Traceback: Start at the maximal position in the matrix, and stop as soon as a 0 is reached.

Example

Multiple sequence alignment

Input: set of sequences with (possibly) different lengths and sharing an evolutionary relationship

$$S = \begin{cases} S_1 = (S_{1,1}, S_{1,2}, ..., S_{1,L_1}) \\ \dots \\ S_N = (S_{N,1}, S_{N,2}, ..., S_{N,L_N}) \end{cases}$$

Output: set of sequences with length $L \geq \max\{L_i \mid i = 1, \dots, N$, from sequences $\in S$ with gaps

$$S' = \begin{cases} S'_1 = (S'_{1,1}, S'_{1,2}, ..., S'_{1,L}) \\ \dots \\ S'_N = (S'_{N,1}, S'_{N,2}, ..., S'_{N,L}) \end{cases}$$

Algorithms

- MSA program (Dynamic programming)
- Progressive alignment (Clustal, T-Coffee)
- Iterative methods (MUSCLE)
- Hidden Markov models
- ...

Clustal

- Perform all **pairwise alignments**
- Apply the **UPGMA** agglomerative (bottom-up) hierarchical clustering to build a **guide tree**
- **Multiple alignment** based on the guide tree.