DP problems of interest

Manohar, Thierry

1 Problems of interest

1.1 Smith-Waterman

Problem: matching two strings S, T.

Alphabets:

- Input: $\Sigma(S) = \Sigma(T) = \{a, c, g, t\}$.
- Cost matrix: $\Sigma(M) = \text{integers}\{0..n\}, n = \max(\text{cost}) \cdot \min(\text{length}(S), \text{length}(T))$
- Backtrack matrix: $\Sigma(B) = \{\mathit{stop}, W, N, NW\}$

Initialization:

- Cost matrix: $M_{(i,0)} = M_{(0,j)} = 0$.
- Backtrack matrix: $B_{(i,0)} = B_{(0,j)} = \mathit{stop}$.

Recurrence:

$$M_{(i,j)} = \max \left\{ \begin{array}{l|l} 0 & \mathit{stop} \\ M_{(i-1,j-1)} + \text{cost}(S(i), T(j)) & NW \\ M_{(i-1,j)} - d & N \\ M_{(i,j-1)} - d & W \end{array} \right\} = B_{(i,j)}$$

1.2 Smith-Waterman with gap extension at different cost

Problem: matching two strings S, T.

Alphabets:

- Input: $\Sigma(S) = \Sigma(T) = \{a, c, g, t\}$.
- Cost matrix: $\Sigma(M) = \text{integers}\{0..n\}, n = \max(\text{cost}) \cdot \min(\text{length}(S), \text{length}(T))$
- Backtrack matrix: $\Sigma(B) = \{\mathit{stop}, W, N, NW\}$

Initialization:

- Cost matrix: $M_{(i,0)} = M_{(0,j)} = 0$.
- Gap opening matrix: $E_{(i,0)} = 0, 0 \leq i \leq \text{length}(S)$
- Gap extending matrix: $E_{(0,j)} = 0, 0 \leq j \leq \text{length}(T)$
- Backtrack matrix: $B_{(i,0)} = B_{(0,j)} = \mathit{stop}$.

Recurrence for the cost matrix:

$$M_{(i,j)} = \max \left\{ \begin{array}{l|l} 0 & stop \\ M_{(i-1,j-1)} + \mathrm{cost}(S(i), T(j)) & NW \\ E_{(i,j)} & N \\ F_{(i,j)} & W \end{array} \right\} = B_{(i,j)}$$

Recurrence for the gap opening/extending matrices:

$$E_{(i,j)} = \max \left\{ \begin{array}{l|l} M_{(i,j-1)} - \alpha & NW \\ E_{(i,j-1)} - \beta & N \end{array} \right\} = B_{(i,j)}$$

$$F_{(i,j)} = \max \left\{ \begin{array}{l|l} M_{(i-1,j)} - \alpha & NW \\ E_{(i-1,j)} - \beta & N \end{array} \right\} = B_{(i,j)}$$

Otherwise written as:

$$M_{(i,j)} = \max \left\{ \begin{array}{l|l} 0 & stop \\ M_{(i-1,j-1)} + \mathrm{cost}(S(i), T(j)) & NW \\ \max_{1 \le k \le j-1} M_{(i,k)} - \alpha - (j-1-k) \cdot \beta & N \\ \max_{1 \le k \le i-1} M_{(k,j)} - \alpha - (i-1-k) \cdot \beta & W \end{array} \right\} = B_{(i,j)}$$

1.3 Nussinov algorithm

Problem: folding a RNA string S over itself.

Alphabets:

- Input: $\Sigma(S) = \{A, C, G, U\}$.
- Cost matrix: $\Sigma(M) = \{0..n\}, n = \text{length}(S)/2$
- Backtrack matrix: $\Sigma(B) = \{stop, W, S, SW, 1..n\}$

Initialization:

- Cost matrix: $\begin{cases} M_{(i,i)} = 0 \\ M_{(i,i-1)} = 0 \end{cases} \forall i \in 1..\text{length}(S)$
- Backtrack matrix: $\begin{cases} B_{(i,i)} = stop \\ B_{(i,i-1)} = stop \end{cases} \forall i \in 1..\text{length}(S)$

Recurrence:

$$M_{(i,j)} = \max \left\{ \begin{array}{l|l} M_{(i+1,j-1)} + \omega(i,j) & SW \\ M_{(i+1,j)} & S \\ M_{(i,j-1)} & W \\ \max_{i<k<j} M_{(i,k)} + W_{(k+1,j)} & k \end{array} \right\} = B_{(i,j)}$$

With $\omega(i,j) = 1$ if i, j are complementary. 0 otherwise.