Algorithms in Computational Biology
Lecture #11: HMM, Viterbi, Sampling from Posterior

Sean Cohen

27.11.2018

1 Possible alignment number

Let S, T be two strings in length of n which we want to align with each other. Note that at each position we have three options of chars alignment (we are not allowed to put gap infront of a gap):

1. s_i in front of t_j
2. s_i in front of a gap
3. t_j in front of a gap

So our allignment will have at least n positions containing char from S, char from T or char from both (whrn $S_i == T_j$). Note that gaps are the worst case so we won't choose them at any case. Finally we will get allignment X of length of at least n and the number of possible results that we can get is $3^{|X|} \geq 3^n$

2 Linear gap penalty

1. **Pre-Init:**
   (a) Let S,T be strings
   (b) let V be a matrix at size $(|S|+1) \times (|T|+1)$
   (c) let Ptr be a matrix that will save the trace at size of $(|S|+1) \times (|T|+1)$
   (d) let isFirst be a boolean variable that equals true while we face single gap and turn to false after one gap heading to another gap.
   (e) let d be the penalty for first gap and e for non-first gap.
2. **Initialization:**
   (a) $isFirst = False$
   (b) $V_{0,0} = 0$
   (c) $for(i = 0...|T|)$
      i. $V_{0,i} = d + e(i-1)$
   (d) $for(j = 0...|S|)$
      i. $V_{J,0} = d + e(i-1)$
3. **Iteration:**
   (a) $for(i = 1...|S|)$

i. $for(j = 1...|T|)$

$$V_{i,j} = max \begin{cases} (V_{i-1,j-1} + \sigma(s_i, t_j), & \\ (V_{i-1,j} + d) & isFirst = true \\ (V_{i-1,j} + e) & isFirst = false \\ (V_{i,j-1} + d) & isFirst = true \\ (V_{i,j-1} + e) & isFirst = false \\ & \end{cases}$$

ii. $if(V_{i,j} = V_{i-1,j-1} + \sigma(s_i, t_j)$
   A. $isFirst = True$

iii. $else$
   A. $isFirst = False$

iv. $$Ptr_{i,j} = \begin{cases} Diagonal & V_{i-1,j-1} \\ up & V_{i-1,j} \\ left & V_{i,j-1} \\ & \\ & \\ & \end{cases}$$

4. **Termination and reconstructing the solution:**
   (a) $bestAligmentScore = V_{|S|,|T|}$
   (b) let X be string which represents the best alignment
   (c) let $i = |S|, j = |T|, k = 0$
   (d) $while(Ptr_{i,j} \neq null)$
      i. $if(Ptr_{i,j} == diagonal)$
         A. $reverseX[k] = S_i = T_j$
         B. $i = i - 1, j = j - 1$
      ii. $elif(Ptr_{i,j} == up)$
         A. $reverseX[k] = S_i$
         B. $i = i - 1$
      iii. $elif(Ptr_{i,j} == left)$
         A. $reverseX[k] = T_j$
         B. $j = j - 1$
   (e) $X = REVERSE(reverseX)$

3 Overlap Alignment

1. **Pre-Init:**
   (a) Let S,T be strings
   (b) let V be a matrix at size $(|S| + 1) \times (|T| + 1)$
   (c) let Ptr be a matrix that will save the trace at size of $(|S| + 1) \times (|T| + 1)$
   (d) let d be the penalty for a gap that isn't at the begining or end of the alignment.
2. **Initialization:**

(a) $V_{0,0} = 0$

(b) $for(i = 0...|S|)$

i. $V_{i,0} = 0$ //no penalty on s_i against gaps at the begining

(c) $for(j = 0...|T|)$

i. $V_{0,j} = d$ //penalty for gaps.

3. **Iteration:**

(a) $for(i = 1...|S|)$

i. $for(j = 1...|T|)$

A. $$V_{i,j} = max \begin{cases} (V_{i-1,j-1} + \sigma(s_i, t_j) & \\ (V_{i-1,j} + d) & \\ (V_{i,j-1} + d) & i < |S| \\ (V_{i,j-1}) & i = |S| \end{cases}$$

B. $$Ptr_{i,j} = \begin{cases} Diagonal & V_{i-1,j-1} \\ up & V_{i-1,j} \\ left & V_{i,j-1} \end{cases}$$

4. **Termination and reconstructing the solution:**

(a) $bestAligmentScore = V_{|S|,|T|}$

(b) let X be string which represents the best alignment

(c) let $i = |S|, j = |T|, k = 0$

(d) $while(Ptr_{i,j} \neq null)$

i. $if(Ptr_{i,j} == diagonal)$

A. $reverseX[k] = S_i = T_j$

B. $i = i - 1, j = j - 1$

ii. $elif(Ptr_{i,j} == up)$

A. $reverseX[k] = S_i$

B. $i = i - 1$

iii. $elif(Ptr_{i,j} == left)$

A. $reverseX[k] = T_j$

B. $j = j - 1$

(e) $X = REVERSE(reverseX)$