Main Topic: Variations to Global Alignments, and BLAST Approximation

1. Local Alignments
   - Goal – to find substrings of sequences S and T with maximum alignment score
   - Motivation - allows evolutionarily ‘interesting’ sub-sequences to be located, even when global alignment is not similar
   - Simplest algorithm – align all subsequences A of S & B of T using previous global alignment algorithm (dynamic programming – Needleman Wunsch)
     1. Very slow $O(n^3m^3)$ and redundant
   - Alternative algorithm (Smith-Waterman): determine the value of optimal alignment of suffixes of $S[1]...S[i]$ and $T[1]...T[j]$. Start at ends of subsequences ($S[i]$ and $T[j]$), and work forward. At each step, there are 4 choices:
     1. Align the next 2 nucleotides
     2. Align gap in T to nucleotide in S (deletion)
     3. Align nucleotide in T to gap in S (insertion)
     4. No alignment (0)
       - $V(i,j)$ is then assigned as the max score of these 4 transition possibilities and entered into matrix. Trace back in matrix until a score of 0 is reached to obtain optimal alignment
       - Faster algorithm $O(mn)$, see slides for example and mathematical details

2. Gap Penalties
   - Goal – to give variable penalties for gaps in alignment based on length of gap
   - Motivation – large deletions or insertions are fairly common (introns/exons, viral DNA insertions to chromosome, etc), so penalty should decrease as length of gap increases
   - Score = func(gap length), and func can be:
     o General (rarely)
     o Convex
     o Affine - linear with large penalty initially, and slow increase with increased length, simple in computation
   - Global alignment with affine gap penalties
     o 4 matrices: $V(i,j)$ is the max of $G(i,j)$, $F(i,j)$, $E(i,j)$
       - $G(i,j)$: value of opt alignment s.t. $S[i]$ matches $T[j]$
       - $F(i,j)$: value of opt alignment s.t. $S[i]$ matches $-$
       - $E(i,j)$: value of opt alignment s.t. $-$ matches $T[j]$
     o $Gap\_penalty = g + s*(gap\_len)$
     o Book keeping issues, as $V(i,j)$ does not represent a unique sequence, 3 cases to track
     o See slides for more details and example

3. BLAST – Basic Local Alignment Scoring Tool
   - Uses an approximation to the dynamic program method with gapless match, usually ‘good enough’
   - Prefers short, strong short matches to long mediocre ones
• Input – sequence and score matrix
• Output – all matches (above threshold) and ‘E-value’ – the measure of improbability that the sequence match could happen at random
• Algorithm:
  o Break input into ‘words’, $w_i$
  o Find neighboring words (by substitutions), $v_{ij}$
  o Look up $v_{ij}$ in database and extend the ‘seed match’ in both directions in database
  o Report scores $>$ threshold, calc. E-value
• See example in slides
• Full implementation includes refinements (e.g. allow some gaps, etc)