DYNAMIC PROGRAMMING SEQUENCE ALIGNMENT

CS340

Sequence Alignment

- Problem: Given two strings S and T, what is the optimal "alignment" between them?
  - Minimum number of changes made to transform S into T, respecting order of characters
- Motivation and Applications:
  - Spell checker
  - Genetic similarity computation: DNA seq. align.
    - "changes" from S to T model mutation events
- Simplest form: Longest Common Subsequence
  - Match or insert a gap (no substitutions)

Sequence Alignment

- O_CURR**A**NCE
- OCCURR**E**NCE
- How do the 2 words align?
  - A gap must be added to ocurrance
  - An A must be replaced by an E

Sequence Alignment

- How about this?
- O_CURR_ANCE
- OCCURRE_NCE
- Which is better?
  - One gap and one mismatch
  - Three gaps and no mismatches

How do we get there?

- Need to compute the measure we are trying to optimize
  - Two general and equivalent formulations:
    - MINIMIZE the “changes” (insertions and mismatches)
      - Later the changes will also involve substitutions
    - MAXIMIZE the “matches”
- Many problems presented in “optimization” frameworks
- To speak more generally:
  - **MINIMIZE PENALTIES (COSTS)**
  - **MAXIMIZE REWARDS**

General Sequence Alignment

- Some mismatches are far likelier than others!
  - In English: (c,k), (a,e), (e,i), (i,y), (q,k),(u,a),(s,c)
  - Unlikely substitutions: (a,t), (k,r),(p,x)
    - The commonality between SELECT and SALEKT is not restricted to SLET
    - That would make them as similar with each other as they are with “SLEET”!
  - For DNA and protein sequences the relative likelihoods are of critical importance!

Gap and Mismatch Penalties

- δ is a gap penalty.
  - Each gap we insert incurs δ cost
  - $\delta > 0$
- α is a mismatch cost
  - For each pair of letters p,q, there is a cost $\alpha_{p,q}$ for lining up letters that do not match.
  - In general, $\alpha_{p,p} = 0$. No cost to exact matches.
- δ and α are external parameters that must be determined.

Which Alignment is Preferred?

- O_CURR**A**NCE
- OCCURR**E**NCE

VS

- O_CURR_ANCE
- OCCURRE_NCE

- Which is better?
- The first is better if $\delta + \alpha_{ae} < 3\delta$

Optimal Alignment Truth

- In an optimal alignment M of 2 strings X and Y, at least one of the following is true:
  - $(m, n) \in M$
  - the mth position of X is not matched
  - the nth position of Y is not matched

  - opt(i, j) = min[
    $\alpha_{xi\ yj}$ + opt(i - 1, j - 1) ↗
    δ + opt(i - 1, j) →
    δ + opt(i, j - 1) ↑
    ]

An example

N	8 ↑				
A	6 ↑				
E	4 ↑				
M	2 ↑				
-	0	2 →	4 →	6 →	8
	-	N	A	M	E

α vowel/vowel = 1
α consonant/consonant = 1
α vowel/consonant = 3
δ = 2

MATCH
MEAN and
NAME

MEAN_
N_AME

An example

N	8	6	5	4	6
A	6	5	3	5	5
E	4	3	2	4	4
M	2	1	3	4	6
-	0	2	4	6	8
	-	N	A	M	E

α vowel/vowel = 1
α consonant/consonant = 1
α vowel/consonant = 3
δ = 2

MATCH
MEAN and
NAME

MEAN_
N_AME