### CS 380 ALGORITHM DESIGN AND ANALYSIS

Lecture 15: LCS, edit distance review Sequence Alignment

References:

Algorithm Design, Pearson, 2006 Kleinberg, Tardos http://en.wikipedia.org/wiki/Needleman-3Wunsch\_algorithm

This article **may be too technical for most readers to understand**. Please help improve this article to make it understandable to non-experts, without removing the technical details. The talk page may contain suggestions. (September 2013) (Learn how and when to remove this template message)

#### CS380 Algorithm Design and Analys

#### LCS Review: Lecture 13

Let  $c_{ij}$  = length of LCS of  $x_1x_2...x_i$  and  $y = y_1y_2...y_i$ .

 $c[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0, \\ 1 + c[i-1,j-1] & \text{if } x_i = y_j, \\ \max(c[i-1,j], c[i,j-1]) & \text{if } x_i \neq y_j. \end{cases}$  $b[i,j] = \begin{cases} ````` & \text{if } x_i = y_j, \\ ````` & \text{if } x_i \neq y_j \text{ and } c[i-1,j] \ge c[i,j-1], \\ ````` & \text{if } x_i \neq y_j \text{ and } c[i-1,j] \le c[i,j-1]. \end{cases}$ 

We compute the c[i,j] and b[i,j] in order of increasing i+j, or alternatively in order of increasing *i*, and for a fixed *i*, in order of increasing *j*.

If characters match, add 1 to value in diagonal, point to that entry. If characters DON'T match, look to cell above and to left and enter that value in current cell. Point to larger cell entry, **default to up if equal.** 



#### CS300 Algorithm Design and Analysis **Edit Distance Review: Lecture 14** • Initialize matrix d: • d(i,0) = i, d(0,j) = j // i = row, j = columnCompute values in a similar row-centric way using a nested-loop the values: $d[i,j] = \begin{cases} d[i-1,j-1] & if x_i = y_i \\ d[i-1,j] + 1 & if x_i \neq y_i \\ d[i,j-1] + 1 & if x_i \neq y_i \\ d[i-1,j-1] + 1 & if x_i \neq y_i \end{cases}$ Match DIAG Deletion from X U/D Insertion to X L/R Substitution DIAG Arrows point to the cell used to compute the

current cell's value (no default on ties, not unique!)



CS38	0 Alg	orithn	n Des	sign a	ind Ai	nalys	is	
E	Xa	an	np	ole	Э.	1:	E	dit Distance, Lecture 14
Le	eve	ens	sht	ei	n [	Dis	sta	nce Calculator
Hov	v mai	ny in	sertio	ons, i	delet	tions	, and	substitutions does it take to turn ATCGTT into AGTTAC ?
Try	elepi	nant	and	relev	ant,	Satu	rday	and Sunday, Google and Facebook.
Ca	ilculate	Leve	nshtei	n dista	ance			
		Α	G	т	т	Α	с	How can this information be used for alignment?
	0	1	2	3	4	5	6	
Α	1	0	1	2	3	4	5	
т	2	1	1	1	2	3	4	
С	3	2	2	2	2	3	3	
G	4	3	2	3	3	3	4	
т	5	4	3	2	3	4	4	
т	6	5	4	3	2	3	4	









## Sequence Alignment: Definition $S_X = \{1, 2, ..., m\}: \text{ positions of characters in string } X$ $S_Y = \{1, 2, ..., n\}: \text{ positions of characters in string } Y$ $S_Y = \{1, 2, ..., n\}: \text{ positions of characters in string } Y$ A alignment M is a set of pairs of positions: $M = \{(1, 2), (2, 3), (4, 4) ... \}$ such that: $each item in S_X \text{ and } S_Y \text{ appear in at most one pair.}$ $each item in S_X \text{ and } S_Y \text{ appear in at most one pair.}$ $each item in S_X \text{ and } S_Y \text{ appear in at most one pair.}$ $each item in S_X \text{ and } S_Y \text{ appear in at most one pair.}$ $each item in S_X \text{ and } S_Y \text{ appear in at most one pair.}$ $each item in S_X \text{ and } S_Y \text{ appear in at most one pair.}$

• Option 3: M = {(1,1), (3, 2), (4, 3)}

# <section-header><section-header><section-header><section-header><equation-block><equation-block><equation-block><equation-block><equation-block><equation-block><equation-block><equation-block>







CS380 Algorithm Design and Analysis	
Sequence Alignm	ent
Alignment(X,Y)	// X =m,  Y =n
Array A[0m, 0n]	
A[i,0]=i*delta	// must use all gaps
A[0,j]=j*delta	// to align these
for $i = 1$ to m	// row centric
for j = 1 to n	
use recurrence	** to compute A[i,j]
Endfor	
Endfor	
Return A[m,n]	



			m	е	а	n
		0	1	2	3	4
	0	0	2	4	6	8
m	1	2				
а	2	4				
n	3	6				











