Edit Distance 9 February 2024 Plan * Edit Distance * Announcements * ED Dynamic Program L' Recurrence L' Reconstructing the sequence of edits. LA Space Complexity ?

	enomics	· · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · ·	Genome	Sequences	$S \in \{A, C, G, T, S\}$
· · · ·	$S_{cat} =$	$A = \begin{bmatrix} C \\ C \\ C \end{bmatrix} = \begin{bmatrix} C \\ C \\ C \end{bmatrix}$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
· · · ·	Sdog =	$A^{(1)} = A^{(1)} + A^{(2)} + A^{($	$\begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \end{array} \end{array} \end{array} \end{array} \end{array} \end{array} \begin{array}{c} \begin{array}{c} \end{array} \end{array} \end{array} \end{array} \begin{array}{c} \begin{array}{c} \end{array} \end{array} \end{array} \end{array} \end{array} \begin{array}{c} \begin{array}{c} \end{array} \end{array} \end{array} \end{array} \end{array} \begin{array}{c} \end{array} \end{array} \end{array} \end{array} \begin{array}{c} \end{array} \end{array} \end{array} \end{array} \end{array} \end{array} \begin{array}{c} \end{array} \end{array}$
· · ·		· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
• • •			
• • •			
• • •			
			· · · · · · · · · · · · · · · · · · ·

Genomics	$\mathcal{L} = \mathcal{L} = $
R Genome sequences	
$S_{cat} = A \cdot C \cdot C \cdot G \cdot A$	
$S_{dog} = A C G G A A$	$ \begin{array}{c} T \\ T $
How related are two spe	2cies
How Similar	are their genomes.
. .	. .
 	· · · · · · · · · · · · · · · · · · ·

Genomics * Genome Sequences SEZA, C, G, TJ* S_{cat} = ACCGATCGAT. Sdog = ACGGAATCGGT How related are two species? L> How "similar" are their genomes? BLAST: Basic Local Alignment Search Tool > 110 K Citations

	Distance	· ·
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	veasity can we	ed_1t
· · · · · · · · · · ·	s a s a s a s a s a s a s a s a s a s a	into a string a train
· · · · · · · · · · · ·	· ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · ·	. .	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · ·		
· · · · · · · · · · ·	. .	
· · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	
· · · · · · · · · · · · ·		

Edit	Distanc		· · · · · ·	 	· · · · · · · ·
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	N. easily	j Can.	$\begin{array}{cccccccccccccccccccccccccccccccccccc$		· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·
· · · · · · · · · ·	· · · · · · · · ·	String				
Allowed	Edits	· · · · · · · ·	· · · · · ·	· · · · · · ·	· · · · · · · · ·	· · · · · · ·
* Insev	tion / De	e le tion	· · · · · ·	· · · · · · ·	 	· · · · · · ·
* Chan	ges.	· · · · · · · ·	· · · · · ·	· · · · · · ·	 	· · · · · · ·
		M = M + M + M + M + M + M + M + M + M +		$\begin{array}{cccccccccccccccccccccccccccccccccccc$	

Edit Distance	· ·
How easily can we edit string S into string t.	
CRYPTOGRAPHY	· · · · · · · · · · · · · · · · ·
ENCRYPTING K	· · ·
1 1	· · · · · · · · · · · · · · · · · · ·

Edit Distance	. .
How easily can we edit string S into	$ \begin{array}{c} \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} $
$\begin{array}{c} \begin{array}{c} \begin{array}{c} \end{array} \end{array} \\ \end{array} \end{array} \\ \begin{array}{c} \end{array} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} $	3 insertions 5 Jeletion
$F = N \left(C R Y P + V R Y R R Y R R R R R R R R R R R R R R$	$\begin{array}{c} \begin{array}{c} \end{array} \\ \end{array} $
	. .
	. .

Given two strings S and T
Compute minimum cost edits from S -> T
insertion / deletion: Y
changing a to b. Aab
$S = \frac{1}{A}GGCTAATC$
$\overline{\Box}$
Changing a to a costs $\Delta aa = 0$.

Announcements
X HW2 due last night. Late deadline Sunday.
XHW3 ort after lecture.
* Prelim Tues Feb 20 7:30 pm
L= Last call to schedule a make-up
See Ed post from Sava Perkins
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·

Given two strings S and T Compute minimum cost edits from S -> T insertion/deletion: Y changing a to b. Aab SAGGCTAATC ---GAGGTAAGCC - ---

Given two strings S and T Compute minimum cost edits from S -> T insertion/deletion: Y changing a to b. Dab SAGGCTAATC ---GAGCTAAGCC --- MA

S_{1}	
Fact, Fix an optimal set of edits One of the following is true:	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
	. .

S	
Eact, Fix an optimal set of edits from S to T. One of the following is true: * Sn is deleted at cost X	
	•
······································	
	٠
	•
	•
	•
	٠
	•

$S = S_{\rm M}$
Fact, Fix an optimal set of edits from S to T.
One of the following is true: * Sn is deleted at cost X * Tun is inserted at cost X

S T T T T
Fact, Fix an optimal set of edits from S to T.
One of the following is true:
* Sn is deleted at cost X
* Tm is inserted at cost X
$x S_n$ changes to T_m at cost $\Delta_{S_n T_m}$

The Edit Distance Recurrence	· · · · · · · · · · · · · · · · · · ·
S[1:i] T[1:j] j	ED(i,j) L Edit distance between prefixes ST1: i] and TT1: j]
Possibilities	
* Si changes to Tj	· · · · · · · · · · · · · · · · · · ·
* Si deleted	· · · · · · · · · · · · · · · · · · ·
* Tj inserted	· · · · · · · · · · · · · · · · · · ·
	· · · · · · · · · · · · · · · · · · ·
	· · · · · · · · · · · · · · · · · · ·

The Edit Distance Recu	
$S\left[1:2\right] = \left[1:2\right] = \left[1$	ED(ij) Minimizes
Possibilities	$= \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_$
* Si changes to Tj * Si deleted	$= \sum_{i=1}^{n} (i-1, j) + \chi$
A I J I I ASEVITED	$ \begin{array}{c} \begin{array}{c} \\ \end{array} \end{array} \\ \end{array} \end{array} \\ \end{array} \\ \begin{array}{c} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ $

 $\leq \text{ED}(i-1, j-1) + \Delta s_i \tau_j$ $ED(i,j) = min \left\{ ED(i-1,j) + \right\}$ ED(i,j-1) + YDynamic Programming Table? Base Cases?

 $E \to E D (n, m)$ ED(i,j) $\leq ED(i-1,j-1) + \Delta s_i T_j$ ED(i-1,j) + YED(i, j) = min $(z \in D(z, j-1)) + \chi$

 $\rightarrow ED(n,m)$ ED(i,j)77 ED(0,0) = O $ED(i,0) = i \cdot Y$ $ED(0,j) = j \cdot \chi$

Edit Distance Algorithm E(o,o) = O $ED(i,0) = i \cdot Y$ $ED(0,j) = j \cdot Y$ For 2=1->n $For j = 1 \longrightarrow M$ $\left\{ \begin{array}{l} ED(i,j) = \min \left\{ ED(i-1,j-1), ED(i-1,j), ED(i,j-1) \right\} \\ + \Delta_{s_i T_j} + \chi + \chi + \chi + \chi \end{array} \right\}$ refurn ED(n,m)

Edit Distance Algorithm ED(o,o) = ORunning Time. - 1+n+m imitialization $ED(i,0) = i \cdot Y$ n outer iterations $ED(O,j) = j \cdot Y$ m mer iterations O(1) per iteration For 2=1->n $For j = 1 \longrightarrow M$ $ED(i,j) = \min \left\{ ED(i-1,j-1), ED(i-1,j), ED(i,j-1) \right\} + \Delta_{s_i T_j} + \chi + \chi + \chi$ refurn ED(n,m)

Edit Distance Algorithm ED(o, o) = ORunning Time. - 1+n+m imitialization $ED(i,0) = i \cdot Y$ n outer iterations miner iterations $ED(0,j) = j \cdot X$ O(1) per iteration For i=1->n $For j = 1 \longrightarrow M$ $ED(i,j) = \min \left\{ ED(i-1,j-1), ED(i-1,j), ED(i,j-1) \right\} + \Delta_{s_i T_j} + X + X + X + X$ Space refurn ED(n,m) DP Table has (m+1) x (n+1) Cells $\left(\bigcup_{i=1}^{n} \left((i + i) \right) \right) = \left((i + i) \right) = \left((i +$

	Distance	E Sequenc	e Alignment
. 	= - C R Y $= - C R Y$ $= - C R Y$	P = T = O G R	A : P : H : A : A : A : A : A : A : A : A : A
Ana	lignment	$\alpha = MON - Cros$	sing matching.
A = A + A + A + A + A + A + A + A + A +	Lignment 	\mathcal{A}	$s_i i \leq g$ $matching =$

Edit Distance = Sequence Alignment insertions - CRYPTEDGRAPHY ENCRYPTING ---deletions An alignment is a non-crossing matching.

Edit Distance =	Sequence Alignment
$= - C R Y P^{2}$ $= - C R Y P^{2}$ $= - C R Y P^{2}$	T = O G R A P H Y $I = I G G =$
An <u>alignment</u> is	a non-crossing matching.
An <u>alignment</u> is	a non-crossing matching.
An <u>alignment</u> is	2 Non-crossing matching.

Edit Distance = Sequence Alignment CRYPT DGRAPHY ENCRYPTING An alignment is non-crossing matching. ED Algorithm computed the distance Can we compute the alignment?

$ \begin{array}{cccccccccccccccccccccccccccccccccccc$		$ = ED((n_1, m_2)) $
	E D(i, j)	
ED(0,0)	$ \begin{bmatrix} & & & & & & & & & & & & & & & & & & $	
ED(i,j) = Min	ED(i-1, j-1) + ED(i, j-1)	$+ \cdot \cdot$

 $\underline{x} \rightarrow \overline{E}D(n, m)$ ED(i,j)Idea Minh of DP Table as a graph ED(0,0) $\leq ED(i-1, j-1) + \Delta s_i T_j$ ED(i-1,j) + YED(i, j) = min $ED(i, j-i) + \chi$

 $\frac{Vertices}{ED(i,j)} \equiv V_{i,j}$ * Edges Correspond to Cost of edit $\left(\begin{array}{c} 1 \\ - \end{array} \right) \left(\begin{array}{c} 1 \\ - \end{array} \right) \left(\begin{array}{c} - \end{array} \right) \left(\begin{array}{c} - \end{array} \right)$ S1T1 $\leq ED(i-1, j-1) + \Delta s_i T_j$ ED(i-1,j) + YED(i, j) = min $(\underline{z}, \underline{z}, \underline{j}, \underline{z}, \underline{z}) + \chi$

 $V_{n,m} = t$ $V_{v,m} = t$ $V_{v,m} = V_{v,m}$ Vertices $ED(i,j) \equiv V_{i,j}$ Correspond to Cost of edit $C = V_{0,0}$ Minimum Cost Alignment Shortest st-Path in ED Graph

Shortest path from V.o. to Vij equals ED(i,j). Theorem. <u>Pf</u>. By induction on i+j. Base Case. Shortest path from V00-> V00 = O = ED(0,0)

Shortest path from Voo to Vij equals ED(i,j). Theorem. Pf. By induction on i+j. Base Case. Shortest path from Voo-> Voo = O = ED(0,0) <u>Enductive Step</u>. Suppose shortest path to $V_{ue} = ED(u,e)$ for all u+l < i+j. V_{i-1}, j Vij Vi-1, Vj-1

Shortest path from Voo to Vij equals ED(i,j). Theorem Pf. By induction on i+j. Base Case. Shortest path from $v_{00} \rightarrow v_{00} = O$ = ED(0,0) $\frac{\text{Enductive Step}}{\text{for all } k+l < i+j,}$ $SP(v_{00} \rightarrow v_{ij}) = Vnin ZSP(v_{00} \rightarrow v_{i-1,j-1}) + \Delta_{SiTj}$ $SP(v_{00} \rightarrow v_{i-1,j}) + \chi$ $SP(v_{00} \rightarrow V_{i,j-1}) + Y$ Vi-i, j Vij Vi-1, Nj-1

Shortest path from Voo to Vij equals ED(i,j). Theorem Pf. By induction on i+j. Base Case. Shortest path from Voo-> Voo = O = ED(0,0) $\frac{\text{Enductive Step}}{\text{for all } k+l < i+j,}$ $SP(v_{00} \rightarrow v_{ij}) = vnin ZED(i-1, j-1) + \Delta s_{iTj}$ ED(i-1, j) + YED(i,j-1) + Y

Shortest path from Voo to Vij equals ED(i,j). Theorem Pf. By induction on i+j. Base Case. Shortest path from Voo-> Voo = O = ED(0,0) $\frac{\text{Fuductive Step}}{\text{for all } k+l < i+j}, \quad \text{Vul} = ED(4, l)$ $SP(v_{00} \rightarrow v_{ij}) = vnin ZED(i-1, j-1) + \Delta_{s_i T_j}$ ED(i-1, j) + YED(i,j-1) + Y= ED(i,j)

Shortest path from V.o. to Vij equals ED(i,j). Theorem Corollary. Diagonal adjes on shortest path from voo > vnn correspond to alignment ACGTTCA 7 A A C C C A

Edit Distance Algorithm ED(o, o) = OHow much space did we / really need? (1,0) = 1.8 $ED(0,j) = j \cdot X$ For i=1->n $For j = 1 \longrightarrow M$ $ED(i,j) = \min \left\{ ED(i-1,j-1), ED(i-1,j), ED(i,j-1) \right\} + A_{s_i T_j}$ refurn ED(n,m)

- ED(n, m)Curv \leq Prev (j-1) + $\Delta s_{1}T_{j}$ Prev (j) + Y $\left(\begin{array}{c} u \\ v \\ v \end{array} \right) = min$

Linear Space ED. $Prev(j) = j \cdot \gamma$ // m-entry 1D arrays Curr(j) = 0For i = 1 - p N $Curr(0) = i \cdot Y$ For $j = 1 \longrightarrow M$ $Curr(j) = min Z Prev(j-1), Prev(j), Curr(j-D), + \Delta_{siTj} + X + X$ Prev & Curr. refurn Curr (m)