

Inexact Matches

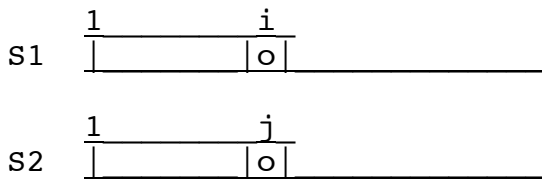
ABBA -> ABBA (exact)
 ABA -> AB-A (insertion/deletion)
 ABCA -> AB~~C~~A (substitution)
 ACA -> A-~~C~~A (insertion/deletion and substitution)

S1 - S2

Edit (Levenshtein) Distance: What is the min # of edits (insertion, deletions, substitutions) made to S1 to change to S2?

Example:

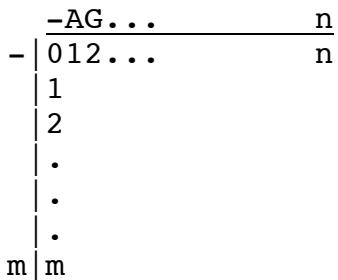
ABCA ---> ABCA
 ACA ---> A-CA <- this is the correct alignment
 ACA -x-> AC-A



E is edit distance between prefixes

$E[i, j] = \min:$
 -> $E[i-1, j-1] + 1$ if $S1[i] \neq S2[j]$
 + 0 otherwise
 -> $E[i, j-1] + 1$
 -> $E[i-1, j] + 1$

Fill dynamic programming array:



There's a much prettier picture here:
<http://lslwww.epfl.ch/biowall/VersionE/ApplicationsE/SequencE.html>

Lists of Important Concepts(? I was distracted making that table when he was talking about this)

Global Alignment

Local Alignment – Did substrings in S1 & S2 that have the lowest edit distance

Gap Penalties – Pay for gaps as a block