CMSC858P Algorithms for Biosequence Analysis

Script: March 6th 2008

Topic: Algorithms for Inexact Pattern Matching

Problem: how do we match two strings that differ in up to k characters?

Q: Why is it interesting in biosequence analysis?
A: We can have DNA/RNA with read failures or mutated DNA/RNA that we still like to match.

First idea to use suffix trees is not promising.

Second idea: assuming two strings S1 and S2 – can we compute the minimum number of actions (deletions, insertions, substitutions of characters) to transform S1 into S2?

We define this number as the **edit distance.**

Example:

```
S1  A C A G T C G A C C T
S2  A C G T G C A A C C
```

```
edit distance between S1 and S2 is 4
S1  A C A G T[2]C G A C C T
    | |   | |   |[3]| | |
S2  A C[1]G T G C A A C C[4]
```

```
to transform S1 into S2:
[1] delete S1[3]=A
[2] insert G after S1[5]
[3] substitute S1[7]=G with A
[4] delete S1[11]
```

```
to transform S2 into S1:
[1] insert A after S2[2]
[2] delete S2[5]=G
[3] substitute S2[7]=A with G
[4] insert T after S2[10]
```

The edit distance is symmetric: S1 -> S2 = S2->S1

To compute the edit distance we can build a table that we fill from the upper left corner to the lower right one using following rule:

```
D(i,j) = min( D(i-1,j)+1 , D(i,j-1)+1 , D(i,j) + t(i,j) )
with: t(i,j)=1 if S1[i] != S2[j] and 0 otherwise
```

Example:

		A	***C***	***A***	***G***	***T***	***C***	***G***	***A***	***C***	***C***	***T***
	0	1	2	3	4	5	6	7	8	9	10	11
A	1	0	1	2	3	4	5	6	7	8	9	10
C	2	1	0	1	2	3	4	5	6	7	8	9
G	3	2	1	1	1	2	3	4	5	6	7	8
T	4	3	2	2	2	1	2	3	4	5	6	7
G	5	4	3	3	2	2	2	3	4	5	6	7
C	6	5	4	4	3	3	2	3	4	4	5	6
A	7	6	5	4	4	4	3	3	3	4	5	6
A	8	7	6	5	5	5	4	4	3	4	5	6
C	9	8	7	6	6	6	5	5	4	3	4	5
C	10	9	8	7	7	7	6	6	5	4	3	4

The yellow path can be traced back in the end and describes the kind of actions to do. At some points the path might not be unique.

The running time and space complexity of the algorithm is O(n*m). The space complexity can be reduced in the case where just the edit distance is requires to O(2n) by row wise computation with two rows (that are exchanged during each step)