Reminders

- Homework Assignment 2 Due September 26
- Homework Assignment 3 Due October 3

Last Time

- Resource allocation problems (e.g. Knapsack)
- TSP
- Evaluation of computational effort
  - Ctools for detailed description of examples from lecture 3

Today

Pattern recognition:

- DNA sequencing
- Speech recognition
- Other pattern recognition problems (e.g. “sound hound”)

String Alignment DP

In many problems it is important to be able to evaluate the similarity of strings

Example**: Consider two strings:

1) C O U N T I N G
2) N T I G

We hypothesize that string 1 and 2 were the same but some characters were lost.

If spaces can be added to either string, what is the optimal alignment that maximizes the number of matches?

**This example is drawn from Abbas and Holmes, 2005, “Bioinformatics and Management Science: Some Common Tools and Techniques”, *Operations Research*, 52(2), 165-190

String Alignment DP

Following are possible alignments of the two sequences:

(A) C O U N T I N G

N T I G – – – –

(B) C O U N T I N G

– – – N T – I G

(C) C O U N T I N G

– – – N T I G –

Optimal String Alignment DP

States: (i, j) where i is the position in string 2 and j is the position in string 1

Actions: modify the strings by adding spaces:

$$a_t \in \{No\ Insert, Insert\ Space\ in\ String\ 1,\ Insert\ Space\ in\ String\ 2\}$$

Rewards: $r(i,j) = \begin{cases} 1 & if\ c_2(i) = c_1(j) \\ 0 & if\ c_2(i) \neq c_1(j) \end{cases}$

Optimality equations:

No space inserted

$$v(i,j) = max \begin{cases} r(i,j) + v(i-1, j-1) \\ v(i-1,j) \\ v(i,j-1) \end{cases}$$

Insert space string 2

Insert space string 1

Boundary Condition:

$v(i,j) = 0$, for positions on the boundary assuming a leading "dummy" character "∅"

DNA Alignment DP Problem Setup

Thinking of the problem in matrix form makes it easy to organize the optimality equations.

String 2 →

	ø	C	O	U	N	T	I	N	G
ø									
N									
T									
I									
G									

↑

String 1

String Alignment DP: Answer

Setup boundary condition:

	ø	C	O	U	N	T	I	N	G
ø	0	0	0	0	0	0	0	0	0
N	0								
T	0								
I	0								
G	0								

String Alignment DP: Answer

Continue filling in optimal value functions:

	ø	C	O	U	N	T	I	N	G
ø	0	0	0	0	0	0	0	0	0
N	0	0	0	0	1	1	1	1	1
T	0	0							
I	0	0							
G	0	0							

String Alignment DP: Answer

Continue filling in optimal value functions:

	ø	C	O	U	N	T	I	N	G
ø	0	0	0	0	0	0	0	0	0
N	0	0	0	0	1	1	1	1	1
T	0	0	0	0	1	2	2	2	2
I	0	0	0						
G	0	0	0						

DNA Alignment DP: Answer

Continue filling in optimal value functions :

	ø	C	O	U	N	T	I	N	G
ø	0	0	0	0	0	0	0	0	0
N	0	0	0	0	1	1	1	1	1
T	0	0	0	0	1	2	2	2	2
I	0	0	0	0	1	2	3	3	3
G	0	0	0	0	1				

DNA Alignment DP: Answer

Continue filling in optimal value functions :

	ø	C	O	U	N	T	I	N	G
ø	0	0	0	0	0	0	0	0	0
N	0	0	0	0	1	1	1	1	1
T	0	0	0	0	1	2	2	2	2
I	0	0	0	0	1	2	3	3	3
G	0	0	0	0	1	2	3	3	4

Q. What's the maximum number of character matches?

String Alignment DP: Answer

From the completed matrix the optimal solution is obtained by “tracing back”:

	ø	C	O	U	N	T	I	N	G
ø	0	0	0	0	0	0	0	0	0
N	0	0	0	0	1	1	1	1	1
T	0	0	0	0	1	2	2	2	2
I	0	0	0	0	1	2	3	3	3
G	0	0	0	0	1	2	3	3	4

Optimal alignment: C O U N T I N G

– – – N T I – G

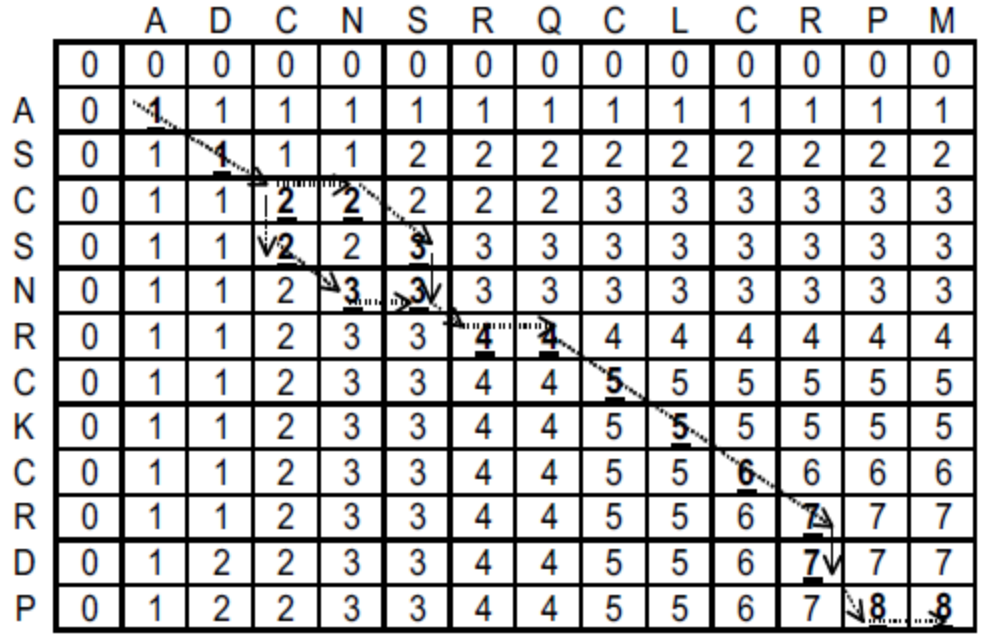
Example 2: String Alignment

"In-class Assignment"

Work in groups 2-3 to solve the alignment problem for the following two strings:

String 1: ASCSNRCKCRDP
String 2: ADCNSRQCLCRPM

Example 2: String Alignment



		A	D	C	N	S	R	Q	C	L	C	R	P	M
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	1	1	1	1	1	1	1	1	1	1	1	1	1
S	0	1	1	1	1	2	2	2	2	2	2	2	2	2
C	0	1	1	2	2	2	2	2	3	3	3	3	3	3
S	0	1	1	2	2	3	3	3	3	3	3	3	3	3
N	0	1	1	2	3	3	3	3	3	3	3	3	3	3
R	0	1	1	2	3	3	4	4	4	4	4	4	4	4
C	0	1	1	2	3	3	4	4	5	5	5	5	5	5
K	0	1	1	2	3	3	4	4	5	5	5	5	5	5
C	0	1	1	2	3	3	4	4	5	5	6	6	6	6
R	0	1	1	2	3	3	4	4	5	5	6	7	7	7
D	0	1	2	2	3	3	4	4	5	5	6	7	7	7
P	0	1	2	2	3	3	4	4	5	5	6	7	8	8

Starting from the bottom right hand corner retrace the path you took to that square.

Example 2: Optimal Alignment

This alignment problem has the following two optimal solutions:

```
A D C N S _ R Q C L C R _ P M
| | | | | | | | | | | | | |
A S C _ S N R _ C K C R D P _
```

```
A D C _ N S R Q C L C R _ P M
| | | | | | | | | | | | | |
A S C S N _ R _ C K C R D P _
```