Dynamic Programming

The following is an example of global sequence alignment using Needleman/Wunsch techniques. For this example, the two sequences to be globally aligned are

G A A T T C A G T T A (sequence #1)
G G A T C G A (sequence #2)

So M = 11 and N = 7 (the length of sequence #1 and sequence #2, respectively)

A simple scoring scheme is assumed where

- $S_{i,j}$ = 1 if the residue at position i of sequence #1 is the same as the residue at position j of sequence #2 (match score); otherwise
- $S_{i,j}$ = 0 (mismatch score)
- w = 0 (gap penalty)

Three steps in dynamic programming

1. Initialization
2. Matrix fill (scoring)
3. Traceback (alignment)

Initialization Step

The first step in the global alignment dynamic programming approach is to create a matrix with M + 1 columns and N + 1 rows where M and N correspond to the size of the sequences to be aligned.

Since this example assumes there is no gap opening or gap extension penalty, the first row and first column of the matrix can be initially filled with 0.

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0											
G	0											
A	0											
T	0											
C	0											
G	0											
A	0											

Matrix Fill Step

One possible (inefficient) solution of the matrix fill step finds the maximum global alignment score by starting in the upper left hand corner in the matrix and finding the maximal score $M_{i,j}$ for each position in the matrix. In order to find $M_{i,j}$ for any i,j it is minimal to know the score for the matrix positions to the left, above and diagonal to i, j. In terms of matrix positions, it is necessary to know $M_{i-1,j}$, $M_{i,j-1}$ and $M_{i-1,\ j-1}$.

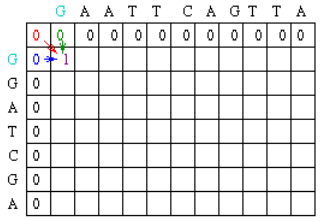
For each position, $M_{i,j}$ is defined to be the maximum score at position i,j; i.e.

```
Mi,j = MAXIMUM[
     Mi-1, j-1 + Si,j (match/mismatch in the diagonal),
     Mi,j-1 + w (gap in sequence #1),
     Mi-1,j + w (gap in sequence #2)]
```

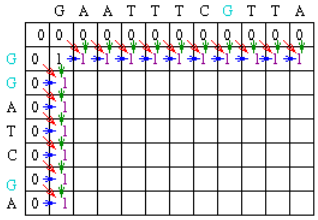
Note that in the example, $M_{i-1,j-1}$ will be red, $M_{i,j-1}$ will be green and $M_{i-1,j}$ will be blue.

Using this information, the score at position 1,1 in the matrix can be calculated. Since the first residue in both sequences is a G, $S_{1,1}$ = 1, and by the assumptions stated at the beginning, w = 0. Thus, $M_{1,1}$ = MAX[$M_{0,0}$ + 1, $M_{1,\ 0}$ + 0, $M_{0,1}$ + 0] = MAX [1, 0, 0] = 1.

A value of 1 is then placed in position 1,1 of the scoring matrix.



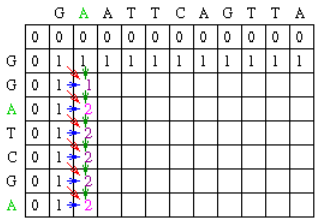
Since the gap penalty (w) is 0, the rest of row 1 and column 1 can be filled in with the value 1. Take the example of row 1. At column 2, the value is the max of 0 (for a mismatch), 0 (for a vertical gap) or 1 (horizontal gap). The rest of row 1 can be filled out similarly until we get to column 8. At this point, there is a G in both sequences (light blue). Thus, the value for the cell at row 1 column 8 is the maximum of 1 (for a match), 0 (for a vertical gap) or 1 (horizontal gap). The value will again be 1. The rest of row 1 and column 1 can be filled with 1 using the above reasoning.



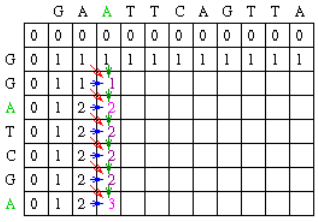
Now let's look at column 2. The location at row 2 will be assigned the value of the maximum of 1(mismatch), 1(horizontal gap) or 1 (vertical gap). So its value is 1.

At the position column 2 row 3, there is an A in both sequences. Thus, its value will be the maximum of 2(match), 1 (horizontal gap), 1 (vertical gap) so its value is 2.

Moving along to position column 2 row 4, its value will be the maximum of 1 (mismatch), 1 (horizontal gap), 2 (vertical gap) so its value is 2. Note that for all of the remaining positions except the last one in column 2, the choices for the value will be the exact same as in row 4 since there are no matches. The final row will contain the value 2 since it is the maximum of 2 (match), 1 (horizontal gap) and 2(vertical gap).



Using the same techniques as described for column 2, we can fill in column 3.



After filling in all of the values the score matrix is as follows:

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	2	2	2	2
A	0	1	2	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3	3
C	0	1	2	2	3	3	4	4	4	4	4	4
G	0	1	2	2	3	3	4	4	5	5	5	5
A	0	1	2	3	3	3	4	5	5	5	5	**6**

Traceback Step

After the matrix fill step, the maximum alignment score for the two test sequences is 6. The traceback step determines the actual alignment(s) that result in the maximum score. Note that with a simple scoring algorithm such as one that is used here, there are likely to be multiple maximal alignments.

The traceback step begins in the M,J position in the matrix, i.e. the position that leads to the maximal score. In this case, there is a 6 in that location.

Traceback takes the current cell and looks to the neighbor cells that could be direct predecessors. This means it looks to the neighbor to the left (gap in sequence #2), the diagonal neighbor (match/mismatch), and the neighbor above it (gap in sequence #1). The algorithm for traceback chooses as the next cell in the sequence one of the possible predecessors. In this case, the neighbors are marked in red. They are all also equal to 5.

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	2	2	2	2
A	0	1	1	2	2	2	2	2	2	2	2	3
T	0	1	2	2	3	3	3	3	3	3	3	3
C	0	1	2	2	3	3	4	4	4	4	4	4
G	0	1	2	2	3	3	4	4	5	5	5	5
A	0	1	2	3	3	3	4	5	5	5	5	6

Since the current cell has a value of 6 and the scores are 1 for a match and 0 for anything else, the only possible predecessor is the diagonal match/mismatch neighbor. If more than one possible predecessor exists, any can be chosen. This gives us a current alignment of

```
(Seq #1)      A
              |
(Seq #2)      A
```

So now we look at the current cell and determine which cell is its direct predecessor. In this case, it is the cell with the red 5.

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	
G	0	1	1	1	1	1	1	1	1	1	1	
G	0	1	1	1	1	1	1	1	2	2	2	
A	0	1	2	2	2	2	2	2	2	2	2	
T	0	1	2	2	3	3	3	3	3	3	3	
C	0	1	2	2	3	3	4	4	4	4		
G	0	1	2	2	3	3	4	4	5	5		
A											6	

The alignment as described in the above step adds a gap to sequence #2, so the current alignment is

```
(Seq #1)      T A
                |
(Seq #2)      _ A
```

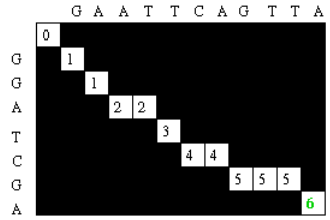
Once again, the direct predecessor produces a gap in sequence #2.

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0		
G	0	1	1	1	1	1	1	1	1	1		
G	0	1	1	1	1	1	1	1	2	2		
A	0	1	2	2	2	2	2	2	2	2		
T	0	1	2	2	3	3	3	3	3	3		
C	0	1	2	2	3	3	4	4	4	4		
G	0	1	2	2	3	3	4	4	5	5	5	
A											6	

After this step, the current alignment is

```
(Seq #1)      T T A
                  |
              _ _ A
```

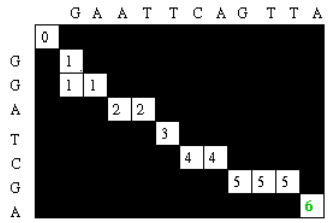
Continuing on with the traceback step, we eventually get to a position in column 0 row 0 which tells us that traceback is completed. One possible maximum alignment is :



Giving an alignment of :

```
G A A T T C A G T T A
|   |   | |   |     |
G G A _ T C _ G _ _ A
```

An alternate solution is:



Giving an alignment of :

```
G _ A A T T C A G T T A
|     |   | |   |     |
G G _ A _ T C _ G _ _ A
```

There are more alternative solutions each resulting in a maximal global alignment score of 6. Since this is an exponential problem, most dynamic programming algorithms will only print out a single solution.

Now you are ready to explore an example using an advanced scoring scheme.

BL5495 Home Page
Eric C. Rouchka
ecr@ibc.wustl.edu