BA3 Dynamic Programming Practical Exampled

The EMBOSS sequence analysis suite, provided from the EBI (European Bioinformatics Institute), can be installed on local machines or servers (see http://emboss.sourceforge.net/) and provides sequence alignment and analysis tools. We can use these tools to test our code. For this practical we will use the versions of these tools available from the EBI web site. You can run Smith Waterman using the water tool, Needleman-Wunsch using the needle software and there is also the tool called `dotmatcher`, that can be used to visualise the relationship between a pair of sequences.

The URLs for these individual tools

Water - https://www.ebi.ac.uk/jdispatcher/psa/emboss_water

Needle- https://www.ebi.ac.uk/jdispatcher/psa/emboss_needle

Dotmatcher - https://www.ebi.ac.uk/jdispatcher/seqstats/emboss_dotmatcher

Next week we will generate our own sequence alignment tool. We can use this to explore the further properties of sequence alignment. We can then use the implementation provided by EMBOSS a reference implementation. We can test the functionality of our code against the EMBOSS implementation.

At the moment we are not using the gap weights that are used in the Smith Waterman paper or those typically used for global alignments. We can use different weights to suite our question and the alignment generated is only guaranteed to be the best possible alignment given the weights we use. Note also that our alignment can generate more than one alignment of equal score- that has equivalent score but a different traceback path through the matrix. In this case, for simplicity we would typically output just one of the alignments.

We can write sequence data into FASTA format by just adding a simple header starting with a “>” character- this allows data to be loaded in this format into alignment tools.

>seq1

GTATATC

Our sequence just has GATC characters- but more complete aligners can take into account redundancy and the biochemical properties of bases.

Global Alignment Part A

Use the weights Match = 1, Mismatch = -1 and Gap Insert Weight -1. This is a global alignment so end gap/insert weights and internal gap/inserts have the same weight=-1. One the table has been filled in, from the bottom right cell traceback the path that led to the highest score back to the top left cell. It can help traceback to draw arrows on the cells showing the path that was used to generate the score in each cell.

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

Write out your alignment here…

Dynamic Programming Practical Part B

Dynamic programming approaches can be used to solve the task of identifying the shortest path between two elements in a grid. This is similar to the image segmentation example shown in class. This algorithm is often used in Bioinformatics for identifying the shortest path through a graph representing a biological network or signalling pathway or other set of interconnected components. Several algorithms exist to solve this problem, but here we will use dynamic programming to find the shortest path across an array of elements.

3	3	3	3	4	5	6	7	8	9
2	2	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
From	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	2	3	4	5	6	7	8	9
3	3	3	3	4	5	6	7	8	9
4	4	4	4	4	5	6	7	8	9
5	5	5	5	5	5	6	7	8	9
6	6	6	6	6	6	6	7	8	To Here

The above example shows the basic concept – just the count of the number of steps it takes to move from one position to another. In the above case, finding the shortest path is not very challenging. Imagine instead we would like to work out the shortest path between **`From`** and **`To Here.`** To solve this problem, we simply work out the number of steps required to go between `From` and `To Here`. We record all possible first step positions, and then from each of those we make another step and record the total number of steps take to traverse from the square From to the square To here. We also avoid the squares coloured in yellow. Then we can use a trace back approach to plot the traversal.

From									
									To Here

Dynamic Programming Practical Part C

There are some extra considerations when interpreting results from alignment algorithms such as Smith Waterman or Needleman-Wunsch. Here is an example of one additional major consideration.

Installed on the servers are alignment programmes from the EMBOSS sequence analysis suite (http://emboss.sourceforge.net/) that offers many different sequence alignment algorithms. Some useful programmes for us are called `water`, `needle` and `dotmatcher`. These provide tools to align larger and more complex sequences using dynamic programming algorithms. To run these programmes just type their name on the command line, if you type " `-h`" after the name you will be given a help page.

- Run the `water` programme on the two example sequences `myseq.fasta` and `myseq2.fasta`. This will generate a local alignment from these two input files. What are your conclusions from this alignment about the relationship between these sequences?
- Now run `dotmatcher` that generates a "dot plot" from the sequences, where each dot in the panel represents one base pair that is identical between sequences. To eliminate noise this dot plot is them smoothed (to remove very short segments with identical sequence) and only runs of identical sequence above a certain threshold score are retained.
- Look at the alignment and the dot plot. What do they tell you about these sequences?
- [Those interested in the biology]. If you want to try to understand the detailed biology of these sequences, the first step is to search DNA sequence databases using Blast. Once you know identity of the sequence, this should give additional meaning to the alignment results.

Example run command for dotmatcher

Note that here I have just compared one sequence (myseq.fasta) to itself! Here I set windowsize =10 and threshold=40. I use the DNAfull matrix. The web page generates a graphic that can be downloaded directly from the web page.