Sequence alignment in bioinformatics – Craig L. Zirbel – Summer 2009

Every living organism has DNA, a sequence of four nucleotides A, C, G, and T. DNA is copied and transmitted from parent to child, but from one generation to the next errors in copying can occur, with one nucleotide being replaced by another, or a new nucleotide being inserted, or an existing nucleotide being deleted. Over many generations, organisms with a common ancestor can end up with fairly different DNA sequences. Now that it is relatively easy to sequence part or all of the DNA of an organism, it is helpful to take DNA sequences and determine how they can most easily be thought of as having come from a common ancestor. The goal is to try to line up as many letters or portions of the sequence that are the same, to minimize the number of substitutions, insertions, and deletions that would have had to happen if the sequences came from a common ancestor.

1. Consider the short sequences ATCTGAT and TCCTGCT. Here are three possible alignments of these sequences, using hyphens (called gaps) to show where there are insertions or deletions:

   ```
   ATC-TGA-T        ATCTGA--T---        ATC---TGA-T
   -TCCTG-CT        ---T--CCTGCT        ---TCCTG-CT
   ```

   The first alignment has 5 matches, the second has 2, and the third has 3. Another way to look at it, the first alignment has 4 differences between the two sequences, the second has 10, and the third has 8. It's easier to make fewer changes in DNA than more, so we prefer the first alignment.
2. Given two DNA sequences, how do we find the alignment of the two that has the most matches? Here are two new short sequences to try to align: TATAGAGC and AATATAGCGC. Take a minute to write down three possible alignments and see how many matches you can make.

3. There are many, many possible alignments even for medium-length sequences, so we need a systematic method. The idea of dynamic programming is to solve smaller problems and use those solutions to solve bigger and bigger problems. Let's start with really small problems related to these two sequences. What is the optimal way to align the first two characters of the two sequences? The first three? These are small enough that we can do them in our heads. Let's fill in small alignments like this in a grid that will help us organize our work. We leave a gap character at the beginning in case we need to align a letter with gaps. Each cell shows the optimal way to align the sequences up to that point. For instance, the cell with bold characters in it shows how to align the first three characters of each sequence.
4. The alignment that goes in each new cell can be figured out from looking at the smaller alignments above, to the left, or diagonally above and to the left of the current cell you're working on, as we'll explain next.

	-	**T**	**A**	**T**	**A**	**G**	**A**	**G**	**C**
-		T -	TA --	TAT ---	TATA ----	TATAG -----	TATAGA ------	TATAGAG -------	TATAGAGC --------
A	- A	T- -A	TA -A	TAT -A-	TATA -A--				
A	-- AA	T-- -AA	TA- -AA	TAT- -A-A					
T	--- AAT	--T AAT	TA-- -AAT	**TA-T** **-AAT**					
A	---- AATA								
T									
A									
G									
C									
G									
C									

Think of how to find the optimal alignment of TATA and AA, which goes in the empty cell in the 3rd row, 5th column. We can think of three ways to get an alignment of these sequences from smaller ones:

```
TAT plus A        TATA plus -         TAT- plus A
-A-       A       -A--       A        -A-A       -
```

These are the smaller alignments above and to the left, above, and to the left of the cell we're trying to fill in. The first of the three options is the best, since it adds a new column to the alignment that brings the total number of matches to 2. Fill in more cells in the table on the previous page, always adding new columns to previous alignments. You do not need to finish the whole table!

5. As you work through the table on the previous page, you will notice that you can add more and more matches, but only when you are in a cell whose column and row heading are the same letter. Instead of writing down all of the intermediate alignments, let's just keep track of how many matches there are in the alignment that goes in each cell. It is the maximum of the numbers in the cell above, the cell to the left and, if the column and row headings have the same letter, the number up and to the left plus 1. It helps to draw diagonal lines in all places in the table where you can add a matched column. This way you don't miss any opportunities to add a match to the alignment. Fill in the whole table below in this way. Notice that this is something like the Manhattan tourist problem, only now we can move diagonally as well, and that we only increase the number of matches when we can move diagonally.

6. After you fill in the table below, look at the lower-right corner. There should be a 7 there, and a diagonal line back to a 6 in the cell up and to the left. This indicates that there is an alignment with 7 matches, and that the last step of creating that alignment is to add a column with two C's in it.

7. Working back through the table, write down the alignment of the two sequences that has 7 matches. When you come to a cell without a diagonal line, go up and then to the left until you come to a cell with a diagonal line in it. This takes care of letters that are aligned to gaps in a consistent way.

	-	T	A	T	A	G	A	G	C
-	0	0	0	0	0	0			
A	0	0	1	1	1				
A	0	0	1	1					
T	0	1	1	2					
A	0	1							
T	0								
A									
G									
C									
G									
C									