Introduction to Bioinformatics
Problem Set 5: Loops, mapping, dotplots, and Blast

1. Why did the search for FMRP in *Drosophila* work well when human FMRP protein was used as the query but failed so abysmally when the corresponding human gene was used as the query? Now's the time to find out.

1a. Hearken back to the tour *Search for FMRP in Drosophila* and consider item 14, the comparison of the human and *Drosophila* proteins. The comparison begins with a stretch of very high amino acid similarity, from the initial M (methionine) up to YK (tyrosine-lysine). How many amino acids are in that stretch?

1b. How many nucleotides are required at the beginning of the gene to encode those amino acids (M through YK)?

Let's get those nucleotides, from the human gene and from the *Drosophila* gene. If the amino acid sequences are extremely similar, then so should be the nucleotide sequences, no? The protein sequence found by Blast comes with a link to the corresponding gene, with the GenBank ID of LD09557. You determined during the tour that the GenBank sequence does not begin with the FMR gene but includes upstream sequence. You'll recall that the gene begins at nucleotide 423.

1c. In BioBIKE, DEFINE a variable (maybe `fly-seq`) as the beginning of the fly gene, using SEQUENCE-OF and specifying the FROM-GENBANK option. The argument should be the GenBank ID (in quotes). Also specify values for FROM and TO, so that you bring in only those nucleotides from the beginning of the gene to the end of the portion of the gene encoding Y and K. To check if you were successful, get the TRANSLATION-OF `fly-seq` and compare it to the amino sequence you considered in **1a**.

1d. Retrace your steps in the tour and find the GenBank ID for the human FMR gene and what nucleotide coordinate the gene starts with. DEFINE a variable (maybe `human-seq`) as in 1c, but using the appropriate GenBank ID.

1e. Align these two DNA sequences, using the ALIGNMENT-OF function. The argument should be a LIST consisting of two items: `fly-seq` and `human-seq`.

1f. Does the alignment look as good as the amino acid alignment? Why not? Do something to test your theory.

2. Make your own DOTPLOT function

The strategy will be to slide a window across one sequence and compare that window against the second sequence. The matches for each window will provide information for one line of the dotplot, using the PLOT function to display the dotplot.

PLOT requires a set of pairs of numbers, e.g. ((1 1) (2 4) (3 9))

2.A. Play with the PLOT function, providing it with a set of different number pairs, until you understand what it does with the pairs.

As a test of the function (and to give us raw material as we build it), we'll try it on the sequences of two small phage: Chlamydia phage 3 (chla-3) and Chlamydia phage phicpg1 (phicpg1).

2.B. DEFINE seq1 as the SEQUENCE-OF chla-3 and seq2 as the SEQUENCE-OF phicpg1

2.C. DEFINE window-size as 11 (each comparison will be 11 nucleotides)

2.D. DEFINE start-of-window as 1 (we'll start the first window at coordinate 1

2.E. DEFINE end-of-window as… as what? Knowing the start-of-window and the window-size, how would you calculate the end coordinate? Use SUM-OF to do this.

2.F. DEFINE window as SUBSTRING seq1 FROM start-of-window TO end-of-window

2.G. DEFINE matches as MATCHES-OF window IN SEQ2. It will be more convenient if the function returns only the position of the match, so specify the POSITION-ONLY option.

2.H. Now we need to turn this information into number pairs. The first number in each pair will be the coordinate of the window (for simplicity, we'll use start-of-window). The second number of each pair will be the coordinate of a match. A pair will be formed by:

LIST start-of-window n

where n is one of the coordinates returned by MATCHES-OF. We don't know how many matches will be found, but we can MAP the LIST function over the list of matches:

APPLY-FUNCTION LIST start-of-window n
REPLACING n WITH matches

DEFINE number-pairs as this function (i.e., APPLY-FUNCTION …)

If all has gone well, you should have the number pairs necessary for PLOT to display one line of a dotplot of seq1 vs seq2. To generalize, i.e. repeat these operations over all windows within seq1, we need a loop.

2.I. Bring down a FOR-EACH loop. We'll treat seq1, seq2, and window-size as constants for the moment. The loop will presume they already exist. The loop will repeat lines 2.E through 2.H and loop through all values of start-of-window. Open up the body of the FOR-EACH loop and create holes for each function made in lines 2.E. through 2.H. Then drag each of those four functions (in order) into one of the four holes.

2.J. Open up a primary control *number from n1 to n2* box. Replace *var*iable with start-of-window. The first value should be 1 of course, so that the window begins at the beginning of seq2. What about the last value? Surely it's related to the length of seq2. Is it equal to the length? Devise a calculation for the last value and put it in the last value box.

2.K. Open up a Results Section, using WHEN… APPEND. Why WHEN? You don't always want to add to the results, just when you found a match. Why APPEND? The difference between APPEND and COLLECT is subtle. We'll explore it later. For now, trust me.

2.L. WHEN what? WHEN matches has a value. That's expressed simply as WHEN matches. Put the variable matches in the condition box.

2.M. APPEND what? APPEND all the number-pairs you found. Put the variable number-pairs in the value box.

2.N. Execute the loop. Do you get the expected collection of number pairs?

2.O. Put the loop within a PLOT function. The easiest way to do this is to mouse over the action icon of FOR-EACH, click Surround, and then click PLOT in either the INPUT-OUTPUT menu or the ALL menu. Execute the resulting function.

2.P. How do you interpret the dotplot?

Now that you have a loop that works, we want to package it into a function, something that you can use like any other BioBIKE function

2.Q. Collapse the now huge PLOT function, by mousing over its action icon and clicking Collapse. Then go the same menu , click Name me, and give it a descriptive name (e.g. Plot Dotplot)

2.R. Bring down the DEFINE-FUNCTION function from the DEFINITION menu.

2.S. Drag Plot Dotplot into the body of the function.

2.T. This function should be given two arguments, seq1 and seq2. Create a second *arg* box using the *More* icon. Then type seq1 in the first box and seq2 in the second

2.U. Allow specification of the value of window-size. Do this by opening the SPECIFICATION section, click Reveal keyword arguments, mouse over the keywords icon, and click keys. Then enter window-size into the *var*iable box and provide a default value, perhaps 12.

2.V. Finally, give the function a name. Dotplot might work for you. Then execute the DEFINE-FUNCTION. You should see a FUNCTION box appear in the palette, and your new function will be represented on its menu.

2.X. Test your new function, bringing it into the workspace from the FUNCTION menu and providing it with various input values and

3. **Display a list of the largest proteins** amongst prokaryotic viruses. Include in the list the name, length, and description of the proteins.

4. **Why are large proteins so large?** Try running a dotplot of a large protein against itself. What do you see? What does it mean?