DOTPLOTS – A graphical comparisonn or sequences:
==========

Here is offered a swift discussion of dotplots.

A simple but effective way to visualise similarities between pairs of sequences.

Knowing how dotplots are constructed can assist the understanding of other, more complex, tools.

<CLICK>

First the fundamentals.

<CLICK>

The dotplot is drawn upon Cartesian axes.

<CLICK>

The two sequences to be compared are laid along the horizontal and vertical axes.

For simplicity, I use a DNA example, but the same software is used for both protein and DNA sequences.

The rules differ, but not the algorithm.

Dotplots are constructed by comparing fixed length words, windows or spans, of residues from each sequence.

A decision is required concerning the size of the word to be used.

<CLICK>

11 base pairs is reasonable for DNA sequences.

<CLICK>

Every contiguous set of 11 base pairs from the horizontal sequence is compared with every 11 base pair word of the vertical sequence in turn.

<CLICK>

The words are compared by considering each pair of adjacent bases.

<CLICK>

A simple numeric evaluation of how similar any two words are is required.

For this a scoring scheme is needed.

For DNA, simply specifying that matching bases score 1 and mismatching bases do not, is fine.

<CLICK>

In the illustration here, the two words being compared would score 9 brownie points out of a possible 11.

Now a declaration of significance is required.

How big must the word score be before it should be regarded as likely to be biologically meaningful.

<CLICK>

8 - I pontificate with assurance!

<CLICK>

Which makes 9 sufficient for celebration.

<CLICK>

A dot is placed to indicate the position of the two windows estimated to be possessed of message!

<CLICK>

Repeat for every possible pair of words in both sequences and a enlightening flurry of dots will jostle eagerly into view.

<CLICK>

Roughly diagonal, more or less unbroken, runs of dots will indicate regions between the two sequences that are interestingly similar, possibly for biologically significant reasons.

<CLICK>

All very vague, but dotplots are intended to give a comprehensive overview rather than any detail.

A dotplot reports all regions between two sequences showing promise of homology.

A textual alignment is required to see the individual residue level detail.

<CLICK>

As already mentioned, the Identity Matrix is perfectly adequate for fully determined DNA sequence comparisons.

<CLICK>

There are some advantages in other matrices however. The EMBOSS default DNA scoring matrix being an example.

<CLICK>

The score range used in this instance is 5 for a match and -4 for a mismatch.

The use of negative numbers is irrelevant for dotplots, but has a justification for other uses of this matrix.

<CLICK>

The wider spread of scores allows the easy inclusion of DNA ambiguity codes.

<CLICK>

Reasonably accurate expansion of the matrix, without the need to introduce non-integer values, is possible.

For a computer, integer arithmetic is very much faster than floating point (or non-integer) arithmetic.

The reason for allowing ambiguity codes in the sequences being compared is to avoid the software failing if one or two ambiguous positions are present.

More than a very few ambiguity codes in either sequence would render the whole operation pointless.

There is little enough message in just the sequence of DNA as it is.

Just 4 letters! A 25% chance of match in the best of circumstances.

It could be argued that the effort to use “accurate” values for all possible pairings of letters, as you see here, is wasted.

Other matrices offer a much less precise scheme for scoring, but are just as effective.

<CLICK>

Clearly, for Protein comparisons, a different matrix is essential.

It cannot be as simplistic as to just look for matching letters!

Amino acid properties must be reflected in some fashion.

More of this later, for now, just accept that protein scoring matrices are trickier by far than those used for DNA comparisons!

<CLICK>

Particularly when comparing very long DNA sequences, or using a computer built in 1985, it is

useful to speed things up.

The most common strategy to make things faster, is to instruct the software to regard only perfectly matching words as significant.

<CLICK>

Having accepted this compromise, the software has new options.

<CLICK>

It could proceed as previously described, with a scoring scheme, and a word size

<CLICK>

and a cut-off value that is the maximum possible.

This works, but there is no gain in speed.

<CLICK>

Alternatively, the words could just be compared as strings of letters, or simple bit patterns.

<CLICK>

There is no need for laborious computations of scores and comparisons with cut-off values.

Simply matching words is massively faster than the conventional dotplot strategies.

<CLICK>

Most dotplot software offers word matching in one form or the other as an option.

<CLICK>

Word matching dotplots are in regular employment in these times when whole genomes need to be compared.

Long and strong regions of similarity are readily highlighted.

<CLICK>

There are three parameters that should be considered when computing a dotplot.

<CLICK>

The scoring scheme.

Anything sensible for DNA will be fine.

For protein sequences the choice depends primarily on the evolutionary distance supposed between the two proteins being compared.

Not such a critical choice for a dotplot as for a textual alignment I would suggest.

<CLICK>

The cut-off score.

Clearly, the higher the cut-off, fewer are the dots that might be expected.

But, the greater the confidence that those dots might mean something.

The lower the cut-off, the less likely it is that meaningful matches will be missed.

However, the more likely that matches detected will just be noise.

<CLICK>

The word size

Arguably the most important dotplot parameter.

The smaller the word size the more “accurate”, in some sense, the plot will be.

However, choose too small and, although nothing will be missed, the plot will be infested with dots that represent nothing but noise.

Larger word sizes generate plots that sometimes miss small features and give only a “broad brush overview”.

But a “broad brush overview” is very often exactly what is required! The detail can be left to textual alignment to reveal.

Computing dotplots is generally fast, it is not such a bad plan to create more than one, using a variety of parameter values.

Then, choose the one that best fits what you wanted to see in the first place!

There are a number of slides that are hidden in this power point. If you really feel the urge for some examples and further life enhancing revelations, unhide them and have a browse.

Otherwise, I stop here and leave you to admire the irritating “The End” slide I found so amusing a decade or so ago.

<CLICK>