(25 March 2014)

Finding Local Repeats

I. How to find ill-defined tandem repeats?

The discussion of tandem repeats in the last notes (Repeats, Tandem Repeats, and Pattern Matching) ended on a sour note. We considered two ways of identifying tandem repeats by pattern matching. The first method worked well if the repeating unit was known and its pattern was highly conserved. The second method worked well even if the repeating unit was unknown, but the pattern had to repeat exactly – no ambiguous positions allowed.

Unfortunately, ambiguous position are common in real tandem repeats (as you saw in the repeats found by Mazel et al), and if the repeating unit is already known, that means someone has been to the problem before you. Generally you have to find the repeating unit yourself. So, we need more powerful methods. There are many that are reasonable candidates. Here I present two, primarily because they illustrate concepts that are generally useful beyond the problem at hand.

II. Dot plots – A general method to find local repeats

Dot plots are graphical representations of comparisons between two sequences, commonly used to give a visual view of where similarities occur. **Fig. 1** shows a typical dot plot, comparing the sequences of two mycobacteriophages, L5 and Che12. Imagine the Che12 sequence spread out from left to right and the L5 sequence spread out from bottom to top. Each point in the graph then represents a position on the L5 genome and a position on the Che12 genome. If a comparison of the sequences at those positions has a desired degree of similarity, then the point is black, otherwise not.

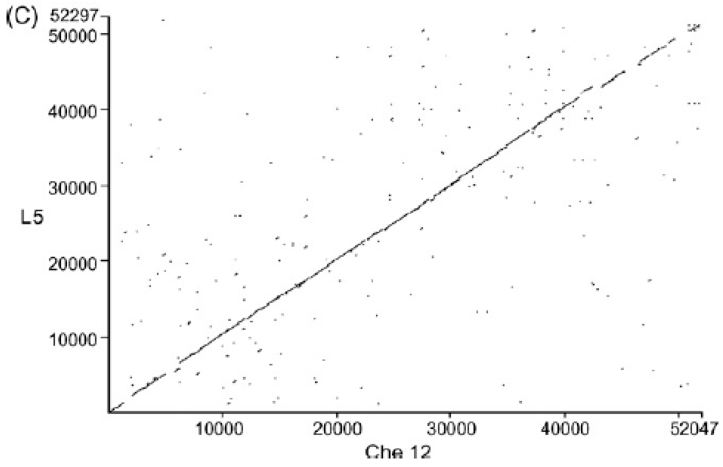


Fig. 1: Dot plot comparing the sequences of mycobacteriophages L5 and Che12. Taken from Fig. 2 of Gomathi NS, et al (2007). Comput Biol Chem 31: 82-91.

SQ1. From my description, what parts (what coordinates) of L5 and che12 are least similar to each other?

SQ2. What graph would you get if Che12 were the sequence used for both axes?

SQ3. What graph would you get if Che12 were the sequence used for both axes and the entire genome of Che12 consisted solely of A's?

There was a lot of hand waving in the previous paragraph. What did I mean by "desired degree of similarity"? How was that similarity determined? We need to answer these questions, because I hope you see from thinking about SQ2 and SQ3 that Dot Plots may be very useful for finding repeated sequences within a single sequence.

In order for you to gain insight into how dot plots work, pay a visit to an implementation of dot plots, at http://www.vivo.colostate.edu/molkit/dnadot/ (also accessible through the course web site, Resources & Links page).* It will take a few seconds for the page to load fully, but when it does, you'll see three boxes near the bottom of the page. This page enables you to make a dot plot (also called dot-matrix) from DNA that you provide.

Try it out! Enter 20 random nucleotides into the left box (DNA Number 1). Don't sit on the A key -- I mean 20 ***random*** nucleotides. Having some trouble producing them? That's OK. Humans shouldn't be asked to be random. Get into BioBIKE and pull down the RANDOM-DNA function, selecting the LENGTH-Of option (give it the number 20). After executing the function, click the result box at the bottom of the page, highlight the 20 nucleotides, and copy them. Then paste them into the left box of the Dot Plot web site. Finally, click the **Copy DNA1 → DNA2** button, and click **Make Plot**.

SQ4. Compare your graph to that in Figure 1 on the previous page. By analogy, what do you think would be the X and Y axes of your graph?

SQ5. Change the value in the Window Size box from 9 to 5 and click Make Plot again. Then repeat the procedure with Window Sizes of 3 and 1. Ideas on what it all means? When the window size was 1, you probably got some boxes of dots. What's the relationship between the size of the boxes and the sequence you entered?

Now that you've seen what the program produces, you might like some enlightenment on how it works. Scroll to the top of the screen and click **Background information on….** [pause] ...With that instruction behind you…

SQ6. Copy/paste the sequence below into both left and right boxes, make sure Window Size is 9, and click Make Plot. Why is the diagonal accompanied by two shorter parallel lines?

GGCCACTGCCCAAGGCCACTGCCCAACCCTCCATCATAAAACTTGGGCTTGGG

SQ7. Click the first point in the lower parallel line. You'll see in the yellow box above the output box the positions in DNA1 and DNA2 represented by that point (of course DNA1 and DNA2 are identical). Identify the nucleotides at those positions. Move down the parallel line, clicking as you go on each point. What is the meaning of the parallel line?

SQ8. Now for the main event... display and copy the SEQUENCE-OF the cpeBA region (all 6867 nucleotides of it), using the DISPLAY-FASTA option to display only nucleotides. Then copy the sequence (without the header) into the left and right Dot Plot boxes. Finally make a dot plot. Anything interesting?

You probably see a blip on the diagonal. Where? Click on its boundaries as best you can. To zoom in on it, try the following:

* ***Warning!*** In order to run Dot Plot, you may need an up-to-date version of Java on your computer and/or listing the Dot Plot web site on a white list of safe sites. On a Windows machine, you can get to this list by going to your system's Control Panel, clicking Java, then on the resulting Java Control Panel, click Security, and Edit Site List. Once at the list, click Add, paste in the URL, and click OK.

SQ9. Display and copy the SEQUENCE-OF the cpeBA region as before, but this time FROM 1001 TO 2000. Now the dot plot has some detail, and it's possible to get the precise boundaries. Locate this region on the highlighted cpeBA region you made went you went through the last notes **(being sure to add 1000 to the coordinates given by Dot Plot). Do you agree with Dot Plot's assessment that you're looking at a large, fuzzy region of repeated sequences?**

SQ10. There's another blip on the diagonal shown by Dot Plot, closer to Dot Plot coordinate 60 (real coordinate 1060). Find its boundaries and highlight it in the cpeBA region. What's the repeating unit? Is it one of Mazel et al's STRRs?

The last two trials of Dot Plot illustrate both the power and the limitations of this method. The power... knowing nothing about tandem repeats, Dot Plot nonetheless shows a blip on the diagonal at the position of an instance of a highly imperfect instance of STRR1 (as you know from going through the last notes). Not only that, Dot Plot manages to find an STRR that was not previously reported. The limitations... using a rather small DNA fragment, blips are just barely visible. If you had given Dot Plot not 6867 nucleotides but 68,670 nucleotides or, worse, an entire genome, you'd never find any tandem repeat. Finally, Dot Plot relies on the ability of the human eye to pick out interesting features. This won't scale up if you want to survey an entire genome.

You are invited to build your own Dot Plot (and thereby appreciate how it works) in Problem 2 of Problem Set 6. In solving the problem, you will make use of the concept of the moving window, found also in Blast and amongst the most commonly encountered tools in genome analysis.

The algorithm used by Dot Plot would need to be modified to allow for automated detection of tandem repeats. This can be done, but now I'm going to turn to a different algorithm that also uses moving windows but is more readily automated.

III. Autocorrelation – Another general method to find local repeats

III.A. What Autocorrelation gives you

To remind you of the problem... We want to be able to detect tandem repeat sequences. MATCHES-OF-PATTERN works if there is a known conserved repeating unit or a unknown repeating unit that is exactly repeated, but that's not how biological repeats work. We need a tool that can detect tandem repeats whose repeating units are unknown and that may have multiple mistakes in them. Dot Plot works well, but it would be difficult to use it for anything but small fragments, and it would be difficult to make it independent of human pattern recognition.

As usual, a good approach to figuring out an algorithm is to ask yourself how you might do it yourself by hand. Here, however, you need to give up your exquisite human ability to recognize patterns at a glance and agree to look at sequences with the tunnel vision of a computer.

SQ11. Given the restriction that you may consider only two letters at a time, imagine how you could find tandem repeats of a 7-nt unit. Consider using two moving windows, each allowing you to see one letter at a time and each moving through the sequence in lockstep, at the same speed.

Of course, the cardinal feature of a tandem repeat is that it repeats, and if the repeating unit is of constant length, then the distance (number of nucleotides) between identical letters should be constant over the breadth of the repeat. The idea is illustrated in **Fig. 2**, below:

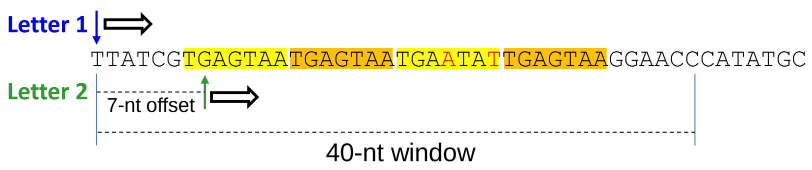


Fig. 2: Autocorrelation algorithm. Two 1-letter windows move in parallel from left to right, separated by 7 nucleotides. When the two letters are the same, 1 point is added to the total for the 40-nt window.

The idea is to count how many coincidences there are between **Letter 1** and **Letter 2** as they move through the 40-nt window.

SQ12. How many comparisons will be performed as Letter 1 and Letter 2 move through the 40-nt window?

The number of comparisons depends on how you define the window. If you define the right boundary as governing how far **Letter 1** travels, you get one answer. If you define it as governing how far **Letter 2** travels, you get another. I chose the first definition so that the window size is equal to the number of comparisons.

SQ13. What score is produced (i.e. how many coincidences occur) from the comparisons? Count if you must, but come up with an actual number.

SQ14. What would be the score if the offset is 1 rather than 7?

Clearly, the ability to detect a tandem repeat depends markedly on what offset size you use.

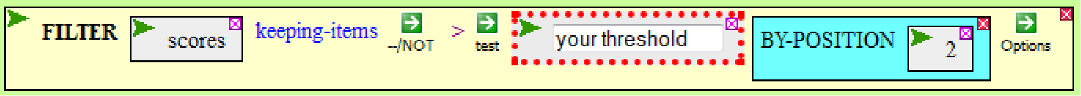
The algorithm is called autocorrelation, because you're correlating the value of letters of a string against letters of the same string, but offset a constant number of nucleotides.

Now try it on the *cpeBA* region. I've implemented an autocorrelation algorithm for your enjoyment in both CyanoBIKE and Phantome/BioBIKE. To get it, mouse over the black **Session** button and click **Workspace – List**. In your list of public workspaces, you should find a workspace called **Autocorrelate**. Click **Restore** for that workspace, and, seemingly by magic, your workspace will fill up with the functions necessary to do an autocorrelation. Here's how you run it:

- DEFINE `seq` as whatever sequence you want to analyze (it comes set up to analyze the *cpeBA* region)
- DEFINE the `from-coord` and `to-coord` to be whatever portion of the sequence you want to consider (it comes set up to consider the sequence from beginning to end)
- DEFINE the `offset` to be whatever you want (it comes set up with an offset of 7, which may more readily detect tandem repeats with repeat units of 7 nt)
- DEFINE the `window` to be whatever you want (it comes set up with a value of 100)
- DEFINE `scores` to be the result returned by the FOR-EACH loop (more on this in a moment)
- PLOT `scores`

SQ14. Execute all of these functions. How do you interpret the results (especially in light of what you already know of the positions of tandem repeats in the *cpeBA* region)?

If this is all the autocorrelation gave you, it wouldn't be much better than Dot Plot – just another figure for the human eye to interpret. But you get much more. Look at the Result pane. The FOR-EACH loop produces a list, consisting of a score for each coordinate, in the format (coordinate score). That list can be FILTERed to keep only those regions with scores higher than whatever threshold you choose to set.



Then use DISPLAY-LIST to see what high-scoring regions you found.

SQ15. Execute FILTER and DISPLAY-LIST. What are the low-coordinate and high-coordinate boundaries of the first high-scoring region? Where does it lie in your highlighted *cpeBA* region? How do you interpret this range?

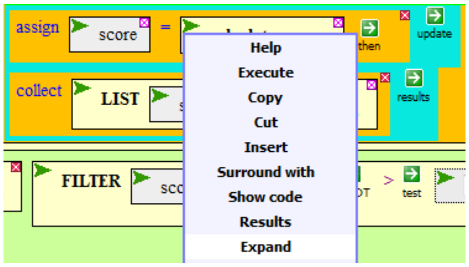
It is by no means easy to interpret these numbers without understanding how the FOR-EACH loop works. So that's where we're going next.

III.B. How does this implementation of autocorrelation work?

The only part of the code on your screen that's mysterious is the FOR-EACH loop, but its outline should be relatively clear.

- FOR-EACH start FROM from-coord TO to-coord
  You probably already understand what from-coord and to-coord mean. As the 100-nt window moves through the *cpeBA* sequence, the start of the window goes from the from-coord that you specified to the to-coord that you specified.
- WHILE something IS-NOT > LENGTH-OF seq
  The loop continues only so long as the window doesn't go beyond the end of the sequence, which you agree could cause problems. I'll leave the specifics of the calculation for another time.
- ASSIGN score = **calculate score**
  I'll get to this, the heart of the loop, in a moment.
- COLLECT (LIST start score)
  This produces the list of coordinates and associated scores that you saw earlier.

So the only thing to understand is how **calculate score** manages to calculate the score. Of course the words themselves don't do anything. They just serve as a descriptive label for the code behind them. To see that code, mouse over the symbol's action icon and click **Expand**, as shown to the right. Now you should be able to see the full code.



The variable score is calculated through another loop. Go back to **Fig. 2** to remind yourself conceptually how the score is calculated, and once you're comfortable with that, let's go through this inner loop line-by-line.

This loop is supposed to calculate the number of matches between **letter1** and **letter2**, within a given window. The most mysterious part of this loop is surely the first line. The variable `position1` (i.e., the position within the *cpeBA* sequence), begins the loop with the value of `start`, then increments by 1 until it reaches `(start + window - 1)`. If you don't see why, then replace the variables with specific values. Suppose the outer loop (`FOR-EACH start FROM from-coord TO to-coord`) has just begun, and `start = 1`. What is the value of `window`? You should know, you set it yourself. So do the calculation...

SQ16. How many iterations will this loop go through? Does that accord with your understanding of the algorithm represented in Fig. 2?

SQ17. Go through the remaining four lines of the inner loop. Do they accord with your understanding of the algorithm represented in Fig. 2?

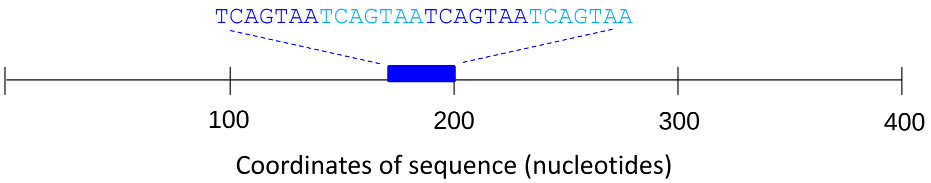
Once you've grasped what the inner loop is doing, you can pack it up again by mousing over the green action icon of the inner FOR-EACH and clicking **Collapse**.

Now you should understand what is the significance of the elements of the list collected at the end of the outer loop.

SQ18. Describe how `score` is calculated and what it means.

SQ19. Describe the meaning of `start` and why it is associated with `score` in the list.

SQ20. Suppose you are using this code to analyze the 400-nt sequence below.


TCAGTAATCAGTAATCAGTAATCAGTAA
100
200
300
400
Coordinates of sequence (nucleotides)

Presume that the sequence is completely random, except for the 28 nucleotides whose sequence is shown. You use the autocorrelation algorithm, with window=100 and offset = 7. Draw what you would expect the plot to be, taking care to make the X-axis (the coordinates) as accurate as you can.

SQ21. With this in mind, interpret the low-coordinate and high-coordinate boundaries you found in SQ15. Highlight features of interest in your highlighted *cpeBA* region.[†]

SQ22. Play with the code, changing the value of `offset` and `window` to see how this affects the plot. Conclusions?

There are still other methods to automate the identification of tandem repeats, but these methods are not as of much general interest as those described above. However, they may be of considerable interest to the Tandem Repeats/CRISPR group.

[†] The region you are considering (bounded by the low- and high-coordinate boundaries) is insanely interesting! I hope the Tandem Repeats group scrutinizes it very carefully, particularly the region where the region ends.