(26 January 2015)

BNFO301: Introduction to Bioinformatics
Computational detection of origins of replication

In our last adventure, you observed that DNA is more than genes – there are DNA sequences that have important functions besides encoding protein. Fuller et al (1984) focused on a short, 9-bp sequence (`TTAT[CA]CA[CA]A`) at the origin of replication of the *E. coli* genome and its role in binding DnaA, a protein essential for the initiation of DNA replication of that genome.

The authors went into the study already with the suspicion that the 9-bp sequence was important. It had been noted in previous articles that the sequence occurs in multiple copies at the origin of replication and in the origins of replication of other replicating DNA molecules.

How could you find this DnaA-binding site in other origins of replication? You might imagine that the task would be easy if the sequence of those origins are known and the binding sequence is the same – just search for it! But what if the binding sequence ***isn't*** the same? What if all you know is that ***some*** 9-bp sequence is repeated a few times at the origin? In fact, what if you didn't know even where the origin was? Would there be any hope?

This level of ignorance is the general state of affairs for almost all bacterial genomes. The genome must have an origin of replication (No origin, no replication. No replication, no genome). And almost all bacteria demonstrably have a DnaA protein (easy to detect because the protein's amino acid sequence is so highly conserved). You might think that in each case that the DnaA protein must bind to the genome at the origin of replication (and possibly other places as well), and you'd probably be right. You might think also that the DnaA protein must bind to the same 9-bp sequence --- but in this case you would be wrong. The overall mechanism of how DnaA works appears to be well conserved over a wide range of bacteria, but the details – like the binding sequence – varies. In cases where the details are known, DnaA appears to bind multiple times to the origin, requiring multiple binding sites -- as is the case in *E. coli* -- but those multiple binding sites do not necessarily have the same sequence as those in *E. coli*. Could you use just this information to find the binding sites and the origin of replication?

This is our task before us, as an example of how pure genome analysis can shed light on the function of cells.

I. The *E. coli* origin of replication

Before trying our hand on little studied genomes, it would be useful to have in hand a well-studied example, the origin of replication from *E. coli*. Of course, this is the very region studied by Fuller et al (1984). Figure 7 from that article gives the sequence from the origin. Take a look at it.

SQ1. Do you see the 9-bp DnaA binding sites? How many sites are there? On which strand (top or bottom) are they?

Hmmm. The quality of that figure is pretty poor. It's virtually impossible to read the highlighted sequence. We need to get something better.

Log into PhAnToMe/BioBIKE so that we can get the origin sequence from there. How to find it? Maybe the sequence is labeled as the origin, but few bacterial genomes have that degree of annotation (description of sequences).

SQ2. From the information given in Fuller et al, how could you find the region of the *E. coli* genome shown in Figure 7?

The *E. coli* genome in BioBIKE may or may not have good annotation, but one thing that it ***does*** have is the sequence. So if you could find the ***sequence*** shown in Figure 7, then you're done.

What tools does BioBIKE have that may be useful? If you mouse over the green STRINGS-SEQUENCES button and then SEARCH/COMPARE, you'll find a few candidates. Standing out are MATCHES-OF-ITEM and SEQUENCE-SIMILAR-TO. Maybe you can match the item – i.e. the sequence found in Figure 7 – with something in the E. coli genome or find something in the genome similar to it. Bring both functions into the workspace by clicking on them.

Try MATCHES-OF-ITEM first. You're asked for the *query* and the *target*. In general, *query* refers to the sequence you have in hand and *target* refers to the sequence you want to search in order to find something similar. In the *query* box, type in the first 10 to 20 nucleotides in Figure 7 and press enter.

In the *target* box,... what's the name of the organism to search? You'll remember from *What is a Gene* that organisms have particular names. Better to find the name in a list of official names rather than guess how BioBIKE might happen to call *E. coli*. Click the *target* box to open it, then mouse over the blue ORGANISMS button and click **Bacteria**. If this is your first time using the button in this session, you'll have to wait a couple of seconds for the bacteria menu to load. Then you'll be invited to try the menu again. Choose the alphabetical list of bacterial names and scroll down to Escherichia coli... whoa! Sequencers have been busy! There are many E. coli to choose from. Which one corresponds to Fuller et al's strain? In principle, you should be able to find the answer from their **Experimental Procedures** section. In fact, however, you'd have to go back through several layers of references. I'll spare you that task and tell you that the *E. coli* strain they got the origin sequence from was reported to be derived from *Escherichia coli* K-12. So go back to the menu and click that strain.[1]

Now you should be ready to roll. Execute the function (recall the two ways to do this from *What is a Gene*).

SQ3. What do you make of the result. (Recall that results will generally appear in the purple Result Pane)

Perhaps you didn't get a result. Instead you got the following error:

```
Execution error:

*** PROBLEM: I don't understand what you mean by
'GATCCTAGGTATTAAAAAGAAGATCTATTTATTTA'

*** ADVICE: Perhaps you misspelled the name of a variable?
    Or perhaps you intended it to be a string, e.g.,
"GATCCTAGGTATTAAAAAGAAGATCTATTTATTTA".
```

SQ4. Whether you got this very common error or not, what do you make of the advice?

Consider that you might have DEFINEd a variable named *GATCCTAGGTATTAAAAAGAAGATCT* (or whatever) and given it a value of 47.

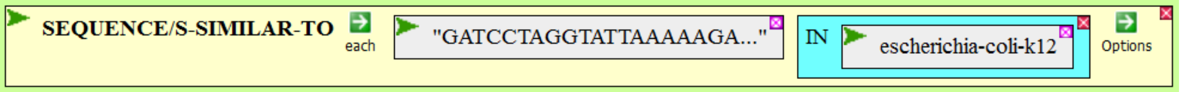
[1] If after all these manipulations, the organism lands outside the *target* box, simply drag it into that box.

SQ5. How does BioBIKE know whether you mean the *sequence* "GATCCTAGG..." or the contents of the variable *GATCCTAGG...*, i.e. 47?

If you don't know the answer to this question, then try reading *BioBIKE syntax and conventions*.

In any event, this strategy didn't work, and the reason why is not immediately clear. So try something else: SEQUENCE-SIMILAR-TO.

That function asks for a *query*, i.e. the same sequence you gave to MATCHES-OF-ITEM. But if you execute the function as it stands, BioBIKE will search every sequence it knows of. You want it to search only *E. coli*. To limit the search, mouse over the **Options** icon, select the **IN** option, and click **Apply**. Fill the *value* box with *E. coli* K12 as before,[2] and execute the function, which should look something like what you see below:



SQ6. How do you interpret the result?

"Q-..." refers to the ***q****uery* you provided, and "T-..." refers to the match that was found in the ***t****arget* genome of E. coli K12. So the match extended in the query from nucleotide 1 to nucleotide 35 and in the genome from nucleotide 3923351 to nucleotide 3923385. We'll discuss E-value in some detail later. Notice %ID, the percentage of the match that was identical. You might expect 100%, but no, it was only 97%.

SQ7. How many nucleotides in the query are not identical to the match found in the genome?

97% of what? How long (how many nucleotides) is the match that was found? What is 97% of that?

Why isn't the genome identical to the query? To check this out, bring up the *E. coli* genome sequence (you've done this before in *What is a Gene*) and go to the region of the match by typing (or copying) 3923351 into the **Go to** box and clicking **Go**.

SQ8. Does the *E. coli* K12 sequence match that found in Figure 7 of Fuller et al? Exactly? Why do you suppose not?

OK, you've gotten the sequence, but it would be nice to have it in a form you can modify, so you could highlight the DnaA-binding sites and perhaps other things. To get the sequence in a format you can copy/paste into another program, go back to SEQUENCE-OF and select a few more options: DISPLAY-ON (to specify putting the sequence in a text window), DOUBLE-STRANDED (to specify that you want both strands displayed, FROM (so you can type in the beginning coordinate), and LENGTH (so you can type in the length of the fragment you want, taken from Figure 7). (Remember to click **Apply**). Execute the function... Hmmm. Perhaps not ideal, since it has spaces every 10 nucleotides. The SEGMENT-LENGTH (as it is called) is 10 nucleotides. If you want the nucleotides to run together without spaces, then apply the

[2] Alternatively, find the nickname of the organism and just type it in. To do that, mouse over the Action Icon (the green wedge) of any box containing Escherichia-coli-K12 and click **View**. This will bring you to a window with lots of information about the organism. Find the line marked **Nickname** and there you'll find one or more nicknames by which the organism is known.

SEGMENT-LENGTH option, enter 50 (as you can see, the number of nucleotides per line), and re-execute the function.

SQ9. Copy/paste the sequence into a word processor and highlight all four 9-bp sequences highlighted in Fig. 7 of Fuller et al. Are all of them TTAT[CA]CA[CA]A? (they should be).

That, in part, is what one bacterial origin of replication looks like.

II. An algorithm to find origins of replication (per Campeau and Pevzner), Part 1

We are by no means the first to be interested in the identification of origins of replication by computational means. In fact, the first chapter of an e-book by Phillip Compeau and Pavel Pevzner is devoted to that purpose, as a means of introducing people to the notion of computational algorithms. That chapter is freely available, either as a web site[3] or a pdf file.[4] Choose one of these formats, because we're going to be going through the first part of the chapter here.

There's probably no need to do more than skim the first two pages, up until the first **Stop and Think** (*Finding oriC*, i.e. the origin of replication).[5] They ask (and so do I):

SQ10. Is the biological problem sufficiently well defined to be implemented on the computer?

The next section, up until the second **Stop and Think (***Hidden Message Problem*), proposes that there must be a hidden message in origins of replications and presents a particular origin that will serve as an example for the remainder of the chapter.

SQ11. Use PhAnToMe/BioBIKE as you did with *E. coli* to locate the origin sequence Compeau and Pevzner present. You'll find that BioBIKE knows of many *Vibrio cholera* genomes. The one the authors mean is Vibrio cholera O1 biovar eltor str N16961. There are at least two pitfalls that may block your way. Paying close attention to the details in the output of SEQUENCE-SIMILAR-TO may help you overcome them.

Keep this sequence handy, because we'll come back to it. In the mean time, the next section, up until the third **Stop and Think (**regarding multiple *k*-mers), introduces the important notion of *k-mer frequency* and how calculating it might help uncover hidden messages in sequences.

SQ12. Just to make sure you understand the concept... what do you think is the most frequent 3-mer on the page that you're reading?

SQ13. Write an English sentence that has more than one most frequent 3-mer.

In the next section we start with algorithms. Compeau and Pevzner considers an algorithm to determine the most frequent *k*-mer in a string, where *k* is a given number, and the string is some text or sequence. The first algorithm they present works but has a fatal flaw. The algorithm proceeds in two steps: (1) breaking the text into *k*-mers, and (2) counting the number of occurrences of each *k*-mer in the text.

[3] https://stepic.org/Bioinformatics-Algorithms-2/Introduction-to-DNA-Replication-1/step/1

[4] https://stepic.org/media/attachments/lessons/1/Bioinformatics%20Algorithms_8.pdf

[5] The biological context the present to justify looking for *oriC* is a bit bogus, but that's all right. We have our own justification.

SQ14. Break the sentence you are reading now into 3-mers. You'll get a list that begins ("Bre" "rea" "eak" "ak " "k t"...). How many 3-mers are in the list? What is the most frequent 3-mer?

To perform this algorithm you're going to have to compare every *k*-mer against every *k*-mer. Since the number of *k*-mers is approximately equal to the size of the text, the number of comparisons is roughly equal to that size squared:

	Match found							
target → **↓query**	"Bre"	"rea"	"eak"	"ak "	"k t"	...	"ers"	**Total**
"Bre"	1					...		1
"rea"		1				...		2
"eak"			1			...		1
"ak "				1		...		1
"k t"					1	...		1
...	...	...	...	...	...	...	...	...
"ers"						...	1	1

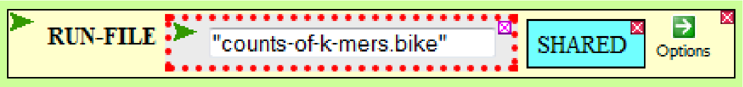
SQ15. If there are *n* letters in the sentence, then how many comparisons must be done to fill in the above table?

This algorithm is fine when you're talking about short English sentences, but when you're dealing with an entire genome, millions of nucleotides, the algorithm becomes ridiculously slow.

SQ16. Using this algorithm, how many comparisons would need to be done to count all 3-mers in the *E. coli* genome?

SQ17. Can you think of an alternative algorithm? Consider how you might do the task.

We will solve this problem in a more reasonable way and implement the algorithm in BioBIKE... but not now. For the moment, I give you a magical tool, available within PhAnToMe/BioBIKE:



RUN-FILE "counts-of-k-mers.bike" SHARED Options

If you close the white box and execute this function, a new button will appear on your palette – FUNCTIONS – and on it will be the new function COUNTS-OF-K-MERS (you'll also get a popup box with a lot of goo ending *'BBL load of "..." ended successfully.'* This is good.).

SQ18. Try out this function, asking for the most frequent 3-mers in the sequence "`CGATATATCCATAG`". Do you understand the results? Repeat execution using the BOTH-STRANDS option. Now do you understand the results?

III. Calculating probabilities and expected counts of *k*-mers

In the next section (ending with a **Stop and Think** question concerning Table 1), Compeau and Pevzner calculate the most frequent *k*-mers in the origin sequence of *Vibrio cholerae* that they presented earlier (and that you located in **SQ11**).

SQ19. Use COUNTS-OF-K-MERS to obtain the results that Compeau and Pevzner present in their Table 1 (or if you're running out of energy, at least obtain the results for 9-mers).

SQ20. In their Stop and Think question, the authors ask if any of the counts in Table 1 are surprisingly large. To get at this question (for now), consider how many counts would you expect of a specific 3-mer in a random sequence as large as the *oriC* region? Of a specific 9-mer?

(Stuck? Simplify the problem. Suppose that *oriC* is just 3 nucleotides in length and consider using the rule of multiplication for a specific 3-nucleotide sequence.)

You might presume (correctly as it turns out in this unusual case) that the frequencies of A, C, G, and T are equal, so the probability of a specific 9-mer – say ACGTTGCAT – occurring in a specific 9-nt region is:

(probability that 1st letter is A)(probability that 2nd letter is C)... (probability that 9th letter is T)

SQ19. What fraction is this joint probability?

But the origin is not a 9-nt region. In fact, it's a 540-nt region.[6]

SQ20. How many ACGTTGCAT's would you expect to find in a 540-nt piece of random DNA?

In general, the expectation is equal to the expectation for one experiment times the number of equivalent experiments. The expectation for one 9-nt region is given in SQ19, and you'd figure that the probability of finding ACGTTGCAT is the same in every 9-nt region within the 540 nucleotides. so we can add their individual probabilities, all 532 of them:

Probability of one ACGTTGCAT in 540 nt = 532 * probability of ACGTTGCAT in 9 nt

SQ21. What is this number? And why did I multiply by 532 and not 540?

SQ22. How many ACGTTGCAT's would you expect to find in a 512,000-nt piece of random DNA?

Most people don't have a lot of faith in such calculations or much of an intuition as to which results are reasonable and which are ridiculous. So let's check your calculation by actually counting. You know how to count sequences (recall *What is a Gene*). So get the COUNT-OF ACGTTGCAT in a random sequence of length 512,000 nt. The RANDOM-DNA function will be useful, availing yourself of the LENGTH option.

SQ23. Execute the COUNT-OF function a few times. Are the counts consistent with your calculation in SQ22?

SQ24. Does your result from SQ22 tell you what is the probability of finding ACGTTGCAT in a 512,000-nt piece of random DNA? If so, what is the probability? If not, why not?

[6] One way I could know this by using the LENGTH-OF function.

IV. An algorithm to find origins of replication (per Campeau and Pevzner), Part 2

We'll come back to their (surprisingly complicated) Stop and Think question at the end. For now on to the next section (ending with the **Stop and Think** question about the four most frequent 9-mers).

Compeau and Pevzner suggest a Detour for calculating probabilities. I'm not going to spend time on it, because in a parallel set of notes, I'll propose a different treatment that I think is of more general utility. I suggest you focus instead on their question for this section regarding the four candidate DnaA-binding sites. I'll reword the question in this way:

SQ25. Do you see any relationship between the four most frequent 9-mers shown in Table 1? Do any of them stand out as different?

There are two important things to think about here. One of those things (the fact that DNA is double stranded) is the focus of the next section. The other... Consider the following experiment. Suppose that I periodically run the sports pages of my newspaper through COUNTS-OF-K-MERS (don't try it... it works only on DNA sequences!). I might find that the most frequent 7-mer is "MANNING". But suppose further that I'm checking only 5-mers. Then I might find that the most frequent 5-mers are "NNING", "ANNIN" and "MANNI". What sense could I make of this? In that light, pay another visit to SQ24.

Now skim the next section, ending at the **Stop and Think** question inviting you to look again at the four most frequent 9-mers. In brief, they note that a binding site may appear on either strand of the DNA and propose that a legitimate algorithm should consider both strands.

SQ26. Suppose that you had in hand a working algorithm that found k-mers in one strand of a given piece of DNA. Treat the algorithm as a black box. What would you add to that algorithm so that it considered both strands of the given DNA?

Compeau and Pevzner suggest that you devise an algorithm to produce the reverse complement of a DNA sequence... well, you've already done this (or will soon), since it is on Problem Set 1! In any event COUNTS-OF-K-MERS offers a simpler solution. It has an option BOTH-STRANDS.

SQ27. Use COUNTS-OF-K-MERS to find the most frequent 9-mers in the Vibrio *oriC* region. How do you interpret the results?

In the next and final (for us) section, Compeau and Pevzner get at the ultimate question: Can we use this sort of analysis to predict with some confidence where an origin of replication resides on a bacterial genome? They pose a counterargument that runs something like this: I'm visiting a town that's new to me, and I want to know whether a certain Chinese restaurant in the town is good. One theory is that you can tell by whether a high percentage of the patrons are native to the culture. I hand out a survey to determine each patron's culinary heritage and discover that 40% of the patrons claim Chinese culinary heritage. Very impressive!

SQ28. Is it? What counterargument could you raise? What counterargument did Compeau and Pevzner raise?

Compeau and Pevzner now want you to devise and implement an algorithm that will find all instances in the genome of a given pattern. This is an excellent exercise, no doubt, but you already have that tool available to you in BioBIKE: MATCHES-OF-PATTERN.

SQ29. Use MATCHES-OF-PATTERN to determine all instances of ATGATCAAG in *Vibrio cholerae.* Compare your results with those of Compeau and Pevzner. If they are not the same, do you have any explanation?

SQ30. Try the same thing, using the BOTH-STRANDS option of MATCHES-OF-PATTERN.

Compeau and Pevzner now move on to another organism – and so will we, but not now. This is enough for the moment.