(23/iii/11)

Introduction to Bioinformatics
Problem Set 7: Genome Analysis

1. You are examining the sequence of a newly sequenced phage and run across the following features. In each case, you're wondering whether you've found something amazing or instead just some chance occurrence. To help decide, provide the probability that the feature might appear in an equivalent random genome. If you don't have enough information, describe what kind of information you're lacking and perhaps how you might get it.

a. You notice that upstream from a gene you're looking at there is a six-nucleotide palindrome. Never mind the specific sequence of the palindrome, what is the likelihood of finding <u>any</u> six-nucleotide palindrome there?

b. You notice a string of six consecutive purines (A or G) immediately upstream from a gene of interest. What is the likelihood that this would arise by chance?

c. You go on to the next gene, and my God, there it is again! What is the likelihood of finding this feature immediately upstream of two genes?

d. Suppose you determine that transmembrane regions occur with a frequency of one every 500 nucleotides, averaged over all genes of a phage. What is the probability of finding <u>two</u> transmembrane regions in the same gene by chance?

2. In the tour of Gomathi et al, GeneMark's ability to find genes improved considerably (if improved = matched the predictions of Gomathi et al) when you told it that the sequence came from a mycobacterium. This would be true if the phage genes had similar characteristics as mycobacterial genes. Most of the pertinent characteristics result (I believe) from biases in codon usages. You'll recall that most amino acids are encoded by multiple codons. Different organisms tend to use some of the possible codons more than others. Does phage Che12 in fact have similar codon biases as a mycobacterium?

a. What is the pattern of codon usage by a mycobacterium? Go to the Resources and Links page on the course web site and from there go to the Codon Usage Table site to find a codon table for Mycobacterium tuberculosis CDC1551 (the strain you had Genemark use during the tour of Gomathi et al). To do this, type all or part of the name of the organism into the query box, click case insensitive, and click Submit. Click the link to the desired strain and examine the codon usage table. Note the explanation of the table at the top. Each codon entry has three fields: the codon triplet, how many times it is used out of 1000 codons, and the total number of times it is used in all genes of the organism. Just to confirm you got there, what is the codon that is used the most by CDC1551, and what percent of total usage does it comprise? What fraction of all alanine codons are GCU? What fraction of all methionine codons are AUG?

b. Look over the table. What generalities can you detect in codon usage?

c. OK, enough Mycobacterium. How about the mycobacterial phage Che12? From the **Genome** menu, **Sequence-analysis** submenu, pull down CODON-FREQUENCIES-OF. Enter Che12 in the *entity* box and execute the function to get the codon frequencies of the genes of Che12.

Hmmm… difficult to know what those numbers mean. Get some help by going back to the function, specify the LABEL-FREQUENCIES option, and execute it again. That's better. Are the labeled frequencies comparable to those you see on the codon table for CDC1551?

d. Too many variables in play. It would be easier to figure out what BioBIKE is giving you if you were working with an organism for which you know the answer. That would be CDC1551! Bring down a new CODON-FREQUENCIES-OF box and find the codon frequencies of CDC1551 as you did with Che12. Are the numbers the same as you got in the codon usage table? Do you recognize the first number BioBIKE gave you, the frequency for GCT?

e. The relative frequency of a codon (compared only to other codons for a specific amino acid) is different from its absolute frequency (compared to all codons). Go back to the CODON-FREQUENCIES-OF function, select the ABSOLUTE option, and execute the function again. Now are the numbers comparable to what you got in the codon usage table?

f. Use the ABSOLUTE option for the function working on Che12. How do the frequencies compare between Che12 and CDC1551?

g. It isn't easy to make that comparison, and a quantitative measure would be more desirable than a qualitative comparison. Go to the **Genome** menu, **Sequence-analysis** submenu and bring down CODON-FREQUENCY-COMPARISON. Enter Che12 in one entity box and CDC1551 in the other, then execute the function. We'll discuss the function in more depth later. For now, I'll just say that it quantitates the comparison of codons, collecting all comparisons into a single number. But what does it mean. For now, try defining an organism taken at random from the list of organisms. Then use that organism in place of CDC1551 in the codon frequency comparison. Is Che12 closer in codon usage to CDC1551 than to random bacteria?

3. Gomathi et al looked for the attachment site of Che12 using special knowledge that the site was similar to one in another characterized phage. Suppose you didn't have this special knowledge? What would you do then? Well, by its very nature, an attachment site is a sequence of nucleotides that is found both in the genome of the phage and the genome of its host.

a. Try comparing the genome of Che12 against the genome of CDC1551 (you'll need to be in PhAnToMe/BioBIKE to do this). Bring down a SEQUENCE-SIMILAR-TO box and use Che12 as the query and CDC1551 as the target. Specify that the comparison is DNA-vs-DNA.

b. Find the actual sequences identified by SEQUENCE-SIMILAR-TO and compare them to the sequence identified by Gomathi et al.

4. Membrane proteins can sometimes be identified as such by recognizing transmembrane domains in the protein sequence. They used the CBS implementation of a transmembrane finder, and so did you, but it wasn't clear what the site was doing. Here, we're going to build our own transmembrane finder. This crude program will declare a region of the membrane to be more likely to lie within a membrane if it is composed of hydrophobic amino acids. A good candidate will be a stretch of 10-20 amino acids that are, on average, hydrophobic.

a. The prime ingredient in this algorithm is the ability to determine the hydrophobicity of a specific amino acid and the average hydrophobicity of a stretch of amino acids. The function HYDROPHOBICITY-OF will do the job. Pull it down either rom the alphabetical list or from the **Genes-proteins** menu **Translation** submenu. Then play with it, giving it single amino acids. What do the numbers mean? To correlate it with your knowledge of amino acids, use a loop or a map to produce a table of each amino acid and its hydrophobicity value. Then compare this table to the characteristics of the amino acids (e.g. from the notes "Proteins" of a few weeks ago). In a general sense, what do the numbers mean?

b. Play a bit more with HYDROPHOBICITY-OF, giving it short strings of amino acids. What do the results mean?

c. Run the entire sequence of the holin protein found by Gomathi et al through HYDROPHOBICITY-OF, then PLOT the results. Can you detect any pattern? Maybe, but the plot is very noisy. There's just too much noise looking at individual amino acids. This is the same problem we faced with dot plots. And the solution is the same: average over a moving window.

d. DEFINE a window-size of 17 (roughly the size of a transmembrane region. DEFINE a start point as 1. DEFINE an end point as the start point + some number that takes it to the end of the window. DEFINE a window as the SEQUENCE-OF the holing protein FROM `start` to `end`. Find the HYDROPHOBICITY-OF scores of this window. Find the MEAN score from all the hydrophobicity scores.

e. It seems most fair to associate the hydrophobicity of a window with the midpoint coordinate of that window. DEFINE midpoint as the average of the start and end.

f. Now make a function that takes three arguments: a sequence, a start point, and a window-size and returns a list consisting of the midpoint coordinate and the mean hydrophobicity.

g. Use this function as the heart of a loop that goes through the sequence of the holin protein and collects number pairs consisting of the midpoint coordinate and the average hydrophobicity at that point.

h. Plot the number pairs produced by the loop. How do you interpret the graph?

5. GeneMark declares a gene to exist, with a specified start and stop codon. Another program specifies another start codon or denies the existence of the gene entirely. Who's right? The machines don't care. It's up to you, the human to make the call. What can you look at that would enable you to do better than the programs? Consider the following scenario.

Suppose you are trying to identify the possible function of a protein called p-Asr1156 in the cyanobacterium *Anabaena* PCC 7120. The automated annotation is no help. It lists the function as "hypothetical".

a. Perhaps you can do better by Blasting the sequence of the protein against GenBank. You can do this in either of two ways: (1) Get the sequence of the protein in FastA format from CyanoBIKE and then proceed to the NCBI Blast site as you've done in the past and use protein-protein Blast; or (2) Use the BioBIKE SEQUENCE-SIMILAR-TO function (**Strings-Sequences** menu) with p-Asr1156 as the query and *Genbank* (DATA menu) as the target (using the IN option). Be sure to specify PROTEIN-VS-PROTEIN as an option, thereby specifying the type of Blast you want to use. Execute the function… any clues? Be sure you scroll all the way to the right to see all the information available. Are the hits you found good hits, i.e. unlikely to have arisen by chance?

b. Seems ridiculous, no? "Anti-sigma-factor antagonist" is a perfectly good annotation! Why couldn't the automated annotator see the similarity you found? Perhaps you should look more deeply into the similarity by making an alignment of p-Asr1156 and similar proteins. First define a set (call it something intelligible) of proteins that are orthologous to p-Asr1156. "Orthologous" means a protein that has evolved in another organism from a common ancestral protein. We don't have a telescope to look back in time, so an ortholog of p-Asr1156 is provisionally identified as the protein that is most similar to p-Asr1156 in another organism and that identifies p-Asr1156 as the most similar protein in *Anabaena*. BioBIKE has an ORTHOLOG-OF[*] function (**Genes-Proteins** menu). If you provide no IN option, it will return orthologs in all cyanobacteria. How many orthologs did you find?

c. Obtain an alignment of the sequences of these proteins, using the ALIGNMENT-OF function (**Strings-Sequences** menu, **Bioinformatics-Tools** submenu). Put the variable you just defined into the sequence-list box. Specify the COLORED option to get pretty output, execute the function, and click the Jalview button. Scroll through the alignment to find p-Asr1156. Do you understand why an automated annotator might have annotated the orthologs as "anti-sigma-factor antagonist" but remain unimpressed with p-Asr1156?

d. Why is p-Asr1156 so different from the rest? ***Maybe it isn't!*** Maybe the computer-generated annotation of the protein is wrong and it really begins much earlier, like the other similar proteins. What does GeneMark have to say? Display (in FASTA format) a ~2000-nt segment of the DNA sequence of the A7120.chromosome[†] that includes the gene *asr1156.* Then offer it to GeneMark to see if it can find an earlier beginning of the gene. You know how to find the coordinates of *asr1156* by displaying the SEQUENCE-OF the *Anabaena* genome and going to *asr1156*. An alternate route is to ask for the DESCRIPTION-OF *asr1156* (with the FULL option). What does GeneMark identify as the start codon of the gene that ends where *asr1156* ends?

e. DEFINE a putative protein (maybe p-Asr1156-alt) as the TRANSLATION-OF the SEQUENCE-OF A7120.chromosome[†] FROM … TO …, where the from/to values are the coordinates identified from GeneMark. Use the LABELED option, so that the sequence is associated with the name of the variable.

[*] At present, ORTHOLOG-OF does not work properly in PhAnToMe/BioBIKE, so better to use CyanoBIKE for this problem.

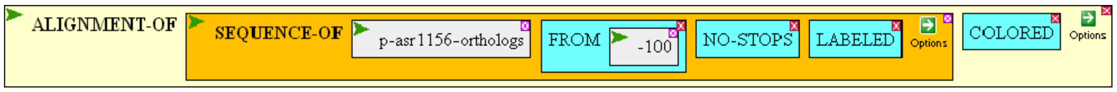
[†] Warning! Be sure to distinguish between the ***genome*** of *Anabaena* PCC 7120 (which you can access through the nickname `A7120`) and the ***chromosome*** of A7120. The genome consists of <u>all</u> DNA within the bacterium, i.e. its chromosome and any plasmids it may have. The SEQUENCE-OF A7120 FROM … TO … will get you the sequences from the chromosome <u>and</u> the plasmids, or if that isn't possible, you'll get an error message.

f. DEFINE a set of sequences consisting of the orthologs of p-Asr1156 JOINed with p-Asr1156-alt. You should get a set that has one more element than the original. Align this set…

   Wait! First, what do you expect from the alignment? The alternative protein is said to begin earlier than the official protein. If the earlier start is correct, then the initial amino acids of the alternative protein may well match the other sequences. If the later start is correct then the initial amino acids of the alternative protein are meaningless and should show no similarity to the other sequences.

   Execute the alignment… which is it? Which start assignment is correct?

g. Heh, heh… that was a trick question. Maybe neither assignment is correct! Suppose we don't rely on GeneMark or any other gene caller but instead look backwards ourselves to see where the similarity with other proteins ends. In fact, the starts of these proteins are pretty ragged in general. So we'll look backwards for all the proteins in the set of orthologs. You can do this surprisingly easily, using the ability of SEQUENCE-OF to extend even protein sequences backwards:



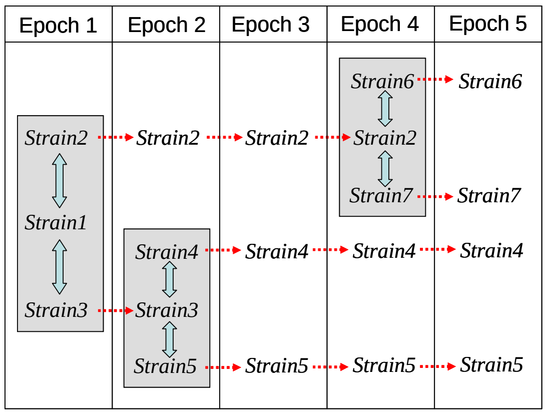
   The sequence of each protein is presumed to begin 100 amino acids before the start site known to BioBIKE. The NO-STOPS option extends the sequence backwards only until a stop codon is encountered. The LABELED option attaches the name of the protein to its extended sequence.

   Where does the region of near universal conservation of amino acid sequence begin?

h. What do you conclude about the true beginning of the protein p-Asr1156, its gene *asr1156*, and the genes of p-Asr1156 orthologs?

i. What procedures might be useful in checking the validity of gene boundaries determined by GeneMark and other automated gene callers?

6. Blast, hydrophobicity,… we've been fooling around long enough. The time has come to evolve ***Life***! What, after all, is life except replication with the possibility of change? If you agree, then here's the plan (see diagram to the right): We'll create a strain (call it *Strain1*), let it replicate and occasionally mutate, and see what we get in several epochs.



The fact is, however, we don't have the ability to watch events unfold over epochs. We can see only what's present today. So the next step will be to erase all knowledge of the course of the events we created and see if we can recreate those events solely on the basis of the final results.

a. You'll be using the MUTATE function in this simulation of life. Bring it into your workspace by using RUN-FILE (**Input-Output** menu). Type in the *file-name* box `"mutate.bike"`, select the SHARED option, and execute the function. If all is well, you should now have the MUTATE function on your **Function** button.

b. Test the MUTATE function by bringing it down from the **Function** button, entering "AAAAA" into the *sequence* box, and executing the completed function. Play with it a bit. Do you see what it does?

Epoch 1

c. DEFINE *Strain1* as a random DNA sequence 40 nucleotides long using the RANDOM-DNA function (**Strings-Sequences** menu, **String-Production** submenu). Use the LABELED option of DEFINE, associating the name of the variable with the sequence.

d. DEFINE *Strain2* as the SEQUENCE-OF *Strain1* (again LABELED), and DEFINE *Strain3* the same way. During Epoch 1, these strains are identical.

Epoch 2

e. MUTATE *Strain2*, by putting *Strain2* in the *sequence* box, and executing the completed function. Do the same with *Strain3*, but don't mutate *Strain1*. Every epoch, mutations arise in the strains, but we'll leave *Strain1* frozen in time.

f. Define *Strain4* and *Strain5* 1 as as the SEQUENCE-OF *Strain3* (again LABELED). During Epoch 2, these strains are identical.

Epoch 3

g. MUTATE *Strain2*, *Strain4*, and *Strain5*. Leave *Strain3* frozen in time.

Epoch 4

h. MUTATE *Strain2*, *Strain4*, and *Strain5*.

i. Define *Strain6* and *Strain7* 1 as as the SEQUENCE-OF *Strain2* (again LABELED). During Epoch 4, these strains are identical.

Epoch 5

j. MUTATE *Strain4*, *Strain5*, *Strain6*, and *Strain7*. Leave *Strain2* frozen in time.

Present day

k. We now have four evolved strains (*Strains 4*, *5*, *6*, and *7*) and three strains frozen in an ancient time (*Strains 1*, *2*, and *3*). Compare their sequences by aligning them by means of the ALIGNMENT-OF function (available from the **Strings-Sequences** menu, **Bioinformatic-tools** submenu). Note that this function requires a single list of sequences, so you'll need to put a LIST function (available from the **List-Tables** menu) into the *sequence-list* argument of ALIGNMENT-OF, create seven boxes within LIST (through the **More** icon), and populate them with the seven sequences you've defined. Does the alignment make sense? Can you identify the mutations?

l. Make a tree from the alignment by dragging the completed ALIGNMENT-OF function into the argument box of TREE-OF (available from the **Strings-Sequences** menu,

Phylogenetic-tree submenu). Does the tree make sense, in light of how the strains evolved?

m. Do the same thing, but this time with only the four present day strains (i.e., delete *Strains 1*, *2*, and *3* from the list to be aligned).

From trees such as these, you can analyze present day sequences, inferring the events that led to them.

7. How closely related are different phage? That's a surprisingly difficult question to answer, owing to mosaicism (as discussed in the Hatfull review article). But we can readily assess the relatedness of specific phage ***proteins***.

a. All bacteriophage should have tails. Find the tail proteins in phage Che12, using the GENES-DESCRIBED-BY function.

b. Choose the two minor tail proteins. What phage proteins are they similar to? Use each of the two proteins as a query to SEQUENCE-SIMILAR-TO and `all-phage` as the target (using the IN option). Be sure you're doing a PROTEIN-VS-PROTEIN comparison. Are the two proteins similar to each other? Which protein has the most similar phage proteins?

c. Choose the minor tail protein with the most similar phage proteins and DEFINE a variable containing those similar proteins. Do this by dragging the completed SEQUENCE-SIMILAR-TO function to the *value* box and choose the RETURN-TARGETS option of SEQUENCE-SIMILAR-TO. This causes the proteins found by Blast to be returned by the function and assigned to the variable.

d. Make a tree of those proteins, using the TREE-OF function on the ALIGNMENT-OF the variable you just created. What groupings do you discern?