Guided Tour in BioLingua

What are genes?

You may have trouble finding a matching pair of socks in the morning. Well, think about finding a gene amongst the few billion nucleotides that comprise the human genome! In this tour, we'll confront the problem of how to find a gene, which will lead us to the question: *What is a gene anyway?* To answer this question, we have to look at real sequences and real genes. Perhaps some generalities will pop out that enable us to come up with a definition.

A. Preliminaries: Genes and gene coordinates

1. Log on to BioLingua by going to http://ramsites.net/~biolingua/ and click on the VCU instance of BioLingua (public version).
2. What organisms does BioLingua know about?
   Enter `*loaded-organisms*` in the command box and find out
3. Who wants to type those long names? Find out the nicknames of all the organisms by entering: `(NICKNAMES-OF *)`
4. Let's focus on one of the smallest cyanobacteria, *Prochlorococcus marinus ss120*. A conveniently short nickname of the organism is ss120. What do you get when you simply type in this nickname?
5. "Small" is a relative term. How much DNA does ss120 have? Let's see what it's got by asking for the sequence. Enter: `(SEQUENCE-OF ss120)`
6. Perhaps that wasn't very edifying. If we can't see the whole sequence, then let's at least find out how big it is. Enter: `(LENGTH-OF ss120)`
7. That's how big it is. How many genes does it have? Display all of ss120's genes by entering: `(GENES-OF ss120)`
8. Again too many to count! Get a count of the genes by entering: `(COUNT-OF *)`. Note that the asterisk means that the function should apply to the previous result, in this case the list of genes.
9. Let's go back to that list of genes and take a look at one of them more or less at random. I chose Pro0047. Click on the blue link to Pro0047, and take a quick look at the kind of information is available.
10. Pro0047 is annotated as a DNA/RNA helicase… never mind what that is now. Apparently whoever entered this information thinks he/she knows what the gene does (or, more accurately, what kind of protein the gene encodes, since genes by themselves don't do anything). The direction of the gene is ***B*** for *backwards*… so genes can go forwards and backwards. We'll have to examine what that means later. The gene encodes a protein (***Encodes-protein*** is set to ***T***rue) – what's that all about? Don't ***all*** genes encode proteins? The gene extends ***from*** coordinate 46600 ***to*** coordinate 49788. OK, enough.
11. Look at another gene. I chose Pro0029. What do you find for this protein? Not much known about it. Of course there's not much of it to know… look at the coordinates: ***from*** 29973 ***to*** 30119. Much shorter than the other gene.

12. Or is it? Why do the math when the computer can. Ask for the length of the first gene by entering: `(LENGTH-OF pro0047)`

13. Now get the length of the second gene… actually, we could have done both at once. Let's do that: `(LENGTHS-OF (pro0047 pro0029))`. Note that when I ask for more than one thing, I put the set of things within parentheses.

14. I vote we work with the smaller gene. What does the gene look like? Everything there is to know about it is contained in its sequence, so let's look at its sequence, by entering: `(DISPLAY-SEQUENCE-OF Pro0029)`

15. Where did that sequence come from? The gene frame said the gene goes from coordinates 29973 to 30119. Those are coordinates of the chromosome of ss120. Let's see if that's true. If it is, we should be able to get the same sequence by entering:
    ```
    (DISPLAY-SEQUENCE-OF (SEQUENCE-OF ss120 FROM 29973 TO 30119))
    ```

PROBLEM 1:

If you understand how coordinates work, then you should be able to go to Pro0001, get its coordinates, and find its sequence in the chromosome. Display the beginning of the chromosome by entering: `(DISPLAY-SEQUENCE-OF (SEQUENCE-OF ss120 FROM 1 TO 500))`. Then using the coordinates you found for Pro0001, find the sequence of the beginning of the gene. To check if you're right, display the sequence of the gene and compare the two sequences.

B. What is the beginning of a gene?

Maybe ***you*** can find the beginning of a gene in a chromosomal sequence, but cells don't have the advantage of a nice coordinate list. How do ***they*** do it? How do you find the beginning of things in a sequence? How did you know where to begin this sentence? How did you know that the beginning of the sentence ***was*** the beginning of the sentence? If you examine your internal processes, perhaps you'll come up with two types of strategies:

a. Look for an internal cue, i.e. a capital letter. Words with capital letters are candidates to begin sentences.

b. Look for an external cue, i.e. punctuation. Words following periods or question marks are candidates to begin sentences.

Perhaps genes have internal or external cues. Let's look.

1. You can stare all you want at a single gene, but unless you have prior knowledge (like what the equivalent of a capital letter looks like in DNA sequences), it won't do any good. What we need to do is to look at ***many*** genes and see if we find any general features. So let's grab the first, say 10 nucleotides of every gene ss120 has and examine the collection to see if anything pops out. First, let's see how to do this with one gene, then we'll generalize. Enter: `(SEQUENCE-OF pro0029 FROM 1 TO 10)`. Does that give you the first 10 nucleotides of the sequence you displayed earlier?

2. OK, if it can work with one gene, it can work with all genes. Replace `pro0029` with `(GENES-OF ss120)` and run the command again. You might be offended by using `SEQUENCE-OF` for multiple genes… it's just not good English. BioLingua doesn't care, but if you like, you can use `SEQUENCES-OF` instead.

3. That's more like it! Can you pick out any generalities as to what kind of sequence begins genes? Just as English has more than one capital letter, genes may have more than one initiating element.

4. BioLingua saves us from generating an overwhelming amount of output. It puts only the first 100 elements of a list on the screen. Note the "…" at the end of the list. How many elements were ***not*** listed? (If you don't know the answer, what fact would you need to know to figure out the answer?)

5. You've no doubt found a pattern that holds for only the first three nucleotides. Let's focus on them. Revise the command you issued in B.1 and generate a list of the first ***three*** nucleotides of each gene in ss120.

6. How many times do your putative initiating elements appear in this list? You can find out by entering: `(COUNT-OF "ATG" IN (RESULT n))`, where n is the number of the result that gave you the list of triplets. Repeat this with all the triplets you think may be initiating elements.

7. Add up all the counts and compare them to the total number of genes. Conclusion?

8. Evidently, there are many exceptions to the rule. The greatest insights are often gained by investigating exceptions. Can you identify any gene of ss120 of the first 100 displayed that is one of the exceptions? Presume that the list of triplets goes in the same order as the list of genes. Click on the gene to learn something about it.

9. If you found the gene, then take a look at its annotation: It says the gene encodes an RNA! What could that mean? Don't ***all*** genes make RNA, which is then translated into protein? Another thing, the field ***Encodes-protein*** does not have the value of ***T***rue! (NIL is BioLingua speak for ***false***). Are there genes that ***do not*** encode protein? It's time to call in reinforcements. Go to Google and type in the annotation of the gene (just copy and paste the whole thing). The first site listed should be *SRPDB Welcome*. Go there and take a look at the overview (click on *About SRP*). Learn something about the SRP cycle – what is it about? What is SRP made of?

10. OK, back to business. Evidently the *ffs* gene does ***not*** encode a protein, and it does ***not*** begin with one of the usual triplets. Is that true of other genes that don't encode proteins… wait a second. ***Are*** there other genes that don't encode proteins, and if so, how many? We need to find out more about ss120. Enter: `ss120` and click on the link that appears.

11. Scrolling down you'll see a (partial) list of genes ss120 contains, and further down, sure enough, a list of noncoding genes. How many are there?

12. What triplet(s) do they begin with? We got the first three nucleotides of ***all*** genes of ss120. Can we get the opening triplet of a subset of those genes? Later we'll learn how to extract information from these screens (called frames). For now take advantage of a function that was made just up to get all noncoding genes. Go back to the web listener and enter: `(NONCODING-GENES-OF ss120)`.

13. *Evaluation failed...* Well, what did you expect. I ***said*** that I just made up the function. It's not part of BioLingua. One beauty of BioLingua is that it is extensible. That means that users can make up new parts of the language and distribute them to others. I've done my part. I've made up a useful function and made it sharable. Now you have to do your part to avail

yourself of the shared function. Do this by entering:

```
  (LOAD-SHARED-FILE "noncoding-genes-of.lisp")
```

14. If you spelled everything correctly, the listener should come back with a message telling you that the file loaded correctly. ***Now*** you can try: `(NONCODING-GENES-OF ss120)`

PROBLEM 2:

Find the first three nucleotides of every gene in ss120 that does not encode a protein. What do you conclude about what you've found thus far?

C. What is the end of a gene?

1. Flush with success at learning how a gene begins, you must be raring to figure out how genes end as well. First, you need to learn how to find the end of ***one*** gene, then you can generalize to all genes as you did in Section A. Think about how you found the first nucleotides of a gene `(SEQUENCE-OF pro0029 FROM 1 TO 10)`. Could you modify this statement so that it gives you the end instead?
2. Unfortunately there's no obvious way. You know the relative coordinate of the beginning of the gene: the first nucleotide is at position 1 of course. But what's the position of the end of the gene? One approach is to find out what the position is. Recall that you learned the length of the gene in step A.13. Using that information, display the last 10 nucleotides of pro0029.
3. Can you generalize this to give the last nucleotides of ***all*** genes of ss120?
4. If you did, congratulations! In the current version of BioLingua, this is not an easy task. There is, however, a more straightforward workaround. Here's how you get the entire sequence of pro0029 but display only the last 10 nucleotides:

```
  (RIGHT (SEQUENCE-OF pro0029) 10)
```

5. Now modify that command so that it works not only on pro0029 but on all the genes of ss120.
6. Use the tricks you learned in **Section B** to try to figure out what marks the end of the gene. As before, quantitate how many genes have the suspected terminating elements.
7. What about noncoding genes? How do they end?
8. What conclusions do you reach regarding how genes end?

PROBLEM 3:

You may be pleased with the elements you found, but perhaps they are red herrings! After all, capital letters don't ALWAYS indicate the beginning of a sentence. Do these elements occur elsewhere in genes? If so, then, just as in English, internal cues must be supplemented by external cues in order to determine the beginning. Use the following syntax:

```
  (PATTERN-MATCH-ALL "ATGCGGGATG" (SEQUENCE-OF pro0029))
```

to find how many times ATGCGGATG appears in the sequence of pro0029. Modify that statement to find how many times the initiating triplets appear in this (and perhaps other) genes.[1]

[1] Warning: PATTERN-MATCH functions at present work on only one sequence at a time. You can't generalize to all genes as you could with other functions.

FINAL COMMENTS

The analysis you performed points to certain elements determining the beginning and end of some genes. The last problem raised the issue that they may not be ***sufficient*** signals. If ***internal*** cues aren't sufficient, how would you go about looking for ***external*** cues? That will be the subject for a future tour.

Identifying a key element is an important first step, but mere identification is not a satisfying ending. ***Why*** do they occur where they do? How do these elements work? Faced with such questions, you might want to learn more about what is known about the process of how protein translation begins and ends and fit that into a framework that includes what you discovered here. This is a typical discovery cycle: gather information, find some surprising generality, hit the literature to find out some rationalization.