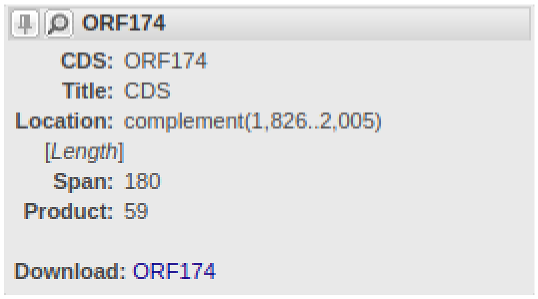
Key: __

BIO/CMPSC 300
Chapter 9 Genome Annotation I
Fall 2017

1. Download the *Enterococcus faecium* resistance plasmid sequence file from the shared repository.

2. Use the ORF Finder tool at NCBI (https://www.ncbi.nlm.nih.gov/orffinder/) to annotate the plasmid.

3. **Briefly** (3 sentences) describe what the ORF Finder tool does.
The ORF Finder tool takes a sequence file and searches all six reading frames for the stretches of in-frame sequence beginning with ATG and ending with one of the three stop codons.
4. What does changing the "Minimal ORF length" option do?
Changing 100 to 300 increases the minimum length required for a stretch of sequence to be considered an ORF.
5. When we click on the ORF174 (for example) we get the box that is displayed in Figure 1. Describe is the CDS, Title Location and Product fields contain?
The CDS: coding sequence, title: Coding sequence title,, location: where in sequence this the DNA of cds is found, and product: gene product name.


ORF174
CDS: ORF174
Title: CDS
Location: complement(1,826..2,005)
[Length]
Span: 180
Product: 59

Download: ORF174

Figure 1: After alicking on the ORF174 annotation, we see this box open.

6. Below is a generic pattern matching algorithm. Explain specifically how each step of the algorithm is being used by the NCBI ORF Finder to find ORFs in the *Enterococcus faecium* resistance plasmid sequence file.

Pattern-Matching Algorithm

a) Initialize parameters of algorithm:
`pattern` = search pattern – ATG
`searchedText` = text that will be searched for pattern – plasmid sequence
`start` = start location of search (assumes first character is position 1) position 1
`stop` = stop location of search (this represents last location to search

from) position three nucleotides from the end (no point in looking at last two)
`increment` = incrementing value (negative number for upstream search, positive number for downstream search) – 1 (in order to search all three possible reading frames)
`threshold` = minimum percentage match required - 100%

b) Compare `pattern` to characters of `searchedText` starting at position `start`. If percentage of matching characters is `>=threshold`, output `start` position and end algorithm. If not, add `increment` to `start` and continue to step 3. In your own words, what is this step doing?
Look for ATG in plasmid sequence beginning at base 1. If you find an ATG (perfect match), output "1" as the start position and end algorithm. If not, move on to position 2 and continue to step 3.

c) If `increment` is positive and `start` is `<=stop`, repeat step 2. If not, pattern was not found, end algorithm. In your own words, what is this step doing?
If your progressing left to right and your start position is less than or equal to your stop position, search for ATG again by repeating step two. If statement's not true, start codon wasn't found, end algorithm.

5. Once a start codon is found, how could you modify the algorithm above to find an open reading frame beginning with an identified start codon and ending with a stop codon? Hint: the modification involves changing just two parameters.
Change the pattern to the stop codons – TAA, TAG, or TGA
Change the increment value to three to stay in frame

6. The algorithm above would find an ATG start codon in one of three reading frames by reading a sequence in the 5' to 3' direction, but really we should consider all *six* possible reading frames: three from the DNA as it was entered and three more on the complementary strand. What changes need to be made to the algorithm above to search for ORFs in the complementary strand?

One solution would be to run the algorithm once on the DNA strand as entered, then get the reverse complement of the DNA and run the algorithm again. Or, you could reverse-complement the pattern (so CAT rather than ATG) and do another search with the same DNA string still starting at position 1 and incrementing 1. The former method is probably better in the long run, as the user will be more comfortable with sequences that look like mRNA as output.

Example of all six reading frames:

5' -TGTCATAGGATAAGCACC -3'

```
1. TGT CAT AGG ATA AGC ACC
2.   GTC ATA GGA TAA GCA
3.      TCA TAG GAT AAG CAC
```

4. GGT GCT TAT CCT ATG ACA
5. GTG CTT ATC CTA TGA
6. TGC TTA TCC TAT GAC