Zool 710: HW5: Hunting for coding regions in DNA

Marguerite A. Butler

DUE: Friday, February 23th by 5pm, please turn in to GitHub

We will build upon our Matching and Function skills. We want to write a script that will find all potential coding regions in a given DNA sequence. To make sure it will work on any sequence, we will use randomly generated DNA.

Background

Coding regions begin with a start codon and end with a stop codon. Furthermore, since DNA is read in "triplets," in order to define a coding region the start codon and the stop codon must be in the same frame to be an open reading frame. For example, there are three related sequences below that are written by triplets in frame 1, frame 2, and frame 3. All of the sequences contain a start codon `ATG` in frame 1, and stop codons `TAA` in frames 1, 2, and 3, respectively. You can see that when the dna is read in triplets, only the first has an open reading frame because start and stop are in the same frame:

1. Frame 1: `...A ATG TCT AAA ATG GGT TAA GCC...`
2. Frame 2: `...AA TGT CTA AAA TGG GTT TAA CC...`
3. Frame 3: `...AAT GTC TAA AAT GGG TTA TAA C...`

In order to find a potential gene, we want to find the longest open reading frame. To do this, we need to find all open reading frames in the six frames (3 forward and 3 reverse).

The Data

Make up random DNA sequence data. Make a vector of 500 base pairs sampled at random. In this DNA "data" we want to find all possible coding regions. We are going to assume that there are no introns.

The Problem

You recall from Molecular Biology class that coding regions are marked with a start and a stop codon.

The start codon is `ATG`. There are three stop codons: "amber" (`TAG`), "ochre" (`TAA`), and "opal" or "umber" (`TGA`).

The major steps are outlined below. We did the starred items in class. Just as we did in class, break each steps down further into concrete (and simple) programming steps. We?ll work on it some more on Tuesday, in the meantime I would encourage you to keep brainstorming with your partner (wasn?t that fun?) and try to tackle some more of the steps to get as far along as possible and come to class with your questions. Go for it!

Each person should write their own code. Please do not search for code online to steal or use `grep`. Remember, the point is to go through the process of developing your own solutions from simple building blocks given the toolkit of skills that you already have.

1. *Make up a random DNA sequence using lower case letters for the base pairs.
2. Save your random sequence in a file called `yourname_dna.csv`
3. *How do you find the start codon? Brainstorm.
4. After figuring out the coding mechanics, turn this into a function. Take what we did in lab and make a robust function out of it. Can you use this function to find the stop codons too?Brainstorm.
5. Start and stop codons must be in the same reading frame. How do you figure out which start and stop codons are in the same frame?
6. Find all start and stop codons – manually check that they are actually correct.
7. Now repeat in the reverse direction. Find all start and stop codons in the reverse direction. Think about an easy way to do this with the machinery you already have. Brainstorm.
8. For each of the six frames (3 forward and 3 reverse), which start and stop codons are in the same frame?
9. Find all of the possible open reading frames. Write output to a text file using the `cat()` function which prints the following information for each open reading frame on separate lines: orf number, start position, stop position, and the sequence (`orf1 start:xx stop:xx sequence:atgaggtc.....taa`), and tell us which one is the longest ORF. Call this file `yourname_orf.txt`. *Make sure you include the complete stop codon (all 3 base pairs) in the output.*
10. Also print to screen: the longest ORF (the sequence), which frame it's in, and start and stop position along the original DNA sequence.
11. Save your script as `yourname_find_gene.R`. Make sure you clear your workspace, test it the code, clean it up and comment it before turning it in. Turn in your script, `yourname_dna.csv`, and `yourname_orf.txt`.