Build Your Own Ribosome

Lourens-Jan Ugen & Peter Kroon

April 9, 2021

1 Ribosome

1.1 Introduction

From wikipedia:

The ribosome is a complex molecular machine found within all living cells, that serves as the site of biological protein synthesis (translation). Ribosomes link amino acids together in the order specified by messenger RNA (mRNA) molecules. Ribosomes consist of two major components: the small ribosomal subunit, which reads the RNA, and the large subunit, which joins amino acids to form a polypeptide chain. Each subunit is composed of one or more ribosomal RNA (rRNA) molecules and a variety of proteins. The ribosomes and associated molecules are also known as the translational apparatus.

Here you will write code that translates DNA into the proteins it encodes, and use it to find all proteins encoded in the genome of E.Coli. The file contains information about the *coding* strand. However, for the sake of completeness, you will *also* find all proteins encoded in the *template* strand (which is the reverse complement of the coding strand).

2 Exercises

In this assignment you will write a function that accepts a DNA sequence and will return the amino-acid sequence of the proteins that it encodes. The file with the genome is large, so test your code with a small part of it (included in Exercise 2a). Do not use `import` for this assignment: you won't need it.

2.1 Exercise 1: Reading a Dictionary

Write a function that can read all the tRNA–amino-acid pairs from the file (`Genetic_code.txt`) provided on Nestor. This file contains the information needed to insert the right amino-acids and stop at the right moment. To store the read tRNA and the corresponding amino-acid, use a dictionary. What should be a key, and what should be a value?

```
def read_dictionary(file_name):
```

2.2 Exercise 2a: Reading the FASTA file

Write a function that can read the file with the genomic information (`E_coli.txt`). Since the file is rather large, add an option to read only the first `n` lines. Note that the first line of the file is a description of it's contents, and you should not read it.

```
def read_DNA(file_name, lines=10):
```

2.3 Exercise 2b: Transcribing DNA

Write a function that can read DNA and return a sequence of RNA. Assume the DNA describes the coding strand.

```
def transcribe_DNA(DNA):
```

2.4 Exercise 3a: Finding a start codon

Proteins always start with a 'start' codon (`AUG`). Write a function that returns the position of the first 'AUG' in your RNA. The function should probably accept both the string to look for, as well as the string to look in, as arguments.

```
def find_start_position(start_codon, sequence):
```

2.5 Exercise 3b: Translating a protein

Now we know where a protein starts we can start to translate the RNA. Write a function that takes as input a) the dictionary with codons you made previously; b) a string containing RNA; and c) a position where to start translating. Iterate over the RNA string in steps of 3 storing all encoded amino acids untill you reach a 'stop' codon.

```
def translate(amino_acid_dict, RNA, start):
```

2.6 Exercise 3c: Translating all proteins

Of course your RNA contains more than one protein. Improve the function you wrote for 3a to return a list with the positions of *all* start codons.

Also write a function that takes as input a) the dictionary with codons you made previously; b) a string containing RNA; and c) *all* positions where to start translating. It should return a list of proteins.

If you remember the DRY (Don't Repeat Yourself) principle, this is probably a very short function.

2.7 Exercise 3d: Read all the proteins

Combine the functions written in 3c: write a function that takes a RNA string, the dictionary of codons, and a start codon as arguments and returns a list of all proteins.

2.8 Exercise 4: The template strand

Let's assume there are also proteins encoded in the template strand. Create the relevant RNA (the reverse complement) and find all proteins it encodes. How many proteins do you find in total encoded in the complete genome?

2.9 Bonus: Plasmid

Of course the E.Coli DNA is a plasmid, which means the DNA molecule is cyclic. Implement this as an optional argument. How does this influence your results?

3 Epilogue

Now that you created a program that can find all possible proteins encoded in any piece of DNA you can start to answer interesting questions: how many proteins are encoded in your DNA? How does that differ between organisms? How is the length of proteins distributed? How are the proteins distributed over the DNA? What percentage of DNA encodes for proteins? How often do encoding genes overlap? ...?

A second set of questions that should be answered is how your program deviates from nature. What simplifications did you make? How do these influence the results? What would you have to do to correct for these? Is it realistic to expect proteins to be encoded in the template strand as well as in the coding strand? Is it realistic to allow for "overlapping" genes? ...?