HW 2: ORF detection in R

An open reading frame (ORF) consists of stretches of (non-stop) codons that begins with a start codon and ends in a stop codon. Prokaryotic genes consist of single and continuous ORF, while eukaryotic genes are interrupted by transcribed, but not translated, sequences called introns. Then a simple gene prediction algorithm for prokaryotes might look for a start codon followed by an ORF that is long enough to encode a typical protein (even if a long ORF by itself is not conclusive evidence for the presence of a gene).

For this homework, we ask you to write a small program in R that finds ORFs in prokaryotic sequences. This program will have, at least, the sequence (in FASTA format) and a minimum length (in number of nucleotides) as input and will return the different ORFs. For each ORF, it will specify the position, the strand (forward or backward) and the frame (1, 2 or 3).

Algorithm
Given a DNA sequence *s*, a positive integer *k*, for each reading frame decompose the sequence into triplets, and find all the **longest non-overlapping** stretches of triplets starting with a start-codon and ending with a stop codon. Repeat also for the reverse complement of the sequence. Output all ORFs longer than the prefixed threshold k.

Execute your program on the mtDNA from the mouse (accession: **NC_005089**). Note that the genetic code from mitochondria is slightly different from the standard one. In particular the one for those vertebrates has different start and stop codons, resulting in different ORFs: the codon TGA means stop in the universal code, but code for tryptophan in mtDNA; AGA and AGG code for arginine in the universal code and the stop codon in mtDNA; and **ATA** represents isoleucine in the universal code and methionine mtDNA. The start codon you have to use are **ATA** and **ATG**. The stop codons are **AGA**, **AGG**, **TAA** and **TAG**.

A separate le with your sript will be attached to the report. In your report you will explain your script and answer these questions:

1. Try k = 10; 50; 100; 300; 500. How does it affect the number of ORFs you find?
2. What fraction of the sequence represent the found ORFs for k = 200?
3. Find an ORF that corresponds to a gene, give its name and its position on the sequence.