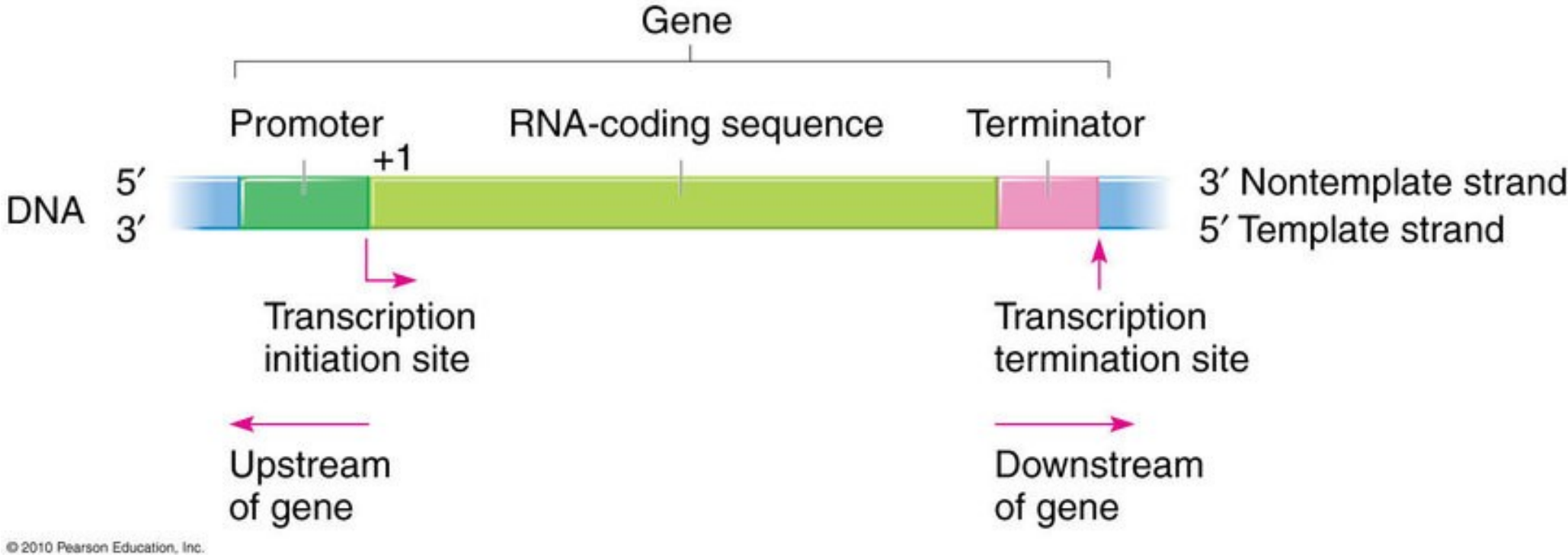
Problem definition

We are interested on identifying and removing genes in plasmids that are never expressed, as these will never be investigated in downstream analyses.

Biological definitions (*might not correspond to true biology, abstraction freedom was taken):

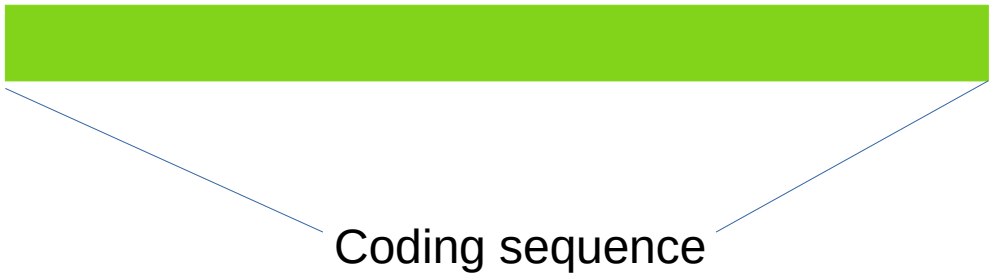
* Genes are defined as a coding sequence between a promoter and a terminator sequences;
* The promoter sequence is fixed: AGGTTGGCAGTCAGTCAGCATCTACTGTTTGCAG
* The terminator sequence is fixed: CGTCTGCTTTTGTCTCTGCTGCTGTCGTTT



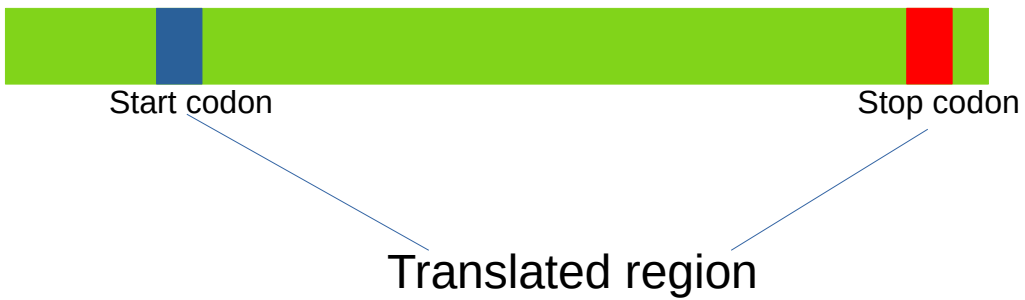
https://www.researchgate.net/figure/Structure-of-a-gene-includes-a-promoter-region-RNA-coding-region-and-terminator-sites_fig4_352678476

Biological definitions (*might not correspond to true biology, abstraction freedom was taken):

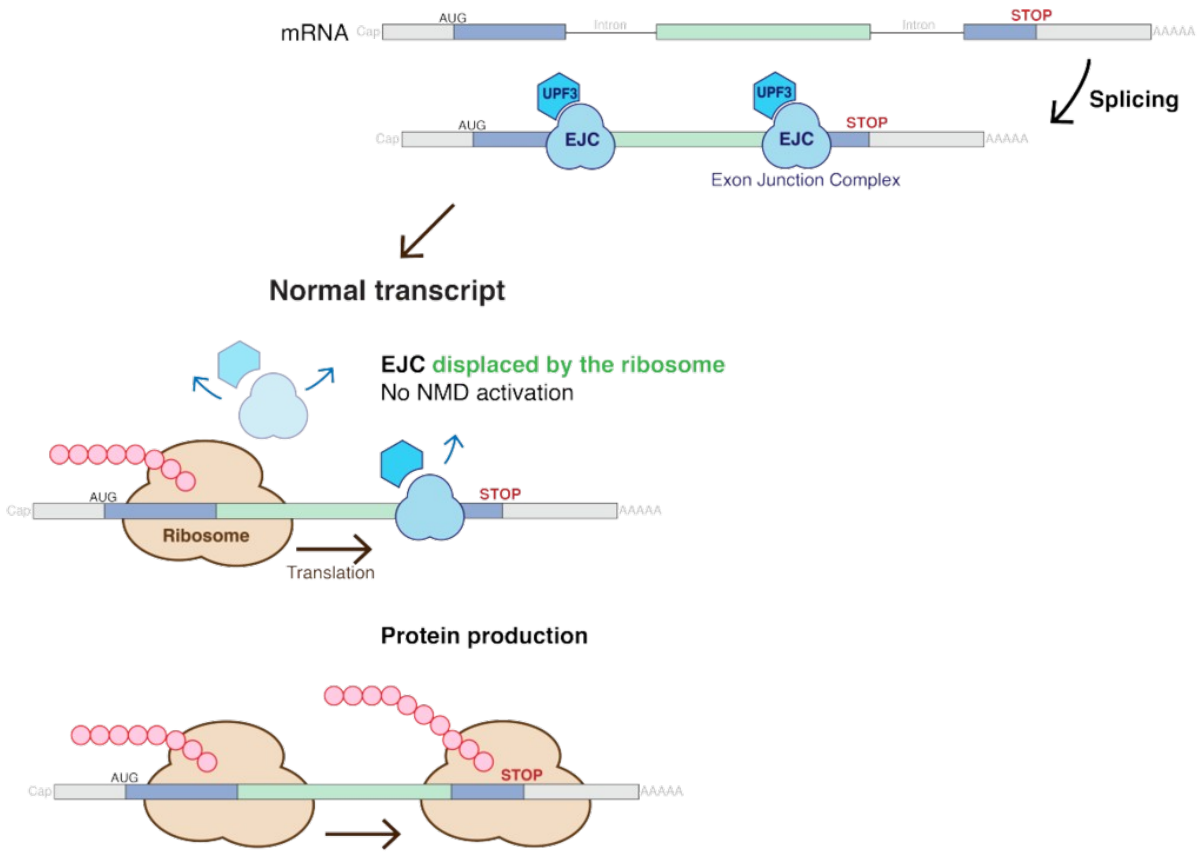
Once we have the coding sequence:



Protein translation starts at the first start codon (ATG) and ends at the last stop codon (TAA, TGA or TAG):

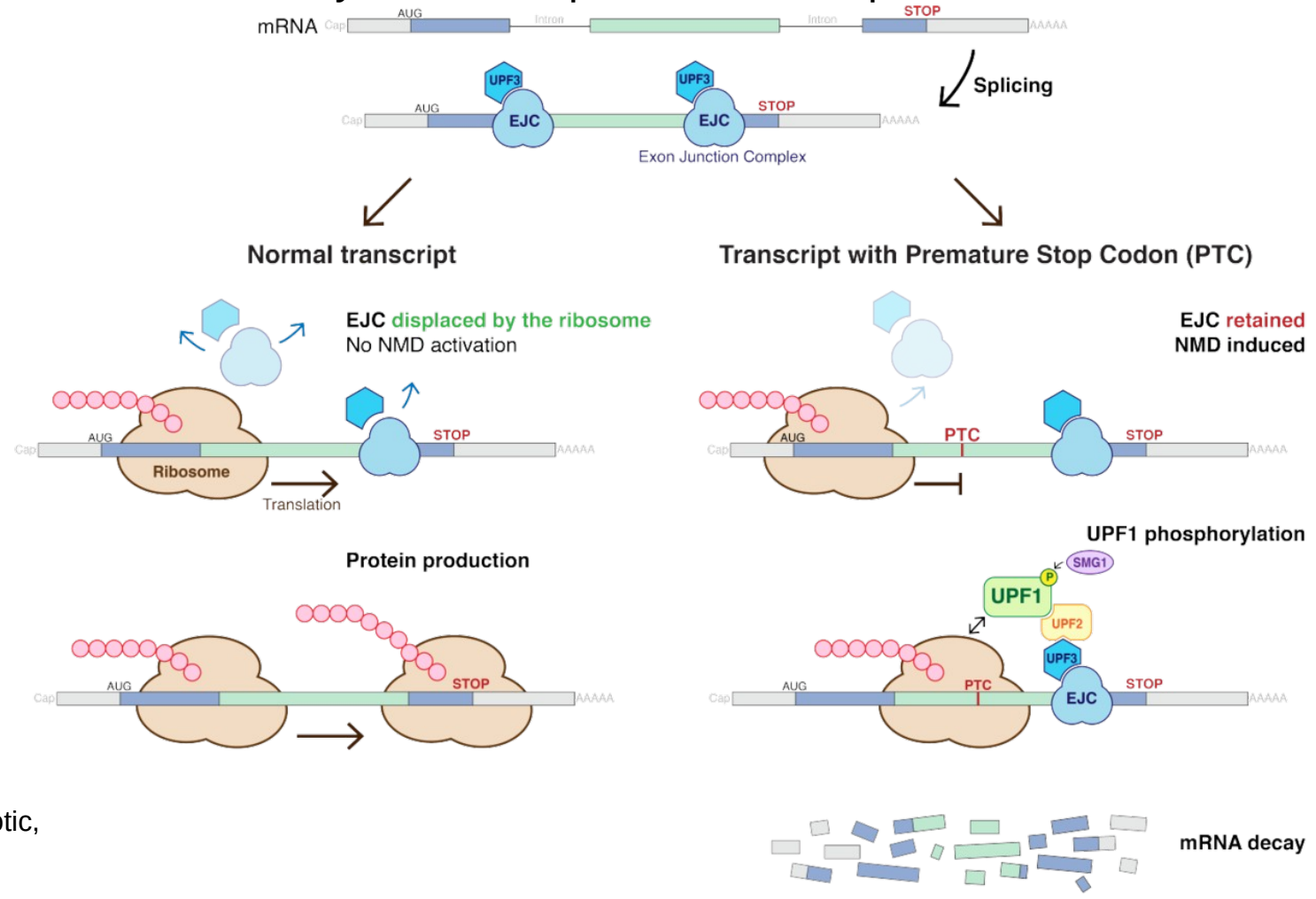


Protein translation should run fine if we have just a single stop codon at the end of the coding region:



Sorry this is eukaryotic, please bear with me

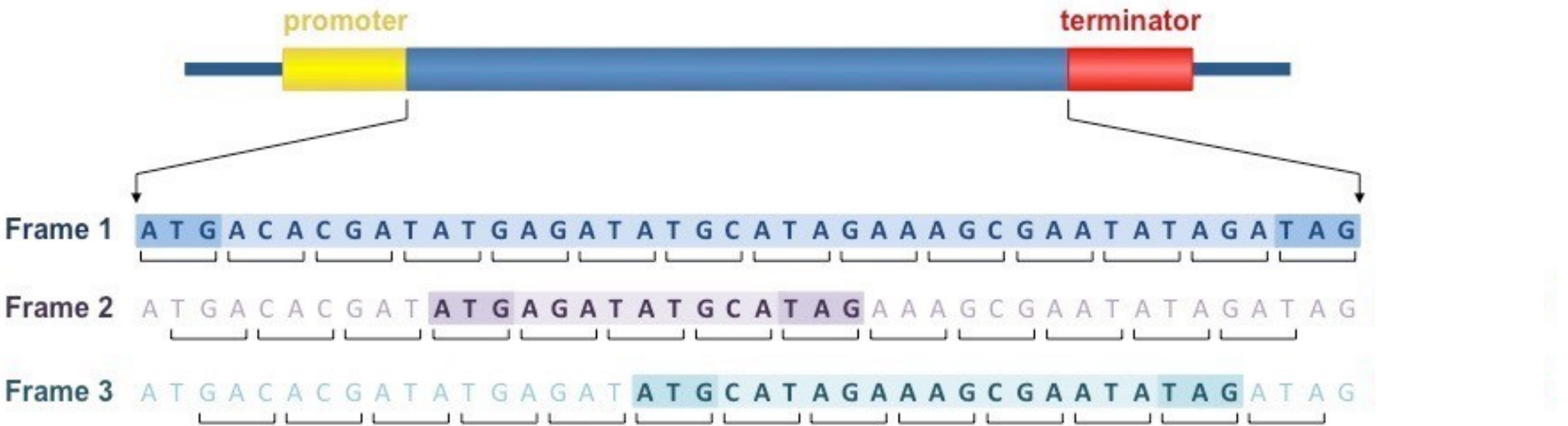
But if we have a premature stop codon (PTC), a stop codon in the middle of the coding region, then the mRNA is decayed and the protein is not expressed:



Sorry this is eukaryotic, please bear with me

However, genes are lucky as they have 6 tries to translate a protein, i.e. 6 frames of translation:

3 in the forward strand:



https://ib.bioninja.com.au/options/untitled/b2-biotechnology-in-agricul/gene-identification.html

Then we reverse complement the coding region, and try the 3 frames in the RC strand

In our little world we can say that a gene won’t be expressed for sure if its 6 frames has a premature stop codon (PTC), as all transcripts will suffer decay

Our job is to process a fasta file representing a bacterial genome, and output which genes can be expressed or will never be expressed from the plasmid sequences

* Simplification: plasmid sequences are fasta records containing the word “plasmid” or “plm” in their header;