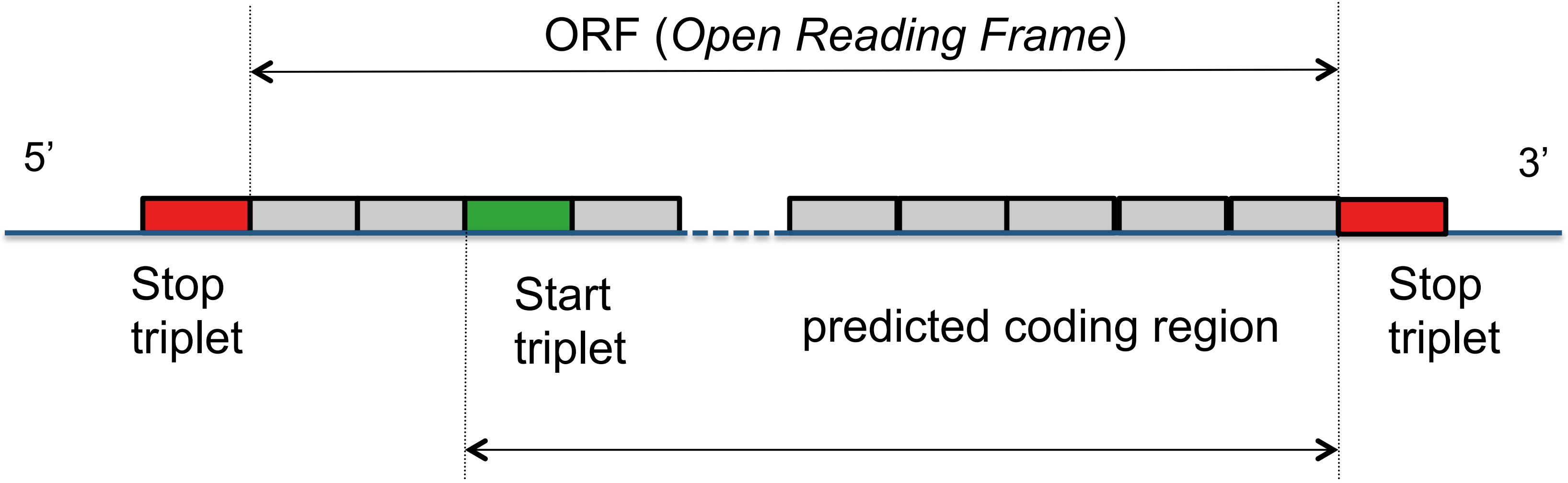
3. Gene prediction

- All genes end on a stop codon
- **A simple algorithm for gene prediction**
- Searching for start and stop codons
- Predicting all the genes in a sequence
- Making the predictions more reliable
- Boyer-Moore algorithm
- Index and suffix trees
- Probabilistic methods
- Benchmarking the prediction methods
- Gene prediction in eukaryotic genomes

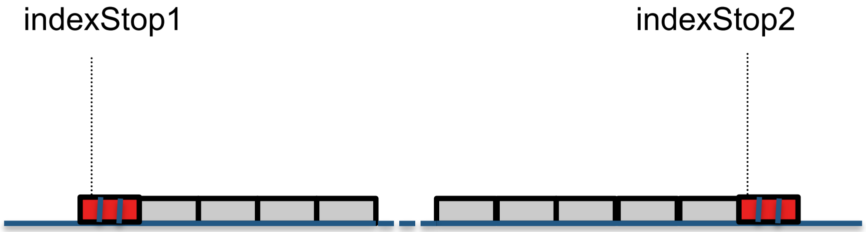
A simple algorithm for gene prediction

A simple algorithm for prokaryotic gene prediction



```
indexStop1, indexStop2, indexStart, indexDNA, iGene: integer
Gene: array [1:*, 1:2] of integer
indexDNA ← 1
iGene ← 1
repeat
  indexStop1 ← NextStopCodon (indexDNA)
  if indexStop1 > 0 then
    indexStop2 ← NextStopCodon (indexStop1+3)
    Length = IndexStop2 - IndexStop1 + 1
    if Length ≥ LengthMin then
          IndexStart <- NextStartCodon (indexStop1+3)
          if indexStart > O and indexStart < IndexStop2 then
               Gene [iGene, 1] ← indexStart
               Gene [iGene, 2] ← indexStop2
               iGene ← iGene +1
          endif
    endif
    endif
    indexDNA ← max (indexStop2, indexStop1, indexDNA+3)
  until (indexStop1 = 0) or (indexDNA > LengthSequence)
```

```
indexStop1, indexStop2, indexStart, indexDNA, iGene: integer
Gene: array [1:*, 1:2] of integer
indexDNA ← 1
iGene ← 1
repeat
  indexStop1 ← NextStopCodon (indexDNA)
  if indexStop1 > 0 then
    indexStop2 ← NextStopCodon (indexStop1+3)
    Length = IndexStop2 - IndexStop1 + 1
    if Length ≥ LengthMin then
          IndexStart <- NextStartCodon (indexStop1+3)
          if indexStart > O and indexStart < IndexStop2 then
               Gene [iGene, 1] ← indexStart
               Gene [iGene, 2] ← indexStop2
               iGene ← iGene +1
          endif
    endif
    endif
    indexDNA ← max (indexStop2, indexStop1, indexDNA+3)
  until (indexStop1 = 0) or (indexDNA > LengthSequence)
```



```
indexStop1, indexStop2, indexStart, indexDNA, iGene: integer
Gene: array [1:*, 1:2] of integer
indexDNA ← 1
iGene ← 1
repeat
  indexStop1 ← NextStopCodon (indexDNA)
  if indexStop1 > 0 then
    indexStop2 ← NextStopCodon (indexStop1+3)
    Length = IndexStop2 - IndexStop1 + 1
    if Length ≥ LengthMin then
          IndexStart <- NextStartCodon (indexStop1+3)
          if indexStart > O and indexStart < IndexStop2 then
               Gene [iGene, 1] ← indexStart
               Gene [iGene, 2] ← indexStop2
               iGene ← iGene +1
          endif
    endif
    endif
    indexDNA ← max (indexStop2, indexStop1, indexDNA+3)
  until (indexStop1 = 0) or (indexDNA > LengthSequence)
```