

2. Gènes et protéines

- La séquence est-elle un bon modèle de l'ADN ?
- Les gènes, de Mendel à la biologie moléculaire
- Le code génétique
- Un algorithme de traduction
- **Implémenter le code génétique**
- Algorithmes + structures de données = programmes
- Les compromis de la conception d'algorithmes
- Les technologies de séquençage de l'ADN
- Le séquençage de génomes complets
- Comment trouver les gènes ?

Il nous faut encore écrire la fonction lookupGeneticCode !

```
indexDNA, indexProt: integer
DNAsequence, ProteinSequence: character string [1:*]
indexProt, indexDNA ← 1
repeat
  ProtSequence [indexProt] ← lookupGeneticCode (DNAsequence [indexDNA],
  DNAsequence [indexDNA+1], DNAsequence [indexDNA+2])
  indexDNA ← indexDNA + 3
  indexProt ← indexProt + 1
until DNAsequence [indexDNA] = "#" or
DNAsequence [indexDNA+1] = "#" or
DNAsequence [indexDNA+2] = "#"
ProteinSequence [indexProt] ← "#"
```

lookupGeneticCode est une fonction que nous écrivons séparément afin de fractionner la difficulté d'expression de l'algorithme

Le code génétique dans un tableau

- Le code génétique peut être représenté par un tableau de **64 lignes** et **2 colonnes**
- L'ordre des triplets est destiné à faciliter l'utilisation du tableau
- Ici, seules les 12 premières lignes ont été représentées

TTT	F	
TTC	F	
TTA	L	
TTG	L	
TCT	S	
TCC	S	
TCA	S	
TCG	S	
TAT	Y	
TAC	Y	
TAA	*	stop
TAG	*	stop

```

Function lookupGeneticCode (char1, char2, char3: character) returns character
/* GeneticCode: array [1:64, 1:4] of character
                                     is supposed to be known inside the function */

for i from 1 to 64 do
    if GeneticCode [i, 1] = char1 then
        for j from i to 64 do
            if GeneticCode [j, 2] = char2 then
                for k from j to 64
                    if GeneticCode [k, 3] = char3 then
                        return (GeneticCode [k, 4])
                    endifor
                endifor
            endifor
        endifor
    endifor
return("?") /* le triplet est inconnu ! */
end lookupGeneticCode

```

```

indexDNA, indexProtein: integer
DNAsequence, ProteinSequence: character string
[1:*]
indexProt, indexDNA ← 1
repeat
  ProtSequence [indexProt] ← lookupGeneticCode
  (DNAsequence [indexDNA], DNAsequence [indexDNA
  +1], DNAsequence [indexDNA+2])
  indexDNA←indexDNA + 3
  indexProt ← indexProt + 1
until DNAsequence [indexDNA] = "#" or
DNAsequence [indexDNA+1] = "#" or
DNAsequence [indexDNA+2] = "#"

ProteinSequence [indexProt] ← "#"

```

```

Function lookupGeneticCode (char1, char2, char3: character) returns
character
/* GeneticCode: array [1:64, 1:4] of character
           is supposed to be known inside the function */
for i from 1 to 64 do
  if GeneticCode [i, 1] = char1 then
    for j from i to 64 do
      if GeneticCode [j, 2] = char2 then
        for k from j to 64
          if GeneticCode [k, 3] = char3
            then return (GeneticCode [k, 4])
        endfor
      endif
    endif
  endif
endfor
return("?")
end lookupGeneticCode

```