# Bioinformatique

Manipulations de séquences ADN, ARN, protéines,...

## Compter les nucléotides d'une séquence ADN

<sxh python; title : Counting_DNA_Nucleotides-01.py> #!/usr/bin/env python # -*- coding: utf-8 -*- “”“ On dispose d'un exemple de chaîne ADN (constituée des symboles 'A', 'C', 'G', 'T') Le programme utilise plusieurs techniques pour donner les nombres d'occurrences respectifs des différentes bases ”“” adn = “AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC”

# utilisation d'une liste et de la méthode .count() bases=[“A”,“C”,“G”,“T”] for base in bases:

```
  print adn.count(base),
```

print

# Variante : for c in 'ACGT':

```
  print adn.count(c),
```

print

# variante un peu moins lisible out = [] for c in 'ACGT':

```
  out.append(str(adn.count(c)))
```

print(' '.join(out))

# utilisation de la technique “list comprehension” count=[adn.count(c) for c in 'ACGT'] for val in count:

```
  print val,
```

print

# autre “list comprehension”, avec impression formatée → version “one line” print “%d %d %d %d” % tuple([adn.count(X) for X in “ACGT”])

# count “à la main”, sans utilisation de fonctions/librairie ACGT = “ACGT” count = [0,0,0,0] for c in adn:

```
  for i in range(len(ACGT)):
      if c == ACGT[i]:
          count[i] +=1
```

for val in count:

```
  print val,
```

print

# count “à la main”, avec .index() ACGT = “ACGT” count = [0,0,0,0] for c in adn:

```
  count[ACGT.index(c)] += 1
```

for val in count:

```
  print val,
```

print

# utilisation de la librairie collections from collections import defaultdict ncount = defaultdict(int) for c in adn:

```
  ncount[c] += 1
```

print ncount['A'], ncount['C'], ncount['G'], ncount['T']

# collections.Counter from collections import Counter for k,v in sorted(Counter(adn).items()): print v, print

# avec un dictionnaire freq = {'A': 0, 'C': 0, 'G': 0, 'T': 0} for c in adn:

```
  freq[c] += 1
```

print freq['A'], freq['C'], freq['G'], freq['T']

# avec un dictionnaire et count(), impression différente dico={} for base in bases:

```
  dico[base]=adn.count(base)
```

for key,val in dico.items():

```
  print "{} = {}".format(key, val)
```

</sxh>

# Références


- Using biological databases to teach evolution and biochemistry
- Rosalind, plateforme d'apprentissage de la programmation en bioinformatique
- GenBank
- Biopython
- https://en.wikipedia.org/wiki/Bioinformatics
- https://en.wikipedia.org/wiki/Open_Bioinformatics_Foundation
- https://en.wikipedia.org/wiki/FASTA_format
- https://en.wikipedia.org/wiki/List_of_open-source_bioinformatics_software
- http://www.amberbiology.com/, “Python For The Life Sciences. A gentle introduction to Python for life scientists” (à paraître)