プログラムを考えてみる

少しボトムアップに

2019-10-21　　山内長承

Codonを数えてみる

1. 配列を先頭から３塩基（＝codon）ずつに分割する
2. Codonごとに数えて、出現数の表を作る
3. Codonをアミノ酸に変換し(表を使う)、アミノ酸列を作る
4. ついでに、synonymous/non-synonymousを判定してみる

繰り返しループを使って３つずつに切る

課題０）　配列 s が与えられているとき、
先頭から１文字ずつ読んでprint するにはどうするか？
forループで繰り返す

```
s = 'ATGAAACGCATTAGCACCAACTGA'
for u in s:
    print(u)
```

（追加）用意した空のリスト c に１字ずつ追加するには？

（更に追加）上記で、Ｔの代わりにＵを書くためには？

```
if _________:
    print(____)
```

繰り返しループを使って３つずつに切る

課題１）　配列 s が与えられているとき、
先頭から３文字ずつ読んでいくにはどうするか？

３文字ずつに切って読むには
0からsの長さlen(s)までの3おきの数列 r を作る

```
r = range(0, len(s), 3)
```

```
print(len(s), list(r))  # 試しに表示してみる
```

次に rの要素ごとに sの(rからr+2文字目まで)を切出す
s = 'ATGAAACGCATTAGCACCAACTGA'
rの要素0 ⇒ s[0:3]を切出す（ATGが得られる）
rの要素1 ⇒ s[3:6]を切出す（AAAが得られる）

試しに表示してみる　print(s[u:u+3])
また、用意した空のリストcに s[u:u+3] を追加する

リストの要素の出現個数を数える

ライブラリ collections のCountを使う

例）
```
import collections
t = [1, 3, 2, 4, 1, 5, 1, 2, 3, 3, 3]
cnt = collections.Counter(t)
print(cnt)
```

では、前述のcodonのリスト c についてcodon数を数えてみる

```
s = 'ATGAAACGCATTAGCACCAACTGA'
c = []
r = range(0, len(s), 3)
for u in r:
    c.append(s[u:u+3])
cnt = collections.Counter(c)
print(cnt)
```

リストの要素の出現個数を数える

ライブラリ collections のCountを使う

おまけ）カウント結果を１つずつ取出す

カウント結果は辞書型　パターン: 回数　になっている
これを取出すには、一工夫必要になる

前頁の
cnt = collections.Counter(c)
の結果 cnt は辞書型 {'ACC': 1, 'CGC': 1,…}　なので、

```
for key in cnt.keys():
    print(cnt[key])
```

辞書cntのキーのリストcnt.keys()を先頭から(key)でなめて
キーkeyに対応する内容 cnt[key] をprintする

Codonをアミノ酸に変換する

ライブラリ Biopython の translate を使う

課題２）塩基文字列をアミノ酸列に変換する

文字列 'ACC' をアミノ酸に変換してみる

```
from Bio.Seq import Seq
seq = Seq('ACC')      # 文字列'ACC'をSeq型に変換
print(repr(seq))      # Seq型でないとtranslateできない
prot = seq.translate() # アミノ酸列に変換
print(repr(prot))
```

長い塩基配列もそのまま変換できる

```
seq = Seq('ATGAAACGCATTAGCACCAACTGA')
prot = seq.translate()
print(repr(prot))
print(prot)
```

MKRISTN*　の　最後の*は終止コドンを表す