CURSO: CC471 2019 -1

Practica Calificada 2.

1. Introducción

Al final Subirá al sitio web en la carpeta tareas un archivo CC471-PCL2-<Nombre-apellido>.zip con los archivos generados incluido un doc CC471-PCL2-<Nombre-apellido>.doc con sus respuestas .

Todos sus programas tendran una cabecera con su nombre y apellidos, el código de alumno, el curso, la fecha y nombre del archivo.
Todos sus programas deberan tener comentarios explicatorios de su código.

DESARROLLO

0. Descarge de los archivos de prueba que corresponden a esta practica Calificada (PCL2.gb)

1. (5 pts) Cree un programa en python que:

1.1 imprima su nombre y código de alumno y
1.2 imprima los identificadores de los registros existentes en PCL2.gb y sus longitudes.
1.3 Imprima los registros que tengan longitud de mas de 223 aminoacidos. Y su numero total.
Grabelo como p1.py . Redirija el resultado a un archivo de texto p1.txt
R:
Pueden haber variantes, pero un programa podria ser el siguiente:

```
from Bio import SeqIO
filename = "PCL2.gb"
for record in SeqIO.parse(filename, "genbank"):
    print("Record " + record.id + ", length " + str(len(record.seq)))

count = 0

for record in SeqIO.parse(filename, "genbank"):
    if len(record.seq)> 223:
        count = count +1
        print("Record con mas de 223")
        print(record.seq + ", length " + str(len(record.seq)))
print("Total Records con LONGITUD mas de 223 AA: " + str(count) )

for record in SeqIO.parse(filename, "genbank"):
    if len(record.seq)> 223:
        print("Record " + record.id + ", length " + str(len(record.seq)))
```

El resultado p1.txt:

```
Record AAM65232.1, length 424
Record NP_180350.1, length 424
Record XP_002880966.1, length 406
Record XP_006295904.1, length 416
Record XP_010473180.1, length 414
Record RQL84124.1, length 426
Record CDY30526.1, length 418
Record VDD13599.1, length 426
Record XP_013630868.1, length 426
Record XP_013743972.1, length 421
```

```
Record XP_013720550.1, length 426
Record XP_013720517.1, length 426
Record XP_013743973.1, length 418
Record XP_018484463.1, length 424
Record XP_009140827.1, length 418

Record con mas de 223
MRTLLPSHTPATVTTAARRRHVIHCAGKRSDSFSINSSSSDWQSSCAILSSKVNSQEQSESLSSNSNGSSSYHVSAVNGHNNGAGVSDLNLVPFNNNQSI
QSKKPLSISDLSPAPMHGSNLRVAYQGVPGAYSEAAAGKAYPNCQAIPCDQFEVAFQAVELWIADRAVLPVENSLGGSIHRNYDLLLRHRLHIVGEVQLP
VHHCLMALPGVRKEFLTRVISHPQGLAQCEHTLTKLGLNVAREAVDDTAGAAEFIAANNIRDTAAIASARAAEIYGLEILEDGIQDDASNVTRFVMLARE
PIIPRTDRPFKTSIVFAHEKGTCVLFKVLSAFAFRNISLTKIESRPNHNVPIRLVDEANVGTAKHFEYMFYIDFEASMAESRAQNALSEVQEFTSFLRVL
GSYPMDMTSWSPSSSSSSSSTFSL, length 424 …
…

Total Records con LONGITUD mas de 223 AA: 15
```

2. (5 pts) Modifique el programa anterior para imprima los identificadores y cuente los registros que terminan en “TFSL”
Grabelo como p2.py . Redirija el resultado a un archivo de texto p2.txt
R:Pueden haber variantes, pero un programa podria ser el siguiente:

```
from Bio import SeqIO
filename = "PCL2.gb"
for record in SeqIO.parse(filename, "genbank"):
    print("Record " + record.id + ", length " + str(len(record.seq)))

count = 0

for record in SeqIO.parse(filename, "genbank"):
    if (record.seq.endswith("TFSL")):
        count = count +1
        print("Record " + record.id + ", termina en TFSL  " )
print("Total de Registros que terminan en TFSL:  "  + str(count) )
```

```
RRecord AAM65232.1, termina en TFSL
Record NP_180350.1, termina en TFSL
Record XP_002880966.1, termina en TFSL
Record XP_006295904.1, termina en TFSL
Record XP_010473180.1, termina en TFSL
Record XP_018484463.1, termina en TFSL
Total de Registros que terminan en TFSL:  6
```

3. (6 pts) Cree un programa en python (p3.py) que
3.1 convierta el archvo PCL2.gb a formato FASTA (PCL2.fasta)
3.2 Genere un archivo PCLFILTRO.fasta con los registros de 3 de los identificadores de la pregunta 2.

```
from Bio import SeqIO
input_filename = "PCL2.gb"
output_filename = "PCL2.fasta"
records_iterator = SeqIO.parse(input_filename, "gb")
count = SeqIO.write(records_iterator, output_filename, "fasta")
print(str(count) + "Registros convertidos")

wanted_ids = ["AAM65232.1", "NP_180350.1", "XP_002880966.1"]
input_filename = "PCL2.fasta"
output_filename = "PCLFILTRO.fasta"
count = 0
total = 0
output_handle = open(output_filename, "w")
# ...
for record in SeqIO.parse(input_filename, "fasta"):
    if record.id in wanted_ids :
        print(record.id)
        SeqIO.write(record, output_handle, "fasta")
# ...
output_handle.close()
print(str(count) + " records selected out of " + str(total))
```

4.1 (2 pts) Busque los Id del archivo PCLFILTRO.fasta y encuentre las secuencias mRNA codificadoras (Encoding mRNA) cree el archivo de nombre mRNA.fasta.
R:
Se debe generar un archivo tipo con tres registros tipo:

```
>AY087695.1 Arabidopsis thaliana clone 37739 mRNA, complete sequence
ACAAAACATTTCCATAACCAAAAAACCTCAAAATAAAAAAATGAGAACTCTCTTACCTTCCCATACTCCG
GCAACGGTAACAACGGCGGCGAGAAGAAGACACGTCATACATTGCGCCGGTAAGAGATCTGACTCTTTCT
CCATTAATTCCAGTAGCTCCGATTGGCAAAGCTCCTGCGCCATTCTCTCAAGCAAAGTTAACTCACAAGA
ACAATCT ....
```

4.2 (2 pts) Haga un alineamiento multiple de estas secuencias y muestre el resultado.
R: Se puede utilizar clustalw para el alineamiento multiple.
El resultado es un archivo mrna.aln del tipo:

```
CLUSTAL 2.1 multiple sequence alignment


XM_013865063.2      ACAACAAAGAAAAAAAACAAATGT---GTGTATATGGTGGTAGGTGAAGGATCGTGAGAG
XM_009142579.2      ------GAAAAAAAAAACAAATGTGTTGTATATATGGTGGTAGGTGAAGGATCGTGAGAG
AY087695.1          ------------------------------------------------------------


XM_013865063.2      GCAGGAGAGCACTCCACGCTCTTCTTCTCTTCTCCTATAACAC--TCAACTTGTCTTCCT
XM_009142579.2      GCAGGAGAGCACTCCACGCTCTTCTTCTCTTCTCCTATAACACACTCAACTTGTCTTCCT
AY087695.1          ------------------------------------------------------------


XM_013865063.2      CCTTTAACAAAGCTT--GAACATCATTCTCACACTCTCAATTCAACAATGAGAACTCTTC
XM_009142579.2      CCTTTAACAAAGCTTTTGAACATCATTCTCACACTCTCAATTCAACAATGAGAACTCTTC
AY087695.1          ------ACAAAACATTTCCATAACCAAAAAAC-CTCAAAATAAAAAAATGAGAACTCTCT
                          ***** * *    * * *      ** ***  ***  ** ************

XM_013865063.2      TACCTTCTTCTCCTCACAATCCAATGCTCCGATATTCCCAAACTCCGGCGACGGTAACGG
XM_009142579.2      TACCTTCCTCTCC---------AATGCTCCGATATTCCCAATCTCCGGCGACGGTAACCC
AY087695.1          TACCTTCCC--------------ATACTCCGGCAA----------CGGTAACAACGGCGG
                    *******               ** *****  *           ***  **     *
```