

CURSO: CC471 2019

Practica 05.

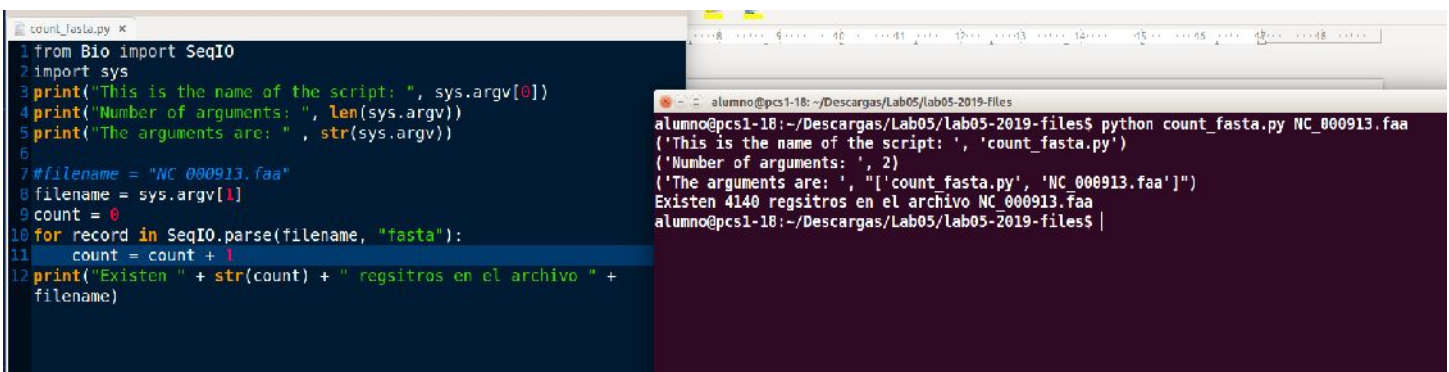
LAZARO CAMASCA EDSON

PRACTICA: Introducción a Biopython: Lectura y escritura de archivos de secuencias

Conteo de registros

Se puede contar el número de proteínas en el archivo NC_000913.faa utilizando grep: el símbolo “^>” indica búsqueda del símbolo “>” al inicio de la línea, es decir buscamos registros (proteínas) en formato Fasta.

P1. Cree el archivo count_fasta.py con el código anterior y modifíquelo utilizando sys.argv para que pueda recibir el nombre del archivo como parámetro y devuelva correctamente los resultados. p. ej utilizando *for filename in sys.argv[1:]*:



The image shows a code editor window on the left and a terminal window on the right. The code editor displays the following Python code in `count_fasta.py`:

```
1 from Bio import SeqIO
2 import sys
3 print("This is the name of the script: ", sys.argv[0])
4 print("Number of arguments: ", len(sys.argv))
5 print("The arguments are: " , str(sys.argv))
6
7 #filename = "NC_000913.faa"
8 filename = sys.argv[1]
9 count = 0
10 for record in SeqIO.parse(filename, "fasta"):
11     count = count + 1
12 print("Existen " + str(count) + " registros en el archivo " + filename)
```

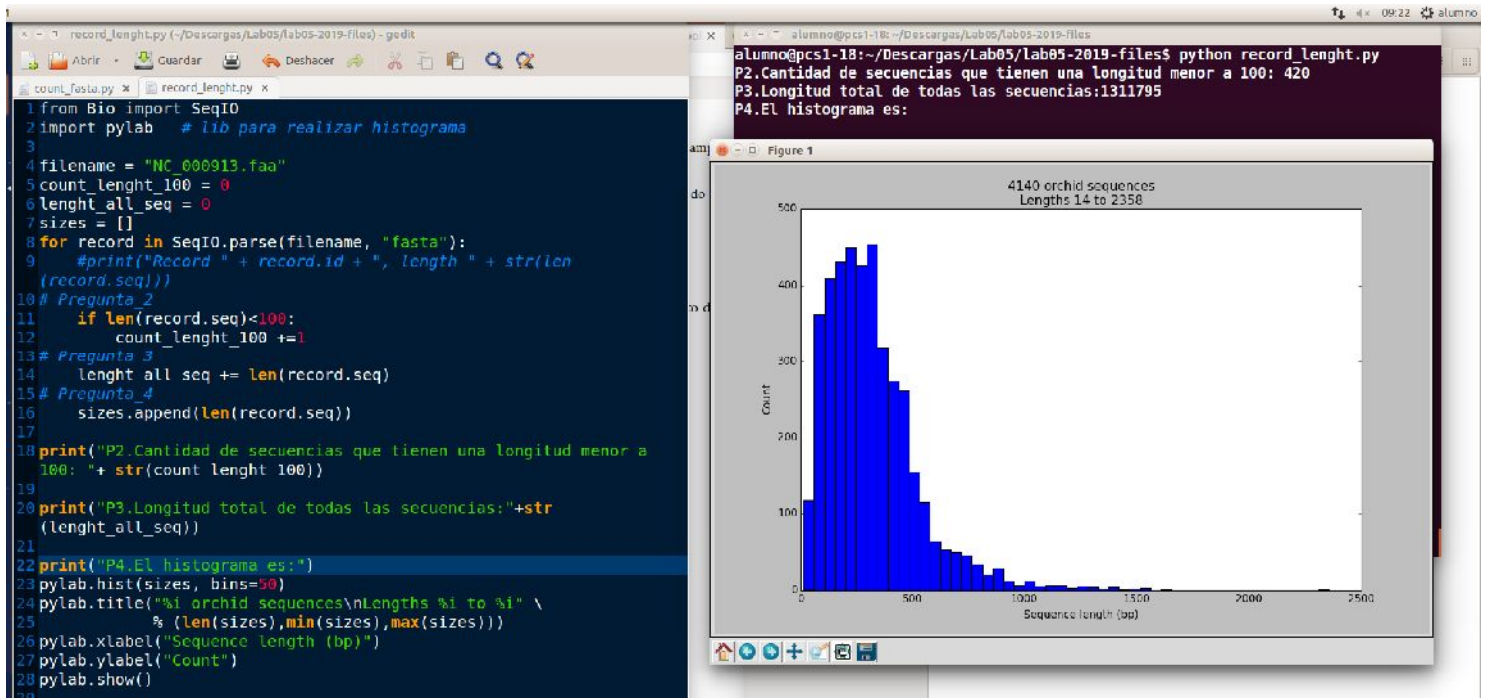
The terminal window shows the command prompt and the execution of the script:

```
alumno@pcs1-18: ~/Descargas/Lab05/lab05-2019-files$ python count_fasta.py NC_000913.faa
('This is the name of the script: ', 'count_fasta.py')
('Number of arguments: ', 2)
('The arguments are: ', "['count_fasta.py', 'NC_000913.faa']")
Existen 4140 registros en el archivo NC_000913.faa
alumno@pcs1-18: ~/Descargas/Lab05/lab05-2019-files$
```

Analizando los registros

La función `SeqIO.parse` crea objetos `SeqRecord`. Los objetos `SeqRecord` de Biopython son un contenedor que contiene la secuencia y cualquier anotación al respecto de ella, lo más importante es su identificador.

- P2. Ejercicio: Cuente cuantas secuencias tienen menos de 100 aminoácidos de longitud.
- P3. Ejercicio: Cuente cual es la longitud total de todas las secuencias.
- P4. Ejercicio. Dibuje un histograma de la distribución de longitudes. [ref. 3]



P5: ¿Cuántas de las proteínas de E. coli de este conjunto se encontraron? Un conjunto de secuencias con “cero registros que no empiezan con M” significa que ha sido correctamente anotado y estandarizado.

```

1 from Bio import SeqIO
2 filename = "PGSC_DM_v3.4_pep_representative.fasta"
3 bad = 0
4 for record in SeqIO.parse(filename, "fasta"):
5     if not record.seq.startswith("M"):
6         bad = bad + 1
7     print(record.id + " starts " + record.seq[0])
8 print("Found " + str(bad) + " records in " + filename + " which did not start with M")

```

PGSC0003DMP400052390 starts M
PGSC0003DMP400031812 starts M
PGSC0003DMP400022757 starts M
PGSC0003DMP400000212 starts M
PGSC0003DMP400014158 starts M
PGSC0003DMP400063170 starts M
PGSC0003DMP400065678 starts M
PGSC0003DMP400037881 starts M
PGSC0003DMP400039478 starts M
PGSC0003DMP40002584 starts M
PGSC0003DMP400032040 starts M
PGSC0003DMP400033247 starts M
PGSC0003DMP400060359 starts M
PGSC0003DMP400047283 starts M
PGSC0003DMP400066202 starts L
PGSC0003DMP400036214 starts N
PGSC0003DMP400026360 starts M
PGSC0003DMP400019313 starts N
PGSC0003DMP400023780 starts N
PGSC0003DMP400065882 starts N
PGSC0003DMP400011215 starts M
PGSC0003DMP400052073 starts M
PGSC0003DMP400066566 starts M
PGSC0003DMP400051773 starts N
PGSC0003DMP400081873 starts N
PGSC0003DMP400066947 starts N
PGSC0003DMP400017931 starts M
PGSC0003DMP400002517 starts N
PGSC0003DMP400035196 starts N
PGSC0003DMP400039490 starts N
PGSC0003DMP400006710 starts M
PGSC0003DMP400054332 starts N
PGSC0003DMP400024185 starts N
PGSC0003DMP400030630 starts N
PGSC0003DMP400046744 starts M
PGSC0003DMP400011481 starts Y
PGSC0003DMP400039277 starts N
PGSC0003DMP400038615 starts N
PGSC0003DMP400018233 starts M
PGSC0003DMP400001659 starts M
PGSC0003DMP400039109 starts N
PGSC0003DMP400040815 starts N
Found 268 records in PGSC_DM_v3.4_pep_representative.fasta which did not start with M
alumno@pcs1-18:~/Descargas/Lab05/Lab05-2019-files\$

P6. Modifique el programa para imprimir la descripción de los registros con problemas, no solo sus identificadores. (Lea la ayuda de Biopython acerca de SeqRecord)

```

1 from Bio import SeqIO
2 #PS
3 filename = "PGSC_DM_v3.4_pep_representative.fasta"
4 bad = 0
5 for record in SeqIO.parse(filename, "fasta"):
6     if not record.seq.startswith("M"):
7         bad = bad + 1
8         #PG
9         print(record.description + " starts " + record.seq[0])
10        #print(record.id + " starts " + record.seq[0])
11
12 print("Found " + str(bad) + " records in " + filename + " which did not
13 start with M")
14

```

```

PGSC0003DMP400058153 PGSC0003DMP400086478 Protein start_err starts K
PGSC0003DMP400034478 PGSC0003DMP400051127 Protein start_err starts L
PGSC0003DMP400051480 PGSC0003DMP400076024 Protein start_err starts E
PGSC0003DMP400046561 PGSC0003DMP400068945 Protein start_err starts I
PGSC0003DMP400057945 PGSC0003DMP400086270 Protein start_err starts G
PGSC0003DMP400041576 PGSC0003DMP400061778 Protein start_err starts L
PGSC0003DMP400011389 PGSC0003DMP400016430 Protein start_err starts V
PGSC0003DMP400031353 PGSC0003DMP400046320 Protein start_err starts V
PGSC0003DMP400063782 PGSC0003DMP400092187 Protein start_err starts T
PGSC0003DMP400067199 PGSC0003DMP400095524 Protein start_err starts K
PGSC0003DMP400034416 PGSC0003DMP400051031 Protein start_err starts L
PGSC0003DMP400001816 PGSC0003DMP400002498 Protein start_err starts I
PGSC0003DMP400058063 PGSC0003DMP400086328 Protein start_err starts P
PGSC0003DMP400028843 PGSC0003DMP400042523 Protein start_err starts Y
PGSC0003DMP400035544 PGSC0003DMP400052723 Protein start_err starts Y
PGSC0003DMP400035526 PGSC0003DMP400052687 Protein start_err starts D
PGSC0003DMP400009388 PGSC0003DMP400013567 Protein start_err starts N
PGSC0003DMP400039559 PGSC0003DMP400058729 Protein start_err starts N
PGSC0003DMP400049857 PGSC0003DMP400073675 Protein start_err starts V
PGSC0003DMP400052699 PGSC0003DMP400077823 Protein start_err starts P
PGSC0003DMP400059731 PGSC0003DMP400088056 Protein start_err starts Y
PGSC0003DMP400052384 PGSC0003DMP400077283 Protein start_err starts I
PGSC0003DMP400029875 PGSC0003DMP400044046 Protein start_err starts D
PGSC0003DMP400000059 PGSC0003DMP400000067 Protein start_err starts V
PGSC0003DMP400059214 PGSC0003DMP400087539 Protein start_err starts K
PGSC0003DMP400004373 PGSC0003DMP400006331 Protein start_err starts V
PGSC0003DMP400062289 PGSC0003DMP400090614 Protein start_err starts F
PGSC0003DMP400035539 PGSC0003DMP400052715 Protein start_err starts L
PGSC0003DMP400005929 PGSC0003DMP400008536 Protein start_err starts Y
PGSC0003DMP400014978 PGSC0003DMP400021992 Protein start_err starts E
PGSC0003DMP400065917 PGSC0003DMP400094242 Protein start_err starts S
PGSC0003DMP400058377 PGSC0003DMP400086782 Protein start_err starts Y
PGSC0003DMP400048726 PGSC0003DMP400072073 Protein start_err starts N
PGSC0003DMP400024527 PGSC0003DMP400036110 Protein start_err starts V
PGSC0003DMP400017379 PGSC0003DMP400025521 Protein start_err starts S
PGSC0003DMP400021482 PGSC0003DMP400031689 Protein start_err starts C
PGSC0003DMP400017939 PGSC0003DMP400026265 Protein start_err starts T
PGSC0003DMP400000040 PGSC0003DMP400000043 Protein start_err starts N
PGSC0003DMP400069297 PGSC0003DMP400097622 Protein start_err starts E
PGSC0003DMP400066282 PGSC0003DMP400094527 Protein start_err starts L
PGSC0003DMP400011481 PGSC0003DMP400016564 Protein start_err starts Y
Found 208 records in PGSC_DM_v3.4_pep_representative.fasta which did not start with M

```

Verificando los caracteres de Parada

P7. Escriba un programa check_stops.py para contar el número de secuencias con un “*” en ellas (en cualquier parte) Y el número de secuencias con el “*” al final. Pruebe el programa con PGSC_DM_v3.4_pep_representative.fasta.


```
temp.py x P8.py x convert_gb_to_fasta.py x check_stops.py x
1 from Bio import SeqIO
2 #P7
3 filename = "PGSC_DM_v3.4_pep_representative.fasta"
4 bad = 0
5 final_stop = 0
6 for record in SeqIO.parse(filename, "fasta"):
7     if record.seq[-1] == '*':
8         final_stop +=1
9
10    for i in record.seq:
11        if i == '*':
12            bad +=1
13    #print(record.description+" starts " + record.seq[0])
14    break
15
16 print("Secuencias que tienen un *: "+str(bad))
17 print("Secuencias que tienen un * al final: "+str(final_stop))
18
```

Terminal de IPython

Terminal 1/A x

```
In [11]: runfile('E:/Lab05/lab05-2019-files/check_stops.py', wdir='E:/Lab05/lab05-2019-files')
Secuencias que tienen un *: 39031
Secuencias que tienen un * al final: 39031
```

P8 cuál es el resultado del siguiente código?

```
temp.py x P8.py x convert_gb_to_fasta.py x
1 from Bio import SeqIO
2
3 #FORMATO FASTA
4 fasta_record = SeqIO.read("NC_000913.fna", "fasta")
5 print("Formato fasta: "+fasta_record.id + " length " + str(len(fasta_record)))
6
7 #FORMATO GBANK
8 genbank_record = SeqIO.read("NC_000913.gbk", "genbank")
9 print("formato gebank: "+genbank_record.id + " length " + str(len(genbank_record)))
```

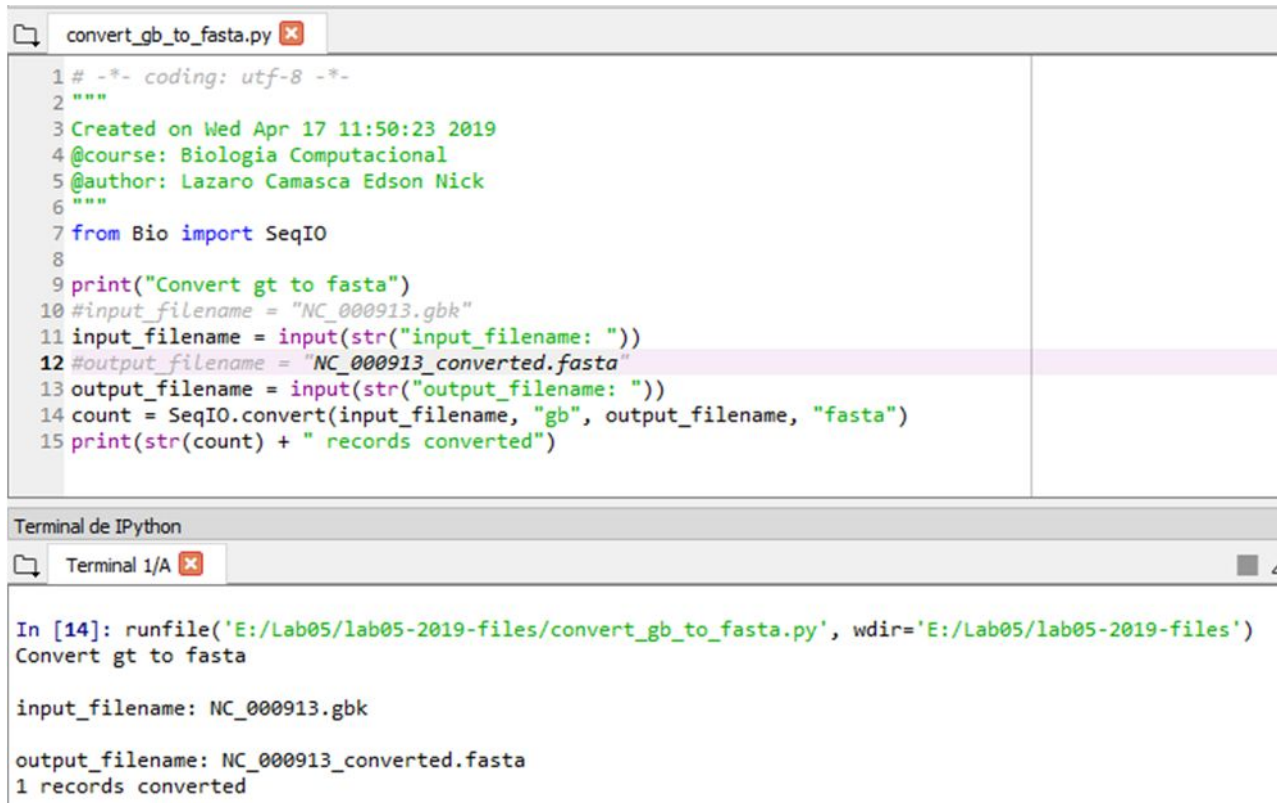
Terminal de IPython

Terminal 1/A x

```
In [7]: runfile('E:/Lab05/lab05-2019-files/P8.py', wdir='E:/Lab05/lab05-2019-files')
Formato fasta: gi|556503834|ref|NC_000913.3| length 4641652
formato gebank: NC_000913.3 length 4641652
```

Convertir un archivo de secuencias

P9: Modifique este programa para que el programa reciba el archivo de entrada y el de salida en la línea de comandos.



The image shows a Jupyter Notebook interface. The top part is a code editor with a file named `convert_gb_to_fasta.py`. The code is a Python script that converts a GenBank file to FASTA format. It includes a docstring with metadata, imports `SeqIO` from `Bio`, and uses `input()` to get the input and output filenames. The script then uses `SeqIO.convert()` to perform the conversion and prints the result.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Apr 17 11:50:23 2019
4 @course: Biología Computacional
5 @author: Lazaro Camasca Edson Nick
6 """
7 from Bio import SeqIO
8
9 print("Convert gt to fasta")
10 #input_filename = "NC_000913.gb"
11 input_filename = input(str("input_filename: "))
12 #output_filename = "NC_000913_converted.fasta"
13 output_filename = input(str("output_filename: "))
14 count = SeqIO.convert(input_filename, "gb", output_filename, "fasta")
15 print(str(count) + " records converted")
```

The bottom part of the image shows the terminal output of the script. It starts with the command `runfile('E:/Lab05/lab05-2019-files/convert_gb_to_fasta.py', wdir='E:/Lab05/lab05-2019-files')`, followed by the prompt `Convert gt to fasta`. The user enters `NC_000913.gb` for the input filename and `NC_000913_converted.fasta` for the output filename. The final output is `1 records converted`.

```
Terminal de IPython
Terminal 1/A

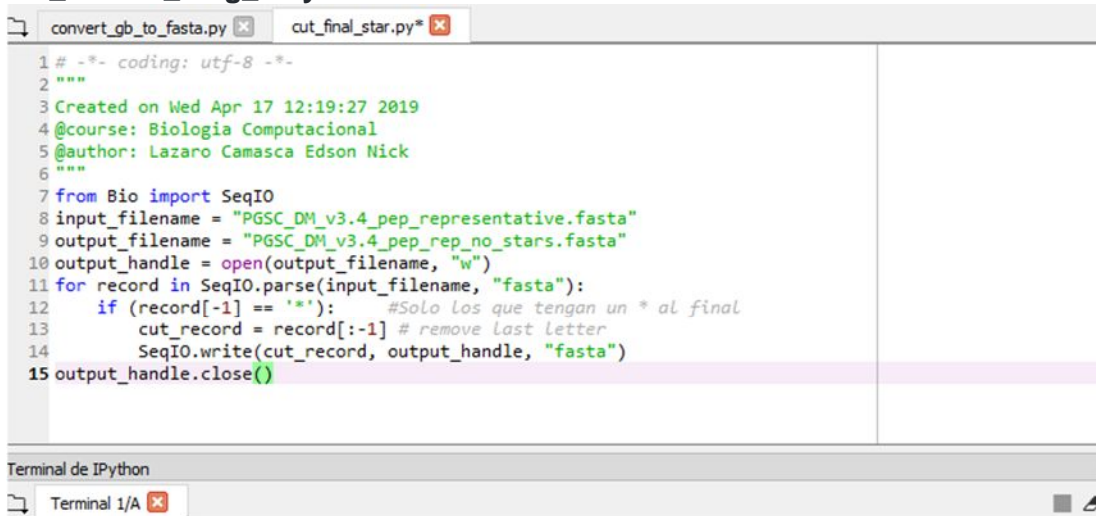
In [14]: runfile('E:/Lab05/lab05-2019-files/convert_gb_to_fasta.py', wdir='E:/Lab05/lab05-2019-files')
Convert gt to fasta

input_filename: NC_000913.gb

output_filename: NC_000913_converted.fasta
1 records converted
```

Filtrando un archivo de secuencias.

P10: ¿Cuál es el resultado de ejecutar este programa? Que contiene NC_000913_long_only.faa ?.



The screenshot shows a Jupyter Notebook with two tabs: 'convert_gb_to_fasta.py' and 'cut_final_star.py*'. The active tab contains a Python script that reads a FASTA file, filters out sequences ending with an asterisk, and writes the results to a new FASTA file. Below the script, the 'Terminal de IPython' shows the command to run the script, and the 'Terminal 1/A' shows the output of the script, which is a FASTA file containing a single protein sequence.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Apr 17 12:19:27 2019
4 @course: Biología Computacional
5 @author: Lazaro Camasca Edson Nick
6 """
7 from Bio import SeqIO
8 input_filename = "PGSC_DM_v3.4_pep_representative.fasta"
9 output_filename = "PGSC_DM_v3.4_pep_rep_no_stars.fasta"
10 output_handle = open(output_filename, "w")
11 for record in SeqIO.parse(input_filename, "fasta"):
12     if (record[-1] == '*'): #Solo Los que tengan un * al final
13         cut_record = record[:-1] # remove last letter
14         SeqIO.write(cut_record, output_handle, "fasta")
15 output_handle.close()
```

Terminal de IPython

Terminal 1/A

In [27]: runfile('E:/Lab05/lab05-2019-files/cut_final_star.py', wdir='E:/Lab05/lab05-2019-files')

In [28]: PGSC_DM_v3.4_pep_rep_no_stars.fasta: Bloc de notas

In [27]: runfile('E:/Lab05/lab05-2019-files/cut_final_star.py', wdir='E:/Lab05/lab05-2019-files')

In [28]: PGSC_DM_v3.4_pep_rep_no_stars.fasta: Bloc de notas

```
Archivo Edición Formato Ver Ayuda
LASTIILDSFPQSQAFFNHTVKPVLIPLLKFLQSTNSYFMLNVYPYDYMQSNSVIPLDY
ALFKPLAANKEAVDSNTLLHYTNVFDAMIDAAYFAMADVNFNIPVMVTESGWPSMGESN
EPDATVDNANTYNSNLKHLNKTGTPKHPIAVSTYIYELYNEDAKAGPLSEKNWGLFS
NNGTPVYILRLTESGSLFANNTSNQTYCVAKEGADTKMLQAGLDWACGTGKVNCAPLMQG
GPCYDPDNVAHAHATYAFDAYYHMMGKAPGTCDFTGATITTTNPSHGTCLFSSYFYQAT
EITHRLRLTRIKVTKYADEITRGTCRVVASPSFVN NVVYRGSQNNLQAATAAQVGGGAAS
GGAKDVTIVIRDGINGKVN TTTSVYSSSSTSYVENNKGICKKAVTLESVSVNDGIRAND
GGGGNVGV DATGGASHAKTSSGTSS
>PGSC0003DMP400020381 PGSC0003DMT400029984 Protein
MLEKDSRDDRLDCVFPKHKDSVEEVSSLSENTRTSNDCSRNNVDSISSEVYPNDPT
```

P11: ¿Cuál es el resultado de correr el siguiente comando? ¿Y qué significa?
\$ grep -c "^>" NC_000913_long_only.faa

P12: Modifique el programa para que remueva la última letra solo si es "*" pero que deje el registro sin cambio si no termina en "*" póngale nombre cut_final_star.py

```

1 """
2 Created on Wed Apr 17 12:03:56 2019
3 @course: Biología Computacional
4 @author: Lazaro Camasca Edson Nick
5 """
6
7 from Bio import SeqIO
8 input_filename = "NC_000913.faa"
9 output_filename = "NC_000913_long_only.faa"
10 count = 0
11 total = 0
12 for record in SeqIO.parse(input_filename, "fasta"):
13     total = total + 1
14     if 100 <= len(record):
15         count = count + 1
16         SeqIO.write(record, output_filename, "fasta")
17 #print("Contenido\n"+record+"\n")
18 print(str(count) + " records selected out of " + str(total))
19
20

```

```

Terminal de IPython
Terminal 1/A

In [23]: runfile('E:/Lab05/lab05-2019-files/lenght_filter_naive.py', wdir='E:/Lab05/lab05-2019-files')
3720 records selected out of 4140

```

```

NC_000913_long_only.faa: Bloc de notas
Archivo Edición Formato Ver Ayuda
>gi|16132220|ref|NP_418820.1| putative methyltransferase [Escherichia coli str. K-12 substr. MG1655]
MRITIILVAPARAENIGAAARAMKTMGFSDLRIVDSQAHLEPATRWVVAHGSGDIIDNIKV
FPTLAESLHDVDFTVATTARSRAKYHYATPVELVPLLEEKSSWMSHAALVFGREDSGLT
NEELALADVLTVGPMVADYPSLNLGQAVMVYCYQLATLIQPAKSDATADQHQLQALRER
AMTLTTTAVADDIKLVDWLQQLGLLEQRDTAMLHRLHLDIEKNITK

```

P13: Cree el programa filter_wanted_ids.py que escribe las proteínas requeridas del archivo de proteínas de la papa.

```

1 #-*- coding: utf-8 -*-
2 """
3 Created on Wed Apr 17 12:35:37 2019
4 @course: Biología Computacional
5 @author: Lazaro Camasca Edson Nick
6 """
7 from Bio import SeqIO
8 wanted_ids = ["PGSC0003DMP400019313", "PGSC0003DMP400020381", "PGSC0003DMP400020972"]
9 input_filename = "PGSC_DM_v3.4_pep_representative.fasta"
10 output_filename = "wanted_potato_proteins.fasta"
11 count = 0
12 total = 0
13 output_handle = open(output_filename, "w")
14 # ...
15 # Your code here
16 for record in SeqIO.parse(input_filename, "fasta"):
17     total += 1
18     # Encuentra la secuencias queridas

```

```

Terminal de IPython
Terminal 1/A

In [48]: runfile('E:/Lab05/lab05-2019-files/filter_wanted_ids.py', wdir='E:/Lab05/lab05-2019-files')
3 records selected out of 39031

```


P14: Modifique el programa para leer la lista de identificadores requeridos de un archivo de texto. Un Id por línea.

```
filter_wanted_ids_read_txt.py
7 from Bio import SeqIO
8 wanted_ids = []
9
10 #ingresar "wanted_ids.txt"
11 #f = open(input(str("Ingrese la direccion del fichero wanted_ids: ")))
12 f = open("wanted_ids.txt")
13 for linea in f:
14     wanted_ids.append(f.readline())
15
16 f.close()
17
18 input_filename = "PGSC_DM_v3.4_pep_representative.fasta"
19 output_filename = "wanted_potato_proteins.fasta"
20 count = 0
21 total = 0
22 output_handle = open(output_filename, "w")

Terminal de IPython
Terminal 1/A
In [57]: runfile('E:/Lab05/lab05-2019-files/filter_wanted_ids_read_txt.py', wdir='E:/Lab05/lab05-2019-files')
1 records selected out of 39031
```

P15: Complete el siguiente programa filter_wanted_id_in_order.py usando SeqIO.index(...) para hacer un archivo fasto con registros que queremos en el orden específico mostrado.

```
8 wanted_ids = ["PGSC0003DMP400019313", "PGSC0003DMP400020381", "PGSC0003DMP400020972"]
9 fileName = 'PGSC_DM_v3.4_pep_representative.fasta'
10 output = 'wanted_potato_proteins_in_order.fasta'
11 fasta_index = SeqIO.index(fileName, 'fasta')
12 total = len(fasta_index)
13 print(str(total) + " secuencias en "+fileName)
14 cont = 0
15 output_handle = open(output, 'w')
16
17 for record_id in wanted_ids:
18     cont = cont + 1
19     record = fasta_index[record_id]
20     SeqIO.write(record, output_handle, 'fasta')
21
22 output_handle.close()
23 print(str(cont)+" secuencias seleccionadas de "+str(total))
24

Terminal de IPython
Terminal 1/A
In [53]: runfile('E:/Lab05/lab05-2019-files/filter_wanted_id_in_order.py', wdir='E:/Lab05/lab05-2019-files')
39031 secuencias en PGSC_DM_v3.4_pep_representative.fasta
3 secuencias seleccionadas de 39031
```