Actividad 2

Manipulación y formateo de archivos: Formato FASTQ y FASTA

Objetivo

El propósito de esta actividad es desarrollar un flujo de trabajo bioinformático completo (denominado en inglés *pipeline*) que nos permita procesar una serie de datos biológicos. Con esta actividad, se pretende que el estudiante adquiera destrezas para interactuar con el sistema operativo a través de la línea de comandos y que sea capaz de desarrollar Shell scripts propios para resolver diferentes retos bioinformáticos focalizados en dos tipos de formato de texto, el formato FASTQ y en formato FASTA.

Detalles sobre la entrega

- La entrega se realizará utilizando este documento como plantilla; adicionando capturas de pantalla con el código empleado y la ejecución del mismo en los espacios determinados a ese uso (podéis incrementar o reducir el tamaño de los mismos).
- Será esencial adicionar en las capturas de pantalla su usuario (adicionar el *promp* completo) e intentar que la resolución de las mismas sea el máximo posible. Recordar que las actividades a realizar están resaltadas en negrita.
- La entrega se realizará a través del Campus VIU en un archivo único descomprimido en formato **PDF**.

Parte I: Creación del directorio de trabajo.

Para inicializar este ejercicio, deberán crear y organizar un nuevo directorio de trabajo que contenga los puntos más importantes que vamos a estudiar. La idea de esta organización es que alguien que no esté familiarizado con su proyecto debería poder mirar los archivos almacenados en el ordenador y comprender en detalle lo que hicimos y por qué lo hicimos.

Usando comandos de Linux a través de la terminal, deberá crear la siguiente estructura de directorios para este proyecto que identificará de la siguiente manera:

User_proyect_year_publication donde:

- **User**: corresponde a su apellido.
- **Project:** corresponde al nombre de su proyecto hipotético en el cual se encuentra trabajando.
- **Year**: es el año de la publicación o investigación que está realizando.
- **Publication**: corresponde a una revista en donde quiere publicar sus resultados.

Este directorio, en mi caso, ***Soler_humanTonsils_2023_Nature*** será creado dentro del directorio ***Documents*** que a su vez contendrá 3 directorios (***data, code*** *y* ***submission***) y dos archivos (***README*** *y* ***LICENSE***). El directorio ***data*** contendrá a su vez dos subdirectorios llamados ***raw*** y **processed.**

Parte II: Obtención de datos

Ahora, va a obtener el conjunto de datos con los cuales va a trabajar. Estos pueden ser encontrados en la propia actividad propuesta en el campus virtual:

- *Actividades/Portafolio de pruebas aplicativas/Prueba aplicativa 2/SRR1984406_1.fastq*

Seguidamente guarde este archivo en el directorio ***data/raw*** y responda a cada una de las preguntas que se le indican, adicionando siempre los comandos empleados en cada caso y una captura de pantalla de ellos junto a su resultado de ejecución.

1. ¿Cuántas secuencias podemos encontrar en el archivo? (0,5 pts)

El número de secuencias en el archivo SRR1984406_1.fastq es **8246.**

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm raw]$ grep "@SRR19" SRR1984406_1.fastq | wc -l
8246
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm raw]$
```

```
grep "@SRR19" SRR1984406_1.fastq | wc -l

# con ‘grep’ junto al argumento “@SRR19” seleccionamos la primera línea de las 4 que forman una lectura en el formato fastq. Así seleccionamos todos los reads que a la vez tienen asignada una secuencia. Concatenando con ‘wc -l’ contamos el número de líneas, lo que nos da el número de secuencias en el archivo fastq.
```

Nota: las preguntas siguientes (2 y 3) van a ser respondidas mediante la ejecución de un script (script_p2_p3.sh) que está guardado en el directorio /code del proyecto.

2. ¿Cuál es la longitud promedio de las secuencias incluidas en el archivo a estudio? (1 pts)

La media de longitud de las secuencias es **130.48775163715740965316**

3. ¿Cuál es la longitud mínima y máxima de las secuencias incluidas en el archivo? (1 pts)

La longitud mínima de las secuencias es **75**

La longitud máxima de secuencia es **135**

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$ tree
.
├── code
│   └── script_p2_p3.sh
├── data
│   ├── processed
│   │   └── read_lengths.txt
│   └── raw
│       └── SRR1984406_1.fastq
├── LICENCE.txt
├── README.txt
└── submission

5 directories, 5 files
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$ bash ./code/script_p2_p3.sh
Respuestas a la pregunta 2:
La suma de los valores de longitud de sequencia es 1076002
El numero de secuencias es 8246
La media de longitud de las secuencias es 130.48775163715740965316

Respuestas a la pregunta 3:
La longitud minima de las secuencias es 75
La longitud maxima de secuencia es 135
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$
```

A continuación, se muestra una captura de pantalla del **script_p2_p3.sh** en el editor de texto 'pluma':

*script_p2_p3.sh

```
#!/bin/bash
#Este script calcula la media de longitud de secuencias de un archivo .fastq y proporciona las longitudes minimas y maximas de las secuencias
#Este script se guarda y ejecuta desde /home/gabriel.tedone/Documentos/Tedone_eDNAread_2023_Nature/code
#El archivo .fastq tiene que estar en el directorio /home/gabriel.tedone/Documentos/Tedone_eDNAread_2023_Nature/data/raw/X.fastq

#selecciona la longitud de secuencia de cada lectura del archivo .fastq y redirige los valores al archivo read_lengths.txt
sed -n '1~4p' /home/gabriel.tedone/Documentos/Tedone_eDNAread_2023_Nature/data/raw/SRR1984406_1.fastq | cut -d '=' -f 2 > /home/gabriel.tedone/
Documentos/Tedone_eDNAread_2023_Nature/data/processed/read_lengths.txt

#se establecen las variables
valores=$(cat /home/gabriel.tedone/Documentos/Tedone_eDNAread_2023_Nature/data/processed/read_lengths.txt) #la variable toma los valores del archivo
read_lengths.txt
num_de_secuencias=$(grep "@SRR19" /home/gabriel.tedone/Documentos/Tedone_eDNAread_2023_Nature/data/raw/SRR1984406_1.fastq | wc -l) #calcula el
numero de lecturas del archivo .fastq
suma=0
media=0
minlength=0
maxlength=0

#bucle iterativo para sumar los valores de longitud a la variable suma
for i in $valores
do
    suma=$((suma+i))
done

#calculo de la media
media=$(echo $suma/$num_de_secuencias | bc -l)  # redirigiendo la operacion aritmetica al comando 'bc -l' se consigue el valor decimal de la division

echo "Respuestas a la pregunta 2:"
echo "La suma de los valores de longitud de sequencia es $suma
El numero de secuencias es $num_de_secuencias
La media de longitud de las secuencias es $media"

#ordenacion del archivo read_lenghts.txt y busqueda de max y min
minlength=$(sort -n /home/gabriel.tedone/Documentos/Tedone_eDNAread_2023_Nature/data/processed/read_lengths.txt | head -n1)
maxlength=$(sort -n /home/gabriel.tedone/Documentos/Tedone_eDNAread_2023_Nature/data/processed/read_lengths.txt | tail -n1)

echo " "
echo "Respuestas a la pregunta 3:"
echo "La longitud minima de las secuencias es $minlength"
echo "La longitud maxima de secuencia es $maxlength"
```

4. ¿Cuántas veces aparece el patrón "ATGATG" en el archivo descargado? (0,5 pts)

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$ tree
.
├── code
│   └── script_p2_p3.sh
├── data
│   ├── processed
│   │   └── read_lengths.txt
│   └── raw
│       └── SRR1984406_1.fastq
├── LICENCE.txt
├── README.txt
└── submission

5 directories, 5 files
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$ grep -o "ATGATG" ./data/raw/SRR1984406_1.fastq
 | wc -l
835
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$
```

```
grep -o “ATGATG” ./data/raw/SRR1984406_1.fastq | wc -l
```

El patrón “ATGATG” aparece **835** veces en el archivo .fastq descargado

A continuación, va a convertir el archivo que se presenta en formato FASTQ en otro archivo con un formato un tanto distinto, el formato FASTA.

5. Para ello, deberá seleccionar la primera y la segunda línea de cada una de las 4 líneas que conforman cada una de las lecturas o *reads* del archivo SRR1984406_1.fastq y así confeccionar un archivo final llamado *all_sequences.fasta* que contenga el *header* o cabecera y la secuencia asociada a cada lectura. Recuerda guardar este archivo en el directorio *data/processed*. Incluya una captura de pantalla con el código empleado para realizar esta conversión de formato y visualice las 5 primeras líneas del archivo *all_sequences.fasta* en el directorio determinado. (1 pts)

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm raw]$ sed -n '1~4s/^@/>/p;2~4p' SRR1984406_1.fastq > ../processed/all_sequen
ces.fasta
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm raw]$ head -n 5 ../processed/all_sequences.fasta
>SRR1984406.1 1 length=135
GACGACTGCCATCTGAACGTGTGGAATCAACGGAGCCACATCTGACTTCCAGTATCCATCCGAAGTTCTCCATTCAATAGTGAGGAATCTGACGACTGCCATCTGAACGTGTGGAATCAACGGAGCCAC
ATCTGA
>SRR1984406.2 2 length=134
TTTGGGAATTTCCTGTATCCATCCGAAGTTCTCCATTCAATAGTGAGGAATCTGACGACTGCCATCTGAACGTGTGGAATCAACGGAGCCACATCTGACTAGTGCCCAGCATGAGCGACTCCACCGCCA
TTGGG
>SRR1984406.3 3 length=134
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm raw]$
```

```
sed -n ‘1~4s/^@/>/p;2~4p’ SRR1984406_1.fastq > ../processed/all_sequences.fasta
```

con el comando sed -n seleccionamos del archivo fastq de raw data, seleccionamos el encabezado de las lecturas (1~4) y mediante la substitución (s) de del carácter inicial (@ por >) lo cambiamos a formato fasta. También se imprime este encabezado modificacado (p). Se seleccionan las secuencias, cada 4 líneas y empezando por la 2ª línea y se imprimen (2~4p). El output de la instrucción se redirige (>) al archivo all_sequences.fasta bajo el directorio /data/processed y se visualizan sus primeras 5 líneas con...

```
head -n 5 ../processed/all_sequences.fasta
```

6. Una vez creado, seleccione aleatoriamente (genere aleatoriamente los números) 5 lecturas (con su cabecera y su secuencia asociada) del archivo all_sequences.fasta y seguidamente guárdelas en 5 archivos de texto distintos denominados secuencia1.fasta, secuencia2.fasta y así sucesivamente. Cada uno de ellos contendrá lo siguiente:

>SRR1984406.1 1 length=135

GACGACTGCCATCTGAACGTGTGGAATCAACGGAGCCACATCTGACTTCCAGTATCCATCCGAAGTTCTCCATTCA
ATAGTGAGGAATCTGACGACTGCCATCTGAACGTGTGGAATCAACGGAGCCACATCTG

Incluya una captura de pantalla con el código empleado para generar cada uno de los archivos creados y con el comando ls muestre la creación de los mismos en el directorio *data/processed* (1 pts)

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$ grep -o ">" all_sequences.fasta | wc -l
8246
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$
```

grep -o “>” all_sequences.fasta | wc -l

Vemos que la cantidad de secuencias en el archivo all_sequences.fasta es **8246.**

Para generar 5 números aleatorios entre 1 y 8246 se utiliza el comando shuf:

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$ shuf -i 1-8246 -n 5
6247
6872
4703
3651
4413
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$
```

shuf -i 1-8246 -n 5

#-i indica el rango de los números y -n la cantidad a generar

Lo que nos genera 5 números aleatorios: 6247, 6872, 4703, 3651 y 4413 con los cuales seleccionar las secuencias.

El **problema** viene al seleccionar las secuencias del archivo *all_sequences.fasta* a partir de estos números, ya que, en un archivo fasta, un número de línea **impar** corresponde con el encabezado de la secuencia mientras que un número de línea **par** corresponde con la secuencia en si.

Entonces, para proseguir, **si el número aleatorio generado es impar**, se selecciona la línea con el número impar en cuestión (que corresponde al encabezado) y la línea que le sigue (que corresponde a su secuencia), mientras que, **si el número aleatorio generado es par**, se selecciona la línea con el número par en cuestión (que corresponde a la secuencia) y la línea que le precede (que corresponde con su encabezado).

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$ sed -n '6247p;6248p' all_sequences.fasta > secuencia1.fasta
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$ sed -n '6871p;6872p' all_sequences.fasta > secuencia2.fasta
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$ sed -n '4703p;4704p' all_sequences.fasta > secuencia3.fasta
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$ sed -n '3651p;3652p' all_sequences.fasta > secuencia4.fasta
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$ sed -n '4413p;4414p' all_sequences.fasta > secuencia5.fasta
```

sed -n ‘6247p;6248p’ all_sequences.fasta > secuencia1.fasta

...

Mediante el comando sed se seleccionan, del archivo all_sequences.fasta, las secuencias que corresponden a los 5 números aleatorios generados teniendo en cuenta si el número es par o impar y se redirigen al archivo secuenciaX.fasta.

Se observa el contenido del directorio /data/processed mediante...

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$ ls -l
total 1376
-rw-r--r-- 1 UNIVERSIDADVIU\gabriel.tedone UNIVERSIDADVIU\domain users 1353645 jun 13 17:39 all_sequences.fasta
-rw-r--r-- 1 UNIVERSIDADVIU\gabriel.tedone UNIVERSIDADVIU\domain users   32477 jun 12 22:45 read_lengths.txt
-rw-r--r-- 1 UNIVERSIDADVIU\gabriel.tedone UNIVERSIDADVIU\domain users     168 jun 13 18:56 secuencia1.fasta
-rw-r--r-- 1 UNIVERSIDADVIU\gabriel.tedone UNIVERSIDADVIU\domain users     168 jun 13 18:56 secuencia2.fasta
-rw-r--r-- 1 UNIVERSIDADVIU\gabriel.tedone UNIVERSIDADVIU\domain users     167 jun 13 18:56 secuencia3.fasta
-rw-r--r-- 1 UNIVERSIDADVIU\gabriel.tedone UNIVERSIDADVIU\domain users     168 jun 13 18:56 secuencia4.fasta
-rw-r--r-- 1 UNIVERSIDADVIU\gabriel.tedone UNIVERSIDADVIU\domain users     168 jun 13 18:57 secuencia5.fasta
```

Y el contenido de los archivos secuenciaX.fasta mediante ...

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$ cat secuencia*.fasta
>SRR1984406.3124 3124 length=134
CATCATCCATTGTAACATCCGATGAGACAGATACTATTGAAAAGATTGCATTAAGTCTCATTCTTTCCCAGAAAACAAGAAGCACACCAAGAGCCAGCATGAGTGAAGAGTTTTTCTTTCAATCAGTTAATGGC
>SRR1984406.3436 3436 length=134
GTTCCCAAGTGTGAATGACAGACTTGTGCGGGTGGATTCTGTGGATTCATCACTCTCAGTGTACAAAAAATTAGTTGAGAGAATGATAAAATCACAGAATCCACCCGCACAAGTCTGTCATTCACACTTGGGAA
>SRR1984406.2352 2352 length=133
CTGCTAGGTCCTGAAGTTTGTGACCTTGTTCGTTCTACATGTTACAGCACTAAGTTGTCAGATCTCCCTACAGACCTGCTCAGAGAGGAGGCTTATAAAAGAATTGTTGGTGAGACAATGTCAAGAGCCTGTC
>SRR1984406.1826 1826 length=134
ATGTAAGATTCCCTGGCCCATATCAAACCAAAAGTGGATTTGTGAAGGCACGGGGTTGTTGGATAGAACAGCATCCACCATAAATGTTTCGAAAGTTCTTTCTTTCCCAATTTGTCTGTCTCTTGATGTAAGCT
>SRR1984406.2207 2207 length=134
GGATAGACTCATAGCCATAAAATATGGTGCACCAACATTGAGAAATATCCCGTAATGCTTGTGAATTTGTAACCGTTGTTGAAAGCATGCTTTCTCCTCTATATTTATTGGAGATTGACAAGAATGAACTTTGG
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm processed]$
```

7. Con cada uno de los cinco archivos creados deberá realizar un script en BASH (esté deberá estar almacenado en el directorio *code*) que sea capaz de leer cada archivo, específicamente su secuencia, y deberá reportar por la salida estándar un informe con las siguientes estadísticas de las secuencias:

- Longitud de cada secuencia.
- Identificación del nucleótido inicial y final de cada secuencia.
- Contenido de GC (porcentaje de nucleótidos G y C) por cada secuencia.
- ¿Cuál es la subsecuencia más larga que contenga un único nucleótido? Por ejemplo, en la secuencia CGAAAAACTACGTTTTCATCCCC, la respuesta correcta sería AAAAA.

Pegue el script generado y adicione una captura de pantalla con la ejecución realizada (2 pts)

Adjunto una captura de pantalla del script en el editor de pluma ya que creo que se lee mejor que pegando el texto 'crudo' del script:

script_p7.sh

```
#!/bin/bash
#Importante que se ejecute el script desde el subdirectorio /code del proyecto!
#Este script parsea varios archivo mono-fasta y proporciona informacion sobre sus correspondientes secuencias

for file in ../data/processed/secuencia*.fasta
do

    seq=($(cat $file | tail -n1)) #se asigna la secuencia a la variable 'seq' para parsear su cadena de nucleotidos
    length=$(echo ${#seq}) #calcula la longitud de la variable
    initial=$(echo ${seq:0:1}) #toma el valor del primer valor de la cadena
    final=$(echo ${seq:$length-1:1}) #toma el valor del ultimo valor de la cadena
    GCcount=$(cat $file | tail -n1 | grep -E -o "[GC]" | wc -l) #cuenta el numero de nucleotidos G o C
    GCcontent=$(echo "($GCcount/$length)*100" | bc -l) #calcula el GC%
    repeatingseq=$(echo $seq | grep -Eo "(A+|T+|G+|C+)" | awk '{print $0, length}' | sort -k 2 -n | tail -n1 | cut -d " " -f1)
    #busca la secuencia de un unico nucleotidos repetido mas larga

    echo "------------------------"
    echo "La secuencia $file tiene:"
    echo "Una longitud de $length nucleotidos"
    echo "Su nucleotido inicial es $initial"
    echo "Su nucleotido final es $final"
    echo "El numero de nucleotidos G/C es $GCcount, lo que, sobre un total de $length nucleotidos, corresponde a  $GCcontent % de G/C"
    echo "La subsecuencia de un mismo caracter mas larga es $repeatingseq"
    echo

done
```

Al ejecutar el script_p7.sh desde el directorio /code se obtienen los resultados:

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm code]$ . script_p7.sh
------------------------
La secuencia ../data/processed/secuencia1.fasta tiene:
Una longitud de 134 nucleotidos
Su nucleotido inicial es C
Su nucleotido final es C
El numero de nucleotidos G/C es 51, lo que, sobre un total de 134 nucleotidos, corresponde a  38.05970149253731343200 % de G/C
La subsecuencia de un mismo caracter mas larga es TTTTT

------------------------
La secuencia ../data/processed/secuencia2.fasta tiene:
Una longitud de 134 nucleotidos
Su nucleotido inicial es G
Su nucleotido final es A
El numero de nucleotidos G/C es 57, lo que, sobre un total de 134 nucleotidos, corresponde a  42.53731343283582089500 % de G/C
La subsecuencia de un mismo caracter mas larga es AAAAAA

------------------------
La secuencia ../data/processed/secuencia3.fasta tiene:
Una longitud de 133 nucleotidos
Su nucleotido inicial es C
Su nucleotido final es C
El numero de nucleotidos G/C es 60, lo que, sobre un total de 133 nucleotidos, corresponde a  45.11278195488721804500 % de G/C
La subsecuencia de un mismo caracter mas larga es AAAA

------------------------
La secuencia ../data/processed/secuencia4.fasta tiene:
Una longitud de 134 nucleotidos
Su nucleotido inicial es A
Su nucleotido final es T
El numero de nucleotidos G/C es 55, lo que, sobre un total de 134 nucleotidos, corresponde a  41.04477611940298507400 % de G/C
La subsecuencia de un mismo caracter mas larga es GGGG

------------------------
La secuencia ../data/processed/secuencia5.fasta tiene:
Una longitud de 134 nucleotidos
Su nucleotido inicial es G
Su nucleotido final es G
El numero de nucleotidos G/C es 49, lo que, sobre un total de 134 nucleotidos, corresponde a  36.56716417910447761100 % de G/C
La subsecuencia de un mismo caracter mas larga es AAAA

(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm code]$
```

8. A continuación, implemente otro script en BASH (esté deberá estar almacenado en el directorio *code*) que lea cada uno de los cinco archivos generados y realice la transcripción de cada secuencia de ADN a su secuencia complementaria reversa de ARN. Una vez transformadas las secuencias, deberá computar la frecuencia de aparición de cada una de las 4 bases nitrogenadas (en porcentaje y redondeado a dos decimales) y reportarlas por la salida estándar para cada una de las 5 secuencias analizadas.

Pegue el script generado y adicione una captura de pantalla de su ejecución por terminal (2 pts)

script_p8.sh

```
#!/bin/bash
#Este script se tiene que ejecutar desde /code y parsea las secuencias*.fasta en ../data/processed/
#Este script reporta por la salida estandar la secuencia de ADN del archivo .fasta, su secuencia reversa y complementaria de ARN
con asociadas sus frecuencias de aparcicion de cada nucleotido (A, U, G, C)
#Las secuencias estan reportadas en sentido 5' --> 3'

for file in ../data/processed/secuencia*.fasta
do
    seq=$(cat $file | tail -n1)                              #obtiene la secuencia
    revcomplRNA=$(echo $seq | tr "ATGC" "UACG" | rev)        #obtiene la cadena complementaria y reversa de ARN
    length=$(echo ${#revcomplRNA})                           #obtiene la longitud de la secuencia

    Acount=$(echo $revcomplRNA | grep -Eo "A" | wc -l)       #cuenta el numero de nucleotidos A en la secuencia de ARN
    freqA=$(echo "100*($Acount/$length)"| bc -l)             #calcula su porcentaje sobre la longitud de la secuencia
    Ucount=$(echo $revcomplRNA | grep -Eo "U" | wc -l)       #idem para "U"
    freqU=$(echo "100*($Ucount/$length)" | bc -l)
    Gcount=$(echo $revcomplRNA | grep -Eo "G" | wc -l)       #idem para "G"
    freqG=$(echo "100*($Gcount/$length)" | bc -l)
    Ccount=$(echo $revcomplRNA | grep -Eo "C" | wc -l)       #idem para "C"
    freqC=$(echo "100*($Ccount/$length)" | bc -l)

    echo "----------------------"
    echo -e "De la secuencia de ADN en $file : \n5' $seq 3'"
    echo -e "Su secuencia reversa y complementaria de ARN es: \n5' $revcomplRNA 3'"
    echo "Donde las frecuencias de aparicion de los nucleotidos son:"
    echo "$freqA" | awk '{printf("%.2f",$1)}' ; echo "% A" #con awk '{printf("%.2f",$1)}' se redondea a 2 decimales el porcentaje
    echo "$freqU" | awk '{printf("%.2f",$1)}' ; echo "% U"
    echo "$freqG" | awk '{printf("%.2f",$1)}' ; echo "% G"
    echo "$freqC" | awk '{printf("%.2f",$1)}' ; echo "% C"
done
```

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm code]$ . script_p8.sh
----------------------
De la secuencia de ADN en ../data/processed/secuencia1.fasta :
5' CATCATCCATTGTAACATCCGATGAGACAGATACTATTGAAAAGATTGCATTAAGTCTCATTCTTTCCCAGAAAACAAGAAGCACACCAAGAGCCAGCATGAGTGAAGAGTTTTTCTTTCAATCAGTTAATGGC 3'
Su secuencia reversa y complementaria de ARN es:
5' GCCAUUAACUGAUUGAAAGAAAAACUCUUCACUCAUGCUGGCUCUUGGUGUGCUUCUUGUUUUCUGGGAAAGAAUGAGACUUAAUGCAAUCUUUUCAAUAGUAUCUGUCUCAUCGGAUGUUACAAUGGAUGAUG 3'
Donde las frecuencias de aparicion de los nucleotidos son:
26.87% A
35.07% U
20.90% G
17.16% C
----------------------
De la secuencia de ADN en ../data/processed/secuencia2.fasta :
5' GTTCCCAAGTGTGAATGACAGACTTGTGCGGGTGGATTCTGTGGATTCATCACTCTCAGTGTACAAAAAATTAGTTGAGAGAATGATAAAATCACAGAATCCACCCGCACAAGTCTGTCATTCACACTTGGGAA 3'
Su secuencia reversa y complementaria de ARN es:
5' UUCCCAAGUGUGAAUGACAGACUUGUGCGGGUGGAUUCUGUGAUUUUAUCAUUCUCUCAACUAAUUUUUUGUACACUGAGAGUGAUGAAUCCACAGAAUCCACCCGCACAAGUCUGUCAUUCACACUUGGGAAC 3'
Donde las frecuencias de aparicion de los nucleotidos son:
26.12% A
31.34% U
20.15% G
22.39% C
----------------------
De la secuencia de ADN en ../data/processed/secuencia3.fasta :
5' CTGCTAGGTCCTGAAGTTTGTGACCTTGTTCGTTCTACATGTTACAGCACTAAGTTGTCAGATCTCCCTACAGACCTGCTCAGAGAGGAGGCTTATAAAAGAATTGTTGGTGAGACAATGTCAAGAGCCTGTC 3'
Su secuencia reversa y complementaria de ARN es:
5' GACAGGCUCUUGACAUUGUCUCACCAACAAUUCUUUUAUAAGCCUCCUCUCUGAGCAGGUCUGUAGGGAGAUCUGACAACUUAGUGCUGUAACAUGUAGAACGAACAAGGUCACAAACUUCAGGACCUAGCAG 3'
Donde las frecuencias de aparicion de los nucleotidos son:
29.32% A
25.56% U
21.05% G
24.06% C
----------------------
De la secuencia de ADN en ../data/processed/secuencia4.fasta :
5' ATGTAAGATTCCCTGGCCCATATCAAACCAAAAGTGGATTTGTGAAGGCACGGGGTTGTTGGATAGAACAGCATCCACCATAAATGTTTCGAAAGTTCTTTCTTTCCCAATTTGTCTGTCTCTTGATGTAAGCT 3'
```

```
21.05% G
24.06% C
--------------------
De la secuencia de ADN en ../data/processed/secuencia4.fasta :
5' ATGTAAGATTCCCTGGCCCATATCAAACCAAAAGTGGATTTGTGAAGGCACGGGGTTGTTGGATAGAACAGCATCCACCATAAATGTTTCGAAAGTTCTTTCTTTCCCAATTTGTCTGTCTCTTGATGTAAGCT 3'
Su secuencia reversa y complementaria de ARN es:
5' AGCUUACAUCAAGAGACAGACAAAUUGGGAAAGAAAGAACUUUCGAAACAUUUAUGGUGGAUGCUGUUCUAUCCAACAACCCCGUGCCUUCACAAAUCCACUUUUGGUUUGAUAUGGGCCAGGGAAUCUUACAU 3'
Donde las frecuencias de aparicion de los nucleotidos son:
32.09% A
26.87% U
20.15% G
20.90% C
--------------------
De la secuencia de ADN en ../data/processed/secuencia5.fasta :
5' GGATAGACTCATAGCCATAAAATATGGTGCACCAACATTGAGAAATATCCCGTAATGCTTGTGAATTTGTAACCGTTGTTGAAAGCATGCTTTCTCCTCTATATTTATTGGAGATTGACAAGAATGAACTTTGG 3'
Su secuencia reversa y complementaria de ARN es:
5' CCAAAGUUCAUUCUUGUCAAUCUCCAAUAAAUAUAGAGGAGAAAGCAUGCUUUCAACAACGGUUACAAAUUCACAAGCAUUACGGGAUAUUUCUCAAUGUUGGUGCACCAUAUUUUAUGGCUAUGAGUCUAUCC 3'
Donde las frecuencias de aparicion de los nucleotidos son:
32.09% A
31.34% U
16.42% G
20.15% C
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm code]$
```

9 . Seguidamente debe completar el archivo de texto llamado LICENSE, donde deberá indicar que tipo de licencia quiere utilizar para limitar o no el uso, la modificación y la distribución de su código BASH desarrollado. Es muy importante comenzar a concienciarse sobre las posibles licencias que podemos otorgarles a nuestros scripts, por ello, busque información en la red sobre los distintos tipos e indique cuál se adaptaría mejor a sus preferencias y el motivo de dicha elección. **Adjunte una captura de pantalla con el contenido de este archivo (0,5 pts).**

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$ cat LICENCE.txt
MIT License

Copyright (c) 2023 Gabriele Tedone

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$
```

10. Además, deberá completar el archivo README, explicando en él, el contenido de cada uno de los directorios creados en el proyecto, especificando las características más relevantes de cada uno de ellos. En este archivo puede añadir la información más relevante que considere. **Adjunte una captura de pantalla con el contenido de este archivo (0,25 pts).**

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$ cat README.txt
This directory contains the source code for the project Tedone_eDNAread_2023_Nature.
This project arised from the need to resolve an assignment for the VIU master's in bioinformatics subject 01MBIF: Bash and Shell
scripting.

This project works on a raw .fastq file, processing the data into a .fasta format, treating the DNA secuences from a biological p
oint of view and obtaining biologicaly relevant information and results from the raw sequencing data.

The raw data containing the  .fastq file can be found in /data/raw/
Created or modified data files from the raw data source can be found under /data/processed
The scripts parsing, modifing and obtaining results from the raw and processed data can be found under /code.

To execute the several scripts, read the script header under the shebang to know where to execute the script from (script_p2_p3.s
h must be executed in the root of the source code (.) while script_p7.sh and script_p8.sh must be executed in the /code subdirect
ory)
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$
```

11. Finalmente, genere una copia de esta actividad cumplimentada en formato PDF y trasládela al directorio submission donde se almacenará la primera versión de su trabajo realizado. **Para acabar este proyecto, incluya una captura de pantalla con la estructura de directorios actual usando el comando *tree* desde el terminal (0,25 pts).**

```
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$ pwd
/home/gabriel.tedone/Documentos/Tedone_eDNAread_2023_Nature
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$ tree
.
├── code
│   ├── script_p2_p3.sh
│   ├── script_p7.sh
│   └── script_p8.sh
├── data
│   ├── processed
│   │   ├── all_sequences.fasta
│   │   ├── read_lengths.txt
│   │   ├── secuencia1.fasta
│   │   ├── secuencia2.fasta
│   │   ├── secuencia3.fasta
│   │   ├── secuencia4.fasta
│   │   └── secuencia5.fasta
│   └── raw
│       └── SRR1984406_1.fastq
├── LICENCE.txt
├── README.txt
└── submission

5 directories, 13 files
(base) [UNIVERSIDADVIU\gabriel.tedone@a-4mgp76lrrpfm Tedone_eDNAread_2023_Nature]$
```