Introducción a Linux para Bioinformática

LandaLab

2025-09-17

Contents

Chapter 1

Bienvenid@s

Este repositorio contiene el material del taller "Introducción a Linux", un espacio abierto para todas las personas que quieran aprender a analizar y visualizar datos con Linux
No importa tu edad, formación o experiencia previa: si tienes curiosidad y ganas de aprender, este taller es para ti.

1.1 Sobre el taller

- **Lenguaje**: Linux
- **Nivel**: Principiante

- **Modalidad**: Práctico, con ejercicios guiados y codificación en vivo

- **Dirigido a**: Cualquier persona interesada en aprender Linux para Bioinformática

- **Requisitos**: Ninguno. No necesitas saber programación ni instalar ningún programa. Sólo necesitas una computadora con acceso a internet

---

1.2 ¿Qué aprenderás?

Al finalizar el taller, podrás:

- Comunicarte con Linux a través de Shell
- Manipular archivos

- Crear proyectos
- Automatizar tareas
- Escribir scripts reproducibles

- Comprender los principios básicos del análisis de datos en Linux
- Descargar secuencias genómicas
- Instalar programas

1.3 Plataforma

Trabajaremos en una plataforma en línea, por lo que **no necesitas instalar nada en tu computadora**. El acceso será gratuito y se proporcionará durante el taller.

1.4 Agradecimientos

Este taller es parte de un esfuerzo por compartir herramientas abiertas, accesibles y colaborativas. Queremos que más personas se acerquen al mundo de los datos y la ciencia sin barreras. ¡Gracias por formar parte!

1.5 ¿Dudas o comentarios?

Puedes abrir un Issue o escribirnos durante el taller.
¡Estamos aquí para aprender junt@s!

Chapter 2

Introducción a Linux

Chapter 3

Navegar en el sistema de ficheros

todo lo de este archivo hasta atajos en el teclado

https://github.com/DianaOaxaca/Introduccion_linux_para_bioinformatica/blob/main/Comandos_utiles.md

Chapter 4

Manejo de archivos

4.1 Código reproducible

"Los sucesos únicos no reproducibles, no tienen importancia para la ciencia" -Karl Popper, the Logic of Scientific, 1959.

¿Qué se necesita para reproducir el resultado de un análisis computacional?

- **Los datos:**
  - La fuente de donde se descargaron.
  - La versión de la fuente asociada a ellos.
- **Los programas utilizados para analizarlos:**
  - La versión de cada uno de los programas.
  - Los valores de cada uno de los argumentos utilizados.

Argumentos a tener en cuenta para tener buenas prácticas:

- Actualización de bases de datos.
- Siempre existirán excepciones que no cumplen con las suposiciones de tu código.
- **Qué un programa genere un resultado no significa que el resultado sea correcto.**
- Todo lo que se te puede olvidar, **¡se te va a olvidar!**
  - Fuente
  - Suposiciones iniciales
  - ¿Qué genera esa función compleja y rebuscada que parecía una joya en su momento?

SIEMPRE, documenta tu código

Comienza por la organización: La estructura de directorios debe estar organizada, lo mejor es tener un directorio de trabajo por proyecto, los pasos del proyecto se organizarán en subdirectorios.

La estructura de directorios propuesta para estas asesorías es:

`Linux` Este es el directorio principal del proyecto.

`data` En este directorio van los datos de entrada para el proyecto.

`src` En este directorio van los scripts ya probados y funcionales.

`results` En este directorio van los resultados generados.

4.2 Bases de datos y secuencias biológicas

Base de datos: Es una colección organizada de información estructurada, o datos, normalmente almacenados electrónicamente en un sistema informático.

En bioinformática se utilizan diversas bases de datos, algunos ejemplos son:

De diversos organismos:

NCBI: https://www.ncbi.nlm.nih.gov/

Genomas de referencia NCBI: https://ftp.ncbi.nlm.nih.gov/genomes/refseq/

ENSEMBL: https://www.ensembl.org/index.html

UCSC Table Browser: https://genome.ucsc.edu/cgi-bin/hgTables

Dedicadas a organismos específicos:

Ecocyc

Flybase

Wormbase

Especializadas en un tema particular:

ENCODE: Elementos funcionales del genoma humano

RegulonDB: Regulación transcripcional de E. coli

Pfam: Familias proteícas

miRBase: Secuencias de miRNA y sus blancos

Secuencias biológicas: Archivo que contiene la secuencia de genes, genomas y/o proteínas.

Fasta: Se compone de un identificador de la secuencia, seguido (por salto de línea) de la secuencia de nucleótidos o aminoácidos de un gene, genoma o proteína.

Fastq: Normalmente se conmpone de cuatro lineas por secuencia

`Line 1` Comienza con '@' seguido del identificador de la secuencia y una descripción opcional.

`Line 2` La secuencia cruda nucleótidos.

`Line 3` Comienza con un '+' opcionalmente incluye el identificador de la secuencia.

`Line 4` Indica los valores de calidad de la secuencia, debe contener el mismo número de símbolos que el número de nucleótidos.

De anotación: GeneBank o tabulares GFF, GTF, GFF3

Tabulares: Una línea por cada elemento. Cada línea DEBE contener 9 campos. Los campos DEBEN estar separados por tabuladores. Todos los campos DEBEN contener un valor, los campos vacíos se denotan con '.'

`seqname` Nombre del cromosoma

`source` Nombre del programa que generó ese elemento

`feature` Tipo de elemento

`start` Posición de inicio

`end` Posición de final

`score` Un valor de punto flotante

`strand` La cadena (+ , -)

`frame` Marco de lectura

`attribute` Pares tag-value, separados por coma, que proveen información adicional

Secuencia FASTA

```
>ID_secuencia,metadatos de identificación
ATGCCCGGTAAAGGATCCCCCCTATGCCGTATAGC
>ID_secuencia,metadatos de identificación
MIPEKRIIRRIQSGGCAIHCQDCSISQLCIPFTLNEHELDQLDNI
```

Secuencia Fastq

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

Agunos foros para pedir ayuda:

Stackoverflow: https://stackoverflow.com/

Biostars: https://www.biostars.org/

Researchgate: https://www.researchgate.net/

IAs

4.3 Manejo de archivos

wget, curl, shasum, md5sum, diff, scp, rsync, gunzip, unzip, tar, head, tail, more, less, cat, >, », nano

Link para el genoma de Raoultella terrigena:

https://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/Raoultella_terrigena/reference/GCF_012029655.1_ASM1202965v1/

Para descargar archivos a nuestra computadora desde la terminal, se requiere utilizar protocolos de transferencia, los comandos `wget` o `curl` funcionan para ello. Veámos un ejemplo:

- Crea el directorio principal de trabajo para este proyecto y sus subdirectorios asociados.
- Accede al directorio data
- Descarga el genoma representativo de Raoultella terrigena de la Refseq de NCBI
- Ahora descarga las secuencias proteícas

¿Notaste algún cambio? - Vuelve a descargar el genoma de Raoultella terrigena, pero ahora asegúrate de guardar la secuencia en un archivo llamado Raoultella_terrigena.fasta.gz

```
wget https://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/Raoultella_terrigena/referen
```

Descarga el archivo de aminoácidos de Raoultella terrigena con el nombre `Raoultella_terrigena.faa.gz`

```
wget -O Raoultella_terrigena.faa.gz https://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacter
```

¿Qué ocurrió?

`curl` Realiza la misma función básica que wget , las diferencias principales son: El output lo imprime a standar output, imprime varias estadísticas útiles sobre la descarga.

```
curl https://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/Raoultella_terrigena/referen
```

Con `curl`

```
curl
https://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/Raoultella_terrigena/reference/GC
```

¿Cómo comprobamos que dos archivos son idénticos? Hemos descargado dos veces la secuencia genómica de *Raoultella terrigena*. Comprobemos que ambos archivos son idénticos.

Prueba con `diff -s`

```
diff GCF_012029655.1_ASM1202965v1_genomic.fna.gz Raoultella_terrigena.fasta.gz
```

¿Y si comparas el faa vs fasta?

```
diff ../data/Raoultella_terrigena.faa.gz ../data/Raoultella_terrigena.fasta.gz
```

4.4 Redireccionamientos y evaluación de la integridad

> Permite direccionar un resultado a un archivo nuevo. Crea el archivo si no existe y lo sobre escribe si existe.

>> Permite redireccionar un resultado en pantalla o un archivo a otro, sin remplazar o sobreescribir. Crea el archivo si no existe y agrega el nuevo contenido al final, si el archivo existe.

shasum y **md5sum** son programas que generan una suma encriptada única para cada archivo.

Revisemos la integridad de los archivos, descarga el archivo md5checksums.txt del directorio genomes/refseq/bacteria/Raoultella terigena de NCBI

```
curl https://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/Raoultella_terrigena/reference/GCF_012
md5sum_R.terrigena.txt
```

```
shasum Raoultella_terrigena.fasta.gz
md5sum Raoultella_terrigena.fasta.gz
cat md5sum_R.terrigena.txt
```

Redireccionemos los resultados de integridad a un archivo nuevo.

```
md5sum Raoultella_terrigena.fasta.gz >
../resuts/R.terrigena.md5sum.check

cat ../results/R.terrigena.msd5sum.check
cat md5sum_R.terrigena.txt >> ../results/R.terrigena.md5sum.check
cat ../results/R.terrigena.msd5sum.check
```

4.5 Transferencia de archivos

Esta parte no la vamos a practicar porque estamos usando una interfaz gráfica con acceso al servidor, pero si no tienes esta forma de acceso es necesario usar estos comandos, asi que te dejamos el ejemplo. :)

```
`scp` [FUENTE][DESTINO]
#FUENTE=Nombre del archivo que quieres transferir
#DESTINO=Ruta de destino
#Ejemplo de mi computadora al servidor:
scp md5sum_R.terrigena.txt hoaxaca@132.248.220.35:/space31/PEG/hoaxaca
#Me pedirá el password
#Funciona de manera inversa si quieres bajar del servidor a tu computadora
scp hoaxaca@132.248.220.35:/space31/PEG/hoaxaca/md5sum_R.terrigena.txt .#ojo el
destino puede ser con ruta absoluta o relativa, aquí fue relativa "."
#También puedes usar rsync
`rsync -e ssh` [FUENTE][DESTINO]
# -e ssh indica que nos conectaremos al servidor a través de una conexión de
tipo ssh
```

4.6 Compresión y descompresión

Para descomprimir archivos usamos **gunzip**.

```
gunzip Raoultella_terrigena.fasta.gz
gunzip *.gz
```

Para comprimir usamos **gzip**.

```
gzip *.f*
```

¿Y si quiero comprimir directorios? Para ello utilizo tar que es un método de ultra compresión, muy utilizada en datos genómicos. Vamos a comprimir el directorio practica1

```
cd ../../
tar cvzf practica1.tar.gz practica1/
# c = crea un nuevo directorio
# v = muestra el progreso dde la compresión
# z = genera un archivo comprimido en zip (.gz)
# f = para indicar el nombre del archivo comprimido
```

Y para descomprimir?

```
rm -r practica1/
tar -xvf practica1.tar.gz
# ¿qué hace el flag x?
```

4.7 Explorar archivos

Veamos las primeras líneas del archivo de anotación .gtf del genoma de Raoultella terrigena

¿Y, cuál es ese, lo tenemos?

Vamos a descargarlo

```
wget -O Raoultella_terrigena.gtf.gz
https://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/Raoultella_terrigena/reference/GCF_012029655
```

Luego a descomprimirlo

```
gunzip Raoultella_terrigena.gtf.gz
```

Ahora si

```
head Raoultella_terrigena.gtf
```

¿Y si quiero ver las primeras 20 lineas?

```
head -n 20 Raoultella_terrigena.gtf
```

Ahora quiero ver las últimas líneas de un archivo

```
tail Raoultella_terrigena.gtf
```

Veamos el genoma

```
more Raoultella_terrigena.fasta
# Enter => Navegar hacia abajo de línea en línea
# Espacio => Navega hacia abajo de pantalla en pantalla
```

Pero podemos ver archivos con algo más potente

```
less Raoultella_terrigena.gtf
```

`Espacio OR Enter` => Navegar hacia abajo

`b OR flecha arriba` => Navegar hacia arriba

`/WORD` => Búsqueda forward

`n` => Siguiente

`?WORD` => Búsqueda backward

`N` => Anterior

`G` => Ir al final del archivo

`g` => Ir al inicio del archivo

`-S` => Mostrar una línea por renglón

4.8 Ejercicio 02

Estás realizando tu proyecto con Raoultella terrigena y tu directora de tesis te pregunta si hay genes nitrogenasa ubicados en la posición 501417 o 3010433 del genoma representativo de los genomas de referencia de esta especie. Tú debes responderle si están o no en esa posición, y si no están, avisarle en qué posición se encuentran y cuántos son.

Pseudocódigo Es la manera en la que planeas, diseñas la ruta de trabajo que usarás para responder una pregunta. Es el diseño experimental.

Pseudocódigo:

1. Crea el directorio sesion02 dentro del directorio results
2. Descarga el archivo de anotación .gff de Raoultella terrigena en el directorio correspondiente.
3. Revisa su integridad y redirecciona el resultado al directorio correspondiente
4. Descomprime el archivo de anotación
5. Navega al final del archivo
6. Navega al inicio del archivo
7. Busca el gene que inicie en la posición 501417
8. Busca el CDS que termine en la posición 3010433
9. Busca los genes nitrogenasa
10. Escribe los resultados

Chapter 5

Filtrar información

En esta sesión, exploraremos los comandos `|`, `sort`, `cut`, `uniq`, `wc` y `grep` para analizar el genoma de *Raoultella terrigena*. Estos comandos son esenciales para procesar datos en la terminal de manera eficiente.

5.1 Recordatorio: Comandos Básicos

1. Lista los contenidos del directorio raíz y busca si existe un directorio llamado `home` usando `less`:

```
ls /
ls / | less
# Resultado esperado: /home
```

2. Lista los primeros 10 archivos del directorio raíz:

```
ls / | head -10
```

3. Lista los últimos 5 archivos del directorio raíz:

```
ls / | tail -5
```

5.2 Introducción a los Comandos

- **Pipe (`|`)**: Permite conectar la salida de un programa con la entrada de otro, procesando datos en RAM para mayor rapidez, evitando escritura/lectura en disco. A diferencia de `&&`, que ejecuta comandos independientes, `|` requiere que la salida de un comando sea la entrada del siguiente.
- **sort**: Ordena líneas de texto.
- **cut**: Extrae secciones de cada línea.

- **uniq**: Filtra líneas repetidas (requiere que el archivo esté ordenado).
- **wc**: Cuenta líneas, palabras, caracteres o bytes.
- **grep**: Busca patrones en archivos.

Estructura de un Archivo de Anotación (GFF)

Un archivo GFF tiene las siguientes columnas:

seqname: Nombre del cromosoma.

source: Programa que generó el elemento.

feature: Tipo de elemento (e.g., gene, CDS).

start: Posición de inicio.

end: Posición de final.

score: Valor de punto flotante.

strand: Cadena (+, -).

frame: Marco de lectura.

attribute: Pares tag-value con información adicional.

5.3 Features en el Genoma

Tamaño del Genoma de Raoultella terrigena

¿Qué archivo necesitamos?

El archivo Raoultella_terrigena.fasta.

¿Qué comando nos ayuda?

```
wc data/Raoultella_terrigena.fasta
```

Nota: Este comando da una estimación aproximada, ya que incluye bytes del encabezado y saltos de línea.

Guarda los resultados en un archivo:

```
mkdir -p results/sesion3
```

```
echo 'El genoma de Raoultella puede medir:' > results/sesion3/Raoultella_caracteristica
```

```
wc data/Raoultella_terrigena.fasta >> results/sesion3/Raoultella_caracteristicas.txt
```

Número de Cromosomas

¿Qué archivo necesitamos?

El archivo Raoultella_terrigena.gff.

¿Qué comando nos ayuda?

```
cut -f1 data/Raoultella_terrigena.gff | head
```

Nota: El resultado incluye las 8 líneas del encabezado. Para excluirlas:

```
grep -v "#" data/Raoultella_terrigena.gff | cut -f1 | sort | uniq
```

Guarda el resultado

```
echo 'El genoma de Raoultella tiene un cromosoma y es' >> results/sesion3/Raoultella_caracteristi
grep -v "#" data/Raoultella_terrigena.gff | cut -f1 | uniq >> results/sesion3/Raoultella_caracter
```

Número de features

¿Qué archivo necesitamos?

El archivo Raoultella_terrigena.gff.

¿Qué comandos requerimos?

```
cut -f3 data/Raoultella_terrigena.gff | uniq
```

Nota: uniq requiere que las líneas estén ordenadas. Prueba:

```
cut -f3 data/Raoultella_terrigena.gff | sort | uniq
# Alternativa:
cut -f3 data/Raoultella_terrigena.gff | sort -u
```

Número de tipos de features

```
cut -f3 data/Raoultella_terrigena.gff | sort -u | wc -l
```

Quita las lineas comentadas

Fuentes de los datos de anotación

```
cut -f2 data/Raoultella_terrigena.gff | sort -u
```

Número de genes y CDS

Pseudocódigo

1. Acceder a la columna 3 (feature).
2. Contar ocurrencias únicas de cada elemento.

```
cut -f3 data/Raoultella_terrigena.gff | sort | uniq -c
```

Para evitar contar elementos repetidos:

```
cut -f3-5 data/Raoultella_terrigena.gff | sort -u | cut -f1 | sort | uniq -c
```

Genes por cadena

Pseudocódigo

1. Cortar las columnas feature y strand.
2. Ordenar y contar ocurrencias únicas.

```
cut -f3,7 data/Raoultella_terrigena.gff | sort | uniq -c
```

Crea un archivo de anotación ordenado por cadena y por región genómica

Pseudocódigo

1. Acceder a las columnas de cadena (strand) y posiciones genómicas.
2. Ordenar por cadena y luego por posición (numéricamente).

```
sort -k7,7 -k4,4n data/Raoultella_terrigena.gff > results/R_terrigena_strand.gff
```

Cuántos genes hay con diferente nombre?

Pseudocódigo

1. Filtrar registros de tipo gene.
2. Acceder a la columna 9 (atributos).
3. Separar por ; y extraer nombres.
4. Contar valores únicos.

```
grep -P "\tgene\t" data/Raoultella_terrigena.gff | cut -f9 | cut -d ';' -f3 | sort -u
```

5.4 Ejercicios

- **1. Cuántos genes hay con distinto ID?**

  Pseudocódigo

1. Filtrar registros de tipo gene.
2. Acceder a la columna 9.
3. Separar por ; y =, quedándote con el ID.
4. Contar valores únicos.

- **2. Cuántas secuencias proteícas?**

  Tip: usa el archivo de proteinas

- **3. Cuánto mide la secuencia de la proteína WP_000448832.1?**

  Pseudocódigo

  1. Localizar el ID `WP_000448832.1`.
  2. Extraer la secuencia y contar caracteres.

- 4. Cuál es el ID del gene `fnr`

Chapter 6

Obtener más información

En esta sesión, exploraremos los comandos `sed`, `tr` y ciclos `for`. Además aprenderemos a crear ambientes de conda e instalar programas dentro de este.

6.1 Descargar secuencias fastq

6.1.1 Subir archivos al servidor

Trabajaremos con datos de amplicones de la región V3-V4 del 16S rRNA de muestras tres tiempos de fermentación del pulque, estos se obtuvieron con una plataforma ILLUMINA MiSeq (2 x 300 pb) y están en formato FASTQ. Los datos fueron depositados en NCBI y ENA bajo el BioProject **PRJEB13870** del artículo **Deep microbial community profiling along the fermentation process of pulque, a biocultural resource of Mexico**.

Vamos a descargar el reporte del BioProject para obtener las ligas de descarga de cada librería, **asegurate de que el nombre del experimento este marcado en las casillas**, descarguemos el archivo tsv del reporte que se encuentra en la siguiente liga:

https://www.ebi.ac.uk/ena/browser/view/PRJNA556980

Ve a la terminal de tu computadora y sitúate en el directorio en donde se descargó el reporte.

```
cd ~/Downloads/
ls ~/Downloads/reporte.txt
```

El reporte que descargamos nos servirá para obtener los archivos **fastq** de cada librería secuenciada, pero necesitamos tenerlos en el servidor y no en nuestra máquina local. Así que vamos a subir el reporte al servidor.

Con `scp` podemos hacer copias de la computadora al servidor y visceversa.

Sitúate en la terminal de tu máquina local, justo en el directorio donde se encuentra el archivo que acabamos de descargar:

```
#Para subir archivos
scp -P 7915 reporte.txt [USUARIO]@IP:/ruta/data

#Para descargar archivos
#scp -P 7915 [USUARIO]@132.248.15.30:/botete/[USUARIO]/amplicones/data/[archivo] .
```

6.1.2 Filtrar sólo la información útil

- ☒ Ahora regresa a la terminal del servidor y lista el contenido del directorio `data`

Podemos ver que el reporte se aloja en este directorio.

```
less -S reporte.txt
```

Podemos notar que contiene la información de todas las librerías que se secuenciaron en el proyecto, sin embargo, a nosotros sólo nos interesan los que son de amplicones del 16S rRNA. Así que nos quedaremos sólo con esta información.

¿Cuántas de estas corresponden a its?

- ☒ Averigua en la ayuda de grep que hacen las banderas `--color` `-c` `-v`

```
grep --color 'its' data/reporte.txt
```

```
grep -c 'its' data/reporte.txt
```

```
grep -v 'its' data/reporte.txt
```

```
grep -c -v 'its' data/reporte.txt
```

La información que necesitamos para descargar los fastq son: el nombre de la muestra y la liga de descarga. Esta información se encuentra en las columnas `experiment_title` y `fastq_ftp`

```
cut -f3,6 reporte.txt
```

Recordemos especificar el delimitador de las columnas usando la bandera `-d`.

```
cut -f3,6 reporte.txt| cut -d' ' -f4-7 | grep -v 'its'
```

:o Ahora si contiene sólo la información de las librerías del 16S, sin embargo aún nos falta un poco… Si recordamos, cada librería está pareada, si notamos el contenido de la columna 2 veremos que hay dos ligas de descarga separadas por un `;` así que necesitamos obtener cada liga por separado.

sed nos ayuda a sustituir patrones de texto o caracteres.

```
cat data/reporte.txt  | grep -v 'its' |  grep -v 'fastq_ftp' | cut -f5 | sed 's/;/\n/g' | less -S
```

6.1.3 Obtener las secuencias

Recordemos que **wget** permite descargar archivos de la web al servidor o a la máquina local.

Pero como son varios los archivos que necesitamos, haremos un *ciclo for* que nos ayude a optimizar esta tarea.

```
for fq in $(cut -f2 data/reporte_16S.txt); do wget $fq data/; done
```

Lista el contenido de tu directorio data, ¿qué contiene?

6.1.4 Verificar la integridad de las secuencias

Obtengamos el identificador md5 desde el reporte de los datos

Con **tr** podemos cambiar carácteres:

- Traducir un caracter a otro

```
echo "este es un ejemplo" | tr 'e' 'a'
```

- Podemos especificar rangos de caracteres

```
echo "este es un ejemplo" | tr a-z A-Z
```

- O clases de caracteres

```
echo "este es un ejemplo" | tr [:lower:] [:upper:]
```

Veamos ...

```
cut -f3,5 reporte.txt | cut -d' ' -f4-7 | grep -v 'its' | tr';' '\n' | grep -v 'experiment' | cut
```

Ahora generemos el identificador de los datos que ya descargamos

Entra al directorio data

```
md5sum SRR9849602_1.fastq.gz
```

Pero otra vez son muchos, hagamos un ciclo for para esto:

```
for gz in $(ls *.gz); do md5sum $gz ; done
```

Nota: También podríamos descargar todos los archivos de un BioProject de NCBI

```
#fastq-dump --split-files SRR1234567
```

6.2 Calcular contenido de GC

Calcula el porcentaje de GC del gene nifH de *Raoultella terrigena*. (Se suma la cantidad de G + C y se divide entre la longitud de la secuencia)

Descargar el archivo de genes del genoma representativo de *Raoultella terrigena*

Algoritmo:

- 1. Obtener el archivo de secuencias codificantes de *R. terrigena*
- 2. Obtener la secuencia del gene nifH
- 3. Calcular la longitud de la secuencia
- 4. Calcular el contenido de GC
- 5. Calcular el porcentaje de GC del gene nifH

1. Obtener el archivo de secuencias codificantes de *R. terrigena*

```
wget -O Raoultella_terrigena.cds.gz https://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteri
```

No olvides descomprimir

2. Obtener la secuencia del gene nifH

2.1 identificar al gene de interés

```
grep nifH data/Raoultella_terrigena.cds
```

2.2 Definir como obtendré la secuencia, hay varias formas, propongo una muy fácil, con **seqtk**.

Pero como no tenemos seqtk pues vamos a instalarlo … Ve a la sección de instalción

2.2.1 Obtener el identificador de la secuencia sin caracteres especiales

```
grep nifH data/Raoultella_terrigena.cds | sed 's/>//g' > results/nif_headers.txt
```

2.2.2 Obtener la secuencia

```
seqtk subseq data/Raoultella_terrigena.cds results/nif_headers.txt > results/nif.fasta
```

Ahora si, revisa el contenido del archivo que generamos

3.1 Calcular la longitud de la secuencia

```
grep -v '>' results/nifH.fasta | tr -d '\n' | wc -c >> results/GC_nifH.txt
```

3.2 Calcular el contenido de GC

```
grep -v '>' results/nifH.fasta | tr -d '\n' | tr [CG] '\n' | wc
```

Comprobemos con el de AT

```
grep -v '>' results/nifH.fasta | tr -d '\n' | tr [AT] '\n' | wc
```

6.3 Isntalar ambientes conda

Usaremos **conda** que es la herramienta más común para la gestión de entornos de software y paquetes en bioinformática.

```
conda create --name seqtk_env seqtk -c bioconda -y
```

`conda create`: Este comando inicia el proceso de creación de un nuevo ambiente.

`--name seqtk_env`: Asigna el nombre seqtk_env al nuevo ambiente. Se puede cambiar el nombre.

`seqtk`: Es el nombre del programa que queremos instalar. Conda buscará el paquete seqtk en sus canales.

`-c bioconda`: Le indica a Conda que busque el paquete en el canal de Bioconda, que es el repositorio principal para herramientas de bioinformática.

`-y`: Responde "sí" automáticamente a la pregunta de confirmación, lo que hace que el proceso sea más rápido y no interactivo.

`seqtk` es una herramienta de línea de comandos de propósito general para el procesamiento de secuencias en formato FASTA/Q. Fue desarrollado por Heng Li y es conocido por ser muy rápido y eficiente, lo que lo hace ideal para tareas como:

- Convertir archivos de FASTA a FASTQ y viceversa.
- Muestrear (submuestrear) secuencias aleatoriamente.
- Filtrar secuencias por longitud.

Es una herramienta fundamental en muchos flujos de trabajo de bioinformática para la manipulación de datos de secuenciación de alto rendimiento.

- Una vez que el comando anterior termine de ejecutarse, el ambiente estará creado. Para empezar a usar `seqtk`, necesitas activar el ambiente:

```
conda activate seqtk_env
```

- Cuando termines de usar seqtk, puedes desactivar el ambiente para volver a tu entorno base:

```
conda deactivate
```

Chapter 7

Recapitulación

Accede a este formulario de google y responde las preguntas

https://docs.google.com/forms/d/e/1FAIpQLSfAODGqJAMaTys2w5RmZg-KX3xWKG3t4d0ufhxlpIUya9C7nw/viewform?usp=sharing&ouid=111135442256323215026

El archivo que necesitas para contestar se encuentra en el siguiente enlace:

https://drive.google.com/file/d/1L3qymmU0bouCH3GjCEWcTS4lXp3sn0aX/view?usp=drive_link

Chapter 8

En construcción también