Práctico 3

Dr Andrés Iriarte

Manejo de textos: sed, awk, uniq, grep, sort y otros

En este práctico nos centraremos en utilizar herramientas para la edición de textos, aunque indirectamente repasaremos muchos de lo aprendido en las clases anteriores. Mediante dos ejercicios distintos, con objetivos similares, aprenderemos las técnicas y estrategias comunes para el procesamiento de textos.

EJERCICIO 1: Extracción de secuencias utilizando comandos de edición de textos

Utilizando comandos básicos del Shell y mediante un string (un conjunto de caracteres, generalmente texto) el estudiante deberá obtener una secuencia codificante determinada. El string se utiliza, en este caso, para buscar en la anotación asignada a cada secuencia codificante en el encabezado de cada una.

Vea abajo el string "enolase", donde coincide en el encabezado: > >lcl|NC_000964.3_cds_NP_389242.1_1382 [gene=mtnW] [locus_tag=BSU13590] [db_xref=GeneID:939339] > [protein=2,3-diketo-5-methylthiopentyl-1-phosphate enolase] [protein_id=NP_389242.1] [location=1427034..1428278] > ATGGATGAAAATGAAAG-GAGCTCTGGCATGAGTGAGTTATTAGCGACATATCTCCTGACCGAACCGGGAGCCGATACAGA > GAAGAAAGCAGAACAAATCGCAACAGGATTGACAGTAGGCTCCTGGACTGATCTGCCC-CTTGTAAAACAGGAGCAAATGC > AAAAGCACAAGGGGCGGGTGATAAAAGTTGAGGA-GAGAGAGGGAACTGCTGCATCAGAAAAACAAGCGGTCATCACAATT > GCCTATCCTGAAAT-CAATTTCTCTCAGGATATTCCGGCTTTGCTGACAACAGTGTTTGGCAAGCTGTCCCTGGACGGAAA > AATCAAATTAATCGATCTTCACTTCTCTGAAGCATTTAAGCGTGCACTGCCGGGACC-GAAGTTTGGCGTATACGGCATCC > GGAAGCTGCTGGGAGAGTTTGAGAGACCGCTGT-TAATGAGCATATTTAAAGGCGTAATCGGAAGGGACCTGAGTGATATT > AAAGAACAGCTCCG-GCAGCAGGCGCTTGGCGGAGTTGACTTGATTAAAGACGATGAAATTTTCTTTGAGACTGGTCTAGC > GCCTTTTGAAACAAGAATTGCAGAAGGAAAGCAAATATTGAAAGAAACGTATGAACAAAC-CGGACATAAAACGCTGTATG > CGGTGAATTTGACCGGACGTACGGCTGATCTGAAAGA-CAAAGCGAGACGCGCCGCCGAATTGGGAGCGGATGCGCTTTTG > TTTAATGTCTTCGCTTACG-GCTTGGACGTTATGCAAGGCCTTGCAGAAGATCCTGAAATCCCAGTTCCGATTATGGCGCA > TCCAGCAGTGAGCGGAGCGTTTACGTCTTCTCCGTTTTACGGTTTTTCTCACGCTCTTT-TACTCGGAAAATTAAACCGAT > ATTGCGGTGCTGACTTCAGCCTCTTTCCGTCTCCATATG-GTTCGGTTGCGCTTCCAAGAGCAGATGCACTGGCGATTCAT > GAAGAATGTGTGAGAGAG-GATGCTTTTAACCAAACATTTGCTGTTCCGTCAGCAGGCATTCATCCCGGCATGGTTCCGCT > ATTAATGCGTGATTTCGGCATAGACCACATTATTAACGCCGGAGGAGGCGTACATGGA-CATCCGAACGGTGCCCAAGGCG > GGGGGCGAGCGTTCAGAGCTATTATTGATGCGGTCC-TAGAGGCCCAGCCGATTGATGAAAAAGCCGAACAATGCAAGGAT > CTTAAGCTTGCGCTA-GATAAATGGGGAAAGGCTGAAGCCGTATGA

1) Obtenemos los datos crudos y analicemos algunas de sus propiedades

En la carpeta PRACT3 dentro de MATERIALES encontraremos archivos multifasta que contienen todas las secuencias codificantes de ciertos organismos procariotas. Algunos de estos han sido intensamente estudiados y otros no tanto, esto puede ser relevante para una discusión posterior.

Copie los archivos a su carpeta personal, dentro de una carpeta "practico3" y trabaje allí adentro.

1. Enlistar los archivos y mediante la interacción con otro comando generar una tabla similar a la que se observa a continuación: > GCF_000005845.2 ASM584v2 > GCF_000007625.1 ASM762v1 > GCF_000009045.1 ASM904v1 > GCF_000498355.1 Ade.TY > GCF_001484935.1 ASM148493v1

   Nota: ls + awk -F'' *'{print $1""*$2...}', notar que F'' *indica que el separador de columnas es el "".*

```
{bash, echo = T, eval = F} ls *.fna | awk -F'_' '{print $1"_"$2,$3}'
```

2. Contar cuantos genes (o secuencias codificantes) fueron predichos en cada uno de los ensamblajes

   Ensamblado N° de CDS GCF_000005845.2 ..... GCF_000007625.1 ..... GCF_000009045.1 ..... GCF_000498355.1 ..... GCF_001484935.1 .....

   Nota: utilizar el comando egrep con opción específica para conteo "'{bash, echo = T, eval = F} egrep -c ">" archive.fas

```
3. Estimar el contenidos de bases:
a. Citosina (C) en el transcriptoma:

> Ensamblado C
> GCF_000005845.2 .................
> GCF_000007625.1 .................
> GCF_000009045.1 .................
> GCF_000498355.1 .................
> GCF_001484935.1 .................

> Nota: utilizar el comando egrep para eliminar los encabezados, el comando sed para cambiar la C por u

```{bash, echo = T, eval = F}
egrep -v ">" archive.fna | sed 's/C/\n&/g' | egrep -c ^[C]
```

   b. Porcentaje de GC (G o C) en el transcriptoma:

   Ensamblado GC% GCF_000005845.2 ..... GCF_000007625.1 ..... GCF_000009045.1 ..... GCF_000498355.1 ..... GCF_001484935.1 .....

   Nota: utilizar una aproximación similar al caso anterior para contar G y C y luego el comando wc para contar el número de caracteres (para obtener el número de bases totales y calcular el porcentaje).

```
{bash, echo = T, eval = F} egrep -v ">" archive.fna | sed 's/[CGAT]/\n&/g' | egrep -c
^[CG]
```

2) Preparamos el archivo multifasta solucionando un problema de falta de estandarización

En este punto prepararemos el archivo fasta para hacer que la búsqueda sea posible. La estrategia planeada debería incluir el uso de egrep con la opción "–A" (Busque para que sirve esta opción!!) y el egrep sería el comando responsable de buscar "enolase". El problema de estandarización de los archivos fasta es que las secuencias, propiamente dichas pueden ubicarse en una única línea o en varias. Si en todos los casos estuviera ubicado en una única línea de largo diverso yo podría utilizar el comando egrep para obtener la secuencia.

¿Por qué le parece usted que esto es un problema, en términos generales y en caso particular de nuestro problema?
```

Para transformar el archivo multifasta y dejar todas las secuencias en una única línea:

```
{bash, echo = T, eval = F} sed 's/>..*/&____/g' ENTRADA | sed ':a;N;$!ba;s/\n//g' |sed
's/____/\n/g' | sed 's/>/\n>/g' | egrep .> SALIDA
```

- Trate de explicar para qué sirve cada sección o proceso dentro de este pipeline. Puede y debe utilizar Google! Una vez que tiene todos los archivos con el formato ajustado. Busque la enolasa en todos los organismos.
- ¿En cuántos organismos esta el gen identificado?
- ¿Qué puede ocurrir con aquellos organismos en los que la secuencia no se encuentra?
- ¿Puede garantizar de alguna manera que la secuencia obtenida es de una enolasa? Verifíquelo.
- ¿Qué posibles inconvenientes o puntos negativos cree usted que tiene esta estrategia a nivel de robustez o confiabilidad de los resultados y a nivel técnico?

EJERCICIO 2: Identificación y extracción de secuencias utilizando el paquete Blast

El problema de la aproximación aplicada en el EJERCICIO 1 es que es absolutamente dependiente de la anotación del genoma. Si bien muchos organismos tienen una anotación confiable, porque han sido muy estudiados, esto no es para nada una regla general. Una aproximación basada en similitud de secuencias a nivel de proteínas sería lo correcto.

La opción más común para realizar una búsqueda de secuencias por similitud es el blast. Los pasos según esta estrategia sería: i) construir base de datos, ii) buscar por identidad, iii) parsear archivo de salida del blast y iv) recuperar las secuencias desde la base de datos.

1) Instalar Blast. Busque en la web como puede instalarse el paquete del blast y en caso de ser necesario pídale a su amable administrador que lo instale.

```
{bash, echo = T, eval = F} sudo apt install ncbi-blast+
```

2) Construir base de datos de las proteínas codificantes en el genoma

Vamos a construir una base de datos de blast utilizando todos de los genomas (archivo de secuencias codificantes) disponibles en el directorio del práctico 3 en MATERIALES. Si no lo hizo aun, copie los archivos de la carpeta MATERIAL/PRACT3 a su carpeta personal. Utilizamos cat para concatenar las secuencias codificantes y con la herramienta transeq (Emboss) es posible traducir las secuencias codificantes en un archivo de nucleótidos en fasta y otros formatos de secuencias.

```
{bash, echo = T, eval = F} cat *.fna > todos.fna transeq todos.fna todos.faa
```

Uno de los primeros pasos habituales para utilizar el blast local es construir una base de datos (binaria) con formato específico. La base de datos se construye a partir de un archivo de secuencias multifasta utilizando el comando makeblastdb.

```
{bash, echo = T, eval = F} makeblastdb -in secuencia -dbtype prot -out base -parse_seqids
```

3) Búsqueda por similitud

La búsqueda por identidad se hace con el programa blast. Este programa se basa en el alineamiento local y en la extensión de patrones coincidentes (strings) iniciales. Como parte del resultado el programa brinda estadísticos de significancia de la identidad (porcentaje de identidad, Score, cobertura e e-value). Existen varias opciones para ajustar y permite entre otras cosas cambiar el formato de salida. En nuestro caso vamos a hacer un blastp sin una identidad mínima, relativamente flexible, e-value =e-5.

Query: Antes de hacer el blast defina que busca, ej.: enolasa, tuf, peptidasa, polimerasa. . . , una vez que defina lo que busca vaya al Genbank, a la base de datos de proteínas (ej. www.ncbi.nlm.nih.gov/protein/?term=(enolase)+AND+%22E
y copie una secuencia de referencia en formato fasta. Utilizando el editor joe puede abrir un archivo vacio, y luego con Shift+Insert (click derecho en la terminal) lo pega. Luego utiliza Ctr+KX para guardar el archivo

```
{bash, echo = T, eval = F} joe SequenciasQuery.fas
```

Nota: Pegue la secuencia fasta obtenida del GenBank al documento SequenciasQuery.fas

Blast:

```
{bash, echo = T, eval = F} blastp -db base -query SequenciasQuery.fas -out GCF_Q.bl
-evalue 1e-5 -num_threads 1 -max_target_seqs 20 -outfmt '7 std qcovs'
```

Como mínimo debe establecerse: el nombre de la base (-db), el query(-query) o secuencia de referencia y la salida (-out). En esta caso también establecemos el número máximo de targets identificados (-max_target_seqs), el formato de salida (-outfmt), el número de procesadores (-num_threads) y e-value (-evalue).

Visualice el resultado e interprete el significado de cada una de las columnas. Explique.

```
{bash, echo = T, eval = F} less GCF_Q.bl
```

4) Procesar el archivo de salida del blast. Utilice el los comandos egrep y awk para seleccionar aquellos hits que cumplan con poseer un 100% de cobertura.

```
{bash, echo = T, eval = F} egrep -v ^# GCF_Q.bl | awk '$13 >=100'
```

¿Puede filtrar con más de un criterio a la vez? Piense en distintas estrategias para resolver este problema.

5) Extraer una secuencia Utilizaremos el comando blastdbcmd para recuperar las secuencias identificadas como similares en nuestra base de datos (con identificadores enlistados en el archivo lista). La siguiente línea puede utilizarse para extraer de a una secuencia por vez:

```
{bash, echo = T, eval = F} blastdbcmd -entry NC_000913.3_cds_NP_417259.1_2742_1 -db
todos.base -out salida.fas
```

PARA HACER EN EL DOMICILIO

6) Utilice el los comandos egrep y awk para seleccionar aquellos hits que cumplan con los siguientes requisitos:

- 62% de identidad y >98% de cobertura
- 50% de identidad y >90% de cobertura

7) Guarde en un archivo los nombres de las secuencias que cumplen con alguno de los criterios del Ejercicio 6. Viendo el manual de blastdbcmd, utilice esta lista para extraer estas secuencias.

8) Utilizar la secuencia para hacer un árbol filogenético. Finalmente utilizaremos muscle y fasttree para reconstruir el árbol filogenético.