

Bitacora prueba blastn *Synechococcus* sp. PCC 7002

Elaborado por: Dra. Edith Elizondo Reyna

Como parte de la estancia posdoctoral en el Departamento de Acuicultura bajo la dirección del Dr. Miguel Ángel del Río Portilla, CICESE, Ensenada. Período 2020-2021

Analisis de secuencias de nucleotidos en cianobacteria

```
In [ ]: ls /home/elizondo/data/synechococcus/
```

Descomprimir archivo .gz (Gzip)

```
In [ ]: !gunzip 9_Synechococcus_s6assembly.fasta.gz
```

Cargar la biblioteca de pandas para analisis de datos como pd

Se requiere visualizar los datos, para ello se utiliza el manejo de bases de datos que tiene Python

se cargan la paquetería correspondientes

```
In [ ]: from pandas import Series, DataFrame
import pandas as pd
from Bio import SeqIO, AlignIO, SeqRecord
```

```
In [ ]: from pandas import Series, DataFrame
import pandas as pd
```

Busqueda del archivo de trucha a analizar

```
In [ ]: ls /home/elizondo/data/synechococcus/
```

Comando que busca las bases de datos de Blastn

```
In [ ]: ls -lh /LUSTRE/bioinformatica_data/BD/blast/db/NT/
```

Comando que busca la version del blas instalado en la maquina

```
In [ ]: ls /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin/bl
astn*
```

```
In [ ]: ls /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin/
```

Comando que nos muestra el archivo a analizar que es equivalente a ls /home/elizondo/data/chrysogaster/

```
In [ ]: ls ~/data/synechococcus/
```

Comando para ubicacion del archivo a analizar

```
In [ ]: cd ~/data/synechococcus/
```

Nos muestra las primeras secuencias que se encuentran en el archivo a analizar

```
In [ ]: !head 9_Synechococcus_s6assembly.fasta
```

Comando que muestra el total de contigs en el archivo a analizar que es equivalente a `!grep ">" T1_Trinity.txt | wc -l`

```
In [ ]: !grep -c ">"  
~/data/synechococcus/9_Synechococcus_s6assembly.fasta
```

```
In [ ]: Describir el siguiente comando
```

```
%%bash` Ejecutar comando desde terminal  
  
`export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db  
/NT ` Exportar la base de datos del blastn  
  
`cd ~/data/chrysogaster/` Dirigirse al directorio donde  
se encuentra el archivo a analizar  
  
`date > tiempo_blastn.tx` Nos muestra el tiempo que dur  
a el analisis  
  
`time /LUSTRE/apps/bioinformatica/ncbi-blast-2.6.0/bin/  
blastn ` Busca el programa de blast instalado en la co  
mputadora  
  
`-query Ochrysogaster_map_contiglist2reads.fasta ` Bus  
ca el archivo a analizar
```

```
`-db /LUSTRE/bioinformatica_data/BD/blast/db/NT/nt ` B
usca la base de datos del blastn

`-out Ochrysogaster_map_contiglist2.tsv ` Nombra el nu
evo archivo que contendra el analisis completo. Y el .t
sv nos arroja un archivo separado por tabuladores.

`-evaluate 1E-6 `` Valor de algoritmo Best-Hits

`-max_target_seqs 1 `` Numero de coincidencias de secue
ncias homologas.

`-num_threads 2 `` Numero de subprocesos utilizados e
n en CPU

`-outfmt "6 std sskingdoms stitle staxids sscinames sco
mnames sbblastnames strand" `` Muestra las caracteristic
as de cada alineacion: Reino, Titulo por el blast, Taxo
n, Nombre cientifico, nombre comun, nombre del blast, l
ongitud de secuencia.

`date >> tiempo_blastn.tx` Muestra el tiempo de lo que
tardo el proceso.
```

Comandos de corrida dentro del jupyter

```
In [ ]: %%bash
export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db/NT/
cd ~/data/synechococcus/
date > tiempo_blastn.txt
time /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin/blastn \
  -query 9_Synechococcus_s6assembly.fasta \
  -db /LUSTRE/bioinformatica_data/BD/blast/db/NT/nt \
  -out 9_Synechococcus_s6assembly_blastn.tsv \
  -evalue 1E-6 \
  -max_target_seqs 1 \
  -num_threads 24 \
  -outfmt "6 std sskingdoms stitle staxids sscinames sco
mnames sbblastnames strand"

date >> tiempo_blastn.txt
```

```
In [ ]: ls -lh
```

Se verifica el tiempo del blast

```
In [ ]: !head tiempo_blastn.txt
```

Se verifica el archivo de salida del Blast

```
In [ ]: !head -2 9_Synechococcus_s6assembly_blastn.tsv
!tail -2 9_Synechococcus_s6assembly_blastn.tsv
```

Cargando la base de datos spid_GO falta....

```
In [ ]: ls /LUSTRE/bioinformatica_data/lga/bigdata/spid_go.csv
```

```
In [ ]: ls /LUSTRE/bioinformatica_data/lga/bigdata/
```

Se inicia el analisis de datos del blastx

Organizar los datos en tabla, el encabezado contiene la definición de cada columna de los resultados de blast. Se tomó la información del manual del Blast (<https://www.ncbi.nlm.nih.gov/books/NBK279690/>).

```
encabezado =("qseqid", "sseqid", "pident", "length", "mismatch", "gapo pen", "qstart",
"qend", "sstart", "send", "evaluate", "bitscore", "sskingdoms", "stitle", "staxids",
"sscinames", "scomnames", "sblastnames")
```

encabezado =

```
"qseqid" Nombre de cada contig del archivo original
"sseqid" Sujeto de la secuencia contig
"pident" Porcentaje de coincidencias identicas
"length" Longitud de alineamiento
"mismatch" Cantidad de nucleotidos que no coinciden en la secuencia
"gapo pen" Numero de aperturas o espacios
"qstart" Inicio de la alineacion de la secuencia de analisis (contigs)
"qend" Final de la alineacion de la secuencia de analisis (contigs)
"sstart" Inicio de la identidad de alineacion de la secuencia homologa
"send" Final de la identidad de alineacion de la secuencia homologa
"evaluate" Valor de identidad de homologia
"bitscore" Puntuacion de bit u homologia
"sskingdoms" Reino de la secuencia homologa
"stitle" Nombre de la proteina homologa
"staxids" Taxonomia de la secuencia homologa separados por orden numerico
"sscinames" Nombre cientifico de la secuencia homologa
"scomames" Nombre comun de la secuencia homologa
"sblastnames" Nombre otorgado por blast a la secuencia homologa
```

Comando para buscar la tabla anterior con el encabezado

```
In [ ]: encabezado = ("qseqid", "sseqid", "pident", "length", "mismatch", "gapopen", "qstart", "qend", "sstart", "send", "evaluate", "bitscore", "sskingdoms", "stitle", "staxids", "sscinames", "scomnames", "sblastnames")
```

```
In [ ]: ftsv=pd.read_csv("9_Synechococcus_s6assembly_blastn.tsv", sep = "\t", header=None , names= encabezado, engine="python")
ftsv.head()
```

Guardando los datos en formato csv

```
In [ ]: ftsv.to_csv("9_Synechococcus_s6assembly_blastn.csv", header=True, index= None)
```

Se verifica el principal contenido del archivo generado

```
In [ ]: !head 9_Synechococcus_s6assembly_blastn.csv
```

En caso de recuperar el archivo

```
In [ ]: ftsv= pd.read_csv("9_Synechococcus_s6assembly_blastn.csv")
ftsv.head(2)
```

Nos muestra en tabla los Reinos de la clasificacion de cada secuencia y el total

Es necesario que se haya cargado la biblioteca DataFrame

```
In [ ]: ftab1= ftsv.groupby("sskingdoms")["qseqid"].count()  
ftab1 = DataFrame(ftab1)  
ftab1
```

Nos despliega una tabla con los Reinos, el nombre que tiene en el Blast y la cantidad de cada clasificacion

```
In [ ]: ftab2= ftsv.groupby(["sskingdoms", "sblastnames"])[ "qseq  
id"].count()  
ftab2 = DataFrame(ftab2)  
ftab2
```

Tabla del Reino Bacteria

```
In [ ]: ftab2= ftsv.groupby(["sskingdoms"]).get_group('Bacteria  
)  
ftab2
```

Guardamos un archivo nuevo con el Reino Bacteria de la tabla anterior ftab2

```
In [ ]: ftab2.to_csv("9_Synechococcus_s6assembly_blastn_bacteri  
as.csv", header=True, index=None)
```


Se verifica el archivo nuevo en el directorio

```
In [ ]: ls ~/data/synechococcus/
```

Se verifica el contenido del archivo nuevo

```
In [ ]: !head 9_Synechococcus_s6assembly_blastn_bacterias.csv
```

Obtener una grafica de barras de los sbblastnames del archivo nuevo bacterias, con eje x frecuencia y eje y sbblastnames

poner ftsv. si así lo guardaste o ftab.

```
In [ ]: ftab2.plot(kind='barh', figsize= (8,6))
plt.axis([-1, int(max(ftab2)+5), -1, ftab2.count()], label=None)
plt.legend().set_visible(False)
plt.xlabel("Frecuencia")
plt.ylabel("sbblastnames")
plt.title("Bacterias encontradas blastn")
yes = input("save figure? (y/) ")
if yes.lower()=="y":
    archivo = f[:f.find("9_Synechococcus_s6assembly_bla
stn_bacterias.csv")]+'bacterias.png'
    plt.savefig(archivo, dpi=400, bbox_inches='tight')
plt.show()
```

Obtener una grafica de barras de los skingdoms del archivo del blastx, donde en el eje "x" se muestra la frecuencia y en el eje "y" skingdoms

```
In [ ]: ftab1.plot(kind='barh', figsize=(8,6))
plt.axis([-1, int(max(ftab1)+5), -1, ftab1.count()], label=None)
plt.legend().set_visible(False)
plt.xlabel("Frecuencia")
plt.ylabel("skindoms")
plt.title("Reinos blastx")
yes = input("save figure? (y/) ")
if yes.lower()=="y":
    archivo = f[:f.find("9_Synechococcus_s6assembly_blastn.csv")]+'bacterias.png'
    plt.savefig(archivo, dpi=400, bbox_inches='tight')
plt.show()
```

Tabla del Reino Eucariota

```
In [ ]: ftab2= ftsv.groupby(["sskingdoms"]).get_group('Eukaryota')
ftab2
```

Analizando Eucariotas

```
In [ ]: f_eucaria= ftsv.groupby(["sskingdoms"]).get_group('Eukaryota')
f_eucaria.head()
```

Nombrar nuevo archivo para analisis del Reino Eucariota

```
In [ ]: f_eucaria.to_csv("Synechococcus_eucariota.csv", header=True, index=None)
```

Se verifica el archivo creado

```
In [ ]: ls ~/data/synechococcus/
```

Comando que muestra todos los nombres del Blast, como su cantidad, exclusivos de Reino Eucariota de los datos analizados

```
In [ ]: f_eucaria= f_eucaria.groupby(["sblastnames"])[ "qseqid" ]  
.count()  
f_eucaria.sort_values(axis = 0, ascending=False, inplace=True)  
#ftab3 = DataFrame(ftab3)  
f_eucaria
```

Grafica de barras con los blastnames del Reino Eucariotaduda

```
In [ ]: f_eucaria.plot(kind='barh', figsize= (8,6))  
plt.axis([-1, int(max(f_eucaria)+5), -1, f_eucaria3.count()], label=None)  
plt.legend().set_visible(False)  
plt.xlabel("Frecuencia")  
plt.ylabel("sblastnames")  
plt.title("Eucariotas blastn")  
yes = input("save figure (y/? )")  
if yes.lower()=="y":  
    archivo = f[:f.find("Synechococcus_eucariota.csv")] + 'eucaria.png'  
    plt.savefig(archivo, dpi=400, bbox_inches='tight')  
plt.show()
```