Bitacora para el manejo de secuencias ensambladas y búsqueda con *Blastn* para microalgas *Chaetoceros*

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

Para el siguiente ejercicio es necesario tener el Blast+ instalado en la computadora

https://www.ncbi.nlm.nih.gov/guide/data-software/ (https://www.ncbi.nlm.nih.gov/guide/data-software/)

Se utilizarán los contigs formados por el ensamblaje obtenido en la bitacora anterior SOAP

"

In [ ]:
```
import os
from pandas import Series, DataFrame
import pandas as pd
from Bio import SeqIO, AlignIO, SeqRecord
from Bio.SeqRecord import SeqRecord
from Bio.Seq import Seq
import matplotlib.pyplot as plt
```

In [ ]:
```
cd /LUSTRE/bioinformatica_data/lga/edith/data/microalgas/
```

In [ ]:
```
os.makedirs('img',exist_ok=True)
```

In [ ]:
```
ls
```

Se analizarán con blastn los contigs obtenidos a la base de datos *nt*

Verifique la localización de la base de datos, en este caso se encuentra en `~/DATA/nt/` o corrija si es necesario

In [ ]:
```
!grep ">" CA2/CA2_out.contig |wc -l
!grep ">" CN2/CN2_out.contig |wc -l
!grep ">" MA2/MA2_out.contig |wc -l
!grep ">" MN2/MN2_out.contig |wc -l
!grep ">" XA2/XA2_out.contig |wc -l
!grep ">" XN2/XN2_out.contig |wc -l
```

In [ ]:
```
!grep -c ">" CA2/CA2.fasta
!grep -c ">" CN2/CN2.fasta
!grep -c ">" MA2/MA2.fasta
!grep -c ">" MN2/MN2.fasta
!grep -c ">" XA2/XA2.fasta
!grep -c ">" XN2/XN2.fasta
```

In [ ]:
```
pwd
```

In [ ]:
```
!find CA2/CA2_out.contig
```

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

Ejecuta el blastn en slurum

In [ ]:
```
fout = open("blastn_CN2.sh", "w")
linea="""#!/bin/sh

#
#SBATCH -p cicese
#SBATCH --job-name=blastn
#SBATCH -e blastn.%N.%j.err

#SBATCH -o blastn.%N.%j.log
#SBATCH -t 6-00:00:00
#
#SBATCH -N 1
#SBATCH -n 24
#
#SBATCH --exclusive

cd $SLURM_SUBMIT_DIR
#

shell=`/bin/basename \`/bin/ps -p $$ -ocomm=\``
if [ -f /usr/share/Modules/init/$shell ]
then
  . /usr/share/Modules/init/$shell
else
  . /usr/share/Modules/init/sh
fi

module load gcc-7.2
export LD_LIBRARY_PATH=/LUSTRE/apps/bioinformatica/ncbi
-blast-2.11.0/lib:$LD_LIBRARY_PATH
export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db/
NT

#
cd /LUSTRE/bioinformatica_data/lga/edith/data/microalga
s/CN2
date > tiempoCN2_blastn.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin
```

```
/blastn \\
 -query CN2.fasta \\
 -db /LUSTRE/bioinformatica_data/BD/blast/db/NT/nt \\
 -out CN2_blastn.tsv \\
 -evalue 1E-6 \\
 -max_target_seqs 1 \\
 -num_threads 24 \\
 -outfmt "6 std sskingdoms stitle staxids sscinames sco
mnames sblastnames strand"
date >> tiempoCN2_blastn.txt

head CN2_blastn.tsv
echo ""
grep -c CN2_blastn.tsv

"""
fout.write(linea)
fout.close()
```

In [ ]:
```
!head -100 blastn_CN2.sh
```

Mandar cola de trabajo en el servidor

In [ ]:
```
!sbatch blastn_CN2.sh
```

In [ ]:
```
!head tiempoCN2_blastn.txt
```

Comando que verifica el nodulo donde esta llevandose a cabo el proceso en slurum

In [ ]:
```
!squeue
```

In [ ]:
```
fout = open("blastn_CA2.sh", "w")
linea="""#!/bin/sh

#
```

```
#SBATCH -p cicese
#SBATCH --job-name=blastn
#SBATCH -e blastn.%N.%j.err

#SBATCH -o blastn.%N.%j.log
#SBATCH -t 6-00:00:00
#
#SBATCH -N 1
#SBATCH -n 24
#
#SBATCH --exclusive

cd $SLURM_SUBMIT_DIR
#

shell=`/bin/basename \`/bin/ps -p $$ -ocomm=\``
if [ -f /usr/share/Modules/init/$shell ]
then
  . /usr/share/Modules/init/$shell
else
  . /usr/share/Modules/init/sh
fi

module load gcc-7.2
export LD_LIBRARY_PATH=/LUSTRE/apps/bioinformatica/ncbi
-blast-2.11.0/lib:$LD_LIBRARY_PATH
export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db/
NT

#
cd /LUSTRE/bioinformatica_data/lga/edith/data/microalga
s/CA2
date > tiempoCA2_blastn.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin
/blastn \\
 -query CA2.fasta \\
 -db /LUSTRE/bioinformatica_data/BD/blast/db/NT/nt \\
 -out CA2_blastn.tsv \\
 -evalue 1E-6 \\
 -max_target_seqs 1 \\
 -num_threads 24 \\
 -outfmt "6 std sskingdoms stitle staxids sscinames sco
```

```
mnames sblastnames strand"
date >> tiempoCA2_blastn.txt

head CA2_blastn.tsv
echo ""
grep -c CA2_blastn.tsv

"""
fout.write(linea)
fout.close()
```

Mandar a la cola de trabajo del servidor

In [ ]:
```
!sbatch blastn_CA2.sh
```

In [ ]:
```
fout = open("blastn_MA2.sh", "w")
linea="""#!/bin/sh

#
#SBATCH -p cicese
#SBATCH --job-name=blastn
#SBATCH -e blastn.%N.%j.err

#SBATCH -o blastn.%N.%j.log
#SBATCH -t 6-00:00:00
#
#SBATCH -N 1
#SBATCH -n 24
#
#SBATCH --exclusive

cd $SLURM_SUBMIT_DIR
#

shell=`/bin/basename \`/bin/ps -p $$ -ocomm=\``
if [ -f /usr/share/Modules/init/$shell ]
then
  . /usr/share/Modules/init/$shell
else
  . /usr/share/Modules/init/sh
```

```
fi

module load gcc-7.2
export LD_LIBRARY_PATH=/LUSTRE/apps/bioinformatica/ncbi
-blast-2.11.0/lib:$LD_LIBRARY_PATH
export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db/
NT

#
cd /LUSTRE/bioinformatica_data/lga/edith/data/microalga
s/MA2
date > tiempoMA2_blastn.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin
/blastn \\
 -query MA2.fasta \\
 -db /LUSTRE/bioinformatica_data/BD/blast/db/NT/nt \\
 -out MA2_blastn.tsv \\
 -evalue 1E-6 \\
 -max_target_seqs 1 \\
 -num_threads 24 \\
 -outfmt "6 std sskingdoms stitle staxids sscinames sco
mnames sblastnames strand"
date >> tiempoMA2_blastn.txt

head MA2_blastn.tsv
echo ""
grep -c MA2_blastn.tsv

"""
fout.write(linea)
fout.close()
```

In [ ]:
```
!sbatch blastn_MA2.sh
```

In [ ]:
```
fout = open("blastn_MN2.sh", "w")
linea="""#!/bin/sh

#
#SBATCH -p cicese
#SBATCH --job-name=blastn
#SBATCH -e blastn.%N.%j.err
```

```
#SBATCH -o blastn.%N.%j.log
#SBATCH -t 6-00:00:00
#
#SBATCH -N 1
#SBATCH -n 24
#
#SBATCH --exclusive

cd $SLURM_SUBMIT_DIR
#

shell=`/bin/basename \`/bin/ps -p $$ -ocomm=\``
if [ -f /usr/share/Modules/init/$shell ]
then
  . /usr/share/Modules/init/$shell
else
  . /usr/share/Modules/init/sh
fi

module load gcc-7.2
export LD_LIBRARY_PATH=/LUSTRE/apps/bioinformatica/ncbi
-blast-2.11.0/lib:$LD_LIBRARY_PATH
export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db/
NT

#
cd /LUSTRE/bioinformatica_data/lga/edith/data/microalga
s/MN2
date > tiempoMN2_blastn.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin
/blastn \\
 -query MN2.fasta \\
 -db /LUSTRE/bioinformatica_data/BD/blast/db/NT/nt \\
 -out MN2_blastn.tsv \\
 -evalue 1E-6 \\
 -max_target_seqs 1 \\
 -num_threads 24 \\
 -outfmt "6 std sskingdoms stitle staxids sscinames sco
mnames sblastnames strand"
date >> tiempoMN2_blastn.txt

echo ""  >> tiempoMN2_blastn.txt
```

```
head MN2_blastn.tsv >> tiempoMN2_blastn.txt
echo ""  >> tiempoMN2_blastn.txt
grep -c MN2_blastn.tsv >> tiempoMN2_blastn.txt

"""
fout.write(linea)
fout.close()
```

In [ ]:
```
!sbatch blastn_MN2.sh
```

In [ ]:
```
fout = open("blastn_XA2.sh", "w")
linea="""#!/bin/sh

#
#SBATCH -p cicese
#SBATCH --job-name=blastn
#SBATCH -e blastn.%N.%j.err

#SBATCH -o blastn.%N.%j.log
#SBATCH -t 6-00:00:00
#
#SBATCH -N 1
#SBATCH -n 24
#
#SBATCH --exclusive

cd $SLURM_SUBMIT_DIR
#

shell=`/bin/basename \`/bin/ps -p $$ -ocomm=\``
if [ -f /usr/share/Modules/init/$shell ]
then
  . /usr/share/Modules/init/$shell
else
  . /usr/share/Modules/init/sh
fi

module load gcc-7.2
export LD_LIBRARY_PATH=/LUSTRE/apps/bioinformatica/ncbi
-blast-2.11.0/lib:$LD_LIBRARY_PATH
export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db/
NT
```

```
#
cd /LUSTRE/bioinformatica_data/lga/edith/data/microalgas/XA2
date > tiempoXA2_blastn.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin/blastn \\
 -query XA2.fasta \\
 -db /LUSTRE/bioinformatica_data/BD/blast/db/NT/nt \\
 -out XA2_blastn.tsv \\
 -evalue 1E-6 \\
 -max_target_seqs 1 \\
 -num_threads 24 \\
 -outfmt "6 std sskingdoms stitle staxids sscinames scomnames sblastnames strand"
date >> tiempoXA2_blastn.txt

echo ""  >> tiempoMN2_blastn.txt
head MN2_blastn.tsv >> tiempoMN2_blastn.txt
echo ""  >> tiempoMN2_blastn.txt
grep -c MN2_blastn.tsv >> tiempoMN2_blastn.txt

"""
fout.write(linea)
fout.close()
```

In [ ]:
```
!sbatch blastn_XA2.sh
```

In [ ]:
```
fout = open("blastn_XN2.sh", "w")
linea="""#!/bin/sh

#
#SBATCH -p cicese
#SBATCH --job-name=blastn
#SBATCH -e blastn.%N.%j.err

#SBATCH -o blastn.%N.%j.log
#SBATCH -t 6-00:00:00
#
#SBATCH -N 1
#SBATCH -n 24
#
```

```
#SBATCH --exclusive

cd $SLURM_SUBMIT_DIR
#

shell=`/bin/basename \`/bin/ps -p $$ -ocomm=\``
if [ -f /usr/share/Modules/init/$shell ]
then
  . /usr/share/Modules/init/$shell
else
  . /usr/share/Modules/init/sh
fi

module load gcc-7.2
export LD_LIBRARY_PATH=/LUSTRE/apps/bioinformatica/ncbi
-blast-2.11.0/lib:$LD_LIBRARY_PATH
export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db/
NT

#
cd /LUSTRE/bioinformatica_data/lga/edith/data/microalga
s/XN2
date > tiempoXN2_blastn.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin
/blastn \\
 -query XN2.fasta \\
 -db /LUSTRE/bioinformatica_data/BD/blast/db/NT/nt \\
 -out XN2_blastn.tsv \\
 -evalue 1E-6 \\
 -max_target_seqs 1 \\
 -num_threads 24 \\
 -outfmt "6 std sskingdoms stitle staxids sscinames sco
mnames sblastnames strand"
date >> tiempoXN2_blastn.txt

echo ""  >> tiempoMN2_blastn.txt
head MN2_blastn.tsv >> tiempoMN2_blastn.txt
echo ""  >> tiempoMN2_blastn.txt
grep -c MN2_blastn.tsv >> tiempoMN2_blastn.txt
echo ""  >> tiempoMN2_blastn.txt
"""
fout.write(linea)
```

```
fout.close()
```

In [ ]:
```
!sbatch blastn_XN2.sh
```

In [ ]:
```
cd /LUSTRE/bioinformatica_data/lga/edith/data/microalgas/
```

Comando para verificar el contenido de los archivos *.err que se generan como resultado de las corridas

In [ ]:
```
!for f in blastn.*.err; do echo $f; ls -lh $f; head $f; echo "--------------"; done
```

Verificar el archivo de blast XA2 que muestra error en blastn.nodo17.166792.err

In [ ]:
```
!head blastn.nodo17.166792.err
```

In [ ]:
```
!head blastn.nodo17.166792.log
```

Este error corresponde al archivo blastn_XA2.sh por lo que hay que visualizarlo

In [ ]:
```
!head -100 blastn_XA2.sh
```

se visualiza el contenido del archivo de salida XA2_blastn.tsv

In [ ]:
```
%%bash
head XA2/XA2_blastn.tsv
echo "numero de resultados es:"
wc -l XA2/XA2_blastn.tsv
```

In [ ]:
```
ls -lh *.tsv
```

In [ ]:
```
ls -ld *.tsv
```

se visualizan los archivos .tsv que son los que tienen la informacion del blastn

In [ ]:
```
ls */*.tsv
```

se copian los archivos .tsv desde Lustre hasta mi caprteta tsv en mi direccion de omica

In [ ]:
```
%%bash
for f in ls */*.tsv
do
echo $f
cp $f ~/data/microalgas/tsv/
done
```

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

El comando tar es usado para comprimir los archivos de interés que posteriormente serán descargados del directorio y así ser analizados en excel cada resultado de blastn. Y para empaquetar una compresión de alguna carpeta se debe usar el algo así como comprimir archivo/s o carpeta/s, se debe realizar de la siguiente manera:

!tar -czv

In [ ]:
```
!tar -czvf tsv.tar.gz ./tsv
```

In [ ]:
```
cd /LUSTRE/bioinformatica_data/lga/edith/data/microalgas
```

In [ ]:
```
!grep ">" ~/data/microalgas
```

In [ ]:
```
!grep ">" ~/Documents/secuenciacion_masiva/sec2019/microalgas/CN2/CN2_clc/Contig_CN2_T.fasta |wc -l
```

In [ ]:
```
!grep ">" ~/Documents/secuenciacion_masiva/sec2019/microalgas/MA2/MA2_clc//Contig_MA2_T.fasta |wc -l
```

In [ ]:
```
!grep ">" ~/Documents/secuenciacion_masiva/sec2019/microalgas/MN2/MN2_clc/Contig_MN2_T.fasta |wc -l
```

In [ ]:
```
!grep ">" ~/Documents/secuenciacion_masiva/sec2019/microalgas/XA2/XA2_clc/Contig_XA2_T.fasta |wc -l
```

In [ ]:
```
!grep ">" ~/Documents/secuenciacion_masiva/sec2019/microalgas/XN2/XN2_clc/Contig_XN2_T.fasta |wc -l
```

In [ ]:
```
pwd
```

In [ ]:
```
ls
```

In [ ]:
```
!head -2 CA2_blastn.tsv
```

In [ ]:
```
encabezado =("qseqid", "sseqid", "pident", "length", "m
ismatch", "gapopen","qstart",
             "qend", "sstart","send", "evalue", "bitsco
re", "sskingdoms", "stitle",
             "staxids", "sscinames", "scomnames", "sbla
stnames")
```

In [ ]:
```
ftab=pd.read_csv("CA2_blastn.tsv", header=None, sep = "
\t" , names= encabezado)
ftab.head()
```

In [ ]:
```
len(ftab)
```

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

Cuántos contigs no son eucariotas?

Guadardo la base de datos en formato csv

In [ ]:
```
ftab.to_csv("CA2_blastn.csv", header=True, index= None)
```

en caso de haber el hecho el análisis previo y querer recuperar el archivo anterior

In [ ]:
```
ftab= pd.read_csv("contigs_blastn.csv")
ftab.head(2)
```

Hay algún contig con más de una asignación (duplicados)?

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

Es necesario eliminar duplicados

In [ ]:
```
ftab1 =ftab.drop_duplicates(subset = 'qseqid', keep='fi
rst', inplace = False)
ftab1
```

Cuántos grupos hay a parte del que pertenece la especie analizada?

In [ ]:
```
ftab2= ftab1.groupby(["sskingdoms", "sblastnames"])["qs
eqid"].count()
ftab2 = DataFrame(ftab2)
ftab2
```

In [ ]:
```
ftab2= ftab1.groupby(["sskingdoms", "sscinames"])["qseq
id"].count()
ftab2.sort_values(axis = 0, ascending=False, inplace=Tr
ue)
ftab2
```

Procedimiento para simplificar tabla y graficar las 10 primeras categorías y el resto ponerlas en "otras"

In [ ]:
```
linea10=ftab2[:10]
linea11=ftab2[10:]
#linea10
otro=sum (linea11)
#otro
otros = pd.DataFrame({0:otro}, index=["Other"])
otros
linea10=linea10.append(otros)
linea10
```

In [ ]:
```
linea10.plot(kind='barh', figsize= (8,6))
plt.axis([-1, max(linea10[0])+5, -1, 10.8], label=None)
plt.legend().set_visible(False)
plt.xlabel("Frecuencia")
plt.ylabel("Especies")
plt.title("Especies con resultado de blastn")
plt.show()
```