Bitácora de análisis de los datos en Blastn

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

Datos obtenidos por secuenciación masiva de *Synechococcus*

Se seleccionaron los contigs con >= 3 lecturas y una cobertura de >=1.5

Se cargan las funciones que se utilizarán en este proceso

In [46]:
```
from Bio import SeqIO, pairwise2, AlignIO, Phylo, Entrez, SeqRecord, Seq, SearchIO
from Bio.Align.Applications import ClustalwCommandline
from Bio.Blast import NCBIWWW, NCBIXML
from Bio.Seq import Seq
from Bio.SeqUtils import GC
from Bio.SeqRecord import SeqRecord

from matplotlib import *
import matplotlib.pyplot as plt
from matplotlib_venn import venn3_unweighted, venn2_unweighted

import os, pylab

from pandas import DataFrame
import pandas as pd

import pylab as pl
from pylab import *
```

Se definen funciones a utilizar en la bitácora

In [1]:
```
def cpg(secuencia):
    g= secuencia.count("G")
    c= secuencia.count("C")
    cg= secuencia.count("CG")
    lar= len(secuencia)
    cpG=0
    try:
        g*c==0
    except:
        cpG=0
    else:
        if g == 0 or c== 0:
            cpG =0
        else:
            cpG=(round(cg/(g*c)*(lar**2/(lar-1)) ,8))
    return (cpG)

def generoespecie(genesp):
    genero=genesp[:genesp.find(" ")]
    #print(genero)
    especie = genesp[genesp.find(" ")+1:]
    especie = especie[:especie.find(" ")]
    #print(especie)
    genesp1 = genero+" "+especie
    return(genesp1)

def gespecie(genesp):
    genero=genesp[:1]+"._"
    #print(genero)
    especie = genesp[genesp.find(" ")+1:]
    especie = especie[:especie.find(" ")]
    #print(especie)
    genesp1 = genero+especie
    return(genesp1)
```

In [2]:
```
ls /LUSTRE/bioinformatica_data/lga/mdelrio/synechococcus/
```

```
9_Synechococcus_blastn_bacterias.csv     blastn.nodo2.166100.err
9_Synechococcus_blastn.csv               blastn.nodo2.166100.log
9_Synechococcus_blastn_eucaria.csv       blastn.nodo2.166101.err
9_Synechococcus_blastn.tsv               blastn.nodo2.166101.log
9_Synechococcus_s6assemblybacterias.png  blastn_synechococcus.sh
9_Synechococcus_s6assemblyeucaria.png    tiempo_blastn.txt
9_Synechococcus_s6assembly.fasta
```

In [3]:
```
cd /LUSTRE/bioinformatica_data/lga/mdelrio/synechococcus/
```

```
/LUSTRE/bioinformatica_data/lga/mdelrio/synechococcus
```

Número de secuencias en el archivo fasta que contiene todas las secuencias generadas por el ensamblaje *De Novo* de *Fundulus lima*

In [7]:
```
!grep -c "^>" 9_Synechococcus_s6assembly.fasta
```

```
8275
```

Descripción de los contigs por tamaño, contenido de GC y Cpg

In [8]:
```
f = '9_Synechococcus_s6assembly.fasta'
```

In [9]:
```
f1 = f[:f.find(".")]
f1
```

Out[9]:
```
'9_Synechococcus_s6assembly'
```

In [10]:
```
sizes = [(rec.name, len(rec), round(GC(rec.seq),4), cpg
(rec.seq)) for rec in SeqIO.parse(open(f, 'r'), "fasta"
)]
sizes = DataFrame(sizes,columns= ["contigs", "length",
"GC", "CpG"])
sizes.describe().round(2)
```

Out[10]:

	length	GC	CpG
count	8275.00	8275.00	8275.00
mean	1355.65	61.61	1.14
std	10700.71	5.68	0.14
min	51.00	23.66	0.00
25%	357.00	59.53	1.06
50%	514.00	62.40	1.15
75%	734.00	65.03	1.23
max	421304.00	77.90	1.84

In [11]:
```
pl.hist(sizes['length'], bins=20)
pl.title("%i secuencias de %s \nLengths %i to %i" \
            % (len(sizes["length"]), f, min(sizes['leng
th']),max(sizes['length'])))
pl.xlabel("Sequence length (bp)")
pl.ylabel("Count")
#pl.legend().set_visible(False)
yes = input("save figure? ")
if yes.lower()=="y":
    archivo = f1+"_length.png"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')
    archivo = f1+"_length.pdf"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')


pl.show()
```

```
save figure?
```

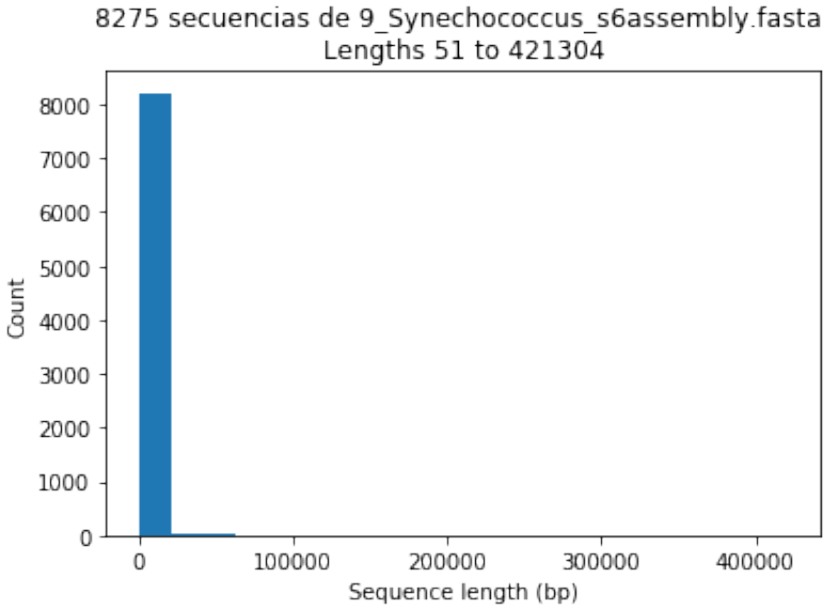


Tabla de distribución de tamaños de los contigs

In [12]:
```
sizes1 = sizes['length'].value_counts(normalize=False,
sort=False, ascending=False,
                                      bins=range(0,sizes['l
ength'].max()+500,1000), dropna=True)
sizes1
```

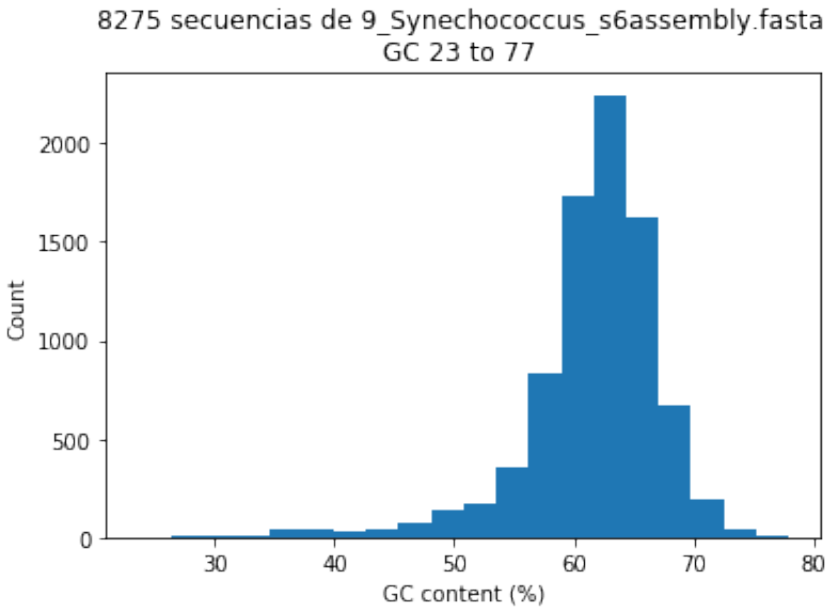
Out[12]:
```
(-0.001, 1000.0]          7308
(1000.0, 2000.0]           795
(2000.0, 3000.0]            57
(3000.0, 4000.0]            14
(4000.0, 5000.0]             2
(5000.0, 6000.0]             5
(6000.0, 7000.0]             1
(7000.0, 8000.0]             1
(8000.0, 9000.0]             4
(9000.0, 10000.0]            1
(10000.0, 11000.0]           2
(11000.0, 12000.0]           1
(12000.0, 13000.0]           1
(13000.0, 14000.0]           2
(14000.0, 15000.0]           4
(15000.0, 16000.0]           1
(16000.0, 17000.0]           3
(17000.0, 18000.0]           2
(18000.0, 19000.0]           3
(19000.0, 20000.0]           1
(20000.0, 21000.0]           3
(21000.0, 22000.0]           1
(22000.0, 23000.0]           0
(23000.0, 24000.0]           0
(24000.0, 25000.0]           1
(25000.0, 26000.0]           1
(26000.0, 27000.0]           1
(27000.0, 28000.0]           2
(28000.0, 29000.0]           2
(29000.0, 30000.0]           1
                          ...
(391000.0, 392000.0]         0
(392000.0, 393000.0]         0
(393000.0, 394000.0]         0
(394000.0, 395000.0]         0
(395000.0, 396000.0]         0
```

```
(396000.0, 397000.0]       0
(397000.0, 398000.0]       0
(398000.0, 399000.0]       0
(399000.0, 400000.0]       0
(400000.0, 401000.0]       0
(401000.0, 402000.0]       0
(402000.0, 403000.0]       0
(403000.0, 404000.0]       0
(404000.0, 405000.0]       0
(405000.0, 406000.0]       0
(406000.0, 407000.0]       0
(407000.0, 408000.0]       0
(408000.0, 409000.0]       0
(409000.0, 410000.0]       0
(410000.0, 411000.0]       0
(411000.0, 412000.0]       0
(412000.0, 413000.0]       0
(413000.0, 414000.0]       0
(414000.0, 415000.0]       0
(415000.0, 416000.0]       0
(416000.0, 417000.0]       0
(417000.0, 418000.0]       0
(418000.0, 419000.0]       0
(419000.0, 420000.0]       0
(420000.0, 421000.0]       0
Name: length, Length: 421, dtype: int64
```

Contenido de GC

In [13]:
```
pl.hist(sizes['GC'], bins=20)
pl.title("%i secuencias de %s \nGC %i to %i " \
            % (len(sizes["GC"]), f, min(sizes['GC']),ma
x(sizes['GC'])))

pl.xlabel("GC content (%)")
pl.ylabel("Count")
#pl.legend().set_visible(False)
yes = input("save figure? ")
if yes.lower()=="y":
    archivo = f1+"_GC.png"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')
    archivo = f1+"_GC.pdf"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')

pl.show()
```

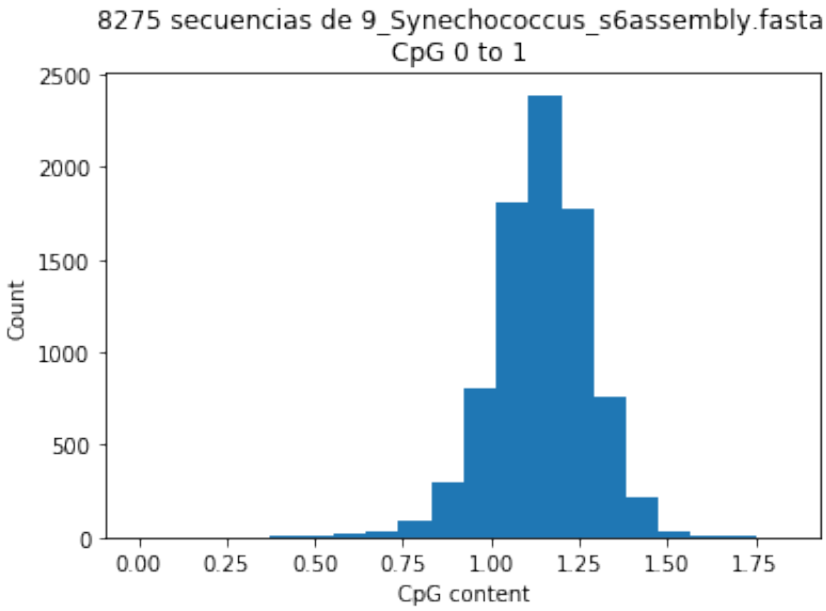
```
save figure?
```



Contenido de CpG

In [14]:
```
pl.hist(sizes['CpG'], bins=20)
pl.title("%i secuencias de %s \nCpG %i to %i " \
            % (len(sizes["CpG"]), f, min(sizes['CpG']),
max(sizes['CpG'])))

pl.xlabel("CpG content")
pl.ylabel("Count")
#pl.legend().set_visible(False)
yes = input("save figure? ")
if yes.lower()=="y":
    archivo = f1+"_CpG.png"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')
    archivo = f1+"_CpG.pdf"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')

pl.show()
```

```
save figure?
```



In [25]:
```
pwd
```

Out[25]:
```
'/LUSTRE/bioinformatica_data/lga/mdelrio/synechococcus
'
```

In [26]:
```
ls *.fasta
```

```
9_Synechococcus_s6assembly.fasta
```

Blastn

Se hace el archivo `.sh` para que no se utilice el nodo maestro de omica en el blast. Esto también libera al Jupyter para seguir haciendo procesos o análisis mientras se ejecuta el blastn.

Se corrio el blastn en la terminal y se obtivieron los resultados, pero se pone la alternativa para enviarlo al slurum

In [ ]:
```
fout = open("blastn_synechococcus.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/mdelrio/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_blastn.tsv \\
 -evalue 1E-6 \\
 -max_target_seqs 1 \\
 -num_threads 24 \\
 -outfmt "6 std sskingdoms stitle staxids sscinames scomnames sblastnames strand"
date >> tiempo_blastn.txt


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

In [9]:
```
## Verificar si las líneas que siguen, sirven para enviar corro electrónico cuando finaliza el proceso.

#SBATCH --mail-type=all
#SBATCH --mail-user=mdelrio@cicese.mx
```

In [4]:
```
ls *.sh
```

```
blastn_synechococcus.sh
```

In [5]:
```
!head -100 blastn_synechococcus.sh
```

```
#!/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
#

export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db
/NT

#
cd /LUSTRE/bioinformatica_data/lga/mdelrio/synechococc
us
date > tiempo_blastn.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bi
n/blastn \
 -query 9_Synechococcus_s6assembly.fasta \
 -db /LUSTRE/bioinformatica_data/BD/blast/db/NT/nt \
 -out 9_Synechococcus_blastn.tsv \
 -evalue 1E-6 \
 -max_target_seqs 1 \
 -num_threads 24 \
 -outfmt "6 std sskingdoms stitle staxids sscinames sc
omnames sblastnames strand"
date >> tiempo_blastn.txt
#SBATCH --mail-type=all
#SBATCH --mail-user=mdelrio@cicese.mx


exit 0
```

Se manda el archvo desde Jupyter a la cola de trabajos

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

Es necesario guardar el trabajo (en este caso `166101`), con ello se puede monitorear si se ha concluido

También se crea el archivo `tiempo_blastn.txt` en donde se guardará la fecha de inicio y de fin

In [19]:
```
ls
```

```
9_Synechococcus_s6assembly.fasta  blastn.nodo2.166100.log
blastn.nodo2.166100.err           blastn_synechococcus.sh
```

comando para verificar que esté corriendo el trabajo en slurum

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

Se revisa `blastn.nodo2.166101.err` y `tiempo_blastn.txt`

In [30]:
```
!head blastn.nodo2.166101.err
```

```
/LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin/blas
tn: error while loading shared libraries: libuv.so.0.1
0: cannot open shared object file: No such file or dir
ectory

real    0m0.001s
user    0m0.001s
sys     0m0.000s
```

In [54]:
```
!head tiempo_blastn.txt
```

```
Fri Dec 11 16:49:43 PST 2020
Fri Dec 11 19:27:02 PST 2020
```

In [39]:
```
!head 9_Synechococcus_blastn.tsv
```

In [52]:
```
ls -lh
```

```
total 13M
-rw-r--r-- 1 mdelrio gen_acuicola 1.5M Dec 11 19:14 9_
Synechococcus_blastn.tsv
-rw-r--r-- 1 mdelrio gen_acuicola  12M Dec  8 10:35 9_
Synechococcus_s6assembly.fasta
-rw-r--r-- 1 mdelrio gen_acuicola  211 Dec 11 16:38 bl
astn.nodo2.166100.err
-rw-r--r-- 1 mdelrio gen_acuicola    0 Dec 11 16:38 bl
astn.nodo2.166100.log
-rw-r--r-- 1 mdelrio gen_acuicola  211 Dec 11 16:42 bl
astn.nodo2.166101.err
-rw-r--r-- 1 mdelrio gen_acuicola    0 Dec 11 16:42 bl
astn.nodo2.166101.log
-rw-r--r-- 1 mdelrio gen_acuicola  787 Dec 11 16:41 bl
astn_synechococcus.sh
-rw-r--r-- 1 mdelrio gen_acuicola   29 Dec 11 16:49 ti
empo_blastn.txt
```

In [ ]:
```
### ya que terminó el proceso se verifica el archivo de salida
```

In [8]:
```
!grep -c ">" 9_Synechococcus_s6assembly.fasta
!grep ">" 9_Synechococcus_s6assembly.fasta|tail
```

```
8275
>9_Synechococcus_s6_contig_8266
>9_Synechococcus_s6_contig_8267
>9_Synechococcus_s6_contig_8268
>9_Synechococcus_s6_contig_8269
>9_Synechococcus_s6_contig_8270
>9_Synechococcus_s6_contig_8271
>9_Synechococcus_s6_contig_8272
>9_Synechococcus_s6_contig_8273
>9_Synechococcus_s6_contig_8274
>9_Synechococcus_s6_contig_8275
```

In [7]:
```
%%bash
for f in 9_Synechococcus_blastn.tsv
do
 head -2 $f
 echo
 tail -2 $f
done
```

```
9_Synechococcus_s6_contig_1     CP016474.1      93.870
24405   1440    27      75934   100295  1572285 154789
4       0.0     36729   Bacteria        Synechococcus
sp. PCC 7003, complete genome   374981  Synechococcus
sp. PCC 7003    Synechococcus sp. PCC 7003
cyanobacteria
9_Synechococcus_s6_contig_1     CP016474.1      95.261
10213   454     10      59769   69952   1589400 157918
9       0.0     16151   Bacteria        Synechococcus
sp. PCC 7003, complete genome   374981  Synechococcus
sp. PCC 7003    Synechococcus sp. PCC 7003
cyanobacteria

9_Synechococcus_s6_contig_8274  CP007202.1      88.667
150     17      0       1       150     1663381 166323
2       1.77e-42        183     Bacteria
Siansivirga zeaxanthinifaciens CC-SAMT-1, complete gen
ome     1454006 Siansivirga zeaxanthinifaciens CC-SAMT
-1      Siansivirga zeaxanthinifaciens CC-SAMT-1
CFB group bacteria
9_Synechococcus_s6_contig_8275  CP000951.1      92.889
225     16      0       1       225     528829  529053
1.17e-85        327     Bacteria        Synechococcus
sp. PCC 7002, complete genome   32049   Synechococcus
sp. PCC 7002    Synechococcus sp. PCC 7002
cyanobacteria
```

Se inicia el análisis de los datos de blastn

In [56]:
```
encabezado =("qseqid", "sseqid", "pident", "length", "m
ismatch", "gapopen", "qstart",
             "qend", "sstart", "send", "evalue", "bitsc
ore", "sskingdoms", "stitle",
             "staxids", "sscinames", "scomnames", "sbla
stnames")
```

In [58]:
```
ftsv=pd.read_csv("9_Synechococcus_blastn.tsv", sep = "\t", header=None , names= encabezado, engine="python")
ftsv.head()
```

Out[58]:

	qseqid	sseqid	pident	length	mismatc
0	9_Synechococcus_s6_contig_1	CP016474.1	93.870	24405	1440
1	9_Synechococcus_s6_contig_1	CP016474.1	95.261	10213	454
2	9_Synechococcus_s6_contig_1	CP016474.1	94.773	5376	260
3	9_Synechococcus_s6_contig_1	CP016474.1	89.007	6568	642
4	9_Synechococcus_s6_contig_1	CP016474.1	94.357	4324	239

In [59]:
```
# Guardando los datos en formato csv
ftsv.to_csv("9_Synechococcus_blastn.csv", header=True, index= None)
```

In [60]:
```
!head "9_Synechococcus_blastn.csv"
```

```
qseqid,sseqid,pident,length,mismatch,gapopen,qstart,qe
nd,sstart,send,evalue,bitscore,sskingdoms,stitle,staxi
```

```
ds,sscinames,scomnames,sblastnames
9_Synechococcus_s6_contig_1,CP016474.1,93.87,24405,144
0,27,75934,100295,1572285,1547894,0.0,36729.0,Bacteria
,"Synechococcus sp. PCC 7003, complete genome",374981,
Synechococcus sp. PCC 7003,Synechococcus sp. PCC 7003,
cyanobacteria
9_Synechococcus_s6_contig_1,CP016474.1,95.261000000000
01,10213,454,10,59769,69952,1589400,1579189,0.0,16151.
0,Bacteria,"Synechococcus sp. PCC 7003, complete genom
e",374981,Synechococcus sp. PCC 7003,Synechococcus sp.
PCC 7003,cyanobacteria
9_Synechococcus_s6_contig_1,CP016474.1,94.773,5376,260
,10,49047,54404,2125402,2120030,0.0,8351.0,Bacteria,"S
ynechococcus sp. PCC 7003, complete genome",374981,Syn
echococcus sp. PCC 7003,Synechococcus sp. PCC 7003,cya
nobacteria
9_Synechococcus_s6_contig_1,CP016474.1,89.007,6568,642
,42,38009,44527,2136286,2129750,0.0,8056.0,Bacteria,"S
ynechococcus sp. PCC 7003, complete genome",374981,Syn
echococcus sp. PCC 7003,Synechococcus sp. PCC 7003,cya
nobacteria
9_Synechococcus_s6_contig_1,CP016474.1,94.357000000000
01,4324,239,4,70911,75234,1579096,1574778,0.0,6628.0,B
acteria,"Synechococcus sp. PCC 7003, complete genome",
374981,Synechococcus sp. PCC 7003,Synechococcus sp. PC
C 7003,cyanobacteria
9_Synechococcus_s6_contig_1,CP016474.1,96.295,2996,107
,4,44553,47547,2128395,2125403,0.0,4915.0,Bacteria,"Sy
nechococcus sp. PCC 7003, complete genome",374981,Syne
chococcus sp. PCC 7003,Synechococcus sp. PCC 7003,cyan
obacteria
9_Synechococcus_s6_contig_1,CP016474.1,89.627999999999
99,3066,268,23,1,3056,2165289,2162264,0.0,3855.0,Bacte
ria,"Synechococcus sp. PCC 7003, complete genome",3749
81,Synechococcus sp. PCC 7003,Synechococcus sp. PCC 70
03,cyanobacteria
9_Synechococcus_s6_contig_1,CP016474.1,95.291,2145,101
,0,55725,57869,2114565,2112421,0.0,3402.0,Bacteria,"Sy
nechococcus sp. PCC 7003, complete genome",374981,Syne
chococcus sp. PCC 7003,Synechococcus sp. PCC 7003,cyan
obacteria
9_Synechococcus_s6_contig_1,CP016474.1,93.062999999999
```

```
99,1456,93,8,57870,59318,1596889,1595435,0.0,2122.0,Ba
cteria,"Synechococcus sp. PCC 7003, complete genome",3
74981,Synechococcus sp. PCC 7003,Synechococcus sp. PCC
7003,cyanobacteria
```

In [ ]:
```
# en caso de recuperar el archivo
ftsv= pd.read_csv("9_Synechococcus_blastn.csv")
ftsv.head(2)
```

In [61]:
```
ftab1= ftsv.groupby("sskingdoms")["qseqid"].count()
ftab1 = DataFrame(ftab1)
ftab1
```

Out[61]:

	qseqid
sskingdoms	
Bacteria	6836
Eukaryota	16

In [62]:
```
ftab2= ftsv.groupby(["sskingdoms","sblastnames"])["qseq
id"].count()
ftab2 = DataFrame(ftab2)
ftab2
```

Out[62]:

		qseqid
sskingdoms	**sblastnames**	
Bacteria	**CFB group bacteria**	195
	a-proteobacteria	5803
	b-proteobacteria	36
	bacteria	12
	cyanobacteria	706
	d-proteobacteria	2
	enterobacteria	1
	firmicutes	13
	g-proteobacteria	57
	high GC Gram+	10
	proteobacteria	1
Eukaryota	**bony fishes**	5
	eudicots	4
	nematodes	1
	primates	1
	red algae	1
	rodents	1
	sea anemones	3

Obteniendo solamente "Bacteria"

In [63]:
```
ftab2= ftsv.groupby(["sskingdoms"]).get_group('Bacteria
')
ftab2
```

Out[63]:

	qseqid	sseqid	pident	length	m
0	9_Synechococcus_s6_contig_1	CP016474.1	93.870	24405	1
1	9_Synechococcus_s6_contig_1	CP016474.1	95.261	10213	4
2	9_Synechococcus_s6_contig_1	CP016474.1	94.773	5376	2
3	9_Synechococcus_s6_contig_1	CP016474.1	89.007	6568	6
4	9_Synechococcus_s6_contig_1	CP016474.1	94.357	4324	2
5	9_Synechococcus_s6_contig_1	CP016474.1	96.295	2996	1
6	9_Synechococcus_s6_contig_1	CP016474.1	89.628	3066	2
7	9_Synechococcus_s6_contig_1	CP016474.1	95.291	2145	1
8	9_Synechococcus_s6_contig_1	CP016474.1	93.063	1456	9

9	9_Synechococcus_s6_contig_1	CP016474.1	90.723	1272	1
10	9_Synechococcus_s6_contig_1	CP016474.1	90.525	1277	1
11	9_Synechococcus_s6_contig_1	CP016474.1	90.701	1269	1
12	9_Synechococcus_s6_contig_1	CP016474.1	90.439	1276	1
13	9_Synechococcus_s6_contig_1	CP016474.1	87.510	1281	1
14	9_Synechococcus_s6_contig_1	CP016474.1	87.510	1281	1
15	9_Synechococcus_s6_contig_1	CP016474.1	87.510	1281	1
16	9_Synechococcus_s6_contig_1	CP016474.1	93.548	930	6
17	9_Synechococcus_s6_contig_1	CP016474.1	91.180	771	5
18	9_Synechococcus_s6_contig_1	CP016474.1	87.307	906	1

19	9_Synechococcus_s6_contig_1	CP016474.1	92.308	676	5
20	9_Synechococcus_s6_contig_1	CP016474.1	92.949	624	4
21	9_Synechococcus_s6_contig_1	CP016474.1	91.652	551	4
22	9_Synechococcus_s6_contig_1	CP016474.1	90.065	463	4
23	9_Synechococcus_s6_contig_1	CP016474.1	86.639	479	3
24	9_Synechococcus_s6_contig_1	CP016474.1	83.527	516	7
25	9_Synechococcus_s6_contig_1	CP016474.1	93.060	317	2
26	9_Synechococcus_s6_contig_1	CP016474.1	93.311	299	2
27	9_Synechococcus_s6_contig_1	CP016474.1	93.289	298	1
28	9_Synechococcus_s6_contig_1	CP016474.1	83.542	480	6

29	9_Synechococcus_s6_contig_1	CP016474.1	93.536	263	1
...	...	...	...	...	..
6823	9_Synechococcus_s6_contig_8236	CP045392.1	79.699	266	5
6824	9_Synechococcus_s6_contig_8237	CP053921.1	87.826	345	4
6825	9_Synechococcus_s6_contig_8240	CP021912.1	79.464	224	4
6826	9_Synechococcus_s6_contig_8243	CP016474.1	92.835	321	2
6827	9_Synechococcus_s6_contig_8244	CP015963.1	96.094	256	8
6828	9_Synechococcus_s6_contig_8249	CP040360.1	98.901	91	1
6829	9_Synechococcus_s6_contig_8249	CP040360.1	98.901	91	1
6830	9_Synechococcus_s6_contig_8250	CP016477.1	88.101	395	4

6831	9_Synechococcus_s6_contig_8251	CP016474.1	94.430	395	2
6832	9_Synechococcus_s6_contig_8252	CP016474.1	94.043	470	2
6833	9_Synechococcus_s6_contig_8253	CP019337.1	77.841	176	3
6834	9_Synechococcus_s6_contig_8254	CP016474.1	99.682	314	1
6835	9_Synechococcus_s6_contig_8255	CP000951.1	92.462	199	1
6836	9_Synechococcus_s6_contig_8256	CP016483.1	91.892	296	2
6837	9_Synechococcus_s6_contig_8257	CP000031.2	87.290	417	4
6838	9_Synechococcus_s6_contig_8258	CP016474.1	96.284	296	1
6839	9_Synechococcus_s6_contig_8259	CP016474.1	89.610	231	2
6840	9_Synechococcus_s6_contig_8261	CP016474.1	91.176	238	1

6841	9_Synechococcus_s6_contig_8264	CP016483.1	89.222	167	1
6842	9_Synechococcus_s6_contig_8265	CP016474.1	94.919	433	1
6843	9_Synechococcus_s6_contig_8266	CP016474.1	93.310	284	1
6844	9_Synechococcus_s6_contig_8267	CP002825.1	75.000	256	5
6845	9_Synechococcus_s6_contig_8268	CP013998.1	93.578	218	1
6846	9_Synechococcus_s6_contig_8269	CP016474.1	92.324	482	3
6847	9_Synechococcus_s6_contig_8270	CP016474.1	97.834	277	6
6848	9_Synechococcus_s6_contig_8271	CP016474.1	95.133	226	1
6849	9_Synechococcus_s6_contig_8272	CP013998.1	85.714	266	3
6850	9_Synechococcus_s6_contig_8273	CP016474.1	94.932	296	1

6851	9_Synechococcus_s6_contig_8274	CP007202.1	88.667	150	1
6852	9_Synechococcus_s6_contig_8275	CP000951.1	92.889	225	1

6836 rows × 18 columns

In [66]:
```
ftab2.to_csv("9_Synechococcus_blastn_bacterias.csv", header=True, index= None)
```

In [67]:
```
ftab3= ftab2.groupby(["sblastnames"])["qseqid"].count()
ftab3.sort_values(axis = 0, ascending=False, inplace=True)

#ftab3 = DataFrame(ftab3)
ftab3
```

Out[67]:
```
sblastnames
a-proteobacteria      5803
cyanobacteria          706
CFB group bacteria     195
g-proteobacteria        57
b-proteobacteria        36
firmicutes              13
bacteria                12
high GC Gram+           10
d-proteobacteria         2
proteobacteria           1
enterobacteria           1
Name: qseqid, dtype: int64
```

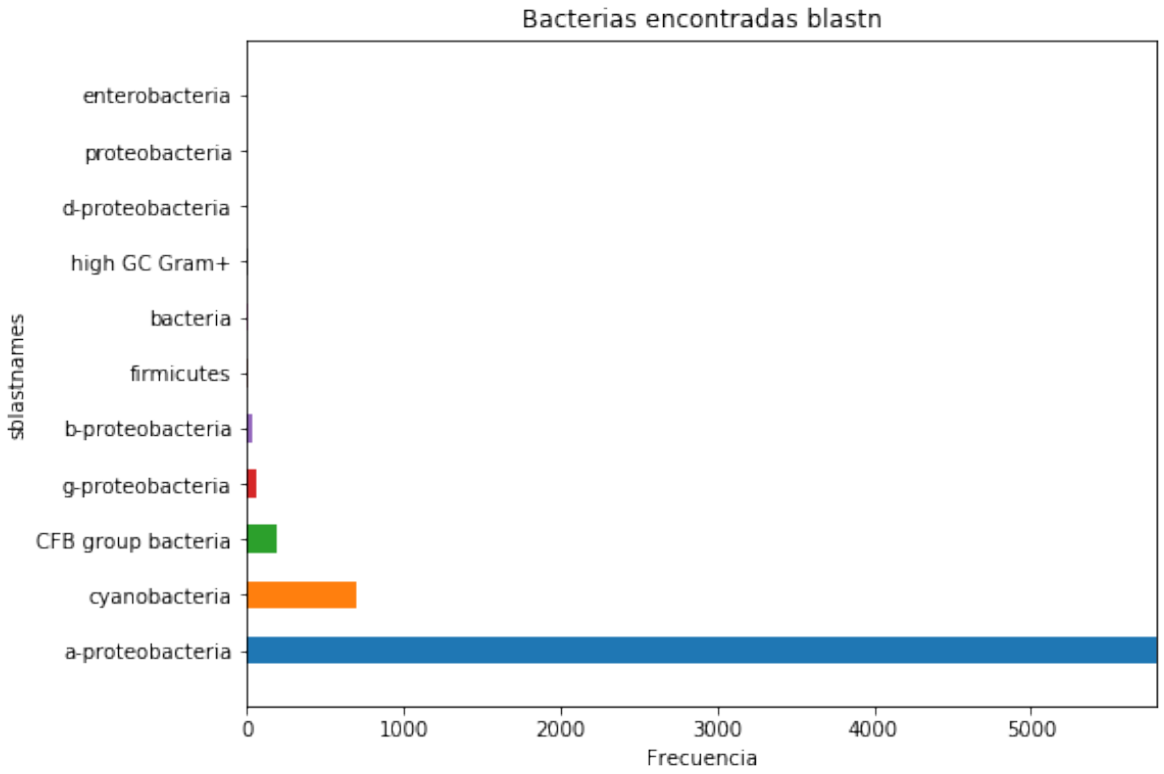
In [68]:
```
# debe ser el nombre del archivo fasta
f
```

Out[68]:
```
'9_Synechococcus_s6assembly.fasta'
```

In [69]:
```
# en caso contrario ejecutar
#f = '9_Synechococcus_s6assembly.fasta'
```

In [70]:
```
ftab3.plot(kind='barh', figsize= (8,6))
plt.axis([-1, int(max(ftab3)+5), -1, ftab3.count()], la
bel=None)
plt.legend().set_visible(False)
plt.xlabel("Frecuencia")
plt.ylabel("sblastnames")
plt.title("Bacterias encontradas blastn")
yes = input("save figure? (y/) ")
if yes.lower()=="y":
    archivo = f[:f.find(".")]+'bacterias.png'
    plt.savefig(archivo, dpi=400, bbox_inches='tight')

plt.show()
```

```
save figure? (y/) y
```



Analizando Eucariotas

In [71]:
```
f_eucaria= ftsv.groupby(["sskingdoms"]).get_group('Euka
ryota')
f_eucaria.head()
```

Out[71]:

	qseqid	sseqid	pident	leng
995	9_Synechococcus_s6_contig_484	XM_001635996.2	94.444	72
1501	9_Synechococcus_s6_contig_1120	KJ532072.1	88.698	407
2082	9_Synechococcus_s6_contig_1863	XM_001635996.2	96.000	75
3030	9_Synechococcus_s6_contig_3067	LN590697.1	93.000	100
3031	9_Synechococcus_s6_contig_3067	LN590697.1	89.899	99

In [72]:
```
f_eucaria.to_csv("9_Synechococcus_blastn_eucaria.csv",
header=True, index= None)
```

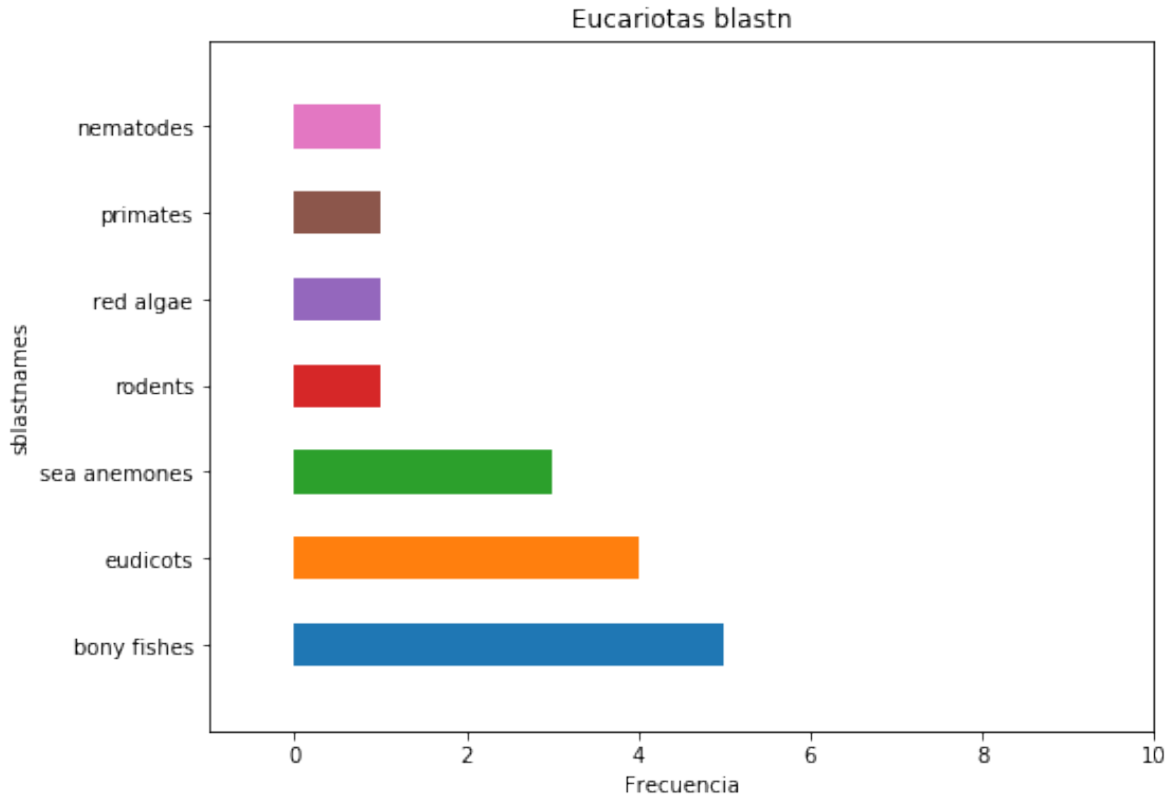
In [ ]:
```
f_eucaria = pd.read_csv("9_Synechococcus_blastn_eucaria
.csv", engine="python")
f_eucaria.head(2)
```

In [73]:
```
f_eucaria3= f_eucaria.groupby(["sblastnames"])["qseqid"
].count()
f_eucaria3.sort_values(axis = 0, ascending=False, inpla
ce=True)

#ftab3 = DataFrame(ftab3)
f_eucaria3
```

Out[73]:
```
sblastnames
bony fishes     5
eudicots        4
sea anemones    3
rodents         1
red algae       1
primates        1
nematodes       1
Name: qseqid, dtype: int64
```

In [74]:
```
f_eucaria3.plot(kind='barh', figsize= (8,6))
plt.axis([-1, int(max(f_eucaria3)+5), -1, f_eucaria3.co
unt()], 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(".")]+'eucaria.png'
    plt.savefig(archivo, dpi=400, bbox_inches='tight')

plt.show()
```

```
save figure (y/)? y
```



Aquí termina el análisis blastn de 9_Synechococcus

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

Blastx

Se hace el archivo `.sh` para que no se utilice el nodo maestro de omica en el blast. Esto también libera al Jupyter para seguir haciendo procesos o análisis mientras se ejecuta el blastx.

In [12]:
```
ls -lh ~/bigdata/swissprot/
```

```
total 630M
-rw-r--r-- 1 mdelrio gen_acuicola 119M Nov 18  2019 sw
issprot.00.phr
-rw-r--r-- 1 mdelrio gen_acuicola 3.7M Nov 18  2019 sw
issprot.00.pin
-rw-r--r-- 1 mdelrio gen_acuicola 4.3M Nov 18  2019 sw
issprot.00.pnd
-rw-r--r-- 1 mdelrio gen_acuicola  18K Nov 18  2019 sw
issprot.00.pni
-rw-r--r-- 1 mdelrio gen_acuicola 1.9M Nov 18  2019 sw
issprot.00.pog
-rw-r--r-- 1 mdelrio gen_acuicola 3.7M Nov 18  2019 sw
issprot.00.ppd
-rw-r--r-- 1 mdelrio gen_acuicola  15K Nov 18  2019 sw
issprot.00.ppi
-rw-r--r-- 1 mdelrio gen_acuicola  26M Nov 18  2019 sw
issprot.00.psd
-rw-r--r-- 1 mdelrio gen_acuicola 614K Nov 18  2019 sw
issprot.00.psi
-rw-r--r-- 1 mdelrio gen_acuicola 172M Nov 18  2019 sw
issprot.00.psq
-rw-r--r-- 1 mdelrio gen_acuicola   31 Nov 18  2019 sw
issprot.pal
-rw-r--r-- 1 mdelrio gen_acuicola 150M Nov 21  2019 sw
issprot.tar.gz
-rw-r--r-- 1 mdelrio gen_acuicola   51 Nov 21  2019 sw
issprot.tar.gz.md5
-rw-r--r-- 1 mdelrio gen_acuicola 137M Nov 19  2019 ta
xdb.btd
-rw-r--r-- 1 mdelrio gen_acuicola  15M Nov 19  2019 ta
xdb.bti
```

In [13]:
```
pwd
```

Out[13]:
```
'/LUSTRE/bioinformatica_data/lga/mdelrio/synechococcus
'
```

In [28]:
```
fout = open("blastx_synechococcus.sh", "w")
linea="""#!/bin/sh
#
#SBATCH -p cicese
#SBATCH --job-name=blastx_9syne
```

```
#SBATCH -e blastx_9syne.%N.%j.err
#SBATCH -o blastx_9syne.%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=~/bigdata/swissprot/

#
cd /LUSTRE/bioinformatica_data/lga/mdelrio/synechococcu
s/
date > tiempo_blastx_synechococcus_final.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin
/blastx \\
 -query 9_Synechococcus_s6assembly.fasta  \\
 -db ~/bigdata/swissprot/swissprot \\
 -out 9_Synechococcus_blastx_final.tsv \\
 -evalue 1E-6 \\
 -max_target_seqs 1 \\
 -num_threads 23 \\
 -outfmt "6 std stitle"
date >> tiempo_blastx_synechococcus_final.txt

#SBATCH --mail-type=all
#SBATCH --mail-user=mdelrio@cicese.mx
```

```
exit 0
"""
fout.write(linea)
fout.close()
```

In [27]:
```
ls ~/bigdata/swissprot/
```

```
swissprot.00.phr  swissprot.00.pog  swissprot.00.psi
swissprot.tar.gz.md5
swissprot.00.pin  swissprot.00.ppd  swissprot.00.psq
taxdb.btd
swissprot.00.pnd  swissprot.00.ppi  swissprot.pal
taxdb.bti
swissprot.00.pni  swissprot.00.psd  swissprot.tar.gz
```

In [33]:
```
!head -100 blastx_synechococcus.sh
```

```
#!/bin/sh
#
#SBATCH -p cicese
#SBATCH --job-name=blastx_9syne
#SBATCH -e blastx_9syne.%N.%j.err
#SBATCH -o blastx_9syne.%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/ncb
i-blast-2.11.0/lib:$LD_LIBRARY_PATH


export BLASTDB=~/bigdata/swissprot/

#
cd /LUSTRE/bioinformatica_data/lga/mdelrio/synechococc
us/
date > tiempo_blastx_synechococcus_final.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bi
n/blastx \
 -query 9_Synechococcus_s6assembly.fasta  \
 -db ~/bigdata/swissprot/swissprot \
 -out 9_Synechococcus_blastx_final.tsv \
 -evalue 1E-6 \
 -max_target_seqs 1 \
 -num_threads 23 \
 -outfmt "6 std stitle"
date >> tiempo_blastx_synechococcus_final.txt

#SBATCH --mail-type=all
#SBATCH --mail-user=mdelrio@cicese.mx


exit 0
```

In [34]:
```
!sbatch blastx_synechococcus.sh
```

```
Submitted batch job 166702
```

In [35]:
```
ls -lh *.err
```

```
-rw-r--r-- 1 mdelrio gen_acuicola 211 Mar  1 11:35 bla
stn.nodo11.166700.err
-rw-r--r-- 1 mdelrio gen_acuicola 211 Mar  1 11:45 bla
stn.nodo11.166701.err
-rw-r--r-- 1 mdelrio gen_acuicola 211 Dec 11 16:38 bla
stn.nodo2.166100.err
-rw-r--r-- 1 mdelrio gen_acuicola 211 Dec 11 16:42 bla
stn.nodo2.166101.err
-rw-r--r-- 1 mdelrio gen_acuicola   0 Mar  1  2021 bla
stx_9syne.nodo11.166702.err
```

In [37]:
```
ls -lh *.err
```

```
-rw-r--r-- 1 mdelrio gen_acuicola 211 Mar  1 11:35 bla
stn.nodo11.166700.err
-rw-r--r-- 1 mdelrio gen_acuicola 211 Mar  1 11:45 bla
stn.nodo11.166701.err
-rw-r--r-- 1 mdelrio gen_acuicola 211 Dec 11 16:38 bla
stn.nodo2.166100.err
-rw-r--r-- 1 mdelrio gen_acuicola 211 Dec 11 16:42 bla
stn.nodo2.166101.err
-rw-r--r-- 1 mdelrio gen_acuicola  61 Mar  1 11:46 bla
stx_9syne.nodo11.166702.err
```

In [21]:
```
pwd
```

Out[21]:
```
'/LUSTRE/bioinformatica_data/lga/mdelrio/synechococcus
'
```

In [36]:
```
!squeue
```

```
             JOBID PARTITION     NAME     USER ST
TIME  NODES NODELIST(REASON)
            166653       d30       GN    dante  R 5-17
:22:51      2 nodo[7,9]
            166676    cicese   blastn   sylvia  R 1-15
:05:46      1 nodo3
            166688       d30  nCopGPB  gvkaren  R   18
:40:45      1 nodo8
            166671       d30  nCopGPB  gvkaren  R 2-19
:02:37      1 nodo10
            166668    cicese  copLGPB  gvkaren  R 2-19
:42:23      1 nodo5
            166666    cicese  copACHE  gvkaren  R 2-19
:51:04      1 nodo4
            166702    cicese blastx_9  mdelrio  R
0:18      1 nodo11
            166699    cicese   blastn  mdelrio  R    1
:35:55      1 nodo6
```

In [38]:
```
!head blastx_9syne.nodo11.166702.err
```

```
Warning: [blastx] Examining 5 or more matches is recom
mended
```

In [39]:
```
%%bash
echo "errores detectados en la corrida"
echo "En caso de observar un comentario de "
echo "'Warning: [blastn] Examining 5 or more matches is
recommended'"
echo "hacer caso omiso"
echo
head blastx_9syne.nodo11.166702.err
echo
echo "archivo log"
head blastx_9syne.nodo11.166702.log
```

```
errores detectados en la corrida
En caso de observar un comentario de
'Warning: [blastn] Examining 5 or more matches is reco
mmended'
hacer caso omiso

Warning: [blastx] Examining 5 or more matches is recom
mended

archivo log
```

In [40]:
```
!squeue
```

```
             JOBID PARTITION     NAME     USER ST
TIME  NODES NODELIST(REASON)
            166653       d30       GN    dante  R 5-18
:58:26      2 nodo[7,9]
            166676    cicese   blastn   sylvia  R 1-16
:41:21      1 nodo3
            166688       d30  nCopGPB  gvkaren  R   20
:16:20      1 nodo8
            166671       d30  nCopGPB  gvkaren  R 2-20
:38:12      1 nodo10
            166668    cicese  copLGPB  gvkaren  R 2-21
:17:58      1 nodo5
            166666    cicese  copACHE  gvkaren  R 2-21
:26:39      1 nodo4
            166699    cicese   blastn  mdelrio  R    3
:11:30      1 nodo6
```

In [41]:
```
%%bash
echo "errores detectados en la corrida"
echo "En caso de observar un comentario de "
echo "'Warning: [blastn] Examining 5 or more matches is
recommended'"
echo "hacer caso omiso"
echo
head blastx_9syne.nodo11.166702.err
echo
echo "archivo log"
head blastx_9syne.nodo11.166702.log
```

```
errores detectados en la corrida
En caso de observar un comentario de
'Warning: [blastn] Examining 5 or more matches is reco
mmended'
hacer caso omiso

Warning: [blastx] Examining 5 or more matches is recom
mended

real    39m38.800s
user    811m59.641s
sys     0m19.518s

archivo log
```

In [42]:
```
ls -lh *.tsv
```

```
-rw-r--r-- 1 mdelrio gen_acuicola 1.5M Dec 11 19:14 9_
Synechococcus_blastn.tsv
-rw-r--r-- 1 mdelrio gen_acuicola 907K Mar  1 12:26 9_
Synechococcus_blastx_final.tsv
```

In [ ]:
```
!head -100 9_Synechococcus_blastx_final.tsv
```

Cargado de la base de datos spid_GO

In [43]:
```
ls /LUSTRE/bioinformatica_data/lga/bigdata/*.csv
```

```
/LUSTRE/bioinformatica_data/lga/bigdata/go_to_goslim.csv*
/LUSTRE/bioinformatica_data/lga/bigdata/spid_go.csv
```

In [28]:
```
fout = open("blastx_synechococcus.sh", "w")
linea="""#!/bin/sh
#
#SBATCH -p cicese
#SBATCH --job-name=blastx_9syne
#SBATCH -e blastx_9syne.%N.%j.err
#SBATCH -o blastx_9syne.%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=~/bigdata/swissprot/

#
cd /LUSTRE/bioinformatica_data/lga/mdelrio/synechococcus/
date > tiempo_blastx_synechococcus_final.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.11.0/bin/blastx \\
```

```
 -query 9_Synechococcus_s6assembly.fasta  \\
 -db ~/bigdata/swissprot/swissprot \\
 -out 9_Synechococcus_blastx_final.tsv \\
 -evalue 1E-6 \\
 -max_target_seqs 1 \\
 -num_threads 23 \\
 -outfmt "6 std stitle"
date >> tiempo_blastx_synechococcus_final.txt

#SBATCH --mail-type=all
#SBATCH --mail-user=mdelrio@cicese.mx


exit 0
"""
fout.write(linea)
fout.close()
```

In [48]:
```
encabezado =("qseqid", "sseqid", "pident", "length", "mismatch", "gapopen", "qstart",
             "qend", "sstart", "send", "evalue", "bitscore",  "stitle",
             "staxids", "sscinames", "scomnames", "sblastnames")
```

In [49]:
```
ftab=pd.read_csv("9_Synechococcus_blastx_final.tsv", se
p = "\t", header=None , names= encabezado)
ftab.head()
```

Out[49]:

	qseqid	sseqid	pident	length	mismatch
0	9_Synechococcus_s6_contig_1	P73412.1	77.778	963	184
1	9_Synechococcus_s6_contig_1	P73412.1	37.037	189	104
2	9_Synechococcus_s6_contig_2	P22106.3	65.884	554	188
3	9_Synechococcus_s6_contig_3	P21524.2	65.894	777	254
4	9_Synechococcus_s6_contig_4	Q55905.1	75.540	834	186

In [ ]:
```
del ftab["staxids"]
del ftab["sscinames"]
del ftab["scomnames"]
del ftab["sblastnames"]
ftab.head(2)
```

In [ ]:
```
ftab["stitle"]= ftab["sskingdoms"]
ftab.head(2)
```

In [ ]:
```
del ftab["sskingdoms"]
ftab.head(2)
```

In [ ]:
```
ftab["uniprotid"] = ftab["sseqid"].str.split(".", n = 1
, expand = True)[0]
ftab.head(2)
```

Se quita `RecName: Full=` de la columna `stitle`

In [ ]:
```
ftab["stitle"] = ftab["stitle"].str.split("=", n = 1, e
xpand = True)[1]
ftab.head(2)
```

In [ ]:
```
!date
fspid = pd.read_csv('/LUSTRE/bioinformatica_data/lga/bi
gdata/spid_go.csv', engine="python")
fspid.head(2)
!date
```

In [ ]:
```
f2=pd.merge(ftab,fspid, on ="uniprotid" , how='inner')
f2.head(2)
```

Se carga la base de datos GO

In [ ]:
```
fgo = pd.read_csv('/LUSTRE/bioinformatica_data/lga/bigd
ata/go_to_goslim.csv', engine="python")
fgo.head(2)
```

Se agregan los datos de go_slim

In [ ]:
```
f3=pd.merge(f2,fgo, on ="GO_id" , how='inner')
f3.head()
```

In [ ]:
```
f3.to_csv("13_Fundulus1_blastx_final.csv", index = None
)
```

In [ ]:
```
f3=pd.read_csv("13_Fundulus1_blastx_final.csv", engine
= "python")
f3.head()
```

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

In [ ]:
```
fspid =[]
```

In [ ]:
```
ls
```

Se eliminan duplicados

In [ ]:
```
f4=f3.drop_duplicates(subset = ('qseqid', "aspect"), in
place = False)
f4.describe().round(2)[['length','evalue']]
```

In [ ]:
```
f4.head(2)
```

In [ ]:
```
f4.to_csv("13_Fundulus1_final_goslim.csv", index= None)
```

In [ ]:
```
ftabpivot = f4.pivot_table(values="sseqid" , index=["qs
eqid"], aggfunc=len, columns="aspect")
ftabpivot.describe().round(2)
```

Proceso para poder obtener los diagramas de Venn

In [ ]:
```
lineaC =[] # data from C
lineaF =[] # data from F
lineaP =[] # data from P
linea = ""
n=1
for row in ftabpivot.index:
    row2=ftabpivot.loc[row]
    if str(row2["C"])=="nan" and str(row2["F"])=="nan"
and str(row2["P"])=="nan" :
        continue
    else:
        if str(row2["C"]) !="nan":
            linea = row
        else:
            linea = ""
        lineaC.append(linea)
        if str(row2["F"]) !="nan":
            linea = row
        else:
            linea = ""
        lineaF.append(linea)

        if str(row2["P"]) !="nan":
            linea = row
        else:
            linea = ""
        lineaP.append(linea)

        n+=1
        #if n==1000:
        #    break

len(lineaC), len(lineaF), len(lineaP)
```

In [ ]:
```
f
```

In [ ]:
```
# si 'f' no está definido, correr esta celda
f ='13_Fundulus1_final.fasta'
```

In [ ]:
```
lineaC = set(lineaC)
lineaF = set(lineaF)
lineaP = set(lineaP)
venn3_unweighted([lineaC, lineaF, lineaP], ('C', 'F', '
P'))
yes = input("save figure (y/? ")
if yes.lower()=="y":
    archivo = f[:f.find(".")]+"_venn.png"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')
    archivo = f[:f.find(".")]+"_venn.pdf"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')
plt.show()
```

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

In [ ]:
```
ls *.csv
```

In [ ]:
```
f4 = pd.read_csv("13_Fundulus1_final_goslim.csv", engin
e= "python")
f4.head()
```

In [ ]:
```
# corroborar que se obtiene aspect y GO slim
fgo=f4.groupby(['aspect','GOSlim_bin'])["qseqid"].count
()
fgo
```

In [ ]:
```
# separacion de los diferentes aspectos
aspect_f = []
aspect_p = []
aspect_c = []
for l in fgo.index:
    #print ( end ="\t")
    linea = l[1], fgo.loc[l]
    if l[0]=="C":
        print (l[0], linea, "Celular")
        aspect_c.append(linea)
    elif l[0]=="F":
        print (l[0], linea, "Function")
        aspect_f.append(linea)
    elif l[0]=="P":
        print (l[0], linea, "Process")
        aspect_p.append(linea)
    else:
        print (l[0], linea, "other")
```

In [ ]:
```
aspect_p = DataFrame(aspect_p, index = None, columns =
("P", "numero"))
aspect_p.set_index("P", inplace=True)
aspect_p.sort_values(by ='numero', inplace=True, ascend
ing=False)
```

In [ ]:
```
#aspect_p
ymaximo = len (aspect_p)
xmaximo = int(round(max (aspect_p["numero"])*1.1,0)/10)
*10
aspect_p.plot(kind='barh', color=('rybg'))
plt.axis([-1, xmaximo, -1, ymaximo], label=None)
plt.xlabel("Count")
plt.ylabel("GOSlim bin")
plt.title("Biological processes")
plt.legend().set_visible(False)
yes = input("save figure? ")
if yes.lower()=="y":
    archivo = f[:f.find(".")]+"_final_GObar_process.png
"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')
    #plt.savefig("../../data/pulposirene/id019/Contigsi
dt019blastx_GObar_process.png", dpi=400, bbox_inches='t
ight')


plt.show()
```

In [ ]:
```
aspect_c = DataFrame(aspect_c, index = None, columns =
("C", "numero"))
aspect_c.set_index("C", inplace=True)
aspect_c.sort_values(by ='numero', inplace=True, ascend
ing=False)
#aspect_c
```

In [ ]:
```
ymaximo = len (aspect_c)
xmaximo = int(round(max (aspect_c["numero"])*1.1,0)/10)
*10
aspect_c.plot(kind='barh', color=('rybg'))
plt.axis([-1, xmaximo, -1, ymaximo], label=None)
plt.xlabel("Count")
plt.ylabel("GOSlim bin")
plt.title("Cellular components")
plt.legend().set_visible(False)
yes = input("save figure? ")
if yes.lower()=="y":
    archivo = f[:f.find(".")]+"_final_GObar_cellcompone
nt.png"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')


plt.show()
```

In [ ]:
```
aspect_f = DataFrame(aspect_f, index = None, columns =
("F", "numero"))
aspect_f.set_index("F", inplace=True)
aspect_f.sort_values(by ='numero', inplace=True, ascend
ing=False)
```

In [ ]:
```
#aspect_f
ymaximo = len (aspect_f)
xmaximo = int(round(max (aspect_f["numero"])*1.1,0)/10)
*10
aspect_f.plot(kind='barh', color=('rybg'))
plt.axis([-1, xmaximo, -1, ymaximo], label=None)
plt.xlabel("Count")
plt.ylabel("GOSlim bin")
plt.title("Biological functions")
plt.legend().set_visible(False)
yes = input("save figure? ")
if yes.lower()=="y":
    archivo = f[:f.find(".")]+"_final_GObar_function.pn
g"
    plt.savefig(archivo, dpi=400, bbox_inches='tight')

plt.show()
```

Se va a realizar blastn al genoma de *Fundulus heteroclitus* (https://www.ncbi.nlm.nih.gov/genome/?term=Fundulus%20genome).

Se descargaron el genoma y el transcriptoma.

In [ ]:
```
# reconstrucción de la base de datos con makedb
!ls /LUSTRE/apps/bioinformatica/ncbi-blast-2.6.0/bin/ma
keblastdb*
```

In [ ]:
```
cd fundulus_heteroclitus/
```

In [ ]:
```
ls
```

Base de datos genoma

In [ ]:
```
!/LUSTRE/apps/bioinformatica/ncbi-blast-2.6.0/bin/makeblastdb \
 -in GCF_000826765.1_Fundulus_heteroclitus-3.0.2_genomic.fna -dbtype nucl -parse_seqids \
 -out fundulus_heteroclitus_genome
```

In [ ]:
```
ls
```

Base de datos transcriptoma

In [ ]:
```
!/LUSTRE/apps/bioinformatica/ncbi-blast-2.6.0/bin/makeblastdb \
 -in GCF_000826765.1_Fundulus_heteroclitus-3.0.2_rna.fna\
 -dbtype nucl -parse_seqids -out fundulus_heteroclitus_mrna
```

In [ ]:
```
ls
```

Blastn al genoma de *Fundulus heteroclitus*

In [ ]:
```
cd ..
```

Se requiere un archivo `.sh` con el fin de tener la información en la bitácora se crea el archivo

In [ ]:
```
fout = open("blastn_13_Fundulus1_final_fundulus_heteroc
litus.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
#

export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db/
NT

#
cd ~/data/Sample_13Fundulus/
date > tiempo_blastn_fhetero.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.6.0/bin/
blastn \\
 -query 13_Fundulus1_final.fasta \\
 -db  fundulus_heteroclitus/fundulus_heteroclitus_genom
e \\
 -out 13_Fundulus1_final_blastn_fhetero.tsv \\
 -evalue 1E-6 \\
 -max_target_seqs 1 \\
 -num_threads 24 \\
 -outfmt "6 std stitle"
date >> tiempo_blastn_fhetero.txt

exit 0
"""
fout.write(linea)
fout.close()
```

In [ ]:
```
!head -100 blastn_13_Fundulus1_final_fundulus_heteroclitus.sh
```

In [ ]:
```
!date
```

In [ ]:
```
!sbatch blastn_13_Fundulus1_final_fundulus_heteroclitus.sh
```

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

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

In [ ]:
```
!head -100 blastn.nodo2.157994.err
```

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

In [ ]:
```
!head -2 13_Fundulus1_final_blastn_fhetero.tsv
!tail -2 13_Fundulus1_final_blastn_fhetero.tsv
```

In [ ]:
```
encabezado_agenoma =("qseqid", "sseqid", "pident", "length", "mismatch", "gapopen", "qstart",
                     "qend", "sstart", "send", "evalue", "bitscore", "stitle")
```

In [ ]:
```
f_agenoma=pd.read_csv("13_Fundulus1_final_blastn_fhetero.tsv", sep = "\t", header=None , names= encabezado_agenoma)
f_agenoma.head()
```

In [ ]:
```
f_agenoma_unicos = f_agenoma.sort_values(by=["qseqid","bitscore"], inplace=False, ascending=[True, False])
f_agenoma_unicos = f_agenoma_unicos.drop_duplicates(subset = 'qseqid', keep='first', inplace = False)
f_agenoma_unicos.sort_index(inplace=True)
f_agenoma_unicos
```

In [ ]:
```
f_agenoma_unicos_genoma = f_agenoma.sort_values(by=["qs
eqid","bitscore"], inplace=False, ascending=[True, Fals
e])
f_agenoma_unicos_genoma = f_agenoma_unicos.drop_duplica
tes(subset = 'sseqid', keep='first', inplace = False)
f_agenoma_unicos_genoma.sort_index(inplace=True)
f_agenoma_unicos_genoma.head(2)
```

In [ ]:
```
len(f_agenoma), len(f_agenoma_unicos), len(f_agenoma_un
icos_genoma)
```

In [ ]:
```
!grep ">"
fundulus_heteroclitus/GCF_000826765.1_Fundulus_heterocl
itus-3.0.2_genomic.fna |wc -l
```

In [ ]:
```
# Se mapearon un total de 4635 contigs de Fundulus hete
roclitus, esto es

print(round(4635/10180*100, 4), "% del total de contigs
existentes en Fundulus heteroclitus")
```

Blastn al transcriptoma de *Fundulus heteroclitus*

In [ ]:
```
cd ~/data/Sample_13Fundulus/
```

Se ejecutó desde la terminal el blastn en *omica* dado que todavía no se tiene el proceso para que, desde Jupyter, no se ejecute en el nodo maestro.
Entonces se requiere ejecutar en un nodo mediante **slurm**.

Para ello se requiere un archivo `.sh` el cual se puede crear mediante *nano*, pero con el fin de tener la información en la bitácora se crea el archivo desde python y se corrigen los parámetros, los archivos y las rutas de búsqueda para que toda la información, quede en esta bitácora.
Se abrió python en la terminal de omica y se ejecutó el siguiente comando:

In [ ]:
```
fout = open("blastn_mrna_fundulus_heteroclitus.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
#

export BLASTDB=/LUSTRE/bioinformatica_data/BD/blast/db/
NT
#SBATCH --mail-type=all
#SBATCH --mail-user=mdelrio@cicese.mx


#
cd ~/data/Sample_13Fundulus/
date > tiempo_blastn_mrna_fhetero.txt
time  /LUSTRE/apps/bioinformatica/ncbi-blast-2.6.0/bin/
blastn \\
 -query 13_Fundulus1_final.fasta \\
 -db  fundulus_heteroclitus/fundulus_heteroclitus_mrna
\\
 -out 13_Fundulus1_final_blastn_mrna_fhetero.tsv \\
 -evalue 1E-6 \\
```

```
 -max_target_seqs 1 \\
 -num_threads 24 \\
 -outfmt "6 std stitle"
date >> tiempo_blastn_mrna_fhetero.txt

#SBATCH --mail-type=all
#SBATCH --mail-user=mdelrio@cicese.mx


exit 0
"""
fout.write(linea)
fout.close()
```

se sale de python exit()

In [ ]:
```
# Se corroboró el archivo ya en unix
```

Se ejecutan los comandos desde la bitácora para guardar los resultados observados en la terminal (si se desea ejecutarlos en la terminal directamente, hay que quitar el `!`.

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

In [ ]:
```
!date
```

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

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

In [ ]:
```
!head -100 blastn.nodo2.157994.err
```

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

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

In [ ]:
```
!head blastn.nodo2.157994.err
```

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

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

In [ ]:
```
!head -2 13_Fundulus1_final_blastn_mrna_fhetero.tsv
!tail -2 13_Fundulus1_final_blastn_mrna_fhetero.tsv
```

In [ ]:
```
ls 13_Fundulus1_final_blastn_mrna_fhetero.tsv
```

In [ ]:
```
encabezado_atrans =("qseqid", "sseqid", "pident", "leng
th", "mismatch", "gapopen", "qstart",
             "qend", "sstart", "send", "evalue", "bitsc
ore", "stitle")
```

In [ ]:
```
f_atrans=pd.read_csv("13_Fundulus1_final_blastn_mrna_fh
etero.tsv", sep = "\t", header=None , names= encabezado
_atrans)
f_atrans.head()
```

In [ ]:
```
f_atrans.to_csv("13_Fundulus1_final_blastn_mrna_fhetero
.csv", index=None)
```

In [ ]:
```
!head -2 13_Fundulus1_final_blastn_mrna_fhetero.csv
```

In [ ]:
```
f_atrans = pd.read_csv("13_Fundulus1_final_blastn_mrna_
fhetero.csv")
f_atrans.head()
```

In [ ]:
```
f_atrans["stitle"] = f_atrans.stitle.str.split("heteroc
litus",expand=True)[1]
f_atrans.head()
```

In [ ]:
```
f_atrans_sseqid = f_atrans.groupby("sseqid")["qseqid"].
count()
f_atrans_sseqid.sort_values(axis = 0, ascending=False,
inplace=True)
f_atrans_sseqid
```

In [ ]:
```
f_atrans_unicos = f_atrans.sort_values(by=["sseqid"], inplace=False, ascending=True)
f_atrans_unicos = f_atrans_unicos.drop_duplicates(subset = 'sseqid', keep='first', inplace = False)
f_atrans_unicos.sort_index(inplace=True)
f_atrans_unicos
```

In [ ]:
```
f_atrans_sseqid = f_atrans_unicos.groupby("sseqid")["qseqid"].count()
f_atrans_sseqid.sort_values(axis = 0, ascending=False, inplace=True)
f_atrans_sseqid
```

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

In [ ]:
```
!grep ">" fundulus_heteroclitus/GCF_000826765.1_Fundulus_heteroclitus-3.0.2_rna.fna |wc -l
```

In [ ]:
```
# Se mapearon un total de 4635 contigs de Fundulus heteroclitus, esto es

print(round(16705/41170*100, 4), "% del total de secuencias de mRNA existentes en Fundulus heteroclitus")
```

In [ ]:
```
f_atrans_unicos = f_atrans.sort_values(by=["sseqid","sstart"], inplace=False, ascending=[True, False])
f_atrans_unicos = f_atrans_unicos.drop_duplicates(subset = 'sseqid', keep='first', inplace = False)
f_atrans_unicos.sort_index(inplace=True)
f_atrans_unicos
```

In [ ]:
```
f_atrans_sseqid = f_atrans.groupby("stitle")["qseqid"].count()
f_atrans_sseqid.sort_values(axis = 0, ascending=False, inplace=True)
f_atrans_sseqid
```

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

In [ ]:
```
n=0
for row in f_atrans_sseqid.index:
    row1= f_atrans.loc[f_atrans["stitle"]==row]
    print(row1["qseqid"], row1["sseqid"], row1["stitle"
])
    n+=1
    if n==10:
        break
```

In [ ]:
```
row
```

In [ ]:
```
row1= f_atrans.loc[f_atrans["stitle"]==row]
row1
```

In [ ]:
```
row1 = row1.sort_values(by=["sstart", "qseqid" ], inpla
ce=False, ascending=[True, True])
row1
```

In [ ]:
```
for r in row1.index:
    r1 = row1.loc[r]
    #print(r1["qseqid"]+ "\t"+ r1["sstart"]+ "\t"+ r1["
send"]+ "\t"+ r1["length"])
    print ('{:<26} {:>5} {:>5} {:>4}'.format(r1["qseqid
"], r1["sstart"], r1["send"], r1["length"] ))
```

In [ ]:
```
len("13_Fundulus1_contig_2038")
```

In [ ]:
```
n=0
for row in f_atrans["stitle"]:
    if row.find("ribosomal")!=-1:
        print(row)
        n+=1
print(n)
```

In [ ]:
```
f_atrans_sseqid = f_atrans.groupby("sseqid")["qseqid"].
count()
f_atrans_sseqid.sort_values(axis = 0, ascending=False,
inplace=True)
f_atrans_sseqid
```

In [ ]:
```
f_atrans_sseqid.sort_values(axis = 0, ascending=False,
inplace=True)
```

pl.hist(sizes['GC'], bins=20) pl.title("%i secuencias_anotadas.fasta\nGC %i to %i" \ % (len(sizes["length"]),min(sizes['length']),max(sizes['length']))) pl.xlabel("GC content (%)") pl.ylabel("Count") #pl.legend().set_visible(False) pl.show()

In [ ]:
```
f_atrans_unicos = f_atrans.sort_values(by=["qseqid","bi
tscore"], inplace=False, ascending=[True, False])
f_atrans_unicos = f_atrans_unicos.drop_duplicates(subse
t = 'qseqid', keep='first', inplace = False)
f_atrans_unicos.sort_index(inplace=True)
f_atrans_unicos
```

In [ ]:
```
len(f_atrans), len(f_atrans_unicos)
```

In [ ]:
```
!grep ">"
fundulus_heteroclitus/GCF_000826765.1_Fundulus_heterocl
itus-3.0.2_rna.fna |wc -l
```

In [ ]:
```
f_atrans_unicos2 = f_atrans.sort_values(by=["qseqid", "
sseqid","bitscore"], inplace=False, ascending=[True, Tr
ue, False])
f_atrans_unicos2 = f_atrans_unicos2.drop_duplicates(sub
set = 'sseqid', keep='first', inplace = False)
f_atrans_unicos2.sort_index(inplace=True)
f_atrans_unicos2
```

In [ ]:
```
print ('{:>6} {:>6} {:>6} {:>6.2f}%'.format(len(f_atran
s), len(f_atrans_unicos), len(f_atrans_unicos2), len(f_
atrans_unicos2)/41170*100))
```

In [ ]:
```
for row in f_atrans_unicos2.index:
    row1=f_atrans_unicos2.loc[row]
    print(row1["qend"]-row1["qstart"]+row1["gapopen"],
abs(row1["send"]-row1["sstart"]), end = "\t")
    if row1["qend"]-row1["qstart"]+row1["gapopen"]== ab
s(row1["send"]-row1["sstart"]):
        print()
    else:
        print("diferente", row1["qend"]-row1["qstart"]+
row1["gapopen"]-abs(row1["send"]-row1["sstart"]))
```

18S busqueda

In [ ]:
```
ls fundulus_heteroclitus/
```

In [ ]:
```
# Se buscó la secuencia en el genbank y se copió parcia
lmente a:
fhetero_18s ="AGCATATGCTTGTCTCAAAGATTAAGCCATGCAAGTCTAAG
TACACACGGCCGGTACAGTGAAACTGCGA"
# con el fin de obtener el contig en donde se localizó.
```

In [ ]:
```
for rec in SeqIO.parse(open("fundulus_heteroclitus/GCF_
000826765.1_Fundulus_heteroclitus-3.0.2_genomic.fna", '
r'), "fasta"):
    if rec.seq.find(fhetero_18s)!=-1:
        print(rec.id, rec.seq[:50])
```

In [ ]:
```
for rec in SeqIO.parse(open("fundulus_heteroclitus/GCF_
000826765.1_Fundulus_heteroclitus-3.0.2_rna.fna", 'r'),
"fasta"):
    if rec.seq.find(fhetero_18s)!=-1:
        print(rec.id, rec.seq[:50])
```

In [ ]:
```
# no está en el rna, solo en el genoma completo
```

In [ ]:
```
fheteroclitus_18s = []
for rec in SeqIO.parse(open("fundulus_heteroclitus/GCF_
000826765.1_Fundulus_heteroclitus-3.0.2_genomic.fna", '
r'), "fasta"):
    if rec.seq.find(fhetero_18s)!=-1:
        fheteroclitus_18s = rec
        print(rec.id, rec.seq[:50])
```

In [ ]:
```
fheteroclitus_18s
```

In [ ]:
```
len(fheteroclitus_18s)
```

Búsqueda del contig para obtener el 18 en *Fundulus lima*

In [ ]:
```
flima_18=[]
n=0
for row in ftsv['stitle']:
    if row.find("18S")!=-1:
        n+=1
        print(n, row)
        flima_18.append(row)
```

In [ ]:
```
flima_18
```

In [ ]:
```
secuencias=[]
for row18s in flima_18:
    for row in ftsv.index:
        row1 = ftsv.loc[row]
        if row1['stitle']==row18s:
            print(row1["qseqid"], row1["sseqid"], row1[
"stitle"][:10])
            secuencias.append(row1["qseqid"])
```

In [ ]:
```
secuencias
```

In [ ]:
```
secuencias_18s = ['13_Fundulus1_contig_580',
 '13_Fundulus1_contig_2887',
 '13_Fundulus1_contig_7095',
 '13_Fundulus1_contig_10455']
```

In [ ]:
```
f
```

In [ ]:
```
# extrayendo las secuencias asociadas al 18s de F. lima
flima_18s = []
for rec in SeqIO.parse(open(f, 'r'), "fasta"):
    if rec.id in secuencias_18s:
        print(rec.id, "\t", len(rec.seq), "\t", rec.seq
[:50])
        flima_18s.append(rec)
```

In [ ]:
```
SeqIO.write(flima_18s, "flima_18s.fasta", "fasta")
!grep ">" flima_18s.fasta
```

In [ ]:
```
secuencia_fas = SeqRecord(Seq(secuencia.upper()), id="P
1a-16S", description="P1a-16S" )
```

In [ ]:
```
# extrayendo las secuencias asociadas al 18s de F. lima
flima_18s = []
for rec in SeqIO.parse(open(f, 'r'), "fasta"):
    if rec.id =="13_Fundulus1_contig_580":
        print(rec.id, "\t", len(rec.seq), "\t", rec.seq
[:50])
        flima_18s = SeqRecord(rec.seq, id=rec.id, descr
iption=rec.description )
```

In [ ]:
```
SeqIO.write(flima_18s, "flima_18s_contig580.fasta", "fa
sta")
!grep ">" flima_18s_contig580.fasta
```

In [ ]:
```
record = SeqIO.read("flima_18s_contig580.fasta", format
="fasta")
result_handle = NCBIWWW.qblast("blastn", "nt", record.f
ormat("fasta"), hitlist_size = 20)
```

In [ ]:
```
with open("flima_18s_contig580.xml", "w") as out_handle
:
    out_handle.write(result_handle.read())

result_handle.close()
```

In [ ]:
```
result_handle = open("flima_18s_contig580.xml")
blast_record = NCBIXML.read(result_handle)
```

debido a que se descargaron las secuencias cortadas, se utilizó

```
len(hsp.sbjct)>2000
```

para obtener solamente el 18S en la línea:

```
if hsp.expect < E_VALUE_THRESH and len(hsp.sbjct)>2000
```

In [ ]:
```
# con valor de corte
E_VALUE_THRESH = 0.001
secuencias = []
for alignment in blast_record.alignments:
    for hsp in alignment.hsps:
         if hsp.expect < E_VALUE_THRESH and len(hsp.sbj
ct)>2000:
            print("****Alignment****")
            print("sequence:", alignment.title)
            print("length:", alignment.length)
            print("e value:", hsp.expect)
            print(hsp.query[0:75] + "...")
            print(hsp.match[0:75] + "...")
            print(hsp.sbjct[0:75] + "...")
            linea =SeqRecord(Seq(hsp.sbjct), id=gespeci
e(alignment.hit_def)+
                                "_"+alignment.accession ,
description=generoespecie(alignment.hit_def))
            secuencias.append(linea)
```

for alignment in blast_record.alignments: for hsp in alignment.hsps: print("**Alignment**") print("sequence:", alignment.title) print("length:", alignment.length) print("e value:", hsp.expect) print(hsp.query[0:75] + "...") print(hsp.match[0:75] + "...") print(hsp.sbjct[0:75] + "...")

In [ ]:
```
n=0
for rec in secuencias:
    n+=1
    print(n,rec.id, rec.description, len(rec.seq))
```

In [ ]:
```
secuencias.append(record)
SeqIO.write(secuencias, 'Flima_alineamiento.fasta', 'fa
sta')
```

Observé que solamente está la secuencia del 18S y al revisarla en el CLC, me di cuenta que solamente hay una parte del 18S, por lo que he decidido descargar las 20 secuencias completas para realizar el alineamiento y análisis posterior.

Para ello se agregan los id de las secuencias en la variable `lineas`

In [ ]:
```
E_VALUE_THRESH = 0.001
secuencias = []
lineas = []
for alignment in blast_record.alignments:
    for hsp in alignment.hsps:
         if hsp.expect < E_VALUE_THRESH and len(hsp.sbj
ct)>2000:
            print("****Alignment****")
            print("sequence:", alignment.title)
            print("length:", alignment.length)
            print("e value:", hsp.expect)
            print(hsp.query[0:75] + "...")
            print(hsp.match[0:75] + "...")
            print(hsp.sbjct[0:75] + "...")
            linea =SeqRecord(Seq(hsp.sbjct), id=gespeci
e(alignment.hit_def)+
                                "_"+alignment.accession ,
description=generoespecie(alignment.hit_def))
            secuencias.append(linea)

            lineas.append(alignment.hit_id.split("|")[3
])
```

In [ ]:
```
# Se corroboran las secuencias
n=0
for rec in lineas:
    n+=1
    print(n,rec)
```

In [ ]:
```
ls *.fasta
```

In [ ]:
```
secuencias= []
Entrez.email = "mdelrio@cicese.mx"
handle = Entrez.efetch(db="nucleotide", rettype="gb", r
etmode="text",
                       id=lineas)
for seq_record in SeqIO.parse(handle, "gb"):
    print("%s %s..." % (seq_record.id, seq_record.descr
iption[:50]))
    print("Sequence length %i, %i features, from: %s"
          % (len(seq_record), len(seq_record.features),
seq_record.annotations["source"]))
    secuencias.append(seq_record)
SeqIO.write(secuencias, "Flima_alineamiento_genban.gb",
"gb")
SeqIO.write(secuencias, "Flima_alineamiento_genban.fast
a", "fasta")

handle.close()
```

In [ ]:
```
!head Flima_alineamiento_genban.gb
```

In [ ]:
```
!grep ">" Flima_alineamiento_genban.fasta
```

In [ ]:
```
record
```

In [ ]:
```
secuencias= []
for rec in SeqIO.parse(open("Flima_alineamiento_genban.
fasta"), "fasta"):
    linea =SeqRecord(rec.seq, id=gespecie(rec.descripti
on[11:])
                             , description=rec.id)

    print("%s %s" % (linea.id, linea.description[:50]))
    secuencias.append(linea)
```

La secuencia de *F. lima* estaba en reverso complementario, por lo que se agregó el reverso complementario a `secuencias`

In [ ]:
```
linea =SeqRecord(record.seq, id=record.id, description=
record.description)
secuencias.append(linea)
```

In [ ]:
```
SeqIO.write(secuencias, 'Flima_alineamiento.fasta', 'fa
sta')
```

Se desea hacer el alineamiento aquí, pero no se cuenta con el clustalw, por lo que no se realizará.

In [ ]:
```
ls ~/analisis/scripts
```

In [ ]:
```
clustalw_exe = r"~/analisis/scripts/clustalw2"
clustalw_cline = ClustalwCommandline(clustalw_exe, infi
le="Flima_alineamiento.fasta")
assert os.path.isfile(clustalw_exe), "Clustal W executa
ble missing"
stdout, stderr = clustalw_cline()
```

Verificar que está el archivo con la secuencia de alneamiento

In [ ]:
```
ls *.aln
```

Visualización de las secuencias alineadas

In [ ]:
```
alignments = AlignIO.parse("Flima_alineamiento.aln", "c
lustal")
for alignment in alignments:
    print(alignment)
    print("")
```

Visualización del árbol con caracteres ASCII

In [ ]:
```
# para visualizar el árbol generado en formato ascii, s
e ve el contenido del archivo .dnd
tree = Phylo.read("P1a_alineamiento.dnd", "newick")
Phylo.draw_ascii(tree, file=None, column_width=80)
```

Visualización del árbol dibujado

In [ ]:
```
tree.rooted = True
Phylo.draw(tree)
```

In [ ]:
```
f_agenoma.head(2)
```

In [ ]:
```
n, n1 = 0, 0

for row in f_agenoma["sseqid"]:
    row2=f_agenoma.loc[f_agenoma['sseqid']==row]['sseqi
d'].values[0]

    if row2!=row:
        print (row2,row)
        n+=1
        break
```

In [ ]:
```
row2 = f_agenoma.loc[f_agenoma['sseqid']=='NW_012224436
.1']
row2
```

In [ ]:
```
# Numero de contigs que alinearon con el conting en don
de está el 18S de Fundulus heteroclitus
n=0
for row in row2.index:
    n+=1
print(n)
```

In [ ]:
```
ftsv.head(2)
```

Microsatelites

In [ ]:
```
cd ~/data/Sample_13Fundulus/fundulus_micros/
```

In [ ]:
```
ls
```

In [ ]:
```
!head "msatcommander.microsatellites.csv"
```

In [ ]:
```
f_micros = pd.read_csv("msatcommander.microsatellites.c
sv", engine="python")
f_micros.rename(columns = {'id': 'msats_id'}, inplace=T
rue)
f_micros.head(2)
```

In [ ]:
```
f_micros["tipo"] = f_micros["motif"].str.len()
f_micros.head(2)
```

In [ ]:
```
micros = []
micro = 0
n = 0
for rows in f_micros["motif"]:
    micro = len(rows)
    #print (rows,micro)
    n+=1
    micros.append(micro)
    #if n==10:
    #    break

#f.head(2)
```

In [ ]:
```
micros = DataFrame(micros, columns=["tipo"])
micros.describe().round(2)
```

In [ ]:
```
f_micros["tipo"]=micros["tipo"]
f_micros.head(2)
```

In [ ]:
```
df1 = f_micros.groupby("tipo")["name"].count()
df1
```

In [ ]:
```
# debe ser el nombre del archivo fasta
f
```

In [ ]:
```
# en caso contrario ejecutar
f = '13_Fundulus1.fa'
```

In [ ]:
```
plt.pie(df1,                # data
        explode=(0, 0, 0, 0, 0.5),    # offset paramete
rs
        labels= ("di", "tri", "tetra","penta", "hexa"),
#df1.index,      # slice labels
        labeldistance=1.1,
        pctdistance = 0.8, #(0., 0., 0., 0., 0., 0.5),
        colors=('b', 'g', 'r', 'c', 'm'),      # array
of colours: colors=('b', 'g', 'r', 'c', 'm', 'y', 'k',
'w')
        autopct='%1.2f%%',  # print the values inside t
he wedges
        #shadow=True,         # enable shadow
        startangle=270       # starting angle
        )
yes = input("save figure (y/)? ")
if yes.lower()=="y":
    archivo = f[:f.find(".")]+'msat_pie.png'
    plt.savefig(archivo, dpi=400, bbox_inches='tight')
plt.show()
```

In [ ]:
```
fig, ax = plt.subplots()

# Example data
nombres = ('Di', 'Tri', 'Tetra', 'Penta', 'Hexa')
y_pos = (1, 2, 3, 4, 5) #np.arange(len(people))
#valores = df1 #3 + 10 * np.random.rand(len(people))
ax.barh(y_pos, df1,  align='center',
        color=list('rybg'), ecolor='black') #xerr=error
,
ax.set_yticks(y_pos)
ax.set_yticklabels(nombres)
ax.invert_yaxis()  # labels read top-to-bottom
ax.set_xlabel('Number')
ax.set_title('Microsatellites of $\it{Fundulus}$ $\it{l
ima}$')
yes = input("save figure (y/)? ")
if yes.lower()=="y":
    archivo = f[:f.find(".")]+'msat_histo.png'
    plt.savefig(archivo, dpi=400, bbox_inches='tight')

plt.show()
```

In [ ]:
```
fig, ax = plt.subplots()

# Example data
nombres = ('Di', 'Tri', 'Tetra', 'Penta', 'Hexa')
y_pos = (1, 2, 3, 4, 5) #np.arange(len(people))
#valores = df1 #3 + 10 * np.random.rand(len(people))
ax.bar(y_pos, df1,  align='center',
        color=list('rybg'), ecolor='black') #xerr=error
,
ax.set_xticks(y_pos)
ax.set_xticklabels(nombres)
#ax.invert_yaxis()  # labels read top-to-bottom
ax.set_xlabel('Number')
ax.set_title('Microsatellites of $\it{Fundulus}$ $\it{l
ima}$')
yes = input("save figure (y/)? ")
if yes.lower()=="y":
    archivo = f[:f.find(".")]+'msat_histo_v.png'
    plt.savefig(archivo, dpi=400, bbox_inches='tight')

plt.show()
```

In [ ]:
```
f_primers1 = open ("msatcommander.primers.csv", 'r')
f_out = open ("msatcommander_primers.csv",'w')
n=0
for line in f_primers1.readlines():
    if n !=1404 and n !=1405:#line !=",,,,,,,,,,,,,,,,
,,,,,,,,,,":
        line1 = line
        f_out.write(line1)

    #else:
    #    print(n, line)
    n=n+1

f_out.close()
f_primers1.close()
print (str(n)+" sequences processed")
```

In [ ]:
```
f_primers = pd.read_csv("msatcommander_primers.csv", nr
ows = 1093,
                        engine="python")
f_primers.head(2)
```

In [ ]:
```
len(f_primers.name)
```

In [ ]:
```
f_primers["tipo"]=0

#secuencias = []
#linea=[]
n, n1 = 0, 0
for rows in f_primers.index:
    row1 = f_primers.loc[rows]
    row = row1["msats_id"]
    rowname = row1["name"]

    if rowname=="Potentially duplicated primers:":
        print ("\n", "\n", rowname)
        f_primers["duplicate"][rows] =1
        continue
    elif rowname=="" or rowname==" " :
        #print ("\nespacio \n")
        f_primers["duplicate"][rows] =1
        continue
    else:
        row2_1 = f_primers.loc[f_primers['msats_id']==r
ow]
        rowmicro = str(row2_1["msats_id"].values[0])
        row2 = str(row2_1["name"] )
        tipo = int(row2_1["tipo"].values[0])
        f_primers["tipo"][rows] = tipo
        print(row, "\t", rowmicro, "\t", row2_1["msats_
id"].values[0], "\t", row2_1["motif"].values[0], "\t",
row1["name"],
              "\t", row1["tipo"], "\t", row2_1["tipo"].
values[0], "\t", f_primers["tipo"][rows] )
        #linea =  (f_primers[rows], ignore_index = True
)
        #secuencias.append(f_primers[rows], ignore_inde
x =True)
```