Bitacora: obtención de lecturas de buena calidad de la secuenciación masiva y ensamblaje con SOAPdenovo

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

En esta bitácora se utilizarán diferentes programas

FastQC

Se puede encontrar más información en el sitio de fastqc (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/)

y de Trim Galore (https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/)

Ensamblaje con Soapdenovo2

Luo, R. et al. 2012. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. Gigascience 1, 18. doi: 10.1186/2047-217X-1-18 (https://academic.oup.com/gigascience/article/1/1/2047-217X-1-18/2656146)
SOAPdenovo2 manual (https://github.com/aquaskyline/SOAPdenovo2)

Todos estos programas se pueden instalar en sistemas Linux o iOS desde anaconda (https://anaconda.org/bioconda/soapdenovo2)

In [ ]:
```
import os
from Bio import SeqIO, Entrez
```

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

In [ ]:
```
ls
```

In [ ]:
```
%%bash
grep "^@" ../8_S356_L001_R1_001.fastq | wc -l
grep "^@" ../8_S356_L001_R2_001.fastq | wc -l
```

In [ ]:
```
%%bash
grep "^@" ../CA2_S27_L001_R1_001.fastq | wc -l
grep "^@" ../CA2_S27_L001_R2_001.fastq | wc -l
grep "^@" ../CN2_S28_L001_R1_001.fastq | wc -l
grep "^@" ../CN2_S28_L001_R2_001.fastq | wc -l
grep "^@" ../MA2_S25_L001_R1_001.fastq | wc -l
grep "^@" ../MA2_S25_L001_R2_001.fastq | wc -l
grep "^@" ../MN2_S26_L001_R1_001.fastq | wc -l
grep "^@" ../MN2_S26_L001_R2_001.fastq | wc -l
grep "^@" ../XA2_S29_L001_R1_001.fastq | wc -l
grep "^@" ../XA2_S29_L001_R2_001.fastq | wc -l
grep "^@" ../XN2_S30_L001_R1_001.fastq | wc -l
grep "^@" ../XN2_S30_L001_R2_001.fastq | wc -l
```

Número de secuencias obtenidas:

archivo	secuencias
8_S356_L001_R1_001.fastq	6660
8_S356_L001_R2_001.fastq	6674

In [ ]:
```
%%bash
time fastqc 8_S356_L001_R1_001.fastq
time fastqc 8_S356_L001_R2_001.fastq
```

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

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

In [ ]:
```
cd fastqc/
```

In [ ]:
```
%%bash
time trim_galore --paired ../8_S356_L001_R1_001.fastq .
./8_S356_L001_R2_001.fastq
```

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

In [ ]:
```
!head -80 8_S356_L001_R1_001.fastq_trimming_report.txt
```

In [ ]:
```
!head -80 8_S356_L001_R2_001.fastq_trimming_report.txt
```

In [ ]:
```
%%bash
grep "^@" 8_S356_L001_R1_001_val_1.fq | wc -l
grep "^@" 8_S356_L001_R2_001_val_2.fq | wc -l
```

Número de secuencias obtenidas:

archivo	secuencias	archivo-trim	recortadas
8_S356_L001_R1_001.fastq	6660	8_S356_L001_R1_001_val_1.fq	6650
8_S356_L001_R2_001.fastq	6674	8_S356_L001_R2_001_val_2.fq	6664

Tomado del manual de Biopython (http://biopython.org/DIST/docs/tutorial/Tutorial.html#htoc292)

In [ ]:
```
def cuentasecuencias(arch1):
    count, total_len, maximo = 0, 0, 0,
    minimo = 100
    for rec in SeqIO.parse(arch1, "fastq"):
        count += 1
        total_len += len(rec.seq)
        if maximo< len(rec.seq):
            maximo = len(rec.seq)
        if minimo > len(rec.seq):
            minimo = len(rec.seq)
    return (count, total_len, maximo, minimo)
```

In [ ]:
```
count1, count2, count3, count4 = 0, 0, 0, 0
total_len1, total_len2, total_len3, total_len4, = 0, 0,
0, 0
minimo1, minimo2, minimo3, minimo4 = 100, 100, 100, 100
maximo1, maximo2, maximo3, maximo4 = 0, 0, 0, 0
# cuenta secuencias en el primer archivo
archivo1 = "../8_S356_L001_R1_001.fastq"
count1, total_len1, maximo1, minimo1 = cuentasecuencias
(archivo1)
archivo2 = "8_S356_L001_R1_001_val_1.fq"
count2, total_len2, maximo2, minimo2 = cuentasecuencias
(archivo2)
archivo3 = "../8_S356_L001_R2_001.fastq"
count3, total_len3, maximo3, minimo3 = cuentasecuencias
(archivo3)
archivo4 = "8_S356_L001_R2_001_val_2.fq"
count4, total_len4, maximo4, minimo4 = cuentasecuencias
(archivo4)

print ("archivo                \t#sec \tmin max \t
archivo-trim\t\t\t #recor\tmin max")
print(archivo1[:17], "\t", count1, "\t", minimo1, maxim
o1, "\t", archivo2, "\t", count2, "\t", minimo2, maximo
2)
print(archivo3[:17], "\t", count3, "\t", minimo3, maxim
o3, "\t", archivo4, "\t", count4, "\t", minimo4, maximo
4)
```

arc = "archivo" contar = "count" tl = "total_len" minimo = "minimo" maximo = "maximo" n = 1 arc1 = ["../8_S356_L001_R1_001.fastq", "8_S356_L001_R1_001_val_1.fq", "../8_S356_L001_R2_001.fastq", "8_S356_L001_R2_001_val_2.fq"] for i in arc1: arc = "archivo"+str(n) contar = "count"+str(n) tl = "total_len"+str(n) minimo = "minimo"+str(n) maximo = "maximo"+str(n) count1, count2, count3, count4 = 0, 0, 0, 0 total_len1, total_len2, total_len3, total_len4, = 0, 0, 0, 0 minimo1, minimo2, minimo3, minimo4 = 100, 100, 100, 100 maximo1, maximo2, maximo3, maximo4 = 0, 0, 0, 0 # cuenta secuencias en el primer archivo archivo1 = "../8_S356_L001_R1_001.fastq" count1, total_len1, maximo1, minimo1 = cuentasecuencias(archivo1) archivo2 = "8_S356_L001_R1_001_val_1.fq" count2, total_len2, maximo2, minimo2 = cuentasecuencias(archivo2) archivo3 = "../8_S356_L001_R2_001.fastq" count3, total_len3, maximo3, minimo3 = cuentasecuencias(archivo3) archivo4 = "8_S356_L001_R2_001_val_2.fq" count4, total_len4,

maximo4, minimo4 = cuentasecuencias(archivo4) print ("archivo \t#sec \tmin max \t archivo-trim\t\t\t #recor\tmin max") print(archivo1[:17], "\t", count1, "\t", minimo1, maximo1, "\t", archivo2, "\t", count2, "\t", minimo2, maximo2) print(archivo3[:17], "\t", count3, "\t", minimo3, maximo3, "\t", archivo4, "\t", count4, "\t", minimo4, maximo4)

Archivo de configuración para el ensamblador Soapdenovo

En caso de ser necesario, cambiar el valor de `max_rd_len=`, en este caso a **251**
Revisar los demás parámetros, p. e. `rd_len_cutoff=`
Cambie las líneas q1 y q2, en este caso a:

```
q1=8_S356_L001_R1_001_val_1.fq
q2=8_S356_L001_R2_001_val_2.fq
```

Leer el manual (https://vcru.wisc.edu/simonlab/bioinformatics/programs/soap/SOAPdenovo2MANUAL.txt
Asimismo es necesario tomar en cuenta el nombre del archivo de la configuración. En este caso:

```
config_soap_8_S356.txt
```

In [ ]:
```
fout = open("config_soap_8_S356.txt", "w")
linea="""#maximal read length
max_rd_len=251
[LIB]
#average insert size of the library
avg_ins=300
#if sequences are forward-reverse of reverse-forward
reverse_seq=0
#in which part(s) the reads are used (only contigs, onl
y scaffolds, both contigs and scaffolds, only gap closu
re)
asm_flags=3
#cut the reads to the given length
rd_len_cutoff=300
#in which order the reads are used while scaffolding
rank=1
# cutoff of pair number for a reliable connection (at l
east 3 for short insert size)
pair_num_cutoff=3
#minimum aligned length to contigs for a reliable read
location (at least 32 for short insert size)
map_len=32
#paired-end fastq files, read 1 file should always be f
ollowed by read 2 file
q1=8_S356_L001_R1_001_val_1.fq
q2=8_S356_L001_R2_001_val_2.fq
#another pair of paired-end fastq files, read 1 file sh
ould always be followed by read 2 file
#q1=input_reads2_pair_1.fq
#q2=input_reads2_pair_2.fq
#paired-end fasta files, read 1 file should always be f
ollowed by read 2 file
#f1= SRX5014491_1_val_1.fq
#f2=SRX5014491_2_val_2.fq
#fastq file for single reads
#q=input_reads.fq
"""

fout.write(linea)

fout.close()
```

Verificando contenido del archivo de configuración

In [ ]:
```
!head -50 config_soap_8_S356.txt
```

In [ ]:
```
ls
```

Archivo para ejecutar sh.

Se debe cambiar el nombre del archivo generado, en este caso a:

```
open("soapdenovo2.submit", "w")
```

In [ ]:
```
fout = open("soapdenovo2.submit", "w")
linea="""#!/bin/sh
#SBATCH --job-name=SOAPdenovo2
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=8
#SBATCH --time=168:00:00
#SBATCH --mem=50gb
#SBATCH --output=SOAPdenovo2.%J.out
#SBATCH --error=SOAPdenovo2.%J.err

#module load soapdenovo2/r240

SOAPdenovo-63mer all -s config_soap_8_S356.txt -K 31 -o
output_directory/output31 -p $SLURM_NTASKS_PER_NODE


"""

fout.write(linea)

fout.close()
```

Se verifica el contenido del archivo

In [ ]:
```
!head -20 soapdenovo2.submit
```

Se crea el directorio de salida, en caso de no existir.

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

Se ejecuta el programa

In [ ]:
```
!time sh soapdenovo2.submit
```

Se verifican los archivos de salida y su contenido

In [ ]:
```
ls output_directory/
```

Escriba en la siguiente celda qué contiene cada archivo.

Verificando el contenido de *.contig

In [ ]:
```
!head output_directory/output31.contig
```

Contando el número de contigs

In [ ]:
```
!grep "^>" output_directory/output31.contig |wc -l
```

Contando el número de secuencias en montaje (scaffolding)

In [ ]:
```
!grep "^>" output_directory/output31.scafSeq |wc -l
```

Estadística

In [ ]:
```
!head -100 output_directory/output31.scafStatistics
```

Una vez concluido el ensamblaje se procede a la anotación de los contigs