Universidad Nacional Autónoma de México

Licenciatura en Ciencias Agrogenómicas

BIOINFORMÁTICA 2

BITÁCORA DE PROYECTO

González Quevedo Cassandra

Profesor: Alejandra Rougon

Fecha de entrega:06 /Diciembre/ 2021

CONTENIDO BIOINFORMÁTICO

Al ingresar a GAIA me traslade a la dirección /projects/cassgonzalez

Posteriormente se creó una carpeta llamada “Intento2” en donde se seleccionó el organismo *Mycoplasma tuberculosis* y se descargó una secuencia de SRR, en la cual se aplicó el comando

```
fastq-dump --split-3 ERR5948382
```

Esto debido a que eran secuencias pareadas y se necesitaba que estuvieran en archivos separados

1) **Control de calidad**

A dichos archivos descargados se les aplicó un control de calidad con el comando

```
fastqc ERR5948382_1.fastq
```

```
fastqc ERR5948382_2.fastq
```

Al terminar de aplicar el control de calidad se crearon unos archivos HTML los cuales se descargaron con el comando

```
scp -P 10022 cassgonzalez@132.247.172.26:/projects/cassgonzalez/Intento2/ERR5948382_1_fastqc.html .
```

```
scp -P 10022 cassgonzalez@132.247.172.26:/projects/cassgonzalez/Intento2/ERR5948382_2_fastqc.html .
```

En estos archivos se pudo analizar la calidad que tenían.

2) **Filtrado de secuencias**

Para realizar el filtrado además de los parámetros básicos se le agregó uno un poco más especifico para tener un mejor filtrado. Se uso el comando

```
fastp -i ERR5948382_1.fastq -I ERR5948382_2.fastq -o Filtrado_1.fastq -O Filtrado_2.fastq >&  Filtrado.log
```

Cuando terminó de correr el código anterior se aplico

```
Wc -l *.fastq
```

Esto se hizo con la finalidad de hacer una comparación entre los archivos originales y ver que tanto mejoraron con el filtrado que se les aplicó

3) Cálculo de cobertura

Despues de haber realizado el control de calidad nos podemos dar cuenta en los reportes generados en HTML que el total de las secuencias es de 1301991 y la longitud de cada una es de 301 nucleotidos. Con eso podemos saber que la cobertura total es de

```
1301991*301 = 391,899,291
```

4) Mapeo (si existe referencia)

Para poder iniciar el mapeo fue necesario empezar con descargar un genoma de referencia desde el NCBI. En este caso se escogió "*Mycoplasma genitalium G37*".

Se subío a GAIA con el comando

```
scp -P 10022 /root/sandbox/Ref.fa cassgonzalez@132.247.172.26:/projects/cassgonzalez
```

Posteriormente se debe realizar un índice, el cual se hizo con

```
bowtie2-build Ref.fa RefParaERDB
```

Ya teniendo nuestro índice se corrió el mapeo con

```
bowtie2 --maxins 1000 -x RefParaERDB -1 /projects/cassgonzalez/Intento2/Filtrado_1.fastq -2
/projects/cassgonzalez/Intento2/Filtrado_2.fastq -S Mapeo3Proyecto.sam
```

Ahora para poder descargar el archivo a nuestra computadora no era tan sencillo ya que era muy pesado por lo que fue necesario eliminar las secuencias que no se mapearon y para esto se uso

```
awk '$3!="*"' Mapeo3Proyecto.sam >MapeoListo.sam
```

Ahora si ya es mas sencillo descargar el archivo con

```
scp -P 10022 cassgonzalez@132.247.172.26:/projects/cassgonzalez/MapeoListo.sam .
```

5) Ensamble de genoma

Para realizar el ensamble del genoma primero es necesario activar el environment con

```
Conda actívate spades
```

Se utilizó spades debido a que la tecnología que se utilizó para mi secuencia fue Illumina

Para correr el ensamble se utilizó

```
spades.py -k 33,37,41 -t 1 -m 7 --pe1-1 /projects/cassgonzalez/Intento2/Filtrado_1.fastq --pe1-2 /projects/cassgonzalez/Intento2/Filtrado_2.fastq -o spades2_ER
```

Después de que se corrió el ensamble, se comprueba la calidad con

```
quast --split-scaffolds -t 1 /projects/cassgonzalez/spades2_ER/scaffolds.fasta
```

Y también se comprobó la calidad con la referencia, pero para esto fue necesario moverse a la carpeta que se generó con el código anterior “quast results”. Dentro de la carpeta se aplicó el siguiente código

```
quast.py --split-scaffolds -t 1 -r ../Ref.fa /projects/cassgonzalez/spades2_ER/scaffolds.fasta
```

Este ultimo código nos generó una carpeta que contiene dos archivos HTML que son los que más no interesa visualizar. Para eso vamos a descargarlos con

```
scp -P 10022 cassgonzalez@132.247.172.26:/projects/cassgonzalez/quast_results/results_2021_12_05_20_09_50/report.html .
```

```
scp -P 10022 cassgonzalez@132.247.172.26:/projects/cassgonzalez/quast_results/results_2021_12_05_20_09_50/icarus.html .
```

6) Anotación (de una porción)

Ya que se tiene el archivo fasta que se generó del ensamble entonces primero se necesita ver cuantos scaffolds se tienen y eso se hace con

Grep ‘>’ -c scaffolds.fasta

Con eso nos damos cuenta que se tienen 469

Ahora para poder seleccionar solo una porción o bien solo un scaffold es necesario primero convertir nuestro archivo scaffolds.fasta a un formato tabular, eso lo hacemos con la ayuda de la siguiente pagina

http://archive.sysbio.harvard.edu/CSB/resources/computational/scriptome/UNIX/Tools/Change.html

Para convertir en tabular se usó el código:

```
perl -e ' $count=0; $len=0; while(<>) { s/\r?\n//; s/\t/ /g; if (s/^>//) { if ($. != 1) { print "\n" } s/ |$/\t/; $count++; $_ .= "\t"; } else { s/ //g; $len += length($_) } print $_; } print "\n"; warn "\nConverted $count FASTA records in $. lines to tabular format\nTotal sequence length: $len\n\n"; ' scaffolds.fasta > scaffolds.tab
```

Posteriormente se debe saber cual es la longitud de los scaffolds y esa esta dada por le última columna del archivo que se generó del código pasado, por lo que se usó el siguiente código para saberlo

```
perl -e ' $col=-1; while (<>) { s/\r?\n//; @F = split /\t/, $_; $len = length($F[$col]); print "$_\t$len\n" } warn "\nAdded column with length of column $col for $. lines.\n\n"; ' scaffolds.tab > seqs_length.tab
```

Ahora podemos visualizar el archivo con

```
less seqs_length.tab
```

Como el archivo es muy grande no se puede visualizar muy bien por lo fue necesario cortar la última columna con

```
Cut -f 4 seqs_length.tab > OnlyLength.list
```

Visualizamos la columna con

```
less OnlyLength.list
```

Y en esta columna solo se presenta el tamaño de los diferentes scaffolds que se generaron, de esta manera se seleccionó el primero que tenía una longitud de 316434

Ahora para poder seleccionar solo ese scaffold de todo el documento primero fue necesario saber cual era el nombre del scaffold y eso se realizó con

```
head -n 1 seqs_length.tab | cut -f 1
```

De esta manera el nombre fue NODE_1_length_316434_cov_797.810397

Antes de cortar el scaffold seleccionado se creó un archivo para mandar ahí la información del scaffold. Esta se creó con

```
Cat > CassG_1.list
```

Y en la primera línea se escribió el nombre del scaffold “NODE_1_length_316434_cov_797.810397”

Para separar el scaffold seleccionado se utilizó

```
perl -e ' ($id,$fasta)=@ARGV; open(ID,$id); while (<ID>) { s/\r?\n//; /^>?(\S+)/; $ids{$1}++; } $num_ids = keys %ids; open(F, $fasta); $s_read = $s_wrote = $print_it = 0; while (<F>) { if (/^>(\S+)/) { $s_read++; if ($ids{$1}) { $s_wrote++; $print_it = 1; delete $ids{$1} } else { $print_it = 0 } }; if ($print_it) { print $_ } }; END { warn "Searched $s_read FASTA records.\nFound $s_wrote IDs out of $num_ids in the ID list.\n" } ' CassG_1.list scaffolds.fasta > Scaffold1.fna
```

Finalmente descargamos ese archivo que se generó con el siguiente código

```
scp -P 10022 cassgonzalez@132.247.172.26:/projects/cassgonzalez/spades2_ER/Scaffold1.fna .
```

Para poder hacer la anotación se hizo uso de la herramienta de Augustus

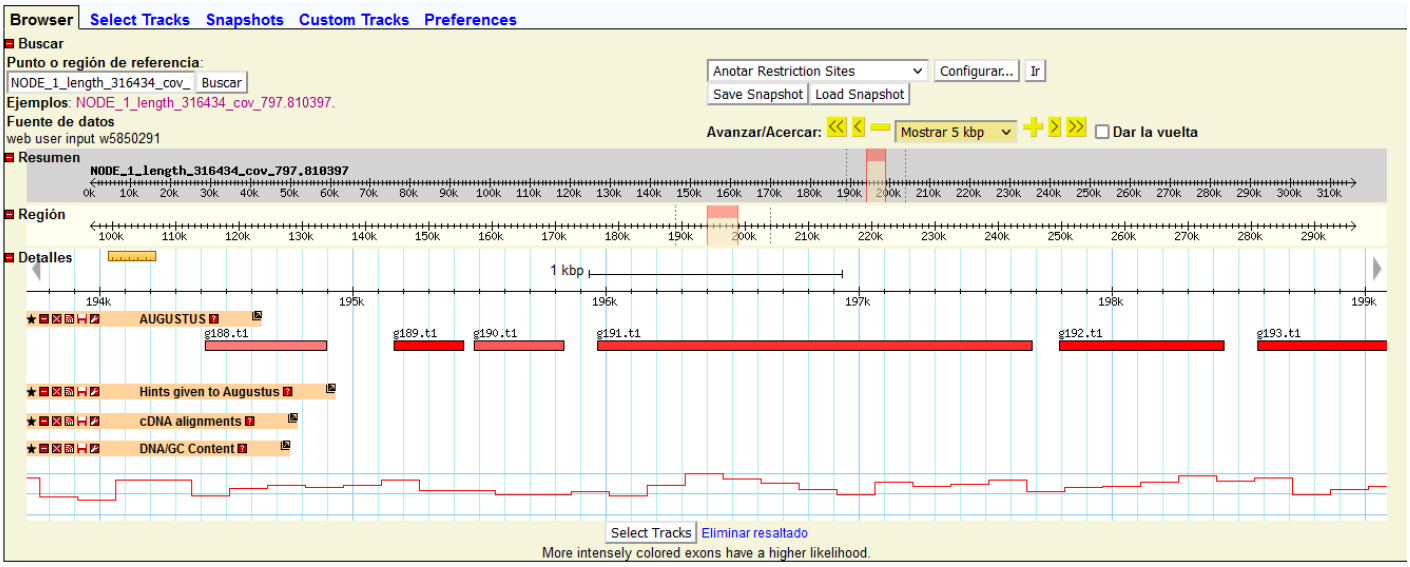
http://bioinf.uni-greifswald.de/augustus/submission.php

En esta pagina además de subir nuestra secuencia se seleccionó el organismo de *Escherichia coli* ya que era el organismo más cercano (por ser una bacteria) para el que hay un algoritmo en esta plataforma y los demás parámetros se quedaron por default. Al finalizar nos generó varios archivos con resultados en texto plano y también unos gráficos sobre los genes que se encontraron

http://bioinf.uni-greifswald.de/augustus/cabinet?folder=AUG-759926626

El primero que se presenta es el grafico en donde se muestran algunos de los genes

http://bioinf.uni-greifswald.de/gb2/gbrowse/w5850291/?name=NODE_1_length_316434_cov_797.810397



El siguiente es el archivo de entrada que se proporcionó con la secuencia del primero scaffold

Unidad León Escuela Nacional de Estudios Superiores

CIENCIAS AGROGENÓMICAS

```
>NODE_1_length_316434_cov_797.810397
GAGTTGCAAGTGTTAGTTATTTTTTAAAAAATCAAAAGAAAGGAGGATTTGACCATTAAA
AAAGTAAAGGTTACAGAGAAAGCTTCTGTTGAAAGAATGATAGAAGTATTAAGTGATTCT
GAAATTTTATCTTATTTATTAAGTAATAAATACACTGCTGATGGAGTTAATGAAATTTAT
GTAACGATGAACATTAAAGACAAAGATTCAAACACTCAAAGGAAGCCAGATAAAATGTCT
TTAGAGCTTGAATTGTTTTTAAATCAGTTTATTGAATTCTTAAAAGAGCAGAGAAAAAGC
TAGCAGCATAAAAATACCAGCAGTTAATGCATAGCTGTTGGTATTTTTACTTAAGTTTTG
ACCATTTTGAATGGATAAAATTTCCTTGCAAATAGAAATTACAAACCAAAAACTATAACA
AAATGAACTAAGCAATTTTCATCTTGGCTTTTTTCTTATTTAAAAGAGATTAAATTTTCA
TCTTAAAAGCTTTTATTTTTACCACACAGGATCTAATAAATTTCAAGAGTTAAACAAGTA
CAAACTTTGTTAATATTTCTCCAAATTTCGGGTTTTTAGTGTTTATGAAATTTAAGTATG
GTGCCATCTTTTTTAGTGGCTTTCTAGGTCTTTCTGCCATTTTGGCTGCCTGTGGTACAA
AGGGTAAATTTGATCAAGTTGATGATGGCAAGATTAAATTGGCTTCCTCTTTAACTTCTA
AAAGTGCATCAAAAGCCTTACAAGCAATTGTTAAAAAATATAACGAAGTTAAAAAACCTG
GTGATTACCCTATTGAAATTACCCAAATCGCTGGTGGTTATGATGGTGGTCGTAGCGATT
TACAAACCCGAGTCAACGTTAAAGACACCACTAACTTTTACAACTTAATCTTAAACTACC
CTGATCTAGTTTCAACTTTAGGTCGTGTTGGTATGGAACTGCCGTTTGACAATGTAAAGG
TTGACAAACTATCACCCCGTTTTTTAGATTTCAACAACCGCATTAGTGCAATTTCTAAAC
```

El siguiente son los resultados en texto plano

```
#
# ----- prediction on sequence number 1 (length = 316434, name = NODE_1_length_316434_cov_797.810397) -----
#
# Predicted genes for sequence number 1 on both strands
# start gene g1
NODE_1_length_316434_cov_797.810397     AUGUSTUS        gene    585     1388    0.82    +       .       g1
NODE_1_length_316434_cov_797.810397     AUGUSTUS        transcript      585     1388    0.82    +       .       g1.t1
NODE_1_length_316434_cov_797.810397     AUGUSTUS        start_codon     585     587     .       +       0       transcript_id "g1.t1"; gene_id "g1";
NODE_1_length_316434_cov_797.810397     AUGUSTUS        single  585     1388    0.82    +       0       transcript_id "g1.t1"; gene_id "g1";
NODE_1_length_316434_cov_797.810397     AUGUSTUS        CDS     585     1388    0.82    +       0       transcript_id "g1.t1"; gene_id "g1";
NODE_1_length_316434_cov_797.810397     AUGUSTUS        stop_codon      1386    1388    .       +       0       transcript_id "g1.t1"; gene_id "g1";
# coding sequence = [atgaaatttaagtatggtgccatctttttttagtggctttctaggtctttctgccattttggctgcctgtggtacaaagg
# gtaaatttgatcaagttgatgatggcaagattaaattggcttcctcttttaacttctaaaagtgcatcaaaagccttacaagcaattgttaaaaaatat
# aacgaagttaaaaaaacctggtgattaccctattgaaattacccaaatcgctggtggttatgatggtggtcgtagcgatttacaaacccgagtcaacgt
# taaagacaccactaacttttacaacttaatcttaaactaccctgatctagtttcaacttttaggtcgtgttggtatggaactgccgtttgacaatgtaa
# aggttgacaaactatcaccccgtttttttagatttcaacaaccgcattagtgcaatttctaaaccaggaatttacggtattccgggtttctttatccacc
# gaagttttatcgatcaacggaccggtgttgcactatattttgaacaacgctaaaaagaaagaaggtactttaaaccaaaaaatgacaagttcttctgta
# aggaaaaaacagcagtggcactttaacagtagcaactgatactgaaacaagtagtttatggaaaaagatagaagattctgcaaaagctaacggtaaaa
# gcgatgaaaaaggaaaaggtaagaagaaagataataaaagtgcaacttttcgcttgtacaactaaaacaaactcaagaaaaaacagatgattcccaa
# gacactaaaaatagtgatgatcaagttaaaaaatcttga]
# protein sequence = [MKFKYGAIFFSGFLGLSAILAACGTKGKFDQVDDGKIKLASSLTSKSASKALQAIVKKYNEVKKPGDYPIEITQIAGG
# YDGGRSDLQTRVNVKDTTNFYNLILNYPDLVSTLGRVGMELPFDNVKVDKLSPRFLDFNNRISAISKPGIYGIPVSLSTEVLSINGPVLHYILNNAKK
# KEGTLNQKMTSSSEGKNSSGTLTVATDTETSSLWKKIEDSAKANGKSDEKGKGKKKDNKSATFSLVQLKQTQEKTDDSQDTKNSDDQVKKS]
# end gene g1
***
```

Después se presenta la predicción de secuencias de aminoácidos

```
>NODE_1_length_316434_cov_797.810397:g1.t1
MKFKYGAIFFSGFLGLSAILAACGTKGKFDQVDDGKIKLASSLTSKSASKALQAIVKKYNEVKKPGDYPIEITQIAGG
YDGGRSDLQTRVNVKDTTNFYNLILNYPDLVSTLGRVGMELPFDNVKVDKLSPRFLDFNNRISAISKPGIYGIPVSLSTEVLSINGPVLHYILNNAKK
KEGTLNQKMTSSSEGKNSSGTLTVATDTETSSLWKKIEDSAKANGKSDEKGKGKKKDNKSATFSLVQLKQTQEKTDDSQDTKNSDDQVKKS
>NODE_1_length_316434_cov_797.810397:g2.t1
MFTSVFAAGGDYNNFFYKIENGRADFSNFKNKGTSYQNLQKVFGDFKGLIDKNGIFVNKGGSYSSNFQKFHQLAYSI
SSTSGFFYSFAGKSAKRLNFGDSFIEYPRFTQEIKAPSKNGENGQTNEGNSTNGEQNLLGTFEVKDDSKPKEEVKSNKNSGKESSQNQGKKSNNNKTI
YLYETKIPDGKTAGDNAILIKDKNVIEKLKSAAKEENKEQTAEATKAAITSNKAKSTKKESSKVIGYTTTDSVREDGKNIFAIDRVNGENYDRKIIVG
AKAETLNQSSTLQSEEAIVLPAPGKYLNGDPKKVTITQGPNIIGIHANEKENAETQKFVD
>NODE_1_length_316434_cov_797.810397:g3.t1
MVAEISLDFPPLEVQEKIAHFLKSFNELSSQLKAELIKRQKQYAFYSDYLLNPKHSQGEEYKLFKLKDIAKKILVGGE
KPSDFQKEKDQVYKYPILSNSRKADDFLGYSKTFRIAEKSITVSARGTIGAVFYRDFSYLPAVSLICFIPKPEFNINFLFHALKATKFHKQGSGTGQL
TMAQFKEYQVYIPSLKKQQEIAATLDPLYYIFANSN
```

Y finalmente se presenta la predicción de secuencias codificantes

>NODE_1_length_316434_cov_797.810397:g1.t1
atgaaatttaagtatggtgccatctttttttagtggctttctaggtctttctgccattttggctgcctgtggtacaaagg
gtaaatttgatcaagttgatgatggcaagattaaattggcttcctctttaacttctaaaagtgcatcaaaagccttacaagcaattgttaaaaaatat
aacgaagttaaaaaacctggtgattaccctattgaaattacccaaatcgctggtggttatgatggtggtcgtagcgatttacaaacccgagtcaacgt
taaagacaccactaacttttacaacttaatcttaaactaccctgatctagtttcaactttaggtcgtgttggtatggaactgccgtttgacaatgtaa
aggttgacaaactatcaccccgtttttttagatttcaacaaccgcattagtgcaatttctaaaccaggaatttacggtattccggtttctttatccacc
gaagttttatcgatcaacggaccggtgttgcactatattttgaacaacgctaaaaagaaagaaggtactttaaaccaaaaaatgacaagttcttctga
aggaaaaaacagcagtggcactttaacagtagcaactgatactgaaacaagtagtttatggaaaaagatagaagattctgcaaaagctaacggtaaaa
gcgatgaaaaaggaaaaggtaagaagaaagataataaaagtgcaacttttcgcttgtacaactaaaacaaactcaagaaaaaacagatgattcccaa
gacactaaaaatagtgatgatcaagttaaaaaatcttga
>NODE_1_length_316434_cov_797.810397:g2.t1
ttgtttacctctgtttttgcagctggcggcggtgattacaataatttcttttacaagattgaaaatggtcgcgctgact
ttagtaactttaagaataaagggacttcttaccaaaatcttcaaaaagtatttggtgatttcaaaggcttaattgacaaaaatggtatctttgtgaat
aagggtggatcttactcatcaaacttccaaaagttccaccaattagcttacagcatttcctctacttccggtttcttctattcgtttgccggtaaaag
tgcaaagcgtttaaatttggtgatagctttattgagtatccgcggtttactcaagaaattaaagcaccatcaaaaaatggagaaaatggccaaacaa
atgaaggaaatagtaccaatggcgaacaaaatctcttaggaactttcgaagtcaaggatgacagtaagcctaaagaagaagttaaaagtaataaaaat
agtggtaaagaaagtagccaaaatcagggtaaaaaatctaacaataacaagactatttacctttatgaaacaaaaattccagatggaaaaactgcagg
tgataatgctatcttaatcaaagataagaacgttatcgaaaagcttaaaagtgcagctaaagaagaaaataaagaacaaactgctgaagcaactaaag
ctgcaattacaagtaataaagcaaaaagcactaaaaaagaaagcagtaaagttatcggttacaccacaactgatagtgttcgcgaagatggcaaaaac
attttcgcaatagatagagttaatggtgaaaactatgaccgtaagattattgtaggtgctaaagcagaaactttaaaccaatctagcaccttacaaag
tgaagaagcgattgttttgcctgcccctggaaagtatctaaatggtgatcccaagaaagtgacaattacccaaggacctaacatcattggcattcatg
ctaacgaaaaggaaaatgcagaaacgcaaaagttcgtagattga