Proyecto Bioinformática II
Ensamble del genoma de *pseudomonas aeruginosa*

Salvador Alejandro Cuevas Villicaña, *Ciencias Agrogenómicas*

***Resumen*—*Pseudomonas aeruginosa* es una bacteria ubicua ambiental, gram-negativa que pertenece a la rama γ de las proteo-bacterias, y es una de las principales causantes de enfermedades oportunistas en animales y plantas; causando en los humanos infecciones en las vías respiratorias, urinarias y tejidos. Esta posición se le debe principalmente a su elevada resistencia a los antibióticos y desinfectantes. Este trabajo presenta el ensamble de un genoma público de *P. aeruginosa* secuenciado mediante la tecnología de Illumina MiSeq, desde como descargar las secuencias hasta la anotación del ensamble de las mismas. Mediante el uso de varias herramientas bioinformaticas**

***Index Terms*—*P. aeruginosa*, bioinformática, genoma, Illumina MiSeq, ensamble, anotación de secuencias**

I. Introducción

La bacteria *Pseudomonas aeruginosa* es una especie de relevancia tanto medica como en actividades agro-productivas. Ya que es la causante de varias enfermedades oportunistas en plantas y animales. Es gram-negativa, aerobia, suele habitar en ecosistemas húmedos, así como en plantas y tejidos vegetales. Las infecciones que suelen afectar al sistema respiratorio, el sistema urinario, y a heridas abiertas causando sepsis. Estas infecciones son imposibles de erradicar, en parte debido a la resistencia natural de la bacteria a los antibióticos, y en última instancia conducen a insuficiencia pulmonar y muerte.

Tiene un genoma significativamente grande que, a comparación de otras bacterias, con un tamaño de 6,3 millones pares de bases (Mbp), la secuenciación y ensamble de su genoma en teoría permitiría encontrar nuevas alternativas para luchas contra sus infecciones.

I-A. Ensamble de genomas

Los archivos obtenidos de una secuenciación contienen millones de fragmentos de ADN que no tienen una utilidad de estudio. Por lo que hace falta ensamblar esos millones de fragmentos en un Genoma completo. Es un proceso complejo que demanda muchas prestaciones computacionales. Existen dos tipos de ensamble de genoma: de novo y ensambles de referencia; los primeros se realizan cuando no se tiene otro genoma ensamblado para poder compararlo y el ensamble de referencia es cuando se realiza hacia un genoma previamente ensamblado. Ambas estrategias presentan sus pros y contras; pero de cualquier forma el procedimiento a seguir es muy similar y consta de los siguientes pasos:

- Control de calidad. Donde se verifica si nuestros archivos de secuenciación son viables para realizar un ensamble y que tan contaminados se encuentran,
- Filtrado. Si vemos que son viable los archivos. Se realiza un filtrado para eliminar adaptadores de secuenciación y secuencias parasitas.
- Calculo de Cobertura. De la secuenciación
- Si existe una referencia previa se puede realizar un mapeo.
- El ensamble del genoma. Existen varias herramientas y el cual usar depende de si se usara una estrategia de novo o de referencia.
- Anotación. Una vez teniendo nuestro genoma ensamblado. Se procede a realizarle la anotación donde se le añade la información biológica relevante; la anotación se divide en estructural y funcional, la estructural es definir la región donde se encuentran los genes en el genoma y la funcional es la función que cumplen estos genes en el organismo.

II. Metodología

Se usaron varias herramientas bioinformáticas, la mayoría de software libre, la mayoría de los cuales se ejecutaron mediante el protocolo SSH en el clúster denominado “GAIA” perteneciente a la Escuela Nacional de Estudios Superior Unidad León de la UNAM, y otro en mi propio ordenador personal con las siguientes prestaciones:

- Procesador: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz
- RAM instalda: 16 GB
- Sistema operativo: Windows 11 Home con WSL2 que corre Ubuntu 20.04 lts

Estudiante de segundo año de la licenciatura en Ciencias Agrogenómicas en la Escuela Nacional de Estudios Superiores Unidad León de la Universidad Nacional Autónoma de México, (e-mail: alejandro.villicana@comunidad.unam.mx)

Así mismo también se usó la base de datos del NCBI

II-1. Descarga de secuencias del NCBI: **Se uso GAIA.** Mediante la herramienta SRAtoolkit https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc previamente instalada en GAIA, se procedió a descargar la secuencia marcada con el indicador SRR17084738, mediante el comando:

```
fastq-dump --split-3 SRR17084738
```

La opción –split-3 nos sirve para bajar secuencias pareadas, como es en el caso de ésta. Con este comando se obtuvieron los archivos SRR17084738_1 y SRR17084738_2 de la secuencia pareada de *P. aeruginosa*

II-2. Control de calidad: **Se uso GAIA.** El primer análisis de control de calidad se realizó mediante FASTQC, https://www.bioinformatics.babraham.ac.uk/projects/fastqc/, previamente instalado en GAIA, el comando que usé es:

```
fastqc SRRSRR17084738*
```

Obteniendo como resultado los reportes html de FASTQC, donde se observó que la muestra estaba contaminada por el adaptador "nextera transposase sequencez un tamaño total de 1,024,419 secuencias. Por lo que el siguiente paso fue eliminar las secuencias de adaptadores, otros adaptadores y secuencias de baja calidad.

II-3. Filtrado de secuencias: **Se uso GAIA.**Para el filtrado de secuencias use FASTP https://github.com/OpenGene/fastp, previamente instalado en GAIA, este se realiza a las secuencias que se obtuvieron en el punto II-1 y mediante el comando:

```
fastp -i SRR17084738_1.fastq -I SRR17084738_2.
    fastq -o filtrado_1.fastq -O filtrado_2.
    fastq --qualified_quality_phred 30
```

La opción elegidas para este comando fue establecer que las secuencias no podían bajar de un nivel de calidad phred 30 mediante –qualified_quality_phred 30; la escala phred nos indica la calidad de nuestra secuenciación mediante la probabilidad de factor de error de cada base, determinada por la ecuación matemática $P = 10^{\frac{-Q}{10}}$, siendo Q el nivel de calidad phred y P la probabilidad de factor de error de base. Por lo que para un nivel calidad 30, la probabilidad de obtener un error es de 1 en 1000 bases. Una vez realizado el filtrado se obtuvieron los archivos filtrado_1.fastq y filtrado_2.fastq ; a los cuales se le realizo nuevamente un FASTQC para comprobar las nuevas calidades. Esto es mediante:

```
fastqc filtrado_*
```

Una vez realizado el nuevo análisis se comprobó que se habían eliminado los adaptadores y que un 74.802107 % de las secuencias habían pasado el filtrado de calidad, es decir 766,287 secuencias que tienen un score phred mayor a 30

II-4. Cálculo de cobertura: **Se uso GAIA.** Para el calculo de cobertura se usaron los siguientes comandos a los archivos filtrado_1.fastq y filtrado_2.fastq

```
cat filtrado_1.fastq | paste - - - - | awk '{
    print$2}' | wc -l
```

Cuyo resultado es 766287, es decir el numero total secuencias.

```
cat filtrado_1.fastq | paste - - - - | awk '{
    print$2}' | wc -c
```

Y su resultado es el numero de saltos de linea. Finalmente el resultado de la cobertura es el restar el segundo valor menos el primero.

$$5300155 - 766287 = 4533868$$

Es decir la cobertura tiene un valor de 4533868

Se obtiene el mismo resultado con el archivo filtrado_2.fastq

II-5. Mapeo: **Se uso GAIA.** Se realizó un mapeo a referencia mediante la herramienta bowtie2, http://bowtie-bio.sourceforge.net/bowtie2/index.shtml , previamente instalada en GAIA, por lo que primero se descargó una secuencia de referencia en el NCBI, eligiendo la que tiene el identificador NC_002516.2 "Pseudomonas aeruginosa PAO1, complete genome.[a]lmacenado locamente como $refgenome.fasta$

Luego se realizó el indice con el comando:

```
bowtie2-build -f refgenome.fasta REFpsea
```

Luego el mapeo se realiza con:

```
bowtie2 -S filtradovsPsea.sam -q --phred33 -p
    4 --fr -x REFpsea -1 filtrado_1.fastq -2
    filtrado_2.fastq >& bowtie_psea.log
```

Obteniendo el archivo de mapeo $filtradovsPsea.sam$ al cual se le eliminaran las secuencias que no mapearon mediante el comando:

```
awk $3!="*" filtradovsPsea.sam >
    filtradovsREF_reducido.sam
```

Obteniendo finalmente el archivo $filtradovsREF\_reducido.sam$

II-6. Ensamble de genomas: **Se uso GAIA.** Para el ensamble del genoma se usó el programa SPAdes, https://github.com/ablab/spades ,previamente instalado en GAIA. El ensamble se realizó con el comando:

```
spades.py -k 21,33,37,41,55,77 -t 4 -m 8 --pe1
    -1 filtrado_1.fastq --pe1-2 filtrado_2.
    fastq -o spades_Psea
```

Las opciones usadas son -k 21,33,37,41,55,77 que indican los valores de tamaños k-mer que se utilizarán, -t 4 que indica el número de hilos, 4 en este caso, -m que indica el número máximo de memoria RAM que puede usar SPAdes, 8GB en este caso y –pe1-1 y –pe1-2 que indica cual es el archivo 1 y el 2 respectivamente del pareo.

Posteriormente se realiza un análisis de calidad del ensamble mediante el archivo scaffolds.fasta generado y es mediante el comando:

```
quast --split-scaffolds -t 1 scaffolds.fasta
```

Posteriormente se realiza una análisis de calidad hacia el genoma de referencia. Mediante el comando:

```
quast.py --split-scaffolds -t 1 -r refgenome.
    fasta scaffolds.fasta
```

II-7. Anotación: **Se uso PC** Una vez teniendo los resultados del ensamble se procede a separar los scaffolds que estan en el archivo scaffolds.fasta para poder realizar la anotación estructural y funcional del mismo. Solo se realizó la preperación del primero porque el porcedimineto es igual para los demas. Y para esto me apoye en la pagina FAS Center dor Systems Biology http://archive.sysbio.harvard.edu/CSB/resources/computational/scriptome/UNIX/Tools/Change.html

Primero se pasa el archivo 'scaffolds.fasta' a formato tabular mediante el comando:

```
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
```

Luego se calcula la longitud de la última columna de 'scaffolds.tab' que es la longitud del scaffolds

```
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
```

Luego se corta esa última columna que nos indica el valor numérico de la longitud del scaffols, mediante el comando:

```
cut -f 4 seqs_length.tab > onlylength.list
```

Luego para ver la longitud de los scaffold que tenemos usamos el comando:

```
cat onlylength.list | more
```

Elegí el primero, el que tiene longitud de 668648

Ahora para ver la nomenclatura de estos scaffolds para poder sepáralos del resto, se usa el comando:

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

Dando como resultado que el primer scaffold se llama: **NODE_1_length_668648_cov_39.096881**

Luego se genera una lista donde se almacena el nombre del scaffold para poder separarlo del resto de los demás:

```
vim SCA1.list
```

Luego se realiza el recorte del scaffold con el comando:

```
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"}' SCA1.list scaffolds.fasta
    > scaffold_NODE_1.fna
```

Ahora teniendo nuestro scaffold NODE_1 se procedió a realizar la anotación estructural y esto es mediante la interfaz web de augustus https://bioinf.uni-greifswald.de/augustus/ . Como parametros se eligió: organismo de referencia a *Escherichia Coli*, Report genes en both strands y Alternative transcripts en middle

Luego de analizar la parte gráfica del reporte de Augustus se opto por elegir los genes g401.t1 al g425.t1 para realizar la anotación funcional. Y para esto de la sección de predicted amino acid sequences y predictes coding sequences del mismo reporte de Agustus se extrajeron dichas secuencias almacenándose en archivos .faa y .fna respectivamente.

Posteriormente para la antoación funcional se uso la herramienta blast del NCBI donde se obtuvieron porcentajes de identidad del 99 % al 100 % en los 26 genes estudiados para el caso del archivo de nucleotidos.fna que se corrió contra la base de datos de non-redundant protein sequences (nr) con los demas valores por default y un 90 % al 100 % de identidad con el archivo aminoacidos.faa y que se corrió con la base de

datos de nucleotide collection (nr/nt) y los demás valores por default.

Finalmente se uso el softwere propietario Blast2Go https://www.blast2go.com/ en su versión basica para comparar ambos resultados unicamente al arhcivo aminoacidos.faa y donde 6 de los genes estudiados completaron el proceso de anotación

III. Resultados

Esta sección contiene los resultados detallados de los procedimientos realizados en la sección ÏI. Metodología"

III-1. Descarga de secuencias del NCBI: La información del genoma usado es:

- Size: 282.1Mb
- Instrument: Illumina MiSeq
- Strategy: WGS
- Source: GENOMIC
- Selection: RANDOM
- Layout: PAIRED

Al ejecutar el comando de descarga se obtienen dos archivos de 579.899 MB cada uno denominados SRR17084738_1 y SRR17084738_2 , los cuales contienen el genoma dividido en cada uno de sus pares.

III-2. Control de calidad: Los resultados del FASTQC de cada uno de los pares son los siguientes:

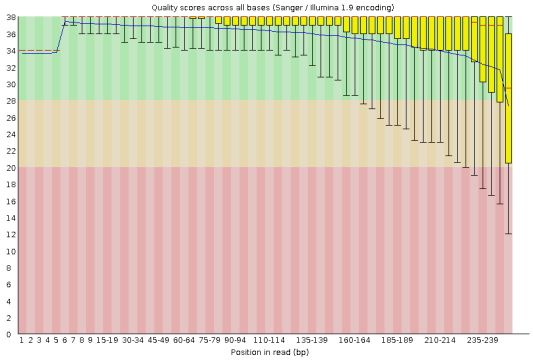

Figura 1. Calidad de SRR17084738_1

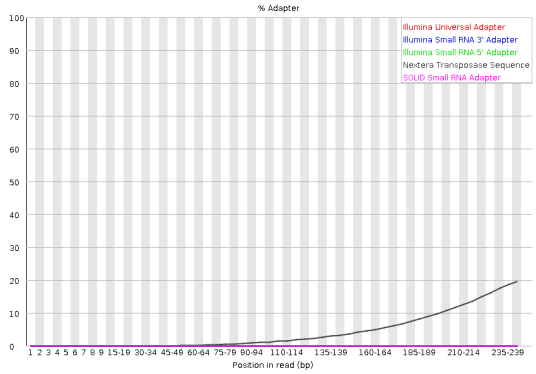

Figura 2. Contaminante de adaptadores SRR17084738_1

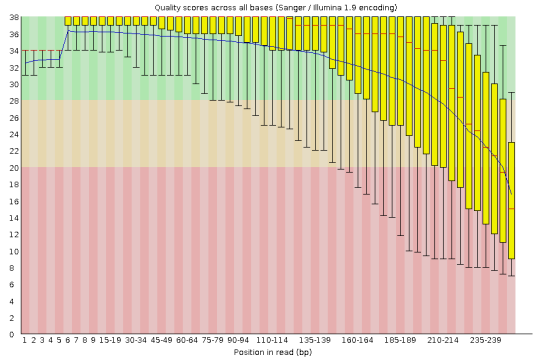

Figura 3. Calidad de SRR17084738_2

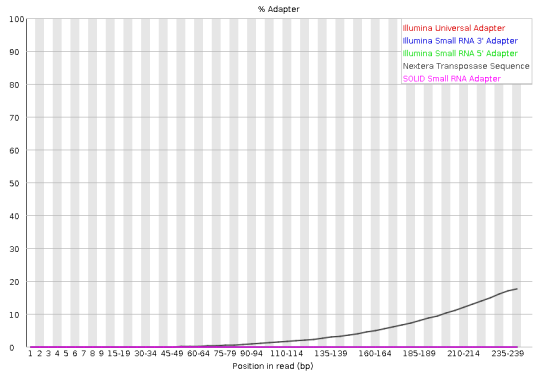

Figura 4. Contaminante de adaptadores SRR17084738_2

Lo más destacable al respecto es que:

- Tamaño total: 1,024,419 secuencias
- %GC : 65, mas que el porcentaje teoríco

También el FastQC arrojó que la muestra estaba contaminada con el adaptador "nextera transposase sequence", por lo que el siguiente paso fue realizar un filtrado de secuencias

III-3. Filtrado de secuencias: Después de realizar un análisis de FASTQC al resultado del filtrado de

FASTP condicionado a una calidad phred mayor a 30 se obtuvieron los siguientes resultados:

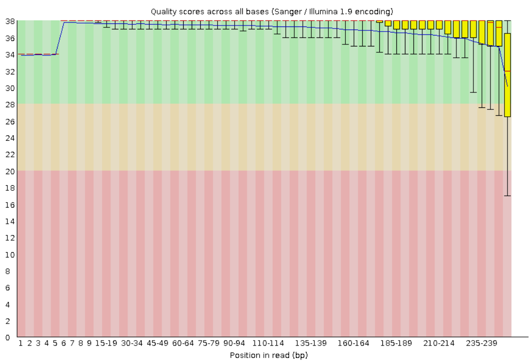

Figura 5. Calidad de filtrado_1

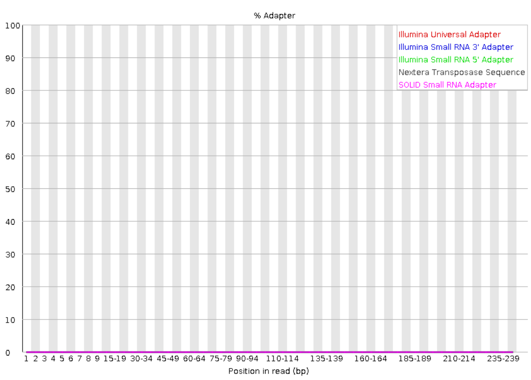

Figura 6. Contaminante de adaptadores de filtrado_1

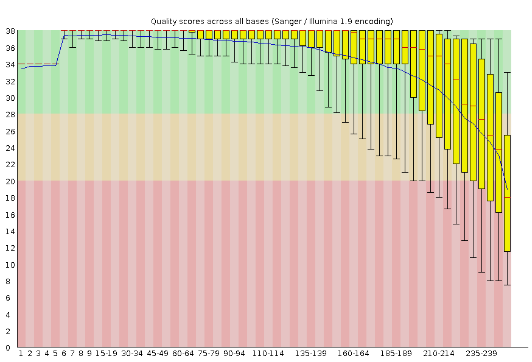

Figura 7. Calidad de filtrado_2

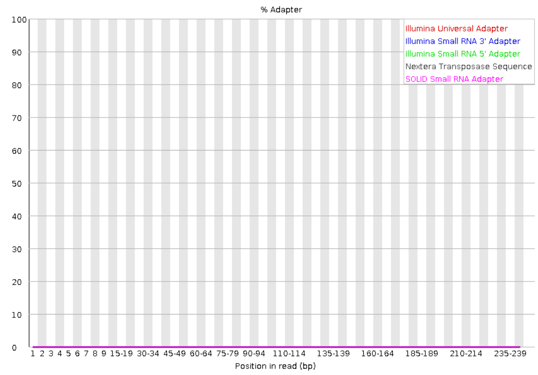

Figura 8. Contaminante de adaptadores de filtrado_2

Del total de secuencias originales solo 766,287 pasaron el filtrado, es decir únicamente 74.802107 % de ellas, asi mismo se logro eliminar los adaptadores de ambos archivos

III-4. Cálculo de cobertura: Aquí se muestra la ejecución de los comandos por los cuales se llega al valor de la cobertura.

```
(base) [avillicana@gaia Proyecto_final]$ cat filtrado_1.fastq | paste - - - - | awk '{print$2}' | wc -l
766287
(base) [avillicana@gaia Proyecto_final]$  cat filtrado_1.fastq | paste - - - - | awk '{print$2}' | wc -c
5300155
(base) [avillicana@gaia Proyecto_final]$ expr 5300155 - 766287
4533868
```

Figura 9. Calculo de cobertura

Dando como resultado el valor de 4533868 para la cobertura

III-5. Mapeo: Las primeras lineas del mapeo sin las secuencias que no mapearon son:

```
(base) alexxacv@Laptop-Omen:/mnt/c/Users/alexc/Desktop/prueba2$ tail filtradovsREF_reducido.sam
SRR17084738.782364      163     NC_002516.2     5774484 42      251M    =       5774497 264     GTGATGGCCTCCGGCCGCGAG
GTGCACATCGACGACCTGCCGCCGGAGCTGCTGACCCAGCCGCAGGACAGCGCGCCGGCGGCGAACTGGGAACAGGCCCTTCGCCAGTGGGCCGACCAGGCGCTGGGCCGCGGCCAG
TCGAACCTGCTGGACAGCGCCGTGCCGGCCTTCGAACGGATCATGATAGAAACCGCCCTCAAGCACACCGCCGGCCGCCGCCGCGATGCCGCCGTGCTGCTCGGCTGGGGCCG    C
C@CCGEFGFEGE@F@CEC+@FCCFFCGGGG8FGEGEEGFFGGGGGCE+CEGGGE=FFFGGGGGGGDGGFFGGGEG7FEE7F:=CG@G<CG:FFCCGF:BF:FC>F7FFB:BFFE7CF
FFGCGEF@G7F<FFFGG58>CEGDFFGGGGGFG>*@FEGGG>8:EFCECCC?:C=8;CE=E=@G99C97:?70CDG5ED757+9*>FFCEGG))/>DC5<CDEF5;58>FF)C3)4C
:????(-87115401(        AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:251        YS:i:-3 YT:Z:CP
SRR17084738.782365      99      NC_002516.2     5207660 42      251M    =       5207691 282     GTATAACGGCGCCGCGAAAGC
ACCGCAGCATTGCCTTGGACGTTCGCCACGGGGCGCGGTTCCTCGCCTGTGGCAGGTGGCGCAGGGCACGCATGCCCGGCGTTTGGCAGATACGGCGCGGCGGATTGTCCCAGGGGG
CTTTCGCCGGCGGGCACTTCGTGACCGGGAATGGGAAAAAAGTCTAAGGAGCGGGGTATTTTTTGATTTTCACCGTTATAATCGCCCCCCGATTATAAGGACGCCTCCTGATC    C
CCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGDGGGGGGGGGGGGGEGGGGGDGFGGGGGGGGGGGDGEGGGGGGGGGGGGCFFGGGG>FGGGGFGGGGE
GEEGGE@CFGEFFCFFGGEEDG>EGGGGGGGGGGGGGGFG8CFDGFEGGGDGGGGGECGDGGCF?CCCF9FFCEGGEGFFCGFGGEGGFF?FGGGGGCGFCGG8;?E8EDG=C5*<C
0?66ECDGG57*<>70        AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:251        YS:i:-14        YT:Z:CP
SRR17084738.782365      147     NC_002516.2     5207691 42      251M    =       5207660 -282    GGGCGTGGACGTTCGCAACGG
GGCGCGGGTCCTTGCCTGTGGCAGGTGGCGCAGGGCACGCATGCCCGGCGTTTGGCAGATACGGCGCGGCGGATTGTCCCAGGGGGCTTTCGCCGGCGGGCACTTCGTGACCGGGAA
TGGGAAAAAAGTCTAAGGAGCGGGGTATTTTTTGATTTTCACCGTTATACTCGCCCCCCGATTATAAGGACGCCTCCTGATCGGGTTCCGGCCCCCGCGCCATCCCCCCACAG    (
((4*82.4(3.7(7-*0D@D>B3@;8))8.1*)1**≤4GC2)EDCC55:CGCCGC*5C7DGCDECGGGGC@CF7FG@6EGGC:8C5E;*F@<**;*55GGGGFCFE>C>>C8*CGG
FA7@A5C>>GGGEEGGF<DGGGDF>FF?7FGGGGGF?**E>>FFFCCCDGGFCC,?7F>CCGGGGGFC:*CGGGGGGFCF=@9GGFGGGDGGGGGGGGGFFEFGGDGGGGGGGGGGGG
GGGEGGGGGDGCCCCC        AS:i:-14        XN:i:0  XM:i:7  XO:i:0  XG:i:0  NM:i:7  MD:Z:0T1C1T11C11T4C153A63       YS:i:
0       YT:Z:CP
SRR17084738.782366      97      NC_002516.2     3407743 42      251M    =       3408193 701     GCCGATGACCGCATTCATCGG
CGTACGGATCTCGTGGCTCATGGTCGCCAGGAAGACGCTCTTCGCCCGATTCGCCTCGTCCGCCTGGCGCATCGCCTCGCGCAACTGGCGGGCCAGGCGCTCCCGTTCGGTGATGTC
GATCCAGCCGTTCATCAGGCCGAGCACCCGTCCCTGGCGATCCCTGTAGGGCGTGACCCAATGGTGGATCTCCAGCAATTGCCCGCCGATGCGCACCGCCATGTCGGTGAAGC    C
CCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGFFGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGEGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGFFGEGGGGGGG
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGGGEFGGGGGGGGGGGGGDGGGGEFGGGGGGGGGGGGGGGGGGGGGGGGCFFGG>EEGGGGGGGGG
GGEGGGGGGGGDFFGF        AS:i:-5 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:168G82     YS:i:-4 YT:Z:DP
SRR17084738.782366      145     NC_002516.2     3408193 42      251M    =       3407743 -701    GACCAGCGCCTGCTTCAATGC
CAGGCGGCCGCGCAGGTCGGCTTCGGTACGTTGCCAACGCCGGAGCCTGAGCTGCAGGACGGTATTCCAGCCCAGGCATCCGACGCAGACCAGGCAGGCCACCAGCAGGCCGAGCAC
CAGGCTGGAAGGGGATGGCGAGCGCAGCCCGTCGGCAGCCTGGCGTAGCGGCGCTTCGTCGACCACATCGGCGTCCCCCGGACGATCCGGAGCGGGAATCCGGAGCATGGAAG    -
311599E?BFFF4F2C4>FF>5EB>BB500)*:CDC5GFDDC5GDFC:5FFEGFGEEDC5:*ECF6GFGCGEGFGGGECEGC<+EFECFGGGFGFFEBEE>GEGGFFGGC@GGGGGF
C*FFGFFGFGGGFDGFCF>FCGGGFGGGGGGGGGEGGGGGGGGGFCGGDGGGGFCFGGGEDGGGGDDGGGGDGGGEGFEGFGEEEGEDGGGGGFEEFBFGGGFFGGGGGGGGGGGGG
```

Figura 10. Primeras lineas del mapeo reducido

III-6. Ensamble de genomas: En el análisis sin referencia tenemos:

Cuadro I
REPORTE CALIDAD DE MAPEO SIN REFERENCIA. ALL STATISTICS ARE BASED ON CONTIGS OF SIZE ≥ 500 BP, UNLESS OTHERWISE NOTED (E.G., "# CONTIGS (≥ 0 BP).ÁND "TOTAL LENGTH (≥ 0 BP)ÏNCLUDE ALL CONTIGS).

Assembly	scaffolds	scaffolds_broken
# contigs (≥ 0 bp)	80	-
# contigs (≥ 1000 bp)	48	57
# contigs (≥ 5000 bp)	41	50
# contigs (≥ 10000 bp)	38	47
# contigs (≥ 25000 bp)	34	41
# contigs (≥ 50000 bp)	30	35
Total length (≥ 0 bp)	**6425745**	-
Total length (≥ 1000 bp)	**6414820**	6413920
Total length (≥ 5000 bp)	**6403294**	6402394
Total length (≥ 10000 bp)	**6385298**	6384398
Total length (≥ 25000 bp)	**6321690**	6286931
Total length (≥ 50000 bp)	**6171899**	6075814
# contigs	55	64
Largest contig	668648	668648
Total length	**6419571**	6418671
GC (%)	66.39	66.39
N50	**303720**	239845
N75	**160338**	106343
L50	**7**	9
L75	**14**	19
# N's per 100 kbp	14.02	**0.00**

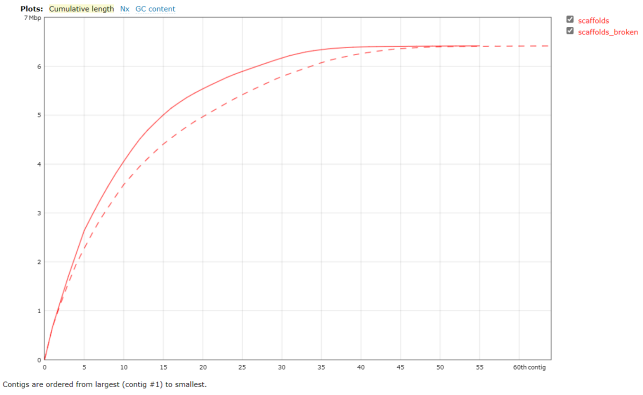

Figura 11. Gráfica de longitud acumulativa de la calidad sin referencia

Cuadro II
REPORTE CALIDAD DE MAPEO CON REFERENCIA. ALL STATISTICS ARE BASED ON CONTIGS OF SIZE ≥ 500 BP, UNLESS OTHERWISE NOTED (E.G., "# CONTIGS (≥ 0 BP).ÁND "TOTAL LENGTH (≥ 0 BP)ÏNCLUDE ALL CONTIGS).

Assembly	scaffolds	scaffolds_broken
# contigs (≥ 0 bp)	80	-
# contigs (≥ 1000 bp)	48	57
# contigs (≥ 5000 bp)	41	50
# contigs (≥ 10000 bp)	38	47
# contigs (≥ 25000 bp)	34	41
# contigs (≥ 50000 bp)	30	35
Total length (≥ 0 bp)	**6425745**	-
Total length (≥ 1000 bp)	**6414820**	6413920
Total length (≥ 5000 bp)	**6403294**	6402394
Total length (≥ 10000 bp)	**6385298**	6384398
Total length (≥ 25000 bp)	**6321690**	6286931
Total length (≥ 50000 bp)	**6171899**	6075814
# contigs	55	64
Largest contig	668648	668648
Total length	**6419571**	6418671
Reference length	6264404	6264404
GC (%)	66.39	66.39
Reference GC (%)	66.56	66.56
N50	**303720**	239845
NG50	**303720**	239845
N75	**160338**	106343
NG75	**160338**	113612
L50	**7**	9
LG50	**7**	9
L75	**14**	19
LG75	**14**	18
# misassemblies	43	**42**
# misassembled contigs	20	20
Misassembled contigs length	5108685	**4067034**
# local misassemblies	42	42
# scaffold gap ext. mis.	**1**	-
# scaffold gap loc. mis.	**6**	-
# unaligned mis. contigs	0	0
# unaligned contigs	**1 + 26 part**	1 + 27 part
Unaligned length	403575	**403122**
Genome fraction (%)	**95.939**	95.937
Duplication ratio	1.001	1.001
# N's per 100 kbp	14.02	**0.00**
# mismatches per 100 kbp	493.93	**493.79**
# indels per 100 kbp	10.48	**10.45**
Largest alignment	**389953**	389622
Total aligned length	**6014261**	6014061
NA50	**125560**	106343
NGA50	**137692**	107066
NA75	**69741**	69474
NGA75	**72080**	69726
LA50	**17**	19
LGA50	**16**	18
LA75	**33**	38
LGA75	**32**	37

En el análisis con referencia tenemos:

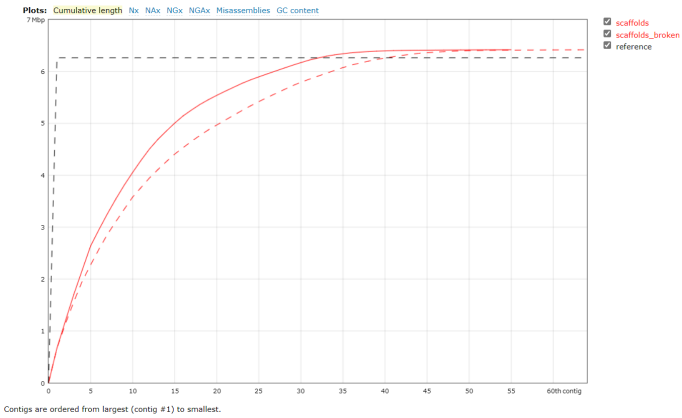

Figura 12. Gráfica de longitud acumulativa de la calidad con referencia

III-7. Anotación: Aqui se muestran los resultados de la anotación funcional, estrucutral y de manera general de scaffold NODE_1_length_668648_cov_39.096881

III-7a. Anotación estructural: Después de correr Augustus en su versión web con los parámetros previamente explicados en la sección II-7 se obtuvo el siguiente resultado, a una escala de 25.5kbp

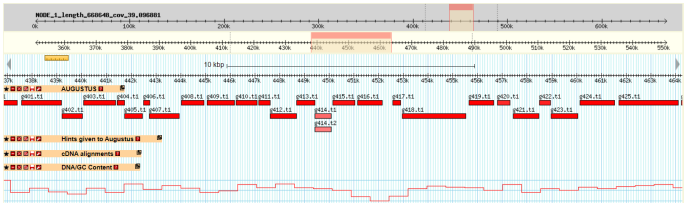

Figura 13. Anotación estructural por medio de augustus

De esta región del scaffold tomamos los genes marcados con los indicadores g401.t1 al g425.1 para que una vez extraídos las secuencias de aminoácidos y nucleótidos del reporte de augustus se procede a realizar la anotación funcional por medio de blast del NCBI y de blast2GO

III-7b. Anotación funcional: Al correr Blast con los parámetros explicados en la seccion II-7 se logró encontrar coincidencias de valores mínimos 90 % para los 24 genes estudiados. Dichos reportes por la cantidad de información se anexan en el repositorio, en la carpeta de Análisis funcional

Al correr el archivo de nucleotidos.fna en Blast2Go y el blast de NCBI se encontraron los siguientes resultados.

Cuadro III
RESULTADOS DE ANOTACIÓN FUNCIONAL DEL SCAFFOLD NODE_1

Nombre Seq	Descripción	Longitud	%Identidad
g401.t1	CTP synthase	1629	100 %
g402.t1	3-deoxy-8-phosphooctulonate synthase	846	99 %
g403.t1	phosphopyruvate hydratase	1290	99 %
g404.t1	cell division protein FtsB	285	99.84 %
g405.t1	2-C-methyl-D-erythritol 4-phosphate cytidylyltransferase	705	99 %
g406.t1	sulfurtransferase-like selenium metabolism protein YedF	249	98 %
g407.t1	selenium metabolism membrane protein YedE/FdhT [Pseudomonas aeruginosa]	1227	99.75 %
g408.t1	glutathione-dependent formaldehyde neutralization regulator	909	99 %
g409.t1	S-(hydroxymethyl)glutathione dehydrogenase/class III alcohol dehydrogenase	1113	99 %
g410.t1	S-formylglutathione hydrolase	852	99 %
g411.t1	2-C-methyl-D-erythritol 2,4-cyclodiphosphate synthase	474	99 %
g412.t1	tRNA pseudouridine(13) synthase TruD	1068	99 %
g413.t1	5'/3'-nucleotidase SurE	640	99 %
g414.t1	O-methyltransferase	672	99 %
g414.t2	O-methyltransferase	669	99 %
g415.t1	peptidoglycan DD-metalloendopeptidase family protein	894	99 %
g416.t1	RNA polymerase sigma factor RpoS	1005	99 %
g417.t1	ferredoxin family protein	324	100 %
g418.t1	DNA mismatch repair protein MutS	2568	100 %
g419.t1	TolB family protein	1008	91 %
g420.t1	CinA family protein	507	99 %
g421.t1	recombinase RecA	1041	96 %
g422.t1	recombination regulator RecX	462	99 %
g423.t1	LOG family protein	1071	99 %
g424.t1	MBL fold metallo-hydrolase	1404	97 %
g425.t1	xylulose 5-phosphate 3-epimerase	2406	99 %

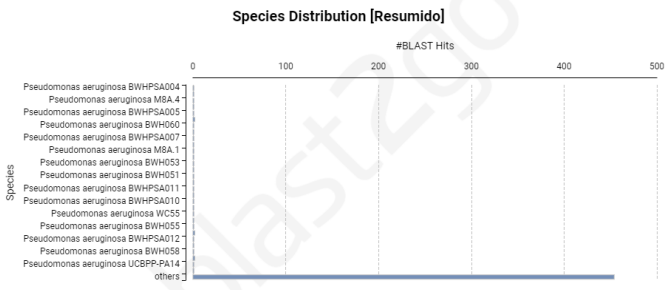

Figura 14. Distribución de especies

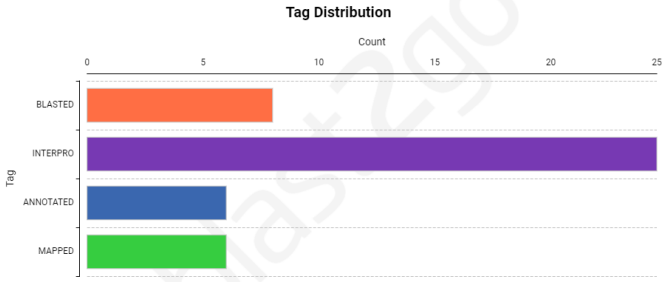

Figura 15. Resultados de blast2GO en histograma de distribución

IV. Discusiones y conclusiones

A partir del análisis de calidad inicial podemos comprobar que la calidad del archivo 2 pareado es inferior al archivo 1, aun así, con el filtrado de fast pig pudimos poco más de 300,000 secuencias que no cumplían un mínimo de calidad de phred 30, Así como eliminar los adaptadores. Del ensamble del genoma podemos observar en el cuadro 2 Que el total de la longitud es similar al del valor de la referencia Siendo de 6419571 para nuestro ensamble contra 6264404, Asimismo también es similar el valor del porcentaje de GC, siendo de 66.39 de nuestro mapeo contra 66.56 del valor de la referencia. Tenemos un total de 80 scaffolds siendo poco menos de la mitad, 38 mayores o iguales a 10000bp. Por lo que podemos concluir que el ensamble es confiable para realizar la anotación dada la similitud con la referencia. En la anotación estudiamos 26 genes del primer scaffold Scaffold Node_1 , del g401.t1 al g425.t1, de los cuales 25 presentan un solo alelo y gen g414 presenta 2 alelos del mismo en nuestro organismo de estudio g414.t1 y g414.t2 respectivamente, y al momento de realizar la anotación funcional mediante BLAST y blast2GO vemos que efectivamente cumplen la misma función que es codificar para O-methyltransferase, y tiene una diferencia de longitud de apenas 3 nucleótidos. Del resto de la anotación podemos decir que en base a los porcentajes de identidad que van del 91 % al 100 % en los genes estudiados esa región del scaffold es muy acordé a lo que se espera en las referencias de las bases de datos. Asimismo, de las gráficas de blast2GO podemos observar que la distribución de especies todas aparecen como *pseudomonas aeruginosa* pero hay un error que aparece como otros, el cual apareció después de terminar de anotar mediante este software por lo que se lo atribuyo un error de configuración ya que sólo 7 genes de los 26 estudiados pudieron completar el blast solamente 6 pudieron mapearse y anotarse. Aun así, la información que arrojó Blast del me permitió llegar al cuadro III en el cual está la descripción de que codifican cada uno de los 26 genes estudiados así como su longitud y porcentaje de similitud arrojado por el Blast.

Apéndice A
Descripción de herramientas bioinformáticas usadas.

- SRAtoolkit : Es un conjunto de herramientas proporcionadas por el NCBI para poder descargar secuencias de la base de datos de Sequence Read Archive (SRA) data
- FastQC : Proporcionar una forma sencilla de realizar algunas comprobaciones de control de calidad en los datos de secuencia sin procesar que provienen de piplines de secuenciación de alto rendimiento. Es una manera rápida y sencilla de usar de realizar comprobaciones de calidad.
- FastP : Herramienta para procesar datos de FASTQ, El algoritmo tiene funciones para control de calidad, recorte de adaptadores, filtrado por calidad y poda de lectura.
- bowtie2 : Herramienta para realizar mapeos de secuencaicion mediante el uso de referencias. Bueno para alinear lecturas de aproximadamente 50 hasta 100 o 1,000 caracteres, y particularmente bueno para alinearse con genomas relativamente largos. Funciona mediante la creación de un indice de referencia
- FAS Center for Systems Biology : Plataformade la Universidad de Harvard que contiene varios scriptos escritos en Perl para el procesamiento de datos biológicos, principalmente en formato FASTA
- SPAdes : Es un ensamblador de genomas diseñado para genomas relativamente pequeños. Principalmente para bacterias.
- Augustus : Es un programa que predice genes de una secuencias genómicas. Es de código abierto, y funciona para la anotación estructural
- Blast2GO : Es un programa propietario para realizar de manera automática la anotación funcional

de datos de secuencias genómicos. Se apoya principalmente del algoritmo de BLAST

- BLAST : Basic Local Alignment Search Tool (BLAST) es una herramienta que permite encontrar similitud de secuencias ya sea de nucleótidos o de proteínas. Lo hace con el apoyo de varias bases de datos de secuencias y se puede utilizar para inferir la anotación funcional y evolutiva de secuencias.

REFERENCIAS

[1] Nachtweide, S., & Stanke, M. (2019). Multi-Genome Annotation with AUGUSTUS. Methods in Molecular Biology (Clifton, N.J.), 1962, 139–160. https://doi.org/10.1007/978-1-4939-9173-0_8

[2] Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. Nature Methods, 9(4), 357. https://doi.org/10.1038/NMETH.1923

[3] Langmead, B., Wilks, C., Antonescu, V., & Charles, R. (2019). Scaling read aligners to hundreds of threads on general-purpose processors. Bioinformatics, 35(3), 421–432. https://doi.org/10.1093/BIOINFORMATICS/BTY648

[4] Stover, C. K., Pham2, X. Q., Erwin, A. L., Mizoguchi, S. D., Warrener, P., Hickey, M. J., Brinkman3, F. S. L., Hufnagle, W. O., Kowalik, D. J., Lagrou, M., Garber, R. L., Goltry, L., Tolentino, E., Westbrock-Wadman, S., Yuan, Y., Brody, L. L., Coulter, S. N., Folger, K. R., Kas2, A., … Olson2, M. v. (2000). Complete genome sequence of Pseudomonas aeruginosa PAO1, an opportunistic pathogen. NATURE, 406. www.nature.com

[5] Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., Lesin, V. M., Nikolenko, S. I., Pham, S., Prjibelski, A. D., Pyshkin, A. v., Sirotkin, A. v., Vyahhi, N., Tesler, G., Alekseyev, M. A., & Pevzner, P. A. (2012). SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing. Journal of Computational Biology, 19(5), 455–477. https://doi.org/10.1089/CMB.2012.0021

[6] Lantz, H., Dominguez Del Angel, V., Hjerde, E., Sterck, L., Capella-Gutierrez, S., Notredame, C., Vinnere Pettersson, O., Amselem, J., Bouri, L., Bocs, S., Klopp, C., Gibrat, J. F., Vlasova, A., Leskosek, B. L., Soler, L., & Binzer-Panchal, M. (2018). Ten steps to get started in Genome Assembly and Annotation. F1000Research, 7. https://doi.org/10.12688/F1000RESEARCH.13598.1

[7] Chen, S., Zhou, Y., Chen, Y., & Gu, J. (2018). fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics, 34(17), i884–i890. https://doi.org/10.1093/BIOINFORMATICS/BTY560