PROTOCOLO DE SECUENCIACIÓN PARA *CAMPYLOBACTER JEJUNI*

Daniela Oliva Solis

ABSTRACT

La bacteria *Campylobacter jejuni* es la principal causa de gastroenteritis bacteriana en el mundo. Sin embargo, la epidemiología de este patógeno solo se comprende parcialmente y es difícil arrojar nueva luz sobre esta área, ya que la mayoría de los casos parecen ser esporádicos y no se notifican. A través de la secuenciación del genoma completo, este trabajo pretende mejorar la predicción de genes además de proveer una guía completa del proceso bioinformático de ensamblaje y anotación.

INTRODUCCIÓN

Campylobacter jejuni, una bacteria móvil gramnegativa, con forma de espiral, es una de las principales causas bacterianas de gastroenteritis transmitida por alimentos en todo el mundo. La campilobacteriosis es una infección autolimitada, caracterizada por un inicio rápido de fiebre, calambres abdominales severos y diarrea que puede incluir sangre, dichos síntomas se hacen evidentes entre 1 y 7 días después del consumo del alimento o líquido contaminado.

Varios estudios epidemiológicos han demostrado un vínculo entre el consumo de aves de corral poco cocidas u otros productos que han estado en contacto. *C. jejuni* coloniza el intestino ciego de los pollos y gallos, a partir de ahí, una vez que se inoculan de manera experimental, la bacteria puede permanecer presente durante toda la vida del ave.

Sin embargo, *a* lo largo de la vida de un pollo se producen cambios drásticos en los niveles de anticuerpos contra. Una vez que un pollo es colonizado con *C. jejuni*, se generan anticuerpos contra la bacteria. Aunque estos anticuerpos pueden no eliminar una población establecida de *Campylobacter*bacterias, se ha observado una disminución en el número de organismos *C. jejuni* que colonizan el tracto intestinal.

El trabajo de ensamblaje y anotación que aquí se trabaja, busca conocer mejor el funcionamiento de algunos genes y ver como se relacionan con ciertas proteínas del tracto digestivo de gallos.

METODOLOGÍA

La secuenciación del genoma se llevó acabo con la tecnología MiSeq de Illumina con archivos de librerías pareadas. El ensamble y la anotación se llevó a cabo mediante plataformas de acceso público acompañado de algunos comandos específicos ejecutados en la terminal de Linux.

Descarga de archivos

Para descargar los datos de *Campylobacter jejuni* se puede realizar a través del link https://www.ncbi.nlm.nih.gov/sra/SRX13305038[accn], del cual se generarán dos archivos fastq por el tipo de librería pareada.

```
$ fastq-dump --split-3 SRR17120534
```

Análisis y Filtrado de calidad

Fastqc es una interfaz abierta en donde se visualiza y analiza de manera gráfica los resultados del tipo secuenciación empleada, para este caso particular es Illumina, pero cabe destacar que funciona igual para cualquier otra tecnología. A través de la terminal en la máquina remota se ejecuta el comando

```
$ fastqc *.fastq -o ResultsFqc/
```

Por automático se generan dos archivos html que se deben bajar a la máquina remota para analizarlos.

Fastp es una herramienta ejecutable que permite hacer cambios y filtrar calidades en los archivos fastq. El filtrado se hará de acuerdo a los posibles errores que fueron detectados en la visualización gráfica. La herramienta por sí sola ya incluye parámetro por default que mejoran la calidad de los reads, sin embargo, siempre está la opción de cambiar los parámetros para ser más estricto y/o específico a la hora del filtrado.

Los parámetros que aquí se emplearon fueron -A (elimina todos los adapatadeores automáticamente) -qualified_quality_phred 30 (todas aquellos reads que sobrepasen este número serán eliminados), -l (se descartarán todas las bases que estén por arriba de 36)

```
$ fastp -
i /home/doliva/ProyectoFinal/SRR171
20534_1.fastq -
I /home/doliva/ProyectoFinal/SRR171
20534_2.fastq --
qualified_quality_phred 30 -A -
l 36 -o SRR17filtfastp_1.fastq -
O SRR17filtfastp_2.fastq >& fastp_S
RR17.log
```

Una vez que los nuevos datos fastq han sido generados, se tiene que realizar otro análisis de calidad con fasqc para veriricar que los parámetros sean más aceptables.

Cálculo de la Cobertura

La palabra cobertura, hace referencia a la profundidad del genoma. En promedio, cuantas veces se está leyendo cada una de las bases que cubren el genoma. La calculamos por medio de la fórmula

$$C = \frac{LN}{G}$$

C: Cobertura

L: Longitud de los reads

N: Número total de reads

G: Longitud del genoma de referencia

$$C = \frac{(21860)(251)}{1378432} = 3.269x$$

Mapeo

Mapear un genoma es necesario para comprender la estructura que lo compone, identificar sitios exactos donde se pueden localizar los genes u otras regiones interesantes. Cuando se mapea a un genoma de referencia, únicamente se busca, que secuencias de los reads logran un match con este genoma de referencia.

Bowtie2 es una herramienta rápida y de memoria eficiente para alinear las lecturas de secuenciación con secuencias de referencia largas. Puede ser instalada en la terminal por medio de un envaironment de anaconda, lo cual facilita su uso.

```
$ conda install -c bioconda bowtie2
```

El primer paso que se debe realizar es generar el índice con la secuencia referencia (para este estudio se utilizó una bacteria muy similar, *Campylobacter Coli*) no sin antes haber activado el environment.

```
$ bowtie2-build -f /home/doliva/ProyectoFinal/Mapeo/CampylobacterColi.fa CampylobacterColi
```

Luego hay que correr el mapeo

```
$ bowtie2 --maxins 1000 -x CampylobacterColi -1 /home/doliva/ProyectoFinal/Fastp/SRR17filtfastp_1.fastq -2 /home/doliva/ProyectoFinal/Fastp/SRR17filtfastp_2.fastq -S CampColi.sam
```

El parámetro –S generará un archivo sam que contendrá todos los reads que fueron mapeado, sin embargo y como en todo siempre hay excepciones, por lo que es recomendable realizar un análisis un análisis y verificar si existe la posibilidad de que algún read no haya mapeado.

```
$ awk '$3!="*"' CampColi.sam >CampColiFilt.sam
```

Una vez que ya se eliminaron los reads que no mapearon se puede confirmar que no haya contaminación en las secuencias antes del ensamble.

Ensamble

El ensamble de los reads se llevó a cabo con el programa SPAdes que pertenece al algoritmo DeBruijn construido con diferentes tamaños de k-meros. El proceso de ensamblaje comienza utilizando gráficos de Bruijn de tamaño múltiple para construir el gráfico de ensamblaje mientras detecta y elimina lecturas quiméricas. Luego, se estiman las distancias entre los k-mers para mapear los bordes del gráfico de ensamblaje. Posteriormente, se construye un gráfico de ensamblaje emparejado y SPAdes genera un conjunto de secuencias de ADN contiguas (contigs).

Para un ensamblaje correcto, es importante que haya suficiente superposición entre las lecturas de secuencia en cada posición del genoma, lo que requiere una alta cobertura de secuenciación. Normalmente, para lecturas de secuencia más largas, se puede esperar más superposición, reduciendo la profundidad de lectura sin procesar.

Entonces, primero se debe activar el environment donde SPAdes fue instalado y posteriormente se va a seguir trabajando con los archivos fastq filtrados por fastp.

```
$ conda activate spades
```

```
$ spades.py -k 33,37,41 -t 1 -m 7 --pe1-1 SRR17filtfastp_1.fastq --pe1-2 SRR17filtfastp_2.fastq -o spades_SRR
```

```
$ conda deactivate
```

Al indicarle la salida del archivo con el parámetro –o, se genera una carpeta de nombre spades_SRR que contiene varios archivos, entre ellos los scaffolds.

Luego, se tiene que checar la calidad del ensamble con la herramienta quast

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

A partir de este checado de calidad, se genera automáticamente un directorio llamado quast_results en cual contiene los resultados en html, este formato es importante para descargar porque en el vienen los resultados del ensamble de manera gráfica.

```
$ scp -P 10022 -
r doliva@132.247.172.26:/home/doliv
a/ProyectoFinal/GenomeAssembly/spad
es_SRR/quast_results/results_2021_1
2_06_15_42_56 .
```

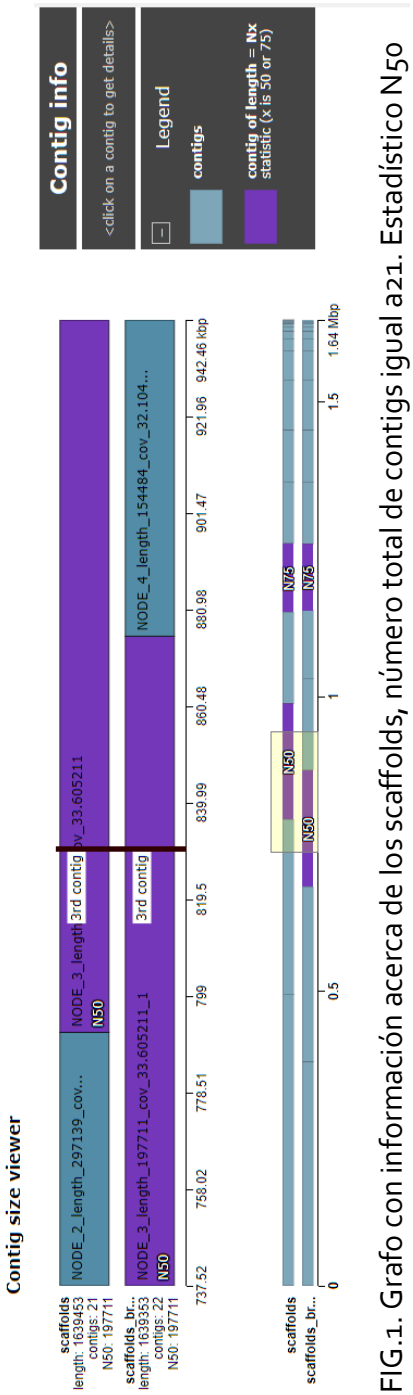

FIG.1. Grafo con información acerca de los scaffolds, número total de contigs igual a21. Estadístico N50 igual a 19711 que quiere decir que al menos la mitad de los reads están por arriba de este número

Worst Median Best ☑ Show heatmap

Statistics without reference	scaffolds	scaffolds_broken
# contigs	21	22
# contigs (>= 0 bp)	38	-
# contigs (>= 1000 bp)	15	16
# contigs (>= 5000 bp)	13	14
# contigs (>= 10000 bp)	11	12
# contigs (>= 25000 bp)	9	10
# contigs (>= 50000 bp)	9	10
Largest contig	494 161	380 493
Total length	1 639 453	1 639 353
Total length (>= 0 bp)	1 642 909	-
Total length (>= 1000 bp)	1 635 488	1 635 388
Total length (>= 5000 bp)	1 633 072	1 632 972
Total length (>= 10000 bp)	1 620 618	1 620 518
Total length (>= 25000 bp)	1 586 678	1 586 578
Total length (>= 50000 bp)	1 586 678	1 586 578
N50	197 711	197 711
N75	116 518	113 568
L50	3	3
L75	5	6
GC (%)	30.47	30.47
Mismatches		
# N's	100	0
# N's per 100 kbp	6.1	0

Tab 1. Estadísticos acerca de los scaffolds

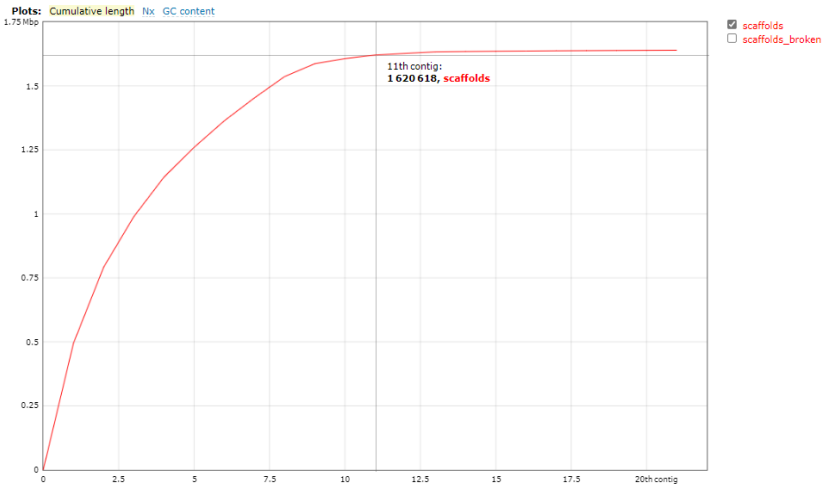

FIG.2. Gráfico de contigs ordenados de meyor a menor, se puede confirmar el estadístico N 50 con el contig 11

Para tener más información sobre el genoma es necesario secuenciar el transcriptoma (mRNA que se expresa en el momento exacto) que ayudará a tener datos concisos de donde están los genes de este organismo, es decir, al secuenciar el transcriptoma se tiene la secuencia confiable y exacta de los genes y si se ejecuta un mapeo con el ensamble que recién se acaba de hacer, se hará notar en que parte del genoma se encuentra dicho gen

Características que se parecen entre los genes de una especie ayudan a entrenar el algoritmo y hacer una mejor predicción de nuevos genes.

La anotación fue solamente de una porción del genoma, es decir, del archivo scaffold.fasta que fue generado por SPAdes solo utilizamos el scaffold número 3

Anotación estructural

La anotación estructural sirve para identificar qué hay y dónde está cada cosa (genes, regiones codificantes, regiones no codificantes, etc.) La herramienta que se utilizó fue el programa de **Augustus** que sirve para predecir genes a partir de algoritmos entrenados. La página brinda tres tipos de bacterias, de las cuales se eligió *Escherichia Coli* por la cercanía con *Campylobacter,* si se tuviera otro algoritmo entrenado de una bacteria más cercana, es posible que la predicción de estos genes fuera más certera.

Los parámetros especificados fueron que el reporte de los genes fuera para ambas cadenas y que el splicing alternativo estuviera en un promedio medio.

Anotación funcional

La anotación funcional que trata sobre precisamente saber las funciones de cada gen se realizó con la herramienta Blast + adaptada para la terminal en donde se podía correr el blastx de forma más rápida. Hay que resaltar que los archivos que se utilizaron fueron los de los aminoácidos generados por Augustus.

El trabajo de blastx es buscar y alinear secuencias de proteínas utilizando una secuencia de nucleótidos como referencia.

Se hicieron dos blastx, uno con *Campylobacter Coli* y otro con *Gallus Galllus*

```
$ makeblastdb -
in ref_seqGallus.fa -
dbtype prot out GallusDB
```

```
$ blastp -db GallusDB -
query aaseqScaffold3.fa -
out ScaffoldVsGallusDB
```

```
$ makeblastdb -
in ref_seqCampColi.fa -
dbtype prot -
out CampylobacterColiDB
```

```
$ blastp -db CampylobacterColiDB -
query aaseqScaffold3.fa -
out ScaffoldVsCampColiDB
```

A través de estos blasts se pretenden encontrar coincidencias fiables que ayuden a predecir la función de la porción que se anotó, que en este caso fue el scaffold 3.

RESULTADOS Y DISCUSIÓN

En cuanto al filtrado de calidad, hubo algunos reads que se eliminaron y en general el estudio fue más óptimo. La cobertura salió un poco baja a lo que se esperaba y aunque el ensamble se vio con buenos resultados, tal vez hubieran sido

mejor. En la anotación funcional se decidió hacer blastx con el organismo de Gallus porque como se mencionó al principio, cuando un ave es infectada por Campylobacter jejuni a menudo las proteínas del tracto digestivo se van acostumbrando a mantener la bacteria, esto no las hace inmunes ni mucho menos, pero sí generan ciertos tipos de anticuerpos que los ayudan soportarlo. Al encontrar unos pocos alineamientos con *C. jejuni* podría estar arrojando algún gen o mecanismo de la bacteria que sea apagado por el tracto digestivo y no cumpla con su función total de infectar al ave, es decir, que solamente se vuelve portador de la enfermedad.

REFERENCIAS

Shoaf-Sweeney, K. D., Larson, C. L., Tang, X., & Konkel, M. E. (2008). Identification of Campylobacter jejuni proteins recognized by maternal antibodies of chickens. *Applied and Environmental Microbiology*, *74*(22), 6867–6875. https://doi.org/10.1128/AEM.01097-08

Llarena A-K, Taboada E, Rossi M. 2017. Whole-genome sequencing in epidemiology of Campylobacter jejuni infections. J Clin Microbiol 55:1269–1275. https://doi.org/10.1128/JCM.00017-17

Ekblom, R., & Wolf, J. B. W. (2014). A field guide to whole-genome sequencing, assembly and annotation. *Evolutionary Applications*, *7*(9), 1026–1042. https://doi.org/10.1111/eva.12178

Pendleton, S., Hanning, I., Biswas, D., & Ricke, S. C. (2013). Evaluation of whole-genome sequencing as a genotyping tool for Campylobacter jejuni in comparison with pulsed-field gel electrophoresis and flaA typing. *Poultry Science*, *92*(2), 573–580. https://doi.org/10.3382/ps.2012-02695