



Ensamble y análisis de SARS-CoV-2

Marcos E. R. Ramírez

Resumen

Después de una jornada semestral de bioinformática, toca utilizar todo lo aprendido para realizar un ensamble. Para ello es importante conocer un poco sobre el SARS-CoV-2 así como la importancia de porque se secuencian el genoma. El uso de herramientas bioinformáticas tales como SRA Toolkit, FASTQC, Trimmomatic, bwa, spades, quast permiten precisar desde una muestra tomada y secuenciada cual es el posible organismo que esta presente y así como cuales son sus funciones y estructuras, sin embargo en este articulo no se completó correctamente todos los pasos del análisis, pero si se llegó a hacer hasta el ensamble del organismo, Por tanto estas prácticas hace que un estudiante tenga la dimensión de lo que implica hacer bioinformática y lo importante de la comprensión de cada cosa.

Introducción

El Coronavirus del Síndrome Respiratorio Agudo Severo de tipo 2 [1] (SARS-CoV-2) es un virus valga la redundancia de la familia de los coronavirus que causan enfermedades respiratorias, tanto comunes como graves como menciona [2], esta enfermedad es conocida como COVID-19 [2] y para las personas de mayor edad y para las que están bajo condiciones médicas como enfermedades cardiovasculares, diabetes, enfermedades crónicas respiratorias o cáncer tienen más probabilidades de desarrollar una enfermedad grave [3].

El surgimiento de SARS-CoV-2 fue observado en casos de neumonía inexplicable en la ciudad de Wuhan China [4]. Se realizaron estudios hasta saber de dónde podría haber brotado el virus, hasta que en murciélagos se identificaron como los hospedadores de importantes virus zoonóticos como el virus de Nipah, el virus Hendra y SARS-CoV, incluyendo coronavirus con

una considerable diversidad genética [4].

El surgimiento de la pandemia por SARS-CoV-2 es debido a la gran capacidad de virulencia. Esto es explicado por el descubrimiento de múltiples detalles estructurales y no estructurales semejantes al sitio único de escisión de FURIN, (SCoV2-PLpro), ORF3b y proteínas no estructurales, y así como cambios conformacionales en la estructura de la proteína spike durante el hospedaje de la célula [5]

Metodología y resultados

El propósito de este artículo es emplear técnicas bioinformáticas para el análisis de un organismo o virus en este caso SARS-CoV-2, el punto es poner en practica lo aprendido en clases y determinar como es estructural y funcionalmente.

Para ese punto primero será necesario entender desde la secuenciación cual método de secuenciación que se usó y sus características, con el propósito de saber que plataformas o re-

cursos como softwares bioinformáticos se usarán y que indicaciones dar.

Una vez identificado cuales son las características de secuenciación se procede a hacer análisis de calidad de la secuenciación para identificar valga la redundancia, la calidad y decidir si son muestras aptas para su análisis o si sirven para el trabajo de investigación, ya que se ha dado la luz verde se filtran la secuencias y/o bases que tengan una muy baja calidad, con ello para minimizar los errores en el análisis.

Ya filtradas las secuencias será necesario hacer mapeos para determinar a que organismo se parece mas en su secuencia de nucleótidos, con ello en mente se procede a hacer el ensamble y determinar la estructura y función de dichos genes que provienen de su muestra.

Dichas herramientas usadas en este articulo serán mencionadas y descritas sus funciones.

Solo con fines prácticos cada una de estas herramientas se

presentará la página para que puedan revisar sus manuales de instalación y uso, pero no se explicara como se instalaran (más que solo se explicara porque se usaron dichas opciones), además cabe destacar que la computadora que se iba a usar para este proyecto no funciono ya que no se instalaron correctamente algunos programas, por lo que se optó por hacerlo con una vía remota.

Para empezar, se descargar la secuenciación del **SARS-CoV-2**, entonces se ingresó a la base de datos de **NCBI** en la base de datos de **SRA** con el ID de **SRP251618**. Por tanto, la primera muestra con el Accession: **SRX10248073**, es seleccionada para ver sus características. Se anexa apoyo visual en la (fig. 1 [A](#)).

Podemos ver las siguientes características. En la parte de **show Abstract** tiene una pequeña descripción, la cuales citada: **Isolate raw read files from the Washington SARS-CoV-2 outbreak**, así como en **Strategy** pues secuenciar el RNA, es con el propósito de saber los patrones de expresión [\[6,7\]](#). Por otra parte, es de tener en cuenta que los reads son single, lo cual ayudara para pasos posteriores.

Descarga

La primera herramienta a disposición es

SRA Toolkit

Este es un software que incluye diversas herramientas y librerías para usar información en el "INSDC Sequence Read Archives (SRA)" [\[8\]](#)

Entonces, ya que dentro de la computadora que se operó no se pudo descargar el SRA, entonces se trabajó mediante una computadora remota, aplicando los siguientes comandos de SRA Toolkit ([comando 1](#)) y el run (fig. 1 [B](#)). Y saldrán los archivos como en la segunda parte del comando.

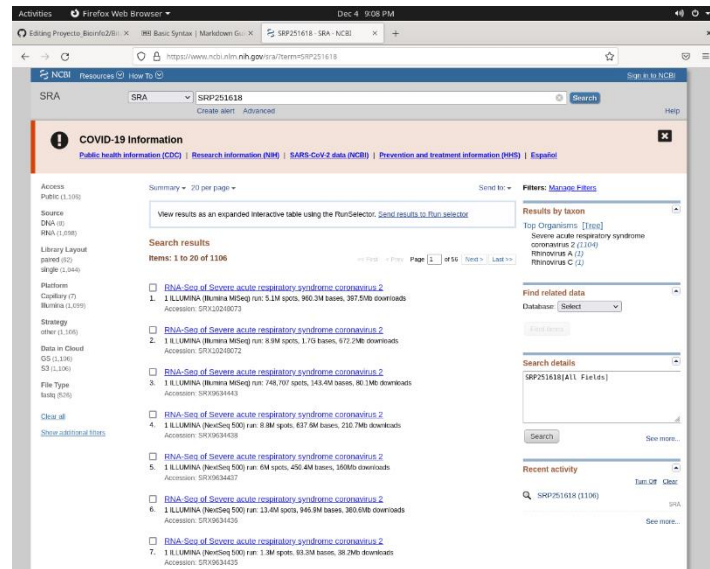
Control de calidad

FastQC

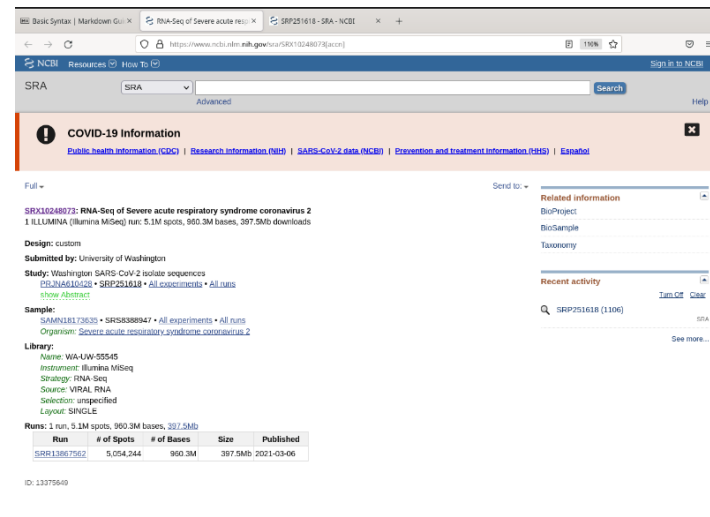
Este software [\[9\]](#) tiene la función de hacer un entorno gráfico la información del control de calidad de secuencias de alto rendimiento

Fig. 1: Ilustración de cómo se ven los pasos en la página

A.



B.



comando fastq-dump

```
$ fastq-dump SRR13867562
```

```
Read 5054244 spots for SRR13867562
```

```
Written 5054244 spots for SRR13867562
```

```
$ ls -lh
```

```
total 2.2G
```

```
-rw-rw-r-- 1 mramirez mramirez 2.2G Dec 5 01:32 SRR13867562.fastq
```

```
drwxrwxr-x 3 mramirez mramirez 28 Dec 5 01:23 ncbi
```

Para ello, con el siguiente comando ([comando 2](#)) y el **SRR13867562.fastq**, saldrán los dos archivos nuevos **SRR13867562_fastqc.html** y **SRR13867562_fastqc.zip** los cuales tendrán que abrirse el html para realizar el análisis de calidad.

En la (fig. 2) las características de la secuenciación se detallan como

Filename: Nombre del archivo fastq

File type: El tipo de archivo (el cual está bien ya que señala que tiene nombrada de forma correcta las bases)

Encoding: Indica que los valores de calidad ASCII están basadas en las tecnologías señaladas

Total sequences: La cantidad de secuencias procesadas es de poco más de 5 millones

Sequences flagged as poor quality: El número de secuencias de baja calidad removidas es de 0

Sequences length: La longitud promedio de las secuencias es de 190 bases, ya que todas son de esa longitud

%GC: Y el contenido de GC consta de un porcentaje de casi la mitad (los detalles en la gráfica correspondiente)

Es observable que la calidad de las secuencias es buena, cerca del 100% de las bases representan calidades por encima de 28 de calidad ASCII (sanger / illumina 1.9) a excepción al final la base numero 190 llega a tener calidades pobres de hasta 11, pero aun en promedio sigue siendo de buena calidad ya que las bases con muy baja calidad (poco menos del 25% de ellas) tienen una pequeña influencia. (fig.3)

En la fig. 4 es más visible, ya que la distribución de las bases ronda en calidades mayores a 36, esto quiere decir que la tasa de error es menor a 0.2%

En la fig. 5 el resultado no es el más óptimo al inicio y al final, ósea que hay diferencias mayores al 10% entre G y C en la posición 5, 150-154, 170-174 y 190. La misma situación entre A y T la diferencia es >10% en las posiciones 5-6, a mitad entre 150-154. Aunque esto será removido posteriormente

comando fastQC

```
$ fastqc SRR13867562.fastq
```

Fig. 2: Estadísticos básicos

Measure	Value
Filename	SRR13867562.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	5054244
Sequences flagged as poor quality	0
Sequence length	190
%GC	41

Fig. 3: calidad de bases por secuencia

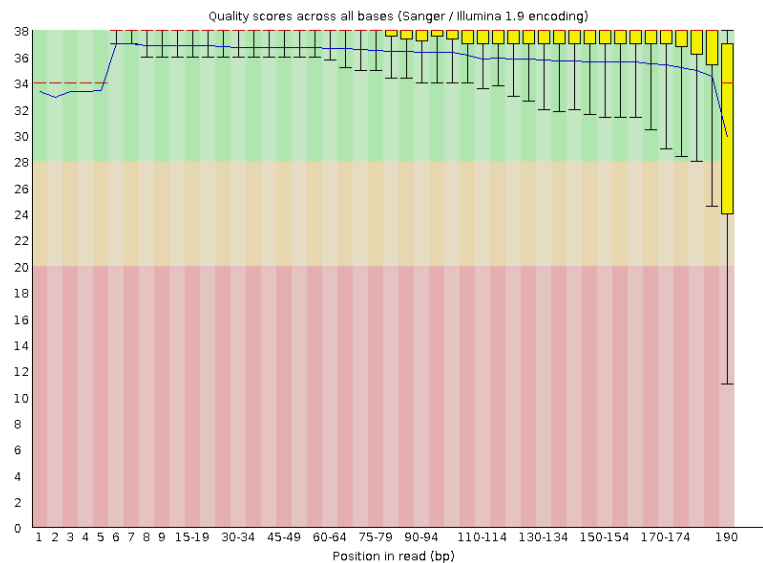


Fig. 4: Puntaje de calidad por secuencia

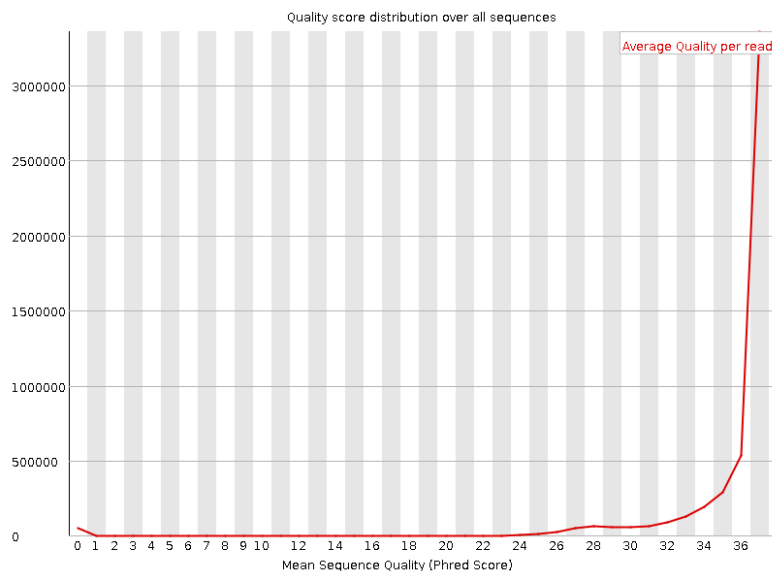


Fig. 5: Contenido de bases por secuencia

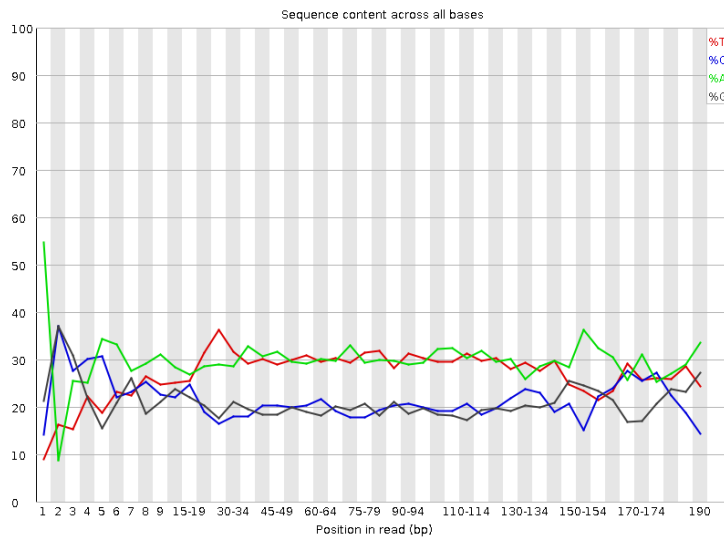


Fig. 6: Contenido GC por secuencia

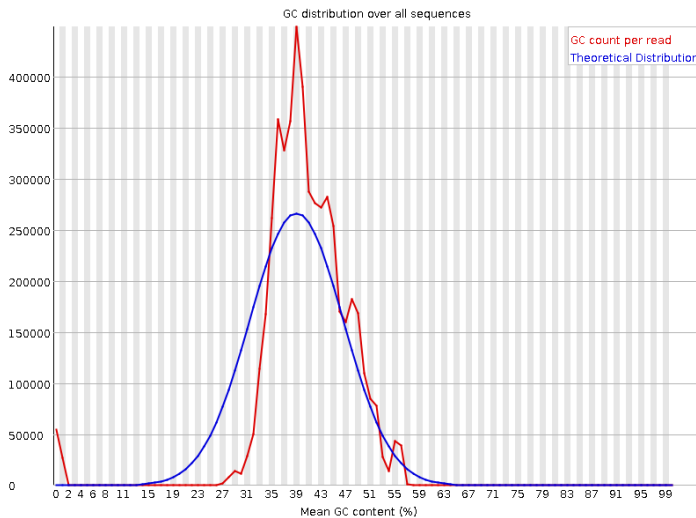
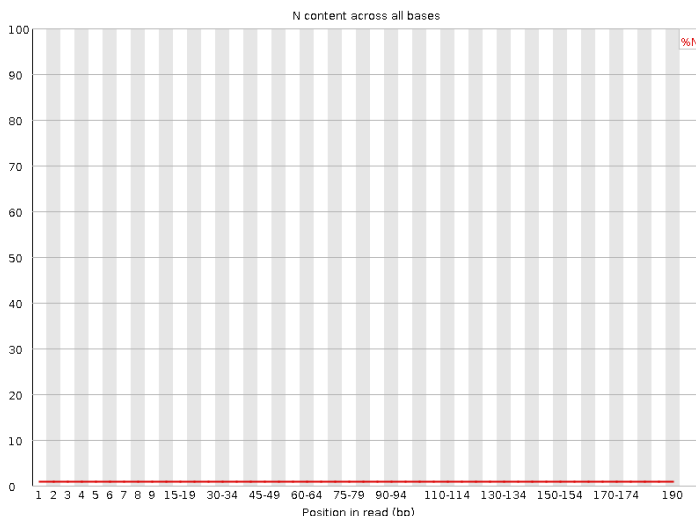


Fig. 7: Contenido de bases N



Es observable que la distribución en el contenido de GC está fuera del teórico, resultando en que mayoría (450,000 secuencias) tiene un porcentaje de GC entre 35-40% por tanto indica que no hay una buena distribución y hay muchas secuencias con tendencia a tener >35% (fig. 6)

Fig. 7 Indica que todas las bases fueron nombradas y no hubo ninguna que no se reconociera.

Como se indicó al inicio, no señalo el mínimo y el máximo en longitud de secuencias, la mayoría o me atrevería a decir que todas tienen longitud de 190 bases (fig. 8)

Los niveles de duplicación son muy altos, de hasta el 60% lo que podría significar que hay un bajo nivel de cobertura de la secuencia objetivo, y por tanto al eliminar los duplicados solo haría que nos quedemos con el 1.58% de secuencias (fig. 9)

Tener sobreexpresión de secuencias tampoco es bueno porque también indica que hay contaminación de organismos (tan solo 3 probablemente) y por otro que no es tan diverso o que también sean bastante signantes biológicamente. (fig. 10)

Finalmente, la cantidad de adaptadores universales Illumina es alta, de hasta el 40% en las posiciones 165-169 (fig. 11)

Filtrado de secuencias

Trimmomatic

Es un software que tiene útiles funciones de filtrado para secuencias Illumina paired-end y single-end. [10].

De acuerdo a las funciones que tiene trimmomatic, se traducen un poco para entender el siguiente comando:

Remover adaptadores
(ILLUMINACLIP:TruSeq3-PE.fa:2:30:10)

Remover las principales bases de baja calidad o bases N (debajo de calidad 3)

(LEADING:3)

Remover las bases finales de baja calidad o bases N (debajo de calidad 3) (TRAILING:3)

Fig. 8: Distribución de la longitud de la secuencia

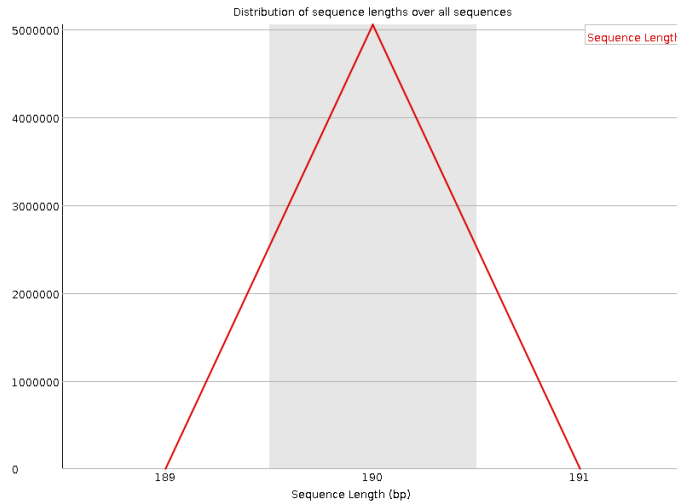
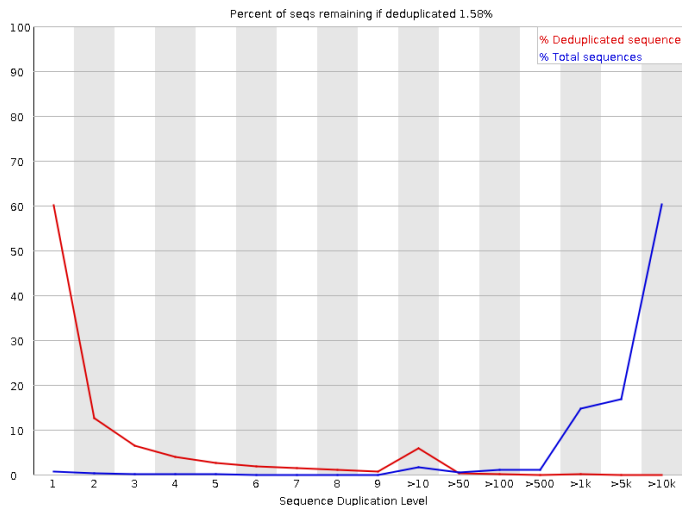


Fig. 9: Niveles de duplicación de las secuencias



Escanea la lectura (read) a razón de 4 en 4 bases y cortando cuando el promedio de calidad cae por debajo de 15 (SLIDINGWINDOW:4:15)

Quita lecturas que estén por debajo de 36 bases de longitud (MINLEN:36)

Entonces el siguiente comando de acuerdo a la sintaxis del programa es para realizar el filtrado ([comando 3](#))

Por lo que cada parte de la estructura indica

1.- La ruta que está después de -jar es para llamar a trimmomatic, al cual se tuvo que buscar para llamarlo.

2.- Por otra parte, el SE es parte del protocolo de trimmomatic para hacer el recorte a fastq: con reads single end.

3.- En tanto -threads es el número de núcleos que usaran (se usaron 2).

4.- Se escoge el tipo de phred al que se va a basar el filtrado, por esta razón dentro de las características que se observaron en el html del fastqc, indicaba que la calidad es en referencia a Illumina 1.9 y por tanto esto lo hace a partir de phred 33.

5.- Lo siguiente se pone la ruta del fastq al que se le quiere limpiar (SRR13867562.fastq), y el siguiente es como debe de salir el resultado en. fastq

(trimmSRR13867562.fastq), aquí se tuvo unos problemas con la ruta del archivo original SRR13867562.fastq, por lo que se decidió cambiar el fastq a la ruta actual a donde se estaba ejecutando el comando de java.

6.- Con la siguiente parte ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 también se decidió copiar el archivo TruSeq3-SE.fa de la ruta original a donde estaba trabajando, por los mismos problemas. Además con las siguientes partes del comando LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36 se añadieron por default como estaba en la descripción anterior.

Mapeo y cobertura

bwa

bwa es un software para realizar el mapeo de secuencias con baja convergencia sobre largas referencias genómicas semejantes al del ser humano. Este funciona bajo tres algoritmos BWA-backtrack, BWA-SW y BWA-MEM

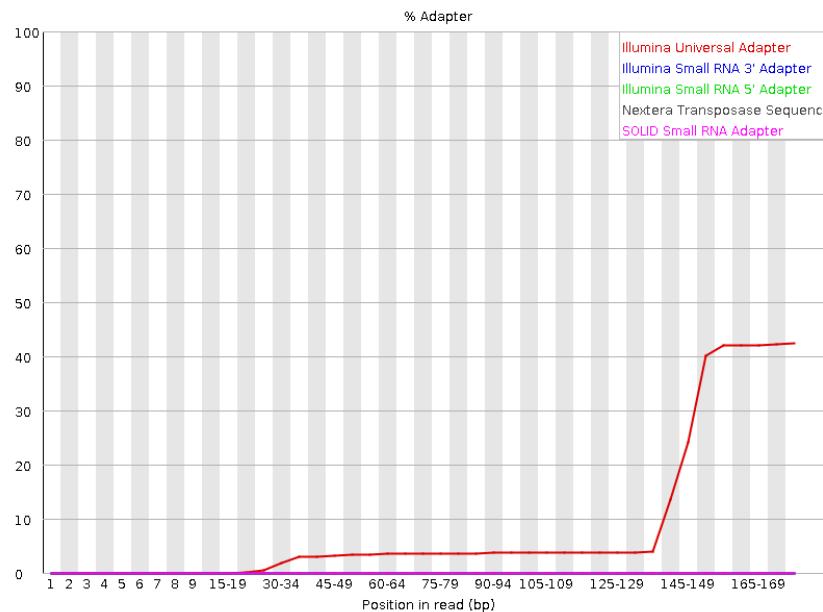
Para obtener la cobertura es necesario hacer el mapeo con bwa, para conseguir el genoma de referencia, este caso será el genoma SARS-CoV-2 debe ser extraído de la base de datos NCBI, entonces en la base de datos de genomas, se escribe SARS-CoV-2 y en la referencia del genoma (reference genome) que aparece en la parte superior, se presiona, así en la siguiente página abrimos el apartado que dice FASTA y se descarga. Finalmente, el nombre del FASTA se cambió a SARS.fasta.

En una carpeta nueva, y dentro de ella seguimos el protocolo bwa index ref.fa [\[11\]](#) para hacer el índice, entonces con el [comando 4](#) podemos ver que salieron nuevos archivos, los cuales serán necesarios para el mapeo. Con la siguiente sintaxis bwa mem ref.fa reads.fq > aln.sam, [\[12\]](#) ya que los reads considere por consenso que en la mayoría de ellos son de 70bp < reads < 190bp, además cabe destacar que provienen de la secuenciación Illumina, tal como señala este repositorio. Entonces el comando resulta en ([comando 5](#))

Fig. 10: Sobreexpresión de secuencias, en la bitácora se muestran el total de ellas

Sequence	Count	Percentage	Possible Source
AGGCAAAGTGTGACGTGTGTTTCTCGTTGAAACAGGGACAAGGC	102026	2.0186203911010234	No Hit
CTGCAGGCTGCTTACGGTTTCGTCCGTGTGACGCCATCATCAGCACAT	75126	1.4863944043857003	No Hit
GTCAAGCTGTACGGCCAATGTTAATGCACTTTTATCTACTGATGGTAAC	73655	1.4572901506140186	No Hit
ACGCCTAAACGAACATGAAATTTCTTGTGTTTCTTAGGAATCATCAAACT	73178	1.447852537392338	No Hit
AGGGGCTGAACATGTCAACAACATCATATGAGTGTGACATACCCATTGGTG	69383	1.3727671240248789	No Hit
AGCTAGTGGGGGACAACCAATCACTAATTGTGTTAAGATGTTGTGTACAC	69239	1.3699180332409753	No Hit

Fig. 11: Contenido de adaptadores



Comando Trimmomatic

```
$ java -jar /app/anaconda3/opt/trinity-2.9.1/trinity-plugins/Trimmomatic/trimmomatic.jar SE
-threads 2 -phred33 SRR13867562.fastq trimmSRR13867562.fastq ILLUMINACLIP:TruSeq3-
SE.fa:2:30:10 LEADING:3 TRAILING:
3 SLIDINGWINDOW:4:15 MINLEN:36
TrimmomaticSE: Started with arguments:
-threads 2 -phred33 SRR13867562.fastq trimmSRR13867562.fastq ILLUMINACLIP:TruSeq3-
SE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences,
0 reverse only sequences
Input Reads: 5054244 Surviving: 4828483 (95.53%) Dropped: 225761 (4.47%)
TrimmomaticSE: Completed successfully

$ ls
SRR13867562.fastq TruSeq3-SE.fa trimmSRR13867562.fastq
```

Comando bwa index

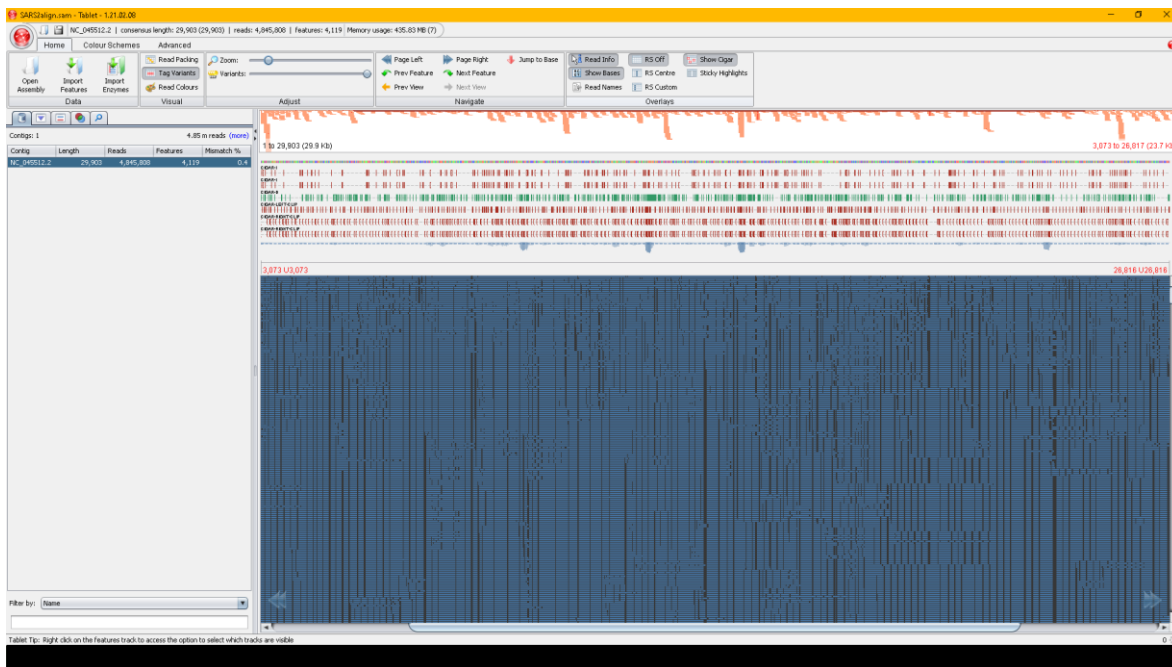
```
(base) [ SARS]$ bwa index SARS.fasta

(base) [ SARS]$ ls
SARS.fasta SARS.fasta.amb SARS.fasta.ann SARS.fasta.bwt SARS.fasta.pac SARS.fasta.sa
```

Comando bwa mem

```
(base) [ SARS]$ bwa mem SARS.fasta ../SRR13867562.fastq >SARS2align.sam
(base) [ SARS]$ ls
SARS.fasta SARS.fasta.amb SARS.fasta.ann SARS.fasta.bwt SARS.fasta.pac SARS.fasta.sa
SARS2align.sam
```

Fig. 12: Tablet, mapeo con el genoma de referencia



Tablet

Es un visualizador gráfico de ensamblajes y alineamientos NGS de alto rendimiento [13]

Entonces para obtener la cobertura, con Tablet en la parte que dice home > Open assembly y en el primer recuadro se coloca el archivo .sam y en el segundo recuadro se pone .fasta, con el ensamblaje terminado (fig. 12), para obtener la cobertura se dirigió a advanced y en donde dice coverage se seleccionó y también Coordinates para finalmente ver el porcentaje de cobertura

1. C = coverage
2. L = read Length (bp)
3. N = number of reads
4. G = haploid genome Length (bp)

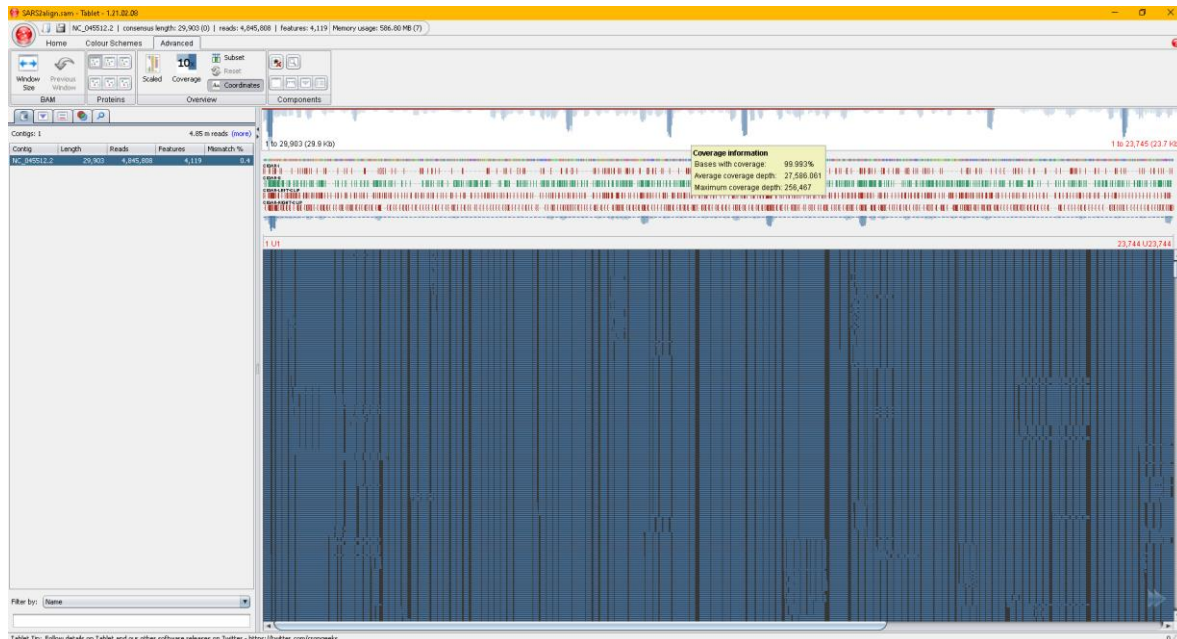
Entonces con G considere la longitud del fasta

```
# grep -v ">" SARS.fasta | wc -c
30332
```

así como $L = 190bp$ tales que pueden ser menores o iguales a 190bp y N es tomado de tablet **4,845,808**.

Por tanto, $C = (L \times N) / G$ talque $C = (190bp \times 4,845,808) / 30,332bp = 30,354x$

Fig. 13: Tablet, cobertura



Así mediante la ecuación de Lander-Waterman [14] también calculamos la cobertura, (cálculos 1)

Comando spades

```
$ spades.py -t 4 -s trimmSRR13867562.fastq -o Ensamble/
```

Ensamble

Spades

Es un ensamblador de genomas, además tiene diferentes opciones para realizar los ensambles [15a]

De acuerdo con la página oficial de spades [15b] en el manual estructuramos el comando así `spades.py [options] -o <output_dir>` entonces se le indica que se active spades con `conda activate spades` y se corre este comando (comando 6) entonces

1.- Con `-t` indica los núcleos, osea 4

2.- Con `-s` le indica el archivo (single end) .fastq, este caso el que ya está filtrado

3.- Por default no está especificada la opción `-k`, pero este corre con los tamaños de k-meros 21, 33, 55 y 77 por que spades estimo que las longitudes de los reads son de 190 osea ≥ 150

4.- finalmente `-o` es la opción para señalar la carpeta (este caso Ensamble Por tanto los archivos resultantes son: (results)

Comando quast

```
(base) [ Ensamble]$ quast.py --split-scaffolds -t 4 -r ../SARS/SARS.fasta
scaffolds.fasta -o REF_CALIDAD_ENSAMBLE/
(base) [ Ensamble]$ cd REF_CALIDAD_ENSAMBLE/
(base) [ REF_CALIDAD_ENSAMBLE]$ ls
aligned_stats  genome_stats  quast.log      report.tex    transposed_report.tex
basic_stats    icarus.html   report.html    report.tsv    transposed_report.tsv
contigs_reports  icarus_viewers  report.pdf     report.txt    transposed_report.txt
```

Quast

Este es una herramienta para evaluar los ensambles por varias métricas.

Entonces este comando se traduce como

`--split-scaffolds` esta opción indica que hace el análisis por scaffolds

`-t` indica el número de núcleos a usar (este caso 4)

`-r` indica la referencia y después se pone la referencia por consiguiente, el nombre del fasta con scaffolds

`-o` finalmente la salida a la carpeta en donde se quiere que se guarden los registros


```
(spades) [ Ensamble]$ ls
K21  assembly_graph.fastg          contigs.fasta  input_dataset.yaml  run_spades.sh      spades.log
K33  assembly_graph_after_simplification.gfa  contigs.paths  misc                run_spades.yaml    tmp
K55  assembly_graph_with_scaffolds.gfa        corrected     params.txt          scaffolds.fasta    warnings.log
K77  before_rr.fasta                       dataset.info   pipeline_state      scaffolds.paths

(spades) [ Ensamble]$ conda deactivate
(base) [ Ensamble]$
```

entonces el **porcentaje de ensamble** fue de tan solo el 46.96%;
 el **radio de duplicación** indica que muchos contigs cubren una misma región ya que el número es >1;
 el **alineamiento más largo** es de 1501;
 la cantidad de bases alineadas correctamente es de 14,433;
 el parámetro NGA50 es nulo ya que no fue necesario contar por bloques de alineamientos ya que no hubo contigo mal ensamblados (por lo que entendí) lo necesario para tenerlos que romper y seguir este protocolo;
 en los **ensamblajes incorrectos** encontré 4 contigs de los cuales se tuvieron que **relocalizar**, o **invertir** entre strands o **buscar en otro cromosoma**;
 la **longitud total de las bases de los contigs** que contienen bases mal ensambladas es de 4628;
 en **promedio el número de bases que no hubo encuentro sobre 100kbp** es de 106.81bp;
 el **número de mutaciones promedio que se encontro en 100kbp** es de ~25;
 todo es en estadísticas sobre 17 **contigs**;
 Con el contig **mínimo** de 1509bp y **máximo** de 14,456bp

1. En el gráfico de Nx si solo se conservaran los contigs con longitudes de 523bp a más bases, se lograría completar por lo menos el ~90% del ensamble y por el contrario, si conservamos solo los contigs con longitudes de 1509bp a más, solo completaríamos ~10% de un ensamble
2. Este grafico NAX es como el caso anterior, solo que es con bloques, por lo que si conservamos bloques de 81bp a más, se completaría el 99.3% del ensamble y si conservamos con longitudes de 1501bp a más solo completaría el ~10%
3. En el gráfico de NGx si conservamos contigs con 517bp o más, solo completaríamos el ~48% del genoma de referencia, mientras que, si conservamos los de 1501bp o más, solo completaríamos el ~10%
4. Este grafico NGAx, si conservamos bloques de 81bp a más, se completaría el ~48% del genoma de referencia y si conservamos con longitudes de 1501bp a más solo completaría el ~5%
5. La grafica misassemblies indica que, si hay contigs hasta como máximo un ensamble incorrecto, se cubriría el 37.8% del genoma y si fueran de hasta como máximo 4 ensambles incorrectos, entonces se cubriría el 48.3% del genoma
6. Y para concluir el grafico GC content podemos denotar que si los contigs se rompieran en ventanas de 100bp y contáramos el porcentaje de GC que tiene cada una, encontraríamos que en los scaffolds hay un máximo de hasta 18 ventanas con 34% de GC, mientras en la referencia como máximo hasta 26 ventanas con 37% de GC

Discusión y Conclusiones

Por tanto, después de realizar este proyecto y no poder concluir correctamente para poder identificar los genes y las funciones de cada uno. Si pude enterarme todo lo que abarca la bioinformática para poder ser practicada, debe de ser entendida desde cómo funcionan los softwares hasta los requerimientos de cada cosa

En tanto a como salieron cada uno de los pasos, sin duda es de pensar que falta mas minuciosidad a la hora de realizar cada comando y revisar bien los parámetros, pero fue el mejor intento de hacerlo, ya que pude darme cuenta de muchas cosas que solo vienen en los manuales del que hizo el software, y por ello es que hubo un buen análisis de calidad, así como de un filtrado y un mapeo, solo que en el ensamble, por falta de rigurosidad pues solo se pudo ensamblar aproximadamente el 50% con respecto al genoma de referencia del SARS-CoV-2.

Bibliografía

- 1.- Instituto Nacional del Cáncer (2 de julio de 2020) *SARS-CoV-2* Definición de SARS-CoV-2 <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/sars-cov-2>
- 2.- Secretaría de Salud (12 de marzo de 2020) *COVID-19 PREGUNTAS FRECUENTES* Gobierno de México <https://www.gob.mx/salud/documentos/covid-19-preguntas-frecuentes>
- 3.- World Health Organization (2021) *Coronavirus disease (COVID-19)* World Health Organization https://www.who.int/health-topics/coronavirus#tab=tab_1
- 4.- WHO-China. (2021). WHO-convened Global Study of Origins of SARS-CoV-2: China Part. <https://www.who.int/publications/i/item/who-convened-global-study-of-origins-of-sars-cov-2-china-part>
- 5.- Kumar A, Prasoon P, Kumari C, Pareek V, Faiq MA, Narayan RK, Kulandhasamy M, Kant K. SARS-CoV-2-specific virulence factors in COVID-19. *J Med Virol.* 2021 Mar;93(3):1343-1350. doi: 10.1002/jmv.26615. Epub 2020 Nov 1. PMID: 33085084.
- 6.- Mackenzie R. J., (6 de abril de 2018) *RNA-seq: conceptos básicos, aplicaciones y protocolo* News Curier <http://www.news-courier.com/genomics/articles/rna-seq-basics-applications-and-protocol-299461>
- 7.- National Human Genome Research Institute (27 sept 2019) *Transcriptoma* National Human Genome Research Institute <https://www.genome.gov/es/about-genomics/fact-sheets/Transcriptoma>
- 8.- Klymenko A. *The NCBI SRA (Sequence Read Archive)* Github <https://github.com/ncbi/sra-tools>
- 9.- Simon Andrews *FastQC* Babraham Bioinformatics <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>
- 10.- Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina Sequence Data. *Bioinformatics*, btu170. <http://www.usadellab.org/cms/?page=trimmomatic>
- 11.- Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25, 1754-1760. [PMID: 19451168] (no es la cita que es para el algoritmo que se usó, pero sirve para tenerlo en cuenta) <http://bio-bwa.sourceforge.net/bwa.shtml#3>
- 12.- Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25, 1754-1760. [PMID: 19451168] (Leer el README.md) <https://github.com/lh3/bwa>
- 13.- Milne I, Stephen G, Bayer M, Cock PJA, Pritchard L, Cardle L, Shaw PD and Marshall D. 2013. Using Tablet for visual exploration of second-generation sequencing data. *Briefings in Bioinformatics* 14(2), 193-202. <https://ics.hutton.ac.uk/tablet/>
- 14.- The Sequencing Center (sin fecha) *What is sequencing coverage?* The Sequencing Center <https://thesequencingcenter.com/knowledge-base/coverage/>
- 15.- Prjibelski A., Antipov D., Meleshko D., Lapidus A., Korobeynikov A. (19 de junio 2020) Using SPAdes De Novo Assembler a) <https://doi.org/10.1002/cpbi.102>
b) https://github.com/ablab/spades/blob/spades_3.15.3/README.md
- 16.- Alla Mikheenko, Andrey Prjibelski, Vladislav Saveliev, Dmitry Antipov, Alexey Gurevich, Versatile genome assembly evaluation with QUASt-LG, *Bioinformatics* (2018) 34 (13): i142-i150. doi: 10.1093/bioinformatics/bty266 First published online: June 27, 2018 <http://quast.sourceforge.net/docs/manual.html>
- 17.- Poojary, M., Jolly, B., Scaria, V., & Shantaraman, A. (2020). Computational Protocol for Assembly and Analysis of SARS-nCoV-2 Genomes. *Research Reports*, 4, 1–14. DOI: 10.9777/rr.2020.10001 <https://www.researchgate.net/publication/341592466>