Análisis de los genes de resistencia a antimicrobianos y su presencia en plásmidos en cepas de *Staphylococcus aureus* resistentes a meticilina

Titulación: Máster en Bioinformática

Alumno/a: Martín Dorta, Lorena

Convocatoria: Julio, 2022

Curso académico: 2021/2022

DNI: 45854091A

Orientación: Herycel Cristina Contreras

Lugar de residencia, mes y año: San Cristóbal de La Laguna, Tenerife.

Director/a: Martín Quijada, Narciso

Créditos: 9 ECTS

De:
Planeta Formación y Universidades

Quiero agradecer a mi tutor, Narciso, por haberme ayudado y aconsejado en todo momento a la hora de hacer este trabajo. Así como a mi novio, Guillermo, que me ha escuchado y apoyado durante todo este proceso.

Índice de contenido

Índice de tablas

Índice de gráficos y figuras

Lista de abreviaturas

ADN: Ácido desoxirribonucleico

AMR: Resistencia antimicrobiana (Antimicrobial Resistance)

CDS: secuencias codificantes de proteínas (CoDing Sequence)

MRSA: *Staphylococcus aureus* resistente a meticilina (Methicillin-Resistant *Staphylococcus aureus*)

HGT: Transferencia genética lateral u horizontal (Horizontal Gene Transfer)

MGE: Elementos genéticos móviles (Mobile Genetic Elements)

NGS: Secuenciación de nueva generación (Next-Generation Sequencing)

WGS: Secuenciación de genoma completo (Whole-Genome Sequencing)

Resumen y palabras clave

Actualmente, las bacterias resistentes a antibióticos son un problema a escala mundial. Una de las más conocidas son las cepas de *Staphylococcus aureus* resistentes a meticilina (MRSA) que se han desarrollado en ambientes hospitalarios, aunque no son exclusivos de ellos. Gracias a los progresos de la secuenciación y análisis bioinformáticos, podemos saber más acerca de los elementos genéticos de estas cepas y poder elaborar estrategias para combatirlas. Durante este trabajo, se analizarán varias cepas secuenciadas de *S.aureus* obtenidas del trabajo de Harris SR *et al.* [10] para detectar los elementos genéticos implicados en las resistencias a antibióticos, así como de su posible presencia en elementos genéticos móviles, centrándonos en el caso de las resistencias a meticilina.

Palabras clave: *Staphylococcus aureus*, MRSA, secuenciación genómica, meticilina, análisis bioinformático, AMR, MGE.

Abstract and keywords

Nowadays, resistant bacteria are a global problem. One of the most known of them are the methicillin resistant of *Staphylococcus aureus* (MRSA) strains which started in hospital environments, although they are not exclusive in those environments. Due to the progress in the sequencing and bioinformatics analysis, we can know more about the genetic elements of these strains and make new strategies to fight them. In this project, we are going to analyse the sequencing data of *S.aureus* that has been used in the study conducted by Harris SR *et al*. [10] to detect the genetic elements involve in the antibiotic resistant, besides of the possibility of being in others mobile genetic elements, especially in the case of methicillin resistant.

Keywords: *Staphylococcus aureus*, MRSA, genomic sequencing, methicillin, bioinformatic analysis, AMR, MGE.

Introducción

Actualmente, se conoce que muchos patógenos microbianos son muy versátiles cuando llega el momento de colonizar nuevos nichos ecológicos, así como huéspedes animales y/o humanos y causarles diferentes tipos de infecciones [1]. El género de *Staphylococcus* incluye bacterias que son parte de la microbiota habitual de la piel y mucosas y que pueden actuar como patógenos oportunistas en determinadas circunstancias causando enfermedades tras el tratamiento con antibióticos [2]. Este género está formado por bacterias gram-positivas, que, aunque pueden ser comunes en humanos, algunas especies y/o cepas pueden actuar como patógenos donde el más famoso de ellos es *Staphylococcus aureus* [3]. Existen varias cepas de esta bacteria que, junto con otras especies del mismo género, pueden causar muchas enfermedades mortales cuya peligrosidad es debida a la gran resistencia que tienen frente a muchos agentes antimicrobianos [4]. Específicamente, las cepas de *S. aureus* resistentes a meticilina (MRSA) son las mayores responsables de las enfermedades que se producen en entornos hospitalarios [5]. Algunas de estas enfermedades son las infecciones nosocomiales que suponen un gran riesgo, ya que se contagian de los pacientes infectados al personal sanitario y, posteriormente, estos pueden llegar a contagiar a pacientes en situaciones de gravedad [6].

La plasticidad genética de *S. aureus* le confiere la capacidad de evolucionar a nuevas cepas virulentes y resistentes a antimicrobianos (AMR). Asimismo, los antibióticos efectivos que se han estado utilizando para tratar a esta bacteria han ido disminuyendo, ya que ha ido adquiriendo resistencias a distintas penicilinas como puede ser la meticilina (en la cual nos centraremos en este estudio), así como a otros tipos de antibióticos como la vancomicina (glucopéptido) [7].

El genoma de *S. aureus*, al igual que muchas otras bacterias, está conformado por un genoma principal y en uno o varios genomas accesorios en muchos casos. El genoma principal contiene todos los genes básicos esenciales para la supervivencia como aquellos genes que controlan el metabolismo. Las bacterias pueden obtener información genética de otras células o de su ambiente por tres métodos distintos: adquisición del ADN del ambiente por transformación, transducción de bacteriófagos y por contacto directo entre células, también llamado conjugación. Esta nueva información se integra en la bacteria en forma de esos genomas accesorios o puede llegar a formar parte del genoma principal [3]. Los elementos genéticos móviles (MGE) son estos fragmentos de ADN que pueden contener una gran variedad de factores de virulencia y resistencia, así como enzimas que pueden controlar su propia transferencia e integración en otros posibles huéspedes. Estos MGE tienen una cierta movilidad y su transferencia entre células diferentes recibe el nombre de transferencia genética lateral u horizontal (HGT) que puede ser entre células eucariotas, entre células procariotas, así como entre células procariotas y eucariotas [3].

Se ha comprobado que la HGT es un mecanismo relevante en la evolución de bacterias como *S. aureus*, donde el ejemplo más conocido es el gen *mec*. Este gen se ha transferido por este mecanismo a distintos trasfondos de los cromosomas de esta especie en varias ocasiones. Por tanto, las cepas resistentes a meticilina han evolucionado de manera independiente varias veces en lugar de una única cepa ancestral [1]. Dada la importancia de este gen en la resistencia frente a la meticilina, en este estudio se pretende buscar si este gen se encuentra en algún MGE, además de otras posibles resistencias que pudiera tener esta especie.

Los plásmidos son uno de los mayores MGE que existen dentro de las comunidades bacterianas que se transmiten por HGT. Estos codifican para genes que tienen todos los elementos necesarios para llevar a cabo su propia replicación y, en el caso de los plásmidos conjugativos, contienen también todos los elementos

necesarios para su transferencia a otra bacteria diferente. Además de estos genes para su replicación, pueden albergar diferentes genes accesorios que otorgan ciertas ventajas selectivas para sobrevivir en ciertos ambientes como puede ser la resistencia frente a los diferentes antibióticos [11]. Asimismo, se ha visto que estos plásmidos pueden estar integrados en el cromosoma principal de la bacteria [9]. Además de las resistencias a antibióticos, especialmente los β-lactámicos, también pueden contener resistencia a desinfectantes y metales pesados, de forma que influye en su heterogeneidad [8].

Secuenciar el genoma completo de cepas de MRSA permite conocer en mayor profundidad los MGE de la bacteria que codifica para los AMR, así como la propia patogenicidad de *S. aureus*. Es por esto por lo que en este estudio se realizará un análisis de distintas cepas de MRSA en busca de la presencia en su genoma de genes de AMR, así como la posibilidad de estos de estar integrados en un MGE.

Actualmente, la bioinformática es uno de los campos más recientes dentro de la investigación biológica que consiste en utilizar métodos matemáticos, estadísticos y computacionales para el procesamiento y análisis de datos biológicos [12]. Esto ha provocado que cada vez más se lleven a cabo numerosos métodos de inteligencia artificial y de machine learning [13].

Estos avances se han visto favorecidos por el surgimiento de la gran cantidad de información de datos generados por la secuenciación de ADN de nueva generación (NGS). Estos datos se tratan con herramientas bioinformáticas que ayudan a entender esos datos genómicos generados y convertirlo en información útil para su utilización en la medicina personalizada [13]. Asimismo, la NGS se está usando cada vez más para detectar e identificar variaciones en la secuencia dando una gran cantidad de marcadores genéticos [14]. En el caso de los análisis bioinformáticos de genomas bacterianos obtenidos por NGS, se divide en varios pasos, el primero consiste en procesar las secuencias crudas obtenidas por las plataformas de NGS; el segundo es realizar un ensamblaje para reconstruir el genoma original y detectar las variantes que pudieran haber; y un tercer paso donde se hace un análisis para dar un contexto a toda la información generada durante los pasos anteriores como pudiera ser la anotación del genoma, búsqueda de AMR, entre otros [15].

Objetivos

Objetivos generales

- Realizar un análisis bioinformático completo mediante "whole genome sequencing" (WGS) de varias cepas de MRSA secuenciadas y directamente desde las secuencias crudas
- Buscar las resistencias a antimicrobianos
- Comprobar la presencia de AMR en plásmidos.

Objetivos específicos

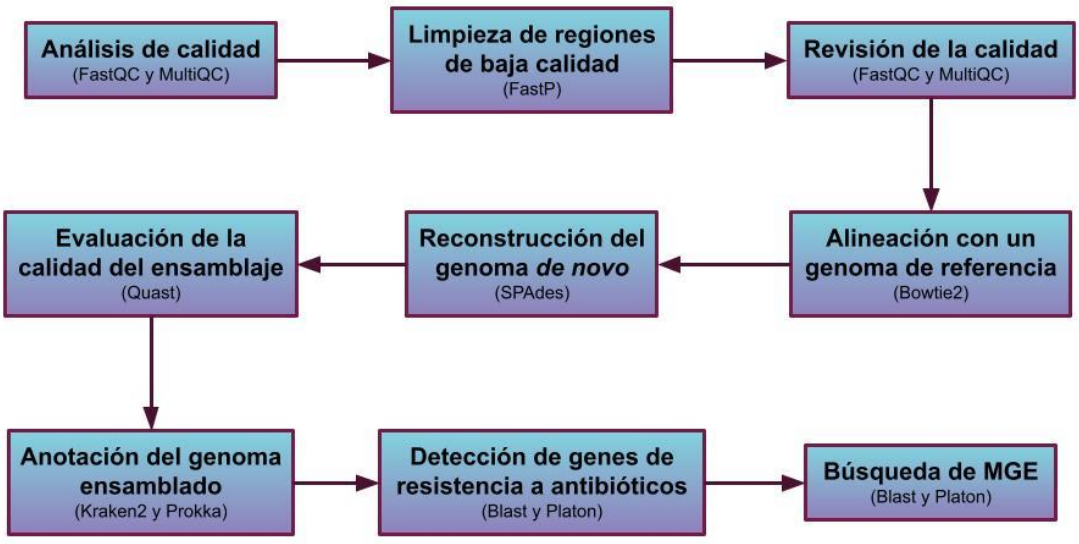
- Dominio de diferentes herramientas bioinformáticas para el análisis, así como su optimización y automatización
- Interpretar los genes de AMR encontrados y su ubicación en las distintas partes del genoma, incluyendo MGE

Metodología

Para realizar este análisis, se utilizaron las secuencias del artículo de Harris SR *et al.* [10] de *S. aureus* resistentes a meticilina que se obtuvieron tras un análisis de un brote en un hospital. Asimismo, se usó el entorno virtual de Amazon WorkSpaces v.4.0.6.2415. Además, se emplearon diferentes herramientas bioinformáticas como puede FastQC v.0.11.9 [17], MultiQC v.1.12 [18], FastP v.0.23.2 [19], Bowtie2 v.2.4.5 [20], SPAdes v.3.15.4 [21] y PRINSEQ-lite v.0.20.4 [22], QUAST v.5.0.2 [23], Kraken2 v.2.1.2 [24], Prokka v.1.14.6 [25], Blast local v.2.12 [26, 27, 28], así como la herramienta platon v.1.6 con sus respectivas bases de datos de plásmidos que tiene asociadas [30, 31, 32, 33, 34]. Además, se utilizaron las bases de datos de resistencia a antimicrobianos de ResFinder [16] y la base de datos de plásmidos de PlasmidFinder [29].

Para el análisis bioinformático de WGS, se siguieron los distintos pasos que se pueden ver de forma simplificada en la Gráfica 1. Brevemente, las herramientas de FastQC como MultiQC, permiten realizar un análisis de calidad de las muestras donde FastQC es un análisis individual de cada muestra y MultiQC genera un resumen de la calidad de todas las muestras analizadas. FastP realiza una limpieza de las muestras según unos parámetros que se establecen y que permite eliminar posibles

adaptadores que haya en estas. Bowtie2 permite alinear nuestras muestras a un genoma de referencia, en este caso, el genoma del fago phiX, el cual se usa como control de la secuenciación, con el fin de eliminar posibles restos de la dataset. La herramienta SPAdes es un ensamblador que intenta reconstruir el genoma completo a partir de las secuencias filtradas. La herramienta Prinseq-lite se utiliza para eliminar aquellos contigs más pequeños que pueden causar ruido de fondo. La calidad resultante de este ensamblaje se comprueba con la herramienta Quast atendiendo a valores como el número de contigs o la N50 para investigar cómo de fragmentados están los genomas ensamblados. La herramienta de Kraken2 permite asignar etiquetas taxonómicas a las muestras y se utilizó para confirmar si todas las muestras pertenecen a *S.*aureus. Prokka predice CDS y los anota según su base de datos del NCBI de AMR o de UniProtKB. Por último, con Blast, concretamente blastn, se utilizó para realizar búsquedas y detectar coincidencias en diversas bases de datos para identificar los AMR y los replicones plasmídicos, al igual que con la herramienta de Platon solo que, en el caso de esta última, contiene información de distintas bases de datos de plásmidos.

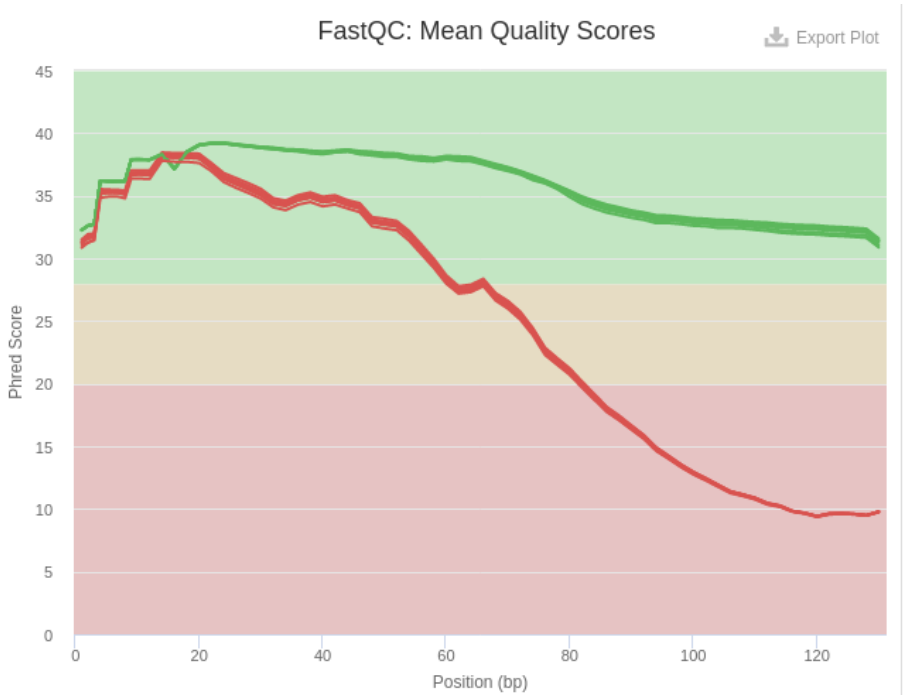


Gráfica 1. Esquema del flujo de trabajo que se va a seguir durante este proyecto.

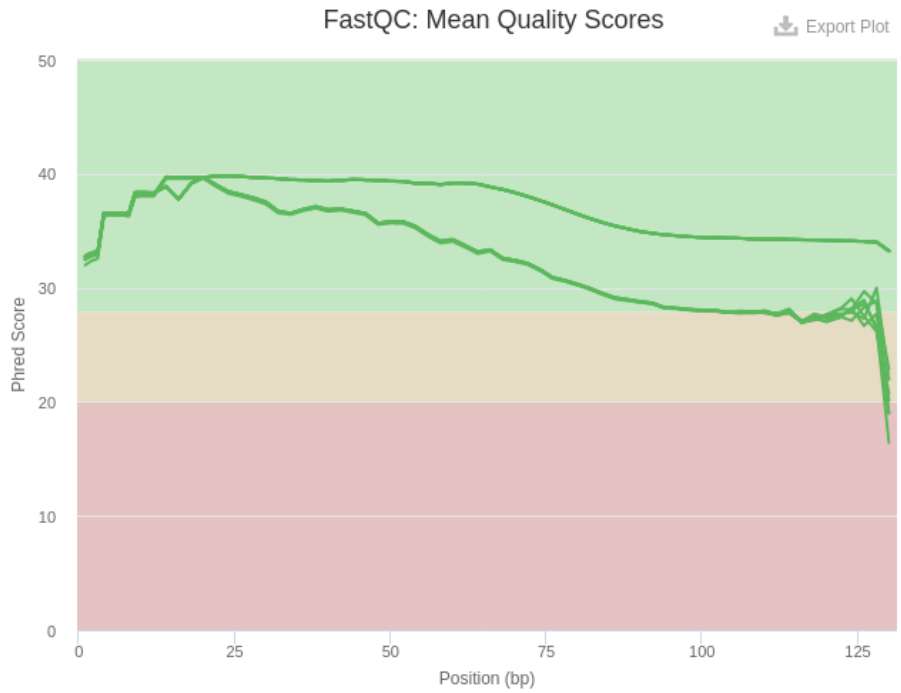
El primer paso del trabajo desarrollado en este TFM consistió en realizar un primer análisis de calidad de las muestras para saber si es necesario hacer una posterior limpieza. Para ello, se empleó la herramienta de FastQC, donde las muestras R1 tienen una mejor calidad en comparación con las R2. Por tanto, es necesario hacer un proceso de limpieza para estas secuencias con mala calidad, especialmente en el

extremo 3'. Asimismo, se utilizó la herramienta MultiQC que permite analizar la calidad de todas las secuencias simultáneamente y nos devuelve un reporte general de la calidad (ver Gráfica 2).

Para hacer el proceso de limpieza, se utilizó la herramienta FastP que permite eliminar los posibles adaptadores que hayan podido quedar en las secuencias, así como refinar la calidad de estas. Como se desconoce el tipo de adaptador que se usó en el artículo general, se añade al programa un fichero .fasta que contiene todos los tipos de adaptadores usados por Illumina descargado de un repositorio de GitHub creado por Bolger T y Usadel B (https://github.com/usadellab/Trimmomatic/tree/main/adapters) [35]. Al igual que en el paso anterior, se hace un análisis de la calidad con FastQC y un análisis general con MultiQC para comprobar que la limpieza ha sido efectiva (ver Gráfica 3) (para ver los resultados de FastQC para una secuencia, ver Gráfica 4 del Anexo).



Gráfica 2. Análisis de calidad por MultiQC de las secuencias crudas. Las muestras en rojo se corresponden a las muestras R2 que tienen peor calidad que las R1 mostradas en verde. Esta peor calidad viene determinada por su puntuación Phred donde se considera que, a partir de un Phred score de 28 (división entre la zona de color amarillo y la zona de color verde), ya se considera una buena calidad.



Gráfica 3. Análisis de calidad por MultiQC de las secuencias limpias. Como se observa, las muestras han mejorado la calidad, aunque desciende un poco en el extremo 3' como es usual.

El estricto proceso de limpieza mediante la herramienta FastP desechó un 19.2% de las secuencias iniciales. El 80.8% de las secuencias restantes (pasaron una media de 755366 secuencias de las muestras) obtuvieron un valor de calidad apropiado para proseguir con el flujo de trabajo.

El siguiente paso fue alinear las secuencias limpias con el genoma del virus phiX que suele utilizarse como control del proceso de la secuenciación de Illumina y, para ello, se utiliza la herramienta de bowtie2 de tal forma que solo se obtendrá el genoma de la bacteria. Una vez filtrados, se obtuvieron una media 755366 secuencias para cada muestra que son las mismas que las que se obtuvieron para las secuencias ya limpias, por tanto, se puede decir que estas secuencias no tenían el genoma del virus (Tabla 6 del Anexo).

Una vez filtradas las secuencias, se someten a un proceso de ensamblado que consiste en la unión de estas secuencias para reconstruir el genoma original de la bacteria. Para ello, se hará un ensamblaje *de novo* usando la herramienta de SPAdes donde se conserva el archivo contigs.fasta de cada una de las muestras. Asimismo, aquellos contigs menores de 200 pares de bases fueron eliminados, ya que pueden ser producto de errores o ruido de fondo por medio de la herramienta prinseq-lite. Una

vez eliminados estos contigs, se procede a evaluar la calidad del ensamblaje realizado con la herramienta quast (ver Tabla 7 del Anexo).

A partir de los contigs generados, se puede elaborar una etapa de anotación del genoma que fue ensamblado con anterioridad. Esta asignación taxonómica se realiza por medio de la herramienta de Kraken2 que permite confirmar que el genoma que hemos ensamblado proviene de *S. aureus* exclusivamente. La predicción de CDS del genoma de cada una de las cepas, así como la anotación funcional de las proteínas resultantes, se realizó mediante la herramienta Prokka.

La búsqueda de genes AMR se realizó mediante un alineamiento de cada genoma con la herramienta blastn contra la base de datos de ResFinder (Zankari et al. [16]), la cual tuvo que ser formateada previamente como se indica a continuación. Se combinaron todos los archivos .fsa, que contenían las secuencias de los genes de resistencia divididos para cada clase de antibióticos, en un único archivo llamado resfinder.fasta para tener, en este único fichero, todas las familias de resistencias y facilitar la búsqueda posterior por blast. Para hacer esta base de datos, se utiliza la herramienta makeblastdb por medio del comando:

```
makeblastdb -in resfinder.fasta -dbtype nucl -
out AntibioticResistanceDB
```

Lo siguiente fue realizar un blastn para cada una de las muestras contra esta base de datos. El mismo procedimiento se realizó para buscar los replicones plasmídicos contenidos en la base de datos de PlasmidFinder [26], posteriormente formateada para ser reconocible por blast, por el comando:

```
makeblastdb -in plasmidfinder.fasta -dbtype nucl -
out PlasmidsDB
```

A continuación, se utiliza la herramienta Platon que permite calcular el RDS (puntuación relacionada con la distribución de los replicones) para cada contig como una medida para discernir si se trata de un contig plasmídico o cromosómicos. Una vez obtenidos todos estos resultados, se optimizaron scripts en BASH para automatizar la generación de tablas que agrupan toda la información obtenida de los

procesos anteriores que facilitarán el posterior análisis de los resultados (cuyo código y función se describe con detalle en el Anexo).

Resultados y discusión

Tras realizar toda la parte de análisis de calidad y de ensamblaje, se obtuvieron unos buenos resultados en general, exceptuando las muestras ERR131806 y ERR131807 que obtuvieron muchos más contigs de forma que su N50 resultó inferior en comparación con el resto, indicando que su genoma está más fragmentado en comparación (ver Tabla 1).

Nombre de la muestra	Nº de secuencias crudas	Nº de secuencias filtradas	Nº de contigs	Tamaño total del contig	Longitud media del contig	N50
ERR131800	1313568	1056893	70	2766006	39514.4	118268
ERR131801	878405	706679	73	2820778	38640.8	107894
ERR131802	853836	693456	71	2824264	39778.4	106622
ERR131804	774828	638379	72	2826480	39256.7	98863
ERR131805	927654	756633	81	2819932	34814	106680
ERR131806	890448	717491	206	2812806	13654.4	33491
ERR131807	903066	718032	196	2809333	14333.3	38221

Tabla 1. Resultado de los análisis hasta el ensamblado.

El análisis por Kraken2 asignó taxonómicamente los genomas como *S. aureus*, confirmando la taxonomía indicada en la publicación de Harris SR et *al.* [10].

La búsqueda de genes AMR reveló que los genomas contenían distintos tipos de AMR más allá de los estrictamente relacionados con la resistencia a meticilina (el gen *mecA*), como son los genes *blaZ*, *aph (3')-III*, *msr(A)*, *mph(C)*, *dfrG*, *aac (6')-aph(2'')*, *aadD*, *bleO* y *erm(C)* (ver Tabla 4).

Como se puede ver, los genomas contenían genes de resistencia a beta-lactámicos, como *mecA* y *blaZ* los cuales dan resistencia frente a antibióticos como amoxicilinas, ampicilinas y, en el caso de *blaZ*, también frente a penicilinas. Además,

se comprobó que los genomas tenían muchas resistencias frente a aminoglucósidos al contener los genes *aph (3')-III*, *aadD*, *aac (6')-aph(2'')* y *bleO* que otorgan resistencias frente a antimicrobianos como pueden ser la kanamicina, amikacina, tobramicina, estreptomicina y bleomicina. Otro gran grupo de resistencias encontrado fueron los macrólidos como el gen *mph(C)*, aunque también se encontraron variantes como el gen *erm(C)* que pertenece al grupo de los MLS_B (Macrólido, Lincosamida, Estreptogramina B) y el gen *msr(A)* que forma parte de los macrólidos y estreptogramina B. Todos estos genes le proporcionan a la bacteria resistencias frente a antibióticos como pueden ser la eritromicina, espiramicina, lincomiacina y la telitromicina. Asimismo, también se encontró el gen *drfG* que afecta a la síntesis de purinas y es un antagonista de la ruta bioquímica del folato, con lo que otorga resistencias frente a las trimetoprimas.

Si se tiene en cuenta los resultados de Prokka y los resultados de la búsqueda de resistencia a antibióticos, se puede comparar los resultados, de tal forma que se puede observar los posibles MGE que hay alrededor de los genes de resistencia (ver Tabla 2). Asimismo, se observaron los CDS que se encontraban los mismos contigs en los que se encontraron los genes *mecA* y *blaZ* (ver Tabla 3).

Muestra	Contigs	Número de CDS
ERR131800	70	2544
ERR131801	70	2544
ERR131802	71	2625
ERR131804	72	2633
ERR131805	81	2623
ERR131806	206	2604
ERR131807	196	2577

Tabla 2. Resultado general de Prokka para las diferentes muestras

Muestra	gen	CDS
ERR131800	mecA	58
	blaZ	40
ERR131801	mecA	58
	blaZ	40
ERR131802	mecA	5
	blaZ	38
ERR131804	mecA	202
	blaZ	51
ERR131805	mecA	5

	blaZ	38
ERR131806	mecA	5
	blaZ	15
ERR131807	mecA	5
	blaZ	15

Tabla 3. Cantidad de CDS encontrados en los mismos contigs en los que se encontraron los genes mecA y blaZ para cada muestra tras ejecutar Prokka.

El análisis del entorno genético de los genes *mecA* y *blaZ* utilizando los archivos de anotación gff generados por prokka, demostró que suelen contener primero una transposasa que, en algunas de las muestras como puede ser la ERR131804 y ERR131805, la transposasa es para el transposón Tn554.

Los genes de *mecA* y *blaZ* no fueron localizados en ningún plásmido en función de los resultados de Platon y PlasmidFinder (ver Tabla 5), con lo que potencialmente estarían integrados en el genoma principal de la bacteria. Sin embargo, gracias al análisis realizado con Prokka, se puede comprobar que estos dos genes, especialmente en el caso de *mecA*, están rodeados por transposasas que podrían permitir el movimiento de estos genes, tanto a otras regiones del genoma, como a un genoma secundario como puede ser el caso de los plásmidos. De esta forma, el movimiento de estos genes de resistencia puede permitir el traspaso de estos a otras bacterias, tanto de la misma especie como a otras distintas, otorgándoles a estas nuevas bacterias una resistencia de la que antes carecían.

Es por esta gran cantidad de resistencias y su capacidad para transmitirse fácilmente por lo que hace que esta clase de bacterias sean muy peligrosas. Por tanto, es importante destacar el uso de este tipo de análisis para evitar usar antimicrobianos inapropiados en este tipo de bacterias que han obtenido resistencias y que podrían matar a aquellas que no las poseen, lo que favorecería la propagación de bacterias resistentes en detrimento de la microbiota habitual sensible.

Gracias a estos análisis bioinformáticos, se puede comprobar las AMR que contienen los genomas de los patógenos pudiendo así elegir el tipo de antimicrobiano más efectivo para acabar con estas poblaciones sin que se generen nuevas resistencias en otras bacterias patogénicas peligrosas o que puedan matar a bacterias beneficiosas de la microbiota habitual. No obstante, para realizar estos análisis bioinformáticos de WGS, se requieren varios días en realizarlo (teniendo en cuenta

con el aislamiento y crecimiento de la bacteria, extracción de ADN, su secuenciación y análisis) que pueden ser cruciales para el tratamiento del paciente. Por tanto, es necesario seguir investigando en este campo para reducir los tiempos requeridos en el proceso, de tal forma que las plataformas de secuenciación, los programas y las herramientas sean cada vez más óptimas y eficientes.

El análisis de MGE por medio de plasmidfinder y Platon reveló que las muestras ERR131802, ERR131805, ERR131806 y ERR131807 tienen un plásmido en la misma zona que contiene el gen de resistencia; no obstante, la detección de los plásmidos es distinta para cada base de datos utilizada (ver Tabla 5).

La diferencia en la detección de los plásmidos se debe a que, mediante plasmidfinder, solo identifica aquellos replicones plasmídicos que estén contenidos en su base de datos [29]. Por otro lado, la herramienta de Platon permite detectar contigs relacionados o parecidos a plásmidos que hayan sido ensamblados a partir del genoma bacteriano. Para ello, realiza un análisis complementado con una caracterización de estos contigs, así como el uso de varios filtros heurísticos para predecir y buscar secuencias de proteínas frente a la base de datos personalizada que está compuesta de marcadores de secuencias proteicas y puntuaciones de la distribución relacionada con los replicones, además de una última búsqueda en bases de datos de plásmidos que se encuentran en el NCBI [31].

Es por estas diferencias por lo que se podría pensar que el análisis realizado con Platon es mucho más estricto y, por lo tanto, la fiabilidad de que ese AMR se encuentre en un plásmido es mucho mayor; aunque este sea más restrictivo.

Muestra	Nombre del contig	Gen	% identidad	Comienzo del gen en el contig	Final del gen en el contig	% cobertura
ERR131800	NODE_14_length_67828	mecA	100.000	44804	46810	100
	NODE_19_length_47352	blaZ	100.000	2658	3545	100
	NODE_20_length_42183	aph(3')-III	100.000	1517	2311	100
	NODE_35_length_9671	msr(A)	98.978	473	1939	100
	NODE_35_length_9671	mph(C)	100.000	2038	2937	100
	NODE_35_length_9671	blaZ	98.778	8707	9606	100
	NODE_44_length_2057	dfrG	100.000	1286	1783	100
	NODE_45_length_1918	aac(6')-aph(2'')	100.000	400	1839	100
ERR131801	NODE_14_length_67828	mecA	100.000	44804	46810	100
	NODE_19_length_47352	blaZ	100.000	2658	3545	100
	NODE_20_length_42183	aph(3')-III	100.000	1517	2311	100
	NODE_35_length_9671	msr(A)	98.978	473	1939	100
	NODE_35_length_9671	mph(C)	100.000	2038	2937	100
	NODE_35_length_9671	blaZ	98.778	8707	9606	100
	NODE_44_length_2057	dfrG	100.000	1286	1783	100
	NODE_45_length_1918	aac(6')-aph(2'')	100.000	400	1839	100
ERR131802	NODE_20_length_39589	blaZ	100.000	16136	17023	100
	NODE_40_length_5087	mecA	100.000	1661	3667	100
	NODE_43_length_2691	aac(6')-aph(2'')	99.931	596	2035	100
	NODE_44_length_2528	erm(C)	100.000	1353	2087	100
ERR131804	NODE_17_length_58037	blaZ	100.000	211304	213310	100
	NODE_18_length_52415	msr(A)	98.978	43808	44695	100
	NODE_18_length_52415	mph(C)	100.000	43217	44683	100
	NODE_18_length_52415	blaZ	98.778	44782	45681	100
	NODE_18_length_52415	aph(3')-III	100.000	51451	52350	100
	NODE_2_length_233969	mecA	100.000	39873	40667	100
	NODE_42_length_3276	dfrG	100.000	2505	3002	100
	NODE_45_length_2026	aac(6')-aph(2''	100.000	134	1573	100
ERR131805	NODE_22_length_39608	blaZ	100.000	22567	23454	100
	NODE_47_length_5070	mecA	100.000	1421	3427	100
	NODE_49_length_3074	aac(6')-aph(2'')	100.000	596	2035	100
	NODE_51_length_2528	erm(C)	100.000	397	1131	100
ERR131806	NODE_101_length_4999	mecA	100.000	151	1038	100
	NODE_104_length_4599	aadD	99.870	1421	3427	100
	NODE_104_length_4599	bleO	100.000	3708	4478	100
	NODE_118_length_2528	erm(C)	100.000	3087	3485	100
	NODE_124_length_1918	aac(6')-aph(2'')	100.000	397	1131	100
	NODE_61_length_17170	blaZ	99.887	400	1839	100
ERR131807	NODE_106_length_2528	erm(C)	100.000	151	1038	100
	NODE_113_length_1918	aac(6')-aph(2'')	100.000	1421	3427	100
	NODE_58_length_17173	blaZ	99.887	3708	4478	100
	NODE_90_length_5004	mecA	100.000	3087	3485	100
	NODE_94_length_4664	aadD	99.870	1352	2086	100
	NODE_94_length_4664	bleO	100.000	400	1839	100

Tabla 4. AMR obtenidos en las muestras ensambladas.

Muestra	Nombre del contig	Gen	% identidad	% Cobertura	PlasmidFinder	Platon
ERR131800	NODE_14	mecA	100	100	NA	NA
	NODE_19	blaZ	100	100	NA	NA
	NODE_20	aph(3')-III	100	100	NA	NA
	NODE_35	msr(A)	98.978	100	NA	NA
	NODE_35	mph(C)	100	100	NA	NA
	NODE_35	blaZ	98.778	100	NA	NA
	NODE_44	dfrG	100	100	NA	NA
	NODE_45	aac(6')-aph(2'')	100	100	NA	NA
ERR131801	NODE_14	mecA	100	100	NA	NA
	NODE_19	blaZ	100	100	NA	NA
	NODE_20	aph(3')-III	100	100	NA	NA
	NODE_35	msr(A)	98.978	100	NA	NA
	NODE_35	mph(C)	100	100	NA	NA
	NODE_35	blaZ	98.778	100	NA	NA
	NODE_44	dfrG	100	100	NA	NA
	NODE_45	aac(6')-aph(2'')	100	100	NA	NA
ERR131802	NODE_20	blaZ	100	100	NA	NA
	NODE_40	mecA	100	100	NA	NA
	NODE_43	aac(6')-aph(2'')	99.931	100	NA	NA
	NODE_44	erm(C)	100	100	rep10_3_pNE131p1	Plasmido
ERR131804	NODE_17	blaZ	100	100	NA	NA
	NODE_18	msr(A)	98.978	100	NA	NA
	NODE_18	mph(C)	100	100	NA	NA

	NODE_18	blaZ	98.778	100	NA	NA
	NODE_18	aph(3')-III	100	100	NA	NA
	NODE_2	mecA	100	100	NA	NA
	NODE_42	dfrG	100	100	NA	NA
	NODE_45	aac(6')-aph(2'')	100	100	NA	NA
ERR131805	NODE_22	blaZ	100	100	NA	NA
	NODE_47	mecA	100	100	NA	NA
	NODE_49	aac(6')-aph(2'')	100	100	NA	NA
	NODE_51	erm(C)	100	100	rep10_3_pNE131p1	Plasmido
ERR131806	NODE_101	mecA	100	100	NA	NA
	NODE_104	aadD	99.870	100	rep22_1b_repB	NA
	NODE_104	bleO	100	100	rep10_3_pNE131p1	Plasmido
	NODE_118	erm(C)	100	100	NA	NA
	NODE_124	aac(6')-aph(2'')	100	100	NA	NA
	NODE_61	blaZ	99.887	100	NA	NA
ERR131807	NODE_106	erm(C)	100	100	rep10_3_pNE131p1	Plasmido
	NODE_113	aac(6')-aph(2'')	100	100	NA	NA
	NODE_58	blaZ	99.887	100	NA	NA
	NODE_90	mecA	100	100	NA	NA
	NODE_94	aadD	99.870	100	rep22_1b_repB	NA
	NODE_94	bleO	100	100	NA	NA

Tabla 5. Plásmidos encontrados que coinciden con un AMR.

Conclusiones

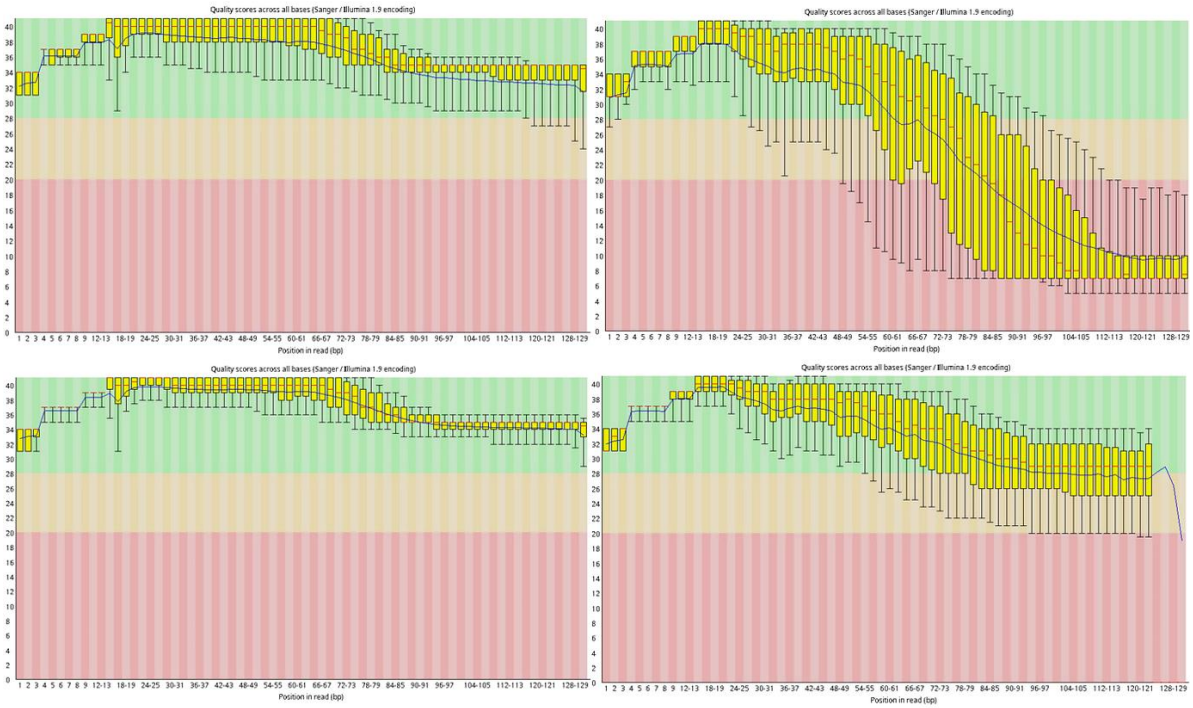
- A pesar de ser MRSA, al realizar un WGS, se pudo observar otros múltiples genes AMR para muchos antimicrobianos.
- Las muestras contenían una gran multitud de resistencias a antimicrobianos donde los genes *mecA* y *blaZ* se encuentran dentro del genoma principal de la bacteria y solo unos pocos genes de resistencia a eritromicinas y estreptomicinas están en plásmidos
- El gen *mecA* está rodeado en el genoma por transposasas que permiten el movimiento de este gen dentro del propio genoma o hacia otros MGE, así como el gen *blaZ* donde, en algunos casos, esa transposasa es para el Tn554.
- Gracias a este tipo de estudios de AMR y al empleo de herramientas bioinformáticas, se podría predecir con alta fiabilidad los genes de AMR presentes en las bacterias que están provocando la enfermedad y de esta manera elegir el tratamiento/terapia antibiótica más apropiado.

Limitaciones y perspectivas futuras

Una de las limitaciones técnicas enfrentadas fueron la limitación de RAM y memoria que ralentizaban el proceso o que impedía utilizar herramientas más precisas y potentes; no obstante, el uso de máquinas con un mayor poder computacional resolvería estas limitaciones y permitiría la paralelización de estos procesos con el fin de analizar más genomas en menos tiempo, de tal forma que se podría realizar un análisis más completo y de forma mucho más eficiente. El análisis de los MGE de este TFM se centró en herramientas para la búsqueda de plásmidos. Los análisis realizados aquí deben complementarse con otros análisis, así como otras herramientas diseñadas para el estudio de otros MGE (como pudiera ser el caso de integrones, islas cromosómicas, etc.) con el fin de tener un espectro más amplio del entorno genético de las AMR.

Para futuros estudios, se puede intentar analizar más muestras para comprobar si existen más AMR asociados a plásmidos. Además, cabe también la posibilidad de realizar más análisis respecto a la identificación de plásmidos por medio de otras herramientas que haya disponibles. Asimismo, debería enfocarse a la búsqueda de otros tipos de MGE.

Anexo



Gráfica 4. Resultado de los análisis de FastQC. En la parte superior, muestra la calidad de las secuencias para una de las muestras antes de limpiarlas. A la izquierda, está la gráfica que se corresponde a las lecturas R1, mientras que, en la derecha, están la gráfica para las lecturas de R2. En la parte inferior, se muestra la calidad de las secuencias después de limpiarlas.

Nombre de la muestra	Número de secuencias crudas	Número de secuencias limpias	Número de secuencias filtradas
ERR131800	1313568	1056893	1056893
ERR131801	878405	706679	706679
ERR131802	853836	693456	693456
ERR131804	774828	638379	638379
ERR131805	927654	756633	756633
ERR131806	890448	717491	717491
ERR131807	903066	718032	718032

Tabla 6. Número de secuencias que hay en cada etapa del análisis para cada muestra. El número de secuencias es el mismo tanto para las R1 como para las R2.

Nombre de la muestra	Número de contigs	Longitud del draft genoma	Longitud media del contig	Valor N50	Profundidad de secuenciación
ERR131800	70	2766006	39514.4	118268	107.634
ERR131801	73	2820778	38640.8	107894	71.366
ERR131802	71	2824264	39778.4	106622	69.86
ERR131804	72	2826480	39256.7	98863	64.228
ERR131805	81	2819932	34814	106680	76.628
ERR131806	206	2812806	13654.4	33491	73.254
ERR131807	196	2809333	14333.3	38221	73.336

Tabla 7. Resultado que obtenemos de ejecutar quast para analizar los contigs obtenido de las muestras.

Scripts empleados

Durante el desarrollo de este TFM, se diseñaron y optimizaron diversos scripts propios, generalmente en el lenguaje de BASH, con el fin de realizar y automatizar cada uno de los procesos de este estudio. Los más relevantes se describen a continuación.

Comando para crear un archivo de texto con los nombres de las muestras

```
1. ls *_1.fastq.gz | sed 's/_1.fastq.gz//' > seqs.txt
```

Con el comando "`ls`", se obtiene una lista de todos los archivos que acaben en "_1.fastq.gz" que luego, por medio del comando "`sed`", se elimina esa parte de tal forma que solo se queden con el nombre de la muestra que se van redireccionando a un archivo de texto. Este formato de comando se utilizará en varias ocasiones para generar un archivo de texto con los nombres de las muestras para ejecutar scripts posteriores. Por tanto, este comando permite que se ejecuten todas las herramientas posteriores para diferente número de muestras, de forma que permite que los siguientes comandos y scripts se ejecuten sin problema en otro estudio con mayor cantidad de muestras.

FastQC.sh

```
#!/bin/bash

#Script para hacer el fastqc en rawseq

for file in ./00_raw_data/*.fastq.gz;
        do fastqc ${file};
        done
```

Aquí se ejecuta la herramienta FastQC en cada una de las muestras que se encuentra en la carpeta de las lecturas crudas.

nseq.sh

```
#!/bin/bash

#Cuenta el número de secuencias de cada muestra

for file in $(ls *gz);
        do
                echo -e "${file}: \c" >> nseq.txt; zgrep -c "^@ERR" ${file} >> nseq.txt;
        done
```

Para obtener el número de cada secuencia, se busca que el bucle vaya por todas las muestras, de forma que mantenga el nombre del archivo y se eligen aquellas que empiezan por ERR. De esta forma, se obtiene el nombre de la muestra y las secuencias que hay en cada una de ellas.

Recuento de las lecturas por muestra

```
awk '{total += $2; count++} END {print total/count}' seqs.txt
```

En este comando, se recorre línea por línea del documento donde se elige el segundo campo que se corresponde al número de secuencias en cada muestra y se van contando, de tal forma que al final se divide la suma total de los números entre la cuenta de secuencias totales que hay a modo de media.

FastP.sh

```
#!/bin/bash

```

```
#Limpieza de los archivos fastq

mkdir 01_clean_reads #Creamos una nueva carpeta para las muestras limpias

for file in $(<seqs.txt); #saca los nombres del txt
        do
                fastp -i 00_raw_data/${file}_1.fastq.gz -I 00_raw_data/${file}_2.fastq.gz \
            -o 01_clean_reads/${file}_1.clean.fastq.gz -O 01_clean_reads/${file}_2.clean.fastq.gz \
            --cut_front --cut_right -l 50 -q 25 --detect_adapter_for_pe --adapter_fasta adapters.fasta
            done
```

Se crea una nueva carpeta para mover los archivos nuevos generados. En el bucle, cada archivo sufre un proceso de limpieza donde se recortan en ambos extremos de la secuencia unos 50 pares de bases y que tengan una calidad de 25, así, también se busca que detecte los adaptadores para nuestras lecturas pareadas según un archivo .fasta donde se encuentran todos los adaptadores posibles.

Bowtie2.sh

```
#!/bin/bash

#Alinea las secuencias con el fago y nos quedamos con las muestras filtradas

mkdir 02_filtered_reads #Crea un nuevo directorio para las muestras filtradas

bowtie2-build phiX174_wg.fasta phiX_index #Convierte el archivo de referencia en un índice

for file in $(<seqs.txt);
        do
            bowtie2 -p 2 -x phiX_index -1 01_clean_reads/${file}_1.clean.fastq.gz \
            -2 01_clean_reads/${file}_2.clean.fastq.gz \
            --un-conc-gz 02_filtered_reads/Sample_host_removed > 02_filtered_reads/Sample_mapped_and_unmapped.sam
        done
```

Al igual que en el script anterior, se crea una carpeta donde estarán las muestras alineadas. Luego, se convierte el archivo que contiene el genoma del fago en un índice que pueda utilizar bowtie2. En el bucle, al igual que en casos anteriores, se ejecuta el comando por todas las muestras donde se usa una representación de las cadenas de

ADN compactadas basándose en el índice del fago creado de tal forma que los resultados se escriben según las muestras originales.

SPAdes.sh

```
#!/bin/bash

#Hacemos un ensamblaje de novo de las lecturas

for file in $(<seqs.txt);
    do
        spades.py -
1 02_filtered_reads/${file}.filtered_R1.fastq.gz \
        -2 02_filtered_reads/${file}.filtered_R2.fastq.gz
\
        --careful -k auto -t 2 -o ${file}_spades_out
         done
```

Una vez se obtienen los archivos en un formato apropiado, se ejecuta el ensamblaje con SPAdes de forma que intenta reducir el número de mismatches y de indels cortos, calcula el número de k-meros de forma automática y usa todos los hilos que dispone el Amazon Workspaces que, en este caso, es de 2 hilos.

Listado de los directorios generados por SPAdes

```
for dir in ERR*/; do echo "$dir" >> spades_dir.tx
t; done;
```

Para ejecutar los siguientes comandos, es necesario crear un archivo de texto que contenga los nombres de los archivos, al igual que al principio del análisis. Con este comando, se guarda el nombre de cada una de las muestras generadas.

PRINSEQ-lite.sh

```
#!/bin/bash

#Eliminamos los contigs menores a 200 pares de bases

mkdir 03_genomes #Creamos una carpeta nueva para guardar
los contigs

for dir in $(<spades_dir.txt);
        do
                prinseq-lite.pl -
fasta ${dir}/contigs.fasta -min_len 200 \
```

```
10.                         -out_good
   03_genomes/$(basename "$dir") -out_bad null
11.             done
```

Se crea una nueva carpeta para colocar los nuevos archivos generados y luego, para cada muestra generada de SPAdes, se eliminan aquellos contigs que tengan un tamaño menor a 200 pares de bases y se guardan aquellos con el nombre del archivo en la nueva carpeta y el resto se descartan.

Quast

```
1. quast.py -o quast_result -m 0 -k --k-mer-
   size 55 03_genomes/*
```

Con quast, se puede analizar las características de los contigs de forma que se genera una carpeta con los resultados de este análisis, el número mínimo de contigs que analiza es de cero para tenerlos todos en cuenta y el tamaño de los k-meros, en este caso, es de 55.

Sizecontigs.sh

```
1. #!/bin/bash
2.
3. #Cuenta el tamaño de cada contig de cada muestra
4.
5. for file in $(<name_fasta.txt);
6.         do
7.                 grep "^>" ${file} | cut -d '_' -
   f 4 > ${file}_contig_number.txt
8.         done
```

Para saber el tamaño de contig medio, es necesario primero saber el tamaño de los contigs por cada muestra de forma que se selecciona el encabezado que tiene el tamaño de cada contig y se elimina el resto del encabezado quedándonos solo con ese valor.

Meancontigs.sh

```
1. #!/bin/bash
2.
3. #Media del tamaño de los contigs para cada muestra
4.
5. for file in $(<sizecontigsname.txt);
6.         do
```

```
7.                 awk '{total += $1;
   count++} END {print total/count}'
   ${file} >> meancontigs.txt
8.         done
```

Se hace una media al igual que se hizo anteriormente con la diferencia de que solo tenemos el número de cada muestra, en lugar de un campo con el nombre y otro con el valor.

blastn1.sh

```
1. #!/bin/bash
2.
3. #Hacemos un blast con nuestra base de datos de resistencia
   a antibióticos
4.
5. for file in ./03_genomes/*.fasta;
6.         do
7.                 blastn -query ${file} -
   db ./resfinder_db/AntibioticResistanceDB \
8.                 -perc_identity 0.8 -culling_limit 1 \
9.                 -outfmt "6 qseqid sseqid pident length
   mismatch gapopen qstart qend sstart send evalue bitscore
   slen" -out ${file}.txt
10.
11.             #Añadimos un encabezado a los archivos
12.                     sed -
   i "1iqseqid\tsseqid\tpident\tlength\tmismatch\tgapopen\tqs
   tart\tqend\tsstart\tsend\tevalue\tbitscore\tslen" ${file}.
   txt
13.
14.             done
```

Se hace una búsqueda de nuestras muestras en la base de datos de resistencia a antibióticos que se ha creado con anterioridad y que el resultado se muestre en una tabla que indique el nombre de nuestra muestra y del mejor gen que encuentra, la longitud de su alineamiento, la posición en la que alinean, el número de gaps que hay y la longitud total del gen. Como por defecto no se añade ningún encabezado, se añade manualmente para tener una visión más clara de los resultados.

coverage.sh

```
1. #!/bin/bash
2.
3. #Calculamos la cobertura para los resultados del blastn de
   resistencia a antibióticos
4.
```

```
for file in $(<blast_name.txt);
        do
           #Calculamos la cobertura
           awk
   'BEGIN {OFS="\t"}{$1=$1}NR==1{print $0, "coverage";next}{p
   rint $0, (($4-$6)*100)/$13}' ${file} > cover.${file}
           #Nos quedamos con los que tengan un valor mayor
   a 80
           awk -v val=80 '!($14 < val)'
   cover.${file} > final.${file}
             done
```

Se añade la cobertura del alineamiento con la base de datos como una nueva columna y se guarda aquellos que tengan un porcentaje superior al 80% en un archivo diferente.

Blastn2.sh

```
#!/bin/bash

#Hacemos un blast con la base de datos de plásmidos

for file in ./03_genomes/*.fasta;
        do
                blastn -query ${file} -
   db ./plasmidfinder_db/PlasmidsDB -perc_identity 0.95 -
   culling_limit 1 \
                -outfmt "6 qseqid sseqid pident length
   mismatch gapopen qstart qend sstart send evalue bitscore
   slen" \
                -out ${file}.plasmid.txt

            #Añadimos un encabezado a los archivos
                    sed -
   i "1iqseqid\tsseqid\tpident\tlength\tmismatch\tgapopen\tqs
   tart\tqend\tsstart\tsend\tevalue\tbitscore\tslen" ${file}.
   plasmid.txt
            done
```

En este script, se hace lo mismo que en el script blastn1.sh, con la diferencia de que se utiliza la base de datos de plásmidos.

Plasmidcover.sh

```
#!/bin/bash

#Calculamos la cobertura para los resultados del blastn de
   plásmidos

```

```
for file in $(<blast2_name.txt);
        do
                #Calculamos la cobertura
                awk 'BEGIN {OFS="\t"}{$1=$1}NR==1{print $0, "coverage";next}{print $0, (($4-$6)*100)/$13}' ${file} > plasmid.cover.${file}
                #Nos quedamos con los que tengan un valor mayor a 95
                        awk -v val=95 '!($14 < val)' plasmid.cover.${file} > Plasmid.${file}
            done
```

Se calcula el porcentaje de cobertura para los resultados del blastn de la base de datos de plásmidos y se añade los resultados a una nueva columna y se filtran los datos, de tal forma que permanecerán aquellos que tengan un porcentaje mayor al 95% y se guardan en un nuevo archivo.

AMRtable.sh

```
#!/bin/bash

#Creamos una tabla para los datos obtenidos de AMR

for sample in $(<seqs.txt);
   do
        #Nos quedamos solo con el contig
        for contig in $(tail -n+2 04_AMRblastn/AMR.${sample}.fasta.txt | cut -f 1 | sort -u);
            do
                        echo ${sample} >> uno.tmp #Nos quedamos con el nombre de la muestra
                        grep -w "^${contig}" 04_AMRblastn/AMR.${sample}.fasta.txt >> dos.tmp
                done
            #Unimos todos los archivos temporales
            paste uno.tmp dos.tmp | sed "1iMuestra\t$(head -n 1 04_AMRblastn/AMR.${sample}.fasta.txt)" > ${sample}_AMR_table.txt
            rm *tmp #Eliminamos los archivos temporales
        done

    #Unimos todas las tablas en una

    for sample in $(<seqs.txt);
       do
```

```
            tail -n+2 ${sample}_AMR_table.txt >> all_AMR_table.txt
        done

    #Añadimos el encabezado final
    sed -i "1i$(head -n 1 ERR131800_AMR_table.txt)" all_AMR_table.txt
```

Por medio de este script, se crea varias tablas con los resultados de la búsqueda de AMR que luego se combinan en una única tabla para facilitar su posterior análisis.

Summarizer.sh

```
#!/bin/bash

#Creamos un único archivo que tenga los resultados del AMR, plasmidfinder y platon

for sample in $(<seqs.txt); #Seleccionamos los resultados de AMR
   do
        for contig in $(tail -n+2 04_AMRblastn/AMR.${sample}.fasta.txt | cut -f 1 | sort -u); #Nos quedamos solo con el contig
            do
                echo ${sample} >> uno.tmp #Nos quedamos con el nombre de la muestra
                grep -w "^${contig}" 04_AMRblastn/AMR.${sample}.fasta.txt >> dos.tmp #Nos quedamos con el nombre del contig
                #Hacemos un if para ver si el contig está en los resultados de plasmidfinder
                if grep -q -w "^${contig}" 05_plasmidfinder/Plasmid.${sample}.fasta.plasmid.txt;
                    then
                        #Si está presente, nos quedamos con el plásmido asociado
                        grep -w "^${contig}" 05_plasmidfinder/Plasmid.${sample}.fasta.plasmid.txt | head -n 1 | cut -f 2 >> tres.tmp
                    else
                        echo "NA" >> tres.tmp #Si no está, que ponga NA
                fi
                #Hacemos un if para ver si el contig está en los resultados de platon
                if grep -q -w "^${contig}" 06_platon/platon.${sample}.fasta.tsv;
                    then
```

```
                                #Si está presente, nos
  quedamos con las características
                                echo "Plasmido" >> cuatro.tmp
                            else
                                echo "NA" >> cuatro.tmp #Si
  no está, que ponga NA
                        fi
                    done
                #Unimos todos los archivos temporales en uno
  para cada muestra
                paste uno.tmp dos.tmp tres.tmp
  cuatro.tmp | sed "1iMuestra\t$(head -n 1
  04_AMRblastn/AMR.${sample}.fasta.txt)\tPlasmidFinder\tPlat
  on" > ${sample}_summary_table.txt
                rm *tmp #Eliminamos los archivos temporales
            done
```

Aquí lo que se busca es generar un archivo para cada muestra que combine la información que se ha obtenido para cada una de ellas tras las diferentes búsquedas en las bases de datos. Para ello, se crean diferentes archivos temporales que luego serán eliminados y que conformarán las diferentes partes que, una vez agrupados, darán lugar a la tabla final de datos para cada muestra.

Tablegenerator.sh

```
#!/bin/bash

#Unimos todas las tablas generadas en una sola

for sample in $(<seqs.txt);
do
tail -
  n+2 ${sample}_summary_table.txt >> all_samples_summary_tab
  le.txt #Unimos todas las tablas
done
#Añadimos el encabezado
    sed -i "1i$(head -n 1
  ERR131800_summary_table.txt)" all_samples_summary_table.tx
  t
```

En este script, lo que se hace es unificar todos los archivos de las muestras generados anteriormente y colocarle el mismo encabezado para todos ellos.

Prokka.sh

```
#!/bin/bash

```

```
#Ejecutamos prokka para cada archivo resultado de los contigs

for file in $(<name_fasta.txt);
   do
        prokka --outdir ../prokka_${file} --addgenes --genus Staphylococcus --species aureus \
        --kingdom Bacteria --usegenus --mincontiglen 200 ${file}
   done
```

A partir de este script, se ejecuta la herramienta de prokka que genera una carpeta para cada muestra que contienen los datos de los CDS a partir de los contigs ensamblados por SPAdes.

Prokka_result.sh

```
#!/bin/bash

#Movemos los archivos que nos interesan de ejecutar prokka y les cambiamos el nombre según la muestra

mkdir 07_prokka_result #Creamos la carpeta para guardar los resultados

for dir in $(<prokka_dir.txt);
   do
        cp ${dir}/PROKKA_*.gff 07_prokka_result/$(basename "$dir")
        done
```

Por medio de este script, se ordenan los resultados deseados de prokka y se cambian los nombres de los archivos para ordenarlos por la muestra a la que pertenecen.

Kraken.sh

```
#!/bin/bash

#Utilizamos kraken2 para confirmar la taxonomía de las muestras

mkdir 08_kraken2 #Creamos un directorio nuevo para guardar los resultados

for sample in $(<seqs.txt);
   do
```

```
        kraken2 03_genomes/${sample}.fasta --memory-mapping --use-mpa-style --threads 2 \
            --db kraken_db/minikraken_8GB_20200312 --report 08_kraken2/${sample}.report \
            --output 08_kraken2/${sample}.txt
    done
```

Por medio de este script, se puede comprobar el origen de las muestras que se están analizando para cada una de ellas y generar sus correspondientes archivos que indican los genomas que están involucrados.

Referencias bibliográficas

1. Fitzgerald JR, Sturdevant DE, Mackie SM, Gill SR, Musser JM. Evolutionary genomics of Staphylococcus aureus: insights into the origin of methicillin-resistant strains and the toxic shock syndrome epidemic. Proc Natl Acad Sci U S A [Internet]. 2001;98(15):8821-6. Disponible en: http://dx.doi.org/10.1073/pnas.161098098
2. Ito T, Okuma K, Ma XX, Yuzawa H, Hiramatsu K. Insights on antibiotic resistance of Staphylococcus aureus from its whole genome: genomic island SCC. Drug Resist Updat [Internet]. 2003;6(1):41-52. Disponible en: http://dx.doi.org/10.1016/s1368-7646(03)00003-7
3. Malachowa N, DeLeo FR. Mobile genetic elements of Staphylococcus aureus. Cell Mol Life Sci [Internet]. 2010;67(18):3057-71. Disponible en: http://dx.doi.org/10.1007/s00018-010-0389-4
4. Firth N, Jensen SO, Kwong SM, Skurray RA, Ramsay JP. Staphylococcal plasmids, transposable and integrative elements. Microbiol Spectr [Internet]. 2018;6(6). Disponible en: http://dx.doi.org/10.1128/microbiolspec.GPP3-0030-2018
5. Palavecino E. Clinical, epidemiological, and laboratory aspects of methicillin-resistant Staphylococcus aureus (MRSA) infections. Methods Mol Biol [Internet]. 2007;391:1-19. Disponible en: http://dx.doi.org/10.1007/978-1-59745-468-1_1
6. Kanemitsu K, Yamamoto H, Takemura H, Shimada J, Kaku M. Characterization of MRSA transmission in an emergency medical center by sequence analysis of the 3'-end region of the coagulase gene. J Infect Chemother [Internet]. 2001;7(1):22-7. Disponible en: http://dx.doi.org/10.1007/s101560170030

7. Holden MTG, Feil EJ, Lindsay JA, Peacock SJ, Day NPJ, Enright MC, et al. Complete genomes of two clinical Staphylococcus aureus strains: evidence for the rapid evolution of virulence and drug resistance. Proc Natl Acad Sci U S A [Internet]. 2004;101(26):9786-91. Disponible en: http://dx.doi.org/10.1073/pnas.0402521101
8. Enright MC. The evolution of a resistant pathogen--the case of MRSA. Curr Opin Pharmacol [Internet]. 2003;3(5):474-9. Disponible en: http://dx.doi.org/10.1016/s1471-4892(03)00109-7
9. Partridge SR, Kwong SM, Firth N, Jensen SO. Mobile genetic elements associated with antimicrobial resistance. Clin Microbiol Rev [Internet]. 2018;31(4). Disponible en: http://dx.doi.org/10.1128/cmr.00088-17
10. Harris SR, Cartwright EJP, Török ME, Holden MTG, Brown NM, Ogilvy-Stuart AL, et al. Whole-genome sequencing for analysis of an outbreak of meticillin-resistant Staphylococcus aureus: a descriptive study. Lancet Infect Dis [Internet]. 2013;13(2):130–6. Disponible en: http://dx.doi.org/10.1016/S1473-3099(12)70268-2
11. Bottery MJ. Ecological dynamics of plasmid transfer and persistence in microbial communities. Curr Opin Microbiol [Internet]. 2022;68(102152):102152. Disponible en: http://dx.doi.org/10.1016/j.mib.2022.102152
12. Tao Z, Shi A, Li R, Wang Y, Wang X, Zhao J. Microarray bioinformatics in cancer- a review. J BUON. 2017;22(4):838-43. PMID: 29155508
13. Dlamini Z, Francies FZ, Hull R, Marima R. Artificial intelligence (AI) and big data in cancer and precision oncology. Comput Struct Biotechnol J [Internet]. 2020;18:2300-11. Disponible en: http://dx.doi.org/10.1016/j.csbj.2020.08.019
14. Nair SV, Madhulaxmi, Thomas G, Ankathil R. Next-generation sequencing in cancer. J Maxillofac Oral Surg [Internet]. 2021;20(3):340-4. Disponible en: http://dx.doi.org/10.1007/s12663-020-01462-4
15. Oliver GR, Hart SN, Klee EW. Bioinformatics for clinical next generation sequencing. Clin Chem [Internet]. 2015;61(1):124-35. Disponible en: http://dx.doi.org/10.1373/clinchem.2014.224360
16. Zankari E, Hasman H, Cosentino S, Vestergaard M, Rasmussen S, Lund O, et al. Identification of acquired antimicrobial resistance genes. J Antimicrob Chemother [Internet]. 2012;67(11):2640–4. Disponible en: http://dx.doi.org/10.1093/jac/dks261

17. Andrews S. FastQC: A Quality control Tool for High Throughput Sequence Data [Internet]. (2010). Disponible en: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
18. Ewels P, Magnusson M, Lundin S, Käller M. MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics [Internet]. 2016;32(19):3047-8. Disponible en: http://dx.doi.org/10.1093/bioinformatics/btw354
19. Chen S, Zhou Y, Chen Y, Gu J. fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics [Internet]. 2018;34(17):i884-90. Disponible en: http://dx.doi.org/10.1093/bioinformatics/bty560
20. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods [Internet]. 2012;9(4):357-9. Disponible en: http://dx.doi.org/10.1038/nmeth.1923
21. Prjibelski A, Antipov D, Meleshko D, Lapidus A, Korobeynikov A. Using SPAdes DE Novo Assembler. Curr Protoc Bioinformatics [Internet]. 2020;70(1):e102. Disponible en: http://dx.doi.org/10.1002/cpbi.102
22. Schmieder R, Edwards R. Quality control and preprocessing of metagenomic datasets. Bioinformatics [Internet]. 2011 [citado 6 de junio de 2022];27(6):863-4. Disponible en: https://pubmed.ncbi.nlm.nih.gov/21278185/
23. Gurevich A, Saveliev V, Vyahhi N, Tesler G. QUAST: quality assessment tool for genome assemblies. Bioinformatics [Internet]. 2013;29(8):1072-5. Disponible en: http://dx.doi.org/10.1093/bioinformatics/btt086
24. Wood DE, Lu J, Langmead B. Improved metagenomic analysis with Kraken 2.Genome Biol[Internet]. 2019;20(1):257. Disponible en: http://dx.doi.org/10.1186/s13059-019-1891-0
25. Seemann T. Prokka: rapid prokaryotic genome annotation. Bioinformatics[Internet]. 2014 [citado 6 de junio de 2022];30(14):2068-9. Disponible en: https://pubmed.ncbi.nlm.nih.gov/24642063/
26. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+: architecture and applications. BMC Bioinformatics [Internet]. 2009;10(1):421. Disponible en: http://dx.doi.org/10.1186/1471-2105-10-421
27. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res [Internet]. 1997;25(17):3389-402. Disponible en: http://dx.doi.org/10.1093/nar/25.17.3389

28. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol [Internet]. 1990;215(3):403-10. Disponible en: http://dx.doi.org/10.1016/S0022-2836(05)80360-2
29. Carattoli A, Zankari E, García-Fernández A, Voldby Larsen M, Lund O, Villa L, et al. In silico detection and typing of plasmids using PlasmidFinder and plasmid multilocus sequence typing. Antimicrob Agents Chemother [Internet]. 2014;58(7):3895-903. Disponible en: http://dx.doi.org/10.1128/AAC.02412-14
30. Schwengers, O. Platon database. Microbial Genomics [Internet]. 2020;95:295. Disponible en: https://doi.org/10.5281/zenodo.4066768
31. Schwengers O, Barth P, Falgenhauer L, Hain T, Chakraborty T, Goesmann A. Platon: identification and characterization of bacterial plasmid contigs in short-read draft assemblies exploiting protein sequence-based replicon distribution scores. Microb Genom [Internet]. 2020;6(10). Disponible en: http://dx.doi.org/10.1099/mgen.0.000398
32. Carattoli A, Zankari E, García-Fernández A, Voldby Larsen M, Lund O, Villa L, et al. In silico detection and typing of plasmids using PlasmidFinder and plasmid multilocus sequence typing. Antimicrob Agents Chemother [Internet]. 2014;58(7):3895-903. Disponible en: http://dx.doi.org/10.1128/AAC.02412-14
33. Garcillán-Barcia MP, Redondo-Salvo S, Vielva L, de la Cruz F. MOBscan: Automated annotation of MOB relaxases. Methods Mol Biol [Internet]. 2020;2075:295-308. Disponible en: http://dx.doi.org/10.1007/978-1-4939-9877-7_21
34. Robertson J, Nash JHE. MOB-suite: software tools for clustering, reconstruction and typing of plasmids from draft assemblies. Microb Genom [Internet]. 2018;4(8). Disponible en: http://dx.doi.org/10.1099/mgen.0.000206
35. Bolger T, Usadel B. Trimmomatic: adapters [Internet]. 2020. Disponible en: https://github.com/usadellab/Trimmomatic/tree/main/adapters