Universidad San Jorge

Facultad de Ciencias de la Salud

Doble Grado en Bioinformática y Farmacia

Proyecto Final de Bioinformática

Comparación de la herramienta MTBseq frente a Snippy y TBProfiler para el análisis de variantes de muestras de *Mycobacterium tuberculosis* en un flujo de trabajo automatizado

Autora del proyecto: Candela Gerediaga Ruiz de Velasco

Director del proyecto: Alberto Cebollada Solanas

Villanueva de Gállego (Zaragoza), 15 de junio de 2025

universidad SANJORGE

GRUPO SANVALERO

Este trabajo constituye parte de mi candidatura para la obtención del título de Graduado o Graduada en Bioinformática por la Universidad San Jorge y no ha sido entregado previamente (o simultáneamente) para la obtención de cualquier otro título. Este documento es el resultado de mi propio trabajo, excepto donde de otra manera esté indicado y referido. Doy mi consentimiento para que se archive este trabajo en la biblioteca universitaria de Universidad San Jorge, donde se puede facilitar su consulta.

Zaragoza, a 15 de junio de 2025.

Fdo: Candela Gerediaga Ruiz de Velasco,
DNI: 79175369P

Dedicatoria y agradecimientos

A Alberto Cebollada, por toda la ayuda y conocimientos que me ha proporcionado. Por su intachable labor como tutor, aconsejándome y dándome su punto de vista para sacar adelante este proyecto. Por confiar en que podía hacerlo.

A los profesores del grado en Bioinformática, que me han ayudado a averiguar lo que es esta profesión y todo lo que puedo aportar al mundo con ella. A mis compañeros, que han hecho que estos seis años se hicieran hasta cortos.

A Ángela, que me ha acompañado desde el día uno y nunca se ha separado. Por hacerme llorar de risa, pero también centrarme cuando lo he necesitado. Por ser mi otra mitad en esta etapa, y saber que siempre estará.

A Andrea, porque, aunque no empezamos de la mejor manera, sé que me llevo una amiga para toda la vida.

A Alex, por haber estado ahí durante todo el camino, ayudarme, aconsejarme y aguantarme. Ya somos los dos bioinformáticos.

A mis aitas y Alba, sin los que esto nunca hubiera sido posible. Gracias por apoyarme en mis peores momentos, por nunca soltar mi mano y sacarme siempre una sonrisa. Por ser la mejor familia que podría desear y acompañarme en el camino.

Y, por último, a Candela. Gracias por arriesgarte a vivir esta aventura hace 6 años. Por no rendirte y seguir siempre adelante, a pesar de que no fuera fácil. Porque gracias a ello eres hoy quién eres. Nunca dejes de perseguir tus sueños y luchar por ellos.

ÍNDICE

RESUMEN

La tuberculosis, causada por el complejo *Mycobacterium tuberculosis* (*M. tuberculosis*), es la principal causa de muertes a nivel mundial provocada por un patógeno infeccioso, con una mortalidad de 1,25 millones de personas en 2023. Detectar variantes genéticas que provocan resistencia a tratamientos es clave para su control.

El objetivo del proyecto es comparar herramientas bioinformáticas para el análisis de variantes (variant calling) en muestras de *M. tuberculosis*, utilizando un pipeline automatizado en Docker. El flujo incluye herramientas para análisis de calidad (FastQC, Fastp, MultiQC) y para análisis genómico (MTBseq, Snippy y TBProfiler).

Los resultados muestran que las tres herramientas detectan SNPs de forma similar, pero MTBseq es más precisa en la detección de inserciones y deleciones. TBProfiler destaca por su capacidad de espoligotipificación y detección de SNPs relacionados con resistencias, y junto con MTBseq, clasifica correctamente los linajes. Las matrices de distancias genéticas generadas por MTBseq y Snippy muestran diferencias entre muestras. En cuanto a rendimiento, MTBseq consume más memoria y tiempo, pero menos CPU; mientras que Snippy y TBProfiler son más rápidos y eficientes, aunque requieren mayor uso de CPU.

Con los resultados obtenidos, se concluye que las tres herramientas son prácticas en el análisis de variantes de *M.tuberculosis*, permitiendo realizar un análisis automatizado completo y útil para la investigación y aplicación clínica.

Palabras clave: *Mycobacterium tuberculosis*, herramientas bioinformáticas, llamado de variantes (variant calling), Docker.

ABSTRACT

Tuberculosis, caused by the *Mycobacterium tuberculosis* complex (*M. tuberculosis*), is the leading cause of death worldwide from an infectious pathogen, with 1.25 million deaths in 2023. Detecting genetic variants responsible for treatment resistance is key to its control

This project compares bioinformatics tools for variant calling in *M. tuberculosis* samples, using an automated pipeline implemented in Docker. The workflow includes tools for quality control (FastQC, Fastp, MultiQC) and for genomic analysis (MTBseq, Snippy, and TBProfiler).

The results show that all three tools detect SNPs similarly, but MTBseq is more accurate in identifying insertions and deletions. TBProfiler stands out for its ability to perform spoligotyping and detect resistance-related SNPs, and along with MTBseq, correctly classifies sample lineages. The genetic distance matrices generated by MTBseq and Snippy reveal differences between samples.

In terms of performance, MTBseq consumes more memory and time but uses less CPU, whereas Snippy and TBProfiler are faster and more efficient in memory usage, although they require higher CPU usage.

Based on the results, the study concludes that all three tools are practical for variant analysis of *M. tuberculosis*, enabling a fully automated analysis useful for both research and clinical applications.

Keywords: *Mycobacterium tuberculosis*, bioinformatic tools, variant calling, Docker.

ÍNDICE DE SIGLAS Y ABREVIATURAS

ADNr	ADN ribosómico
BAGEP	Bacterial Genome Pipeline
BAM	Binary Alignment Map
BCF2	Binary Call Format
BWA	Burrows Wheeler Aligner
CIBA	Centro de Investigación Biomédica de Aragón
CPU	Unidad Central de Procesamiento (Central Processing Unit)
CRAM	Compressed Reference-oriented Alignment Map
DR	Repeticiones Directas (Direct Repeats)
DRV	Repeticiones de Variantes Directas (Deirect Variant Repeats)
DST	Test de Susceptibilidad de Fármacos (Drug Susceptibility Tests)
GATK	Genome Analysis Toolkit
IACS	Instituto Aragonés de Ciencias de la Salud
LR	Lecturas Largas (Long Read)
LSP	Polimorfismo de Secuencias Largas (Large Sequence Polymorphism)
MIRU-VNTR	Mycobacterial Interspersed Repetitive Unit – Variable Number Tandem Repeats
MNP	Polimorfismo de Múltiples Nucleótidos (Multiple Nucleotide Polymorphism)
MTBC	Complejo *Mycobacterium tuberculosis* (*Mycobacterium tuberculosis* Complex)
NGS	Secuenciación de Nueva Generación (Next Generation Sequencing)
RAM	Memoria de Acceso Aleatorio (Random Access Memory)
SAM	Sequence Alignment Map
SNP	Polimorfismo de un Solo Nucleótido (Single Nucleotide Polymorphism)
SR	Lecturas Cortas (Short Reads)
SV	Variante Estructural (Structural Variant)
TB	Tuberculosis
VCF	Formato de Llamado de Variantes (Variant Call Format)
WGS	Secuenciación de Genoma Completo (Whole Genome Sequencing)

1. INTRODUCCIÓN

La tuberculosis (TB) es una enfermedad infecciosa, originada por el complejo *Mycobacterium tuberculosis* (MTBC). Este complejo está compuesto por 7 especies de bacterias: *Mycobacterium tuberculosis, Mycobacterium africanum, Mycobacterium bovis, Mycobacterium pinnipedii, Mycobacterium caprae, Mycobacterium canettit y Mycobacterium microti.* Todas ellas son bacilos Gram positivo, ácido-alcohol resistentes y aerobias estrictas, siendo *M. tuberculosis* la más frecuente en humano (1).

La enfermedad se transmite de persona a persona, principalmente a través de las gotas que una persona enferma (con tuberculosis pulmonar o laríngea) expulsa cuando estornuda, tose o escupe. Las gotas contienen los bacilos tuberculosos, y al ser tan pequeños, son capaces de mantenerse en suspensión en el aire durante varias horas. La infección por vía mucosa ocurre cuando las gotas infectadas entran en contacto con ellas, mientras que si se trata de una inoculación accidental se transmitiría por vía percutánea (1 – 3).

Una vez los bacilos entran en el nuevo organismo huésped, han de pasar al menos 21 días para que se manifiesten síntomas y la infección se vuelva contagiosa. También es posible que no se desarrolle una enfermedad activa y que las bacterias permanezcan dentro del organismo inactivas. Este fenómeno se conoce como tuberculosis latente y está caracterizado por la ausencia de síntomas y la imposibilidad de contagio (2, 3).

La infección más común es la pulmonar, con síntomas como fiebre, sudoración (nocturna principalmente), fatiga, expectoraciones y dolor torácico. Estos primeros síntomas suelen pasar desapercibidos. Es por esto por lo que es tan importante su rápida detección y tratamiento, ya que, si no se trata, la infección puede progresar y propagarse a sistema nervioso central, sistema circulatorio o sistema gastrointestinal y convertirse en tuberculosis extrapulmonar (1 – 3) .

La tuberculosis es la principal causa de muertes a nivel mundial provocada por un patógeno infeccioso, siendo especialmente mortífera para personas con VIH. En 2023, después de tres años en las que el virus del COVID-19 fue la causa de muerte predominante, 1,25 millones de personas murieron a causa de la tuberculosis, de las cuales 161000 estaban infectadas con VIH. Además, la tuberculosis está considerada una de las principales causas de muerte relacionadas con resistencia a antimicrobianos (3).

Está presente en todos los países y grupos de edad, y en 2023 10,8 millones de personas la contrajeron. El mayor porcentaje de contagios ocurrió en Asia Sudoriental, seguida de África y

Pacífico Oriental. A pesar de las altas tasas de mortalidad e infección, es una enfermedad prevenible y curable, siendo clave el diagnóstico y tratamiento (3).

Existen pruebas rápidas de diagnóstico con altos porcentajes de precisión, lo que permite detectar la enfermedad de forma temprana. También es eficaz la prueba de la tuberculina, las pruebas cutáneas con antígenos o el ensayo de liberación de interferón. Una vez diagnosticada, el tratamiento consiste en la administración de antibióticos específicos, como la isoniazida, la rifampicina, la pirazinamida o el etambutol. Es importante completar siempre el tratamiento prescrito. En el caso de que la bacteria sea resistente a los antimicrobianos, se denomina "farmacorresistente o multirresistente", y requiere otro tipo de tratamiento. En estos escenarios, los dos tratamientos de primera línea (isoniazida y rifampicina) no son efectivos, teniendo que utilizar fármacos alternativos que no son tan beneficiosos. La tuberculosis multirresistente constituye una crisis de salud pública, así como una gran amenaza para la seguridad sanitaria (1, 3).

Por lo tanto, es importante detectar las variaciones en el genoma de la bacteria (procedimiento conocido como *variant calling*) que producen la resistencia al tratamiento y que generan las cepas farmacorresistentes. Y para ello, es necesario tener un conocimiento previo del genoma de *Mycobacterium tuberculosis*.

Por norma general, se utiliza como referencia el genoma de la cepa H37Rv, aislada originalmente en 1905 (4). Aunque la cepa original ya no existe, las cepas H37Rv TMC102 (ATCC 27294) y NR-123 (ATCC 25618), aisladas del mismo paciente, se siguen utilizando como modelo. Sin embargo, todavía hoy hay ciertas incertidumbres sobre la secuencia de referencia actual. La secuenciación del genoma completo de la bacteria fue realizada en 1998, identificando un único cromosoma circular con alrededor de 3974 genes y 4411529 pares de bases. El contenido de Guanina y Citosina (G/C) es del 65,6%, siendo la segunda secuencia bacteriana más grande disponible, después de la de *Escherichia coli* (4 – 6).

Estudios de secuenciación y epidemiología molecular han permitido establecer que *M. tuberculosis* posee una estructura poblacional de tipo clonal, con poca variación genética entre cepas y un intercambio horizontal mínimo. Esto significa que las distintas cepas de la bacteria se generan por la expansión clonal del ancestro común. En estos casos, la epidemiología molecular es muy útil, ya que comprende una serie de técnicas que permiten comparar secuencias de ácidos nucleicos de 2 o más aislamientos. Los aislamientos están relacionados entre sí al descender de un mismo ancestro común, y se organizan en grupos formando clones.

Aunque las diferencias entre ellos son mínimas, se han detectado cambios debidos a polimorfismos de un solo nucleótido (Single Nucleotide Polymorphism (SNP)), polimorfismos de secuencias largas (Large Sequence Polymorphism (LSP)) conformados principalmente por deleciones o inserciones, y variaciones en regiones repetidas (7).

Los SNPs están causados por un cambio en un único nucleótido de la secuencia del genoma, y pueden ser sinónimos o no sinónimos. En los sinónimos, el cambio no produce una diferencia en la codificación del aminóacido, por lo que no está sometido a presión selectiva. En cambio, los polimorfismos no sinónimos sí que tienen potenciales consecuencias funcionales, ya que el aminoácido resultante cambia y con él se pueden ver alteradas funciones y características importantes. Los SNPs del segundo tipo constituyen dos terceras partes de la variabilidad genética de la bacteria *M. tuberculosis.*
Los LSPs, en cambio, están relacionados con la deleción de secuencias largas, que se distribuyen a lo largo del genoma en regiones conocida como regiones diferenciales o de deleción. Estas regiones sirven para identificar las bacterias y clasificarlas, llegando a la conclusión de que *M. tuberculosis* constituye el ancestro más antiguo, y que a partir de ella han evolucionado las demás cepas (6, 7).

Por otra parte, la variación genética también puede estar causada por variaciones de los elementos repetitivos del genoma de la bacteria. Se estima que alrededor del 10% del genoma de la bacteria está constituido por zonas con altas repeticiones de genes. Estas regiones, conocidas como Direct Repeat (DR) locus, están compuestas por múltiples repeticiones de variantes directas (Direct Variant Repeats (DRVs)). Las DRVs están formadas por repeticiones directas (DR) de 36 bp que se combinan con una secuencia espaciadora (inter-DR) no repetitiva de tamaño similar (35-41 bp) (8, 9).

Para intentar identificar estas variaciones, se han utilizado diversas técnicas moleculares. Entre ellas se encuentran las sondas de ácidos nucleicos, la amplificación genética (a través de la PCR-RFLP del gen *hsp65* o secuenciación de ADN ribosómico (ADNr) 16S), la hibridación en fase sólida, la espoligotipificación (o Spoligotyping por su nombre en inglés) y la técnica MIRU-VNTR (Mycobacterial Interspersed Repetitive Unit – Variable Number Tandem Repeats) (4 – 7).

Las sondas constituyeron la primera prueba basada en detección de ácidos nucleicos, diseñando sondas fluorescentes de ADN que fuera complementario al ADNr de la bacteria. Cuando ambos se unen, se produce una reacción quimio-luminiscente que permite identificar de forma específica y sensible la micobacteria.

La amplificación genética consiste en amplificar ácidos nucleicos de determinadas regiones genéticas, con el objetivo de poder analizarlo después. En la PCR-RFLP, gracias a las enzimas de restricción, se obtienen patrones de bandas que permiten diferenciar las especies de micobacterias. Una muy utilizada es la RFLP-IS6110, que utiliza una sonda correspondiente a un fragmento de la secuencia de inserción IS6110 de la bacteria.

En cambio, en la secuenciación de ADNr 16S permite establecer relaciones filogenéticas muy precisas entre los microorganismos. La hibridación en fase sólida obtiene también un patrón de bandas que permite identificar la especie (6, 10).

La espoligotipificación (o Soligotyping por su nombre en inglés), es un método que permite genotipar de forma rápida el complejo MTBC usando el principio de hibridación inversa. Amplifica 43 secuencias cortas no repetitivas, conocidas como espaciadores, localizadas en la región de repetición directa (DR1). La espoligotipificación es una alternativa a la IS6110-RFLP (8, 11).

Por último, MIRU-VNTR es una herramienta molecular basada en la cuantificación del número de repeticiones en tándem en un loci específico del genoma. Se ha utilizado ampliamente en el ámbito de la epidemiología molecular y en la caracterización genotípica de *M. tuberculosis*. Gracias a este método, es posible obtener un perfil genético específico y único de cada cepa, lo que ha facilitado la vigilancia epidemiológica activa y la identificación rápida de agrupaciones clonales, así como la clasificación de linajes (9).

Los avances en Secuenciación de Nueva Generación (Next Generation Sequencing (NGS)) en esta última década han reducido los costes de la Secuenciación del Genoma Completo (Whole Genome Sequencing (WGS)) e introducir esta herramienta de forma rutinaria para caracterizar las cepas bacterianas (debido a las diferencias en su genoma) y predecir las resistencias. Esto ha permitido mejorar considerablemente la vigilancia epidemiológica de las enfermedades causadas por muchos patógenos, entre ellos el MTBC (12).

Sin embargo, el análisis de los datos obtenidos por estas técnicas es costoso y requiere conocimientos y habilidades específicas. La Bioinformática desempeña un papel crucial en este proceso, ya que permite analizar los datos obtenidos en la secuenciación de una manera precisa y específica. Los bioinformáticos son capaces de estudiar grandes volúmenes de datos mediante algoritmos que aplican métodos estadísticos y computacionales avanzados. Al combinar datos de diferentes fuentes, obtienen una visión global del problema. Además, los conocimientos en biología les permiten interpretar los resultados con precisión, extrayendo información valiosa.

A lo largo de los años, se han desarrollado pipelines que permiten detectar SNPs, variantes de la bacteria o genes relacionados con la resistencia. Los pipelines están conformados por un conjunto de procesos y herramientas, que buscan recopilar datos brutos (Raw Data), para posteriormente analizarlos y presentar resultados en un formato fácilmente comprensible e interpretable (13).

Estos pipelines son útiles en estudios de WGS y entre ellos se encuentran servicios web (CASTB, PhyResSE,TBProfiler) o softwares locales (KvarQ, o Mykrobe Predictor TB). Estas herramientas permiten inferir resistencias a fármacos y clasificar filogenéticamente las cepas sin tener un conocimiento tan especializado. Sin embargo, es peligroso no hacer una comprobación de los resultados obtenidos, por eso es altamente recomendable que un profesional con conocimientos específicos lo ratifique (12).

MTBseq surge como un nuevo pipeline automatizado que permite analizar datos del MTBC a partir de datos obtenidos por NGS. TBProfiler es otra alternativa específica para *M. tuberculosis*, que permite analizar datos de WGS de la bacteria para predecir linajes y resistencias a fármacos.

Además de estas herramientas que son específicas para tuberculosis, otras como Snippy o VariantDetective también son útiles en los análisis de detección de variantes. Snippy se utiliza para detectar SNPs de bacterias de forma rápida. También permite alinear el genoma central (por su nombre en inglés "core genome") de una especie, que no es más que el conjunto de genes comunes a todos los individuos de la especie, frente a un genoma de referencia. Este genoma es especialmente útil, ya que se mantiene invariable, independientemente de las variaciones genéticas individuales de cada individuo. Por otra parte, VariantDetective es una herramienta diseñada para identificar variantes cortas y estructurales en secuencias con formato FASTA o en lecturas con formato FASTQ (12, 14 – 17).

Para utilizar estos softwares, es posible descargarlos localmente en el ordenador, utilizarlos en línea u obtener su imagen de Docker para poder usarla de forma contenerizada. Docker es una plataforma que permite crear, desplegar y ejecutar aplicaciones en contenedores. Estos contendores constituyen un entorno ligero y portátil, que incluye todo lo necesario para la funcionalidad de la aplicación: código, bibliotecas, dependencias, configuraciones… Los contenedores se pueden ejecutar en cualquier sistema que tenga Docker instalado, independientemente de si es una máquina local, un servidor o la nube. Esto garantiza que la aplicación se ejecute en diferentes entornos de la misma manera. Al tener todo encapsulado en un mismo contenedor, se evitan problemas de incompatibilidades. Cada contenedor es independiente, por lo que no hay interferencias con otras imágenes, y al ser más ligeros que las

máquinas virtuales tradicionales son más eficaces, ya que todos ellos comparten el mismo núcleo del sistema operativo (18, 19).

En este trabajo se ha buscado utilizar Docker, porque permite un sencillo control de versiones. Además, la existencia de un repositorio como Docker Hub proporciona acceso directo a muchas imágenes, facilitando aún más su descarga e implementación. Casi todas las herramientas planteadas para este trabajo tienen una imagen de Docker disponible. Sin embargo, VariantDetective no disponía de una imagen descargable. Es por esto por lo que finalmente no será utilizada para este trabajo. En su lugar, se ha utilizado la herramienta TBProfiler, que sí que dispone de imagen de Docker para poder utilizar.

2. ANTECEDENTES

El estudio del genoma de *M. tuberculosis* y de las variantes que aparecen en esta micobacteria ha sido objeto de investigación desde que Robert Koch describió por primera vez el bacilo en 1882 (20). A lo largo de más de un siglo, los esfuerzos se han focalizado en comprender la diversidad genética de esta especie y su relación con las características clínicas y epidemiológicas de la misma.

Muchos estudios se han centrado en buscar cuáles son las diferencias entre las cepas de la bacteria, que afectan tanto a las implicaciones clínicas como epidemiológicas. Para ello, se han utilizado distintas técnicas moleculares, como se ha explicado previamente en la introducción.

Con el desarrollo y expansión de la secuenciación masiva o NGS, se ha logrado un salto significativo hacia la secuenciación del genoma completo. La resolución que alcanza es considerablemente superior a la de las técnicas anteriores, lo que permite obtener información más precisa sobre la evolución genética y las mutaciones específicas responsables de resistencia a fármacos antituberculosos. Un estudio realizado en 2013 por Roetzer et al. reveló que los grupos (clusters) generados por genotipado clásico mediante espoligotipificación en 1997 eran erróneos. En este estudio, el análisis WGS de 86 aislamientos reveló 85 SNPs, además de 36 perfiles con SNPs únicos (21).

A través de WGS, se han podido detectar de manera precisa SNPs y LSPs en el genoma, útiles no solo para rastrear brotes recientes, sino también para estudios filogeográficos y evolutivos. Otro estudio llevado a cabo ese mismo año por Walker et al. secuenció 390 aislamientos de 254 pacientes, incluyendo representación de los principales linajes de *M. tuberculosis*. La divergencia encontrada fue mínima, de raramente más de 5 SNPs por genoma en 3 años (22).

Gracias a la gran conservación del genoma de las cepas del MTBC, es posible analizar datos obtenidos de WGS de cualquiera de las cepas y compararla con una referencia común. Como se ha nombrado previamente, la cepa H37Rv es la que se toma como referencia, y las diferencias respecto a este genoma se consideran SNPs, inserciones y deleciones. Sin embargo, la presencia de regiones repetitivas sigue siendo difícil de analizar de forma precisa, y es por esto por lo que las regiones repetitivas suelen excluirse de los análisis. Hoy en día no hay un estándar internacional que establezca las regiones que se deben excluir, los softwares y parámetros a utilizar para el análisis o la calidad y cantidad de muestras que se requieren para los datos de secuenciación. En un estudio llevado a cabo por Walker et al, se estableció como límite una distancia máxima de 12 SNPs y no de 5, ya que también se encontraron conexiones epidemiológicas consistentes entre aislados de *M.tuberculosis* difiriendo en hasta 12 SNPs (22).

Los avances en WGS han redefinido las relaciones filogenéticas dentro del MTBC. Actualmente, la integración de herramientas bioinformáticas avanzadas permite realizar análisis genómicos exhaustivos que incluyen la identificación automatizada de variantes genéticas, el establecimiento de relaciones epidemiológicas precisas y el rastreo rápido de brotes, lo que representa un avance notable respecto a metodologías previas. Por ello, la secuenciación del genoma completo se ha convertido en la técnica preferida para estudios epidemiológicos, clínicos y diagnósticos (7, 20).

Los datos obtenidos de la secuenciación se procesan a través de diversos pipelines.

Un estudio llevado a cabo por Jajou et al. hizo una comparación de distintos pipelines para el análisis de datos de WGS de 535 aislamientos positivos de MTBC obtenidos de los registros de Países Bajos. Estos datos se analizaron en cuatro institutos europeos distintos:

- Instituto Nacional de Salud Pública y Medioambiente de Países Bajos (RIVM).
- Universidad de Oxford, Reino Unido.
- Centro de Investigación Borstel, en Alemania.
- Instituto Statens Serum de Copenhague, Dinamarca (SSI).

En total, se utilizaron cuatro pipelines basados en el análisis de SNPs, y un método de gen por gen (cgMSLT). Entre todas ellos, el pipeline propuesto por el Centro de Investigación Borstel fue MTBseq. Los resultados obtenidos por cada pipeline fueron después comparados con respecto a su habilidad de establecer nexos epidemiológicos entre los casos de TB. La investigación epidemiológica previa para poder comparar estos resultados fue llevada a cabo por el Servicio Municipal de Salud de Países Bajos. El estudio confirmó que todos los pipelines habían sido capaces de establecer claramente las diferencias entre los casos relacionados epidemiológicamente y los casos no relacionados (23).

Muchos de los pipelines son flujos de trabajo autónomos que requieren grandes requerimientos computacionales para analizar múltiples genomas. Un estudio publicado en 2020 presentó un pipeline automatizado y escalable, conocido como Bacterial Genome Pipeline (BAGEP). Entre sus múltiples funciones, era capaz de realizar el control de calidad de las muestras, mapear las lecturas a un genoma de referencia para encontrar variantes o construir árboles filogenéticos a partir del genoma central, creando una visualización interactiva de SNPs en localizaciones específicas del genoma. El objetivo del estudio era crear un pipeline fácil de usar a partir de herramientas bioinformáticas ya existentes. En este caso se utilizó el marco de trabajo Snakemake, utilizando Fastp para preprocesar las muestras, Snippy para la detección de variantes, Centrifuge para la clasificación taxonómica o vcfR para la visualización de SNPs. BAGEP fue probada y validada con muestras de *Mycobacterium tuberculosis* y *Salmonella enterica*, fácilmente implementable en un ordenador portátil (24).

Por otra parte, herramientas web como TBProfiler también han sido útiles para predecir la resistencia de muestras de tuberculosis a distintos fármacos.

Un estudio publicado en 2019 utilizó WGS como herramienta diagnóstica para el ámbito de la tuberculosis, a través de este servidor web. El estudio buscaba predecir la resistencia de las muestras de *M. tuberculosis* a los fármacos antituberculosos, además de los linajes de las cepas. Se desarrolló una nueva versión de la herramienta TBProfiler para predicciones *in silico* utilizando herramientas nativas como Trimmomatic, BWA/bowtie, SAMtools o BCFtools. Para ello, se añadieron un total de 178 nuevas mutaciones nuevas (marcadores de resistencia frente a 16 fármacos) a la librería existente de la herramienta. Se utilizó una base de datos de 17239 cepas, procesando las muestras con el pipeline de TBProfiler. Para evaluar la especificidad y sensibilidad de la librería, así como la capacidad predictiva de la herramienta, fue necesario poder comparar los resultados de la predicción con un "control". En este caso, se usaron datos de tests fenotípicos de susceptibilidad de fármacos (Drug Susceptibility Test (DST)), así como resultados de un pipeline de análisis con MinION, realizando WGS en 34 replicados de 3 aislamientos con conocidas resistencias, y resultados obtenidos con otra herramienta alternativa, específica de tuberculosis: Mykrobe-predictor. El estudio concluyó que la sensibilidad y especificidad de la herramienta propuesta (TBProfiler) eran altas (rangos entre 83%-94% y 96%-98% respectivamente). Además, al comparar los resultados con el de otras herramientas como Mykrobe predictor, se demostró la superioridad predictiva para detectar resistencia a fármacos de primer y segunda línea (16) .

Similarmente, otro estudio realizado en 2021 trató de evaluar el rendimiento de la secuenciación de lecturas largas (long-read (LR) y cortas (short-read (SR)) para la detección de mutaciones de resistencias anti-TB usando las herramientas TBProfiler y Mykrobe predictor. Para ello se utilizaron 24 muestras, incluyendo sus perfiles de susceptibilidad a fármacos (obtenidos con DST). Los resultados pusieron de manifiesto que la predicción de resistencias era más precisa en el caso de los ensamblados SR, siendo ambas herramientas útiles y fiables (25).

Con toda esta información, se pone de manifiesto la verdadera importancia y utilidad de las herramientas bioinformáticas a la hora de analizar muestras de bacterias tan relevantes como *M. tuberculosis*, permitiendo una comprensión más profunda de su diversidad genética y resistencia a fármacos. Estos avances no sólo han mejorado la precisión en el diagnóstico y tratamiento, sino que también han facilitado el seguimiento epidemiológico y la implementación de estrategias de control más efectivas. Sin embargo, en la literatura disponible no hay ninguna investigación previa que estudie la combinación de herramientas dockerizadas para el análisis genómico de *M.tuberculosis,* justificando la utilidad de este trabajo.

3. OBJETIVOS

El objetivo principal del proyecto consiste en comparar diferentes herramientas disponibles para el análisis de variantes de muestras de *Mycobacterium tuberculosis*. En concreto, la superioridad, o no, de MTBseq frente a Snippy y TBProfiler. Se establecerá un flujo de trabajo automatizado para facilitar todo el proceso.

Entre los objetivos específicos, se encuentran los siguientes:

- Automatizar la ejecución del flujo de trabajo al detectar nuevas muestras en una carpeta específica.
- Desarrollar un flujo de análisis comparativo entre múltiples muestras con TBjoin, TBamend, y TBgroups.
- Desarrollar un flujo de trabajo para las herramientas MTBseq, Snippy y TBProfiler.
- Detectar, usando las tres herramientas, diferencias entre pares de muestras con menos de 15 SNPs y determinar qué SNPs presentan dichas diferencias.
- Determinar si los resultados de las 3 herramientas utilizadas son comparables o no.

Como objetivo adicional, se ha planteado encapsular el flujo de trabajo completo en un único contenedor Docker listo para ser desplegado en cualquier entorno.

4. MATERIAL Y MÉTODOS

4.1. Fuentes de datos

Los datos que se van a utilizar son muestras de *M.tuberculosis* obtenidas del Instituto Aragonés de Ciencias de la Salud (IACS).

El conjunto consiste en 8 aislados de *M.tuberculosis* procedentes del Grupo de Genética de Micobacterias y de la Unidad de Biocomputación del Instituto Aragónes de Ciencias de la Salud. Las muestran fueron secuenciadas con tecnología Illumina y un secuenciador Miseq. Los resultados obtenidos en FASTQ son el punto de partida del trabajo, siendo 8 muestras con 2 direcciones cada una (16 ficheros FASTQ).

4.2. Tabla comparativa de las herramientas internas utilizadas por los pipelines

Nombre	Herramientas internas que utiliza	Objetivos	Utilización	Ficheros de entrada	Ficheros de salida
MTBseq	BWA, SAMtools, PICARD, GATK	Detección y anotación de variantes, clasificación filogenética, análisis comparativo de muestras múltiples	Flujo de trabajo estandarizado y modular en Perl	Archivos FASTQ	.vcf, .bam, .pileup, .tab
Snippy	BWA-MEM, Freebayes	Llamado de variantes y alineamiento de genoma central	Rápido, produce un conjunto de archivos en una sola carpeta	Archivos FASTQ, secuencia de referencia en formato FASTA o GENBANK	.vcf, .tab, .gff, .bam .tsv
TBProfiler	Trimmomatic, Bowtie2, BWA, Minimap2, Freebayes, BCFtools, GATK, LoFreq, Pilon	Predicción de linaje y resistencia a fármacos	Pipeline para datos de secuenciación de genoma completo	Archivos FASTQ, BAM	.json, .bam, .vcf

Tabla 1. Tabla comparativa de las tres pipelines y sus herramientas internas.

La Tabla 1 muestra la relación entre los pipelines comerciales utilizados y las herramientas internas que usa cada uno de ellos. En el siguiente punto, se explica de forma resumida el proceso de llamado de variantes. Internamente, los tres pipelines que se han implementado en el trabajo realizan estos pasos, a través de las distintas herramientas internas que se pueden apreciar en la tabla.

4.3. Análisis básico de llamado de variantes (variant calling): herramientas internas

El llamado de variantes se define como el proceso de identificar variantes a partir de datos de secuenciación. Para ello, se parte de datos en formato FASTQ, obtenidos de la secuenciación masiva (NGS) del genoma o exoma completo. Este proceso es de gran importancia en la identificación de SNPs, inserciones y deleciones, y es muy útil para buscar y analizar variaciones genéticas y relaciones filogenéticas, entre otras aplicaciones. Sin embargo, constituye un análisis bioinformático predictivo, por lo que es necesario realizar una investigación experimental a la hora de realizar conclusiones sobre los resultados obtenidos (26).

El esquema básico que sigue el proceso de llamado de variantes se podría resumir en los siguientes pasos (27):

4.3.1. Control de calidad y preprocesado de datos

Es necesario evaluar la calidad de las lecturas obtenidas en la secuenciación antes de comenzar cualquier análisis. No es exclusivo de los análisis de llamado de variantes, se utiliza como paso inicial para la mayoría de los análisis bioinformáticos. Para ello se pueden utilizar distintos recursos.

FastQC proporciona un conjunto modular de análisis que permite obtener una primera impresión de los datos antes de continuar con análisis posteriores. Es una herramienta basada en Java que analiza tanto los errores originados en el secuenciador como en el material de inicio (Raw data). A través de un archivo HTML que se obtiene como salida (output), se puede ver de forma sencilla la calidad de las lecturas a varios niveles (28).

Por otra parte, Trimmomatic, Cutadapt o Fastp permiten realizar un preprocesado de los datos obtenidos, eliminando zonas no deseadas como secuencias de adaptadores o primers, o filtrando lecturas de baja calidad.
En este caso se ha utilizado **Fastp**, ya que es la herramienta más rapida de las tres, realizando tanto análisis de calidad como preprocesado de las lecturas. Es una herramienta que permite filtrar secuencias de baja calidad y eliminar secuencias adaptadoras, proporcionando también un reporte de la calidad de las muestras. Permite analizar datos de lecturas de extremos emparejados (paired-ends) y de lecturas de un sólo extremo (single-end) de forma rápida y eficiente, lo que la hace una herramienta versátil y útil. Hay una gran variedad de filtros y opciones que se pueden aplicar para hacer el análisis más completo y adecuarlo a las secuencias de partida (29, 30).

Una vez hecho el análisis, se pueden utilizar herramientas para unirlos todos en un único reporte. **MultiQC** es una herramienta desarrollada en Python que permite la creación de un único reporte con gráficos interactivos para la visualización de diferentes análisis realizados en múltiples muestras. Para ello, se utilizan los ficheros generados por las distintas herramientas previas (en este caso FastQC y Fastp), generando un resumen de las estadísticas en un archivo HTML (31).

4.3.2. Alineamiento de secuencias

Es un proceso fundamental en el análisis bioinformático, y consiste en alinear dos o más secuencias de ADN, ARN o proteínas por sus regiones similares, intentando identificar similitudes y diferencias. En el campo de la bioinformática, hay varias herramientas ampliamente utilizadas para el alineamiento de secuencias:

Burrows Wheeler Aligner (BWA) es uno de los alineadores más utilizados. Es un alineador basado en la transformada de Burrows-Wheeler, eficiente, preciso y flexible, lo que lo convierte en una alternativa muy popular, base de muchos pipelines comerciales. Hay tres variaciones de BWA, dependiendo de lo que se quiera analizar. BWA-backtrack está indicado para lecturas cortas (<70 bp), mientras que BWA-SW está diseñado para lecturas largas y de baja calidad. BWA-MEM es el algoritmo más reciente, y está recomendado para secuencias de longitud media larga (70-1000 bp), habiendo reemplazado a los dos anteriores. Para utilizar BWA, es necesario disponer de un genoma de referencia adecuado, proporcionando una base sólida para el posterior llamado de variantes (32).

Bowtie2 es un alineador basado en BWA, pero optimizado. Es más eficiente en memoria y rápido que su antecesor, lo que lo hace útil en pipelines de genómica comparativa. Sin embargo, es ligeramente menos sensible en determinados casos (33).

Minimap2 es otra herramienta versátil utilizada en el alineamiento de secuencia de ADN y ARN, contra una base de datos de referencia grande. Esta herramienta es significativamente más rápida que otras, tanto en lecturas cortas como largas, manteniendo la precisión. Además, es especialmente útil en datos de secuenciación de alta diversidad genética (34).

4.3.3. Mapeado y procesamiento de alineamientos

Antes de proceder con el análisis de variantes, es necesario eliminar posibles errores de secuenciación, manteniendo la variabilidad genética de origen de las muestras. Para ello, se utilizan distintas herramientas (27).

SAMtools es un conjunto de programas que son útiles para interactuar con datos de secuenciación de alto rendimiento. Está compuesto por tres repositorios diferentes: SAMtools, BCFtools y HTSlib.

El primero se utiliza en archivos de formato SAM (Sequence Alignment Map), BAM (Binary Alignment Map) y CRAM (Compressed Reference-oriented Alignment Map), y permite la lectura, escritura y manipulación de estos. SAM es el formato original y contiene datos mapeados de secuenciación. BAM y CRAM son formas comprimidas del archivo SAM, y se diferencian en el rango de pérdidas que se producen al comprimir el archivo (BAM sin pérdidas, CRAM puede variar dependiendo de la cantidad de compresión).

BCFtools permite leer y escribir archivos en varios formatos: formato de llamado binario (Binary Call Format (BCF2)), formato de llamado de variantes (Variant Call Format (VCF)) y formato de VCF genómico (genomic VCF (gVCF)). Además, realiza llamado y filtrado de SNPs y variantes de inserción y deleción cortas (también conocidas como indels).

Por último, HTSlib es una librería en C que permite leer y escribir datos de secuenciación de alto rendimiento (35, 36) .

Picard es una línea de comandos basada en Java utilizada para marcar duplicados, generar estadísticas del alineamiento o crear diccionarios de secuencias. Es capaz de manejar varios formatos de datos, como BAM, SAM, CRAM o VCF, esenciales en el análisis de datos genómicos (37).

GATK (Genome Analysis Toolkit) HaplotypeCaller es una colección de herramientas de líneas de comando para procesar y analizar datos de secuenciación de alto rendimiento. El objetivo es descubrir variantes genómicas, pudiendo utilizar las herramientas por separado o unidas en un flujo de trabajo completo. GATK contiene ya de por sí una versión de Picard. Fueron inicialmente diseñadas para el análisis de genoma humano, pero se pueden adaptar para organismos no humanos, incluso para organismos no diploides. Entre los algoritmos que utiliza, se encuentra el de recalibración de la puntuación de la calidad de las bases o el de recalibración de la puntuación de la calidad variantes. El primero es un paso de preprocesado de los datos que detecta errores sistemáticos producidos por la máquina de secuenciación a la hora de estimar la precisión de cada base (a través de Base Quality Recalibrator). El segundo algoritmo es una técnica de filtrado que se aplica al conjunto en el que se hará el llamado de variantes, permitiendo modelar el perfil técnico de variantes (36).

4.3.4. Llamado de variantes

Una vez hecho el mapeado y procesamiento de alineamientos, es posible hacer el llamado de variantes, analizando las variantes que las lecturas generadas presentan con respecto al genoma de referencia (27).

Freebayes es una herramienta que detecta las variantes utilizando un detector de variantes genética Bayesiano. Está diseñado para encontrar polimorfismos pequeños, específicamente SNPs, indels, o polimorfismos de múltiples nucleótidos. Para ello, partiendo de un archivo BAM, calcula la probabilidad posterior de cada genotipo en cada posición de la referencia, y selecciona el genotipo más probable. Después, lo compara con el de referencia y obtiene las diferencias. Por lo tanto, el archivo de entrada es un archivo BAM y el genoma de referencia, y la salida un archivo VCF. Es una detección basada en haplotipos, definidos en la RAE como "conjunto de alelos localizados en una región de un cromosoma, que suelen heredarse como un bloque". De esta manera, el llamado de variantes se basa en las secuencias actuales de las lecturas que han sido alineadas a la referencia, y no a su alineamiento exacto (38, 39).

GATK también se puede utilizar como algoritmo de detección de variantes estructurales (Structural Variants (SVs)) en uno o más individuos. Las SVs son reordenamientos del DNA que implican al menos 50 nucleótidos. Entre ellas encontramos deleciones, duplicaciones, inversiones o translocaciones. Representan una de las mayores fuerzas que dirige la evolución del genoma. Algunas de ellas pueden provocar interrupciones en genes que codifican proteínas, o afectar a la regulación de distintos mecanismos. El proceso de llamado de variantes utiliza un enfoque basado en reensamblaje local de haplotipos (36).

VarScan es una herramienta de software de independiente desarrollada por el Instituto de Genome en la universidad de Washington. Busca detectar variantes en datos de NGS, tanto en exoma como en genoma completo. Permite detectar distintos tipos de variaciones, tanto somáticas como germinales:

- Variantes germinales (SNPs e indels), tanto en muestras individuales como en pools de muestras.
- Variantes multimuestra (multisample).
- Mutaciones somáticas.
- Alteraciones en el número de copias (somáticas).

Para ello, utiliza una estrategia heurística/estadística desarrollada en Java, a diferencia de muchos programas de llamado de variantes que emplean estadística Bayesiana (40).

Por último, **BCFtools** es un programa utilizado para llamado de variantes y manipulación de ficheros de tipo VCF y BCF, que es su homólogo binario. Realiza el llamado de variantes a partir del archivo de salida obtenido del comando *samtools mpileup*, que produce la probabilidad de genotipos en ambos formados VCF y BCF. Se basa en un enfoque más tradicional de pila de lecturas (pileup) (41).

4.4. Pipelines automatizados para análisis de variantes de datos NGS de MTBC

A continuación, se explica cómo funciona cada uno de los tres pipelines, basándose en las herramientas internas comentadas previamente.

4.4.1. MTBseq

Es una herramienta automatizada creada para el análisis de datos de WGS de aislamientos de *M. tuberculosis*. Proporciona un flujo de trabajo estandarizado que permite la detección de variantes, la anotación de variantes relacionadas con la resistencia o clasificar las cepas en función de sus relaciones filogenéticas. Además, MTBseq posibilita realizar un análisis comparativo de muestras múltiples, generando listas conjuntas de variantes. Utiliza softwares de acceso libre para el mapeado de lecturas (SAMtools o BWA), para el recalibrado de las bases (PICARD, GATK) o llamado de variantes (SAMtools). Todo ello está implementado en un flujo de trabajo en lenguaje de programación Perl. Al presentar una arquitectura modular, se puede personalizar el pipeline según las necesidades del usuario. Por defecto, utiliza el genoma de *M.tuberculosis* H37Rv como genoma de referencia, tanto para el mapeo de referencia como para la anotación el llamado de variantes. Esta herramienta puede ser ejecutada tanto en entornos locales como en máquinas virtuales (12, 42).

Utiliza como archivo de entrada (input) archivos FASTQ, que deben presentar un formato de nombre específico, indicando el ID de la muestra, el ID de la librería y la dirección. Se puede correr el flujo de trabajo completo (utilizando la opción --TBfull) o ir módulo por módulo. El flujo de trabajo es el siguiente:

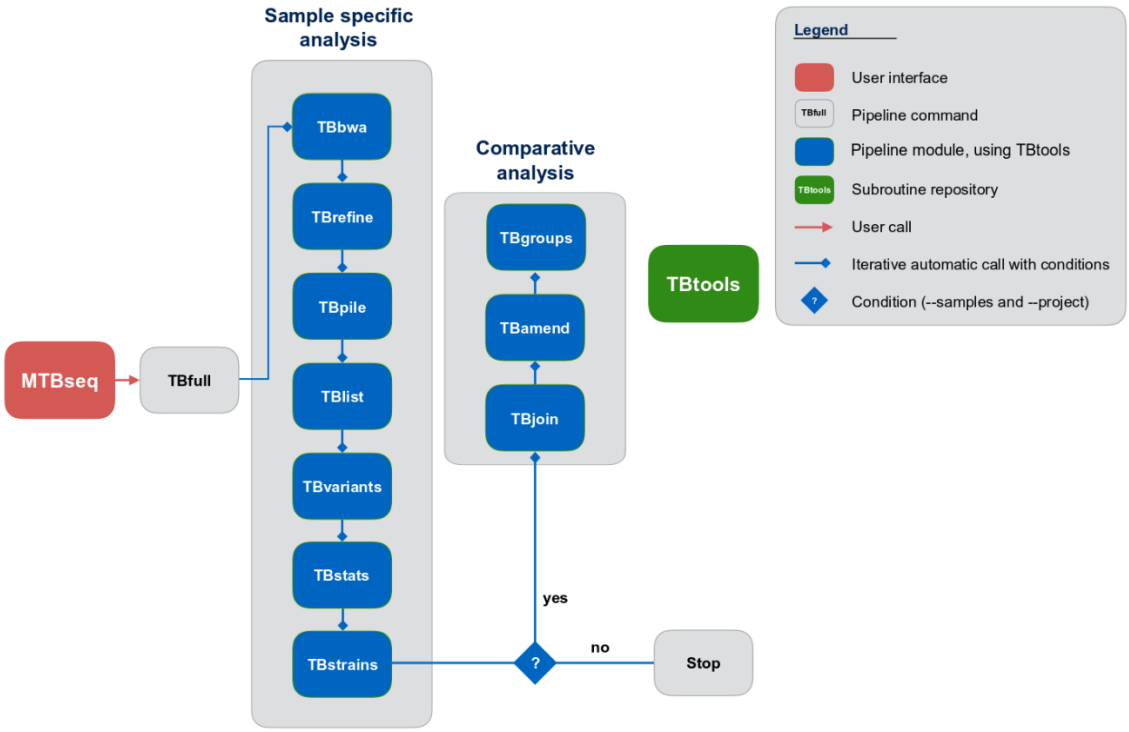

Figura 1. Explicación del flujo de MTBseq obtenida del manual de la herramienta (42).

En la Figura 1 se puede apreciar el flujo que sigue el programa. Los tres últimos pasos, permiten realizar el análisis comparativo de muestras. Utilizando TBfull se realizan de forma automática todos los pasos sin necesidad de llamarlos uno por uno. En cambio, si se prefiere ir implementando los pasos por separado, el flujo es el siguiente:

- **TBbwa**: mapeo a la secuencia de referencia, utilizando la herramienta BWA. Los resultados se guardan en la carpeta Bam.
- **TBrefine**: realineamiento alrededor de las inserciones y deleciones, además de recalibración de llamado de bases, utilizando GATK. Los resultados se guardan en la carpeta GATK_Bam.
- **TBpile**: creación de archivos pileup a partir de los archivos de mapeado refinados, utilizando SAMtools. Los resultados se guardan en la carpeta Mpileup.
- **TBlist**: creación de listas de posiciones a partir de archivos pileup, capturando la información esencial del mapeado, y creando una lista que contiene el contaje de nucleótidos de las lecturas apeados para cada posición de la referencia. Los resultados se guardan en la carpeta Position_Tables.
- **TBvariants**: detección de variantes de la lista de posiciones, usando SAMtools. Además de la detección, también se realiza la anotación de variantes. Los resultados se guardan en la carpeta Called. Se pueden añadir parámetros para personalizar este paso:
    - --snp_vars: añade un filtro extra para sólo detectar las variantes que sean SNPs.

  - --mincovf: especifica la mínima cobertura que debe presentar una lectura en dirección hacia delante (Forward). Por defecto es 4.
  - --mincovr: especifica la mínima cobertura que debe presentar una lectura en dirección reversa (Reverse). Por defecto es 4.
  - --minphred20: establece un mínimo número de lecturas que indiquen un alelo con un puntaje phred de al menos 20. Por defecto es 4.
  - --minfreq: establece la frecuencia mínima de un alelo para que sea reportado. Por defecto es 75.
- **TBstats**: cálculo de calidad del mapeo y variantes detectadas, utilizando SAMtools flagstat, que proporciona un resumen de las estadísticas de alineamiento. Los resultados se guardan en la carpeta Statistics.
- **TBstrains**: clasificación de linajes basada en un conjunto de SNPs filogenéticos. Los resultados se guardan en la carpeta Classification. Se pueden añadir parámetros para personalizar el paso:
  - --mincovf
  - --mincovr
  - --minphred20
  - --minfreq
- **TBjoin**: análisis comparativo de SNPs del conjunto de muestras especificado por el usuario. Los resultados se guardan en la carpeta Joint. Se pueden añadir parámetros para personalizar este paso:
  - --project indica el nombre del proyecto.
  - --samples: indica el fichero samples.txt necesario para el análisis, con una estructura específica.
  - --snp_vars
  - --mincovf
  - --mincovr
  - --minphred20
  - --minfreq
- **TBamend**: post-procesamiento de las tablas de variantes unidas en el paso anterior. Los resultados se guardan en la carpeta Amend. Se pueden añadir parámetros para personalizar el paso:
  - --project
  - --samples
  - --mincovf
  - --mincovr
  - --minphred20

  - --minfreq
- **TBgroups**: inferencia de aislamientos probablemente relacionados por sus distancias entre distintas posiciones de SNPs, a través de un proceso aglomerativo. Los resultados se guardan en la carpeta Groups. Se puede añadir una opción de distancia:
  - --distance: establece una distancia entre SNPs que se utiliza para clasificar las muestras en grupos de muestras. Si la distancia entre SNPs es igual o menor al valor establecido, las muestras se agrupan juntas. Por defecto es 12.

Todas las llamadas permiten además establecer un número de hilos (--threads), refiriéndose al número de núcleos de CPU que se van a utilizar para ese paso en concreto. Esto permite ejecutar tareas en paralelo, lo cual es especialmente útil en análisis bioinformáticos intensivos como el mapeo de secuencias o el llamado de variantes (12, 42).

4.4.2. Snippy

Snippy permite hacer llamado de variantes y alineamiento de genoma central de forma rápida. La herramienta es capaz de mapear las lecturas obtenidas con NGS frente a un genoma de referencia haploide utilizando BWA-MEM. Después, detecta variantes entre el genoma de referencia y los datos de las lecturas mapeadas, utilizando la herramienta nativa Freebayes, encontrando polimorfismos de único nucleótido (SNPs) o de múltiples nucleótidos (MNPs), de tipo complejo (combinación de las dos anteriores), inserciones y deleciones.

Está diseñada para ser rápida y producir un conjunto de archivos de salida en una sola carpeta. A partir de los resultados obtenidos, se puede generar un alineado de los SNPs centrales ("core SNPs"), muy útil para crear un árbol filogenético de alta resolución. Un sitio central es una posición genómica que está presente en todas las muestras, ya sea con el mismo nucleótido en cada muestra (sitio central monomórfico) o con variaciones en alguna de las muestras (sitio central polimórfico o variante). Si no se tienen en cuenta las inserciones y deleciones, los sitios variantes conformarán el genoma de SNPs central.

En este caso, es necesario introducir una secuencia de referencia en formato FASTA (sin anotar) o GENBANK (anotada). Si es anotada, además de detectar las variaciones, también intentará averiguar cuál es el efecto de esos cambios en los genes. Además de la referencia, se introduce como entrada un fichero de lecturas en formato FASTQ o FASTA y una el nombre de una carpeta donde almacenar los resultados obtenidos. Es posible utilizar parámetros para personalizar el análisis:

- --mincov hace referencia al mínimo número de lecturas necesaria cubriendo un sitio, siendo por defecto 10.
- --minfrac se refiere a la mínima proporción de esas lecturas que deben diferir de la referencia para que sean reportadas. Por defecto 0,9.

- --minqual representa la mínima calidad del llamado de variantes (por defecto 100).

Los ficheros de salida se pueden obtener en diversos formatos, entre los que se encuentran .tab, .vcf, .gff o .bam, además de ficheros FASTA con los alineamientos.

En el caso de querer automatizar la ejecución de múltiples análisis de Snippy, lo que se puede hacer es utilizar la opción snippy-multi, que es una herramienta auxiliar de Snippy. Para ello, es necesario disponer de un archivo llamado "input.tab" con el nombre y la dirección de las muestras a analizar. Con ese fichero de entrada se genera un script de Bash (run_snippy.sh), que lanza Snippy para cada muestra por separado, pero de manera automatizada. Además, la última línea del script es un comando que ejecuta snippy-core, que permite alinear SNPs comunes (17, 43).

Esta herramienta auxiliar es útil porque genera una serie de archivos de salida que la ejecución simple de Snippy no crea, y que son necesarios para análisis posteriores con snp-dists, que calcula distancias entre muestras. Entre ellos el esencial es core.full.aln, un fichero FASTA que contiene un alineamiento múltiple de secuencias del genoma central de la bacteria (43).

Tanto Snippy como snippy-multi permiten, al igual que MTBseq, establecer un número de hilos (--cpus) para realizar el análisis de forma paralela con varios núcleos de CPU.

Snp-dists es una herramienta independiente a Snippy, creada por el mismo autor, que permite convertir un fichero FASTA en una matriz de distancias de SNPs (formato .tsv). Es necesario disponer del archivo core.full.aln, ya que contiene el alineamiento múltiple de todas las muestras procesadas con Snippy, y es esencial para calcular las distancias entre SNPs. Al igual que en el resto de las herramientas, es posible añadir opciones para personalizar la salida (44).

4.4.3. TBProfiler

Es un pipeline que permite predecir el linaje y la resistencia a fármacos de datos de secuenciación de genoma completo de *M. tuberculosis*. Los ficheros de entrada en formato FASTQ primero se filtran con Trimmomatic. Después, se hace un alineamiento al genoma de referencia de *M. tuberculosis* H37Rv utilizando Bowtie2, BWA o Minimap2. En este caso, la referencia por defecto es la misma que la utilizada en la herramienta MTBseq, con una longitud de 4411532 pares de bases). Es posible utilizar un archivo BAM y así evitar el alineamiento, pero se debe asegurar que se ha creado con una versión específica que corresponda con el genoma de referencia. Por último, se realiza una búsqueda de variantes pequeñas y grandes deleciones que pueden estar asociadas con resistencias a fármacos, usando Freebayes (por defecto), BCFtools, GATK, LoFreq o Pilon. Las variables obtenidas pueden filtrarse en función de varios parámetros:

- --call_whole_genome: permite realizar el llamado de variantes en un genoma completo.

- --snp_dist: establece la distancia mínima entre SNPs. No hay valor predefinido. Es una función experimental, por lo que los resultados pueden ser inesperados.

Como archivos de salida por defecto se obtienen ficheros .json, .bam y .vcf, aunque también es posible guardarlo en formatos .txt (--txt) o .csv (--csv). Adicionalmente, es posible realizar espoligotipificación experimental, con la opción --spoligotyping. Se considera experimental porque aún sigue en desarrollo y no está completamente validada. Permite predecir el patrón de espoligotipos a partir de los archivos .fastq y .bam (14, 45).

Al igual que las dos herramientas anteriores, en TBProfiler también es posible establecer el número de núcleos de CPU que se quieren utilizar para el análisis (-t).

4.5. Manejo de Docker

El trabajo se ha desarrollado utilizando Docker en través de Visual Studio Code. Como se ha comentado previamente, se basa en el lanzamiento de contenedores a partir de imágenes previamente descargadas (a través del comando *docker run*). Esto permite tener herramientas contenerizadas, con un sencillo de las dependencias y las versiones, independientemente del servidor en el que se esté trabajando. Los contenedores se pueden lanzar de maneras distintas usando diferentes comandos. Es necesario conocer el nombre de la imagen de Docker que se va a usar (Repository) y la etiqueta específica de la imagen (TAG).

4.5.1. docker run –-rm -v "dirección de la carpeta/:/data" repository:TAG sh -c "acción(es) a realizar".

Por una parte, se puede lanzar el comando para que el contenedor se elimine automáticamente al detenerse (utilizando la opción *--rm*). De esta manera, se evita acumular contenedores inactivos en el sistema, ya que, una vez se ejecuten las acciones que hemos pedido, el contenedor se eliminará automáticamente.

La opción *-v* permite montar un volumen, siendo *dirección de la carpeta* la ruta del sistema local (en este caso el servidor) y */:data* la ruta dentro del contenedor donde se montará la carpeta. Así, se permite el acceso del contenedor a los archivos del sistema local.

El comando *sh –c "acción(es) a realizar"* ejecuta un shell (*sh*) dentro del contenedor, y utiliza la opción *-c* para ejecutar las acciones que se especifican en la cadena de comandos. Esas acciones se ejecutarán dentro del contenedor.

4.5.2. docker run -v "dirección de la carpeta/:/data" repository:TAG sh -c "acción(es) a realizar".

Es una variación de la llamada anterior. Al quitar la opción --rm, se evita la eliminación automática del contenedor al finalizar. Esto permite mantener el contenedor vivo por si se necesita para algún otro propósito, como utilizar docker stats o docker logs.

4.5.3. docker run –it -v "dirección de la carpeta/:/data" repository:TAG /bin/bash

Por otra parte, es posible lanzar el comando de manera que el contenedor se vuelva interactivo y se puedan ejecutar comandos dentro del contenedor. Para esto se utiliza la opción *-it* (modo iterativo y asignación de pseudo-terminal). El comando completo sería *docker run –it -v "dirección de la carpeta/:/data" repository:TAG /bin/bash*.

La opción *-v* permite crear el volumen, igual que en la opción anterior. */bin/bash* ejecuta el shell de Bash dentro del contenedor, lo que permite interactuar con el contenedor en tiempo real. En este caso, al lanzar el contenedor se creará una pseudo-terminal que permite al usuario interactuar con el contenedor directamente, como si estuviera en una terminal de Linux. Una vez terminado, es necesario salir para volver a la sesión actual (46).

4.6. Análisis de recursos y costes

Para hacer el análisis de recursos consumidos por cada contenedor se ha utilizado docker stats. Este comando devuelve un flujo de datos a tiempo real de los contenedores que están lanzados. Permite hacer un seguimiento detallado del uso de recursos de los contenedores. Es posible establecer un formato personalizado, en función de las necesidades del usuario, incluyendo las columnas que se quiera entre las disponibles del comando (CONTAINER ID, NAME, CPU% and MEM%, MEM USAGE/LIMIT, NT I/O, BLOCK I/O, PIDs) (47).

4.7. Control de errores

Los propios scripts tienen control de errores, comprobando que las muestras han sido correctamente copiadas o que siguen la estructura necesaria. Sin embargo, es útil generar ficheros de registro (también conocidos como logs) .log que recojan lo que se muestra por pantalla (stdout) y los errores que surjan a lo largo del proceso (stderr). Para ello, es necesario crear un .log en cada script que vaya a recoger toda la información, y posteriormente volcar esa información en el fichero .log. Para recoger también la información de las llamadas a los contenedores, se utiliza el comando docker logs, que recupera los logs presentes en el tiempo de ejecución (48, 49).

4.8. Flujo de trabajo

La Imagen 2 muestra el flujo de trabajo de las herramientas que se ha seguido a lo largo del proyecto. En las muestras iniciales, se realiza primero el análisis de calidad, que consiste en ejecutar las herramientas FastQC y Fastp, y unificar los informes usando MultiQC.

A continuación, se realiza el análisis genómico, a través de las herramientas MTBseq, Snippy y TBProfiler.

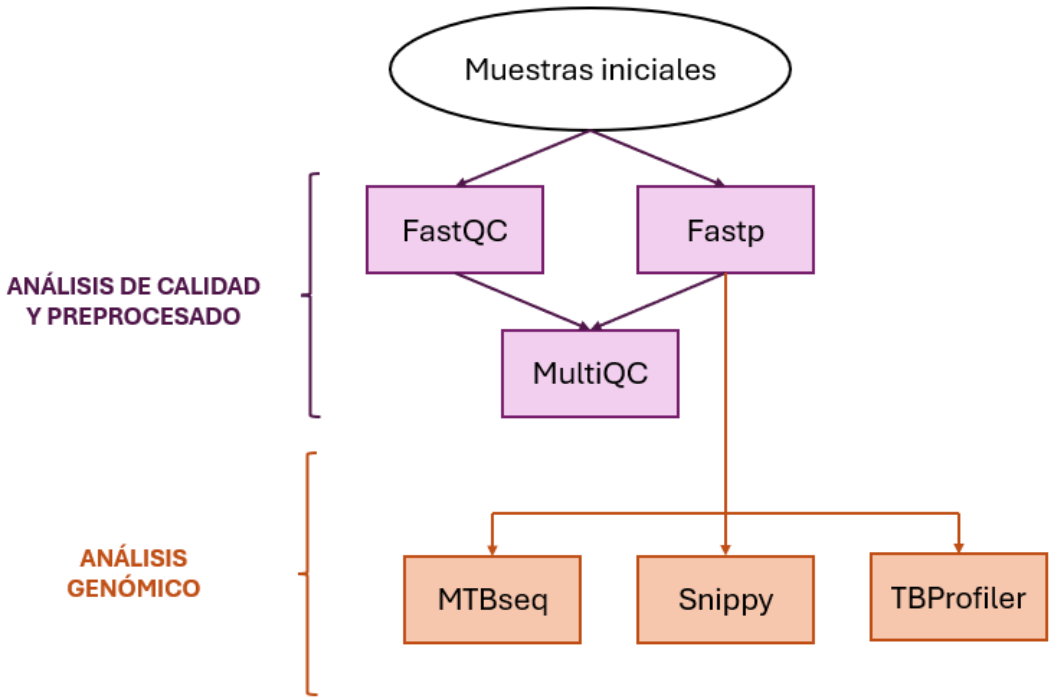

Figura 2. Flujo de trabajo seguido en el proyecto. Óvalo blanco: muestras iniciales; rectángulos violetas: herramientas pertenecientes a la etapa de análisis de calidad y preprocesado; rectángulos naranjas: herramientas pertenecientes a la etapa de análisis genómico.

5. IMPLEMENTACIÓN

En este trabajo se ha automatizado el proceso de análisis de datos genómicos de muestras de secuenciación, cubriendo las etapas de validación de muestras, control de calidad, filtrado, análisis de variantes y perfiles genómicos. Para ello, se ha desarrollado un código mayoritariamente en Bash, utilizando diversas herramientas bioinformáticas integradas en contenedores Docker. De esta manera, se garantiza la reproducibilidad y escalabilidad de los resultados, a través de un flujo de trabajo sencillo.

Para este trabajo, únicamente se ha realizado parte del código, ya que el resto había sido previamente desarrollado en las prácticas curriculares (en concreto, el código de FastQC, Fastp, multiQC y parte del de MTBseq).

Todo el código está disponible en el repositorio GitHub MTB-Pipeline-VariantAnalysis (50).

5.1. Imágenes de Docker utilizadas

La Tabla 2 muestra todas las imágenes de Docker que se han utilizado para este proyecto:

NAME	REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
FastQC	staphb/fastqc	latest	2f5ea875ab87	2 years ago	486MB
Fastp	quay.io/biocontainers/fastp	0.24.0--heae3180_1	c6641ec44abd	5 months ago	29.4MB
MultiQC	quay.io/biocontainers/multiqc	1.27.1--pyhdfd78af_0	c80c86364291	2 months ago	888MB
MTBseq	quay.io/biocontainers/mtbseq	1.1.0--hdfd78af_0	164ff6a15c0c	20 months ago	1.47GB
Snippy	quay.io/biocontainers/snippy	4.6.0--hdfd78af_5	202cf5d385a1	6 months ago	2.14GB
snp-dists	staphb/snp-dists	latest	f319891b91eb	3 years ago	269MB
TBProfiler	quay.io/staphb/tbprofiler	latest	3a3fee4fd2d1	2 months ago	2.53GB

Tabla 2. Imágenes de Docker utilizadas para el proyecto, incluyendo Nombre (Name), Repositorio (Reporistory), Etiqueta (Tag), identificador de la imagen (Image ID), fecha de creación (Created) y tamaño (Size).

5.2. Código desarrollado

Esta sección se divide en dos subsecciones. Por un lado, hay scripts que fueron desarrollados en prácticas *(main.sh, check_samples.sh, FastQC.sh, fastp.sh y parte de MTBseq.sh y filterTab.py*), y que para este proyecto no se ha cambiado a excepción de pequeñas modificaciones: mejorar la documentación para una sencilla comprensión y añadir lo necesario para analizar la gestión de recursos. Aun así, se tendrán en cuenta los resultados obtenidos, porque son relevantes para el flujo completo del análisis.

Por otra parte, hay scripts desarrollados expresamente para el proyecto (final MTBseq.sh, corrección filterTab.py, Snippy.sh, TBProfiler.sh, select_option.sh y capture_stats.sh).

El flujo completo se ha organizado en una serie de scripts modulares que se ejecutan secuencialmente a partir del script principal *main.sh*. Los scripts se distribuyen de la siguiente manera:

- Validación y preprocesamiento de datos: *check_samples.sh*, *FastQC.sh* y *fastp.sh*.
- Selección de flujo de trabajo para el análisis genómico: *select_option.sh*.
- Análisis genómico: *MTBseq.sh*, *Snippy.sh* y *TBProfiler.sh*.
- Filtrado de datos estadísticos: *filterTab.py*.
- Monitoreo y estadísticas: *capture_stats.sh*.

El script principal *main.sh* recibe como parámetro el directorio de trabajo. Controla la ejecución en orden lógico de los pasos que se mencionan, gestionando posibles errores que se produzcan a lo largo del proceso. Además, crea las carpetas donde se van a almacenar los resultados (Analysis) y las muestras iniciales (Raw_Data). Desde este script se llama a *check_samples.sh*, *FastQC.sh* y *fastp.sh*. Además, se realiza una unión de los reportes obtenidos en FastQC y Fastp utilizando MultiQC, siempre que haya alguna muestra nueva.

5.2.1. Check_samples.sh

Este script se encarga de comprobar si existen archivos .fastq.gz en el directorio proporcionado como directorio de trabajo. Si es así, y las muestras no existen todavía en la carpeta Raw_Data, establece que hay muestras nuevas (necesario para después lanzar MultiQC) y las mueve a la carpeta Raw_Data. Si las muestras ya existen en Raw_Data, esa muestra se salta. Si no hay archivos que coincidan con el formato necesario, se avisa al usuario y el script finaliza.

5.2.2. FastQC.sh

Este script se encarga de realizar el análisis de calidad utilizando la herramienta FastQC. Para ello, primero comprueba que existen muestras que analizar en Raw_Data con el formato necesario (.fastq.gz). Si no existen termina el proceso. Si existen muestras comienza el análisis. Lo primero es crear las variables para los ficheros de entrada y de salida, evitando introducir a mano cada muestra y permitiendo una automatización del proceso. Para cada muestra de la carpeta Raw_Data, comprobará que no se existan los ficheros de salida (lo que indicaría que FastQC ya se ha realizado previamente para esa muestra). Si los encuentra, no realiza el análisis para esa muestra, si no, llama al contenedor de Docker para hacer el análisis. El comando utilizado para la llamada es el siguiente:

```
docker run -d -v "$directory/:/data" --name $container_name staphb/fastqc:latest sh -c "fastqc /data/Raw_Data/$sample"
```

El contenedor se ejecuta en segundo plano, permitiendo así capturar las estadísticas y almacenarlas en un .txt con el nombre del contenedor. Además, se lanza utilizando la Opción explicada previamente en el punto 4.5.2, manteniendo vivo el contenedor tras su ejecución.
Los archivos de salida obtenidos en este caso son un .html y un .zip, con los resultados del análisis. Ambos se guardan en la carpeta Analysis, y se comprueba que se han movido correctamente.

5.2.3. fastp.sh

Funciona de forma similar a *FastQC.sh*. Este script se encarga de hacer un preprocesado y análisis de calidad de las muestras iniciales. Es una alternativa a FastQC. Comienza creando un fichero .txt llamado sample_names, útil para saber de un vistazo las muestras que hay y si son de un sólo extremo (Forward, R1 o Reverse, R2) o de extremos emparejados (R1 y R2). Una vez creado el fichero, se recorre la carpeta Raw_Data en busca de archivos con formato .fastq.gz para analizar. Al igual que antes, si no se encuentran archivos de este tipo se termina el proceso. Si se encuentran, lo primero es crear variables con los nombres adecuados para poder posteriormente llamar al contenedor de Docker. Aquí es cuando tiene importancia el .txt creado previamente. Si la muestra es de extremos emparejados, el script automáticamente lanza el contenedor con la opción "Paired Ends". Si la muestra sólo tiene un extremo, lanza el contenedor correspondiente al extremo (R1 o R2). De esta manera, se permite el análisis en todos los casos. Para nuestro trabajo sólo se han utilizado muestras con extremos emparejados. Al igual que para FastQC, se lanza el contenedor en segundo plano, permitiendo la captura de estadísticas, utilizando el siguiente comando:

```
docker run -d -v "$directory/:/data" --name $container_name quay.io/biocontainers/fastp:0.24.0--heae3180_1 sh -c "cd data/Raw_Data; fastp -i $input1 -I $input2 -o $output1 -O $output2 -q 20 --json $outputJSON --html $outputHTML"
```

En este caso se ha añadido una opción en el análisis de Fastp, *-q20*, un filtro de calidad. Establece el valor de corte que indica si una base es de calidad o no. Por defecto es 15. Nosotros lo hemos aumentado a una calidad phred mínima mayor de 20, buscando así resultados de mayor calidad.

Los archivos de salida en este caso son los siguientes: un .fastq.gz, que es el archivo preprocesado del que se han eliminado las lecturas de baja calidad, y que se utilizará posteriormente para el análisis de MTBseq, Snippy y TBProfiler). Además, un .json y un .html que

contienen un reporte con el resumen del análisis. Es necesario que el archivo de salida .fastq.gz se renombre de forma correcta para que el formato coincida con el requerido posteriormente. Por último, se comprueba que todos los ficheros hayan sido correctamente movidos a la carpeta Analysis.

5.2.4. Select_option.sh

Una vez acabado el análisis de calidad y preprocesado de las muestras, el usuario puede elegir la herramienta que quiere utilizar para el análisis genómico. Este script contiene un menú de opciones que permite seleccionar una o varias herramientas, entre las opciones MTBseq, Snippy y TBProfiler. Para ello, es necesario seguir el formato de la llamada al menú, incluyendo los siguientes tres parámetros: ./select_option.sh -OPCIÓN /directorio. Si alguno de los parámetros introducidos en la llamada contiene errores (por ejemplo, se ha introducido un directorio que no existe, o la opción es incorrecta, o simplemente falta algo de lo que se necesita para que la llamada sea completa), se vuelve al menú. Una vez elegida la opción (es posible seleccionar una, dos o las tres), automáticamente se llama al script/scripts que corresponden a esa opción. Al finalizar, se pregunta si se quiere ejecutar alguna otra herramienta (siempre que no se haya seleccionado la opción de todas las herramientas), en cuyo caso se vuelve al menú de opciones. Si no, el análisis genómico finaliza.

5.2.5. MTBseq.sh

Como se ha explicado previamente, MTBseq es una herramienta automatizada para el análisis de datos WGS de aislamientos de *M. tuberculosis*. Para este trabajo, se ha decidido ir implementando los pasos uno por uno, para un mayor control de los posibles errores que surgieran. El script comienza comprobando que los nombres de todos los archivos de entrada (archivos .f(ast)q.gz) de las muestras (guardados en la carpeta "Analysis") presenten el formato específico requerido por MTBseq. Si alguna de las muestras no lo tiene se detecta y se para el proceso para que el usuario lo pueda subsanar. Si todas las muestras tienen el formato requerido, se crea el archivo samples.txt a partir del fichero sample_names.txt generado en *fastp.sh*. Este archivo es necesario para los pasos TBvariants, TBjoin, TBamend y TBgroups, y contiene el ID de la muestra y el ID de la librería. Sólo las muestras que presenten ambas direcciones (R1 y R2) estarán incluidas en este fichero y se utilizarán para el análisis. Después, se comienza con las llamadas a Docker. La estructura de todas las llamadas es la misma, ejecutando los contenedores en segundo plano para capturar las estadísticas y sin eliminar los contenedores tras la ejecución. Las diferencias radican en el paso que se añade como opción, y en las opciones específicas de cada paso que han sido mencionadas previamente en el punto 4.4.1 A continuación, se van explicando todas las llamadas y las opciones especificadas en cada una de ellas:

TBbwa, TBrefine, TBpile y TBlist: sólo se establece el número de hilos a 16.

```
docker run -d -v "/$directory:/data" --name $container_name quay.io/biocontainers/mtbseq:1.1.0--hdfd78af_0 sh -c "cd data/Analysis ; MTBseq --step TBbwa --threads 16"

docker run -d -v "/$directory:/data" --name $container_name quay.io/biocontainers/mtbseq:1.1.0--hdfd78af_0 sh -c "cd data/Analysis/ ; MTBseq --step TBrefine --threads 16"

docker run -d -v "/$directory:/data" --name $container_name quay.io/biocontainers/mtbseq:1.1.0--hdfd78af_0 sh -c "cd data/Analysis ; MTBseq --step TBpile --threads 16"

docker run -d -v "/$directory:/data" --name $container_name quay.io/biocontainers/mtbseq:1.1.0--hdfd78af_0 sh -c "cd data/Analysis/ ; MTBseq --step TBlist --threads 16"
```

TBvariants: se establece el número de hilos a 16, además de introducir el fichero .txt necesario para que se lean las muestras. También se incluyen las opciones mincovf, mincovr, minphred20 y minfreq por defecto (4, 4, 4 y 75 respectivamente).

```
docker run -d -v "/$directory:/data" --name $container_name quay.io/biocontainers/mtbseq:1.1.0--hdfd78af_0 sh -c "cd data/Analysis ; MTBseq --step TBvariants --samples samples.txt --mincovf 4 -- mincovr 4 --minphred20 4 --minfreq 75 --threads 16"
```

TBstats: sólo se establece el número de hilos a 16.

```
docker run -d -v "/$directory:/data" --name $container_name quay.io/biocontainers/mtbseq:1.1.0--hdfd78af_0 sh -c "cd data/Analysis/ ; MTBseq --step TBstats --threads 16"
```

Después de este paso, se da la opción al usuario a filtrar el archivo de estadísticas obtenido, utilizando un determinado valor para la cobertura media y mediana. En caso de que el usuario quiera filtrar, se llama a un script de Python que realiza el filtrado (*filterTab.py*), generando un archivo de estadísticas que se guarda en una carpeta nueva (Filtered_Statistics), junto con un nuevo .txt, filtered_samples.txt, que contiene los valores del filtrado y las muestras que han superado el filtro. Si no, se salta al siguiente paso.

TBstrains: se establece el número de hilos a 16, además de los valores por defecto de mincovf, mincovr, minphred20 y minfreq.

```
docker run -d -v "/$directory:/data" --name $container_name quay.io/biocontainers/mtbseq:1.1.0--hdfd78af_0 sh -c "cd data/Analysis/ ; MTBseq --step TBstrains --mincovf 4 -- mincovr 4 --minphred20 4 --minfreq 75 --threads 16"
```

TBjoin y TBamend: se establece el número de hilos a 16, y los valores por defecto de mincovf, mincovr, minphred20 y minfreq. En este paso es también necesario indicas el fichero samples.txt, además de un nombre de proyecto si se quiere (en este caso GenomicAnalysis).

```
docker run -d -v "/$directory:/data" --name $container_name quay.io/biocontainers/mtbseq:1.1.0--hdfd78af_0 sh -c "cd data/Analysis ; MTBseq --step TBjoin --samples samples.txt --project GenomicAnalysis --mincovf 4 -- mincovr 4 --minphred20 4 --minfreq 75 --threads 16"
```

```
docker run -d -v "/$directory:/data" --name $container_name quay.io/biocontainers/mtbseq:1.1.0--hdfd78af_0 sh -c "cd data/Analysis/ ; MTBseq --step TBamend --samples samples.txt --project GenomicAnalysis --mincovf 4 -- mincovr 4 --minphred20 4 --minfreq 75 --threads 16"
```

TBgroups: se establece el número de hilos a 16, además de indicar el fichero samples.txt, el nombre del proyecto y la distancia para que dos muestras se consideren del mismo grupo, en este caso 15.

```
docker run -d -v "/$directory:/data" --name $container_name quay.io/biocontainers/mtbseq:1.1.0--hdfd78af_0 sh -c "cd data/Analysis/ ; MTBseq --step TBgroups --samples samples.txt --project GenomicAnalysis --distance 15 --threads 16"
```

Para todas las llamadas que requieren fichero samples.txt y el nombre del proyecto, ambos deben coincidir para que no se produzcan errores de lectura o incongruencias en las llamadas y búsqueda de archivos. El número de hilos puede ser variable en cada llamada, pero se ha decidido unificarlo para poder generar unas estadísticas más homogéneas. Tras la llamada de cada contenedor, se llamada al script *capture_stats.sh*, que permite capturar las estadísticas mientras el contenedor se está ejecutando.

Al ser un pipeline, en prácticamente todos los pasos es necesario haber realizado el anterior, ya que los ficheros de entrada de un paso suelen ser los ficheros de salida del anterior. De la misma manera, si un paso se ha realizado previamente, la propia herramienta lo detecta de manera automática (ya que existen los archivos de salida de ese paso), y salta el análisis para esa muestra. Todos los archivos generados en cada paso se almacenan en una carpeta específica del paso, como se ha mencionado en el punto 4.4.1.

5.2.6. Snippy.sh

Snippy es otra de las herramientas utilizadas para el análisis genómico de las muestras. Al igual que MTBseq, utiliza como archivos de entrada los archivos de salida de Fastp. Para aquellos que presenten ambas direcciones (se comprueba en el fichero sample_names.txt generado en Fastp), se hace la llamada al Docker. El análisis requiere una referencia, ya sea anotada o no. En nuestro caso, se ha utilizado la misma referencia que utiliza MTBseq. La referencia utilizada es

"M._tuberculosis_H37Rv.gbk", obtenida del NIH, y coincide con la de MTBseq presentando el mismo número de pares de bases (4411532), intentando asemejar lo máximo posible los resultados de ambas herramientas (51). Una vez guardada la referencia en una variable y copiada al directorio de trabajo, se realiza la llamada al Docker, también en segundo plano.

```
docker run -d -v "$directory/:/data" --name $container_name quay.io/biocontainers/snippy:4.6.0--hdfd78af_5 sh -c "cd data/Analysis snippy --mincov 4 --minfrac 0.75 --minqual 20 --cpus 16 --outdir mysnps_$sample --report --ref refe.gbk --R1 $input1 --R2 $input2"
```

En este caso, se establecen los valores de mincov, minfrac y minqual a 4, 0,75 y 20 respectivamente, además del número de hilos a 16. También se establece el nombre de la carpeta de salida, se pide que se genere un informe del análisis, se proporciona la referencia y los dos archivos de entrada (que son los archivos de salida .fastq.gz generados por Fastp). Para cada muestra, se genera una carpeta llamada mysnps_nombreMuestra, que contiene todos los resultados del análisis, entre los que se encuentran un .csv, un .vcf o un .gff, además del snps.report.txt y los ficheros FASTA con los alineamientos. Si alguna de las carpetas de salida ya existe, el análisis de esa muestra no se realiza.

Con el objetivo de obtener la matriz de distancias que se obtiene en MTBseq, se han implementado también Snippy-multi y snp-dists. El primero genera determinados archivos de alineamientos que la salida de Snippy no obtiene, y que son necesarios para después ejecutar snp-dists. Primero es necesario crear una tabla de entrada (input.tab), que contenga el identificador de la muestra y las rutas completas a los ficheros de ambas direcciones, R1 y R2. Para ello se recorre el fichero sample_names.txt generado en Fastp. Una vez hecho esto, es posible hacer la llamada del contenedor de Snippy-multi, utilizando las mismas opciones que para Snippy, manteniendo así la consistencia entre ambas. Por lo tanto, también se establece el número de hilos a 16 y se dan los mismos valores a mincov, minfrac y minqual. Se establece además que, una vez ejecutado este primer comando, se ejecute el script que se genera (*run_snippy.sh*):

```
docker run -v "$directory:/data" --name $container_name quay.io/biocontainers/snippy:4.6.0--hdfd78af_5 sh -c "cd data/Analysis&& snippy-multi input.tab --ref refe.gbk --mincov 4 --minfrac 0.75 --minqual 20 --cpus 16 > run_snippy.sh && bash run_snippy.sh"
```

Disponiendo de los archivos para ejecutar snp-dists, el último paso para obtener la matriz de distancias es hacer la llamada de Docker correspondiente:

```
docker run -v "$directory:/data" --name $container_name staphb/snp-dists:latest sh -c 'cd /data/Analysis&& snp-dists -b core.full.aln > snp-distances_snippy.tsv'
```

En este caso, simplemente se establece el nombre del archivo de salida (snp-distances_snippy.tsv) y la opción -b, que omite el nombre de la herramienta y la versión en el fichero de salida.

5.2.7. TBProfiler.sh

TBProfiler es la tercera herramienta utilizada para el análisis genómico. Como para los últimos dos scripts, se obtienen los nombres de las muestras del fichero sample_names.txt, ya que los archivos de entrada son los archivos de salida de Fastp. Para cada muestra con ambas direcciones R1 y R2 comprueba que no exista previamente el archivo de salida, ya que eso indicaría que ya se ha analizado y esa muestra se saltaría. Si no existe, se hace la llamada de Docker en segundo plano con el siguiente comando:

```
docker run –d -v "$directory/:/data" --name $container_name quay.io/staphb/tbprofiler:latest sh -c "cd Analysis; tb-profiler profile --call_whole_genome --snp_dist 15 --spoligotype -1 $input1 -2 $input2 -p $output -d tbprofiler_results_$sample --csv -t 16"
```

En este caso, se establece la opción --call_whole_genome, permitiendo realizar el llamado de variantes en un genoma completo, además de la distancia máxima entre SNPs a 15 y la opción de la espoligotipificación experimental. Al igual que para Snippy, esta herramienta también genera una carpeta por muestra donde almacena los resultados, llamada en este caso tbprofiler_results_nombreMuestra. Por último, se añade la opción --csv, que genera un .csv adicional con los resultados de análisis.

5.2.8. FilterTab.py

Este script en Python permite filtrar el archivo .tab generado en el paso TBStats, introduciendo por pantalla los valores de cobertura media y mediana que se quieren establecer. Para ello, primeramente, se definen las variables necesarias para el filtrado del archivo .tab, y después se guarda del archivo .tab filtrado y se crea un nuevo fichero .txt que contenga sólo las muestras que han pasado el filtrado (filtered_samples.txt). Estos dos últimos archivos se guardan en la carpeta Filtered_Statistics.

5.2.9. Capture_stats.sh

Este script se centra en la captura de estadísticas. Gracias al uso de Docker stats, es posible analizar los recursos empleados y los costes de cada herramienta. El script genera para cada contenedor lanzado un archivo .txt que almacena los datos de porcentaje de CPU (CPU %) utilizado, memoria utilizada con respecto al límite (MEM USAGE/LIMIT), porcentaje de memoria utilizado (MEM %) y tiempo. Esta última columna se obtiene gracias a la función date disponible en el propio ordenador. Buscando una mayor comprensión, se ha establecido una variable llamada sampleStep, que permite diferenciar la muestra (en el caso de FastQC, Fastp, Snippy y TBProfiler) o el paso en el que se encuentra (para MTBseq y Snippy-multi, snp-dists).

Para ello, los contenedores se lanzan en segundo plano, y mientas están ejecutándose, se lanza el comando docker stats. Se filtra por nombre de contenedor, evitando posibles interferencias con otros contenedores que estén ejecutándose a la vez, y se recogen los datos establecidos en el formato de la llamada. La opción --no-stream permite que la llamada a Docker stats se haga de forma única, ya que, sin esta opción, se muestras estadísticas en vivo que se actualizan de forma constante (necesitando pulsa Ctrl+C para salir de la pantalla que se abre).

Como la llamada a docker stats está dentro de un bucle que no termina hasta que el contenedor deja de ejecutarse, podemos simular las estadísticas en vivo, aunque esté la opción --no-stream. Tras cada llamada a docker stats, se establece un tiempo de parada de 5 segundos, lo que permite unificar las estadísticas y que cada línea del fichero .txt se genere en un intervalo de tiempo constante. Si no, la siguiente línea se ejecuta en cuanto la anterior termina, lo que puede resultar en ficheros .txt demasiado largos. Entre docker stats y sleep, se ha añadido también el comando docker logs, que permite redirigir la salida del contenedor a un .log. De esta manera, es posible almacenar la salida del contenedor, aunque se esté ejecutando en segundo plano. Tanto docker stats como docker logs están dentro de una condición if, que sólo los lanza en el caso de que el contenedor exista.

Por último, al finalizar cada llamada al contenedor, se muestra el tiempo inicial y final, además del tiempo total que ha requerido. Una vez hecho todo esto, el contenedor se elimina utilizando el comando rm, asegurando así que el contenedor no se mantiene después de ejecutarse.

De esta manera, es posible hacer un análisis sencillo de las estadísticas principales, y poder comparar las distintas herramientas en función de los recursos utilizados y del tiempo que se ha empleado.

5.2.10. Creación de archivos .log

Como se ha explicado previamente, se generará un archivo .log para el control de errores. En este caso, tendrán un fichero .log las dos herramientas de análisis de calidad (FastQC y Fastp), además de las tres de análisis genómico (MTBseq, Snippy y TBProfiler). Aunque MTBseq ya crea de manera automática un .log, se ha querido crear uno adicional por tener también los mensajes añadidos del script creado para el proyecto. Para ello, se han añadido al principio de cada uno de estos scripts las líneas que definen el archivo .log y que redirigen toda la salida al mismo:

```
log_file="$directory/Analysis/FastQC_$(date '+%Y-%m-%d')_.log"
exec > >(tee -a "$log_file") 2>&1
```

Lo único que cambia de un script a otro es el nombre del .log. Al final de cada script, como hay algunos que están encadenados, se para la redirección al .log, evitando así que se mezclen los .log de diferentes scripts. Esto se consigue mediante la línea:

```
exec > /dev/tty 2>&1
```

5.3. Flujo de trabajo

La Figura 3 muestra el flujo de trabajo explicado previamente.

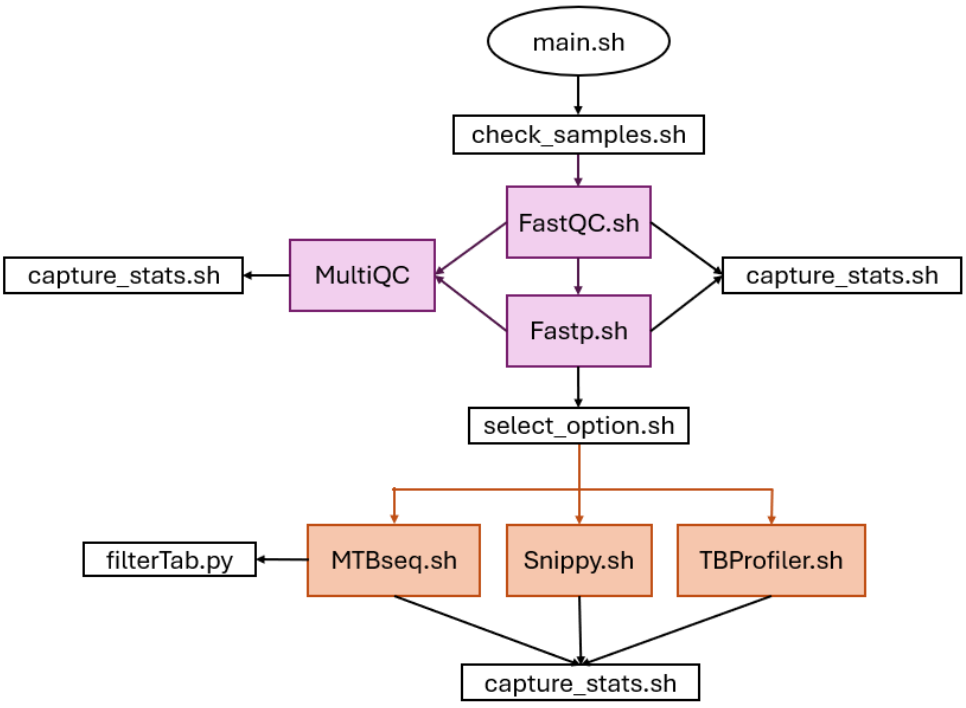

Figura 3. Flujo de trabajo de scripts seguido en el proyecto. Óvalo blanco: script inicial; rectángulos blancos: scripts complementarios a los de las herramientas principales; rectángulos violetas: scripts relacionados con las herramientas pertenecientes a la etapa de análisis de calidad y preprocesado; rectángulos naranjas: scripts relacionados con las herramientas pertenecientes a la etapa de análisis genómico

6. ESTUDIO ECONÓMICO

El objetivo del estudio económico es valorar los costes asociados a la realización del proyecto de forma general, haciendo una estimación realista de lo que supondría llevarlo a cabo en un contexto profesional. Se parte de la base de que es un proyecto que ha sido realizado en un ámbito universitario con los recursos disponibles para ello.

6.1. Costes materiales

El proyecto se ha realizado utilizando un equipo informático personal, además del servidor proporcionado por la universidad. Ambos presentan especificaciones técnicas adecuadas para el tratamiento de datos de secuenciación y la ejecución de herramientas bioinformáticas. El equipo personal dispone de un procesador de gama media-alta (Intel Core i5), 8 GB de RAM y un disco SSD de 512 GB. En caso de tener que adquirir un equipo de estas características para el proyecto, el coste aproximado sería de unos 600€. Sin embargo, al tratarse de un dispositivo ya disponible, no requiere un gasto adicional. El servidor también es un coste a tener en cuenta. En este caso se disponía de forma gratuita del servidor de la universidad (512GB de almacenamiento, 16GB de RAM), por lo que este costo ha sido 0€. Sin embargo, puede darse la situación en la que se tuviese que alquilar un servidor. Cada análisis completo requiere alrededor de 2 horas, por lo que se estiman unas 30 horas totales a lo largo del proyecto, teniendo en cuenta pruebas, modificaciones y optimizaciones del proceso que se hayan llevado a cabo a lo largo del mismo. El coste de infraestructura de computación de altas prestaciones para un análisis de este tipo en el IACS sería de 0,41€/hora/CPU. Considerando unas 30 horas, ascendería a 12,3€ (52).

6.2. Herramientas utilizadas

Se parte de muestras previamente analizadas por el IACS, lo que no aporta un coste añadido al proyecto ya que se ha realizado en colaboración. Sin embargo, si hubiera que analizar las muestras, habría que sumar el coste correspondiente al análisis de NGS, aproximadamente de unos 4000€. Este precio incluiría el análisis cuantitativo y cualitativo de ADN, la preparación de librerías para la PCR, el material fungible necesario para la secuenciación y el proceso de secuenciación en sí. El proyecto se ha apoyado únicamente en software libre y de código abierto. Se han empleado herramientas como FastQC, Fastp, MultiQC, MTBseq, Snippy o TBProfiler, todas ellas disponibles de forma gratuita y sin necesidad de licencias. Además, se ha trabajado utilizando Docker, lo que simplifica considerablemente el proceso. Esto es debido a que sólo es necesario descargar las imágenes de las herramientas y no las herramientas como tal, lo que minimiza errores de configuración y aumenta la reproducibilidad del trabajo en distintos entornos y sistemas operativos.

6.3. Externalización del análisis (opcional)

En el caso de tener que externalizar el análisis completo, éste se podría realizar en el IACS, utilizando los servicios científico-técnicos de la Unidad de Biocomputación del Centro de Investigación Biomédica (CIBA). Un análisis de estas características, considerado dentro de la tarifa Análisis Avanzado bioestadístico/bioinformático, considerando alrededor de 30 horas de trabajo a 55,54€/hora, supondría un coste de 1666,2€ (52).

6.4. Almacenamiento

Se han analizado un total de ocho muestras (cada una con dos archivos FASTQ obtenidos por NGS). Cada muestra ocupa entre 500MB y 1GB, incluyendo datos intermedios y resultados, lo que supone un total aproximado de entre 4 y 8 GB. Además, hay que tener en cuenta el tamaño de descarga de las imágenes de Docker, que asciende a alrededor de 6,7GB, como se puede ver en la Tabla 2. La suma de ambos hace un total de alrededor 13GB. Estos datos se han gestionado en el servidor, a excepción de aquellos que han sido descargados y tratados de forma local para obtener los gráficos de los resultados. Considerando que el precio por terabyte en discos duros está en un rango entre 50 y 80€, el coste proporcional del almacenamiento requerido sería alrededor de 10€.

6.5. Coste de personal

El proyecto ha requerido una dedicación aproximada de 160 horas de un sólo bioinformático, incluyendo la revisión bibliográfica, el diseño del flujo de trabajo, la ejecución del análisis, el tratamiento de datos para obtener resultados y gráficos, y la redacción de la memoria. El trabajo ha sido realizado por la autora del proyecto sin compensación económica y dentro del marco de sus estudios. En el caso de tener que estimar un coste orientativo de lo que supondría si se realizase por una persona contratada, se podría calcular de la siguiente manera: suponiendo un salario bruto de 25€/hora, el salario bruto mensual ascendería a 4000€. Si se considera también el coste para la empresa, incluyendo cargas sociales (aproximadamente un 36%), el coste final se situaría en 5440€ mensuales. Asumiendo un salario retribuido en 12 mensualidades, el salario total anual bruto ascendería a 48000€ y el coste de la empresa anual a 65280€.

6.6. Resumen de costes

La siguiente tabla muestra el desglose de lo que costaría el proyecto en el caso de tener en cuenta todos los gastos añadidos que se han ido mencionando, frente a los gastos reales de este proyecto.

MOTIVO DE GASTO	**DESGLOSE**	**COSTE TOTAL**
Costes materiales	Equipo informático (opcional) + Servidor	600 € + 12,3€
Herramientas utilizadas	Licencias de software + NGS (opcional)	0 € + 4000€
Externalización del análisis (opcional)	Análisis bioestadístico/bioinformático completo	1666,2€
Almacenamiento	Muestras iniciales y resultados	10 €
Coste de personal	Coste de empresa	5440€/mes
		11728,50€

Tabla 3. Desglose de los costes del proyecto

El desarrollo del proyecto ha sido económicamente viable gracias al uso de herramientas gratuitas, el aprovechamiento de recursos propios y la ejecución en entornos disponibles para el mismo. Esto ha permitido evitar inversiones en infraestructura o licencias, lo cual hubiera encarecido el proyecto. Al partir de muestras ya analizadas, se elimina el coste de realizar una secuenciación. Esto, junto con no tener la necesidad de externalizar el proceso, permite que el coste total del proyecto sea asequible en un escenario real.

En un contexto profesional, el principal coste estaría determinado por el coste de personal, siendo menos relevante el resto de los costes. Si además el análisis se tuviese que externalizar o se necesitase hacer la secuenciación de las muestras, los costes subirían aún más, llegando a lo que se puede ver en la Tabla 3.

Por todo ello, el enfoque que se ha planteado demuestra que es posible realizar análisis bioinformáticos completos y rigurosos, sin necesidad de una gran inversión económica.

7. RESULTADOS Y DISCUSIÓN

Al aplicar la metodología explicada, se ha conseguido automatizar el flujo del análisis, permitiendo realizar de forma secuencial todos y cada uno de los pasos a partir de la detección de muestras en una carpeta específica. Además, el usuario puede interactuar con el código en varias partes, permitiendo personalizar aún más el análisis. Una vez ejecutado todo el pipeline, los resultados obtenidos han sido los siguientes:

7.1. Análisis de calidad y preprocesado de las muestras

Los informes obtenidos en FastQC y Fastp han sido combinados en un sólo informe utilizando MultiQC. De esta manera, es sencillo ver si las muestras presentan o no una calidad adecuada.

MUESTRA	LECTURAS ANTES DEL FILTRADO	LECTURAS DESPUÉS DEL FILTRADO	PORCENTAJE Q30 ANTES DEL FILTRADO	PORCENTAJE Q30 DESPUÉS DEL FILTRADO	LECTURAS DE BAJA CALIDAD (<Q20)
HCU24001_S1	1978218	1744158	68,55%	73,17%	234060
HCU24002_S2	1562746	994656	62,18%	76,31%	568090
HCU24010_S6	2253624	1988222	70,44%	75,71%	265402
HCU24011_S7	556414	510314	67,05%	70,79%	46100
HMS24017_S15	1042378	965440	77,36%	83,58%	76938
HMS24046_S23	2202100	1817738	68,11%	74,73%	384362
HMS24051_S27	1540640	1431330	68,68%	71,95%	109310
HMS24052_S28	3364164	2905730	71,18%	77,29%	458434

Tabla 4. Resultados del preprocesado de muestras realizado por Fastp. Incluye muestra, lecturas antes y después del filtrado, porcentaje q30 antes y después del filtrado, lecturas descartadas por baja calidad (<q20).

La Tabla 4 Presenta los resultados obtenidos en el preprocesado de las muestras. Se puede observar que todas las ellas mejoran su porcentaje de calidad q30 tras el procesado. Además, ninguna de las muestras presenta una gran cantidad de lecturas de baja calidad, estando entre 76938 y 568090. Considerando que la mayoría de las muestras tienen alrededor de 1-2 millones de lecturas iniciales, no es un número muy elevado. En cuanto a la cobertura, la muestra HCU24011_S7 es la que presenta una menor cobertura, con una cantidad considerablemente menor de lecturas iniciales. Esta diferencia de lecturas puede ocasionar una disminución en la calidad global de todo el conjunto. Sin embargo, al ser un caso aislado, no se considera de gran relevancia.

En cuanto al reporte de MultiQC, en las figuras 4, 5, 6 y 7 se puede observar que las ocho muestran presentan buena calidad, disminuyendo la calidad de la secuencia a medida que se acerca a la parte final del gráfico. Esto es totalmente normal a medida que el proceso progresa, con lo que las muestras se consideran de buena calidad para seguir con el análisis.

Los reportes por muestra y el resto del reporte de MultiQC están en el repositorio de GitHub creado para este proyecto (50).

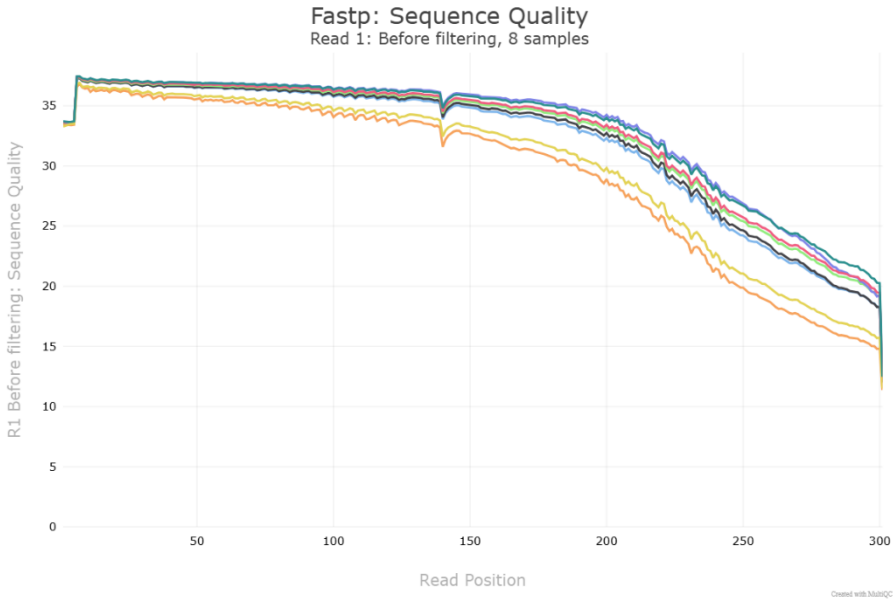

Figura 4. Gráfico de calidad de secuencia de la lectura 1 antes del filtrado.

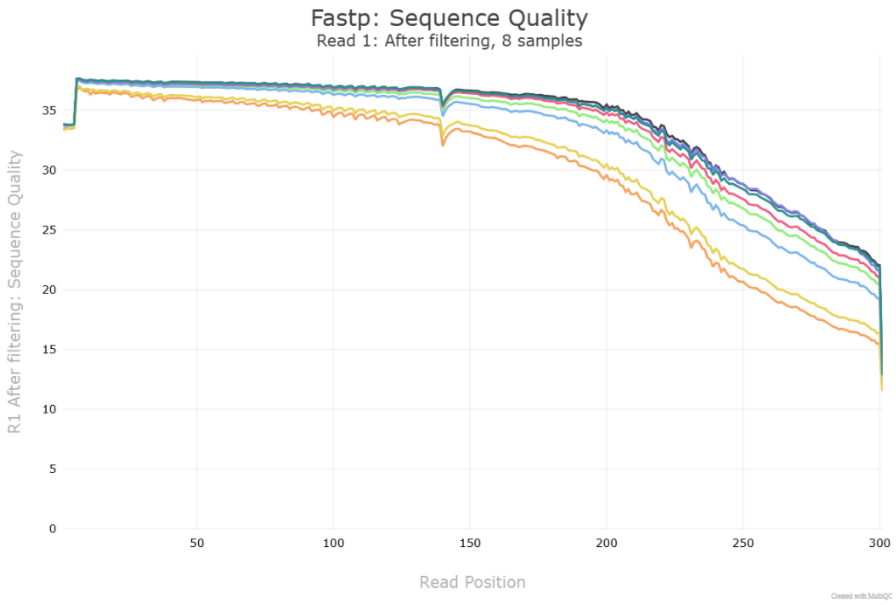

Figura 5. Gráfico de calidad de secuencia de la lectura 1 después del filtrado.

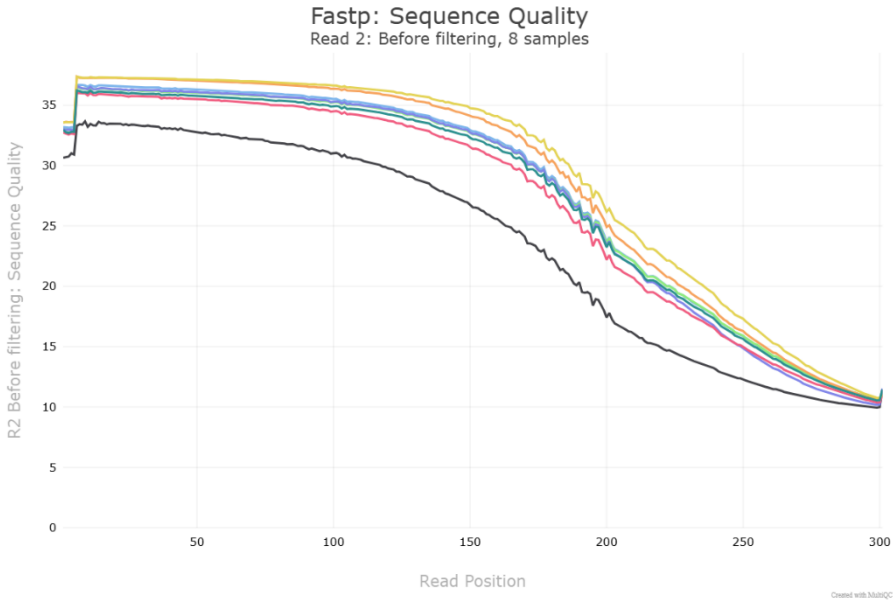

Figura 6. Gráfico de calidad de secuencia de la lectura 2 antes del filtrado.

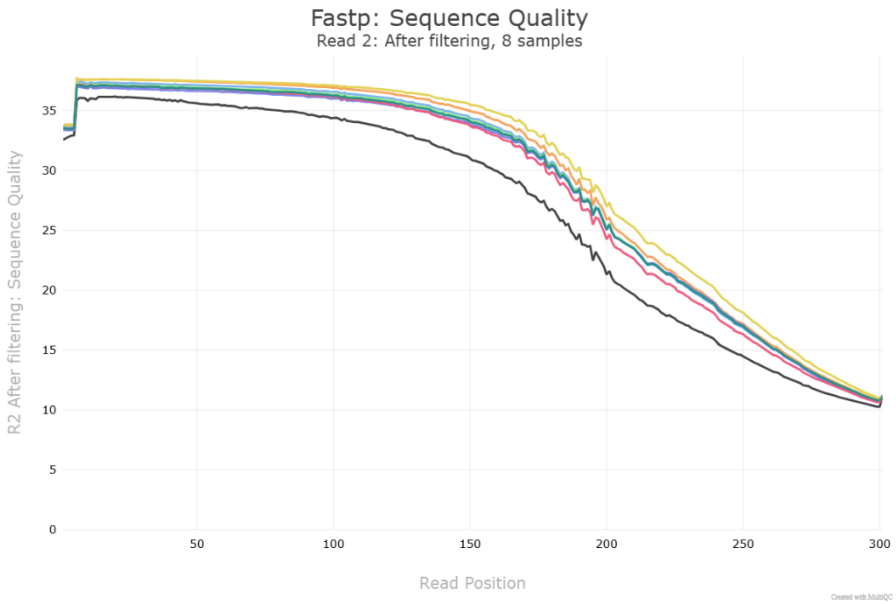

Figura 7. Gráfico de calidad de secuencia de la lectura 2 después del filtrado.

7.2. Clasificación de las muestras por linaje

Las ocho muestras analizadas pertenecen a tres especies bacterianas que forman parte del MTBC: *M. tuberculosis, M. africanum* y *M.bovis.* La distribución es la siguiente: 5 de *M. tuberculosis,* 2 de *M. africanum* y 1 de *M.bovis.*

La primera muestra pertenece al linaje Euro-americana, y corresponde a las muestras HCU24001_S1, HCU24010_S6, HMS24017_S15, HMS24046_S23 y HMS24051_S27. Las dos muestras *M.africanum* pertenecen a dos linajes distintos: West-Africa 2 para HCU24011_S7 y West-Africa 1 para HMS24052_S28. Por último, la muestra HCU24002_S2 es la única que pertenece a la especie *M.bovis*. Estos resultados se pueden observar en la Figura 8 y en la Tabla 5.

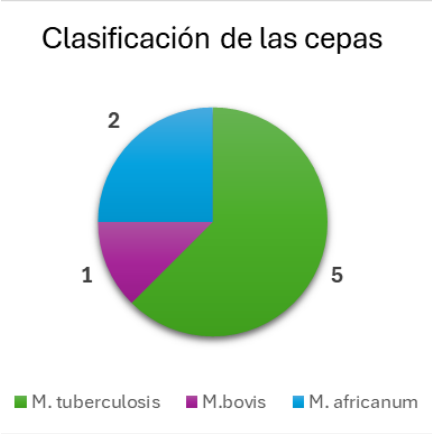

Figura 8. Distribución de cepas.

MUESTRA	LINAJE	ESPECIE
HCU24001_S1	Euro-American	*M. tuberculosis*
HCU24002_S2	*M.bovis*	*M.bovis*
HCU24010_S6	Euro-American	*M. tuberculosis*
HCU24011_S7	West-Africa 2	*M. africanum*
HMS24017_S15	Euro-American	*M. tuberculosis*
HMS24046_S23	Euro-American	*M. tuberculosis*
HMS24051_S27	Euro-American	*M. tuberculosis*
HMS24052_S28	West-Africa 1	M. africanum

Tabla 5. Clasificación de muestra, linaje y cepa.

Esta clasificación se obtiene tanto en TBprofiler como en MTBseq. La primera herramienta muestra la cepa de cada muestra, y quizás es algo menos intuitiva. En cambio, la segunda tiene un paso específico, TBstrains, en el que genera una tabla con la clasificación de las muestras analizadas, incluyendo además de la cepa (Homolka species), el linaje Homolka, el grupo, la calidad o el linaje Coll.

La diferencia entre los dos linajes mencionados radica en el sistema de clasificación genotípica utilizado para agrupar las cepas de *M. tuberculosis*. El linaje Homolka está basado en el esquema propuesto por Homolka et al. En 2012 y ofrece una mayor resolución filogenética dentro de los principales linajes. En cambio, el linaje Coll está basado en la clasificación propuesta por Coll et al. en 2014 y agrupa las cepas en linajes mayores con relevancia filogenética y epidemiológica. La clasificación en este caso es más agrupada (53, 54).

7.3. Detección de SNPs, inserciones y deleciones de las diferentes herramientas

Las tres herramientas son capaces de detectan SNPs, inserciones y deleciones. Además, Snippy y TBProfiler detectan MNPs y Snippy variaciones complejas (compuestas por inserciones y deleciones múltiples, u otro tipo de variaciones). Las diferencias en las detecciones de SNPs,

inserciones y dependen de la distancia que separa a la muestra del genoma de la cepa de referencia (en este caso H37Rv).

7.3.1. Detección de SNPs

Las tres herramientas detectan un número muy similar de SNPs en todas las muestras. Este número es ligeramente superior para TBProfiler en todas ellas. Las muestras que más SNPs presentan son HCU24002_S2 y HMS24052_S28, con un total de 2304 y 2248 para MTBseq, 2189 y 2100 para Snippy y 2540 y 2459 para TBProfiler respectivamente. En el otro extremo, la muestra que menos SNPs presenta es HCU24001_S1, con un total de 559 SNPs detectados por MTBseq, 520 por Snippy y 680 por TBProfiler.

Como se ha comentado previamente, TBProfiler realiza una búsqueda adicional de SNPs relacionados con resistencia a fármacos. Sin embargo, no se han encontrado SNPs de este tipo para ninguna de las ocho muestras.

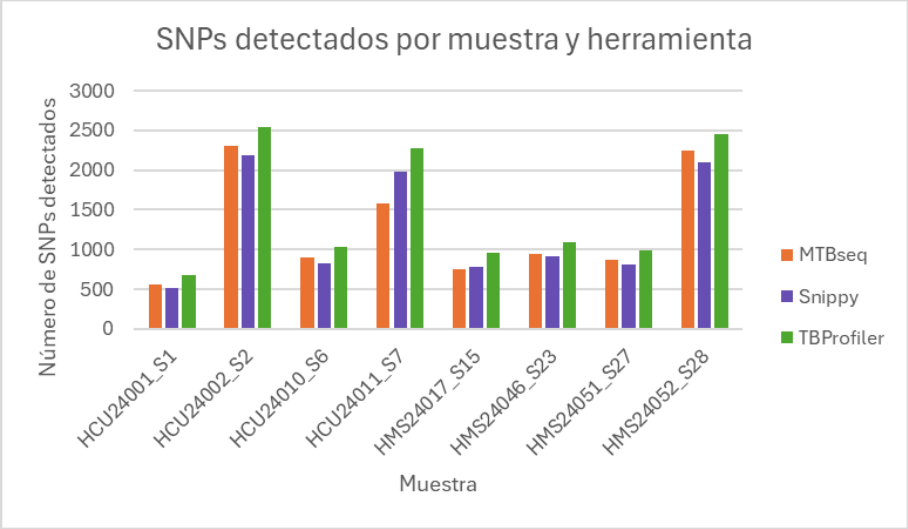

Figura 9. Gráfico de barras que muestra las diferencias entre las herramientas y muestras en la detección de SNPs.

7.3.2. Detección de inserciones

En el caso de las inserciones, hay una clara superioridad de detección por parte de MTBseq frente a las otras dos herramientas. HMS24052_S28 es la muestra en la que más se aprecia esta diferencia, donde MTBseq detecta 403 inserciones, frente a 104 y 119 detectados por Snippy y TBProfiler respectivamente.

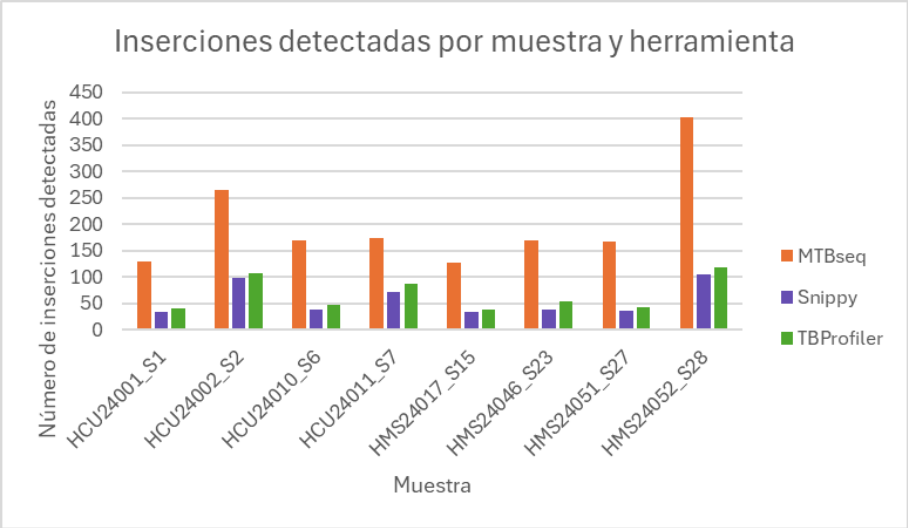

Figura 10. Gráfico de barras que muestra las diferencias entre las herramientas y muestras en la detección de inserciones.

7.3.3. Detección de deleciones

De la misma manera que para las inserciones, MTBseq es considerablemente superior en la detección de deleciones, si se compara con Snippy y TBProfiler. La muestra HMS24017_S15 es la única de las oxho que presenta números de detección similares en las tres herramientas, siendo 67 para MTBseq, 34 para Snippy y 38 para TBProfiler.

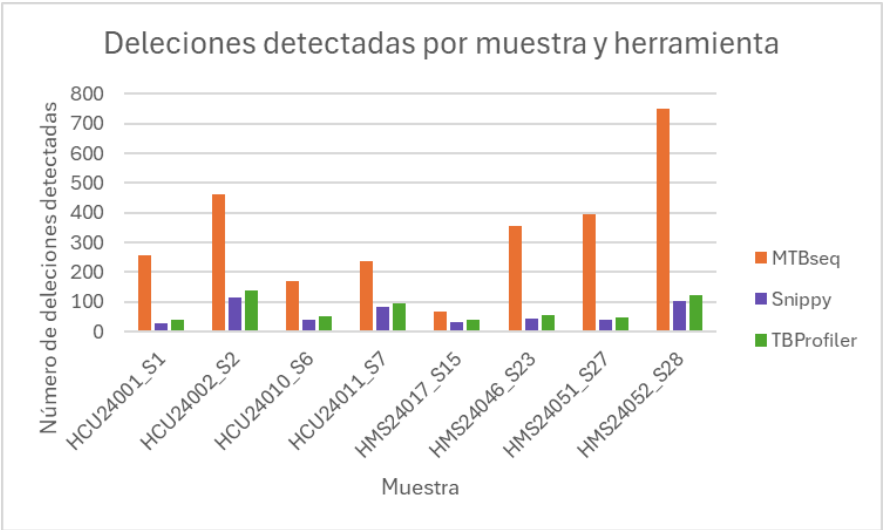

Figura 11. Gráfico de barras que muestra las diferencias entre las herramientas y muestras en la detección de inserciones.

Varios estudios son consistentes con la idea de que las diferencias en la detección de SNPs, e indels son debidas a las herramientas utilizadas para el llamado de variantes. En el caso de

MTBseq, se utiliza la herramienta GATK, mientras que para Snippy se utiliza FreeBayes y para TBProfiler GATK, Freebayes y BCFtools.

Un estudio llevado a cabo por Laurie et al. evaluó la robustez de herramientas de detección de variantes a través de una evaluación comparativa (benchmarking en inglés) de combinaciones de alineadores (BWA-MEM y GEM3) y herramientas de llamado de variantes (FreeBayes, GATK HaplotypeCaller y SAMtools). Los resultados obtenidos demostraron que para la detección de SNPs los resultados eran similares para todas las combinaciones. Sin embargo, para la detección de indels, la combinación con GATK fue ligeramente superior a las otras combinaciones (55).
Otro estudio llevado a cabo por Bo-Young Kim et al. estudió las diferencias entre varias herramientas de llamado de variantes, obteniendo una superioridad de indels al utilizar GATK frente a FreeBayes o la combinación de las dos anteriores (56).
Por último, un estudio llevado a cabo por Chao Tang et al. obtuvo conclusiones similares, siendo la detección de SNPs similar y algo superior para FreeBayes y BCFtools en comparación a GATK, pero la detección de indels considerablemente superior utilizando GATK frente a FReeBayes y BCFtools (57).

Internamente, las tres herramientas utilizan formas distintas de procesado y algoritmos para realizar la detección de variantes. GATK, como reconstruye posibles haplotipos en regiones candidatas, permite detectar indels con mayor sensibilidad, especialmente en regiones complejas. Por otra parte, FreeBayes no realiza el reensamblaje local que utiliza GATK, lo que puede provocar una pérdida de indels más pequeños o en regiones ambiguas. Por último, BCFtools utiliza un enfoque más rápido, pero menos sensible a indels, especialmente aquellos que están cerca de otros polimorfismos o en regiones repetitivas (36, 38, 39, 41).

7.3.4. Profundidad de la cobertura de detección

La profundidad de cobertura es muy útil, ya que es la medida del número de lecturas que detectan cada SNP. Para ello, se han obtenido las coberturas medias por muestra, pudiendo así comparar las tres herramientas en un diagrama de cajas y bigotes:

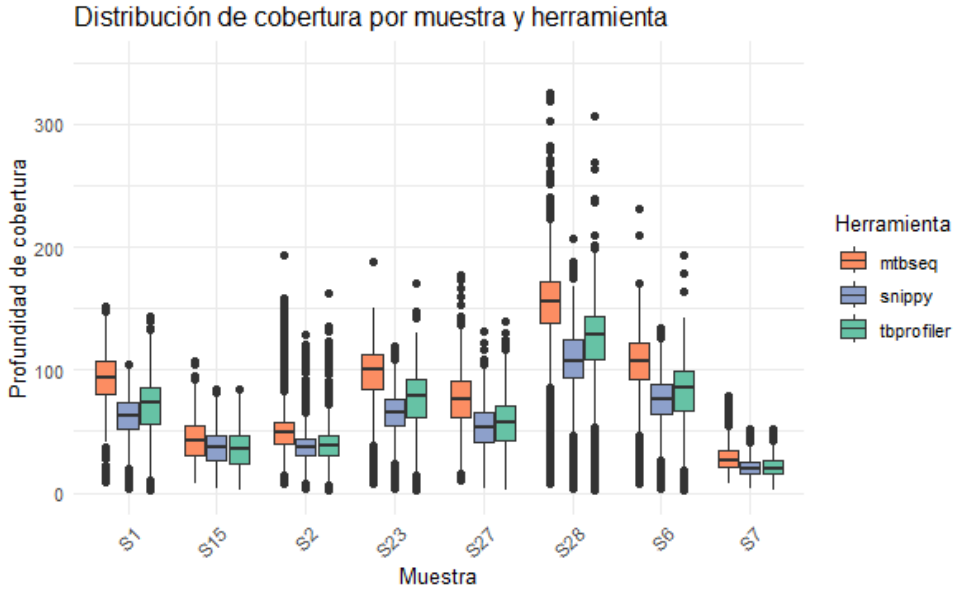

Figura 12. Gráfico de cajas y bigotes de la profundidad de cobertura por muestra y herramienta.

En el gráfico se puede apreciar la distribución de la profundidad de cobertura de las tres herramientas en las distintas muestras. Para todas ellas, MTBseq presenta una profundidad de cobertura mayor, lo que indica que hay más lecturas por SNP. Snippy y TBProfiler presentan una profundidad de cobertura similar, siendo mayor la de TBProfiler en algunas de las muestras, pero sin superar en ninguna de ellas a MTBseq. Como se ha comentado anteriormente, la muestra HCU24011_S7 es la que muestra una menor cobertura para las tres herramientas.

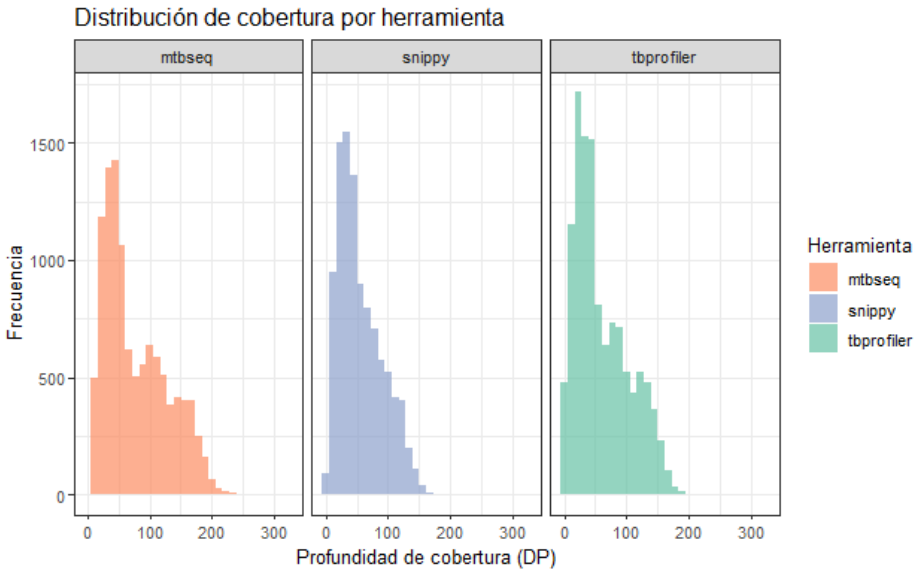

Figura 13. Distribución de cobertura por herramienta.

Como la distribución de cobertura por herramienta no es normal, lo que se puede apreciar en la Figura 13, se decide realizar un test de Kruskal Wallis en lugar de uno de ANOVA para ver si hay diferencias significativas entre la profundidad de cobertura de las herramientas. Kruskal Wallis es una prueba no paramétrica que no asume normalidad, y que compara las medianas entre grupos (58).

El p-valor obtenido en el test de Kruskal Wallis es $2{,}2e^{-16}$, lo que indica que hay diferencias significativas entre la profundidad de cobertura de las herramientas. Adicionalmente, se ha realizado un test de Wilcoxon por pares de muestras, que permite ver si hay diferencias significativas entre dos herramientas. Es también un test estadístico no paramétrico que contrasta si dos muestras proceden de poblaciones equidistribuidas (59). Los resultados muestras que existen diferencias significativas entre todas las herramientas entre sí, siendo mucho mayor la diferencia entre MTBseq y Snippy (p-valor $4{,}94e^{-207}$) y MTBseq y TBProfiler (p-valor $7{,}15e^{-166}$) que entre Snippy y TBProfiler ($2e^{-3}$).

7.3.5. Diagramas de Venn

Para realizar una comparación más detallada y profunda de la detección de SNPs, se han creado seis diagramas de Venn, uno por muestra, que plasman las diferencias entre los SNPs de genes anotados detectados por cada herramienta.

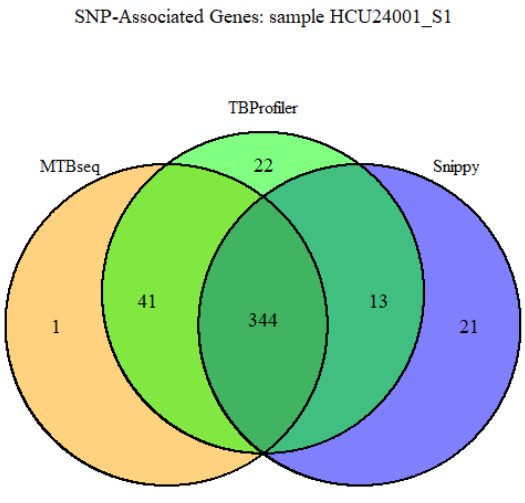

Figura 14. Diagrama de Venn HCU24001_S1.

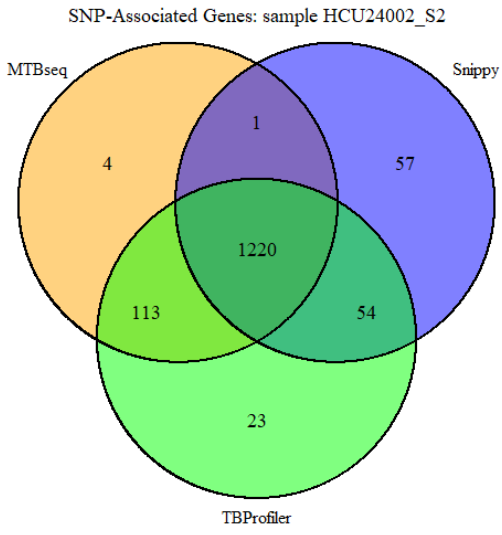

Figura 15. Diagrama de Venn HCU24002_S2.

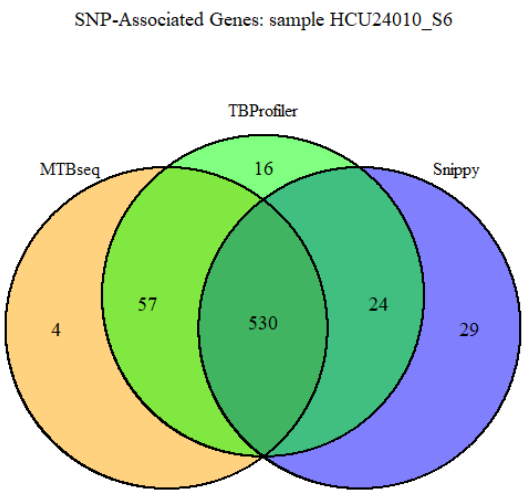

Figura 16. Diagrama de Venn HCU24010_S6.

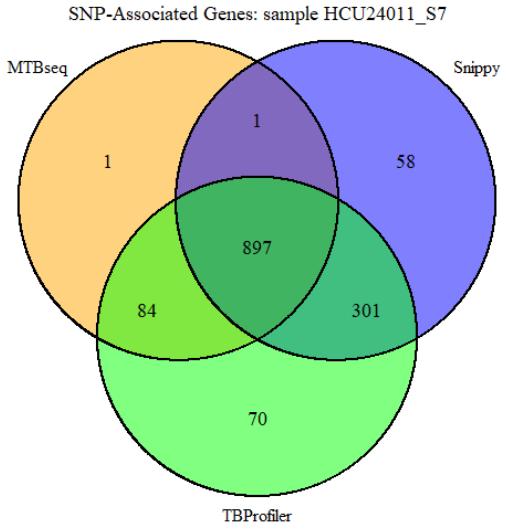

Figura 17. Diagrama de Venn HCU24011_S7.

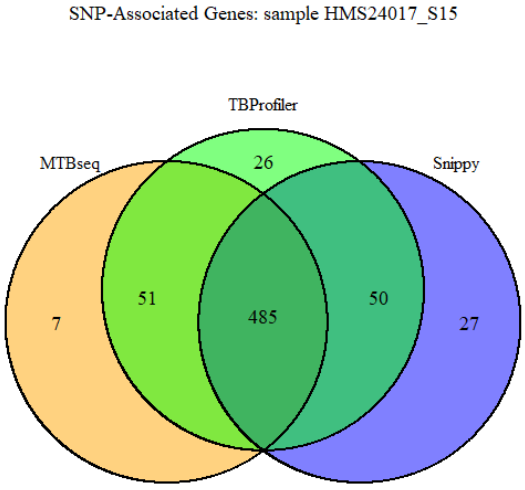

Figura 18. Diagrama de Venn HMS24017_S15.

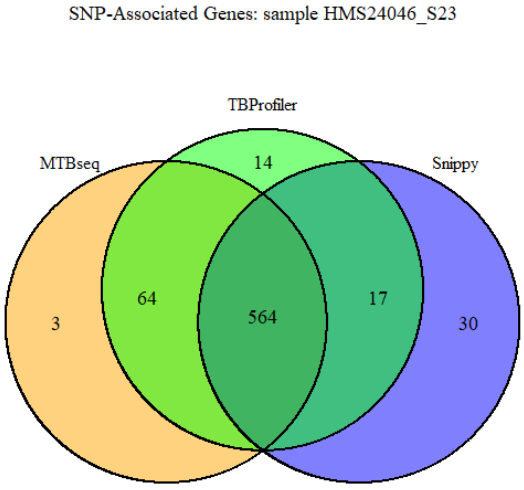

Figura 19. Diagrama de Venn HMS24046_S23.

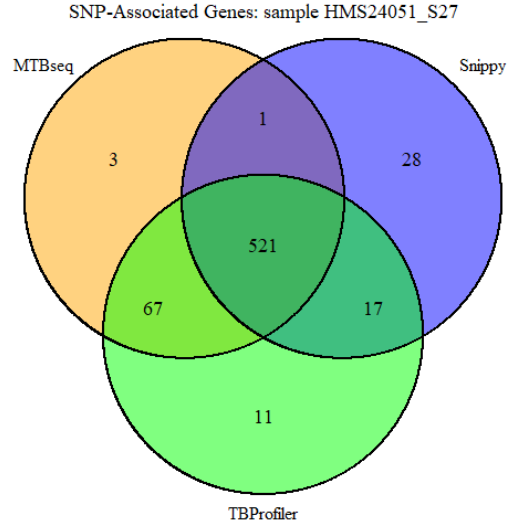

Figura 20. Diagrama de Venn HMS24051_S27.

Figura 21. Diagrama de Venn HMS24052_S28.

En estos diagramas se puede apreciar que la herramienta que más SNPs detecta es Snippy, seguida de TBProfiler y MTBseq. Esto puede ser debido a que la referencia utilizada por Snippy es anotada. Los resultados de TBProfiler han sido anotados de forma manual posteriormente al análisis, ya que éste no permite la anotación durante el propio proceso.

También se puede observar que la cantidad de SNPs detectados por muestra es considerablemente menor a los obtenidos en el punto 7.3.1. Esto está directamente relacionado con las anotaciones, ya que es mucho menor la cantidad de SNPs anotados que sin anotar que se detectan. Se han obtenido dos tipos de diagramas de Venn, dependiendo de la muestra:

El **primer tipo** sería como el de la muestra HCU24001_S1, y se puede interpretar de la siguiente manera: MTBseq detecta de forma única 1 SNP, Snippy 21 y TBProfiler 22. Además, 41 son detectados por MTBseq y TBProfiler, pero no por Snippy; 13 por TBProfiler y Snipp, pero no por MTBseq; y 344 por las 3 herramientas. En este caso, no hay SNPs detectados conjuntamente por Snippy MTBseq. Este primer tipo de diagrama se ha obtenido para las muestras HCU24001_S1, HCU24010_S6, HMS24017_S15 y HMS24046_S23, y plasmados en las figuras 14, 16, 18 y 19.

Por otra parte, se ha obtenido **otro tipo** de diagrama de Venn, similar al de la muestra HCU24002_S2. Se puede interpretar de la siguiente manera: MTBseq detecta de forma única 4 SNPs, Snippy 57 y TBProfiler 23. Además, 1 SNP es detectado de forma conjunta por MTBseq y Snippy, 54 por Snippy TBProfiler y 113 por TBProfiler y MTBseq. Las tres herramientas detectan de forma conjunta 1220 SNPs. Este sería el caso para las muestras HCU24002_S2, HCU24011_S7, HMS24051_S27 y HMS24052_S28, plasmadas en las figuras 15, 17, 20 y 21.

Las herramientas utilizan distintos mapeadores y alineadores en el llamado de variantes, lo que posiblemente causa las diferencias en los SNPs únicamente detectados por una herramienta.

Es bastante llamativo que MTBseq detecta de forma sistemática un número mucho menor de SNPs para todas las muestras, en comparación a las otras dos herramientas. Por ello, se ha hecho una búsqueda de los genes implicados en esos SNPs, presentes en la Tabla 6.

MUESTRA	NÚMERO DE SNPS ÚNICOS	GENES DE LOS SNPS ÚNICOS
HCU24001_S1	1	Rvnr02
HCU24002_S2	4	Rv1587c, Rv1765c, Rv2350c, Rv2386a
HCU24010_S6	4	Rv0797, Rvnr01, Rv1587c, Rv2386a
HCU24011_S7	1	Rv2386a
HMS24017_S15	7	Rv0797, Rv1041c, Rv1199c, Rv2386a, Rv2512c, Rv3023c, Rv3115
HMS24046_S23	3	Rv1873, Rv2386a, Rv2665
HMS24051_S27	3	Rv0797, Rv1199c, Rv2386a
HMS24052_S28	3	Rv0297, Rv1765c, Rv2386a

Tabla 6. Tabla que muestra los genes a los que perteneces los SNPs únicos detectados por MTBseq.

Algunos de ellos no tienen un producto o función definida, y se tratan como hipotéticos. Es el caso de Rv2386a o Rv2665. Sin embargo, hay otros que producen elementos de inserción de secuencias de transposasas, como es el caso de Rv0797 para el elemento IS1547 y Rv1199c para el elemento IS1081. También destacan los genes Rv2350c y Rv0297, que están relacionados con la fosfolipasa B asociada a membrana y con la familia de proteínas PE-PEGRS respectivamente. Por último, los genes RVnr01 y Rvnr02, relacionados con el ARN ribosómico.

Estos resultados coinciden tanto en los resultados obtenidos por la herramienta MTBseq como en las bases de datos de genes de tuberculosis y del Instituto Nacional de Salud (NIH) (60, 61).

7.4. Espoligotipificación

Como se ha explicado previamente, TBProfiler permite realizar una espoligotipificación experimental de las muestras analizadas.

MUESTRA	PATRÓN BINARIO ESPACIADORES (ESPACIADOR 1 – ESPACIADOR 43)
HCU24001_S1	■■■■■■■■■■■■■■■■■■■■■■■■■□□□□□□□□□□□□■■■■■■■
HCU24002_S2	■■□■■■■■■□■■■■■■■□■□■■■■■■■■■■■■■■■■■■■■□□□□□
HCU24010_S6	■■■■■■■■□□□■■■■■■■■■□□□□□□□□□□□□□□□□□□□□■■■■■
HCU24011_S7	■■■■■■□□□■■■■□□□□□□□■■□■■□□■□□□■■■■■■■□■■■■
HMS24017_S15	□■■■■□■■□□□■■■■■□□□■□□□□□□□□□□□□□□□□□□■■□■■■
HMS24046_S23	■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■□□□□■■■■■■■
HMS24051_S27	■■■■■■■■■■■■■■■■■■■□□□□■■■■■■■■□□□□■■■■■■■
HMS24052_S28	■■■■■■■□□□□□■■■■■■■■□□□□■■□■■■■■■□■■■□□□■■■■

Tabla 7. Representación de los patrones binarios de espoligotipificación. Cuadrado negro indica presencia del espaciador, cuadrado blanco ausencia.

Los patrones de espoligotipificación permiten relacionar las muestras entre sí. Cuanto más parecidos sean, más probabilidad hay de que pertenezcan a la misma especie.
En este caso, a simple vista se puede apreciar que HCU24001_S1, HCU24010_S6 y HMS24017_S15 tienen patrones muy similares. Las tres muestras pertenecen a la especie *M. tuberculosis*. De la misma manera, esta similitud se puede apreciar en las muestras HMS24046_S23 y HMS24051_S27, perteneciendo ambas también a la especie *M. tuberculosis*. Por otra parte, la muestra HCU24002_S2 presenta un patrón único y distinto al resto, coincidiendo esto con que sea la única muestra perteneciente a la especie *M.bovis.* Por último, los patrones de las muestras HCU24011_S7 y HMS24052_S28 también son parecidos, con una presencia y ausencia de espaciadores intermitente a lo largo de todo el patrón. Estas dos muestras se corresponden con la especie *M. africanum.*

Además de los patrones, TBProfiler proporciona todas las secuencias espaciadoras, lo que puede ser útil para estudios más avanzados.

Estos resultados coinciden en prácticamente todas las muestras con los obtenidos de forma experimental. En la Imagen 1 se pueden observar los patrones de espoligotipificación que se obtuvieron de forma experimental en el IACS.

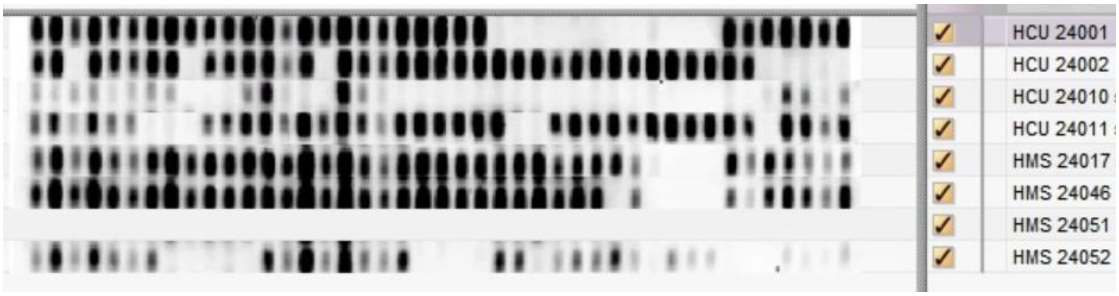

Imagen 1. Patrones de espoligotipificación obtenidos de forma experimental. Fuente: Alberto Cebollada Solanas (IACS).

Cabe destacar que de forma experimental no se realizó el patrón de la muestra HMS24051_S27. Además, al ser un proceso con un punto de subjetividad, y dependiendo del proceso de revelado, los resultados pueden variar.

Con esto se comprueba que ambos métodos obtienen resultados muy similares, teniendo la ventaja de que la espoligotipificación a partir de datos de NGS, como ha sido el caso de este proyecto, es mucho más rápida que la que se realiza en el laboratorio. Sin embargo, al ser esta función experimental y todavía no estar validada completamente, es posible que alguno de los resultados no sea del todo correcto o presente ligeras incongruencias.

7.5. Comparación de matrices de distancias de Snippy y MTBseq

En MTBseq, es posible hacer un análisis comparativo entre múltiples muestras en los pasos TBjoin, TBamend y TBgroups. De estos pasos se puede obtener una matriz de distancias que refleja cómo de separadas estás las muestras entre ellas. De la misma manera, también se ha obtenido una matriz de distancias entre muestras con la herramienta Snippy, implementando Snippy-multi y snps-dists.

Partiendo de estas matrices de distancias, se han generado dos mapas de calor que incluyen la agrupación filogenética de las muestras en grupos (pheatmap), representados en las figuras 22 y 23.

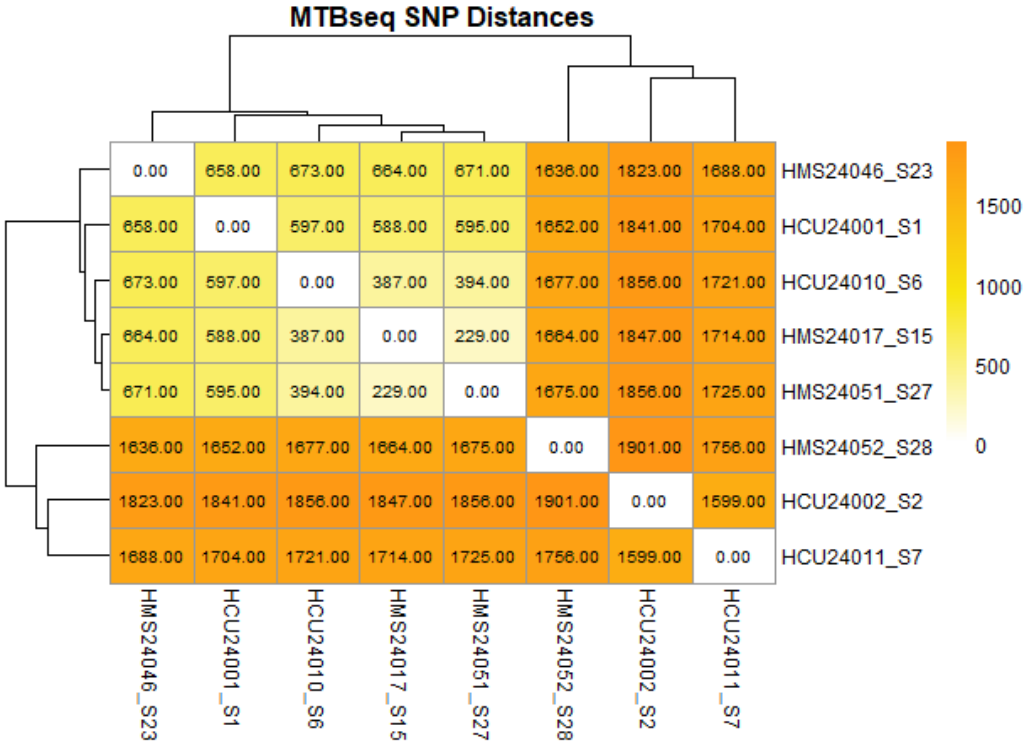

Figura 22. Mapa de calor con agrupación filogenética (pheatmap) de las muestras analizadas por MTBseq.

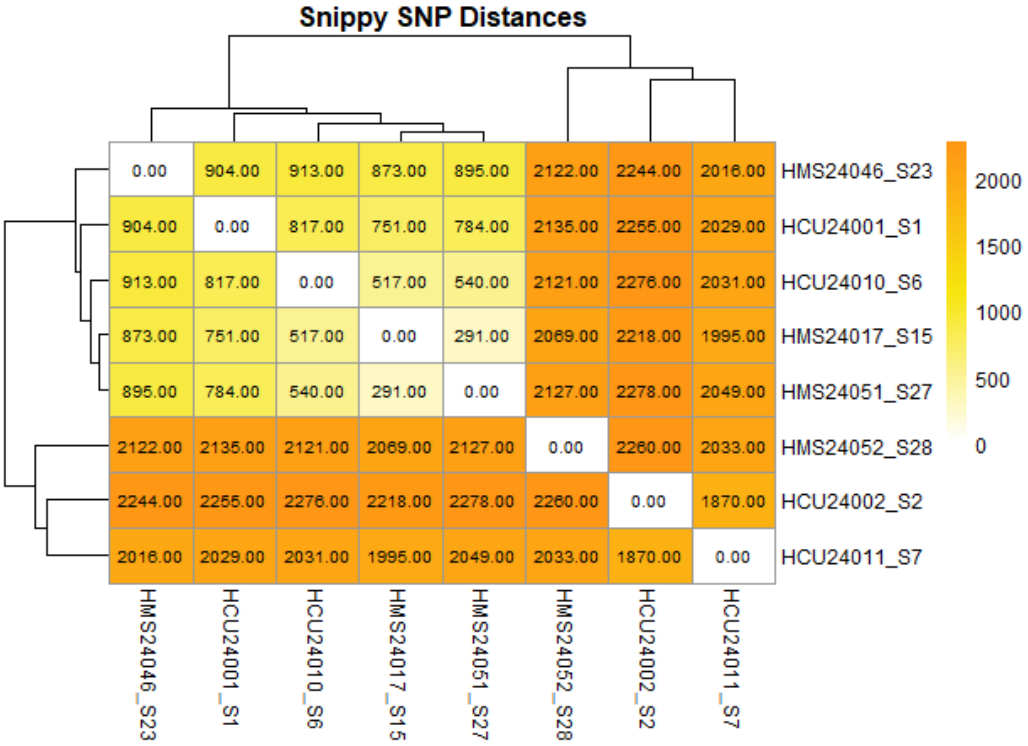

Figura 23. Mapa de calor con agrupación filogenética (pheatmap) de las muestras analizadas por Snippy.

Observando ambos gráficos, se puede ver que, en los dos casos, las muestras se agrupan de la misma manera, lo que también coincide con el linaje Homolka del que se ha hablado previamente en el punto 7.2. Además, teniendo en cuenta los distintos tonos de color, se observa que las muestras HCU24002_S2 y HMS24052_S28 son las más separadas, con una distancia de 1901 en MTBseq y de 2260 en Snippy y tonos más oscuros en sus casillas. Esto indica que estas dos muestras, pertenecientes a las especies *M. bovis* y *M. africanum*, son las menos similares entre sí.

También se puede apreciar que todas las distancias reportadas por el mapa de calor de Snippy son superiores a las de MTBseq, lo que puede ser debido a una diferencia en el criterio de filtrado de las variantes, o por diferencias en el alineamiento o en el conjunto de posiciones analizadas.

Sin embargo, TBgroups no consigue agrupar las muestras cuando se establece una distancia máxima de 15 SNPs, lo que indica que todas las muestras estás más separadas entre sí. Esto es consistente con lo explicado en los puntos 7.2 y 7.4, coincidiendo ambos en las diferentes especies de las muestras.

7.6. Comparación de las herramientas: análisis de costes y tiempo

Las herramientas pueden ser comparadas no sólo por los resultados que obtienen, sino también computacionalmente y por el tiempo que tardan en ejecutarse. Para cada contenedor, se ha obtenido un fichero .txt que contiene las estadísticas desglosadas de cada llamada al Docker, con información sobre el porcentaje de uso de la CPU, el uso de memoria con respecto a la memoria disponible, el porcentaje de uso de la memoria, y el tiempo requerido. Observando estos archivos, se puede ver fácilmente cuántos recursos consume cada una de las herramientas y el tiempo empleado.

Para una mayor claridad de los resultados, se ha calculado para cada contenedor el porcentaje medio de uso de la CPU, el uso de memoria promedio (en MiB) y el tiempo de ejecución total, plasmado en la Tabla 8:

HERRAMIENTA	PORCENTAJE MEDIO DE USO DE LA CPU	USO PROMEDIO DE MEMORIA (MIB)	TIEMPO DE EJECUCIÓN
MTBseq	31,21 %	4404,089	1:28:48
Snippy	46,78 %	943,18	0:10:17
TBprofiler	47,31 %	926,39	0:06:27

Tabla 8. Tabla comparativa de costes y tiempo de las tres herramientas utilizadas para el análisis genómico.

En cuanto al uso de la CPU, TBProfiler presenta el porcentaje medio más alto, seguido de Snippy y MTBseq. Esto significa que los procesadores del sistema, en este caso el servidor, están siendo utilizados más intensamente en TBProfiler y Snippy que en MTBseq, lo que puede reflejarse en una mayor velocidad de procesado.
Por otra parte, MTBseq es la herramienta que tiene un uso promedio de memoria más alto, lo cual es coherente ya que es la herramienta más exhaustiva y que más pasos necesita debido a su enfoque modular. En cambio, Snippy TBProfiler son mucho más ligeras en este aspecto.
Por último, si se tiene en cuenta el tiempo total de ejecución, se observa que se repite el mismo patrón: MTBseq presenta un tiempo considerablemente más alto que Snippy y TBProfiler, lo que es consistente con los resultados obtenidos en los otros dos apartados.

El estudio llevado a cabo por Laurie et al. también hizo una comparación de las combinaciones de herramientas a nivel de tiempo total y tiempo de CPU. Los resultados pusieron de manifiesto un mayor tiempo total para GATK, seguido de SAMtools y FreeBayes. Con respecto al tiempo de CPU, GATK obtuvo tiempos considerablemente superiores, seguido de FreeBayes y SAMtools. Estos resultados son consistentes con los obtenidos en la Tabla 8, donde MTBseq, que utiliza

GATK internamente, presenta un tiempo de ejecución bastante superior a las otras dos herramientas, que utilizan FreeBayes, BCFtools (que pertenece a SAMtools) (55).

Hay que tener en cuenta que estos valores se han obtenido de una ejecución, pero pueden variar ligeramente dependiendo de las condiciones específicas de cada momento. Por lo tanto, no deben tomarse como valores inamovibles.

8. LIMITACIONES DEL ESTUDIO

A pesar de que el proyecto ha reportado resultados muy positivos, existen algunas limitaciones que habría que destacar.

El tamaño muestral es bastante pequeño (8 muestras), por lo que para obtener resultados más fiables y robustos sería conveniente aumentarlo añadiendo más muestras. Al haber creado un flujo de trabajo automatizado, bastaría con añadir nuevos ficheros FASTQ al directorio y volver a lanzar los scripts.

Por otra parte, los scripts *check_samples.sh*, *FastQC.sh* y *fastp.sh* están preparados para detectar muestras iniciales en formato .fastq.gz, por lo que si las muestras de partida no estuviesen comprimidas el script determinaría que no hay muestras en el directorio. En el desarrollo del proyecto este factor no se tuvo en cuenta, ya que normalmente este tipo de muestras siempre vienen comprimidas. Sin embargo, se podría plantear como mejora y optimización del código considerar este aspecto.

Por falta de tiempo, no se ha podido llegar a completar el objetivo adicional de encapsular el flujo de trabajo completo en un único contenedor Docker, lo que hubiera sido de gran utilidad en el caso de querer escalar el proyecto a un contexto de práctica real.

9. CONCLUSIONES

Con los resultados obtenidos y la discusión realizada previamente, se pueden extraer las siguientes conclusiones.

Gracias a la automatización del flujo de trabajo, se han podido analizar las ocho muestras de manera sencilla y rápida, sin la necesidad de tener que ejecutar las herramientas una a una. Tras el análisis de calidad y el preprocesado de las muestras, se concluye que todas ellas son aptas y de una calidad suficiente para poder realizar el posterior análisis genómico.

En este segundo análisis, las tres herramientas propuestas para el estudio han sido comparadas en varios aspectos, incluyendo la detección de SNPs (anotados y no anotados), inserciones y deleciones, la clasificación de linajes, la profundidad de cobertura, las matrices de distancias generadas o los costes y tiempo que requieren. Además, se ha podido explorar la espoligotipificación que obtiene TBProfiler, lo cual permite también diferenciar y agrupar las muestras dependiendo del patrón de espoligotipificación que presenten.

Se ha llegado a la conclusión de que la detección general de SNPs es similar en todas las herramientas, mientras que MTBseq es la herramienta que más inserciones y deleciones detecta, con diferencias significativas frente a las otras dos herramientas en la profundidad de la cobertura. Sin embargo, para los SNPs anotados detectados, que se pueden observar en los diagramas de Venn generados, se concluye que es precisamente MTBseq la herramienta que menos SNPs detecta de forma única. Estos SNPs se encuentran de forma recurrente en genes proteínas y ARN ribosómico.

Las matrices de distancias no han revelado pares de muestras con distancias menores a 15 SNPs, pero sí que sirven de guía para la clasificación filogenética de las muestras, al igual que la clasificación de linajes generada por MTBseq y TBProfiler. Las muestras pertenecen principalmente a la especie *M. tuberculosis,* seguido de *M. africanum* y *M. bovis.*

Por último, el análisis de recursos y costes permite concluir que la herramienta que más tiempo requiere es MTBseq, y que además es la herramienta que menor porcentaje medio de uso de la CPU y más uso promedio de memoria presenta. En cuanto a Snippy y TBProfiler, ambas herramientas son similares en tiempo, CPU y memoria.
Con estos resultados, se ratifica que MTBseq prioriza la exhaustividad y modularidad del proceso, a costa de tiempo y memoria, mientras que Snippy y TBProfiler son más eficientes en tiempo y memoria, pero usan más CPU para lograrlo de manera rápida.

A pesar de que el objetivo adicional de encapsulación en un único contenedor no ha sido satisfecho, y se presentan ciertas limitaciones en el tamaño muestral y el código, el proyecto realizado engloba un flujo automatizado sencillo y completo, que permite un análisis de calidad y genómico automatizado. Por todo ello, queda patente que la integración de la secuenciación del genoma completo con las herramientas bioinformática proporciona una base sólida para futuras investigaciones y aplicaciones clínicas.

10. BIBLIOGRAFÍA

1. Instituto Nacional de Seguridad y Salud en el Trabajo (INSST). *Mycobacterium tuberculosis* [en línea]. Madrid: INSST, 2022 [Citado: 2 de mayo de 2025]. Disponible en: https://www.insst.es/agentes-biologicos-basebio/bacterias/mycobacterium-tuberculosis
2. Centers for Disease Control and Prevention (CDC). *Tuberculosis: Causes and How It Spreads* [en línea]. Atlanta, Georgia: CDC, 2024 [Citado: 2 de mayo de 2025]. Disponible en: https://www.cdc.gov/tb/causes/index.htm
3. Organización Mundial de la Salud (OMS). *Tuberculosis* [en línea]. Ginebra: OMS, 14 de marzo de 2025 [Citado: 2 de mayo de 2025]. Disponible en: https://www.who.int/es/news-room/fact-sheets/detail/tuberculosis
4. CHITALE, Poonam et al. *A comprehensive update to the Mycobacterium tuberculosis H37Rv reference genome* [en línea]. Nature Communications 13, nº artículo 7068 (1 Nov 2022). [Citado: 2 de mayo de 2025]. Disponible en: https://www.nature.com/articles/s41467-022-34853-x
5. FONTALVO, Dilia; GÓMEZ Doris. *Genes del Mycobacterium tuberculosis involucrados en la patogenicidad y resistencia a antibióticos durante la tuberculosis pulmonar y extrapulmonar* [en línea]. Revista Médicas UIS vol. 28, n.º 1 (1 Abr 2015). [Citado: 4 de abril de 2025]. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-03192015000100004
6. VIÑUELAS-BAYÓN Jesús; VITORIA, María Asunción; SAMPER, Sofía. *Diagnóstico rápido de la tuberculosis. Detección de mecanismos de resistencia* [en línea]. Enfermedades Infecciosas y Microbiología Clínica (1 Oct 2017). [Citado: 2 de mayo de 2025]. Disponible en: https://www.elsevier.es/es-revista-enfermedades-infecciosas-microbiologia-clinica-28-articulo-diagnostico-rapido-tuberculosis-deteccion-mecanismos-S0213005X17300678
7. COLL, Pere; GARCÍA DE VIEDMA, Darío. *Epidemiología molecular de la tuberculosis* [en línea]. Enfermedades Infecciosas y Microbiología Clínica (1 Abr 2018). [Citado: 7 de abril de 2025] Disponible en: https://www.elsevier.es/es-revista-enfermedades-infecciosas-microbiologia-clinica-28-articulo-epidemiologia-molecular-tuberculosis-S0213005X18300016
8. SOLA, C et al. *Recent developments of spoligotyping as applied to the study of epidemiology, biodiversity and molecular phylogeny of the Mycobacterium tuberculosis complex* [en línea]. Web of Science, vol 48, pp 921-932 (1 Dic 2000). [Citado: 2 de marzo de 2025]. Disponible en: https://www.webofscience.com/wos/woscc/full-record/WOS:000166330100008
9. VAN EMBDEN, JDA et al. *Genetic variation and evolutionary origin of the direct repeat locus of Mycobacterium tuberculosis complex bacteria* [en línea]. Journal of Bacteriology

(1 May 2000). [Citado: 28 de febrero de 2025]. Disponible en: https://journals.asm.org/doi/10.1128/jb.182.9.2393-2401.2000

10. CAPCHA, Luis et al. *Perfiles genéticos (IS6110) y patrones de resistencia en aislamientos de M. tuberculosis de pacientes con tuberculosis pulmonar* [en línea]. SciELO Perú: Revista Peruana de Medicina Experimental y Salud Pública, vol. 22, nº 1 (En/Mar 2005). [Citado: 6 de marzo de 2025]. Disponible en: http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1726-46342005000100002&lng=es&nrm=iso&tlng=es
11. ALVAREZ-CORRALES, Nancy Montserrat et al. *Evaluación de spoligotyping a partir de baciloscopías como metodología alterna e independiente de cultivo para la genotipificación de Mycobacterium tuberculosis* [en línea]. SciELO: Revista chilena de infectología, vol. 38, nº 1 (Feb 2021). [Citado: 10 de junio de 2025]. Disponible en: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0716-10182021000100061&lng=es&nrm=iso&tlng=es
12. KOHL, Thomas Andreas et al. *MTBseq: A comprehensive pipeline for whole genome sequence analysis of Mycobacterium tuberculosis complex isolates* [en línea]. PeerJ (13 Nov 2018). [Citado: 16 de abril de 2025]. Disponible en: https://peerj.com/articles/5895
13. DataScientist. *Pipeline: Definición, funcionamiento y uso en Data Science* [en línea]. [Citado: 25 de abril de 2025]. Disponible en: https://datascientest.com/es/pipeline-definicion-funcionamiento-y-uso-en-data-science
14. TB-Profiler [en línea]. [Citado: 25 de abril de 2025]. Disponible en: https://tbdr.lshtm.ac.uk/
15. CHARRON, Philippe; KANG, Mingsong. *VariantDetective: an accurate all-in-one pipeline for detecting consensus bacterial SNPs and SVs* [en línea]. OXFORD ACADEMICU: Bioinformatics, vol. 40 (1 Feb 2024). [Citado: 8 de junio de 2025]. Disponible en: https://academic.oup.com/bioinformatics/article/40/2/btae066/7609103
16. PHELAN, Jody E. et al. *Integrating informatics tools and portable sequencing technology for rapid detection of resistance to anti-tuberculous drugs* [en línea]. Genome Medicine, article 41 (2019). [Citado: 9 de junio de 2025]. Disponible en: https://genomemedicine.biomedcentral.com/articles/10.1186/s13073-019-0650-x
17. SYME, Anna; GLADMAN, Simon; SEEMANN, Torsten. *Microbial Variant Calling* [en línea]. Galaxy Training (s.f.). [Citado: 10 de junio de 2025]. Disponible en: https://training.galaxyproject.org/training-material/topics/variant-analysis/tutorials/microbial-variants/tutorial.html
18. Red Hat. *¿Qué es Docker y cómo funciona?* [en línea]. [Citado: 2 de mayo de 2025]. Disponible en: https://www.redhat.com/es/topics/containers/what-is-docker

19. Docker. *Accelerated Container Application Development* [en línea]. [Citado: 2 de mayo de 2025]. Disponible en: https://www.docker.com/
20. GÓMEZ-VELASCO, Anaximando et al. *Diversidad genética del Complejo de Mycobacterium tuberculosis: implicaciones clínicas y epidemiológicas* [en línea]. TIP Revista Especializada en Ciencias Químico-Biológicas (29 Jun 2023). [Citado: 24 de marzo de 2025]. Disponible en: https://tip.zaragoza.unam.mx/index.php/tip/article/view/563
21. ROETZER Andreas et al. *Whole Genome Sequencing versus traditional genotyping for investigation of a Mycobacterium tuberculosis outbreak: a longitudinal molecular epidemiological study* [en línea]. PLoS Medicine (12 Feb 2013). [Citado 26 de febrero de 2025]. Disponible en: https://pubmed.ncbi.nlm.nih.gov/23424287/
22. WALKER, Timothy M et al. *Whole-genome sequencing to delineate Mycobacterium tuberculosis outbreaks: a retrospective observational study* [en línea]. Lancet Infect Dis. (Feb 2013). [Citado: 2 de mayo de 2025]. Disponible en: https://pubmed.ncbi.nlm.nih.gov/23158499/
23. JAJOU, Rana et al. *Towards standardisation: Comparison of five whole genome sequencing (WGS) analysis pipelines for detection of epidemiologically linked tuberculosis cases* [en línea]. Eurosurveillance (27 Jun 2019). [Citado: 16 de abril de 2025]. Disponible en: https://www.eurosurveillance.org/content/10.2807/1560-7917.ES.2019.24.50.1900130
24. OLAWOYE, Idowu B; FROST, Simon D.W.; HAPPi, Christian T. *The bacteria genome pipeline (BAGEP): an automated, scalable workflow for bacteria genomes with Snakemake* [en línea]. PeerJ (27 Oct 2020). [Citado: 17 de abril de 2025]. Disponible en: https://peerj.com/articles/10121/
25. PEKER, Nilay et al. *Evaluation of whole-genome sequence data analysis approaches for short- and long-read sequencing of Mycobacterium tuberculosis* [en línea]. Microbial Genomics (Nov 2021). [Citado: 27 de abril de 2025]. Disponible en: https://pubmed.ncbi.nlm.nih.gov/34825880/
26. Bio Knowledge Lab. *Variant Calling* [en línea]. [Citado: 8 de junio de 2025]. Disponible en: https://www.b-kl.eu/variant-calling/
27. LÓPEZ, Carla et al. *Procedimiento de Microbiología Clínica 71: Aplicaciones de las técnicas de secuenciación masiva en la Microbiología Clínica* [en línea]. Sociedad Española de Enfermedades Infecciosas y Microbiología Clínica (2021). [Citado: 25 de abril de 2025]. Disponible en: https://seimc.org/contenidos/documentoscientificos/procedimientosmicrobiologia/seimc-procedimiento71.pdf

28. Babraham Bioinformatics. *FastQC A Quality Control tool for High Throughput Sequence Data* [en línea]. Babraham Bioinformatic (s.f.). [Citado: 2 de mayo de 2025]. Disponible en: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
29. BioQueue Encyclopedia. *Fastp: An ultra-fast all-in-one FASTQ preprocessor* [en línea]. BioQueue Encyclopedia (s.f.). [Citado: 2 de mayo de 2025]. Disponible en: https://open.bioqueue.org/home/knowledge/showKnowledge/sig/fastp
30. CHEN, Shifu. *Ultrafast one-pass FASTQ data preprocessing, quality control, and deduplication using fastp* [en línea]. iMeta (May 2023). [Citado: 25 de mayo de 2025]. Disponible en: https://pubmed.ncbi.nlm.nih.gov/38868435/
31. EWELS, Philip et al. *MultiQC: summarize analysis results for multiple tools and samples in a single report* [en línea]. Bioinformatics (Oct 2016). [CItado: 4 de junio de 2025]- Disponible en: https://pubmed.ncbi.nlm.nih.gov/27312411/
32. Manual bwa – Burrows-Wheeler Alignment Tool [en línea]. [Citado: 2 de mayo de 2025]. Disponible en: https://bio-bwa.sourceforge.net/bwa.shtml
33. Bowtie2 Tutorial [en línea]. Tinybio (s.f.). [Citado: 2 de mayo de 2025]. Disponible en: https://docs.tinybio.cloud/docs/bowtie2-tutorial
34. Minimap2 Tutorial [Internet]. Tinybio (s.f.). [Citado: 2 de mayo de 2025]. Disponible en: https://docs.tinybio.cloud/docs/minimap2-tutorial
35. Samtools [en línea]. [Citado: 2 de mayo de 2025]. Disponible en: https://www.htslib.org/
36. GATK: genome Analysis Toolkit. *Variant Discovery in High-Throughput Sequencing Data* [en línea]. GATK (s.f.). [Citado: 2 de mayo de 2025]. Disponible en: https://gatk.broadinstitute.org/hc/en-us
37. Picard Tutorial [en línea]. Tinybio (s.f.). [Citado: 2 de mayo de 2025]. Disponible en: https://docs.tinybio.cloud/docs/picard-tutorial
38. *Variant Calling using Freebayes* [en línea]. OmicsBox User Manual (s.f.). [Citado: 2 de mayo de 2025]. Disponible en: https://docs.omicsbox.biobam.com/latest/Variant-Calling-using-Freebayes/
39. *Freebayes* [en línea]. Docs CSC (s.f.). [Citado: 2 de mayo de 2025]. Disponible en: https://docs.csc.fi/apps/freebayes/
40. VarScan - Variant Detection in Massively Parallel Sequencing Data manual [en línea]. VasScan (s.f.). [Citado: 2 de mayo de 2025]. Disponible en: https://varscan.sourceforge.net/
41. Bcftools by samtools [en línea]. [Citado: 2 de mayo de 2025]. Disponible en: https://samtools.github.io/bcftools/
42. MTBseq_source MANUAL [en línea]. GitHub (2023). [Citado: 3 de marzo de 2025]. Disponible en: https://github.com/ngs-fzb/MTBseq_source/blob/master/MANUAL.md

43. SEEMANN, Torsten. *snippy: :scissors: Rapid haploid variant calling and core genome alignment* [en línea]. GitHub (2019). [Citado: 3 de marzo de 2025]. Disponible en: https://github.com/tseemann/snippy
44. SEEMANN, Torsten. *snp-dists: Pairwise SNP distance matrix from a FASTA sequence alignment* [en línea]. GitHub (2021). [Citado: 3 de marzo de 2025]. Disponible en: https://github.com/tseemann/snp-dists
45. PHELAN, *Jody. TBProfiler: Profiling tool for Mycobacterium tuberculosis to detect resistance and strain type from WGS data* [en línea]. GitHub (2019). [Citado: 3 de marzo de 2025]. Disponible en: https://github.com/jodyphelan/TBProfiler
46. AMEIJEIRAS, Rafael. *Cómo usar el comando docker run para ejecutar tus contenedores Docke*r [en línea]. PANDROA Tech Blog (17 Jun 2024). [Citado: 6 de marzo de 2025]. Disponible en: https://pandorafms.com/blog/es/docker-run/
47. Dockerdocs. *Docker container stats* [en línea]. Dockerdocs (s.f.). [Citado: 20 de abril de 2025]. Disponible en: https://docs.docker.com/reference/cli/docker/container/stats/
48. Dockerdocs. *docker container logs* [en línea]. Dockerdocs (s.f.). [Citado: 20 de abril de 2025]. Disponible en: https://docs.docker.com/reference/cli/docker/container/logs/
49. LERENA, Sancho. *Logs: qué son y por qué monitorizarlos* [en línea]. PANDROA Tech Blog (5 Mar 2024). [Citado: 20 de abril de 2025]. Disponible en: https://pandorafms.com/blog/es/logs/
50. GEREDIAGA, Candela. *MTB-Pipeline-Variant-Analysis* [en línea]. GitHub (14 Jun 2025). [Citado: 14 de junio de 2025]. Disponible en: https://github.com/candelagerediaga
51. NIH. *Mycobacterium tuberculosis H37Rv complete genome - Nucleotide – NCBI* [en línea]. National Institute of Health (s.f.). [Citado: 10 de junio de 2025]. Disponible en: https://www.ncbi.nlm.nih.gov/nuccore/AL123456.3?report=genbank
52. Instituto Aragonés de Ciencias de la Salud (IACS). *Biocompitación. Servicios Científico Técnicos* [en línea]. IACS (s.f.). [Citado: 11 de junio de 2025]. Disponible en: https://www.iacs.es/servicios/biocomputacion/
53. SUSANNE, Homolka et al. *High Resolution Discrimination of Clinical Mycobacterium tuberculosis Complex Strains Based on Single Nucleotide Polymorphisms* [en línea]. PLOS One (2 Jul 2012). [Citado: 10 de junio de 2025]. Disponible en: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0039855
54. COLL, Francesc et al*. A robust SNP barcode for typing Mycobacterium tuberculosis complex strains* [en línea]. Nature communications, nº artículo 4812 (1 Sep 2014). [Citado: 3 de junio de 2025]. Disponible en: https://www.nature.com/articles/ncomms5812
55. STEVE, Laurie et al. *From Wet-Lab to Variations: Concordance and Speed of Bioinformatics Pipelines for Whole Genome and Whole Exome Sequencing* [en línea]. Human Mutation

(1 Dec 2016). [Citado: 11 de junio de 2025]. Disponible en: https://pubmed.ncbi.nlm.nih.gov/27604516/

56. BO – YOUNG, Kim et al. *Optimized detection of insertions/deletions (INDELs) in whole-exome sequencing data* [en línea]. PLOS One (9 Aug 2017). [Citado: 11 de junio del 2025]. Disponible en: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0182272
57. TANG, Chao et al. *Sequence Fusion Algorithm of Tumor Gene Sequencing and Alignment Based on Machine Learning* [en línea]. Computational Intelligence and Neuroscience (31 Dec 2021). [Citado: 9 de junio 2025]. Disponible en: https://pubmed.ncbi.nlm.nih.gov/35003249
58. AMAT RODRIGO, Joaquín. *Test Kruskal-Wallis* [en línea]. Ciencia de Datos. RPubs (Ene 2016). [Citado: 11 de junio de 2025]. Disponible en: https://rpubs.com/joaquin_ar/219504
59. AMAT RODRIGO, Joaquín. *Test de Wilcoxon Mann Whitney como alternativa al t-test* [en línea]. Ciencia de Datos. RPubs (Jul 2017). [Citado: 11 de junio de 2025]. Disponible en: https://cienciadedatos.net/documentos/17_mann%E2%80%93whitney_u_test
60. TB Database [en línea]. Genes (s.f.). [Citado: 10 de junio de 2025]. Disponible en: http://tbdb.bu.edu/tbdb_sysbio/GeneIndex.html
61. NIH. *Home – Gene [en línea].* National Institute of Health (s.f.). [Citado: 8 de junio de 2025]. Disponible en: https://www.ncbi.nlm.nih.gov/gene

11. ANEXOS

11.1. Propuesta del Trabajo Fin de Grado presentada

Título del proyecto

Comparación de la herramienta MTBseq frente a Snippy y VariantDetective para el análisis de variantes de muestras de *Mycobacteryum tuberculosis* en un flujo de trabajo automatizado.

Tipología del proyecto

Este proyecto se clasifica como un proyecto de investigación aplicada. Se centra en la comparación y evaluación de herramientas bioinformáticas para el análisis de variantes genómicas en muestras de *Mycobacterium tuberculosis*. Además, incluye el desarrollo y automatización de un flujo de trabajo que integra estas herramientas, con el objetivo de mejorar la eficiencia y reproducibilidad del análisis genómico en un entorno automatizado.

Descripción y justificación del tema a tratar

La bioinformática desempeña un papel fundamental en el análisis de datos genómicos. En este trabajo, se propone la comparación de diferentes herramientas de análisis de variantes de variantes en secuencias genómicas de *Mycobacterium tuberculosis*, como MTBseq, Snippy o VariantDetective. Todo ello se desarrollará en un flujo de trabajo automatizado, lo que permitirá procesar grandes volúmenes de muestras. Se implementará un sistema automatizado que detectará nuevas muestras en una carpeta específica e iniciará automáticamente los procesos de control de calidad, llamada a variantes y análisis comparativo. Se intentará determinar si los resultados obtenidos con las tres herramientas son comparables, viendo si la herramienta MTBseq es mejor o no que las otras dos. Además, en caso de que diese tiempo, se integrará todo el flujo en un contenedor Docker desplegable en cualquier máquina para garantizar la reproducibilidad del análisis.

Objetivos del proyecto

1. **Objetivo principal:**
   Comparar las diferentes herramientas disponible de análisis de variantes de muestra en un flujo de trabajo automatizado.
2. **Objetivos específicos:**
   - Automatizar la ejecución del flujo de trabajo al detectar nuevas muestras en una carpeta específica.
   - Desarrollar un flujo de análisis comparativo entre múltiples muestras con **TBjoin**, **TBamend**, y **TBgroups**.

- Desarrollar un flujo de trabajo para las herramientas MTBseq, Snippy y VariantDetective.
- Detectar, usando las tres herramientas, diferencias entre pares de muestras con menos de 15 SNPs y determinar qué SNPs presentan dichas diferencias.
- Determinar si los resultados de las 3 herramientas utilizadas son comparables o no.

3. **Objetivos adicionales (ampliaciones si diese tiempo):**
   - Opción 1: Encapsular el flujo de trabajo completo en un único contenedor Docker listo para ser desplegado en cualquier entorno.

Metodología

El trabajo se estructurará en tres fases principales:

1. Detección automática y preprocesamiento de muestras

- Sistema de detección de nuevas muestras:
  - Monitorización de una carpeta específica del sistema de archivos.
- Control de calidad de secuencias
- Ejecución MTBseq, Snippy y VariantDetective

2. Análisis de variantes

- Detección de pares de muestras con menos de 15 SNPs de diferencia.
- Listado detallado de los SNPs diferenciadores.

3. Comparación de las diferentes herramientas propuestas

- Identificación de diferencias:
  - Para las mismas muestras, comparación de los resultados obtenidos en cada herramienta.
  - Identificación de las ventajas y desventajas de cada herramienta y determinación de la superioridad o no de MTBseq frente a las otras dos en muestras de *M.tuberculosis*.

---

4. Encapsulación completa en Docker (si diese tiempo):

- Desarrollo de un **contenedor Docker único** que encapsule todo el flujo de trabajo.
- Preparación de scripts de despliegue automático para facilitar la ejecución del análisis en cualquier entorno.
- Documentación del uso del contenedor para el usuario final.

Planificación temporal

1. Desarrollo del sistema de detección automática y preprocesamiento con FastQC y Fastp. (HECHO)
2. Automatización del flujo de trabajo con MTBseq. (HECHO)
3. Implementación del análisis comparativo de variantes.
4. Automatización del flujo de trabajo con Snippy y VariantDetective.
5. Comparativa de las tres herramientas.
6. Pruebas de funcionalidad y optimización del flujo.
7. Posible Extensión: encapsulación completa en Docker.
8. Documentación final y presentación del proyecto.

Observaciones adicionales

El tutor de este proyecto es Alberto Cebollada Solanas (también fue el tutor de prácticas tuteladas en empresa). La propuesta de este Trabajo Fin de Grado está basada en las prácticas curriculares ya realizadas, en las que se trató de automatizar un flujo de trabajo de análisis de muestras de *M.tuberculosis*. Durante el desarrollo de las mismas, se consiguió automatizar con éxito el análisis de calidad y preprocesamiento (FastQC y Fastp) y comenzar con los primeros de MTBseq (puntos 1 y 2 de la planificación temporal, marcados como HECHO).
Se vio que el desarrollo completo de la automatización del flujo podría ser útil para el análisis de las muestras, fundamentando la propuesta de completar la automatización, esta vez centrándose en comparar las diferentes herramientas disponibles de comparación de variantes, en el trascurso del TFG.

Por lo tanto, el TFG se centrará en automatizar el Análisis comparativo de variantes de la herramienta MTBseq (que no se pudo realizar durante las prácticas), automatizar las otras dos herramientas disponibles y hacer la comparativa entre ellas, así como completar la extensión opcional si es que diese tiempo. El código desarrollado durante las prácticas se utilizará para hacer el análisis de calidad y el preprocesado, así como la primera parte de la herramienta MTBseq.