UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Diseño y desarrollo de un pipeline para la identificación de variaciones genómicas en un contexto de medicina de precisión

Trabajo Fin de Máster

Máster Universitario en Ingeniería Biomédica

AUTOR/A: Ibañez Henein, Moises

Tutor/a: García Simón, Alberto

Cotutor/a: Pastor López, Oscar

Director/a Experimental: Costa Sánchez, Mireia

CURSO ACADÉMICO: 2023/2024

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas las personas que han hecho posible la realización de este Trabajo de Fin de Máster. En primer lugar, agradezco a Alberto y Mireia por su invaluable guía, paciencia y apoyo constante a lo largo de este proceso. Su experiencia y consejos han sido fundamentales para la culminación de este proyecto.

También quiero agradecérselo a mi familia y a mi novia, por su amor incondicional y por creer en mí en todo momento. Su apoyo ha sido mi mayor motivación. Este TFM es el resultado de un esfuerzo colectivo y estoy profundamente agradecido con todos los que han contribuido a su realización.

RESUMEN

La medicina de precisión es una disciplina médica que permite personalizar el tratamiento y la prevención de enfermedades considerando las diferencias individuales en genes, entorno y estilo de vida. Este enfoque ofrece una atención médica proactiva, completa y adaptada a cada paciente.

Estas diferencias individuales en los genes se conocen como variantes genómicas, que son cambios permanentes en la secuencia de ADN. Por ejemplo, como sustituciones, deleciones o inserciones de nucleótidos. Identificar y evaluar la patogenicidad de dichas variantes en los pacientes es una tarea crucial en la medicina de precisión para la prevención, identificación temprana y tratamiento personalizado de enfermedades. Sin embargo, este proceso es altamente complejo debido a múltiples factores, como por ejemplo la diversidad genómica o la alta frecuencia de mutaciones. Este proceso se divide en tres etapas: análisis primario (secuenciación del ADN a partir de una muestra biológica), secundario (limpieza de datos e identificación de variantes) y terciario (interpretación de la patogenicidad de variantes). La identificación de variantes en el análisis secundario es compleja y opaca debido a los altos costes, la complejidad de los procesos, la falta de estandarización y la poca transparencia de las herramientas disponibles.

El objetivo principal de este trabajo es **diseñar, desarrollar y validar un pipeline de código abierto y transparente para la identificación de variantes genómicas**. Se han analizado las principales técnicas y herramientas existentes y se ha propuesto una solución más eficiente, estandarizada y transparente. Al ser comparada con la empresa líder del sector, se han identificado resultados superiores a un 0.85 en la métrica F1, demostrando su utilidad y validez.

Palabras Clave: Variant calling, variantes genómicas, medicina de precisión.

ABSTRACT

Precision medicine is a medical discipline that allows for the personalization of treatment and disease prevention by considering individual differences in genes, environment, and lifestyle. This approach offers proactive, comprehensive, and tailored medical care for each patient.

These individual differences in genes are known as genomic variants, which are permanent changes in the DNA sequence. For example, they can be substitutions, deletions, or insertions of nucleotides. Detecting and evaluating the pathogenicity of these variants in patients is a crucial task in precision medicine for the prevention, early detection, and personalized treatment of diseases. However, this process is highly complex due to multiple factors, such as genomic diversity or the high frequency of mutations. This process is divided into three stages: primary analysis (DNA sequencing from a biological sample), secondary analysis (data cleaning and identification of variations), and tertiary analysis (interpretation of the pathogenicity of variants).

The identification of variations in the secondary analysis is complex and opaque due to high costs, process complexity, lack of standardization, and low transparency of the available tools.

The main objective of this work is to design, develop, and validate an open-source and transparent pipeline for the detection of gene variants. The main existing techniques and tools have been analyzed, and a more efficient, standardized, and transparent solution has been proposed. When compared with the leading company in the sector, results superior to 0.85 in the F1 metric have been identified, demonstrating its utility and validity.

Keywords: Variant calling, genomic variants, precision medicine.

Índice

Documentos

Documento I. Memoria

Documento II. Presupuesto

Documento III. Anexo

Índice de la memoria

Índice del presupuesto

Índice del anexo

Índice de figuras

Índice de tablas

Documento I. Memoria

Documento II. Presupuesto

Índice de Ejemplos

Índice de abreviaciones

ADN: Ácido Desoxirribonucleico

ARN: Ácido Ribonucleico

BQSR: *Base Quality Score Recalibration*

CPU: *Central Processing Unit*

CSV*: Comma-Separated Values*

DBSNP: *Database Single Nucleotide Polymorphisms*

GPU*: Graphics Processing Unit*

MHC: *Major Histocompatibility Complex*

NGS: *Next Generation Sequencing*

ONT: *Oxford Nanopore Technologies*

PCR*: Polimerase Chain Reaction*

SNP*: Single Nucleotide Polymorphisms*

SNV: *Single Nucleotide Variant*

WES: *Whole Exome Sequencing*

WGS: *Whole Genome Sequencing*

GATK: *Genomic Analysis Toolkit*

BED: *Browser Extensible Data*

VCF: *Variant Call Format*

BAM: *Binary Aligned Map*

SIF: *Singularity Image Format*

DEF: *Singularity Definition File*

MEMORIA

Diseño y Desarrollo de un pipeline para la identificación de variaciones genómicas en un contexto de medicina de precisión

Documento I

Moisés Ibáñez Henein

Máster en Ingeniería Biomédica

Curso académico 2023/2024

1 INTRODUCCIÓN

La medicina de precisión, fundamentada en el análisis genómico, promete transformar el cuidado de nuestra salud. Sin embargo, enfrenta diversos retos que nos impiden desarrollar todo su potencial. Estos retos se fundamentan en los altos costes de ejecución y en la falta de transparencia en los métodos actuales de identificación de variantes. Este Trabajo Fin de Máster aborda estos retos de manera directa, ubicándose en el campo de la bioinformática y enfocándose en el diseño, desarrollo y validación de un pipeline económico y transparente para la identificación de variantes genómicas.

Este trabajo se ha desarrollado en colaboración con un grupo de biólogos moleculares, genetistas y biólogos, incluyendo el presidente del grupo chileno de cáncer hereditario[1]. Su experiencia en este dominio ha sido fundamental para identificar las necesidades específicas en el mismo: por un lado, aumentar significativamente la transparencia en los procesos de identificación de variantes, dando una mayor explicabilidad y confianza en los resultados; por otro lado, reducir los costes asociados al proceso, lo que permitiría un mayor acceso a estas tecnologías. Por lo tanto, se han abordado tanto la reducción de costes como el incremento en la transparencia mediante un enfoque basado en el uso de alternativas de código abierto. Esta estrategia permite una completa trazabilidad del proceso, aumentando así su transparencia, y elimina los altos costes asociados a licencias de software propietario.

En este capítulo, se establecerá el contexto del trabajo, la motivación que nos impulsa, los trabajos relacionados con nuestro proyecto, y la estructura que se ha seguido para redactar la memoria de este Trabajo Fin de Máster.

1.1 IMPORTANCIA Y MOTIVACIÓN DEL TRABAJO

Como se ha mencionado anteriormente, este Trabajo Final de Máster surge de la necesidad de ofrecer a diferentes grupos de profesionales alternativas más económicas y transparentes a las opciones de mercado actuales. Una de las principales barreras en la implementación de análisis genómicos es el coste económico. Mientras que algunos costes, como los asociados a los equipos, reactivos y almacenamiento son ineludibles, existen otras áreas donde se pueden reducir costes de manera significativa, en concreto, los gastos asociados al análisis de los datos. Es por esto por lo que el Trabajo Final de Máster se ha enfocado en reducir estos costes al máximo.

La otra dimensión de interés es la falta de transparencia en los métodos comerciales actuales, lo que impide una comprensión completa del proceso y reduce la explicabilidad y entendibilidad de los resultados obtenidos. La existencia de algoritmos propietarios impide que los profesionales puedan evaluar sus resultados y compararlos con los de otros profesionales. Por otro lado, desde una perspectiva puramente científica, es muy difícil realizar una verificación independiente de los procesos.

[1] https://cancerhereditario.cl/

Por lo tanto, las principales motivaciones de este trabajo son:

- **Reducir costes**: Ofrecer alternativas más económicas al diagnóstico genómico, lo que permitirá una mayor accesibilidad y adopción de estas tecnologías en la práctica clínica.
- **Aumentar la transparencia**: Garantizar la comprensibilidad y trazabilidad de todos los procesos involucrados, desde la recolección de muestras hasta el análisis de datos, asegurando que todos los pasos sean claros y verificables.

1.2 MEDICINA DE PRECISIÓN Y SALUD

La medicina moderna ha experimentado una transformación en las últimas décadas con la llegada de las nuevas técnicas de secuenciación y la medicina de precisión. Esta transformación representa un cambio en el paradigma tradicional de enfoque generalista para transitar hacia un paradigma donde el foco se pone en el individuo y aquellas características que lo hacen único.

Los análisis de variantes genómicas se enmarcan en la medicina de precisión. Estos análisis identifican alteraciones en el ADN de un individuo y estudian cómo las mismas afectan a la susceptibilidad de sufrir ciertas enfermedades o a la respuesta a medicamentos. La capacidad de identificar y comprender estas variantes de una manera efectiva ha abierto nuevas vías para el diagnóstico, la prevención y el tratamiento de enfermedades considerando la variabilidad genómica de cada persona (Precision Medicine, s. f.).

La aplicación de la medicina de precisión se extiende a muchas áreas clínicas, siendo particularmente importante en la oncología y en las enfermedades raras, donde permite una comprensión más profunda de los mecanismos moleculares de estas patologías. Un ejemplo paradigmático de la aplicación de la medicina de precisión en la oncología consiste en la identificación de mutaciones en los genes BRCA1 y BRCA2. Estas pruebas permiten identificar a mujeres con un alto riesgo de desarrollar cáncer de mama y ovario (Kuchenbaecker et al., 2017). Otro ejemplo es el de la fibrosis cística y el gen CFTR (Farinha & Callebaut, 2022)

Para poder entender con mayor nivel de detalle el potencial y los desafíos asociados a la medicina de precisión y la identificación de variantes, es necesario un conocimiento de los principios básicos de la genómica molecular. En la siguiente sección, se explorarán en detalle dichos principios básicos, sentando las bases necesarias para comprender la complejidad de los análisis genómicos y la importancia de desarrollar métodos económicos y transparentes de identificación de variantes.

1.2.1 Componentes, estructura del ADN y productos génicos.

El ácido desoxirribonucleico o ADN, es la molécula portadora de las instrucciones esenciales para el desarrollo y funcionamiento de los seres vivos y algunos virus. Durante la reproducción, el ADN se replica y se transmite de padres a hijos, recibiendo cada descendiente una combinación única del material genómico de ambos progenitores.

El ADN está formado por nucleótidos, que están compuestos por un grupo fosfato, un azúcar (desoxirribosa) y una base nitrogenada, que puede ser adenina, timina, guanina o citosina.

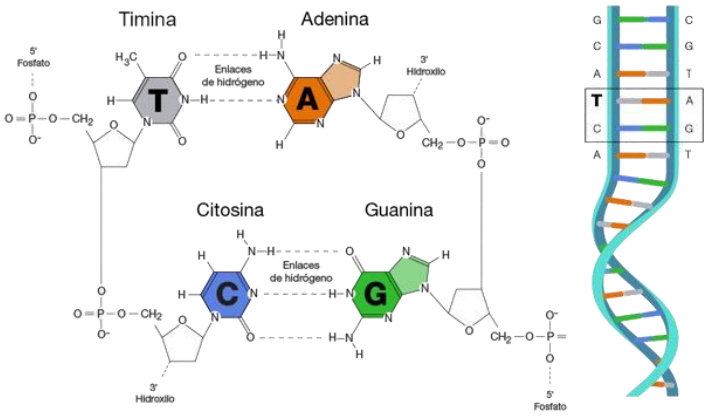

Figura 1 Esquema modificado del ADN (Timina, s.f.).

En humanos, el ADN se organiza en una doble hélice, donde las bases de ambas cadenas se unen mediante puentes de hidrógeno. El ADN se encuentra en forma de cromatina en el núcleo de todas las células a excepción de los glóbulos rojos. Durante la división celular esta cromatina se condensa en cromosomas. Los humanos poseemos 23 pares de cromosomas, incluyendo 22 pares de autosomas y un par de cromosomas sexuales, siendo XX en mujeres y XY en hombres.

Además, en el ADN hay regiones a las que se les denominan genes. Estos componen una unidad de información en una posición del ADN que codifica un producto génico, generalmente una proteína esencial para el desarrollo y funcionamiento celular. La síntesis de proteínas comienza con la transcripción de una región del ADN en ácido ribonucleico o ARN, mediada por la enzima ARN polimerasa. Después de la traducción se ensamblan las proteínas en los ribosomas del citoplasma utilizando la información codificada en el ARN.

Por lo tanto, las variantes en el ADN pueden alterar la secuencia de aminoácidos de las proteínas, afectando su estructura y función, lo que puede llevar a diferencias en características físicas, susceptibilidad a enfermedades y otras funciones biológicas.

1.2.2 Tipos de variaciones en el ADN

Habiendo sentado las bases de la genómica en la sección anterior, ahora es el momento de adentrarnos en la variabilidad genómica que se ha mencionado al principio. En esta sección, se explorarán las variantes en el ADN y cómo pueden afectar a las características y a la salud de las personas.

Estas variantes son cambios moleculares que afectan a las bases de los genes, incluyendo deleciones (pérdida de material genómico), inserciones (adición de material genómico) y sustituciones (reemplazo de material genómico)

Este Trabajo Final de Máster se centra en las variantes genómicas, que se dividen según la célula de la que se extrae el ADN en dos tipos: somáticas y germinales. Las variantes somáticas son cambios genómicos en células no reproductoras, causados por factores como radiación o errores de replicación del ADN. Por otro lado, las variantes germinales son alteraciones en células que forman gametos, y a diferencia de las anteriores son transmisibles a la descendencia.

Cabe destacar que no todas las variantes son perjudiciales; algunas son neutras o beneficiosas, fomentando la diversidad genómica, crucial para la evolución y adaptación de las especies.

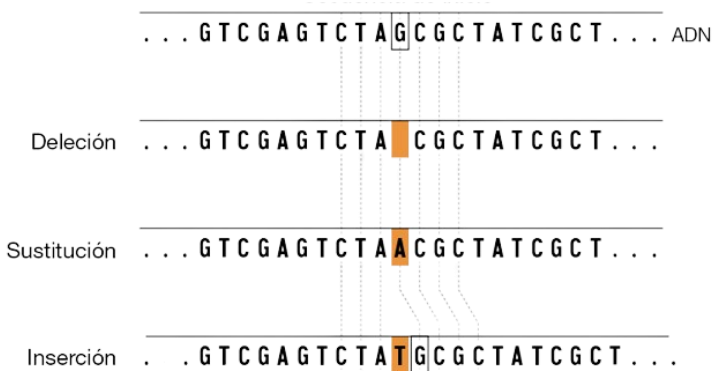

Figura 2 Ejemplo de variantes genómicas (Mutación, s.f.).

1.3 OBJETIVOS

El objetivo de este Trabajo Final de Máster es **diseñar, desarrollar y validar un pipeline de código abierto y transparente para la identificación de variantes genómicas.** Este objetivo se puede desglosar en los siguientes subobjetivos:

Subobjetivo 1: Entender el proceso de identificación de variantes (*variant calling*). Se realizará un estudio del estado del arte y las herramientas existentes, identificando sus fortalezas y limitaciones.

Subobjetivo 2: Diseñar y desarrollar un pipeline para identificar variantes genómicas.

Subobjetivo 3: Validar el pipeline con datos de pacientes reales frente a los resultados dados por una empresa líder en el sector (*gold standard*).

1.4 OBJETIVOS DE DESARROLLO SOSTENIBLE (ODS)

En este apartado se van a exponer cuáles son los objetivos de desarrollo sostenible, pertenecientes a la agenda 2030 de la Organización de las Naciones Unidas (ONU), que se busca cumplir a lo largo de este trabajo.

En primer lugar, se abordará el ODS 3: Salud y bienestar. Este Trabajo Final de Máster de identificación de variantes en el genoma tiene un impacto directo en la salud y el bienestar. Al mejorar la transparencia y reducir los costes de las herramientas genómicas, se logra que las pruebas genómicas sean más accesibles para una mayor parte de la población. Esto puede llevar a diagnósticos más tempranos y tratamientos personalizados, mejorando así la calidad de vida.

También se va a afrontar el ODS 9: Industria, innovación e infraestructuras. Al desarrollar un pipeline eficiente para la identificación de variantes genómicas, se está contribuyendo al avance tecnológico en el campo de la bioinformática. Esto no solo impulsa la industria, sino que también puede fomentar la creación de infraestructuras más robustas y eficientes para la investigación y el desarrollo en genómica.

Finalmente, el último objetivo que se planteará es el ODS 10: Reducción de las desigualdades. Al hacer que las herramientas genómicas sean más transparentes y asequibles, el proyecto ayuda a reducir las desigualdades en el acceso a la atención médica avanzada. Las poblaciones que antes no podían permitirse pruebas genómicas ahora pueden beneficiarse de ellas, lo que contribuye a una distribución más equitativa de los recursos.

1.5 ESTRUCTURA DEL TRABAJO FINAL DE MÁSTER

En esta sección se presenta la estructura seguida para la elaboración de la memoria de este Trabajo de Fin de Máster, que consta de un total de siete capítulos:

- **Capítulo 1. Introducción.** Se introduce el marco contextual del Trabajo Fin de Máster, así como la motivación y objetivos para llevarlo a cabo.
- **Capítulo 2. Metodología.** Se describe la metodología Design Science, la cual ha guiado el desarrollo de este Trabajo Fin de Máster.
- **Capítulo 3. Investigación del problema.** Se propone entender el proceso de identificación de variantes. Así como identificar y explicar con detalle qué aspectos son más relevantes en éste.
- **Capítulo 4. Diseño y desarrollo del pipeline.** Se describe el proceso seguido para desarrollar el pipeline. Se indican los datos, recursos y métodos empleados para la realización de este.
- **Capítulo 5. Validación del pipeline.** Se lleva a cabo una comparación de las diferentes técnicas disponibles en el pipeline para evaluar cual es la mejor.
- **Capítulo 6. Discusión.** Se discuten los resultados obtenidos en el Trabajo Final de Máster, analizando su relevancia y las implicaciones que tienen en el campo de estudio.
- **Capítulo 7. Conclusiones y trabajo futuro.** Se sintetizan los resultados del Trabajo Final de Máster y se llegan a conclusiones. Además, se especifican las líneas para un trabajo futuro.
- **Capítulo 8. Bibliografía**. Se presenta la bibliografía de Trabajo Final de Máster.

2 METODOLOGÍA

2.1 DESIGN SCIENCE

Design Science es un enfoque de investigación que se centra en la creación y evaluación de artefactos con el objetivo de resolver problemas en un contexto específico. En este enfoque, tanto el artefacto como su interacción con el contexto tienen una gran importancia porque, dependiendo del contexto, un artefacto puede ser útil y solucionar el problema o no.

Este enfoque se utiliza comúnmente en disciplinas como la ingeniería y la informática, donde el objetivo es diseñar y construir sistemas y tecnologías que mejoren la eficiencia, la efectividad o la experiencia del usuario con respecto a un proceso específico. Design Science implica un proceso iterativo de identificación de problemas, diseño y validación de soluciones y, finalmente, la implementación.

Para el desarrollo de este Trabajo de Fin de Máster, se ha seguido la metodología de investigación de ciencia del diseño (Design Science) propuesta por Wieringa en (Wieringa, 2014). El artefacto de este Trabajo de Fin de Máster es un **pipeline diseñado específicamente para la identificación de variantes genómicas en el ámbito de la medicina de precisión**. Este pipeline es una herramienta que permite analizar y procesar datos genómicos con el fin de identificar variantes en el ADN que pueden tener implicaciones clínicas. En el contexto de la **medicina de precisión**, la identificación precisa y rápida de estas variantes genómicas es necesaria para personalizar tratamientos y mejorar los resultados en los pacientes. Por lo tanto, el pipeline no solo facilita el análisis de datos genómicos, sino que también juega un papel importante en la implementación de estrategias terapéuticas adaptadas a las características genómicas individuales de cada paciente.

Dentro de la metodología del Design Science se contemplan dos tipos de problemas:

- I) Problemas de diseño, asociados a un ciclo de diseño.
- II) Problemas de conocimiento, asociados a un ciclo empírico o experimental.

En el caso de los problemas de diseño se busca aplicar la metodología de Design Science para provocar un cambio en el mundo real, de forma que se diseñe una solución que tenga en cuenta las necesidades de todos aquellos que vayan a utilizar la solución o se beneficien de ella. En cambio, para los problemas de conocimiento o experimentales, la metodología de Design Science ofrece un marco para obtener conocimiento acerca del mundo real mediante la respuesta de preguntas, sin crear nuevas soluciones.

Basándonos en las definiciones anteriores, este Trabajo Final de Máster instancia un ciclo de diseño porque crea una solución tangible a un problema. Un ciclo de diseño se divide en tres fases:

- **Investigación del problema.** En esta primera etapa del ciclo se caracteriza el problema a resolver, identificando las necesidades de los usuarios, el marco conceptual del problema, qué se pretende mejorar con el diseño, y las necesidades de esta mejora.
- **Diseño de la solución**. En esta etapa se especifican todos los requerimientos de la solución propuesta para el problema práctico.

- **Validación de la solución**. En esta etapa se estudia si la solución diseñada en la segunda etapa permite dar cobertura a todas las dimensiones del problema, definidas en la primera etapa del ciclo.

2.1.1 Preguntas de investigación

Siguiendo el esquema del Design Science para alcanzar nuestros subobjetivos, se plantearán una serie de preguntas de investigación asociadas, que nos ayudarán a profundizar en cada aspecto del proyecto. Estas preguntas son las siguientes

Subobjetivo 1: Entender el proceso de identificación de variantes (*variant calling*). Se realizará un estudio del estado del arte y las herramientas existentes, identificando sus fortalezas y limitaciones.

- PI1: ¿Cómo se obtienen las secuencias de nucleótidos para el *variant calling*?
- PI2: ¿Cómo se codifican las secuencias de nucleótidos para su análisis?
- PI3: ¿Cómo se identifican las variantes genómicas?

Subobjetivo 2: Diseñar y desarrollar un pipeline para identificar variantes genómicas.

- PI4: ¿Cómo obtener un pipeline para identificar variantes genómicas?
- PI5: ¿Qué hardware es necesario para implementar este pipeline?

Subobjetivo 3: Validar el pipeline con datos de pacientes reales frente a los resultados dados por una empresa líder en el sector (*gold standard*).

- PI6: ¿En cuánto se diferencian los resultados de nuestro pipeline al *gold standard*?
- PI7: ¿Hasta qué punto nuestra herramienta es útil?

Las respuestas a estas preguntas se irán respondiendo a lo largo y al final de este trabajo para facilitar al lector la comprensión del documento.

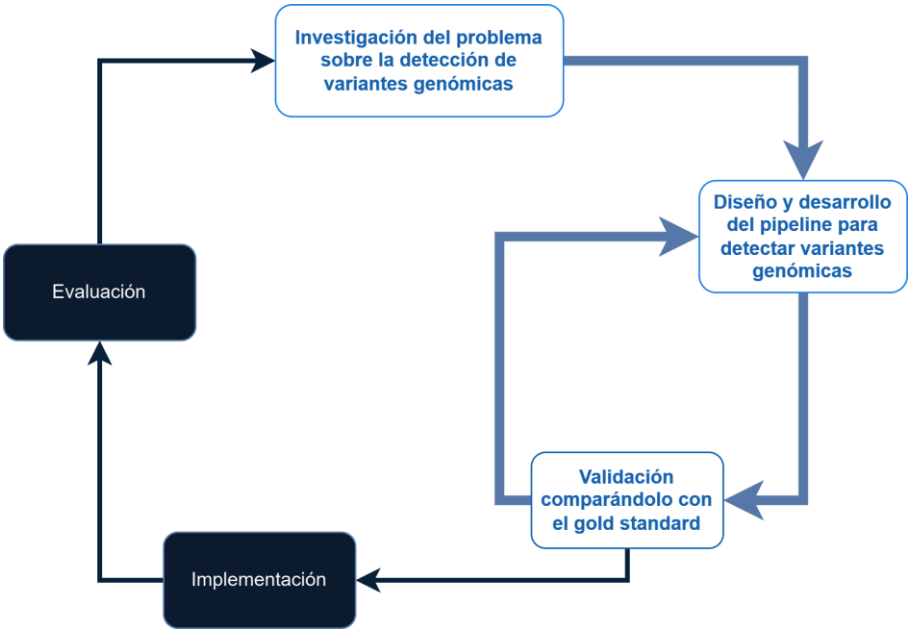

Figura 3 Ciclo iterativo de diseño en la metodología Design Science.

3 INVESTIGACIÓN DEL PROBLEMA

En esta sección se aborda el subobjetivo 1 de nuestro proyecto, dando respuesta a las preguntas de investigación asociadas. Empezará abordando la problemática, seguido de un análisis del estado del arte. Luego, se explicará cómo se obtienen y codifican estas secuencias, y finalmente, se explorarán los alineadores utilizados, las herramientas de recalibracion, y las técnicas de *variant calling*.

3.1 PROBLEMÁTICA

El proceso de identificación de variantes presenta varios problemas importantes. Primero, los altos costes en tiempo y recursos computacionales pueden ser un obstáculo para laboratorios con presupuestos limitados. Además, la complejidad del proceso, que requiere conocimientos profundos en genómica y bioinformática, representa un desafío para investigadores sin formación en estas áreas. También, la falta de estandarización en los métodos y herramientas utilizados dificulta la comparación de datos entre estudios. Finalmente, la opacidad de algunas técnicas puede afectar la confianza en los resultados y llevar a errores en el diagnóstico. Por lo tanto, es necesario desarrollar soluciones a estos desafíos para impulsar el progreso en la medicina de precisión.

3.2 ESTADO DEL ARTE

Un paso previo en cualquier investigación es preguntarnos cuál es el estado del arte del campo de interés, por lo que en esta sección se analizará el estado del arte en cuanto a la identificación de variantes genómicas.

Para analizar el estado del arte se expondrán los resultados del Truth Challenge V2 de PrecisionFDA. Esta competición se centró en la evaluación y el rendimiento de diferentes metodologías de *variant calling* en un marco de referencia común, con un enfoque en el análisis comparativo en diferentes regiones (Olson et al., 2022).

Este desafío invitó al público a evaluar el rendimiento de estas metodologías utilizando conjuntos de datos bien caracterizados. Dichos datos consistían en tres personas de ascendencia judía askenazí (HG002, HG003 y HG004), proporcionados en forma de lecturas cortas y largas (FASTQ).

Tecnología	GIAB ID	Numero de lecturas
Illumina	HG002	415.086.209
	HG003	419.192.650
	HG004	420.312.085
	HG002	8.449.287

PacBio HiFi	HG003	7.288.357
	HG004	7.089.316
ONT	HG002	19.328.993
	HG003	23.954.632
	HG004	29.319.334

Tabla 1 Datos usados por tecnología y número de lecturas en el Challenge de PrecisionFDA V2 (Olson et al., 2022).

Según los autores, se pudo observar, que el uso combinado de diversas tecnologías de secuenciación (MULTI) es necesario si se quiere aspirar a los mejores resultados. Esta metodología multifacética probó ser superior, especialmente en comparación con enfoques que dependen de una sola tecnología. En particular, se benefician enormemente de esta estrategia integrada las regiones que presentan desafíos significativos en el mapeo, las duplicaciones segmentales y en el Complejo Mayor de Histocompatibilidad (MHC).

Tecnología	**Región Genómica**	**Participante**	**F1**
MULTI	Todas las Regiones	Sentieon	0.999
MULTI	Todas las Regiones	Roche Sequencing Solutions	0.999
MULTI	Todas las Regiones	The Genomics Team in Google	0.999
MULTI	Regiones Difíciles de Mapear	Roche Sequencing Solutions	0.994
MULTI	MHC	Sentieon	0.998
ILLUMINA	Todas las Regiones	DRAGEN	0.997
ILLUMINA	Regiones Difíciles de Mapear	DRAGEN	0.969
ILLUMINA	MHC	Seven Bridges Genomics	0.992
PACBIO	Todas las Regiones	The Genomics Team in Google	0.992
PACBIO	Regiones Difíciles de Mapear	Sentieon	0.993
PACBIO	MHC	Sentieon	0.995
ONT	Todas las Regiones	The UCSC CGL and Google	0.965
ONT	Regiones Difíciles de Mapear	The UCSC CGL and Google	0.983

ONT	MHC	Wang Genomics Lab	0.972

Tabla 2 Resultados por tecnología y región analizada en el challenge de PrecisionFDA V2 (Olson et al., 2022).

Además, se puede observar cómo DeepVariant, utilizado por el equipo de genómica de Google Health y el UCSC CGL and Google, sobresale en la mayoría de las tecnologías. Sin embargo, en la tecnología de Illumina, es DRAGEN quien lidera, con la excepción del complejo mayor de histocompatibilidad. Según un estudio realizado en 2010 (Leinonen et al., 2010) para NGS, aproximadamente el 80% del mercado estaba dominado por Illumina, y actualmente la situación es muy similar. Por lo tanto, aunque DRAGEN no se distingue en otras tecnologías ni en el uso de varias, su rendimiento en los datos de Illumina es notable y, debido a su amplia presencia en el mercado, durante este Trabajo Final de Máster se considerará el *gold standard.*

3.3 OBTENCIÓN DE SECUENCIAS

Para entender cómo se realiza el *variant calling* primero se debe entender cómo se obtienen estas secuencias de nucleótidos, y la evolución histórica de la obtención de estas.

En 1990 se inició el Proyecto del Genoma Humano el cual fue un hito científico que desentrañó las aproximadamente 3.000 millones de bases del genoma humano. En 2003, se llegó a completar el 85% del primer genoma y en 2022, se subsanaron las lagunas y la secuencia quedó completa. En total, la secuenciación del primer genoma humano llevó 32 años.

Hoy en día, gracias a las técnicas *Next Generation Sequencing* o NGS, es posible secuenciar el genoma completo de una persona en tan solo un día. La diferencia radica en la cantidad de cadenas de ADN secuenciadas simultáneamente. Mientras que las técnicas de NGS procesan miles de millones de cadenas de ADN al mismo tiempo, el Proyecto Genoma Humano se basaba en la secuenciación Sanger, la cual solo permitía secuenciar una cadena a la vez. No obstante, las tecnologías de NGS solo son posibles gracias a que tras el Proyecto del Genoma Humano se pudo crear una secuencia de ADN humana de referencia.

3.3.1 Sanger Sequencing

La secuenciación del ADN se inició con la técnica desarrollada por Frederick Sanger. Esta ha sido fundamental para numerosos avances en la genómica y la biología molecular. Con los años esta técnica se ha mejorado, incluyendo el uso de PCR. El método de Sanger explicado en (Crossley et al., 2020) comienza con la toma de una muestra de ADN de doble cadena. Este ADN se desnaturaliza mediante el uso de calor, lo que resulta en una única cadena de ADN.

Después se preparan cuatro tubos de ensayo, cada uno con una solución de cadena de ADN y la enzima ADN polimerasa, amplificando la cadena previa mediante una PCR. A continuación, se introducen en los tubos de ensayo dos tipos diferentes de nucleótidos: los oxirribonucleótidos y los dideoxirribonucleótidos.

Los oxirribonucleótidos, son los componentes básicos del ADN y no tienen un grupo hidroxilo (OH) presente en el carbono número dos. Por otro lado, los dideoxirribonucleótidos son nucleótidos

modificados que carecen de un grupo hidroxilo en el carbono número tres. Esta ausencia impide que otro nucleótido se una a él, actuando como una señal de parada para el proceso de replicación.

Cada uno de los cuatro tubos de ensayo contiene un tipo diferente de dideoxirribonucleótido: ddTTP, ddATP, ddGTP y ddCTP, correspondientes a timina, adenina, guanina y citosina, respectivamente. Finalmente, se puede utilizar la electroforesis capilar para visualizar las bandas.

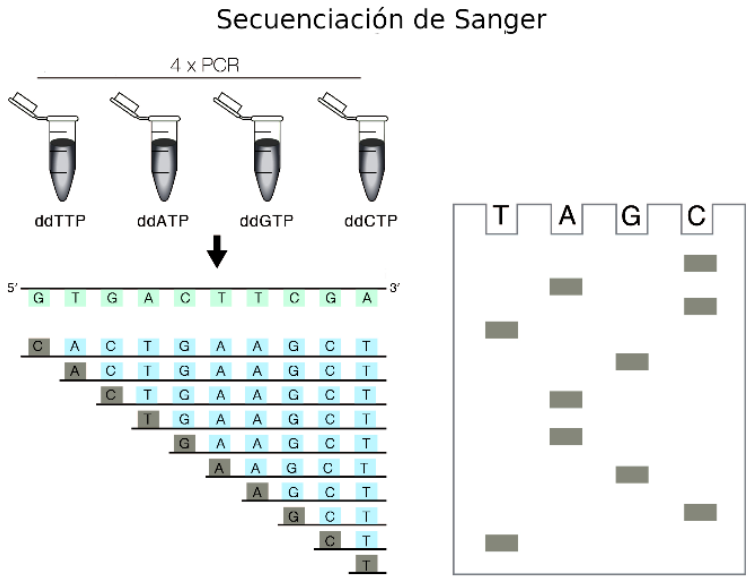

Figura 4 Esquema modificado de la secuenciación de Sanger (Rodriguez & Krishnan, 2023).

3.3.2 Next Generation Sequencing

Posteriormente al método de Sanger emergieron otras técnicas de secuenciación que se denominan NGS. Son una amalgama de técnicas con muchas diferencias entre ellas. Por razones de practicidad se describirá la técnica empleada por Illumina la cual es la secuenciación por síntesis, explicada en (Rodriguez & Krishnan, 2023).

Esta técnica permite la secuenciación tanto del ADN como del ARN. Puede abarcar todo el genoma o transcriptoma, únicamente las regiones codificantes del ADN, o genes específicos en el ADN o ARN.

En primer lugar, se recogen las muestras y se purifica el ADN o el ARN. A continuación, se comprueba que el ADN o el ARN sean puros y no estén degradados. Si se secuencia el ARN, previamente debe transcribirse a ADN.

A continuación, se prepara una biblioteca a partir del ADN. Una biblioteca es una colección de fragmentos cortos de un tamaño determinado. Estos fragmentos se crean cortando el ADN mediante ondas sonoras de alta frecuencia o enzimas.

Después, a cada extremo de estos fragmentos se añaden secuencias de ADN, llamadas adaptadores. Estos adaptadores contienen la información necesaria para la secuenciación. También incluyen un índice para identificar la muestra. Por último, se eliminan los adaptadores no unidos y la biblioteca queda completa.

La secuenciación se produce en una superficie de vidrio de una celda de flujo. En la superficie de la celda de flujo se fijan fragmentos cortos de ADN, denominados oligonucleótidos. Estos oligonucleótidos coinciden con las secuencias de los adaptadores de la biblioteca.

Una vez obtenida la biblioteca, se desnaturaliza para formar cadenas individuales de ADN. Luego, se añade a la celda de flujo, donde se une a uno de los dos oligonucleótidos, formando la cadena directa. Posteriormente, se crea la cadena reversa y se lava la cadena directa.

Con la biblioteca unida a la celda de flujo si la secuenciación comenzara en este punto, la señal fluorescente sería demasiado baja para la identificación. Por tanto, cada fragmento de la biblioteca debe amplificarse.

Para esta amplificación las cadenas de ADN se unen al segundo oligonucleótido en la celda de flujo para formar un puente. Estas hebras se copian y luego, estos fragmentos de doble cadena se desnaturalizan. Este proceso de copia y desnaturalización se repite continuamente, formando grupos localizados. Siguiendo en el proceso, se cortan las hebras inversas, estas hebras se eliminan mediante lavado, dejando la hebra directa lista para la secuenciación.

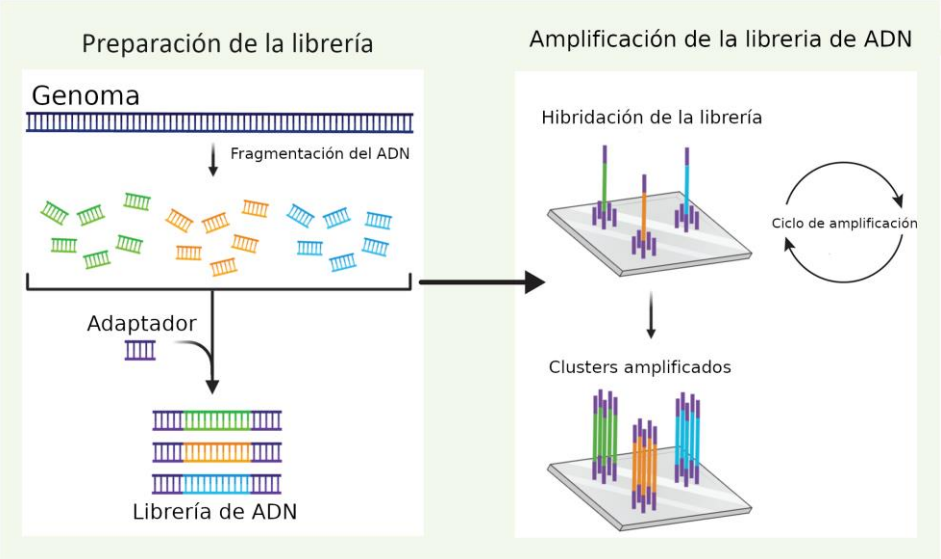

Figura 5 Esquema de la preparación de librerías y de la amplificación en NGS por síntesis (Rodriguez & Krishnan, 2023).

A continuación, los nucleótidos fluorescentes G, C, T y A se añaden a la celda de flujo junto con la ADN polimerasa. Cada nucleótido tiene una etiqueta fluorescente de un color diferente y un terminador, lo que significa que sólo se puede secuenciar un nucleótido a la vez. Primero, la base complementaria se une a la secuencia. Luego, la cámara lee y registra el color de cada grupo.

Posteriormente, una nueva solución fluye y elimina los terminadores. Los nucleótidos y la ADN polimerasa vuelven a fluir y se secuencia otro nucleótido. Estos ciclos de lectura continúan durante el número de lecturas establecido en el secuenciador. Una vez completados, estas secuencias se lavan. Luego, se secuencia el primer índice y se vuelve a lavar. Si sólo se necesita una lectura, la secuenciación terminaría aquí.

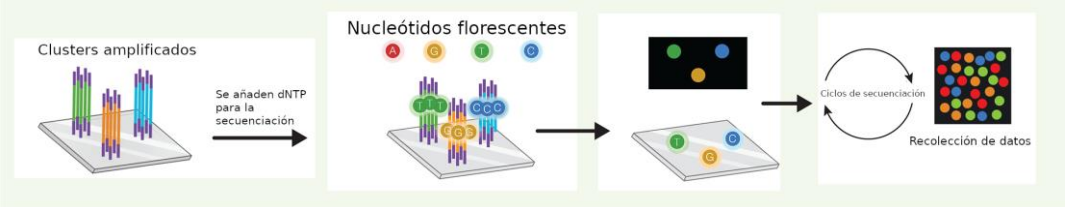

Figura 6 Esquema de la obtención de las secuencias en NGS por síntesis (Rodriguez & Krishnan, 2023).

Sin embargo, para la secuenciación por pares, se secuencia el segundo índice, así como la hebra inversa de la biblioteca. No hay cebador para la segunda lectura de índice, en su lugar, se crea un puente para que el segundo oligonucleótido actúe como cebador. Luego, se secuencia el segundo índice.

Luego, se genera la hebra inversa y se cortan y lavan las hebras delanteras. Después, se secuencian las hebras inversas.

Una vez finalizada la secuenciación, se filtran las lecturas erróneas. Entre ellas se incluyen los clústeres que se solapan, adelantan o retrasan la secuenciación o son de baja intensidad.

En estas técnicas, la profundidad de lectura es una métrica clave, representando el número de lecturas por nucleótido. La profundidad media de lectura se refiere al promedio de lecturas en toda la región secuenciada. Diferentes regiones y aplicaciones requieren distintas profundidades. Por ejemplo, en la secuenciación del genoma completo se recomienda una profundidad media de 30, mientras que para el cáncer se sugiere una profundidad media de 1500. Este aumento en la profundidad se traduce en un mayor número de ciclos en el secuenciador.

3.3.3 Genoma de referencia y sus versiones

Como se ha señalado previamente, los genomas que se examinan no siempre se adquieren a través de las técnicas NGS. En ocasiones, se recurre a otros métodos, como la secuenciación de Sanger, que analiza únicamente una cadena de ADN. De hecho, la existencia de esta técnica y los datos generados por ella han sido fundamentales para el desarrollo de los sistemas NGS, dado que para mapear las lecturas de NGS se usa un genoma obtenido mediante Sanger. Este genoma se denomina genoma de referencia.

En esta sección se señalarán algunos de los genomas de referencia, sus características y en qué escenarios es recomendable su uso según la documentación.

3.3.3.1 Genoma enmascarado y sin enmascarar

A veces al genoma de referencia independientemente de la versión se le aplica un enmascaramiento esto es recomendable para identificar con mayor precisión regiones del genoma que presentan numerosas variantes, lo que puede complicar el alineamiento de secuencias. Esto se puede observar en el Ejemplo 1 y en el Ejemplo 2.

```
11401 gcacgcccac ctgctggcag ctggggacac tgccgggccc tcttgctcca acagtactgg
11461 cggattatag ggaaacaccc ggagcatatg ctgtttggtc tcagtagact cctaaatatg
11521 ggattcctgg gtttaaaagt aaaaaataaa tatgtttaat ttgtgaactg attaccatca
11581 gaattgtact gttctgtatc ccaccagcaa tgtctaggaa tgcctgtttc tccacaaagt
```

Ejemplo 1 Sección del cromosoma 1 del genoma toplevel de Ensembl sin enmascarar

```
11401 nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnca acagtactgg
11461 cggattatag ggaaacaccc ggagcatatg ctgtttggtc tcagnnnnnn nnnnnnnnnn
11521 nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn
11581 nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn nnnnnnnnnn
```

Ejemplo 2 Sección del cromosoma 1 del genoma toplevel de Ensembl enmascarado.

3.3.3.2 Comparativa entre GRCh37 y GRCh38

Para el *variant calling*, los genomas de referencia más utilizados son GRCh37/hg19 y GRCh38/hg38. Aunque también existe el genoma de referencia *Telomere to Telomere* (T2T), se ha optado por no utilizarlo en este proyecto debido a su reciente aparición y falta de adopción generalizada en la comunidad científica.

La versión GRCh37/hg19 ha sido el genoma de referencia más empleado y ha servido como base para numerosos estudios que han producido una gran cantidad de datos. Sin embargo, presenta varias regiones que no están representadas con precisión. Para solucionar este problema de representación, se creó la versión GRCh38/hg38; la cual incluye actualizaciones y mejoras que abordan las imprecisiones presentes en hg19. Dado que hg38 es una versión más reciente, se han realizado menos investigaciones que con hg19.

Un estudio (Pan et al., 2019) comparó las variantes identificadas utilizando hg19 y hg38. Los resultados mostraron que las tasas de conversión de hg38 a hg19 eran más bajas que las tasas de conversión de hg19 a hg38. Alrededor del 1,5% de las variantes se convirtieron de manera discordante entre hg19 y hg38.

Por lo tanto, la elección entre estos dos genomas de referencia depende de los requisitos específicos del caso de uso. A primera vista, podría parecer obvio usar hg38, ya que es la versión más precisa y completa. Sin embargo, si el proyecto implica comparar los resultados con conjuntos de datos generados con hg19, entonces, usar hg19 podría ser más apropiado.

3.3.3.3 Versiones de GRCh38 y sus características

El genoma de referencia GRCh38 es publicado por el **Consorcio de Referencia del Genoma** (*Genome Reference Consortium*). Este tiene varios elementos clave que están explicados en (*Reference Genome Components*, 2024). Estos se explicarán a continuación.

Primero, un *contig* es una secuencia de ADN que se ha ensamblado a partir de lecturas de secuenciación. Estos *contigs* actúan como los bloques de construcción del genoma de referencia completo.

Por otro lado, los gaps en el genoma de referencia representan regiones que no han sido secuenciadas o que están incompletamente ensambladas. Estos gaps a menudo se deben a la complejidad inherente de ciertas regiones genómicas, y se encuentran principalmente en telómeros, centrómeros y secuencias largas y repetitivas.

También se pueden encontrar las secuencias alternativas (Alt), estas son representaciones de variantes genómicas y proporcionan información valiosa sobre la diversidad genómico en la población.

Es importante destacar que el genoma de referencia GRCh38/h38, aunque estable, no es inmutable. Se actualiza periódicamente para reflejar los avances en la comprensión del genoma humano y actualmente está en la versión GRCh38.p14. Estas actualizaciones pueden tomar varias formas:

- Los parches FIX implican cambios en las secuencias de ensamblaje existentes para corregir errores o mejorar el ensamblaje
- Los parches NOVEL, por otro lado, implican la adición de nuevas regiones alternativas al ensamblaje.

3.3.4 Regiones de análisis y tipos de secuenciación en NGS

En la etapa de *variant calling* y en la secuenciación, algunas veces se pasan por alto ciertas regiones del genoma. Estas suelen ser las secciones no codificantes, los centrómeros, las regiones repetitivas, los telómeros, las regiones intergénicas o las áreas con secuencias altamente variables.

Teniendo en cuenta lo anteriormente mencionado, es posible focalizar tanto el *variant calling* como la secuenciación en los exomas. La secuenciación de solo los exomas, que son las regiones del genoma que codifican proteínas, es de gran interés por un enfoque basado en la eficiencia dado que los exomas constituyen aproximadamente el 2% del genoma humano, pero contienen la mayoría de las variantes genómicas conocidas por causar enfermedades (Petersen et al., 2017).

Por lo tanto, dependiendo de las regiones de interés o si todo el genoma es secuenciado se pueden diferenciar dos tipos de secuenciaciones, WES (Whole Exome Sequencing) y WGS (Whole Genome Sequencing).

La secuenciación de este Trabajo Final de Máster es WES. Según los archivos de estratificación proporcionados por el proyecto Genome in a Bottle (GIAB) y utilizados en el Truth Challenge V2 de PrecisionFDA, el 99.86% de los exomas están en regiones difíciles de mapear. Aproximadamente 300 millones de bases componen las regiones difíciles de mapear, de las cuales 45 millones son exomas.

Es importante señalar que, para obtener resultados de máxima calidad, no se debe subestimar la importancia de las regiones excluidas del genoma, ya que pueden influir significativamente en la activación o desactivación de genes y en la organización cromosómica.

3.4 CODIFICACIÓN DE SECUENCIAS

En esta sección, se explorarán los formatos más importantes para representar datos genómicos, como FASTA y FASTQ, entre otros. Estos formatos basados en texto están estandarizados y permiten almacenar y analizar secuencias de ADN y ARN. También existen otros formatos donde se anotan

características genómicas y variantes. La comprensión de estos es imperativa si se quieren entender estos procesos con profundidad.

3.4.1 Formato FASTA

El formato FASTA, que se puede observar en el Ejemplo 3, representa las secuencias de nucleótidos o aminoácidos, utilizando códigos de una sola letra. Comienza con un carácter "mayor que" (">") seguido de una descripción, y las líneas siguientes contienen la secuencia. Suele usarse para codificar el genoma de referencia.

```
>KI270394.1 dna:scaffold scaffold:GRCh38:KI270394.1:1:970:1
REF
AAGTGGATATTTGGATAGCTTTGAGGATTTCGTTGGAAACGGGATTACATATAAAATCTA
GAGAGAAGCATTCTCAGGAACTTCTTTGTGATGTTTGCATTCAAGTCACAGAACTGAACA
TTCCCTTTCATAGAGCAGGTTTGAAACACTCTTTCTGTAGTATCTGCAAACGGACATTTC
ATACGCTTTCAGGCCTATGGTGAGAAAGGAAATATCTTCAAATAAAAACTAGACAGAAGC
```

Ejemplo 3 Sección del genoma de referencia.

Tomando el ejemplo anterior en la Tabla 3 se puede ver como cada letra se corresponde con un ácido nucleico.

Código de ácido nucleico	Significado
A	**A**denosina
C	**C**itosina
G	**G**uanina
T	**T**imina
U	**U**racilo

Tabla 3 Tabla de la correspondencia de las letras y los ácidos nucleicos.

3.4.2 Formato FASTQ

El formato FASTQ al igual que el formato FASTA almacena una secuencia biológica, generalmente una secuencia de nucleótidos. Originalmente, se desarrolló en el Wellcome Trust Sanger Institute para combinar una secuencia formateada en FASTA con sus datos de calidad, recientemente se ha convertido en el estándar de facto para almacenar la salida de instrumentos de NGS.

Un archivo FASTQ como el que se puede ver en el Ejemplo 4 consta de cuatro campos separados por líneas en cada secuencia:

- Campo 1: Comienza con el carácter "@" y va seguido de un identificador de secuencia y una descripción opcional.
- Campo 2: Contiene las letras de la secuencia en bruto.
- Campo 3: Comienza con el carácter "+" y opcionalmente va seguido del mismo identificador de secuencia (y cualquier descripción) nuevamente.
- Campo 4: Codifica los valores de calidad para la secuencia del segundo campo y debe contener el mismo número de símbolos que letras en la secuencia.

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

Ejemplo 4 Secuencia de nucleótidos en un archivo FASTQ.

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
```

Ejemplo 5 Símbolos de menor a mayor calidad para un FASTQ.

3.4.3 Formato BED

El formato *Browser Extensible Data* o BED, que se puede observar en el Ejemplo 6, es utilizado para almacenar regiones genómicas como coordenadas y anotaciones asociadas. Los datos se presentan en forma de columnas separadas por espacios o tabulaciones.

Consta de un mínimo de tres columnas a las que se pueden agregar nueve columnas opcionales para un total de doce columnas. Las primeras tres columnas contienen los nombres de los cromosomas, el inicio y las coordenadas finales de las secuencias consideradas.

```
chr7    127471196    127472363
chr7    127472363    127473530
chr7    127473530    127474697
```

Ejemplo 6 Líneas de un archivo BED.

3.4.4 Formato BAM

El formato *Binary Alignment Map* o BAM almacena alineaciones de secuencias de nucleótidos de forma compacta y con índices. Esto permite un almacenamiento eficiente y acceso rápido a regiones

específicas del genoma, facilitando la identificación de diferencias entre el genoma secuenciado y el de referencia, útil para la llamada de variantes.

Un archivo BAM se compone de dos secciones principales:

- **Encabezado:** Esta sección almacena información como el nombre de la muestra, su longitud y el método de alineación utilizado. Las alineaciones de la siguiente sección se vinculan a esta información para su interpretación.
- **Alineamientos:** Esta sección contiene información detallada de cada lectura (fragmento de ADN secuenciado): su nombre, secuencia, calidad, datos de alineación y etiquetas personalizadas. El nombre de la lectura suele incluir el cromosoma, coordenada inicial, calidad de alineación y una cadena descriptiva de las coincidencias.

3.4.5 Formato VCF

El formato *Variant Call Format* o VCF (Danecek et al., 2011) es un estándar utilizado para representar llamadas de variantes de SNP, indel y variantes estructurales.

Un archivo VCF válido está compuesto por dos partes principales, el encabezado y los registros de llamadas de variantes.

El encabezado de un archivo VCF generalmente comienza con una etiqueta que indica la versión que sigue el archivo. Además, incluye información sobre datos y fuentes de referencia relevantes, como el organismo y la versión de construcción del genoma de referencia como se observa en el Ejemplo 9. También pueden encontrarse líneas contig, como las del Ejemplo 8, que contienen los nombres de los contigs y sus longitudes.

Asimismo, se definen las anotaciones usadas para calificar y cuantificar las propiedades de las llamadas de variantes del archivo VCF. Las líneas FILTER, las cuales se pueden ver un en el Ejemplo 7, en el archivo VCF indican los filtros aplicados a los datos. A veces, los encabezados de los VCF generados pueden incluir la línea de comandos utilizada para generarlos, aunque no es un requisito obligatorio. Por último, en el encabezado se pueden encontrar las líneas FORMAT e INFO que definen las anotaciones contenidas en las columnas FORMAT e INFO del archivo VCF. Estas varían según la técnica de llamada de variantes utilizada.

```
##FILTER=<ID=DRAGENSnpHardQUAL,Description="Set if true:QUAL < 3.0103">
##FILTER=<ID=DRAGENIndelHardQUAL,Description="Set if true:QUAL < 3.0103">
##FILTER=<ID=LowDepth,Description="Set if true:DP <= 1">
##FILTER=<ID=LowGQ,Description="Set if true:GQ = 0">
```

Ejemplo 7 Líneas filter de un VCF.

```
##contig=<ID=chrUn_GL000216v2,length=176608>
##contig=<ID=chrUn_GL000218v1,length=161147>
##contig=<ID=chrEBV,length=171823>
```

Ejemplo 8 Líneas contig de un VCF.

```
##reference=file:///data/scratch/hg38-alt_masked.cnv.graph.hla.rna-9-
1679615250-1.v9/DRAGEN/9/reference.bin
```

Ejemplo 9 Línea del genoma de referencia de un VCF.

Después de las líneas de encabezado y los nombres de los campos se pueden encontrar los registros de llamadas de variantes. Cada línea representa una única variante, con varias propiedades de esa variante representadas en las columnas.

#CHROM	POS	ID	REF	ALT	QUAL	FILTER
chr1	729454	.	T	G	12.35	PASS
chr1	826893	.	G	A	46.79	PASS
chr1	826940	.	C	T	29.53	PASS

Ejemplo 10 Campos de un VCF.

Estas propiedades se componen del cromosoma (CHROM), la posición (POS), el identificador (ID), el alelo de referencia (REF), el alelo alternativo (ALT), la calidad (QUAL), el filtro (FILTER) y la información adicional (INFO).

3.5 IDENTIFICACIÓN DE VARIANTES.

Después de explicar cómo se obtienen y codifican las secuencias, se va a explicar la identificación de variantes, que se divide en dos partes: preprocesado e identificación.

En el preprocesado, las secuencias se alinean basándose en un genoma de referencia y se recalibran los datos. Luego, en la identificación de variantes, se utilizan diversas herramientas, cada una con su propio método de funcionamiento.

3.5.1 Alineadores

Para el análisis de variantes, las lecturas codificadas en un archivo FASTQ deben alinearse utilizando alineadores. Estos alineadores tienen la función principal de alinear las secuencias de lectura corta generadas por las máquinas de NGS con una secuencia de referencia, permitiendo así identificar la ubicación exacta de cada lectura. En esta sección, se mencionarán algunos de los alineadores más utilizados y sus características.

3.5.1.1 DRAGMAP

Un alineador muy utilizado es DRAGMAP (Illumina, s. f.). Este es una implementación del alineador de DRAGEN, que el equipo de Illumina creó para tener una forma de código abierto de producir los mismos resultados que en su software propietario DRAGEN. Según los autores DRAGMAP está destinado a reemplazar a BWA como alineador dado que tiene un rendimiento mayor como se puede ver en la Figura 7. Es importante mencionar que, dado que es una versión alfa es posible que existan errores. Por desgracia, en la documentación no se proporciona una explicación detallada sobre su funcionamiento específico. Solo se menciona su rendimiento.

Tras investigar las alineaciones de BWA, los autores de DRAGMAP encontraron que muchos falsos positivos caen en regiones del genoma donde hay secuencias alternativas que comparten mucha secuencia con las secuencias primarias. Esto plantea un desafío para el alineador del genoma, que tiene que decidir si tiene más sentido alinear las lecturas a las secuencias primarias o a las secuencias alternativas (Introducing DRAGMAP, the New Genome Mapper in DRAGEN-GATK, s.f.).

Por lo tanto, para solucionar esto DRAGMAP utiliza una estrategia de enmascaramiento del genoma de referencia y en todas las pruebas se demostró una precisión superior en comparación con los enfoques anteriores, con menos falsos positivos en la llamada de variantes subsiguientes. Esta implementación contribuyó al rendimiento de DRAGEN en el Precision FDA Truth Challenge 2 en lecturas de Illumina.

Cabe destacar que las pruebas de DRAGMAP se centraron en muestras disponibles del proyecto GIAB, y no se evaluaron específicamente en genomas de poblaciones que exhiben mayor diversidad, por lo que puede haber limitaciones aún no reconocidas.

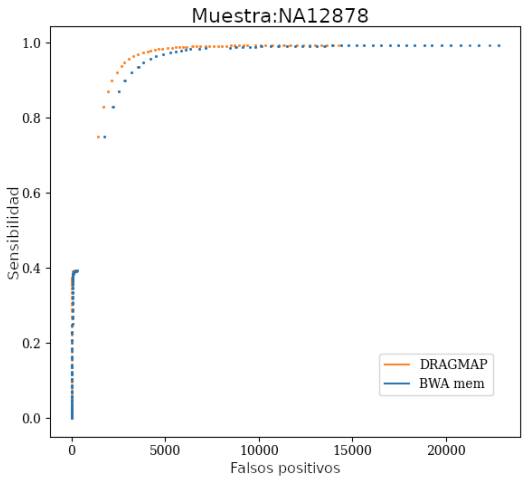

Figura 7 Curva ROC modificada que compara BWA Y DRAGMAP (Introducing DRAGMAP, the New Genome Mapper in DRAGEN-GATK, s.f.).

3.5.1.2 Borrow Wheeler Aligner

Borrow Wheeler Aligner o BWA es un alineador desarrollado por (Li & Durbin, 2009). Es ampliamente utilizado y es considerado el estado del arte en la alineación de lecturas en NGS. En la publicación original los autores compararon el rendimiento de BWA con otros alineadores, como Bowtie, MAQ y SOAP2, utilizando datos sintéticos generados mediante WGSSIM. Estas lecturas simuladas incluyeron una tasa de mutación del 0.02% para indels, un 0.09% para SNPs y un 2% de error de lectura. Los resultados revelaron que BWA superaba a los otros alineadores en todos los casos evaluados.

Este alineador toma el genoma de referencia y aplica una transformación conocida como *Burrows-Wheeler Transform* o BWT. La transformación se realiza ordenando todas las rotaciones circulares de un texto en orden alfabético y extrayendo la última columna y el índice de la cadena original en el conjunto de permutaciones ordenadas de S.

Concretamente dada una cadena de texto de entrada S = ^BANANA$, para obtener la transformación, esta cadena rota N veces, donde N = 8 y es la longitud de la cadena, considerando también el carácter ^ que representa el inicio de la cadena y el carácter $ que representa el puntero el final de la cadena.

Por último, estas rotaciones, o desplazamientos circulares, se ordenan por orden alfabético. Esta transformación se puede apreciar en la Tabla 4.

Transformación				
1. Input	**2. Todas las rotaciones**	**3. Ordenación por orden alfabético**	**4. Obtenemos la última columna**	**5. Output**
^BANANA$	^BANANA$	**$^**BANANA	$^BANAN**A**	A$NNB^AA
	$^BANANA	^BANANA$	^BANANA**$**	
	A$^BANAN	**A**$^BANAN	A$^BANA**N**	
	NA$^BANA	**A**NA$^BAN	ANA$^BA**N**	
	ANA$^BAN	**A**NANA$^B	ANANA$^**B**	
	NANA$^BA	**B**ANANA$^	BANANA$^	
	ANANA$^B	**N**A$^BANA	NA$^BAN**A**	
	BANANA$^	**N**ANA$^BA	NANA$^B**A**	

Tabla 4 Pasos seguidos para realizar la transformación de Burrows-Wheeler (Wikipedia contributors, 2024).

Después de transformar el genoma, se realiza la búsqueda de una secuencia. Suponiendo que el genoma completo es “acaacg$”, su transformada BWT es “gc$aac” y que se quiere buscar la secuencia “aac”.

El proceso de búsqueda comienza desde el final del patrón a buscar. Se buscan coincidencias entre el último valor del patrón y el primero de las cadenas, así como entre el primer valor del patrón y el último de las cadenas. Si no hay una coincidencia completa, se repite la secuencia actual, actualizando el último valor a buscar. A continuación, en la Tabla 5 se puede ver un ejemplo simplificado.

Paso 1, Query aac	Paso 2, Query aac	Paso 3, Query aac
$acaacg	$acaacg	$acaacg
aacg$ac	aacg$ac	**aacg$ac**
acaacg$	**acaacg$**	acaacg$
acg$aca	**acg$aca**	acg$aca
caacg$a	caacg$a	caacg$a

cg$acaa	cg$acaa	cg$acaa
g$acaac	g$acaac	g$acaac

Tabla 5 Pasos a seguir para encontrar una secuencia S siendo un subconjunto de B.

3.5.2 Recalibración con GATK4

Además de los alineadores, también son importantes técnicas como MarkDuplicates, BaseRecalibrator y BQSR. Estas herramientas forman parte de GATK4 (Van Der Auwera et al., 2013). A continuación, se explicarán en qué consisten estas técnicas.

GATK MarkDuplicates identifica y etiqueta lecturas duplicadas en un archivo BAM o SAM. Las lecturas duplicadas se definen como aquellas que provienen de un solo fragmento de ADN. Después de recoger las lecturas duplicadas, la herramienta diferencia las lecturas primarias y duplicadas utilizando un algoritmo que clasifica las lecturas por la suma de sus puntuaciones de calidad de base.

Después se puede encontrar a GATK BaseRecalibrator. Esta herramienta genera una tabla de recalibración basada en varias covariables. Hace un recorrido operando solo en sitios que están en el VCF de sitios conocidos, es decir, en el archivo dbSNP.

Por último, GATK ApplyBQSR. Esta herramienta realiza el segundo paso en un proceso de dos etapas que empieza con BaseRecalibrator. Específicamente, recalibra las calidades de base de las lecturas de entrada basándose en la tabla producida por la herramienta BaseRecalibrator, y genera un archivo BAM o CRAM recalibrado.

3.5.3 Técnicas de llamadas de variantes

Para comprender la etapa de *variant calling*, una parte importante son las técnicas de llamadas de variantes. Estas técnicas se emplean para identificar variantes genómicas, para esto se emplean los archivos BAM como entrada. En esta sección se comentarán cómo funcionan algunas de estas técnicas, las cuales son DeepVariant, FreeBayes, Strelka2 y HaplotypeCaller.

3.5.3.1 DeepVariant

DeepVariant (Poplin et al., 2018) es una técnica de llamada de variantes desarrollada por Google basada en redes convolucionales. En esta sección se explicará que es una red convolucional y luego cual es la arquitectura que han empleado y cómo se codifican los datos para esta.

Las redes convolucionales son un tipo de arquitectura de red neuronal diseñada específicamente para procesar datos con una estructura de cuadrícula, como imágenes o videos.

La operación de convolución de la que se puede ver un ejemplo en Figura 8 es fundamental en estas redes. Consiste en aplicar filtros (*kernels*) a regiones locales de la entrada para extraer características relevantes, como bordes, texturas y patrones. Estos filtros se deslizan sobre la imagen y generan mapas de características.

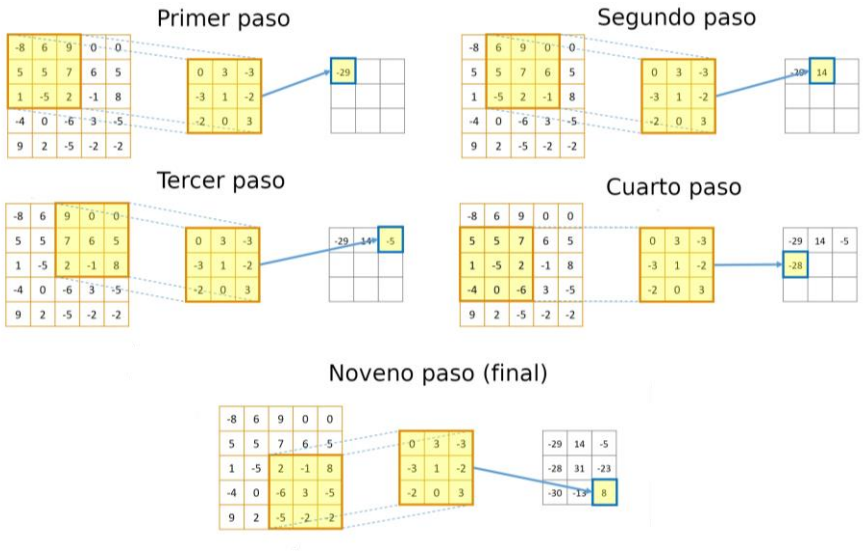

Figura 8 Operaciones y orden de una convolución.

La unión de varios de estos *kernels* genera una capa convolucional. Estas capas se combinan con capas de activación, que pueden ser sigmoides, tangentes hiperbólicas o las más común la ReLU, de esta última derivan otras muchas. Estas activaciones introducen no linealidad en el modelo. Dentro de una misma red suelen usarse diferentes activaciones. No se entrará en detalle dado que varía mucho según el propósito de la red y el tipo de entrada de esta, pero por simplificar y dado que DeepVariant es un clasificador, estas redes cuentan con dos tipos de activaciones la final y el resto.

Si la tarea busca clasificar dos clases la capa final suele ser una Sigmoide, especificada en la Ecuación 2. Sin embargo, para clasificaciones de más de una clase, se utiliza la Softmax. Cabe destacar que el uso de algunas de estas activaciones, sobre todo la ReLU que se puede observar en la Ecuación 1, o aquellas que deriven de ella, puede implicar en una explosión del gradiente, por lo que suelen ir acompañadas de capas como *BatchNormalization* o *LayerNormalization*.

$$ReLU(x)\begin{cases} x\ si\ x \geq 0 \\ 0\ si\ x < 0 \end{cases}$$

Ecuación 1 Función de activación Rectified linear unit (ReLU).

$$Sigmoide(x) = \frac{1}{1+e^{-x}}$$

Ecuación 2 Función de activación Sigmoide.

Después de estas capas y activaciones, se pueden utilizar capas de *pooling* que reducen la resolución espacial de los mapas de características. Esto ayuda a disminuir la cantidad de parámetros y a capturar características que son invariantes a pequeñas traslaciones. Un ejemplo de estas capas se puede ver en la Figura 9.

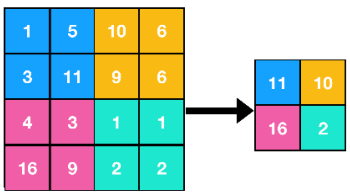

Figura 9 Representación de la operación de max pooling.

También pueden incluir una capa o varias de *Dropout.* Estas capas 'apagan' algunas neuronas de manera aleatoria esto se logra poniendo a cero algunos mapas de características a la entrada de la capa. Estas capas se usan para reducir el sobreajuste de la red.

La optimización de los pesos en una red neuronal, esencialmente cómo la red "aprende" para llevar a cabo una tarea, se logra mediante un método de optimización llamado descenso del gradiente, aplicado a través de un proceso conocido como retropropagación. Este proceso se realiza automáticamente si la función de coste es diferenciable.

El gradiente es un vector que contiene todas las derivadas parciales de la función de pérdida con respecto a los parámetros. Este vector es crucial para la actualización de los parámetros durante el entrenamiento, utilizando técnicas como el descenso del gradiente.

Además, en el descenso del gradiente la tasa de aprendizaje es un hiperparámetro de gran importancia. Esta tasa determina el tamaño del paso en cada iteración mientras se busca el mínimo de la función de coste. Una tasa de aprendizaje demasiado alta puede llevar a la divergencia, y una baja puede hacer que el entrenamiento sea demasiado lento. Para este tipo de modelos los marcos de trabajo populares son TensorFlow y PyTorch. Proporcionan operaciones de autodiferenciación, facilitando enormemente este proceso.

Ahora de manera más concreta la arquitectura usada en DeepVariant es la de Inception v3, esta consta de múltiples módulos llamados "Módulos Inception". Estos módulos contienen una combinación de filtros convolucionales de 1x1, 3x3, concatenaciones de capas y capas de poolings. Además, hacia el final de la red también cuenta con capas de *BatchNormalization* y un *Dropout*.

En la actualidad esta arquitectura está desfasada, y existen otro tipo de alternativas para este tipo de problemáticas. Aun así, puede ser que se haya especificado Inception V3 y luego por algún motivo se haya usado una versión modificada de este implementando algunos elementos más innovadores como por ejemplo el *BatchNormalization/LayerNormalization* tras cada capa, o incluso otras activaciones como la Gelu.

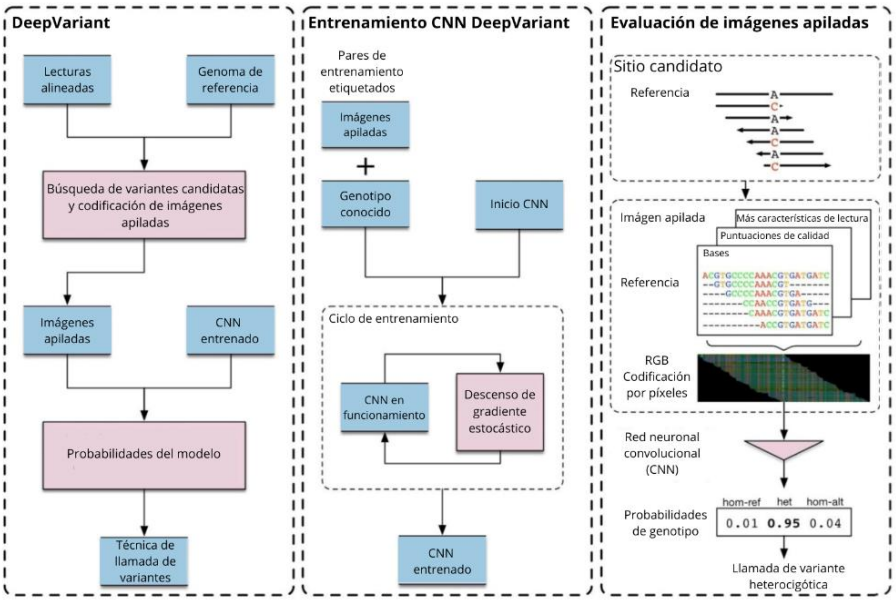

Figura 10 Caja izquierda: Visión general de DeepVariant. Caja del medio: Entrenamiento de DeepVariant. Caja de la derecha: Inferencia.

DeepVariant tiene 3 etapas. En la primera etapa, las lecturas de NGS se alinean primero con un genoma de referencia y se limpian con un marcado de duplicados y, opcionalmente, un reensamblaje local. En la etapa intermedia es cuando la red neuronal realiza su clasificación para calcular las probabilidades de genotipo para los tres estados de genotipo diploide de referencia homocigota (hom-ref), heterocigoto (het) o alternativo homocigota (hom-alt). Por último, se interpretan las clasificaciones para así producir los VCFs.

Entre la primera y la segunda etapa las lecturas alineadas se escanean en busca de sitios que pueden ser diferentes del genoma de referencia. Además, los datos de lectura y referencia se codifican como un tensor de ocho canales que se puede observar en la Figura 11.

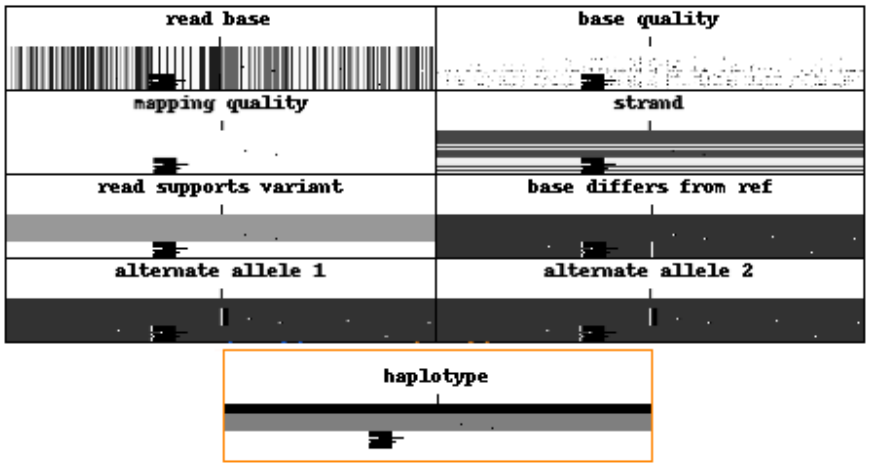

Figura 11 Canales de entrada para DeepVariant (Cook, 2021).

3.5.3.2 FreeBayes

Freebayes (Garrison & Marth, 2012) usa el teorema de Bayes el cual se explica en la Ecuación 3.

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

Ecuación 3 Teorema de Bayes.

Donde $P(B|A)$ es al a posteriori $P(A|B)$ es la probabilidad condicional $P(A)$ es el a priori. A continuación, se explicará cómo se aplica esto dadas unas secuencias para calcular el genotipo dado un lugar y así poder identificar variantes.

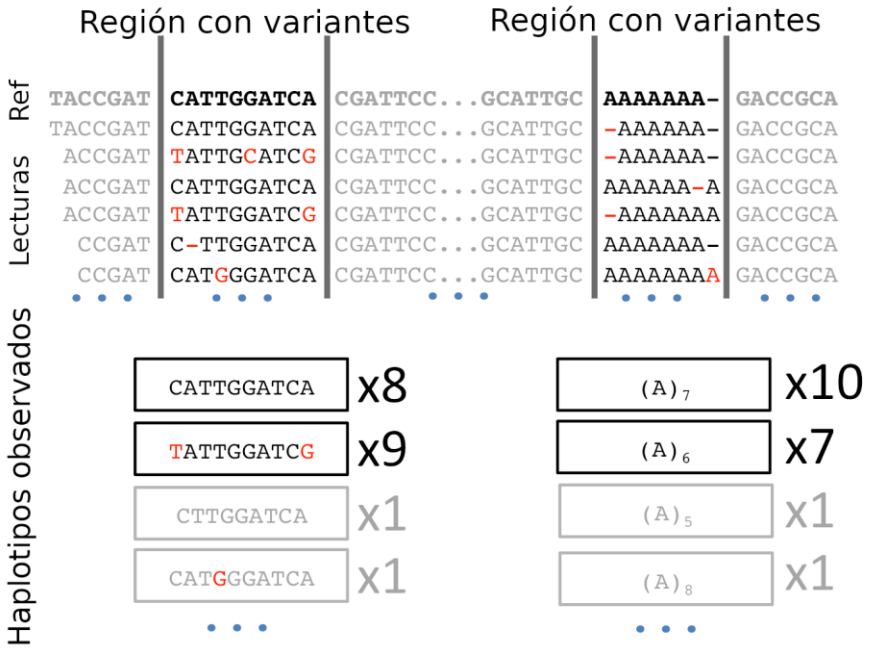

Figura 12 Imagen modificada que ilustra Freebayes (Freebayes, s.f.).

Antes de empezar con la explicación hace falta definir la nomenclatura que se usara en las siguientes formulas. En un lugar genómico dado, se tienen *n* muestras extraídas de una población, cada una de las cuales tiene un número de copias o multiplicidad *m* dentro del lugar. Se denomina el número de copias de los lugares presentes en nuestro conjunto de muestras como $M = \sum_{i=1}^{n} m_i$. Entre estas *M* copias, se tiene *K* alelos distintos, $b_1, \dots b_k$, con frecuencias alélicas $f_1, \dots f_k$. Cada individuo tiene un genotipo no emparejado G_i compuesto por k_i alelos distintos, $b_{i_1}, \dots b_{k_i}$, y las correspondientes frecuencias alélicas $f_{i_1}, \dots f_{k_i}$, que también pueden expresarse de manera equivalente como un multiconjunto de alelos B_i: $|B| = m_i$. Suponiendo un conjunto s_i de observaciones de secuenciación $r_{i_1}, \dots r_{i_{s_i}} = R_i$ cada muestra en nuestro conjunto de muestras, de modo que hay $\sum_{i=1}^{n} s_i$ lecturas en el lugar genómico bajo análisis. Q_i denota la calidad de mapeo, o la probabilidad de que la lectura r_i esté mal mapeada con respecto a la referencia.

Primero lo que hace es generarse unos haplotipos. Para esto usa el teorema de Bayes de la formula anterior, a nuestro caso de uso se podría entender con la Ecuación 4.

$$P(\, G_1, \dots, G_n \mid R_1, \dots, R_n \,) = \frac{P(G_1, \dots, G_n) \prod_{i=1}^{n} P(R_i \mid G_i)}{\sum_{\forall (G_1, \dots, G_n)} \left(P(G_1, \dots, G_n) \prod_{i=1}^{n} P(R_i \mid G_i) \right)}$$

Ecuación 4 Teorema de bayes en Freebayes.

Dentro de la formula anterior si se desarrolla la probabilidad condicional $P(R_i \mid G_i)$, esta modelada como una distribución multinomial con un elemento corrector en el que se introducen las calidades del mapeo siendo este $P(b_l'|b_l)$. Desarrollando este elemento se obtiene que $P(b_l'|b_l) = 1 - qi$.

$$P(R_i \mid G_i) = \sum_{\forall(R_i \in G_i)} \left(\frac{s_i!}{\prod_{j=1}^{k_i} o_j'!} \prod_{j=1}^{k_i} \left(\frac{f_{i_j}}{m_i}\right)^{o_j'} \prod_{l=i}^{s_i} P(b_l'|b_l) \right)$$

Ecuación 5 Definición del a posteriori en Freebayes.

Si se sigue con la formula inicial faltarían los a prioris $P(G_1, \dots, G_n)$. La probabilidad de una combinación de genotipos dada es equivalente a la intersección de esa probabilidad y la probabilidad del conjunto correspondiente de frecuencias alélicas. Esto se deriva del hecho de que las frecuencias alélicas se obtienen a partir del conjunto de genotipos y siempre se tendrán los mismos $f_1, \dots f_k$ para cualquier $G_1, \dots, G_n$equivalente. Siguiendo la Regla de Bayes, esta identidad se descompone aún más.

$$P(G_1, \dots, G_n \cap f_1, \dots, f_k) = P(G_1, \dots, G_n \mid f_1, \dots, f_k) P(f_1, \dots, f_k)$$

Ecuación 6 Definición del a priori en Freebayes.

Siendo las probabilidades del genotipo dadas las frecuencias alélicas:

$$P(G_1, \dots, G_n \mid f_1, \dots, f_k) = \frac{1}{M!} \prod_{l=1}^{k} f_l \prod_{i=1}^{n} \frac{m_i!}{\prod_{j=1}^{k_i} f_{i_j}!}$$

Ecuación 7 Probabilidades del genotípico dadas las frecuencias alélicas.

Como en nuestro caso se están buscando las variantes germinales estas frecuencias serian el caso bialélico en el que nuestro conjunto de muestras tiene dos alelos con frecuencias f_1 y f_2 tal que $f_1 + f_2 = M$ y siendo θ la ratio de mutaciones de la población.

$$P(f_1, \dots, f_k) = P(a_1, \dots, a_M) = P(a_{f1} = 1, a_{f2} = 1) = \frac{M!}{\prod_{z=1}^{M-1}(\theta + z)} \frac{\theta}{f_1 f_2}$$

Ecuación 8 Probabilidad de las frecuencias alélicas.

Todo esto lo hace en ventanas que se superponen en todos los valores es decir la ventana se mueve solo una base por iteración. Esta ventana es dinámica, y se aplica un conjunto de filtros de entrada para excluir alelos del análisis que sean altamente improbables. Estos filtros requieren un número mínimo de observaciones alternas y una suma mínima de calidades de bases en una sola muestra para incorporar un alelo y sus observaciones en el análisis.

Luego, se determina una longitud de haplotipo. Primero, determina el alelo que pasa los filtros de entrada y que es más largo en relación con la referencia. Luego, se analizan las observaciones de haplotipo de todas las alineaciones que se superponen completamente a esta ventana, encontrando el extremo derecho del alelo de haplotipo más largo que comienza dentro de la ventana. Esta posición más a la derecha se utiliza para actualizar la longitud de la ventana de haplotipo, y se ensambla un nuevo conjunto de observaciones de haplotipo a partir de las lecturas que se superponen completamente a la nueva ventana. Este proceso se repite hasta que el extremo derecho de la ventana no se superponga parcialmente con ninguna observación de haplotipo que pase los filtros de entrada. Este método convergerá siempre que las lecturas tengan una longitud finita y solo se utilicen las lecturas que se superponen completamente con la ventana de identificación en el análisis.

Una vez que se ha determinado una ventana para el análisis, se analizan todas las lecturas que se superponen completamente en observaciones de haplotipo que están ancladas en los límites de la ventana.

Dadas estas series de observaciones de secuenciación $r_{i_1}, \dots r_{i_{s_i}} = R_i$ y las verosimilitudes de datos $P(R_i \mid G_i)$ para cada muestra y genotipo posible derivado de los alelos putativos, se determina la probabilidad de polimorfismo en el lugar. Luego se establece una estimación a posteriori máxima del genotipo para cada muestra.

$$G_1, \dots, G_n = argmax_{G_i} P(R_i \mid G_i)$$

Ecuación 9 Obtención del genotipo más probable.

En cuanto a la identificación de alelos, no se especifica exactamente cómo se realiza, pero dados los pasos anteriores, se entiende que se hace una suposición bialélica cuando los genotipos logran suficiente evidencia, es decir, si las probabilidades para los dos genotipos más probables, en función de las lecturas, superan cierto umbral.

3.5.3.3 Strelka 2

Strelka2 (Kim et al., 2018) puede usarse tanto como para germinales y tumorales. Destaca en su eficiencia en tiempos con respeto a otras herramientas según los autores. Como este Trabajo Final de Máster trata sobre variantes germinales solo se tratará cómo funciona el modelo en estas. Un esquema de este se puede encontrar en la Figura 13.

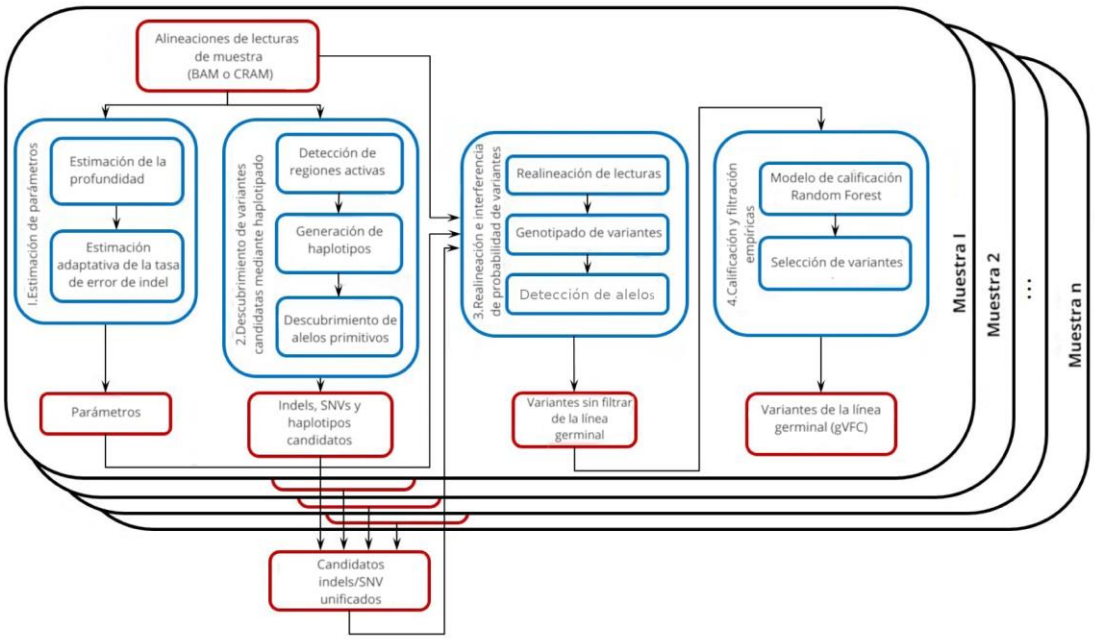

Figura 13 Esquema modificado del funcionamiento de Strelka 2 para variantes germinales (Kim et al., 2018).

Un paso inicial en todos los flujos de trabajo es la estimación rápida de la profundidad de secuenciación para cada cromosoma.

Los errores de secuenciación de indel se modelan como un proceso que ocurre de forma independiente en cada lectura. Estas probabilidades de error se estiman a partir de los datos de

secuenciación de cada muestra de entrada en dos pasos. Primero, los datos de secuenciación alineados se analizan para producir recuentos de errores. Segundo, los recuentos se utilizan para estimar los parámetros de interés.

A continuación**,** para cada indel se modela como perteneciente a un estado 'limpio' (que genera esencialmente cero errores indel) o un estado 'ruidoso' (que genera errores indel independientemente en las lecturas según un conjunto de probabilidades de error a ser estimadas), con las probabilidades generales de error derivadas de un modelo de mixtura de dos estados.

Los conteos de alelos, a su vez, se modelan como extraídos de una mezcla sobre posibles genotipos, con las distribuciones específicas de genotipo siendo multinomiales para genotipos homocigotos y mezclas de dos multinomiales para genotipos heterocigotos.

Una vez estén estimados estos parámetros, se puede comenzar a buscar las regiones activas. Estas regiones se evalúan en función de si antes o después de la variante observada hay un indel o si difiere de la referencia. Los indels anteriores y posteriores son los que más contribuyen a esta puntuación.

Un lugar con una puntuación de evidencia de variante c y una cobertura *d* se convierte en un lugar variante si:

- $c \geq 0.35d$
- $c \geq 9$ y $c \geq 0.2d$

Ecuación 10 Para la identificación de regiones activas en Strelka

Después se agrupan regiones activas cercanas si están a menos de 13 bases entre sí. Para agrupaciones con dos o más variantes, se extiende la región siguiendo la misma regla mencionada anteriormente.

Con las regiones activas ya identificadas siendo de 250 bases o menos, se intentan generar haplotipos utilizando dos modelos.

Sí más del 65% de todas las lecturas que se superponen con la región activa están completamente cubiertas se utiliza el modelo basado en alineamiento. En este para cada lectura se extrae el segmento alineado a la región activa como un haplotipo candidato.

Por el contrario, si menos del 65% de las lecturas que se superponen con la región activa son lecturas de cobertura se utiliza el modelo basado en ensamblaje. En este se ejecuta un ensamblaje local utilizando un enfoque de grafo de Bruijn.

Una vez obtenidos los haplotipos se clasifican según el soporte de lecturas, en orden decreciente. Se excluyen aquellos con menos de tres lecturas de soporte o que se encuentren por debajo del top x, donde x es la ploidía esperada de la muestra.

La prueba asume que el haplotipo candidato con el mayor soporte de lecturas, h1, es verdadero. Después se identifica si el haplotipo candidato con el siguiente nivel más alto de soporte de lecturas, h2, es perteneciente a un alelo alternativo del mismo cromosoma.

A continuación, se hace un barrido para seleccionar los candidatos a indel. Si el modelado de haplotipos está habilitado, un indel candidato que pertenezca a una región activa donde el haplotipado tuvo éxito también debe haber sido descubierto a través del alineamiento de haplotipos en al menos una muestra y debe tener al menos dos lecturas de soporte en al menos una muestra.

Además, se evalúa su estado de candidatura mediante una prueba exacta binomial unilateral, con la hipótesis nula de que la cobertura del indel es generada por procesos de error de indel. El indel se considera una variante candidata si el valor de P resultante es menor que 10^-9.

Tras identificar los indels candidatos, las lecturas se realinean a ellos. La realineación tiene dos funciones principales: generar alineaciones probables y crear una alineación representativa. El proceso de búsqueda de alineación utiliza una alineación inicial del archivo de entrada y un conjunto de indels candidatos que intersecan con la lectura. Una vez generadas las alineaciones probables, se evalúa la probabilidad de que cada variante (SNP o indel) sea real.

Para calcular la probabilidad de estos genotipos se usa el "*Shared probability model*" especificado en Strelka2 en el que se usan todos los parámetros, generados en los pasos anteriores.

Después se realiza la identificación de alelos de las variantes. Se pueden clasificar en diferentes casos:

- Homref: No variante (ambos alelos son de referencia).
- Het: Heterocigoto (un alelo de referencia y uno no de referencia).
- Homalt: Homocigoto (ambos alelos no son de referencia).
- Hetalt: Heterocigoto con dos alelos no de referencia.

Como último paso, se extrae información adicional en forma de un conjunto de características predictivas. Estas características se combinan con la probabilidad calculada por el modelo de llamada de variantes para mejorar la precisión de la llamada.

Esto se realiza mediante, un *random forest* entrenado con datos etiquetados de ejecuciones de secuenciación. Además, proporciona una puntuación de calidad agregada para cada variante. Existen dos modelos de bosque aleatorio entrenados, uno para las variantes de un solo nucleótido (SNPs) y otro para las inserciones/deleciones (indels).

3.5.3.4 HaplotypeCaller

HaplotypeCaller es una técnica de llamada a variantes creada por BOARD *institute* y está dentro de gatk4 (Van Der Auwera et al., 2013). En la Figura 14 se puede apreciar un esquema del funcionamiento de HaplotypeCaller.

Este se puede dividir en varios pasos que se comentaran a continuación. Previamente se debe realizar un mapeo de lecturas a la referencia, marcado de duplicados y una llamada al BQSR.

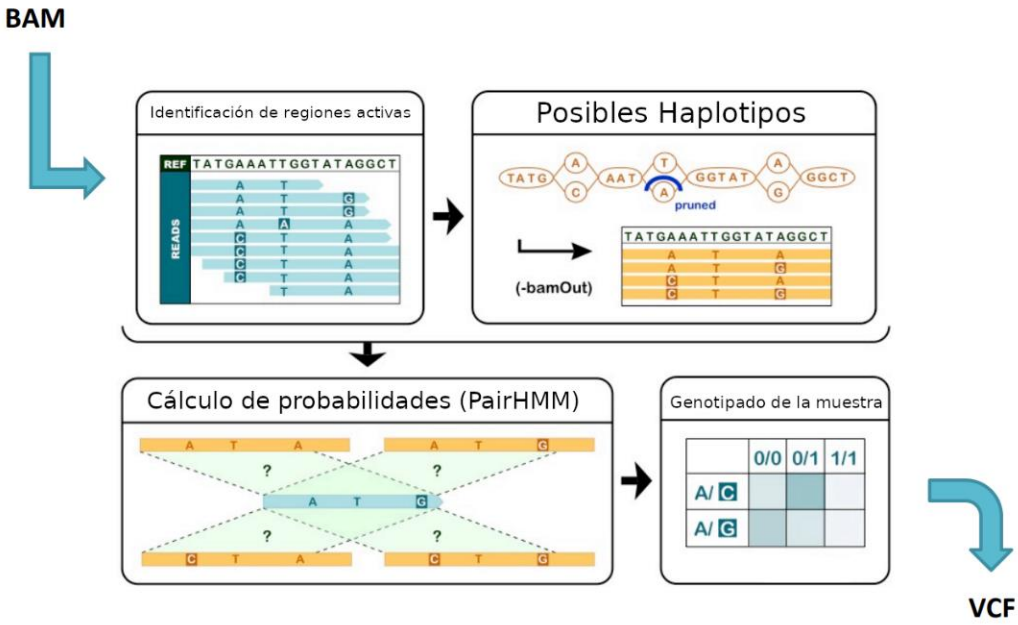

Figura 14 Esquema modificado del funcionamiento de HaplotypeCaller (Getting Started With GATK4, 2024).

Primero se deben identificar las regiones activas mediante una ventana deslizante, para identificar regiones en el genoma que probablemente contengan variantes, es decir, discrepancias entre las lecturas y la secuencia de referencia. Esto se hace con el objetivo de ahorrar costes computacionales.

Para la identificación se calcula un perfil de actividad para cada posición del genoma, este es un vector de ceros y unos. Cada valor de perfil se divide y se extiende utilizando un *kernel* gaussiano que abarca un radio de hasta 50 centrado en su posición actual, con una desviación estándar predeterminada en 17. Después se realiza un filtrado en el cual se recomienda quedarse con las regiones mayores a 0.

Después de la identificación de las regiones activas como se observa en la Figura 15 se realiza un número *N* de posibles haplotipos y se codifican como un grafo de Bruijn. Además, este grafo se aprovecha para realizar un realineamiento de las lecturas mediante el algoritmo de Smith Walterman.

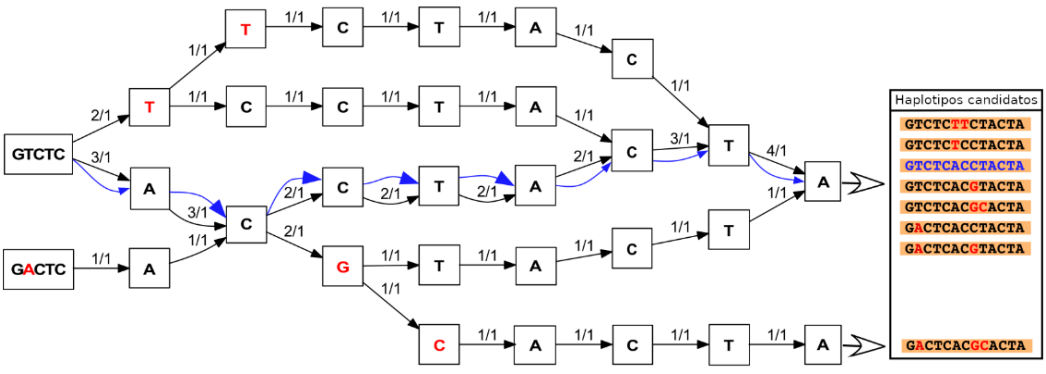

Figura 15 Esquema modificado del ensamblado de haplotipos mediante un grafo de Bruijn (Getting Started With GATK4, 2024).

Las alineaciones anteriores se ignoran, además si algún camino de este grafo solo está respaldado por una lectura se elimina.

A continuación, con estos haplotipos se realiza un Pair Hidden Markov Model o PairHMM para obtener las probabilidades de que esa lectura sea correcta. Los parámetros del PairHMM se obtienen del BQSR.

También existe una variante de este en el cual se incluye con DRAGMAP. Sobre el proceso entero no se encontró información detallada, pero presumiblemente DRAGMAP generará unos valores que luego se usarán en el PairHMM.

Antes de realizar el PairHMM que se puede observar en la Figura 16, primero se tiene que decidir qué columnas deben ser representadas por estados de deleción y qué columnas deben ser modeladas por estados de inserción.

Una regla simple sería comparar el número de símbolos y el número de huecos. Si la columna tiene más símbolos que huecos, se tratan los huecos como eliminaciones de símbolos. Por lo tanto, se modela la columna usando un estado de coincidencia *Mk* (para los símbolos en la columna dada) y un estado de eliminación *Dk* (para los huecos en la misma columna).

Por el contrario, si se tienen más huecos que símbolos, tendría más sentido ver los símbolos como inserciones, por lo que se usaría un estado de inserción Ik para representar la columna.

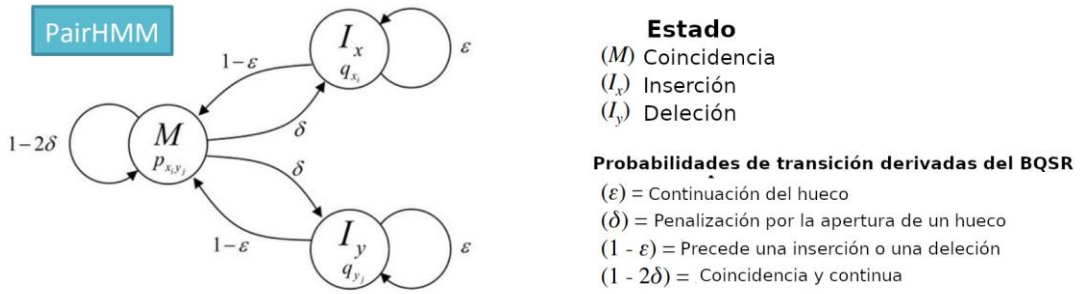

Figura 16 Esquema modificado del PairHMM y sus parámetros (Getting Started With GATK4, 2024).

Una vez que se han decidido qué columnas deben ser representadas por estados de coincidencia y cuáles deben ser representadas por estados de inserción, se conoce la secuencia de estados subyacente para cada secuencia de símbolos en la alineación.

Después se estiman las probabilidades de transición y las probabilidades de emisión del perfil-HMM contando el número de cada transición de estado o emisión de símbolo y calculando sus frecuencias relativas. Estos cálculos de probabilidades se hacen para cada haplotipo y lectura.

$$\text{Haplotipos}$$
$$\text{Lecturas} \begin{bmatrix} A_{1_2} & \dots & A_{1_n} \\ A_{2_2} & \dots & A_{1_n} \\ A_{3_2} & \dots & A_{3_n} \\ A_{n_2} & \dots & A_{n_n} \end{bmatrix}$$

Figura 17 Resultado del PairHMM para cada par haplotipo/lectura (Getting Started With GATK4, 2024).

Una vez se tienen las probabilidades se debería obtener una matriz donde las columnas son los haplotipos y las filas son las lecturas, es decir para cada haplotipo se compara con las N lecturas En la Figura 17 A_{i_j} es la probabilidad del haplotipo dada la lectura. En esta matriz se filtra obteniendo el máximo de lectura y las variantes siendo las columnas, esto se hace para una única variante.

Después para la identificación de alelos de los haplotipos se usa el Teorema de Bayes para cada fila de la matriz. Se escogen los dos valores máximos de las variantes, dado que, se asume la diploidía. En la Figura 18 se puede apreciar un ejemplo de esto último.

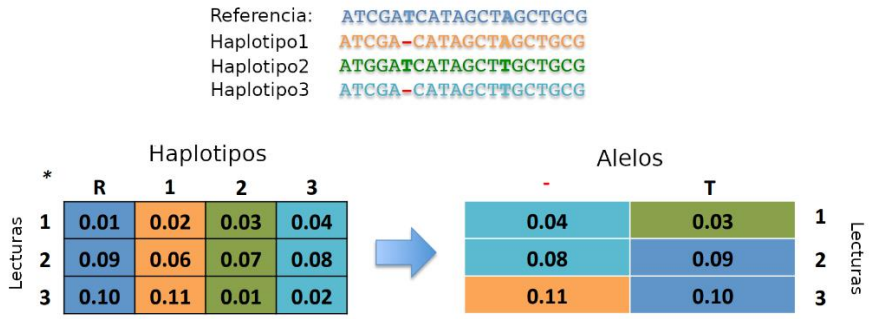

Figura 18 Ejemplo de selección de probabilidades para la identificación del genotipo (Getting Started With GATK4, 2024)

$$P(G|D) = \frac{P(G)P(D|G)}{\sum_i P(G_i)P(D|G_i)}$$

Ecuación 11 Teorema de Bayes.

El denominador es despreciable dado que será igual para todos los haplotipos y $P(G)$ es un valor fijo según los autores. Si asumimos un organismo diploide la probabilidad de $P(D|G)$ es la siguiente.

$$P(D|G) = \prod_J \left(\frac{P(D_J|H_1)}{2} + \frac{P(D_J|H_2)}{2}\right)$$

Ecuación 12 Probabilidad a priori de los alelos.

Este cálculo de probabilidad se realiza para cada posible genotipo basado en los alelos que se observan en el sitio, considerando todas las combinaciones posibles de alelos. Por ejemplo, si se observa una A y una T en un sitio, los genotipos posibles son AA, AT y TT, por lo tanto, se obtienen 3 probabilidades. Una vez conseguidas estas ya se tiene el genotipo más probable, y se asigna a la muestra.

3.6 RESPUESTAS A LAS PREGUNTAS DE INVESTIGACIÓN DEL PRIMER SUBOBJETIVO

Ahora a modo de resumen de la sección donde se trata el primer subobjetivo se responderán a las preguntas de investigación asociadas a este.

PI1: ¿Cómo se obtienen las secuencias de nucleótidos para el *variant calling*?

En la sección 3.3 se han definido y explicado las técnicas para obtener secuencias de nucleótidos, destacando las técnicas de Sanger y la secuenciación de nueva generación (NGS). Actualmente, el variant calling se realiza con secuencias obtenidas mediante NGS, un proceso que incluye varios pasos. Se ha tratado en detalle la secuenciación por síntesis, utilizada por Illumina, la empresa líder en el mercado. En este método, el ADN se fragmenta y se le añaden adaptadores generando una biblioteca. Luego, la biblioteca se desnaturaliza para formar cadenas individuales de ADN, que se fijan a una celda de flujo y se amplifican. Finalmente, se añaden nucleótidos fluorescentes y se secuencian las cadenas.

PI2: ¿Cómo se codifican las secuencias de nucleótidos para su análisis?

En la sección 3.4 se ha explicado cómo se codifican las secuencias de nucleótidos y los archivos usados como pasos intermedios, tales como FASTA, FASTQ, BAM y VCF. Estas codificaciones son un estándar y su comprensión ha sido necesaria para extraer información relevante, como la decodificación del filtrado de los VCFs. Las secuencias de nucleótidos se codifican para su análisis utilizando varios formatos estandarizados, cada uno con un propósito específico:

1. **FASTA**: Representa secuencias de nucleótidos o aminoácidos usando códigos de una sola letra. Comienza con un carácter “>” seguido de una descripción, y las líneas siguientes contienen la secuencia.
2. **FASTQ**: Combina la secuencia de nucleótidos con datos de calidad. Cada secuencia consta de cuatro campos: un identificador, la secuencia de nucleótidos, un separador “+” y los valores de calidad.
3. **BED**: Almacena regiones genómicas como coordenadas y anotaciones asociadas. Presenta los datos en columnas separadas por espacios o tabulaciones.
4. **BAM**: Es una versión comprimida y binaria de los archivos SAM, que almacena alineaciones de secuencias de nucleótidos de forma compacta y con índices, permitiendo un acceso rápido a regiones específicas del genoma.
5. **VCF**: Representa llamadas de variantes de SNP, indels y variantes estructurales. Incluye un encabezado con información sobre datos y fuentes de referencia, seguido de registros de llamadas de variantes con varias propiedades observadas en el sitio de la variante.

PI3: ¿Cómo se identifican las variantes genómicas?

En las secciones 3.5 se han explicado algunas de las herramientas más utilizadas para la identificación de variantes, el preprocesamiento de las secuencias de nucleótidos y la extracción de información sobre la calidad de estas.

La identificación de variantes genómicas se realiza en dos etapas principales: preprocesado y identificación. En la etapa de preprocesado, las secuencias de nucleótidos se alinean contra un genoma de referencia y los datos se recalibran para corregir errores y mejorar la precisión.

En la etapa de identificación, se utilizan diversas herramientas de análisis, cada una con su propio método de funcionamiento. Algunas de las más comunes son DeepVariant, que utiliza redes neuronales convolucionales para clasificar e identificar variantes; FreeBayes, que emplea el teorema de Bayes para calcular la probabilidad de variantes en un lugar genómico; Strelka2, que modela errores de secuenciación y utiliza un modelo de mezcla para identificar variantes; y HaplotypeCaller, que genera haplotipos y utiliza un PairHMM para identificar variantes.

4 DISEÑO Y DESARROLLO DEL PIPELINE

En esta sección, se detallará el pipeline para la identificación de variantes genómicas. Además, se indicarán las fuentes de datos utilizadas, las discrepancias entre ellas y otras herramientas empleadas. Para la ejecución del pipeline fue necesario un servidor de alto rendimiento. Por lo tanto, también se describirán las características del hardware y software de dicho servidor.

4.1 PIPELINE PARA LA IDENTIFICACIÓN DE VARIANTES GENÓMICAS

El pipeline diseñado se centra en identificar variantes genómicas en células germinales en los exomas, utilizando datos de entrada en formato FASTQ. El proceso incluye varios pasos: comienza con la evaluación de la calidad, seguida de la alineación y la identificación de variantes. Finalmente, se realiza un filtrado por regiones, con la opción de una recalibración después de la alineación en ciertos casos.

Para este Trabajo Final de Máster, los genetistas con los que se ha colaborado recomendaban utilizar NF-Sarek o GATK4. Se decidió usar NF-Sarek porque integra técnicas de GATK4, lo que lo convierte en una herramienta muy completa. A diferencia de GATK4, no solo cuenta con HaplotypeCaller como técnica de identificación de variantes, sino que cuenta con otras técnicas interesantes a evaluar como por ejemplo DeepVariant o Strelka. Cabe destacar que NF-Sarek también tiene algunas limitaciones, como la imposibilidad del uso de genomas codificados como grafos usados en DRAGEN. Se desarrollará más sobre esto último en las conclusiones.

En las siguientes secciones, se proporcionará una descripción detallada de NF-Sarek, las fuentes de datos utilizadas y sus discrepancias. Además, se resumirán los recursos del pipeline, incluyendo sus componentes y herramientas. Finalmente, se explicará cómo se ha realizado la ejecución del pipeline.

4.1.1 NF-Sarek

El pipeline NF-Sarek, esta implementado en Nextflow. Este último simplifica la creación de pipelines computacionales complejos y distribuidos. Nextflow permite combinar múltiples tareas, reutilizar scripts y herramientas existentes sin necesidad de aprender un nuevo lenguaje o API. Además, admite diferentes tipos de contenedores, facilitando la creación de pipelines autocontenidos, la gestión de versiones y la rápida reproducción de configuraciones. También proporciona una capa de abstracción entre la lógica del pipeline y la capa de ejecución, permitiendo su ejecución en múltiples plataformas sin modificaciones. La paralelización la define implícitamente mediante las declaraciones de entrada y salida de los procesos. Por último, todos los resultados intermedios producidos durante la ejecución del pipeline se rastrean automáticamente (Di Tommaso et al., 2017).

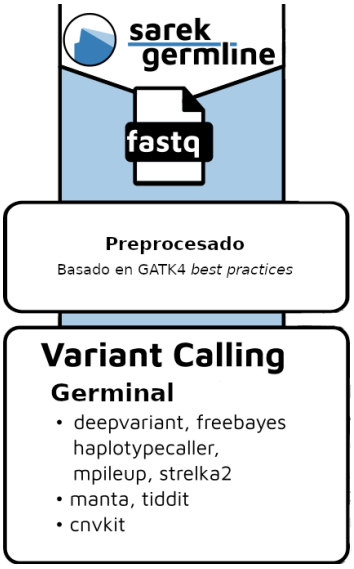

Figura 19 Esquema modificado de las opciones de NF-Sarek/Germline (Garcia et al. 2020).

NF-Sarek del que se puede ver un esquema en Figura 19 es un flujo de trabajo diseñado para identificar variantes en datos de secuenciación de genoma completo o dirigido y puede trabajar con cualquier especie que tenga un genoma de referencia (Garcia et al., 2020).

Para usar NF-Sarek, se debe preparar un archivo CSV con las rutas de los datos de entrada. En la Tabla 6 cada fila representa un par de archivos FAST.

patient	sample	fastq_1	fastq_2
PATIENT1	SAMPLE_PAIRED_END1	p1_R1.fastq.gz	p1_R2.fastq.gz
PATIENT2	SAMPLE_PAIRED_END2	p2_R1.fastq.gz	p2_R2.fastq.gz
PATIENT3	SAMPLE_PAIRED_END3	p3_R1.fastq.gz	p3_R2.fastq.gz

Tabla 6 Ejemplo del CSV con el formato indicado para el correcto funcionamiento de NF-Sarek.

De NF-Sarek se han seleccionado algunas herramientas para evaluarlas y así determinar cuál es la configuración que más se aproxima al gold standard.

Se decidió descartar Sentieon debido a la necesidad de una licencia, y no se utilizó Manta ni Tiddit porque el enfoque del pipeline se centra exclusivamente en los Polimorfismos de Nucleótido Simple (SNPs) y las inserciones y deleciones (Indels), no en las Variantes Estructurales (SVs), que son el foco principal de Manta y Tiddit. También se descartó el uso de mpileup, ya que, aunque puede ser útil en ciertos contextos, no es un llamador de variantes en sí mismo.

Por lo tanto, se decidió utilizar únicamente DeepVariant, HaplotypeCaller, Strelka2 y Freebayes, y como alineadores se decidió usar BWA y DRAGMAP para todos los pacientes.

Es importante destacar que la documentación oficial recomienda desactivar las herramientas de recalibración al usar DRAGMAP. Esta recomendación se basa en que la metodología de DRAGMAP, puede entrar en conflicto con dichas herramientas.

4.1.2 Fuentes de datos usadas

En el contexto del *variant calling* en WES, se deben seleccionar varios archivos esenciales estos son los siguientes:

- **dbSNP**: Se emplea en la recalibración y en HaplotypeCaller.
- **Genoma de referencia**: Se utiliza para alinear las secuenciaciones de NGS.
- **Archivo BED**: Se emplea para delimitar las regiones de los exomas.

Para el fichero dbSNP, se ha utilizado el archivo del proyecto 1000 Genomes concretamente la versión v3. Este archivo es una recopilación de los SNP más frecuentes o de relevancia clínica. En cuanto al genoma de referencia, se ha escogido la versión GRCh38.p14 de Ensemble, que tiene dos variantes: 'dna toplevel rm' (que enmascara repeticiones) y 'dna toplevel' (sin enmascaramiento). Para delimitar las regiones de los exomas, se empleará el archivo Illumina Exome TargetedRegions v1.2.hg38. En este archivo se pueden encontrar 45 millones de bases, una cantidad muy reducida si se compara con los aproximadamente 3.000 millones de bases que componen el genoma completo.

Los FASTQ que contienen las secuencias de nucleótidos que se usarán para validar el pipeline fueron obtenidos mediante el hardware de Illumina y proporcionados por los profesionales de Chile. La Tabla 7 muestra el tamaño según la dirección de secuenciación y dos identificadores diferentes, lo que facilitará la representación de los datos en gráficos y tablas durante la validación.

Identificador	Tamaño R1 (KB)	Tamaño R2 (KB)	Id Auxiliar
F495P1	3.173,409	3.168.208	1
F510P1	390.686	388.543	2
F520P1	416.303	418.626	3
F523P1	3.153.631	3.148.200	4
F532P1	4.200.753	4.195.725	5
F538P1	3.052.733	3.040.634	6
F567P1	3.680.027	3.659.876	7
F584P1	359.697	361.026	8
F586P1	387.811	386.155	9
F589P1	2.779.541	2.745.069	10

Tabla 7 Pacientes, tamaño en KB de los fastq y el id auxiliar.

Además, para las comparaciones de los resultados y validación se usarán unos archivos VCF generados mediante el pipeline de DRAGEN; estos también fueron facilitados por los profesionales de Chile. Las características de esto se pueden ver en la Tabla 8.

Identificador	N.º de variantes	Tamaño
F495	38688	12.054 KB
F510	38322	10.632 KB
F520	-	-
F523	38954	11.933 KB
F538	38714	12.127 KB
F567	38120	12.091 KB
F584	35348	9.898 KB
F586	31901	10.390 KB
F589	32939	11.853 KB

Tabla 8 N.º de variantes y tamaño de los VCF obtenidos con DRAGEN por paciente.

Además, los archivos, tanto el BED de Illumina como los VCFs de DRAGEN, tuvieron que ser modificados eliminando el prefijo 'chr' de la columna del nombre del cromosoma para facilitar su procesamiento. Cabe destacar que el paciente con el id auxiliar número 3 fue excluido por no contar con el VCF obtenido con DRAGEN.

4.1.3 Discrepancias entre diferentes fuentes de datos

En la selección del genoma de referencia y el dbSNP el motivo de usar los datos que nos proporciona Ensembl es porque después de un breve análisis, se pudo verificar que la notación de los cromosomas era sustancialmente diferente entre NCBI, Ensembl y el BED de Illumina. Un ejemplo de esto se observa en la Tabla 9.

Cromosoma	GenBank	RefSeq	Ensembl	Illumina Bed	DRAGEN .VCF
1	CM000663.2	NC_000001.11	1	chr1	chr1
2	CM000664.2	NC_000002.12	2	chr2	chr2

Tabla 9 Diferencias en los nombres de los cromosomas en diferentes fuentes.

Por lo tanto, se optó por usar Ensembl como fuente de datos por ser la que más facilitaba la estandarización de estas notaciones.

Además, es importante mencionar que las versiones *toplevel* del genoma de referencia tanto de NCBI como de Ensemble deberían ser idénticas en términos de contenido. Sin embargo, en lo que respecta a las versiones enmascaradas, no se ha hallado documentación que confirme su equivalencia.

4.1.4 Resumen de recursos

Por concluir con todo lo anterior en la Figura 20, se presentan las técnicas y fuentes de datos que se utilizarán en nuestro pipeline inicial.

En este pipeline, se toma cada uno de los pares y se alinean con el genoma enmascarado y sin enmascarar, utilizando BWA y DRAGMAP. Para este último, se omite la recalibración. Luego, con los resultados de cada uno de estos alineadores y genomas de referencia, se genera un VCF utilizando diferentes técnicas: FreeBayes, HaplotypeCaller, DeepVariant y Strelka. Posteriormente, todos los resultados se filtrarán utilizando el BED de Illumina con la herramienta BCFtools. Los resultados obtenidos se discutirán en el apartado de validación, donde se concluirá con la mejor alternativa entre estas técnicas y conjuntos de datos para acercarnos al gold standard.

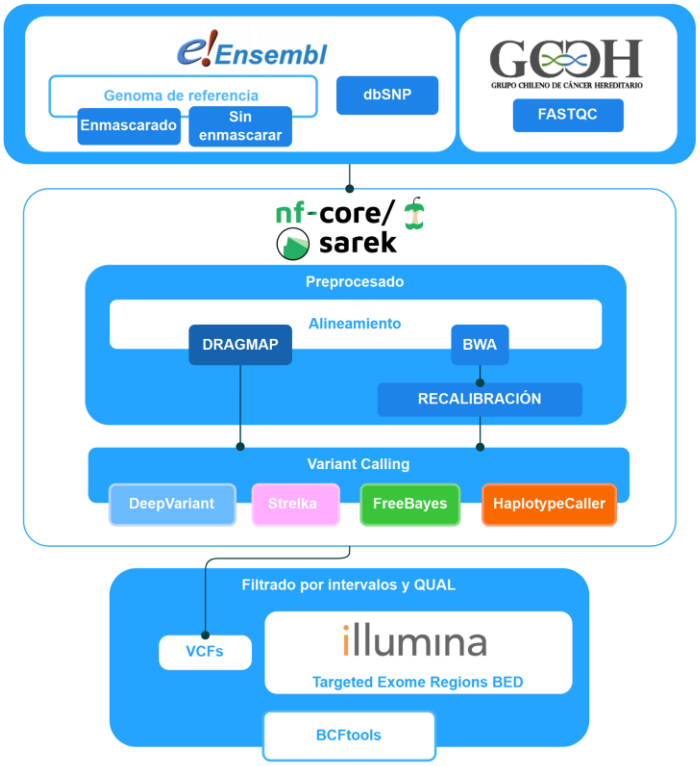

Figura 20 Selección de herramientas y fuentes de datos para la validación.

4.2 SERVIDOR

Como se ha especificado anteriormente, para ejecutar estos pipelines son necesarios grandes recursos de computación. Estos recursos son poco habituales o inexistentes en sistemas convencionales de uso doméstico. Por lo tanto, se requiere una infraestructura de computación de alto rendimiento (High-Performance Computing, HPC). Un HPC aprovecha varios grupos de ordenadores trabajando de manera conjunta para resolver problemas complejos que requieren de computación masiva. Estos sistemas son muy usados en áreas enfocadas a avances médicos y de la ciencia de materiales.

Frecuentemente en estos sistemas se aprovecha la paralelización, es decir, dividir un problema en partes más pequeñas que se pueden resolver simultáneamente, utilizando múltiples procesadores o núcleos.

4.2.1 Hardware del HPC

La plataforma donde se ha realizado esta ejecución cuenta con un nodo de acceso que actúa como el portal de entrada al HPC, donde se organizan y preparan los trabajos computacionales. Una vez listos, estos trabajos se distribuyen para su ejecución en un conjunto de seis ordenadores.

Estos últimos sirven como nodos de cómputo dedicados y, por lo general, no se acceden directamente por los usuarios. Cada uno de estos nodos de cómputo está equipado con dos procesadores AMD

EPYC 7453, que cuentan con 28 núcleos cada uno, una memoria de 512 GB de RAM, y 8 GPUs NVIDIA A40 con 48 GB de memoria cada una.

En este Trabajo Final de Máster, se ha utilizado una CPU con un uso completo de todos los núcleos y sin limitaciones en la memoria RAM.

4.2.2 Software del HPC

Dentro de un HPC, el software no es como el de un ordenador convencional. Mientras que los ordenadores personales están diseñados para tareas cotidianas y aplicaciones generales, el software de un HPC está optimizado para manejar cálculos complejos y grandes volúmenes de datos. En esta sección se expondrá el software del HPC que se ha utilizado para la realización del pipeline.

4.2.2.1 Sistema operativo

Los sistemas operativos en la mayoría de los HPC suelen ser variantes de Linux, un sistema operativo tipo Unix bajo la licencia GNU GPL. Linux es especialmente popular en bioinformática dado que las herramientas en su mayoría se encuentran en estos sistemas operativos.

La distribución de Linux empleada en este Trabajo Final de Máster es Ubuntu la cual es una de las distribuciones de Linux más populares y se orienta al usuario promedio con un fuerte enfoque en la facilidad de uso.

4.2.2.2 SLURM

Para asignar las ejecuciones a los nodos, se utilizó SLURM (Yoo et al., 2003), un sistema de gestión de colas que asigna recursos de manera eficiente en un HPC. SLURM puede asignar nodos de cómputo a usuarios y administrar colas de trabajos pendientes.

SLURM ofrece una gran cantidad de opciones. Aunque no se profundizará en todas ellas debido a su extensión, se mencionarán algunas que han sido esenciales para la ejecución del pipeline, estas opciones son SBATCH, SQUEUE y SCANCEL.

El comando SBATCH se utiliza para enviar un script a SLURM. Al script no necesariamente se le otorgan recursos de inmediato, es decir, puede permanecer en la cola de trabajos pendientes durante algún tiempo antes de que los recursos requeridos estén disponibles.

Después se puede encontrar el comando SQUEUE, el cual se utiliza para ver la información de los trabajos. Por defecto, SQUEUE imprimirá el ID del trabajo, la partición, el nombre de usuario, el estado del trabajo, el número de nodos y el nombre de los nodos para todos los trabajos en cola o en ejecución dentro de SLURM. Por último, el comando SCANCEL, este se utiliza para cancelar trabajos.

Además, si se quiere que el trabajo puesto en cola tenga ciertos criterios se puede crear un archivo SH donde se especifique exactamente la cantidad de recursos necesarios para la tarea.

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=28
singularity run --fakeroot pipeline.sif
```

Ejemplo 11 Archivo SH donde se especifican los recursos para SLURM.

En el Ejemplo 11 se puede ver como primero, se especifica el lenguaje que interpretará el texto. Luego, se define el número de nodos. A continuación, se indica el número de tareas a realizar y, después, se menciona la cantidad de CPUs necesarias. Por último, se muestra el código para ejecutar un contenedor de Singularity, una herramienta que se explicará a continuación.

4.2.2.3 Singularity

Singularity, una plataforma de contenedores que permite la creación y ejecución de contenedores que empaquetan componentes de software de manera portátil y reproducible, es utilizada por el HPC como medida de seguridad (Kurtzer et al., 2017). La ventaja es que el contenedor es un solo archivo, lo que significa que no es necesario instalar todo el software necesario en varios sistdemas operativos.

Los contenedores se pueden crear a partir de archivos llamados *recipes*. Estos son archivos de texto con un esquema específico que contienen instrucciones para construir un contenedor personalizado, incluyendo detalles sobre el sistema operativo base, el software a instalar, las variables de entorno, los archivos a agregar desde el sistema host y los metadatos del contenedor. Un esquema simplificado de estos *recipes* sería el que se puede ver en el Ejemplo 12.

```
Bootstrap: docker # Aquí se especifica de donde se obtiene la imagen del OS
From: ubuntu      # Aquí se especifica el OS y la versión


%setup
      # Aquí iría toda la instalación de NF-sarek y demás recursos
      necesarios
%runscript
      # Aquí vendría la ejecución en sí, estos scripts se especifican mas
```

Ejemplo 12 Recipe para generar el contenedor en Singularity.

Al igual que en el caso de SLURM, existen numerosas opciones disponibles. No se profundizará en todas ellas debido a su gran cantidad, pero se mencionarán algunas que han sido esenciales para la ejecución. Estas son RUN, BUILD y FAKEROOT.

El comando BUILD se utiliza para construir un contenedor Singularity. Puede tomar como entrada un contenedor existente, un directorio, o un *recipe*.

También, se puede encontrar el comando RUN. Este ejecuta un contenedor Singularity y ejecuta el script de ejecución si está definido para el contenedor.

Por último, una característica importante de Singularity es FAKEROOT, que permite a un usuario no privilegiado ejecutar contenedores con privilegios de *root* dentro del contenedor, sin tener privilegios de *root* en el *host* fuera del contenedor. Esta característica es útil para realizar tareas dentro del contenedor sin comprometer la seguridad del sistema anfitrión.

4.2.3 Ejecución del pipeline

Una vez se han explicado los elementos del pipeline y los recursos de *software* como de *hardware* necesarios, se profundizará en el proceso de ejecución. Primero en un sistema externo al servidor principal, se instala el pipeline NF-Sarek dentro de un contenedor de singularity, creando un entorno aislado y controlado para su ejecución. Luego, tanto los datos como el contenedor se suben al HPC. Dentro del contenedor, se encuentra el script de ejecución con los comandos necesarios para ejecutar el pipeline. Estos son los que se han usado en este Trabajo Final de Máster:

```
nextflow run nf-core/sarek -r 3.4.2 -profile singularity --input samplesheet.csv --outdir salida --tools strelka,deepvariant,haplotypecaller,freebayes --fasta Genoma/Homo_sapiens.GRCh38.dna.toplevel.fa --nucleotides_per_second 400000 --fasta_fai Genoma/Homo_sapiens.GRCh38.dna.toplevel.fa.fai --igenomes_ignore true --dbsnp Genoma/1000GENOMES-phase_3_.vcf.gz --aligner dragmap --skip_tools baserecalibrator
```

Ejemplo 13 Comando de NF-Sarek para DRAGMAP sin enmascarar.

```
nextflow run nf-core/sarek -r 3.4.2 -profile singularity --input samplesheet.csv --outdir salida --tools strelka,deepvariant,haplotypecaller,freebayes --fasta Genoma/Homo_sapiens.GRCh38.dna.toplevel.fa --nucleotides_per_second 400000 --fasta_fai Genoma/Homo_sapiens.GRCh38.dna.toplevel.fa.fai --igenomes_ignore true --dbsnp Genoma/1000GENOMES-phase_3_.vcf.gz
```

Ejemplo 14 Comando de NF-Sarek para BWA sin enmascarar.

```
nextflow run nf-core/sarek -r 3.4.2 -profile singularity --input samplesheet.csv --outdir salida --tools strelka,deepvariant,haplotypecaller,freebayes --fasta Genoma/Homo_sapiens.GRCh38.dna_rm.toplevel.fa --nucleotides_per_second 400000 --fasta_fai Genoma/Homo_sapiens.GRCh38.dna_rm.toplevel.fa.fai --igenomes_ignore true --dbsnp Genoma/1000GENOMES-phase_3_.vcf.gz
```

Ejemplo 15 Comando de NF-Sarek para BWA enmascarado.

```
nextflow run nf-core/sarek -r 3.4.2 -profile singularity --input samplesheet.csv --outdir salida --tools strelka,deepvariant,haplotypecaller,freebayes --fasta Genoma/Homo_sapiens.GRCh38.dna_rm.toplevel.fa --nucleotides_per_second 400000 --fasta_fai Genoma/Homo_sapiens.GRCh38.dna_rm.toplevel.fa.fai --igenomes_ignore true --dbsnp Genoma/1000GENOMES-phase_3_.vcf.gz --aligner dragmap --skip_tools baserecalibrator
```

Ejemplo 16 Comando de NF-Sarek para DRAGMAP enmascarado.

Estos comandos generan múltiples directorios de salida. Su estructura se puede apreciar en la Figura 21. Dentro de estos directorios, se encuentran los resultados del pipeline.

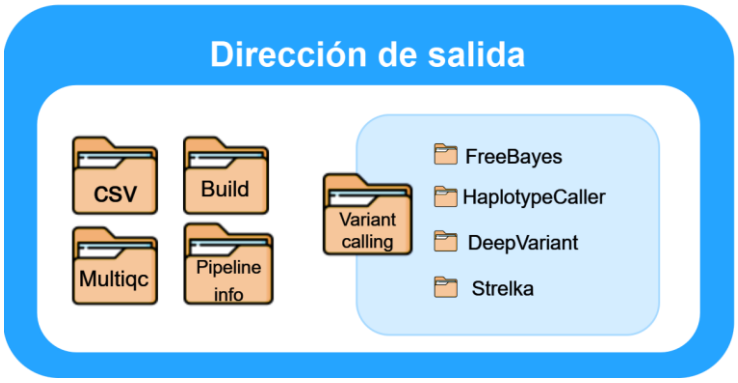

Figura 21 Esquema de carpetas del pipeline.

Una vez obtenidos los archivos VCF a través del pipeline, el siguiente paso es filtrarlos utilizando el archivo BED proporcionado por Illumina. También se filtran los indels con QUAL mayor que 0 y los SNPs con QUAL mayor o igual a 3, siguiendo el mismo criterio observado en los VCF de DRAGEN, que también fueron filtrados con el archivo BED mencionado. Para la validación, se obtendrán las intersecciones entre los VCF generados por nuestro pipeline y los de DRAGEN.

Para esto, se utilizará un script personalizado en Bash que emplea herramientas de BCFtools, una suite de utilidades para trabajar con archivos VCF y BED.

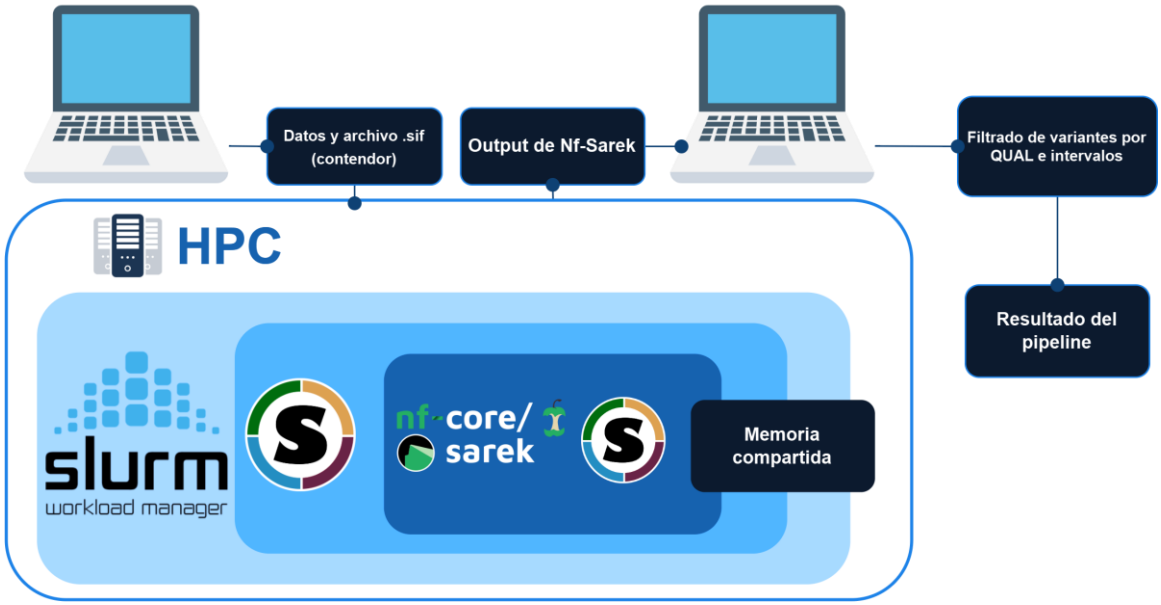

Figura 22 Esquema de la ejecución del pipeline.

Una de las fortalezas de nuestro trabajo es la versatilidad del esquema que se ha desarrollado. Aunque está diseñado específicamente para una infraestructura específica, con sus propias características y limitaciones, el enfoque subyacente es adaptable y sirve como un punto de partida para otras implementaciones que puedan ser ejecutadas en otros entornos de computación de alto rendimiento. Cabe destacar la estructura modular y bien documentada del esquema, permitiendo realizar personalizaciones de una manera sencilla y eficiente.

4.3 RESPUESTAS A LAS PREGUNTAS DE INVESTIGACIÓN DEL SEGUNDO SUBOBJETIVO

Ahora a modo de resumen de la sección donde se trata el segundo subobjetivo se responderán a las preguntas de investigación asociadas a este.

PI4: ¿Cómo obtener un pipeline para identificar variantes genómicas?

Para identificar variantes genómicas, primero se seleccionan herramientas de alineación e identificación. Además, es necesario escoger una fuente de datos para el genoma de referencia y el dbSNP. En la sección 4, tras explicar las herramientas, se procederá a utilizarlas y detallar su ejecución para una posterior evaluación en la sección 5 y, a partir de esta validación, realizar una selección.

Una herramienta que integra un gran número de estas es NF-Sarek, debido a su capacidad para combinar múltiples herramientas de análisis genómico de alta precisión, incluyendo técnicas de GATK4, con alineadores BWA y DRAGMAP. Con los archivos BAM generados, se obtienen los archivos VCF utilizando DeepVariant, Strelka, HaplotypeCaller y FreeBayes.

En general, BWA es considerado el estado del arte en cuanto a alineadores, aunque los autores de DRAGMAP afirman que esta herramienta es superior y está destinada a reemplazarlo. En cuanto a las técnicas de identificación de variantes, DeepVariant puede considerarse un pseudo estado del arte, dado su rendimiento notable en el desafío PrecisionFDA V2. Sin embargo, en WES, DRAGEN es superior.

Respecto al uso de otras herramientas, aunque no sean las mejores en términos de fiabilidad de resultados, son significativamente más rápidas, según la documentación. Por esta razón, se eligieron para evaluar si la posible pérdida de precisión podría compensarse con los beneficios de menores tiempos de procesamiento y recursos utilizados.

PI5: ¿Qué hardware es necesario para implementar este pipeline?

Se requirió una infraestructura de alto rendimiento (HPC) para implementar el pipeline. Esta infraestructura, descrita en la Sección 4.2, contaba con un mínimo de 28 núcleos de CPU y 60 GB de RAM, siendo estas configuraciones habituales en el campo de la identificación de variantes.

5 VALIDACIÓN DEL PIPELINE

Una vez explicado el *pipeline* y su ejecución, se van a validar las herramientas de este, lo cual es nuestro subobjetivo 3. Para la validación en las siguientes secciones, se realizará un análisis de la calidad de las secuenciaciones, se comparan los resultados con los VCFs generados con DRAGEN con los del pipeline desarrollado y, además, se estimarán los tiempos de ejecución por core de CPU y los gastos de memoria RAM.

5.1 ANÁLISIS DEL NÚMERO DE LECTURAS Y COBERTURA

Para evaluar la calidad de las secuenciaciones, se necesita conocer el número de lecturas, así como el número de duplicación de estas y cuáles no se han podido mapear incluyendo además la información acerca de su cobertura. Estas características son importantes en la etapa de *variant calling*, ya que, como se verá más adelante, un número reducido de estas puede afectar negativamente a los resultados.

En el contexto de NGS, el número de lecturas se refiere a las secuencias de nucleótidos obtenidas durante el proceso de secuenciación. Por otro lado, la cobertura se refiere al número de veces que una base específica del genoma es leída durante el proceso de secuenciación. Una mayor cobertura implica una mayor fiabilidad en la identificación de variantes genómicas, ya que cada base del ADN es secuenciada múltiples veces, lo que reduce la posibilidad de errores.

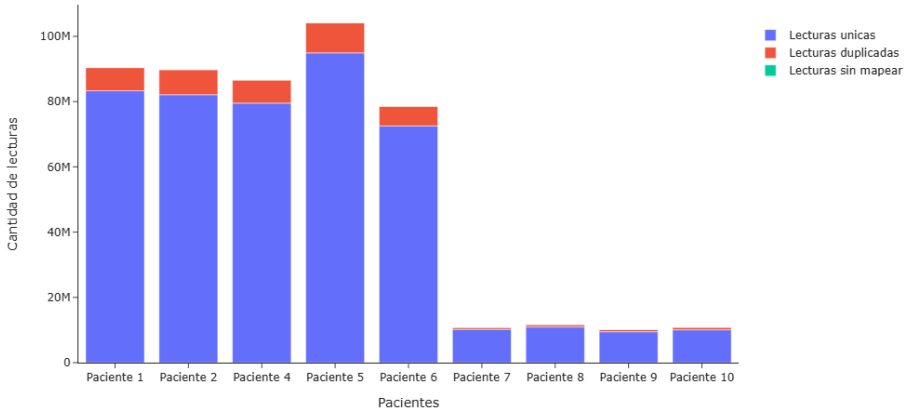

Figura 23. Gráfico de barras apiladas indicando el número de lecturas únicas, duplicadas y sin mapear de cada paciente tras el alineador BWA usando el genoma de referencia sin enmascarar.

Como se puede ver en la Figura 23 con el genoma sin enmascarar BWA logra mapear casi todas las lecturas de los FASTQ. Además, también se aprecia que los pacientes 7, 8, 9 y 10 tienen un número muy reducido de lecturas.

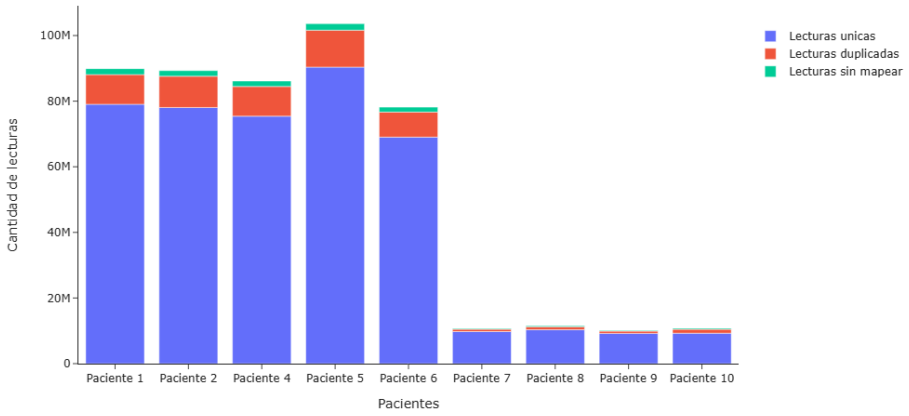

Figura 24 Gráfico de barras apiladas indicando el número de lecturas únicas, duplicadas y sin mapear por paciente tras el alineador BWA usando el genoma de referencia enmascarado.

En Figura 24 se puede observar que con el uso de un genoma enmascarado BWA no logra mapear algunas de las lecturas.

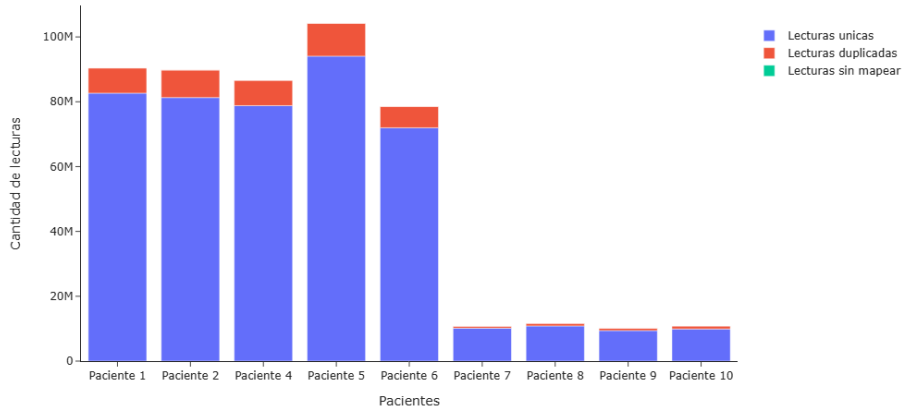

Figura 25 Gráfico de barras apiladas indicando el número de lecturas únicas, duplicadas y sin mapear por paciente tras el alineador DRAGMAP usando el genoma de referencia sin enmascarar.

En la Figura 25 se puede apreciar cómo DRAGMAP tiene un comportamiento muy similar a BWA con el genoma sin enmascarar.

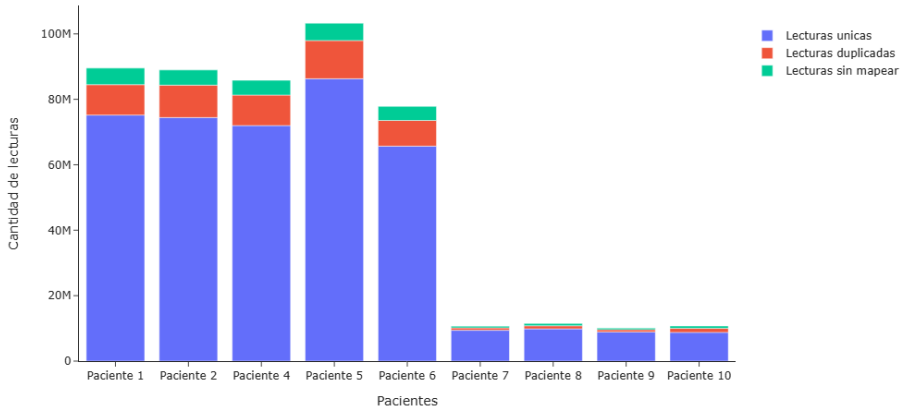

Figura 26 Gráfico de barras apiladas indicando el número de lecturas únicas, duplicadas y sin mapear por paciente tras el alineador DRAGMAP usando el genoma de referencia enmascarado.

En la Figura 26 se puede observar que con el genoma enmascarado DRAGMAP deja muchas más lecturas sin mapear con respecto a BWA. Para ejemplificar todavía más los resultados, se van a ver en una tabla los valores de un paciente que tiene una cobertura de 80x para los exomas.

Descripción	N.º de lecturas	Tipo de Lectura
DRAGMAP sin enmascarar	82664806	Lecturas únicas
	7774540	Lecturas duplicadas
	4369.0	Lecturas sin mapear
BWA sin enmascarar	83312674	Lecturas únicas
	71114378	Lecturas duplicadas
	11108.0	Lecturas sin mapear
DRAGMAP enmascarado	75195924	Lecturas únicas
	9287180	Lecturas duplicadas
	5059911	Lecturas sin mapear
BWA enmascarado	79047298	Lecturas únicas
	9051982	Lecturas duplicadas
	1804974	Lecturas sin mapear

Tabla 10 de lecturas únicas, duplicadas y sin mapear por paciente tras el alineador BWA y DRAGMAP usando el genoma de referencia enmascarado y sin enmascarar para un solo paciente.

En la Tabla 10 se aprecia cómo DRAGMAP logra alinear más lecturas que BWA si el genoma está sin enmascarar, pero en caso contrario BWA logra alinear más lecturas.

Es importante mencionar que, aunque se logren alinear más lecturas, esto no garantiza una alineación correcta. Sin embargo, es una forma de realizar un análisis de calidad basado en la alineación. Para evaluar su rendimiento, sería necesario llevar a cabo simulaciones. Aun así, como se ha señalado en la sección de investigación del problema, según los autores, el rendimiento de DRAGMAP supera al de BWA.

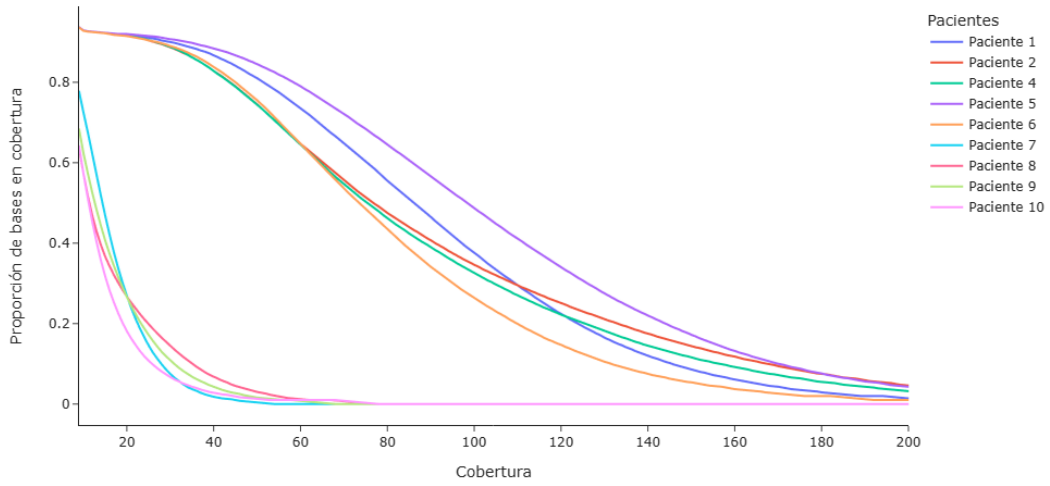

Figura 27 Distribución acumulada de la cobertura en los exomas dividido por paciente.

En la Figura 27 se puede ver la cobertura de los exomas. Aproximadamente el 50 % de las zonas de interés para los pacientes 1, 2, 4, 5 y 6 tienen aproximadamente más de 80 lecturas que las respaldan. La cobertura recomendada en la literatura para WES suele estar entre 80x y 120x.

Luego se puede encontrar al grupo de los pacientes 7 ,8 ,9 y 10 que tienen una cobertura que no sería apta para el análisis secundario en WES. Esta diferencia de cobertura entre unos y otros se debe en esencia al tiempo que la muestra biológica pasa dentro del secuenciador que se traduce en la cantidad de ciclos de lectura que se realizan. Cabe destacar que estas coberturas eran muy similares para cada alineador y genoma de referencia.

5.2 ANÁLISIS DEL DESEMPEÑO

En esta sección se evaluarán los falsos positivos, falsos negativos y verdaderos positivos en comparación con nuestro *gold standard,* tanto para todas las variantes como para indels y SNPs. Se utilizarán diagramas de caja para representar estos valores, dividiendo según el genoma alineador y la técnica de llamada de variantes. Además, se empleará el F1-score, una métrica muy común en estos casos, que nos permite resumir todos los valores anteriormente nombrados. Además de estos F1 scores se obtendrán la media, mediana y la desviación típica, para diferentes agrupaciones que se especificarán más adelante.

El F1-score se calcula con la siguiente fórmula:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

Ecuación 13 Fórmula del F1 score.

Siendo la Precisión:

$$Precision = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ positivos}$$

Ecuación 14 Fórmula de la Precision.

Y siendo el Recall:

$$Recall = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ negativos}$$

Ecuación 15 Fórmula del Recall.

5.2.1 Análisis de los verdaderos positivos

En esta sección se presentarán gráficos de diagramas de cajas que mostrarán la cantidad de verdaderos positivos divididos por técnica de alineamiento y tipo de genoma de referencia. Se realizará para todas las variantes y, posteriormente, de manera separada para SNPs e indels.

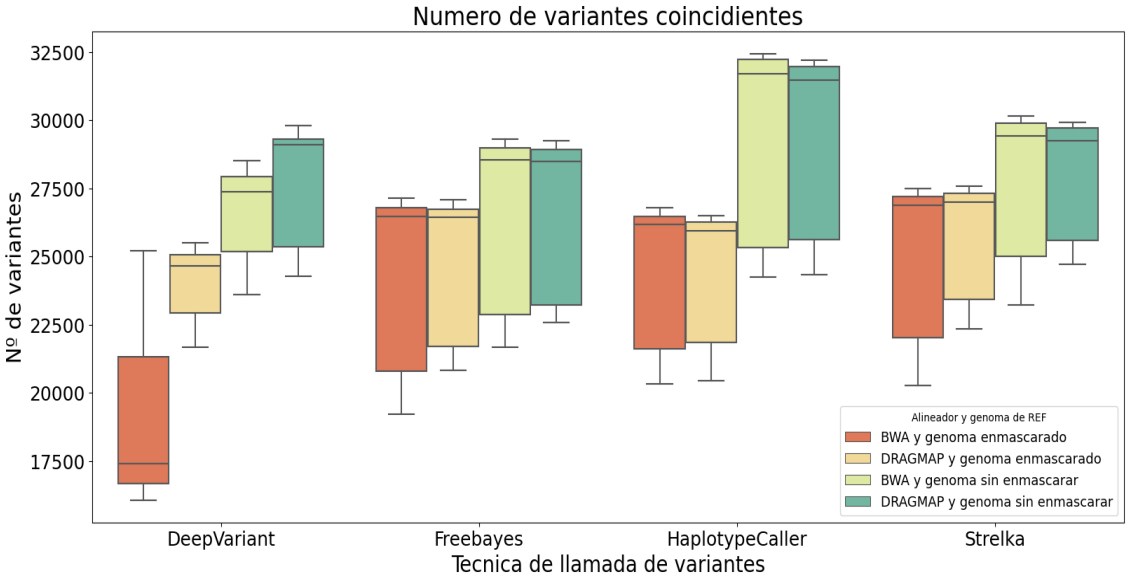

Figura 28 Diagramas de caja del nº de verdaderos positivos por técnica, genoma de referencia y alineador.

En la Figura 28, se observa que para maximizar el número de verdaderos positivos, es preferible optar por el uso de HaplotypeCaller. También se puede ver que, en todos los casos, el empleo de un genoma enmascarado conduce a resultados inferiores.

Además, es apreciable que DeepVariant, muestra una mayor susceptibilidad tanto al enmascaramiento del genoma como al tipo de alineador utilizado si se compara con el resto de las herramientas. Tanto Freebayes, como DeepVariant y Strelka tienen, un rendimiento similar.

A continuación, se va a realizar el mismo análisis, pero en este caso diferenciando entre SNPs e indels, para obtener una visión más detallada y específica del impacto de estas herramientas en cada tipo de variante.

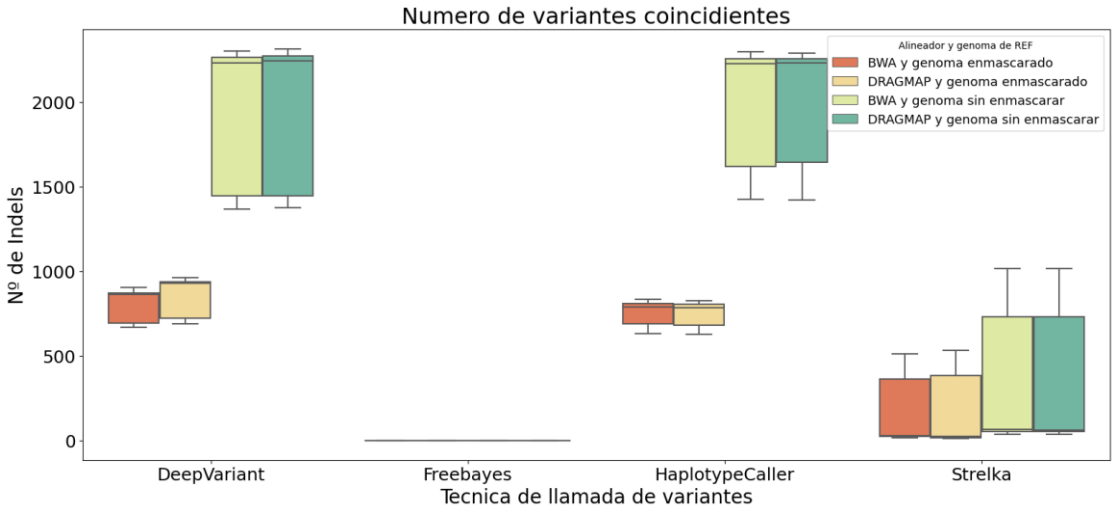

Figura 29 Diagramas de caja del nº verdaderos positivos (solo indels) por técnica, genoma de referencia y alineador.

Para el caso de los indels se puede observar en la Figura 29 el mismo comportamiento en cuanto al enmascaramiento del genoma, es decir, la aplicación de un genoma sin enmascarar es más efectivo. DeepVariant y HaplotypeCaller emergen como las más competentes en esta categoría.

Por otro lado, tanto FreeBayes como Strelka tienen un rendimiento considerablemente bajo en la identificación de indels. FreeBayes no logra identificar ningún indel que esté presente en el *gold standard* mientras que Strelka identifica una cantidad muy reducida en comparación con DeepVariant y HaplotypeCaller.

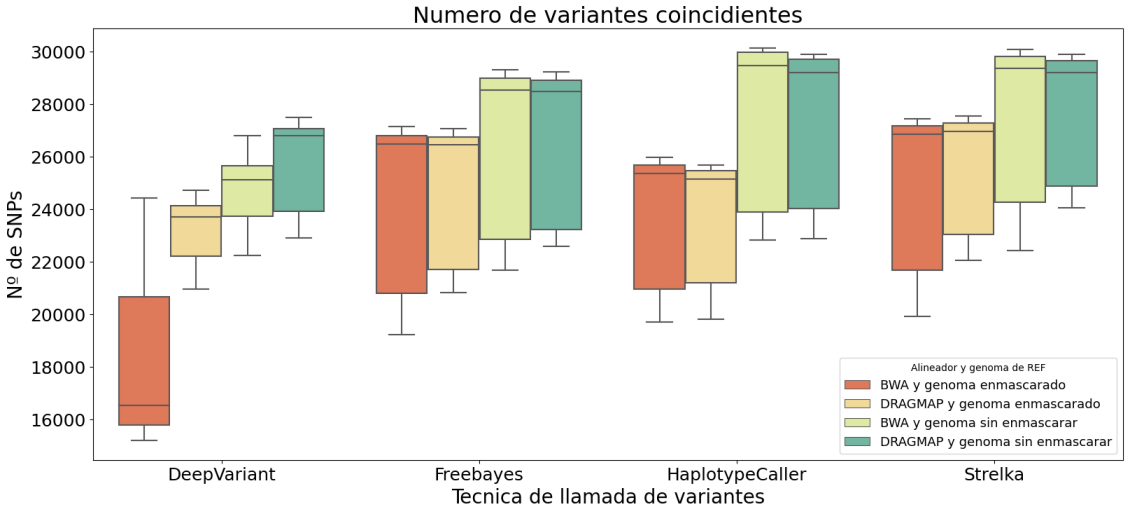

Figura 30 Diagramas de caja del nº de verdaderos positivos (solo SNPs) por técnica, genoma de referencia y alineador.

Dado que el número de SNPs es significativamente mayor que el número de indels, la Figura 30 es prácticamente igual a la, con algunas diferencias. Entre ellas, tanto FreeBayes como Strelka se posicionan mucho mejor si solo se tienen en cuenta los SNPs.

5.2.2 Análisis de los falsos positivos

Al igual que en el caso anterior se presentarán gráficos de diagramas con las mismas divisiones, pero ahora teniendo en cuenta los falsos positivos. Estos son la cantidad de variantes que solo se encuentran en los VCFs generados mediante nuestro pipeline.

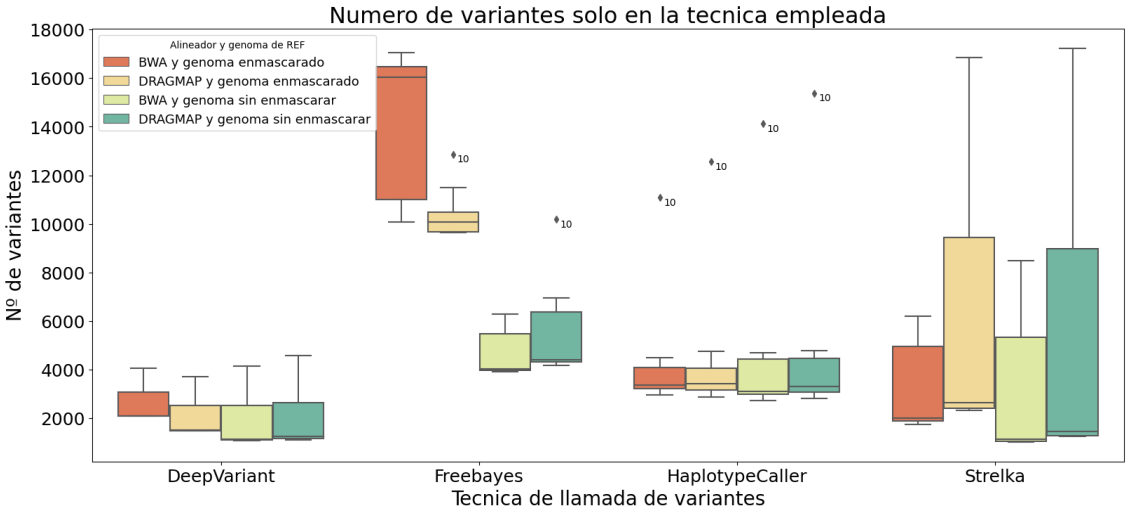

Figura 31 Diagramas de caja del nº de falsos positivos por técnica, genoma de referencia y alineador.

Tras analizar la Figura 31 se puede indicar que Strelka se posiciona como es la técnica de la identificación de variantes con la menor cantidad de falsos positivos, aunque su variabilidad es alta. Si se considera la variabilidad como algo negativo, el ganador en este caso sería DeepVariant.

A pesar de que en la anterior sección DeepVariant no mostraba un rendimiento óptimo, en la Figura 31 se puede apreciar como su desempeño no se ve tan afectado por la cantidad limitada de datos. Esto sugiere que DeepVariant podría ser una herramienta robusta y confiable incluso en condiciones donde la información disponible es escasa.

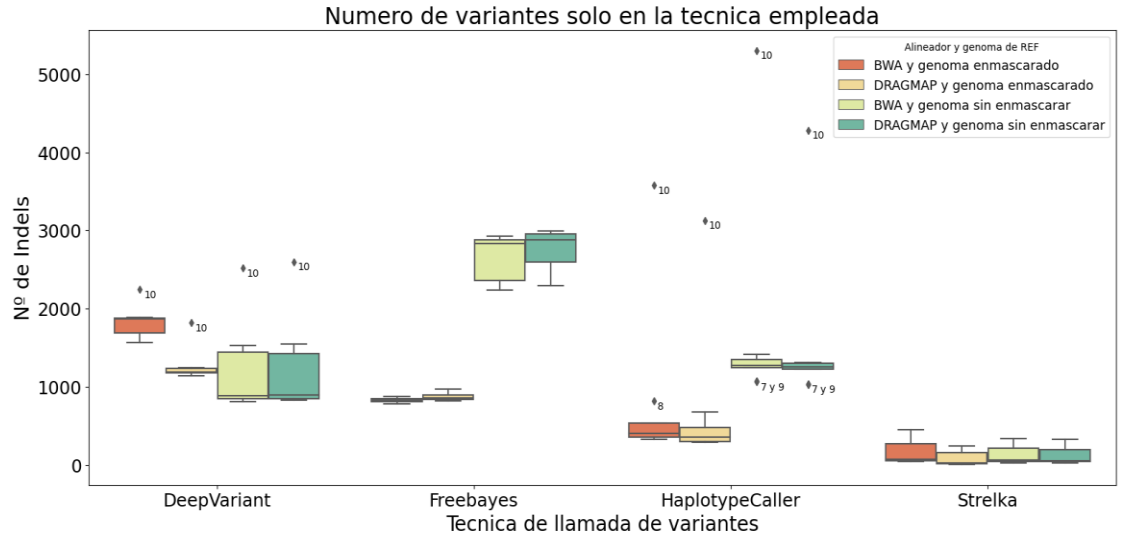

Figura 32 Diagramas de caja del nº de falsos positivos (solo indels) por técnica, genoma de referencia y alineador.

Como se puede observar en la Figura 32, Strelka presenta menos falsos positivos en cuanto a indels. Sin embargo, este bajo número se debe al deficiente desempeño de esta técnica para identificar este tipo de variantes.

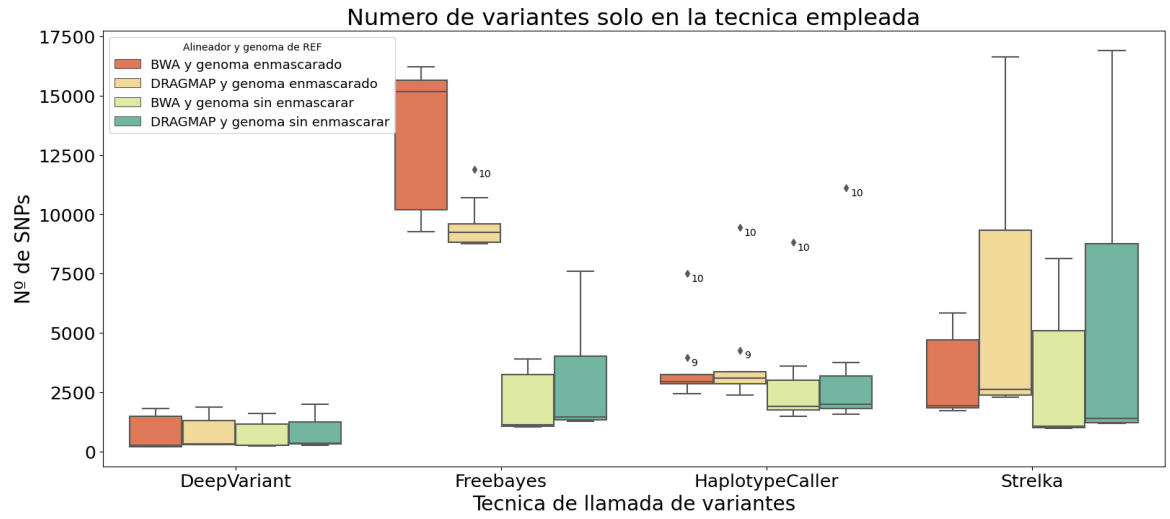

Figura 33 Diagramas de caja del nº de falsos positivos (solo SNPs) por técnica, genoma de referencia y alineador.

Si se ven los falsos positivos de los SNPs en la Figura 33 aquí el ganador indiscutible es DeepVariant para todos los alineadores y referencias con una variabilidad muy reducida.

5.2.3 Análisis de los falsos negativos

Para concluir el análisis inicial se va a realizar lo mismo que en secciones anteriores, pero para los falsos negativos, es decir, las variantes que solo se encuentran en los VCFs generados por DRAGEN.

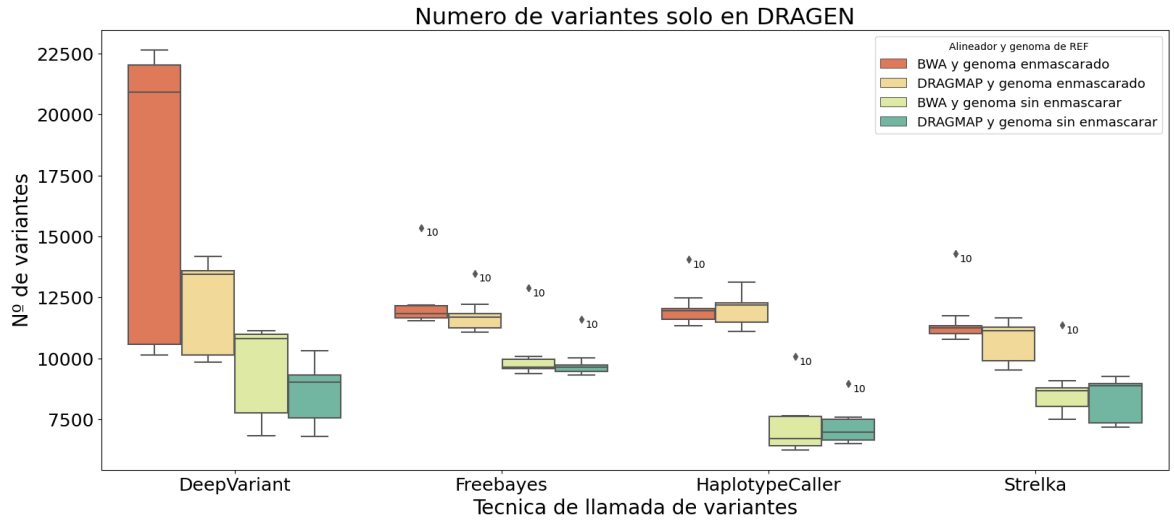

Figura 34 Diagramas de caja del nº de falsos negativos por técnica, genoma de referencia y alineador.

Como se puede observar en la Figura 34, el ganador es HaplotypeCaller, siendo muy reducidos estos falsos negativos. Le siguen Strelka y DeepVariant.

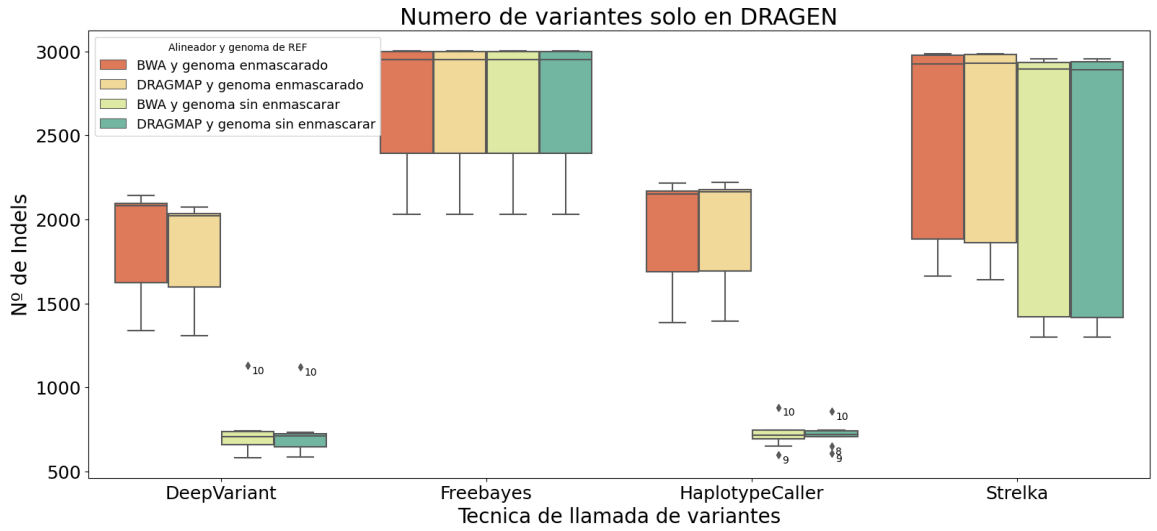

Figura 35 Diagramas de caja del nº de falsos negativos (solo indels) por técnica, genoma de referencia y alineador.

Se puede apreciar en la Figura 35 que tanto DeepVariant como HaplotypeCaller son las mejores alternativas a DRAGEN en lo que respecta a los falsos negativos en indels. Además, se nota un gran aumento en el desempeño de estas dos técnicas al aplicar un genoma de referencia sin enmascarar. Este comportamiento no se observa en Strelka ni en FreeBayes.

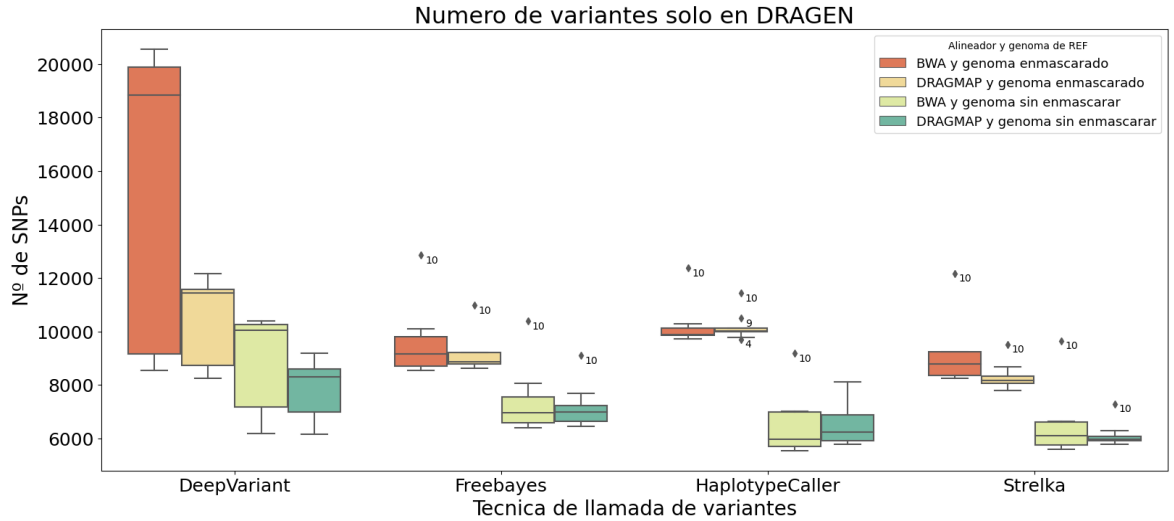

Figura 36 Diagramas de caja del nº de falsos negativos (solo SNPs) por técnica, genoma de referencia y alineador.

Si solo se tienen, en cuenta los SNPs que son falsos negativos en la Figura 36 se puede observar cómo Strelka rivaliza con HaplotypeCaller y Freebayes mejora el rendimiento.

Tras analizar los falsos negativos, falsos positivos y verdaderos positivos, se puede concluir que, en general, los mejores resultados se obtienen con DeepVariant y HaplotypeCaller. Además, es preferible usar un genoma sin enmascarar entre las opciones que se han evaluado y los datos que se han utilizado.

5.2.4 Análisis F1 scores y diagramas de Venn

En esta sección, se utilizará la métrica F1, anteriormente definida, para resumir los gráficos anteriores y evaluar los resultados de las técnicas de manera objetiva. Se hará siguiendo el enfoque del caso anterior, centrándonos en las técnicas, el alineador y el genoma de referencia separando entre todas las variantes, SPNs y indels. Además, se analizarán los resultados a nivel de paciente para comprobar si los valores de cobertura y número de lecturas influyen realmente en los resultados del pipeline.

Técnica de llamada de variantes	Alineador y REF	Media	Mediana	STD
DeepVariant	DRAGMAP y genoma sin enmascarar	0.837943	0.849464	0.029438
HaplotypeCaller	BWA y genoma sin enmascarar	0.831718	0.867623	0.067484
HaplotypeCaller	DRAGMAP y genoma sin enmascarar	0.829627	0.860249	0.062897
DeepVariant	BWA y genoma sin enmascarar	0.820390	0.824561	0.027326
Strelka	BWA y genoma sin enmascarar	0.818339	0.855327	0.057460
Strelka	DRAGMAP y genoma sin enmascarar	0.802171	0.849303	0.069742
Freebayes	BWA y genoma sin enmascarar	0.779705	0.806342	0.042692
Freebayes	DRAGMAP y genoma sin enmascarar	0.773427	0.801335	0.045432
DeepVariant	DRAGMAP y genoma enmascarado	0.768967	0.768586	0.023238
Strelka	BWA y genoma enmascarado	0.767106	0.798855	0.051510
Strelka	DRAGMAP y genoma enmascarado	0.749909	0.793174	0.066639
HaplotypeCaller	BWA y genoma enmascarado	0.746434	0.769696	0.051830
HaplotypeCaller	DRAGMAP y genoma enmascarado	0.745341	0.768103	0.048548
Freebayes	DRAGMAP y genoma enmascarado	0.687289	0.706863	0.034346
DeepVariant	BWA y genoma enmascarado	0.659252	0.601940	0.094824
Freebayes	BWA y genoma enmascarado	0.646571	0.651982	0.021598

Tabla 11 Media mediana y desviación típica de la métrica F1 agrupada por técnica, tipo de genoma de referencia y alineador.

Según la Tabla 11, las mejores opciones para todas las variantes, basadas en los valores de F1 serían HaplotypeCaller BWA con el genoma sin enmascarar y DeepVariant DRAGMAP con el genoma sin enmascarar. Es interesante notar que este último muestra muy poca variabilidad en las puntuaciones F1.

Técnica de llamada de variantes	Alineador y REF	Media	Mediana	STD
DeepVariant	BWA y genoma sin enmascarar	0.659223	0.737966	0.111532
DeepVariant	DRAGMAP y genoma sin enmascarar	0.659020	0.736652	0.113200
HaplotypeCaller	DRAGMAP y genoma sin enmascarar	0.638437	0.687124	0.099279
HaplotypeCaller	BWA y genoma sin enmascarar	0.629462	0.682403	0.114125
HaplotypeCaller	DRAGMAP y genoma enmascarado	0.372943	0.385458	0.046894
HaplotypeCaller	BWA y genoma enmascarado	0.368477	0.383906	0.051930
DeepVariant	DRAGMAP y genoma enmascarado	0.354633	0.363495	0.029918
DeepVariant	BWA y genoma enmascarado	0.302121	0.304861	0.021704
Strelka	DRAGMAP y genoma sin enmascarar	0.233392	0.039948	0.239423
Strelka	BWA y genoma sin enmascarar	0.232361	0.042430	0.237893
Strelka	DRAGMAP y genoma enmascarado	0.130135	0.014681	0.143710
Strelka	BWA y genoma enmascarado	0.121313	0.017781	0.130302
Freebayes	BWA y genoma enmascarado	-	-	-
Freebayes	BWA y genoma sin enmascarar	-	-	-
Freebayes	DRAGMAP y genoma enmascarado	-	-	-
Freebayes	DRAGMAP y genoma sin enmascarar	-	-	-

Tabla 12 Media mediana y desviación típica de la F1 agrupada por técnica, tipo de genoma de referencia y alineador para indels.

Si se observan los F1 en la Tabla 12, solo para los indels, se puede ver que Freebayes no identifica ninguno. Además, en este caso, DeepVariant muestra el mejor rendimiento, seguido por HaplotypeCaller. Por otro lado, Strelka no funciona de manera satisfactoria si se compara con DeepVariant y HaplotypeCaller.

Técnica de llamada de variantes	Alineador y REF	Media	Mediana	STD
DeepVariant	DRAGMAP y genoma sin enmascarar	0.855299	0.859859	0.022039
HaplotypeCaller	BWA y genoma sin enmascarar	0.852230	0.885580	0.058550
HaplotypeCaller	DRAGMAP y genoma sin enmascarar	0.848031	0.876912	0.057266
Freebayes	BWA y genoma sin enmascarar	0.846329	0.878072	0.049314

Strelka	BWA y genoma sin enmascarar	0.846312	0.893565	0.068021
Freebayes	DRAGMAP y genoma sin enmascarar	0.839636	0.873405	0.053021
DeepVariant	BWA y genoma sin enmascarar	0.836138	0.832348	0.024635
Strelka	DRAGMAP y genoma sin enmascarar	0.828704	0.886702	0.081891
DeepVariant	DRAGMAP y genoma enmascarado	0.803011	0.802448	0.021504
Strelka	BWA y genoma enmascarado	0.798352	0.836101	0.057930
Strelka	DRAGMAP y genoma enmascarado	0.777887	0.829213	0.075800
HaplotypeCaller	BWA y genoma enmascarado	0.772667	0.795281	0.046173
HaplotypeCaller	DRAGMAP y genoma enmascarado	0.770270	0.791979	0.044429
Freebayes	DRAGMAP y genoma enmascarado	0.722849	0.745245	0.036934
DeepVariant	BWA y genoma enmascarado	0.693863	0.633823	0.098957
Freebayes	BWA y genoma enmascarado	0.678234	0.683768	0.021781

Tabla 13 Media mediana y desviación típica de la F1 agrupada por técnica, tipo de genoma de referencia y alineador para SNPs.

Al observar la Tabla 13 de los valores de F1 para SNPs, las mejores opciones serían HaplotypeCaller BWA con el genoma sin enmascarar y DeepVariant DRAGMAP con el genoma sin enmascarar. También se puede ver cómo en SNPs Freebayes y Strelka logran tener un funcionamiento mejor, pero sin lograr los resultados de DeepVariant y HaplotypeCaller.

Paciente	Media	Mediana	STD
1	0.790893	0.809055	0.081415
7	0.790461	0.791456	0.056397
6	0.787855	0.803760	0.079387
5	0.787661	0.808182	0.083636
2	0.786502	0.801727	0.075686
4	0.784256	0.800095	0.078691
9	0.751312	0.742556	0.052395

8	0.747265	0.750148	0.056078
10	0.672402	0.673772	0.050172

Tabla 14 Media mediana y desviación típica de la métrica F1 agrupada por paciente.

En la Tabla 14 se puede apreciar que la media de la métrica F1 por paciente tiene cierta correlación con la cobertura, a excepción del paciente 7, aunque si se aprecia la mediana esta correlación se cumple.

Paciente	Media	Mediana	STD
1	0.824014	0.843368	0.081920
6	0.821568	0.839160	0.079523
5	0.820224	0.841232	0.084477
7	0.819949	0.839936	0.054272
2	0.819671	0.835033	0.075835
4	0.817474	0.832657	0.078891
9	0.776587	0.787455	0.051282
8	0.775240	0.793256	0.054936
10	0.708292	0.714934	0.054694

Tabla 15 Media mediana y desviación típica de la métrica F1 agrupados por paciente para SNPs.

Para la media Tabla 15 por paciente de los F1 para SNPs se puede ver cómo la tendencia en función de la cobertura se repite, menos otra vez para el paciente siete.

Paciente	Media	Mediana	STD
7	0.477430	0.466860	0.141819
9	0.459241	0.447517	0.134039
8	0.421680	0.407396	0.137901

1	0.372437	0.387926	0.301270
6	0.369468	0.379851	0.296871
5	0.366749	0.381684	0.300561
2	0.366345	0.374268	0.292571
4	0.363598	0.369128	0.294868
10	0.329190	0.310615	0.089749

Tabla 16 Media mediana y desviación típica de la métrica F1 agrupados por paciente para indels.

Luego para los indels como se puede observar en la Tabla 16 la tendencia cambia completamente siendo los que más F1 tienen, los que tienen menor cobertura.

La razón por la cual los F1 son diferentes para los indels al calcular la media y la mediana es que, por alguna razón, Strelka diverge más del gold standard a medida que aumenta la cobertura de las bases. La causa exacta de esta divergencia es desconocida. Quizás se deba a que a la hora de generar los haplotipos según la cobertura elige entre dos técnicas.

Técnica	Paciente	Alineador y genoma de referencia	F1
Strelka	2	DRAGMAP y genoma sin enmascarar	0.039948
Strelka	2	BWA y genoma sin enmascarar	0.038033
Strelka	2	BWA y genoma enmascarado	0.017781
Strelka	2	DRAGMAP y genoma enmascarado	0.014681

Tabla 17 F1 scores de Strelka y DRAGEN en indels para datos con buena cobertura.

Técnica	Paciente	Alineador y genoma de referencia	F1
Strelka	8	DRAGMAP y genoma sin enmascarar	0.449966
Strelka	8	BWA y genoma sin enmascarar	0.444444
Strelka	8	DRAGMAP y genoma enmascarado	0.231604
Strelka	8	BWA y genoma enmascarado	0.220605

Tabla 18 F1 scores de Strelka y DRAGEN en indels para un paciente con mala cobertura.

Ahora se va a pasar a ver mediante diagramas de Venn cual es el mejor rendimiento y el peor según si se están evaluando todas las variantes, indels o SNPs

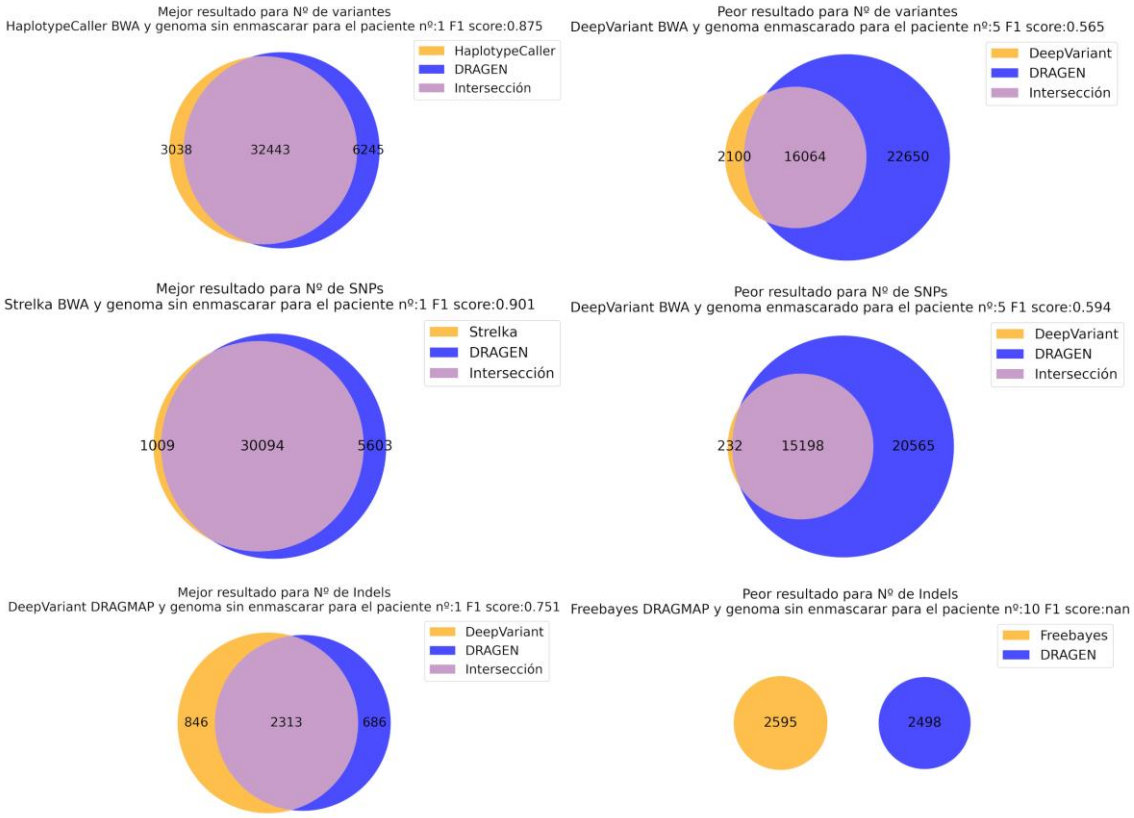

Figura 37 Diagramas de Venn para el mejor y, peor caso según el F1 score en todas las variantes, indels y SNPs.

En los diagramas de Venn de la Figura 37 se puede observar al igual que en la sección anterior que los mejores resultados se logran con el genoma sin enmascarar, utilizando HaplotypeCaller para todas las variantes, DeepVariant para los indels y Strelka para los SNPs. También se observa una diferencia en el rendimiento entre los SNPs y los indels, siendo estos últimos menos precisos. Curiosamente, el paciente 1 obtiene los mejores resultados en todos los tipos de variantes a pesar de no ser el que mayor cobertura tiene.

En contraste, los peores resultados se obtienen con el genoma enmascarado, y DeepVariant muestra una alta variación según los datos de entrada, pudiendo ofrecer tanto los mejores como los peores resultados. En el peor escenario, la suma de errores por falsos positivos y negativos asciende a varios miles de variantes. Aunque esta cifra es significativa, debe considerarse en contexto. Por ejemplo, 25.000 variantes en discrepancia pueden parecer muchas, pero en comparación con los 45 millones de bases examinadas, la proporción de errores se reduce drásticamente.

Se tiene que señalar que dentro de los indels hay algunos en los que las variantes identificadas son similares, y a la hora de hacer la intersección han sido excluidas.

Estas diferencias entre pares en su mayoría se encuentran en la identificación de alelos. Esto también pasa con algunos SNPs, pero proporcionalmente no son tantos. Este análisis solo se realizó para el paciente número 1 y se encontraron un total de 180 variantes que cumplían con este tipo de diferencias.

CHROM	POS	REF DRAGEN	Indel DRAGEN	REF Ensembl	Indel DeepVariant
1	15388335	C	CCATT, T	C	CCATT
1	21176231	TGCC	T, TGCCGCCGCC	TGCC	T
1	61406543	GC	G	GC	G, GCC
1	85072972	A	AAG, AG	A	AG
1	100196431	GAAAAAAA	GAA, G	GAAAAAA	G, GA
X	7275929	CTT	C	CT	C
X	70744241	A	AT, AAAT	A	AT
X	123911477	C	CAAA, CA	C	CAAA
X	123912216	G	GAAA	G	GAA

Tabla 19 Análisis preliminar de los falsos positivos en indels.

Es importante señalar que ninguno de los archivos VCFs ha sido filtrado por un valor de calidad (QUAL) superior a 20, como se recomienda en la literatura. Si se aplicara este filtro, es posible que las métricas mejoren.

5.3 ANÁLISIS DE TIEMPOS

En esta sección se va a realizar un análisis de tiempos de todas las herramientas usadas y explicadas en apartados anteriores así se podrá decidir mejor cuáles se usarán en nuestro pipeline final, dado que en función del caso de uso puede convertirse en una limitación significativa. Por ejemplo, si se desea secuenciar grandes poblaciones de individuos, el tiempo requerido para procesar cada muestra se puede convertir en un recurso limitante.

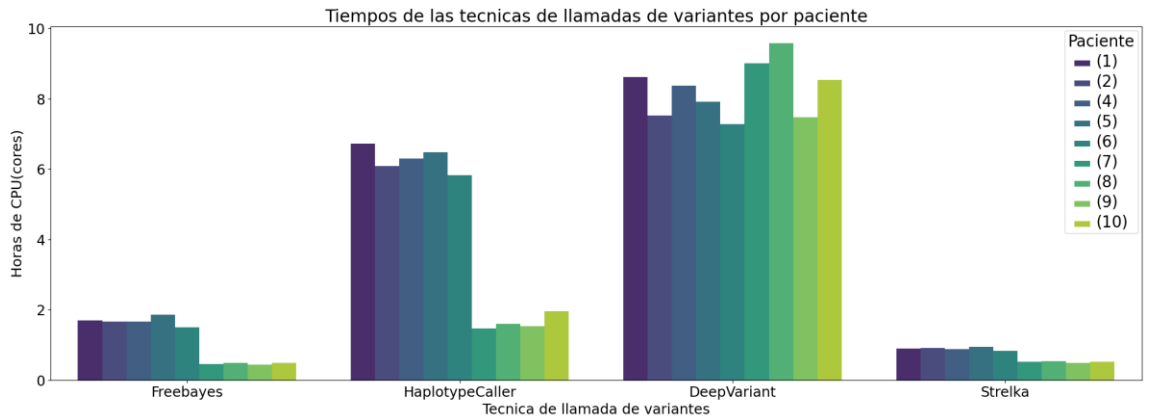

Figura 38 Horas de CPU por paciente y técnica de variantes.

En la Figura 38 se muestra que DeepVariant es la herramienta que más tiempo consume en comparación con las demás, y se observa una consistencia en los tiempos entre diferentes pacientes. HaplotypeCaller, siendo el segundo en términos de tiempo de procesamiento, muestra variaciones más notables entre pacientes. Curiosamente, estas dos herramientas son también las que han demostrado mejores resultados en el análisis de desempeño.

Por otro lado, FreeBayes y Strelka destacan por sus tiempos menores de procesamiento. La eficiencia de tiempo de estas herramientas puede ser ventajosa en situaciones donde se requiera rapidez, pero es fundamental equilibrar la velocidad con la precisión para asegurar la fiabilidad de los resultados, y como se ha visto en el apartado de desempeño en indels tanto Strelka como Freebayes tienen un rendimiento muy lejos del ideal.

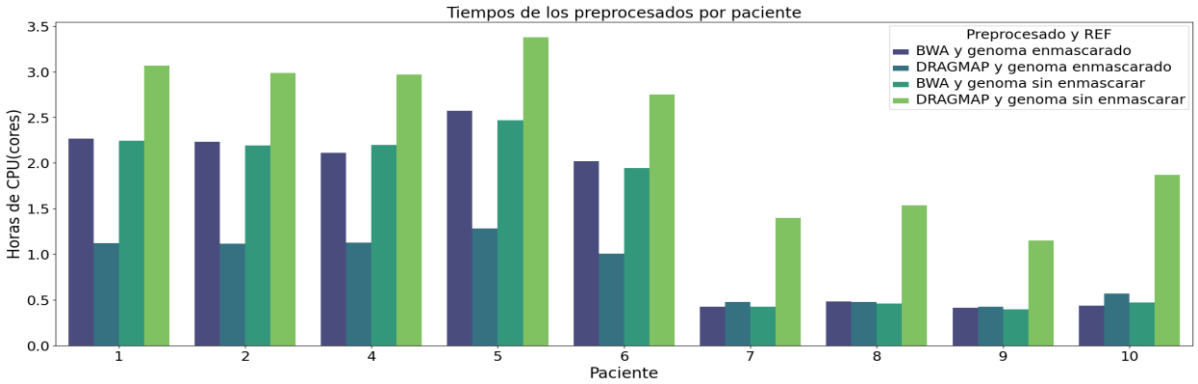

Figura 39 Tiempo de preprocesados por paciente y tipo.

Luego en la fase de preprocesamiento de los FASTQ, se ha observado que el uso de DRAGMAP y un genoma enmascarado resulta en un tiempo de procesamiento menor en comparación con BWA. Con BWA, la diferencia de tiempo entre utilizar un genoma enmascarado o no es mínima. Por otro lado, DRAGMAP con un genoma sin enmascarar muestra un tiempo de procesamiento considerablemente mayor.

5.4 ANÁLISIS DE COSTES COMPUTACIONALES

Al analizar los costes computacionales, es esencial centrarse en los aspectos más críticos que influyen en el rendimiento y la eficiencia. Uno de estos aspectos clave es el pico de memoria en RAM.

Se pudo observar que los picos de memoria en la ejecución del pipeline en todas las ejecuciones se encontraban en el alineador. Por lo tanto, solo se van a graficar estos picos según el alineador y el genoma de referencia.

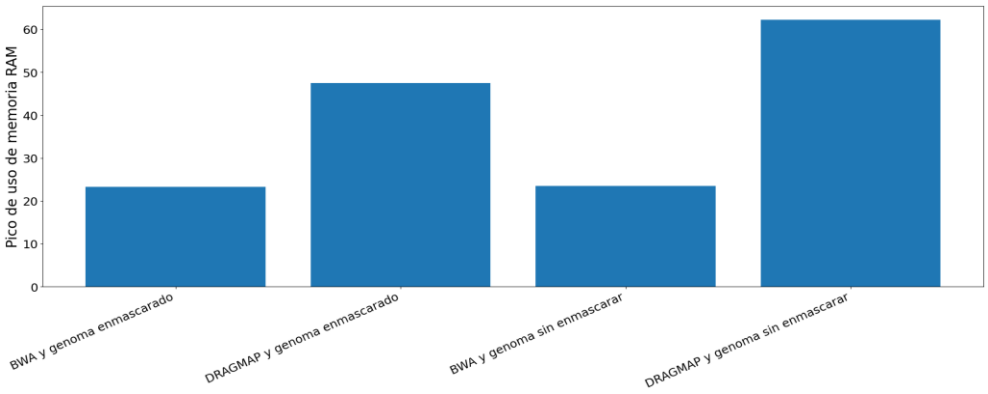

Figura 40 Picos de memoria del pipeline.

Como se puede observar en la Figura 40 DRAGMAP tiene un uso más intensivo de la RAM y se pueden encontrar diferencias en este si se usa con un genoma enmascarado o sin enmascarar.

Para BWA, el uso de la RAM es más reducido y no hay diferencias significativas en el uso de genomas enmascarados.

5.5 CONFIGURACIÓN DEL PIPELINE PARA APROXIMARNOS AL GOLD STANDARD

Tras evaluar las opciones disponibles, es momento de definir el enfoque de nuestro pipeline, diseñado para maximizar los valores de F1.

La incapacidad de Strelka y FreeBayes para identificar indels de manera efectiva sugiere que no son adecuadas para aplicaciones donde la precisión en la identificación de este tipo de variantes es esencial. Por lo tanto, se recomienda la utilización de herramientas más especializadas y precisas para la identificación de indels, como DeepVariant y HaplotypeCaller, que han demostrado un rendimiento superior en las evaluaciones.

En cuanto al genoma, utilizar uno sin enmascarar aumenta el rendimiento de estas herramientas. Para el alineador, se optará por usar DRAGMAP, ya que DeepVariant tiene un rendimiento inferior sin este. Se podría considerar usar BWA y HaplotypeCaller únicamente, pero no sería deseable, dado que DeepVariant tiene un mejor rendimiento con pocas lecturas en comparación con HaplotypeCaller. Por ello, finalmente, el pipeline se dividirá en dos opciones, dependiendo de si se cuenta con una cobertura de 80x o no.

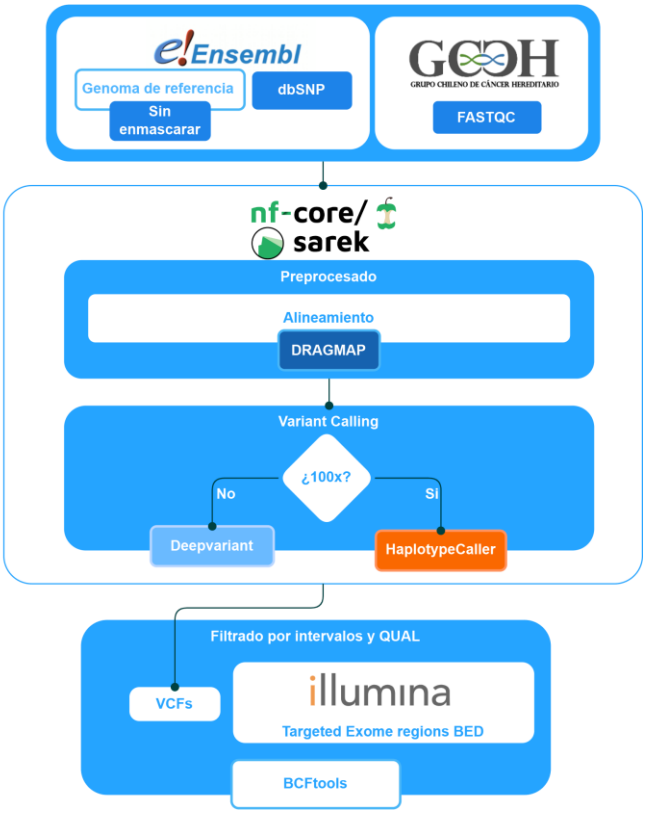

Figura 41 Diagrama de flujo del pipeline.

6 DISCUSIÓN Y COMPARACIÓN CON OTRAS ALTERNATIVAS

En esta sección se discutirán algunos elementos que servirán para el apartado de conclusiones. Siguiendo el enfoque del Design Science, donde se evalúa el artefacto en su contexto, se realizará un análisis de mercado para evaluar la reducción de costes. Luego, se discutirán aspectos importantes del Trabajo Final de Máster.

6.1 COMPARACIÓN CON LAS ALTERNATIVAS DEL MERCADO

Como se ha dicho anteriormente se realizará una comparación de mercado para resolver la duda sobre si nuestro pipeline realmente reduce costes. Para hacer esta estimación solo se tendrá en cuenta el precio de la estación de trabajo y el coste energético. La justificación de estas estimaciones se puede encontrar en el apartado de presupuesto.

En este Trabajo Fin de Máster, se han realizado un total de 144 análisis de variantes. En total estos análisis han requerido un tiempo de ejecución de entre 5 y 6 días, utilizando un sistema con características técnicas similares a las de la estación de trabajo mencionada en la parte del presupuesto. Para fines de este análisis, se supondrá que se realizan 100 secuenciaciones semanales.

Según Illumina en Azure DRAGEN, el coste aproximado por muestra de WGS es de 0.9 céntimos de euro. Estos costes serían incluso menores para WES.

Para este análisis, se va a excluir el coste asociado al almacenamiento de datos en Azure. La razón de la exclusión del almacenamiento es que, si los datos se borran y se realiza una copia de seguridad de manera temprana, este coste no tendría un impacto significativo en nuestras conclusiones. Aun así, un mal uso de estos sistemas cloud puede llevar a un coste adicional sin ser necesario.

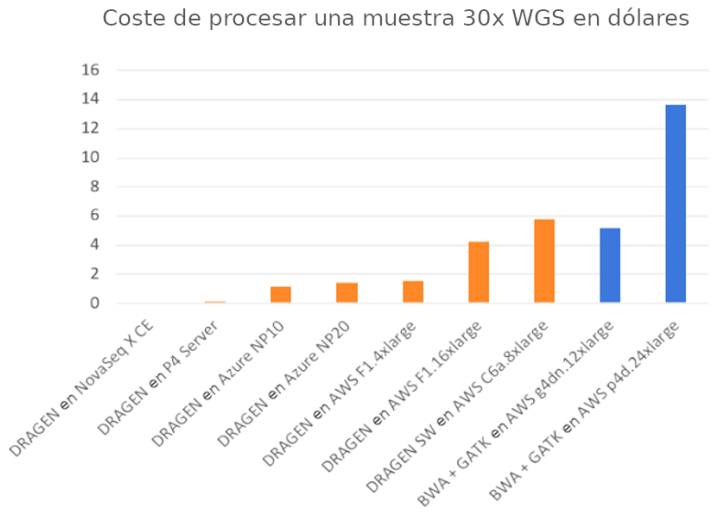

Figura 42 Gráfico de barras modificado del coste de DRAGEN para una muestra de WGS en diferentes plataformas (Illumina s.f.).

Si se observa el coste mes a mes en la Figura 43 para el mismo número de análisis, que en este caso es 400, hasta el mes 23 que correspondería a aproximadamente a dos años, empezaría a ser rentable la inversión de una estación de trabajo.

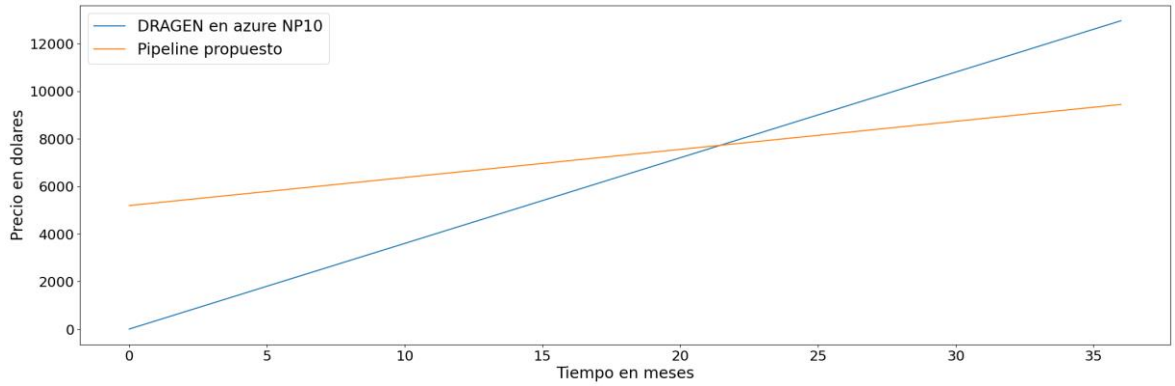

Figura 43 Costes por mes de la estación de trabajo + gasto energético y DRAGEN en Azure P10.

Es importante mencionar que como se observa en Figura 44, si se contara con un HPC con un uso reducido, el pipeline sería rentable desde el primer día.

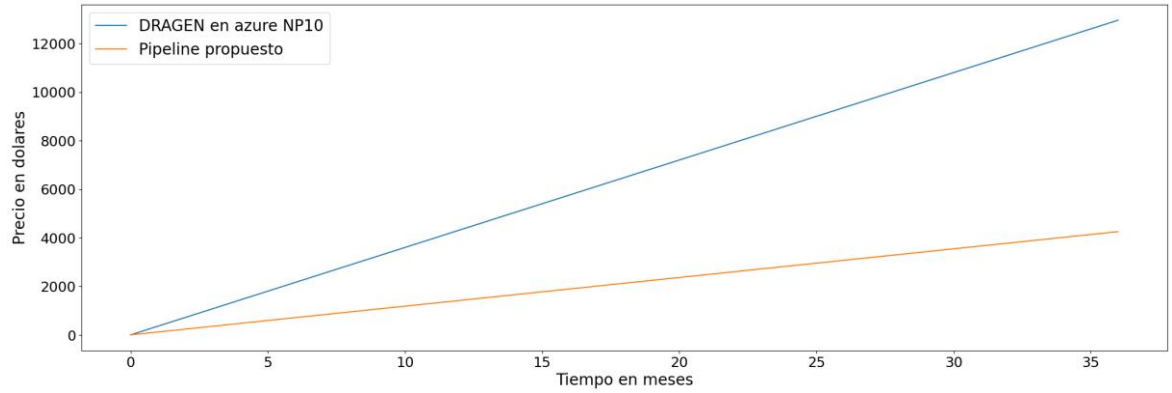

Figura 44 Costes de únicamente el gasto energético y DRAGEN en Azure NPP10.

Aunque esta aproximación es básica y carece de datos validados, las conclusiones preliminares sugieren que, con un HPC o una economía fuerte para aprovechar la economía de escala y más de 400 secuenciaciones mensuales, sería conveniente realizar los análisis localmente si la falta de transparencia de las alternativas de mercado fuese un problema.

Ahora bien, si no se cumplen estas condiciones, se recomienda encarecidamente usar estos servicios de en la nube. Esto se debe a que se está suponiendo que la estación de trabajo no falla en dos años con un uso muy intensivo, cosa que es poco probable. En resumen, la decisión entre utilizar el servicio de DRAGEN o invertir en un HPC propio dependerá de la escala y duración del proyecto, así como del presupuesto disponible.

6.2 DISCUSIÓN

A continuación, se discutirán aspectos clave del *variant calling*, como la cobertura y el número de lecturas del genoma de referencia. También se abordarán las técnicas de variantes, los costes económicos y las limitaciones del *gold standard*. Además, se explorará la falta de *benchmarks* en este campo.

6.2.1 Cobertura, lecturas y genoma enmascarado, y su importancia

Se observó que tanto la cobertura, el número de lecturas y el enmascaramiento del genoma son cruciales para los resultados del pipeline. Para acercarnos al gold standard, es necesario utilizar un genoma sin enmascaramiento, lo cual contradice las afirmaciones del blog de Illumina. Esta discrepancia podría deberse a que Illumina usa un genoma de referencia propio con un enmascaramiento menos restrictivo.

Además, la cobertura es esencial para maximizar la eficacia de estas técnicas. La cobertura recomendada para WES es de entre 80x y 120x, esta métrica indica la mediana de la cobertura de las bases. Sin embargo, el otro 50% puede no tener la calidad adecuada, lo que puede llevar a errores en las técnicas usadas en esas zonas. Sería recomendable hacer un análisis basado en las regiones de cobertura, pero por falta de tiempo no ha sido posible.

6.2.2 Técnicas de llamadas de variantes

En cuanto a las técnicas específicas, tanto DeepVariant como HaplotypeCaller son las que mejor funcionan, considerando tanto los SNPs como los indels. En el caso de los indels, Freebayes y Strelka no obtuvieron resultados satisfactorios

Es relevante señalar que DeepVariant es muy sensible al alineador utilizado, y a diferencia de otras técnicas, es más consistente en su desempeño incluso cuando hay pocas lecturas que respalden las variantes. También se observó que alrededor del 30% o 20% de los indels que no se encuentran en DRAGEN, en comparación con DeepVariant, se deben a problemas en la identificación de alelos.

Si se observan los tiempos, siendo los que menor tiempo tardan los mejores, los que destacan son Strelka2 y FreeBayes.

En cuanto a los picos de memoria RAM, estos se encuentran para todos los casos en el alineador, siendo el que mayores picos tiene DRAGMAP. Este último aumenta si se emplea un genoma sin enmascarar; BWA es más estable en este aspecto.

6.2.3 Costes económicos

En cuanto a los costes económicos, dado que este pipeline requiere un clúster o servidor, y debido a no alcanzar un F1 mayor a 0.99, si no se espera realizar una secuenciación masiva (más de 100 secuenciaciones por semana), sería recomendable utilizar la alternativa de DRAGEN en los servidores de Azure. Solo en casos en los que se necesite comprender el funcionamiento detallado de cada herramienta, se recomendaría utilizar nuestro propio pipeline para así poder identificar posibles errores o incluso pensar en mejoras de estos.

6.2.4 El gold standard no es perfecto

Es importante destacar que lo que se ha considerado gold standard no está exento de imperfecciones. Los desarrolladores reconocen que sus métodos pueden generar un número considerable de resultados erróneos, fluctuando entre sobre los 4.000 y 2.000 para indels y entre 30.000 y 7.000 para SNPs si se suman los falsos positivos y negativos (Demystifying the Versions of GRCh38/Hg38 Reference Genomes, How They Are Used in DRAGEN and Their Impact on Accuracy).

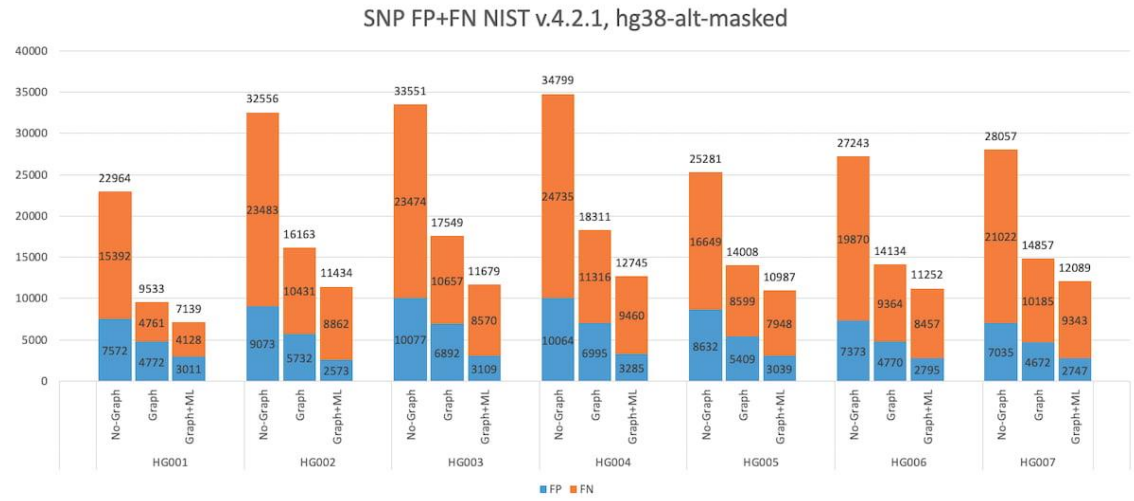

Figura 45 Rendimiento de DRAGEN en SNPs en los benchmarks del GIAB (Demystifying the Versions of GRCh38/Hg38 Reference Genomes, How They Are Used in DRAGEN and Their Impact on Accuracy).

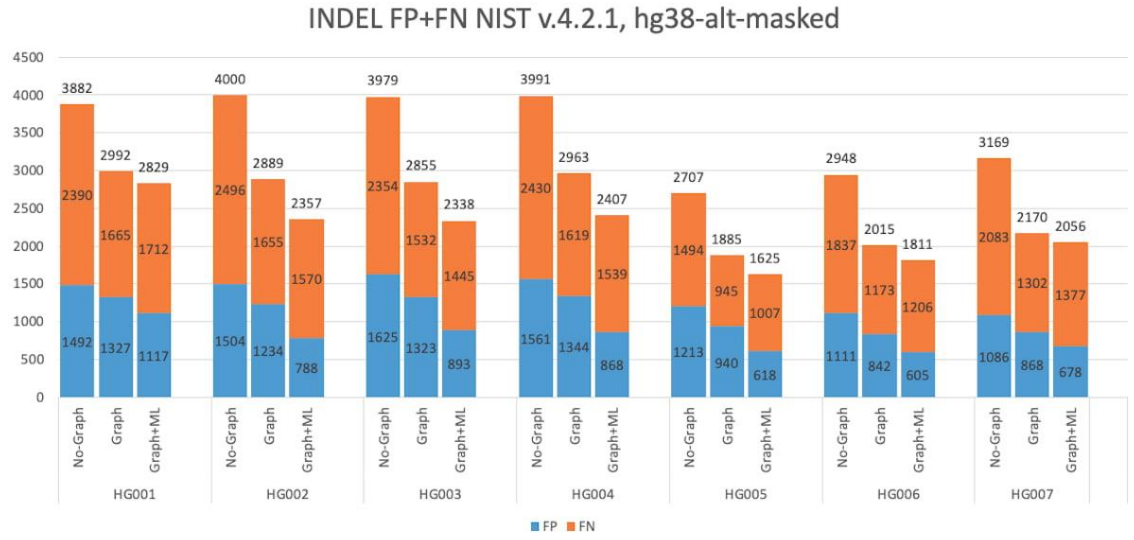

Figura 46 Rendimiento de DRAGEN en indels en los benchmarks del GIAB (Demystifying the Versions of GRCh38/Hg38 Reference Genomes, How They Are Used in DRAGEN and Their Impact on Accuracy).

No se especifica claramente si se refieren a la región "difficult to map" o a todas las regiones. Sin embargo, se pueden hacer algunas suposiciones. Sabiendo que la mayoría de los exomas están dentro de esta región, se pueden estimar los errores entre falsos positivos y negativos en los exomas.

Para ello, se necesita saber cuántas variantes se pueden encontrar en los exomas. Según (Seaby et al., 2015), en promedio, hay 25.000 variantes en los exomas humanos. Sabiendo esto, y que DRAGEN obtiene un F1 de 0.96 en el *benchmark* del GIAB, se puede concluir que en los VCFs de DRAGEN hay aproximadamente 1.600 variantes erróneas, asumiendo que los falsos negativos y positivos son iguales, es decir, 800 cada uno. También se asume que los errores en los aproximadamente 300 millones de nucleótidos analizados en la región "difficult to map" son uniformes.

6.2.5 Benchmarks y las diferencias con nuestro caso de uso

Otro aspecto que considerar es que los *benchmarks* predeterminados HG003, HG004 y HG002, son todos pertenecientes a la misma etnia siendo estos padre madre e hijo. También existen otros *benchmarks*, como el del trío chino (HG005, HG006, HG007). Debido a la escasez de pacientes y la mínima variabilidad, una condición que no refleja la diversidad humana es posible que estos *benchmarks* sean insuficientes para validar estas herramientas.

Además, estos pacientes tienen una cobertura de 30x en WGS y según la literatura esto es mejor que un mapeo de 80x en WES (Sun et al., 2021). Por lo tanto, sin duda, se necesitan *benchmarks* sólidos que representen un verdadero reto para estos sistemas y que estén dedicados solo a WES.

6.3 RESPUESTAS A LAS PREGUNTAS DE INVESTIGACIÓN DEL TERCER SUBOBJETIVO

Ahora a modo de resumen de la sección donde se trata el tercer subobjetivo se responderán a las preguntas de investigación asociadas a este.

PI6: ¿En cuánto se diferencian los resultados de nuestro pipeline al *gold standard*?

En la sección 5, se evalúa el rendimiento de las herramientas seleccionadas y se eligen las más adecuadas. Los mejores resultados se obtuvieron utilizando DeepVariant y DRAGMAP con un genoma sin enmascarar. Además, DeepVariant ha demostrado ser más efectivo en datos con una cobertura de secuenciación menor a 80x.

El mejor resultado, según la métrica F1 para todas las variantes, es de 0.875 en el mejor de los pacientes, con una media de 0.83 en todos los pacientes. Al diferenciar entre indels y SNPs, se observa una mayor similitud en los SNPs que en los indels. En estos últimos, muchas de las diferencias se debían a la identificación de alelos en el mejor paciente entre DRAGEN y DeepVariant.

Sin embargo, debido a la alta sensibilidad de DeepVariant al mapeo de lecturas, en nuestro pipeline final se incluirá una bifurcación condicional basada en la cobertura de secuenciación del paciente. Si la cobertura es inferior a 80x, se empleará DeepVariant; en caso contrario, se optará por HaplotypeCaller, mapeando todos los casos con DRAGMAP y un genoma sin enmascarar.

PI7: ¿Hasta qué punto nuestra herramienta es útil?

En la sección 6 se analiza la utilidad de nuestra herramienta, la cual depende de varios factores clave: rendimiento, tiempo, coste y comprensión del proceso completo.

Nuestra herramienta resulta especialmente útil en varios escenarios. Por ejemplo, si se dispone de un HPC con poco uso, se puede aprovechar al máximo su capacidad sin incurrir en costes adicionales. Además, para aquellos que desean entender todos los procesos involucrados en la secuenciación y análisis de datos genómicos, nuestra herramienta ofrece una transparencia y control que otras soluciones, como DRAGEN, pueden no proporcionar.

Las diferencias con DRAGEN son asumibles, especialmente si se considera la flexibilidad y personalización que nuestra herramienta permite. Finalmente, si se planea realizar más de 100 secuenciaciones semanales, nuestra herramienta se convierte en una opción muy viable, ya que puede reducir los costes si el proyecto es de larga duración.

7 CONCLUSIONES Y TRABAJO FUTURO

Al inicio de este Trabajo de Fin de Máster, se formularon las preguntas de investigación relacionadas con nuestro objetivo principal, el cual es **diseñar, desarrollar y validar un pipeline de código abierto y transparente para la identificación de variantes genómicas**. Por lo tanto, la principal contribución de este trabajo es un pipeline para la identificación de variantes genómicas.

En esta sección se presentan las respuestas a las preguntas de investigación y las conclusiones de los tres subobjetivos específicos de este Trabajo Fin de Máster, descritos previamente. Todas las preguntas se enmarcan en el objetivo principal de desarrollar un pipeline para la identificación de variantes genómicas, el cual se divide en subobjetivos que contienen una o varias preguntas de investigación.

El análisis realizado en este trabajo demuestra que, a pesar de costes computacionales y tiempos de procesamiento, **DeepVariant** y **HaplotypeCaller** son las herramientas más efectivas para la identificación de variantes genómicas. Estas herramientas superan a otras en términos de precisión y consistencia, aunque DeepVariant muestra una mayor sensibilidad al alineador. Además, se ha observado que el uso de un genoma enmascarado tiende a empeorar los resultados, sugiriendo que es preferible trabajar con genomas sin enmascarar para obtener una mayor exactitud en la identificación de variantes. Por otro lado, herramientas como **Strelka** y **FreeBayes**, no alcanzan la misma precisión, especialmente en la identificación de indels.

Por esta razón, se han seleccionado DeepVariant, HaplotypeCaller, DRAGMAP y un genoma sin enmascarar para construir el pipeline objetivo.

La implementación del pipeline requiere un HPC, lo que implica altos costes económicos. Por lo tanto, si no se planea una secuenciación masiva, puede ser más rentable utilizar servicios en la nube de DRAGEN en Azure.

Acabadas las conclusiones se pasará a los subobjetivos y las preguntas de investigación asociadas.

Subobjetivo 1: Entender el proceso de identificación de variantes (*variant calling*). Se realizará un estudio del estado del arte y las herramientas existentes, identificando sus fortalezas y limitaciones.

PI1: ¿Cómo se obtienen las secuencias de nucleótidos para el *variant calling*?

En la sección 3.3 se han definido y explicado las técnicas para obtener secuencias de nucleótidos, destacando las técnicas de Sanger y la secuenciación de nueva generación (NGS). Actualmente, el variant calling se realiza con secuencias obtenidas mediante NGS, un proceso que incluye varios pasos. Se ha tratado en detalle la secuenciación por síntesis, utilizada por Illumina, la empresa líder en el mercado. En este método, el ADN se fragmenta y se le añaden adaptadores generando una biblioteca. Luego, la biblioteca se desnaturaliza para formar cadenas individuales de ADN, que se fijan a una celda de flujo y se amplifican. Finalmente, se añaden nucleótidos fluorescentes y se secuencian las cadenas.

PI2: ¿Cómo se codifican las secuencias de nucleótidos para su análisis?

En la sección 3.4 se ha explicado cómo se codifican las secuencias de nucleótidos y los archivos usados como pasos intermedios, tales como FASTA, FASTQ, BAM y VCF. Estas codificaciones son un estándar y su comprensión ha sido necesaria para extraer información relevante, como la decodificación del

filtrado de los VCFs. Las secuencias de nucleótidos se codifican para su análisis utilizando varios formatos estandarizados, cada uno con un propósito específico. Estos propósitos se pueden ver en la sección 3.4 o en las respuestas de investigación de la sección 3.

PI3: ¿Cómo se identifican las variantes genómicas?

En las secciones 3.5 se han explicado algunas de las herramientas más utilizadas para la identificación de variantes, el preprocesamiento de las secuencias de nucleótidos y la extracción de información sobre la calidad de estas.

La identificación de variantes genómicas se realiza en dos etapas principales: preprocesado y identificación. En la etapa de preprocesado, las secuencias de nucleótidos se alinean contra un genoma de referencia y los datos se recalibran para corregir errores y mejorar la precisión.

En la etapa de identificación, se utilizan diversas herramientas de análisis, cada una con su propio método de funcionamiento. Algunas de las más comunes son DeepVariant, que utiliza redes neuronales convolucionales para clasificar e identificar variantes; FreeBayes, que emplea el teorema de Bayes para calcular la probabilidad de variantes en un lugar genómico; Strelka2, que modela errores de secuenciación y utiliza un modelo de mezcla para identificar variantes; y HaplotypeCaller, que genera haplotipos y utiliza un PairHMM para identificar variantes.

Subobjetivo 2: Diseñar y desarrollar un pipeline para identificar variantes genómicas.

PI4: ¿Cómo obtener un pipeline para identificar variantes genómicas?

Para identificar variantes genómicas, primero se seleccionan las herramientas de alineación e identificación. Además, es necesario escoger una fuente de datos para el genoma de referencia y el dbSNP. En la sección 4, tras explicar las herramientas, se procederá a utilizarlas y detallar su ejecución para una posterior evaluación en la sección 5 y, a partir de esta validación, realizar una selección.

Una herramienta que integra un gran número de estas es NF-Sarek, debido a su capacidad para combinar múltiples herramientas de análisis genómico de alta precisión, incluyendo técnicas de GATK4, con alineadores BWA y DRAGMAP. Con los archivos BAM generados, se obtienen los archivos VCF utilizando DeepVariant, Strelka, HaplotypeCaller y FreeBayes.

En general, BWA es considerado el estado del arte en cuanto a alineadores, aunque los autores de DRAGMAP afirman que esta herramienta es superior y está destinada a reemplazarlo. En cuanto a las técnicas de identificación de variantes, DeepVariant puede considerarse un pseudo estado del arte, dado su rendimiento notable en el desafío PrecisionFDA V2. Sin embargo, en WES, DRAGEN es superior.

Respecto al uso de otras herramientas, aunque no sean las mejores en términos de fiabilidad de resultados, son significativamente más rápidas según la documentación. Por esta razón, se eligieron para evaluar si la posible pérdida de precisión podría compensarse con los beneficios de menores tiempos de procesamiento y recursos utilizados.

PI5: ¿Qué hardware es necesario para implementar este pipeline?

Se requirió una infraestructura de alto rendimiento (HPC) para implementar el pipeline. Esta infraestructura, descrita en la Sección 4.2, contaba con un mínimo de 28 núcleos de CPU y 60 GB de RAM, siendo estas configuraciones habituales en el campo de la identificación de variantes.

Subobjetivo 3: Validar el pipeline con datos de pacientes reales frente a los resultados dados por una empresa líder en el sector (*gold standard*).

PI6: ¿En cuánto se diferencian los resultados de nuestro pipeline al *gold standard*?

En la sección 5, se evalúa el rendimiento de las herramientas seleccionadas y se eligen las más adecuadas. Los mejores resultados se obtuvieron utilizando DeepVariant y DRAGMAP con un genoma sin enmascarar. Además, DeepVariant ha demostrado ser más efectivo en datos con una cobertura de secuenciación menor a 80x.

El mejor resultado, según la métrica F1 para todas las variantes, es de 0.875 en el mejor de los pacientes, con una media de 0.83 en todos los pacientes. Al diferenciar entre indels y SNPs, se observa una mayor similitud en los SNPs que en los indels. En estos últimos, muchas de las diferencias se debían a la identificación de alelos en el mejor paciente entre DRAGEN y DeepVariant.

Sin embargo, debido a la alta sensibilidad de DeepVariant al mapeo de lecturas, en nuestro pipeline final se incluirá una bifurcación condicional basada en la cobertura de secuenciación del paciente. Si la cobertura es inferior a 80x, se empleará DeepVariant; en caso contrario, se optará por HaplotypeCaller, mapeando todos los casos con DRAGMAP y un genoma sin enmascarar.

PI7: ¿Hasta qué punto nuestra herramienta es útil?

En la sección 6 se analiza la utilidad de nuestra herramienta, la cual depende de varios factores clave: rendimiento, tiempo, coste y comprensión del proceso completo.

Nuestra herramienta resulta especialmente útil en varios escenarios. Por ejemplo, si se dispone de un HPC con poco uso, se puede aprovechar al máximo su capacidad sin incurrir en costes adicionales. Además, para aquellos que desean entender todos los procesos involucrados en la secuenciación y análisis de datos genómicos, nuestra herramienta ofrece una transparencia y control que otras soluciones, como DRAGEN, pueden no proporcionar.

Las diferencias con DRAGEN son asumibles, especialmente si se considera la flexibilidad y personalización que nuestra herramienta permite. Finalmente, si se planea realizar más de 100 secuenciaciones semanales, nuestra herramienta se convierte en una opción muy viable, ya que puede reducir los costes si el proyecto es de larga duración.

7.1 TRABAJO FUTURO

En esta sección, se propondrán diversas líneas de trabajo futuro con el objetivo de mejorar el pipeline. Estas propuestas no solo buscarán optimizar su funcionamiento, sino que también se centrarán en la crucial tarea de caracterizar y comprender los errores que puedan surgir. Al identificar y analizar estos errores de manera detallada, se podrá incrementar significativamente la fiabilidad de su aplicación clínica, asegurando así resultados más precisos y confiables en el ámbito médico.

Estas líneas de trabajo futuro se pueden englobar en tres principales puntos:

- **Utilización de otras tecnologías de secuenciación**: Sería altamente beneficioso contar con datos provenientes de diversas tecnologías de secuenciación, como ONT (Oxford Nanopore Technologies) y PacBio (Pacific Biosciences). La integración de estas tecnologías avanzadas puede contribuir significativamente a mejorar la calidad y la precisión de los datos genómicos.
- **Análisis de diferentes fuentes del genoma de referencia**: Es necesario analizar las distintas fuentes del genoma de referencia, como UCSC, NCBI, Ensembl e Illumina. Aunque las versiones 'toplevel' deberían ser idénticas, pueden existir variaciones entre versiones que pueden influir en los resultados. Por ello, es esencial cuantificar y comprender estas diferencias para obtener resultados más precisos. Nuestro pipeline no puede utilizar datos de grafo como genoma de referencia. Este genoma se emplea en la versión de HaplotypeCaller con DRAGMAP, que, según la escasa documentación disponible, es equivalente a DRAGEN. Su uso requiere un genoma de referencia específico que enmascara las alteraciones. Por lo tanto, sería interesante considerar esta técnica y evaluar también este genoma (Functional Equivalence in DRAGEN-GATK, 2024).
- **Cuantificar los errores de las herramientas con datos sintéticos**: Además sería muy interesante probar estas herramientas con datos sintéticos, ya que, como se ha mencionado antes, hay poca variabilidad en los *benchmarks*. Esta aproximación ha sido utilizada para validar otras herramientas biológicas ampliamente utilizadas como BWA.
- **Cuantificar el *trade off* entre WGS y WES:** Es necesario cuantificar los tiempos y costes de los kits, así como la etapa de variant calling subsiguiente para cada técnica. Dado que, lo que se encuentra en la literatura no está claramente definido, lo que genera cierta ambigüedad (Barbitoff et al., 2020)**.**

8 BIBLIOGRAFÍA

Olson, N. D., Wagner, J., McDaniel, J., Stephens, S. H., Westreich, S. T., Prasanna, A. G., Johanson, E., Boja, E., Maier, E. J., Serang, O., Jáspez, D., Lorenzo-Salazar, J. M., Muñoz-Barrera, A., Rubio-Rodríguez, L. A., Flores, C., Kyriakidis, K., Malousi, A., Shafin, K., Pesout, T., . . . Zook, J. M. (2022). PrecisionFDA Truth Challenge V2: Calling variants from short and long reads in difficult-to-map regions. Cell Genomics, 2(5), 100129. https://doi.org/10.1016/j.xgen.2022.100129

Van Der Auwera, G. A., Carneiro, M. O., Hartl, C., Poplin, R., Del Angel, G., Levy-Moonshine, A., Jordan, T., Shakir, K., Roazen, D., Thibault, J., Banks, E., Garimella, K. V., Altshuler, D., Gabriel, S., & DePristo, M. A. (2013). From FastQ Data to High-Confidence Variant Calls: The Genome Analysis Toolkit Best Practices Pipeline. Current Protocols in Bioinformatics, 43(1). https://doi.org/10.1002/0471250953.bi1110s43

Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. Bioinformatics, 25(14), 1754-1760. https://doi.org/10.1093/bioinformatics/btp324

Yoo, A. B., Jette, M. A., & Grondona, M. (2003). SLURM: Simple Linux Utility for Resource Management. En Lecture notes in computer science (pp. 44-60). https://doi.org/10.1007/10968987_3

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., Whitwham, A., Keane, T., McCarthy, S. A., Davies, R. M., & Li, H. (2021). Twelve years of SAMtools and BCFtools. Gigascience, 10(2). https://doi.org/10.1093/gigascience/giab008

Poplin, R., Chang, P., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., Newburger, D., Dijamco, J., Nguyen, N., Afshar, P. T., Gross, S. S., Dorfman, L., McLean, C. Y., & DePristo, M. A. (2018). A universal SNP and small-indel variant caller using deep neural networks. Nature Biotechnology, 36(10), 983-987. https://doi.org/10.1038/nbt.4235

Illumina. (s. f.). GitHub - Illumina/DRAGMAP: DRAGEN open-source mapper. GitHub. https://github.com/Illumina/DRAGMAP

Di Tommaso, P., Chatzou, M., Floden, E. W., Barja, P. P., Palumbo, E., & Notredame, C. (2017). Nextflow enables reproducible computational workflows. Nature Biotechnology, 35(4), 316-319. https://doi.org/10.1038/nbt.3820

Garcia, M., Juhos, S., Larsson, M., Olason, P. I., Martin, M., Eisfeldt, J., DiLorenzo, S., Sandgren, J., De Ståhl, T. D., Ewels, P., Wirta, V., Nistér, M., Käller, M., & Nystedt, B. (2020). Sarek: A portable workflow for whole-genome sequencing analysis of germline and somatic variants. F1000Research, 9, 63. https://doi.org/10.12688/f1000research.16665.2

Kim, S., Scheffler, K., Halpern, A. L., Bekritsky, M. A., Noh, E., Källberg, M., Chen, X., Kim, Y., Beyter, D., Krusche, P., & Saunders, C. T. (2018). Strelka2: fast and accurate calling of germline and somatic variants. Nature Methods, 15(8), 591-594. https://doi.org/10.1038/s41592-018-0051-x

Garrison, E., & Marth, G. (2012, 17 julio). Haplotype-based variant detection from short-read sequencing. arXiv.org. https://arxiv.org/abs/1207.3907

Danecek, P., Auton, A., Abecasis, G., Albers, C. A., Banks, E., DePristo, M. A., Handsaker, R. E., Lunter, G., Marth, G. T., Sherry, S. T., McVean, G., & Durbin, R. (2011). The variant call format and VCFtools. Bioinformatics, 27(15), 2156-2158. https://doi.org/10.1093/bioinformatics/btr330

Demystifying the versions of GRCh38/hg38 reference genomes, how they are used in DRAGEN and their impact on accuracy. (s. f.). https://emea.illumina.com/science/genomics-research/articles/dragen-demystifying-reference-genomes.html

Inside DRAGEN and what enables efficient secondary analysis at scale. (s. f.). https://assets.illumina.com/science/genomics-research/articles/secondary-analysis-at-scale.html

Genome in a Bottle | NIST. (2024, 16 abril). NIST. https://www.nist.gov/programs-projects/genome-bottle

Harrison, P. W., Amode, M. R., Austine-Orimoloye, O., Azov, A. G., Barba, M., Barnes, I., Becker, A., Bennett, R., Berry, A., Bhai, J., Bhurji, S. K., Boddu, S., Lins, P. R. B., Brooks, L., Ramaraju, S. B., Campbell, L. I., Martinez, M. C., Charkhchi, M., Chougule, K., . . . Yates, A. D. (2023). Ensembl 2024. Nucleic Acids Research, 52(D1), D891-D899. https://doi.org/10.1093/nar/gkad1049

Kurtzer, G. M., Sochat, V., & Bauer, M. W. (2017). Singularity: Scientific containers for mobility of compute. PloS One, 12(5), e0177459. https://doi.org/10.1371/journal.pone.0177459

Matplotlib: Visualization with Python. (2024). Zenodo. https://doi.org/10.5281/zenodo.11201097

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E. S., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Van Kerkwijk, M. H., Brett, M., Haldane, A., Del Río, J. F., Wiebe, M. W., Peterson, P., . . . Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2

Chen, S., Zhou, Y., Chen, Y., & Gu, J. (2018). fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics, 34(17), i884-i890. https://doi.org/10.1093/bioinformatics/bty560

Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data. (s. f.). https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Pedersen, B. S., & Quinlan, A. R. (2017). Mosdepth: quick coverage calculation for genomes and exomes. Bioinformatics, 34(5), 867-868. https://doi.org/10.1093/bioinformatics/btx699

McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings Of the Python In Science Conferences. https://doi.org/10.25080/majora-92bf1922-00a

Pan, B., Kusko, R., Xiao, W., Zheng, Y., Liu, Z., Xiao, C., Sakkiah, S., Guo, W., Gong, P., Zhang, C., Ge, W., Shi, L., Tong, W., & Hong, H. (2019). Similarities and differences between variants called with human reference genome HG19 or HG38. BMC Bioinformatics, 20(S2). https://doi.org/10.1186/s12859-019-2620-0

Crossley, B. M., Bai, J., Glaser, A., Maes, R., Porter, E., Killian, M. L., Clement, T., & Toohey-Kurth, K. (2020). Guidelines for Sanger sequencing and molecular assay monitoring. Journal of

Veterinary Diagnostic Investigation, 32(6), 767–775. https://doi.org/10.1177/1040638720905833

Reference genome components. (2023, August 29). GATK. https://gatk.broadinstitute.org/hc/en-us/articles/360041155232-Reference-Genome-Components

Precision medicine. (s. f.). Genome.gov. https://www.genome.gov/genetics-glossary/Precision-Medicine#:~:text=Precision%20medicine%20or%20precision%20healthcare,healthcare%20to%20their%20unique%20attributes.

Wieringa, R. J. (2014). Design Science Methodology for Information Systems and Software Engineering. En Springer eBooks. https://doi.org/10.1007/978-3-662-43839-8

Leinonen, R., Sugawara, H., & Shumway, M. (2010). The sequence read archive. Nucleic Acids Research, 39(Database), D19–D21. https://doi.org/10.1093/nar/gkq1019

Rodriguez, R., & Krishnan, Y. (2023). The chemistry of next-generation sequencing. Nature Biotechnology, 41(12), 1709–1715. https://doi.org/10.1038/s41587-023-01986-3

Petersen, B., Fredrich, B., Hoeppner, M. P., Ellinghaus, D., & Franke, A. (2017). Opportunities and challenges of whole-genome and -exome sequencing. BMC Genomic Data, 18(1). https://doi.org/10.1186/s12863-017-0479-5

Timina. (s.f.). Genome.gov. https://www.genome.gov/es/genetics-glossary/Timina

Mutación. (s.f.). Genome.gov. https://www.genome.gov/es/genetics-glossary/Mutacion

Introducing DRAGMAP, the new genome mapper in DRAGEN-GATK. (s.f.). GATK. https://gatk.broadinstitute.org/hc/en-us/articles/4410953761563-Introducing-DRAGMAP-the-new-genome-mapper-in-DRAGEN-GATK

Wikipedia contributors. (2024, April 26). Burrows–Wheeler transform. Wikipedia. https://en.wikipedia.org/wiki/Burrows%E2%80%93Wheeler_transform

Cook, D. (2021, February 8). Improving Variant Calling using Haplotype Information. DeepVariant Blog. https://google.github.io/deepvariant/posts/2021-02-08-the-haplotype-channel/

Freebayes. (s.f.). GitHub - freebayes/freebayes: Bayesian haplotype-based genetic polymorphism discovery and genotyping. GitHub. https://github.com/freebayes/freebayes

Getting started with GATK4. (2024, June 25). GATK. https://gatk.broadinstitute.org/hc/en-us/articles/360036194592-Getting-started-with-GATK4

Kuchenbaecker, K. B., Hopper, J. L., Barnes, D. R., Phillips, K., Mooij, T. M., Roos-Blom, M., Jervis, S., Van Leeuwen, F. E., Milne, R. L., Andrieu, N., Goldgar, D. E., Terry, M. B., Rookus, M. A., Easton, D. F., Antoniou, A. C., McGuffog, L., Evans, D. G., Barrowdale, D., Frost, D., . . . Olsson, H. (2017). Risks of Breast, Ovarian, and Contralateral Breast Cancer for BRCA1 and BRCA2 Mutation Carriers. JAMA, 317(23), 2402. https://doi.org/10.1001/jama.2017.7112

Seaby, E. G., Pengelly, R. J., & Ennis, S. (2015). Exome sequencing explained: a practical guide to its clinical application. Briefings In Functional Genomics, 15(5), 374-384. https://doi.org/10.1093/bfgp/elv054

Functional equivalence in DRAGEN-GATK. (2024, July 8). GATK. https://gatk.broadinstitute.org/hc/en-us/articles/4410456501915-Functional-equivalence-in-DRAGEN-GATK

Farinha, C. M., & Callebaut, I. (2022). Molecular mechanisms of cystic fibrosis – how mutations lead to misfunction and guide therapy. Bioscience Reports, 42(7). https://doi.org/10.1042/bsr20212006

Barbitoff, Y. A., Polev, D. E., Glotov, A. S., Serebryakova, E. A., Shcherbakova, I. V., Kiselev, A. M., Kostareva, A. A., Glotov, O. S., & Predeus, A. V. (2020). Systematic dissection of biases in whole-exome and whole-genome sequencing reveals major determinants of coding sequence coverage. Scientific Reports, 10(1). https://doi.org/10.1038/s41598-020-59026-y

Sun, Y., Liu, F., Fan, C., Wang, Y., Song, L., Fang, Z., Han, R., Wang, Z., Wang, X., Yang, Z., Xu, Z., Peng, J., Shi, C., Zhang, H., Dong, W., Huang, H., Li, Y., Le, Y., Sun, J., & Peng, Z. (2021). Characterizing sensitivity and coverage of clinical WGS as a diagnostic test for genetic disorders. *BMC Medical Genomics*, *14*(1). https://doi.org/10.1186/s12920-021-00948-5

PRESUPUESTO

Diseño y Desarrollo de un pipeline para la identificación de variaciones genómicas en un contexto de medicina de precisión

Documento II

Moisés Ibáñez Henein

Máster en Ingeniería Biomédica

Curso académico 2023/2024

1 PRESUPUESTO

1.1 INTRODUCCIÓN

En esta sección se van a detallar los costes asociados a la ejecución del Trabajo Final de Máster. Se incluyen los costes de mano de obra, materiales, precios unitarios, mediciones y precios parciales, así como el presupuesto total de ejecución por contrata.

Para el cálculo de los costes de personal, se van a considerar las horas dedicadas por el ingeniero que realiza el trabajo, las horas empleadas por los tutores y otros miembros del grupo PROS para el asesoramiento a lo largo del trabajo, así como el coste unitario por hora de cada profesional implicado. Los costes unitarios por hora se han determinado de acuerdo con los criterios de elaboración de presupuestos de I+D publicados en 2018 por la Universidad Politécnica de València.

Para llevar a cabo la reproducción de estos resultados de manera eficiente, no es estrictamente necesario disponer de un HPC completo; sin embargo, es altamente recomendable para maximizar la paralelización y, por ende, minimizar el tiempo de procesamiento. Aun así, para los efectos prácticos de la realización de este presupuesto se supondrá el mínimo producto viable.

Para ver el coste de la estación de trabajo se va a suponer que la vida útil es de tres años y se ha estado usando durante 6 meses, así se podrán calcular las amortizaciones. Para el gasto energético se ha estimado un uso constante de dos semanas a pleno rendimiento.

Para calcular las amortizaciones se usará la siguiente fórmula.

$$Amortizacion\ anual = \frac{Valor\ de\ adquisición}{Vida\ util}$$

1.2 CUADRO DE MANO DE OBRA

En esta sección, se presentará un análisis detallado de la mano de obra involucrada en nuestro proyecto.

	Horas Totales	Coste Total
Científico de datos: Titulado Superior	500h	12.495 €
Tutor: Catedrático de Universidad	45h	2.313 €
Cotutor: Doctor Contratado	80 h	2.516 €
Miembro grupo PROS: Investigador Predoctoral	15h	374,85 €
Coste Total de Personal	17.698,85€	

Tabla 20 Costes de personal

1.3 CUADRO DE MATERIALES

En esta sección, se ofrece un desglose exhaustivo de los materiales necesarios para la ejecución de nuestro proyecto. Primero se verá el software y a continuación el hardware.

Software		
Componente	**Nombre**	**Precio**
Sistema operativo	Debian 12	0 €

Tabla 21 Costes de software

Ahora se evaluará el hardware. Estas son estimaciones, ya que estas ejecuciones han sido realizadas en un HPC que no ha sido utilizado en su totalidad. Por lo tanto, para la estimación de precios, se ha escogido una estación de trabajo con características similares a las usadas en el HPC.

Estación de trabajo		
Componente	**Nombre**	**Precio**
Procesador	Intel Xeon w7-3465X	2889,00 €
Placa base	ASUS Pro W790E-SAGE SE	1257 €
Ventilación	Noctua NH-U14S DX-4677	139,90 €
RAM	Corsair Vengeance LPX DDR4 4x16GB	167,98 €
Caja	MSI MAG FORGE M100R	59,90 €
Fuente de alimentación	Corsair SF1000L 1000W	208,09 €
Almacenamiento primario	Samsung 980 SSD 1TB	81,90 €
Almacenamiento secundario	Seagate Exos X20 SATA HDD 20TB SATA	378,99 €
Monitor	LG 22MR410-B 21.5" FullHD 100Hz FreeSync	79.90 €
Teclado y ratón	Iggual CMK-BUSINESS Teclado + Ratón Negros	8.47 €
	Total	5.271,13 €

Tabla 22 Costes de la estación de trabajo.

Si se calculan las amortizaciones de la estación de trabajo para los 6 meses suponiendo que tiene una vida útil de tres años los 5.271,13€ se reducen a 878,52 €.

1.4 CUADRO DE PRECIOS UNITARIOS

En esta sección, se presentará un desglose detallado de los precios unitarios de los materiales y servicios necesarios para nuestro proyecto.

	Unidad	Coste Unitario
Científico de datos: Titulado Superior	Horas	24,99 €
Tutor: Catedrático de Universidad	Horas	51, 40 €
Cotutor: Doctor Contratado	Horas	31,45 €
Miembro grupo PROS: Investigador Predoctoral	Horas	24,99 €

Tabla 23 Cuadro de precios unitarios

Los precios unitarios se pueden encontrar en las tablas de la sección 9.3, ya que el precio de los materiales es por unidad única.

1.5 CUADRO DE MEDICIONES

Algo que se debe calcular es el coste energético de esta estación de trabajo, dado que es un coste adicional que hay que tener en cuenta al tener un consumo energético tan elevado.

El cálculo del coste energético es complejo debido a las fluctuaciones en el precio. Sin embargo, se puede hacer una estimación basándonos en el coste medio por kilovatio hora, y asumiendo que la estación de trabajo opera continuamente durante dos semanas. El coste por kilovatio hora en España a fecha de 05/06/2024 es de 0.164€.

Para calcular el consumo de energía en kilovatios-hora (kWh) del Corsair SF1000L 1000W SFX 80 Plus Gold Full Modular durante un día, se necesitaría saber cuánto tiempo está en uso y la carga a la que está operando.

Para fines de este trabajo se va a suponer que gasta lo 1000W constantemente cosa que no sería así, pero por realizar una aproximación simple.

Potencia en vatios (W): Es la potencia máxima que la fuente de alimentación puede entregar, en este caso, 1000W.

Tiempo en horas (h): El número de horas que la fuente de alimentación está en funcionamiento.

La fórmula para calcular el consumo de energía sería:

$$Consumo\ en\ kWh\ = \frac{Potencia\ en\ W\ * Tiempo\ en\ h}{1000}$$

Por ejemplo, si la fuente de alimentación opera a su máxima capacidad (1000W) durante una hora, el cálculo sería:

$$Consumo\ en\ kWh\ = \frac{1000\ * 1}{1000}\ =\ 1kWh$$

Ahora si se calcula el consumo en kWh durante dos semanas de la estación de trabajo, se obtiene:

$$Consumo\ en\ kWh\ = \frac{1000\ * 24 * 14}{1000}\ =\ 336kWh$$

Por lo tanto, el coste energético del TFM es de 55,10 €.

1.6 CUADRO DE PRECIOS PARCIALES

Una vez obtenidos todos los costes, se pueden ver en la tabla de los costes parciales de cada uno de los apartados anteriores.

Presupuesto Total Coste Total de Personal	17.698,85€
Coste Total de Software	0 €
Coste Total de Hardware	878,52 €
Coste energético	55,10 €

Tabla 24 Cuadro de precios parciales.

1.7 PRESUPUESTO DE EJECUCIÓN POR CONTRATA

Sumando los valores de la tabla anterior, se obtiene que el coste de ejecución por contrata es de 18.632,47 €. Para el total, hay que añadir el 21% de IVA, lo que resulta en 22.545,28 €.

ANEXO

Diseño y Desarrollo de un pipeline para la identificación de variaciones genómicas en un contexto de medicina de precisión

Documento III

Moisés Ibáñez Henein

Máster en Ingeniería Biomédica

Curso académico 2023/2024

ANEXO

Software	Versión
bcftools	1.18
gatk4	4.5.0.0
deepvariant	1.5.0
dragmap	1.2.1
bwa	0.7.17.post1188
pigz	2.3.4
samtools	1.19.2
fastp	0.23.4
fastqc	0.12.1
freebayes	1.3.6
mosdepth	0.3.8
samtools	1.19.2
strelka	2.9.10
vcftools	0.1.16
Nextflow	23.10.1
nf-core/sarek	v3.4.2-gb5b766d

Todos los archivos son descargados en la fecha 02/05/2024