

Universidad San Jorge
Facultad de Ciencias de la Salud
Bioinformática

Proyecto Final

**Análisis comparativo de metodologías y
herramientas para el estudio de los
distintos tipos de variantes en el genoma
completo**

Autor del proyecto: Álvaro Gaya Pérez
Director del proyecto: Dra. Ángela Jimeno Martin
Zaragoza, 8 de septiembre de 2024

Este trabajo constituye parte de mi candidatura para la obtención del título de Graduado en Bioinformática por la Universidad San Jorge y no ha sido entregado previamente (o simultáneamente) para la obtención de cualquier otro título.

Este documento es el resultado de mi propio trabajo, excepto donde de otra manera esté indicado y referido.

Doy mi consentimiento para que se archive este trabajo en la biblioteca universitaria de Universidad San Jorge, donde se puede facilitar su consulta.

Firma



Fecha 06/09/2024

Contenido

1. Resumen	4
1. Abstract	4
2. Introducción	5
2.1. El genoma humano	5
2.2. Secuenciación	7
2.3. Variantes	8
2.4. Aplicaciones del análisis de variantes	13
3. Antecedentes / Estado del Arte	15
4. Objetivos	17
5. Metodología	18
5.1. Selección de herramientas	18
5.2. Evaluación de FreeBayes para SNP e INDEL	19
5.3. Evaluación de Manta para variaciones estructurales (SV)	20
5.4. Evaluación de CNVnator para variaciones en el número de copias (CNV)	21
5.5. Proyecto 1000 Genomas	22
6. Implementación	24
6.1. Instalación de Conda usando Miniconda en Linux	25
6.2. Freebayes con Conda	27
6.3. Manta con Conda	28
6.4. CNVnator con Conda	30
6.5. Filtrado de variantes	31
6.5.1. Filtrado de variantes – SNP e INDEL	32
6.5.2. Filtrado de variantes – variantes estructurales (SV, en inglés)	35
6.5.3. Filtrado de variantes – variación del número de copias (CNV, en inglés)	35
7. Estudio Económico	38
8. Resultados	40
8.1. Freebayes	40
8.2. Manta	47
8.3. CNVnator	52
8.4. Fortalezas y limitaciones de cada herramienta	56
8.5. Integración de un protocolo de análisis que combine las tres herramientas	59
9. Conclusiones	61
10. Referencias	62
11. Anexos	68
11.1. Anexo I	68
11.2. Anexo II	71
11.3. Anexo III	73
11.4. Anexo IV	76
11.5. Anexo V	80

1. Resumen

Existen numerosas herramientas bioinformáticas diseñadas para la identificación de variantes en el genoma humano. El objetivo principal del proyecto es diseñar un protocolo de análisis de variantes que combine distintas herramientas para mejorar la detección de cada tipo de variante. Tras realizar una revisión bibliográfica, se seleccionó FreeBayes como la herramienta principal para la identificación de polimorfismos de un solo nucleótido (SNP) y pequeñas inserciones y deleciones (INDEL) inferiores a 50 bases, Manta como detector de variantes estructurales y CNVnator como identificador de variaciones en el número de copias (CNV). FreeBayes muestra un alto rendimiento en la identificación de SNP y una capacidad moderada para detectar INDEL, con limitaciones en la detección de variantes estructurales grandes. Manta destaca en la identificación de variaciones estructurales grandes, como inserciones y deleciones, pero es menos efectiva para variantes pequeñas. Por su parte, CNVnator demuestra ser altamente efectivo en la detección de CNV, mostrando robustez incluso con baja cobertura. En conclusión, cada herramienta complementa las limitaciones de las otras, sugiriendo que su uso combinado puede proporcionar un análisis integral de la variabilidad genética en estudios genómicos complejos.

Palabras clave: Bioinformática; Freebayes; Manta; CNVnator; Genómica.

1. Abstract

There are numerous bioinformatics tools designed for the identification of variants in the human genome. The main objective of this project is to design a variant analysis protocol that combines different tools to enhance the detection of each type of variant. After conducting a literature review, FreeBayes was selected as the primary tool for identifying single nucleotide polymorphisms (SNPs) and small insertions and deletions (INDELs) of less than 50 bases, Manta as the detector of structural variants, and CNVnator as the identifier of copy number variations (CNVs). FreeBayes shows high performance in SNP identification and moderate capacity for detecting INDELs, with limitations in detecting large structural variants. Manta excels in identifying large structural variations, such as insertions and deletions, but is less effective for smaller variants. Meanwhile, CNVnator proves to be highly effective in detecting CNVs, demonstrating robustness even with low coverage. In conclusion, each tool complements the limitations of the others, suggesting that their combined use may provide a more comprehensive analysis of genetic variability in complex genomic studies.

Keywords: Bioinformatics; Freebayes; Manta; CNVnator; Genomics.

2. Introducción

2.1. El genoma humano

El genoma se define como el conjunto completo de ADN de un organismo que contiene todas las instrucciones necesarias para el desarrollo, funcionamiento y reproducción del mismo. Este material genético está organizado en estructuras llamadas cromosomas. Los seres humanos tienen un total de 23 pares de cromosomas, 46 en total, de los cuales 22 pares son autosomas y un par son cromosomas sexuales (XX en mujeres y XY en hombres).

Cada cromosoma está compuesto por una larga molécula de ADN de doble hebra, que se encuentra superenrollada para formar una estructura compacta, denominada cromatina, que le permite ser almacenada en el núcleo celular. El ADN, a su vez, es un polímero de nucleótidos. Hay cuatro tipos en el ADN humano: adenina (A), timina (T), citosina (C) y guanina (G). La secuencia en la que estos nucleótidos están dispuestos es lo que determina la información genética.

El ADN humano tiene una longitud aproximada de 3 mil millones de pares de bases. Aunque la mayoría de este ADN es idéntico en todos los seres humanos, pequeñas variaciones en la secuencia de nucleótidos, conocidas como **variantes genéticas**, contribuyen a las diferencias individuales. Estas variantes pueden ser tan simples como un solo cambio de base (SNP, por sus siglas en inglés) o más complejas como inserciones, deleciones y variaciones estructurales (Nurk et al., 2021).

Además de los cromosomas nucleares, todas las células eucariotas también contienen ADN en sus mitocondrias, conocido como ADN mitocondrial (ADNmt). En la especie humana, este ADN es heredado exclusivamente de la madre y tiene una estructura circular, en contraste con el ADN lineal de los cromosomas nucleares. Aunque el ADNmt representa solo una pequeña fracción del genoma total, juega un papel crucial en la producción de energía celular.

El genoma humano está compuesto por regiones codificantes, aquellas cuya secuencia servirá para fabricar proteínas, y regiones no codificantes en las que se encuentran los genes que dan lugar a diversos tipos de ARN funcionales (ARNr, ARNt, etc.) y, además, secuencias que desempeñan roles esenciales en la regulación y expresión de los genes o en el empaquetamiento y disposición de la cromatina. Comprender estas regiones es crucial para entender cómo se controla la expresión de la información genética y el impacto que pueden tener las variantes genéticas en las funciones biológicas en el organismo humano.

A continuación, se presenta una breve descripción de los elementos que constituyen estas regiones:

1. Genes y su estructura

Los genes son segmentos de ADN que contienen la información necesaria para la síntesis de proteínas y moléculas de ARN funcionales (Kanehisa & Goto, 2000). Están compuestos por:

- **Exones:** Secuencias codificantes que son transcritas en ARN mensajero (ARNm) y luego traducidas en proteínas.
- **Intrones:** Secuencias no codificantes que se encuentran entre los exones y son eliminadas del ARNm durante el procesamiento (splicing). Aunque no codifican proteínas, pueden contener elementos reguladores que influyen en la expresión génica y permiten la generación de múltiples isoformas de ARNm, aumentando la diversidad proteica (Shaul, 2017).

2. Secuencias reguladoras

El control de la expresión génica es un proceso esencial que permite a las células responder a señales internas y externas, así como adaptar sus funciones según las necesidades del organismo. Este control se logra en gran medida gracias a las regiones reguladoras del genoma, que incluyen promotores, potenciadores, silenciadores y aislantes.

- **Promotores:** Ubicados antes del inicio de un gen, sirven como sitio de unión para la ARN polimerasa y los factores de transcripción, iniciando la transcripción del ADN a ARN (Yella & Bansal, 2017).
- **Potenciadores:** Secuencias que aumentan la tasa de transcripción de genes específicos. Pueden estar ubicados cerca o lejos del gen que regulan y pueden funcionar en cualquier orientación. Actúan como sitios de unión para factores de transcripción y facilitan la formación del complejo de transcripción (Zabidi & Stark, 2016).
- **Silenciadores:** Secuencias que disminuyen la expresión de genes específicos. Funcionan uniéndose a proteínas represoras que inhiben la transcripción, ya sea bloqueando la unión de la ARN polimerasa o remodelando la cromatina (Hughes et al., 2014).
- **Aislantes:** Barreras que evitan la interacción inapropiada entre elementos reguladores y promotores. Pueden bloquear la acción de los potenciadores para evitar la activación de genes no deseados o impedir la propagación de heterocromatina, lo cual garantiza

que ciertos genes permanezcan accesibles y puedan ser transcritos. De esta forma, los aislantes aseguran una regulación específica de la expresión génica.(Maston et al., 2006).

3. Secuencias repetitivas y regiones no codificantes

- El genoma humano incluye secuencias repetitivas y regiones no codificantes que, aunque no producen proteínas directamente, son cruciales para la regulación génica, la estabilidad del genoma y la evolución.**Secuencias repetitivas:** Incluyen elementos dispersos como retrotransposones (L1) y elementos SINE (Alu), que se encuentran en diferentes partes del genoma y pueden influir en la expresión génica (Biscotti et al., 2015). También incluyen secuencias tandem como microsatélites y minisatélites, cuya variabilidad proporciona información útil en estudios genéticos, aunque su expansión puede causar enfermedades (Tørresen et al., 2019).
- **Regiones no codificantes:** No se transcriben en ARNm para la síntesis de proteínas, pero son cruciales para la regulación génica y la estabilidad del genoma. Incluyen intrones, ARN no codificantes (como microARNs y ARN largos no codificantes), y regiones intergénicas. Los microARNs regulan la expresión génica al unirse a ARNm, mientras que los ARN largos no codificantes pueden influir en la transcripción y la estructura del genoma (Mattick et al., 2023). Las regiones intergénicas, aunque no codifican proteínas, pueden contener elementos reguladores que modulan la actividad de los genes cercanos (Hacisuleyman et al., 2014).

2.2. Secuenciación

La secuenciación del genoma es el proceso que permite determinar la secuencia exacta de nucleótidos en el ADN de un organismo. Este proceso es crucial para entender la estructura y la función del genoma, así como para identificar variantes genéticas que puedan influir en el desarrollo o en la manifestación de patologías. Sin embargo, la secuenciación del genoma presenta una serie de desafíos y requerimientos que han evolucionado a lo largo del tiempo.

Este proceso se realiza en varias etapas que incluyen la preparación de la muestra, la fragmentación del ADN, la lectura de las secuencias y el ensamblado de los fragmentos secuenciados para reconstruir la secuencia completa. Tradicionalmente, este proceso se realizaba mediante técnicas de secuenciación Sanger que, aunque precisas, eran laboriosas y costosas. Estas técnicas requerían la amplificación del ADN y el uso de terminadores fluorescentes para identificar cada base secuencialmente (Margulies et al., 2005). Además, solo permitían secuenciar solo pequeños fragmentos de ADN a la vez y eran limitadas en cuanto a la cantidad de datos que podían generar.

Otra metodología ampliamente utilizada en el análisis de la secuencia de ADN, fueron los microarrays, especialmente para el análisis de variaciones genéticas, como se describirá más adelante. Esta técnica permite el análisis simultáneo de miles de variantes genéticas mediante la hibridación de secuencias de ADN en una matriz sólida. A pesar de su utilidad, los microarrays tienen limitaciones significativas en términos de resolución y capacidad para detectar variantes raras o complejas (Hänzelmann et al., 2013). La necesidad de secuenciar grandes cantidades de ADN, combinada con la demanda de mayor precisión y velocidad, impulsó el desarrollo de tecnologías más avanzadas. La secuenciación de nueva generación (NGS) emergió como una solución revolucionaria, capaz de realizar secuenciaciones masivas en paralelo. Las tecnologías NGS han reducido drásticamente el tiempo y el costo asociados con la secuenciación, permitiendo el análisis de genomas completos en cuestión de días o semanas, a una fracción del costo de los métodos tradicionales y con alta precisión (Korneliussen et al., 2014). Y, lo que es más, pueden proporcionar un alto nivel de cobertura del genoma analizado, altamente relevante para la búsqueda e identificación de diferencias entre individuos. (McKernan et al., 2009).

Estas metodologías modernas también han facilitado el análisis de variaciones genéticas en poblaciones más grandes, lo que es esencial para estudios genómicos de enfermedades complejas o raras, es decir, poco frecuentes. La capacidad de obtener grandes volúmenes de datos de manera rápida y económica ha transformado el campo de la genómica, permitiendo no solo el estudio detallado de variaciones individuales sino también la integración de datos en grandes estudios de asociación genética y en el análisis funcional del genoma.

2.3. Variantes

Las variantes genéticas son diferencias en la secuencia del ADN que ocurren entre individuos, y pueden ser tan simples como un cambio en una única base (nucleótido) o tan complejas como grandes reordenamientos de segmentos cromosómicos. Estas variaciones son fundamentales para la diversidad genética, ya que son responsables de muchas de las diferencias observables entre individuos, como el color de los ojos, la altura y la susceptibilidad a enfermedades.

Las variantes genéticas pueden surgir de diversas maneras. Pueden ser heredadas de los padres (variantes germinales) o pueden surgir *de novo* en una célula somática (variantes somáticas).

La importancia de las variantes genéticas radica en su impacto sobre la función genética y, por ende, en la salud y la enfermedad. Algunas variantes son benignas y no afectan significativamente al individuo, mientras que otras pueden alterar la función de los genes y proteínas, llevando a condiciones patológicas. Por ejemplo, una mutación puntual en un gen puede resultar en una proteína mal plegada que no funciona correctamente, mientras que una delección de una gran región del genoma puede eliminar completamente uno o más genes esenciales. Además, algunas

variantes pueden aumentar la susceptibilidad a enfermedades comunes como el cáncer, la diabetes o enfermedades cardiovasculares, mientras que otras pueden estar directamente asociadas con enfermedades raras y hereditarias (Kircher et al., 2014).

Las variantes genéticas se pueden clasificar en varias categorías, dependiendo de su naturaleza y tamaño.

1. Polimorfismos de **nucleótido único (SNP)**

Los **polimorfismos de nucleótido único (SNP)**, por sus siglas en inglés) son el tipo más común de variación genética en el genoma humano (véase Ilustración 1). Representan un cambio en una sola base de ADN, es decir, una posición específica en el genoma donde puede haber una variación de una sola letra (nucleótido). Por ejemplo, en una secuencia de ADN donde la mayoría de los individuos tienen una adenina (A), algunos pueden tener una guanina (G), citosina (C) o timina (T).

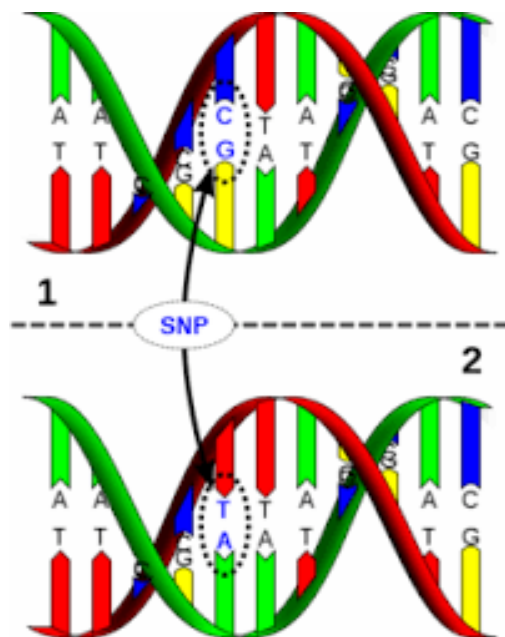


Ilustración 1. Identificación de un SNP.

Estos polimorfismos son altamente frecuentes en el genoma, ocurriendo aproximadamente una vez cada 300 nucleótidos, lo que suma alrededor de 10 millones en el genoma humano. Esta alta densidad hace que sean extremadamente útiles como marcadores genéticos en estudios de asociación genómica. Se pueden categorizar según su ubicación en el genoma: aquellos en las regiones codificantes (exónicos) se encuentran en las áreas de los genes que codifican proteínas. Los polimorfismos sinónimos no alteran la secuencia de la proteína y generalmente se consideran neutros desde el punto de vista funcional. En contraste, los no sinónimos cambian la secuencia de aminoácidos de una proteína, lo que puede

afectar su estructura y función. Estos cambios pueden ser perjudiciales, beneficiosos o neutros. Ejemplos de impactos perjudiciales incluyen la pérdida de función de una enzima o la alteración de una ruta metabólica crítica. Los polimorfismos en regiones reguladoras pueden afectar la expresión génica al alterar sitios de unión para factores de transcripción u otros elementos regulatorios, influyendo en el nivel y la localización de la expresión de los genes (Buniello et al., 2018).

La detección de estos polimorfismos se realiza mediante técnicas de secuenciación y genotipado. Por una parte, la secuenciación de nueva generación (NGS) permite identificar variantes a gran escala con alta precisión, como se ha mencionado anteriormente; por otra, las tecnologías de microarray son ampliamente utilizadas para genotipar polimorfismos en estudios de asociación genómica (GWAS, por sus siglas en inglés), ofreciendo herramientas poderosas para identificar variantes asociadas con enfermedades complejas.

En medicina personalizada, estas variantes son cruciales para predecir la respuesta a medicamentos, la susceptibilidad a enfermedades y su progresión. Por ejemplo, ciertas variantes pueden influir en cómo un paciente metaboliza un principio activo, permitiendo ajustar la dosis para mejorar la eficacia y reducir efectos secundarios (Frazer et al., 2007).

Además, proporcionan información valiosa sobre la evolución humana y la diversidad genética. Al estudiar la distribución de estas variantes en diferentes poblaciones, se pueden inferir patrones de migración, selección natural y la historia evolutiva de la especie.

2. Inserciones y deleciones (INDEL)

Indel: Insertion

Reference Genome	A	C	A	A	T	A	
Example Sequence	A	C	A	G	A	T	A

Indel: Deletion

Reference Genome	A	C	A	G	C	C	A	T	A
Example Sequence	A	C	A		A	T	A		

Ilustración 2. Representación de una inserción (arriba) y una deleción (abajo).

Las **inserciones y deleciones**, comúnmente referidas como **INDEL** (del inglés *insertions and deletions*), son tipos de variantes genéticas que implican la adición o eliminación, respectivamente, de uno o más nucleótidos en una secuencia de ADN (véase Ilustración 2). Estas variantes pueden tener un rango de tamaños desde un solo

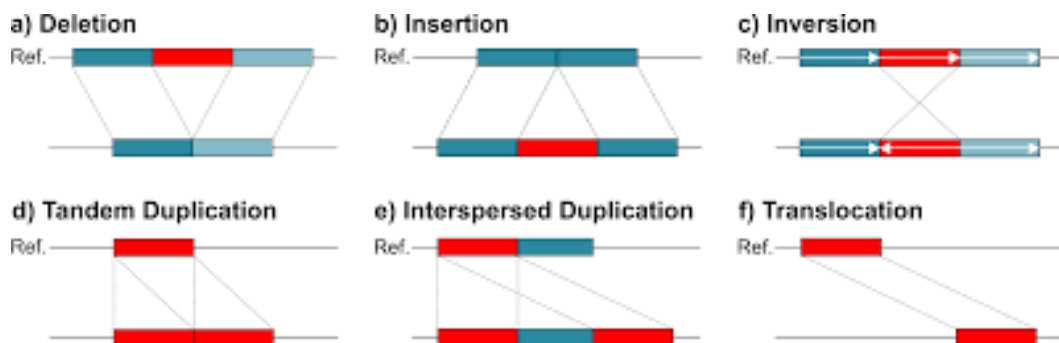
nucleótido hasta varios miles de bases y su presencia puede tener efectos significativos en la función génica y en el fenotipo del organismo. Cabe destacar, entre otras anomalías, que cuando no se inserta o elimina un múltiplo de tres nucleótidos, esto resulta en una alteración del marco de lectura del gen, lo cual puede llevar a la producción de proteínas truncadas con pérdida de función (Choi et al., 2012).

Pueden surgir de varios mecanismos biológicos, como errores durante la replicación o reparación del ADN o por recombinación homóloga. Su identificación y análisis es crucial en estudios genómicos, ya que estas variantes pueden ser responsables de una variedad de enfermedades genéticas y trastornos hereditarios. Por ejemplo, algunas deleciones en genes específicos están

asociadas con condiciones como la fibrosis quística y ciertas formas de distrofia muscular (England et al., 2011).

3. Variantes estructurales (SV)

Las **variantes estructurales (SV)**, por sus siglas en inglés) son cambios en la estructura del genoma que involucran segmentos grandes de ADN, típicamente de 50 bases o más (véase Ilustración 3). Lógicamente, estas variaciones pueden tener un impacto significativo en la estructura de los genes y su regulación. Existen varios tipos de variantes estructurales, incluyendo deleciones, duplicaciones, inversiones, translocaciones e inserciones complejas.



Las deleciones estructurales implican la pérdida de segmentos grandes de ADN, lo que puede

Ilustración 3. Representación de las principales variantes estructurales.

a) Delección, b) Inserción, c) Inversión, d) Duplicación tandem, e) Duplicación intercalada, f) Translocación

resultar en la eliminación de uno o más genes, afectando así la función genética. Por ejemplo, una deleción en el cromosoma 22q11.2 está asociada con el síndrome de DiGeorge, una enfermedad que afecta múltiples sistemas del cuerpo (Sullivan, 2019).

Las duplicaciones estructurales ocurren cuando se repite un segmento del genoma, lo que puede llevar a una dosis génica aumentada de los genes contenidos en ese segmento. Esto puede tener efectos tanto beneficiosos como perjudiciales, dependiendo del contexto. Un ejemplo notable es la duplicación en el gen PMP22, que causa la enfermedad de Charcot-Marie-Tooth tipo 1A, una neuropatía hereditaria (Pentao et al., 1992).

Las inversiones son variantes estructurales en las que un segmento de ADN se invierte en su orientación dentro del genoma. Si bien las inversiones no necesariamente alteran la secuencia de los genes, pueden afectar a su regulación, lo que puede tener consecuencias patológicas.

Las translocaciones implican el intercambio de segmentos de ADN entre diferentes cromosomas. Las translocaciones pueden ser equilibradas, cuando no hay ganancia ni pérdida de material

genético, o desequilibradas, cuando hay una pérdida o ganancia de material genético. Las translocaciones desequilibradas a menudo están asociadas con enfermedades genéticas y ciertos tipos de cáncer, como la leucemia mieloide crónica, que está relacionada con la translocación entre los cromosomas 9 y 22 (conocida como el cromosoma Filadelfia) (Radich et al., 2006).

Por último, las inserciones complejas son variantes estructurales que implican la adición de segmentos grandes de ADN en ubicaciones específicas del genoma. De este modo, al igual que las alteraciones descritas anteriormente, pueden suponer el origen de enfermedades genéticas.

La detección y análisis de variantes estructurales son esenciales para comprender si hay un origen genético en enfermedades poco conocidas o avanzar en el entendimiento de la evolución genómica y el origen de la diversidad en las especies. Así, el análisis detallado de estas variantes continúa siendo una prioridad en la investigación genética y en el desarrollo de tratamientos personalizados en la medicina de precisión.

4. Variaciones del número de copias (CNV)

Las **variaciones del número de copias (CNV)**, por sus siglas en inglés) son alteraciones en la cantidad de copias de segmentos de ADN que pueden variar en tamaño desde unos pocos cientos de bases hasta megabases (véase Ilustración 4). Estas variaciones afectan la cantidad de material genético en un genoma y pueden tener un impacto significativo en la función génica y el fenotipo del organismo.

Las CNV se dividen en dos categorías principales: deleciones y duplicaciones. Las deleciones en el número de copias implican la pérdida de un segmento de ADN en uno o ambos cromosomas homólogos, lo que puede llevar a una reducción en la expresión de los genes contenidos en esa región. Estas deleciones pueden ser responsables de una variedad de trastornos genéticos y enfermedades, como el síndrome de Williams, que resulta de una deleción en el cromosoma 7 que afecta varios genes importantes para el desarrollo (Sanders et al., 2011).

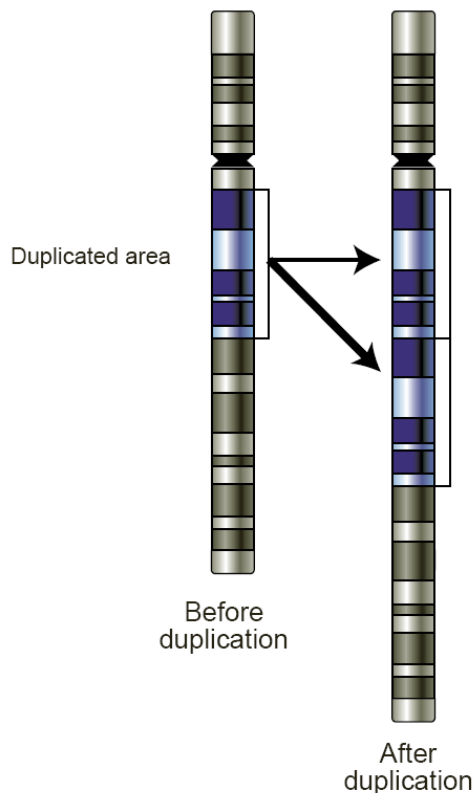


Ilustración 4. Representación de una duplicación en el número de copias.

Las duplicaciones en el número de copias pueden provocar un aumento en la dosis génica, lo que a su vez puede contribuir a la aparición de enfermedades genéticas. Un ejemplo de esto es el síndrome de Down, causado por la trisomía del cromosoma 21. En lugar de los dos cromosomas 21 normales, las personas con síndrome de Down tienen tres copias completas de dicho cromosoma. Esta circunstancia lleva a una sobreexpresión de los genes localizados en dicho cromosoma, lo que resulta en una serie de características físicas y alteraciones cognitivas asociadas con el síndrome de Down (Cooper et al., 2011).

La detección de CNV se realiza mediante varias técnicas, incluyendo el uso de microarrays y métodos basados en secuenciación. Los microarrays, aunque limitados en resolución, han sido herramientas fundamentales en la

identificación de CNV durante años. Sin embargo, las tecnologías de secuenciación de nueva generación (NGS) han mejorado significativamente la capacidad para detectar y caracterizar CNV con una mayor precisión y resolución.

El análisis de variantes genéticas implica identificar estas diferencias y determinar su posible impacto funcional. Esto es crucial para comprender la genética de las enfermedades, desarrollar terapias personalizadas y avanzar en la medicina de precisión. Las técnicas modernas de secuenciación y bioinformática permiten la detección y el análisis de variantes a una escala sin precedentes, facilitando estudios de asociación genética, identificación de biomarcadores y el desarrollo de nuevos tratamientos basados en la información genética específica de un individuo.

2.4. Aplicaciones del análisis de variantes

El análisis de variantes genéticas desempeña un papel crucial en múltiples aspectos de la biomedicina y la investigación genética contemporánea. Una de sus aplicaciones más significativas se encuentra en el diagnóstico y manejo de enfermedades raras, que afectan a una pequeña fracción de la población, pero cuya base genética puede ser compleja y variada. Identificar variantes genéticas específicas asociadas con estas enfermedades es fundamental para

precisar diagnósticos y desarrollar estrategias de tratamiento personalizadas, mejorando así la calidad de vida de los pacientes.

Además, la medicina personalizada utiliza información genética para adaptar tratamientos médicos a las características individuales de los pacientes. Este enfoque, conocido como medicina de precisión, permite prever respuestas a medicamentos, identificar riesgos genéticos y diseñar planes de tratamiento optimizados. Esto no solo mejora la efectividad de los tratamientos, sino que también reduce los efectos secundarios y optimiza los recursos farmacológicos disponibles.

En el ámbito de la investigación en genética humana, el análisis de variantes genéticas es esencial para profundizar en la comprensión de la biología humana. Estudiar cómo diferentes variantes afectan la función génica y contribuyen a la diversidad fenotípica permite descubrir nuevos genes y vías biológicas implicadas en la salud y la enfermedad. Estos conocimientos son fundamentales para el desarrollo de nuevas terapias y medicamentos innovadores.

En oncología, el análisis de variantes genéticas juega un papel crucial en la identificación de mutaciones somáticas que impulsan el crecimiento del cáncer. Este conocimiento permite desarrollar terapias dirigidas específicamente contra las mutaciones presentes en las células cancerosas, mejorando la precisión y la eficacia del tratamiento mientras se minimizan los daños a las células normales.

Finalmente, el análisis de variantes genéticas también es vital en estudios de genética de poblaciones y evolución. Comparar las variantes genéticas entre diferentes poblaciones y especies proporciona información sobre la historia evolutiva, patrones de migración y adaptaciones genómicas a diferentes entornos y presiones selectivas.

En conjunto, el análisis de variantes genéticas continúa siendo una herramienta poderosa que impulsa avances significativos en la medicina personalizada, la investigación biomédica y nuestra comprensión de la genética humana y la evolución. Los avances tecnológicos en secuenciación y bioinformática seguirán potenciando estas aplicaciones, facilitando descubrimientos que transformarán el diagnóstico y tratamiento de enfermedades en el futuro próximo.

3. Antecedentes / Estado del Arte

La identificación de variantes genéticas ha sido un campo de investigación en rápida evolución durante las últimas décadas. Con el desarrollo de la secuenciación de nueva generación (NGS), el campo ha experimentado un avance significativo, permitiendo un análisis más detallado y a gran escala de los genomas. Las herramientas y metodologías para la detección de variantes, especialmente las que permiten un análisis bioinformático, han progresado considerablemente, ofreciendo una mayor precisión y eficiencia durante el proceso de identificación.

En este contexto, cabe destacar HaplotypeCaller, como parte del conjunto de herramientas GATK (Genome Analysis Toolkit). Este programa se ha convertido en una de las herramientas estándar para la búsqueda de variantes debido a su alta precisión en la detección de SNP e INDEL. Utiliza un enfoque basado en haplotipos para identificar variantes, lo que mejora la precisión y reduce las tasas de falsos positivos (Poplin et al., 2017).

FreeBayes es otro programa popular para la detección de variantes, especialmente SNP e INDEL. A diferencia de HaplotypeCaller, FreeBayes utiliza un enfoque bayesiano para identificar variantes en una población de muestras. Esta herramienta es particularmente útil en estudios donde se requiere la identificación de variantes a nivel poblacional (Garrison & Marth, 2012).

RepeatSeq se centra en la identificación de repeticiones en tándem cortas (STR), un tipo específico de variante genética. Las STR pueden tener un impacto significativo en el fenotipo y están asociadas con diversas enfermedades genéticas. RepeatSeq utiliza un enfoque basado en la secuenciación para identificar y cuantificar estas repeticiones, proporcionando información detallada sobre su variabilidad (Highnam et al., 2012).

Scalpel es una herramienta diseñada específicamente para la detección de INDEL en regiones genómicas complejas. Utiliza un enfoque de ensamblaje local para identificar variantes en regiones repetitivas y de baja complejidad, superando algunas de las limitaciones de los métodos tradicionales de búsqueda de variantes (Fang et al., 2015).

Lumpy es una herramienta dedicada a la identificación de variaciones estructurales (SV), como deleciones, duplicaciones, inversiones y translocaciones. Utiliza un enfoque integrador que combina diferentes señales (lecturas discordantes, pares de fragmentos y lecturas divididas) para detectar variantes estructurales con alta precisión. Lumpy ha sido ampliamente adoptado debido a su capacidad para manejar grandes volúmenes de datos de NGS y su eficacia en la detección de SV (Layer et al., 2012).

Manta es una herramienta de bioinformática utilizada para la detección de variantes estructurales (SV) y variantes de número de copias (CNV) en datos de secuenciación de nueva generación. A diferencia de otros métodos, Manta emplea un enfoque de búsqueda a nivel de genoma completo, combinando señales de lecturas discordantes, lecturas de pares de fragmentos y lecturas alineadas parcialmente para identificar variantes con alta precisión. Esta herramienta destaca por su capacidad de trabajar tanto con datos de secuenciación de genoma completo como de exoma y su compatibilidad con muestras de individuos y tumores, incluyendo datos de muestras pareadas sanos/tumorales. La flexibilidad y precisión de Manta en la identificación de variantes estructurales la han convertido en una herramienta popular en estudios genómicos y de cáncer (Chen et al., 2016).

CNVNator es una herramienta especializada en la detección de variaciones en el número de copias (CNV). Utiliza un enfoque basado en la profundidad de lectura para identificar regiones del genoma que han sido duplicadas o eliminadas. CNVNator es conocido por su alta sensibilidad y precisión en la identificación de CNV, lo que lo hace ideal para estudios genómicos a gran escala (Abyzov et al., 2011).

ERD (Extended Read Depths) es otra herramienta para la detección de CNV que se basa en la profundidad de lectura extendida, es decir, un análisis más detallado de la cobertura de secuenciación a través del genoma, para mejorar la precisión en la identificación de estas variantes. ERD es particularmente útil en el análisis de genomas complejos donde las CNV pueden tener un impacto significativo en el fenotipo.

PennCNV es una de las herramientas más utilizadas para la detección de CNV. Combina información de intensidad de señal y datos de genotipos para identificar CNV con alta precisión. PennCNV ha sido ampliamente validado y utilizado en numerosos estudios de asociación genómica, lo que respalda su eficacia y fiabilidad (Wang et al., 2007).

A lo largo de los últimos diez años, las herramientas para la detección de variantes genómicas han evolucionado significativamente. Cada una de las herramientas mencionadas ha contribuido de manera importante al campo, ofreciendo distintas ventajas según el tipo de variante y el contexto del estudio. Por ejemplo, los microarrays, aunque limitados en resolución, fueron pioneros en la identificación de variantes genéticas a gran escala. Sin embargo, con la llegada de la tecnología NGS, herramientas como HaplotypeCaller y FreeBayes han demostrado una mayor precisión y capacidad para manejar grandes volúmenes de datos (Cutler et al., 2012).

La identificación de INDEL y variantes estructurales ha sido mejorada con herramientas como Manta y Lumpy, que utilizan enfoques innovadores para superar las limitaciones de las metodologías tradicionales. Por otro lado, la detección de CNV ha avanzado considerablemente

con el uso de herramientas como CNVNator y PennCNV, que ofrecen alta precisión y han sido validadas en múltiples estudios (Duan et al., 2013; Yao et al., 2017).

A lo largo de los años, se han desarrollado diversas herramientas para la detección de variantes genómicas, cada una con enfoques y características particulares, como HaplotypeCaller, Freebayes, Lumpy, Manta, Scalpel, CNVNator, y ERDs, entre otras. Estas herramientas abordan distintos tipos de variantes, desde pequeñas indels hasta grandes reordenamientos estructurales y variaciones en el número de copias (CNVs). Sin embargo, a pesar de su relevancia y avances, no existe una única herramienta que sea capaz de detectar con precisión todos los tipos de variantes genómicas en una muestra. Este vacío metodológico representa una limitación importante en los estudios genómicos, lo que subraya la necesidad urgente de contar con herramientas más completas y versátiles que permitan obtener resultados más integrales y precisos en la investigación genética.

4. Objetivos

El objetivo principal de este proyecto es identificar y evaluar las herramientas y metodologías más efectivas para la detección de diferentes tipos de variantes genómicas. Este análisis comparativo se enfocará en las siguientes categorías de variantes: SNP, INDEL, variaciones estructurales y CNV. Los objetivos específicos son los siguientes:

1. Identificar herramientas de búsqueda de variantes optimizadas para cada tipo de variante:

- Seleccionar y revisar la documentación y bibliografía que describa herramientas de bioinformática diseñadas y validadas en la detección de variantes genómicas en los últimos años, incluyendo Microarray, HaplotypeCaller, FreeBayes, RepeatSeq, Scalpel, Lumpy, Manta, CNVNator, ERD, y PennCNV.
- Determinar cuáles de estas herramientas son las más adecuadas para detectar los tipos de variantes de interés.

2. Evaluar el rendimiento de las herramientas seleccionadas:

- Analizar la precisión y eficiencia de cada herramienta para el tipo de variante para el que está más indicada, según su documentación de uso y la bibliografía publicada.
- Realizar comparaciones basadas en criterios como sensibilidad, especificidad, tasa de falsos positivos y negativos, y capacidad de manejo de datos de gran volumen.

3. Realizar pruebas empíricas y análisis comparativos:

- Implementar estas herramientas en el análisis de variantes de distintas poblaciones humanas.
- Comparar los resultados obtenidos con las herramientas seleccionadas para cada tipo de variante, destacando las fortalezas y limitaciones de cada una.

4. Establecer un protocolo de análisis de variantes en el que se combinen distintas herramientas para mejorar la detección.

Estos objetivos permitirán una evaluación exhaustiva de herramientas existentes para la búsqueda de variantes, seleccionando la más adecuada para cada categoría de modo que, finalmente, se proponga una metodología de trabajo que las combine y así hallar de la forma más eficiente los distintos tipos de variantes de interés, desde SNP a CNV, cuando se analicen datos de secuenciación de genomas completos. La aplicación de este trabajo en el futuro podría ser de gran utilidad tanto para la búsqueda de posibles variantes desconocidas que den origen a enfermedades raras o mutaciones presentes en genomas tumorales.

5. Metodología

La metodología empleada en este proyecto ha sido diseñada para identificar y evaluar las herramientas más efectivas para la detección de distintos tipos de variantes genómicas. A lo largo del proceso, se ha seguido un enfoque sistemático que abarca desde la selección inicial de herramientas a través de la búsqueda bibliográfica y repositorios especializados, hasta la evaluación de su rendimiento basado en criterios definidos. A continuación, se describe en detalle cada etapa del proceso.

5.1. Selección de herramientas

La primera fase del proyecto consistió en la revisión y selección de herramientas de identificación de variantes que han sido ampliamente utilizadas y validadas (Fang et al., 2017). Se consideraron diversas herramientas, incluyendo Microarray, HaplotypeCaller, FreeBayes, RepeatSeq, Scalpel, Lumpy, Manta, CNVNator, ERDs y PennCNV. Se evaluaron sus capacidades y limitaciones con base en literatura científica y estudios previos, centrándose en su eficacia para detectar SNP, INDEL, variaciones estructurales y CNV (véase Tabla 1).

Tabla 1. Comparativa de las distintas herramientas de búsqueda de variantes según su eficacia a la hora de detectar SNP, INDEL, variaciones estructurales y CNV. El color verde simboliza que la herramienta reconoce ese tipo de variantes correctamente, el amarillo que es capaz de reconocerlas en algunos casos y el rojo que no está diseñada para detectar esas variantes. El criterio de selección de colores para cada herramienta se hizo en base a la lectura bibliográfica de artículos recogidos tanto en la sección Antecedentes / Estado del Arte como en la evaluación de cada herramienta que se describe más abajo.

	SNP	INDEL	Variaciones estructurales	CNV
Microarray				
HaplotypeCaller				
Freebayes				
RepeatSeq				
Scalpel				
Lumpy				
Manta				
CNVnator				
ERDs				
PennCNV				

Se seleccionaron Freebayes para la detección de SNP e INDEL, Manta para la identificación de variantes estructurales y CNVnator para el reconocimiento de CNV. El método de selección en base a criterios de flexibilidad, precisión, eficacia y tasa de falsos positivos/negativos, entre otros, se explica en las secciones siguientes.

5.2. Evaluación de FreeBayes para SNP e INDEL

La elección de FreeBayes como herramienta principal para la detección de SNP e INDEL se fundamenta en varias características distintivas que la destacan dentro del campo del análisis genómico. FreeBayes utiliza un enfoque bayesiano, lo que le permite calcular la probabilidad de variantes basándose en la evidencia observada en los datos de secuenciación. Este método no solo mejora la precisión en la identificación de variantes genéticas, sino que también optimiza la capacidad para discernir variantes en regiones genómicas complejas con cobertura variable (Rimmer et al., 2014)

Una ventaja significativa es su alta flexibilidad y capacidad de personalización. Es posible ajustar numerosos parámetros según los requisitos específicos del estudio, incluyendo la profundidad mínima de cobertura, el umbral de calidad de las lecturas y las probabilidades previas de variantes. Esta flexibilidad permite adaptar FreeBayes para obtener resultados óptimos en diferentes tipos de datos de secuenciación y contextos de investigación genética.

En términos de precisión, FreeBayes ha demostrado consistentemente una baja tasa de falsos positivos en comparación con otras herramientas similares como HaplotypeCaller y RepeatSeq (Liu et al., 2019). Este aspecto es crucial especialmente en estudios que manejan muestras con alta cobertura y complejidad genómica, donde la exactitud en la identificación de SNP e INDEL es fundamental para la interpretación precisa de los resultados genéticos.

Otra fortaleza radica en su capacidad para manejar eficientemente grandes volúmenes de datos de secuenciación masiva. Esto es altamente relevante en investigaciones genómicas que implican cohortes extensas de pacientes o el análisis de genomas completos, donde la eficiencia en el procesamiento de datos contribuye significativamente a la viabilidad y el éxito del estudio (Poplin et al., 2018)

Finalmente, la validación extensiva de FreeBayes en múltiples estudios genómicos (Hwang et al., 2015; Laurie et al., 2016) refuerza su reputación como una herramienta fiable y precisa para la detección de variantes genéticas. Su aplicación en investigaciones genéticas ha consolidado su posición como una opción preferida para investigadores que buscan herramientas robustas y validadas para análisis genómicos avanzados.

5.3. Evaluación de Manta para variaciones estructurales (SV)

Manta se ha consolidado como una de las herramientas más efectivas para la detección de variaciones estructurales (SV) en datos de secuenciación de nueva generación (NGS) debido a su versatilidad, precisión y eficiencia computacional. Su capacidad para identificar una amplia gama de SV proporciona a los investigadores una visión integral de las alteraciones estructurales en el genoma. Esto es especialmente valioso para estudios que buscan asociar variabilidad genética con enfermedades o características fenotípicas específicas, posicionándola por encima de herramientas como LUMPY y BreakDancer, que pueden tener limitaciones en la detección de ciertos tipos de SV (Chen et al., 2016).

Una de las fortalezas clave de Manta es su facilidad de integración con otros protocolos de análisis genómicos. Su compatibilidad con herramientas como GATK y Strelka permite su incorporación en flujos de trabajo más amplios, mejorando la coherencia y eficiencia en la detección de variantes genéticas. Esta integración simplificada destaca a Manta frente a herramientas como DELLY y SvABA que, a menudo, requieren configuraciones más complejas y no siempre se integran tan fácilmente (Wala et al., 2018).

Manta también sobresale en precisión y sensibilidad sobre herramientas similares. Utiliza un enfoque basado en gráficas de evidencias, que combina información de pares de lectura y profundidad de cobertura para identificar variantes estructurales con alta fidelidad. Esto permite

a Manta detectar SV incluso en regiones genómicas complejas y repetitivas, reduciendo la tasa de falsos positivos en comparación con herramientas como Pindel, que se centran en la detección de puntos de ruptura (Ye et al., 2009).

La eficiencia computacional de Manta es otra de sus ventajas significativas, especialmente en estudios de gran escala que manejan grandes volúmenes de datos. Su diseño permite realizar análisis complejos sin un uso excesivo de recursos, lo que la diferencia de herramientas como CNVnator y TIDDIT, que pueden requerir más tiempo de procesamiento y recursos computacionales (Eisfeldt et al., 2017).

Por último, Manta es adaptable a diversos tipos de datos de secuenciación, incluyendo exoma y genoma completo, lo que la hace adecuada para una variedad de estudios genómicos. Su flexibilidad, combinada con el respaldo continuo de Illumina y una activa comunidad de usuarios, asegura que Manta se mantenga actualizada y capaz de responder a las necesidades emergentes de la investigación genómica. Estas características hacen de Manta una herramienta preferente en la detección de variaciones estructurales en estudios genómicos avanzados.

5.4. Evaluación de CNVnator para variaciones en el número de copias (CNV)

CNVnator es una herramienta ampliamente reconocida para la detección de variaciones en el número de copias (CNV) en estudios de secuenciación del genoma completo, destacándose por su precisión, velocidad y robustez. Utiliza un enfoque basado en la profundidad de lectura, que permite identificar CNV a través de la variación en la cobertura de secuenciación a lo largo del genoma (Abyzov et al., 2011). Este método es efectivo para detectar tanto CNV grandes como pequeñas, ajustando la resolución de detección mediante ventanas de longitud variable, lo que le confiere una alta precisión en comparación con otras herramientas como Control-FREEC (Boeva et al., 2011).

La capacidad de CNVnator para procesar grandes volúmenes de datos de manera rápida y eficiente es crucial para estudios de gran escala que involucran miles de muestras o datos de alta cobertura. Esto lo diferencia de herramientas como PennCNV, que suelen ser más lentas y requieren una pre-procesación más extensa (Wang et al., 2007). CNVnator ofrece una solución directa y eficiente, facilitando su aplicación en estudios de secuenciación del genoma completo.

Además de su velocidad, CNVnator ha demostrado una alta precisión en la detección de CNV, mostrando concordancia con métodos de microarrays y otros enfoques experimentales. A diferencia de herramientas como ExomeDepth, que están más limitadas a datos de exoma, CNVnator proporciona una detección precisa en todo el genoma, incluso en regiones con alta repetitividad o sesgos en la cobertura (Plagnol et al., 2012).

La facilidad de uso de CNVnator también contribuye a su popularidad. Aunque requiere conocimientos básicos de manejo de línea de comandos, su interfaz es accesible y permite obtener resultados fiables con un mínimo de ajustes. Esto es ventajoso en comparación con herramientas como ERDS, que requieren una mayor personalización para obtener resultados óptimos.

Finalmente, CNVnator proporciona una salida detallada que incluye información sobre la ubicación, tamaño y ganancia o pérdida de copias de las CNV detectadas. Esta salida informativa permite un análisis más profundo de las regiones afectadas, siendo comparable, y en algunos casos superior, a la de otras herramientas como GATK CNVCaller, que puede requerir pasos adicionales para una interpretación completa (Broad Institute, 2020).

En conclusión, CNVnator se destaca como una herramienta ideal para la detección de CNV en estudios de secuenciación del genoma completo, ofreciendo un balance óptimo entre precisión, velocidad y facilidad de uso. Su capacidad para manejar grandes volúmenes de datos y su enfoque robusto la posicionan como una elección preferente en el análisis de variaciones en el número de copias en el campo de la genómica.

5.5. Proyecto 1000 Genomas

Con el fin de valorar mediante un análisis real el desempeño de las herramientas candidatas seleccionadas, se buscaron datos de secuenciación de genoma completo en bases de datos especializadas. Este es el caso del repositorio perteneciente al Proyecto 1000 Genomas (*1000 Genomes Project*), lanzado en 2008. Esta es una iniciativa internacional cuyo objetivo es crear el catálogo más completo y detallado de variación genética humana. Este proyecto ha secuenciado los genomas de más de 2,500 individuos de 26 poblaciones diferentes alrededor del mundo, proporcionando una base de datos de gran valor para estudios de asociación genética y análisis de variaciones genómicas (Fairley et al., 2020).

Una de las principales ventajas del Proyecto 1000 Genomas es su enfoque en la diversidad genética global. Al incluir muestras de diversas poblaciones, el proyecto permite un análisis más inclusivo y representativo de la variabilidad genética humana. Este enfoque es crucial para comprender las diferencias genéticas que pueden influir en la predisposición a enfermedades, la respuesta a medicamentos y otros rasgos fenotípicos.

En el contexto de este proyecto de fin de grado, se han utilizado ficheros bam de alineamiento del genoma completo correspondiente a sujetos de la población Punjabi de Lahore, Pakistán (PJL), como base para realizar el análisis de variantes genómicas (<http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/data/HG01583/alignment/>). El genoma de

referencia (hs37d5) empleado para el alineamiento se diferencia del genoma humano GRCh37 estándar por incluir secuencias de decoy y del virus Epstein-Barr. Estas adiciones mejoran la precisión del alineamiento de lecturas de secuenciación, al reducir la alineación errónea en regiones ambiguas o repetitivas del genoma. Los ficheros de alineamiento proporcionan una representación detallada de las secuencias de ADN alineadas, lo que facilita la identificación y caracterización de variantes genómicas como SNP, INDEL, variaciones estructurales y CNV.

El uso de datos del Proyecto 1000 Genomas asegura que el análisis se realiza sobre una base de datos de alta calidad y reconocida internacionalmente. La población PJL, al igual que otras incluidas en el proyecto, ofrece una visión única de la diversidad genética, lo que permite una comparación más precisa y relevante de las herramientas de detección de variantes.

En este proyecto, la utilización de datos de la población PJL ha sido fundamental para realizar un análisis exhaustivo y comparativo de las metodologías y herramientas para la detección de variantes en el genoma completo.

6. Implementación

La primera etapa de la implementación consistió en la descarga de los ficheros BAM de alineamiento correspondientes a la población Punjabi in Lahore, Pakistán (PJL). Estos archivos fueron obtenidos desde el repositorio público del proyecto 1000 Genomes, como se ha comentado anteriormente.

Debido a que el análisis de variantes requiere de una comparación precisa con el genoma de referencia, también fue necesario descargar el genoma de referencia utilizado en el alineamiento de estos ficheros, la versión **hs37d5** publicada específicamente desde por el proyecto 1000 Genomes (Fase II).

La versión **hs37d5** del genoma humano es una adaptación del ensamblaje de referencia GRCh37, que incluye 35.4 Mb de secuencias adicionales para mejorar la precisión en el análisis genómico. Algunas de las modificaciones específicas que incorpora son:

1. **Secuencias de *Decoy* y EBV:** Se añaden secuencias *decoy* para mitigar la alineación incorrecta en regiones ambiguas o altamente repetitivas, así como secuencias del virus Epstein-Barr (EBV), lo cual es crucial para identificar regiones que podrían ser confundidas con secuencias humanas.
2. **Regiones Alternativas y Haplotipos:** La inclusión de regiones alternativas y haplotipos adicionales permite capturar variabilidad genética que no está presente en el ensamblaje GRCh37 estándar. Esto es particularmente relevante para estudios que buscan una representación más completa de la diversidad genómica en diferentes poblaciones.
3. **Correcciones y Ajustes en Secuencias:** Incorpora correcciones y actualizaciones basadas en datos de secuenciación más recientes, mejorando la calidad del alineamiento y la identificación de variantes. Estos ajustes son importantes para la identificación precisa de variaciones genéticas, especialmente en estudios de alineamiento del genoma completo, como en el caso de la población punjabi de Lahore, Pakistán.

Estas características hacen de **hs37d5** una referencia más adecuada para estudios que requieren alta precisión en el alineamiento y en la detección de variantes genómicas, proporcionando una representación más completa y exacta del genoma humano.

El siguiente paso en el proyecto fue instalar todas las herramientas necesarias para la búsqueda de variantes. Para facilitar el proceso y asegurar que fuera lo más eficiente posible, se creó una

guía detallada de instalación y uso para cada una de las herramientas empleadas, que incluye instrucciones de instalación, configuraciones necesarias y ejemplos de comandos para realizar la búsqueda de variantes. Este enfoque estructurado no solo facilitó la implementación de las herramientas, sino que también garantizó la reproducibilidad del análisis.

6.1. Instalación de Conda usando Miniconda en Linux

En este proyecto, los análisis se han llevado a cabo en el servidor del grupo de investigación CoMBA, en un sistema operativo Ubuntu (Ubuntu 22.04.4 LTS). Además, se ha optado por utilizar un entorno Conda para la instalación y gestión de herramientas de búsqueda de variantes. La elección de Conda se fundamenta en las siguientes razones:

1. **Facilidad de gestión de dependencias:** Las herramientas de bioinformática, como Freebayes, Manta y CNVnator, a menudo requieren múltiples bibliotecas y dependencias específicas de sistema. Conda permite la instalación y gestión de estas dependencias de manera automática, reduciendo significativamente los problemas de incompatibilidad y asegurando que todas las versiones de las bibliotecas requeridas estén alineadas con las necesidades de cada herramienta.
2. **Creación de entornos aislados:** Conda facilita la creación de entornos de trabajo aislados. Esto significa que cada herramienta puede ejecutarse en un entorno que contiene exclusivamente las versiones del programa y bibliotecas necesarias para su funcionamiento. Al utilizar entornos aislados, se evita la contaminación del sistema operativo anfitrión y la interferencia entre diferentes herramientas y sus dependencias, garantizando que las configuraciones de una herramienta no afecten a otras.
3. **Portabilidad y reproducibilidad:** Un aspecto crucial en la bioinformática es la reproducibilidad de los resultados. Conda permite crear entornos replicables que pueden ser compartidos entre diferentes usuarios y máquinas, asegurando que las herramientas y sus dependencias se mantengan idénticas en cualquier entorno de ejecución. Esto facilita la replicación de los análisis y refuerza la fiabilidad de los resultados obtenidos en el proyecto.
4. **Actualizaciones y mantenimiento simplificados:** Conda proporciona una manera sencilla de actualizar las herramientas instaladas y sus dependencias. Mediante comandos simples, es posible mantener las herramientas al día con las últimas versiones, mejorando la seguridad y las funcionalidades disponibles, sin riesgo de romper las configuraciones de los entornos.

5. **Soporte multiplataforma:** Conda es compatible con múltiples sistemas operativos, incluyendo Linux, macOS y Windows. Esta característica asegura que las herramientas de llamado de variantes puedan ser instaladas y utilizadas de manera uniforme en diferentes plataformas, lo que es particularmente útil en entornos de investigación donde se utilizan diferentes tipos de sistemas.
6. **Facilidad de instalación:** Instalar herramientas de bioinformática directamente desde su código fuente puede ser complicado y propenso a errores, especialmente para usuarios sin experiencia avanzada en administración de sistemas. Conda simplifica este proceso, proporcionando versiones precompiladas de las herramientas que pueden ser instaladas con un solo comando, reduciendo la complejidad y los posibles errores de instalación.

En resumen, la utilización de un entorno Conda para la instalación y gestión de herramientas de búsqueda de variantes proporciona un marco robusto y flexible que mejora la gestión de dependencias, asegura la reproducibilidad de los resultados y simplifica el proceso de instalación y mantenimiento del programa. En el **Anexos**

Anexo I se encuentra una **guía detallada de instalación de Conda**.

6.2. Freebayes con Conda

Instalar Freebayes en un entorno Conda proporciona una gestión eficaz de las dependencias y evita conflictos con otras herramientas instaladas en el sistema, como ya se ha expuesto.

Posteriormente, se llevó a cabo la ejecución de Freebayes en el servidor. Un ejemplo básico de cómo ejecutar FreeBayes para hacer búsqueda de variantes en un archivo BAM es el siguiente:

```
freebayes \  
-f /ruta/al/genoma_referencia.fasta \  
/ruta/al/archivo_entrada.bam \  
> resultados.vcf
```

Explicación de los parámetros:

- **-f:** Especifica el archivo FASTA del genoma de referencia.
- **/ruta/al/archivo_entrada.bam:** Especifica el archivo BAM de entrada con las lecturas alineadas.
- **resultados.vcf:** Especifica el archivo de salida donde se guardarán las variantes llamadas en formato VCF.

En el **Anexo II** se encuentra la **guía de instalación y ejecución de Freebayes en un entorno Conda.**

6.3. Manta con Conda

El uso de Conda para la instalación de Manta proporciona las mismas ventajas mencionadas anteriormente. Un ejemplo para hacer un llamado de variantes estructurales (SV) de una o varias muestras hay que ejecutar el siguiente comando:

```
../configManta.py \  
--bam ../muestra1.bam \  
--bam ../muestra2.bam (opcional) \  
--bam ../muestra3.bam (opcional) \  
--referenceFasta ../referencia.fa \  
--runDir directorio_analisis_manta
```

El comando es una instrucción para configurar Manta. A continuación, se explican los parámetros usados en el comando:

- **../configManta.py:** Este es el archivo de configuración de Manta. Es el archivo ejecutable que inicializa la configuración para un análisis particular, basándose en los parámetros y opciones proporcionados. Este archivo se ejecuta una sola vez para preparar el entorno y los archivos necesarios para correr Manta.
- **--bam ../muestra1.bam:** Este parámetro especifica la ruta al archivo BAM de una muestra de secuenciación. En este caso, muestra1.bam es el archivo BAM de la primera muestra que se analizará.
- **--bam ../muestra2.bam y --bam ../muestra3.bam (opcionales):** Estos son parámetros adicionales que indican archivos BAM adicionales para incluir en el análisis. Son opcionales y permiten realizar un análisis conjunto de múltiples muestras si se proporciona más de un archivo BAM.
- **--referenceFasta ../referencia.fa:** Este parámetro especifica la ruta al archivo FASTA del genoma de referencia que se utilizará en el análisis. El archivo FASTA contiene las secuencias de ADN del genoma de referencia contra las cuales se alinearon las lecturas en los archivos BAM. Manta utiliza este archivo para mapear y comparar las secuencias alineadas para identificar variantes estructurales.
- **--runDir directorio_analisis_manta:** Este parámetro define el directorio donde se almacenarán los archivos de salida y los archivos intermedios generados por Manta durante su ejecución. Es el directorio de trabajo donde Manta crea su estructura de carpetas y organiza los resultados del análisis. Usar un directorio específico ayuda a mantener los archivos del proyecto organizados y facilita el acceso y la revisión de los resultados.

Una vez configurado, simplemente hay que ejecutar el archivo de trabajo:

```
| ./runWorkflow.py
```

En el **Anexo III** se encuentra la **guía de instalación y ejecución de Manta en un entorno Conda.**

En el repositorio de Github hay información más detallada sobre este proceso (Illumina. (2016).

Manta: Structural variant and indel caller for mapped sequencing data. Repositorio GitHub.

<https://github.com/Illumina/manta>)

6.4. CNVnator con Conda

El uso de Conda para la instalación de CNVnator supuso un paso esencial para poder ejecutarlo correctamente ya que CNVnator requiere de varios permisos de nivel administrador, un soporte extra para el análisis de datos desarrollado por CERN, así como una versión específica de SAMtools y otros paquetes. El proceso está detalladamente en la **Guía de instalación y ejecución de CNVnator en un entorno Conda** del **Anexo IV**. Una vez conseguida la instalación, la búsqueda de variantes con CNVnator se realiza de la siguiente forma:

```
cnvnator -root ejemplo.root -call tamaño_bin > llamado_variantes.calls
```

-root ejemplo.root:

- La opción -root especifica el archivo .root que contiene los datos ya procesados por CNVnator.
- ejemplo.root es el nombre del archivo raíz generado en pasos anteriores del análisis. Este archivo contiene información estructurada que CNVnator utiliza para detectar variantes en el número de copias.

-call tamaño_bin:

- La opción -call indica a CNVnator que debe realizar la búsqueda de variantes basándose en el tamaño de bin especificado.
- tamaño_bin se refiere a una medida de la resolución del análisis. Es un valor numérico que determina el tamaño de las ventanas deslizantes que CNVnator utilizará para escanear el genoma y detectar CNV. Un tamaño de bin comúnmente utilizado es de 100 a 1000 bases, pero puede variar dependiendo del tamaño de los fragmentos en el archivo de datos y la cobertura deseada.

llamado_variantes.calls:

- Este es el nombre del archivo de salida en el que se almacenarán las variantes, en este caso los CNV, detectados).
- CNVnator guardará en este archivo los resultados de la búsqueda de variantes, es decir, las regiones del genoma donde se ha identificado una variación en el número de copias.

Para analizarlo posteriormente, este fichero .calls se puede convertir a un formato típico de análisis de variantes como es .vcf usando el conversor proporcionado por CNVnator.

6.5. Filtrado de variantes

Una vez realizada la búsqueda de variantes con el programa bioinformático correspondiente, el filtrado posterior de los resultados contenido en archivos VCF (del inglés, *Variant Call Format*) es una etapa crucial. Mediante el proceso de filtrado se pueden identificar y seleccionar las variantes genómicas de mayor calidad y relevancia biológica, entre otras ventajas como se señala a continuación:

- **Reducción de falsos positivos:** Uno de los principales objetivos del filtrado de variantes es minimizar la presencia de falsos positivos, es decir, aquellas variantes que son incorrectamente identificadas como reales debido a errores de secuenciación, problemas de alineamiento o artefactos del proceso de búsqueda. Filtrar las variantes basándose en métricas de calidad, como la profundidad de cobertura, la calidad de la búsqueda (QUAL) y las tasas de error de mapeo, ayuda a asegurar que solo las variantes más fiables sean retenidas para análisis posteriores.
- **Mejora de la interpretación biológica:** Al filtrar las variantes, se eliminan aquellas que son menos fiables o tienen poca relevancia, lo que facilita la interpretación de los resultados.
- **Optimización de los recursos computacionales:** Trabajar con archivos VCF que contengan un gran número de variantes no filtradas puede ser computacionalmente intensivo y costoso. Al aplicar filtros, se reduce la cantidad de datos a procesar, lo que facilita el manejo de los datos y acelera los análisis subsecuentes.

Existen varias herramientas para llevar a cabo el filtrado de variantes. En primer lugar, cabe destacar **VCFTools**, una herramienta ampliamente utilizada para la manipulación y análisis de archivos VCF. Proporciona una serie de funciones que permiten realizar filtros basados en la calidad de las variantes, la profundidad de cobertura, el número de muestras genotipadas, etc. VCFTools es fácil de usar y se integra bien en flujos de trabajo bioinformáticos, haciendo que sea una elección popular para filtrar variantes en proyectos de investigación.

Similar a VCFTools, **BCFTools** ofrece funcionalidades robustas para el manejo de archivos VCF o BCF (del inglés, *Binary VCF*). Además de proporcionar opciones de filtrado, BCFTools también permite convertir entre diferentes formatos de archivos de búsqueda de variantes e indexar archivos VCF para un acceso más rápido. La capacidad de manejar archivos comprimidos en formato BCF hace que BCFTools sea una herramienta eficiente para proyectos de gran escala, ya que reduce el espacio en disco y mejora la velocidad de procesamiento.

Para complementar el proceso de filtrado, en este trabajo además se ha utilizado **RTGtools** (del inglés, *Real-Time Genomics tools*) para generar estadísticas detalladas sobre los archivos VCF

antes y después del filtrado. RTGtools permite evaluar la calidad de las variantes mediante métricas como el número de variantes que pasan o no pasan los filtros establecidos, la proporción de variantes heterocigotas frente a homocigotas, y otros parámetros relevantes. Estas estadísticas proporcionan una visión general de la calidad y cantidad de las variantes y ayudan a validar la efectividad del filtrado aplicado.

La guía de instalación de cada una de estas tres herramientas viene recogida en el **Anexo V**.

6.5.1. Filtrado de variantes – SNP e INDEL

Los comandos de filtrado principales usados para el filtrado de SNP e INDEL con VCFTools han sido los siguientes:

```
# Filtrar SNP de un archivo VCF
vcftools --vcf input.vcf \
    --remove-indels \ # Eliminar INDEL
    --minQ 30 \ # Calidad mínima de mapeo
    --minDP 10 \ # Profundidad mínima de cobertura
    --maxDP 100 \ # Profundidad máxima de cobertura
    --maf 0.05 \ # MAF mínimo (Minor Allele Frequency)
    --max-maf 0.5 \ # MAF máximo
    --minGQ 20 \ # Calidad mínima de genotipo
    --max-alleles 2 \ # Número máximo de alelos
    --recode --recode-INFO-all \ # Reescribir archivo VCF
    --out SNP_filtrados

# Filtrar INDEL de un archivo VCF
vcftools --vcf input.vcf \
    --keep-only-indels \
    --minQ 30 \
    --minDP 10 \
    --maxDP 100 \
    --minGQ 20 \
    --max-alleles 2 \
    --recode --recode-INFO-all \
    --out INDEL_filtrados
```

Comando principal: `vcftools --vcf input.vcf`

`vcftools`: Es la herramienta que se ha utilizado para procesar el archivo VCF.

`--vcf input.vcf`:

- Especifica el archivo VCF de entrada que contiene las variantes genéticas (SNP e INDEL). Este es el archivo que se va a filtrar.

Parámetros de Filtrado

--remove-indels:

- Este parámetro elimina todas las variantes que son INDEL (inserciones y deleciones) del archivo VCF, dejando solo los SNP (polimorfismos de un solo nucleótido).

--minQ 30:

- Filtra las variantes en función de la calidad de mapeo. Solo se mantienen las variantes con una calidad de mapeo mínima de 30.
- Motivo: La calidad de mapeo (QUAL) es una medida de confianza en la búsqueda de la variante. Un valor de 30 es generalmente considerado como un umbral razonable para garantizar que las variantes sean fiables.

--minDP 10:

- Filtra las variantes basándose en la profundidad mínima de cobertura. Solo se retienen las variantes que han sido cubiertas al menos 10 veces por las lecturas de secuenciación.
- Motivo: La profundidad de cobertura (DP) refleja cuántas veces una región ha sido secuenciada. Una baja cobertura podría indicar una variante no fiable, por lo que se establece un umbral mínimo.

--maxDP 100:

- Filtra las variantes basándose en la profundidad máxima de cobertura. Se eliminan las variantes con una cobertura superior a 100.
- Motivo: Coberturas extremadamente altas pueden indicar errores de secuenciación, duplicaciones u otras anomalías. Establecer un límite máximo ayuda a evitar estos problemas.

--maf 0.05:

- Filtra las variantes según la frecuencia del alelo menor (del inglés, *Minor Allele Frequency*). Solo se retienen las variantes con un MAF de al menos 0.05 (5%).
- Motivo: El MAF indica cuán común es el alelo menos frecuente en la población. Variantes raras (MAF muy bajo) podrían ser errores o menos relevantes, dependiendo del estudio.

--max-maf 0.5:

- Filtra las variantes para eliminar aquellas donde el alelo menor tenga una frecuencia mayor al 50%.

- Motivo: En estudios genéticos, a menudo se buscan variantes con un MAF dentro de un rango específico. Variantes con un MAF superior a 0.5 suelen ser variantes mayoritarias y pueden ser menos informativas para ciertos análisis.

`--minGQ 20:`

- Filtra las variantes según la calidad mínima del genotipo (GQ). Solo se retienen los genotipos con un valor mínimo de 20.
- Motivo: El GQ es una medida de la confianza en el genotipo asignado a cada muestra. Un valor de 20 es un umbral común para asegurar que los genotipos sean fiables.

`--max-alleles 2:`

- Filtra variantes para asegurarse de que solo se incluyan aquellas con hasta 2 alelos (uno de referencia y uno alternativo).
- Motivo: SNP típicos solo tienen dos alelos (uno de referencia y uno alternativo). Variantes con más de dos alelos podrían ser errores o representaciones complejas que no son SNP típicos.

Opciones de salida

`--recode --recode-INFO-all:`

- Estas opciones indican que el archivo VCF filtrado debe ser recodificado y que toda la información en el campo INFO debe ser mantenida en la salida.
- Motivo: Queremos conservar toda la información disponible en las variantes filtradas y generar un nuevo archivo VCF con solo las variantes que pasan los filtros aplicados.

`--out filtered_snps:`

- Especifica el nombre de salida para el archivo VCF filtrado.

6.5.2. Filtrado de variantes – variantes estructurales (SV, en inglés)

El filtrado de variantes es un paso crucial en el análisis genómico, especialmente cuando se trata de variantes estructurales (SV, en inglés), que incluyen deleciones, duplicaciones, inserciones, translocaciones y otras alteraciones a gran escala en el genoma. Estas variantes pueden tener un impacto significativo en la función génica y, por lo tanto, es fundamental distinguir entre variantes genuinas y posibles falsos positivos.

En este apartado, se empleará la herramienta bcftools para filtrar las variantes estructurales detectadas por Manta. A través de la aplicación de criterios estrictos de calidad y soporte de lectura, se busca refinar el conjunto de datos, garantizando que solo las variantes estructurales más fiables y relevantes se incluyan en el análisis final. Este enfoque permite optimizar la interpretación de las variantes, reduciendo la probabilidad de errores y mejorando la precisión de los hallazgos genómicos.

Comando de filtrado (archivo diploidSV.vcf.gz)

```
# Filtrar variantes con QUAL > 30, GQ > 20, y que pasen el filtro  
'PASS'  
# Adicionalmente, excluye variantes con problemas como alta  
profundidad o sin soporte  
  
bcftools view -i 'QUAL>30 && FORMAT/GQ>20 && FILTER="PASS" &&  
FILTER!="MinQUAL" && FILTER!="MinGQ"' diploidSV.vcf.gz -o  
resultado_filtrado.vcf
```

El comando completo filtra un archivo VCF (diploidSV.vcf.gz) para incluir únicamente aquellas variantes que:

- Tienen una calidad (QUAL) mayor a 30.
- Tienen una calidad de genotipo (GQ) mayor a 20.
- Han pasado todos los filtros estándar (FILTER="PASS").
- No han sido excluidas por tener una calidad baja (FILTER!="MinQUAL") ni por tener una baja calidad de genotipo (FILTER!="MinGQ").

6.5.3. Filtrado de variantes – variación del número de copias (CNV, en inglés)

El proceso de filtrado de variantes estructurales en estudios de variación del número de copias (CNV) puede verse afectado por la calidad de los datos de alineamiento, particularmente en el contexto de baja cobertura de secuenciación. En estudios donde los archivos BAM presentan baja cobertura, se observan frecuentemente problemas que impactan directamente en los indicadores de calidad de las variantes detectadas.

Uno de los parámetros afectados es *natorQ0*, que representa la fracción de lecturas con una calidad de mapeo igual a cero. Un valor alto de *natorQ0* en regiones con baja cobertura puede indicar que una gran proporción de las lecturas son de baja calidad o no se han alineado de manera confiable al genoma de referencia. Esto sucede porque, con menos datos disponibles, hay una mayor probabilidad de que las lecturas no se alineen adecuadamente o se asignen con baja calidad.

La baja cobertura no solo afecta la calidad del mapeo, sino también la profundidad de lectura normalizada (*natorRD*). Esto puede llevar a una reducción en la capacidad de detectar variantes estructurales con confianza, ya que el número limitado de lecturas puede no proporcionar suficiente evidencia para confirmar la presencia de una variante. Como resultado, se puede enfrentar un incremento en el ruido y los falsos positivos, dado que las lecturas de baja calidad o alineamientos ambiguos pueden ser malinterpretados como variantes estructurales.

Es crucial considerar estas implicaciones al aplicar filtros para la detección de variantes. Un filtrado muy estricto basado en *natorQ0* en condiciones de baja cobertura puede eliminar variantes reales que, debido a la limitación en la cobertura, presentan altos valores de *natorQ0*. Por tanto, al filtrar variantes, es esencial balancear entre la sensibilidad y la especificidad, teniendo en cuenta las limitaciones impuestas por la cobertura de secuenciación.

Con esta información, el comando de filtrado seleccionado es el siguiente:

```
bcftools filter -i 'INFO/natorQ0 <= 0.9 \
&& INFO/natorRD >= 0.01 \
&& INFO/natorP1 <= 1e-5 && INFO/natorP2 <= 1e-5' \
input.vcf.gz -o resultado_filtrado.vcf
```

1. **INFO/natorQ0 <= 0.9**

Este filtro selecciona variantes donde la fracción de lecturas con calidad de mapeo cero (*natorQ0*) sea del 90% o menos. Se usa para evitar variantes en regiones con problemas graves de calidad de lectura. Un valor alto de *natorQ0* indica que una gran proporción de las lecturas tienen baja calidad o no se alinean bien, lo cual puede afectar la confiabilidad de la variante detectada.

2. **INFO/natorRD >= 0.01**

Este filtro incluye solo variantes con una profundidad de lectura normalizada (*natorRD*) de al menos 0.01. Esto asegura que las variantes con una cobertura mínima razonable sean consideradas, evitando la inclusión de variantes en regiones con cobertura extremadamente baja, que podrían no tener suficiente evidencia para confirmar la presencia de la variante.

3. INFO/natorP1 <= 1e-5 && INFO/natorP2 <= 1e-5

Estos filtros aplican umbrales de significación estadística para los valores p calculados por las pruebas estadísticas (natorP1 y natorP2). Solo se conservan las variantes con valores p menores o iguales a 10^{-5} , lo cual asegura que las variantes seleccionadas tengan una alta probabilidad de ser verdaderas y minimiza el riesgo de incluir falsos positivos.

Estos filtros combinados ayudan a mantener la calidad de las variantes estructurales detectadas mientras se reducen los posibles artefactos debidos a baja cobertura y calidad de lectura.

7. Estudio Económico

Para realizar un análisis de búsqueda de variantes utilizando las herramientas seleccionadas (FreeBayes, Manta y CNVnator), en el presupuesto se deben considerar los costeos de los programas, equipo y almacenamiento de datos empleados, además de las instalaciones y el tiempo de trabajo de un bioinformático profesional. A continuación, se detalla un estudio económico de los costos asociados al proyecto realizado para este trabajo:

1. Costos de Personal:

- **Bioinformático profesional:** La tarifa promedio de un bioinformático profesional es de aproximadamente 60 euros por hora. Estimando que el proyecto requiere alrededor de 150 horas de trabajo para análisis de datos, interpretación de resultados y redacción de informes, el costo total en mano de obra sería de:
 - $60 \text{ euros/hora} * 150 \text{ horas} = \mathbf{9000 \text{ euros.}}$

2. Costeos derivados de herramientas informáticas:

- **Licencias de programa:** FreeBayes, Manta y CNVnator son herramientas de código abierto y no requieren costos de licencia. El resto de las herramientas (RTGtools, VCFtools, BCFtools, IGV) pueden ser usadas gratuitamente.

3. Costes derivados del equipo informático:

- **Servidor y almacenamiento de datos:** El análisis de genomas completos requiere una infraestructura de alto rendimiento. El servidor tiene un coste de 9000 euros y un tiempo de vida útil de 5 años. Se estima que el alquiler o mantenimiento de un servidor adecuado, junto con almacenamiento de datos durante la duración del proyecto, asciende a:
 - Servidor y almacenamiento(alquiler/uso de 3 meses): 450 euros.

Total: **450 euros.**

4. Costos de Insumos y Materiales:

- **Consumo de energía:** Estimando un costo de electricidad para mantener los equipos en funcionamiento durante el análisis:
 - Aproximadamente 300 euros.
- Otros insumos (p. ej., suscripciones a bases de datos científicas, costos de impresión y documentación): 200 euros.

5. Costos totales:

Tipo de Coste	Importe
Costes de personal	9,000 euros
Costes derivados de herramientas informáticas	0 euros
Costes derivados del equipo informático	450 euros
Insumos y materiales	500 euros
Costo total estimado del proyecto	9,950 euros

6. Beneficios económicos:

Este análisis puede generar múltiples beneficios económicos para una empresa dedicada a la investigación genética o farmacogenómica. Entre los beneficios potenciales se incluyen:

- **Mejora en la precisión de diagnóstico genético:** Contribuyendo a diagnósticos más precisos, reduciendo costos de pruebas adicionales y acelerando tratamientos personalizados.
- **Optimización de recursos en proyectos futuros:** Un protocolo estandarizado mejora la eficiencia, reduce tiempos de análisis y, por lo tanto, costos operativos.
- **Oportunidades de negocio y colaboración:** Resultados robustos pueden atraer contratos de investigación, colaboraciones y financiaciones, potencialmente generando ingresos adicionales.

En conclusión, aunque el costo inicial del análisis genómico es considerable, la inversión puede verse recuperada a través de mejoras en la eficiencia del diagnóstico, reducción de errores y ampliación de oportunidades de negocio en el sector de la bioinformática.

8. Resultados

8.1. Freebayes

Una vez analizados los datos con Freebayes, la herramienta RTG es útil para crear una tabla resumen de lo que se ha obtenido (Tabla 2).

```
# generar estadísticas
rtg vcfstats resultados.vcf > rtg_estadisticas.out
```

Tabla 2. Resultados RTG vcfstats para el fichero VCF sin filtrar generado por Freebayes

Location	../resultados.vcf
Failed Filters	0
Passed Filters	4203532
SNPs	3500772
MNPs	84966
Insertions	147622
Deletions	188412
Indels	26039
SNP Transitions/Transversions	2.03 (3350500/1649019)
Total Het/Hom ratio	1.28 (2213959/1733852)
SNP Het/Hom ratio	1.34 (2002925/1497847)
MNP Het/Hom ratio	1.83 (54926/30040)
Insertion Het/Hom ratio	0.69 (60325/87297)
Deletion Het/Hom ratio	0.81 (84405/104007)
Indel Het/Hom ratio	0.78 (11378/14661)
Insertion/Deletion ratio	0.78 (147622/188412)
Indel/SNP+MNP ratio	0.10 (362073/3585738)

Calidad y filtrado de variantes

- **Failed Filters:** 0
- **Passed Filters:** 4,203,532

El hecho de que todas las variantes hayan pasado los filtros (*Passed Filters:* 4,203,532) y ninguna haya fallado (*Failed Filters:* 0) es un indicativo positivo de la calidad general de las variantes en el archivo. Esto sugiere que las variantes identificadas cumplen con los criterios de calidad predefinidos, que suelen estar relacionados con factores como la profundidad de lectura, la calidad y la fiabilidad de la búsqueda de variantes. En estudios genéticos, tener una alta tasa de variantes que pasan los filtros es crucial, ya que aumenta la confianza en los datos y reduce la posibilidad de que errores técnicos influyan en los resultados.

Distribución de tipos de variantes

- **SNP:** 3,500,772
- **MNP:** 84,966
- **Insertions:** 147,622
- **Deletions:** 188,412
- **Indels:** 26,039

La mayoría de las variantes identificadas son **SNP** (3,500,772), lo cual es típico, ya que los SNP son la forma más común de variación genética en los genomas. Los **MNP** (84,966) representan un tipo menos común de variación, donde varios nucleótidos consecutivos cambian. La presencia de un número significativo de **Inserciones** (147,622) y **Deleciones** (188,412) indica que hay una considerable cantidad de variación estructural en la secuencia genómica, lo que podría tener implicaciones funcionales importantes, como la alteración de marcos de lectura o la interrupción de genes.

Los **Indel** (26,039), son menos frecuentes que los SNP y MNP, pero son importantes porque pueden causar cambios más drásticos en la secuencia de ADN y, por consiguiente, en las posibles secuencias codificantes como la introducción de paradas de lectura prematuras o cambios en la secuencia codificante que pueden afectar la función proteica.

Proporciones de variantes

- ***SNP Transitions/Transversions:*** 2.03 (3,350,500/1,649,019)

La proporción de transiciones a transversiones (2.03) es un dato importante en genómica, ya que refleja la naturaleza del cambio de bases en las variantes SNP. Las transiciones son más comunes que las transversiones. Un cociente mayor a 2 es típico en la mayoría de los organismos y sugiere que las variantes observadas siguen un patrón evolutivo esperado.

- ***Total Het/Hom ratio:*** 1.28 (2,213,959/1,733,852)

El cociente las proporciones de heterocigosidad y homocigosidad por tipo de variante, observamos que los SNP y MNP tienen cocientes superiores a 1, lo que refuerza la idea de alta diversidad genética para estas variantes. En contraste, las proporciones para inserciones, deleciones e INDEL son menores que 1, indicando una prevalencia mayor de variantes homocigotas en estos casos. Esto podría sugerir que las variantes estructurales son más estables en estado homocigoto o que hay una selección en contra de variantes heterocigotas en estas categorías.

- ***Insertion/Deletion ratio:*** 0.78 (147,622/188,412)

Este indicador muestra que hay más deleciones que inserciones en el genoma analizado. Esto podría tener diferentes implicaciones según el contexto biológico, como una mayor tendencia a perder segmentos de ADN en ciertas regiones del genoma, lo que podría afectar la estabilidad genómica o la función génica.

-
- **Indel/SNP+MNP ratio:** 0.10 (362,073/3,585,738)

La proporción de INDEL en relación con SNP y MNP es bajo (0.10), lo que refleja la naturaleza generalmente menos frecuente de los INDEL en comparación con las variantes puntuales. Sin embargo, a pesar de su menor frecuencia, los INDEL pueden tener efectos más significativos en la estructura y función del ADN, ya que pueden alterar el marco de lectura de los genes o causar mutaciones más disruptivas.

Interpretación general

En resumen, la tabla indica que el archivo VCF contiene un gran número de variantes de alta calidad, con una predominancia de SNP, lo que es típico en estudios genómicos. La diversidad genética parece ser alta, como lo sugiere la proporción de heterocigotos vs. homocigotos en varias categorías de variantes. Las métricas de transiciones a transversiones y los cocientes de inserciones vs. deleciones ofrecen información sobre la naturaleza y frecuencia de los cambios genómicos, con implicaciones potenciales en la estabilidad genómica y la evolución. Estas estadísticas pueden ser útiles para estudios de asociación genética, investigaciones sobre la variabilidad genética dentro de una población, o para comprender las bases genéticas de ciertas enfermedades o rasgos.

Estadísticas tras el filtrado por SNP e INDEL

Un paso importante a la hora de visualizar variantes es filtrar por tipos. La *Tabla 3* recoge las estadísticas generadas por RTGtools tras realizar un filtrado por SNP.

Tabla 3. Estadísticas del fichero VCF tras el filtrado por SNP.

Location	SNP_filtrados.vcf
Failed Filters	0
Passed Filters	179457
SNPs	179457
MNPs	0
Insertions	0
Deletions	0
Indels	0
SNP Transitions/Transversions	1.81 (115699/63758)
Total Het/Hom ratio	- (179457/0)
SNP Het/Hom ratio	- (179457/0)
MNP Het/Hom ratio	- (0/0)
Insertion Het/Hom ratio	- (0/0)
Deletion Het/Hom ratio	- (0/0)
Indel Het/Hom ratio	- (0/0)
Insertion/Deletion ratio	- (0/0)
Indel/SNP+MNP ratio	0.00 (0/179457)

Tras el proceso de filtrado de calidad aplicado al archivo VCF original, se observó una reducción significativa en el número de variantes detectadas. Inicialmente, el archivo contenía más de 4,2 millones de variantes, incluyendo SNP, MNP, inserción y delección. Después del filtrado, solo quedaron 179,457 variantes, todas ellas SNP, eliminándose completamente otros tipos de variantes como MNP e INDEL. Además, la relación de transición a transversión en los SNP pasó de 2.03 a 1.81, indicando un cambio en la composición de las variantes seleccionadas. Esto refleja un enfoque en retener solo las variantes de mayor calidad y relevancia, especialmente el SNP, tras el proceso de filtrado.

Otras variantes características de Freebayes son las inserciones y deleciones pequeñas. Para observarlas, se realizó un filtrado en el fichero original de Freebayes con el fin de obtener ese tipo de variantes. Las estadísticas se recogen en la *Tabla 4*.

Tabla 4. Estadísticas del fichero VCF tras el filtrado por INDEL.

Location	INDEL_filtrados.vcf
Failed Filters	0
Passed Filters	322922
SNPs	5432
MNPs	7057
Insertions	6245
Deletions	7238
Indels	968
Missing Genotype	295975
SNP Transitions/Transversions	0.97 (3574/3683)
Total Het/Hom ratio	2.03 (18041/8899)
SNP Het/Hom ratio	2.06 (3658/1774)
MNP Het/Hom ratio	3.65 (5539/1518)
Insertion Het/Hom ratio	1.25 (3474/2771)
Deletion Het/Hom ratio	1.79 (4647/2591)
Indel Het/Hom ratio	2.95 (723/245)
Insertion/Deletion ratio	0.86 (6245/7238)
Indel/SNP+MNP ratio	1.16 (14451/12489)

Las estadísticas obtenidas por **rtg vcfstats** sobre el archivo VCF filtrado para INDEL muestran un total de 322,922 variantes, de las cuales hay 6,245 inserciones, 7,238 deleciones y 968 indels, junto una cantidad de SNP (5,432) y MNP (7,057). Aunque el objetivo del filtrado era obtener únicamente INDEL (inserciones, deleciones e indels), la presencia de SNP y MNP sugiere que los parámetros de filtrado utilizados no lograron excluir completamente estas variantes puntuales. Esto podría deberse a la manera en que se definieron los filtros o a la interpretación de ciertas variantes complejas que pueden no ser categorizadas estrictamente como INDEL por las herramientas de filtrado, lo cual permitió que algunos SNP y MNP se incluyeran en el archivo filtrado final.

Visualización con IGV

Una vez analizados, podemos pasar a observar los datos a través del programa IGV, que permite navegar por el genoma alineado e identificar regiones de posicionamiento de variantes que puedan ser de interés. Para hacer una toma de contacto, vamos a visualizar el fichero VCF sin filtrar (Ilustración 5) y posteriormente los filtrados por SNP e INDEL (Ilustración 6):

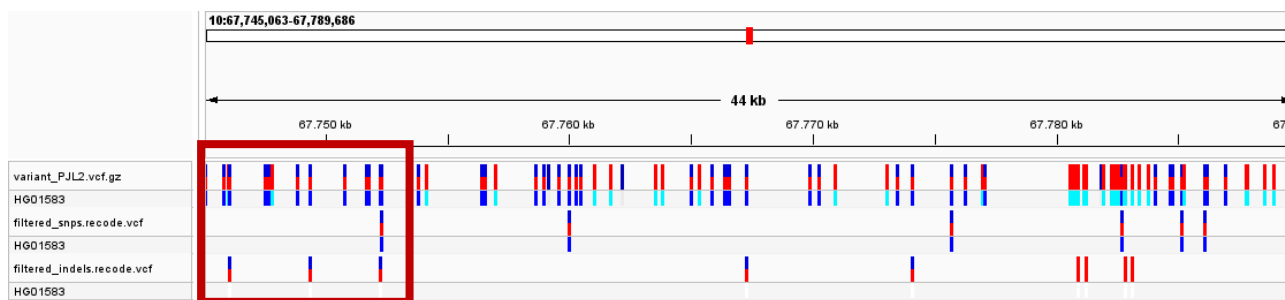


Ilustración 5. Visión general de las variantes de una región cromosómica de 44kb contenida en el cromosoma 10 del genoma. En la región recuadrada en rojo se encuentra el gen CTNNA3, el cual codifica una proteína conocida como alfa T-catenina, que está involucrada en la formación y mantenimiento de las uniones adherentes entre las células.

Para apreciar mejor los cambios, se ha seleccionado una región más concreta de la imagen anterior:

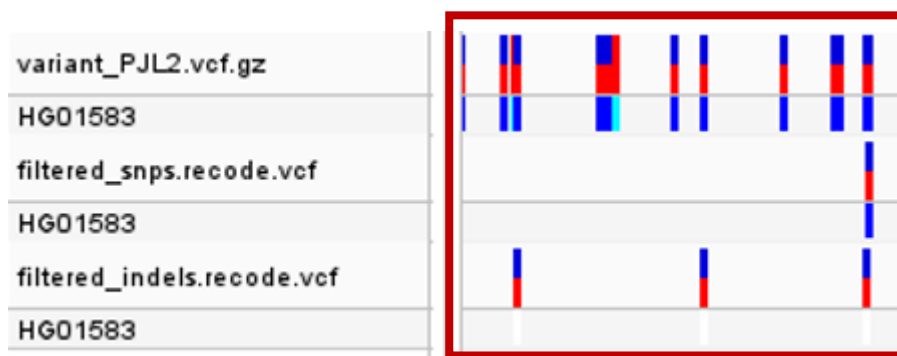


Ilustración 6. Comparativa de variantes. En la sección superior (variant_PJL2.vcf.gz) se muestran todas las variantes, e inmediatamente debajo aquellas que son SNP (filtered_snps.recode.vcf) e inserciones y deleciones pequeñas (filtered_indels.recode.vcf)

El fichero **variant_PJL2.vcf.gz** es el resultado de ejecutar Freebayes. Como se puede observar, contiene tanto variantes de un solo nucleótido (SNP) como inserciones y deleciones (INDEL). Tras aplicar el filtrado por SNP e INDEL, acompañado a su vez de un filtrado de falsos positivos detallado en el apartado "Filtrado de variantes – SNP e INDEL", muchas variantes previamente detectadas en el análisis simple que realiza Freebayes desaparecen tras el filtrado, como se puede observar en la representación gráfica de los ficheros **filtered_snps.recode.vcf** y **filtered_indels.recode.vcf**.

8.2. Manta

Los principales resultados de Manta son un conjunto de archivos VCF 4.1, que se encuentran en `${RUTA_DIRECTORIO_ANALISIS}/results/variants`. Se crean 3 archivos VCF para un análisis germinal, y se produce un VCF somático adicional para la sustracción tumor/sano. Estos archivos son:

- **diploidSV.vcf.gz**
 - Variantes estructurales (SV) e INDEL puntuados y genotipados bajo un modelo diploide para el conjunto de muestras en un análisis conjunto de muestra diploide o para la muestra normal en un análisis de sustracción tumor/normal. En el caso de una sustracción tumor/normal, las puntuaciones en este archivo no reflejan ninguna información de la muestra tumoral.
- **somaticSV.vcf.gz**
 - SV e indel puntuados bajo un modelo de variantes somáticas. Este archivo solo se producirá si se suministra un archivo de alineamiento de muestra tumoral durante la configuración. En mi caso, al no proporcionarlo, este fichero no se crea.
- **candidateSV.vcf.gz**
 - Candidatos a SV e INDEL no puntuados. Solo se requiere una cantidad mínima de evidencia de soporte para que un SV se registre como candidato en este archivo. Un SV o INDEL debe ser candidato para ser considerado para puntuación, por lo tanto, un SV no puede aparecer en los otros resultados de VCF si no está presente en este archivo. Cabe señalar que, por defecto, este archivo incluye INDEL de tamaño 50 y mayores.
- **candidateSmallIndels.vcf.gz**
 - Subconjunto del archivo candidateSV.vcf.gz que contiene solo variantes simples de inserción y delección de tamaño inferior al tamaño mínimo de variantes puntuadas (50 por defecto). Pasar este archivo a un buscador de variantes pequeñas proporcionará una cobertura continua sobre todos los tamaños de INDEL cuando se evalúen juntos los resultados del programa de búsqueda de variantes pequeñas y los de Manta. Se pueden extraer conjuntos alternativos de

candidatos a INDEL pequeños del archivo candidateSV.vcf.gz si este conjunto de candidatos no es apropiado.

Obtención de estadísticas

Se obtuvieron unas estadísticas iniciales del fichero VCF sin filtrar de la misma forma que con Freebayes, es decir, usando la herramienta RTGtools.

*Tabla 5. Estadísticas iniciales del fichero VCF generado por Manta. Se obtuvieron siguiendo las pautas descritas para **RTGtools** en la sección de **filtrado de variantes**.*

Location	diploidSV.vcf.gz
Failed Filters	204
Passed Filters	29
SNPs	0
MNPs	0
Insertions	3
Deletions	18
Indels	6
Breakends	2
SNP Transitions/Transversions	- (0/0)
Total Het/Hom ratio	2.22 (20/9)
SNP Het/Hom ratio	- (0/0)
MNP Het/Hom ratio	- (0/0)
Insertion Het/Hom ratio	0.00 (0/3)
Deletion Het/Hom ratio	3.50 (14/4)
Indel Het/Hom ratio	2.00 (4/2)
Breakend Het/Hom ratio	- (2/0)
Insertion/Deletion ratio	0.17 (3/18)
Indel/SNP+MNP ratio	- (27/0)

El análisis de variantes estructurales con Manta ha identificado un total de 29 variantes que pasaron los filtros, incluyendo 3 inserciones, 18 delecciones, 6 indel y 2 puntos de ruptura. Sin embargo, 204 variantes no pasaron los filtros, lo que probablemente se deba a la baja cobertura en el archivo BAM de alineamiento. Para mejorar la precisión de los resultados en estudios posteriores, se recomienda utilizar una mayor cobertura de secuenciación.

*Tabla 6. Estadísticas del fichero VCF tras aplicar el comando de filtrado con bcftools. El comando completo puede observarse en la sección de **filtrado de variantes**.*

Location	resultado_filtrado.vcf
Failed Filters	0
Passed Filters	18
SNPs	0
MNPs	0
Insertions	0
Deletions	12
Indels	4
Breakends	2
SNP Transitions/Transversions	- (0/0)
Total Het/Hom ratio	17.00 (17/1)
SNP Het/Hom ratio	- (0/0)
MNP Het/Hom ratio	- (0/0)
Insertion Het/Hom ratio	- (0/0)
Deletion Het/Hom ratio	- (12/0)
Indel Het/Hom ratio	3.00 (3/1)
Breakend Het/Hom ratio	- (2/0)
Insertion/Deletion ratio	0.00 (0/12)
Indel/SNP+MNP ratio	- (16/0)

El análisis del archivo resultado_filtrado.vcf muestra que 18 variantes han pasado los filtros, incluyendo 12 delecciones, 4 indel y 2 puntos de ruptura, sin variantes que hayan fallado en los filtros. Sin embargo, la baja cantidad de variantes identificadas podría deberse también a una baja cobertura en el archivo BAM de alineamiento. Para obtener un análisis más completo en estudios futuros, se recomienda aumentar la cobertura de secuenciación.



Contenido del fichero VCF

El fichero **diploidSV.vcf.gz** contiene variantes estructurales puntuadas y genotipadas, por lo que es ideal para la visualización y el análisis detallado en herramientas como IGV.

Como ejemplo, esta sería la información relativa a la primera variante estructural encontrada por Manta en el fichero BAM de alineamiento **HG01583**, que corresponde con el genoma completo de la población Punjabi de Lahore, Pakistan (Ilustración 7).

```
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT HG01583
1 4676497 MantaDEL:14:0:0:0:0:0 AGGGCACCTGGTAGGCAGGTAGGATACCTCTTATACAGCT
AGGGAACCTGGTAGGCAGAGAGGACACCTCTGATACAGCTAGGGCACCTGGTAGGCAGGGAGGACACCTGGTAGACTGGA
AGGGGAGAGAAGGACGCCTTGTCAGATC ATA 88 SampleFT END=4676644;SVTY
PE=DEL;SVLEN=-147;CIGAR=1M2I147D GT:FT:GQ:PL:PR:SR 0/1:MinGQ:11:137
,0,8:0,0:1,3
```

Ilustración 7. Información de la primera variante estructural contenida en el fichero **diploidSV.vcf.gz** obtenido del llamado de variantes del genoma completo de la población Punjabi, con identificador **HG01583**.

Esta variante es una delección (DEL) en el cromosoma 1, que comienza en la posición 4,676,497 y termina en 4,676,644, con una longitud total de 147 bases. La calidad de la variante es alta (QUAL = 88), pero el filtro aplicado indica que la muestra no pasó el filtro de calidad del genotipo (SampleFT). La información de CIGAR muestra que la variante se identifica con una combinación de coincidencias, inserciones y eliminaciones en el alineamiento. El genotipo para la muestra HG01583 es heterocigoto (0/1), pero la calidad del genotipo (GQ = 11) es baja, lo que sugiere que la confianza en esta llamada de variante es relativamente baja.

Tras aplicar el filtrado destinado para variantes estructurales, se obtiene el siguiente resultado al visualizar la primera variante (Ilustración 8):

```
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT HG01583
4 34453419 MantaINS:2557:0:0:0:0:0 AT ACATTCCATATATATAATACATTC
CATGTATATAAATACATGGAATACATTCCATGTATA 128 PASS END=34453420;SVTYPE=INS;
SVLEN=59;CIGAR=1M59I1D GT:FT:GQ:PL:PR:SR 0/1:PASS:42:178,0,39:0,0:4,4
```

Ilustración 8. Información de la primera variante estructural contenida en el fichero **diploidSV.vcf.gz** filtrado.

Como se puede observar, la variante que aparecía en el fichero sin filtrar ha desaparecido y en su lugar hay una inserción que cumple con los criterios de calidad establecidos tras el filtrado.

El contenido de esta variante describe una inserción en el cromosoma 4, específicamente en la posición 34,453,419, en el genoma de la muestra HG01583. La referencia en esta posición es una adenina y una timina ("AT"), mientras que la variante alternativa introduce una secuencia considerablemente más larga:

"ACATTCCATATATATAATACATTCCATGTATATAAATACATGGAATACATTCCATGTATA", con una longitud total de 59 bases.

El campo "QUAL", que indica la calidad de la variante, tiene un valor de 128, lo que sugiere que se confía en la precisión de esta búsqueda de variante. El filtro "PASS" indica que la variante ha pasado todos los criterios de filtrado.

En la sección "INFO", se proporcionan detalles clave sobre esta inserción. El parámetro "END" señala que la variante termina en la posición 34,453,420, lo que implica que la inserción ocurre entre estas dos posiciones. El tipo de variante ("SVTYPE") es una inserción ("INS"), como lo indica el campo. El tamaño de la inserción ("SVLEN") es de 59 nucleótidos. Además, el campo "CIGAR" describe el patrón de alineación entre las secuencias de referencia y alternativa, indicando que hay una base coincidente ("1M") seguida de una inserción de 59 nucleótidos ("59I"), y finalmente una eliminación de una base ("1D"), lo que se corresponde con la secuencia presentada.

En la parte de "FORMAT" se presentan varios parámetros relacionados con el genotipo de la muestra. El "GT" muestra el genotipo "0/1", lo que indica que el individuo es heterocigoto para esta inserción, es decir, tiene una copia del alelo de referencia y una copia del alelo alternativo. El campo "FT" muestra "PASS", lo que sugiere que no hay problemas con la calidad del genotipo de esta muestra. La calidad del genotipo ("GQ") es de 42, lo que indica una buena confiabilidad en la búsqueda del genotipo. El campo "PL" representa las probabilidades Phred escaladas para los posibles genotipos, con valores de "178,0,39", lo que sugiere que el genotipo heterocigoto es el más probable (0/1), mientras que el genotipo homocigoto alternativo (1/1) tiene una menor probabilidad.

Finalmente, "PR" y "SR" son indicadores de soporte de lectura. "PR" refleja el soporte de lecturas pareadas, y en este caso, ambos valores son cero, lo que sugiere que no hay lecturas emparejadas que respalden ni el alelo de referencia ni el alternativo. En cambio, "SR" se refiere a las lecturas divididas, y muestra que hay cuatro lecturas que respaldan tanto el alelo de referencia como el alternativo, lo que confirma la presencia de la inserción.

Visualización de los resultados

Las variantes también se pueden visualizar en IGV con la respectiva información asociada que proporciona el programa (Ilustración 9 e Ilustración 10):

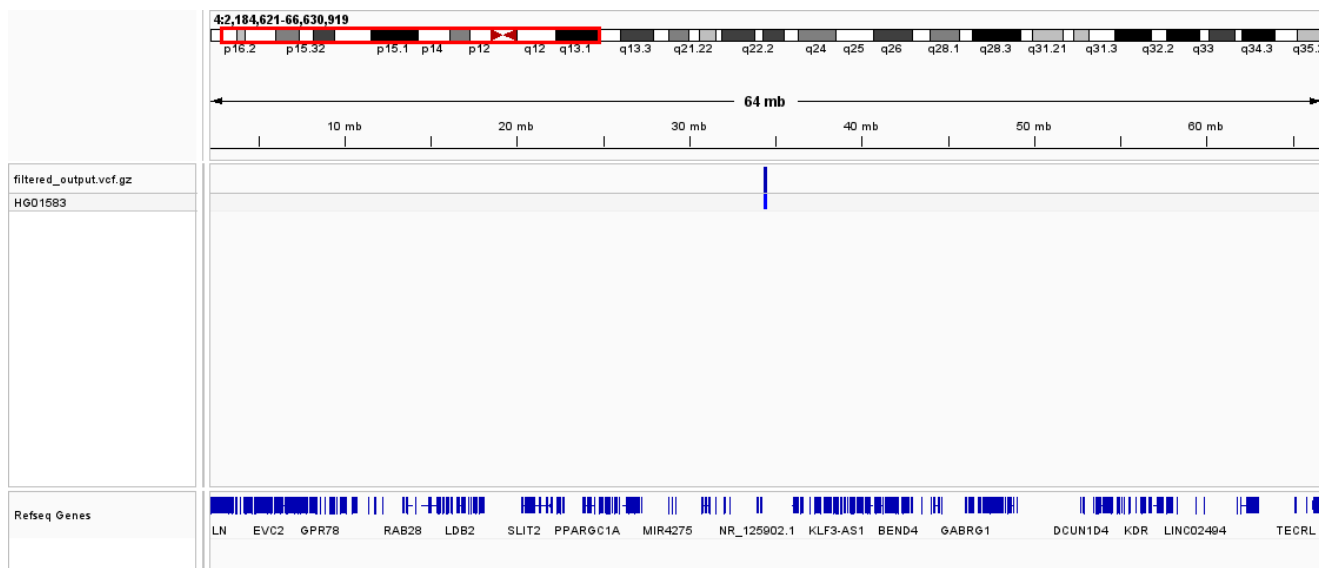



Ilustración 9. Visualización en IGV de la primera variante estructural (cromosoma 4) contenida en el fichero **diploidSV.vcf.gz** filtrado

 filtered_output.vcf.gz

ID: MantaNS:2557:0:0:0:0

Chr: 4

Position: 34,453,419-34,453,420

Reference: AT*

Alternate: ACATTCCATATATATAATACATTCCATGTATATAAATACATGGAATACATTCCATGTATA

Qual: 128

Type: INDEL

Is Filtered Out: No

Alleles:

Alternate Alleles: ACATTCCATATATATAATACATTCCATGTATATAAATACATGGAATACATTCCATGTATA

Variant Attributes

CIGAR: 1M59I1D

SVTYPE: INS

END: 34453420

SVLEN: 59

Ilustración 10. Información adicional asociada a la variante señalada en la ilustración 10

8.3. CNVnator

Al comenzar a analizar un fichero VCF es una buena idea revisar algunas estadísticas preliminares para obtener una visión general de los datos. En este caso, es útil observar que no hay SNP ni INDEL presentes en el fichero. Esto es especialmente relevante cuando se utiliza CNVnator, ya que es una herramienta diseñada exclusivamente para la detección de CNV. Esto se puede

observar en la Tabla 7 del informe, donde los "Symbolic SV" (variantes estructurales simbólicas) reflejan las detecciones de CNV que CNVnator ha identificado.

Tabla 7. Estadísticas generadas por RTG vcfstats del fichero VCF de llamado de variantes del genoma completo de 3 muestras (HG01583, HG01586, HG01589) de la población Punjabi en Lahore, Pakistán.

Location	PJL_complete.vcf.gz
Failed Filters	0
Passed Filters	1464
SNPs	0
MNPs	0
Insertions	0
Deletions	0
Indels	0
Symbolic SVs	1332
Partial Genotype	132
SNP Transitions/Transversions	- (0/0)
Total Het/Hom ratio	1.72 (843/489)
SNP Het/Hom ratio	- (0/0)
MNP Het/Hom ratio	- (0/0)
Insertion Het/Hom ratio	- (0/0)
Deletion Het/Hom ratio	- (0/0)
Indel Het/Hom ratio	- (0/0)
Symbolic SV Het/Hom ratio	1.72 (843/489)
Insertion/Deletion ratio	- (0/0)
Indel/SNP+MNP ratio	- (0/0)

CNVnator crea un fichero VCF con la siguiente estructura:

```
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT PJJL_complete
1 1 PJJL_CNVnator_del_1 N <DEL> . PASS END=10000;SVTYPE
=DEL;SVLEN=-10000;IMPRECISE;natorRD=0;natorP1=1.59373e-11;natorP2=3.32278e-122;natorP3=1
.99216e-11;natorP4=5.11456e-96;natorQ0=-1 GT:CN 1/1:0
```

Ilustración 11. Contenido del fichero VCF generado por CNVnator.

Esta variante en el cromosoma 1 indica una delección imprecisa de 10.000 bases, identificada por CNVnator (ID: PJJL_CNVnator_del_1). La calidad de la detección ha pasado los filtros, y el valor de SVTYPE confirma que es una delección (DEL), mientras que el campo SVLEN (-10000) refleja la longitud de la variante. Los valores de natorP1-P4 muestran la significancia estadística de la variante, y el genotipo (GT) 1/1 sugiere que la delección está presente en ambas copias del genoma de la muestra (CN = 0, lo que indica la ausencia de copias en la región).

Teniendo en cuenta los aspectos mencionados acerca de la calidad de los ficheros en la sección de filtrado de variantes, las estadísticas en relación con el fichero VCF filtrado son las siguientes (Tabla 8):

*Tabla 8. Estadísticas del fichero VCF generado por CNVnator tras aplicarle los filtros del apartado **filtrado de variantes**.*

Location	resultado_filtrado.vcf
Failed Filters	0
Passed Filters	177
SNPs	0
MNPs	0
Insertions	0
Deletions	0
Indels	0
Symbolic SVs	142
Partial Genotype	35
SNP Transitions/Transversions	- (0/0)
Total Het/Hom ratio	1.37 (82/60)
SNP Het/Hom ratio	- (0/0)
MNP Het/Hom ratio	- (0/0)

Insertion Het/Hom ratio	- (0/0)
Deletion Het/Hom ratio	- (0/0)
Indel Het/Hom ratio	- (0/0)
Symbolic SV Het/Hom ratio	1.37 (82/60)
Insertion/Deletion ratio	- (0/0)
Indel/SNP+MNP ratio	- (0/0)

Como se puede observar, se ha pasado de 1464 variantes en el fichero sin filtrar a 177 en el filtrado. Al comparar el bajo filtro que se ha aplicado con la cantidad de variantes estructurales simbólicas que han sido filtradas, se puede confirmar que los ficheros BAM de alineamiento originales para hacer el llamado de variantes no cumplen con unos criterios de calidad muy estrictos, al estar clasificados como "baja cobertura". Quizás sería de interés incluir ficheros con mayor cobertura en otros análisis comparativos.

Visualización de resultados

Las variantes se pueden visualizar a través de IGV (Ilustración 12), usando como referencia el genoma humano b37 junto con sus secuencias "decoy". El nombre específico de este genoma en la librería genómica de IGV es "Human (1kg, b37+decoy)" y equivale al genoma de referencia empleado anteriormente (hs37d5.fa).

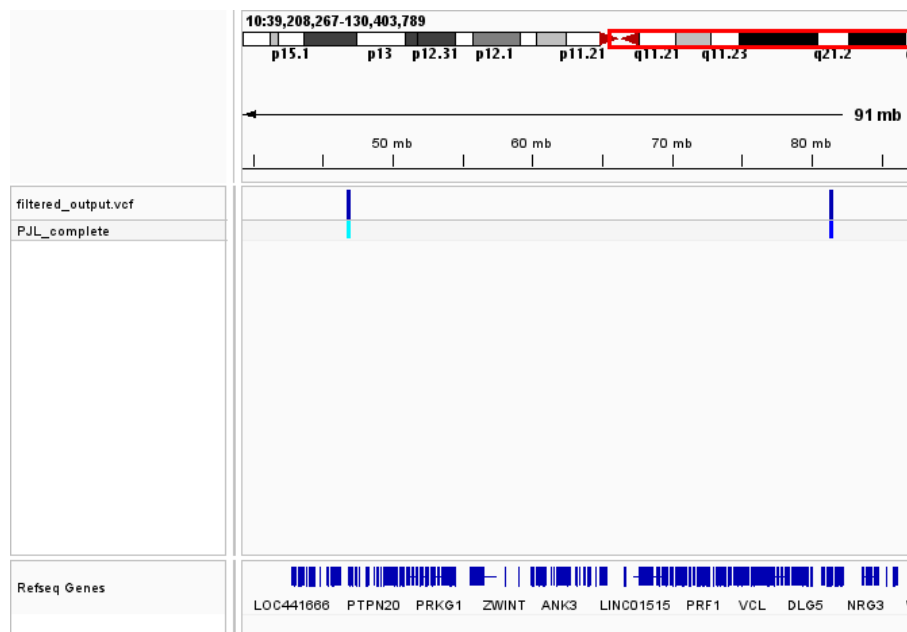


Ilustración 12. Visualización en IGV de una duplicación y deleción en el fichero filtrado tras el llamado de variantes de CNVnator. Ambas están localizadas en el mismo cromosoma (10), pero la duplicación está en la región del gen PTPN20 y la deleción en la región del gen DLG5.

La **primera variante estructural simbólica** indica una **duplicación** imprecisa de aproximadamente 204.000 bases en el cromosoma 10 (desde la posición 46,947,001 hasta 47,151,000), identificada por CNVnator (ID: PJJ_CNVnator_dup_527). El campo SVTYPE señala que es una duplicación (DUP), y el valor de SVLEN confirma la longitud del evento. El natorRD (2.05) indica un aumento en la densidad de lecturas, apoyando la duplicación. Los valores de natorP1-P4 sugieren una alta significancia estadística en la detección, y el genotipo (GT) 1/1 indica que la duplicación está presente en ambas copias del genoma, con un número de copias (CN) igual a 2.

La **segunda variante estructural simbólica** corresponde a una **delección** imprecisa en el cromosoma 10, entre las posiciones 81,474,001 y 81,505,000, con una longitud de aproximadamente 31.000 bases (SVLEN=-31000), detectada por CNVnator (ID: PJJ_CNVnator_del_541). El campo SVTYPE indica que se trata de una delección (DEL), y el valor de natorRD (0.46) sugiere una reducción en la densidad de lecturas, lo que respalda la delección. Los valores de natorP1-P4 indican una alta significancia estadística para la detección. El genotipo (GT) 0/1 sugiere que la delección está presente en una de las dos copias del genoma, y el número de copias (CN) es 1.

Ambos CNVs son detectados como imprecisos, lo que hace referencia a que su ubicación no ha podido ser definida con claridad. A pesar de esto, ambos eventos muestran alta significancia estadística según los valores p proporcionados, lo que sugiere que estos eventos son reales y relevantes. Sin embargo, la calidad de mapeo y la profundidad de lectura pueden afectar la interpretación de la fiabilidad de estos eventos, siendo especialmente relevante en el caso de la delección donde la cobertura es baja y la fracción de lecturas con baja calidad es alta.

8.4. Fortalezas y limitaciones de cada herramienta

A continuación, se detallan los aspectos positivos y ventajosos del uso de cada herramienta, junto con sus potenciales debilidades.

1. FreeBayes: Detección de SNP e INDEL

Fortalezas:

- **Precisión en SNP:** FreeBayes mostró un alto rendimiento en la detección de SNP, como lo evidencia el gran número de SNP detectados en el análisis no filtrado (3,500,772 variantes). La herramienta se destaca por su capacidad de identificar polimorfismos de un solo nucleótido con alta sensibilidad y especificidad, lo cual es fundamental en estudios de variabilidad genética y asociación de enfermedades.

- **Detección de INDEL:** Aunque la mayoría de las variantes detectadas fueron SNP, FreeBayes también demostró una competencia moderada en la identificación de INDEL, registrando un total de 147,622 inserciones y 188,412 deleciones. La capacidad de detectar variantes pequeñas (tanto inserciones como deleciones) es valiosa en el estudio de micro variaciones que podrían tener implicaciones funcionales.

Limitaciones:

- **Variantes estructurales grandes:** FreeBayes no está optimizado para la detección de variantes estructurales grandes, como duplicaciones, translocaciones o inversiones, lo que limita su aplicabilidad a la identificación de variantes de mayor escala.
- **Filtro de calidad:** Tras el proceso de filtrado, el número de variantes SNP se redujo significativamente, lo que podría indicar una susceptibilidad a falsos positivos en los datos sin filtrar o un enfoque de filtrado demasiado estricto que podría omitir variantes relevantes.

2. Manta: Detección de Variaciones Estructurales

Fortalezas:

- **Identificación de variantes estructurales:** Manta mostró una clara efectividad en la detección de variantes estructurales como inserciones, deleciones y puntos de ruptura (breakends). En los análisis no filtrados, se identificaron un total de 29 variantes, lo cual es indicativo de su capacidad para captar variaciones complejas en el genoma.
- **Adaptación a diferentes tipos de análisis:** Manta puede manejar análisis germinales y somáticos, proporcionando flexibilidad en estudios de genomas humanos normales y en contextos de cáncer, respectivamente.

Limitaciones:

- **Sensibilidad a la cobertura:** La herramienta mostró una alta tasa de fallos en los filtros (204 variantes fallidas), probablemente debido a una baja cobertura en los archivos BAM de alineamiento. Esto sugiere que Manta podría requerir una alta calidad de datos de entrada para producir resultados fiables.
- **Limitación en variantes pequeñas:** Manta está diseñado para variantes estructurales grandes y no captura eficientemente INDELs menores de 50 bases, lo que limita su aplicación en el análisis de variantes más pequeñas que podrían ser funcionalmente significativas.

3. CNVnator: Detección de CNV (Variaciones en el Número de Copias)

Fortalezas:

- **Especialización en CNV:** CNVnator es altamente efectivo en la detección de variaciones en el número de copias (CNV), lo cual fue evidente al detectar 1,332 variantes estructurales simbólicas en el análisis inicial. Su diseño especializado lo hace ideal para estudios que buscan identificar ganancias y pérdidas de secuencias en el genoma.
- **Manejo de variantes simbólicas:** La herramienta mostró una capacidad robusta para identificar CNV incluso en contextos de baja cobertura, destacando su efectividad en situaciones en las que otras herramientas podrían fallar.

Limitaciones:

- **Ausencia de detección de SNP e INDEL:** CNVnator no está diseñado para identificar SNP e INDEL, lo que lo hace menos versátil en comparación con herramientas como FreeBayes que pueden detectar una gama más amplia de variantes. Esto limita su aplicabilidad a estudios que requieren un análisis integral de todas las formas de variación genética.
- **Requerimiento de calidad en datos de entrada:** Aunque mostró solidez en la detección de CNV, la calidad de los datos de entrada sigue siendo crucial, ya que la herramienta puede pasar por alto CNV en contextos de cobertura extremadamente baja o ruido elevado en los datos.

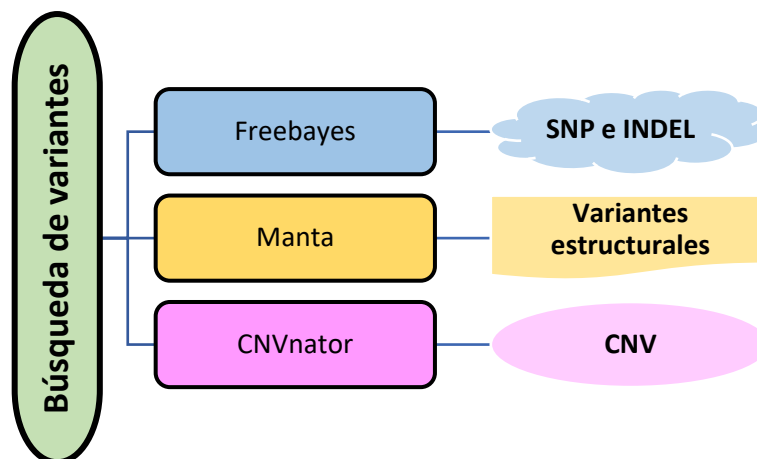
8.5. Integración de un protocolo de análisis que combine las tres herramientas

Uno de los objetivos del proyecto es establecer un protocolo de análisis de variantes en el que se combinen distintas herramientas especializadas, optimizando la detección de variantes genéticas , cubriendo desde variantes pequeñas hasta grandes alteraciones estructurales.

FreeBayes se especializa en la detección de **SNP** e **INDEL**, mostrando alta precisión en polimorfismos de un solo nucleótido y microvariaciones, esenciales en estudios de variabilidad genética. Sin embargo, su capacidad es limitada para detectar **variantes estructurales grandes**, por lo que es necesario complementarlo con otras herramientas. **Manta** cubre este vacío al ser altamente eficaz en la detección de **variantes estructurales complejas**, como grandes inserciones, translocaciones y puntos de ruptura, lo que la hace ideal para estudios que requieren la identificación de alteraciones a gran escala. Aunque su eficacia en variantes pequeñas es limitada, esta herramienta compensa la falta de FreeBayes en grandes variantes.

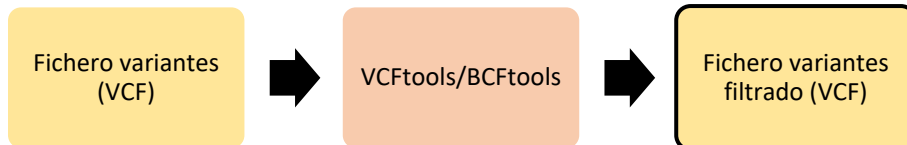
CNVnator se incorpora al protocolo para identificar **variaciones en el número de copias** , un tipo de variante que FreeBayes y Manta no pueden detectar eficientemente. Su especialización en CNV lo convierte en una herramienta clave para estudios genómicos completos.

En resumen, este protocolo asegura una **detección exhaustiva de variantes genéticas**, combinando la precisión de FreeBayes en variantes pequeñas, la capacidad de Manta en variaciones estructurales grandes y la especialización de CNVnator en CNV, proporcionando así un análisis integral del genoma. La Secuencia 1 resume gráficamente el protocolo de trabajo:



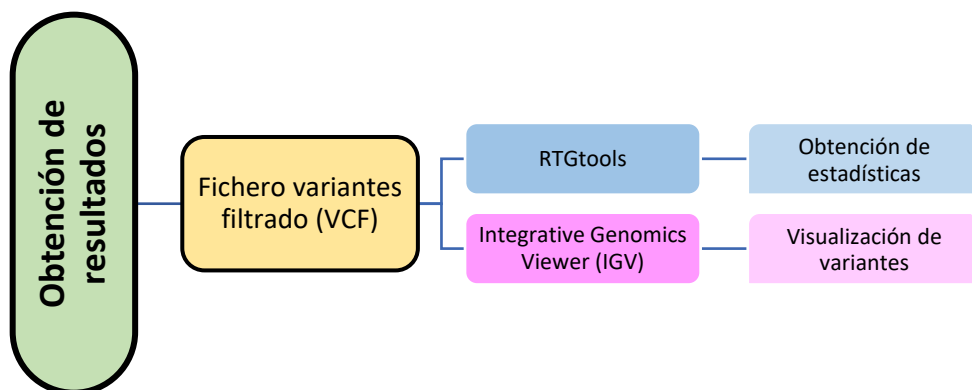
Secuencia 1. Tras obtener los ficheros BAM de alineamiento y el fichero FASTA del genoma de referencia, se realiza la búsqueda de variantes combinando las 3 herramientas. El objetivo es ampliar el rango de variantes, destinando a Freebayes para la detección de SNP e INDEL, Manta para identificar variantes estructurales y CNVnator para detectar CNV.

Cada herramienta genera un fichero VCF con las variantes encontradas. El siguiente paso es filtrar esas variantes usando herramientas como VCFtools o BCFtools . Las variantes se filtran en función de varios parámetros, como la calidad mínima de mapeo, la profundidad máxima y mínima de cobertura, el número máximo de alelos, etc. El protocolo de filtrado para cada tipo de variante viene recogido en el apartado **Filtrado de variantes**.



Secuencia 2. Filtrado de ficheros con variantes (VCF). Se emplean herramientas como VCFtools o BCFtools para filtrar los resultados obtenidos de cada herramienta. Este paso adicional antes de la obtención de resultados es bastante útil para eliminar falsos positivos y analizar solamente variantes que cumplan los criterios de calidad.

Una vez obtenidos los ficheros VCF filtrados, se obtienen estadísticas mediante la herramienta RTGtools. Entre las principales estadísticas se incluye el número total de variantes identificadas, desglosando entre variantes SNP y variaciones de tipo indel (inserciones y deleciones). También ofrece información sobre la distribución de los tipos de variantes, como transiciones y transversiones. Además, se muestran datos sobre la calidad de las variantes, como los valores de filtrado, que indican cuántas variantes pasaron o no ciertos criterios de calidad, y la cantidad de variantes por muestra en los casos de estudios con múltiples individuos. A su vez, se emplea el programa IGV para visualizar las variantes en comparación con el genoma de referencia, además de obtener información adicional sobre las mismas.



Secuencia 3. Obtención de resultados. Se utiliza la herramienta RTGtools para generar estadísticas sobre los ficheros VCF filtrados y el programa IGV para visualizar y obtener información adicional de las variantes

9. Conclusiones

En base a los resultados obtenidos, a continuación, se describen las conclusiones extraídas del presente trabajo:

- Para el análisis de variantes SNP, el programa FreeBayes mostró una excelente capacidad de detección de variantes, con alta sensibilidad y especificidad. Así mismo, se ha observado que también detecta INDEL de forma moderadamente eficaz. Sin embargo, no es adecuado para identificar variantes estructurales grandes como duplicaciones o translocaciones, puede ser susceptible a falsos positivos en datos no filtrados, aunque, es preciso evitar un filtrado estricto para evitar la pérdida de variantes relevantes.
- En la detección de variantes estructurales grandes, como inserciones y deleciones complejas, el programa Manta ha probado su solvencia, tanto análisis germinales como somáticos. No obstante, su precisión depende en gran medida de una alta cobertura de los datos de entrada, pudiendo fallar en la detección de variantes estructurales si la cobertura de los archivos BAM es baja.
- En el caso de la detección de variaciones en el número de copias, CNVnator es un programa altamente eficiente, incluso en ficheros con baja cobertura. Ahora bien, la calidad de los datos de entrada es crucial: en contextos de baja calidad, puede pasar por alto CNV.
- La búsqueda de variantes combinando el uso de FreeBayes, Manta y CNVnator ofrece un análisis integral del genoma completo, permitiendo sumar las fortalezas de las tres herramientas y superar las limitaciones que presenta la identificación de los distintos tipos de variantes existentes en los genomas. Su combinación proporciona una mayor precisión y confianza en la identificación de variantes genéticas, lo que supone una estrategia útil para estudios de variabilidad genética y su asociación con enfermedades complejas.

10. Referencias

- Abyzov, A., Urban, A. E., Snyder, M., & Gerstein, M. B. (2011a). CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Research*, 21 6, 974–984. <https://api.semanticscholar.org/CorpusID:35941407>
- Abyzov, A., Urban, A. E., Snyder, M., & Gerstein, M. B. (2011b). CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Research*, 21 6, 974–984. <https://api.semanticscholar.org/CorpusID:35941407>
- Biscotti, M. A., Olmo, E., & Heslop-Harrison, P. J. S. (2015). Repetitive DNA in eukaryotic genomes. *Chromosome Research*, 23, 415–420. <https://api.semanticscholar.org/CorpusID:16210689>
- Boeva, V., Popova, T. G., Bleakley, K., Chiche, P., Cappel, J., Schleiermacher, G., Janoueix-Lerosey, I., Delattre, O., & Barillot, E. (2011). Control-FREEC: a tool for assessing copy number and allelic content using next-generation sequencing data. *Bioinformatics*, 28, 423–425. <https://api.semanticscholar.org/CorpusID:270225952>
- Broad Institute. (2020). *(How to) Call rare germline copy number variants*. <https://gatk.broadinstitute.org/Hc/En-US/Articles/360035531152--How-to-Call-Rare-Germline-Copy-Number-Variants>.
- Buniello, A., MacArthur, J. A. L., Cerezo, M., Harris, L. W., Hayhurst, J. D., Malangone, C., McMahon, A. C., Morales, J., Mountjoy, E., Sollis, E., Suveges, D., Vrousitou, O., Whetzel, P. L., Amode, R. M., Guillen, J. A., Riat, H. S., Trevanion, S. J., Hall, P., Junkins, H., ... Parkinson, H. E. (2018). The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. *Nucleic Acids Research*, 47, D1005–D1012. <https://api.semanticscholar.org/CorpusID:53568759>
- Chen, X., Schulz-Trieglaff, O., Shaw, R., Barnes, B., Schlesinger, F., Källberg, M., Cox, A. J., Kruglyak, S., & Saunders, C. T. (2016). Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics*, 32 8, 1220–1222. <https://api.semanticscholar.org/CorpusID:7620813>
- Choi, Y., Sims, G. E., Murphy, S., Miller, J. R., & Chan, A. P. (2012). Predicting the Functional Effect of Amino Acid Substitutions and Indels. *PLoS ONE*, 7. <https://api.semanticscholar.org/CorpusID:7171705>
- Cooper, G. M., Coe, B. P., Girirajan, S., Rosenfeld, J. A., Vu, T. H., Baker, C. A., Williams, C. A., Stalker, H. J., Hamid, R., Hannig, V. L., Abdel-Hamid, H. Z., Bader, P. I., McCracken, E., Niyazov, D. M., Leppig, K. A., Thiese, H. A., Hummel, M., Alexander, N., Gorski, J. L., ...

- Eichler, E. E. (2011). A Copy Number Variation Morbidity Map of Developmental Delay. *Nature Genetics*, 43, 838–846. <https://api.semanticscholar.org/CorpusID:9186422>
- Duan, J., Zhang, J.-G., Deng, H.-W., & Wang, Y. (2013). Comparative Studies of Copy Number Variation Detection Methods for Next-Generation Sequencing Technologies. *PLoS ONE*, 8. <https://api.semanticscholar.org/CorpusID:2347274>
- Eisfeldt, J., Vezzi, F., Olason, P. I., Nilsson, D., & Lindstrand, A. (2017). TIDDIT, an efficient and comprehensive structural variant caller for massive parallel sequencing data. *F1000Research*, 6. <https://api.semanticscholar.org/CorpusID:6974705>
- England, N., Ramsey, M. D. B. W., Davies, M. J., Ch.B., N. G. M., M.D., E. T., Bell, M. S. C., Ĥ evínek M.D., P. D., M.D., M. G., M.D., E. F. M., Wainwright, M. C. E., B.S., M. W. K., M.D., R. M., M.D., F. R., Sermet-Gaudelus, Ph. I., Ph.D., S. M. R., M.S.P.H., Q. D., Ph.D., S. R., Yen, M. K., M.D., C. O., & Elborn, M. J. S. (2011). A CFTR potentiator in patients with cystic fibrosis and the G551D mutation. *The New England Journal of Medicine*, 365 18, 1663–1672. <https://api.semanticscholar.org/CorpusID:4788308>
- Fairley, S., Lowy-Gallego, E., Perry, E., & Flicek, P. (2020). The International Genome Sample Resource (IGSR) collection of open human genomic variation resources. *Nucleic Acids Research*, 48(D1), D941–D947. <https://doi.org/10.1093/nar/gkz836>
- Fang, H., Bergmann, E. A., Arora, K., Vacic, V., Zody, M. C., Iossifov, I., O’Rawe, J., Wu, Y., Barrón, L. T. J., Rosenbaum, J., Ronemus, M., Lee, Y., Wang, Z.-H., Dikoglu, E., Jobanputra, V., Lyon, G. J., Wigler, M., Schatz, M. C., & Narzisi, G. (2015). Indel variant analysis of short-read sequencing data with Scalpel. *Nature Protocols*, 11, 2529–2548. <https://api.semanticscholar.org/CorpusID:14947825>
- Fang, H., Wu, Y., Yang, H., Yoon, M., Jiménez-Barrón, L. T., Mittelman, D. A., Robison, R. J., Wang, K., & Lyon, G. J. (2017). Whole genome sequencing of one complex pedigree illustrates challenges with genomic medicine. *BMC Medical Genomics*, 10. <https://api.semanticscholar.org/CorpusID:27616314>
- Frazer, K. A., Ballinger, D. G., Cox, D. R., Hinds, D. A., Stuve, L. L., Gibbs, R. A., Belmont, J. W., Boudreau, A. A., Hardenbol, P., Leal, S. M., Pasternak, S., Wheeler, D. A., Willis, T., Yu, F., Yang, H., Zeng, C., Gao, Y., Hu, H., Hu, W., ... Stewart, J. (2007). A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449, 851–861. <https://api.semanticscholar.org/CorpusID:4431729>
- Garrison, E. P., & Marth, G. T. (2012). Haplotype-based variant detection from short-read sequencing. *ArXiv: Genomics*. <https://api.semanticscholar.org/CorpusID:15153602>
- Hacisuleyman, E., Goff, L. A., Trapnell, C., Williams, A., Henao-Mejia, J., Sun, L., McClanahan, P. D., Hendrickson, D. G., Sauvageau, M., Kelley, D. R., Morse, M., Engreitz, J. M., Lander, E. S., Guttman, M., Lodish, H. F., Flavell, R. A., Raj, A., & Rinn, J. L. (2014). Topological organization of multichromosomal regions by the long intergenic noncoding RNA Firre.

-
- Nature Structural & Molecular Biology*, **21**, 198–206.
<https://api.semanticscholar.org/CorpusID:115422065>
- Hänzelmann, S., Castelo, R., & Guinney, J. (2013). GSVA: gene set variation analysis for microarray and RNA-Seq data. *BMC Bioinformatics*, **14**, 7–7.
<https://api.semanticscholar.org/CorpusID:10196486>
- Highnam, G., Franck, C. T., Martin, A., Stephens, C., Puthige, A., & Mittelman, D. A. (2012). Accurate human microsatellite genotypes from high-throughput resequencing data using informed error profiles. *Nucleic Acids Research*, **41**, e32–e32.
<https://api.semanticscholar.org/CorpusID:10591004>
- Hughes, J. R., Roberts, N., McGowan, S. J., Hay, D., Giannoulatou, E., Lynch, M. D., Gobbi, M., Taylor, S., Gibbons, R. J., & Higgs, D. R. (2014). Analysis of hundreds of cis-regulatory landscapes at high resolution in a single, high-throughput experiment. *Nature Genetics*, **46**, 205–212. <https://api.semanticscholar.org/CorpusID:205348099>
- Hwang, S., Kim, E., Lee, I., & Marcotte, E. M. (2015). Systematic comparison of variant calling pipelines using gold standard personal exome variants. *Scientific Reports*, **5**.
<https://api.semanticscholar.org/CorpusID:17722772>
- Kanehisa, M., & Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, **28**, 27–30. <https://api.semanticscholar.org/CorpusID:7449269>
- Kircher, M., Witten, D. M., Jain, P., O’Roak, B., Cooper, G. M., & Shendure, J. A. (2014). A general framework for estimating the relative pathogenicity of human genetic variants. *Nature Genetics*, **46**, 310–315. <https://api.semanticscholar.org/CorpusID:2593453>
- Korneliussen, T. S., Albrechtsen, A., & Nielsen, R. (2014). ANGSD: Analysis of Next Generation Sequencing Data. *BMC Bioinformatics*, **15**.
<https://api.semanticscholar.org/CorpusID:6472679>
- Laurie, S., Fernández-Callejo, M., Marco-Sola, S., Trotta, J.-R., Camps, J., Chacón, A., Espinosa, A., Gut, M., Gut, I. G., Heath, S. C., & Beltran, S. (2016). From Wet-Lab to Variations: Concordance and Speed of Bioinformatics Pipelines for Whole Genome and Whole Exome Sequencing. *Human Mutation*, **37**, 1263–1271.
<https://api.semanticscholar.org/CorpusID:17758737>
- Layer, R. M., Chiang, C., Quinlan, A. R., & Hall, I. M. (2012). LUMPY: a probabilistic framework for structural variant discovery. *Genome Biology*, **15**, R84–R84.
<https://api.semanticscholar.org/CorpusID:1528024>
- Liu, F., Zhang, Y., Zhang, L., Li, Z., Fang, Q., Gao, R., & Zhang, Z. (2019). Systematic comparative analysis of single-nucleotide variant detection methods from single-cell RNA sequencing data. *Genome Biology*, **20**. <https://api.semanticscholar.org/CorpusID:208144352>
- Margulies, M., Egholm, M. W., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y.-J., Chen, Z., Dewell, S. B., Du, L., Fierro, J. M., Gomes, X. V,
-

- Godwin, B. C., He, W., Helgesen, S., Ho, C. H., Irzyk, G. P., ... Rothberg, J. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, *437*, 376–380. <https://api.semanticscholar.org/CorpusID:85690135>
- Maston, G. A., Evans, S. K., & Green, M. R. (2006). Transcriptional regulatory elements in the human genome. *Annual Review of Genomics and Human Genetics*, *7*, 29–59. <https://api.semanticscholar.org/CorpusID:12346247>
- Mattick, J. S. A., Amaral, P. P., Carninci, P., Carpenter, S. B., Chang, H. Y., Chen, L.-L., Chen, R., Dean, C., Dinger, M. E., Fitzgerald, K. A., Gingeras, T. R., Guttman, M., Hirose, T., Huarte, M., Johnson, R., Kanduri, C., Kapranov, P., Lawrence, J. B., Lee, J. T., ... Wu, M. (2023). Long non-coding RNAs: definitions, functions, challenges and recommendations. *Nature Reviews Molecular Cell Biology*, 1–17. <https://api.semanticscholar.org/CorpusID:255456200>
- McKernan, K. J., Peckham, H. E., Costa, G. L., McLaughlin, S., Fu, Y., Tsung, E. F., Clouser, C., Duncan, C., Ichikawa, J. K., Lee, C. C., Zhang, Z., Ranade, S. S., Dimalanta, E., Hyland, F. C., Sokolsky, T. D., Zhang, L., Sheridan, A., Fu, H., Hendrickson, C. L., ... Blanchard, A. P. (2009). Sequence and structural variation in a human genome uncovered by short-read, massively parallel ligation sequencing using two-base encoding. *Genome Research*, *19*, 1527–1541. <https://api.semanticscholar.org/CorpusID:8786465>
- Nurk, S., Koren, S., Rhie, A., Rautiainen, M., Bizakadze, A. V., Mikheenko, A., Vollger, M. R., Altemose, N., Uralsky, L. I., Gershman, A., Aganezov, S. S., Hoyt, S. J., Diekhans, M. E., Logsdon, G. A., Alonze, M., Antonarakis, S. E., Borchers, M., Bouffard, G. G., Brooks, S. Y., ... Phillippy, A. M. (2021). The complete sequence of a human genome. *Science (New York, N.Y.)*, *376*, 44–53. <https://api.semanticscholar.org/CorpusID:235233625>
- Pentao, L., Wise, C. A., Chinault, A. C., Patel, P., & Lupski, J. R. (1992). Charcot–Marie–Tooth type 1A duplication appears to arise from recombination at repeat sequences flanking the 1.5 Mb monomer unit. *Nature Genetics*, *2*, 292–300. <https://api.semanticscholar.org/CorpusID:11304278>
- Plagnol, V., Curtis, J., Epstein, M., Mok, K. Y., Stebbings, E., Grigoriadou, S., Wood, N. W., Hambleton, S., Burns, S. O., Thrasher, A. J., Kumararatne, D., Doffinger, R., & Nejentsev, S. (2012). A robust model for read count data in exome sequencing experiments and implications for copy number variant calling. *Bioinformatics*, *28*, 2747–2754. <https://api.semanticscholar.org/CorpusID:7479524>
- Poplin, R., Chang, P.-C., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., Newburger, D., Djamco, 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>
- Poplin, R., Ruano-Rubio, V., DePristo, M. A., Fennell, T., Carneiro, M. O., der Auwera, G. A. Van, Kling, D. E., Gauthier, L. D., Levy-Moonshine, A., Roazen, D., Shakir, K., Thibault, J.,

- Chandran, S., Whelan, C. W., Lek, M., Gabriel, S., Daly, M. J., Neale, B., MacArthur, D. G., & Banks, E. (2017). Scaling accurate genetic variant discovery to tens of thousands of samples. *BioRxiv*. <https://api.semanticscholar.org/CorpusID:89880024>
- Radich, J. P., Dai, H., Mao, M., Oehler, V. G., Schelter, J. M., Druker, B. J., Sawyers, C. L., Shah, N. P., Stock, W., Willman, C., Friend, S. H., & Linsley, P. S. (2006). Gene expression changes associated with progression and response in chronic myeloid leukemia. *Proceedings of the National Academy of Sciences of the United States of America*, 103 8, 2794–2799. <https://api.semanticscholar.org/CorpusID:1154298>
- Rimmer, A., Phan, H., Mathieson, I., Iqbal, Z., Twigg, S. R. F., Wilkie, A. O. M., McVean, G., Lunter, G., & Consortium, W. (2014). Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nature Genetics*, 46(8), 912–918. <https://doi.org/10.1038/ng.3036>
- Sanders, S. J., Ercan-Sencicek, A. G., Hus, V., Luo, R., Murtha, M. T., Moreno-De-Luca, D., Chu, S. H., Moreau, M. P., Gupta, A. R., Thomson, S. A., Mason, C. E., Bilguvar, K., Celestino-Soper, P. B. S., Choi, M., Crawford, E. L., Davis, L. K., Wright, N. R. D., Dhodapkar, R. M., DiCola, M. J., ... State, M. W. (2011). Multiple Recurrent De Novo CNVs, Including Duplications of the 7q11.23 Williams Syndrome Region, Are Strongly Associated with Autism. *Neuron*, 70, 863–885. <https://api.semanticscholar.org/CorpusID:3779933>
- Shaul, O. (2017). How introns enhance gene expression. *The International Journal of Biochemistry & Cell Biology*, 91 Pt B, 145–155. <https://api.semanticscholar.org/CorpusID:3306612>
- Sullivan, K. E. (2019). Chromosome 22q11.2 deletion syndrome and DiGeorge syndrome. *Immunological Reviews*, 287, 186–201. <https://api.semanticscholar.org/CorpusID:56480443>
- Tørresen, O. K., Star, B., Mier, P., Andrade, M., Bateman, A., Jarnot, P., Gruca, A., Grynberg, M., Kajava, A. V., Promponas, V. J., Anisimova, M. O., Jakobsen, K. S., & Linke, D. (2019). Tandem repeats lead to sequence assembly errors and impose multi-level challenges for genome and protein databases. *Nucleic Acids Research*, 47, 10994–11006. <https://api.semanticscholar.org/CorpusID:203661355>
- Wala, J. A., Bandopadhyay, P., Greenwald, N. F., O'Rourke, R., Sharpe, T., Stewart, C., Schumacher, S. E., Li, Y., Weischenfeldt, J., Yao, X., Nusbaum, C., Campbell, P. J., Getz, G., Meyerson, M. L., Zhang, C.-Z., Imieliński, M., & Beroukhim, R. (2018). SvABA: genome-wide detection of structural variants and indels by local assembly. *Genome Research*, 28, 581–591. <https://api.semanticscholar.org/CorpusID:3870389>
- Wang, K., Li, M., Hadley, D., Liu, R., Glessner, J. T., Grant, S. F. A., Hakonarson, H., & Bucan, M. (2007a). PennCNV: an integrated hidden Markov model designed for high-resolution copy

- number variation detection in whole-genome SNP genotyping data. *Genome Research*, 17 11, 1665–1674. <https://api.semanticscholar.org/CorpusID:40294294>
- Wang, K., Li, M., Hadley, D., Liu, R., Glessner, J. T., Grant, S. F. A., Hakonarson, H., & Bucan, M. (2007b). PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Research*, 17 11, 1665–1674. <https://api.semanticscholar.org/CorpusID:40294294>
- Yao, R., Zhang, C., Yu, T., Li, N., Hu, X., Wang, X., Wang, J., & Shen, Y. (2017). Evaluation of three read-depth based CNV detection tools using whole-exome sequencing data. *Molecular Cytogenetics*, 10. <https://api.semanticscholar.org/CorpusID:10052586>
- Ye, K., Schulz, M. H., Long, Q., Apweiler, R., & Ning, Z. (2009). Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, 25, 2865–2871. <https://api.semanticscholar.org/CorpusID:5906713>
- Yella, V. R., & Bansal, M. (2017). DNA structural features of eukaryotic TATA-containing and TATA-less promoters. *FEBS Open Bio*, 7(3), 324–334. <https://doi.org/10.1002/2211-5463.12182>
- Zabidi, M. A., & Stark, A. (2016). Regulatory enhancer-core-promoter communication via transcription factors and cofactors. *Trends in Genetics*, 32(12), 801–814. <https://doi.org/10.1016/j.tig.2016.10.001>

11. Anexos

11.1. Anexo I

Guía de instalación de Conda usando Miniconda en Linux

1. Descarga del instalador de Miniconda

En primer lugar, se debe descargar el archivo de instalación de Miniconda desde el repositorio oficial (<https://repo.anaconda.com/miniconda/>).

- Abrir una terminal.
- Descargar la última versión del instalador de Miniconda para Linux utilizando “wget” o “curl”:

```
wget \
https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

Alternativa con “curl”:

```
curl -O \
https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

2. Ejecutar el archivo de instalación

- Dar permisos de ejecución al archivo descargado:

```
chmod +x Miniconda3-latest-Linux-x86_64.sh
```

- Ejecutar el instalador:

```
./Miniconda3-latest-Linux-x86_64.sh
```

- Seguir las instrucciones en pantalla:
 - Presionar Enter para continuar.
 - Leer y aceptar el acuerdo de licencia.
 - Elegir la ubicación de instalación (la predeterminada es \$HOME/miniconda3).

Si se desea una instalación automática sin interacción del usuario, se puede usar la opción -b (modo batch):

```
./Miniconda3-latest-Linux-x86_64.sh -b -p $HOME/miniconda
```

3. Configurar el entorno

Una vez que Miniconda esté instalada, es necesario configurar el entorno para usar conda desde cualquier terminal.

- Cargar Conda en la sesión actual:

```
eval "$($HOME/miniconda3/bin/conda shell.bash hook)"
```

Si se usó una ubicación de instalación diferente (con -p), se debe ajustar la ruta en el comando anterior.

- Inicializar Conda para que se cargue automáticamente en futuras sesiones de terminal:

```
conda init
```

Esto modificará el archivo de configuración del shell (como .bashrc o .zshrc) para incluir Conda.

4. Verificar la instalación

- Cerrar y abrir la terminal, o recargar la configuración del shell:

```
source ~/.bashrc
```

- Verificar la instalación de Conda:

```
conda --version
```

Se debería ver algo como conda 23.x.x (la versión puede variar).

5. Crear y usar entornos Conda

Ahora que Conda está instalado, es posible comenzar a crear entornos para gestionar proyectos y dependencias de forma aislada.

- Crear un nuevo entorno:

```
| conda create -n mi_entorno (python=X.X)
```

No es obligatorio declarar la versión de Python, pero a veces es recomendable crear el entorno con una versión adecuada para el programa que se vaya a instalar.

- Activar el entorno:

```
| conda activate mi_entorno
```

- Instalar paquetes dentro del entorno activo:

```
| conda install paquete
```

- Desactivar el entorno:

```
| conda deactivate
```

6. (Opcional) Eliminar el instalador

Una vez que Miniconda esté instalada y configurada correctamente, se puede eliminar el archivo de instalación para liberar espacio:

```
| rm Miniconda3-latest-Linux-x86_64.sh
```

11.2. Anexo II

Guía de instalación y ejecución de Freebayes en un entorno Conda

1. Crear un entorno Conda

Primero, se debe crear un entorno Conda específico para FreeBayes llamado, por ejemplo, `freebayes_env`:

```
| conda create -n freebayes_env
```

Después, es necesario activar el entorno recién creado:

```
| conda activate freebayes_env
```

2. Configurar los canales de Conda

Antes de proceder con la instalación de Freebayes, es importante asegurarse de que los canales adecuados estén configurados en Conda. Para ello, se ejecutan los siguientes comandos, los cuales añaden los canales defaults, bioconda y conda-forge a la configuración de Conda:

```
| conda config --add channels defaults  
| conda config --add channels bioconda  
| conda config --add channels conda-forge
```

Estos canales son fundamentales para acceder a paquetes de bioinformática y asegurar la compatibilidad de las herramientas instaladas.

3. Instalar Freebayes

Con el entorno activado, se procede a instalar Freebayes utilizando Conda. El siguiente comando instala Freebayes junto con todas las dependencias requeridas:

```
| conda install freebayes
```

Este método de instalación garantiza que Freebayes esté correctamente integrado con su entorno y que las dependencias se resuelvan automáticamente.

4. Verificar la instalación

Para verificar que FreeBayes se haya instalado correctamente, ejecute:

```
| freebayes --version
```

Esto debería mostrar la versión de FreeBayes que ha sido instalada.

7. Desactivar el entorno

Tras terminar de trabajar con FreeBayes, es posible desactivar el entorno con:

```
| conda deactivate
```

El repositorio de Github contiene más información sobre Freebayes (Garrison, E. (2011). FreeBayes: Bayesian haplotype-based variant detector. Repositorio GitHub. <https://github.com/freebayes/freebayes>)

11.3. Anexo III

Guía de instalación y ejecución de Manta en un entorno Conda

1. Crear un entorno Conda

Primero, se debe crear un entorno Conda dedicado para Manta. Esto ayudará a manejar las dependencias y evitar conflictos con otras herramientas. Para ello, se ejecuta el siguiente comando:

```
| conda create -n manta_env python=2.7
```

En este caso, se utiliza Python 2.7, que es la versión compatible con Manta.

2. Activar el entorno

A continuación, se debe activar el entorno recién creado:

```
| conda activate manta_env
```

3. Instalar Manta

Una vez activado el entorno, se procede a instalar Manta utilizando el canal de Bioconda. Este comando instalará Manta junto con sus dependencias necesarias:

```
| conda install manta=1.6.0=py27_0
```

4. Configurar la carpeta de trabajo

Para continuar, es necesario localizar el directorio donde se ha instalado Manta. Por defecto, el directorio de instalación se encuentra en:

```
| ~/miniconda3/envs/manta_env/share
```

Dentro de este directorio, se debería encontrar una carpeta llamada /manta-1.6.0-0. A continuación, se debe cambiar al directorio de trabajo y empezar a configurar Manta. Para facilitar el proceso, se creará un enlace simbólico al archivo de configuración.

```
cd directorio_de_trabajo  
ln -s \  
~/miniconda3/envs/manta_env/share/manta-1.6.0-0/bin/configManta.py \  
configManta
```

Manta opera a través de un archivo de configuración llamado configManta.py, cuyo propósito es establecer una serie de carpetas en el directorio de trabajo de acuerdo con el comando a ejecutar. En este caso, para realizar un llamado de variantes estructurales (SV) de varias muestras, se debe ejecutar el siguiente comando:

```
./configManta \  
--bam ../muestra1.bam \  
--bam ../muestra2.bam \  
--bam ../muestra3.bam \  
--referenceFasta ../referencia.fa \  
--runDir directorio_analisis_manta
```

El comando es una instrucción para configurar Manta. A continuación, se explican los parámetros usados en el comando:

- **../configManta.py:** Este es el archivo de configuración de Manta. Es el archivo ejecutable que inicializa la configuración para un análisis particular, basándose en los parámetros y opciones proporcionados. Este archivo se ejecuta una sola vez para preparar el entorno y los archivos necesarios para correr Manta.
- **--bam ../muestra1.bam:** Este parámetro especifica la ruta al archivo BAM de una muestra de secuenciación. En este caso, muestra1.bam es el archivo BAM de la primera muestra que se analizará.
- **--bam ../muestra2.bam y --bam ../muestra3.bam (opcionales):** Estos son parámetros adicionales que indican archivos BAM adicionales para incluir en el análisis. Son opcionales y permiten realizar un análisis conjunto de múltiples muestras si se proporciona más de un archivo BAM.
- **--referenceFasta ../referencia.fa:** Este parámetro especifica la ruta al archivo FASTA del genoma de referencia que se utilizará en el análisis. El archivo FASTA contiene las secuencias de ADN del genoma de referencia contra las cuales se alinearon las lecturas en los archivos BAM. Manta utiliza este archivo para mapear y comparar las secuencias alineadas para identificar variantes estructurales.

- **--runDir directorio_analisis_manta:** Este parámetro define el directorio donde se almacenarán los archivos de salida y los archivos intermedios generados por Manta durante su ejecución. Es el directorio de trabajo donde Manta crea su estructura de carpetas y organiza los resultados del análisis. Usar un directorio específico ayuda a mantener los archivos del proyecto organizados y facilita el acceso y la revisión de los resultados.

El directorio de salida debería verse así:

```
configManta results runWorkflow.py runWorkflow.py.config.pickle workspace
```

5. Ejecutar el archivo de trabajo

Una vez configurado, simplemente hay que ejecutar el archivo **runWorkflow.py**:

```
cd directorio_analisis_manta  
./runWorkflow.py
```

Esta es la forma más sencilla de ejecutar Manta. Para obtener más información, se puede consultar la guía de usuario disponible en su página oficial de GitHub: (Illumina. (2016). Manta: Structural variant and indel caller for mapped sequencing data. Repositorio GitHub. <https://github.com/Illumina/manta>)

11.4. Anexo IV

Guía de instalación y ejecución de CNVnator en un entorno Conda

Instalación de CNVnator usando Mamba

A continuación, se presentan los pasos para instalar CNVnator utilizando Mamba, una alternativa eficiente a conda para la gestión de entornos y paquetes.

Paso 1: Instalación de Mamba

1. Instalar Miniconda: Primero, es necesario tener Miniconda instalado. Para ello, se debe descargar e instalar Miniconda utilizando los siguientes comandos:

```
wget \
https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
```

Se deben seguir las instrucciones proporcionadas durante la instalación y, una vez completada, reiniciar la terminal o ejecutar `source ~/.bashrc` para aplicar los cambios.

2. Instalar Mamba: Con Miniconda ya instalado, se puede proceder a instalar Mamba con el siguiente comando:

```
conda install mamba -c conda-forge
```

Paso 2: Crear un entorno con Mamba

1. Crear un nuevo entorno: Se debe crear un entorno dedicado para CNVnator utilizando Mamba:

```
mamba create -n cnvnator-env
```

2. Activar el entorno:

```
conda activate cnvnator-env
```

Paso 3: Instalación de CNVnator

1. Configurar los canales: Como CNVnator puede no estar disponible en los canales predeterminados, se debe agregar el canal bioconda y conda-forge para facilitar la instalación:

```
mamba config --add channels bioconda  
mamba config --add channels conda-forge
```

2. Instalar CNVnator: Intentar instalar CNVnator directamente desde el canal bioconda con el siguiente comando:

```
mamba install cnvnator
```

En caso de que CNVnator no esté disponible en el canal "bioconda", es posible que se deba instalar desde la fuente. Los pasos para esta instalación se detallan más a fondo en el github oficial de CNVnator (<https://github.com/abyzovlab/CNVnator>).

Ejecución de CNVnator

Una vez instalado, la búsqueda de variantes a través de CNVnator requiere la creación de un archivo raíz con información acerca del fichero BAM de alineamiento que se va a utilizar como muestra. Los pasos están detallados a continuación:

1. Generar el archivo de árbol de eventos (-tree):

```
cnvnator -root ejemplo.root -tree ejemplo.bam
```

- Propósito: Este comando toma el archivo BAM (ejemplo.bam) y lo convierte en un formato de árbol de eventos dentro del archivo raíz (ejemplo.root). Un archivo raíz es un formato de archivo utilizado por CNVnator para manejar grandes volúmenes de datos de cobertura genómica de manera estructurada.
- Qué sucede: CNVnator lee el archivo BAM, que contiene los alineamientos de las secuencias al genoma de referencia, y organiza estos datos en una estructura de árbol dentro del archivo raíz. Esta estructura facilita el acceso rápido y eficiente a los datos de cobertura.

2. Crear el histograma de cobertura (-his):

```
cnvnator -root ejemplo.root -his tamaño_bin \  
-d /ruta/al/genoma/referencia/
```

- Propósito: Este paso calcula la profundidad de cobertura en el genoma dividido en "bins" (ventanas) de un tamaño específico, definido por tamaño_bin (por ejemplo, 100 bases).
- Qué sucede: CNVnator usa los datos almacenados en el archivo ROOT para calcular cuántas lecturas se alinean en cada bin del tamaño especificado por tamaño_bin. El argumento -d especifica el directorio donde se encuentran los archivos FASTA descomprimidos del genoma de referencia. Esto es necesario para mapear las lecturas correctamente a las regiones correspondientes del genoma.
- Importancia: Crear histogramas de cobertura es crucial para detectar variaciones en la cobertura que podrían indicar CNV. Los bins con cobertura significativamente mayor o menor que el promedio sugieren duplicaciones o deleciones, respectivamente.

3. Calcular estadísticas de cobertura (-stat):

```
cnvnator -root ejemplo.root -stat tamaño_bin
```

- Propósito: Este paso calcula estadísticas de la distribución de la profundidad de cobertura en los bins de tamaño_bin bases.
- Qué sucede: CNVnator calcula la media y la desviación estándar de la cobertura en los bins. Estos valores se utilizan para normalizar la cobertura en los siguientes pasos, permitiendo detectar desviaciones que sugieran CNV.
- Importancia: Sin este paso, no podríamos interpretar correctamente si la cobertura en una región específica es anormal o no.

4. Particionar el genoma para detección de CNVs (-partition):

```
cnvnator -root ejemplo.root -partition tamaño_bin
```

- Propósito: Este comando segmenta el genoma en regiones contiguas de cobertura similar, basándose en los histogramas y estadísticas calculados en los pasos anteriores.

- Qué sucede: CNVnator utiliza un algoritmo de segmentación para agrupar bins contiguos que tienen una cobertura similar. Esto ayuda a definir límites claros de las regiones con diferentes números de copias, lo que facilita la identificación de CNV.
- Importancia: Este paso es esencial para definir las regiones específicas donde hay duplicaciones o deleciones. Ayuda a reducir el ruido y a concentrarse en regiones que realmente presentan una variación en el número de copias.

Resumen de los pasos:

1. Primero, los datos del BAM se transforman en un formato ROOT manejable.
2. Luego, se calcula la cobertura en ventanas de un tamaño específico para detectar variaciones.
3. Las estadísticas de cobertura ayudan a normalizar y detectar desviaciones significativas.
4. Finalmente, el genoma se segmenta en regiones de cobertura uniforme para identificar CNV claramente.

Estos pasos deben ser ejecutados en orden, ya que cada uno depende del anterior para proporcionar la información necesaria para la detección precisa de CNV. Después de completar estos pasos, se puede proceder a la llamada de CNV con el siguiente comando:

```
cnvnator -root ejemplo.root -call tamaño_bin > llamado_variantes.calls
```

Este comando utilizará la información generada para identificar y listar las variantes en el número de copias a lo largo del genoma.

Lamentablemente, el llamado de variantes que hace CNVnator está en un formato ilegible para la mayoría de las herramientas de filtrado o visualización. Sin embargo, CNVnator incluye un conversor de su formato particular a VCF, y se ejecuta con el siguiente comando:

```
cnvnator2VCF.pl -prefix estudio1 -reference GRCh37  
llamado.variantes.calls /ruta/a/archivos/fasta/individuales
```

El repositorio de Github de CNVnator recoge más información: (Abyzov Lab. (2011). CNVnator: A tool for CNV detection from depth-of-coverage by mapped reads. Repositorio GitHub. <https://github.com/abyzovlab/CNVnator>)

11.5. Anexo V

Guía de instalación de VCFtools en Linux

VCFtools es una colección de programas diseñados para trabajar con archivos VCF. A continuación, se describen los pasos para instalar VCFtools en un sistema Linux, siguiendo las instrucciones del [repositorio oficial en GitHub](#).

1. Preparativos para la instalación

Antes de comenzar, se debe asegurar de que el sistema tenga instaladas las herramientas necesarias para compilar el código. En la mayoría de los sistemas Linux, estas herramientas ya están presentes. De no ser así, se pueden instalar utilizando los siguientes comandos:

```
sudo apt-get update  
sudo apt-get install build-essential
```

Nota: Si no se tienen permisos de sudo, es recomendable consultar con el administrador del sistema para instalar estas herramientas o utilizar un entorno Conda para manejar las dependencias.

2. Clonar el repositorio de VCFtools

Se debe clonar el repositorio oficial de VCFtools desde GitHub utilizando git:

```
git clone https://github.com/vcftools/vcftools.git
```

Después, se debe cambiar al directorio del repositorio clonado:

```
cd vcftools
```

3. Compilar e instalar VCFtools

Se debe ejecutar el archivo autogen.sh para preparar el entorno de compilación:

```
./autogen.sh
```


Luego, se configura la instalación especificando el directorio de instalación (opcional). Si no se especifica un directorio, VCFtools se instalará en /usr/local por defecto:

```
| ./configure
```

(Opcional) Para instalar en un directorio personalizado (por ejemplo, \$HOME/vcftools):

```
| ./configure --prefix=$HOME/vcftools
```

A continuación, se compila el código con el siguiente comando:

```
| make
```

Finalmente, se instala VCFtools:

```
| make install
```

Si se utilizó un directorio personalizado en el paso de configuración, se debe añadir la ruta a la variable PATH para acceder a VCFtools fácilmente:

```
| export PATH=$HOME/vcftools/bin:$PATH
```

Esta línea se puede añadir al archivo .bashrc o .zshrc para que la ruta se cargue automáticamente en cada sesión.

4. Verificar la instalación

Para verificar que VCFtools se haya instalado correctamente, se debe comprobar su versión ejecutando:

```
| vcftools --version
```

Debería aparecer un mensaje indicando la versión de VCFtools instalada.

5. Uso de VCFtools

Una vez instalado VCFtools, se puede comenzar a utilizarlo para trabajar con archivos VCF. A continuación, se presentan algunos ejemplos básicos:

- Filtrar variantes en un archivo VCF (por ejemplo, mantener solo las variantes con una calidad mayor a 30):

```
| vcfutils --vcf input.vcf --minQ 30 --recode --out output_filtered
```

- Contar el número de variantes en un archivo VCF:

```
| vcfutils --vcf input.vcf --out count_output --counts
```

- Extraer un subconjunto de muestras de un archivo VCF:

```
| vcfutils --vcf input.vcf --keep sample_list.txt --recode --out  
subset_output
```

6. (Opcional) Eliminar archivos de instalación

Para limpiar los archivos de compilación y liberar espacio, se puede ejecutar el siguiente comando:

```
| make clean
```

Guía de instalación de RTG Tools en Linux

RTG Tools es un conjunto de herramientas utilizadas para el análisis de variantes genéticas, como la comparación de archivos VCF y la validación de variantes. A continuación, se presenta una guía paso a paso para instalar RTG Tools en un sistema Linux.

1. Descargar RTG Tools

Para comenzar, se debe abrir una terminal y dirigirse al sitio web de RTG Tools para descargar la última versión disponible. El archivo generalmente está disponible en formato .tar.gz. Es importante obtener el enlace de descarga correcto desde el sitio web oficial de RTG Tools.

Para descargar el archivo, se puede utilizar wget o curl:

Usando wget:

```
wget https://www.rtg-genomics.com/download/rtg-tools-latest-linux-x86_64.tar.gz
```

Usando curl:

```
curl -O https://www.rtg-genomics.com/download/rtg-tools-latest-linux-x86_64.tar.gz
```

2. Descomprimir el archivo descargado

Una vez descargado el archivo, se debe descomprimir utilizando el siguiente comando:

```
tar -xvzf rtg-tools-latest-linux-x86_64.tar.gz
```

Después, se debe cambiar al directorio del archivo descomprimido:

```
cd rtg-tools-latest-linux-x86_64
```

3. Configurar el entorno

Para poder utilizar RTG Tools, es necesario añadir el directorio de binarios al PATH del sistema.

Se puede añadir RTG Tools al PATH temporalmente en la sesión actual con el siguiente comando:

```
| export PATH=$PATH:$(pwd)/bin
```

Para hacer este cambio permanente, se debe añadir la línea anterior al final del archivo de configuración del shell (.bashrc, .zshrc, etc.):

```
| echo 'export PATH=$PATH:/ruta/a/rtg-tools/bin' >> ~/.bashrc
```

A continuación, se recarga el archivo de configuración del shell para aplicar los cambios:

```
| source ~/.bashrc
```

Nota: Es necesario reemplazar /ruta/a/rtg-tools con la ruta real donde se descomprimió RTG Tools.

4. Verificar la instalación

Para verificar que RTG Tools se haya instalado correctamente, se debe comprobar su versión con el siguiente comando:

```
| rtg --version
```

Se debería ver un mensaje que indique la versión de RTG Tools instalada.

5. Usar RTG Tools

Una vez instalado, RTG Tools está listo para ser utilizado en análisis genéticos. A continuación, se presentan algunos ejemplos básicos:

Comparar archivos VCF:

```
| rtg vcfeval -b archivo_referencia.vcf -c archivo_comparar.vcf
```

Convertir archivos VCF:

```
rtg vcftools --vcf archivo_entrada.vcf --out archivo_salida
```

Sin embargo, el principal objetivo de RTG Tools en este proyecto es obtener un archivo con las estadísticas del fichero VCF. Para ello, se debe ejecutar el siguiente comando:

```
rtg vcfstats ejemplo.vcf > estadísticas_ejemplo.txt
```

Explicación del comando:

- `rtg`: Llama a la herramienta RTG.
- `vcfstats`: Especifica la herramienta dependiente para generar estadísticas de archivos VCF.
- `estadísticas-ejemplo.vcf`: Es el archivo VCF sobre el que se obtienen estadísticas.

Interpretar la salida:

La salida del comando proporcionará diversas estadísticas sobre el archivo VCF. Estas pueden incluir:

- Número total de variantes: Cuántas variantes (SNPs, indels, etc.) están presentes en el archivo.
- Distribución de las variantes por tipo: La cantidad de variantes de cada tipo (SNP, indel, etc.).
- Calidad de las variantes: Estadísticas relacionadas con la calidad de las variantes, como la puntuación de calidad media.
- Información sobre las muestras: Estadísticas sobre las muestras en el archivo, si es un VCF multiclinico.

Ejemplo de salida (simplificado):

```
Number of variants: 1000  
Number of SNPs: 800  
Number of indels: 200  
Number of samples: 10  
Mean quality score: 50
```

El comando `rtg vcfstats` es una herramienta útil para obtener estadísticas detalladas sobre archivos VCF. Permite revisar la calidad y el contenido de tus datos genéticos de manera eficiente.

Guía de instalación de BCFTools en un entorno Conda

Paso 1 (Opcional): Crear un Entorno Conda para BCFTools

Se sugiere crear un entorno específico para BCFTools para mantener las dependencias organizadas y evitar conflictos con otros paquetes. El siguiente comando crea un entorno llamado `bcftools_env`:

```
| conda create -n bcftools_env
```

Una vez creado, el entorno debe activarse para que las instalaciones de paquetes y las ejecuciones se realicen dentro de él:

```
| conda activate bcftools_env
```

Paso 2: Instalar BCFTools

Conda facilita la instalación de BCFTools utilizando canales que contienen el paquete. El canal `bioconda` es un repositorio especializado en programa de bioinformática. Se puede instalar BCFTools ejecutando el siguiente comando:

```
| conda install -c bioconda bcftools
```

- `-c bioconda` especifica el canal de donde se obtendrá el paquete.
- `bcftools` es el nombre del paquete que se desea instalar.

Este comando instalará BCFTools junto con todas las dependencias necesarias automáticamente.

Verificar la Instalación: Después de la instalación, se puede verificar que BCFTools se haya instalado correctamente ejecutando:

```
| bcftools --version
```

Este comando debería mostrar la versión de BCFTools instalada, confirmando que el proceso de instalación se ha realizado correctamente.