Desarrollo de un protocolo de análisis de datos de NGS y comparación de algoritmos de detección de variantes.

Miguel Barquín del Romo
Máster Bioinformática y Bioestadística
Bioinformática Clínica

Joan Maynou Fernández

Enero 2017

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Descripción del trabajo*
Nombre del autor:	*Miguel Barquin del Romo*
Nombre del consultor/a:	*Joan Maynou Fernández*
Fecha de entrega (mm/aaaa):	01/2018
Titulación::	*Máster Bioinformática y Bioestadística*
Área del Trabajo Final:	*Bioinformática Clínica*
Idioma del trabajo:	*Castellano*
Palabras clave	*Pipeline, NGS, SNP.*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

La implementación de algoritmos de trabajo es fundamental en el ámbito hospitalario, debido a la creciente demanda y utilización de las tecnologías de secuenciación masiva. Uno de estos casos es el estudio de los genes BRCA 1 y BRCA2, ambos implicados en el cáncer de mama y de ovario entre otros.

La finalidad de este trabajo es la obtención de un script con el cual se puedan analizar datos clínicos obtenidos a través de secuenciación masiva, así como comparar determinados algoritmos usados para estudiar su rendimiento.

Como resultado se ha obtenido un script completo con el que poder analizar dichas muestras, obteniendo un resultado claro de las variantes encontradas, así como su implicación clínica.

También se han podido comparar distintos algoritmos de detección de variantes, resaltando la gran variabilidad entre ellos y la creciente necesidad de conocer y validar las diferentes opciones que se pueden encontrar.

Abstract (in English, 250 words or less):

The implementation of work algorithms is fundamental in the hospital environment, due to its growing demand and the use of mass sequencing technologies. An example of these cases is the study of the BRCA 1 and BRCA2 genes, which are involved in breast and ovarian cancer, among others.

The purpose of this work is to obtain a sequence of commands which would aid the data analysis through massive sequencing, as well as compare algorithms used to study their performance.

As a result, a complete script has been developed, which allows the reading of the samples to obtain a clear result of the parameters found, as well as their clinical implications.

It was also possible to compare different algorithms for variant detection, highlighting the great variability between them and the rising need to contemplate the different options that could be found.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

En el año 2004 se comenzó a desarrollar y comercializar nuevas tecnologías de secuenciación que redujeran el tiempo, los recursos y los costes necesarios para llevar a cabo la secuenciación del genoma humano (1). Dichos avances han progresado hasta obtener una tecnología de secuenciación denominada en inglés “Next generation Sequencing” o NGS, la cual nos permite obtener una gran cantidad de secuencias en un único proceso.

El funcionamiento de la tecnología es el siguiente: el ADN (Ácido desoxiribonucleico) se fragmenta en trozos de secuencia más pequeños, denominados amplicones, en los cuales se van a ir incorporando una serie de secuencias que actúan como adaptadores. Dichos fragmentos de ADN se amplifican, y forman un grupo o clustering, actuando como molde o template en el proceso de secuenciación. En el extremo de los adaptadores se van a ir incorporando dNTPs (desoxinucleósidos trifosfato) marcados con fluorescencia, los cuales van a poder ser detectados a través de captación de imagen (2–5).

El uso de tecnologías de amplificación de genes seleccionados (targeted genes en inglés) permite secuenciar regiones específicas del genoma, pudiendo centrar la amplificación en aquellos genes en los cuales tenemos un interés específico en analizar, debido a que existen evidencias científicas de que variaciones en su secuencia puedan ser causantes de alguna patología, o puede ser una diana para uso terapéutico (6).

Las distintas variaciones genéticas que pueden contener una secuencia de ADN se dividen en tres categorías: polimorfismo de nucleótido simple, conocido como SNP (acrónimo del inglés Single Nucleotid Polymorphism); inserciones o deleciones de pequeños fragmentos de secuencia, conocidos como Indels (acrónimo del inglés insertions and deletions), y variaciones en el número de copias, conocidos como CNV (acrónimo del inglés Copy Number Variation) (7).

Tal es el caso del cáncer de mama y del cáncer de ovario. En ambas enfermedades, las mutaciones en los genes BRCA1 y BRCA2 han sido las únicos que presentan alta significación para ambos tipos de cánceres (8).

Existe además el riesgo de que dichas mutaciones sean de tipo hereditarias, lo que implica que su presencia puede encontrarse en otros miembros de la familia (9). El riesgo de malignidad en individuos con mutaciones germinales en BRCA1 puede aumentar hasta un 87% para el cáncer de mama, y un 63% para el de ovario; el riesgo en mutaciones en BRCA2 aumenta hasta un 84% para el cáncer de mama, y un 27% para el de ovario (10).

Es por ello, por lo que el estudio de la secuencia de ambos genes a través de las nuevas técnicas de secuenciación masiva se ha convertido en una herramienta indispensable en la práctica clínica (11–13).

1.1.1 Descripción general

El objetivo de este Trabajo de Fin de Máster (TFM) es realizar un protocolo de análisis de datos obtenidos por técnicas de secuenciación masiva (en inglés NGS), así como una comparativa del comportamiento de diferentes algoritmos involucrados en el desarrollo de dicho protocolo.

Para llevar a cabo este objetivo, se va a realizar una búsqueda y una posterior aplicación de diferentes algoritmos bioinformáticos usados en el proceso de búsqueda de variantes (SNP e Indels), así como su posterior anotación. Una vez obtenido un resultado para cada algoritmo, se llevará a cabo una comparación de éstos.

1.1.2 Justificación del TFM

Existen diferentes tipos de variaciones genómicas: SNPs, mutaciones puntuales o variaciones en un solo nucleótido; indels (pequeñas inserciones y deleciones), CNV (deleciones, inserciones o duplicaciones de fragmentos de ADN con un tamaño entre 1Kb y 5 Mb) y SV (grandes deleciones, inserciones o duplicaciones, mayores de 5Mb) (7).

Dichas variaciones se han encontrado relacionadas en diferentes casos: descrito como causantes de la aparición de tumores (14), así como importantes dianas terapéuticas para el uso de medicamentos (15). Es por ello que su correcta identificación se ha convertido en un punto importante para el diagnóstico y tratamiento del paciente.

Existe una gran variedad de algoritmos bioinformáticos para su identificación (16,17), y la combinación de estos, junto con otros algoritmos involucrados en otros procesos, como pre-procesado o alineamiento, en diferentes pasos del pipeline bioinformático, puede ocasionar la obtención de diferentes resultados e interpretaciones del mismo individuo.

Es por ello por lo que el desarrollo de un protocolo de análisis de datos, y una comparación de los distintos algoritmos utilizados, se haya convertido en un paso importante para obtener los resultados robustos.

1.2 Objetivos del Trabajo

Objetivo1. Implantar un protocolo de análisis de datos obtenidos por NGS en entornos clínicos.

Objetivo2. Realizar una comparativa del comportamiento de diferentes algoritmos para detectar variantes (SNP/Indels).

1.3 Enfoque y método seguido

Para la realización del protocolo de análisis, se va a desarrollar un script que contenga el código para llevar a cabo el análisis. El script se escribió en el editor de texto emacs, en un ordenador con un sistema operativo Ubuntu 16.04.3.

El flujo de trabajo en el que va a consistir el pipeline se resume en la siguiente imagen:

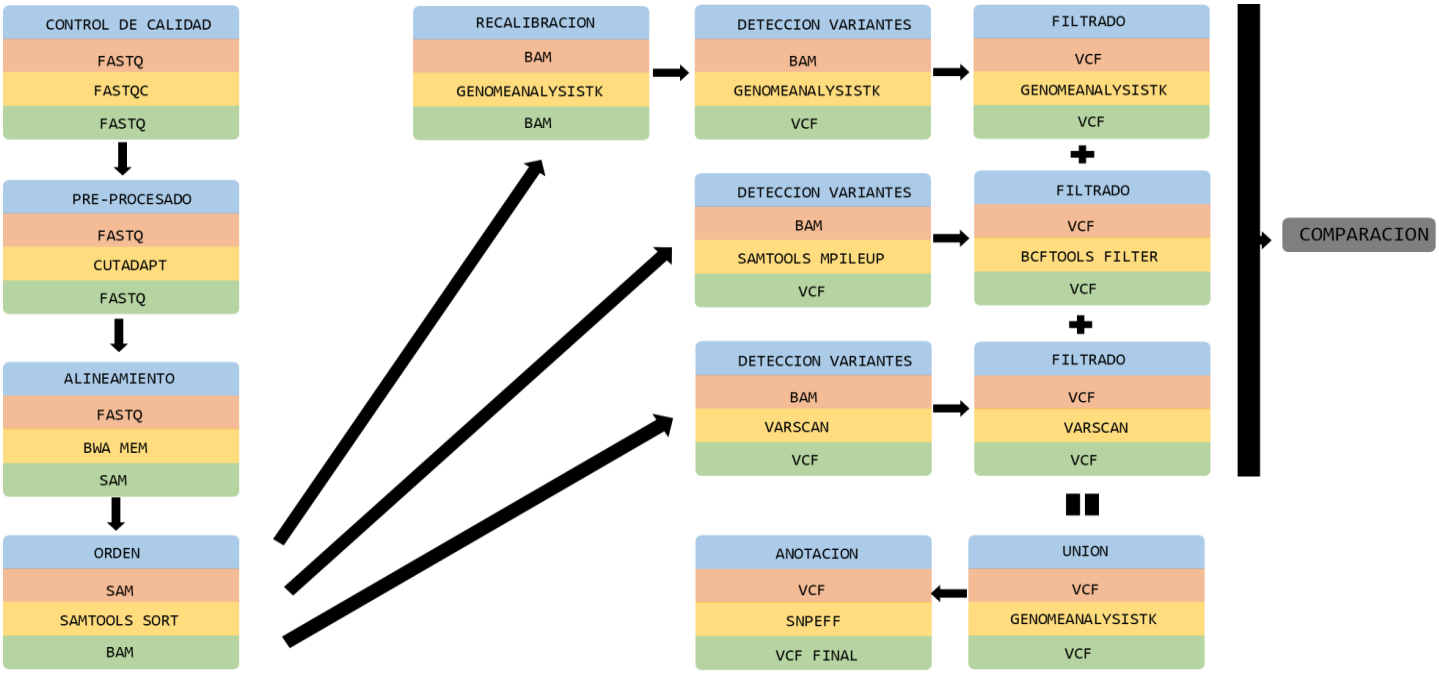

Figura1: esquema de trabajo. En la casilla azul viene el nombre del proceso, en marrón el nombre del archivo con el que se va a trabajar, en amarillo el programa que se utiliza, y en verde el tipo de archivo que se obtiene después de dicho proceso.

Pese a existir una gran cantidad de algoritmos con los que poder realizar el proceso de análisis, sólo se eligió uno para cada proceso, exceptuando la detección de variantes que se eligieron tres algoritmos diferentes para poder llevar a cabo una mejor optimización del proceso, y una comparación entre ellos.

1.4 Planificación del Trabajo

Las tareas que se van a llevar a cabo son:

1. Selección de las muestras que van a ser analizadas.
2. Búsqueda bibliográfica de los distintos algoritmos que se usan durante todo el proceso.
3. Creación e implementación del algoritmo. Obtención de las variantes encontradas en las muestras.
4. Comparación de los resultados.

1.5 Breve sumario de productos obtenidos

- Obtención de un script completo, por el cual se pueden analizar los datos obtenidos por procesos de secuenciación masiva, obteniendo como resultado final de dicho script la clasificación de las variantes encontradas.
- Archivo por cada muestra que contiene toda sus variantes detectadas y anotadas.
- Comparación de los algoritmos usados para la detección de variantes.

1.6 Breve descripción de los otros capítulos de la memoria

- Capítulo 2: Material y métodos. Este capítulo viene dividido en tres partes: en la primera parte se introduce el tipo de muestras usadas; en la segunda se explica el pipeline creado para llevar a cabo el análisis; y en la tercera se explica el proceso para llevar a cabo la comparación.
- Capítulo 3: Resultados y discusión. Se abordan los resultados obtenidos, y se argumenta la posible causa de los mismos.
- Capítulo 4: Conclusiones. Se explican las diferentes conclusiones obtenidas de la realización del trabajo.
- Capítulo 5: Glosario: Se definen las palabras y términos más relevantes.
- Capítulo 6: Bibliografía: Relación de la bibliografía utilizada para la elaboración del trabajo.

2. Material y métodos

2.1 Muestras

Los datos a analizar son de sujetos mujeres con cáncer de mama y/o ovario, que entran en el estudio de las variantes germinales de los genes BRCA1 y BRCA2. La interpretación biológica de las posibles variables encontradas, así como los posibles informes médicos que derivan de las mismas no son el objetivo de este trabajo, por lo que no aparecerán en esta memoria.
El ADN se ha obtenido de muestras de sangre periférica, realizando una extracción de ADN genómico de las células mononucleares, a través del kit Maxwell® 16 Blood DNA Purification Kit. La concentración obtenida se midió usando Quantus™ Fluorometer, mientras que la pureza de las muestras se midió con NanoDrop 200c espectofotómetro. La creación de las librerías se realizó con el kit BRCA MASTR™ Dx de Multiplicom, siguiendo el workflow establecido por la casa comercial. La secuenciación de las librerías se realizó en Illumina Miseq.

2.2 Pipeline

1. Control de calidad

Los archivos con los que se va a empezar a trabajar son los datos brutos obtenidos directamente del secuenciador. Dichos datos se encuentran en archivos FASTQ, un formato desarrollado por Jim Mullikin (18).

En dicho formato aparecen una serie de características y valores (18):

- En la primera línea aparece el símbolo '@' seguido de un identificador para dicho archivo.
- En la segunda aparece la secuencia de bases nitrogenadas en formato FASTA (A, G, C, T).
- En la tercera línea aparece el símbolo '+' para indicar el final de la secuencia anterior.
- En la cuarta línea aparecen los datos de calidad para cada una de las bases obtenidas en la línea 2.

Para cada sujeto se obtienen dos archivos FASTQ: uno para la lectura forward, y otro para la lectura reverse de la secuencia.

El primer paso que se debe llevar a cabo es un control de calidad de los archivos FASTQ. Gracias a este paso, realizado a través de la herramienta FastQC, se va a obtener una serie de parámetros y valores, a través de los cuáles se van a poder determinar si los archivos cumplen o no unos parámetros mínimos de calidad para poder continuar efectuando el análisis. Puesto que se va a realizar con posterioridad un pre-procesado de los archivos, la calidad de las muestras que se observe no tiene por qué ser óptima, ya que este paso la mejorará sensiblemente.

De entre todas las salidas que ofrece Fastqc, sólo se va a tener en cuenta el parámetro denominado “Per base sequence quality”. Se trata de un gráfico de barras en el cual aparece un rango de valores de calidad para cada base en cada posición del archivo. En el eje de las abscisas se encuentran las bases leídas, en el de las ordenadas la calidad de cada base.

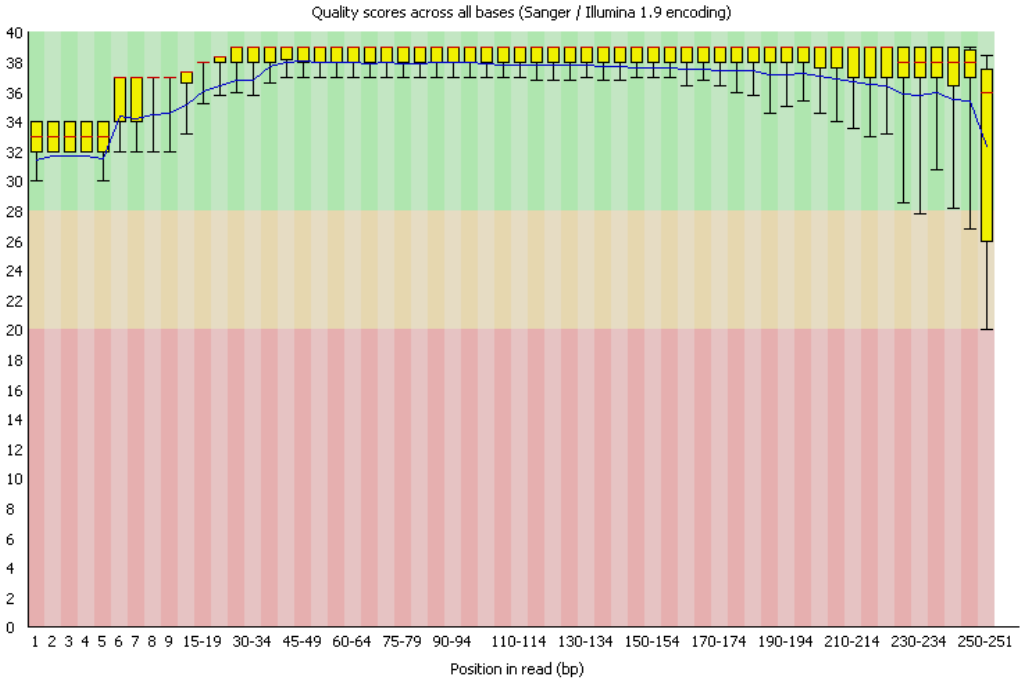

Figura 2: Ejemplo de gráfica obtenida: se observa que la calidad de la lectura es buena, decayendo en las últimas bases debido a que la química utilizada se va degradando, afectando a las últimas bases en secuencias largas.

2. Pre-procesado

Uno de los grandes problemas que tiene la secuenciación masiva es que la tasa de error aumenta con respecto a las tecnologías utilizadas con anterioridad, como la tecnología Sanger, lo que implica la necesidad de realizar un pre-procesado con el que mejorar las secuencias leídas (19–21). En este sentido, existen diferentes de herramientas con las que poder realizar el preprocesado de las muestras: Trimmomatic, PRINSEQ o Cutadapt (22), siendo esta ultima la que se va a utilizar en este pipeline. Cutadapt está escrito principalmente en Python, y desarrollado en Ubuntu Linux (23).

El código que se va a utilizar es el siguiente:

```
cutadapt -q 20,20 --pair-filter=any -m 100 --
output=out.MUESTRA.1.fastq --paired-
output=out.MUESTRA.2.fastq NOMBREFASTQREAD1.fastq
NOMBREFASTQREAD2.fastq
```

- -q 20,20: se establece la mínima calidad de las terminaciones de las lecturas, descartándose las que presenten menor calidad. Se decidió establecer un mínimo de 20 en ambos extremos.
- --pair-filter=any: el par de lecturas se descarta si una de las lecturas no cumple con los parámetros de calidad que se establecen.
- -m 100: se descartan aquellas lecturas que tengan una longitud menor al número establecido. Se decidió establecer una longitud mínima de 100 pares de bases.
- --output=out.MUESTRA.1.fastq --paired-output=out.MUESTRA.2.fastq: son los archivos obtenidos, se crean dos: uno para cada archivo fastq, es decir, uno para cada dirección de lectura. Al igual que los archivos iniciales, se encuentran en el mismo formato.
- NOMBREFASTQREAD1.fastq NOMBREFASTQREAD2.fastq: son los archivos fastq originales que se están analizando, se tienen que introducir los dos archivos obtenidos de la misma muestra.

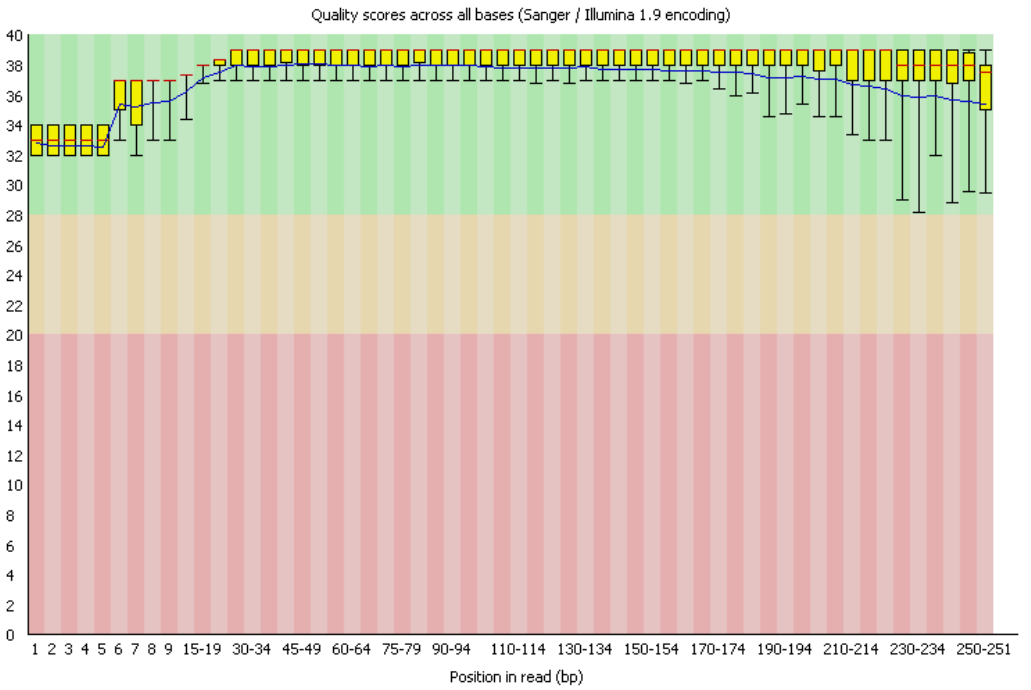

Figura 3: siguiendo con el ejemplo de la imagen 1, se ha subido el gráfico “Per base sequence quality” del fastq obtenido después del pre-procesado. Se observa cómo se mejora la calidad de las lecturas al final de la secuencia.

3. Alineamiento

Una vez que se han obtenido los fastq con unas calidades de lecturas mejores que las obtenidas inicialmente, se va a llevar a cabo el proceso de alineamiento con estos nuevos archivos.

Este proceso consiste en mapear las lecturas obtenidas frente a un genoma conocido, presentando alta exactitud (24).

Existe una serie de algoritmos para llevar a cabo el alineamiento de las lecturas. Entre las diferentes opciones que se pueden encontrar aquellos que están basados en el algoritmo Burrows–Wheeler Transformation (BWT), como son Bowtie y BWA; y aquellos que se basan en el algoritmo Smith–Waterman (SW), como son MOSAIK, SHRiMP2, y Novoalign (22).

En este trabajo se va a utilizar el algoritmo bwa-mem, que se ha evidenciado que presenta mayor precisión que otros (25), estando además indicado para amplicones con un tamaño de entre 70 y 1000 pares de bases (26).

Antes de llevar a cabo el alineamiento, se va a establecer y preparar el genoma de referencia con el que se van a alinear las secuencias utilizadas en este trabajo. Como sólo se han secuenciados dos genes (el gen BRCA1 y BRCA2), basta con tener la secuencia de ambos para realizar el alineamiento.

Accediendo al navegador (genome browser en inglés) Ensembl (27), se puede obtener en formato fasta las regiones del genoma que más nos interesan. Se va a acceder a la versión GRCH37, para obtener el genoma de referencia en esta versión. Para obtener la secuencia de BRCA1, se especifica el cromosoma 17, y las coordenadas 411959912 y 41277787, obteniendo el archivo BRCA1.fasta. Para BRCA2 se especifica el cromosoma 13, y las coordenadas 32887600 y 32975850, para obtener el archivo BRCA2.fasta. Se tienen que unir para tener ambas secuencias en el mismo archivo (se puede usar, por ejemplo el comando `cat * .fasta > BRCA.fasta`).

Una vez obtenido el archivo fasta con la secuencia referencia, se tiene que llevar a cabo el indexado para obtener distintos archivos que van a ser utilizados por los diferentes programas a lo largo de todo el proceso.

Con este comando se crean cinco archivos (.amb, .ann, .bwt, .pac, .sa).

```
bwa index BRCA.fasta
```

Con el siguiente se obtiene un archivo .dict que contiene los tamaños y nombres de la referencia.

```
java -jar picard.jar CreateSequenceDictionary REFERENCE=
BRCA.fasta OUTPUT=BRCA.dict
```

Con este comando se crea el archivo .fai de la referencia, con el cual se podrá acceder a esta.

```
samtools faidx BRCA.fasta
```

Una vez que se ha obtenido la referencia y sus correspondientes archivos, se puede llevar a cabo el alineamiento.

```
bwa mem -R '@RG\tID:'"$i"'\tLB:'"$i"'\tSM:'"$i"'\tPL:ILLUMINA' BRCA.fasta out.MUESTRA.1.fastq out.MUESTRA.2.fastq > aln.MUESTRA.sam
```

- -R: La identificación del grupo vendrá anclada en cada lectura del archivo de salida.
- '@RG\tID:'"$i"'\tLB:'"$i"'\tSM:'"$i"'\tPL:ILLUMINA': Identificación del grupo. Hay que cambiar el símbolo $i por el nombre de la muestra.
- BRCA.fasta: se indica el genoma de referencia.
- out.MUESTRA.1.fastq out.MUESTRA.2.fastq: se alinean los dos fastq obtenidos en la parte de pre-procesado. A partir de este punto, en vez de trabajar con dos archivos por cada muestra, se trabaja con uno.
- aln.MUESTRA.sam: el archivo obtenido es un archivo en formato SAM.

Los archivos SAM están en un formato separado por tabulaciones, con un formato de texto line oriented. Está formado por un encabezado, donde aparecen algunos datos, y una sección de alineamiento, donde cada línea contiene información de cada alineamiento (28).

4. Sort e indexado

La ordenación de las lecturas (sort en inglés) no es un proceso obligatorio, pero si útil para agilizar el proceso de datos y evitar cargar alineamientos extras (29) para su realización se va a utilizar la herramienta samtools (29).

```
samtools sort aln.MUESTRA.sam > aln.MUESTRA.bam
```

```
samtools index aln.MUESTRA.bam
```

Con estos comandos se obtienen dos archivos: un archivo en formato BAM, se trata de un archivo binario que contiene de forma comprimida la información que tiene el archivo SAM (28); y un archivo bai, se trata de un archivo con la información indexada del archivo bam, contiene datos que resumen el número de lecturas por secuencia de frecuencia (30). Con estos datos no se tiene que llevar a cabo el proceso de eliminación de duplicados. Se trata de una carrera de amplicones,

Una vez que se ha obtenido el BAM, se puede llevar a cabo el proceso de detección de variantes y el de filtrado. Se van a utilizar tres herramientas diferentes para llevar a cabo este proceso: GATK HaplotypeCaller, Samtools mpileup y Varscan.

5. GATK HaplotypeCaller

GATK (31) corresponde a las iniciales de Genome Analysis Tool Kit. Se van a seguir las recomendaciones que la propia página web de GATK aconseja continuar.

- Recalibración de lecturas

La recalibración de la calidad de las bases (en inglés Base Quality Score Recalibration [BQSR]) es un método por el cual se ajustan los valores de calidad de las bases de las lecturas a través de técnicas de machine learning. Esto se debe a que dichos valores, que vienen determinados por los secuenciadores, pueden estar erróneamente calculados a causa de diferentes problemas de la química de la secuenciación, defectos de fábrica del equipo, u otros. Estos errores influyen en el proceso de detección de variantes, ya que dichos algoritmos utilizan estos datos para mostrar la veracidad de dicha detección. Este proceso comprende dos fases: en una primera fase se calculan las variaciones teniendo en cuenta nuestros datos y una lista de variantes conocidas, en una segunda parte se ajustan los valores a nuestros datos (32).

En primer lugar, se tiene que obtener e indexar el archivo .vcf donde se encuentran tanto los SNP como los Indels conocidos para llevar a cabo la recalibración. Tiene que estar realizado con el mismo genoma de referencia con el que se ha estado trabajando a lo largo de todo el proceso (en este caso, el GRCh37). De entre los distintos archivos que existen, en este trabajo se ha optado por vcf, de la página web del NCBI (33).

Una vez obtenido, se lleva a cabo la indexación del archivo. Se va a obtener un archivo tbi.

```
tabix -p vcf All_20161121.vcf.gz
```

- -p vcf: indica el tipo de archivo de entrada que hay que indexar. En nuestro caso un archivo vcf.
- All_20161121.vcf.gz: es el archivo que se va a indexar.

Primera parte del código, donde se va a obtener un archivo en formato table donde están calculadas las variaciones:

```
java -jar GenomeAnalysisTK.jar \
     -T BaseRecalibrator \
     -R BRCA.fasta \
     -I aln.MUESTRA.bam \
     -knownSites All_20161121.vcf.gz \
     -o recal_data_aln_MUESTRA.table
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T BaseRecalibrator: Orden específica. En este caso, BaseRecalibrator para la calibración de bases.
- -R BRCA.fasta: se indica el genoma de referencia.
- -I aln.MUESTRA.bam: el archivo de entrada, en este caso el bam en el que va a trabajar.
- -knownSites All_20161121.vcf.gz: el archivo donde aparecen los SNP e Indels conocidos.
- -o recal_data_aln_MUESTRA.table: el archivo de salida.

Segunda parte del código, donde se va a aplicar los datos obtenidos en la parte anterior a nuestros datos.

```
java -jar GenomeAnalysisTk.jar \
     -T PrintReads \
     -R BRCA.fasta \
     -I aln.MUESTRA.bam \
     -BQSR recal_data_aln_MUESTRA.table  \
     -o recal_aln_MUESTRA.bam
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T PrintReads: Orden específica. En este caso, PrintReads para la calibración de bases.
- -R BRCA.fasta: se indica el genoma de referencia.
- -I aln.MUESTRA.bam: el archivo de entrada, en este caso el bam en el que va a trabajar.
- -BQSR recal_data_aln_MUESTRA.table: indica que se lleve a cabo una recalibración con los datos obtenidos en el archivo.
- -o recal_aln_MUESTRA.bam: el archivo de salida, se trata del bam con los valores recalibrados.

  - Detección de variantes

Es el proceso en el cual se identifican y etiquetan las diferencias de secuencia encontradas entre nuestros datos y la referencia (8).
Una vez obtenido el bam, se utiliza el algoritmo de detección de variantes de GATK HaploTypeCaller.

```
java -jar GenomeAnalysisTk.jar \
    -T HaplotypeCaller \
    -R BRCA.fasta \
    -I recal_aln_MUESTRA.bam \
    --genotyping_mode DISCOVERY \
    -o recal_raw_aln_MUESTRA_variants.vcf
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T HaplotypeCaller: Orden específica. En este caso, HaplotypeCaller.
- -R BRCA.fasta: se indica el genoma de referencia.
- -I recal_aln_MUESTRA.bam: el archivo de entrada, se trata del bam obtenido después de la recalibración.
- --genotyping_mode DISCOERY: especificación sobre la determinación de los alelos alternativos. Con el argumento DISCOVERY, el programa buscara aquellos que sean los más probables, no los predeterminados.
- -o recal_raw_aln_MUESTRA_variants.vcf: el archivo de salida, se trata del vcf con las variantes encontradas.

  - Filtrado de variantes

El filtrado de variantes es un proceso por el cual se establecen una serie de criterios de calidad a las muestras encontradas, evitando así aquellas que por tener una baja calidad pueden no ser ciertas. El criterio que se va a aplicar a las variantes encontradas por el sistema GATK es el recomendado por los propios creadores de la herramienta, (34).

❖ Obtención de SNPs y aplicación de los parámetros de filtro:

En esta primera parte se va a aplicar los parámetros específicos para los SNPs encontrados. Primero se seleccionan aquellos que se encuntran en el vcf:

```
java -jar GenomeAnalysisTk.jar \
     -T SelectVariants \
     -R BRCA.fasta \
     -V recal_raw_aln_MUESTRA_variants.vcf \
     -selectType SNP  \
     -o recal_raw_snps_MUESTRA.vcf
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T SelectVariants: Orden específica. En este caso, SelectVariants.
- -R BRCA.fasta: se indica el genoma de referencia.
- -V recal_raw_aln_MUESTRA_variants.vcf: el archivo de entrada, se especifica que se trata de un vcf. Se trata del vcf obtenido en el apartado de Detección de variantes.
- -selectType SNP: se seleccionan los snp.
- -recal_raw_snps_MUESTRA.vcf: el archivo de salida es un vcf con los SNPs.

En esta segunda parte se va a aplicar los parámetros, que son: QD < 2.0, FS > 60.0, MQ < 40.0, MQRankSum < -12.5, ReadPosRankSum < -8.0. Aquellas variantes que no cumplan ninguna de estas condiciones serán marcadas como PASS.

```
java -jar GenomeAnalysisTk.jar \
     -T VariantFiltration \
     -R BRCA.fasta \
     -V recal_raw_snps_MUESTRA.vcf \
     - - filterExpression "QD < 2.0 || FS > 60.0 || MQ <
40.0 || MQRankSum < -12.5 || ReadPosRankSum < -8.0" \
     - -filterName “NO PASS” \
     -o filtered_snps_MUESTRA.vcf
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T VariantFiltration: Orden específica. En este caso, VariantFiltration.
- -R BRCA.fasta: se indica el genoma de referencia.
- -V recal_raw_snps_MUESTRA.vcf: el archivo de entrada, es el vcf con sñolo los SNPs.
- --filterExpression "QD < 2.0 || FS > 60.0 || MQ < 40.0 || MQRankSum < -12.5 || ReadPosRankSum < -8.0": son los distintos parámetros que se establecen.
- --filterName “NO PASS”: indica que aquellos cambios que no pasen alguno de los filtros, se marcará como “NO PASS”.
- -o filtered_snps_MUESTRA.vcf: el archivo final, donde se encuentra los SNPSs que se encontraban desde el principio, pero marcadas como ‘PASS’ o ‘NO PASS’ según hayan pasado o no, respectivamente, los filtros.

❖ Obtención de indels, y aplicación de los parámetros de filtro:

Se va a realizar el mismo proceso que el realizado para los SNPs, pero esta vez para los indels.

```
java -jar GenomeAnalysisTk.jar \
     -T SelectVariants \
     -R BRCA.fasta \
     -V recal_raw_aln_MUESTRA_variants.vcf \
     -selectType INDEL  \
     -o recal_raw_indel_MUESTRA.vcf
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T SelectVariants: Orden específica. En este caso, SelectVariants.
- -R BRCA.fasta: se indica el genoma de referencia.
- -V recal_raw_aln_MUESTRA_variants.vcf: el archivo de entrada, al igual que anteriormente, especificando que se trata del vcf obtenido en el apartado de Detección de variantes.

- -selectType INDEL: se seleccionan los indels.
- -o recal_raw_indel_MUESTRA.vcf: los indels quedan guardados en este vcf.

Como se ha señalado con anterioridad, se va a filtrar los indels con una serie de parámetros diferentes a los establecidos en los SNPs. En este caso son: QD < 2.0, FS > 200.0, ReadPosRankSum < -20.0. Se aplican estos valores para obtener el vcf con los indels.

```
java -jar GenomeAnalysisTk.jar \
     -T VariantFiltration \
     -R BRCA.fasta \
     -V recal_raw_indel_MUESTRA.vcf \
     -- filterExpression "QD < 2.0 || FS > 200.0 ||
ReadPosRankSum < -20.0" \
     --filterName “NO PASS” \
     -o filtered_indels_MUESTRA.vcf
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T VariantFiltration: Orden específica. En este caso, VariantFiltration.
- -R BRCA.fasta: se indica el genoma de referencia.
- -V recal_raw_snps_MUESTRA.vcf: el archivo de entrada que contiene sólo los indels.
- --filterExpression "QD < 2.0 || FS > 200.0 || ReadPosRankSum < -20.0"
- --filterName “NO PASS”
- -o filtered_indels_MUESTRA.vcf: archivo final, donde se encuentran los indels con el parámetro ‘PASS’o ‘NO PASS’.

Llegados a este punto, se han obtenido dos vcf: uno con los SNPs, y otro con los indels. En ambos aparecen las mismas variantes que en el archivo sin filtrar (el llamado recal_raw_aln_MUESTRA.vcf), pero con las muestras anotadas como PASS o NO PASS.

Se va a proceder a la unión de ambos vcf y a filtrarlo para obtener sólo las variantes que han pasado el filtro.

```
java -jar GenomeAnalysisTk.jar \
     -T CombineVariants \
     -R BRCA.fasta \
     -- variant filtered_snps_MUESTRA.vcf \
     -- variant filtered_indels_MUESTRA.vcf \
     --filterName “NO PASS” \
     -o union_recalMUESTRA.vcf \
     -genotypeMergeOptions UNIQUIFY
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T CombineVariants: Orden específica. En este caso, CombineVariants.
- -R BRCA.fasta: se indica el genoma de referencia.

- --variant filtered_snps_MUESTRA.vcf: vcf con los SNPs filtrados.
- --variant filtered_indels_MUESTRA.vcf: vcf con los indels filtrados.
- -o union_recalMUESTRA.vcf: archivo final, se han juntado en un mismo vcf.
- -genotypeMergeOptions UNIQUIFY: se crea un solo registro para cada variante.

Se va a llevar a cabo el proceso de selección para aquellas variantes que hayan pasado el filtro.

```
java -jar GenomeAnalysisTk.jar \
     -T SelectVariants \
     -R BRCA.fasta \
     --variant union_recalMUESTRA.vcf \
     -select 'vc.isNotFiltered()' \
     -o gk_filtro_MUESTRA.vcf
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T SelectVariants: Orden específica. En este caso, SelectVariants.
- -R BRCA.fasta: se indica el genoma de referencia.
- variant union_recalMUESTRA.vcf: se trabaja con el vcf obtenido en el paso anterior.
- -select 'vc.isNotFiltered()': se seleccionan solo las que son PASS
- -o gk_filtro_MUESTRA.vcf: vcf final.

Con este vcf final se ha obtenido aquellas variantes que, una vez habiendo realizado la recalibración de bases, hayan pasado los parámetros mínimos de calidad.

6. Samtools

Se va a generar un segundo vcf donde se guardan las variantes obtenidas por la herramienta Samtools (9). No se va a utilizar el bam recalibrado obtenido anteriormente, sino el bam obtenido en el apartado 4, llamado aln.MUESTRA.bam.

Conjuntamente a samtools, se va a utilizar la herramienta bcftools.

```
samtools mpileup -gf BRCA.fasta aln.MUESTRA.bam | bcftools call -vmO v -o st_aln_MUESTRA.vcf
```

- samtools mpileup: orden general samtools mpileup
- -gf BRCA.fasta: referencia que se va a utilizar.
- | bcftools call: se concatena la ejecución del comando bcftools call
- -vmO
- v: se indica que sólo salgan las variantes.
- -o st_aln_MUESTRA.vcf: el archivo de salida, donde estarán las variantes encontradas por samtools.

Una vez que se ha obtenido el archivo vcf con las variantes, se va a aplicar unos parámetros mínimos de calidad para filtrarlos. Siguiendo las recomendaciones de Cornish (35) y Song (36), se establece que el valor mínimo de calidad (QUAL) es mayor o igual a 20, y que el valor mínimo de profundidad de lectura (DP) sea mayor o igual a 10. Ambas características deben cumplirse.

Se va a seguir utilizando la herramienta bcftools, aunque esta vez con la característica de filtrado.

```
bcftools filter -i 'QUAL>=20 & DP>=10' st_aln_MUESTRA.vcf > st_filtro_MUESTRA.vcf
```

- bcftools filter: especificación de la herramienta filtrado dentro de samtools.
- -i 'QUAL>=20 & DP>=10': se establecen los criterios de filtrado.
- st_aln_MUESTRA.vcf: archivo vcf con el que se trabaja.
- st_filtro_MUESTRA.vcf: el vcf obtenido, donde se encuentran las variantes filtradas.

7. Varscan

El tercer vcf se va a generar a través de la herramienta VarScan (37). En combinación con esta herramienta, se va a utilizar samtools (29).

Como en el anterior caso, se va a utilizar el archivo bam llamado aln.MUESTRA.bam, el que se ha generado en el paso número 4. En este caso, se va a aplicar un mínimo de cobertura de 100 lecturas.

```
samtools mpileup -f BRCA.fasta aln.MUESTRA.bam | java -jar VarScan.v2.3.9.jar mpileup2cns --min-coverage 100 --output-vcf --variants > vs_aln_MUESTRA.vcf
```

- samtools mpileup:orden general.
- -f BRCA.fasta: genoma de referencia.
- aln.MUESTRA.bam: archivo con el que se va a trabajar.
- java -jar VarScan.v2.3.9.jar: archivo VarScan
- mpileup2cns: comando que implica detectar cualquier genotipo.
- --min-coverage 100: se establece un mínimo de cobertura en 100.
- --output-vcf: se especifica el formato del archivo de salida.
- --variants: indicación para que sólo se archive en el final las variantes.
- vs_aln_MUESTRA.vcf: nombre del archivo obtenido

Ahora se va a proceder a hacer el mismo paso, pero aplicando una serie de parámetros mínimos de calidad que deben cumplir las variaciones encontradas. Se siguen las recomendaciones de calidad de Warden (38) y se implanta que dichos parámetros de calidad sean: cobertura mínima (--min-coverage) de 100, cobertura mínima del alelo variante (--min-reads2) de 4, calidad mínima de las lecturas (--min-avg-qual) de 20, y una mínima frecuencia del alelo variante de (--min-var-freq) de 0.30.

```
samtools mpileup -f BRCA.fasta aln.MUESTRA.bam | java -jar VarScan.v2.3.9.jar mpileup2cns -v --min-coverage 100 --min-reads2 4 --min-avg -qual 20 --min-var-freq o.30 --output-vcf --variants > vs_filtro_MUESTRA.vcf
```

- samtools mpileup: orden general.
- -f BRCA.fasta: genoma de referencia.
- aln.MUESTRA.bam: archivo con el que se va a trabajar.
- java -jar VarScan.v2.3.9.jar: archivo VarScan
- mpileup2cns: comando que implica detectar cualquier genotipo.
- --min-coverage 100 --min-reads2 4 --min-avg -qual 20 --min-var-freq 0.30: se trata de los parámetros establecidos para llevar a cabo el filtrado.
- --output-vcf: se especifica el formato del archivo de salida.
- --variants: indicación para que sólo se archive en el final las variantes.
- vs_filtro_MUESTRA.vcf

8. Unión de VCF

Una vez que se han obtenido los 3 vcf diferentes, cada uno aplicando un algoritmo diferente, se va a llevar a cabo el proceso de unión. Se establece que las variantes encontradas tienen que estar en un mínimo de dos vcf, es decir, que dicha variante ha de ser captada por un mínimo de dos algoritmos para que aparezcan en el archivo final: no se puede correr el riesgo de aceptar como válidas aquellas que sólo cape un algoritmo, de ahí que se ponga un mínimo de presencia; al igual que no se puede exigir que los tres algoritmos hayan captado el cambio, ya que un posible fallo en uno supondría la eliminación de aquellas variantes que sí que estén en los otros.

Para llevar a cabo dicha unión, se va a volver a utilizar la herramienta GATK.

```
java -jar GenomeAnalysisTk.jar \
     -T CombineVariants \
     -R BRCA.fasta \
     -minN 2 \
     -V: gk_filtro_MUESTRA.vcf \
     -V: st_filtro_MUESTRA.vcf \
     -V: vs_filtro_MUESTRA.vcf \
     -o union_MUESTRA.vcf \
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T CombineVariants: Orden específica. En este caso, CombineVariants.
- -R BRCA.fasta: se indica el genoma de referencia.
- -V: gk_filtro_MUESTRA.vcf: vcf filtrado obtenifo por GATK
- -V: st_filtro_MUESTRA.vcf: vcf filtrado obtenido por SamTools
- -V: vs_filtro_MUESTRA.vcf: vcf filtrado obtenido por VarScan.

-o union_MUESTRA.vcf: vcf que se obtiene

9. Anotación

El último paso que se lleva a cabo es el proceso de anotación, que consiste en predecir los posibles efectos que pueda tener la variante encontrada (28).

Para ello se va a utilizar dos recursos: el primero, la herramienta Snpeff (39), un programa que permite llevar a cabo la anotación de nuestras variantes; para ello se va a necesitar de una base de datos o repositorio que contenga la información sobre dichas variantes. Para ello se necesita el segundo recurso: un listado de las diferentes variantes que se han encontrado que tienen significancia clínica. Dicha base de datos se ha obtenido del archivo ClinVar (40).

```
java -Xmx7g -jar SnpSift.jar annotate -a
clinvar_20171203.vcf union_MUESTRA.vcf >
final_clinvar_MUESTRA.vcf
```

2.3 Comparación de algoritmos

Una vez que se ha realizado el análisis completo de las 18 muestras, se va a proceder a la comparación de los algoritmos de detección de variantes. Como se ha mencionado anteriormente, se han utilizado tres herramientas diferentes: GATK, SamTools, y VarScan. Durante el proceso, se ha obtenido un archivo vcf para cada algoritmo, teniendo además una serie de parámetros mínimos de calidad con los que llevar a cabo el proceso de filtrado.

Se va a llevar a cabo el proceso de unión de los archivos vcf, pero esta vez, y a diferencia del pipeline, se van a unir todos los vcf obtenidos del mismo algoritmo, pero de distinta muestra. De esta manera se va a poder comparar la cantidad de variantes, y que porcentaje de concordancia existen entre las herramientas.

El proceso de unión se llevará a cabo utilizando la herramienta usada en el apartado 6 del punto anterior.

```
java -jar GenomeAnalysisTk.jar \
-T CombineVariants \
-R BRCA.fasta \
-V: gk_filtro_MUESTRA1.vcf \
-V: gk_filtro_MUESTRA2.vcf \
-V: gk_filtro_MUESTRA3.vcf \
- … … etc
-o gk_filtro.todas.vcf \
```

- java -jar GenomeAnalysisTK.jar: Parte general del comando común a la herramienta GATK.
- -T CombineVariants: Orden específica.
- -R BRCA.fasta: se indica el genoma de referencia.
- -V: gk_filtro_MUESTRA1.vcf: se va a trabajar con todas las muestras obtenidas: gk_filtro_MUESTRA2.vcf, gk_filtro_MUESTRA3.vcf, etc.
- -o gk_filtro.todas.vcf

Ahora hay que seguir haciendo este mismo paso, pero con las muestras de los otros conjuntos: st_filtro_MUESTRA1.vcf, st_filtro_MUESTRA2.vcf, etc; y vs_filtro_MUESTRA1.vcf, vs_filtro_MUESTRA2.vcf, etc.

Dado que ya se han obtenido los tres vcf diferentes, conteniendo cada uno las variantes detectadas por cada algoritmo, se va a llevar a cabo el proceso de comparación:

- Anotación

Primero se lleva a cabo un proceso de anotación, por el que se crea una identidad para cada una las variantes encontradas. Se va a utilizar la herramienta bcftools.

```
bcftools annotate -Ob -x 'ID' -I
+'%CHROM:%POS:%REF:%ALT' gk_filtro.todas.vcf >
gk_filtro.todas.bcf
```

- bcftools annotate: especificación para anotar.
- -Ob: el objeto de salida será un archivo bcf
- -x 'ID' -I +'%CHROM:%POS:%REF:%ALT': se crea el ID por variante que contiene el cromosoma, la posición, la base de referencia y la base variante.
- gk_filtro.todas.vcf: el vcf con el que se va a trabajar. Se realiza con los tres vcf creados.
- gk_filtro.todas.bcf: creación del bcf.

- Obtención del listado de variantes.

Gracias a este proceso, se va a poder obtener un listado con el ID creado de cada variante.

```
bcftools view gk_filtro.todas.bcf | cut -f3 | grep -V "^##" | grep -v "^ID"
```

- bcftools view: orden específica para ver los datos de las variantes.
- gk_filtro.todas.bcf: bcf con el que se trabaja. Se realiza en los tres bcf creados.
- | cut -f3 | grep -V "^##" | grep -v "^ID": commando para extraer el ID de las variables.

Una vez se ha finalizado este paso, se va a obtener todas las variantes identificadas por cada uno de los algoritmos, pero llamadas de una manera uniforme para que la comparación pueda ser realizada de una manera correcta.

3. Resultados y discusión

Las distintas variantes encontradas en los procesos de secuenciación masiva en el ámbito clínico requieren de una precisión y validación que nos permitan estar seguros de que los cambios encontradas son de verdad mutaciones en el ADN del paciente y no errores de la tecnología o del procesado de datos. Una incorrecta detección tanto por la obtención de falsos positivos, como por la no captación de variantes reales, hacen que los estudios de análisis del comportamiento de los algoritmos sea un punto importante a tener en cuenta antes de aplicar esta tecnología.

En el apartado del pipeline se ha conseguido obtener dos productos: por un lado, se ha obtenido un script que contiene todos los comandos necesarios para llevar a cabo el análisis de datos de secuenciación masiva que se originen en el laboratorio; así como los conocimientos necesarios como para continuar mejorando y ampliando el presente protocolo, así como para crear nuevos con los que analizar datos de otra índole.

Así mismo, el otro producto que se ha obtenido de ese pipeline ha sido la creación de las distintas bases de datos por cada uno de los pacientes, que contienen tanto las variantes encontradas, como la naturaleza de las mismas.

Por otro lado, usando el pipeline creado para llevar a cabo el Objetivo 1 de este trabajo, se consiguieron 3 archivos de variantes (SNPs e Indels) obtenidos de tres algoritmos de detección de variantes diferentes: GATK, Samtools y VarScan.

El estudio comparativo que se ha realizado se desarrolló mediante una comparación cuantitativa entre ellos estudiando el número de variantes comunes, para ver su solapamiento y las posibles disconformidades que presentan.

Para llevar a cabo dicha comparación se decidió realizar un diagrama de Venn, obteniendo el siguiente resultado:

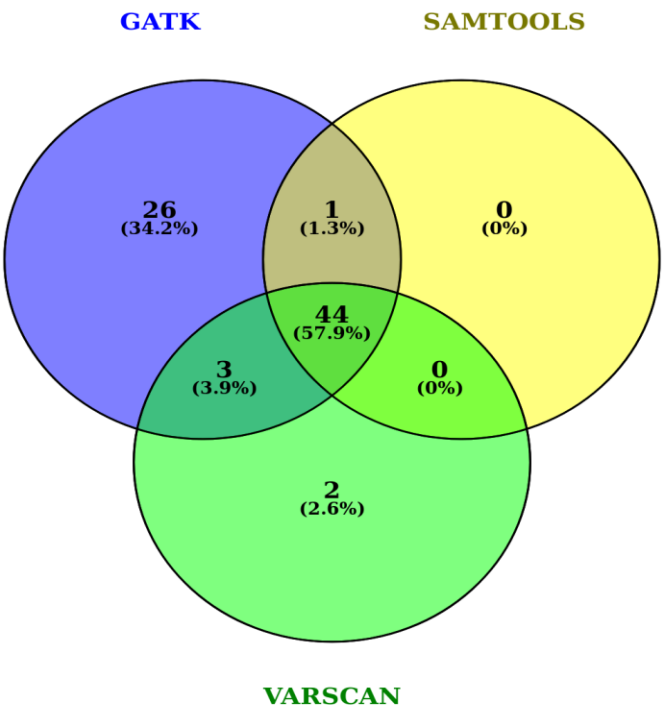

Figura 4: Diagrama de Venn realizado con las variantes encontradas en cada algoritmo.

Se observó que existía una concordancia entre los tres algoritmos de un 57.9%, obteniendo un dato más o menos similar a otros estudios donde obtenían un 57% de concordancia (41), y un 70% (35).

En cuanto a las discrepancias encontradas, se detectó que tanto las tres variantes encontradas por GATK y VarScan y no encontradas por Samtools, como las dos variantes encontradas sólo por VarScan, se trataban en ambos casos de Indels, siendo discordante con otros estudios donde la combinación bwa mem- samtools mpileup (nuestra caso) es la que más Indels detecta (22).

En cuanto al número total de variantes encontradas, VarScan y Samtools presentan números parecidos (49 y 45 respectivamente), en cambio las variantes encontradas por GATK son muy superiores, habiendo detectado 74 variantes, existiendo estudios donde también GATK es la herramienta que más variantes detecta (42).

Del total de variantes encontradas por los tres algoritmos, sólo dos no han sido captadas por GATK (las dos captadas por VarScan).

Las discrepancias encontradas en nuestra comparación, y las diferencias con otros estudios, puede deberse a diferencias en los criterios de filtración

que haya podido tener cada grupo, las distintas versiones que se estén usando de los algoritmos, y a la propia naturaleza de la tecnología.

Sin haber podido realizar pruebas de precisión o estadística más certeras (al no tener una referencia real de las verdaderas variantes) que hubieran permitido determinar con más exactitud la veracidad de los resultados obtenidos, se puede valorar que, de los tres algoritmos analizados, la herramienta GATK sería la más aconsejable de usar en el ámbito clínico.

Puesto que el uso de la secuenciación Sanger se sigue utilizando para confirmar la detección de las variantes por NGS (43), es preferible usar algoritmos que capten mayor número de variantes frente a otros que detecten menos. Si bien es verdad que el sobrecoste es mayor el uso de algoritmos que puedan no detectar variaciones importantes es un riesgo demasiado alto.

4. Conclusiones

Gracias a la realización de este trabajo se ha podido conocer de primera mano las distintas herramientas utilizadas para llevar a cabo los análisis de datos obtenidos por NGS, así como ampliar mis conocimientos de informática. Existen una gran cantidad de algoritmos, distintas estrategias, y variabilidad en las tecnologías, por lo que este trabajo me ha permitido acercarme a una parte de ellas.

Los objetivos iniciales habían sido demasiado altos, por lo que se ha tenido que hacer un reajuste durante el proceso de realización. Uno de los anteriores objetivos principales tuvo que ser eliminado del trabajo. El principal motivo ha sido la falta de tiempo, y la falta de conocimientos que tenía antes de realizar el trabajo.

La metodología no ha cambiado a lo largo del proceso de realización de este trabajo. Como se comentaba anteriormente, la falta de tiempo ha ocasionado un cambio en los objetivos del trabajo, con lo que ha conllevado un cambio en la planificación.

Debido a la gran cantidad de algoritmos que existen, las opciones de líneas futuras son bastante extensas. La implantación de algoritmos de detección de CNV, la detección de variantes somáticas, así como la utilización de algoritmos paralelos a los usados puede ser un punto a seguir desarrollándose.

5. Glosario

- ADN: ácido desoxiribonicleico.
- SNP: del inglés Single Nucleotid Polymorphism.
- INDEL: del inglés, insertions and deletions
- Next generetion Sequencing: secuenciación masiva en castellano. Conjunto de nuevas tecnologías para la obtención de la secuencia de ácidos nucleicos.
- Pipeline: proceso informático continuo.
- Script: archivo de procesamiento.

6. Bibliografía

1. van Dijk EL, Auger H, Jaszczyszyn Y, Thermes C. Ten years of next-generation sequencing technology. Trends Genet [Internet]. 2014 Sep [cited 2018 Jan 9];30(9):418–26. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0168952514001127
2. Rodríguez-Santiago B, Armengol L. Tecnologías de secuenciación de nueva generación en diagnóstico genético pre- y postnatal. Diagnóstico Prenat [Internet]. Elsevier; 2012 Apr [cited 2018 Jan 10];23(2):56–66. Available from: http://linkinghub.elsevier.com/retrieve/pii/S2173412712000273
3. Muzzey D, Evans EA, Lieber C. Understanding the Basics of NGS: From Mechanism to Variant Calling. Curr Genet Med Rep [Internet]. Springer; 2015 [cited 2017 Oct 26];3(4):158–65. Available from: http://www.ncbi.nlm.nih.gov/pubmed/26566462
4. Hodkinson BP, Grice EA. Next-Generation Sequencing: A Review of Technologies and Tools for Wound Microbiome Research. Adv wound care [Internet]. Mary Ann Liebert, Inc.; 2015 Jan 1 [cited 2018 Jan 9];4(1):50–8. Available from: http://www.ncbi.nlm.nih.gov/pubmed/25566414
5. Reuter JA, Spacek D V, Snyder MP. High-throughput sequencing technologies. Mol Cell [Internet]. NIH Public Access; 2015 May 21 [cited 2018 Jan 9];58(4):586–97. Available from: http://www.ncbi.nlm.nih.gov/pubmed/26000844
6. Jennings LJ, Arcila ME, Corless C, Kamel-Reid S, Lubin IM, Pfeifer J, et al. Guidelines for Validation of Next-Generation Sequencing-Based Oncology Panels: A Joint Consensus Recommendation of the Association for Molecular Pathology and College of American Pathologists. J Mol Diagn [Internet]. Elsevier; 2017 May 1 [cited 2018 Jan 8];19(3):341–65. Available from: http://www.ncbi.nlm.nih.gov/pubmed/28341590
7. Zhao M, Wang Q, Wang Q, Jia P, Zhao Z. Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives. BMC Bioinformatics [Internet]. BioMed Central Ltd; 2013;14(Suppl 11):S1. Available from: http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-S11-S1
8. Mehrgou A, Akouchekian M. The importance of BRCA1 and BRCA2 genes mutations in breast cancer development. Med J Islam Repub Iran [Internet]. Iran University of Medical Sciences; 2016 [cited 2018 Jan 5];30:369. Available from: http://www.ncbi.nlm.nih.gov/pubmed/27493913
9. Petrucelli N, Daly MB, Feldman GL. Hereditary breast and ovarian cancer due to mutations in BRCA1 and BRCA2. Genet Med [Internet]. 2010 May 12 [cited 2018 Jan 5];12(5):245–59. Available from: http://www.ncbi.nlm.nih.gov/pubmed/20216074
10. Petrucelli N, Daly MB, Pal T. BRCA1- and BRCA2-Associated Hereditary Breast and Ovarian Cancer [Internet]. GeneReviews®. University of Washington, Seattle; 1993 [cited 2018 Jan 5]. Available from: http://www.ncbi.nlm.nih.gov/pubmed/20301425
11. Next generation sequencing, a new gold standard for clinical gene panel testing | Atlas of Science [Internet]. [cited 2018 Jan 8]. Available from: https://atlasofscience.org/next-generation-sequencing-a-new-gold-standard-

for-clinical-gene-panel-testing/
12. D'Argenio V, Esposito MV, Telese A, Precone V, Starnone F, Nunziato M, et al. The molecular analysis of BRCA1 and BRCA2: Next-generation sequencing supersedes conventional approaches. Clin Chim Acta [Internet]. Elsevier; 2015 Jun 15 [cited 2018 Jan 8];446:221–5. Available from: https://www.sciencedirect.com/science/article/pii/S0009898115001989#bb0005
13. Smith KL, Isaacs C. BRCA mutation testing in determining breast cancer therapy. Cancer J [Internet]. NIH Public Access; 2011 [cited 2018 Jan 5];17(6):492–9. Available from: http://www.ncbi.nlm.nih.gov/pubmed/22157293
14. Shlien A, Malkin D. Copy number variations and cancer. Genome Med [Internet]. 2009 Jun 16 [cited 2017 Oct 15];1(6):62. Available from: http://www.ncbi.nlm.nih.gov/pubmed/19566914
15. Rios J, Puhalla S. PARP inhibitors in breast cancer: BRCA and beyond. Oncology (Williston Park) [Internet]. 2011 Oct [cited 2017 Oct 15];25(11):1014–25. Available from: http://www.ncbi.nlm.nih.gov/pubmed/22106552
16. Mielczarek M, Szyda J. Review of alignment and SNP calling algorithms for next-generation sequencing data. J Appl Genet [Internet]. Journal of Applied Genetics; 2016;57(1):71–9. Available from: http://dx.doi.org/10.1007/s13353-015-0292-7
17. Zare F, Dow M, Monteleone N, Hosny A, Nabavi S. An evaluation of copy number variation detection tools for cancer using whole exome sequencing data. BMC Bioinformatics [Internet]. BMC Bioinformatics; 2017;18(1):286. Available from: http://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1705-x
18. Cock PJA, Fields CJ, Goto N, Heuer ML, Rice PM. SURVEY AND SUMMARY The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. [cited 2018 Jan 5]; Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2847217/pdf/gkp1137.pdf
19. Chen C, Khaleel SS, Huang H, Wu CH. Software for pre-processing Illumina next-generation sequencing short read sequences. Source Code Biol Med [Internet]. BioMed Central; 2014 [cited 2018 Jan 5];9:8. Available from: http://www.ncbi.nlm.nih.gov/pubmed/24955109
20. Shin S, Park J. Characterization of sequence-specific errors in various next-generation sequencing systems. Mol Biosyst [Internet]. 2016 Mar [cited 2018 Jan 10];12(3):914–22. Available from: http://www.ncbi.nlm.nih.gov/pubmed/26790373
21. Fox EJ, Reid-Bayliss KS, Emond MJ, Loeb LA. Accuracy of Next Generation Sequencing Platforms. Next Gener Seq Appl [Internet]. NIH Public Access; 2014 [cited 2018 Jan 10];1. Available from: http://www.ncbi.nlm.nih.gov/pubmed/25699289
22. Bao R, Huang L, Andrade J, Tan W, Kibbe WA, Jiang H, et al. Review of current methods, applications, and data management for the bioinformatics analysis of whole exome sequencing. Cancer Inform [Internet]. SAGE Publications; 2014 [cited 2018 Jan 5];13(Suppl 2):67–82. Available from: http://www.ncbi.nlm.nih.gov/pubmed/25288881

23. Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.journal [Internet]. 2011 May 2 [cited 2018 Jan 5];17(1):10. Available from: http://journal.embnet.org/index.php/embnetjournal/article/view/200
24. Bao R, Huang L, Andrade J, Tan W, Kibbe WA, Jiang H, et al. Review of Current Methods, Applications, and Data Management for the Bioinformatics Analysis of Whole Exome Sequencing. Cancer Inform [Internet]. 2014 Jan 21 [cited 2018 Jan 10];13s2(Suppl 2):CIN.S13779. Available from: http://www.ncbi.nlm.nih.gov/pubmed/25288881
25. Li J, Batcha AMN, Grüning B, Mansmann UR. An NGS Workflow Blueprint for DNA Sequencing Data and Its Application in Individualized Molecular Oncology. Cancer Inform [Internet]. SAGE Publications; 2015 [cited 2018 Jan 5];14(Suppl 5):87–107. Available from: http://www.ncbi.nlm.nih.gov/pubmed/27081306
26. Li H, Durbin R. Fast and accurate long-read alignment with Burrows-Wheeler transform. Bioinformatics [Internet]. 2010 Mar 1 [cited 2018 Jan 6];26(5):589–95. Available from: http://www.ncbi.nlm.nih.gov/pubmed/20080505
27. The GRCh37 assembly in Ensembl [Internet]. [cited 2018 Jan 6]. Available from: https://www.ensembl.org/info/website/tutorials/grch37.html
28. Albert I. The Biostar Handbook. 2017.
29. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics [Internet]. Oxford University Press; 2009 Aug 15 [cited 2018 Jan 6];25(16):2078–9. Available from: http://www.ncbi.nlm.nih.gov/pubmed/19505943
30. Sequence Alignment/Map Format Specification. 2017 [cited 2018 Jan 6]; Available from: https://samtools.github.io/hts-specs/SAMv1.pdf
31. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res [Internet]. Cold Spring Harbor Laboratory Press; 2010 Sep [cited 2018 Jan 6];20(9):1297–303. Available from: http://www.ncbi.nlm.nih.gov/pubmed/20644199
32. Base Quality Score Recalibration (BQSR) — GATK-Forum [Internet]. [cited 2018 Jan 6]. Available from: https://gatkforums.broadinstitute.org/gatk/discussion/44/base-quality-score-recalibration-bqsr
33. National Center for Biotechnology Information [Internet]. [cited 2018 Jan 6]. Available from: https://www.ncbi.nlm.nih.gov/
34. (howto) Apply hard filters to a call set — GATK-Forum [Internet]. [cited 2018 Jan 6]. Available from: https://gatkforums.broadinstitute.org/gatk/discussion/2806/howto-apply-hard-filters-to-a-call-set
35. Cornish A, Guda C. A Comparison of Variant Calling Pipelines Using Genome in a Bottle as a Reference. Biomed Res Int [Internet]. Hindawi Limited; 2015 [cited 2018 Jan 7];2015:456479. Available from: http://www.ncbi.nlm.nih.gov/pubmed/26539496
36. Song K, Li L, Zhang G. Coverage recommendation for genotyping analysis of highly heterologous species using next-generation sequencing technology. Sci Rep [Internet]. Nature Publishing Group; 2016 Oct 20 [cited 2018 Jan 7];6:35736. Available from:

http://www.ncbi.nlm.nih.gov/pubmed/27760996
37. Koboldt DC, Chen K, Wylie T, Larson DE, McLellan MD, Mardis ER, et al. VarScan: variant detection in massively parallel sequencing of individual and pooled samples. Bioinformatics [Internet]. 2009 Sep 1 [cited 2018 Jan 9];25(17):2283–5. Available from: http://www.ncbi.nlm.nih.gov/pubmed/19542151
38. Warden CD, Adamson AW, Neuhausen SL, Wu X. Detailed comparison of two popular variant calling packages for exome and targeted exon studies. PeerJ Inc.; 2014 Jun 12 [cited 2018 Jan 9]; Available from: https://peerj.com/preprints/403/
39. Cingolani P, Platts A, Wang LL, Coon M, Nguyen T, Wang L, et al. A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff. Fly (Austin) [Internet]. 2012 Apr 27 [cited 2018 Jan 10];6(2):80–92. Available from: http://www.ncbi.nlm.nih.gov/pubmed/22728672
40. Landrum MJ, Lee JM, Riley GR, Jang W, Rubinstein WS, Church DM, et al. ClinVar: public archive of relationships among sequence variation and human phenotype. Nucleic Acids Res [Internet]. Oxford University Press; 2014 Jan [cited 2018 Jan 10];42(Database issue):D980-5. Available from: http://www.ncbi.nlm.nih.gov/pubmed/24234437
41. O'Rawe J, Jiang T, Sun G, Wu Y, Wang W, Hu J, et al. Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing. Genome Med [Internet]. 2013 [cited 2018 Jan 10];5(3):28. Available from: http://www.ncbi.nlm.nih.gov/pubmed/23537139
42. You N, Murillo G, Su X, Zeng X, Xu J, Ning K, et al. SNP calling using genotype model selection on high-throughput sequencing data. Bioinformatics [Internet]. 2012 Mar 1 [cited 2018 Jan 9];28(5):643–50. Available from: http://www.ncbi.nlm.nih.gov/pubmed/22253293
43. Mu W, Lu H-M, Chen J, Li S, Elliott AM. Sanger Confirmation Is Required to Achieve Optimal Sensitivity and Specificity in Next-Generation Sequencing Panel Testing. J Mol Diagnostics [Internet]. 2016 Nov [cited 2018 Jan 10];18(6):923–32. Available from: http://www.ncbi.nlm.nih.gov/pubmed/27720647