Desarrollo de un pipeline Bioinformático mediante R: Análisis basados en panel de cáncer de pulmón.

David Masip Galaso
Máster Universitario en Bioinformática y Bioestadística UOC - UB
Computación e Inteligencia artificial en problemas biológicos y clínicos

Consultora: Romina Astrid Rebrij
Profesor responsable de la asignatura: David Merino Arranz

Fecha Entrega: Junio 2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Desarrollo de un pipeline Bioinformático mediante R:* *Análisis basados en panel de cáncer de pulmón.*
Nombre del autor:	*David Masip Galaso*
Nombre del consultor/a:	*Romina Astrid Rebrij*
Nombre del PRA:	*David Merino Arranz*
Fecha de entrega (mm/aaaa):	06/2019
Titulación:	*Máster Universitario en Bioinformática y Bioestadística*
Área del Trabajo Final:	*Computación e Inteligencia Artificial en problemas biológicos y clinicos*
Idioma del trabajo:	*Castellano*
Palabras clave	*Data mining; Lung cancer; Fastq files*

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

En la actualidad, el cáncer de pulmón es uno de los tipos de cáncer más comunes y de los más extensos. En comparación con el resto de cánceres, alrededor del 14% de todos los cánceres nuevos son cánceres procedentes de este órgano. Debido a las altas estadísticas, se cree interesante envolver el presente trabajo de Fin de Máster con la realización de un pipeline bioinformático a partir de un panel de genes de cáncer de pulmón. Para ello, se parte de una base con datos tipo “Fastq files” procedente de un panel de genes para su posterior parametrización [1][2][3][4].

Con el presente objetivo y en plena expansión e integración genómica en el campo del diagnóstico clínico, encontramos una razón de peso para la ejecución de dicho particular estudio dentro del campo de diagnóstico. El tipo de lenguaje utilizado para este proyecto será la plataforma de software libre “R” y sus distintos paquetes de Bioconductor utilizados a lo largo del pipeline, los cuáles nos permitirán filtrar y definir con gran detalle toda la información proveniente de los archivos seleccionados, sacar datos estadísticos representarlos de una forma rápida, simple, coherente y de la mejor manera posible.

Esta aplicación puede utilizarse para cualquier tipo de estudio siempre y cuando el punto de inicio sea una base con datos de clase fastq files, ya sea precedente de Pubmed, de cualquier tipo de panel de genes procedentes de un secuenciador o de cualquier otra fuente. Al publicarse en la red, ésta además puede ser utilizada por cualquier usuario, incluso aquellos sin conocimientos en lenguajes de programación específico.

Abstract (in English, 250 words or less):

Nowadays, lung cancer is one of the most common and the most extensive sort of cancer. At a glance, in comparison with others, around 14% of all new diagnosed cancers are lung cancer. Due to the high likelihood, it is interesting to link the current final Project Master thesis with the relationship of a bioinformatic pipeline from a lung cancer genes panel. For its deployment, we will start with a certain amount of data coming from a panel of genes in fastq files format for its parametrization [1][2][3][4].

With the main goal defined and with the genomic integration processes in expansion onto clinical diagnostics field, we find a good reason to start deploying such a particular study onto this specific field. The programming language used for this project is the free software platform “R” and different packages used for pipelines, which will allow us to filter and define the correct information, graphic and represent them and get it as most simple, fast, coherent and correct way as possible, getting the decided outcome.

Aside from this, this application will be able to be used for any kind of study as long as the starting point is a database with fastq files class data, either coming from Pubmed, any type of genes panel, from any sequencer instrument or from any other source. As it is going to be published on the network, this can also be used by any user, even those with no further knowledge on specific programming languages.

Índice

Indice de Figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

Desde el inicio de los años 90, dónde se empezó a gestar el inicio del proyecto del [6] Genoma Humano, y cuya culminación tuvo lugar alrededor de 2006 con la publicación de la secuencia del último cromosoma, gran cantidad de información se ha podido ver representada y analizada, tanto desde un punto de conocimiento científico, pasando por el diagnóstico de enfermedades y su posterior tratamiento hasta la edición de partes de la cadena. Pese a que los dos grandes objetivos principales del proyecto fueron, por un lado tratar de averiguar la posición de todos los nucleótidos (secuenciación) y por otro, la localización de los genes en cada uno de los 23 pares de cromosomas del ser humano (cartografía), este hito del S.XXI, ha hecho posible que poco a poco, se haya podido sacar varios patrones genéticos relacionados con enfermedades, para intentar si más bien no, poderlas prevenir parcialmente en un futuro, atrasarlas en la manera de lo posible o tratarlas de una forma más precisa y temprana. Actualmente, la capacidad de un diagnóstico clínico seguro a partir de un exoma ya es una realidad estandarizada en muchos campos del diagnóstico. Entre otras, algunas enfermedades degenerativas, enfermedades denominadas raras o actualmente una de las mayores causas de mortalidad en el mundo: el cáncer.

Partiendo de esta enfermedad, la relevancia en encontrar un patrón de rasgos genéticos que sean los promotores de la alteración de dichas células que posteriormente sufren una mutación y por consiguiente esto se transforma en enfermedad, se cree interesante envolver el presente trabajo con la realización de un pipeline bioinformático a partir de un panel específico de genes con cáncer de pulmón.

El punto de partida de los datos a tratar, son archivos tipo [7] fastq files estructurados junto con su índice adicional, pero simplemente se conoce que son específicos de un panel de genes sobre este tipo cáncer, por lo que centrándonos en su tratamiento, en un inicio, se establece un alcance de proyecto para su tratamiento, análisis, clasificación y obtención de resultados. Ante la posibilidad de obtener los datos del ARN correspondiente al panel, si fuera posible, se podría hacer una clasificación según el tipo de expresión, aunque en un inicio no se contempla dentro del marco del presente proyecto.

Para llevar a cabo el estudio de dichos datos, éste se va a generar mediante un estudio analítico sobre dicho panel de genes y si fuera posible y el tiempo lo permitiera, dentro del marco del proyecto se va a ampliar con una herramienta de lectura tipo aplicativa, dónde a partir de datos del estudio, provenientes de un panel de genes de pulmón, permita a los facultativos, de una forma rápida, obtener un resultado específico sobre la calidad de estos estudios, y facilitar así, la labor de diagnóstico clínico. Para llevar a cabo dicho el estudio se utilizará la plataforma de lenguaje libre R sobre distintos paquetes de procesado y reporting, explicados en los siguientes apartados.

Dada la amplitud de dicho trabajo y teniendo en cuenta que el punto de partido son datos tipo fastq files, esta aplicación podría ser útil para cualquier tipo de análisis procedentes de otras fuentes, tipo NCBI, Pubmed, cualquier otro panel de genes o datos de un secuenciador. Asimismo, este proyecto tiene el alcance de introducir unos

principios de tratamiento de datos de secuenciación para cualquier tipo de usuario, incluso aquellos sin conocimientos en lenguajes de programación específico.

1.2 Objetivos del Trabajo

1.2.1 Objetivos generales:

- Desarrollar un pipeline bioinformático a partir de unos archivos de datos fastq files sobre un panel de genes de cáncer de pulmón, utilizando Bioconductor a partir del uso de software R y otras aplicaciones para la visualización de los datos.

1.2.2 Objetivos específicos:

- Los objetivos específicos del proyecto se basan en cada uno de los pasos a desarrollar en el pipeline: Carga de datos, análisis de calidad y filtraje/recorte si es necesario, conversión de archivos para alineamiento, verificación y estudio de variantes.

- El Software es un estudio analítico en formato “R” que toma uno o varios archivos fastq files y ejecuta una serie de pruebas en él para generar un informe de control de calidad completo. Esto dirá si hay algo inusual en la secuencia seleccionada. Cada prueba se marca como un “pasa”, “advertencia” o “fallo”, dependiendo de los lejos que éste se aleja de lo que cabría esperar de un conjunto de datos normal sin esos sesgos. Es importante destacar que las advertencias o incluso fallos, no necesariamente significan que haya un problema con los datos, sino que es inusual. Además, adicionalmente se hace una transformación de archivo y posterior alineamiento y visualización de las anotaciones, iendo en profundidad sobre algunos aspectos a resaltar de la secuencia seleccionada.

1.3 Enfoque y método seguido

Dado que el entonro R ha sido la herramienta vehicular de trabajo a lo largo del presente máster, éste va a ser el lenguaje a partir se centre el análisis del tratamiento de los datos, trabajando en cada caso con las librerias necesarias para la computación de datos estadísticos y gráficos.

Para el análisis de los paquetes de datos se van a utilizar varias librerias, la principal y más extendida es Bioconductor. Esta herramienta proporciona unos análisis y comprensión de alto nivel procedentes de datos genómicos mediante R. Dentro de Bioconductor, se eligirán los mejores paquetes, según la fase de trabajo en la cual se ejecuten las sentencias: Importación, filtraje/ordenación/clasificacación, transformación, visualización, modelo.

A lo largo de la figura 1. Metodología utilizada, se intenta resumir el "workflow" de trabajo que se utilizará para obtener el resultado final a través de la pantalla.
Una vez importados los archivos, éstos deben de ser pre-tratados, filtrados y ordenados para su podterior análisis. Una vez estructurados, se debe de pensar en qué tipo y en qué fundamentos se quiere crear el modelo que respresente esos datos. Finalmente y en paralelo, una vez creado el modelo que se quiera ver, los datos previamente importados serán transformados en información y gráficos de interés que al mismo tiempo se verán representados para su posterior visualización y extracción de conclusiones.

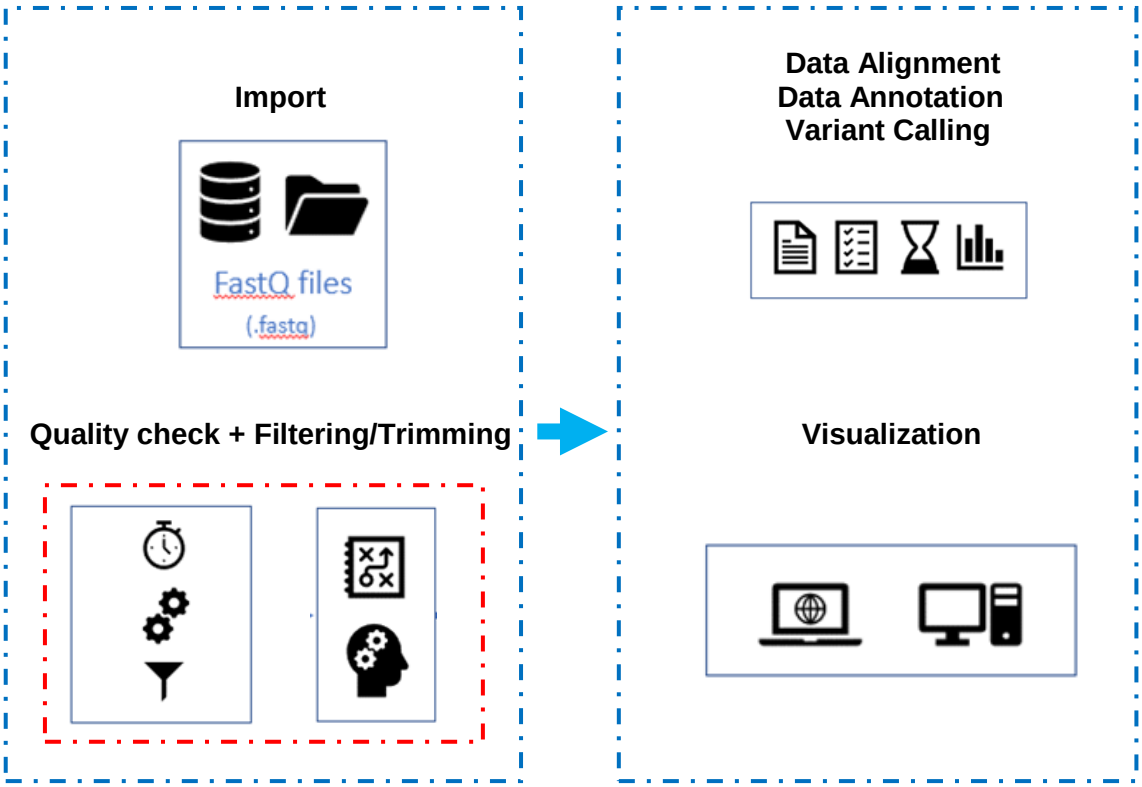


Figura 1. Metodología utilizada

1.4 Planificación del Trabajo

El trabajo se estructura según los hitos temporales a lo largo del presente proyecto, teniendo en cuenta su temporalización correspondiente a cada una de las entregas parciales de las PEC. Por consiguiente, se ha creído oportuno dividir las entregas según muestra a continuación:

1.4.1 Tareas:

1.4.1.1 Plan de Trabajo inicial

- Definición de contenidos del TFM.
- Definición de tareas e hitos asignados al cronograma.

1.4.1.2 Desarrollo de trabajo FASE I

- Requisitos previos a la ejecución del pipeline:
    - Búsqueda de información y documentación para la ejecución del TFM mediante R y paquetes Bioconductor.
    - Aprendizaje sobre paquetes Bioconductor: ShortRead, Bowtie2, GenomicsAlignments, entre otros.
    - Análisis de la herramienta mediante Rstudio

- Creación de proyecto Rmarkdown e implementación pasos fase I:
    - Instalación librerías correspondientes.
    - Lecturas y de datos del pipeline.
    - Exploración de datos cargados y procesados.
    - Control de Calidad de los datos cargados
- Implementación de cálculos de la fase I siguiendo proyecto Rmarkdown.
- Documentación en memoria.

1.4.1.3 Desarrollo de trabajo FASE II

- Implementación pasos fase II:
    - Lectura y decisión de paquete a utilizar para Alineamiento: Bowtie2
    - Obtención de alineación con Genoma de Referencia
    - Reporting final
- Muestra de datos de cálculos de la fase II.
- Documentación en memoria.

1.4.1.4 Elaboración de la memoria

- La memoria se realiza durante toda la duración del proyecto y en cada caso se analizan y estudian los datos obtenidos en cada fase.

1.4.1.5 Elaboración de la presentación

Elaboración de presentación mediante format .ppt para la exposición del trabajo y datos más destacados del trabajo.

1.4.1.6 Defensa pública

El trabajo se presentará delante del tribunal designado para la ocasión, siendo objeto y alcance de defensa el presente alcance del proyecto.

1.4.1.7 Calendario

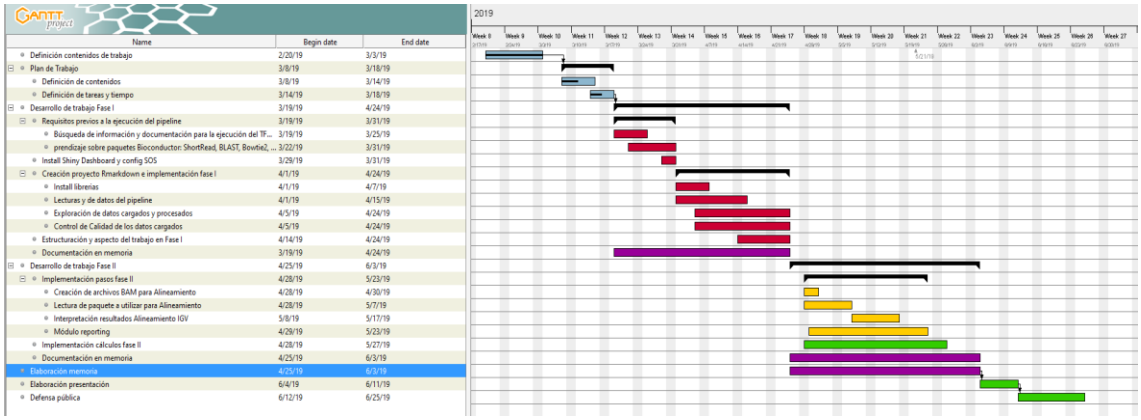


Figura 2. Calendario

1.5 Breve sumario de productos obtenidos

1. Plan de trabajo del TFM
2. Memoria
    a. Documento acreditativo en el que se recapitulan todos los pasos ejecutados a lo largo del proyecto. Este documento se mantiene vivo a lo largo de toda la duración del proyecto, actualizándose en cada fase.
3. Producto
    a. Estudio detallado sobre los archivos seleccionados, donde de una forma estructurada se presentan las distintas fases ejecutadas:

        - Carga de ficheros
        - Viabilidad de estudios de calidad
        - Filtrado y recorte (si aplica)
        - Alineamiento
        - Anotaciones
        - Variantes
        - Representación gráfica

4. Presentación
    a. Exposición mediante presentación de archivo .ppt

1.6 Análisis de riesgos

1.6.1 Riesgos asociados a los fastq files:

Los riesgos asociados a los datos del proyecto provienen de unos archivos tipo fastq files relacionados con paneles de genes procedentes de cáncer de pulmón. En un principio estos paneles deberían de ser estables y no deberían de generar mayores incovenientes, aunque se desconoce el estado de los mismos. Asimismo, no viene acompañados de ningún archivo FASTA ni BAM file, por lo que parte del alcance de este proyecto será generar dichos archivos con extensión .BAM para su posterior tratamiento y alineación. Los Fastq files, son archivos de lectura tipo single read, y cuyo identificador viene representado en un archivo aparte. En la sección del cuerpo del proyecto, se definirán cada una de las partes de dichos archivos. No obstante, como se ha comentado en capítulos anteriores, el objeto de este pipeline, se podría extender a cualquier tipo de archivos con la misma extensión, ya fueran exomas, secuencias procesadas, paneles de genes o cualquier otro archivo con extensión fastq. Debido a esto último, se deberían de extremar las precauciones de obtención de datos, debido a que existe una posibilidad de que haya problemas con los datos seleccionados y el procesamiento de estos de obtenga el producto final esperado.

1.6.2 Riesgos asociados a la búsqueda de información:

Parte del proyecto demanda una exigente búsqueda bibliográfica sobre los paquetes y métodos analizados para llevar a cabo el pipeline bioinformático sobre dichos paneles. Debido a este aspecto, es importante tener un grado de familiarización con los archivos del tipo fastq files y el significado de cada paquete a utilizar, además de tener cierto dominio práctico. Es posible que la búsqueda y herramientas a utilizar sea más extensa que la propuesta en el alcance del proyecto.

1.6.3 Riesgos asociados al tiempo y alcance del proyecto:

Teniendo en cuenta el alcance el proyecto y dependiendo del tipo de pipeline a ejecutar, asi como el módulo de reporting asociado, el planing podría verse afectado, dependiendo de la velocidad y calidad en la que se efectúen los análisis y se consoliden los hitos del plan de trabajo.

1.7 Breve descripción de los otros capítulos de la memoria

En el capítulo 2 se presenta un estudio estadístico sobre el cáncer de pulmón.

En el capítulo 3 se presenta una introducción sobre los paquetes de Bioconductor más desarrollados a lo largo del Pipeline: ShortRead para la preparación y lectura de datos.

En el capítulo 4 se detallan los resultados en cuanto a calidad de los archivos, así como la conclusión de la necesidad de su posterior filtrado o recorte de secuencias para la obtención de los datos previos a la alineación con Genoma elegido.

En el capítulo 5 se detalla el proceso de alineación mediante paquete Bowtie2.[30]

En el capítulo 7 se presenta el proceso seguido en este TFM para realizar el pipeline a partir del input de la carga de los archivos fastq files y como se aplican los consiguientes pasos, para la obtención de las lecutras finales.

Se incluye un anexo (Anexo 1) con el código fuente de la aplicación desarrollada y un glosario de términos y expresiones utilizados durante el proyecto.

2. Estudio sobre el cáncer de pulmón

El término cáncer engloba un grupo numeroso de enfermedades que se caracterizan por el desarrollo de células anormales, que se dividen, crecen y se diseminan sin control en cualquier parte del cuerpo. Las células normales se dividen y mueren durante un periodo de tiempo programado. Sin embargo, la célula cancerosa o tumoral "pierde" la capacidad para morir y se divide casi sin límite. Tal multiplicación en el número de células llega a formar unas masas, denominadas "tumores" o "neoplasias", que en su expansión pueden destruir y sustituir a los tejidos normales.[10].

En los últimos 20 años, el número de tumores diagnosticados ha experimentado un crecimiento constante en España debido no sólo al aumento poblacional, sino también a las técnicas de detección precoz y al aumento de la esperanza de vida (ya que el riesgo de desarrollar tumores aumenta con la edad). Entre ellos, el cancer de pulmón.

El cáncer de pulmón generalmente se forma en el tejido de las células que recubren los conductos de aire en los pulmones. Los dos tipos principales son el cáncer pulmonar de células pequeñas y el cáncer pulmonar de células no pequeñas, que tiende a crecer más lentamente y tarda más tiempo en extenderse más allá de los pulmones. Entre las causas del primer tipo, el tabaquismo representa el principal carcinógeno ambiental conocido para padecer cáncer, siendo la primera causa de cáncer de pulmón, y un factor importante en otros cánceres como los de cabeza y cuello, esófago, estómago, páncreas,etc.

A continuación se muestran un par de figuras donde residen pruebas realizadas para el diagnóstico patológico del cáncer de pulmón:

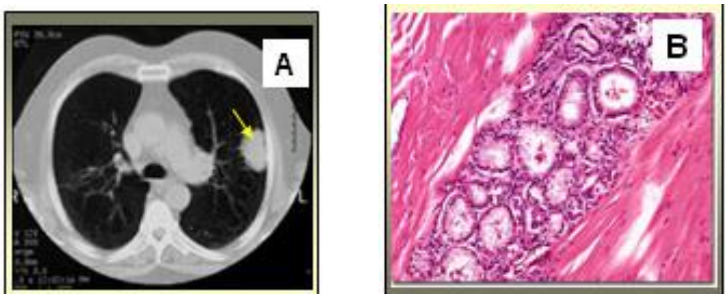


Figura 3. A. Aspecto radiológico, en la tomografía computerizada (TC), de un carcinoma de pulmón.
B. Aspecto microscópico por el que se confirma un tumor de pulmón cuya histología es adenocarcinoma. [10].

A nivel estadístico, los tumores más frecuentes a nivel mundial fueron los de pulmón, mama, colorrecto, próstata, estómago e hígado.

Por lo que representa España, se estima que los cánceres más frecuentes diagnosticados en varones en 2019 serán los de próstata, colon y recto, pulmón y vejiga urinaria. A mucha distancia, los siguientes cánceres más frecuentes serán los de cavidad oral y faringe, riñón, hígado y estómago, los linfomas no hodgkinianos y el cáncer de páncreas, todos ellos con más de 4.000 casos al año.

A continuación se muestran los gráficos de probabilidad de desarrollo de cáncer por sexo y franja de edad, asi como la frecuencia de tumores diagnosticados a nivel mundial y la estimación de fallecimientos en ambos sexos. Resaltar que el cáncer de pulmón es la primera causa de mortalidad a nivel mundial y su estimación con respecto alas caulsas por fallecimiento respecto a otros tumores, todavía es muy lejano.

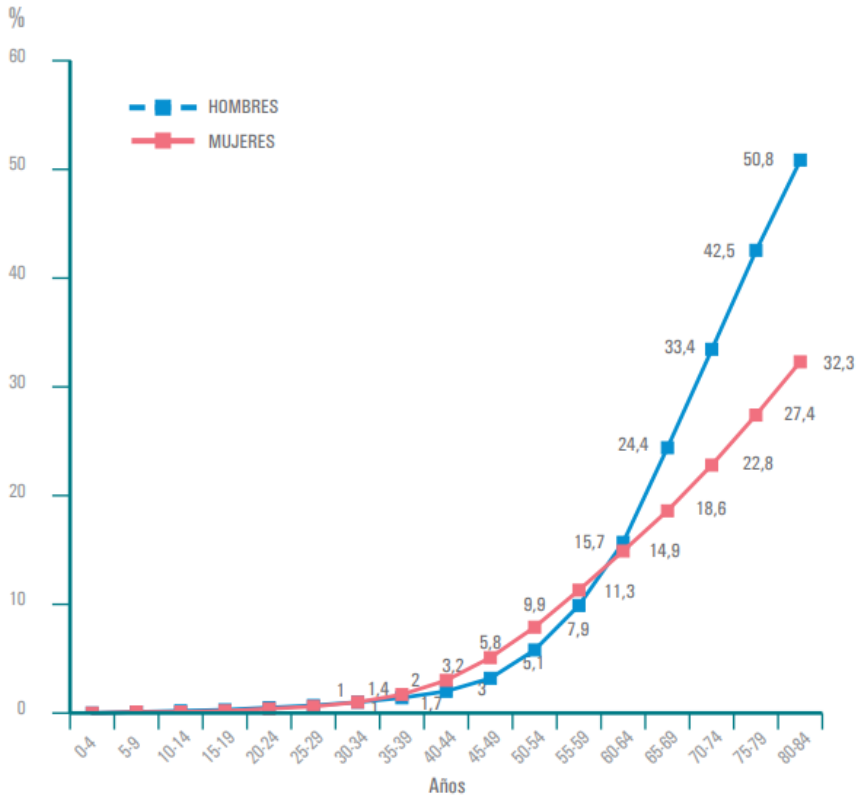


Figura 4. Probabilidad (%) de desarrollar un cáncer (excluidos los tumores cutáneos no melanoma) por sexos en España en el año 2019. [10].

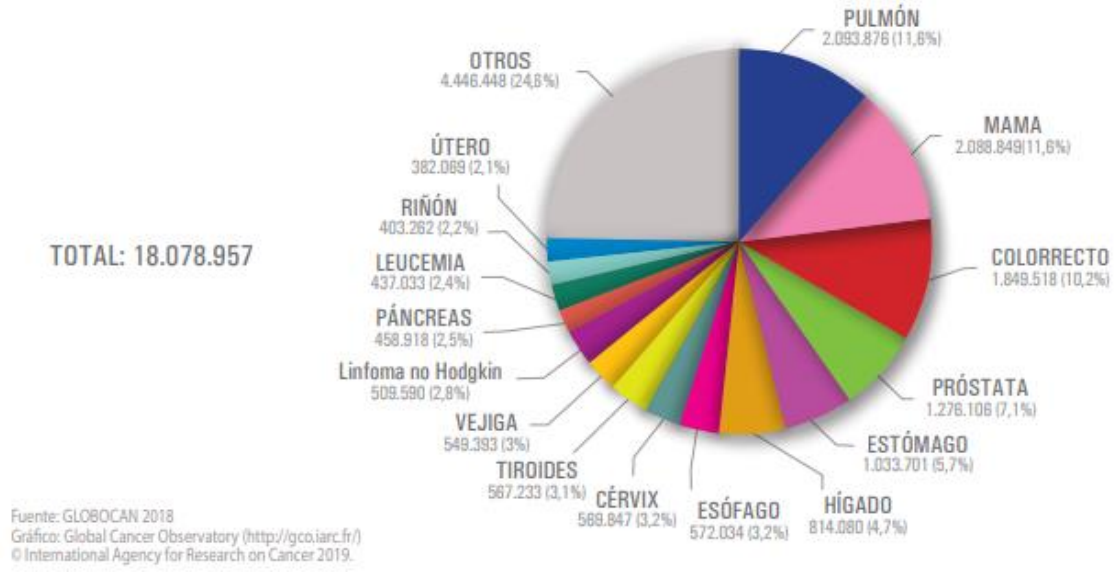


Figura 4. Tumores más frecuentemente diagnosticados en el mundo. Estimación para el año 2018, ambos sexos [10].

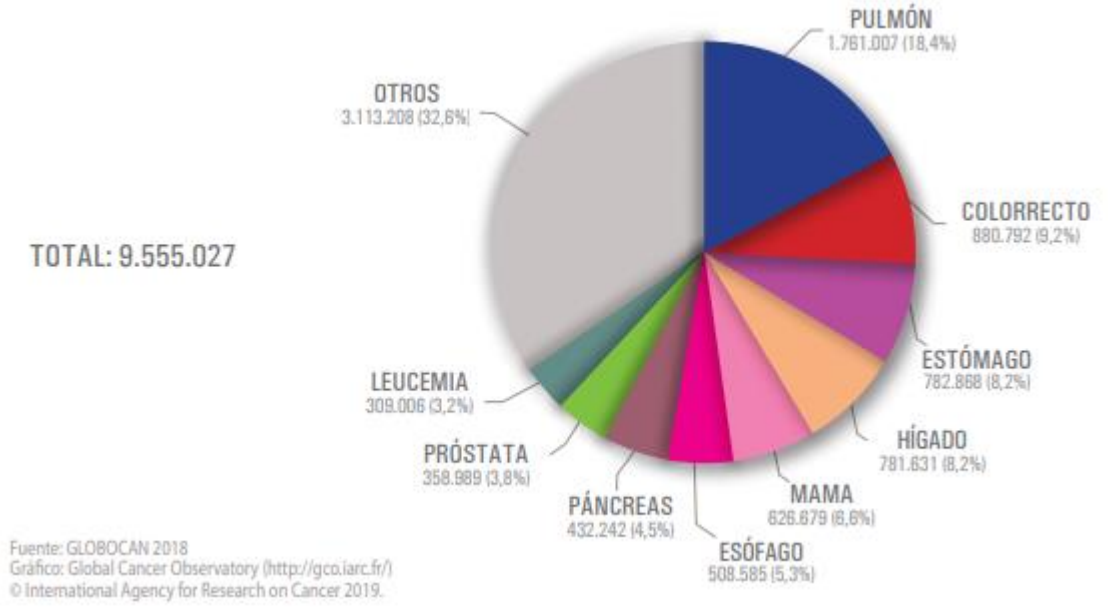


Figura 5. Estimación del número de fallecimientos por tumores en el mundo en el año 2018, ambos sexos [10].

A nivel estatal, el índice de mortalidad estandarizada por cancer de pulmón sigue el patrón internacional, siendo la principal causa de fallecimiento.

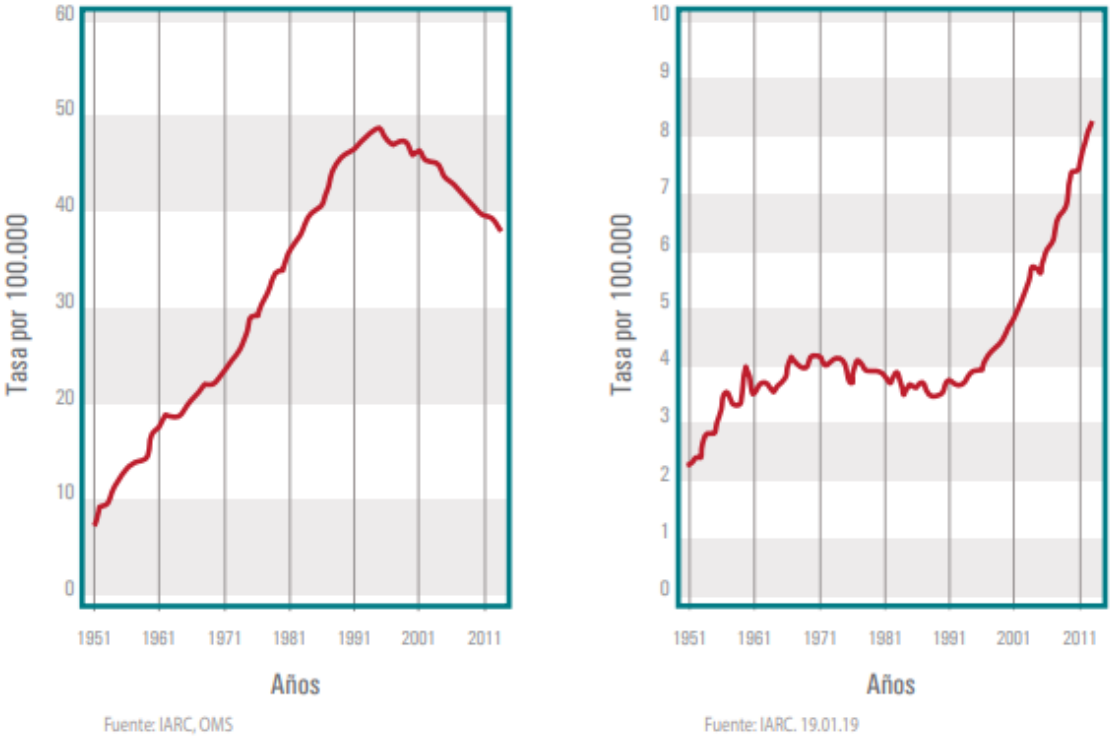


Figura 6. Mortalidad estandarizada por edad por cáncer de pulmón en España, varones (izquierda) y mujeres (derecha) (periodo 1951-2011). [10].

3. Next Generation Sequencing (NGS)

La NGS (Secuenciación de la próxima generación), también conocida como secuenciación de alto rendimiento, es el término general para describir una serie de diferentes tecnologías modernas de secuenciación. Estas tecnologías permiten la secuenciación del ADN y el ARN de forma mucho más rápida y económica que la secuenciación de Sanger utilizada anteriormente, y como tal revolucionaron el estudio de la genómica y la biología molecular.

El procesamiento bioinformático de datos NGS puede ser operacionalmente dividido en tres pasos principales:

1. Generación de una secuencia de lectura que contiene una secuencia lineal de nucleótidos (por ejemplo, ACTGGCA) realizada utilizando software específico del instrumento (archivo FASTQ).

2. Control de Calidad de la secuenciación, verificación de filtraje y/o ejecución de recorte si es necesario.

3. Mapeo y alineación de lecturas de secuencia contra una secuencia de referencia e identificando las diferencias (variantes) entre la secuencias leídas y la referencia seleccionada (archivo SAM / BAM).

4. Anotación de las variantes (archivo VCF) a generación de informe.

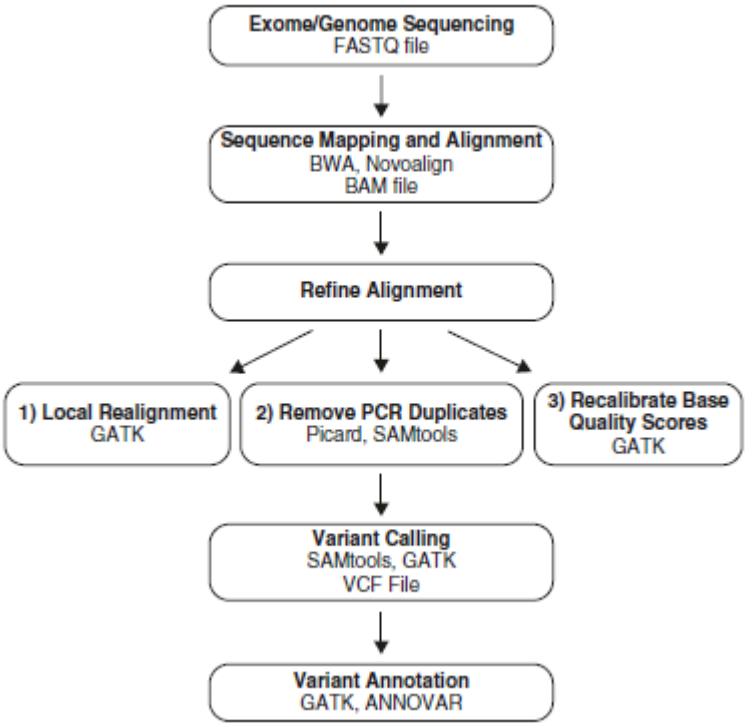


Figura 7. NGS Pipeline Workflow

4. Paquetes Bioconductor: ShortRead

Bioconductor es un proyecto de código abierto para el análisis de datos en Genómica. El principal objetivo de desarrollar e integrar software para el análisis estadístico de datos de laboratorio en biología molecular. Está basado en el lenguaje de programación R. Su principal aplicación es el análisis de microarrays. Los archivos FASTQ pueden usarse como entrada de secuencia para alineación y otro software de análisis secundario. Dentro de este proyecto, constan varios paquetes que se pueden implementar a través del lenguaje de programación. Para nuestro Pipeline, se irán añadiendo varios, pero a continuación se definen aquellos con más cuerpo a lo largo del proyecto.[21]

El paquete ShortRead, implementa el muestreo, la iteración y la entrada de archivos cuya extensión es FASTQ. El formato de estos archivos FastQ es el siguiente:

- Identificador de secuencia
- Secuencia
- Línea de identificación de puntuación de calidad (representada por un +)
- Nivel de calidad

La primera línea, identificando la secuencia, contiene los siguientes elementos:

@<instrument>:<run number>:<flowcell ID>:<lane>:<tile>:<x-pos>:<y-pos>:<UMI> <read>:<is filtered>:<control number>:<index>

A modo de resumen, en la siguiente table se detallan todos los elementos de cada una de las partes que describe un archive tipo .fastq.

Element	Requirements	Description
@	@	Each sequence identifier line starts with @
<instrument>	Characters allowed: a–z, A–Z, 0–9 and underscore	Instrument ID
<run number>	Numerical	Run number on instrument
<flowcell ID>	Characters allowed: a–z, A–Z, 0–9	
<lane>	Numerical	Lane number
<tile>	Numerical	Tile number
<x_pos>	Numerical	X coordinate of cluster
<y_pos>	Numerical	Y coordinate of cluster
<read>	Numerical	Read number. 1 can be single read or Read 2 of paired-end
<is filtered>	Y or N	Y if the read is filtered (did not pass), N otherwise
<control number>	Numerical	0 when none of the control bits are on, otherwise it is an even number. On HiSeq X and NextSeq systems, control specification is not performed and this number is always 0.
<sample number>	Numerical	Sample number from sample sheet

Figura 8. Descripción archivo formato fastq file

Un ejemplo de una entrada válida es la siguiente:

```
@SIM:1:FCX:1:15:6329:1045 1:N:0:2
TCGCACTCAACGCCCTGCATATGACAAGACAGAATC
+
<>;##=><9=AAAAAAAAAA9#:<#<;<<<????#=
```

Figura 9. Formato ejemplo Fastq file

Brevemente, podemos ver que la segunda y cuarta líneas del registro fastq son los nucleótidos y las calidades de cada ciclo de lectura.
Esta información se muestra en una orientación de 5 'a 3', que es en un principio como la ve el secuenciador. Una letra N en la secuencia se usa para indicar las bases que el secuenciador no pudo ver. La cuarta línea del registro, codifica la calidad (confianza) de la llamada en relación a la base correspondiente. El nivel de calidad se codifica siguiendo una de varias transformaciones, con la noción general de que las letras más adelante en el alfabeto ASCII visible es la de mayor calidad.
Las letras se asignan a números y los números corresponden (más comúnmente) a la fórmula -10 log10 p. En el ejemplo de a continuación, (I) corresponde a una puntuación de 40, por lo tanto, p = 0.0001. La calidad máxima de las lecturas suele estar sobre 40, una calidad media entre 20-35 y una calidad mala por debajo de 20.

```
##  !  "  #  $  %  &  '  (  )  *  +  ,  -  .  /  0  1  2  3  4  5  6  7  8  9  :  ;  <  =  >
##  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
##  ?  @  A  B  C  D  E  F  G  H  I  J
## 30 31 32 33 34 35 36 37 38 39 40 41
```

Figura 10. Calidades formato fastq

A lo largo de la fase I del proyecto se detallarán las calidades de los archivos cargados para entender en profundidad su contenido, asi como su descripción.

Además de lo anteriormente añadido, el paquete ShortRead incluye funciones para filtrar y recortar lecturas, y para generar un informe de evaluación de la calidad. Los datos se representan como objetos derivados de DNAStringSet y se manipulan fácilmente para una variedad de propósitos.

5. Fastq files QC (Quality Check)

Para entender mejor la fase cualitativa de un análisis de archivo, el proceso de decidir qué base está presente en cada ciclo de cada fragmento viene con cierta probabilidad (p) de que cometamos un error. El puntaje de calidad expresa nuestra confianza en una base particular. A mayor puntuación de calidad, mayor confianza en la lectura de esa base. Las probabilidades de llamadas básicas sin procesar se convierten en caracteres de texto para facilitar su almacenamiento en un archivo.

Para su cálculo, cada carácter tiene una relación directa de calidad y una probabilidad correspondiente. Estos cálculos pueden hacerse mediante programas especializados de obtención de calidad de las secuencias cargadas o mediante una iteración en R, como será nuestro caso. Una vez cargados y analizados dichos datos, veremos si el reporte de calidad requiere o no de un proceso de filtraje y/o recorte, previos a sus finales interpretaciones.

6. Alineamiento – Bioconductor

6.2 SAMtools

Una vez cargados los archivos y ejecutado el reporte de calidad correspondiente, se puede proceder a uno de los siguientes pasos, que es el alineamiento. Si la calidad no es correcta, se precede a un filtraje o recorte y se remueven las secuencias de los adaptadores para lograr obtener lecturas de calidad y proceder al alineamiento. En otros casos, algunos pipelines trabajan con contigs que son lecturas ensambladas en secuencias más largas. Para la generación de estos contigs existen programas ensambladores. Estos contigs se generan de novo o por alineación.

En el alcance de este proyecto y después de analizar las distintas posibles clases de paquetes en Bioconductor para su alineamiento, se ha decidido y recomendado utilizar el paquete Bowtie2.

Previo a la alineación con este paquete, asi como con su gran mayoría, debe de pasar previamente por una conversión de los ficheros a formato SAM/BAM. Dado que se parte de unos archivos con formato Fastq, éstos se van a transformar utilizando una de las muchas aplicaciones disponibles en la red. En este caso particular se ha utilizado la aplicación Galaxy, que permite un taspaso rápido y seguro a dichos formatos.

El Mapa de Alineación de Secuencias (SAM) es un formato basado en texto originalmente para almacenar secuencias biológicas alineadas con una secuencia de referencia. Se utiliza ampliamente para almacenar datos, como las secuencias de nucleótidos, generadas por las tecnologías de secuenciación de la próxima generación, y el estándar se ha ampliado para incluir secuencias no asignadas. El formato es compatible con lecturas cortas y largas (hasta 128 Mbp) producidas por diferentes plataformas de secuenciación y se utiliza para mantener datos mapeados/alineados. Los archivos tipo .BAM son la versión binaria comprimido de los archivos SAM.

Es un formato de texto delimitado por tabulación (TAB) que consiste en una sección de encabezado, que es opcional, y una sección de alineación. Si está presente, el encabezado debe ser anterior a las alineaciones.

Las líneas de encabezado comienzan con ‘@’, mientras que las líneas de alineación no lo hacen. Cada línea de alineación tiene 11 campos obligatorios para la información esencial de alineación, como la posición del mapeo y el número variable de campos opcionales para información flexible o específica del alineador.

Entre sus principales características y objetivos, destacan:

- Ser lo suficientemente flexible para almacenar toda la información de alineación generada por los varios programas de alineación.
- Simplicidad como para ser generado por programas de alineación o convertido en formatos de alineación existentes.
- Compacto en tamaño.
- Permite que la mayoría de las operaciones en la alineación funcionen en un flujo sin cargar toda la alineación en la memoria.
- Permite que el archivo sea indexado por posición genómica para todas las recuperaciones de todas las lecturas alineadas con el locus.

- El paquete SAMtools proporcionan varias utilidades para manipular alineaciones en el formato SAM, incluyendo la clasificación, fusión, indexación y generación de alineaciones en un formato por posición.

6.3 Bowtie2

Para el paso de la alineación, como bien se ha comentado en el apartado anterior, se he decidido apostar por el paquete Bowtie2, dada, básicamente su mayor rapidez, teniendo en cuenta la longitud de la secuencia. No obstante, en unos inicios surgió la duda de la utilización del paquete Burrows-Wheeler Aligner. Tras una comparación, y viendo que ambos paquetes eran parecidos (lecturas de hasta 100bp o mayores, según el algoritmo utilizado y para un mapeo contra genomas largos, como es el Genoma Humano), se decidió aposat por el segundo.[37]

Bowtie2 es una herramienta ultrarrápida y eficiente en memoria para alinear las lecturas de secuencia a largas secuencias de referencia. Es particularmente bueno para alinear lecturas de aproximadamente 50bp hasta 100bp o 1000bp, y particularemente aconsejable para alineaciones con genomas de referencia relativamente largos. Las diferencias mas destacadas con su “antecesor” son:

- Para lecturas de más de 50 bp, Bowtie 2 es generalmente más rápido, más sensible y usa menos memoria que Bowtie 1. Para lecturas relativamente cortas (por ejemplo, menos de 50 bp) Bowtie 1 a veces es más rápido y / o más sensible.
- Bowtie 2 admite la alineación con huecos con penalizaciones por huecos afines. El número de espacios y las longitudes de los espacios no están restringidos, excepto por el esquema de puntuación configurable. Bowtie 1 encuentra alineaciones sin tapar.
- No hay un límite superior en la longitud de lectura en Bowtie 2. Bowtie 1 tenía un límite superior de alrededor de 1000 pb.
- Bowtie 2 permite que las alineaciones se superpongan con caracteres ambiguos (por ejemplo, Ns) en la referencia. Bowtie 1 no lo hace.
- Bowtie 2 elimina la noción de “estrato” de alineación de Bowtie 1 y su distinción entre los modos “tipo Maq” y “extremo a extremo”. En Bowtie 2, todas las alineaciones se encuentran en un espectro continuo de puntuaciones de alineación donde el esquema de puntuación, similar a Needleman-Wunsch y Smith-Waterman.
- La alineación de extremo emparejado de Bowtie 2 es más flexible. P.ej. Para parejas que no se alinean de manera emparejada, Bowtie 2 intenta encontrar alineaciones no emparejadas para cada compañero.
- Bowtie 2 reporta un espectro de cualidades de mapeo, en contraste para Bowtie 1 que reporta 0 o alto.
- Bowtie 2 no alinea las lecturas del espacio de color.

Una ventaja es que Bowtie2 indexa el genoma con un índice de FM para mantener su un estado de gasto de memoria bajo: en el caso del genoma humano (como nuestro alcance), su huella de memoria suele ser de alrededor de 3.2 GB. Bowtie2 es compatible con los modos de alineación de extremos separados, locales y emparejados.

El equipo para la realización de la alineación es un PC con procesador Intel core i5 64 bits, 16GB de RAM y 250GB de SDD.

7. Desarrollo del trabajo

7.1 Desarrollo Fase I

El pipeline desarrollado a lo largo de este proyecto es un estudio centrado en analizar unos archivos en formato fastq y contiene varias fases. Destacar, que dichos archivos en bruto provienen de un secuenciador de Illumina, modelo MiSeq y contienen datos sobre un panel de genes sobre cáncer de pulmón. El estudio se ha realizado sobre uno de éstos, de forma que representa un paso a paso ejemplificado para la ejecución completa de dicho estudio. No obstante, en los anexos, se nombrarán todos los archivos correspondientes al panel, asi como su posible ejecución en cada uno de ellos. Este estudio se puede extender a cualquier tipo de archivo, tanto los mostrados en el anexo o cualquier otro que parta de una extensión tipo fastq. Para su tratamiento y rapidez, entre otros, dependerá del peso del archivo y de la capacidad de procesamiento de datos del ordenador. Dichos archivos complementarios serán también colgados en una nube, para su consulta.

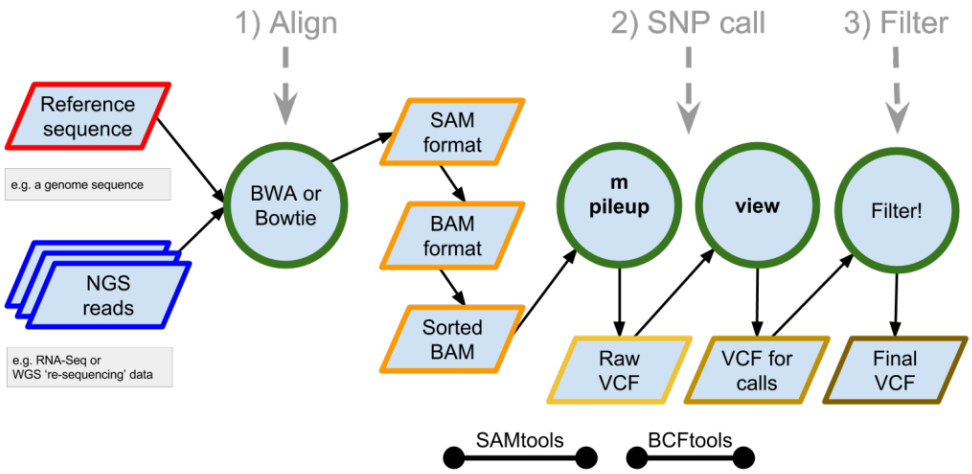


Figura 11. Pipeline Overview

Cada uno de estos archivos, viene con su denominado Índice. Este índice o “barcode”, es otro archivo de formato fastq, por lo tanto, cumple el mismo formato de cualquier otro archivo con dicha extensión (normalmente más corto, unos 8bp), pero lo más interesante es la razón de su uso. Su principal objetivo, como bien indica el nombre, es identifcar una muestra específica dentro de una combinación de ellas. Es decir, estos índices permiten mezclar muestras de diferentes pacientes (por ejemplo), en un mismo run/análisis del secuenciador. Estos identificadores, están enlazados a cada una de las extracciones, tantas muestras a secuenciar, tantos identificadores, y permite que cada una de los archivos, tenga su correspondiente libreria e independencia.
Gracias a estos identificadores, al terminar la secuenciación, el software de postprocesado, estará capacitado para identificar y diferenciar cada una de esas muestras, para las siguientes acciones, como puede ser un filtraje, recorte, etc.

Los datos en bruto derivados de los secuenciadores de Illumina están compuestos por imágenes grabadas ópticamente mediante 4 colores fluorescentes (ATGC)después de cada ciclo de secuenciación. De hecho, dichos scanners, contienen unos lásers de clase 4, cada uno correspondiente a una de dichas bases.
Por otra parte, Illumina utiliza una medida de control de calidad denominada filtrado de calidad de "castidad" para aceptar o rechazar un grupo individual que se aplica después de los primeros 25 ciclos de una secuenciación. Específicamente, durante los

primeros 25 ciclos, se registra la base de intensidad fluorescente más alta incorporada en un grupo, y su intensidad se compara con la siguiente base fluorescente más alta registrada para el grupo. Esta información se utiliza para calcular la relación de filtro de castidad que se obtiene al tomar la fluorescencia de la base de intensidad fluorescente más alta y dividirla por la fluorescencia de la misma base de intensidad fluorescente más alta más la fluorescencia de la siguiente base de intensidad fluorescente más alta. Una proporción de 0,6 o mayor se considera una proporción de "aprobación". De este modo, y en el objeto de nuestro estudio, veremos que obtendremos datos de alta calidad.
Un "error" de grupo se define cuando dos o más eventos de incorporación de base tienen valores de relación de castidad inferiores a 0,6 en los primeros 25 ciclos de una secuenciación. Este filtro de calidad elimina los clusters superpuestos y de baja intensidad que a menudo representan altas densidades de clúster, lo que provoca el solapamiento del clúster y señales de secuencia mixtas. El modo de error dominante para la secuenciación de Illumina es la categoría de sustituciones de un solo nucleótido.

Los archivos utilizados son archivos de longitud 76 nucleótidos y en formato “paired end”. Esto se refiero a los dos extremos de la misma molécula de DNA. Por lo que se puede secuenciar un extremo (5’ → 3’) y luego girarlo y secuenciar el otro extremo (3’ → 5’). Las dos secuencias que se obtendrán, serán “lecturas finales emparejadas”.

En el primer paso, leemos los datos dentro de la ruta y elegimos el primero.

```
```{r}
library(ShortRead)
showMethods(readFastq)
getwd()
fastqDir <-
file.path("C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pip
eline/data/File")
fastqFiles <- dir(fastqDir, full=TRUE)
fastqFiles

#Dentro de la ubicación de los archivos, decidimos elegir por ejemplo
el primer de los archivos

fq <- readFastq(fastqFiles)
sequenceOfReads <- sread(fq)
class(sequenceOfReads)
```
```

El chunk nos devuelve el nombre del archivo y el tipo de archivo.

```
[1]
"C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline/data
/File/15-0991_S1_L001_R1_001.fastq"
[1] "DNAStringSet"
```

Dentro del archivo, analizamos ancho de lectura y se representa por ejemplo las 3 primeras DNAStringSet. Los datos representados son objetos de clase ShortReadQ:

```
```{r}
readLengths <- width(fq)
readLengths[1:3]
head(sread(fq), 3)
```
```

```
[1] 76 76 76
  A DNAStringSet instance of length 3
    width seq
```

```
[1]    76
CTTAAAACTGATTTTACATGGTACATGAAACAAGGCAAATAACTGCGATTTTTTTCTTCCTTCTGCTCCT
TCCCCT
[2]    76
CTGTGCTTGCACCTGACATCCTCGTGCTCCTGTTTACCTTGTTTCATTTTATCTTTTTTTTTTTTTTTTT
TTCTCT
[3]    76
CATTTCTTTGGTAAAAAGCTGAAGCCAGAGAAGTGTCTTTGAATCCTTTAATCTCCCTTCTCTTTTTCCT
CCCTCC
```

Como se ha comentado, vemos claramente que el ancho de las lecturas es de 76 bases.

De forma genérica, el paquetes ShortRead inclue una función para genera un informe de control de calidad simple. En nuestro estudio, para no alargar la secuencia, vamos a utilizar un sólo archivo, aunque realmente la función se podría ejecutar sobre todos los archivos. Dicha función accepta un archivo tipo FastQ file y devuelve un objeto.

```
```{r}
qa <-
qa("C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline/d
ata/File/15-0991_S1_L001_R1_001.fastq")
qa
```
class: FastqQA(10)
QA elements (access with qa[["elt"]]):
  readCounts: data.frame(1 3)
  baseCalls: data.frame(1 5)
  readQualityScore: data.frame(512 4)
  baseQuality: data.frame(95 3)
  alignQuality: data.frame(1 3)
  frequentSequences: data.frame(50 4)
  sequenceDistribution: data.frame(373 4)
  perCycle: list(2)
    baseCall: data.frame(304 4)
    quality: data.frame(1950 5)
  perTile: list(2)
    readCounts: data.frame(0 4)
    medianReadQualityScore: data.frame(0 4)
  adapterContamination: data.frame(1 1)
```

Obtenemos varios datos de calidad sobre el archivo:

```
ShortRead:::.ppnCount(qa[["readCounts"]])
    read      filter   aligned
1   2228315
  ShortRead:::.plotReadCount(qa)
```

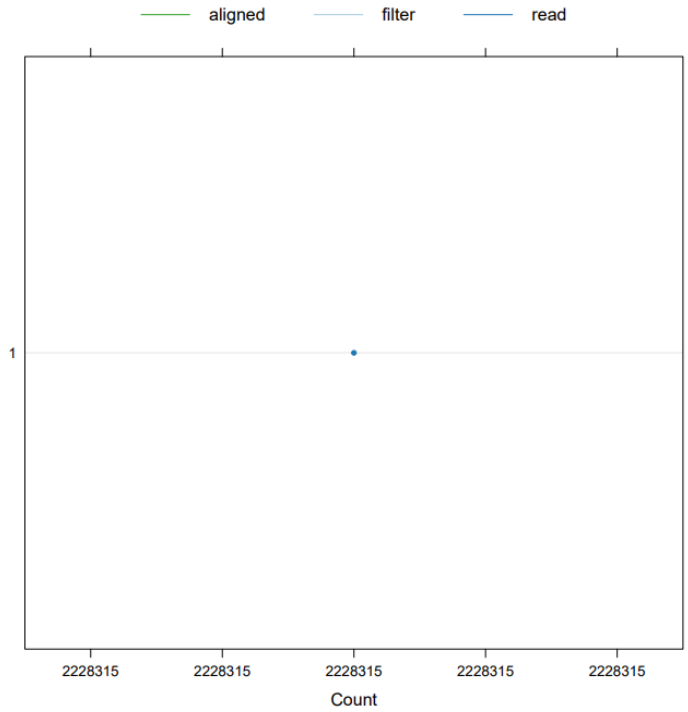


Figura 12. Lectura sobre archivo fastq

Y a continuación se muestra la frecuencia de cada una de las bases sobre lectura.
Las frecuencias base deben reflejar con precisión las frecuencias de las regiones secuenciadas.

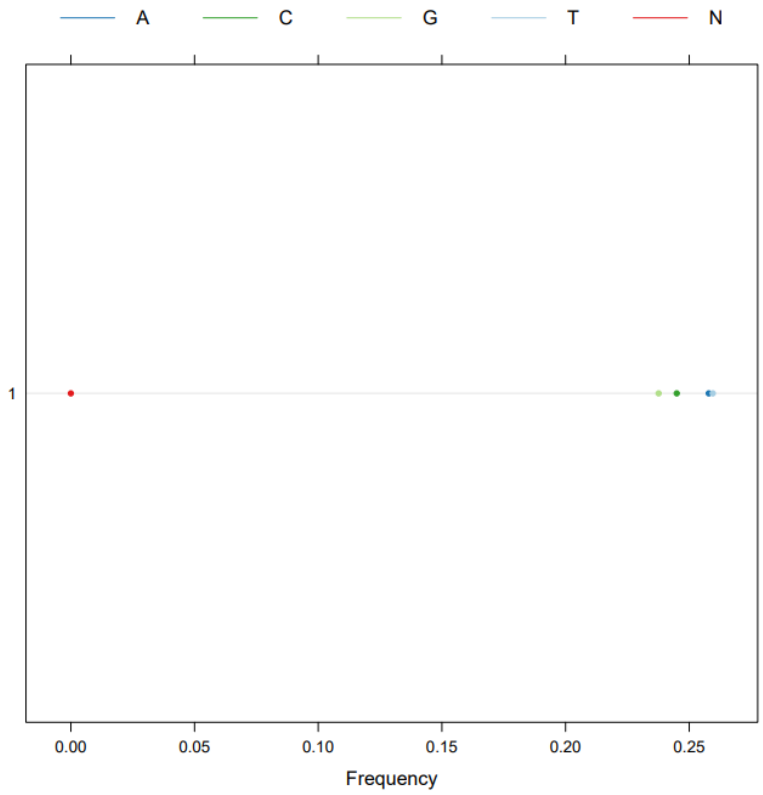


Figura 13. Frecuencia de nucleótidos en lectura

Para el estudio de calidad, en el siguiente gráfico se muestran la lectura de carácter general.. Los carriles con lecturas consistentemente de buena calidad tienen picos fuertes a la derecha del panel.

```
df <- qa[["readQualityScore"]]
  ShortRead:::.plotReadQuality(df[df$type=="read",])
```

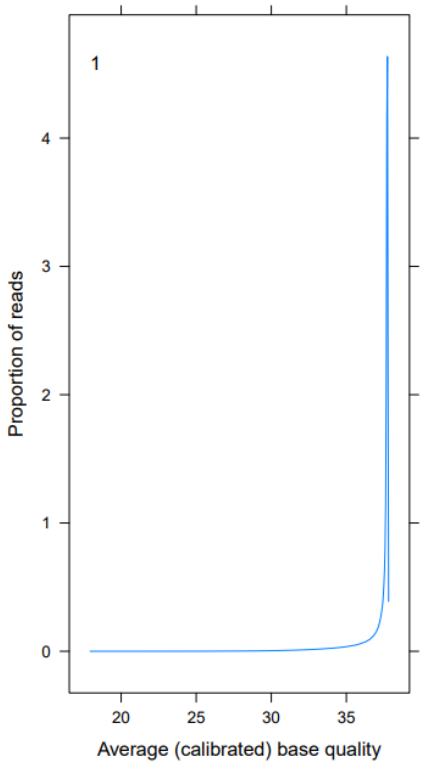


Figura 14. Calidad media de las lectuars sobre las bases

```
  df <- qa[["sequenceDistribution"]]
  ShortRead:::.plotReadOccurrences(df[df$type=="read",], cex=.5)
```

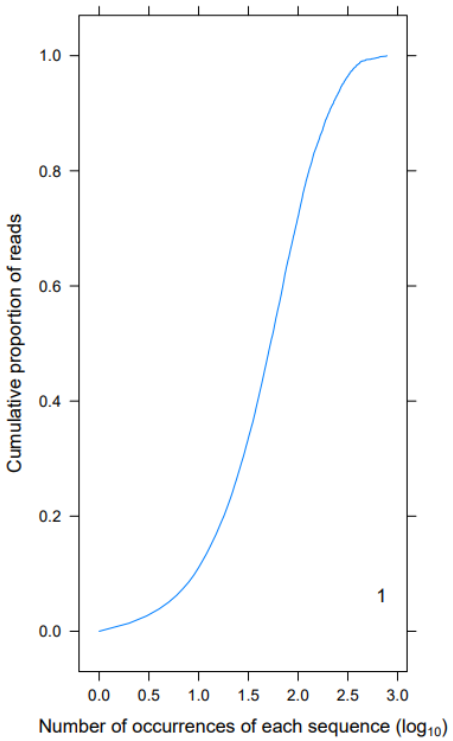


Figura 15. Distribución de las lecturas

Estas curvas muestran cómo se distribuye la cobertura entre las lecturas. Idealmente, la proporción acumulativa de lecturas hará una transición brusca de baja a alta.

Las partes a la izquierda de la transición podrían corresponder aproximadamente a errores de secuenciación o procesamiento de muestras, y corresponder a lecturas que se representan con poca frecuencia. 10-15%; de lecturas en un típico "control" del Genome Analyzer caen en esta categoría.

Las partes a la derecha de la transición representan lecturas que están sobre representadas en comparación con las expectativas. Estos pueden incluir secuencias de cebadores o adaptadores secuenciados inadvertidamente, artefactos de secuenciación o llamadores de bases, o características de la muestra de ADN (regiones altamente repetidas) que no se eliminaron adecuadamente durante la preparación de la muestra. Alrededor del 5% de las lecturas de "control" del Analizador del Genoma se encuentran en esta categoría.

Las transiciones amplias de baja a alta proporción acumulativa de lecturas pueden reflejar un sesgo de secuenciación o características (quizás intencionales) de la preparación de la muestra, lo que da como resultado una cobertura no uniforme. la transición es aproximadamente 5 veces más ancha de lo esperado, a partir de un muestreo uniforme en el carril de control del Analizador del Genoma.

Las lecturas duplicadas comunes pueden proporcionar pistas sobre el origen de las secuencias sobre representadas. Algunas de estas lecturas están filtradas por los algoritmos de alineación; otras lecturas duplicadas pueden apuntar a problemas de preparación de muestra.

```
ShortRead:::.freqSequences(qa, "read")
```

Sequence	count	lane
GCCCCTCCCGGGAAGGTGCCGTCTCCT CCGGCCCCTCGGGTCCCTGCTCTGTCA CTGACTGCTGTGACCCACTCTG	781	1
GTCCCCGGCCTGGCAGGGCGCCCTGGA GTGGGAGGAAGAGGTAACCACAGGGGG GCTGGAGCTGGCCTCGGACTTG	771	1
CACCAGAGCAGCTGCAGTTCCCTGAGG AGCCCCTGATTCTGCACCTCAGCCCCG TGTGTATCCTCCTGGCTGATCA	673	1
AGCACATCTGCCACCACTTGCACTGCG GTCTGGCTAACCACATGAGCATGGCCA CTGATGAGGTGGATGGAGGGTG	655	1
GGCCACAGCGTCTGCTCCACCTCCAGC TTGTACCTGCAGGATCTGAGCGCCGCC GCCTCAGAGTGCATCGACCCCT	648	1
ACTCGGTGCAGCCGTATTTCTACTGCG ACGAGGAGGAGAACTTCTACCAGCAGC AGCAGCAGAGCGAGCTGCAGCC	617	1
TAGCCATGGCAAGGTCCCCATGACAAG TGCCTCCTCTCCCATCTTCCACAGAGC GGTCCCCTCACCGGCCCATCCT	598	1
CCCAGCATCCAGTGGGGGAGTGAAGGG CAATGAAGGGTACATCCTGGGGTCCAG GCCAACCTGGCTCTCTGCTCGG	575	1
CTCTTTAGCCATGGCAAGGTCCCCATG ACAAGTGCCTCCTCTCCCATCTTCCAC AGAGCGGTCCCCTCACCGGCCC	543	1
CAGGGGCATGGACAGGTCCAGGTACTC CTGTGATGGGCGAGAGGAAGCAGCGAT GGGCCGGGCCCCTCCTCCCTGC	539	1
GGGAACCTTTGCTCTTTTTCAGTGACC AAAATCATCTGTGCCCAGCAGTGCTCC GGGCGCTGCCGTGGCAAGTCCC	483	1
GTTGGGGGCCGAGCCCTCACTGCTGTT CACCCAGGGCAGCCCGCCCACCCCGGT ATCCAGCCATGTCAGCCCTCTG	480	1
TTCCTGAACAGTGAGCACAAGGAGATC CGCATGGAGGCTGCCCGCACCTGCTCC CGCCTGCTCACACCCTCCATCC	475	1
CCTGGAACAGCCCCCAACTGGGCCTGG AGACAGCAGCCCCCCCGAGCACCCTGT CCTCCACCCCTCAGAGCCTTCA	474	1
CTGGGATTATATTCCGACTGCAAGCAT AGGCAAGTTCAGGGATCTATCTGTTCT CTCCAAAGACAGAGGGAAGTGG	457	1

CCACTCTCTGACCCAATTTGTTCGCCA CTGTGCGGATCATTTCCTGAACAGTGA GCACAAGGAGATCCGCATGGAG	440	1
CCCTGTGCAGCTGTGGGTTGATTCCAC ACCCCCGCCCGGCACCCGCGTCCGCGC CATGGCCATCTACAAGCAGTCA	433	1
TGTGCTCCCCCGAGGGCTGCTGGGGCC CGGAGCCCAGGGACTGCGTCTCTTGCC GGAATGTCAGCCGAGGCAGGGA	432	1
CTGCAGCCAGAAAGACTCCAGGGGACT CATTACTCTGTGCAGTCAGACATCTGG AGCATGGGACTGTCTCTGGTAG	426	1
TCCAGGATACTCGGCACAGCCTGGAGG CTCTGAGTAGCACATGCAGACCTGCAG GCCAGGCTCCCCAGAAAGGCAC	421	1

Figura 16. Frecuencia de secuencias

Lecturas duplicadas comunes después del filtrado:

```
  ShortRead:::.freqSequences(qa, "filtered")
```

```
NA
```

Las lecturas duplicadas alineadas comunes no se contemplan:

```
  ShortRead:::.freqSequences(qa, "aligned")
```

```
NA
```

La llamada básica por ciclo generalmente debe ser aproximadamente uniforme en todos los ciclos. Los resultados de la línea 'control' del Analizador de Genoma a menudo muestran una deline en A y aumentan en T a medida que avanzan los ciclos. Este es probablemente un artefacto de la tecnología subyacente.

```
perCycle <- qa[["perCycle"]]
  ShortRead:::.plotCycleBaseCall(perCycle$baseCall)
```

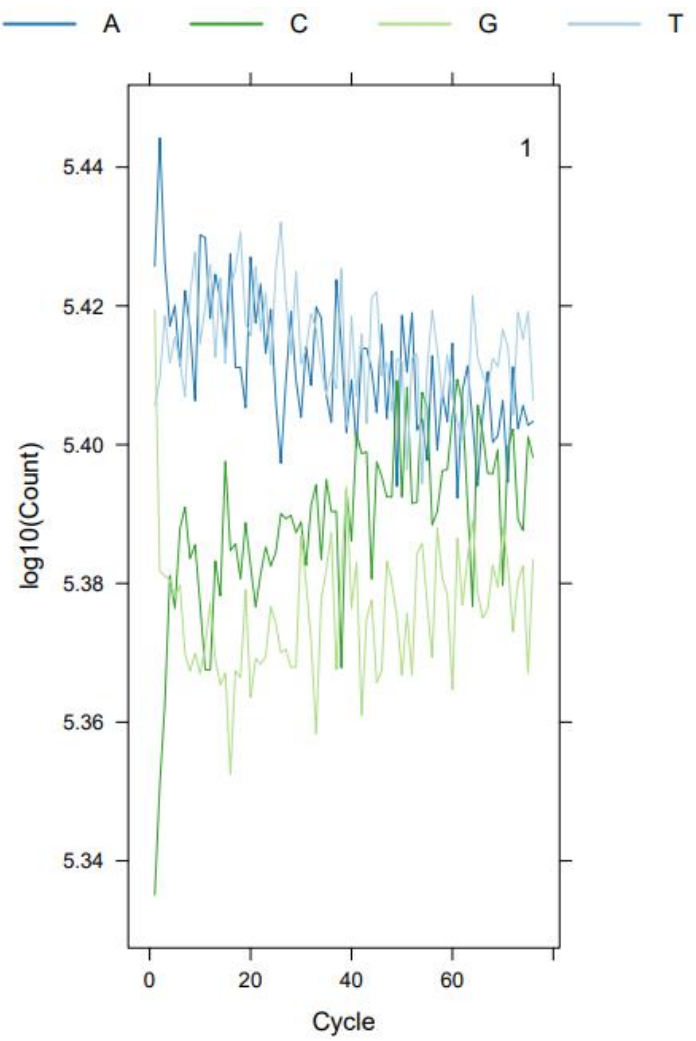


Figura 17. Llamadas base específicas de ciclo y calidad de lectura

Puntaje de calidad por ciclo. Los puntajes de calidad reportados están "calibrados", es decir, incorporan ajustes similares a los de la secuencia de alineación. Estos suelen declinar con el ciclo, de manera acelerada. Las transiciones abruptas en la calidad entre los ciclos hacia el final de la lectura pueden producirse cuando solo algunos de los ciclos se usan para la alineación: los ciclos incluidos en la alineación se calibran más efectivamente que las lecturas excluidas de la alineación.

Las líneas rojizas son cuartiles (sólido: mediano, punteado: 25, 75), la línea verde es la media. El sombreado es proporcional al número de lecturas.

```
  perCycle <- qa[["perCycle"]]
  ShortRead:::.plotCycleQuality(perCycle$quality)
```

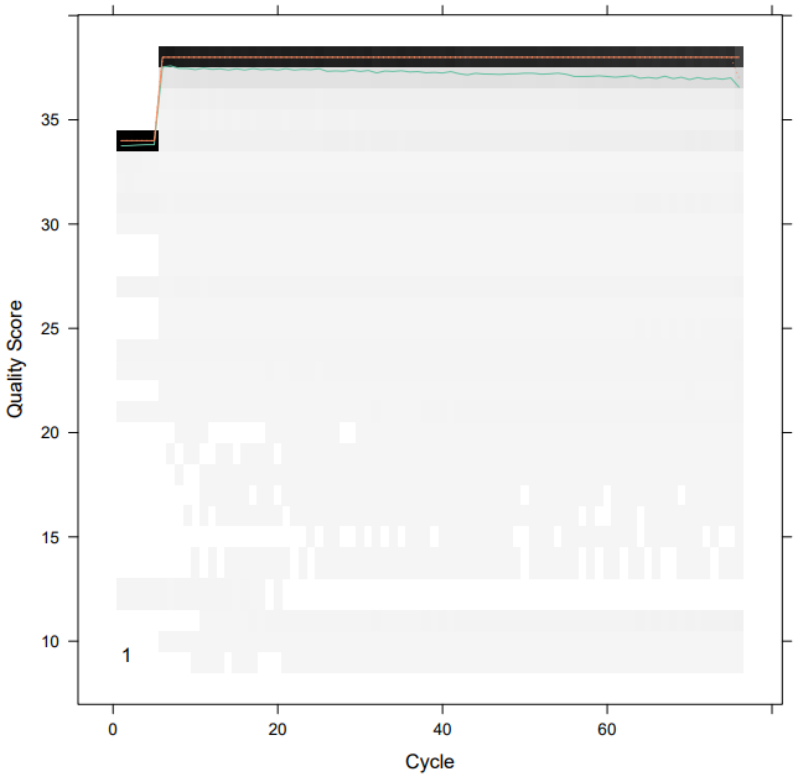


Figura 18. Llamadas por ciclo de base. Forward

Contaminación:

La contaminación del adaptador se define aquí como secuencias no genéticas adjuntas en uno o ambos extremos de las lecturas. La medida de 'contaminación' es el número de lecturas con una coincidencia derecha o izquierda con la secuencia del adaptador sobre el número total de lecturas. Las tasas de desajuste son 10% a la izquierda y 20% a la derecha con una superposición mínima de 10 nt.

```
ShortRead:::.ppnCount(qa[["adapterContamination"]])
    contamination
1   NA
```

En particular, analizando los resultados, se puede apreciar claramente en los gráficos de calidad que no existe apenas contaminación, sospesado además por el código anterior. Esto es debido al tipo de archivo utilizado, en nuestro scope, estos archivos provienen de un panel de genes de cancer de pulmón, sin contaminación y listos para poder alinearlos con el Genoma de referencia. En el estudio se hace un detalle sobre la posibilidad de la interpretación cualitativa debido a que dicho pipeline puede ser ampliado a otros archivos fastq files, previamente sin limpiar y en los cuales se deberá hacer un paso de filtering and trimming previa a la alineación.

Al ser archivos tipo paired end read, miramos la calidad de ambas de las lecturas, para poder ver la posible contaminación entre el anverso y el reverso de las consiguientes cadenas.

```
```{r}
library(dada2)
packageVersion("dada2")
path <-
"C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline/data
/File"
list.files(path)
```
```

```
fnFs <- sort(list.files(path, pattern="15-0991_S1_L001_R1_001.fastq",
full.names = TRUE))

fnRs <- sort(list.files(path, pattern="15-0991_S1_L001_R2_001.fastq",
full.names = TRUE))

# Extract sample names, assuming filenames have format:
SAMPLENAME_XXX.fastq

sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`, 1)
sample.names2 <- sapply(strsplit(basename(fnRs), "_"), `[`, 1)

plotQualityProfile(fnFs[1:2])
plotQualityProfile(fnRs[1:2])

```
```

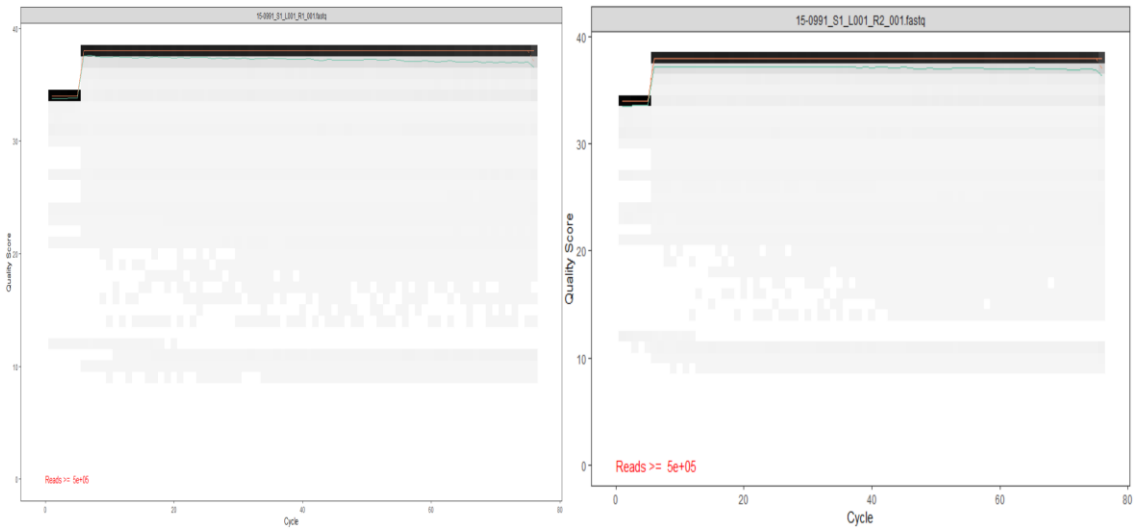


Figura 19. Anverso y reverso Strand Quality Contamination

En escala de grises, se plasma la frecuencia del resultado de cada muestra en la posición de cada base.

La calidad media se muestra en forma de línea verde y los quartiles de distribución de resultados sobre la línea de color naranja. En color roja, se muestra la proporción de lecuras que se extienden almenos sobre esa posición. En particular, como los fastq files se secuencian sobre plataforma Illumina, éstas lecturas son típicamsente todas de la misma longitud, por lo tanto, de ahí que me muestre una línea de color rojo, completamente plana.

Dicha lectura tiene buena calidad. Generalmente se advierte un proceso de triming sobre los últimos nucleótidos para tener un mayor control sobre errores y tenerlos bien controlados. Estos perfiles cualitativos no sugieren que es necesario un trimming.

Si en alguna de las lecturas, sufre una pérdida en cuanto a calidad, por ejemplo en el fin de la secuencia, el paquete DADA2 de Bioconductor incorpora información de calidad en su modelo de error, lo que hace que el algoritmo sea robusto a una secuencia de calidad inferior, pero el recorte a medida que se desplome la calidad promedio mejorará la sensibilidad del algoritmo a variantes de secuencia raras. Sobre la base de estos perfiles, truncaremos las lecturas inversas en la posición 160 donde se bloquea la distribución de calidad. Cogeremos una anverso y un reverso de una de las muestras.

```{r}
library(Rqc)
```
```

```
qa<-rqc(path =
"C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline/data
/File", pattern = ".fastq")

```

El paquete Rqc acepta el formato de archivo R Markdown como archivo de plantilla para generar informes personalizados. Markdown es un lenguaje de marcado para el desarrollo web. Los archivos R Markdown son archivos Markdown normales con códigos R. Cada fragmento de código se ejecuta durante la compilación realizada por el paquete knitr. Knitr toma el archivo Markdown R y genera un archivo Markdown combinado con los códigos R y sus salidas, como texto, tablas y figuras. Rqc usa el archivo Result Markdown para generar el informe final en formatos HTML o PDF. El archivo de origen del informe predeterminado de Rqc es una buena referencia para escribir nuevos informes de plantilla. El código de ejecución a continuación devuelve la ruta del archivo del sistema a este archivo fuente.

Los datos de resultados de Rqc están disponibles dentro de los archivos de plantilla a través del objeto rqcResultSet. Este objeto es una lista de estad?sticas resumidas sobre los archivos de entrada y es utilizado por todas las funciones de acceso y gráficos proporcionados por el paquete Rqc. La función rqcReport toma la ruta del archivo de plantilla como argumento y genera informes personalizados.

Para cada gráfico generado por Rqc, hay una función que da forma a los datos de manera apropiada. La información configurada se utiliza para producir la trama final.

Calidad media de lectura de distribución de archivos:

Esta gráfica describe una visión general de la distribución de calidad media por lectura de todos los archivos

```
rqcReadQualityBoxPlot (rqcResultSet)
```

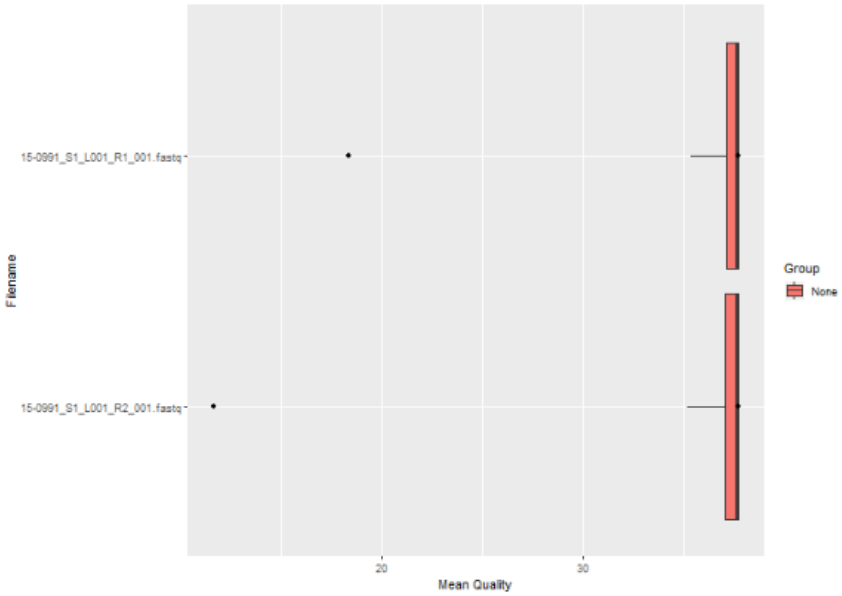


Figura 20. Distribución de calidad media por lectura. Anverso y reverso

Average Quality:

Esta gráfica describe el patrón de calidad promedio al mostrar en los umbrales de calidad del eje X y en el eje Y el porcentaje de lecturas que exceden ese nivel de calidad.

```
groups <- unique(perFileInformation(rqcResultSet)$group)
for (group in groups) {
  rqcResultSet.sub <- subsetByGroup(rqcResultSet, group)
  print(rqcReadQualityPlot(rqcResultSet.sub) +
          ggtitle(paste("Group", group)))
}
```

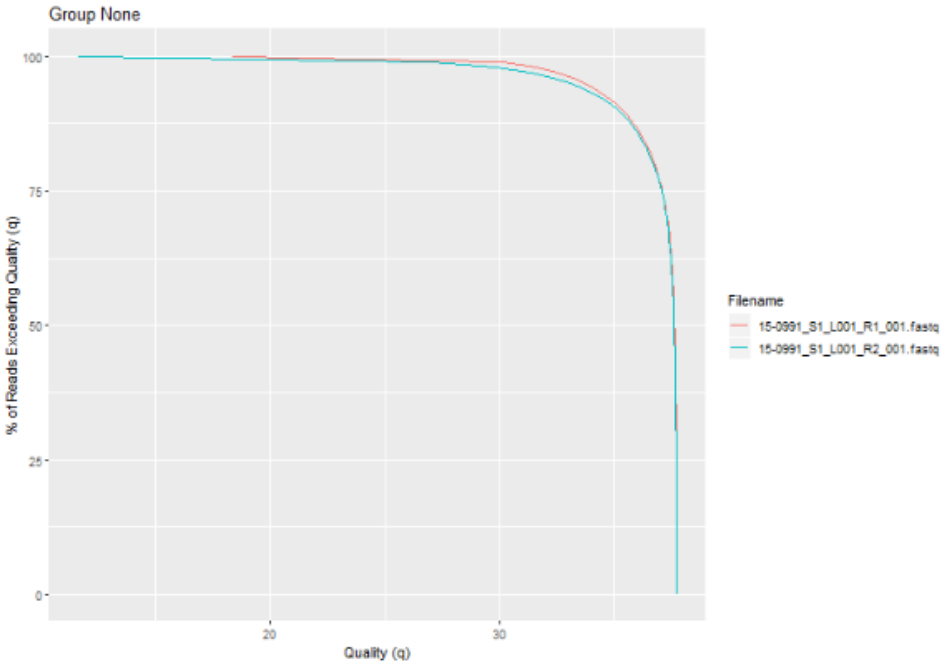


Figura 21. Patrón de calidad

Calidad media específica del ciclo:

Esta gráfica describe los puntajes promedio de calidad para cada ciclo de secuenciación.

```
for (group in groups) {
  rqcResultSet.sub <- subsetByGroup(rqcResultSet, group)
  print(rqcCycleAverageQualityPlot(rqcResultSet.sub) +
          ggtitle(paste("Group", group)))
}
```

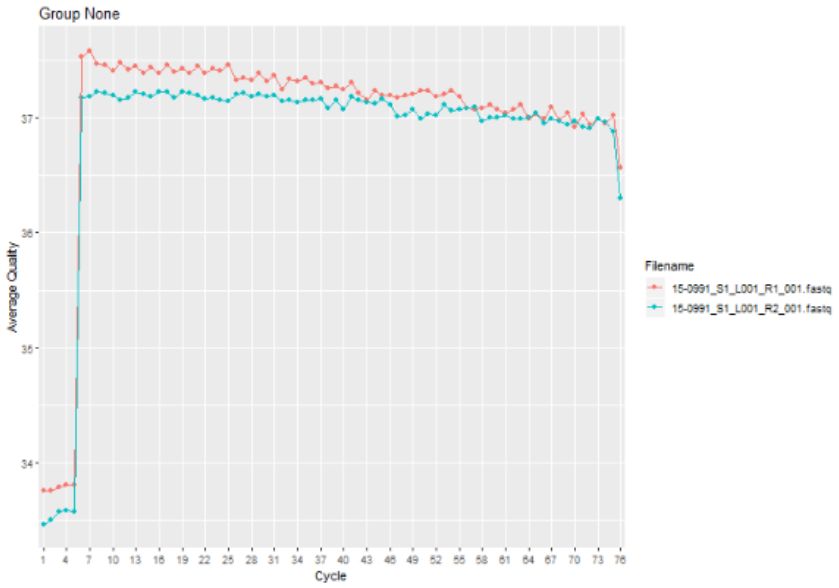


Figura 22. Calidad media específica del ciclo

Frecuencia de lectura:

Esta gráfica muestra la proporción de lecturas que aparecieron muchas veces. A menor proporción a lo largo de las ocurrencias, mas estable y de más calidad es la muestra.

```
for (group in groups) {
  rqcResultSet.sub <- subsetByGroup(rqcResultSet, group)
  print(rqcReadFrequencyPlot(rqcResultSet.sub) +
          ggtitle(paste("Group", group)))
}
```

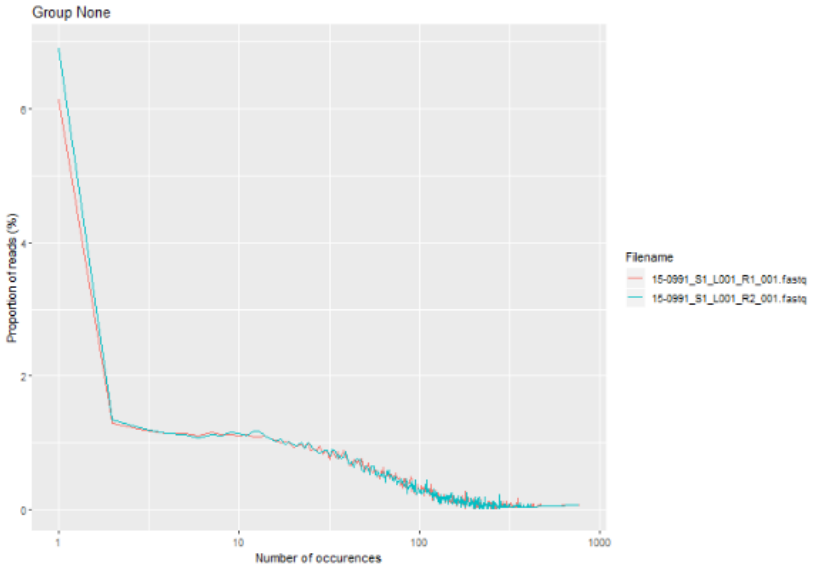


Figura 23. Frecuencia de lectura

Contenido en guanina y citosina (GC) por secuencia específico del ciclo:

Gráfico de líneas que muestra el contenido promedio de GC para cada ciclo de secuenciación. En el ejemplo hay una variación máxima de 4% en GC, por lo que se puede denotar que existe una buena especificidad y por consiguiente, su calidad tenderá a ser óptima.

```
for (group in groups) {
  rqcResultSet.sub <- subsetByGroup(rqcResultSet, group)
  print(rqcCycleGCPlot(rqcResultSet.sub) +
          ggtitle(paste("Group", group)))
}
```

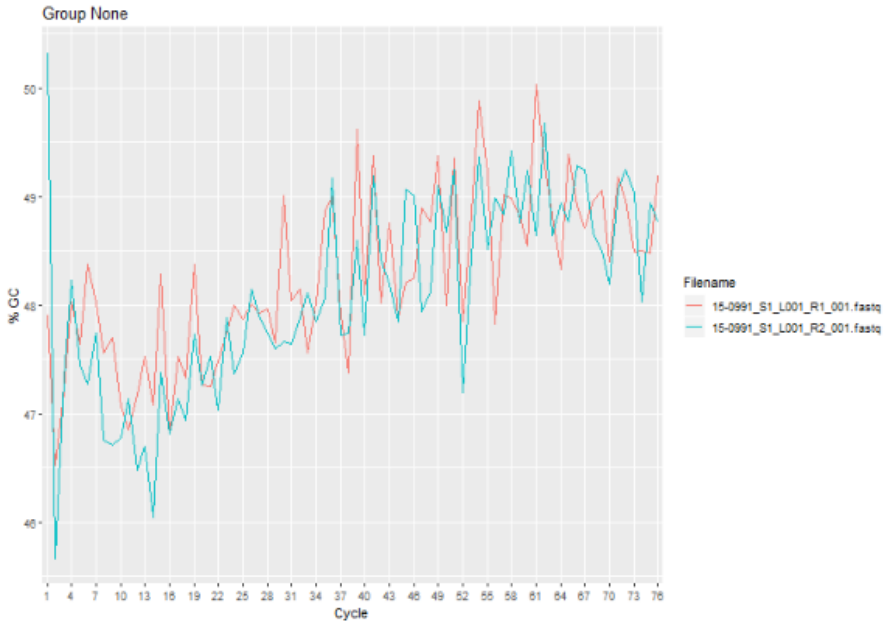


Figura 24. Contenido de GC específico del ciclo

Distribución de calidad específica del ciclo:

Gráfico de barras que muestra la proporción de llamadas de calidad por ciclo. Los colores se presentan en un degradado rojo-azul, donde el rojo identifica las llamadas de menor calidad. Se prefiere esta visualización porque es más limpia que las gráficas de caja que se describen a continuación.

```
for(pair in pairs) {
  rqcResultSet.sub <- subsetByPair(rqcResultSet, pair)
  print(rqcCycleQualityPlot(rqcResultSet.sub))
}
```

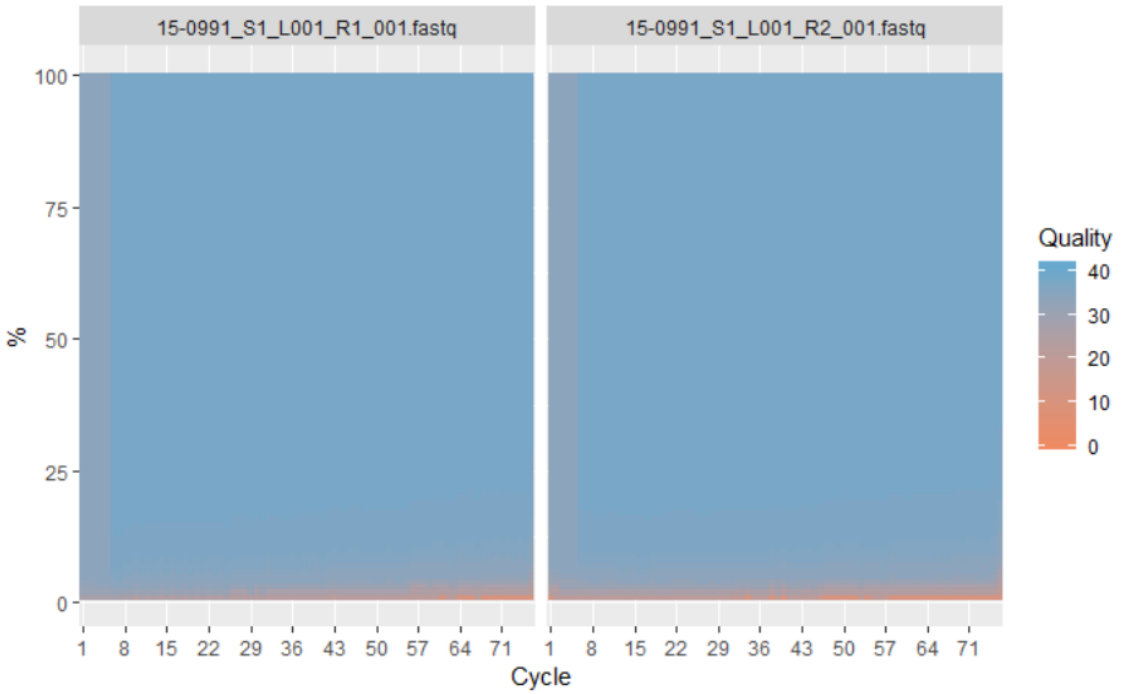


Figura 25. Contenido de GC específico del ciclo

<u>Distribución de calidad específica del ciclo - Gráfica de caja:</u>

Diagramas de caja que describen patrones empíricos de distribución de calidad en cada ciclo de secuenciación. Siguiendo el guión, claramente se ve como todas las calidades son óptimas.

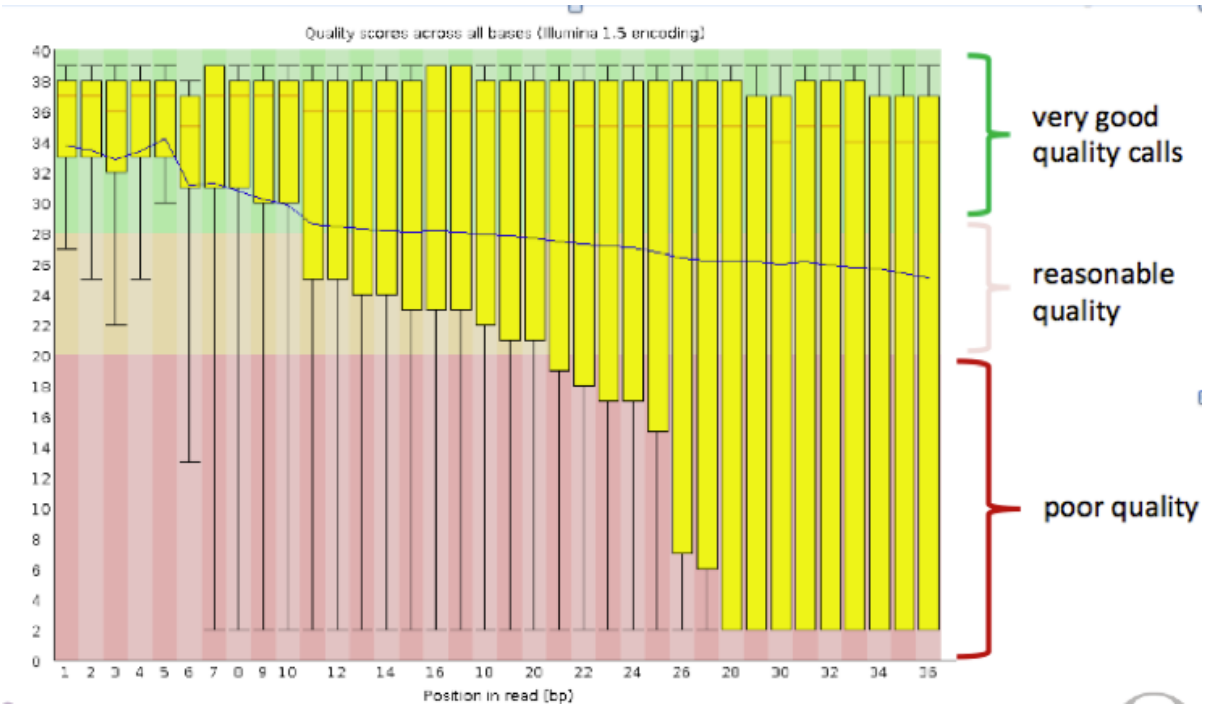


Figura 26. Patrón de referencia

```
for(pair in pairs) {
  rqcResultSet.sub <- subsetByPair(rqcResultSet, pair)
  print(rqcCycleQualityBoxPlot(rqcResultSet.sub))
}
```

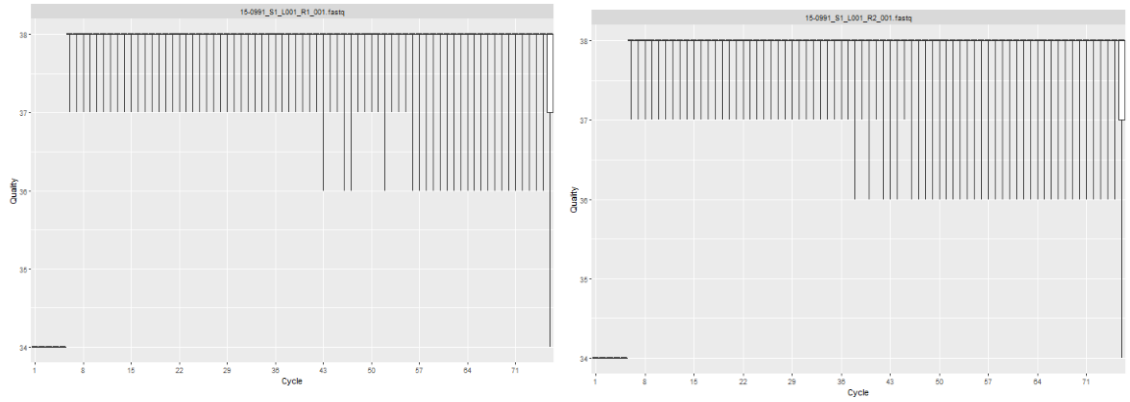


Figura 27. Resultados

Proporción de llamada base específica del ciclo:

Este gráfico de barras apiladas describe la proporción de cada nucleótido requerido para cada ciclo de secuenciación.

```
for(pair in pairs) {
  rqcResultSet.sub <- subsetByPair(rqcResultSet, pair)
  print(rqcCycleBaseCallsPlot(rqcResultSet.sub))
}
```

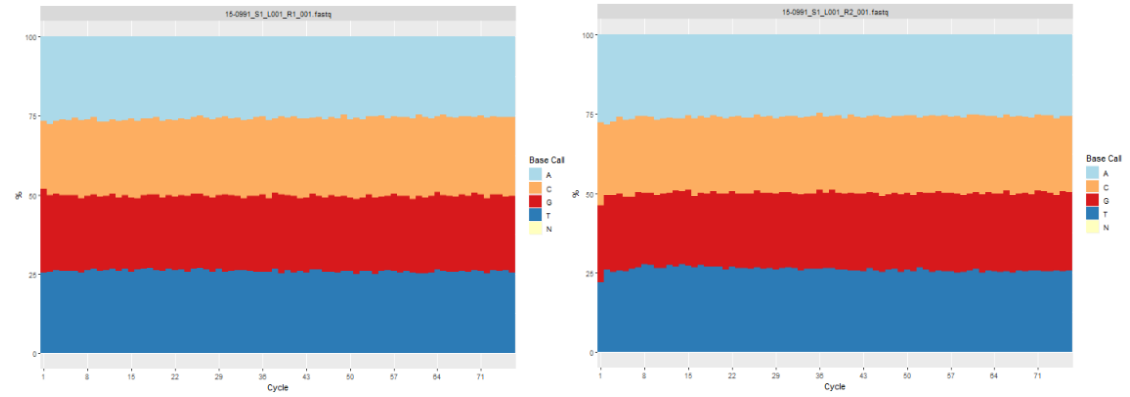


Figura 28. Llamada por base de ciclo

La gráfica de líneas que se muestra a continuación contiene la misma información que la gráfica de arriba. Sin embargo, algunos pueden encontrar esto más fácil de leer al comparar las tasas de llamadas para cada uno de los nucleótidos.

```
for(pair in pairs) {
  rqcResultSet.sub <- subsetByPair(rqcResultSet, pair)
  print(rqcCycleBaseCallsLinePlot(rqcResultSet.sub))
}
```

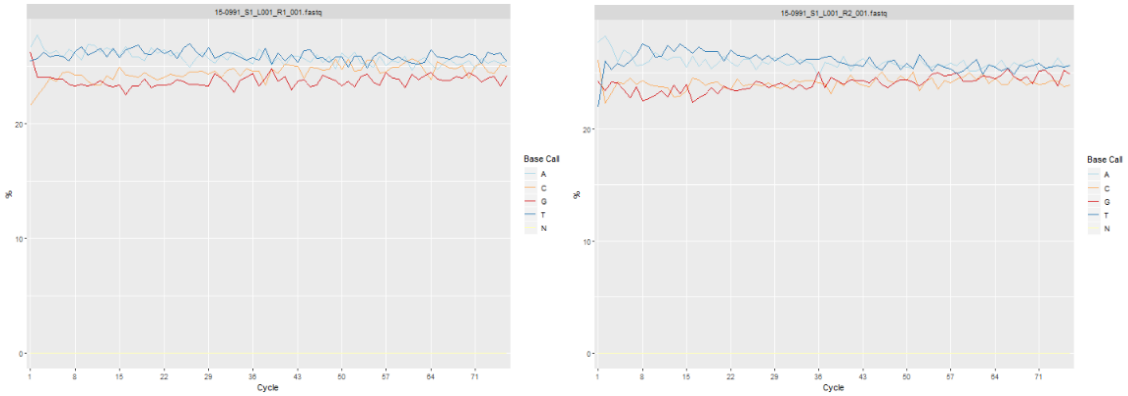


Figura 29. Llamada por base de ciclo

Por otro lado, como se comentó en la explicación de calidad en la definición de los fastq files, si analizamos análogamente la calidad de los archivos, éstos los ejecuta a partira partir de la función quality. Con esta función y teniendo en cuenta el abecedario de valores, seremos capaces de obtener una primera aproximación sobre la calidad de nuestra muestra. En nuestro caso y como se ha comentado anteriormente, no es necesario pasar por dicho proceso, pero a modo de ejemplo se va a ejemplificar las 10 primeras cadenas escogidas como muestra, se muestra el alfabeto de calidad correspondiente a cada uno de los archivos.

```{r}
head(quality(fq), 3)
```

```
class: FastqQuality
quality:
  A BStringSet instance of length 3
    width seq
[1]    76
@BCCCGAEFC9FFGGFFDGGFF9@,E,,;,C,,;,,,,,,,,,<,6,+,,,,<@@@C,<,;,;,;,6<,;,
9,9,9,
[2]    76
8A@C@FGGGFCGGGGDFDGGFGF7D,BC<CE,CFE9;;CE,,;,<,,,,,,,;,,<,,++++8+8+++++
++9,99
[3]    76
<6BCCGGGGCFGFGGGGGGGF9<,;CC,,,,,,<,CCEFC,,,,<;C,,,,,;<6,;C,6;CC,<,,;6,
666@68
```

Hay varias formas de manipular dichos objetos. Mediante la función alphabetByCycle", se obtiene un sumario de los nucleótidos usados en cada ciclo es una instancia de igual tamaño que mediante ShortReadQ o DNAStringSet. En nuestro caso, se muestran las 4 bases sobre los primeros 8 ciclos.

```{r}
alpbycy <- alphabetByCycle(sread(fq))
alpbycy[1:4, 1:8]
matplot(t(alpbycy[c("A","G","T","C"),]), type="l")
```

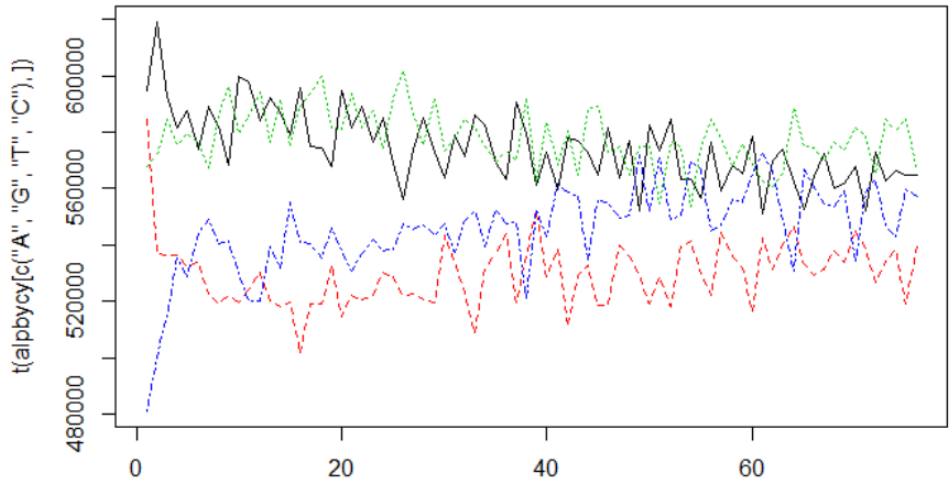


Figura 30. Distribución por base de ciclo

Y mediante otras funciones como alphabetFrequency, se obtiene el sumario de nucle?tidos por archivo.En nuestro caso hemos seleccionado los 10 primeros:

```
```{r}
alpFreq <- alphabetFrequency(sread(fq))
alpFreq[1:10,]
par(mfrow=c(1,2))
hist(alpFreq[,c("G", "C")],
     main = "Histogram of gc Content",
     xlab="individual reads" )
hist(alpFreq[,c("A", "T")],
     main = "Histogram of gc Content",
     xlab="individual reads" )

```
```

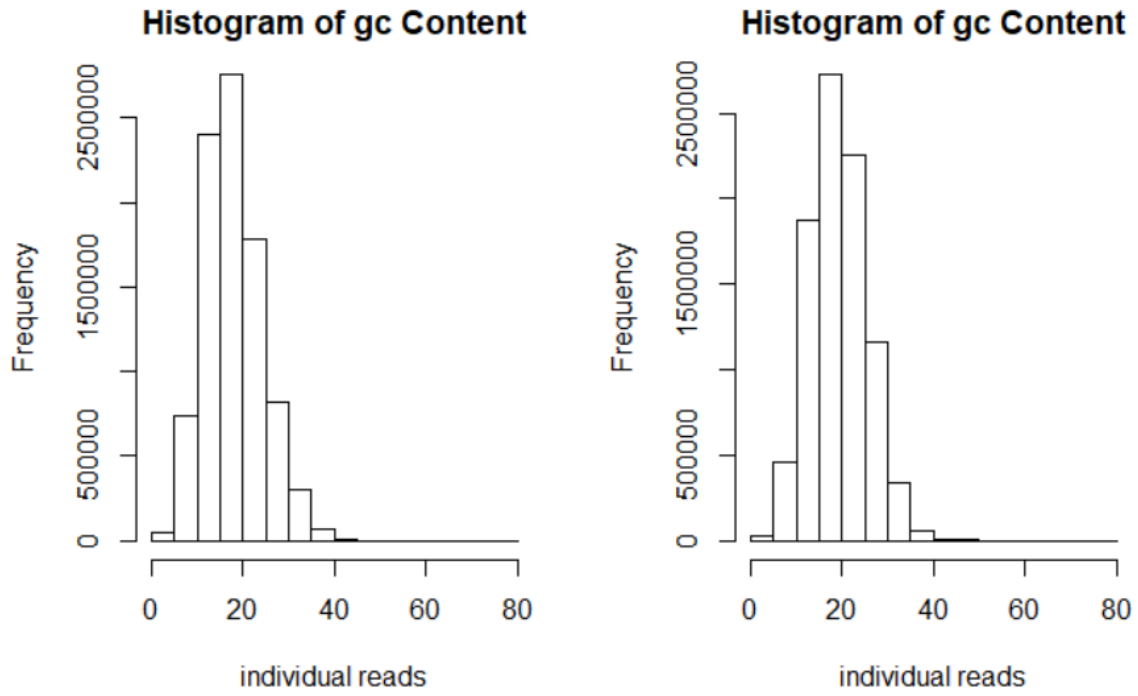


Figura 31. Distribución G-C (Izq). Distribución A-T (Drcha)

Asi como obtener la suma -log10 de los pvalues mediante la función alphabetScore para todas las lecturas. En nuestro caso de la 1 a la 10.

```
[1] 1731 1825 1898 1963 1781 1848 1928 1934 1862 1789
```

En algunos casos, las secuencias de los archivos fastq files son extremadamente largas, como podríaa ser el caso de un fastq de exoma. En nuestro particular, como los datos provienen de un panel de genes, son secuencias cortas y no es necesario sacar una muestra para el análisis, se analizan directamente completas. En un ejemplo cualquiera, la nomenclatura a seguir sería la siguiente, para una extracción de muestra de 1M de lecturas:

```
sampler <- FastqSampler(fastqFiles[1], 1000000)
yield(sampler) # sample of 1000000 reads
```

Del mismo modo, FastqStreamer() trabaja de un modo parecido, ejecutando de manera aleatoria una serie de muestras del archivo.
Por otro lado, se puede usar la función "tabla", para identificar el número de veces que una secuencia aparece en la lectura de nuestro archivo fastQ.

```
```{r}
readOccurence <- table(sread(fq))
sort(readOccurence,decreasing = TRUE)[1:25]
```
```

```
GTCCCCGGCCTGGCAGGGCGCCCTGGAGTGGGAGGAAGAGGTAACCACAGGGGGGCTGGAGCTGGCCTCG
GACTTG

1703
GCCCCTCCCGGGAAGGTGCCGTCTCCTCCGGCCCCTCGGGTCCCTGCTCTGTCACTGACTGCTGTGACCC
ACTCTG

1675
AGCACATCTGCCACCACTTGCACTGCGGTCTGGCTAACCACATGAGCATGGCCACTGATGAGGTGGATGG
AGGGTG

1674
```

Del mismo modo se pueden identificar lecturas duplicadas, potencialmente provenientes de PCR por amplificatión utilizando la función srduplicated y nuestro ShortReadQ object.

La función nos retorna un vector con aquellas secuencias duplicadas.A tener en cuenta que la primera vez qeu aparece la secuencia, es dicha secuencia en sí, no su duplicado. A partir de la primera, el resto son dichos duplicados. A continuación se muestra un resumen:

```
```{r}
duplicates <- srduplicated(fq)
duplicates[1:10]
table(duplicates)
```
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
Duplicates

  FALSE    TRUE
 350358 4106272
```

El siguiente escenario es el pre procesado de lecturas, por ejemplo para el recorte de las colas de baja calidad, los adaptadores o los añadidos para la preparación de la muestra.

En el marco de nuestro proyecto no es necesario dicha filtración por los motivos mencionados anteriormente, pero dado el caso en el que se observen baja calidad al final de las lecturas, puede ser posible la eliminación de las colas de baja calidad para su posterior alineamiento con el genoma. La función trimTails() , recorta las lectyuras desde 3', eliminando aquellas bases que están por debajo de un umbral deseado. Esta función acepta argumentos que especifican sobre el objeto ShortReadQ, el número mínimo de bases sucesivas requeridas para estar por debajo del límite de calidad para el recorte y la puntuación de corte real.

```
TrimmedFastq <- trimTails(fq,20,"5")
TrimmedFastq
```

Adicionalmente, una vez recortadas, podemos exportar nuestras lecturas FastQ para un análisis adicional utilizando la función writeFastq ().

```
writeFastq(TrimmedFastq,"myTrimmed_Fastq.fastq")
```

Una vez aplicados los filtros, se pueden volver a procesar los pasos indicados sobre la calidad de las muestras, asi como obtener la distribución de calidad especifico en cada ciclo mediante qa.

Antes de adentrarse en el siguiente paso, Galaxy ofrece la posibilidad de ejecutar un control de calidad sobre las secuencias, por lo que previo a su transformación a archivo .BAM. Sin demasiada extensión, se ve que el resultado de calidad a lo largo de todas las bases y se corrobora que son datos de alta calidad.

Measure	Value
Filename	15-0991_S1_L001_R1_001_fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	2228315
Sequences flagged as poor quality	0
Sequence length	76
%GC	48

Measure	Value
Filename	15-0991_S1_L001_R2_001_fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	2228315
Sequences flagged as poor quality	0
Sequence length	76
%GC	48

Figura 32. Galaxy Quality Check. Estadísticas.

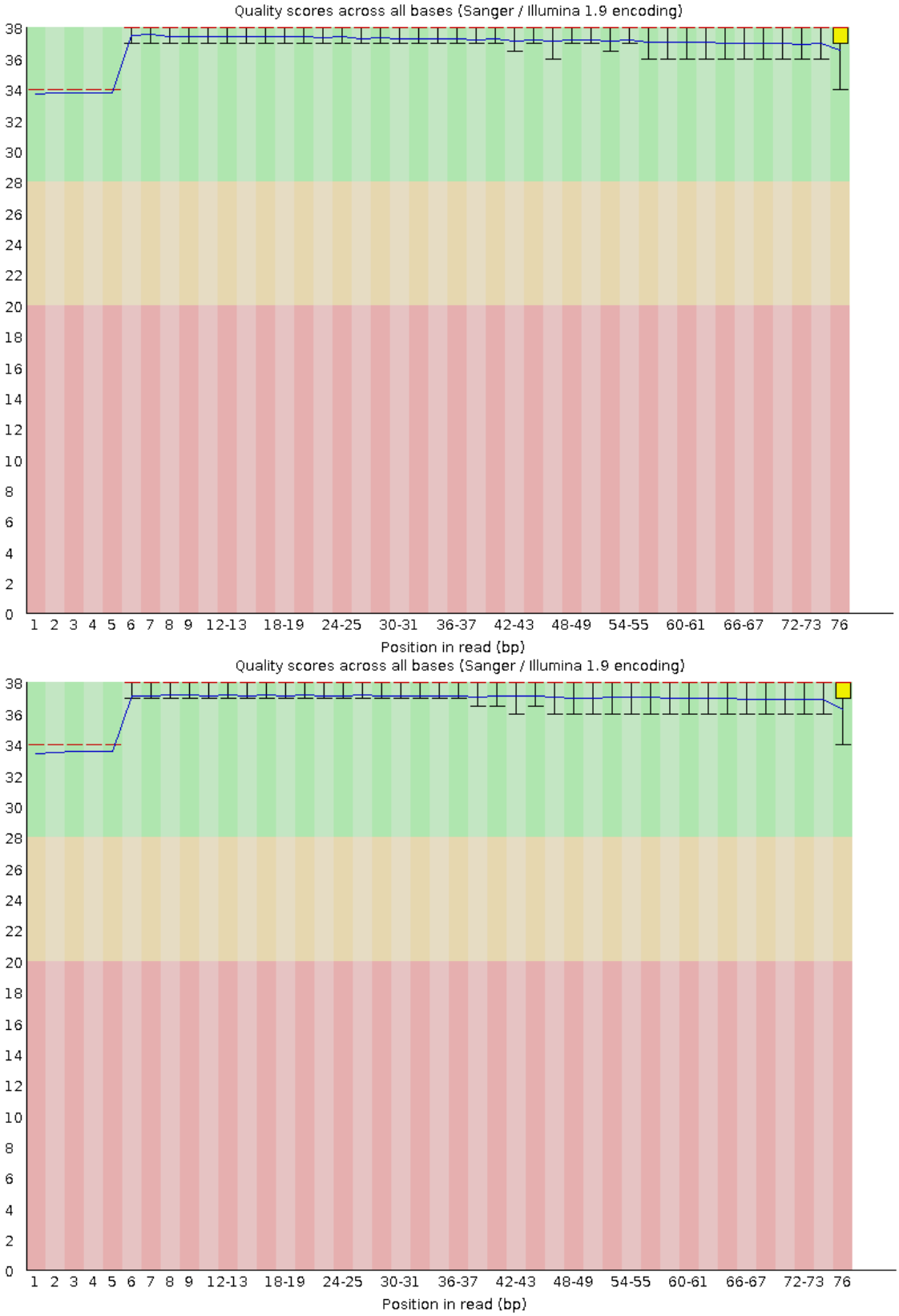


Figura 33. Galaxy Quality Check. Calidad a través de todas las bases

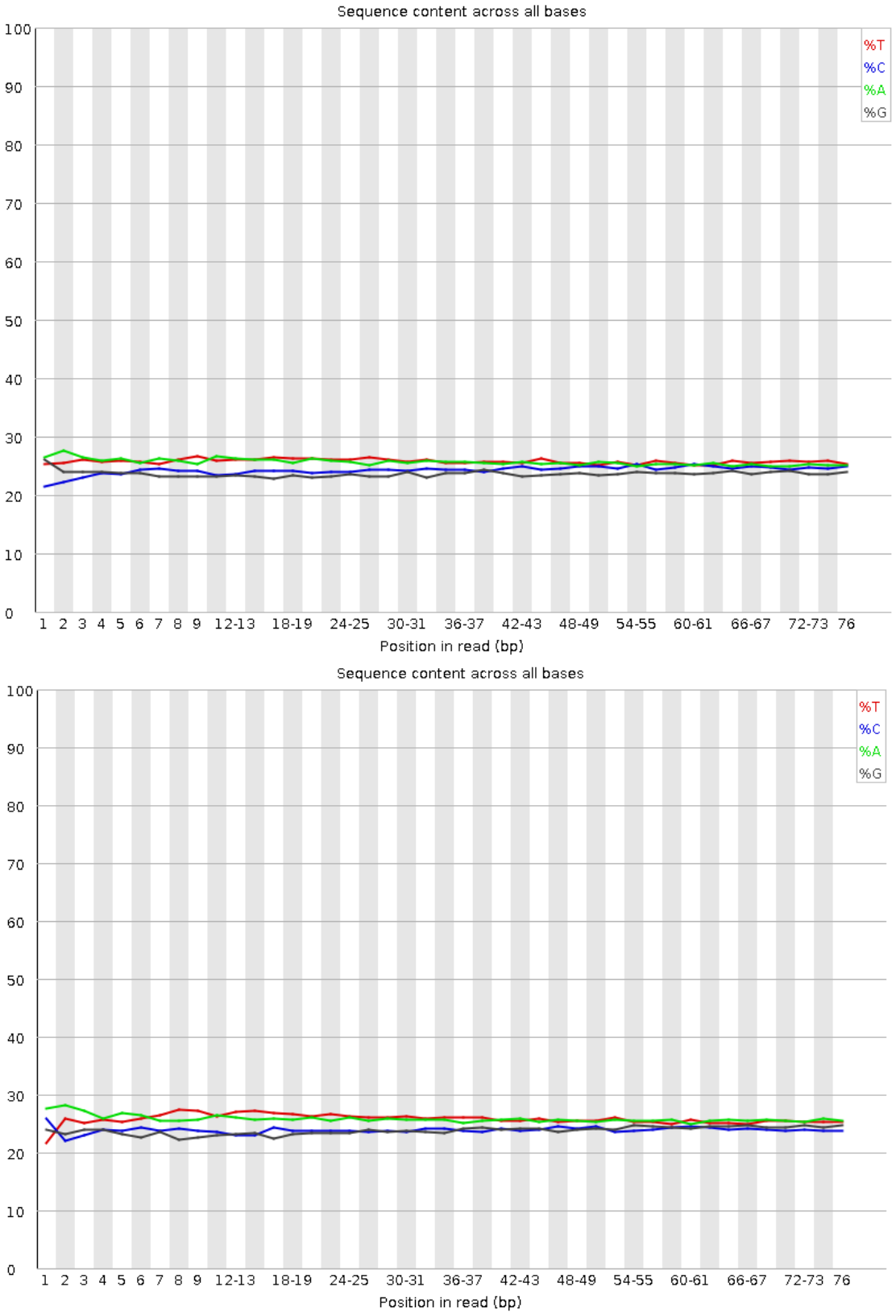


Figura 34. Contenido de bases a lo largo de la secuencia

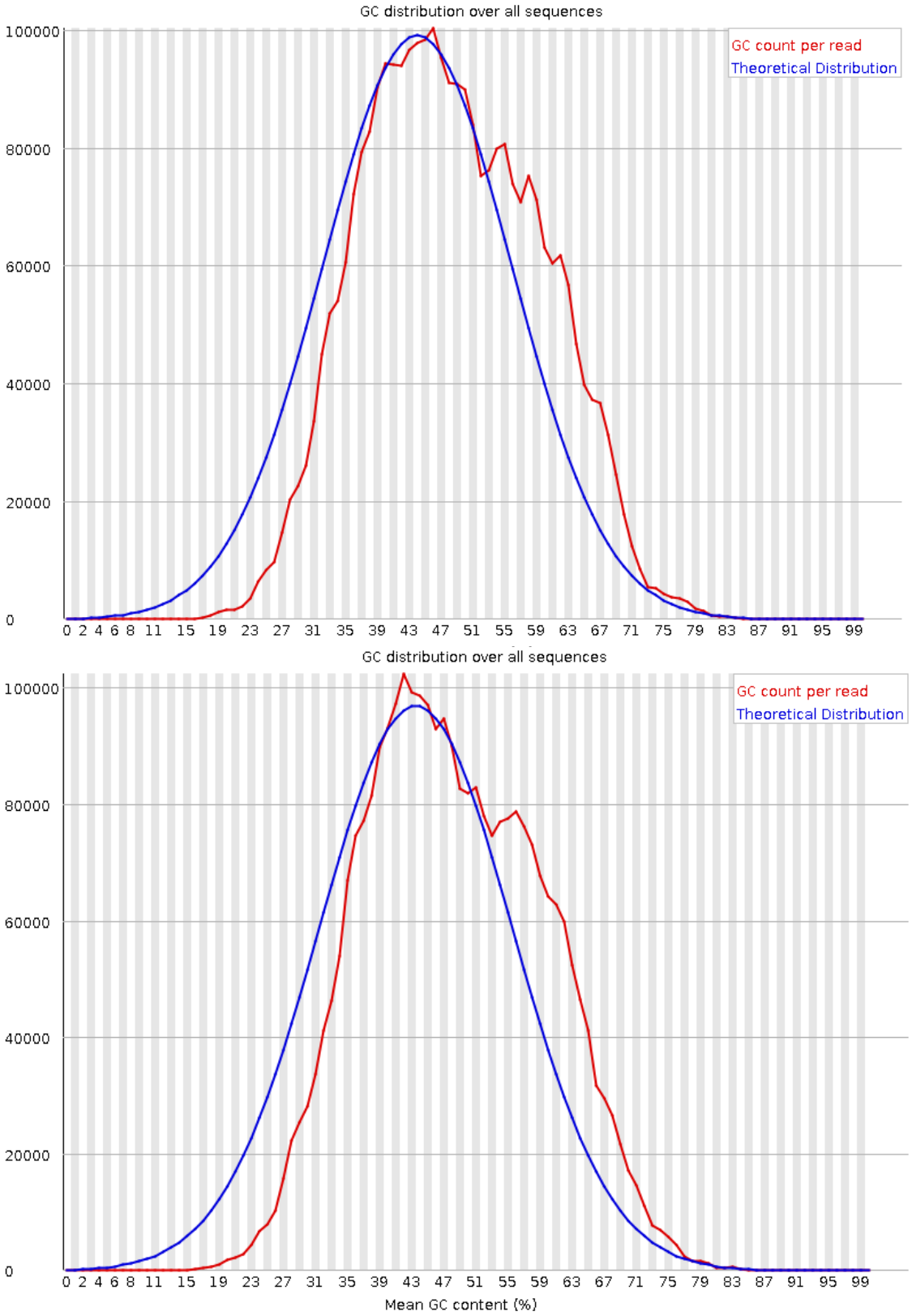


Figura 35. Duplicidad a lo largo de las secuencias. Teórico vs Medido

- La línea azul muestra el total de las secuencias
- La línea roja muestra las secuencias duplicadas

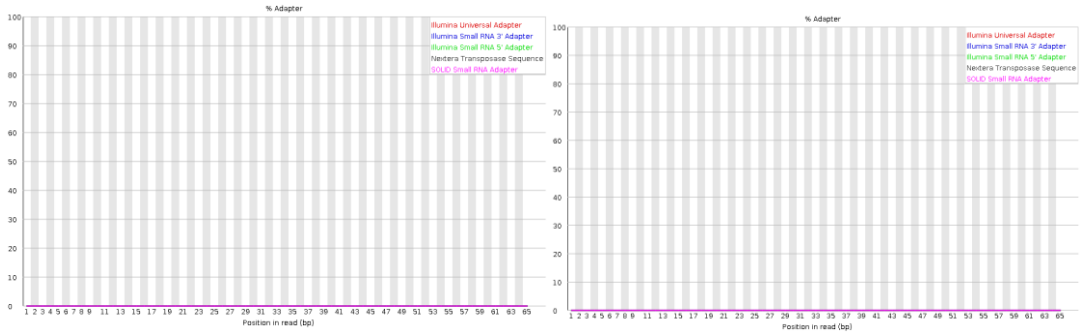


Figura 36. Verificación de no adaptadores en secuencias

7.2 Desarrollo Fase II

La secuenciación produce una colección de secuencias sin contexto genómico. No sabemos a qué parte del genoma corresponden las secuencias. El mapeo de las lecturas de un experimento a un genoma de referencia es un paso clave en el análisis de datos genómicos modernos. Con el mapeo, las lecturas se asignan a una ubicación específica en el genoma y se pueden obtener conocimientos como el nivel de expresión de los genes.

Las lecturas cortas no vienen con información de posición, por lo que no se sabe de qué parte del genoma provienen. Se necesitaría usar la secuencia de la lectura en sí para encontrar la región correspondiente en la secuencia de referencia. Pero la secuencia de referencia puede ser bastante larga (~ 3 mil millones de bases para humanos), lo que hace que sea una tarea desalentadora encontrar una región que coincida. Ya que en este caso, las lecturas son cortas, puede haber varios lugares, igualmente probables en la secuencia de referencia, desde los cuales se podrían haber leído. Esto es especialmente cierto para las regiones repetitivas.

En principio, se podría hacer un análisis de BLAST para averiguar dónde encajan mejor las piezas secuenciadas en el genoma conocido. Se tendría que hacer para cada uno de los millones de lecturas en nuestros datos de secuenciación. Sin embargo, alinear millones de secuencias cortas de esta manera puede llevar un par de semanas. En nuestro caso específico, se es conocedor que están mapeadas frente al cromosoma 10 (chr10), por lo que tanto a nivel de visualización como de anotaciones, ésta va a ser nuestra referencia.

Una vez revisados los archivos FASTQ y eliminados todas las secuencias de adaptadores que pudieran estar presentes, esta listo para mapearlos a un genoma de referencia. Si bien las herramientas como BLAST y BLAT son métodos poderosos, no están especializadas para la gran cantidad de datos generados por los secuenciadores de la próxima generación. Se recomienda encarecidamente que utilice un programa de alineación de lectura específico de la próxima generación. Es por eso que se va a emplear el paquete Bowtie2.

El workflow a seguir, partiendo de la base de los archivos fastq, es el siguiente:

1. Preparar los datos
2. Mapa leído en un genoma de referencia
3. Inspección de un archivo BAM
4. Visualización utilizando un navegador genoma (IGV) o mediante un navegador genoma (IGV)

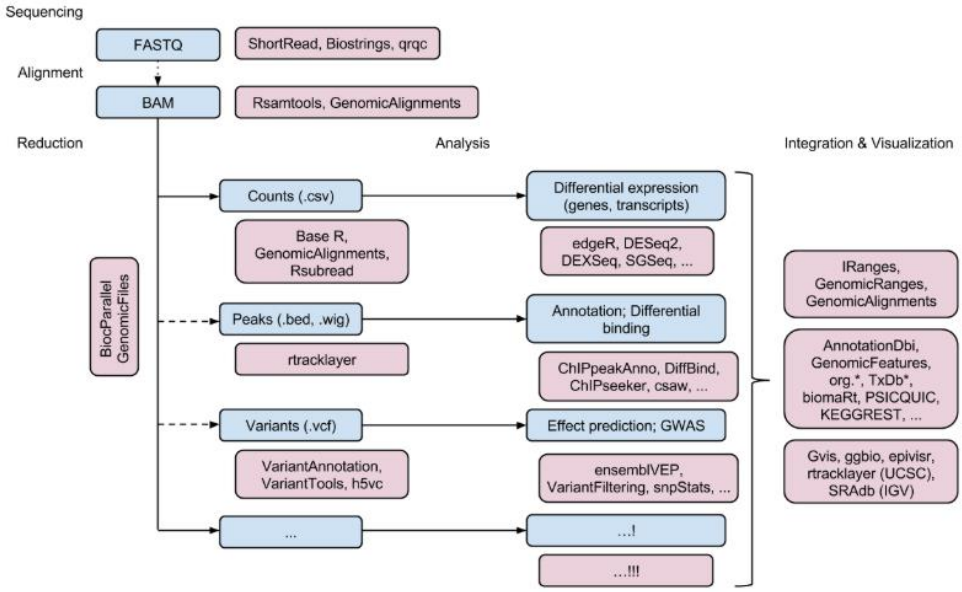


Figura 37. Workflow

Preparación de datos

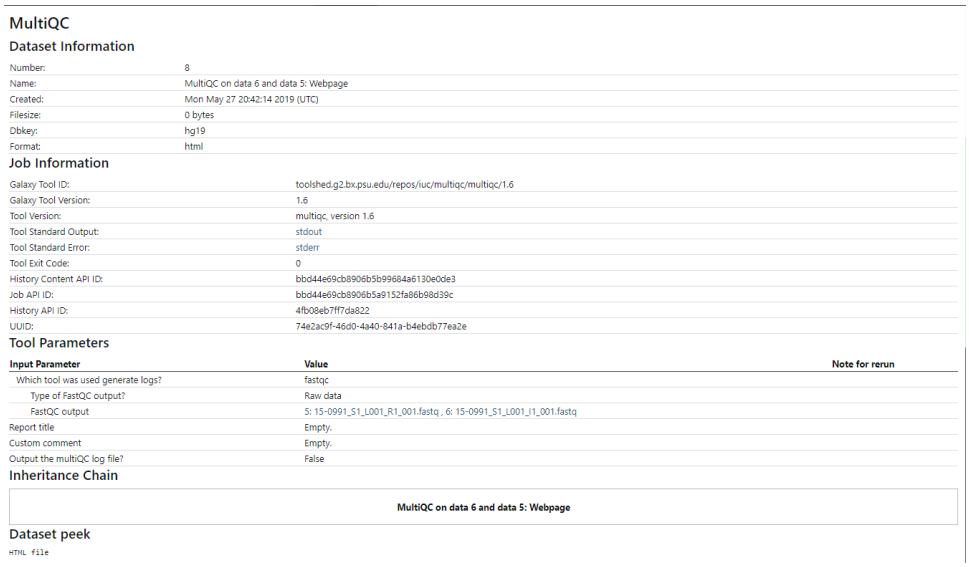
La salida de la mayoría de los algoritmos de alineación es un formato de archivo denominado SAM (mapa de alineación de secuencia) que contiene información de lectura y orientación en relación con la secuencia de referencia y un valor de confianza para la alineación.

A tamaño reducido, la versión binaria de SAM es el formato BAM.
Los archivos BAM se suelen comprimir y permiten mejorar la eficiencia de procesamiento. Al utilizar criterios de alineación iniciales, que suelen ser más permisivos que los algoritmos secundarios, la salida es un conjunto de datos que se sabe que contiene imprecisiones.

En este caso, la herramienta Galaxy, permite de una forma senzilla, transformar un archivo fastq en BAM, para su posterior análisis y alineación. Resaltar que en momento de la carga del archivo, definimos sobre el Genoma que está referenciado.

Figura 38. Carga de datos fastq y resumen de datos cargados (3 lecturas)

A continuación, ejecutamos la función MultiQC. MultiQC agrega los resultados de los análisis de bioinformática en muchas muestras en un solo informe. Toma los resultados de múltiples análisis y crea un informe. Es una herramienta de uso general, perfecta para resumir el resultado de numerosas herramientas bioinformáticas.



MultiQC

Dataset Information

Number:	8
Name:	MultiQC on data 6 and data 5: Webpage
Created:	Mon May 27 20:42:14 2019 (UTC)
Filesize:	0 bytes
Dbkey:	hg19
Format:	html

Job Information

Galaxy Tool ID:	toolshed.g2.bx.psu.edu/repos/iuc/multiqc/multiqc/1.6
Galaxy Tool Version:	1.6
Tool Version:	multiqc, version 1.6
Tool Standard Output:	stdout
Tool Standard Error:	stderr
Tool Exit Code:	0
History Content API ID:	bbd44e69cb8906b5b99684a6130e0de3
Job API ID:	bbd44e69cb8906b5a9152fa86b98d39c
History API ID:	4fb08eb7ff7da822
UUID:	74e2ac9f-46d0-4a40-841a-b4ebdb77ea2e

Tool Parameters

Input Parameter	Value	Note for rerun
Which tool was used generate logs?	fastqc	
Type of FastQC output?	Raw data	
FastQC output	5: 15-0991_S1_L001_R1_001.fastq , 6: 15-0991_S1_L001_I1_001.fastq	
Report title	Empty.	
Custom comment	Empty.	
Output the multiQC log file?	False	

Inheritance Chain

MultiQC on data 6 and data 5: Webpage

Dataset peek

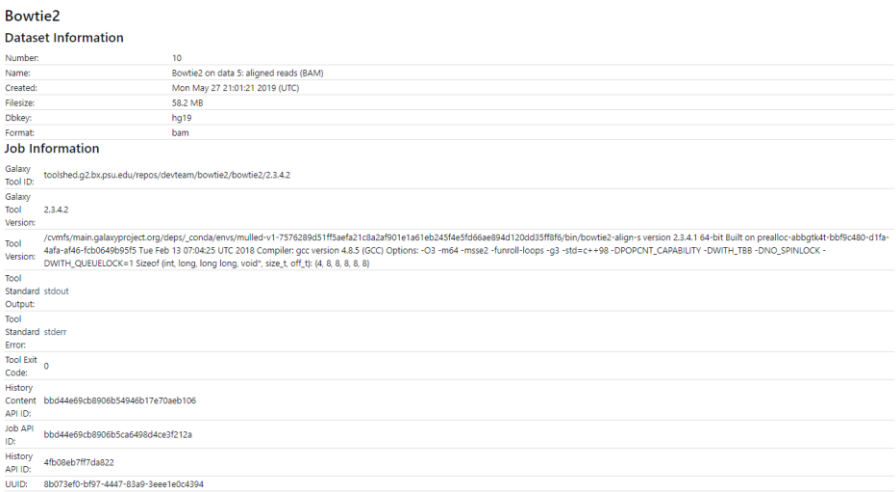
HTML file

Figura 39. MultiQC

Preparación del Genóma de referencia

Los genomas de referencia se pueden descargar desde los recursos del genoma UCSC, Ensembl o NCBI. En nuestro estudio utilizaremos la referencia humana GRCh37.hg19 pues los fastQ files están enfrentados contra ese Genoma.

Las alineaciones genómicas y las posteriores anotaciones pueden consumir mucho tiempo y no ser realistas en el corto tiempo que tenemos. Por lo tanto, una vez obtenido el archivo en formato BAM y preprocesamos un solo el cromosoma (chr10) del conjunto de datos anterior para ahorrar tiempo. El archivo BAM se puede obtener fácilmente sobre el programa Galaxy,a si como su posterior representación y anotaciones. Para el alineamiento, se utilizará el paquete Bowtie2.



Bowtie2

Dataset Information

Number:	10
Name:	Bowtie2 on data 5: aligned reads (BAM)
Created:	Mon May 27 21:01:21 2019 (UTC)
Filesize:	58.2 MB
Dbkey:	hg19
Format:	bam

Job Information

Galaxy Tool ID:	toolshed.g2.bx.psu.edu/repos/devteam/bowtie2/bowtie2/2.3.4.2
Galaxy Tool Version:	2.3.4.2
Tool Version:	/cvmfs/main.galaxyproject.org/deps/_conda/envs/mulled-v1-7576289d51ff5aefa21c8a2af901e1a61eb245f4e5fd66ae894d120dd35ff8f6/bin/bowtie2-align-s version 2.3.4.1 64-bit Built on prealloc-abbgtk4t-bbf9c480-d1fa-4afa-af46-fcb0649b95f5 Tue Feb 13 07:04:25 UTC 2018 Compiler: gcc version 4.8.5 (GCC) Options: -O3 -m64 -msse2 -funroll-loops -g3 -std=c++98 -DPOPCNT_CAPABILITY -DWITH_TBB -DNO_SPINLOCK -DWITH_QUEUELOCK=1 Sizeof {int, long, long long, void*, size_t, off_t}: {4, 8, 8, 8, 8, 8}
Tool Standard Output:	stdout
Tool Standard Error:	stderr
Tool Exit Code:	0
History Content API ID:	bbd44e69cb8906b54946b17e70aeb106
Job API ID:	bbd44e69cb8906b5ca6498d4ce3f212a
History API ID:	4fb08eb7ff7da822
UUID:	8b073ef0-bf97-4447-83a9-3eee1e0c4394

Figura 40. Encabezado BAM file

Para la definición de los campos de un BAM file, se puede tomar como referencia el siguiente ejemplo. Los archivos BAM constan de 2 partes principalmente:

- Una sección de encabezado (las líneas que comienzan con @) que contiene metadatos, en particular los nombres y longitudes del cromosoma (líneas que comienzan con el símbolo @SQ).

- Una sección de alineación que consta de una tabla con 11 campos obligatorios, así como un número variable de campos opcional

Col	Field	Type	Brief Description
1	QNAME	String	Query template NAME
2	FLAG	Integer	bitwise FLAG
3	RNAME	String	References sequence NAME
4	POS	Integer	1- based leftmost mapping POSition
5	MAPQ	Integer	MAPping Quality
6	CIGAR	String	CIGAR String
7	RNEXT	String	Ref. name of the mate/next read
8	PNEXT	Integer	Position of the mate/next read
9	TLEN	Integer	observed Template LENgth
10	SEQ	String	segment SEQuence
11	QUAL	String	ASCII of Phred-scaled base QUALity+33

Figura 41. Explicación campos BAM files

Ejemplificando:

@SQ SN:chr10 LN:135534747

QNAME: M01621:321:000000000-BV2JC:1:2108:23966:23449
FLAG: 16
RNAME: chr10
POS: 1739232
MAPQ: 42
CIGAR: 76M
MRNM: *
MPOS: 0
ISIZE: 0
SEQ:
CTTCTTCTGATGTGGCCCAGGGAAGCCAAAAGATTGGACACCCCTGATCTAAAGGTTTTCATTTCTGTTTCTT
CTC
QUAL:
FGFGGGGGGGGGGGGGGGGGGGFGGGGGFGGGGGGGGGGGGGGGFGGGGGGGGGGGGGGGGGGG
GGGGGCCCCC

Mapa leído en un genoma de referencia

El mapeo de lectura es el proceso para alinear las lecturas en un genoma de referencia. Un mapeador toma como entrada un genoma de referencia y un conjunto de lecturas. Su objetivo es alinear cada lectura en el conjunto de lecturas en el genoma de referencia, permitiendo desajustes, indeles y recortes de algunos fragmentos cortos en los dos extremos de las lecturas:

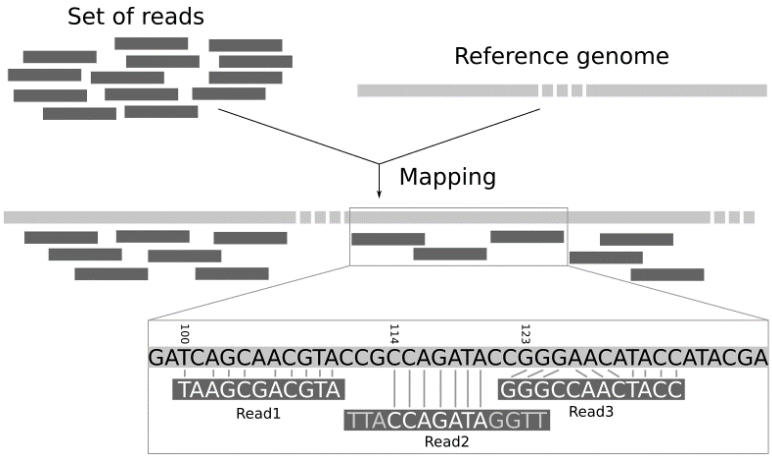


Figura 42. Ejemplo de proceso de mapeo. La entrada consiste en un conjunto de lecturas y un genoma de referencia. En el medio, proporciona los resultados del mapeo: las ubicaciones de las lecturas en el genoma de referencia. La primera lectura se alinea en la posición 100 y la alineación tiene dos desajustes. La segunda lectura se alinea en la posición 114. Es una alineación local con recortes a la izquierda y a la derecha. La tercera lectura se alinea en la posición 123. Consiste en una inserción de 2 bases y una eliminación de 1 base.

La verificación de las estadísticas de mapeo es un paso importante que se debe hacer antes de continuar con cualquier análisis. Existen varias fuentes potenciales de errores en el mapeo, que incluyen (pero no se limitan a):

- Reacción en cadena de la polimerasa (PCR): muchos métodos HTS involucran uno o varios pasos de PCR. Los errores de PCR se mostrarán como desajustes en la alineación, y especialmente los errores en las primeras rondas de PCR se mostrarán en múltiples lecturas, lo que sugiere falsamente una variación genética en la muestra. Un error relacionado serían los duplicados de PCR, donde el mismo par de lectura se produce varias veces, sesgando los cálculos de cobertura en la alineación.

- Errores de secuenciación: la máquina de secuenciación puede realizar una llamada errónea, ya sea por razones físicas (por ejemplo, aceite en un portaobjetos de Illumina), o por las propiedades del ADN secuenciado (por ejemplo, homopolímeros). Como los errores de secuencia son a menudo aleatorios, se pueden filtrar como lecturas singleton durante la variante de llamada.

- Errores de asignación: el algoritmo de asignación puede asignar una lectura a la ubicación incorrecta en la referencia. Esto sucede a menudo alrededor de repeticiones u otras regiones de baja complejidad.

Si alguna de los supuestos anteriores se identifican en los mapeos, se debe de investigar la causa de estos errores antes de continuar con sus análisis.

Anotaciones:

Para poder utilizar la información dentro de la secuencia de un genoma es fundamental anotarla con datos biológicos relevantes. Las anotaciones de un genoma ayudan a los científicos a entender qué significa su secuencia, cómo está estructurado y cómo funciona.

La parte fundamental de los métodos de anotación es la localización de genes en un genoma, así como la determinación de su estructura y de las proteínas que producen. Las herramientas de anotación intentan realizar esto a través de análisis computacionales y enfoques experimentales. A pesar de que existe una gran diversidad de procedimientos para anotar un genoma, todos comparten un conjunto de características esenciales.

Al final, el mejor enfoque depende del tiempo y recursos disponibles.

Entrada/Muestra - Ensamble de genoma:

Se necesita un ensamble de alta calidad, cuya secuencia esté determinada al menos en un 90% y que tenga la menor cantidad de huecos posible Bases de datos de genes y/o de genomas similares.

Datos de información genética:

Información biológica de expresión de genes proveniente de tecnologías como RNA-seq Computadora con alta capacidad de almacenamiento y procesamiento. Se requiere de equipo de cómputo con alta capacidad de memoria y procesamiento, ya que se maneja una gran cantidad de datos biológicos. Programas computacionales de anotación. El conjunto de programas que se utilizarán depende en gran medida en la naturaleza del genoma, en el tipo de datos que serán utilizados, el objetivo de la anotación y de los recursos disponibles.

Procedimiento - Preparación del genoma:

Antes de iniciar se necesita preparar el ensamble identificando y descartando las partes del genoma que no contienen genes. Estas regiones suelen complicar el proceso y pueden generar anotaciones erróneas

Fase computacional:

La primera fase consta de un proceso computacional donde se identifican los elementos del genoma con base en información de experimentos de expresión y en genes de otros genomas. Esta información es mapeada a la secuencia del genoma de referencia y utilizada de manera conjunta para predecir genes. Una alternativa complementaria es la predicción ab initio (mediante modelos matemáticos), sobre todo cuando no se cuenta con evidencia externa.

Fase de anotación:

En esta fase el objetivo es combinar los elementos mapeados, adjuntarles información biológica y finalmente definir un conjunto óptimo de anotaciones. Es una tarea difícil, por lo que se necesitan combinaciones de distintos procedimientos computacionales para abordarla con precisión; también se suele utilizar la curación manual Validación.

Las dos primeras fases dan buenos resultados, mas es necesario validar las anotaciones cualitativamente. Esto se logra mediante inspecciones manuales, comprobaciones experimentales y medidas de calidad Publicación del genoma. Para la publicación de la anotación en una base de datos es posible crear una propia, enviarla a una de las más importantes o a alguna temática. Hacer públicos los datos incentiva el mejoramiento del ensamble y la anotación.

Salida/Resultado:

Un conjunto diverso de datos biológicos localizados a lo largo de nuestro genoma de interés. Al principio se cuenta con una simple secuencia y al final podemos saber qué genes hay en ella, qué hacen y en dónde se encuentran.

Fuentes de error más frecuentes:

- No filtrar las regiones en el genoma que no contienen genes.
- Fallas al elegir los programas computacionales.
- Los datos de referencia contienen errores.
- Se utiliza un genoma de referencia con anotaciones erróneas.
- Aplicaciones Predicción de genes Predicción de funciones de genes.

La anotación de genomas tiene numerosas aplicaciones en la investigación biológica, ya sea en el desarrollo de hipótesis o en los análisis de genómica comparativa. Además, son un medio importante para la anotación de otros genomas.
Para la alineación, existen infinidades de SW, entre otros:

- Lookseq
- IGV
- JBrowse
- Genome Workbench
- ...

En el caso particular del estudio, una vez ejecutado la transformación a BAM file y la alineación con el Genoma de referencia, obtenemos su visualización a través del software IGV para sus anotaciones y posteriores variantes. Básicamente, se ha elegido este visualizador porque es una herramienta muy potente, con posibilidad de descarga en escritorio local y con una gran capacidad y rápidez de análisis.
En particular, este Interface es ofrecido directamente por Galaxy, una vez obtenidos los archivos BAM, por lo que eso, facilita en su mayoría su posterior visualización y análisis. Antes de visualizarla, abrimos el archivo obtenido de la alineación. Se observa que:

```
2228315 reads; of these:
  2228315 (100.00%) were unpaired; of these:
    4428 (0.20%) aligned 0 times
    2139611 (96.02%) aligned exactly 1 time
    84276 (3.78%) aligned >1 times
99.80% overall alignment rate
```

Con dichos porcentajes, se podría determinar, que la secuencia está lista para ser visualizada y posteriormente comprobar si existen anotaciones.

Si visualizamos las imágenes, observamos lo siguiente:

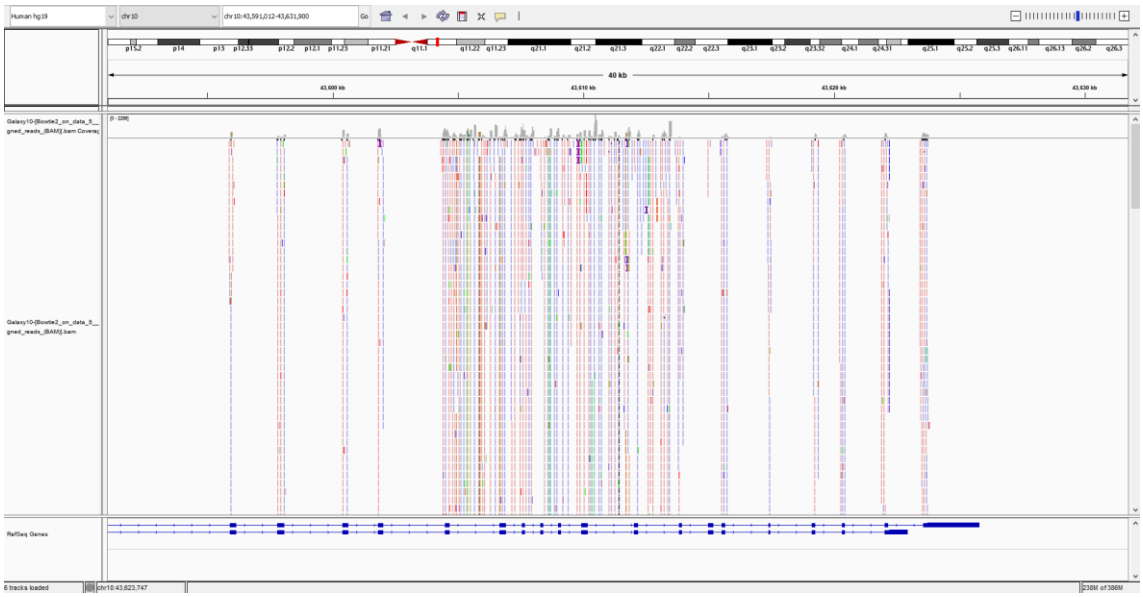


Figura 43. Captura principal IGV

- Para la carga el archivo BAM, se necesita el archivo.bai de igual modo, el cual representa el “índice” por tal de poder visualizarla.
- Un vez cargados, es necesario encontrar el foco de la secuencia dentro del cromosoma correspondiente y según la longitud de las lecturas. En el caso del estudio, estas lecturas corresponden al cromosoma 10 y la posición, viene determinado por la distancia dada en la figura 40 (POS).
- En rasgos generales, podemos observa que en la parte superior se encuentra la longitud total del cromosoma y en rojo, aquella parte del cromosoma a la cual se está alineando.
- En la parte inferior, se puede ver el Gen/Genes a los cuales está referenciada nuestra lectura (RefSeq Genes). En el ejemplo, se muestra como la secuencia está relacionado con la expresión del gen RET. En los anexos, se puede encontrar una breve descripción del Gen RET, aunque interactivamente seleccionando dicho nombre, este redirige la llamada a la página NCBI para su descripción y contexto genómico, asi como su expresión. (https://www.ncbi.nlm.nih.gov/gene?term=NM_020630)

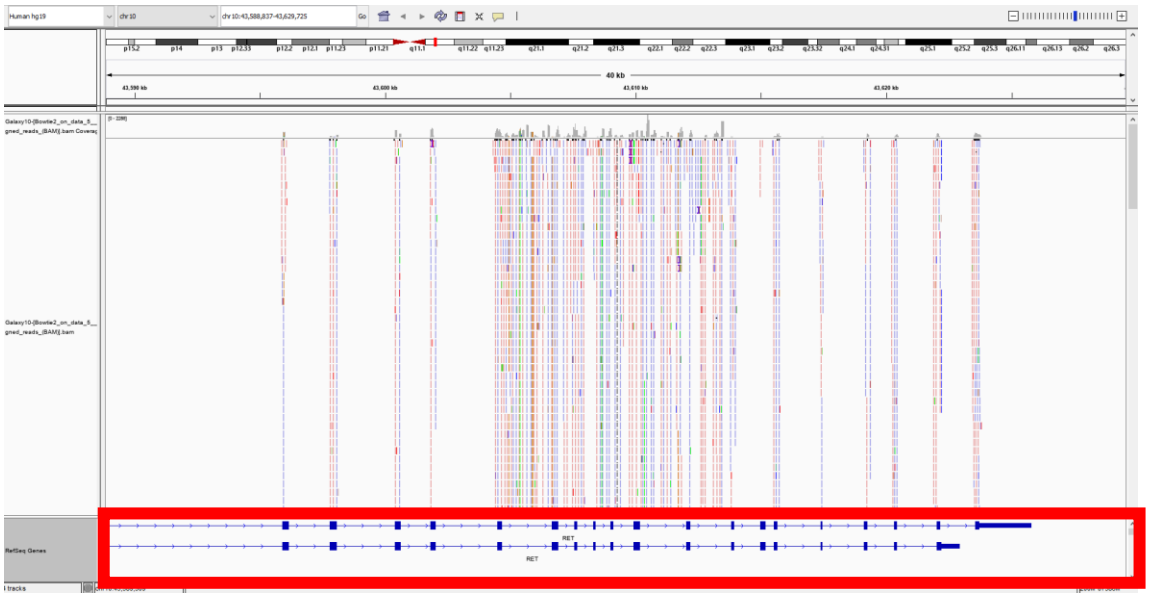


Figura 44. Representación de la región particual del gen.

Analizando el Gen en cuestión y ampliando la imágen, obtenemos las posibles variaciones, en este caso en particular, se observan 2 posibles variaciones. La representación de dichos genes sigue el siguiente formato:

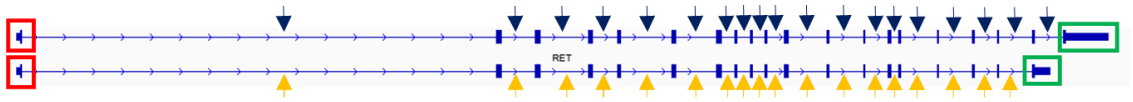


Figura 45. Variaciones gen RET

- 5’ UTR en la parte izquierda (roja)
- 3’ UTR en la parte derecha (verde)
- 19 Intrones en la variación 1(Negro) y 18 en la variación 2 (Amarillo)
- 18 exones en la variación 1 y 17 en la variación 2

Por defecto, el string está representado de 5’ a 3’, pero si se selecciona la señal, se gira hacia el reverso.

Para la visualización de cada una de las secuencias, una vez hecho “zoom in” en la zona de la secuencia seleccionada, navegando a través, se puede ver qué lecturas han sido alineadas respecto al genoma de referencia.

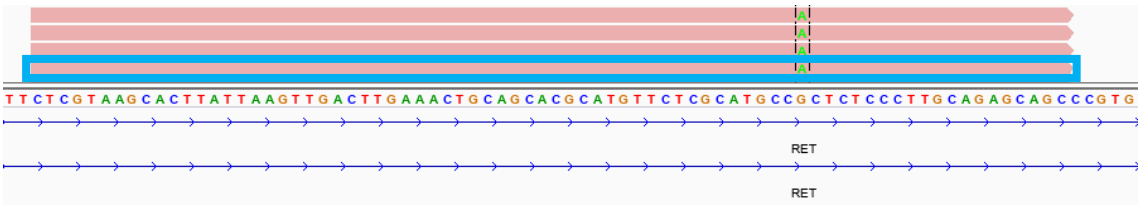


Figura 46. Lecturas frente a genoma de referencia (azul)

En la cabecera, se puede observar que el número de alineaciones referenciadas en una localización particular, viene determinada por la amplitud de la barra superior (azul vs rojo). Para el resto de lecturas, se pueden visualizar simplemente bajando el cursor del ratón.

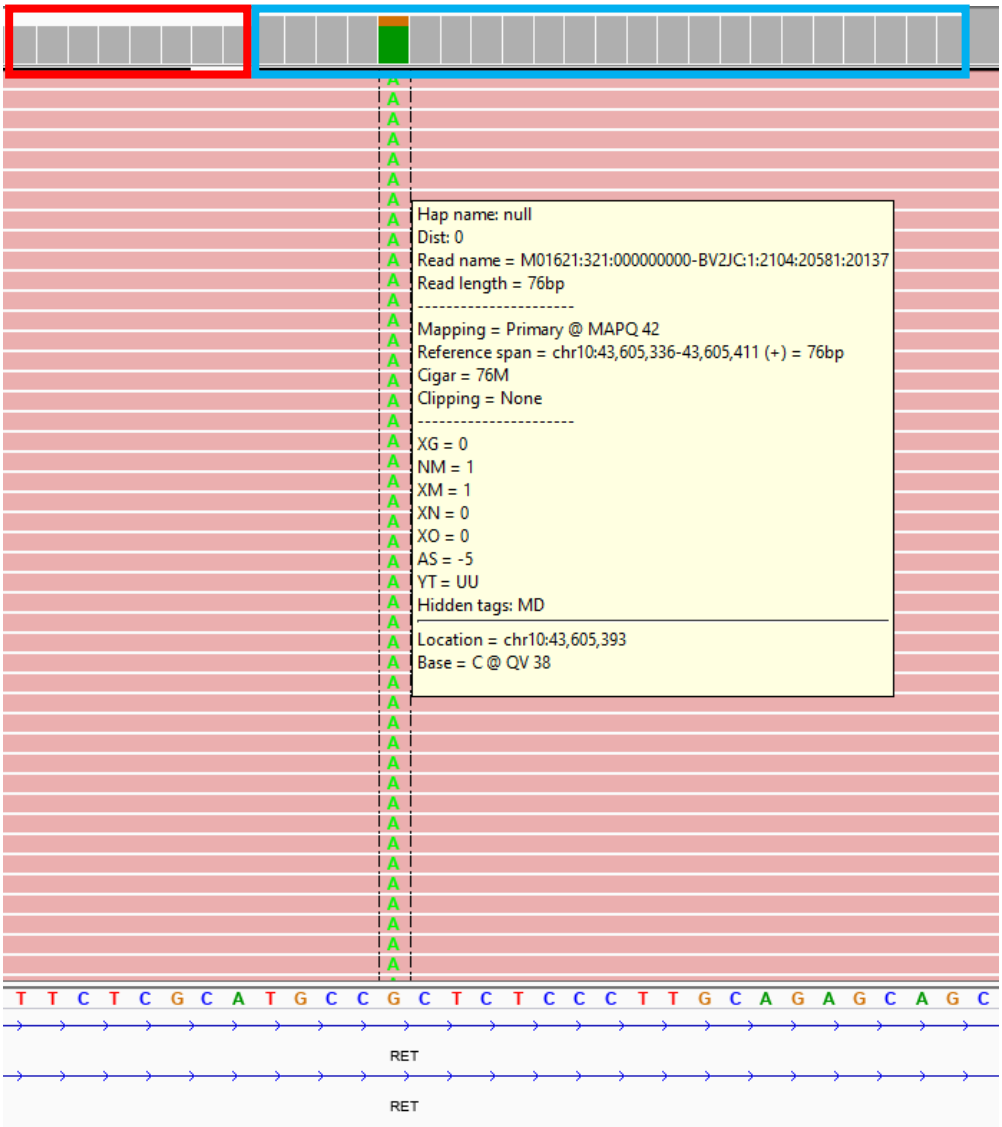


Figura 47. Diferencia de lecturas alineadas

Los eventos interesantes, estás resaltados en color. En el ejemplo anterior, la base de referencia es Guanina (G), pero en la mayoría de las lecturas está representado como Adenina (A). Si el ratón se posa en la posición, representa el número de bases de la lectura seleccionada. Además, si existe otro color, como este caso es el de la Guanina (21%), éste se muestra cuando un nucleótido difiere de su referencia en más de un 20%. Este porcentaje de visualización es modificable en las opciones, mediante el ajuste del límite de cobertura del alelo. En este caso en particular, cuando además las calidades están expresada de mayor intensidad a menor, se puede concluir que es un heterocigoto SNP (Hetereazygous SNP), es decir, una secuencia que afecta a una sola base o Polimorfismo de nucleótido simple.

Un polimorfismo de un solo nucleótido o SNP (Single Nucleotide Polymorphism, pronunciado snip) es una variación en la secuencia de ADN que afecta a una sola base (adenina (A), timina (T), citosina (C) o guanina (G)) de una secuencia del genoma. Destacara que si no se llega al 1% no se considera SNP y sí una mutación puntual.

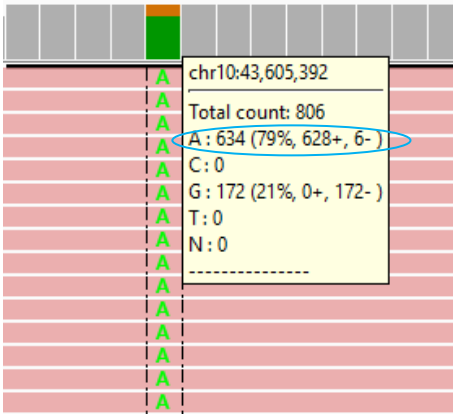


Figura 48. Recuento lectura vs base referencia

Si nos desplazamos a través de las lecturas, nos podemos encontrar en que haya posiciones dónde no existen eventos diferenciados de la referencia, lo cual indica que las lectura y la referencia, coinciden por completo.

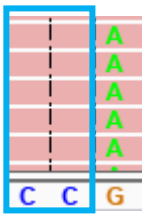


Figura 49. Lectura vs referencia, sin diferencia

Las bases representadas en un color más intenso representarán una mayor intensidad y por consiguiente, de mayor calidad, mientras que las más débiles, representarán menor calidad.

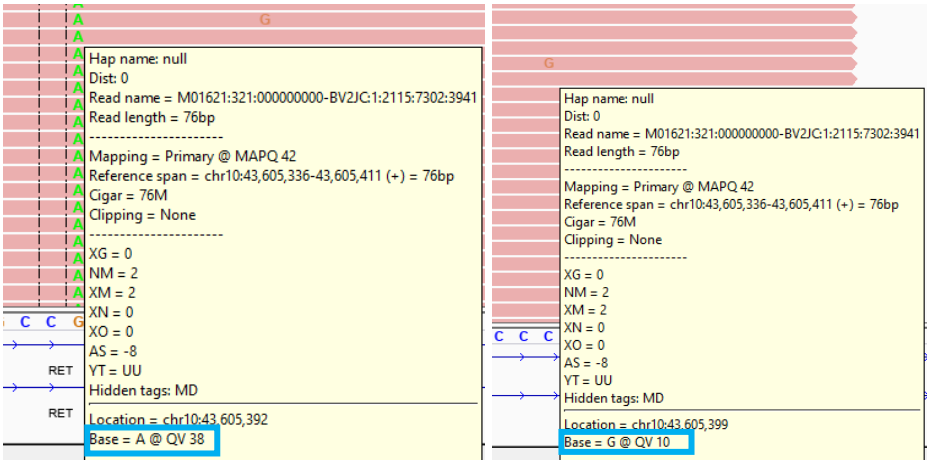


Figura 50. Diferencia calidad lectura (A@ QV38 vs G@ QV10)

Otro tipo de anotaciones que se pueden observar es que dentro de una misma lectura, existen mapeads con calidad superior a otras. En el ejemplo, se muestra una lectura con calidad 42 vs 0.

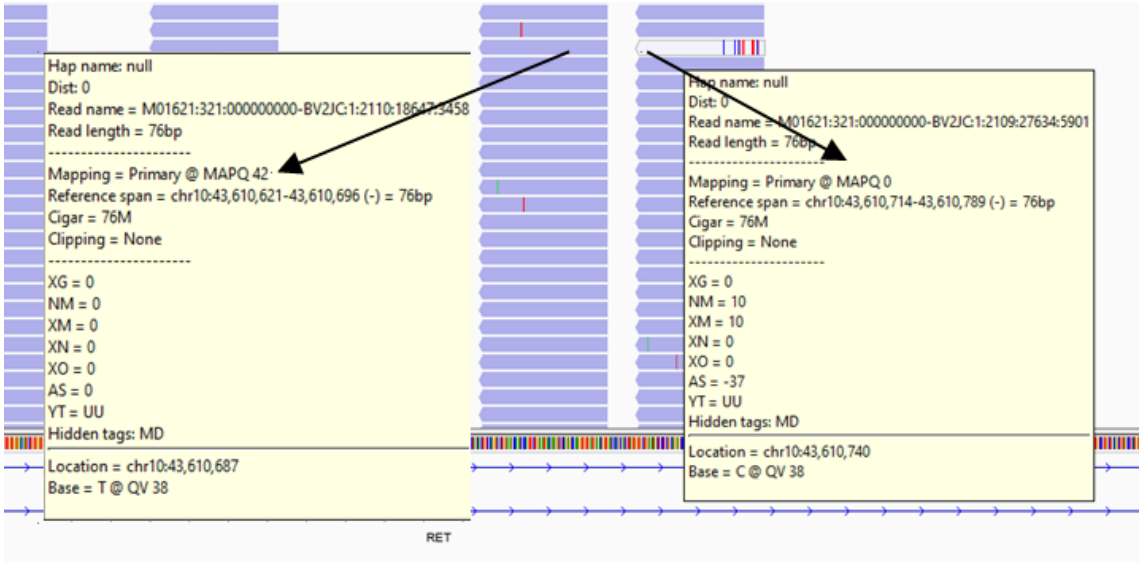


Figura 51. Diferencia mapping

Existe la posibilidad de encontrar inserciones y delecciones a lo largo de las lecturas. En el siguiente ejemplo se muestran 5 inserciones de bases entre

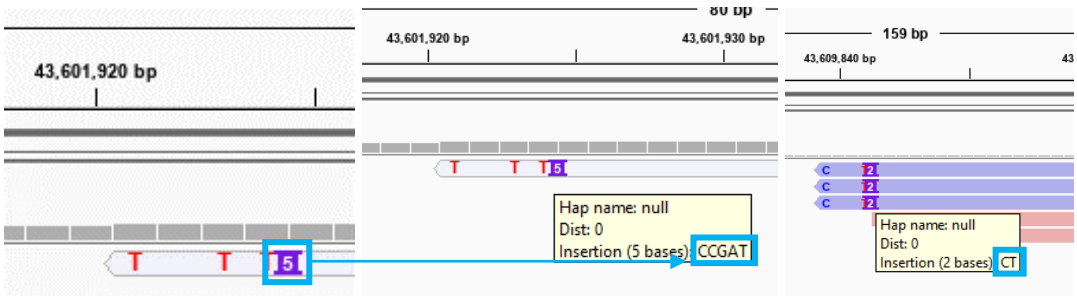


Figura 52. Inserciones

En el caso de las delecciones, se representan mediante una línea entre lecturas. En nuestro caso en particular, en la zona del chr10 no se han encontrado.

Figura 53. Delecciones de bases

Otro potencial SNP, puede ser debido a que no hay llamadas respecto al umbral de representación y del genoma de referencia. Para entender mejor este apartado, causante muchas veces de falsos positivos en los datos de secuenciación, se colorean las lecturas de forma que se diferencian las anversas (rojas) y las reversas (azules). En varias zonas de los datos cargados se puede observar que mientras puede haber un posible SNP en la cadena anversa, esta no existe en la reversa. Strand Bias, succede cuando un genotipo infiere cuando una lectura anversa y reversa son distintas. En este ejemplo se podría concluir que no hay SNP en el reverso, pero si un posible Heterocito SNP. Si esto fuera verdad, se esperaría también una Timina en el reverso, pero al existir sólo en el anverso, se puede decir con certeza que es un caso de falso positivo.

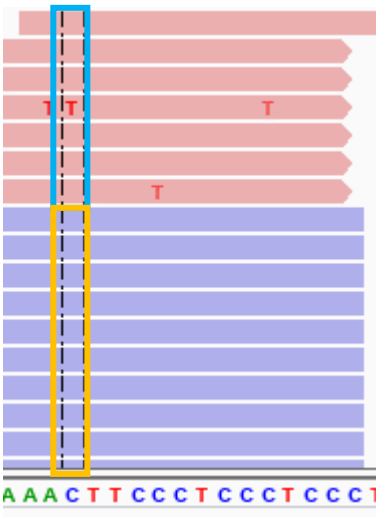


Figura 54. SNP Anverso vs reverso

En resumen, para la detección de los SNP más destacados en nuestra secuencia, se puede utilizar la función pileup dentro de Galaxy. Esta función facilita la selección e identificación de los SNPs más relevantes.

La descripción de los pileups viene determinada por 6 columnas:

```
   1      2         3     4      5     6
----------------------------------------
chr10  1739232      c     1      ^K,   E
chr10  1739233      t     1      ,     G
chr10  1739234      t     1      ,     F
```

dónde:

```
Column  Definition
------- ------------------------------
   1    Chromosome
   2    Position (1-based)
   3    Reference base at that position
   4    Coverage (# reads aligning over that position)
   5    Bases within reads where (see Galaxy wiki for more info)
   6    Quality values (phred33 scale, see Galaxy wiki for more)
```

Para no obtener una gran cantidad de datos y que no se pueden representar en pantalla ni tampoco analizar, es necesario un filtrado.

Pre Filtrado y filtrado de variantes

En este proceso se eliminarán aquellas variantes fuera de las regiones codificantes, así como los errores de secuenciación. Esta sección se puede dividir en dos pasos: el primero es prácticamente automático y consiste en eliminar aquellas variantes fuera de las regiones codificantes; el segundo es más manual, y consiste en visualizar las lecturas para eliminar posibles errores de secuenciación que no se han detectado durante el análisis previo.

Con esta aproximación los SNPs se detectan tras el mapeo de las reads frente a un genoma de referencia. Usando esta aproximación clásica, en los genomas bajo análisis los SNPs se detectan alineando las reads con la secuencia ensamblada del genoma de referencia.

La anotación usada para evaluar el impacto funcional de los SNPs es la anotación del genoma de referencia.

Por tanto, se obtiene un archivo VCF con las variantes y una archivo BAM con su correspondiente archivo BAI con las reads alineadas al genoma de referencia. Este enfoque permite saber el número de reads que respaldan cada SNP.

Hay que tener en cuenta que el cromosoma 10 tiene una longitud aproximada de 128mb, lo que significa que probablemente tendremos muchísimos SNPs, si intentamos ejecutarlo aparecen alrededor de 1,3 millones. A medida que se vaya filtrando, por ejemplo, c7>50 (la columna 7 contiene un score que combina varias medidas de calidad, como la cobertura o la calidad por posición), iremos reduciendo el número de SNP para poderlo mostrar con mejor detalle.

Como se ha mencionado anteriormente, la lista de variantes que proporciona el pipeline de análisis, comercial o propio, será bastante extensa debido a la detección de polimorfismos (Single Nulceotide Polymorphism, SNP), errores de secuenciación, variantes que no están situadas dentro de la zona codificante del gen, etc. Por ello, antes de comenzar a interpretar las variantes candidatas, se deben filtrar aquellas variantes que no se vayan a informar. Cabe destacar que los criterios de filtrado de variantes están en evaluación continua, ya que, dependiendo de los avances que se

realicen en el campo de la secuenciación masiva, puede ser conveniente modificar la forma de evaluar las variantes.

Se ejecuta un pileup para a partir del archivo BAM creado para la visualiación de las variantes y se filtra para obtener un número de filas no demasiado grande.

Un pileup es una representación en columnas de las lecturas alineadas, en el nivel base, a la referencia. El archivo de pila resume todos los datos de las lecturas en cada región genómica que está cubierta por al menos una lectura. Cada fila del archivo de pila proporciona información similar a una sola columna vertical de lecturas en la vista IGV.

Se filtra siguiendo el patrón:

- Filtrado de 6 columnas
- Sin reporte de posiciones con mayor cobertura de 30
- Reporte de sólo variantes. Esto hará efectivamente la llamada de la variante: solo le dará ubicaciones que tengan alguna evidencia de que podría haber una variante presente. Esta variante de llamada no es muy estricta, por lo que aún obtendrá muchas filas. Podríamos filtrar aún más, por ejemplo, las variantes con puntuaciones de alta calidad.

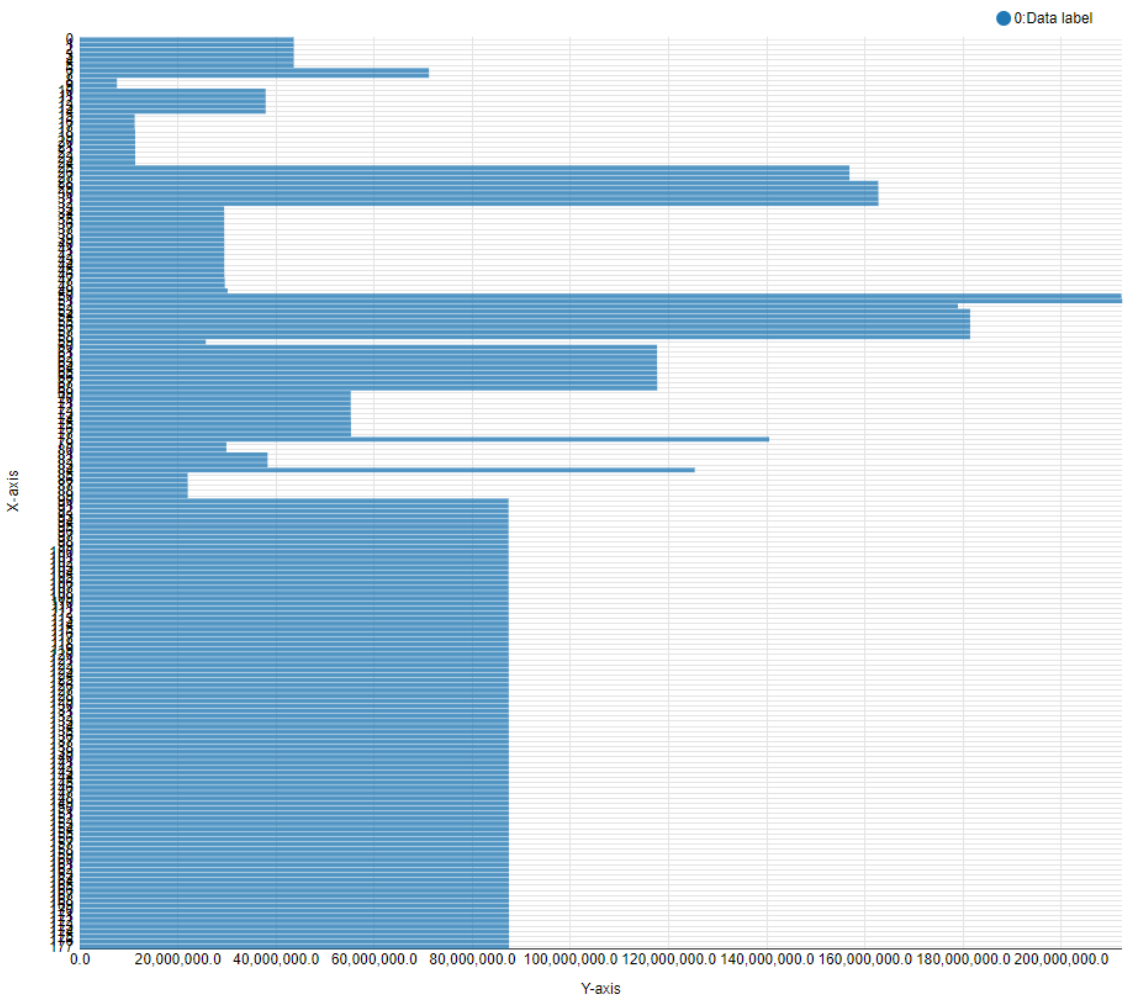


Figura 55. SNP Pileup filter

FreeBayes es un llamador variante bayesiano que evalúa la probabilidad de cada genotipo posible para cada posición en el genoma de referencia, dadas las lecturas observadas en esa posición, e informa la lista de variantes posibles.

La generación actual de herramientas de llamada de variantes no genera archivos de pila, y no necesita hacer esta sección para usar FreeBayes en la siguiente sección. Sin embargo, un archivo de pila es una buena ilustración de la evidencia de que la variante que llama es internamente, y produciremos uno para ver esta evidencia.

La información para el filtrado del cromosoma 10 utilizando esta técnica es la siguiente:

FreeBayes

Dataset Information

Number:	54
Name:	FreeBayes on data 10 (variants)
Created:	Tue Jun 4 07:32:37 2019 (UTC)
Filesize:	12.3 KB
Dbkey:	hg19
Format:	vcf

Job Information

Galaxy Tool ID:	toolshed.g2.bx.psu.edu/repos/devteam/freebayes/freebayes/1.1.0.46-0
Galaxy Tool Version:	1.1.0.46-0
Tool Version:	
Tool Standard Output:	stdout
Tool Standard Error:	stderr
Tool Exit Code:	0
History Content API ID:	bbd44e69cb8906b5a47882b4b27d97f1
Job API ID:	bbd44e69cb8906b52a6571f0de1fa2cf
History API ID:	4fb08eb7ff7da822
UUID:	ad1af132-c4b3-474f-985d-8995eacdb726

Tool Parameters

Input Parameter	Value
Choose the source for the reference genome	cached
Run in batch mode?	individual
BAM dataset	10: Bowtie2 on data 5: aligned reads (BAM)
Using reference genome	hg19
Limit variant calling to a set of regions?	limit_by_region
Region Chromosome	chr10
Region Start	43605073
Region End	43605710
Choose parameter selection level	simple_w_filters
Require at least this coverage to process a site	0
Job Resource Parameters	no

Figura 56. FreeBayes calling

De vuelta a IGV, tras la exportación del archivo .vcf de freebayes, se obtienen las posiciones de variantes por la zona que se ha analizado. En el ejemplo, se muestran 3 bases que contienen varias variantes en la zona cromosómica elegida. A contiuación se muestra una de ellas tanto en IGV como en genome.ucsc.edu:

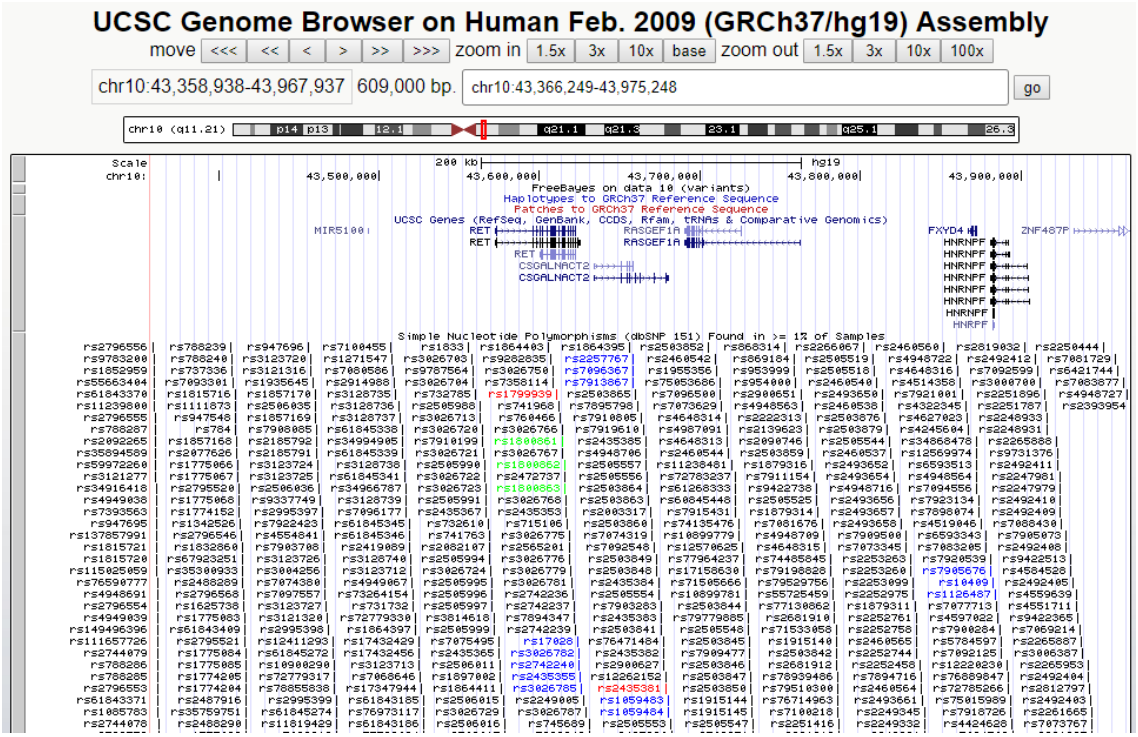


Figura 57. chr10:43,366,249-43,975,248

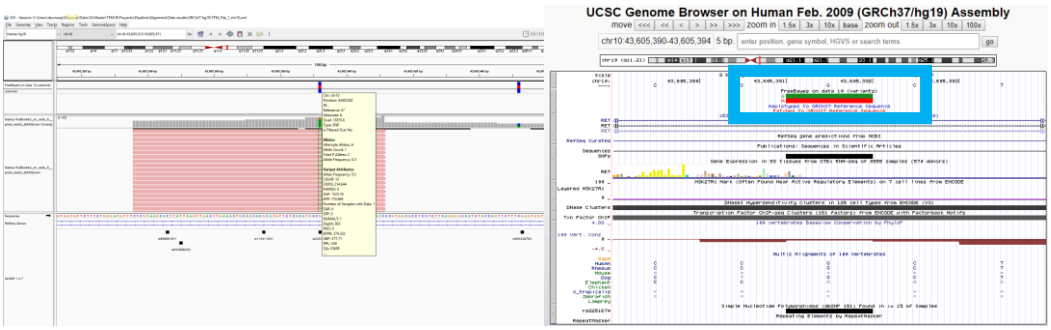


Figura 58. SNP en chr10:43,605,313-43,605,471

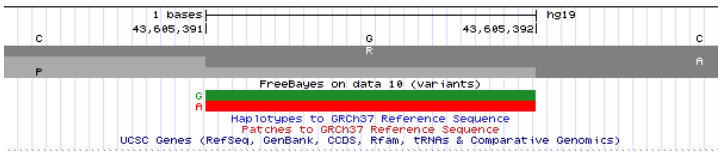


Figura 59. Polimorfismo en chr10:43,605,313-43,605,471

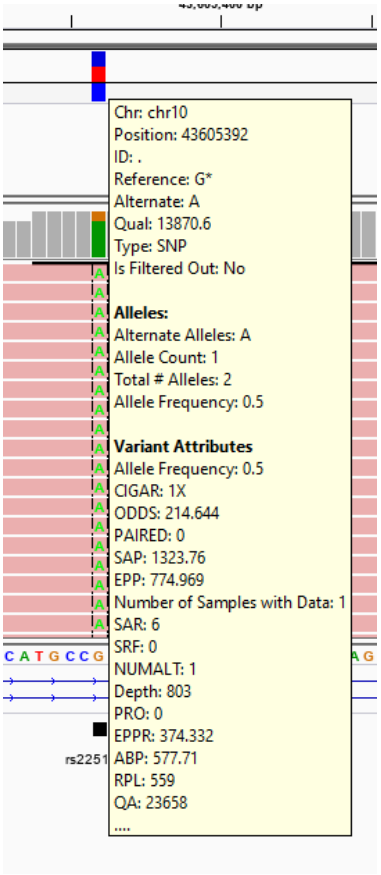


Figura 60. Atributos de la variante

Evaluación de variantes:

Los SNP en color rojo representan cambios en los aminoácidos.
Los SNP en color verde representan variantes sinónimas.
Los SNP en color azul son regiones traducidas o empalmadas.
Los SNP en negro son regiones de intrones.

El filtrado y evaluación del impacto de las variantes se lleva a cabo analizando la localización de los SNPs con respecto a los genes anotados del genoma de referencia. Esta anotación clasifica a los SNP en diferentes tipos según su diferente impacto funcional:

- SNPs sinónimos que no causan alteración de la secuencia de proteína codificada por ese gen.
- SNPs missense que sí alteran la secuencia de proteína.
- SNPS que producen la ganancia de un codón STOP.
- SNPs que producen la pérdida de un codón de inicio.
- SNPs en una región de secuencias repetitivas.
- SNPs o en una región no codificante.

Una vez observado el SNP, podemos ver la frecuencias de los alelos correspondiente:

A: 0.003% (4 / 123244);
C: 79.557% (98049 / 123244);
G: 20.440% (25191 / 123244)

Las anotaciones por dbSNP:

RET (NM_020630): synonymous_variant S (TCC) --> S (TCG)
RET (NM_020975): synonymous_variant S (TCC) --> S (TCG)

Las anotaciones UCSC:

Esta sustitución de base única es tri-alélica.

Y la predicción relativa a los tracks del gen seleccionado.

RET (uc010qez.1) synonymous_variant S (TCC) → S (TCA) → STOP
RET (uc010qez.1) synonymous_variant S (TCC) → S (TCG) → Cysteina [15]
RET (uc001jal.3) synonymous_variant S (TCC) → S (TCA) → STOP
RET (uc001jal.3) synonymous_variant S (TCC) → S (TCG) → Cysteina[15]
RET (uc001jak.1) synonymous_variant S (TCC) → S (TCA) → STOP
RET (uc001jak.1) synonymous_variant S (TCC) → S (TCG) → Cysteina[15]

Validación:

Una vez visualizados, es posible comparar las variantes obtenidas con las anotadas en las base de datos dbSNP para su posterior validación.
La Base de Datos de Polimorfismo de Nucleótido Único (dbSNP) es un archivo público gratuito para la variación genética dentro y entre las diferentes especies, desarrollado y hospedado por el Centro Nacional de Información Biotecnológica en colaboración con el Instituto Nacional de Investigación del Genoma Humano.

Pinchando sobre el SNP que aparece puede ver la anotación que hay en dbSNP sobre esta variante:

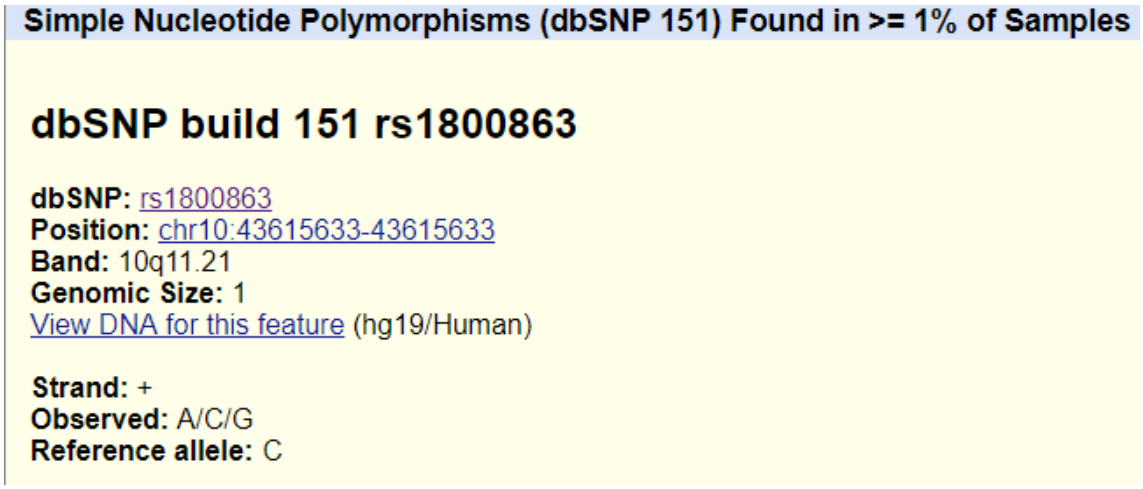


Figura 61. dbSNP

Efectos biológicos de las variantes

Se puede observar que los efectos biológicos de las varaintes presentadas, hacen referencia a variaciones de intrones. Observando una realinación de las secuencias flanqueantes de SNP a la secuencia genómica, se muestran las diferencias siguientes.
En el anexo 3 se adjunta un enlace para la consulta de los archivos generados en el proceso de mapeo, filtraje, anotación y variante procedentes de Galaxy.

Genomic sequence around rs1800863 (chr10:43615133-43616133, + strand):

CGACCTCATCTCATTTGCCTGGCAGATCTCACAGGGGATGCAGTATCTGGCCGAGATGAAGGTGCGTGCATATGGCTCTGCACCCAGCCAGCCCCGgcca
ggccacaccctgacccaccacgcccctgccacccacaccctggcctgccactcccccaccatgccacactctagcccaccatgcccctgccatggcatgc
catgctatggctcaccacgcccctgccatgtcacaccctgactccaccacgcccctgccatgccacaccCCCGGCCCAGGTCTCACCAGGCCGCTACCCG
GGCCACACACCACCCCCTCTGCTGGTCACACCAGGCTGAGCCAGTGACCGCTGCTGCCTGGCCATGGCCTGACGACTCGTGCTATTTTTCCTCACAGCTCG
TTCATCGGGACTTGGCAGCCAGAAACATCCTGGTAGCTGAGGGGCGGAAGATGAAGATTTCGGATTTCGGCTTGTCCCGAGATGTTTATGAAGAGGATTC
C
TACGTGAAGAGGAGCCAGGTGCCCAGTCCCGGGGATGAGGCGGGGCTCCCAGGGATCCCAGGTGCACCATGGGGCAGGCAGTGCCCTTGGGAAGCCTAGG
AAAGATACCGAAGATTAGTGGAGCTCTAAGCTTTTTATAGCCCTCACCCCAAATCTTTCTGACCCTGGGTCCCCAAGGACCCAATTAGAACTCCGCTCAG
CCTCTGCCATGTCCTTCTCCTCCAGGGCCTCCAGGGCACCCCTCCCTGGCAGCATACTGACCCGAGGCCCTTGCCGCACTTTTCAGAGGCCACCTCATGC
TGCGGAACTAACAGTCCTCTTCTGCAGAATAAAGGTCACCGTTCTGATATGACCTTAGCTCTTTTCTCAAAGAAGGGTGGGATGAAATTAGCAGGATCGT
CATTCTTTGCAAAAAGGAATGAACTGCTTTACAAGTGAGGCTTCTCCCGCACAGGGGCCTTGGACACTGGGCTGGGTGAGTTTAGAGGCATAGGAACCCC

dbSNP flanking sequences and observed allele code for rs1800863:

(Uses IUPAC ambiguity codes)

CGACCTCATCTCATTTGCCTGGCAGATCTCACAGGGGATGCAGTATCTGGCCGAGATGAAGGTGCGTGCATATGGCTCTGCACCCAGCCAGCCCCGGCCA
GGCCACACCCTGACCCACCACGCCCCTGCCACCCACACCCTGGCCTGCCACTCCCCCACCATGCCACACTCTAGCCCACCATGCCCCTGCCATGGCATGC
CATGCTATGGCTCACCACGCCCCTGCCATGTCACACCCTGACTCCACCACGCCCCTGCCATGCCACACCCCCGGCCCAGGTCTCACCAGGCCGCTACCCG
GGCCACACACCACCCCCTCTGCTGGTCACACCAGGCTGAGCCAGTGACCGCTGCTGCCTGGCCATGGCCTGACGACTCGTGCTATTTTTCCTCACAGCTCG
TTCATCGGGACTTGGCAGCCAGAAACATCCTGGTAGCTGAGGGGCGGAAGATGAAGATTTCGGATTTCGGCTTGTCCCGAGATGTTTATGAAGAGGATTC
V
TACGTGAAGAGGAGCCAGGTGCCCAGTCCCGGGGATGAGGCGGGGCTCCCAGGGATCCCAGGTGCACCATGGGGCAGGCAGTGCCCTTGGGAAGCCTAGG
AAAGATACCGAAGATTAGTGGAGCTCTAAGCTTTTTATAGCCCTCACCCCAAATCTTTCTGACCCTGGGTCCCCAAGGACCCAATTAGAACTCCGCTCAG
CCTCTGCCATGTCCTTCTCCTCCAGGGCCTCCAGGGCACCCCTCCCTGGCAGCATACTGACCCGAGGCCCTTGCCGCACTTTTCAGAGGCCACCTCATGC
TGCGGAACTAACAGTCCTCTTCTGCAGAATAAAGGTCACCGTTCTGATATGACCTTAGCTCTTTTCTCAAAGAAGGGTGGGATGAAATTAGCAGGATCGT
CATTCTTTGCAAAAAGGAATGAACTGCTTTACAAGTGAGGCTTCTCCCGCACAGGGGCCTTGGACACTGGGCTGGGTGAGTTTAGAGGCATAGGAACCCC

Figura 62. Secuencia genómica y alelos observados alrededor de rs2251674

Alignment between genome (hg19 chr10:43615133-43616133, + strand; 1001 bp) and dbSNP sequence (rs1800863; 1001 bp)
ID (including gaps) 99.9%, coverage (of both) 100.0%

```
43615133 CGACCTCATCTCATTTGCCTGGCAGATCTCACAGGGGATGCAGTATCTGGCCGAGATGAAGGTGCGTGCATATGGCTCTGCACCCAGCCAGCCCCGgcca 43615232
         ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||++++
00000001 CGACCTCATCTCATTTGCCTGGCAGATCTCACAGGGGATGCAGTATCTGGCCGAGATGAAGGTGCGTGCATATGGCTCTGCACCCAGCCAGCCCCGGCCA 00000100

43615233 ggccacaccctgacccaccacgcccctgccacccacaccctggcctgccactcccccaccatgccacactctagcccaccatgcccctgccatggcatgc 43615332
         ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
00000101 GGCCACACCCTGACCCACCACGCCCCTGCCACCCACACCCTGGCCTGCCACTCCCCCACCATGCCACACTCTAGCCCACCATGCCCCTGCCATGGCATGC 00000200

43615333 catgctatggctcaccacgcccctgccatgtcacaccctgactccaccacgcccctgccatgccacaccCCCGGCCCAGGTCTCACCAGGCCGCTACCCG 43615432
         ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
00000201 CATGCTATGGCTCACCACGCCCCTGCCATGTCACACCCTGACTCCACCACGCCCCTGCCATGCCACACCCCCGGCCCAGGTCTCACCAGGCCGCTACCCG 00000300

43615433 GGCCACACACCACCCCCTCTGCTGGTCACACCAGGCTGAGCCAGTGACCGCTGCTGCCTGGCCATGGCCTGACGACTCGTGCTATTTTTCCTCACAGCTCG 43615532
         ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
00000301 GGCCACACACCACCCCCTCTGCTGGTCACACCAGGCTGAGCCAGTGACCGCTGCTGCCTGGCCATGGCCTGACGACTCGTGCTATTTTTCCTCACAGCTCG 00000400

43615533 TTCATCGGGACTTGGCAGCCAGAAACATCCTGGTAGCTGAGGGGCGGAAGATGAAGATTTCGGATTTCGGCTTGTCCCGAGATGTTTATGAAGAGGATTC 43615632
         ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
00000401 TTCATCGGGACTTGGCAGCCAGAAACATCCTGGTAGCTGAGGGGCGGAAGATGAAGATTTCGGATTTCGGCTTGTCCCGAGATGTTTATGAAGAGGATTC 00000500

43615633 C 43615633

00000501 V 00000501

43615634 TACGTGAAGAGGAGCCAGGTGCCCAGTCCCGGGGATGAGGCGGGGCTCCCAGGGATCCCAGGTGCACCATGGGGCAGGCAGTGCCCTTGGGAAGCCTAGG 43615733
         ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
00000502 TACGTGAAGAGGAGCCAGGTGCCCAGTCCCGGGGATGAGGCGGGGCTCCCAGGGATCCCAGGTGCACCATGGGGCAGGCAGTGCCCTTGGGAAGCCTAGG 00000601

43615734 AAAGATACCGAAGATTAGTGGAGCTCTAAGCTTTTTATAGCCCTCACCCCAAATCTTTCTGACCCTGGGTCCCCAAGGACCCAATTAGAACTCCGCTCAG 43615833
         ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
00000602 AAAGATACCGAAGATTAGTGGAGCTCTAAGCTTTTTATAGCCCTCACCCCAAATCTTTCTGACCCTGGGTCCCCAAGGACCCAATTAGAACTCCGCTCAG 00000701

43615834 CCTCTGCCATGTCCTTCTCCTCCAGGGCCTCCAGGGCACCCCTCCCTGGCAGCATACTGACCCGAGGCCCTTGCCGCACTTTTCAGAGGCCACCTCATGC 43615933
         ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
00000702 CCTCTGCCATGTCCTTCTCCTCCAGGGCCTCCAGGGCACCCCTCCCTGGCAGCATACTGACCCGAGGCCCTTGCCGCACTTTTCAGAGGCCACCTCATGC 00000801

43615934 TGCGGAACTAACAGTCCTCTTCTGCAGAATAAAGGTCACCGTTCTGATATGACCTTAGCTCTTTTCTCAAAGAAGGGTGGGATGAAATTAGCAGGATCGT 43616033
         ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
00000802 TGCGGAACTAACAGTCCTCTTCTGCAGAATAAAGGTCACCGTTCTGATATGACCTTAGCTCTTTTCTCAAAGAAGGGTGGGATGAAATTAGCAGGATCGT 00000901

43616034 CATTCTTTGCAAAAAGGAATGAACTGCTTTACAAGTGAGGCTTCTCCCGCACAGGGGCCTTGGACACTGGGCTGGGTGAGTTTAGAGGCATAGGAACCCC 43616133
         ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
00000902 CATTCTTTGCAAAAAGGAATGAACTGCTTTACAAGTGAGGCTTCTCCCGCACAGGGGCCTTGGACACTGGGCTGGGTGAGTTTAGAGGCATAGGAACCCC 00001001
```

Figura 63. Alineación entre el genoma (hg19 chr10: 43604892-43605892, cadena +; 1001 pb) y la secuencia dbSNP (rs2251674; 1001 pb)

Buscando información sobre la variante, se encuentra que existe una posible relación entre las alteraciones mencionadas y el cáncer de tiroides de origen folicular y posible impacto del polimorfismo RET y su asociación haplotípica modula la susceptibilidad al cáncer de tiroides [15] [16].

8. Conclusiones

Con este trabajo se ha representado un pipeline genérico partiendo de unos datos en crudo y pudiendo ir paso a paso en la evaluación y obtención de resultados en cada uno de los pasos. Además, puede utilizarse como guía para cualquier otro tipo de estudio siempre y cuando se parta de unos archivos con extensión fastq.

Con la utilización del lenguaje R y algunas aplicaciones interactivas como IGV para las visualizaciones o UCSC Genome Browser se ha sido capaz de mostrar visualmente unas alteraciones en las cadenas representadas y por consiguiente unas variaciones que podrían tener relación con el cáncer de pulmón, aunque se ha encontrado una relación con una mutacion del gen en cuestión, relacionada a una enfermedad pedriátrica, encontrada en la población China (posible causante de la enfermedad de Hirschsprungs).

Si evaluamos el alcance de los objetivos propuestos podemos concluir que han sido alcanzados, ya que sí que se ha conseguido generar un estudio/pipeline, dada una región específica y partiendo de unos datos en bruto. Este ejercicio puede servir de ayuda y/o complemento para una mejor representación paso a paso de un estudio genético, ya que en ciertos puntos del trabajo, puede permitir afinar y filtrar mucha mas información según se crea.

Como parte de mejora, se podría proponer una mayor ampliación en la búsqueda de posibles variante a lo largo de todo el cromosoma 10 y sus posibles afecciones y relaciones con el cáncer de pulmón.

También se podría generar una aplicación mediante shiny, que una vez cargados los datos fastq, esta permitiera la transformación a BAM file, para su posterior alineación y anotación, así como estudio de variantes. En el alcance del proyecto, esta parte se estimo como posible, si en tiempo y recursos era factible. No obstante, debido a la carga de trabajo actual se cree interesante dejarlo para una línea de exploración futura. En este ámbito, resaltar que se ha añadido en el Anexo 2 el marco de la aplicación, así como su código, y listar el archivo cargado en fastq.

https://tfmbioinformatica2019.shinyapps.io/TFM_BIO/

En rasgos generales, el estudio, ha sido una buena herramienta para la alineación de conceptos, así como su estructuración. La metodología a seguir además de la gran cantidad de ayudas en red y aplicaciones, hace que muchas veces los conceptos no queden del todo alineados. Con este estudio, se ha intentado crear un tipo de pautado sobre una posible vía a trabajar con archivos en bruto.

9. Glosario

FASTQ: almacena secuencias y la información de calidad de la secuencia juntos.

SAM / BAM: El Mapa de Alineación de Secuencias es un formato basado en texto originalmente para almacenar secuencias biológicas alineadas con una secuencia de referencia desarrollada por Heng Li y Bob Handsaker et al. El formato SAM consiste en un encabezado y una sección de alineación.El equivalente binario de un archivo SAM es un archivo de mapa de alineación binaria (BAM), que almacena los mismos datos en una representación binaria comprimida.

Secuenciación: La secuenciación de ADN es el proceso que determina la secuencia de bases de los nucleótidos (As, Ts, Cs y Gs) de un fragmento de ADN. Hoy en día, con el equipo y los materiales adecuados, secuenciar un fragmento pequeño de ADN es relativamente sencillo.

NGS: Secuenciación de Segunda Generación. Las técnicas de secuenciación de nueva generación son estrategias nuevas a gran escala que aumentan la velocidad y reducen el costo de la secuenciación del ADN.

Heterocigoto: De un organismo o individuo diploide se dice en genética que es heterocigoto para un gen determinado cuando los cromosomas homólogos portadores de ese locus presentan allí dos versiones distintas del gen, es decir, dos alelos. La condición de heterocigoto se denomina heterocigosis.

SNP: Un polimorfismo de un solo nucleótido o SNP (Single Nucleotide Polymorphism, pronunciado snip) es una variación en la secuencia de ADN que afecta a una sola base (adenina (A), timina (T), citosina (C) o guanina (G)) de una secuencia del genoma.

dbSNP: base de datos global de todos los SNPs. Está contenida en NCBI (del inglés, National Center for Biotechnology Information).

Alineamiento: Un alineamiento de secuencias en bioinformática es una forma de representar y comparar dos o más secuencias o cadenas de ADN, ARN, o estructuras primarias proteicas para resaltar sus zonas de similitud, que podrían indicar relaciones funcionales o evolutivas entre los genes o proteínas consultados. Las secuencias alineadas se escriben con las letras (representando aminoácidos o nucleótidos) en filas de una matriz en las que, si es necesario, se insertan espacios para que las zonas con idéntica o similar estructura se alineen.

Mapeo: Un mapa genómico es unidimensional, es decir, es linear como las moléculas de ADN que lo conforman. La diferencia de este con la secuencia genómica consiste en el nivel de detalle mostrado. El mapa genómico es menos detallado que la secuencia genómica.

Anotación: Para poder utilizar la información dentro de la secuencia de un genoma es fundamental anotarla con datos biológicos relevantes. Las anotaciones de un genoma ayudan a los científicos a entender qué significa su secuencia, cómo está estructurado y cómo funciona.

9. Bibliografía

1. https://www.cancer.gov/espanol/cancer/naturaleza/estadisticas (03/2019)
2. https://seom.org/seomcms/images/stories/recursos/Las_cifras_del_cancer_en_Esp_2017.pdf (03/2019)
3. https://www.cancer.org/es/cancer/cancer-de-pulmon-no-microcitico/acerca/estadisticas-clave.html (03/2019)
4. https://www.cancer.org/es/cancer/cancer-de-pulmon-no-microcitico/acerca/estadisticas-clave.html (03/2019)
5. https://www.genome.gov/10001772/all-about-the--human-genome-project-hgp/ (03/2019)
6. Datos fastq files: https://agilent.sharefile.com/d-s0ec0901031443efb (02/2019)
7. https://www.r-project.org/nosvn/pandoc/shiny.html(03/2019)
8. http://shiny.rstudio.com/(03/2019) (03/2019)
9. https://www.ncbi.nlm.nih.gov/genome/51?genome_assembly_id=368248(04/2019) (05/2019)
10. https://seom.org/dmcancer/wp-content/uploads/2019/Informe-SEOM-cifras-cancer-2019.pdf (05/2019)
11. http://genome.ucsc.edu/cgi-bin/hgc?hgsid=728819215_pRkrEVFa2MgykLjfntHWJaPaNSa6&c=chr10&l=43605387&r=43605395&o=43605391&t=43605392&g=snp151Common&i=rs2251674 (05/2019)
12. https://www.ncbi.nlm.nih.gov/snp/rs2251674#variant_details (05/2019)
13. https://es.wikipedia.org/wiki/Histidina (05/2019)
14. https://en.wikipedia.org/wiki/Cysteine (05/2019)
15. https://www.ncbi.nlm.nih.gov/pubmed/30089490 (05/2019)
16. https://www.ncbi.nlm.nih.gov/pubmed/25736215 (05/2019)

17. Langmead B, Wilks C, Antonescu V, Charles R. Scaling read aligners to hundreds of threads on general-purpose processors. Bioinformatics. 2018 Jul 18. doi: 10.1093/bioinformatics/bty648. (05/2019)

18. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature Methods. 2012 Mar 4;9(4):357-9. doi: 10.1038/nmeth.1923. (05/2019)

19. Ekblom R, Wolf JB. A field guide to whole-genome sequencing, assembly and annotation. Evol Appl. 2014 Nov;7(9):1026-42. PMID de PubMed:25553065. Yandell M, Ence D. A beginner's guide to eukaryotic genome annotation. Nat Rev Genet. 2012 Apr 18;13(5):329-42. PMID de PubMed:22510764. (05/2019)

20. Jesús Alvarado Valverde. (2016). Anotación de genoma. 2019, Mayo 29, Conogasi.org Sitio web: http://conogasi.org/articulos/anotacion-de-genoma/. (05/2019)

Bioconductor:

21. https://bioconductor.org/packages/devel/bioc/manuals/ShortRead/man/ShortRead.pdf (04/2019)
22. https://www.bioconductor.org/packages//2.10/bioc/manuals/ShortRead/man/ShortRead.pdf (04/2019)
23. http://bioconductor.org/packages/release/bioc/html/iASeq.html (03/2019)
24. https://es.wikipedia.org/wiki/Compresi%C3%B3n_de_Burrows-Wheeler (04/2019)
25. https://www.bioconductor.org/help/course-materials/2014/SeattleFeb2014/Bioconductor.pdf (04/2019)
26. https://bioconductor.org/packages/release/bioc/manuals/Rsamtools/man/Rsamtools.pdf (04/2019)
27. http://bioconductor.org/packages/release/bioc/vignettes/Rsamtools/inst/doc/Rsamtools-Overview.pdf (04/2019)
28. http://bioconductor.org/packages/release/bioc/vignettes/GenomicAlignments/inst/doc/GenomicAlignmentsIntroduction.pdf (04/2019)
29. https://rpubs.com/sahiilseth/flowr_fq_bam (05/2019)
30. http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml#what-isnt-bowtie-2 (05/2019)
31. https://bioinformatics-core-shared-training.github.io/cruk-summer-school-2018/Introduction/SS_DB/Materials/Practicals/Practical2_alignment_SS.html (05/2019)
32. http://www-dep.iarc.fr/WHOdb/WHOdb.htm (05/2019)
33. http://ci5.iarc.fr/CI5plus/old/Graph4p.asp?cancer%5B%5D=110&male=1&female=2&country%5B%5D=72400000&sYear=1950&eYear=2012&stat=3&age_from=1&age_to=18&orientation=1&window=1&grid=1&line=2&moving=1&scale=0&submit=%C2%A0%C2%A0%C2%A0Execute%C2%A0%C2%A0%C2%A0 (05/2019)
34. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/ (05/2019)
35. http://www.sthda.com/english/wiki/fastqcr-an-r-package-facilitating-quality-controls-of-sequencing-data-for-large-numbers-of-samples (05/2019)
36. http://girke.bioinformatics.ucr.edu/GEN242/pages/mydoc/Rsequences.html (05/2019)
37. http://bowtie-bio.sourceforge.net/bowtie2/faq.shtml (05/2019)

10. Anexos

Anexo 1: Código en R.

Fast Q Pipeliner analysis

David Masip Galaso

June 23th, 2019

Abstract

En la actualidad, el cáncer de pulmón es uno de los tipos de cáncer más comunes y de los más extensos. En comparación con el resto de cánceres, alrededor del 14% de todos los cánceres nuevos son cánceres procedentes de este órgano. Debido a las altas estadísticas, se cree interesante envolver el presente trabajo de Fin de Máster con la realización de un pipeline bioinformático a partir de un panel de genes de cáncer de pulmón. Para ello, se parte de una base con datos tipo "Fastq files" procedente de un panel de genes para su posterior parametrización.

Con el presente objetivo y en plena expansión e integración genómica en el campo del diagnóstico clínico, encontramos una razón de peso para la ejecución de dicho particular estudio dentro del campo de diagnóstico. El tipo de lenguaje utilizado para este proyecto será la plataforma de software libre "R" y sus distintos paquetes de Bioconductor utilizados a lo largo del pipeline, los cuáles nos permitirán filtrar y definir con gran detalle toda la información proveniente de los archivos seleccionados, sacar datos estadísticos representarlos de una forma rápida, simple, coherente y de la mejor manera posible.

Esta aplicación puede utilizarse para cualquier tipo de estudio siempre y cuando el punto de inicio sea una base con datos de clase fastq files, ya sea precedente de Pubmed, de cualquier tipo de panel de genes procedentes de un secuenciador o de cualquier otra fuente. Al publicarse en la red, ésta además puede ser utilizada por cualquier usuario, incluso aquellos sin conocimientos en lenguajes de programación específico.

Descripción General

Esta parte del análisis se utilizan datos de un determinado panel de cáncer de pulmón de ADN proveniente de secuenciadores de Illumina, MiSeq con una extensión de 76bp y paired end reads. El alcance tendrá a cargo la lectura, la alineación, el resumen de las lecturas alineadas y el trazado, asi como de sus anotaciones.

Platforma usada y tipo de experimento

Archivos de datos Fastq y algunos paquetes de bioconductores como "ShortRead", "Bowtie2", "SAMtools" entre otros.

Chunk 1: Directories creation

Establecer directores si no están creados:

```
workingDir <- getwd()
if(!dir.exists("data")) dir.create("data")
dataDir <-file.path(workingDir, "data")
if(!dir.exists("results")) dir.create("results")
resultsDir <- file.path(workingDir,"results")
```

Chunk 2: GetData from Folder

Lectura de datos

Escogeremos sólo una lectura porque son bastante pesadas, aunque nos podríamos basar en cualquier archivo para elegirlas.

```
library(ShortRead)
showMethods(readFastq)

## Function: readFastq (package ShortRead)
## dirPath="character"
## dirPath="FastqFile"
## dirPath="SolexaPath"

getwd()

## [1]
"C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline"

fastqDir <-
file.path("C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pip
eline/data/File")
fastqFiles <- dir(fastqDir, full=TRUE)
fastqFiles

## [1]
"C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline/data
/File/15-0991_S1_L001_R1_001.fastq"
## [2]
"C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline/data
/File/15-0991_S1_L001_R2_001.fastq"

#Dentro de la ubicación de los archivos, decidimos elegir por ejemplo
el primer de los archivos

fq <- readFastq(fastqFiles)
sequenceOfReads <- sread(fq)
class(sequenceOfReads)

## [1] "DNAStringSet"
## attr(,"package")
## [1] "Biostrings"
```

Chunk 3: Comprobación general

Analizamos ancho de lectura y se representa por ejemplo las 10 primeras DNAStringSet. Los datos representados son objetos de clase ShortReadQ.

```
readLengths <- width(fq)
readLengths[1:3]

## [1] 76 76 76

head(sread(fq), 3)

##   A DNAStringSet instance of length 3
##     width seq
## [1]    76
CTTAAAACTGATTTTACATGGTACATGAAAC...CGATTTTTTTCTTCCTTCTGCTCCTTCCCCT
## [2]    76
CTGTGCTTGCACCTGACATCCTCGTGCTCCT...ATTTTATCTTTTTTTTTTTTTTTTTTTCTCT
## [3]    76
CATTTCTTTGGTAAAAAGCTGAAGCCAGAGA...CTTTAATCTCCCTTCTCTTTTTCCTCCCTCC
```

Chunk 4: Comprobación de calidad de los datos.

De forma genérica, el paquetes ShortRead inclue una función para genera un informe de control de calidad simple. En nuestro estudio, para no alargar la secuencia, vamos a utilizar un sólo archivo, aunque realmente la función se podría ejecutar sobre todos los archivos. Dicha función accepta un archivo tipo FastQ file y devuelve un objeto.

```
library(stats)
qa <-
qa("C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline/data/File", pattern = "15-0991_S1_L001_R1_001.fastq")
qa

## class: FastqQA(10)
## QA elements (access with qa[["elt"]]):
##   readCounts: data.frame(1 3)
##   baseCalls: data.frame(1 5)
##   readQualityScore: data.frame(512 4)
##   baseQuality: data.frame(95 3)
##   alignQuality: data.frame(1 3)
##   frequentSequences: data.frame(50 4)
##   sequenceDistribution: data.frame(365 4)
##   perCycle: list(2)
##     baseCall: data.frame(304 4)
##     quality: data.frame(1943 5)
##   perTile: list(2)
##     readCounts: data.frame(0 4)
##     medianReadQualityScore: data.frame(0 4)
##   adapterContamination: data.frame(1 1)
```

Una vez analizado mediante la función, existe la posibilidad de generar un report mediante la siguiente función:

```
myReport <- report(qa)
myReport

## [1]
"C:\\Users\\davmasip\\AppData\\Local\\Temp\\RtmpmuA4UF\\file3e8cb9d16e8/index.html"
```

En particular, analizando los resultados, se puede apreciar claramente en los gráficos de calidad que no existe apenas contaminación. Esto es debido al tipo de archivo utilizado, en nuestro scope, estos archivos provienen de un panel de genes de cancer de pulm?n, son archivos sin contaminación y listos para poder alinearlos con el Genoma de referencia. En el estudio se hace un detalle sobre la posibilidad de la interpretación cualitativa debido a que dicho pipeline puede ser ampliado a otros archivos fastq files, previamente sin limpiar y en los cuales se deberá hacer un paso de filtering and trimming previa a la alineación.

Al ser archivos tipo paired end read, miramos la calidad de ambas de las lecturas, para poder ver la posible contaminación entre el anverso y el reverso de las consiguientes cadenas.

```
library(dada2)
packageVersion("dada2")

## [1] '1.12.1'

path <-
"C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline/data
/File"
list.files(path)

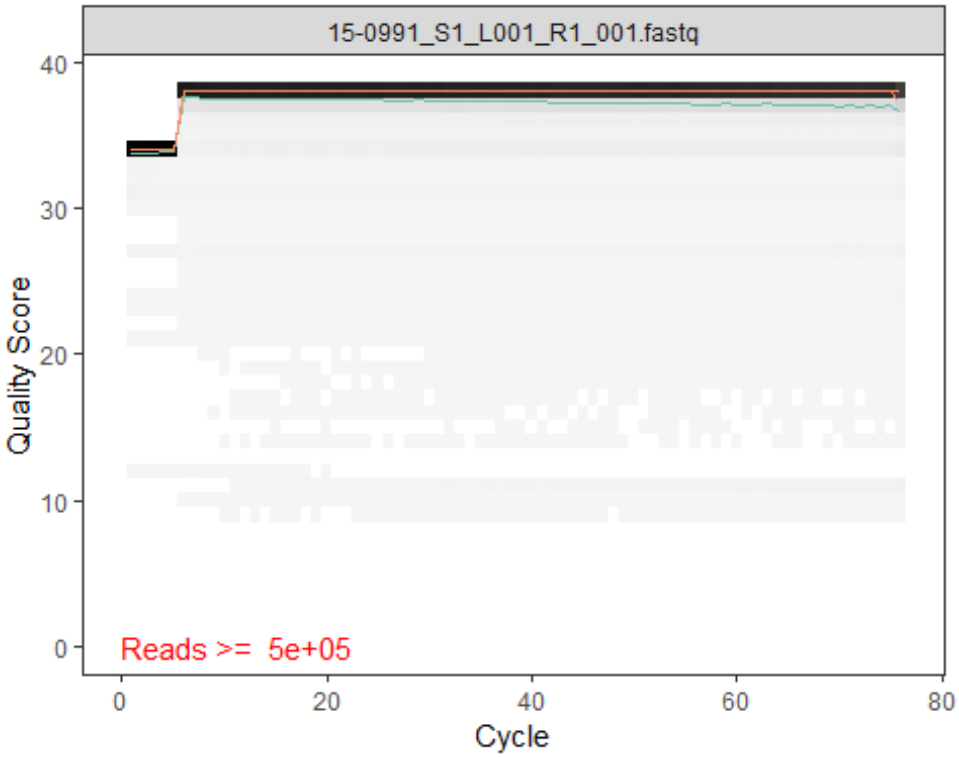
## [1] "15-0991_S1_L001_R1_001.fastq" "15-0991_S1_L001_R2_001.fastq"

fnFs <- sort(list.files(path, pattern="15-0991_S1_L001_R1_001.fastq",
full.names = TRUE))
fnRs <- sort(list.files(path, pattern="15-0991_S1_L001_R2_001.fastq",
full.names = TRUE))

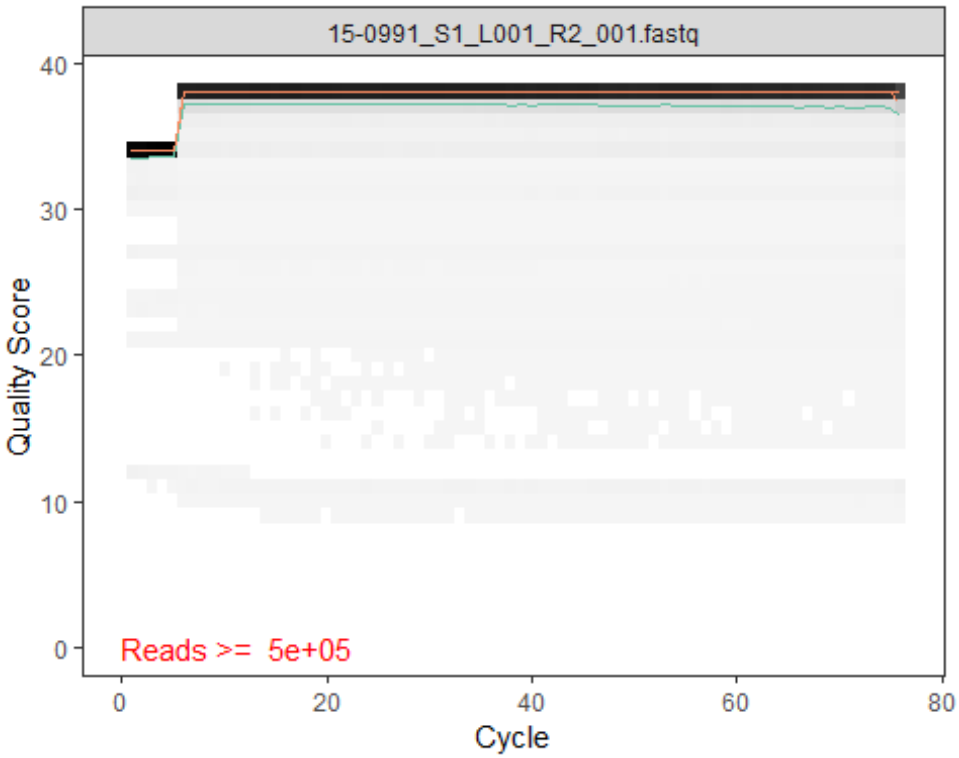
# Extract sample names, assuming filenames have format:
SAMPLENAME_XXX.fastq

sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`, 1)
sample.names2 <- sapply(strsplit(basename(fnRs), "_"), `[`, 1)

plotQualityProfile(fnFs[1:2])
```




```
plotQualityProfile(fnRs[1:2])
```



En escala de grises, se plasma la frecuencia del resultado de cada muestra en la posici?n de cada base. La calidad media se muestra en forma de l?nea verde y los quartiles de distribucu?n de resultados sobre lal ?nea de color naranja. En color roja, se muestra la proporci?n de lecuras qeu se extienden almenos sobre esa posici?n. En particular, como los fastq files se secuencian sobre plataforma Illumina, ?stas lecturas son t?picamsente todas de la misma longitud, por lo tanto, de ah? que me muestre una l?nea de color rojo, completamente plana.

Dicha lectura tiene buena calidad. Generalmente se advierte un proceso de triming sobre los ?ltimos nucle?tidos para tener un mayor control sobre errores y tenerlos bien controlados. Estos perfiles cualitativos no sugieren que es necesario un trimming.

Si en alguna de las lecturas, sufre una pérdida en cuanto a calidad, por ejemplo en el fin de la secuencia, DADA2 incorpora informaci?n de calidad en su modelo de error, lo que hace que el algoritmo sea robusto a una secuencia de calidad inferior, pero el recorte a medida que se desplome la calidad promedio mejorar? la sensibilidad del algoritmo a variantes de secuencia raras. Sobre la base de estos perfiles, truncaremos las lecturas inversas en la posici?n 160 donde se bloquea la distribuci?n de calidad. Cogeremos una anverso y un reverso de una de las muestras.

```
#library(Rqc)
#qa<-rqc(path =
"C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline/data
/File", pattern = ".fastq")
```

Chunk 5: Graficos

El paquete Rqc acepta el formato de archivo R Markdown como archivo de plantilla para generar informes personalizados. Markdown es un lenguaje de marcado para el desarrollo web. Los archivos R Markdown son archivos Markdown normales con c?digos R. Cada fragmento de c?digo se ejecuta durante la compilaci?n realizada por el paquete knitr. Knitr toma el archivo Markdown R y genera un archivo Markdown combinado con los c?digos R y sus salidas, como texto, tablas y figuras. Rqc usa el archivo Result Markdown para generar el informe final en formatos HTML o PDF. El archivo de origen del informe predeterminado de Rqc es una buena referencia para escribir nuevos informes de plantilla. El c?digo de ejecuci?n a continuaci?n devuelve la ruta del archivo del sistema a este archivo fuente.

Los datos de resultados de Rqc est?n disponibles dentro de los archivos de plantilla a trav?s del objeto rqcResultSet. Este objeto es una lista de estad?sticas resumidas sobre los archivos de entrada y es utilizado por todas las funciones de acceso y gr?ficos proporcionados por el paquete Rqc. La funci?n rqcReport toma la ruta del archivo de plantilla como argumento y genera informes personalizados.

Para cada gr?fico generado por Rqc, hay una funci?n que da forma a los datos de manera apropiada. La informaci?n configurada se utiliza para producir la trama final. El siguiente ejemplo muestra c?mo el usuario puede acceder a estos datos para generar gr?ficos con otras herramientas.

```
#df <- rqcCycleAverageQualityCalc(qa)
#cycle <- as.numeric(levels(df$cycle))[df$cycle]
#plot(cycle, df$quality, col = df$filename, xlab='Cycle',
ylab='Quality Score')
#sublist <- qa
#rqcCycleQualityPlot(sublist)
```

Chunk 6: Calidad por secuencia

Por otro lado, si analizamos analogamente la calidad de los archivos, ?stos los ejecuta a partira partir de la funci?n quality. Con esta funci?n y teniendo en cuenta

el abecedario de valores, seremos capaces de obtener una primera aproximaci?n sobre la calidad de nuestra muestra. En nuestro caso y como se ha comentado anteriormente, no es necesario pasar por dicho proceso, pero a modo de ejemplo se va a ejemplificar las 10 primeras cadenas escogidas como muestra, se muestra el alfabeto de calidad correspondiente a cada uno de los archivos.

! " # $ % & ’ ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 : ; < = > ? @ A B C D E F G H I J 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41

```
head(quality(fq), 3)
```

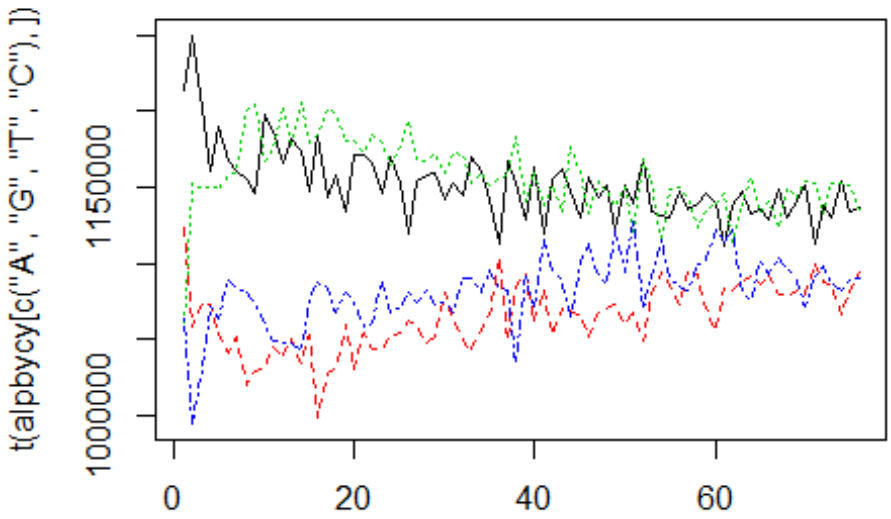
```
## class: FastqQuality
## quality:
##   A BStringSet instance of length 3
##     width seq
## [1]    76
@BCCCGAEFC9FFGGFFDGGFF9@,E,,;,C...6,+,,,,<@@@C,<,;,;,,6<,;,9,9,9,
## [2]    76
8A@C@FGGGFCGGGGDFDGGFGF7D,BC<CE..,,,,,,,,;,,<,,++++8+8+++++++9,99
## [3]    76
<6BCCGGGGCFGFGGGGGGGF9<,;CC,,,,,,.;C,,,,,;<6,;C,6;CC,<,,;6,666@68
```

Hay varios formas de manipular dichos objetos. Mediante la funci?n “alphabetByCycle”, se obtiene un sumario de los nucl?otidos usados en cada ciclo es una instancia de igual tama?o que mediante ShortReadQ o DNAStringSet. En nuestro caso, se muestran las 4 bases sobre los primeros 8 ciclos.

```
alpbycy <- alphabetByCycle(sread(fq))
alpbycy[1:4, 1:8]
```

```
##          cycle
## alphabet    [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]
[,8]
##        A 1213777 1249765 1202169 1160812 1189462 1167489 1160641
1154840
##        C 1063287  995047 1030747 1072223 1063897 1089419 1083337
1080841
##        G 1124176 1059069 1074543 1072726 1054755 1041417 1052263
1020263
##        T 1055390 1152749 1149171 1150869 1148516 1158305 1160389
1200686
```

```
matplot(t(alpbycy[c("A","G","T","C"),]), type="l")
```

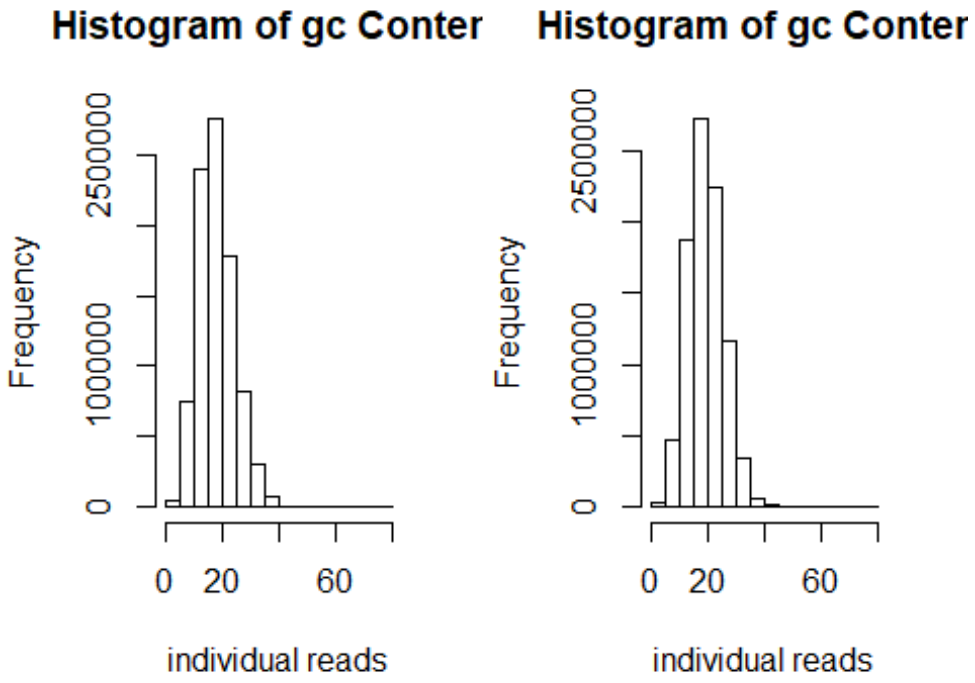


Y mediante otras funciones como alphabetFrequency, se obtiene el sumario de nucle?tidos por archivo.En nuestro caso hemos seleccionado los 10 primeros:

```
alpFreq <- alphabetFrequency(sread(fq))
alpFreq[1:10,]

##        A  C  G  T M R W S Y K V H D B N - + .
##  [1,] 20 19  9 28 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [2,]  6 18  8 44 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [3,] 16 21 10 29 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [4,] 23 15 10 28 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [5,] 11 29 11 25 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [6,]  8 28 11 29 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [7,]  5 33 15 23 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [8,] 10 34 12 20 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [9,] 28 16 10 22 0 0 0 0 0 0 0 0 0 0 0 0 0
## [10,] 23 11  9 33 0 0 0 0 0 0 0 0 0 0 0 0 0

par(mfrow=c(1,2))
hist(alpFreq[,c("G", "C")],
     main = "Histogram of gc Content",
     xlab="individual reads" )
hist(alpFreq[,c("A", "T")],
     main = "Histogram of gc Content",
     xlab="individual reads" )
```



Asi como obtener la suma -log10 de los pvalues mediante la funci?n alphabetScore para todas las lecturas. En nuestro caso de la 1 a la 10.

```
readScores<- alphabetScore(fq)
readScores[1:10]

##  [1] 1731 1825 1898 1963 1781 1848 1928 1934 1862 1789
```

En algunos casos, las secuencias de los archivos fastq files son extremadamente largas, como podr?a ser el caso de un fastq de exoma. En nuestro particular, como los datos provienen de un panel de genes de c?ncer de pulm?n, son secuencias muy cortas y no es necesario sacar una muestra para el an?lisis. Sin embargo, s?lo mencionar, que existe dicha posibilidad. En un ejemplo cualquiera, la nomenclatura a seguir ser?a la siguiente, para una extracci?n de muestra de 1M de lecturas:

```
sampler <- FastqSampler(fastqFiles[1], 1000000)

yield(sampler) # sample of 1000000 reads
```

Del mismo modo, FastqStreamer() trabaja de un modo parecido, ejecutando de manera aleatoria una serie de muestras del archivo.

Por otro lado, se puede usar la funci?n “tabla”, para identificar el n?mero de veces qeu una secuencia aparece en la lectura de nuestro archivo fastQ.

```
readOccurence <- table(sread(fq))
sort(readOccurence,decreasing = TRUE)[1:25]

##
##
GTCCCCGGCCTGGCAGGGCGCCCTGGAGTGGGAGGAAGAGGTAACCACAGGGGGGCTGGAGCTGGCCTCG
GACTTG
```

##
1703
##
GCCCCTCCCGGGAAGGTGCCGTCTCCTCCGGCCCCTCGGGTCCCTGCTCTGTCACTGACTGCTGTGACCC
ACTCTG
##
1675
##
AGCACATCTGCCACCACTTGCACTGCGGTCTGGCTAACCACATGAGCATGGCCACTGATGAGGTGGATGG
AGGGTG
##
1674
##
AAGGTTGCACTTGTCCACGCATTCCCTGCCTCGGCTGACATTCCGGCAAGAGACGCAGTCCCTGGGCTCC
GGGCCC
##
1665
##
GCAGCACACAGCCCTGCCAAGACAGTCCCCGCTACAACCCCAGCCCTCCCAAGACTGGGGCTACCGTCTG
ACCCTG
##
1646
##
CAGCCCATGCACCGCTACGACGTGAGCGCCCTGCAGTACAACTCCATGACCAGCTCGCAGACCTACATGA
ACGGCT
##
1638
##
CTCAGCAAGGAAGACCTTCCCAAAGGCGCCCTCCCCCAGCTCCCACTTGAGCACGATGTCCCGGCGCTTG
ATGTGG
##
1638
##
TCAACACCGCTTTGCCGATAGAATATGGTCCACTTGTGGAAGAGGAAGAGAAAGTGCCTGTGAGGCCCAA
GGACCC
##
1625
##
CACCAGAGCAGCTGCAGTTCCCTGAGGAGCCCCTGATTCTGCACCTCAGCCCCGTGTGTATCCTCCTGGC
TGATCA
##
1516
##
GGCCACAGCGTCTGCTCCACCTCCAGCTTGTACCTGCAGGATCTGAGCGCCGCCGCCTCAGAGTGCATCG
ACCCCT
##
1480
##
AGGAGGTGGTAGGCAGAGGTGGTGGGGCAGCTGGGCTGCGCTCCTCCTCCCGTTTTGCCTGTTGAGAGAC
CAGGAG
##
1470
##
CCAAGTCTACCTGGCTCCCCTTCCCTCCAGGGGAGCAGGGGAGCAAAGGGCCCCTCCTCTGAGCATCTAG
AGCAAC

```
##
1454
##
GGCCACTGTTTTGTTGGCGGGCAACCCTGCTTGCAGGATGGGCCGGTGAGGGGACCGCTCTGTGGAAGAT
GGGAGA
##
1447
##
TCAGAAACCGATTTTCCTATCTCTCTGCCTGGAGGGTGGTGGAGGGCTGGTTTGGGGAAGAGTGGGCTAG
TGCATT
##
1447
##
CCAGGGACACAGGGGTGCATAAGCCATGGCTTTTCCCCGGGGGAGGCTCATGCTCCTCGGGGAGAGACAG
ATCACT
##
1431
##
GGGGCTCGGGGCTGCCCTGCGGGGAGGACTCCGTCGAGGAGAGCAGAGAATCCGAGGACGGAGAGAAGGC
GCTGGA
##
1426
##
ACTCGGTGCAGCCGTATTTCTACTGCGACGAGGAGGAGAACTTCTACCAGCAGCAGCAGCAGAGCGAGCT
GCAGCC
##
1416
##
CAACTCCATGACCAGCTCGCAGACCTACATGAACGGCTCGCCCACCTACAGCATGTCCTACTCGCAGCAG
GGCACC
##
1384
##
CCCAGCATCCAGTGGGGGAGTGAAGGGCAATGAAGGGTACATCCTGGGGTCCAGGCCAACCTGGCTCTCT
GCTCGG
##
1383
##
GTCACCATCTCCAGCTGGTCGGCCGTGGAGAAGCTCCCGCCACCGCCGTCGTTGTCTCCCCGAAGGGAGA
AGGGTG
##
1320
##
GGGAACCTTTGCTCTTTTTCAGTGACCAAAATCATCTGTGCCCAGCAGTGCTCCGGGCGCTGCCGTGGCA
AGTCCC
##
1296
##
AGCGGCCCATGAATGCCTTCATGGTGTGGTCCCGCGGGCAGCGGCGCAAGATGGCCCAGGAGAACCCCAA
GATGCA
##
1291
##
GTTTAAATAGTTAAGAAAAGCTGCTCTGCTACGATGACACTGGGGCCAAGAGTGAGAGAAAGTGAGGGAG
GGAGCT
```

```
##
1269
##
TAGCCATGGCAAGGTCCCCATGACAAGTGCCTCCTCTCCCATCTTCCACAGAGCGGTCCCCTCACCGGCC
CATCCT
##
1240
##
CCTTATGTGACTTGTTTCCTTCCCCTCAGCCCTGCCCTCACTGCAGTGCTCTACGACCTGAGCCGTCAGA
TTCCAC
##
1210
```

Del mismo modo se pueden identificar lecturas duplicadas, potencialmente provenientes de PCR por amplificatión utilizando la función srduplicated y nuestro ShortReadQ object.

La función nos retorna un vector con aquellas secuencias duplicadas.A tener en cuenta que la primera vez qeu aparece la secuencia, es dicha secuencia en sí, no su duplicado. A partir de la primera, el resto son dichos duplicados. A continuación se muestra un resumen:

```
duplicates <- srduplicated(fq)
duplicates[1:10]

##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

table(duplicates)

## duplicates
##   FALSE    TRUE
##  350358 4106272
```

Chunk 7: Filtering and Trimming Data

El siguiente escenario es el pre procesado de lecturas, por ejemplo para el recorte de las colas de baja calidad, los adaptadores o los añadidos para la preparación de la muestra. En el marco de nuestro proyecto no es necesario dicha filtración por los motivos mencionados anteriormente, pero dado el caso en el que se observen baja calidad al final de las lecturas, puede ser posible la eliminación de las colas de baja calidad para su posterior alineamiento con el genoma. La función trimTails() , recorta las lectyuras desde 3', eliminando aquellas bases que están por debajo de un umbral deseado. Esta función acepta argumentos que especifican sobre el objeto ShortReadQ, el número m?nimo de bases sucesivas requeridas para estar por debajo del límite de calidad para el recorte y la puntuación de corte real.

```
TrimmedFastq <- trimTails(fq,20,"5")

TrimmedFastq
```

Adicionalmente, una vez recortadas, podemos exportar nuestras lecturas FastQ para un análisis adicional utilizando la función writeFastq ().

```
writeFastq(TrimmedFastq,"myTrimmed_Fastq.fastq")
```

Una vez aplicados los filtros, se pueden volver a procesar los pasos indicados en el chunk 3 sobre la calidad de las muestras, asi como obtener la distribución de calidad especifico en cada ciclo mediante qa.

Chunk 8: Alignment/Mapping

La secuenciación produce una colección de secuencias sin contexto genómico. No sabemos a qué parte del genoma corresponden las secuencias. El mapeo de las lecturas de un experimento a un genoma de referencia es un paso clave en el análisis de datos genómicos modernos. Con el mapeo, las lecturas se asignan a una ubicación específica en el genoma y se pueden obtener conocimientos como el nivel de expresión de los genes.

Las lecturas cortas no vienen con información de posición, por lo que no sabemos de qué parte del genoma provienen. Necesitamos usar la secuencia de la lectura en sí para encontrar la región correspondiente en la secuencia de referencia. Pero la secuencia de referencia puede ser bastante larga (~ 3 mil millones de bases para humanos), lo que hace que sea una tarea desalentadora encontrar una región que coincida. Ya que nuestras lecturas son cortas, puede haber varios lugares, igualmente probables en la secuencia de referencia, desde los cuales se podrían haber leído. Esto es especialmente cierto para las regiones repetitivas.

En principio, podríamos hacer un análisis de BLAST para averiguar dónde encajan mejor las piezas secuenciadas en el genoma conocido. Tendríamos que hacerlo para cada uno de los millones de lecturas en nuestros datos de secuenciación. Sin embargo, alinear millones de secuencias cortas de esta manera puede llevar un par de semanas. Y realmente no nos importa la correspondencia exacta de base a base (alineación). Lo que realmente nos interesa es “de dónde provienen estas lecturas”. Este enfoque se llama mapeo. En nuestro caso específico, sabemos que están mapeadas frente al cromosoma 10 (chr10), por lo que tanto a nivel de visualización como de anotaciones, ésta va a ser nuestra referencia.

Una vez revisados los archivos FASTQ y eliminados todas las secuencias de adaptadores que pudieran estar presentes, esta listo para mapearlos a un genoma de referencia. Si bien las herramientas como BLAST y BLAT son métodos poderosos, no están especializadas para la gran cantidad de datos generados por los secuenciadores de la próxima generación. Se recomienda encarecidamente que utilice un programa de alineación de lectura específico de la próxima generación. Es por eso que se va a emplear el paquete Bowtie2.

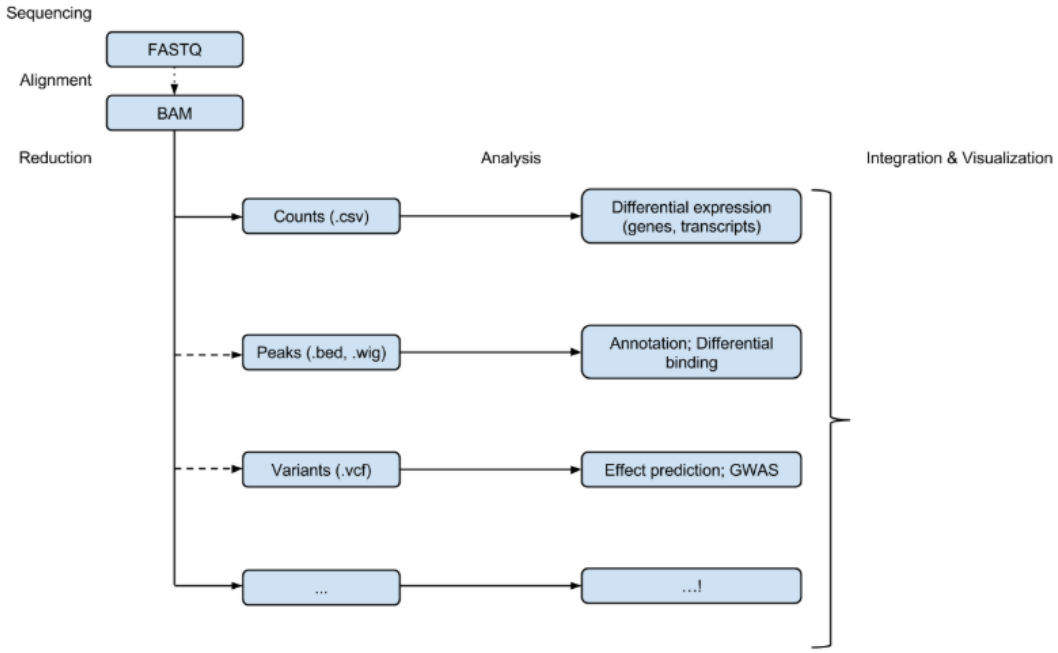
El workflow a seguir, partiendo de la base que tenemos los archivos fastq, es el siguiente:

1. Preparar los datos
2. Mapa leído en un genoma de referencia
3. Inspección de un archivo BAM
4. Visualización utilizando un navegador genoma (IGV) o mediante un navegador genoma (IGV)

```
file.path("C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline")
```

```
## [1]
"C:/Users/davmasip/Desktop/Dako/02.Master/TFM/R/Proyecto/Pipeline"
```

```
knitr::include_graphics("Workflow.png")
```



Preparación del Genóma de referencia

Los genomas de referencia se pueden descargar desde los recursos del genoma UCSC, Ensembl o NCBI. En nuestro estudio utilizaremos la referencia humana GRCh37.hg19 pues los fastQ files están enfrentados contra ese Genoma.

Las alineaciones genómicas y las posteriores anotaciones pueden consumir mucho tiempo y no ser realistas en el corto tiempo que tenemos. Por lo tanto, una vez obtenido el archivo en formato BAM y preprocesamos un solo el cromosoma (chr10) del conjunto de datos anterior para ahorrar tiempo. El archivo BAM se puede obtener fácilmente sobre el programa Galaxy,a si como su posterior representación, anotaciones y variaciones. Para el alineamiento, se utilizará el paquete Bowtie2.

Anexo 2: Código shiny app.

Ui.R

```
library(shiny)
library(shinydashboard)
library(rsconnect)
library(ShortRead)
library(FastqCleaner)
library(systemPipeR)
library(rlang)
library(Biostrings)
library(seqinr)
library(ade4)
library(FastqCleaner)
library(DT)
```

```
library(httpuv)
library(BiocGenerics)
library(logging)
library(shinyjs)
library(shinyBS)
library(ggplot2)
library(GenomicAlignments)
options(repos = BiocInstaller::biocinstallRepos())
getOption("repos")


header <- dashboardHeader(title = "FastQ Pipeliner App")

#####

sidebar <- dashboardSidebar(
  tags$style(".left-side, .main-sidebar {padding-top: 100px}"),
  tags$style(".left-side, .main-sidebar {padding-bottom: 100px}"),
  width = 275,

  sidebarMenu(

    menuItem("Sidebar Menu", tabName = "sidebar", icon = icon("atom")),
    hr(),

    #Loading tab

    menuItem("Loading tab",icon = icon("file-upload"),startExpanded = FALSE,

           br(),

           menuSubItem(text = "Upload your file/s",tabName = "upload", icon =
icon("upload"), href = NULL, newtab = TRUE),
             fileInput(inputId = "file",
                       label = h6(""),
                       multiple = F,
                       accept = c("fastq/fasta",
                                  ".fastq",
                                  ".fasta"
                                  ),
                       width = 450,
                       buttonLabel = "Browse...",
                       placeholder = "No file selected"
             ),
           h6(helpText("Default max. file size 1GB", align = 'right')),
           menuSubItem(text = "Data summary", tabName = "summary", icon =
icon("info"), href = NULL, newtab = TRUE)
           ),

    br(),


    #Alignment

    menuItem("Alignment tab", tabName = "alignment", icon = icon("align-justify"),
startExpanded = F,
```

```
        br(),

        h5(helpText("Choose the Human Genome type below")),
        radioButtons(inputId = 'genome', label = "", choices = c(GRCh37 = 'GRCh37',
GRCh38 = 'GRCh38'), selected = 'GRCh38'),

        h5(helpText("Align Data to Genome")),
        fluidRow(
          column(12, offset = 4,
                 actionButton("align","Apply!"))
        ),
        uiOutput("apply1"),

        br()
    ),
    br(),

    #Charts

    menuItem("Chart", tabName = "chart", icon = icon("chart-bar"), startExpanded =
FALSE,
        selectInput("plotType",
        label = p(class = "controlcheck-p3", "Diagnostic plots"),
        choices = c(
          "Select a plot..." = "",
          "Per cycle quality" = "a",
          "Per cycle mean base quality" = "b",
          "Mean quality distribution" = "c",
          "% reads with Phred scores > threshold" = "d",
          "Per cycle base proportion (barplot)" = "e",
          "Per cycle base proportion (lineplot)" = "f",
          "CG content over all reads" = "g",
          "Read length" = "h",
          "Read occurrence" = "i",
          "Relative k-mer diversity" = "j"


        ),
        selected = "none"
    ),
    br()
                    )

  )
)

#########################################################################

body <- dashboardBody(
  useShinyjs(),
  tags$style(type="text/css",
        ".shiny-output-error { visibility: hidden; }",
        ".shiny-output-error:before { visibility: hidden; }"
  ),
  tabItems(
```

```
   tabItem(tabName = "sidebar",

           p("Welcome to the Pipeliner App",
             br(),
             "To begin, please upload a fastq file.", align = 'center',
             br(),
             br(),
             br(),
             br(),
             br(),
             br(),
             br(),
             br(),
             h5("Powered by", tags$img(src = 'logo1.png', height=50, width=150), align =
'center'))

           ),

   tabItem(tabName = "upload",
           uiOutput("upload")
           ),

   tabItem(tabName = "summary",
           fluidRow(
             tabsetPanel(
               tabPanel(("Summary"),

                        column(12, offset = 1, style = "float:left; margin-left:2em;margin-top:
2em",
                               p(class="controlcheck-p3", "File selected:"),
                               verbatimTextOutput("name")
                        ),
                        column(12, offset = 1,style = "float:left; margin-left:2em; margin-top:
2em",
                               p(class="controlcheck-p3", "Size:"),
                               verbatimTextOutput("size")
                        ),
                        column(12, offset = 1,style = "float:left; margin-left:2em; margin-top:
2em",
                               p(class="controlcheck-p3", "Datapath:"),
                               verbatimTextOutput("datapath")
                        ),
                        uiOutput("summary")),

               tabPanel("Raw Data",
                        dataTableOutput('raw_data')),
               tabPanel("Test",
                        dataTableOutput('test'))
             )
           ),


   #Filters

   tabItem(tabName = "a",
```

```
        fluidRow(
          tabsetPanel(
            tabPanel("Data"),

                    h4("Explorar secuencias"),
                    h5("Seleccione una secuencia para filtrar"),
                    br(),
                    br()

            )
        )
        ),


    #Charts

    tabItem(tabName = "chart",

        fluidRow(

          htmlOutput("plot1"),
          htmlOutput("plot2"),
          uiOutput("plotPanel"),
          bsModal("modalTable",
                  title = "Most represented sequences",
                  trigger = "tabBut",
                  size = "large",
                  DT::dataTableOutput('freqTable1'),
                  DT::dataTableOutput('freqTable2')
          ),
          uiOutput("tabBut")

        )
    )

  )
)

ui <- dashboardPage(header, sidebar, body)
```

---

server.R

```
shinyServer(function(input, output) {

  options(shiny.maxRequestSize = 1024*1024^2)
  #LOADING TAB
  ##########################################################################
  ##########################################################################

  #Data reactive file

  data <- reactive({

    file = input$file
    if (is.null(file)) {return(
```

```
  )}
  else{
    fastQ <- readFastq(dirPath = file$datapath)

  }

})

##loading tab
output$upload<- renderUI({
  file = input$file
  if(is.null(input$file)){
      p("Welcome to the Pipeliner App",
        br(),
        "To begin, please upload a fastq file.", align = 'center',
        br(),
        br(),
        br(),
        br(),
        br(),
        br(),
        br(),
        br(),
        br(),
        h5("Powered by", tags$img(src = 'logo1.png', height=50, width=150), align =
'center'))}
})

#Output raw_data

output$raw_data <- DT::renderDataTable({

    raw<-data.frame(
      ID = (idOfReads<-id(data())),
      Sequences = (sequences <- sread(data())),
      Length = (lengths_size <- width(data())),
      Frequence = (alpFreq <- alphabetFrequency(sequences,baseOnly = TRUE)))

})

output$test <- renderPrint({

})



output$summary <- renderPrint({
  if (is.null(data()))
  {return(

    )}
    else{

      output$name <- renderPrint(input$file$name)
      br()
      output$size <- renderPrint(input$file$size)
      br()
```

```
      output$datapath <- renderPrint(input$file$datapath)


    }
  })

  #fILTER TAB
  ################################################################################
  ################################################################################

  #ALIGNMENT TAB
  ################################################################################
  ################################################################################



  #CHART TAB
  ################################################################################
  ################################################################################

})
```

Anexo 3: Galaxy files

https://agilent-my.sharepoint.com/:f:/p/david_masip/EuIqDCX5W7dCi-Atk2kYpWUB0terPtoaz6JT0QNe-_478w?e=n8Q9ZD

Anexo 4: Fastq files

https://agilent-my.sharepoint.com/:f:/p/david_masip/EvsS-ejPoXhCiQhWGWRshwIB6qZfqDrQ_wR-TWCzMpqI9Q?e=ftdVSw