UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática

Memoria del Trabajo de Fin de Título de Carrera de Adaptación
al Grado de Ingeniería en Informática
de la Universidad de Las Palmas de Gran Canaria

TÍTULO

Identificación de regiones del ADN con baja frecuencia de lectura

AUTOR

Deyán Fabricio Guacarán Sabogal

TUTORES

Antonio Tugores Cester. *Jefe de Servicio Unidad de Investigación del Complejo Hospitalario Universitario Insular – Materno Infantil.*

Francisca Quintana Domínguez. *Profesora Titular de Universidad del Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria.*

FECHA

Julio de 2014

Contenido

1. Agradecimientos.

Quiero dar las gracias a todas las personas que han participado en el desarrollo de este Trabajo Fin de Grado:

A mi tutor Antonio Tugores, por enseñarme y compartir sus conocimientos, ya que sin él no hubiera sido posible redactar esta memoria. Por otro lado a mi tutora, Francisca Quintana, por darme la oportunidad de participar en este proyecto, darme todo su apoyo y ayudarme en la divulgación y rectificación de dicha memoria.

A mi compañero y amigo Pascual, por haberme ayudado durante el transcurso del desarrollo del proyecto y compartir sus conocimientos conmigo. Sin él, no hubiera sido posible la finalización del mismo. Este Trabajo Fin de Grado también le pertenece a él.

A todos las personas pertenecientes a UICHUIMI (Juan Carlos, Paloma, Cristina, Teresa…), por los momentos buenos que hemos pasado y que me han ayudo a realizar este proyecto de forma más amena y agradable.

A todos mis compañeros de clase y amigos (David, Dalia, Xerach, Francisco, Aythami, Cynthia, Román, Aitor Cardona, Aitor Hernández, Cristián, Natalia, Pablo, Reynier…), ya que con ellos he compartido y vivido momentos muy bonitos durante toda mi carrera y he aprendido mucho de cada uno de ellos.

A mis amigos que he tenido el placer de conocerlos este año (Helena, Alba, Ylenia, Patricsua, Adrián, Allende, Borja, Carlos, Patricia, Elena, Jezabel, Pablo, Vicente,…) y que me han dado momentos muy divertidos y alegres durante todos estos meses. Todos ellos también forman parte de este proyecto.

A todos mis profesores de la Universidad, a ellos les debo mis conocimientos y cualidades adquiridas en todos estos años de mi carrera.

A toda mi familia y a Jesús, por haberme dado su apoyo en todos los momentos de mi vida universitaria. Sin ellos, no habría sido posible llegar hasta donde he llegado y ser la persona que soy en estos momentos de mi vida.

A mi segunda familia, porque los considero como tal, por todo el apoyo incondicional que me han dado y por estar ahí para mostrarme todo

su apoyo en todos los momentos de mi vida en estos años. En especial también a mis abuelos por darme todos sus buenos consejos y de ellos he aprendido los valores de la vida y que hay que seguir adelante superando todas las adversidades que nos ponga la vida.

A Rita, una persona muy especial en mi vida, por haberme dado todo su apoyo para seguir adelante y disfrutar y sufrir cada uno de los momentos durante mi paso en la vida universitaria. Gracias por estar ahí siempre que lo he necesitado, sin ti esto tampoco sería posible. Tú también formas parte de cada una de estas líneas de este proyecto.

Y en especial, a mi madre, gracias por todos los esfuerzos que has puesto en mí y por darme la oportunidad de estudiar lo que he querido. Sin ti, no llegaría a donde he llegado ni tendría la oportunidad de escribirte estas líneas para ti. Por estar ahí en todos mis momentos tanto buenos como difíciles que he pasado en mi vida, y que por ti he aprendido que hay que luchar y seguir para adelante en todas las barreras que nos pongan. Gracias por todos tus consejos.

Muchas Gracias a todos.

Deyán Fabricio Guacarán Sabogal

2. Resumen del Trabajo Fin de Grado.

El ADN es un polímero [i] que contiene la mayor parte de la información necesaria para el desarrollo y funcionamiento de todos los organismos vivos conocidos. La información está fraccionada en diferentes segmentos, los genes, que contienen variables que son individuales y que determinan las características de cada persona. Entre ellas, hay dos que son de especial importancia para la atención sanitaria: la susceptibilidad genética de padecer una enfermedad y la capacidad de responder de forma diferencial a un medicamento, denominado farmacogenética [ii]. Poder identificar dichas variantes puede ayudar a comprender la enfermedad e individualizar el tratamiento del paciente respectivamente.

Para conocer estas variantes debemos conocer la secuencia de ADN de los genes implicados en las patologías o en las características farmacogenéticas para un individuo determinado, un proceso denominado secuenciación. La secuenciación completa del genoma, por su complejidad, está todavía lejos de encontrar un hueco entre las aplicaciones médicas de uso rutinario. Sin embargo, existen técnicas para seleccionar y secuenciar el exoma, que es la parte del genoma que contienen los exones, fracciones de los genes que contienen la información necesaria para la fabricación de las proteínas. El exoma supone aproximadamente un 2% del total del genoma, y su secuenciación se realiza en la Unidad de Investigación del Complejo Hospitalario Materno Infantil (UICHUIMI) para identificar la causa de enfermedades raras y de origen genético.

La secuenciación de exoma cubre la mayor parte de los exones del genoma, pero no detecta algunas regiones, lo que imposibilita la detección de variantes en ellas. Este hecho crea una incertidumbre diagnóstica, lo que limita el poder de esta herramienta para la detección de mutaciones patogénicas [iii], ya que genes clínicamente relevantes pueden haberse quedado sin analizar por un defecto experimental. Por ello, necesitamos saber con precisión qué regiones exómicas han quedado fuera del análisis. Así, el objetivo principal del Trabajo Fin de Grado es la creación de una herramienta informática que permita al personal clínico, sin conocimientos informáticos la detección de regiones del exoma con poca cobertura de secuenciación, es decir, regiones del ADN con una frecuencia de lectura baja comparándolo con respecto al genoma de referencia (ADN estándar).

3. Introducción.

El ácido desoxirribonucleico (ADN) es un ácido nucleico que contiene la información genética de los organismos vivos y es responsable de su almacenamiento y transmisión hereditaria. La totalidad de la información genética que posee un organismo o una especie en particular se denomina *genoma*, es decir, un conjunto de genes donde están almacenadas las claves para la diferenciación de las células que forman los diferentes tejidos y órganos de un individuo.

El genoma codifica una serie de características o rasgos observables de un organismo, como su morfología, desarrollo, propiedades bioquímicas, fisiología y comportamiento. A todas estas características la denominamos *fenotipo*, y a toda la información genética, el *genotipo*.

El ADN está estructurado, como observamos en la Figura 3.1, por dos largas cadenas complementarias de nucleótidos unidas entre sí formando una doble hélice. Las dos cadenas de nucleótidos que constituyen una molécula de ADN, se mantienen unidas entre sí porque se forman enlaces entre las bases nitrogenadas [iv] de ambas cadenas que quedan enfrentadas.

Esta configuración le aporta estabilidad a la molécula de ADN. Las cuatro bases nitrogenadas que se encuentran en el ADN son la Adenina (A), Timina (T), Citosina (C) y Guanina (G). La unión de las bases se realiza mediante puentes de hidrógeno, y este apareamiento está condicionado químicamente de forma que la Adenina se complementa con la Timina y la Guanina con la Citosina.

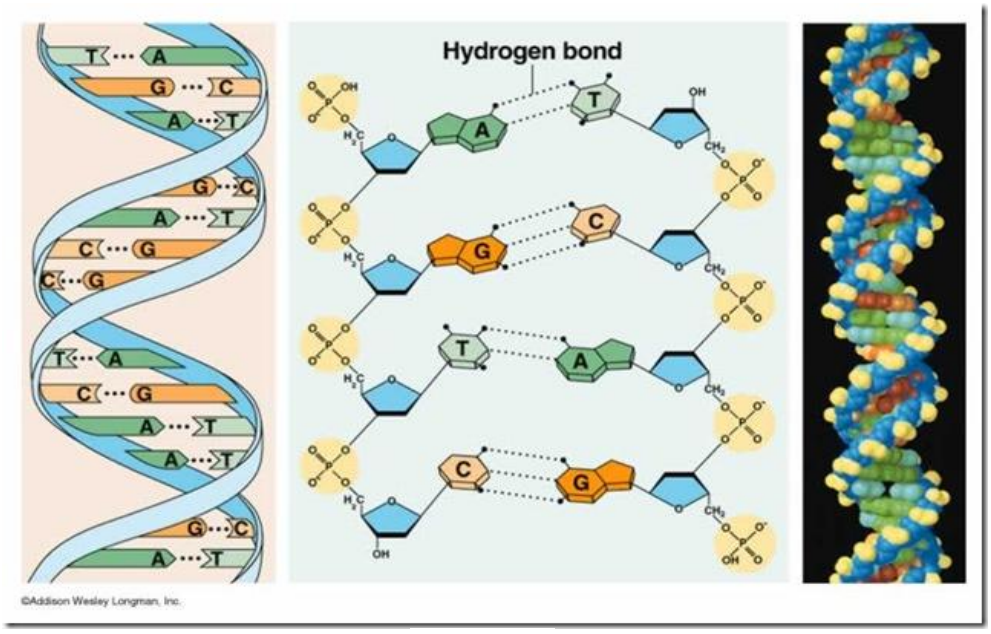


Figura 3.1

En el ser humano, las células somáticas [v] poseen 46 cadenas de ADN, llamados *cromosomas*[vi], distribuidos en 23 parejas. Cada par está formado por un cromosoma de origen materno y otro paterno. Veintidós de estas parejas tienen la misma estructura y el tamaño, es decir, son homólogos. Pero el sexo de la persona está determinado por el último par de cromosomas, los cromosomas sexuales. La mujer contiene dos cromosomas X, mientras que el hombre contiene un cromosoma X y otro Y (es decir, cromosomas heterólogos). Las células germinales contienen la mitad de la información, que resulta de la recombinación de los cromosomas paterno y materno para originar cromosomas únicos responsable de la herencia.

El genoma humano contiene aproximadamente entre 20.000 y 25.000 genes. Se consideran genes las regiones del genoma que codifican para proteínas, esto es, las regiones que se transcriben para generar ARN mensajeros que luego se traducen en proteínas en los ribosomas. De media, los genes tienen un tamaño de 3.000 nucleótidos o bases. Teniendo en cuenta que el total del genoma asciende a 3.164,7 millones de bases, la suma de todos los genes representa un 2% del genoma humano. Por tanto, el 98% del genoma humano es ADN-no codificante, o lo que es lo mismo, no contiene información relevante para la síntesis de proteínas.[2]

La estructura de los genes contiene, además, elementos que participan en la regulación de su expresión como los *promotores* [vii] y *potenciadores*[viii] de la transcripción.

La estructura intrón-exón de los genes fue descubierta en 1977 (podemos observar dicha estructura en la Figura 3.2). [2] Estructuras que diferencian a Eucariotas de Procariotas[ix].

Un *intrón* es una región del ADN que debe ser eliminada del transcrito de ARN antes de que salga del núcleo y sea traducido. Los *exones* representan las secciones en la versión final del ARNm que se unen entre sí durante el proceso de *splicing* (también llamado corte y empalme) y, por tanto, se traducirán finalmente en proteínas.

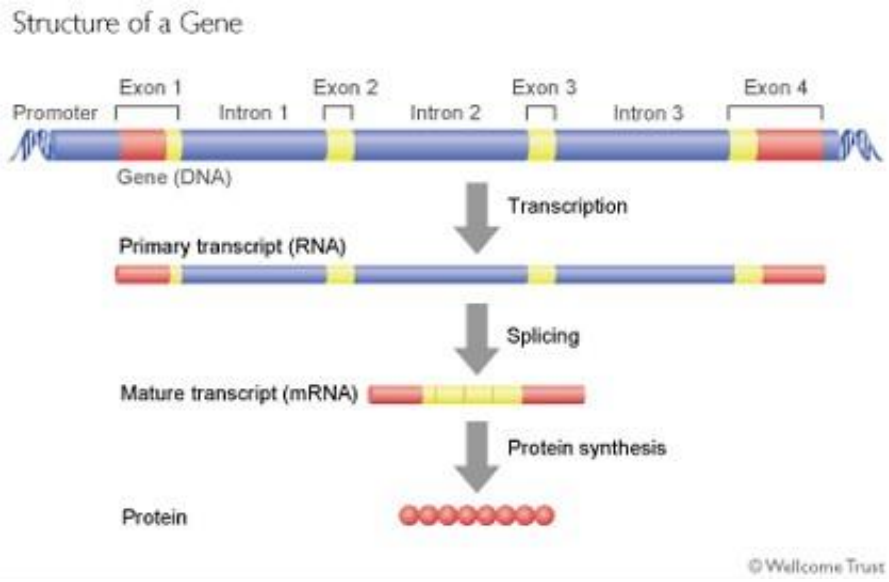


Figura 3.2

La expresión génica, encargada de la síntesis de proteínas a partir de los genes, comprende por dos etapas: *fase de transcripción* y *la fase de traducción*. En la fase de transcripción consiste en que a partir de un fragmento del ADN es transcrito por ARN polimerasa (ARNP) en un ARN nuclear que, tras el procedimiento (splicing), se convierte en mensajero (ARNm). En la fase de traducción se genera la proteína a partir del ARNm y está comprendida por tres subfases: *iniciación de la síntesis proteica*, *elongación de la cadena polipeptídica* y *finalización de la síntesis de las proteínas*. Podemos observar estas dos etapas en las siguientes imágenes Figura 3.3 y Figura 3.4 respectivamente.**¡Error! No se encuentra el origen de la referencia.**

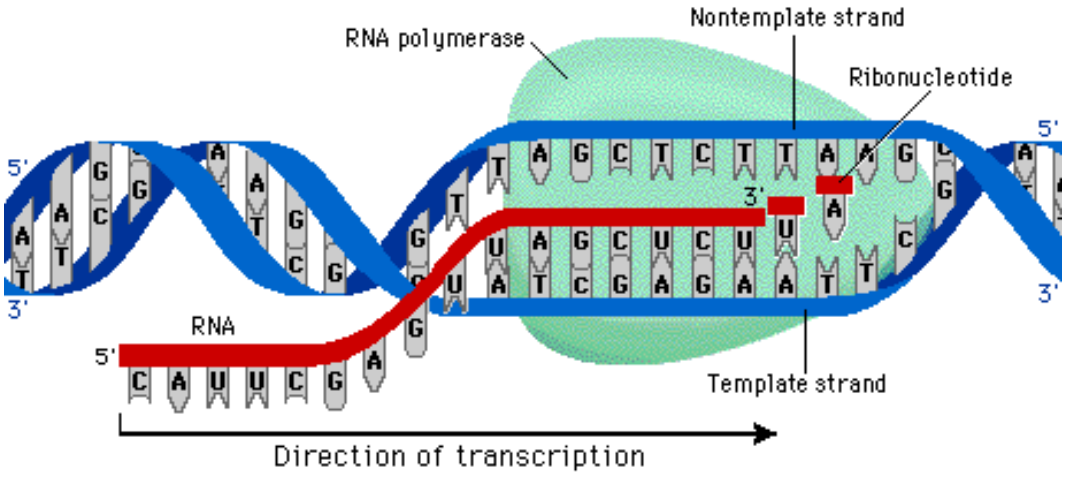


Figura 3.3

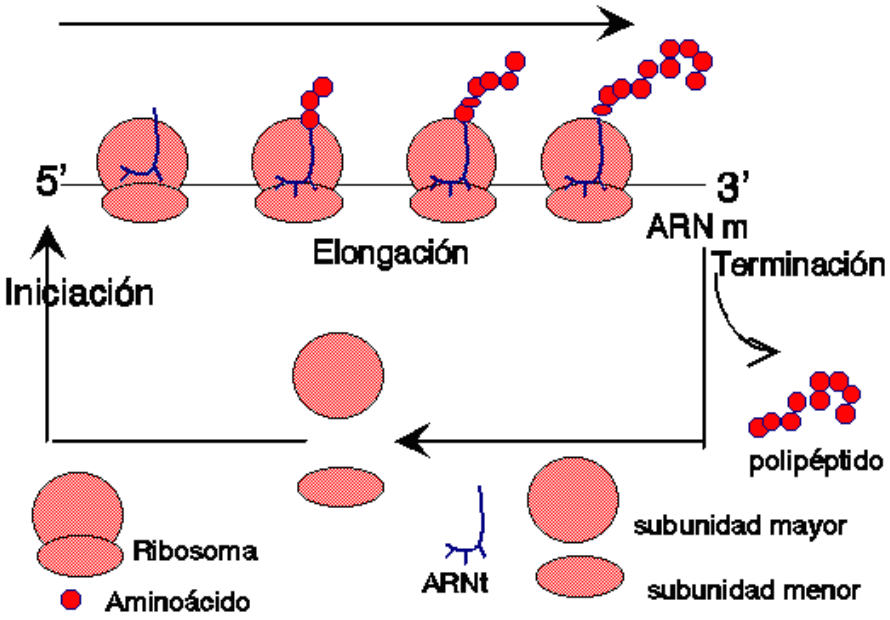


Figura 3.4

3.1 Aplicaciones Médicas.

La presencia de variantes genéticas puede provocar cambios en la estructura final de las proteínas, lo que puede ser origen de enfermedad. El diagnóstico genético de enfermedades se ha convertido en una técnica que se emplea cada vez más en la Medicina. Entre las diferentes técnicas de análisis de ADN, la secuenciación se considera como la técnica de referencia, dada su alta sensibilidad y especificidad. La importancia creciente que tiene la secuenciación de ADN en el diagnóstico ha provocado un gran desarrollo en todos los aspectos relacionados con las tecnologías de secuenciación de ADN, así como en las herramientas informáticas para el procesado y análisis de las secuencias de ADN.

Entre las nuevas técnicas de secuenciación de ADN que se han desarrollado en los últimos años destacan las técnicas de nueva generación, que se suelen conocer como *Next Generation Sequencing*, o simplemente NGS. **¡Error! No se encuentra el origen de la referencia.**

Pero cabe preguntarse ¿qué método de secuenciación de ADN se considera hoy en día como el método de referencia para el diagnóstico

genético de enfermedades? La respuesta a esta pregunta es muy directa. El método de referencia para la secuenciación de ADN para su uso en el diagnóstico genético de enfermedades sigue siendo el método de Sanger, en combinación con la electroforesis capilar (EC). El método de Sanger, que es considerado como la "técnica clásica" de secuenciación es el único que, por el momento, aparece en las directrices y estándares de calidad (Best Practice Guidelines) de los organismos internacionales que se encargan de supervisar el buen hacer de los laboratorios de diagnóstico genético.

En principio, el concepto del NGS es muy similar al método de Sanger, ya que las bases son secuencialmente identificadas a través de señales emitidas.

La diferencia es que el método de Sanger se limita a secuenciar un segmento de un específico tamaño, mientras que el NGS extiende este proceso a través de millones de reacciones de forma paralela y masiva.

Para ello, cada muestra de ADN es fragmentada dentro de una librería de pequeños segmentos que pueden ser secuenciados de manera uniforme y precisa en millones de reacciones paralelas. Las recién identificadas cadenas de bases, llamadas *lecturas*, son entonces reensambladas usando un genoma de referencia conocido. En el caso de que haya una ausencia del genoma de referencia para la especie que se está secuenciando, se utiliza un ensamblado de *novo*.

En resumen, el método NGS tiene una alta profundidad de cobertura y es más rentable y barato a la hora de secuenciar largas regiones. Sin embargo no es tan efectivo como el método Sanger en cuanto a las regiones repetitivas de secuenciación.

También existen otras aplicaciones médicas de secuenciación que se están utilizando en el mundo de la Medicina, como es la farmacogenética y el análisis de enfermedades no reveladas. La primera consiste en el estudio en las diferentes respuestas que puede dar un determinado paciente ante los medicamentos. Los medicamentos constituyen hoy en día una de las causas de reacciones adversas, que resultan en una importante morbilidad y mortalidad en pacientes así como en un aumento de los costos en los tratamientos e infraestructuras. Esta disciplina ayuda a personalizar los tratamientos médicos en los pacientes, mejorando el uso de los medicamentos.

La segunda consiste en que a través de la estrategia de la secuenciación del exoma se puede averiguar la aparición de enfermedades graves en los pacientes. Esto permite la aplicación de procedimientos preventivos, que pueden hacer mejorar el estado de salud de los pacientes y ayudan beneficiosamente en cuanto a los costos económicos que por ello acarrea.

Finalmente, el proceso de secuenciación del exoma permite hallar la originalidad genética de una enfermedad, permitiendo el tratamiento de la enfermedad de forma más eficaz. Dicho proceso es recomendable utilizarlo cuando es difícil llegar al diagnóstico por criterios clínicos o de laboratorios convencionales (ej.: síntomas compartidos entre varias enfermedades), o cuando el número de genes candidatos es demasiado elevado.

4. Estado Actual y Objetivos.

La Unidad de Investigación del Complejo Hospitalario Universitario Insular – Materno Infantil (UICHUIMI) durante muchos años lleva estudiando descubrir el origen de enfermedades genéticas en los habitantes locales con la finalidad de determinar y proponer un tratamiento apropiado para estas nuevas enfermedades. Esta Unidad de Investigación ha avanzado en muchas de sus investigaciones logrando, por ejemplo, la identificación de una mutación endémica[x] en la población perjudicada por la enfermedad de Wilson[xi].

Se evalúa la secuenciación del exoma para identificar la causa de enfermedades poco comunes y de origen genético. Para ello existe una herramienta de software llamada "DNAnalytics" (desarrollada en el 2013 en la Unidad de Investigación) que realiza dicho proceso, y en el que se trabaja con ficheros de datos grandes, que son poco intuitivos.

Uno de los problemas que se plantea es que la cobertura (número de veces que se ha secuenciado de forma fiable una región determinada) es variable. Esto es, hay regiones del ADN donde no se ha secuenciado el exoma, y por tanto hay imposibilidad de identificar variantes en dichas regiones. Esto crea una incertidumbre diagnóstica, lo que limita el poder de esta herramienta para la detección de mutaciones patogénicas, ya que genes clínicamente relevantes pueden haberse quedado sin analizar por un defecto experimental. Por ello, necesitamos saber que regiones exómicas han quedado fuera del análisis.

En UICHUIMI a partir del empleo de la secuenciación del ADN por NGS (Next-generation sequencing) se trabaja con ficheros de datos grandes. Esto implica la manipulación, búsqueda y extracción de información de los datos de la secuencia del ADN (bioinformática) sobre grandes tamaños de datos.

En primer lugar, debemos señalar que este trabajo está basado en la continuación y mejora sobre la aplicación global ya creada, y que hemos nombrado con anterioridad. La herramienta general se encarga de procesar y encapsular el flujo de análisis del ADN, llegando a realizar de forma informatizada desde el alineamiento del mismo hasta la anotación y filtración de las variantes descubiertas. En nuestro caso, como trataremos

con ficheros de alineamiento de las secuencias del ADN, nos enfocaremos en esa fase.

Por otro lado, destacamos que inicialmente el objetivo de este Trabajo Fin de Grado era la creación de un módulo software que permita al personal de la Unidad de Investigación sin conocimientos informáticos, la identificación de regiones del ADN con una **frecuencia de lectura baja** (por debajo de un **umbral** definido por el usuario) comparándolo con respecto al genoma de referencia (ADN estándar). Dicho módulo una vez finalizado se integraría en la aplicación global ya existente (DNAnalytics).

Pero en el transcurso del desarrollo del proyecto, se ha visto necesario la transformación del desarrollo del módulo a la creación de una interfaz de usuario (GUI) que realiza dicho proceso. Esto es, porque el personal clínico se veía con la necesidad de tener un entorno visual sencillo que permita la interactuación con el algoritmo desarrollado.

Además, hemos añadido un objetivo más, que consiste en la obtención de números estadísticos sobre los resultados que se generan una vez finalizado el proceso de búsqueda de regiones con baja lectura del ADN. Así, podíamos saber de manera más precisa cuántos genes/exones son sospechosos de estar mal secuenciados.

Por tanto se pueden definir los siguientes objetivos:

- A partir de un fichero de datos (en nuestro caso es un fichero de alineamiento del ADN con formato .BAM) extraído de la secuenciación del ADN, calcular el número de veces que se ha leído un posición en el secuenciamiento del ADN.

- Filtrar y extraer las regiones que tengan un número de veces de lectura determinado por un específico umbral. Cabe destacar que nos interesa que dicho umbral sea **bajo** (o también no haberse leído nunca) para la localización de regiones del ADN sin secuenciar.

- Comparar estos datos con los exones ya conocidos de los humanos (Homo Sapiens), para la localización de regiones que no han sido secuenciadas.

- Calcular resultados estadísticos de los diferentes ficheros extraídos.

- Crear una Interfaz Gráfica (GUI) para los usuarios de la Unidad de Investigación.

5. Flujo de Análisis y búsqueda de variantes en el ADN.

Para la identificación de variantes genéticas que causan posibles enfermedades en los pacientes, la UICHUIMI realiza un proceso de secuenciación del exoma a través del NGS, ya que este método permite cubrir todo el exoma con una serie de lecturas bastantes fiables con respecto a una alta proporción del exoma. El ser humano posee muchas variantes en las cuales algunas de ellas pueden ser sospechosas de producir la enfermedad de análisis. Investigar este proceso es bastante difícil, ya que en la mayoría de los casos incluso hay que realizarlo de forma manual.

Por ello la UICHUIMI necesita de conocimientos informáticos y que ayude a orientarse hacia un proceso tanto estándar como económico en la práctica sanitaria.

En las siguientes fases describiremos el proceso que se realiza en el análisis del ADN, desde el estudio de una posible hipótesis o suposición hasta que se adquieren variantes que pueden originar enfermedades.

5.1 Preparación de hipótesis o suposición.

Primeramente, se realiza un análisis de los familiares del paciente o del paciente sospechoso de padecer una posible enfermedad genética. Para ello se confecciona un árbol genealógico como se puede observar en la Figura 5.1.

Dicha representación gráfica facilita la identificación de síndromes genéticos y el establecimiento de diagnósticos presintomáticos[xii]. A su vez permite un mejor cálculo del riesgo y los patrones de herencia de una enfermedad, es decir, permite conocer la probabilidad de tener una enfermedad o de heredarla.

La construcción de un árbol genealógico constituye el análisis genético más fácil y barato. Solamente con esta herramienta pueden desecharse algunas de las hipótesis posibles respecto a la enfermedad del paciente en estudio, evitando así un gasto considerable de dinero en la realización de pruebas diagnósticas más caras.

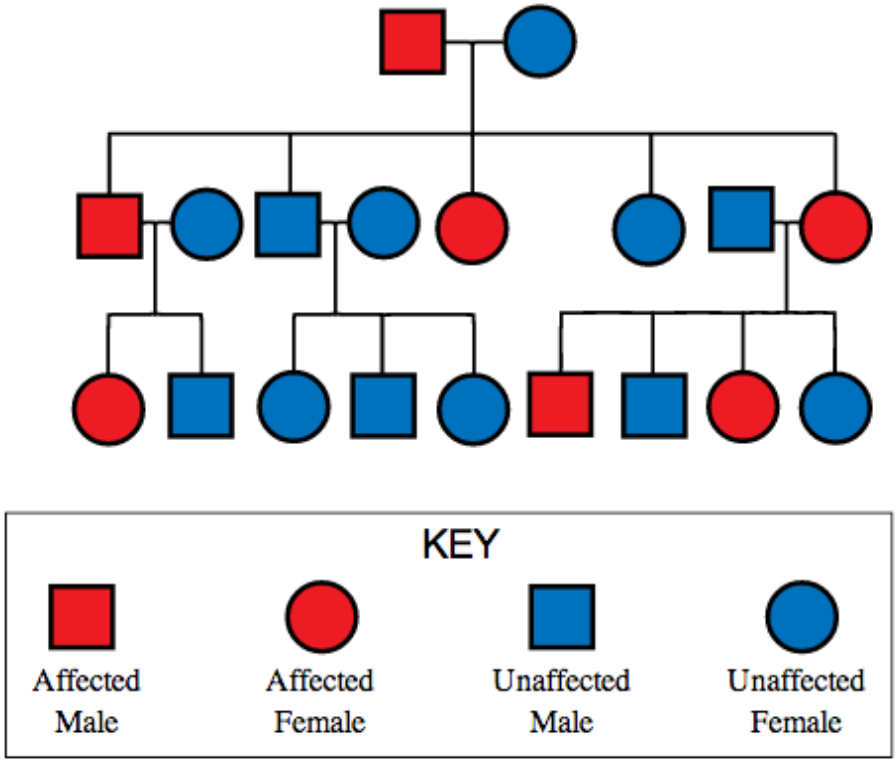


Figura 5.1

Una vez establecidas una o varias hipótesis, podremos decidir qué segmentos del ADN se van a secuenciar y realizar diferentes comparaciones de los resultados obtenidos. En la Unidad de Investigación se trabaja con tres tipos de hipótesis: homocigotos, dominantes recesivos y uno o más genes candidatos (variantes).

5.2 Obtención del ADN.

En esta fase se les extrae a los pacientes que queremos investigar una pequeña muestra de sangre con una cantidad aproximada de 10 mililitros, de la que se extrae el material genético de las células blancas según el método de precipitación fraccionada de Miller.[3]

5.3 Secuenciación del ADN.

Una vez obtenidas las muestras de ADN, son enviadas al Instituto de Genómica de Pekín (Beijing Genomics Institute – BGI) para que se encarguen de realizar la secuenciación por el método NGS.[4]

En este instituto se realiza el proceso de fragmentación del ADN en diferentes fracciones de aproximadamente 150 pares de bases. Con

diferentes métodos físico-químicos se ejecuta una separación de las fracciones que no forman parte del exoma y contiene una calidad menor.

Mediante el sistema "Illumina HiSeq 2000" las fracciones restantes correspondientes a una calidad positiva se secuencian en una dirección y su inversa, obteniéndose información de las dos cadenas del ADN.[5]

Este sistema de secuenciación no sólo ofrece unos resultados sin precedentes y de bajo coste, sino también una experiencia de usuario avanzada. Con dicha tecnología supera las deficiencias de las técnicas de secuenciación tradicionales como de alto coste en mano de obra y materiales. Los métodos alternativos de preparación de muestras permiten una amplia gama de aplicaciones incluyendo la expresión génica, pequeño descubrimiento de ARN o las interacciones proteína-ácido nucleico.

Los resultados obtenidos se guardan en diferentes archivos informatizados en el que incluyen una sucesión de secuencias con sus correspondientes valores de calidad. Esta información está almacenada en dos archivos en formato ".FASTQ", en el cual cada uno de ellos contiene las secuencias de una de las dos direcciones del ADN(5' y 3'). Para saber como está estructurado estos ficheros basta con irnos al final de la memoria al anexo "Formato de los ficheros".

5.4 Alineamientos de las Secuencias.

Los archivos recibidos de la secuenciación normalmente contienen 30 millones de secuencias con longitudes de aproximadamente de 90 pares de bases, por lo que almacena unos 3 mil millones en su totalidad.

El problema que existe es que la información que aporta estos ficheros no determina en qué posiciones se encuentran las secuencias, por lo que se lleva a cabo el proceso de alineamiento para ordenar cada secuencia en su posición exacta o aproximada. Para llevar a cabo este proceso se utiliza un genoma de referencia, en el cual se hace las comparaciones con cada una de las secuencias recibidas.

Los objetivos al hacer la comparación de dos o más secuencias son:

- Determinar (y cuantificar) el grado de similitud que hay entre ellas.
- Determinar si existe algún tipo de relación entre ellas o si el parecido es simplemente fruto de la casualidad.

- Detectar la presencia de motivos estructurales y/o funcionales conservados.
- Construir árboles filogenéticos [xiii] que reflejen sus relaciones evolutivas.

Realizar el alineamiento de las secuencias requiere un gasto muy alto tanto en computación como en memoria, por lo que utilizar métodos habituales se hace inviable. Para ello se emplea diferentes algoritmos que contribuyan a una mejor eficiencia del proceso en vez de una mayor eficacia. Existen dos tipos de algoritmos que pueden solucionar esta cuestión: *algoritmos en función de una tabla hash* y *algoritmos basados en árboles sufijo/prefijo*.

En función del número de secuencias que se comparan podemos distinguir:

- **Alineamiento de dos secuencias**: se comparan dos secuencias utilizando diversos métodos como, por ejemplo, la matriz de puntos (dot-plot), algoritmos de programación dinámica (Needleman-Wunsch o Smith- Waterman) o algoritmos heurísticos (FAST, BLAST).
- **Alineamiento de múltiples secuencias**: se comparan más de dos secuencias. Para ello se pueden utilizar diversos programas basados en algoritmos heurísticos como, por ejemplo, CLUSTALW.

Para el método de alineamiento de dos secuencias las técnicas más conocidas para realizar este proceso son:

- **Alineamiento global**: Es especialmente útil cuando las secuencias se parecen bastante, tienen una longitud similar y los dominios conservados se encuentran en el mismo orden. Abarca la totalidad de las secuencias comparadas, es decir, intenta alinear todos y cada uno de los residuos de las dos secuencias. Este tipo de alineamiento nos permitirá determinar si las secuencias son homólogas o no, si pertenecen a una misma familia o construir un árbol filogenético. Para este proceso se utiliza el algoritmo de Needleman-Wunsch.

- **Alineamiento local**: Es el más utilizado y resulta especialmente útil cuando se comparan secuencias muy divergentes, de igual o distinta longitud, pero que pueden contener una o más regiones conservadas con similitud local. Las regiones conservadas suelen

corresponder a dominios estructurales o funcionales que resultan cruciales para el mantenimiento de la estructura y/o función de la molécula. Para este procedimiento se emplea el algoritmo de Smith – Waterman.

- **Alineamiento semiglobal**: Es especialmente útil cuando se comparan secuencias de longitud muy distinta o secuencias en las que el final de una se solapa con el inicio de otra. Permite (1) detectar regiones de solapamiento que nos permitan ensamblar *contigs* a partir de fragmentos más pequeños, (2) comparar EST (*expressed sequence tags*) con ADN genómico para así poder distinguir los exones de los intrones y determinar la estructura del gen, y (3) descubrir patrones en una secuencia mucho más larga. Para este proceso se utiliza el algoritmo de Smith – Waterman ya que no aplica penalizaciones ni al inicio ni al final de la secuencia más larga.

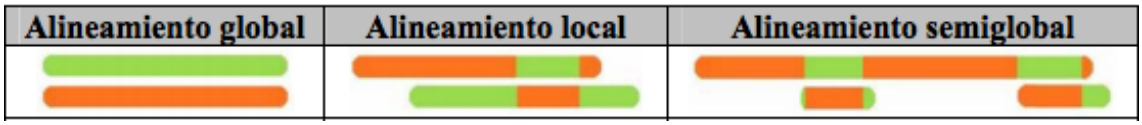


Figura 5.2

Para el método de alineamiento de múltiples secuencias las técnicas más utilizadas para realizar este proceso son:

- **Programación dinámica**: La técnica de programación dinámica es teóricamente aplicable a cualquier número de secuencias; sin embargo, y puesto que es computacionalmente costosa tanto en tiempo como en memoria, raramente se usa en su forma más básica para más de tres o cuatro secuencias.

- **Métodos progresivos**: Los métodos progresivos, jerárquicos, o por árbol, generan un alineamiento múltiple de secuencias alineando primero las secuencias más similares, para ir añadiendo sucesivamente al alineamiento secuencias o grupos menos relacionados, hasta que el conjunto problema completo ha sido incorporado a la solución.

- **Métodos iterativos**: Los métodos iterativos intentan mejorar el punto débil de los métodos progresivos: su fuerte dependencia de la precisión de los alineamientos de los emparejamientos iniciales. Los métodos iterativos optimizan una función objetivo basada en un

método seleccionado de puntuación de alineamiento mediante la asignación de un alineamiento global inicial y el posterior realineamiento de subconjuntos de secuencias.

Primeramente sabemos que tenemos un genoma de referencia que posee aproximadamente 3x10^9 nucleótidos. Como sabemos, las cadenas aisladas del ADN (muestras) se envían a Beijing Genome Institute para que ellos se encarguen de la secuenciación por NGS.

Una vez recibida la fragmentación de dicho ADN en millones de trozos entre 150 y 200 pares (en ficheros informáticos en formato ".fastq), hacemos una comparación de cada secuencia con el genoma de referencia en el que se obtiene diferentes lecturas (DP) en todas las posiciones. Estas lecturas se guardan en el fichero de alineamiento (.bam) además de información adicional (cromosoma, calidad del mapeo, longitud de la secuencia…). Mediante la herramienta desarrollada, podremos recorrer todas esas lecturas y obtener las regiones determinadas por el umbral introducido por el usuario.

Como hemos visto existen diferentes programas de alineamiento de secuencias, sin embargo, en UICHUIMI se utiliza uno en concreto llamado BWA. Describiremos a continuación este tipo de software.

Burrows – Wheeler Aligner

BWA es un paquete de software para el mapeo de secuencias pequeñas contra un genoma de referencia, como por ejemplo el genoma humano. Dicho software se compone de tres algoritmos: BWA-backtrack, BWA-SW y BWA-MEM. El primer algoritmo está diseñado para la secuencia de Illumina en el cual lee hasta 100 pares de bases, mientras que las dos restantes están diseñadas para las secuencias más largas que oscilan entre 70 y 1 millón de pares de bases. BWA-MEM y BWA-SW comparten características similares, como por ejemplo el soporte de lecturas largas y alineamientos de rupturas, pero BWA-MEM, por lo general se recomienda para consultas de alta calidad, ya que es más rápido y preciso. BWA-MEM también ofrece un mayor rendimiento que BWA-BackTrack para lecturas Illumina de 70 a 100 pares de bases.

BWA-back está principalmente diseñado para las tasas de error de secuenciación por debajo del 2%. Aunque los usuarios pueden tolerar más errores a través de líneas de comandos, su rendimiento se degrada rápidamente. Hay que tener en cuentas que para lecturas Illumina, bwa-

backtrack puede opcionalmente suprimir bases de baja calidad del alelo 3' antes de la alineación y por lo tanto es capaz de alinear más lecturas con una alta tasa de error en la cola, que es una postura típica de Illumina.

BWA-SW y BWA-MEM toleran más errores con alineamientos más largos. Las simulaciones sugieren que pueden funcionar bien con un error del 2% para una alineación de 100 pares de bases, el error de 3% para una de 200 pb (pares de bases), 5% para 500 pb y 10% para 1000 pb o más.[8]

El resultado de los algoritmos BWA está en formato ".SAM", adecuado para la búsqueda de las variantes.

5.5 Localización de variantes.

Una vez hemos hecho el alineamiento de las secuencias, en esta fase iremos desde el principio hasta el final del genoma de referencia comparándolo con el exoma reconstruido, para así ir identificando en qué posiciones es diferente. Este proceso es el más complicado de todos, ya que en cada posición normalmente se encuentra varias lecturas alineadas y hay que saber que existen variantes en las que sobra o falta un fragmento.

Para ello antes de realizar el proceso de localización de variantes, se hace un refinamiento a los alineamientos. Para realizar este proceso se emplea un sistema llamado *Genome Analysis Toolkit*. Por otra parte existe una herramienta llamada *Picard* en el cual éste se encarga de realizar una eliminación de lecturas repetidas que no agregan información, analizar coberturas de profundidad y recalibrar los valores de calidad. Con todas estas funcionalidades ayudan a facilitar la detección de Indels[xiv].[9] [10]

A continuación observamos en la siguiente Figura 5.3 como se ubican las secuencias en un fichero en formato ".SAM":

```
Ref  : TCGTAATCACGACA
seq1 : --GTAAACAC----
seq2 : TCGTAAACACG---
sep3 : ---TAAACACGA--
seq4 : -CGTAATCACG---
```

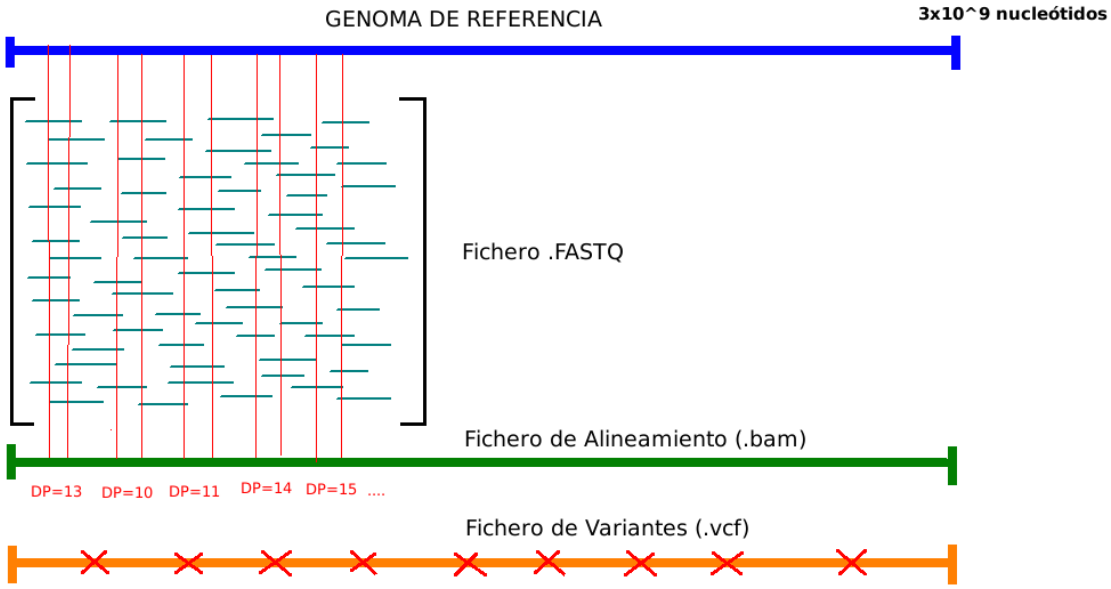
Figura 5.3

En esta imagen podemos observar como en la posición 7 de la referencia hay una Timina, sin embargo, en tres de las cuatro secuencias a

comparar vemos que hay una Adenina. Con esto podemos llegar a la hipótesis de que puede haber una variante en dicha posición, pero también cabe destacar que hay que tener en cuenta algunos puntos de vista.

Sabemos que a cada una de las lecturas le corresponde un valor de calidad, por lo que podemos a llegar a plantearnos que este resultado puede ser debido a un error del aparato de secuenciamiento. También puede haber sido un fallo en el programa de alineamiento, ya que cada uno de los alineamientos le corresponde un valor de bondad.

El objetivo es utilizar un programa que localice el mayor número de variantes verídicas posibles. En este caso la búsqueda de variantes es más fácil ya que sabemos que existen variantes por la inexistencia de una o más bases, por una repetición de un fragmento de cadena o por la inclusión de una cadena.



5.6 Anotación de variantes.

En esta fase obtenemos un fichero con miles de variantes, en el cual nuestro objetivo es identificar unas cuantas entre todas ellas. Para ello debemos hacer una serie de filtros con respecto a unas ciertas normas.

La información sobre la ubicación de las variantes es pobre ya que sólo indica la posición en la que se ha producido la variante, por lo que es

necesaria agregar más información sobre ello. Los campos más frecuentes que se añaden son:

- **Localización**: Indica que codón pertenece al genoma de referencia y qué codón pertenece la variante.

- **Sinonimia**: Las variantes son sinónimas cuando no representan un cambio en el aminoácido que codifica el codón donde se encuentra.

- **Peligro**: Dentro de las variantes que no son sinónimas, se hallan también las toleradas y peligrosas, dependiendo del cambio de las variantes.

- **Frecuencia**: Repetición de la variante en las distintas bases de datos. Habitualmente están clasificadas por regiones, por lo que se puede saber en qué partes del mundo son más usuales o no.

- **Conocimiento**: Base de datos e información donde se encuentra la variante, y así nos informa de hasta qué punto es conocida la variante.

Aunque existen herramientas de anotación que permiten ejecutarse de forma local, es más recomendable realizarla en los servicios web ya que los resultados estarán actualizados.

5.7 Filtrado.

Esta etapa es la más importante ya que en la anterior fase de localización de variantes, se identifican una media de 150.000 variantes. La finalidad de esta fase es hacer un filtrado de las variantes que pueden ser sospechosas de causar una enfermedad.

Para ello se hace un filtrado por el método por frecuencias, en el cual se suprimen las variantes que contengan una frecuencia muy alta, ya que se consideran que con dicha frecuencia perjudicaría a muchas personas. Por otra parte, también se suprimen las variantes con lecturas menores que 10,

que tengan valores de calidad por debajo de 50 y que no sean la opción más evidente.

Por último se clasifican las variantes según el tipo de hipótesis que se trate (homocigotos, dominante recesivo o uno o más variantes). También se pueden buscar variantes comunes entre diferentes pacientes sobre una misma enfermedad.

6. Desarrollo de la aplicación.

En esta etapa describiremos todo el proceso en la creación de la aplicación sobre la identificación de regiones de baja frecuencia de lectura en el ADN, desde la planificación del trabajo hasta la implementación de la herramienta. Se ha decidido que la aplicación sea llamada **MIST** (Missing Sequencing Tools).

6.1 PLANIFICACIÓN.

En este capítulo hablaremos sobre la metodología de trabajo que se va a utilizar, las diferentes plataformas usadas y los requisitos asignados para el desarrollo.

6.1.1 Metodología de Trabajo.

Para el desarrollo del módulo de software de identificación de regiones pobres en el ADN se ha utilizado como procedimiento de trabajo la metodología de desarrollo ágil. Este método basado en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto organizado y multidisciplinario. Cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad, sino que el objetivo es tener una especie de "demo" al final de cada iteración. Con esto permite al cliente probar y utilizar la versión ya desarrollada. Por ello se puede realizar los cambios en los requisitos desde el principio del desarrollo.

La finalidad es que los desarrolladores tengan una facilidad en los cambios del software a la hora de añadir o suprimir requisitos, y así no haya duración entre la iteración actual y la siguiente.

En el 2001 se creo un manifiesto ágil con la finalidad de descubrir mejores formas mejores de desarrollar software para nuestra propia experiencia, y no sólo para reducir el tiempo de desarrollo sino para mejorar también la eficiencia. Estos son los 4 puntos del manifiesto ágil:

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.

- Respuesta ante el cambio sobre seguir un plan.

Además también se crearon 12 principios en dicho manifiesto:

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

- El software funcionando es la medida principal de progreso.

- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma inmediata.

- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

- A intervalos regulares el equipo reflexiona cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

También hay que tener que en cuenta que dicho método tiene sus inconvenientes o desventajas. Por ejemplo estos pueden ser:

- Falta de documentación del diseño. Al no haber documentación es el código lo que se toma documentación.

- Problemas derivados de la comunicación oral. No hace falta decir que algo que está escrito "no se puede borrar", en cambio, algo dicho es muy fácil crear ambigüedad.

- Fuerte dependencia de las personas.

- Falta de reusabilidad derivada de la falta de documentación.

- Restricciones en cuanto a tamaños de los proyectos.

- Problemas derivados del fracaso de los proyectos ágiles.

En nuestro proyecto como sólo habrá un desarrollador y un cliente, no podremos utilizar dicha metodología ya que esta enfocada a grupos. Pero podemos utilizar algunas características de dicho método a nuestro desarrollo. Por ejemplo podemos realizar reuniones frecuentes para observar cómo va progresando el desarrollo del proyecto, lanzamientos habituales del desarrollo, mejoras progresivas del proyecto o cambios sin muchas modificaciones en los requisitos.

6.1.2 Plataformas de desarrollo.

En este apartado hablaremos sobre las diferentes plataformas que se ha utilizado para el desarrollo del trabajo fin de grado. En nuestra aplicación podríamos haber utilizado cualquier lenguaje de programación, pero la finalidad es que la herramienta desarrollada sea de fácil portabilidad y mantenimiento. Por ello el lenguaje de programación que hemos utilizado es Java. Hay diferentes razones por las que hemos hecho dicha elección. La principal razón es que ya existe una aplicación global desarrollada llamada, "**DNAnalytics**", en dicho lenguaje. Dicha aplicación se encarga de encapsular los distintos procesos para el análisis de los datos genéticos y permite al personal de la Unidad conocer de manera rápida las variantes candidatas de una enfermedad.

Así que utilizar este lenguaje será mucho más factible a la hora realizar una posible integración de nuestro módulo a la aplicación general. Otro de los argumentos es que Java es un lenguaje independiente de la plataforma, es decir, cualquier programa creado a través de Java podrá funcionar correctamente en ordenadores de todo tipo y con sistemas operativos distintos.

También se ha utilizado como código de programación Shell, en el que hemos realizado pequeños scripts para nuestros filtrados de datos en los archivos de los alineamientos de las secuencias de diferentes pacientes.

6.1.3 Recursos.

A continuación detallaremos los recursos tanto hardware como software que hemos utilizado para el desarrollo de nuestra aplicación para la localización de regiones pobres en el ADN.

Recursos Hardware

Los ficheros de alineamiento del ADN requieren alto costo en lo computacional, en el cual se necesita poseer una cantidad alta de memoria y de procesadores con bastante potencia. Para ello se necesita múltiples núcleos para comparar y analizar los datos en paralelo. También es muy recomendable tener un disco de almacenamiento externo, con la finalidad de conservar tanto los resultados obtenidos como los datos a investigar.

Actualmente, en UICHUIMI se utiliza dos ordenadores con los que poder trabajar con estos ficheros. A continuación especificaremos los requisitos que tienen ambos:

Componente	Descripción
Memoria RAM	16 GB DDR3
CPU	Intel i5 de dos núcleos
Almacenamiento de datos	500 GB

Componente	Descripción
Memoria RAM	16 GB DDR3
CPU	Intel i7 3770 (4 núcleos / 8 subprocesos)
Almacenamiento de datos	SSD 250 GB / 3TB

Recursos Software

Muchos de los programas relacionados con el análisis del ADN han sido desarrollados en una distribución de Linux, por lo que ha sido recomendable utilizar el sistema operativo CentOS (versión 6.4). Se ha elegido este sistema operativo porque ofrece velocidad, estabilidad y confiabilidad. Comparado con otros sistemas operativos basados en Linux, CentOS sólo ejecuta las versiones más básicas y estables de programas, reduciendo el riesgo de bloqueos del sistema. Por otra parte puede operar mucho más rápido que los sistemas operativos basados en Linux similares porque sólo ejecuta las versiones básicas de software. También se puede ejecutar en una computadora durante mucho tiempo sin requerir ninguna actualización adicional del sistema.

Como hemos elegido el lenguaje de programación Java, se utilizará el entorno de programación Netbeans IDE 7.3. También hemos tenido que utilizar para filtros de datos el lenguaje Shell. Además para la lectura de los ficheros de alineamiento del ADN hemos instalado una herramienta para dicho formato llamado *Samtools*.

6.2 DESARROLLO.

En esta fase explicaremos el proceso de desarrollo del módulo de software para la identificación de regiones del ADN con baja frecuencia de lectura. También informaremos de las progresiones, los cambios y las correcciones que se han ido realizando durante el desarrollo del módulo.

Como ya sabemos el flujo de análisis del ADN está compuesto por varias fases (las podemos observar en la Figura 6.1) que hemos explicado en el Capítulo 5:

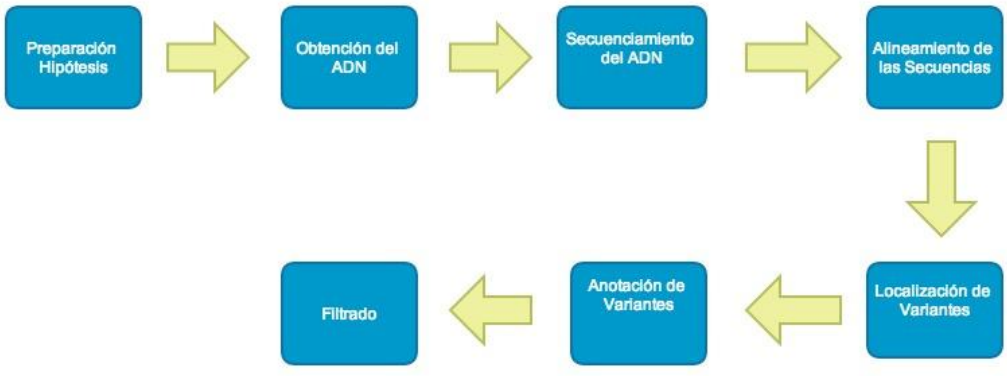


Figura 6.1

En este Trabajo de Fin de Grado pondremos toda la atención en la fase de “Alineamiento de Secuencias” ya que los ficheros con los que vamos a trabajar pertenecen a esta etapa.

Como hemos observado en el flujo de análisis del ADN, podremos utilizar un programa llamado “*Burrows – Wheeler Aligner”* para realizar el alineamiento de las secuencias, generando un fichero en formato “.SAM” y su correspondiente binario “.BAM”. En nuestro caso trabajaremos con el último ya que es una versión más comprimida que el primero. [12]

En un archivo “.SAM” recibimos los alineamientos con sus respectivas calidades.

Primeramente, para poder leer estos tipos de ficheros debemos tener instalada en nuestra distribución de Linux una herramienta llamada “Samtools”.[13] Para saber más sobre las diferentes opciones de esta herramienta, basta con dirigirnos al anexo “Herramientas adicionales”.
En nuestro caso para el desarrollo de la identificación de regiones pobres en el ADN utilizaremos el siguiente comando:

```
samtools view
```

Dicho comando nos permite visualizar estos ficheros en formato “.SAM”. Para ello abriremos un terminal en nuestra distribución Linux y ejecutaremos el comando en el cual observamos el contenido como el de la Figura 6.2 de algún determinado paciente:

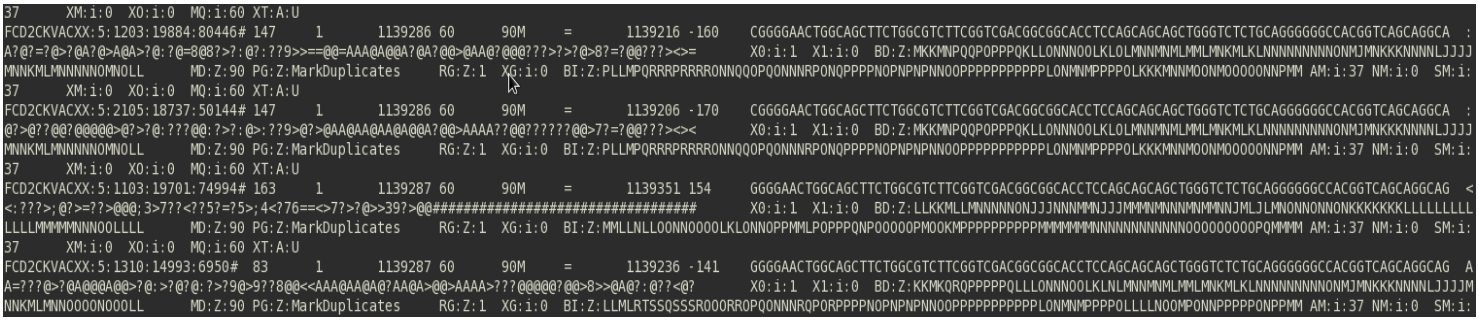


Figura 6.2

A continuación analizaremos con detalle cómo están estructurados estos tipos de ficheros en formato “.BAM” y describiremos cada uno de los campos asociados:

- *QNAME*: Es una especie de identificador. Es usado para agrupar / identificar alineamientos que están juntos, como alineamientos paralelos o una lectura que aparece en múltiples alineaciones.
- *FLAG*: Una serie de bit a bit de información que describe la alineación. Contiene varias informaciones.
- *RNAME*: Nombre de la secuencia de referencia, a menudo contiene el nombre del cromosoma (1, 2, 3…).
- *POS*: La posición más a la izquierda de donde este se asigna a la referencia de alineación. En formato ".SAM", la referencia empieza por 1 y por el formato ".BAM" comienza en 0.
- *MAPQ*: Calidad de mapeo. Un valor de 255 indica que la calidad del mapeo no está disponible.
- *CIGAR*: Es una string que indica como las bases que se alinean, ya sea emparejado / desajuste con la referencia (M), son eliminados de la referencia (D), y son insertados en los que no están en la referencia (I). Este formato extendido permite, además, realizar más operaciones.
- *RNEXT*: Informa del nombre de la secuencia de referencia de la siguiente alineación en este grupo. Si aparece un "=" quiere decir que la siguiente se encuentra en el mismo cromosoma.
- *PNEXT*: Te indica la posición de la alineación principal de la siguiente en la plantilla. Se establece con valor 0 cuando la información no se encuentra disponible. Este campo es igual a POS en la línea principal de la siguiente lectura. Si NEXT es 0, las no suposiciones pueden ser hechas sobre NEXT y bit 0x20.
- *TLEN*: La longitud de este grupo desde la posición más a la izquierda a la posición más a la derecha.
- *SEQ*: Las cadena de secuencia para este alineamiento.
- *TAGS*: Etiquetas con información adicional que también se encuentra dentro de la alineación.

Por otra parte, también a través del siguiente comando:

```
samtools view –H
```

Podremos observar el número de pares de bases que contiene cada uno de los cromosomas. Podemos ver este caso mediante la siguiente ilustración (Figura 6.3):

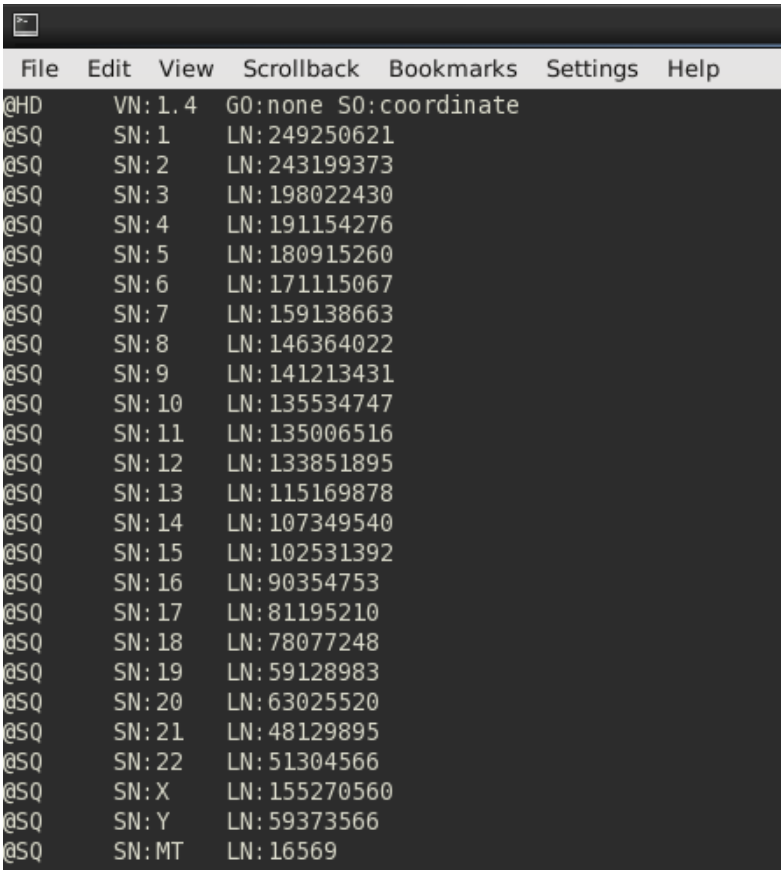


Figura 6.3

En dicha imagen podremos ver que la información está estructurada además de la cabecera, con los siguientes campos:

- @SQ: Identificador.
- SN: Nombre del cromosoma.
- LN: Número de pares de bases que contiene dicho cromosoma.

6.2.1 Objetivos.

En este capítulo describiremos los diversos objetivos que necesitan los usuarios en UICHUIMI y sus diferentes características. El desarrollo de estos objetivos se darán por correctos cuando sean validados por el/los clientes. A continuación mostraremos estos objetivos:

Objetivo 1	Localización de regiones pobres
Versión	1.0
Autores	Deyán F. Guacarán Sabogal Antonio Tugores Cester
Descripción	La herramienta debe permitir hacer una búsqueda de regiones con lecturas de baja frecuencia, determinada por un umbral introducido por el usuario. Además debe realizar una comparación con las regiones de los exones de los humanos.
Importancia	Muy Alta
Estado	Validado

Tabla 6.1: Objetivo 1

Objetivo 2	Intersección entre regiones pobres.
Versión	1.0
Autores	Deyán F. Guacarán Sabogal Antonio Tugores Cester
Descripción	La herramienta debe permitir hacer una intersección de las regiones con baja frecuencia de lectura entre múltiples pacientes.
Importancia	Muy Alta
Estado	Validado

Tabla 6.2: Objetivo 2

Objetivo 3	Resultados Estadísticos
Versión	1.0
Autores	Deyán F. Guacarán Sabogal Antonio Tugores Cester
Descripción	La herramienta debe permitir extraer resultados estadísticos de los ficheros tanto de las regiones con baja frecuencia de lectura como en la intersección entre múltiples pacientes.
Importancia	Alta
Estado	Validado

Tabla 6.3: Objetivo 3

6.2.2 Requisitos Funcionales

Los requisitos funcionales son aquellos que definen una función del sistema de software o sus componentes, con la finalidad de que los usuarios logren sus objetivos.

Requisito Funcional 1	Progreso
Versión	1.0
Autores	Deyán F. Guacarán Sabogal
Descripción	La herramienta deberá mostrar el progreso de ejecución mediante una barra de progreso.
Importancia	Alta
Estado	Validado

Tabla 6.4: Requisito Funcional 1

6.2.2.1 Casos de Uso

En este capítulo hablaremos sobre los casos de uso, aquellos que realizan una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Antes de explicar los casos de uso, señalamos el actor que interactúa con la aplicación.

Actor 1	Usuario
Versión	1.0
Autores	Deyán F. Guacarán Sabogal
Descripción	El actor representa a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad.

Tabla 6.5: Actor 1 1

A continuación mostraremos todos los casos de uso que intervienen en el programa y sus relaciones, por lo que para ello realizaremos un esquema en UML. Lo podemos observar en la siguiente Figura 6.4:

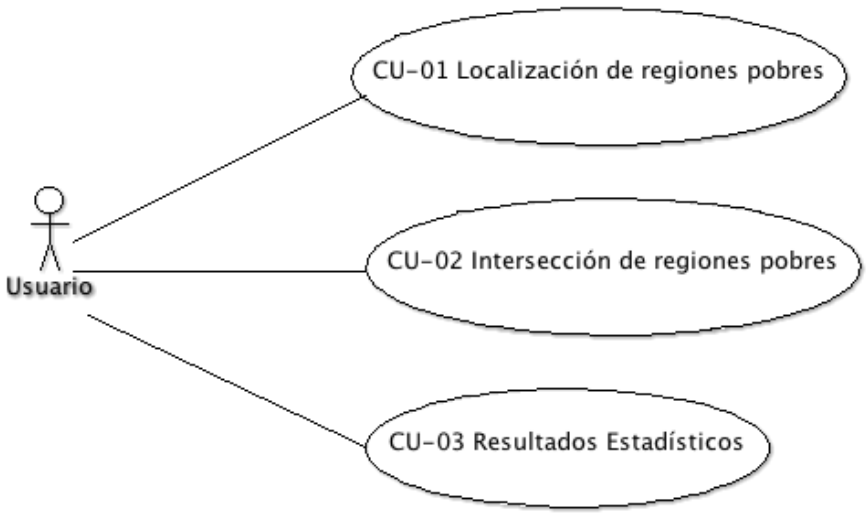


Figura 6.4

Una vez especificados los casos de uso que el programa contiene, procederemos a hacer una descripción del flujo de acciones de cada uno de ellos. Para ello también haremos capturas de pantalla de cómo está organizado todos ellos.

Caso de Uso 1	Localización de regiones pobres
Versión	1.0
Autores	Deyán F. Guacarán Sabogal
Actor Principal	Usuario
Descripción	El usuario localiza las regiones con una frecuencia de lectura baja.
Precondición	Que haya un fichero en formato ".bam" correspondiente al alineamiento de las secuencias de un determinado paciente.
Flujo	
1. El usuario selecciona el fichero en formato ".BAM" como fichero de entrada. 2. El usuario selecciona el fichero de referencia de los exones de los humanos como fichero también de entrada. 3. El usuario introduce el umbral de lecturas. 4. El usuario guarda el fichero de salida en la ubicación que desee. 5. El usuario pulsa el botón "Start". 6. El sistema muestra el progreso de ejecución mediante la barra de progreso. 7. El sistema muestra un panel de la consola con los cálculos que se están realizando en ese momento.	

Tabla 6.6: Caso de Uso 1

A continuación veremos una ilustración del caso de uso "Localización de regiones pobres" mediante la Figura 6.5, en el cual podremos ver el proceso de ejecución de búsqueda de regiones con baja frecuencia. Para ello hemos puesto de ejemplo un umbral de 10, es decir, regiones que tengan lecturas de cero a nueve:

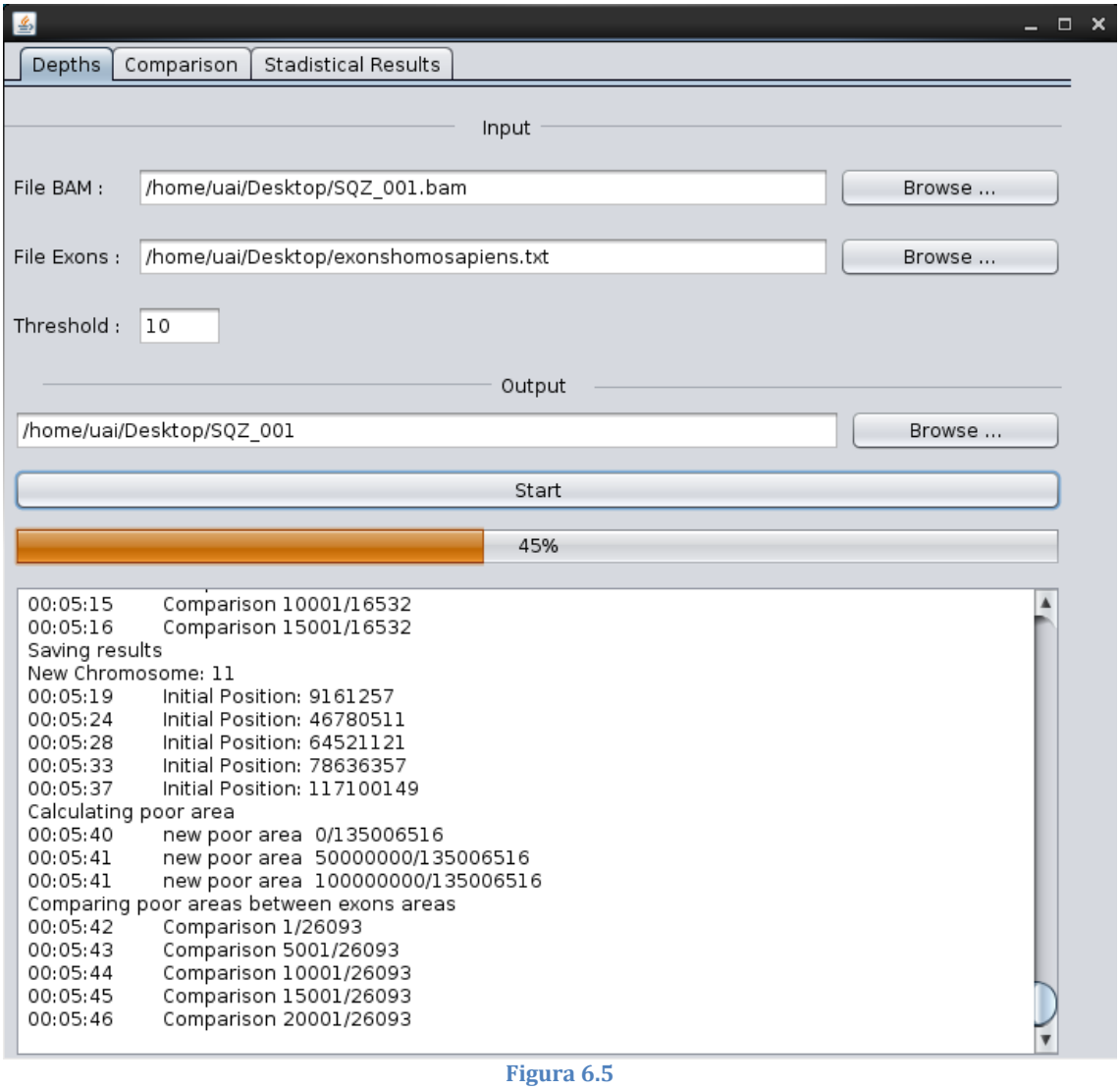


Figura 6.5

Caso de Uso 2	Intersección entre regiones pobres
Versión	1.0
Autores	Deyán F. Guacarán Sabogal
Actor Principal	Usuario
Descripción	El usuario realiza la intersección de regiones pobres entre múltiples pacientes.

Precondición	Que haya disponible como mínimo dos o más ficheros de diferentes pacientes con sus correspondientes regiones pobres.
Flujo	
1. El usuario selecciona una lista de ficheros de los diferentes pacientes con regiones pobres como ficheros de entrada. 2. El usuario guarda el fichero de salida en la ubicación que desee. 3. El usuario pulsa el botón "Start".	

Tabla 6.7: Caso de Uso 2

A continuación mostraremos una ilustración de cómo está estructurado el caso de uso "Intersección entre regiones pobres" mediante la Figura 6.6:

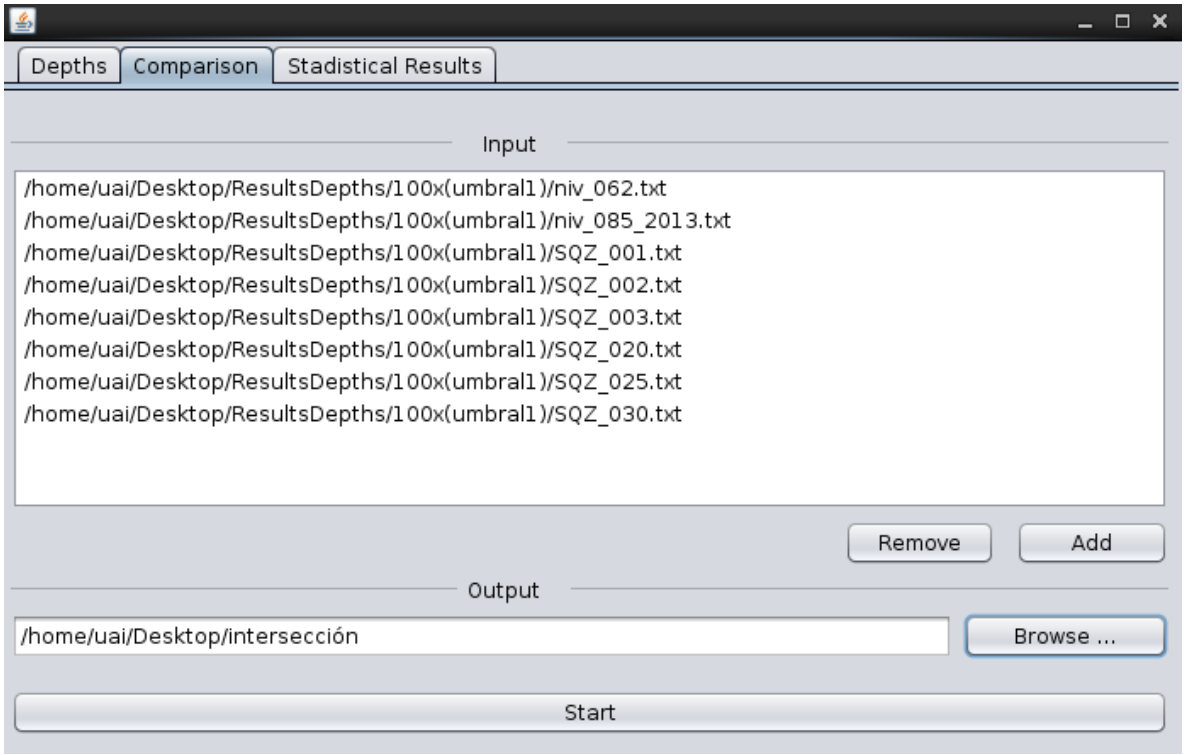


Figura 6.6

Caso de Uso 3	Resultados Estadísticos
Versión	1.0
Autores	Deyán F. Guacarán Sabogal
Actor Principal	Usuario
Descripción	El usuario obtiene diferentes resultados estadísticos tanto para la intersección de regiones pobres de

	múltiples pacientes como también para pacientes individuales.
Precondición	Que haya disponible ficheros con regiones con baja frecuencia de lectura de diferentes pacientes o la intersección entre ellos.
Flujo	
1. El usuario selecciona el fichero para obtener los resultados estadísticos como fichero de entrada. 2. El usuario guarda el fichero de salida en la ubicación que desee. 3. El usuario pulsa el botón "Start".	

Tabla 6.8: Caso de Uso 3

A continuación mostraremos una ilustración de cómo está estructurado el caso de uso "Resultados Estadísticos" mediante la Figura 6.7:

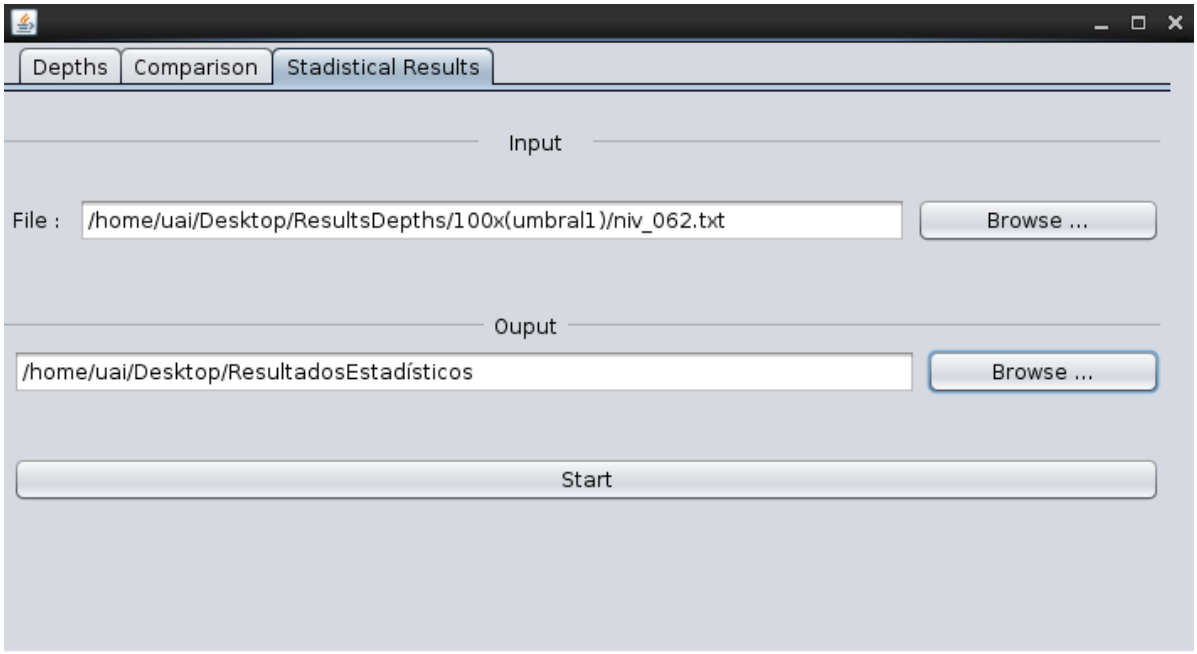


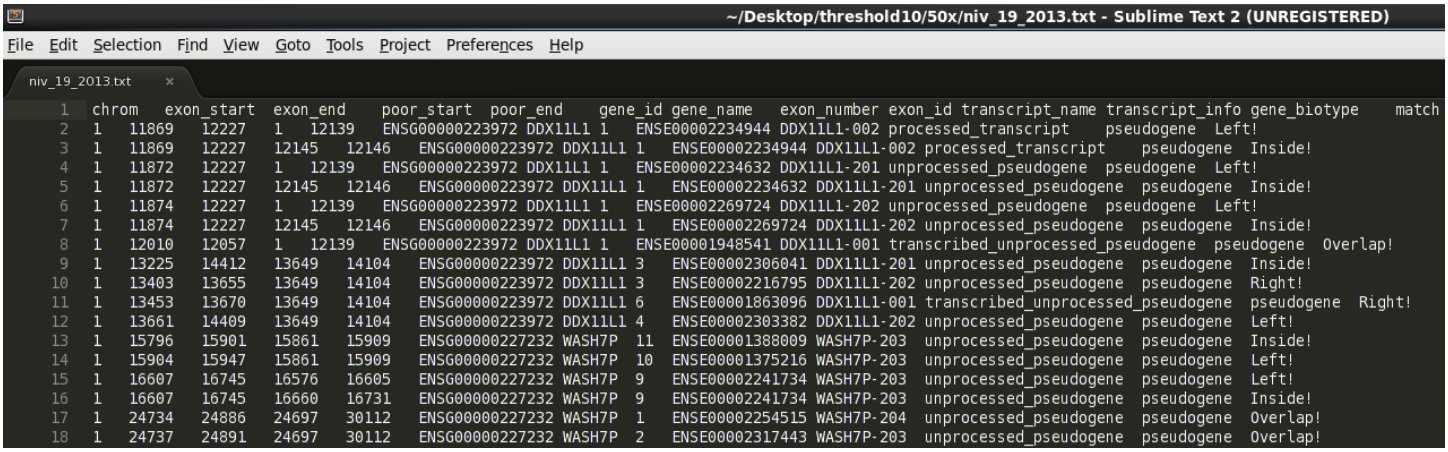
Figura 6.7

6.3 RESULTADOS

En este capítulo hablaremos sobre los diferentes resultados que generan la aplicación sobre la localización de regiones con baja frecuencia de lecturas, además de otras funciones adicionales como los resultados estadísticos de dichos resultados generados.

6.3.1 Depths

En esta fase hablaremos sobre el fichero que se genera al realizar el cálculo del número de lecturas de todos los cromosomas. Durante el proceso se extraen las regiones con respecto al umbral introducido por el usuario. Una vez obtenido dichas regiones pobres, se realiza una comparación con las regiones del genoma de referencia de los "homo sapiens" para saber en que zonas son afectadas. A continuación vemos una ilustración de un fichero generado de un determinado paciente al realizar la ejecución en la Figura 6.8 y explicaremos cada uno de los campos:



	chrom	exon_start	exon_end	poor_start	poor_end	gene_id	gene_name	exon_number	exon_id	transcript_name	transcript_info	gene_biotype	match
2	1	11869	12227	1	12139	ENSG00000223972	DDX11L1	1	ENSE00002234944	DDX11L1-002	processed_transcript	pseudogene	Left!
3	1	11869	12227	12145	12146	ENSG00000223972	DDX11L1	1	ENSE00002234944	DDX11L1-002	processed_transcript	pseudogene	Inside!
4	1	11872	12227	1	12139	ENSG00000223972	DDX11L1	1	ENSE00002234632	DDX11L1-201	unprocessed_pseudogene	pseudogene	Left!
5	1	11872	12227	12145	12146	ENSG00000223972	DDX11L1	1	ENSE00002234632	DDX11L1-201	unprocessed_pseudogene	pseudogene	Inside!
6	1	11874	12227	1	12139	ENSG00000223972	DDX11L1	1	ENSE00002269724	DDX11L1-202	unprocessed_pseudogene	pseudogene	Left!
7	1	11874	12227	12145	12146	ENSG00000223972	DDX11L1	1	ENSE00002269724	DDX11L1-202	unprocessed_pseudogene	pseudogene	Inside!
8	1	12010	12057	1	12139	ENSG00000223972	DDX11L1	1	ENSE00001948541	DDX11L1-001	transcribed_unprocessed_pseudogene	pseudogene	Overlap!
9	1	13225	14412	13649	14104	ENSG00000223972	DDX11L1	3	ENSE00002306041	DDX11L1-201	unprocessed_pseudogene	pseudogene	Inside!
10	1	13403	13655	13649	14104	ENSG00000223972	DDX11L1	3	ENSE00002216795	DDX11L1-202	unprocessed_pseudogene	pseudogene	Right!
11	1	13453	13670	13649	14104	ENSG00000223972	DDX11L1	6	ENSE00001863096	DDX11L1-001	transcribed_unprocessed_pseudogene	pseudogene	Right!
12	1	13661	14409	13649	14104	ENSG00000223972	DDX11L1	4	ENSE00002303382	DDX11L1-202	unprocessed_pseudogene	pseudogene	Left!
13	1	15796	15901	15861	15909	ENSG00000227232	WASH7P	11	ENSE00001388009	WASH7P-203	unprocessed_pseudogene	pseudogene	Inside!
14	1	15904	15947	15861	15909	ENSG00000227232	WASH7P	10	ENSE00001375216	WASH7P-203	unprocessed_pseudogene	pseudogene	Left!
15	1	16607	16745	16576	16605	ENSG00000227232	WASH7P	9	ENSE00002241734	WASH7P-203	unprocessed_pseudogene	pseudogene	Left!
16	1	16607	16745	16660	16731	ENSG00000227232	WASH7P	9	ENSE00002241734	WASH7P-203	unprocessed_pseudogene	pseudogene	Inside!
17	1	24734	24886	24697	30112	ENSG00000227232	WASH7P	1	ENSE00002254515	WASH7P-204	unprocessed_pseudogene	pseudogene	Overlap!
18	1	24737	24891	24697	30112	ENSG00000227232	WASH7P	2	ENSE00002317443	WASH7P-203	unprocessed_pseudogene	pseudogene	Overlap!

Figura 6.8

- *Chrom*: Nombre del cromosoma. En nuestros ficheros irán del cromosoma 1 al cromosoma Y.
- *Exon_Start*: Comienzo de la posición de la región del exón.
- *Exon_End*: Final de la posición de la región del exón.
- *Poor_Start*: Comienzo de la posición de la región pobre.
- *Poor_End*: Final de la posición de la región pobre.
- *Gene_ID*: El identificador estable para el gen.
- *Gene_Name*: Nombre del gen.
- *Exon_Number*: Posición del exón en la transcripción.
- *Exon_ID*: El identificador estable para el exón.
- *Transcript_Name*: El nombre de la transcripción.
- *Transcript_Info*: Información de la transcipción.
- *Gene_Biotype*: El biotipo de este gen.
- *Match*: La zona del exón que resulta afectada. Pueden ser de cuatro opciones: left, right, inside y overlap. Hablaremos sobre ello a continuación.

En el primer caso, puede ocurrir que la zona que esté afectada sea una parte izquierda de la región del exón. Podemos ver un caso en la siguiente ilustración (Figura 6.9):

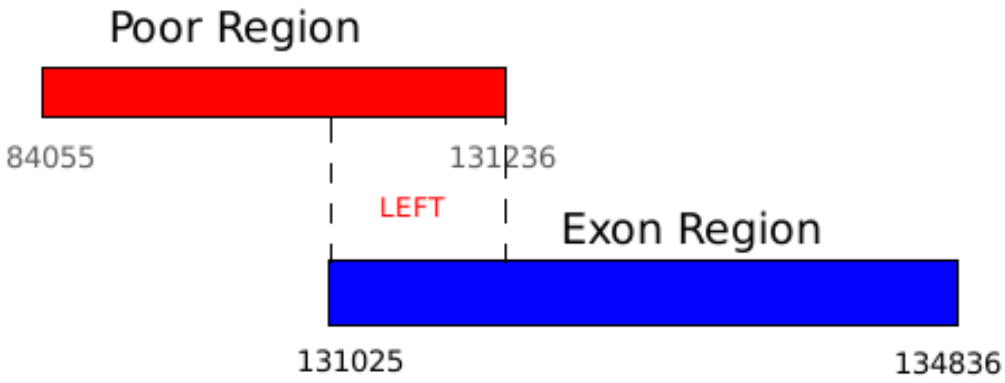


Figura 6.9

En el segundo caso, puede ocurrir que la zona que esté afectada sea una parte derecha de la región del exón. Podemos observar un caso en la siguiente ilustración (Figura 6.10):

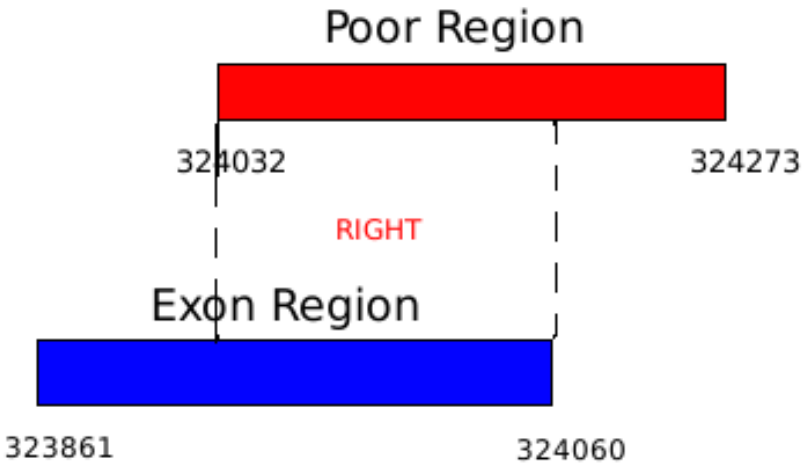


Figura 6.10

En el tercer caso, puede ocurrir que la zona afectada se encuentre dentro de la región del exón. Podemos ver un caso en la siguiente ilustración (Figura 6.11):

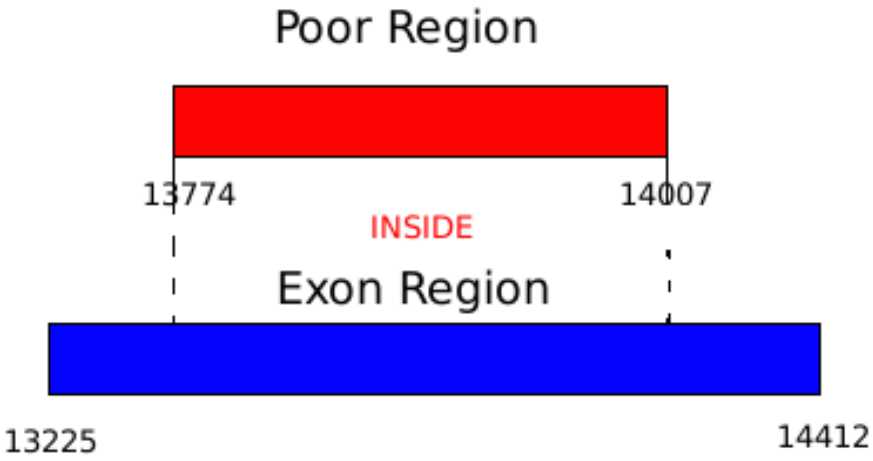


Figura 6.11

En el último caso, puede darse la opción de que la zona afectada sea su totalidad ya que la región con baja de frecuencia de lectura sea de más tamaño que la región del genoma de referencia. Podemos ver el caso en la siguiente ilustración (Figura 6.12):

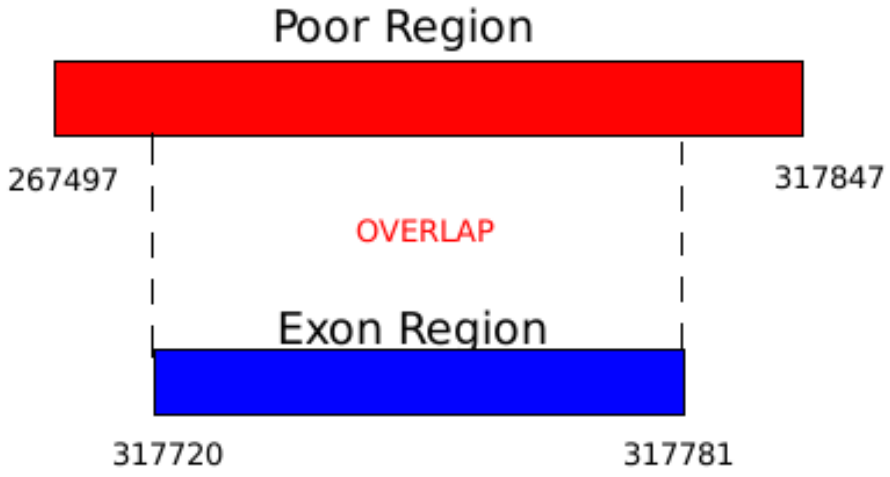


Figura 6.12

6.3.2 Intersection

En la siguiente Figura 6.14 describiremos cómo está estructurado el fichero generado con sus respectivos campos:

int100x(thr1,lenght50).txt

	chrom	exon_start	exon_end	gene_id	gene_name	exon_number	exon_id	transcript_name	transcript_info	gene_biotype
2	1	52473	53312	ENSG00000268020	OR4G4P	1	ENSE00003698237	OR4G4P-001	unprocessed_pseudogene	pseudogene
3	1	62948	63887	ENSG00000240361	OR4G11P	1	ENSE00001830178	OR4G11P-001	unprocessed_pseudogene	pseudogene
4	1	69091	70008	ENSG00000186092	OR4F5	1	ENSE00002319515	OR4F5-001	protein_coding	protein_coding
5	1	89295	91629	ENSG00000238009	RP11-34P13.7	4	ENSE00001846804	RP11-34P13.7-001	lincRNA	lincRNA
6	1	89551	90050	ENSG00000239945	RP11-34P13.8	2	ENSE00001927725	RP11-34P13.8-001	lincRNA	lincRNA
7	1	90287	91105	ENSG00000239945	RP11-34P13.8	1	ENSE00001907785	RP11-34P13.8-001	lincRNA	lincRNA
8	1	110953	111357	ENSG00000238009	RP11-34P13.7	3	ENSE00001879696	RP11-34P13.7-002	lincRNA	lincRNA
9	1	141474	143011	ENSG00000241860	RP11-34P13.13	2	ENSE00001911218	RP11-34P13.13-004	antisense	processed_transcript
10	1	146386	146509	ENSG00000241860	RP11-34P13.13	8	ENSE00001853409	RP11-34P13.13-001	lincRNA	processed_transcript
11	1	160446	160690	ENSG00000241599	RP11-34P13.9	1	ENSE00001882727	RP11-34P13.9-001	lincRNA	lincRNA
12	1	161314	161525	ENSG00000241599	RP11-34P13.9	2	ENSE00001895361	RP11-34P13.9-001	lincRNA	lincRNA

Figura 6.14

- *Chrom*: Nombre del cromosoma. En nuestros ficheros irán del cromosoma 1 al cromosoma Y.
- *Exon_Start*: Comienzo de la posición de la región del exón.
- *Exon_End*: Final de la posición de la región del exón.
- *Gene_ID*: El identificador estable para el gen.
- *Gene_Name*: Nombre del gen.
- *Exon_Number*: Posición del exón en la transcripción.
- *Exon_ID*: El identificador estable para el exón.
- *Transcript_Name*: El nombre de la transcripción.
- *Transcript_Info*: Información de la transcripción.
- *Gene_Biotype*: El biotipo de este gen.

Como vemos en la imagen anterior, las regiones con baja de frecuencia de lectura no se encuentran en los ficheros, ya que sólo haremos la comparación de los exones comunes correspondientes al genoma de referencia, y no tendremos necesidad de añadir dichas regiones pobres.

6.3.3 Statistical Results

En esta fase haremos balance de los resultados que se han generado tanto en los ficheros con baja frecuencia de lectura como en los ficheros con las intersecciones entre múltiples pacientes. Para ello obtendremos de cada uno de ellos una serie de estadísticas para hacernos una idea de la cantidad de información que se está perdiendo en el genoma de referencia.

En la Figura 6.15 mostraremos una imagen como ejemplo del fichero que se genera una vez ejecutado el procedimiento:

StadisticalResults.txt

```
/******** Genes ********/
Number of Genes: 51798
Protein-coding genes : 23327
Long non-coding RNA genes : 7720
Small non-coding RNA genes : 1906
Processed Pseudogenes genes : 9467
Unprocessed Pseudogenes genes : 2460
Unitary Pseudogenes genes : 172
Polymorphic Pseudogenes genes : 30
Pseudogenes genes : 376
--------------------
Processed Transcript genes: 13008
Transcribed Unprocessed Pseudogene genes: 638
Retained Intron genes: 8833

/******** Exons *******/
Number of exons: 277582
Protein-coding exons : 98833
Long non-coding RNA exons : 26635
Small non-coding RNA exons : 1919
Processed Pseudogenes exons : 23312
Unprocessed Pseudogenes exons : 9586
Unitary Pseudogenes exons : 686
Polymorphic Pseudogenes exons : 102
Pseudogenes exons : 1059
-------------------
Processed Transcript exons: 40991
Transcribed Unprocessed Pseudogene exons: 3067
Retained Intron exons: 29382
```

Figura 6.15

A continuación haremos un análisis de todos los componentes estadísticos que hemos obtenido tanto de un fichero con baja frecuencia de lectura como de un fichero de intersección entre múltiples pacientes:

- *Number of Genes*: Número de genes afectados en todos los cromosomas.
- *Proteing - Coding Genes*: Número de genes afectados en el cual codifican proteínas.
- *Long non – coding RNA Genes*: Número de genes afectados en el cual codifican transcritos no proteicos (contienen más de 200 nucleótidos)

- *Small non – coding RNA Genes*: Significa lo mismo que el anterior, pero la diferencia es que estos son pequeños.
- *Processed Pseudogenes Genes*: Número de genes afectados en el cual son pseudogenes procesados.
- *Unprocessed Pseudogenes Genes*: Número de genes afectados en el cual son pseudogenes que no han sido procesados.
- *Unitary Pseudogenes Genes*: Número de genes afectados en los cuales varias mutaciones pueden detener un gen transcrito o traducido con éxito.
- *Polymorphic Pseudogenes Genes*: Número de genes que están intactos en el genoma de otros individuos de la misma especie.
- *Pseudogenes Genes*: Número de genes que han perdido su capacidad de codificación de proteínas o ya no están expresados en la célula.
- *Process Transcript Genes*: Número de genes afectados que no han sido transcritos.
- *Transcribed Unprocessed Pseudogenes Genes*: Número de pseudogenes transcritos no procesados.
- *Retained Intron Genes*: Número de genes que contienen secuencias intrónicas.

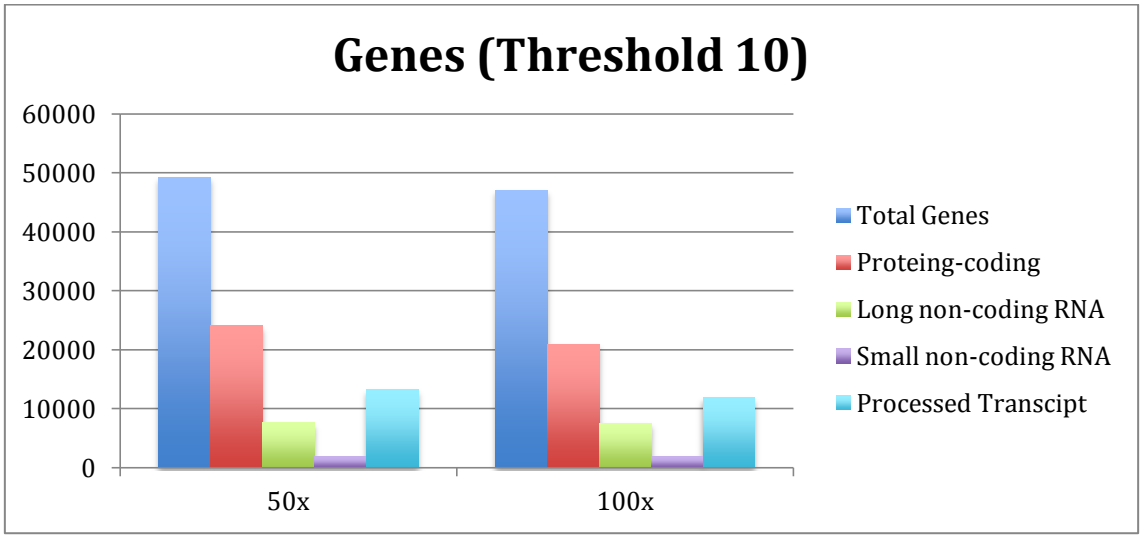
Al igual que con las estadísticas a nivel de gen, también podemos observar en la anterior imagen que hemos obtenido resultados a nivel de exón.

6.3.4 Caso Práctico

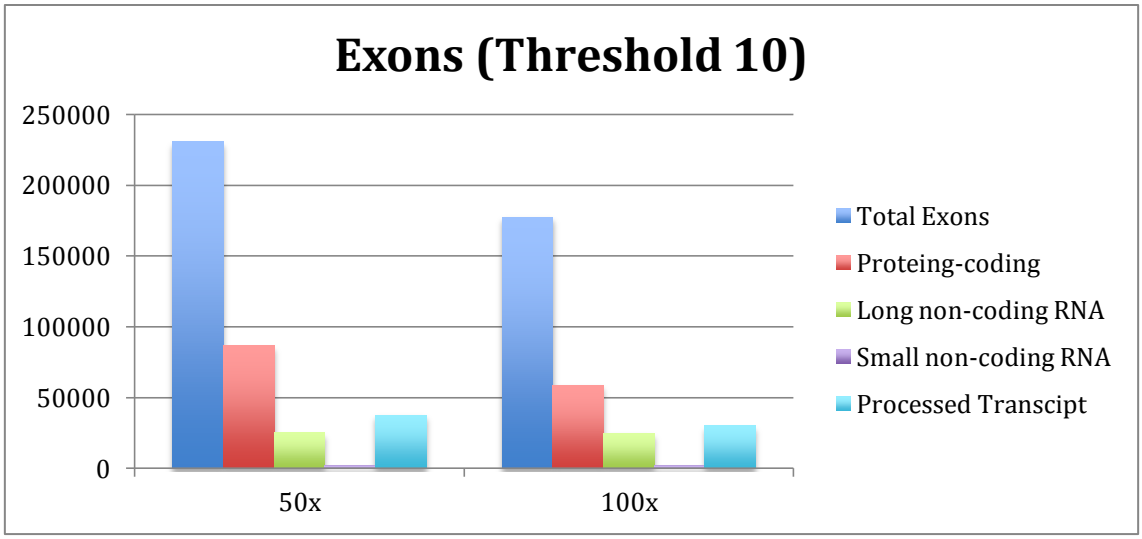
Una vez localizadas las regiones que contienen baja frecuencia de lectura del ADN en un determinado paciente, el objetivo es hacer el mismo proceso para múltiples pacientes. Luego agrupamos a los que han sido secuenciados con una cobertura de 50x (50 veces que se ha secuenciado de forma fiable una región) y a los de cobertura de 100x (100 veces que se ha secuenciado). A cada grupo le haremos la intersección con la finalidad de encontrar los exones comunes. Así, encontraremos las regiones de exones que están afectadas en el genoma de referencia.

Hemos llegado a varias conclusiones una vez hecho el estudio que explicaremos en el capítulo 7. Podemos ver en las siguientes ilustraciones los resultados obtenidos, así como sus gráficas correspondientes.

Threshold 10	50x	100x
Total Genes	49277	47108
Proteing-coding	24088	20952
Long non-coding RNA	7640	7547
Small non-coding RNA	1909	1901
Processed Transcipt	13213	11949



Threshold 10	50x	100x
Total Exons	230980	177617
Proteing-coding	86697	58784
Long non-coding RNA	25259	24795
Small non-coding RNA	1909	1902
Processed Transcipt	37802	30128


Exons (Threshold 10)
250000
200000
150000
100000
50000
0
50x
100x
Total Exons
Proteing-coding
Long non-coding RNA
Small non-coding RNA
Processed Transcipt

7. Conclusiones y Trabajos Futuros.

Como ya sabemos, existe un problema en el que la cobertura de la secuenciación de un determinado paciente es variable. Por ello, hay regiones del ADN en el que es posible que no se haya secuenciado el exoma, y por consecuente la imposibilidad de identificar dichas variantes en dichas regiones.

Para ello, hemos creado la herramienta que se encarga de detectar regiones con poca cobertura de secuenciación (MIST- Missing Sequencing Tools). Como ya sabemos dicha aplicación filtra y calcula regiones del fichero de alineamiento con un número de lecturas determinado por un umbral introducido por el usuario.

Para llegar a una veracidad convincente como sabemos que heredamos dos cromátidas[xv] (una de nuestra madre y otra del padre, en el cual las dos forman un cromosoma) en el que cada una de las cromátidas está formada por dos cadenas, hemos contemplado que el mínimo de lecturas leídas debe ser 4. Cabe destacar que en todos los cromosomas tenemos dos alelos[xvi], menos en el cromosoma Y en el que sólo tenemos uno.

Aun así nosotros hemos puesto un umbral de 10, ya que además de ser un límite estándar podemos obtener una mayor veracidad en nuestras investigaciones.

Después de haber hecho multitud de procesos con diferentes ficheros los resultados obtenidos han sido sorprendentes. Se ha observado que hay demasiados genes con baja frecuencia de lectura y que son de tamaño considerable. Hay que destacar que en cada uno de los genes podemos encontrar que hay múltiples exones esparcidos en diferentes zonas de éstos sospechosos de haber sido mal secuenciado.

Esto nos ha creado una incertidumbre diagnóstica, ya que en casi todos los resultados obtenidos hemos observado que un 1/3 del genoma ha sido mal secuenciado. Por ello, podemos decir que es una cantidad aproximadamente considerable aunque por otra parte sabemos con seguridad que el otro 2/3 restante está bien secuenciado a ciencia cierta.

Por una parte, al realizar la comparación entre las dos intersecciones realizadas hemos descubierto genes sospechosos de estar mal secuenciados

en la cobertura 100x que no encontramos en los pacientes con tasa de cobertura 50x.

La otra conclusión que hemos llegado es que al hacer una comparación estadística a nivel tanto de genes como de exones que codifican proteínas, podemos analizar que existe una mejora al hacer una cobertura de 100x con respecto al 50x.

Secuenciar con una profundidad de 100x es mucho mejor que con la de 50x, ya que la cobertura de encontrar genes codificantes mal secuenciados es mucho mayor, por lo que la rentabilidad en cuánto a coste monetario sería más beneficiosa secuenciar directamente con una profundidad mucho mayor (100x).

Es obvio que con MIST (Missing Sequencing Tools) solo hemos hurgado una pequeña parte de algo más grande. La búsqueda de regiones en el ADN que son sospechosos de haber sido mal secuenciados representa un logro científico y tecnológico para la sanidad.

Poder con dicha tecnología descubrir regiones causantes de originar una posible enfermedad, o averiguar si el individuo es más o menos susceptible de un determinado medicamento constituye un gran avance en el tratamiento médico, y supone tanto un beneficio económico bastante grande como preciso que los sistemas actuales.

Aun así, todavía quedan cosas por hacer en el flujo de trabajo de la localización de regiones mal secuenciadas en el ADN. A continuación exponemos unas cuantas:

- Optimización y mejora de la herramienta MIST de localización de regiones del ADN con baja frecuencia de lectura.

- Añadir la herramienta desarrollada al software general DNAnalytics como módulo adicional.

- Implementar, o encontrar nuevos algoritmos para la búsqueda de identificación de regiones mal secuenciadas en el ADN.

- Reducir el tiempo de procesamiento de los resultados obtenidos, adaptando la herramienta en máquinas más potentes.

- Encontrar y estudiar otras herramientas ya desarrolladas similares a nuestro programa, con el fin de realizar una comparación sobre éste.

8. Aportaciones.

En este capítulo hablaremos de las diferentes aportaciones que presenta nuestro Trabajo Fin de Grado. Con la implementación de la aplicación hemos avanzado considerablemente en muchos factores muy importantes.

En primer lugar, dicha aplicación (MIST) aporta saber qué zonas del exoma han sido mal secuenciadas. Por ello si, por ejemplo, un 90 % ha sido de las regiones han sido bien secuenciadas el programa detecta qué zonas del 10 % restante no lo han sido. Con esto nos podemos ahorrarnos tiempo y recursos (ya que reduce el número de exones a secuenciar por Sanger).

Por otra parte, en cuánto al factor científico-médico, buscar qué zonas han sido mal secuenciadas es de vital importancia ya que existen regiones que pueden causar enfermedades o incluso la muerte del paciente. Así que las detecciones de estos posibles genes candidatos, permite conocer qué variantes son sospechosas de causar riesgos de salud o nuevas variantes que no han sido descubiertas hasta ahora.

Además esta aplicación reduce, claramente, la posibilidad de error por parte de los empleados de la Unidad de Investigación, ya que no sólo se evalúan los resultados obtenidos sino que el experto puede realizar sus comparaciones con los datos suyos llegando así a conclusiones más óptimas.

En la actualidad, no existe este método informatizado y que cubra esta necesidad. Con el desarrollo de esta aplicación, conseguimos que los profesionales de dicho sector se encuentren con una herramienta sencilla, intuitiva y útil que agilizará y mejorará las investigaciones para este tipo de estudios.

9. Competencias Cubiertas.

Para la realización y evaluación de este trabajo fin de grado, se requiere de una serie de competencias, de las cuales enumeraremos y justificaremos las siguientes:

9.1 CII01

Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.

1. Se han diseñado, desarrollado, seleccionado y evaluado aplicaciones y sistemas informáticos ya que se ha tenido que hacer un estudio previamente sobre como sería la obtención de las regiones con baja frecuencia de lectura en el genoma de referencia.

2. Se ha desarrollado una aplicación con la capacidad de automatizar dicha búsqueda, incluyendo las diferentes pruebas tanto de fiabilidad y seguridad como las de calidad, considerando las diferentes normas vigentes.

3. En dicha memoria podemos ver que hemos incluido los apartados "Estado Actual y Objetivos" y "Recursos" (en el apartado de "Planificación"), con el fin de cubrir por completo dicha competencia.

9.2 CII02

Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.

1. Para la realización de este proyecto de dicha magnitud, tener una planificación es imprescindible para alcanzar los plazos establecidos del mismo.

2. Si se alcanza satisfactoriamente la presentación del proyecto, ratifica que la competencia ha sido cubierta.

3. Cabe destacar que dicho proyecto aunque esté en un proceso de investigación y experimentación, puede tener en un futuro un importante impacto y valor económico. Esto es así, porque dicha herramienta proporciona un ahorro tanto de tiempo como económico muy considerable para los investigadores.

9.3 CII04

Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.

Para una acertada utilización de la herramienta desarrollada se ha planteado las siguientes condiciones técnicas:

1. La implementación del módulo de software debe tener una serie de requisitos para llevar a cabo la instalación del mismo:

    - Fiabilidad.
    - Comportamiento constante.
    - Buena calidad del software.
    - Que sea seguro.

2. La intención es que dicha herramienta sea utilizada de forma global para los investigadores quieran hacer uso de ella. En un principio la aplicación se ha planteado en el idioma español, pero cabe la posibilidad de ampliarla a otro idioma si es necesario.

3. En el caso de que fuera necesario realizar diferentes correcciones a dicha herramienta, el proceso de pruebas no debería de afectar a la actividad del mismo. Es preciso analizar y comprobar el funcionamiento antes de efectuar la difusión del producto.

9.4 CII08

Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

1. Como hemos dicho en la anterior competencia, la aplicación debe contener una serie de requisitos para que sea robusta, segura y eficiente.

2. Para la implementación de la aplicación, es recomendable que se tome como paradigma la orientación a objetos y que dicho lenguaje sea también imperativo.

3. La portabilidad es técnicamente difícil de lograr, por ello se ha utilizado como lenguaje de desarrollo Java ya que actúa independientemente de la plataforma que estemos trabajando.

9.5 CII018

Conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.

Actualmente las leyes que están puestas en diferentes países guiadas a proteger la utilización desmedida o/e ilegal de la información obtenida y tratado en equipos informáticos.

Desde hace varias años, la mayoría de los países en el ámbito europeo e internacional, han intentado crear/ejecutar leyes relacionados con el acceso ilegal a los sistemas informáticos o el mantenimiento indebido de estos accesos, la propagación de virus u otros métodos para adquirir información ilegal.

Todos estos enfoques en países occidentales son muy parecidos a los europeos, ya que el objetivo también es proteger la información que puede ser adquirida de forma ilícita mediante diferentes métodos como la comunicación segura o que la transferencia información sea de lo más confidencial posible.

9.6 TFG01

Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.

Esta competencia ha sido totalmente cubierta en la presente memoria sobre este Trabajo Fin de Grado, en el cual también se ha desarrollado la aplicación a la que se hace mención.

Para ello hemos hecho previamente un estudio sobre el desarrollo del mismo, desde la planificación e implementación de la herramienta hasta los resultados obtenidos al realizar la ejecución de los diferentes algoritmos desarrollados.

Todos estos aspectos pertenecen, evidentemente, a ámbitos de las tecnologías específicas de la Ingeniería en Informática.

10. Normativa y Legislación.

A continuación se incluirá la legislación vigente que afecta a este Trabajo Fin de Grado en materia de seguridad informática:

10.1 Ley de Protección de datos.

En este apartado explicaremos las razones por las cuales esta ley es importante en nuestro proyecto.

Una de las principales razones por la que se utiliza esta ley es que la aplicación realiza sus procesos a partir de unos ficheros relacionados con el alineamiento del ADN. Estos ficheros vienen originalmente de muestras de sangre extraída en determinados pacientes, por lo que es de vital importancia proteger cualquier información concerniente a dichas personas identificadas.

Además durante la implementación de la aplicación, hacemos operaciones y procedimientos técnicos los cuales nos permiten la recogida, elaboración y modificación de los datos.

En nuestro caso, estos datos de carácter personal recogidos en pacientes, sólo podrán ser usados para finalidades relacionadas con las investigaciones biomédicas. Por otro lado, el tratamiento de dichos datos requerirá del consentimiento inequívoco del afectado, salvo que la ley disponga otra cosa.

10.2 Leyes sobre Seguridad.

En este apartado hablaremos sobre los diferentes aspectos relacionados con la seguridad que necesita la aplicación desarrollada en este Trabajo Fin de Grado. Para ello enumeramos las siguientes consideraciones a tener en cuenta:

1. La información de los archivos que vayamos a tratar relacionado con determinados pacientes sean de total confidencialidad.

2. Con la posibilidad de que la información en dicho archivos sufra peligro de confidencialidad e integridad en los datos, los equipos informáticos donde vayamos a instalar la aplicación deberán tener un sistema de protección.

3. Generalmente las amenazas internas pueden ser más serias que las externas, ya aquellas personas dentro la investigación conocen la red y saben cómo es su funcionamiento, ubicación de la información, datos de interés, etc. Por ello es necesario realizar copias de seguridad, e incluso, si es necesario sistemas de respaldo remoto que permitan mantener la información en dos ubicaciones de forma asíncrona.

4. Aunque en nuestros equipos informáticos tienen el sistema operativo Linux, los virus son uno de los medios más tradicionales de ataque a los sistemas y a la información que los sostienen. Para poder evitar su contagio se deben vigilar los equipos y los medios de accesos a ellos, principalmente la red.

11. Manual de Usuario y Software.

En este apartado exponemos un sencillo manual de usuario sobre la aplicación desarrollada para los usuarios correspondientes tanto al personal clínico como a los usuarios externos. Para facilitar la interpretación del funcionamiento del mismo, mostraremos varios ejemplos mediante capturas de pantalla.

11.1 Acceso a la aplicación.

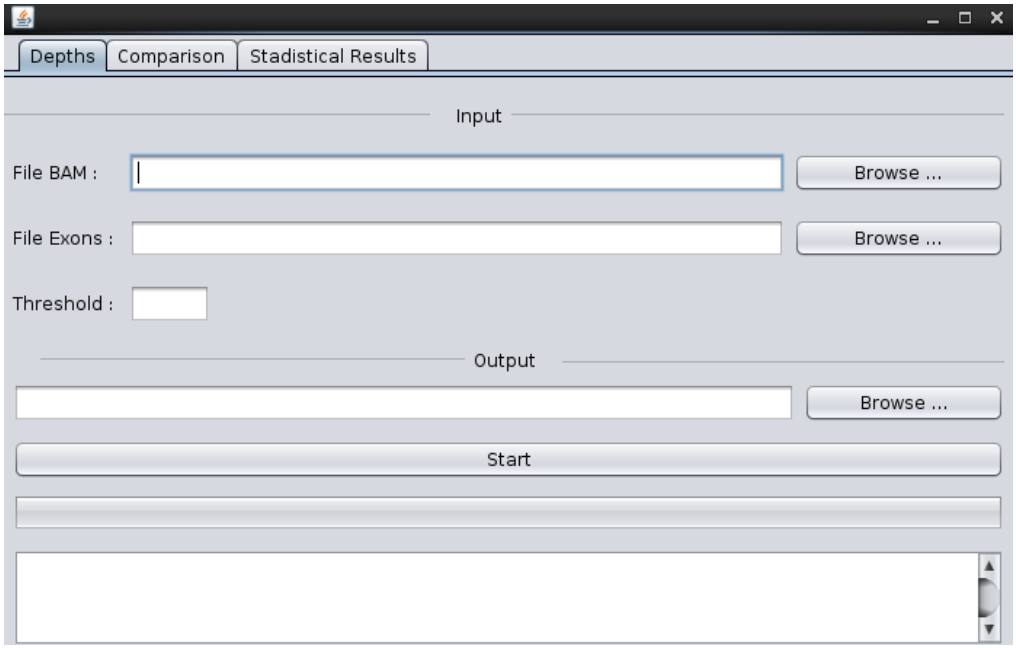
1. Haga clic dos veces sobre el archivo llamado "MIST" o selecciónelo y pulse la tecla "Enter".

2. A continuación se abrirá la interfaz del programa en el cual está compuesta por tres pestañas superiores: "Depths", "Comparison" y "Statistical Results".

11.2 Proceso "Depths".

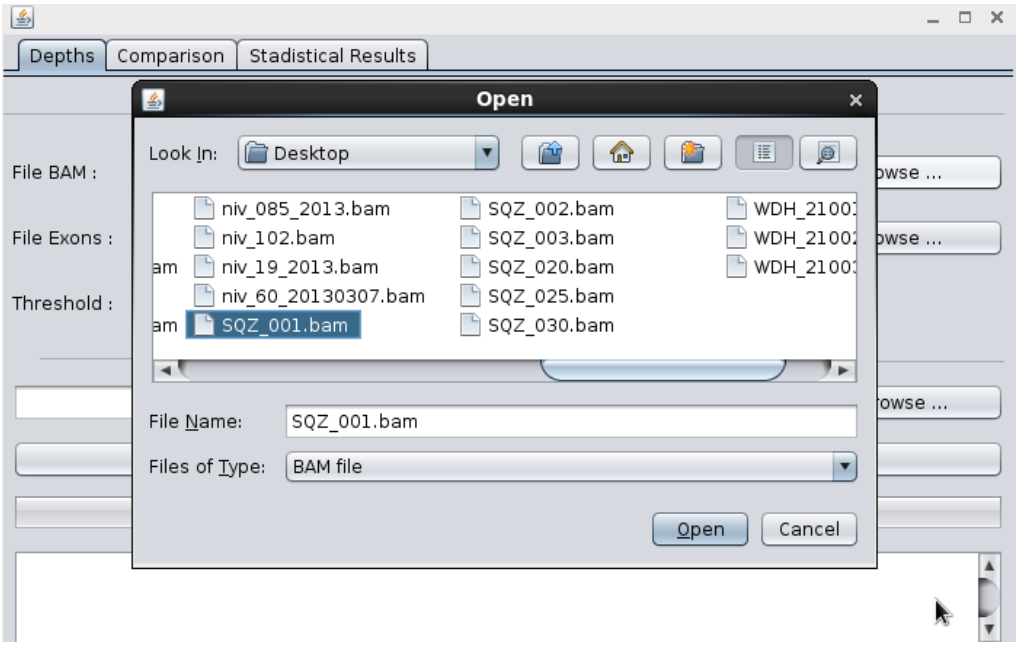
El proceso que realiza es la localización de regiones del ADN con lecturas por debajo del límite que hemos introducido en el campo numérico "Threshold".

1. Hacer clic en la pestaña "Depths". Se ofrecerá la interfaz:



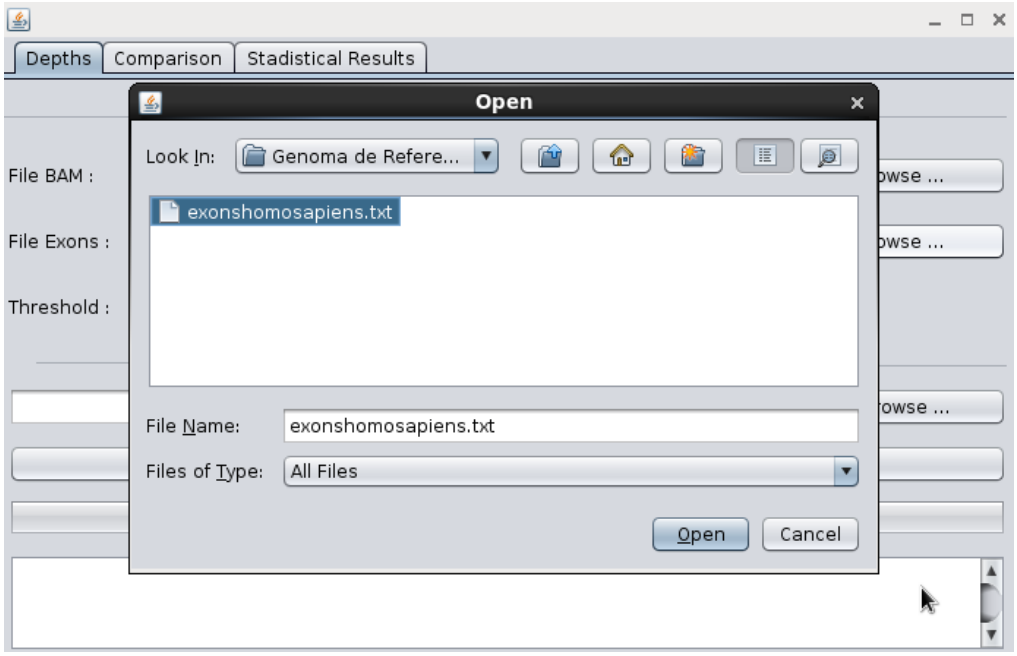
2. Dicha interfaz está compuesta por dos secciones: “Input” y “Output”.

3. Hacer clic en el botón “Browse…” correspondiente a “File BAM” para elegir el fichero de alineamiento del ADN.

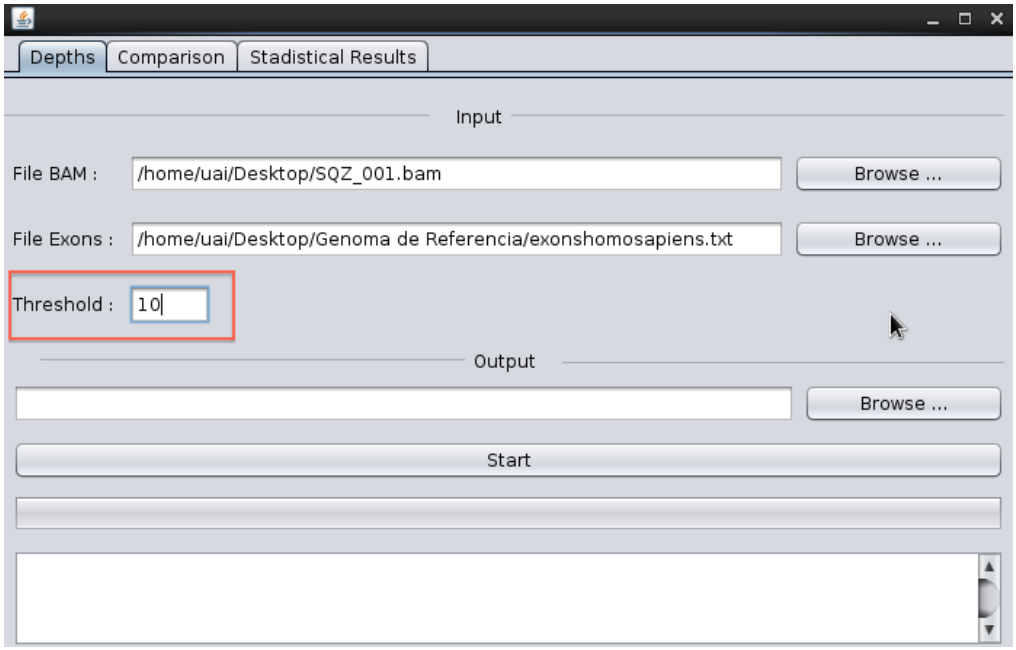


4. MIST sólo permite elegir ficheros con extensión “.BAM”.

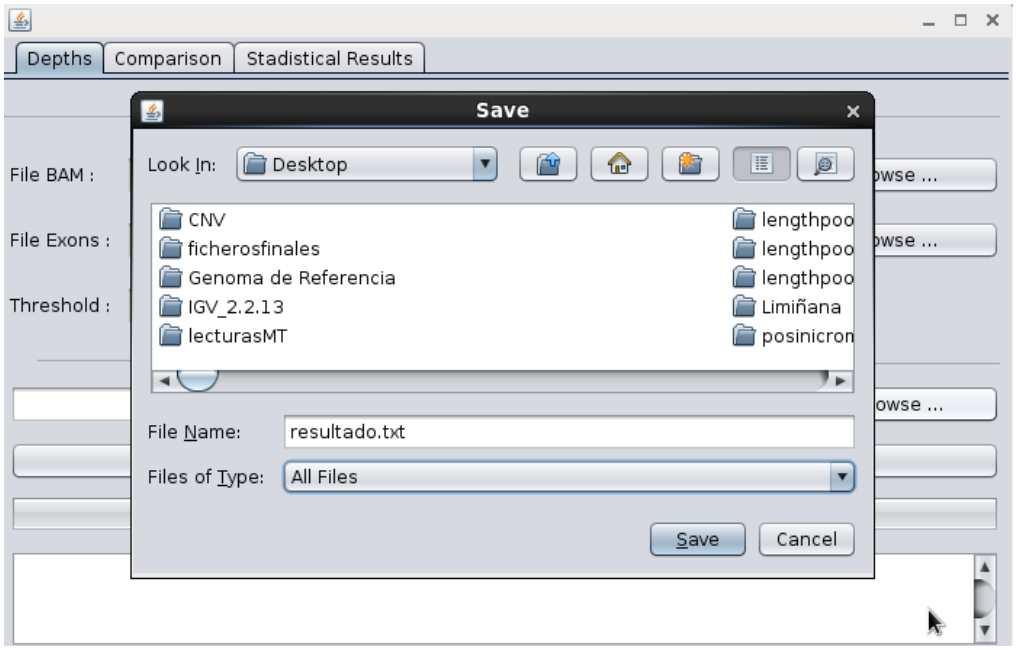
5. Hace clic en el botón “Browse…” correspondiente al “File Exons” (Base de datos de exones).



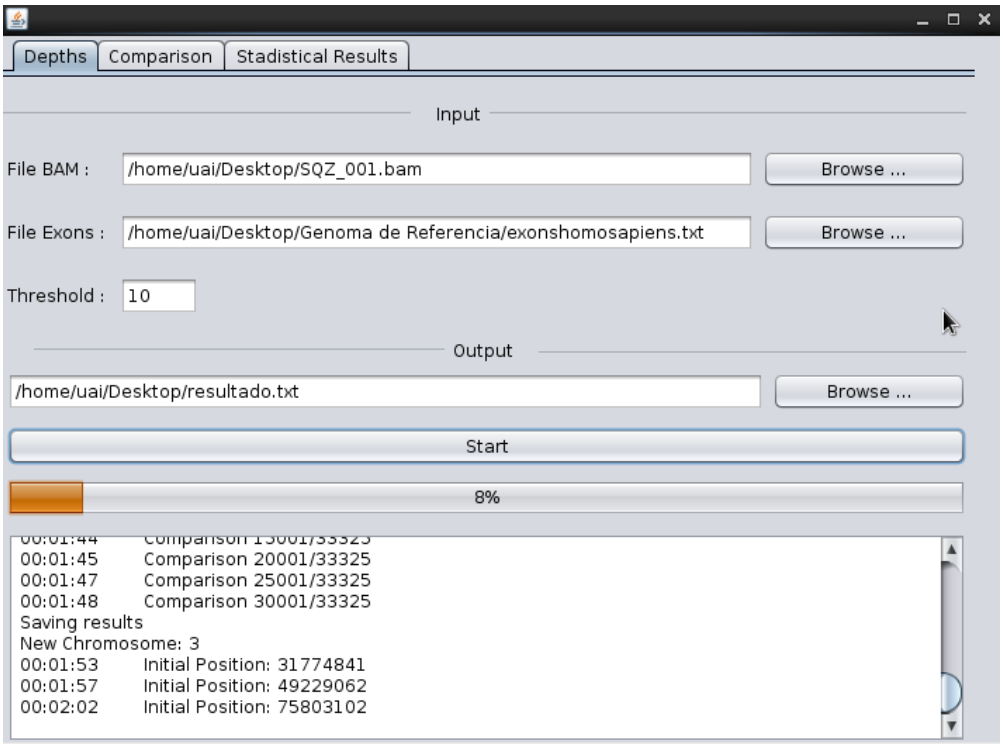
6. En el campo “Threshold” introducir el umbral para determinar el límite de lecturas.



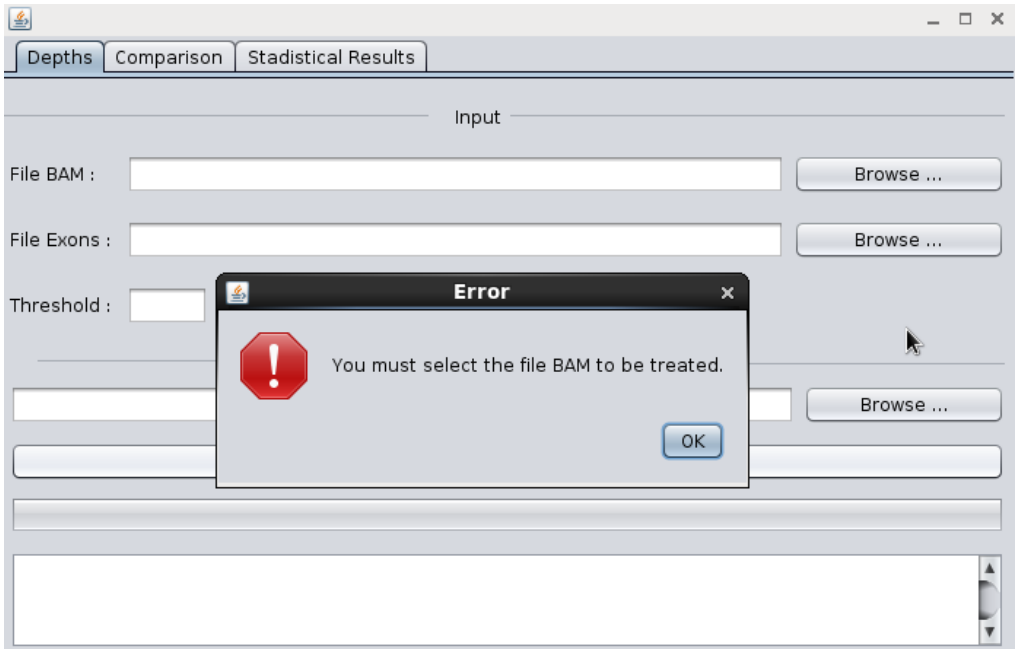
7. Hacer clic en el botón “Browse…” correspondiente a la sección “Output”, para seleccionar la ubicación del archivo dónde queremos general nuestros resultados. El archivo lo podemos guardar con formato del tipo “.txt” o “.tsv” (Tab Separated Values).



8. Hacer clic en “Start” para ejecutar el proceso del programa.



9. El programa mostrará un mensaje de error al pulsar el botón “Start” en el caso de que no se haya realizado ninguna de las anteriores instrucciones.

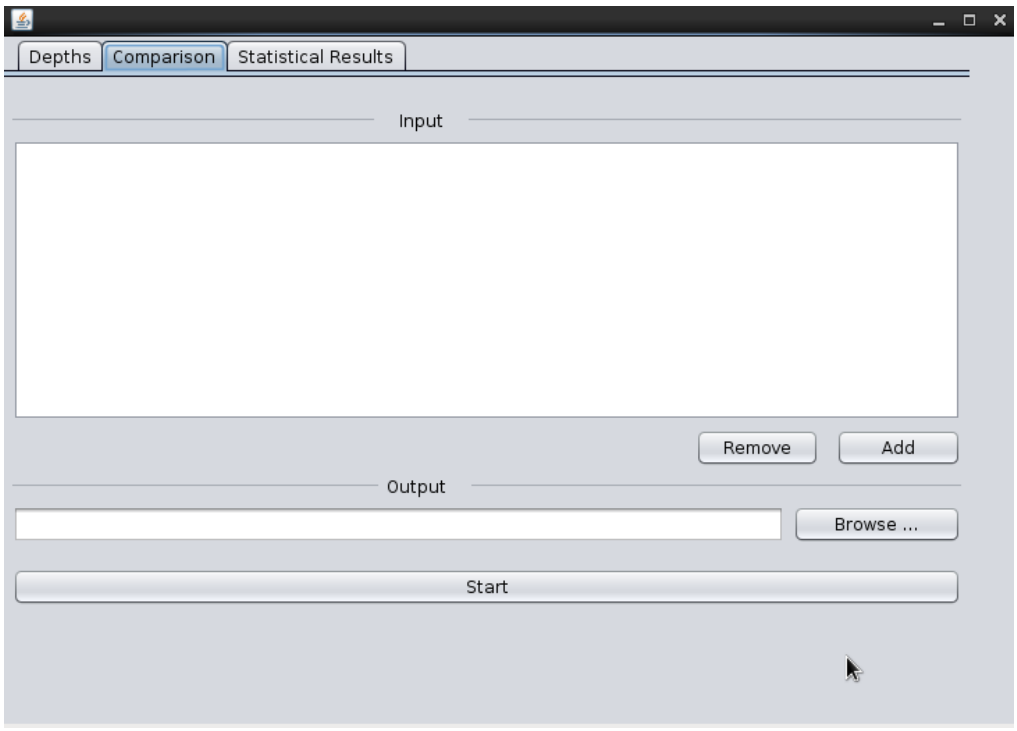


10. Una vez que el proceso de la aplicación haya terminado, podemos volver a cerrar la aplicación o directamente pasar a otra pestaña.

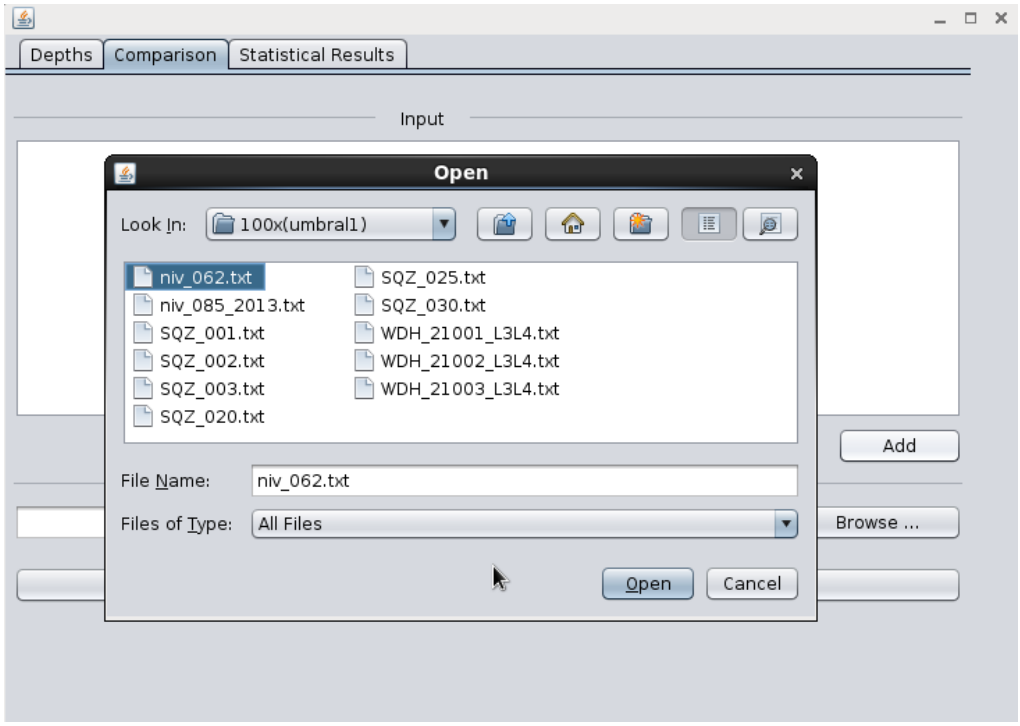
11.3 Proceso “Comparison”.

El proceso que realiza es la intersección de las regiones con baja frecuencia de lectura de todos los pacientes.

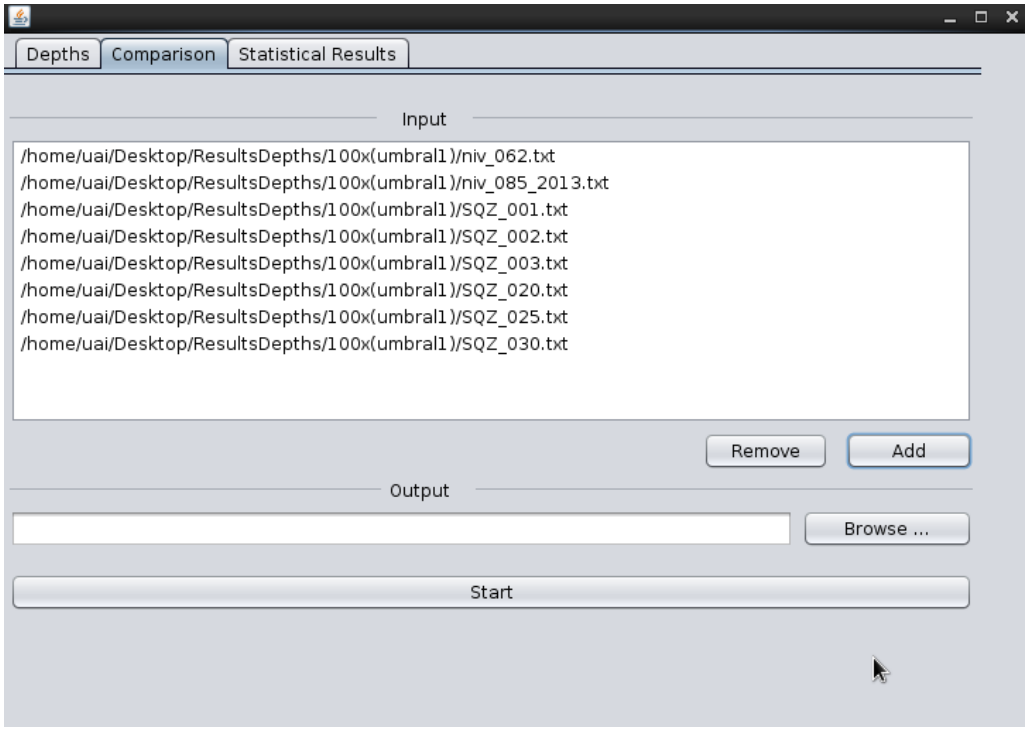
1. Hacer clic en la pestaña “Comparison”. Se mostrará el siguiente panel, dividido en las secciones “Input” y “Ouput”:



2. Hacer clic en el botón “Add” para añadir los ficheros de entrada. Sólo se pueden añadir ficheros generados en la etapa “Depths”.

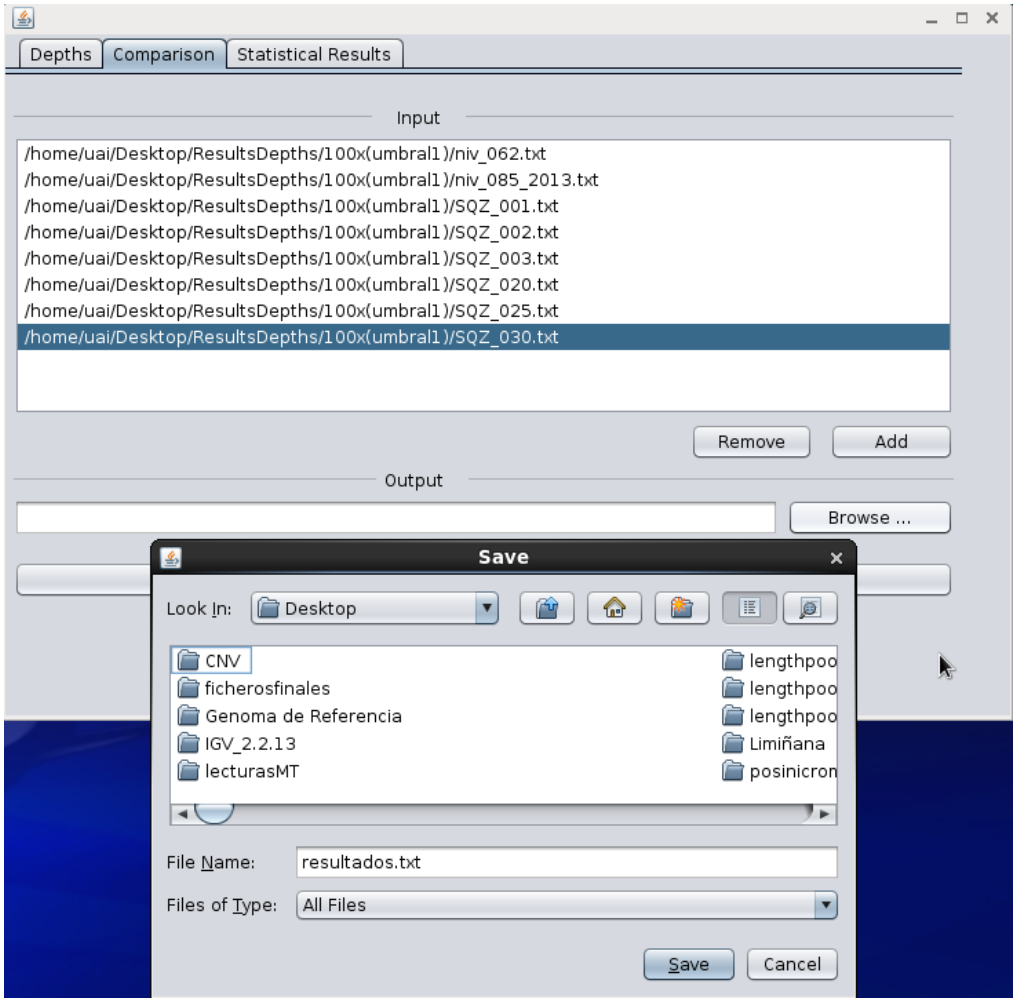


1. La siguiente interfaz nos muestra las rutas de cada uno de los ficheros que hemos añadido para realizar la comparación.



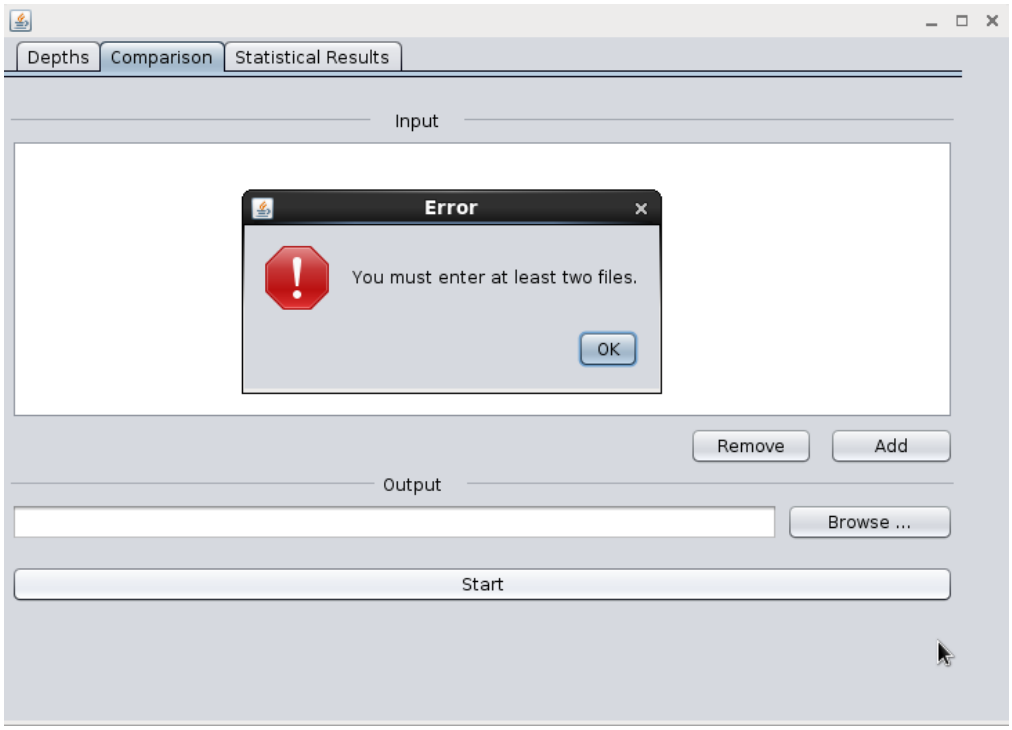
2. Hace clic botón “Remove” para eliminar un fichero añadido previamente.

3. En la sección “Output” hacer clic en el botón “Browse…”, para seleccionar la ubicación del archivo dónde queremos guardar nuestros resultados. El archivo lo podemos generar del tipo “.txt” o “.tsv” (Tab Separated Values).



4. Hacer clic en “Start” para ejecutar el proceso del programa.

5. El programa mostrará un mensaje de error al pulsar el botón “Start” en el caso de que no se haya realizado ninguna de las anteriores instrucciones.

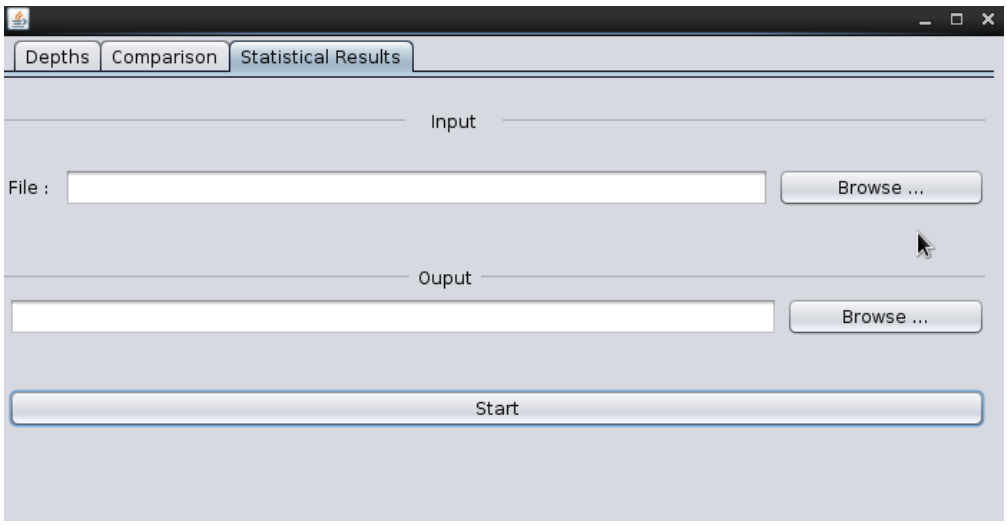


6. Una vez que el proceso de la aplicación haya terminado, podemos volver a cerrar la aplicación o directamente pasar a otra pestaña.

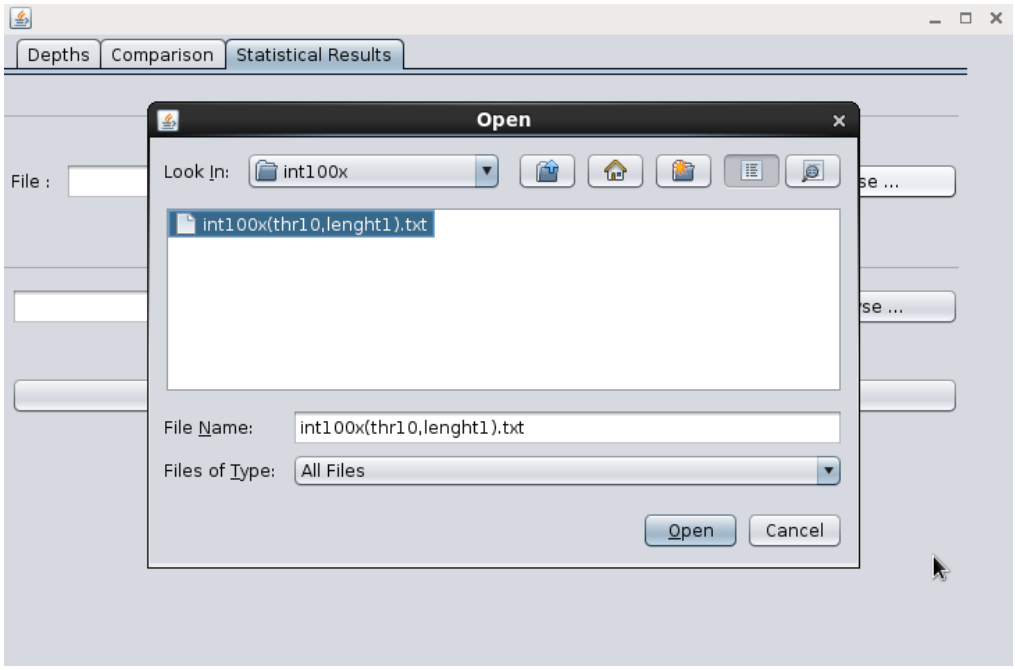
11.4 Proceso “Statitiscal Results”.

El proceso que realiza es la obtención de resultados estadísticos tanto para la intersección de regiones pobres entre pacientes como para cada uno de forma individual.

1. Hacer cli en la pestaña “Statistical Results”. Se muestra la interfaz, dividida en las secciones “Input” y “Output”.



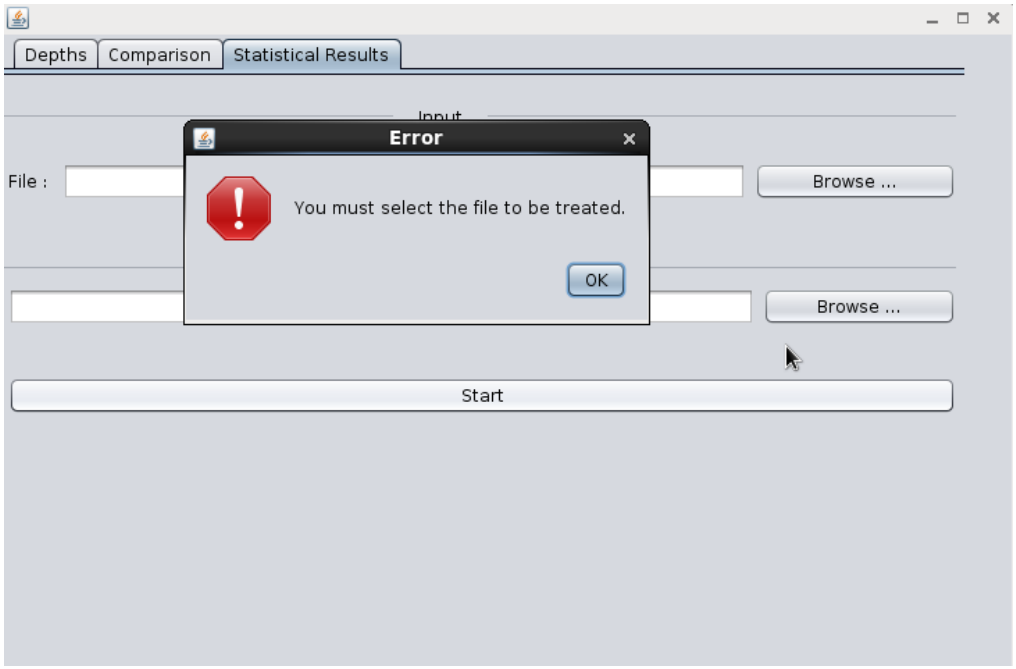
2. Hacer clic en el botón “Browse…” correspondiente al campo “File” para elegir el fichero al que queremos realizar el tratamiento.



3. En la sección “Output” hacer clic en el botón “Browse…”, para seleccionar la ubicación del archivo dónde queremos guardar nuestros resultados. El archivo lo podemos generar del tipo “.txt” o “.tsv” (Tab Separated Values).

4. Hacer clic en el botón “Start” para ejecutar el proceso del programa.

5. El programa mostrará un mensaje de error al pulsar el botón “Start” en el caso de que no se haya realizado ninguna de las anteriores instrucciones.



6. Una vez que el proceso de la aplicación haya terminado, podemos volver a cerrar la aplicación o directamente pasar a otra pestaña.

12. Fuentes de información.

[1] U.S. DOE Human Genome Project.

[2] Peter J. Russell. Genetics. 4ª. Ed. Harper Collins, 1996.

[3] Miller SA, Dykes DD y Polesky HF. "A simple salting out procedure for extracting DNA from human nucleated cells". En: *Nucleic Acids Research* 16.3 (feb. de 1988), pág. 1215.

[4] About BGI. URL: www.genomics.cn

[5] Illumina HiSeq 2000. URL: http://www.genomics.cn/en/navigation/show_navigation?nid=4145

[6] An Introduction to Next-Generation Sequencing Technology. URL: http://res.illumina.com/documents/products/illumina_sequencing_introduction.pdf

[7] Miller SA, Dykes DD y Polesky HF. "A simple salting out procedure for extracting DNA from human nucleated cells". En: *Nucleic Acids Research* 16.3 (feb. de 1988), pág. 1215.

[8] Burrows-Wheeler Aligner. URL: http://bio-bwa.sourceforge.net/

[9] McKenna A y col. "The Genome Analysis Toolkit: a Ma- pReduce framework for analyzing next-generation DNA sequencing data". En: *Genome Res.* 20.9 (sep. de 2010), pa´gs. 1297-1303.

[10] Picard. URL: http://picard.sourceforge.net/

[11] Working with BAM Files. URL: http://www.ncbi.nlm.nih.gov/tools/gbench/tutorial6/

[12] BamView. URL: http://bamview.sourceforge.net/

[13] Samtools. URL: http://samtools.sourceforge.net/

[14] Li H. et al. **The Sequence Alignment/Map format and SAMtools.** Bioinformatics. 2009 Aug 15; 25(16):2078-9.

[15] *The Variant Call Format and VCFtools*, Petr Danecek, Adam Auton, Goncalo Abecasis, Cornelis A. Albers, Eric Banks, Mark A. DePristo, Robert Handsaker, Gerton Lunter, Gabor Marth, Stephen T. Sherry, Gilean McVean, Richard Durbin and 1000 Genomes Project Analysis Group, **Bioinformatics**, 2011.

[16] Danecek P. et al. **The variant call format and VCFtools**. Bioinformatics. 2011 Aug 1; 27(15):2156-8.

[17] Fernando García-Alcalde, Konstantin Okonechnikov, José Carbonell, Luis M. Cruz, Stefan Götz, Sonia Tarazona, Joaquín Dopazo, Thomas F. Meyer, and Ana Conesa **"Qualimap: evaluating next-generation sequencing alignment data."** Bioinformatics 28, no. 20 (2012): 2678-2679.

[18] Xosé M. Fernández-Suárez and Michael K. Schuster
Using the Ensembl Genome Server to Browse Genomic Sequence Data.
UNIT 1.15 in *Current Protocols in Bioinformatics*, Jun 2010.
www.ncbi.nlm.nih.gov/pubmed/20521244

[19] Blast. Altschul S.F., Gish W., Millwe W., Myers E.W. and Lipman D.J. (1990) Basic local alignment search tool.
J. Mol. Biol. **215**: 403-410.

[20] Paul Flicek, M. Ridwan Amode, Daniel Barrell, Kathryn Beal, Konstantinos Billis, Simon Brent, Denise Carvalho-Silva, Peter Clapham, Guy Coates, Stephen Fitzgerald, Laurent Gil, Carlos García Girón, Leo Gordon, Thibaut Hourlier, Sarah Hunt, Nathan Johnson, Thomas Juettemann, Andreas K. Kähäri, Stephen Keenan, Eugene Kulesha, Fergal J. Martin, Thomas Maurel, William M. McLaren, Daniel N. Murphy, Rishi Nag, Bert Overduin, Miguel Pignatelli, Bethan Pritchard, Emily Pritchard, Harpreet S. Riat, Magali Ruffier, Daniel Sheppard, Kieron Taylor, Anja Thormann, Stephen J. Trevanion, Alessandro Vullo, Steven P. Wilder, Mark Wilson, Amonida Zadissa, Bronwen L. Aken, Ewan Birney, Fiona Cunningham, Jennifer Harrow, Javier Herrero, Tim J.P. Hubbard, Rhoda Kinsella, Matthieu Muffato, Anne Parker, Giulietta Spudich, Andy Yates, Daniel R. Zerbino, and Stephen M.J. Searle **Ensembl 2014**
Nucleic Acids Research 2014 42 Database issue:D749-D755.
doi: 10.1093/nar/gkt1196

13. Anexo 1: Formato de ficheros.

FASTQ

En la siguiente Figura 1.1 observamos como está estructurado los primeras líneas de un fichero ".fastq". La primera línea es el identificador (comenzando por @), después del cual viene la secuencia. La línea que comienza con + suele contener otra vez el mismo identificador, aunque se puede omitir para ahorrar espacio en disco. Al final viene una serie de caracteres que representan la calidad, un carácter por cada letra de la secuencia.

Para que la cadena de valores de calidad tenga la misma longitud que la secuencia, se codifica el número convirtiéndolo en un carácter ascii. Para evitar caracteres ascii que no se pueden desplegar en pantalla, se suele agregar 33 al número de calidad antes de codificarlo.

```
@IL21_4392:7:1:1487:3943/1
AATTGCATCTCGTATGCCGTCTTCTGNTTGAANNNNA
+
GFGEACFF=F@<FBEGA;FGFF38FB)C@E88&&&&1
```

Figura 1.1

VCF

En las siguientes Figuras 1.2 y 1.3 podemos observar como está estructurado un archivo de variantes de formato ".VCF" [15] :

En esta primera imagen observamos las primeras líneas del archivo ".vcf", en el cual nos expone una información (meta-información) de cómo será el contenido y la estructura que tendrá dicho fichero. Esta meta-información es incluida después de la cadena "##" y debe ser formada por "clave = valor".

En estas primeras líneas encontramos los siguientes campos asociados a sus valores:

- **Fileformat**: Este campo siempre es requerido y debe ser la primera línea del fichero. Informa del número de la versión del fichero de las variantes.
- **Info**: Este campo puede ser de varios tipos: entero, real, flag, carácter y cadena. También contiene un campo "Number" que es de tipo entero en el cuál se describe el número de valores que puede ser incluido con el campo "Info".
- **Filter**: Este campo informa de los filtros que se han aplicado a los datos.
- **Format**: Este campo indica los diferentes campos del Genotipo que se van a exponer.
- **Contig**: Es un conjunto de lecturas contiguas que están relacionadas entre sí mediante la superposición de secuencias. Al igual que con las secuencias cromosómicas es muy recomendable que la cabecera incluya etiquetas que describen los "contigs" que hacen referencia al fichero ".vcf". El formato es idéntico a la de una secuencia de referencia, pero con una etiqueta URL adicional que indica dónde puede ser encontrado la secuencia.

En la Figura 5.6 vemos como está estructurado las líneas de los datos del fichero. Está formado por 8 campos (además de 2 campos más que son opcionales), en el cuál todas sus líneas están delimitados por tabuladores. Los valores perdidos están especificados por un ".".

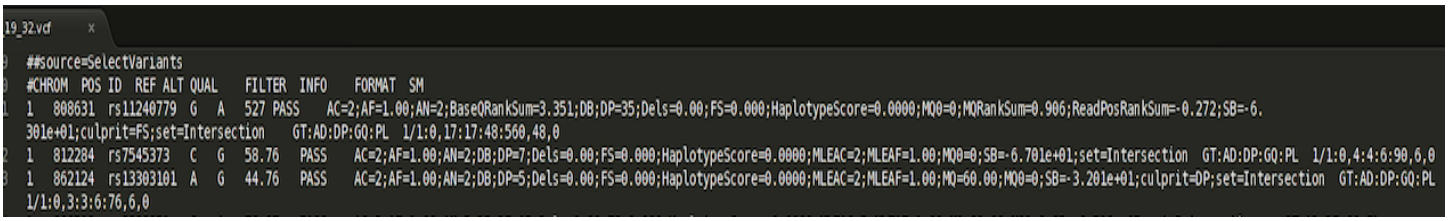


Figura 1.3

A continuación explicaremos cada uno de los campos que está estructurado dichas líneas:

- **CHROM (cromosoma)**: Contiene un identificador a partir del genoma de referencia o una cadena ID entre corchetes que apunta a un "contig" en el archivo de ensamblado. Todas las entradas para un específico cromosoma deben formar un bloque contiguo en el VCF.

- **POS**: La posición de referencia, con la primera base que tiene la posición 1. Las posiciones son ordenadas numéricamente, en orden creciente, dentro de cada secuencia de referencia CHROM. Se permite tener múltiples registros con el mismo punto de venta. Los telómeros[xvii] son indicados usando las posiciones 0 o N+, dónde N es la longitud del correspondiente cromosoma o contig.

- **ID**: Identificador de la variante. Si se trata de una variante de tipo "dbSNP" se incita a utilizar el/los número/s rs. Sin identificador debería estar presente en más de un registro de datos. Si no existiera ninguna identificación posible, se le asigna el valor perdido por defecto.

- **REF (base de referencia)**: Cada base debe ser uno de A, C, G, T, N. Se permiten múltiples bases. El valor en el campo POS se refiere a la posición de la primera base de la cadena. Para inserciones y deleciones en los que tanto REF o uno de los alelos de ALT deberían de otro modo ser vacíos/nulos, las cadenas REF y ALT deben incluir la base antes del evento (debe reflejarse en el

campo POS), a menos que el evento se produzca en la posición 1 en el contig en cuyo caso se debe incluir la base después del evento.

- **ALT (base alternativa)**: Las opciones son cadenas de bases formadas por las bases A, C, G, T, N o un identificador ID o una cadena alternativa llamada "breakend". Si no hay alelos alternativos, entonces se debe usar el valor perdido.

- **QUAL**: En campo se utiliza para asignar un valor de calidad con respecto al campo ALT. Las puntuaciones altas indican llamadas de alta confianza.

- **FILTER**: Este campo tendrá el valor "PASS" si la posición ha pasado todos los filtros, es decir, si se ha realizado una llamada a esta posición. De lo contrario, si no se ha pasado todos los filtros, se hace una lista separados por ";" de los filtros que han sido frustrados.

- **INFO**: En este campo se añade información adicional si se requiere. Está codificado como una serie separada por ";" con el formato "clave = valor".

14. Anexo 2: Herramientas Adicionales.

Existen herramientas adicionales para ser utilizadas en el tratamiento de ficheros específicos, aunque no están relacionadas con alguna fase en particular en el flujo de análisis del ADN.

Una de las herramientas que son imprescindibles para el tratamiento con el ADN es el conjunto de paquetes Samtools/Picard. Estos dos paquetes se encargan de la manipulación de los ficheros en formato SAM y su correspondiente binario BAM, referente al alineamiento de secuencias.

Samtools

Es un conjunto de utilidades que manipulan los alineamientos en el formato de BAM. Importa y exporta a partir del formato SAM (Sequence Alignment / Map), la no clasificación, la fusión y la indexación, y permite recuperar lecturas en cualquier región con rapidez. [14]

Samtools está diseñado para trabajar en "stream" (lectura continua). Considera un archivo de entrada "-" como la entrada estándar (stdin) y un archivo de salida "-" como la salida estándar (stdout). Varios comandos de se pueden combinar con "pipes"[xviii] de Unix. Samtools siempre alerta de las salidas y los mensajes de error a la salida de error estándar (stderr).

También es capaz de abrir un BAM (no SAM) de archivos en un FTP remoto o servidor HTTP si el nombre de archivo de BAM comienza con "ftp://" o "http://". Samtools comprueba el directorio de trabajo actual para el archivo de índice y se descargará el índice sobre la ausencia. No recupera el archivo de alineación entero, a menos que le pidamos que lo haga.

Una vez instalada la herramienta, podremos visualizar este tipo de ficheros de alineamiento de secuencias. Dicha herramienta posee diferentes opciones que vamos a ver a continuación:

- *samtools view*: Extrae o imprime todo o parte de los alineamientos en el formato SAM o BAM. Si no se especifica ninguna región, se imprimirán todas las regiones; de lo contrario sólo alineamientos superpuestos de las regiones especificadas serán mostradas. Un alineamiento se puede administrar varias veces si es la superposición de varias regiones.

- *samtools sort*: Ordena alineamientos por las coordenadas del extremo izquierdo.

- *samtools index*: Indexa el alineamiento ordenado para acceso aleatorio rápido. Genera un fichero en formato ".BAI".

- *samtools idxstats*: Recupera e imprime estadísticas en el fichero índice. La salida está delimitado por tabuladores con cada línea que consiste en nombre de la secuencia de referencia, longitud de la secuencia, lecturas mapeadas y lecturas no mapeadas.

- *samtools merge*: Combina múltiples alineamientos ordenados. La listas de referencias de cabecera de todos los archivos de BAM de entrada, y los @SQ cabeceras, si los hay, todos deben referirse al mismo conjunto de secuencias de referencia.

- *samtools faidx*: Indexa la secuencia de referencia en el formato FASTA o extrae una subsecuencia de la secuencia de referencia indexada. Si no se especifica ninguna región, "faidx" indexará el archivo y crea <ref.fasta> en el disco. Si las regiones son especificadas, las subsecuencias se recuperarán y se imprimen por la salida estándar en el formato FASTA. El archivo de entrada puede ser comprimido en el formato RAZF.

- *samtools mpileup*: Genera BCF o "pileup" [xix] para uno varios archivos de BAM. Los registros de alineamiento son agrupados por identificadores de muestra en @RG líneas de cabecera. Si los identificadores se encuentran ausentes, cada archivo de entrada se considera una muestra.

- *Samtools tviews*: Visor de alineamiento de texto (basado en la biblioteca ncurses).

Otra herramienta que explicaremos a continuación se llama "VCFTools" diseñado para la manipulación de ficheros VCF.

VCFTools

Es un paquete de programas diseñados para trabajar con archivos VCF, como los generados por el Proyecto 1000 genomas. El objetivo de

VCFTools es proporcionar métodos de fácil acceso para trabajar con datos de variación genética complejos en la clase de archivos VCF. [16]

Este conjunto de herramientas se puede utilizar para realizar las siguientes operaciones sobre archivos VCF:

- Filtrar variantes específicas.
- Comparar archivos.
- Resumir variantes.
- Convertir a diferentes tipos de archivos.
- Validar y combinar archivos.
- Crear intersecciones y subconjuntos de variantes.
- VCFtools consta de dos partes: un módulo perl y un ejecutable binario. El módulo de perl es una API general de Perl para manipular archivos VCF, mientras que el binario ejecutable proporciona rutinas de análisis generales.

Picard

Picard abarca utilidades de líneas de código basados en Java que manipulan archivos SAM y una API de Java (HTSJDK) para la creación de nuevos programas que leen y escriben archivos SAM. Tanto el formato de texto y formato binario SAM (BAM) son compatibles.

GATK

El kit de herramientas de análisis del genoma o GATK es un paquete de software desarrollado en el Instituto Broad capaz de analizar los datos de secuenciación. El conjunto de herramientas ofrece una amplia variedad de herramientas, con un enfoque principal en la localización de variantes y genotipado, así como una fuerte énfasis en la garantía de calidad de los datos. Su arquitectura robusta, potente motor de procesamiento y las características de computación de alto rendimiento hacen que sea capaz de asumir proyectos de cualquier tamaño. Hace uso de otra utilidad llamada "Queue" para realizar el análisis de flujo de trabajo completo de un modo sin supervisión, y mediante la aplicación de una secuencia de herramientas predefinida.

Qualimap

Es una aplicación independiente de la plataforma, escrito en Java y R que proporciona una interfaz gráfica de usuario (GUI) y una interfaz de línea de comandos para facilitar el control de calidad de los datos de secuenciación de alineación.

Qualimap examina los datos de alineamientos de secuencias en archivos SAM/BAM de acuerdo con las características de las lecturas asignadas y proporciona una visión general de los datos que ayuda a los sesgos de detectar en la secuencia y/o mapeo de los datos y facilita la toma de decisiones para su posterior análisis. [17]

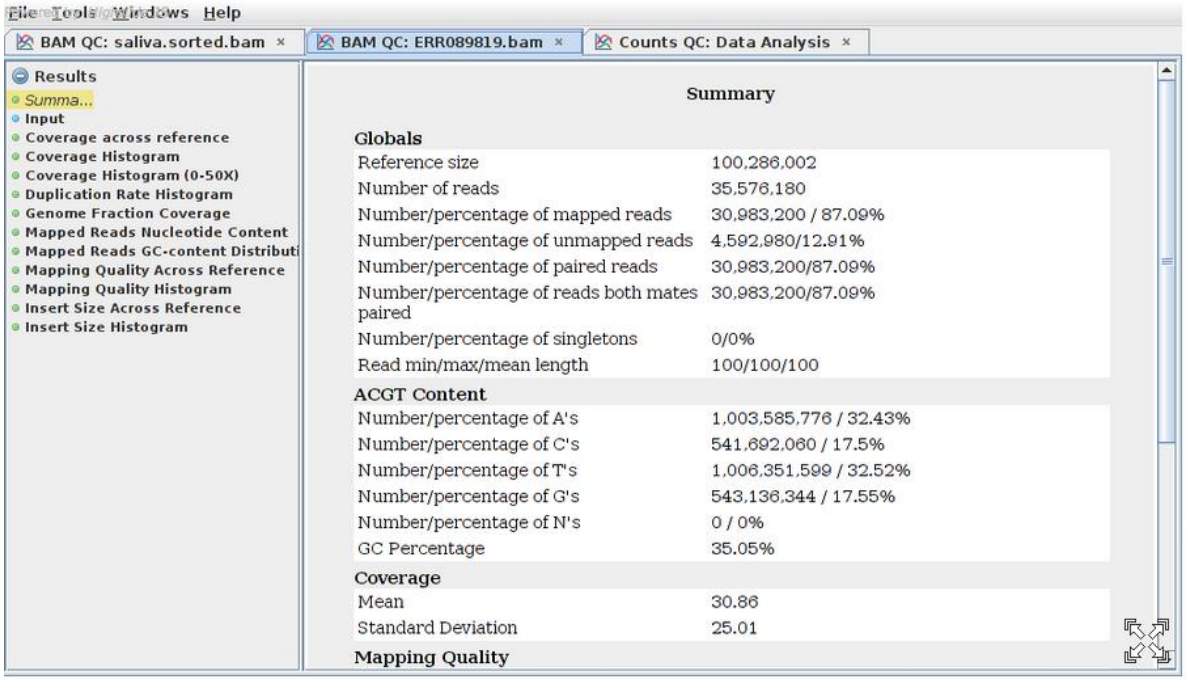
1. Examina los datos de alineación de secuenciación de acuerdo con las características de las lecturas mapeadas y sus propiedades genómicas.

2. Proporciona una visión general de los datos que ayuda a detectar sesgos en la secuenciación y/o mapeo de los datos y facilita para su posterior análisis.

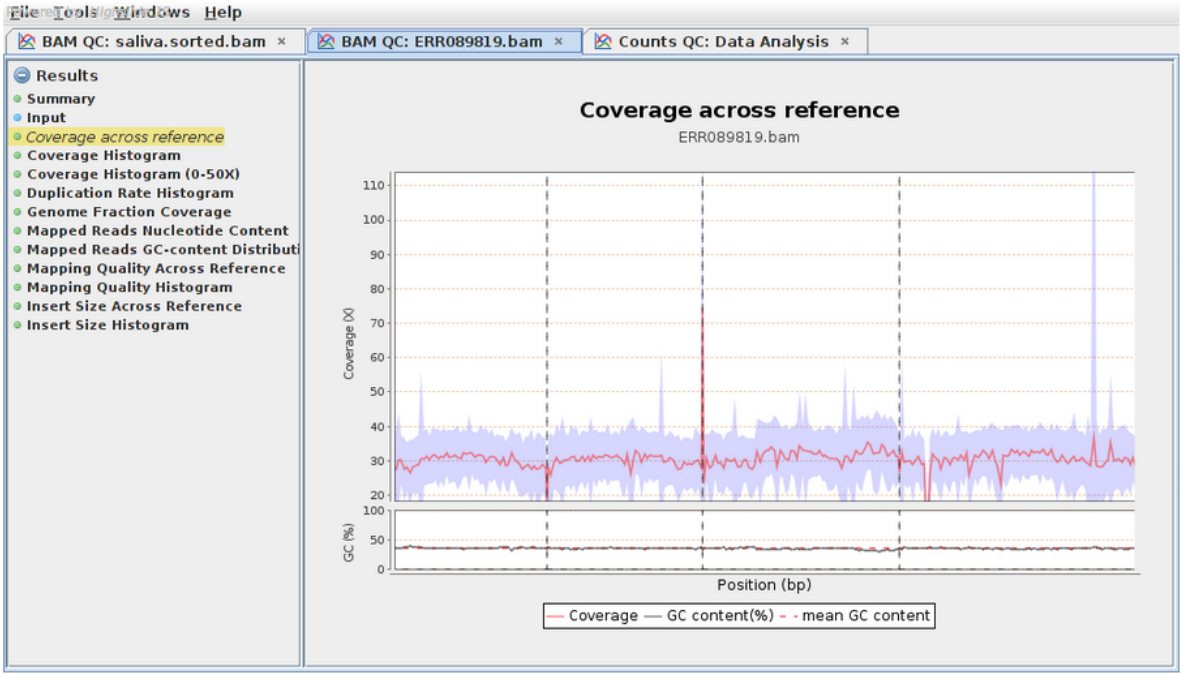
Las principales características que ofrece Qualimap son:

- Análisis rápido en todo el genoma de referencia de la cobertura de la cartografía y la distribución de nucleótidos.
- Resumen fácil de interpretar de las principales propiedades de los datos de la alineación.
- Análisis de las lecturas mapeadas dentro/fuera de las regiones definidas en una referencia de la anotación.
- El análisis de la adecuación de la profundidad de la secuenciación en los experimentos de RNA-seq.
- Agrupación de los perfiles epigenómicos[xx].

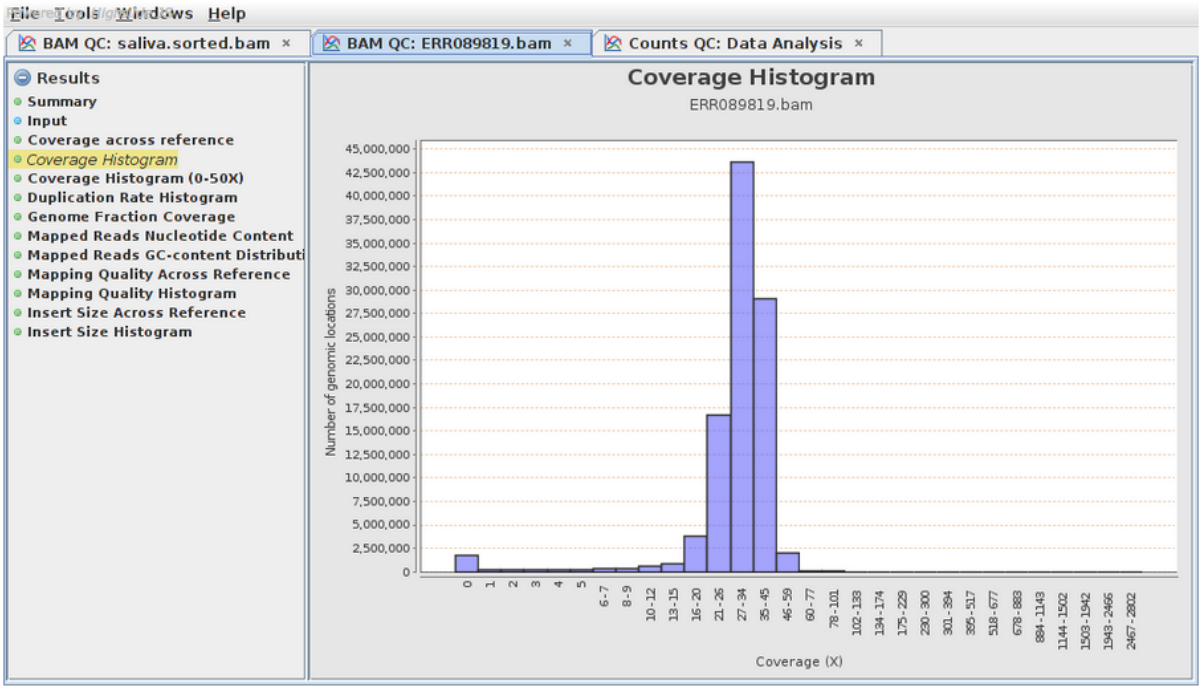
A continuación mostraremos una serie de imágenes del programa:



Summary



Coverage across reference



Coverage histogram

15. Glosario.

[i] **Polímero**: Macromoléculas (generalmente orgánicas) formadas por la unión de moléculas más pequeñas llamadas monómeros.
[ii] **Farmacogenética**: Disciplina que estudia el efecto de la variabilidad genética de un individuo en su respuesta a determinados fármacos.
[iii] **Mutación patogénica**: Mutación que causa la aparición de enfermedad con una elevada penetrancia (proporción de individuos que teniendo el cambio expresan la enfermedad).
[iv] **Base nitrogenada**: 1. Compuesto orgánico cíclico, que incluye dos o más átomos de nitrógeno. 2. Unidad mínima de información equivalente a un bit informático.
[v] **Célula somática**: Célula que conforma el crecimiento de los tejidos y órganos de un ser vivo pluricelular.
[vi] **Cromosoma**: Estructura que se encuentra en el centro (núcleo) de las células que transporta fragmentos largos de ADN.
[vii] **Promotor**: Secuencia que señala el comienzo de la transcripción del ADN a ARNm.
[viii] **Potenciador**: Secuencia reguladora que se encuentra lejos del inicio de la secuencia. Este aumenta la intensidad de la actividad de la transcripción.
[ix] **Eucariota y Procariota**: Tipos de células que se diferencian por tener o no núcleo y por sus sistemas de membranas internas. Ambas tienen material genético heredable y una membrana que las separa del exterior.
[x] **Endémica**: Se aplica a la enfermedad que se desarrolla habitualmente en una región determinada.
[xi] **Enfermedad de Wilson**: Trastorno hereditario en el cual hay demasiado cobre en los tejidos corporales. El exceso de cobre causa daño al hígado y al sistema nervioso.
[xii] **Diagnóstico Presintomático**: Identificación de individuos saludables que pueden haber heredado un gen determinístico para una enfermedad.
[xiii] **Árbol Filogénetico**: Árbol que muestra las relaciones evolutivas entre varias especies u otras entidades que se cree que tienen un ascendencia común.
[xiv] **Indel**: La palabra indel es una contradicción de "inserción o deleción", en referencia a los dos tipos de mutaciones genéticas que se consideran a menudo juntas a causa de su efecto similar y la incapacidad de distinguir entre ellas en una comparación de dos secuencias.
[xv] **Cromátida**: Una de las unidades longitudinales de un cromosoma duplicado, unida a su cromátida hermana por el centrómero, es decir, la cromátida es toda la parte derecha o izquierda del centrómero del cromosoma.

[xvi] **Alelo**: Cada una de las formas alternativas que puede tener un mismo gen que se diferencian en su secuencia y que se puede manifestar en modificaciones concretas de la función de ese gen.
[xvii] **Telómero**: Extremo de un cromosoma.
[xviii] **Pipes**: Elemento informático que lleva la información de un comando a otro en Linux.
[xix] **Pileup**: Formato de texto basado para resumir las lecturas de las bases de llamadas de alineamientos a una secuencia de referencia.
[xx] **Epigenética**: En un sentido amplio, estudio de todos aquellos factores no genéticos que intervienen en la determinación de la ontogenia o desarrollo de un organismo, desde el óvulo fertilizado hasta su senescencia, pasando por la forma adulta; y que igualmente interviene en la regulación heredable de la expresión génica sin cambio en la secuencia de nucleótidos.