Sección de Biología
Universidad de La Laguna

Análisis de datos en biocomputación: tecnología minION y herramientas bioinformáticas

Data analysis in biocomputing: minION technology and bioinformatic tools

Eva Tosco Herrera

Trabajo de Fin de Grado tutorizado por Carlos Pérez González

(Departamento de Matemáticas, Estadística e Investigación Operativa)

Julio de 2019

ÍNDICE DE CONTENIDOS

Resumen del trabajo

El MinION es un dispositivo de secuenciación lanzado al mercado en el año 2015, cuya función se basa en el uso de nanoporos como herramientas para la identificación de bases nucleotídicas del ADN en cadenas sencillas. Esto posibilita una secuenciación más rápida de mayores fragmentos de ácidos nucleicos, especialmente en el caso de genomas bacterianos. En el presente trabajo se estudia, gracias a las herramientas de bioinformática basadas en el lenguaje de programación *Python*, los resultados de este nuevo sistema de secuenciación, gestionando sus lecturas y ensamblándolas mediante software de uso habitual para facilitar el flujo de trabajo y la obtención de resultados del secuenciador de ONT. Todo este procedimiento se realiza sobre la plataforma *Jupyter Notebooks*, un sistema de cuadernos interactivos que facilita el aprendizaje del código, la exposición de datos y su distribución libre en repositorios online. Además, presentamos varias modificaciones en el código para mejorar la visualización de gráficas y tablas de parámetros, que nos permite evaluar la calidad de la secuenciación, proporcionando mayor dinamismo e interacción con los datos.

Abstract

The MinION is a sequencer device launched onto the market in 2015. Its functionality is based on the use of nanopores as tools to identify nucleotides of ADN in a single chain (*single-molecule*). This provides faster processes of sequencing something possible with longer fragments of ADN, especially in the case of bacterial genomes. This project analyzes the results from this new sequencing system, using *Python*-based bioinformatic tools, by handling the reads and assembling them using the usual software to simplify the workflow and the data obtainment from the sequencer. This entire process has been carried out on the *Jupyter Notebooks* platform, an interactive notebook system that eases code learning, data exhibition and free distribution in online repositories. Besides, we present some improvements in the code to enhance the graphs and parameter tables visualization that allow us to evaluate the sequencing quality, providing more dynamism and data interaction.

Introducción

El estudio de la información codificada en los genomas recibe el nombre de **bioinformática,** y reúne la ciencia computacional y la biología molecular, centrándose actualmente en el desarrollo de bases de datos, algoritmos de búsqueda y programas de predicción genética en ordenadores, entre otras herramientas analíticas utilizadas para entender los datos de los distintos tipos de secuencias (ADN, ARN y proteínas). Gracias al avance en este campo, los datos de las secuencias de ADN y proteínas se encuentran almacenados en bases de datos, algunas de ellas públicas, de forma que es posible consultarlas fácilmente con una conexión a Internet. Se han establecido distintos tipos de bases de datos, como las *bases de datos primarias*, que contienen las secuencias junto a información de su origen; en contraposición, las *bases de datos secundarias* contienen resultados de análisis sobre la información de las bases de datos primarias, tales como esquemas de secuencias especiales, mutaciones, relaciones evolutivas o incluso variaciones de genes (Pierce, 2010).

A pesar de que conocemos los tripletes que codifican cada aminoácido, la mayoría del ADN es no codificante: alberga funciones reguladoras y de otros tipos, sobre las cuales nos queda mucho por descubrir. Sin embargo, un lugar por donde empezar puede ser el desentrañado del conjunto de nucleótidos que conforman su ADN, es decir, su **secuenciación**.

La secuenciación de ADN consiste en la determinación de la secuencia de nucleótidos que condiciona el genotipo (y, por tanto, el fenotipo) de cada organismo vivo. Durante los dos últimos siglos se han desarrollado múltiples técnicas de secuenciación para conocer el genoma de múltiples especies de seres vivos y analizar la función de cada gen en el metabolismo, descubriendo nuevas mutaciones y sus consecuencias, tanto desde un punto de vista sanitario como desde una perspectiva biológica (Klug, Cummings, Spencer, & Palladino, 2013). Ésta última define la biodiversidad como un fenómeno enriquecedor y universal, aunque parezca contradictorio que se considere como "global" un fenómeno que recoge todos los aspectos que hacen única a cada especie.

En este Trabajo de Fin de Grado se presentará una parte bibliográfica, para poner en contexto las técnicas de secuenciación previas a las actuales y nombrar las más novedosas, incluyendo algunas herramientas computacionales con las que podemos contar hoy en día. Posteriormente se explica la parte práctica del trabajo: el análisis de datos mediante cuadernos de *Python*, explicando los métodos utilizados y las conclusiones correspondientes.

Técnicas de secuenciación

El esquema global de la generación de mapas genómicos es el siguiente:

1. **Rotura** del genoma en millones de segmentos al azar.
2. **Lectura** de cada segmento.
3. Solapado entre segmentos con secuencias idénticas mediante herramientas de **alineamiento.**
4. Unión final entre todos los pequeños segmentos (también llamados *contigs*), resultando así una **secuencia** completa, en el caso de ensamblado *de novo* (sin previas referencias documentadas).

Este proceso requiere de automatización por una razón principal: las reacciones de secuenciación individuales (*lecturas de secuenciación*) resuelven cadenas de aproximadamente 600 pb de longitud, mientras que un sólo cromosoma contiene unos $3x10^8$ pb de ADN: la posibilidad actual de longitud máxima de lectura equivaldría a la lectura del 0,0002% de un cromosoma.

El principal problema de los proyectos de secuenciación de genomas es el ensamblado de las secuencias y la construcción de una **secuencia consenso**, que representa los nucleótidos de cada molécula de ADN en el genoma. Pero incluso el uso de secuenciación automatizada conlleva problemas en este aspecto: la tasa de error no es constante, y depende de múltiples factores (colorantes, pureza, homogeneidad de la muestra, etc.…). Para reducir este margen de error, en los proyectos de este estilo se obtienen múltiples lecturas independientes para cada par de bases, de forma que los errores al azar no causen una falsa secuencia consenso. De todo ello se extrae la enorme utilidad de la automatización: la necesidad de múltiples lecturas de un mismo fragmento de ADN, multiplicado por el resto de las secuencias del genoma de la muestra. A lo largo de este siglo se han ido desarrollando máquinas de secuenciación masiva con menor tasa de error y menores requerimientos respecto al tiempo de trabajo: estas máquinas se utilizan en paralelo, formando *cadenas de montaje de secuenciación*, donde varias máquinas trabajan de forma ininterrumpida, algo imposible de conseguir mediante trabajo humano.

Los resultados de estos procesos se categorizan a su vez de la siguiente forma (Griffiths, Wessler, Lewontin, Carroll, & Ayllón Gómez, 2008):

- **Secuencias borrador**, que consisten en un esbozo general con errores tipográficos, gramaticales, pequeños huecos, secciones por reordenar, etc.
- **Secuencias finalizadas**, con una tasa de errores muy baja.
- **Secuencias completas**, donde no hay errores y la secuencia es real y correcta.

Para llegar a la capacidad de secuenciación de la que disfrutamos en la actualidad se han desarrollado nuevas técnicas de secuenciación, cada vez más automatizadas aún, que se denominan por orden de desarrollo o por "generación", siendo las más antiguas de primera generación. Las distintas generaciones de secuenciación se explicarán a continuación de forma resumida.

Primera generación: método de Sanger

Desde un punto de vista histórico, el método más utilizado para secuenciar ADN ha sido el de **secuenciación por dideoxinucleótidos**, también denominado **método de Sanger** en honor a su creador Fred Sanger, Premio Nobel de Química en 1980. En esta técnica, la molécula completa de ADN que se desea secuenciar se convierte en fragmentos más pequeños, hasta reducirla a cadenas sencillas que luego se utilizan como moldes para sintetizar cadenas complementarias. Es entonces cuando se añaden cebadores complementarios al ADN diana, además de ADN polimerasa y los cuatro desoxirribonucleótidos trifosfato (dATP, dCTP, dGTP y dTTP). La clave de la técnica reside en la adición de dideoxinucleótidos (ddNTPs), que poseen un radical hidrógeno 3' en vez de un grupo hidroxilo 3'. A este tipo de nucleótidos se les suele llamar **nucleótidos de terminación de cadena**: al incluir los ddNTPs durante una PCR, la ADN polimerasa inserta cada cierto tiempo uno de estos nucleótidos modificados dentro de la cadena en crecimiento. Debido a la ausencia del grupo hidroxilo en la posición 3' del anillo central del nucleótido, se trunca la posibilidad de formar enlaces fosfodiéster y es entonces cuando la síntesis de la hebra de ADN complementaria se detiene. Conforme vaya avanzando la PCR junto a los ddNTPs, se irán truncando fragmentos de ADN en cada posición consecutiva de nucleótidos. Así, se irán liberando secuencias casi idénticas y de longitud similar, cuyo nucleótido de diferencia sea el siguiente a la secuencia idéntica de ambos fragmentos. Según esta técnica, es posible entonces insertar un ddNTP en cada posición del ADN en síntesis, para luego poder separar los fragmentos mediante una electroforesis en gel.

En el método inicial de Sanger se utilizaban 4 tubos de ensayo distintos para las reacciones, con un ddNTP distinto en cada una, marcados radiactivamente para el análisis de la secuencia mediante autorradiografía de cada gel de poliacrilamida, midiendo los patrones de bandas. En los últimos 20 años se ha modificado el método tradicional de Sanger, hasta convertirse en **secuenciación de ADN automatizada de alto rendimiento.** Actualmente, los ciclos de reacción ocurren en un único tubo (en vez de cuatro), donde cada uno de los cuatro tipos de ddNTPs lleva unido un colorante fluorescente de un color distintivo. Así, los productos de las reacciones se separan en un solo gel de poliacrilamida, dentro de un **gel capilar**. Según los fragmentos de ADN se van desplazando por el gel, van siendo detectados por un láser que excita los colorantes fluorescentes

y provoca la emisión de cada longitud de onda característica de los ddNTP marcados. La luz emitida se percibe mediante un detector, que amplifica las señales y las transmite a un ordenador, cuya función es convertir los patrones de luz en una secuencia expresada gráficamente (**electroferograma**) mediante picos coloreados que se corresponden con cada nucleótido del ADN de la muestra. Esta técnica potenció la velocidad de los resultados a un mínimo coste, cumpliendo un papel esencial en el desarrollo del Proyecto Genoma Humano, comenzado en 1990 y completado en 2016.

A pesar de la gran importancia del método de Sanger como base metodológica y su aplicación, se ha producido una enorme evolución en el campo de las técnicas de secuenciación, algo necesario para satisfacer la demanda de secuenciación de genomas completos y el desarrollo de la **genómica** (Klug et al., 2013).

NGS: secuenciación de segunda generación

Todas las tecnologías que llevan más allá la técnica de Sanger se denominan tecnologías de secuenciación de nueva generación **(NGS)**, más conocidas actualmente como "**técnicas de segunda generación**"**.** En ellas se implementan técnicas de procesamiento en paralelo, combinadas con tratamientos avanzados de imágenes de fluorescencia, que disparan la velocidad de secuenciación y reducen mucho los costes.

Ilustración 1: aspecto de Ion Torrent Personal Machine, obtenido de (Thermo Fisher Scientific, 2019).

En la actualidad existen unas 5 tecnologías distintas de secuenciación de segunda generación, aunque algunas se utilizan más que otras: las más usadas son *Ion Torrent Personal Genome Machine* y *Illumina HiSeq*. Funcionan de manera similar, utilizando secuenciación por síntesis. Las técnicas de Illumina son ligeramente distintas, ya que unen los fragmentos de ADN a un soporte sólido (parecido a un microchip) para usarlos como molde de síntesis y determinar las secuencias de fragmentos de lectura más pequeños (cerca de 100 pb) mediante reacciones similares a las de Sanger (dideoxinucleótidos), pero a una velocidad mayor (300 millones de lecturas por cada flujo). Consiste principalmente en la adición de nucleótidos marcados individualmente, lavando los que no se unan a las hebras de ADN y detectando los que sí se han unido, de forma que la repetición de este ciclo cause la resolución de la secuencia de estudio (Klug, Cummings, Spencer, Palladino, & Killian, 2016).

TGS: secuenciación de tercera generación y *single-molecule sequencing*

Poco después de que se comercializaran las técnicas de secuenciación de segunda generación ya se estaban creando **técnicas de secuenciación de tercera generación (TGS)**. Éstas últimas están basadas en estrategias para secuenciar moléculas completas de cadenas sencillas de ADN (*single-molecule sequencing*). Actualmente se presentan distintas formas de aproximación a este sistema, y una de ellas es el *PacBio* (de la empresa Pacific Biosciences). Es capaz de leer secuencias con un máximo de 1500 bp, y consiste en la unión de cadenas sencillas de ADN de la muestra a una sola ADN polimerasa para posteriormente observar la síntesis de ADN en tiempo real. La polimerasa está anclada a un sustrato: un nanoporo de unos 10 nm de diámetro, localizado en una fina capa de metal, sobre un sustrato de vidrio. Esta configuración permite la emisión de luz necesaria para detectar los nucleótidos unidos a fluoróforos, con un color correspondiente para cada tipo de nucleótido, según se vayan uniendo a la cadena de ADN (Klug et al., 2016).

MinION

Otra aproximación al *single-molecule sequencing* es el MinION, desarrollado por la empresa Oxford Nanopore Technologies. Consiste en un dispositivo portátil, que se conecta a un PC mediante un puerto USB para la visualización de los datos de rendimiento en tiempo real: muestra valores como el porcentaje de nanoporos activos, indicando el correcto funcionamiento del dispositivo. El sistema consiste en hacer pasar una corriente eléctrica a través de membranas resistentes a la electricidad, donde están fijados una serie de nanoporos proteicos. A medida de que el ADN o ARN pasa a través de los nanoporos, se producen alteraciones en la corriente eléctrica, emitiendo señales que pueden ser analizadas para definir la secuencia de bases (Oxford Nanopore Technologies, 2018).

Ilustración 2: aspecto superior del MinION. Debajo de la cubierta blanca se encuentra la "flow cell".

El dispositivo devuelve un archivo digital con formato FAST5, donde se han traducido las señales eléctricas y los datos en bruto. Este archivo se somete a varios análisis mediante distintos programas informáticos, donde cada uno utiliza el archivo saliente del programa anterior, formando así una cadena de procesamiento: esto lo denominamos *pipeline*, *workflow* o flujo de trabajo.

Sólo es posible entender estos datos si, después de la secuenciación, se traducen las señales eléctricas a nucleótidos: este proceso se denomina *basecalling*, y lo puede realizar un programa concreto (*basecaller*) en el ordenador al que tengamos conectado el dispositivo. También puede realizarlo otro dispositivo complementario, del mismo fabricante, que reclame los datos del archivo de entrada y los convierta en letras (haciendo referencia a nucleótidos) en un archivo de texto saliente con formato FASTQ.

Para poder entender el flujo de trabajo al que deben ser sometidos los datos salientes del secuenciador, debemos explicar brevemente los tipos de archivos que se utilizan de forma general en el software de gestión de datos, de forma que podamos seguir el proceso y la función de cada programa en cada etapa del método (explicado con mayor detalle en el apartado de objetivos), donde se expresa la utilidad de la herramienta bioinformática que se expone en este trabajo.

Estructura de datos en bioinformática

Las nuevas tecnologías de secuenciación han acelerado tanto la obtención de datos, que el siguiente problema es su gestión, la grabación y el archivo de enormes secuencias de ADN. Actualmente se trabaja con una serie de distintos formatos, estándares para el estudio y el análisis de las secuencias nucleotídicas.

Archivos FAST5

Los archivos .fast5 que obtenemos cuando se realizan lecturas con el MinION también se conocen como archivos de tipo HDF5: consisten en un formato que pertenece a una suite tecnológica y librería de datos muy flexible que sirve para la gestión de datos, ya que permite el almacenamiento de un número ilimitado de valores de distintos tipos. Se trata entonces de un software librería que representa información compleja; funciona sobre distintas plataformas, implementando también interfaces de programación de distintas clases, como C, C++, e incluso Java (Hughes, 2017).

Archivos FASTQ

El formato FASTQ fue creado en el Trust Sanger Institute para guardar la secuencia de nucleótidos junto al nivel de calidad de las lecturas, pero actualmente se ha convertido en un formato común para compartir secuencias y datos. Consiste en un texto plano con extensiones ".fq" o ".fastq", que se pueden visualizar directamente a través de las terminales de comandos de los sistemas operativos Unix y Linux. En estos archivos, cada secuencia está definida por 4 líneas de texto: la primera empieza con una arroba ("@"), seguida de un código identificador de secuencia y una descripción opcional. La segunda línea es la secuencia de letras o nucleótidos: A, C, G, T y N (base nucleotídica desconocida), indicada en gris. La tercera línea comienza con un signo más ("+"), que puede funcionar como un marcador de fin de secuencia y puede estar seguido del mismo identificador de secuencia, además de cualquier otra descripción o comentario relevante. La cuarta línea está formada por los valores de calidad (marcada en rosa pálido) de la secuencia presente en la segunda línea, y debe contener tantos símbolos como letras haya en la secuencia. Siguiendo un ejemplo de una secuencia sencilla obtenida a través de sistema de secuenciación Illumina:

```
@HWI-ST193:542:C2H0GACXX:8:1101:4404:2179:Y:0:ACACGA
ATGCNTTTTATAATCAAAAGCGAAGACCTAGCAGGAGGTTAAAAACCTTT
+
<<<<#2<@@5:9@44:@@?4(-8@(<9@<<658.1/41/451/40<>???????9??
```

La primera línea es el identificador de secuencia y los ítems o elementos de descripción, separados por dos puntos (":") o espacios. Las descripciones pueden variar según los secuenciadores y las fuentes de datos. En este ejemplo, cada ítem se identifica de la siguiente forma:

- HWI-ST193:542: nombre de la secuencia (debe ser un identificador único).
- C2H0GACXX:8:1101: identificador de la celda de flujo utilizada.
- 4404: coordenada X del clúster, dentro de la celda.
- 2179: coordenada Y del clúster, dentro de la celda.

- Y: estado de filtración (Y si la lectura está filtrada, N en caso contrario).
- 0:ACACGA: estado de los bits de control y secuencias índice (parámetros propios del secuenciador).

En dicho ejemplo, la segunda línea contiene la secuencia; la tercera línea sólo tiene un signo "+" y no incluye información adicional; la cuarta línea posee las puntuaciones de calidad de lectura (**puntuación Phred**) para cada base correspondiente en la segunda línea. La puntuación Phred está asociada a la fiabilidad de cada carácter o letra de la secuencia (probabilidad de que la lectura sea correcta). Así, las puntuaciones pueden abarcar desde 0 hasta 95, y se codifican con los siguientes caracteres en código ASCII en orden, de izquierda a derecha, siendo "!" la menor puntuación y "~" la mayor:

```
!"#$%&'()*+,./0123456789:;<1⁄4>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]
^_`abcdefghijklmnopqrstuvwxyz{|}~
```

La codificación presentada puede variar según los secuenciadores, y los caracteres no siempre pueden ser comparados entre sí: deben haber sido previamente convertidos a probabilidades mediante un algoritmo matemático, definido por el fabricante del secuenciador (Zhang, 2016). En el caso de los secuenciadores Illumina, por ejemplo, una puntuación de 50 equivale a un 99.999% de fiabilidad, mientras que una puntuación de 10 indica un 90% (Illumina, 2011).

Estos archivos FASTQ no tienen límite en el número total de secuencias, pero como las lecturas en las técnicas NGS son tan extensas, cada muestra utilizará muchos archivos FASTQ para reflejar la lectura completa. En la mayoría de los casos se comprimen en archivos comprimidos de software libre (archivos ".gz") para reducir el tamaño total.

Archivos FASTA

Con extensión ".fa" o ".fasta", estos archivos constituyen el formato de texto más usado para secuencias de ácidos nucleicos y secuencias de aminoácidos, incluso en las últimas etapas de la secuenciación de primera generación. Sin embargo, también lo es dentro de las NGS, en las herramientas software para alineamiento o mapeado de secuencias. Éstas permiten múltiples entradas en el mismo archivo, lo que se denomina *multi-fasta*.

En los archivos ".fasta" encontramos que cada secuencia contiene un encabezado iniciado por el carácter ">" (identificador de secuencia) y terminado por una nueva línea. También puede albergar información descriptiva. La secuencia en sí consiste en caracteres que representan bases nucleotídicas de la misma forma que en los archivos .fastq (A, T, G, C ó N si es indeterminada), normalmente en líneas con el mismo número de caracteres. Lo más usual son 80 caracteres, a pesar

de que cada línea no tiene límite real. Estos archivos pueden usarse para mostrar secuencias de un solo gen, un cromosoma o más de uno. A continuación se presentan, como ejemplo, la línea de encabezado y las primeras dos líneas de la secuencia del gen humano p53 (NC_000017.11, marcado en gris claro) en formato FASTA:

```
>gi|568815581:c7687550-7668402 Homo sapiens chromosome 17,
GRCh38 Primary Assembly
GATGGGATTGGGGTTTTCCCCTCCCATGTGCTCAAGACTGGCGCTAAAAGT TTT
GAGCTTCTCAAAAGTC
TAGAGCCACCGTCCAGGGAGCAGGTAGCTGCTGGGCTCCGGGGACACTTTG CGT
TCGGGCTGGGAGCGTG
```

Para un único cromosoma, el identificador de secuencia sólo puede ser el nombre del propio cromosoma. Por ejemplo, a continuación, se presenta una secuencia parcial del cromosoma 1 humano en formato FASTA:

```
>chr1
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN NNNNNNNNNNNNNNNNNN
TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTA
ACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAAC
CCTAACCCAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCC
```

Un archivo FASTA pequeño puede visualizarse con herramientas básicas de edición de texto y comandos en terminales Unix/Linux. Para archivos más grandes, como es el caso de los genomas de referencia, se puede usar software basado en Java, como el programa del Broad Institute de Boston, *Integrative Genomics Viewer*, que es comúnmente utilizado (Zhang, 2016).

Archivos VCF

A pesar de que no nos centraremos especialmente en su uso en este trabajo, los archivos .vcf consisten en los formatos más utilizados a la hora de coleccionar variantes génicas. Se estableció durante el proyecto "1000 Genomes" y juega un papel fundamental en aplicaciones de secuenciación clínica. Este formato de archivo almacena información sobre variaciones de secuencia, tales como inserciones, deleciones o incluso SNP's (polimorfismos de un solo nucleótido), junto a anotaciones complementarias (Endrullat, Glökler, Franke, & Frohme, 2016).

Otros archivos

Algunos otros archivos que se utilizan frecuentemente y vale la pena mencionar son los archivos **tipo SAM** (extensión .sam): son los archivos de salida habituales del software de alineamiento o mapeado con el genoma de referencia. Constan de texto plano delimitado por tabuladores que contienen datos del proceso de alineamiento de las secuencias obtenidas. Podrán servir a su vez como archivos de entrada para el software posterior, como en la etapa de detección de variantes (*variant calling* en inglés). Como este tipo de archivos suelen ser de tamaño considerable, existen los archivos con **formato BAM** (extensión .bam), como versión binaria y reducida de un archivo SAM, aunque suelen contener la misma información y cualquiera de los dos formatos es fácilmente convertible en el otro.

Otros ejemplos de tipos de archivos comunes son los archivos **BED** (.bed), **GFF** o **GTF** (.gff o .gtf, *general feature/transfer format*), utilizados para albergar resúmenes finales de datos y anotaciones de interés genómico: por ejemplo, información sobre un gen, localización cromosómica, pauta de lectura del gen, posición de inicio y de terminación, números de identificación, etc. Podemos encontrar estos archivos al descargar información de algunas bases de datos (Zhang, 2016).

Jupyter Notebooks

Jupyter Notebooks es una aplicación web de software libre que permite compartir documentos con el código de programación y ejecutarlos de forma interactiva, mostrando los diferentes resultados en forma de ecuaciones, visualizaciones y texto narrativo. Tiene múltiples usos, tales como mostrar y analizar datos científicos, simulaciones numéricas, modelización estadística, etc. (Project Jupyter, 2019).

En este trabajo utilizamos esta herramienta y *Python* como lenguaje de programación, con lo cual podemos modificar el código en tiempo real, así como comprobar de forma instantánea los cambios y sus repercusiones. También aporta la posibilidad de visualizarlos en cualquier ordenador y una gran facilidad para mostrar gráficas. Además, la compatibilidad con la inclusión de nuevos módulos para personalizar las gráficas de salida facilita mucho las tareas de procesamiento y análisis.

Objetivos

En el presente trabajo hemos utilizado como base estructural una serie de cuadernos realizados en trabajos previos de bioinformática (Rodríguez-Pérez et al., 2019), donde se explica con un enfoque educativo y de divulgación, pero abordado desde una perspectiva informática y de programación.

Así, se pretende explicar su funcionamiento y utilidad de los resultados bajo una nueva perspectiva: desde un punto de vista biológico, presentando las herramientas disponibles y desarrolladas para la simplificación y mejora del flujo de trabajo que se lleva a cabo a nivel bioinformático, partiendo de archivos procedentes de técnicas de secuenciación de última generación; concretamente, en ensamblado *de novo*, una técnica explicada más adelante. Antes de entrar a ello en profundidad, necesitamos una visión resumida sobre los pasos habituales que conforman este proceso, y así poder valorar la simplificación de las etapas en este tipo de procedimientos.

Flujo de trabajo en bioinformática

Como contexto general, los flujos de trabajo para la obtención de resultados a partir de secuenciadores pueden dividirse en tres partes (Front Line Genomics, 2017):

1. Secuenciación

En esta etapa entran en juego las técnicas de **creación de librerías**: aislamiento y extracción de ácidos nucleicos, centrado en la calidad, formato y cantidad de muestra. Sin embargo, debemos tener en cuenta que existen requerimientos especiales según el secuenciador a utilizar: los sistemas de tercera generación (como *Pacific Biosciences RS II*) requieren omitir el paso de amplificación de ADN. Existen otras especificaciones concretas respecto a factores como el pH, agentes quelantes (o ausencia de ellos), detergentes, desnaturalizantes, etc. En el caso de estar trabajando con MinION, en esta etapa se preparan las muestras por analizar con los reactivos estandarizados y el kit correspondiente, para su posterior micropipeteado en la *flow cell* (o celda de flujo). Es posible adquirir distintos tipos de kits para la creación rápida de librerías adecuadas, y es el paso que más tiempo conlleva.

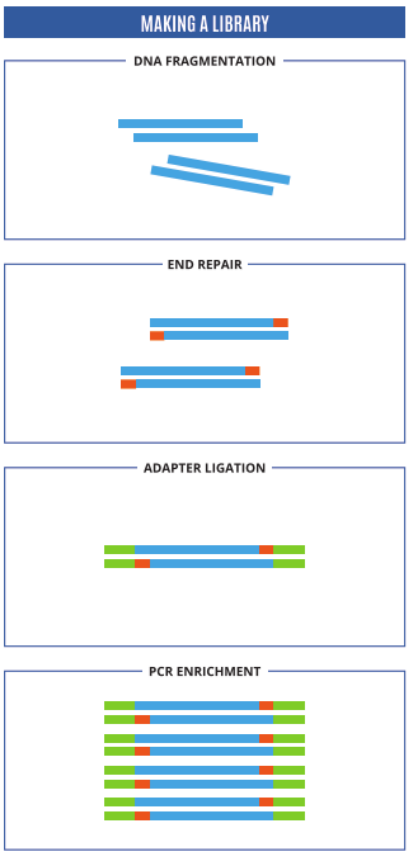

Ilustración 3: diagrama de formación de librerías, obtenido de (Front Line Genomics, 2018).

Los pasos para la creación de librerías pueden resumirse en las siguientes subetapas:

1. Fragmentación de ADN: esencial para asegurar que los *contigs* son de tamaño similar antes de secuenciar. Para los sistemas de secuenciación de lecturas largas, suelen sintetizarse fragmentos de 10 kb.
2. Reparación de extremos: tras la fragmentación, se aplican una serie de tratamientos que rellenan huecos y eliminan segmentos de cadenas sobrantes, resultando en una muestra llena de fragmentos con extremos perfectamente truncados en cadena doble.
3. Ligado de adaptadores: se añaden los adaptadores (de cadena conocida), para unirse a los fragmentos de ADN.
4. Amplificación: se produce una PCR para crear una librería robusta de fragmentos de ADN para secuenciar. Este paso asegura que sólo las secuencias con adaptadores sean las secuenciadas.
5. Purificación y cuantificado: es importante cuantificar los ácidos nucleicos en la librería, para conocer si hay suficientes clones de la longitud requerida. Algunas técnicas de cuantificación son la espectrofotometría o la fluorimetría (Front Line Genomics, 2018a).

2. Análisis primario

En esta etapa se consideran las pruebas de una buena extracción, cuantificación y determinación de pureza de la muestra secuenciada. Alberga procesos de entrada de datos, *basecalling* y simulaciones (en caso de experimentos de prueba para comprobar la fiabilidad del secuenciador). El proceso de asignación de bases o *basecalling* se realiza en paralelo junto a la asignación de fiabilidad o de calidad a cada base reconocida: ambos conjuntos de datos se presentan en los archivos salientes del *basecaller* o software utilizado, como ya se explicó en el apartado de archivos de estructura de datos.

Entra en el proceso el **control de calidad:** esto asegura que el análisis a realizar será preciso y que los datos que se generen serán de alta calidad, o al menos serán de calidad suficiente como para aportar conclusiones válidas y fiables. Así, el control de la calidad de los datos se convierte en un aspecto esencial para la validez del experimento y la confianza atribuida a sus conclusiones. Dependiendo del sistema de secuenciación, este control puede realizarse a nivel de laboratorio, mediante reactivos, o a nivel de software, mediante algoritmos específicos que calculan la fiabilidad de los datos. Estos algoritmos pueden aportar información sobre la eficiencia de los reactivos utilizados, la calidad de la muestra e incluso el nivel de confianza de sus propios resultados.

Tal y como hemos comentado previamente, en las últimas técnicas de secuenciación se utiliza el *Phred score,* donde se asocia un valor de calidad dentro de una escala de símbolos a cada base

nucleotídica leída. Este aspecto también se encuentra incluido en el control de calidad. Así, tras esta etapa de análisis primario se suele obtener un archivo **.fastq,** con las secuencias y sus *scores* correspondientes (Front Line Genomics, 2017).

3. Análisis secundario

El análisis primario produce series de nucleótidos no alineados procedentes de los fragmentos de la muestra, algo que no aporta una visión general del genoma del organismo en cuestión: por ello, se procede después al alineado del genoma y su ensamblado correspondiente. Además, se realiza un filtrado de resultados según sus calidades, así como la definición de las variantes encontradas de un gen, en el caso de que proceda en el experimento. Algunas de las técnicas más utilizadas para llevar a cabo el ensamblado de las lecturas se resumen en los siguientes párrafos (Front Line Genomics, 2018b):

- **Mapeado en genomas de referencia:** ha sido históricamente la técnica más utilizada para el ensamblaje de genomas, principalmente por la sencillez del uso de un genoma completo (representativo de la especie estudiada) como una plantilla, para encajar los fragmentos de ADN de la muestra y conformar una imagen del genoma bajo estudio. Existen múltiples herramientas computacionales (ver tabla 1) que identifican las posiciones más probables de cada *contig* en el genoma mediante distintos algoritmos, y son capaces de extraer secuencias consenso de cada región, solapando secuencias de cada grupo de fragmentos similares. Sin embargo, la principal limitación de esta técnica es la necesidad de un genoma de referencia, que estará disponible o no, dependiendo del conocimiento previo disponible sobre la especie que se analice en el experimento.
- **Genomas gráficos de referencia:** con funcionamiento similar al mapeado en genomas de referencia. Sin embargo, esta técnica utiliza como plantilla un gráfico que representa los datos genómicos de cientos de miles de individuos de la especie en concreto, calculando las probabilidades de localización de cada nucleótido en cada posición y teniendo en cuenta también las variantes presentes en la población de referencia. Esta técnica resulta ser muchísimo más fiable, pero requiere una capacidad computacional difícilmente alcanzable en la actualidad.
- **Ensamblado de secuencias *de novo:*** esta última técnica ha sido muy popular recientemente; no requiere genomas de referencia, ya que se basa en el solapamiento computacional de los fragmentos de lectura mediante la búsqueda de las secuencias presentes al inicio y al final de los fragmentos, calculando las probabilidades de los *contigs* en cada posición del genoma y desarrollando así una secuencia continua. La fiabilidad del

método depende entonces del número de fragmentos que hacen referencia a la misma zona del genoma: esto se denomina "cobertura" (o *depth*, en inglés). Así, la secuencia final tendrá mayor fiabilidad conforme mayor cobertura se haya efectuado en el proceso, algo que depende de la técnica de secuenciación. Esta técnica puede usarse en el estudio de cualquier genoma, sin que resulte relevante hasta qué punto ha sido estudiado dicha especie previamente. Sin embargo, requiere altas capacidades de computación y sus resultados pueden estar limitados, a causa de la enorme cantidad de zonas de secuencias repetitivas en el genoma.

El último paso del análisis secundario consiste en la **detección de variantes** (*variant calling* en inglés). Las variantes pueden tener consecuencias a nivel fenotípico (serían variantes beneficiosas o perjudiciales, en el caso de enfermedades genéticas) o no tenerlas (neutrales). Además, pueden presentarse de muchas formas: cambios en un solo nucleótido (SNPs), inserciones o deleciones (INDELs), translocaciones, inversiones estructurales o cambios en el número de copias de cada región específica del genoma (CNVs) son algunos ejemplos. Estas variantes también son detectables en los genomas mediante algoritmos y comparaciones entre genomas de referencia, sean lineales o gráficos, aunque la limitación de la capacidad computacional permanece aún presente.

4. Análisis terciario

Es la etapa más práctica y más aplicable a nivel clínico y biológico, pero también es la más compleja: consiste en la anotación y el filtrado de variantes, según el tipo de investigación y el nivel de especificidad deseada. Tiene el objetivo de integrar la información genómica y los datos fenotípicos, conectándolos entre sí para poder obtener información sobre las funciones del genoma. También se pueden identificar variantes asociadas con enfermedades. En esta etapa, el archivo .vcf generado contendrá una lista de variantes neutrales y con consecuencias fenotípicas, de la cual se podrán descartar, en investigaciones clínicas, las variantes sin repercusiones patológicas, para estudiar aquellas que parezcan estar relacionadas con patologías. Cada variante se encontrará clasificada en una categoría de patogenicidad de las siguientes:

- **Variante patogénica:** causantes directas o relacionadas con fenotipo de enfermedades.
- **Variante probablemente patogénica:** variantes que parecen influir en patologías, pero aún sin confirmar, a falta de mayor investigación al respecto.
- **Variante benigna:** suelen descartarse en las primeras etapas del análisis terciario, ya que no suelen estar conectadas a fenotipos dañinos o patogénicos.

- **Variante probablemente benigna:** variantes que parecen resultar benignas, pero se encuentran pendientes de confirmar, a falta de más estudios al respecto.
- **Variantes de significado incierto (VUS):** variantes sobre las cuales hay muy poca bibliografía o no pueden clasificarse en ninguna otra categoría por el momento.

Existe una enorme variedad de programas analíticos disponibles que resuelven el problema de la filtración de variantes; sin embargo, ante la enorme oferta de herramientas, la elección puede resultar confusa (Front Line Genomics, 2018b). En la tabla 1 se presentan, junto a muchos otros programas del resto de etapas del *workflow* (muchos de ellos se utilizan en la herramienta presentada en el apartado de objetivos), algunos ejemplos de detectores de variantes.

Todo este proceso puede simplificarse mediante el uso de herramientas, que simplifican los pasos del flujo de trabajo de *basecalling*, control de calidad y ensamblado. En la siguiente sección se describirá la funcionalidad de una de estas aplicaciones, llamada *NanoDJ*, que engloba gran parte de las herramientas descritas anteriormente. El uso de esta aplicación pretende facilitar los métodos y procedimientos relacionados con el tratamiento de datos de secuenciación genómica.

Tabla 1: Lista de algunos ejemplos de programas informáticos relevantes en el flujo de trabajo expuesto.

Etapas		
2	**Análisis primario**	Entrada de datos, *basecalling* y simulaciones
3.1	**Análisis secundario**	Control de calidad, resumen y filtrado de datos
3.2	**Análisis secundario**	Ensamblado genómico y comparaciones
3.3	**Análisis secundario**	Detección de variantes genéticas
4	**Análisis terciario**	Priorización y análisis de variantes patogénicas

Etapa	Software	Definición	Función	Referencia bibliográfica
2	***Guppy***	*Basecaller* de ONT	Transforma señales del archivo .fast5 en secuencias de bases en un archivo .fastq.	(Wick, 2018)
2	***Albacore***	*Basecaller* de ONT	Identifica la secuencia de DNA a partir de los datos en bruto. Sustituido últimamente por *Guppy.*	(Vera, 2018)
2	***NanoSim***	Simulador de ONT	Simula lecturas del MinION de forma estadística y las alinea con genomas de referencia.	(Yang et al., 2017)
3.1	***Porechop***	Supresor de adaptadores de ONT	Detecta y elimina adaptadores de las secuencias leídas del MinION.	(Wick, 2019)
3.1	***Biopython***	Paquete de librerías	Paquete optimizado para el trabajo con secuencias y datos biológicos basado en el lenguaje de programación *Python.*	(Cock et al., 2009)
3.2	***BWA***	Ensamblador de Illumina (referencia)	Especializado para el mapeo y ensamblado de secuencias con genoma de referencia, optimizado para Illumina.	(Li y Durbin, 2009)
3.2	***Rebaler***	Simulador de ensamblaje (referencia)	Ensambla lecturas largas usando un ensamblado de referencia, optimizado para ADN bacteriano.	(Wick, 2019)
3.2	***BLAST***	*Basic Local Alignment Search Tool*	Encuentra regiones similares entre secuencias, tanto de nucleótidos como de aminoácidos.	(NCBI, 2019)
3.2	***Unicycler***	Ensamblador (Illumina, Pac-Bio y ONT)	Ensambla lecturas largas, optimizado para genomas bacterianos. Capaz de ensamblados híbridos.	(Wick et al., 2017)
3.2	***MaSuRCA***	Ensamblador de novo (lecturas cortas)	Ensambla mezclas de lecturas de Illumina, 454 o Sanger, siendo posible añadir lecturas largas de otros secuenciadores.	(Zimin et al., 2013)
3.2	***Miniasm***	Ensamblador de novo (Pac-Bio y ONT)	Ensambla lecturas largas mediante la concatenación de fragmentos, sin utilizar genoma de referencia.	(Li, 2019)
3.2	***Canu***	Ensamblador de novo (*single-molecule*)	Ensambla lecturas largas, optimizado para *single-molecule sequencing* de genomas microbianos (bacterias y eucariotas).	(Koren et al., 2017)
3.2	***Flye***	Ensamblador de novo (Pac-Bio y ONT)	Ensambla fragmentos largos y metagenomas, diseñado para trabajar desde genomas bacterianos hasta de mamíferos.	(Kolmogorov, 2019)
3.2	***QUAST***	Paquete de ensamblaje	Ensambladores de genomas y metagenomas. Capaz de comparar y evaluar múltiples ensamblados con o sin genoma de referencia.	(Gurevich et al., 2013)
3.2	***Bandage***	Visor de ensamblado	Muestra los gráficos de ensamblado de forma interactiva, incluyendo las conexiones entre contigs o fragmentos.	(Wick et al., 2015)
3.3	***smCounter2***	Detector de variantes genéticas	Detecta variantes con bajas frecuencias, especialmente en zonas no codificantes.	(Xu et al., 2019)
3.3	***Pisces***	Detector de variantes genéticas (Illumina)	Reconoce variantes, especializado para comparar secuencias con genomas de referencia.	(Dunn et al., 2019)
4	***Exomiser***	Filtro de variantes patogénicas	Detecta y filtra variantes patogénicas a partir de secuencias de exomas o genomas completos.	(Smedley et al., 2015)

Funcionamiento de *NanoDJ*

NanoDJ es un paquete de software libre desarrollado por el ITER (Instituto Tecnológico y de Energías Renovables, España) que consigue aunar múltiples pasos del análisis de las secuencias directas que genera el MinION (Rodríguez-Pérez et al., 2019).

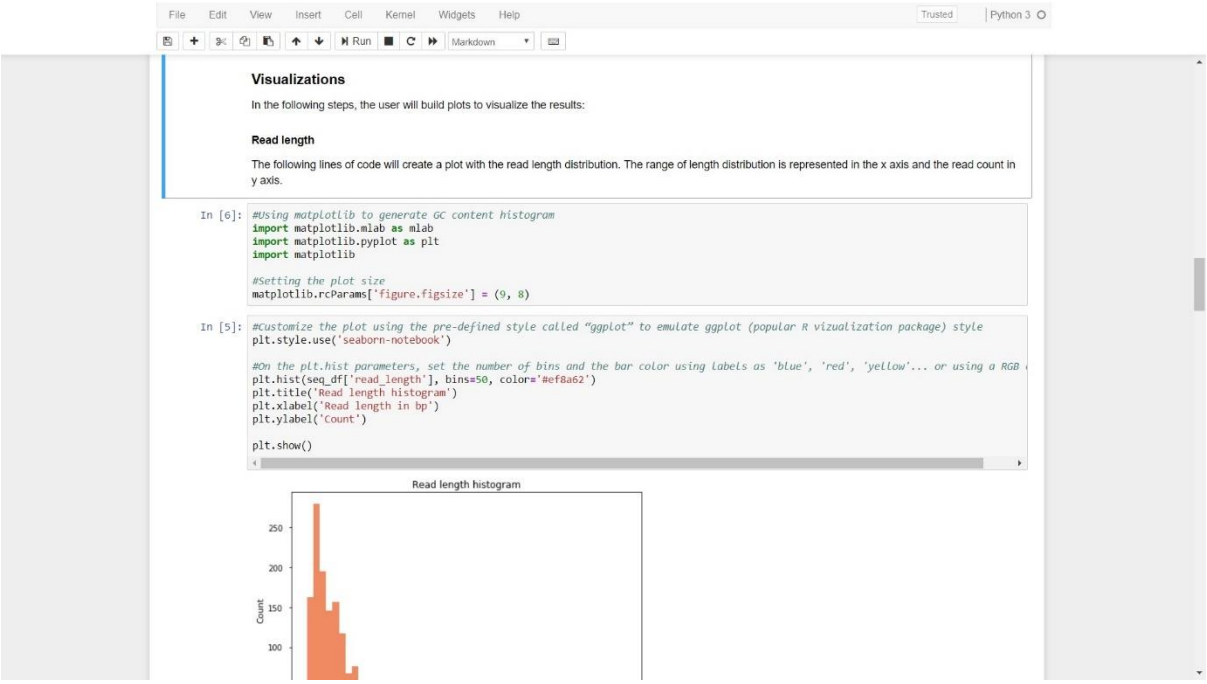

Ilustración 4: visualización del cuaderno de NanoDJ al abrirlo con Jupyter.

El objetivo principal de *NanoDJ* es el procesamiento de los datos de las secuencias obtenidas por el secuenciador MinION de la forma más eficiente posible. El software está basado en *Python*, un lenguaje de programación sencillo, y podemos utilizarlo en un cuaderno de *Jupyter*. A continuación se describe una visión superficial de las tareas que lleva a cabo esta aplicación.

En primer lugar, el cuaderno lee el archivo .**fast5** (que procedería directamente del MinION) a través de un paquete de *Python*, para convertir los datos de lectura (*reads*) en secuencias de nucleótidos (proceso de *basecalling*). Sin embargo, si usamos el MinIT (ver ilustración 5), un dispositivo complementario de la misma empresa, este paso se realizaría dentro de él, utilizando otro software (llamado *guppy*) pero resultando en archivos del mismo tipo: un archivo .**fastq**. En este paso también convertiremos el archivo .fastq en .**fasta**, no para el desarrollo de gráficos, sino para etapas posteriores de alineamiento mediante el uso de BLAST (como paquete de *Biopython,* librería especializada en la gestión de datos biológicos), que compara la muestra con un archivo de genoma de referencia, también en formato .fasta.

Ilustración 5: aspecto del dispositivo complementario MinIT, de ONT.

Tras la realización del control de calidad mediante el uso de los comandos de creación de gráficos, es posible clasificar las secuencias obtenidas en el experimento según las especies presentes. Existen distintos programas para ello, pero *NanoDJ* se encarga de esta función usando BLAST, de forma que podamos identificar los organismos o el organismo que se encontraba en la muestra biológica original. Para ejecutarlo, primero debemos definir los archivos que se usarán como genomas de referencia (o base de datos), y luego ejecutar el paquete, indicando que analice el archivo de lectura .**fasta** que obtuvimos a partir del archivo .fastq. El resultado en este caso es una tabla en formato .**csv**, que será más sencilla de procesar a la hora de generar gráficos. Los datos se encuentran organizados por grupos en dicha tabla, de forma que podemos contar el número de organismos presentes en la muestra, así como el número de secuencias alineadas con cada genoma de referencia.

En el cuaderno de *NanoDJ* se utilizan *numpy* y *pandas*: paquetes de *Python* para leer la tabla .**csv** y referenciar en el cuaderno los datos del archivo, definiendo las filas y las columnas de la tabla de secuencias alineadas y no alineadas, con sus respectivas proporciones de alineamiento (lecturas alineadas de cada organismo entre el total de lecturas alineadas de todos los organismos). Así, se obtiene una representación básica de las especies en la muestra secuenciada, y podemos empezar a sacar conclusiones sobre el experimento.

Por último, el cuaderno utiliza el módulo *matplotlib* (ver ilustración 4), para desarrollar gráficos de sectores sobre los datos visualizados previamente en forma de tabla, asignando cada proporción a la sección correspondiente del gráfico.

Resultados y discusión

Outputs de los cuadernos

En el cuaderno educacional del proyecto que mostramos como ejemplo, se estudian varias muestras de distintos alimentos para identificar los organismos presentes en la misma mediante el alineamiento del ADN extraído con los genomas de referencia. En dicho ejemplo se presentan distintas tablas de datos que resumen los principales parámetros estadísticos, de forma que podamos visualizar que el proceso se desarrolla correctamente. Después, se muestran los distintos gráficos, diseñados con las plantillas predeterminadas de los paquetes de *Biopython*:

- **Contenido en guanina y citosina (GC),** como indicador de la calidad de lectura y como criterio de fragmentos obtenidos con mayor calidad.
- **Longitud de secuencias leídas**, para visualizar el tamaño de los fragmentos y la eficacia del proceso de preparación de librerías como etapa previa a todo el proceso analítico.
- **Calidad media de lecturas,** mediante la representación de la media aritmética de las distintas calidades de los fragmentos secuenciados, a partir de las puntuaciones de calidad de cada base leída dentro de cada fragmento.

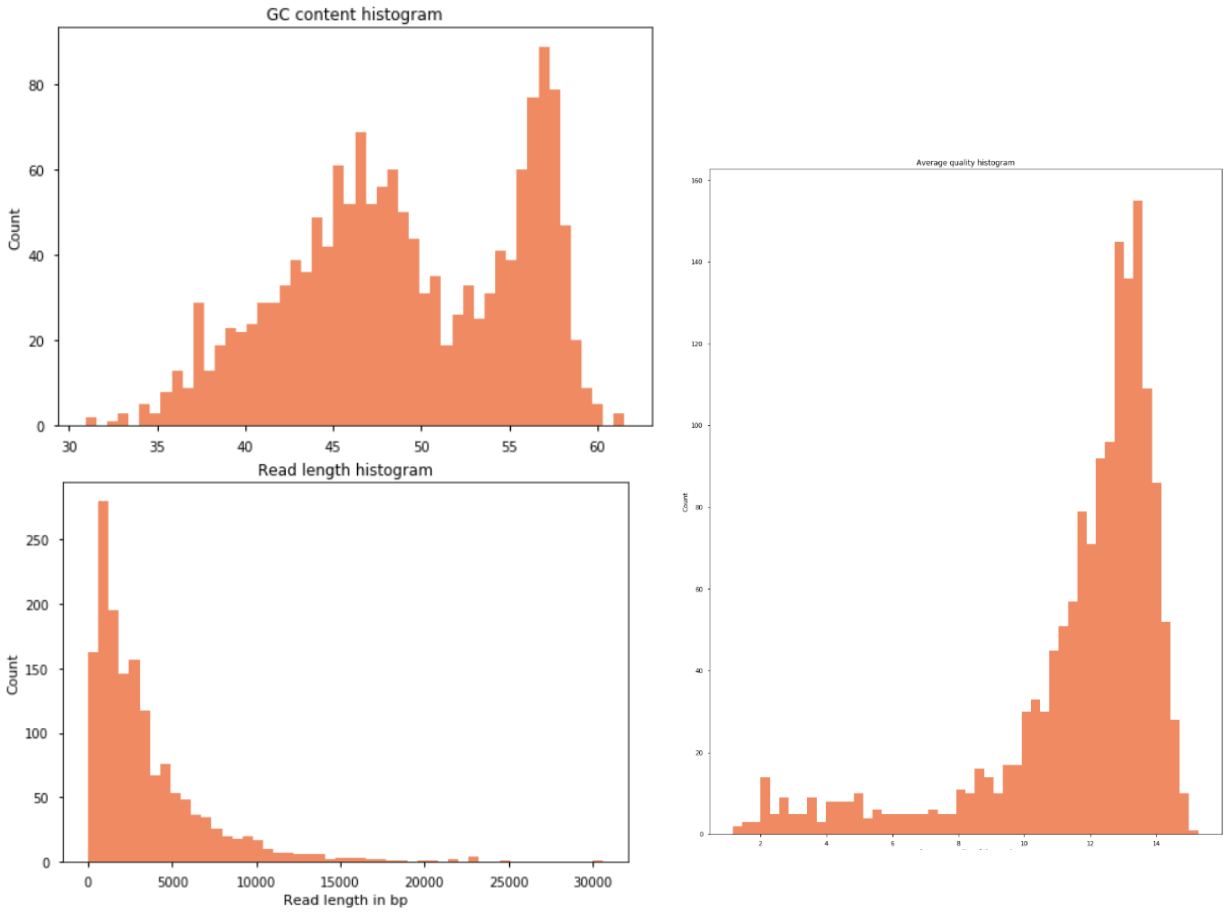

Ilustración 6: gráficas de calidad de lectura de contenido en GC (arriba), longitud de lecturas (debajo) y calidad media de lecturas (derecha).

Tras definir la calidad de los resultados a través de las gráficas, el *notebook* procede a ejecutar el paquete BLAST, encargado de comparar las secuencias leídas con el genoma de referencia. Para que esto se lleve a cabo, el paquete requiere definir previamente una base de datos para comparar. Utilizamos la versión basada en *Python* y no las herramientas disponibles en NCBI para ahorrar tiempo y capacidad computacional; tal y como se presenta en forma de comando, es el usuario el que define el genoma de referencia y las lecturas con las que desea trabajar. En nuestro caso, el *notebook* descrito reconoce un archivo con todos los genomas de referencia de los organismos que esperamos encontrar en la muestra. Todo este proceso desemboca en la visualización de una tabla que muestra la proporción de secuencias alineadas con cada organismo respecto a la totalidad de secuencias alineadas con cualquier organismo del archivo de genomas de referencia. En la última fila de esta tabla también se muestra el recuento y la proporción de secuencias no alineadas con ningún organismo de referencia.

Por último, el *notebook* utiliza el paquete *matplotlib* para realizar un diagrama de sectores (ver ilustración 7), donde el ángulo de cada sector sea directamente proporcional al recuento de secuencias y su proporción, en el caso de cada especie. Podemos observar que, en el caso del ejemplo, partíamos de una muestra de mezcla de comida, y cada sector corresponde a un organismo distinto: rúcula (*Arugula sp.*), cerdo (*pig*), pollo (*chicken*), etc. El gráfico indica que la mayor cantidad de ADN presente en la muestra pertenece a maíz (*corn*) y a levaduras (*S. cerevisiae*).

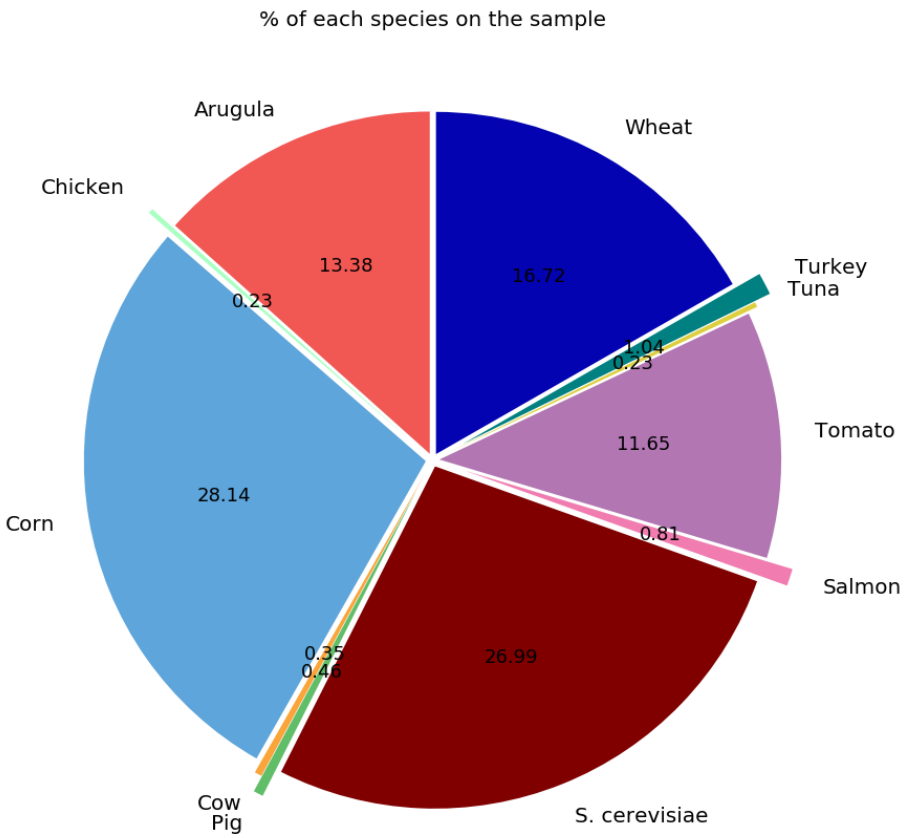

Ilustración 7: diagrama de sectores de ejemplo, con etiquetas y proporciones correspondientes.

Mejoras de los *outputs*

Para dar un paso más allá en las condiciones de los gráficos mostrados, acudimos a diversos paquetes de *Python* para mejorar la parte visual e interactiva de los mismos. Algunos ejemplos son los paquetes *numpy* y *pandas* para definir previamente los datos a analizar, el paquete *bokeh* que posibilita la interacción y el dinamismo de los histogramas o módulos del proyecto *Nanoplot* o los paquetes *nanomath* y *nanoget* para conformar una tabla de estadísticos básicos (De Coster, D'Hert, Schultz, Cruts, & Van Broeckhoven, 2018).

Utilizando el paquete *bokeh*, los gráficos presentados previamente se convierten en ventanas de observación de datos desplegables con información complementaria. Estas mejoras se presentan en un nuevo cuaderno, que lee los datos a partir del archivo .**fastq** original y comienza por definir el conjunto de datos para analizar, mostrando los parámetros de calidad calculados.

A continuación, definimos las características del histograma que deseamos presentar, además de la implementación de la herramienta *hovertool* (datos sobre la gráfica) para luego ejecutar el código y mostrar el gráfico interactivo. En las ilustraciones 8 y 9 podemos observar dichos gráficos.

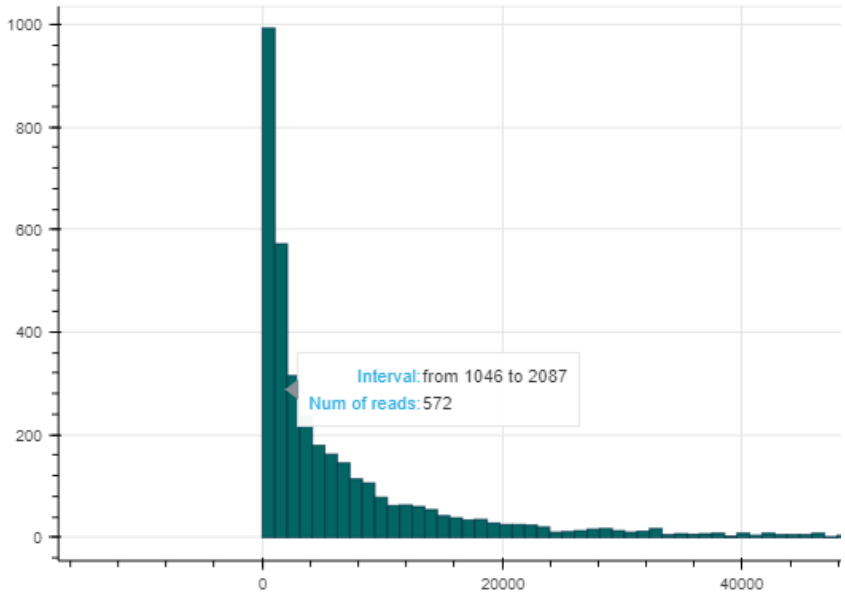

*Ilustración 8: gráfica interactiva de longitud de fragmentos y densidad de lecturas con datos de demostración ("*hovertool*") tras las mejoras.*

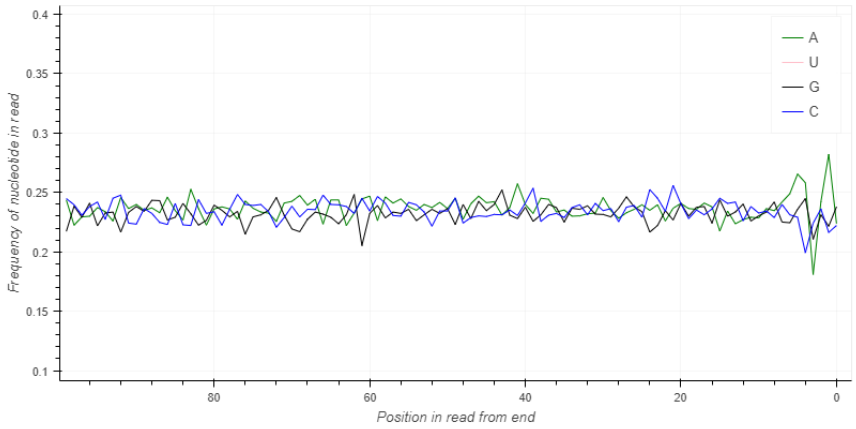

Ilustración 9: gráfica de frecuencia de nucleótidos al final de la lectura de cada fragmento.

feature	
General summary	
Mean read length	3,442.2
Mean read quality	8.5
Median read length	2,426.5
Median read quality	9.1
Number of reads	1,544.0
Read length N50	5,534.0
Total bases	5,314,681.0
Number, percentage and megabases of reads above quality cutoffs	
>Q5	1430 (92.6%) 5.2Mb
>Q7	1292 (83.7%) 4.7Mb
>Q10	242 (15.7%) 1.0Mb
>Q12	0 (0.0%) 0.0Mb
>Q15	0 (0.0%) 0.0Mb
Top 5 highest mean basecall quality scores and their read lengths	
1	11.2 (4980)
2	11.0 (2934)
3	11.0 (3254)
4	11.0 (1477)
5	11.0 (2789)
Top 5 longest reads and their mean basecall quality score	
1	30588 (8.8)
2	25001 (8.4)
3	22904 (9.7)
4	22801 (9.2)
5	22667 (6.9)

Ilustración 10: tabla de parámetros básicos de los datos mediante los paquetes nanomath *y* nanoget.

Respecto a la tabla de estadísticos básicos (ver ilustración 10), utilizando *nanomath* y *nanoget* podemos construir y mostrar una tabla de parámetros estadísticos básicos de los datos con los que estamos trabajando: la longitud media de lectura, la calidad media, medianas, número total de lecturas , número de bases totales, etc. También se presentan algunas listas de los cinco mejores fragmentos con mayor calidad de lectura, los cuartiles que los contienen y los fragmentos de mayor longitud con su calidad correspondiente.

Discusión

El secuenciador MinION produce una enorme cantidad de datos de forma muy rápida, fácil y portátil. Aunque en la actualidad no genera resultados de calidad suficiente como para obtener un ensamblado de genomas de alta fiabilidad por sí mismo, sabemos que sus lecturas complementan las secuencias de otros secuenciadores, como el *MiSeq* de Illumina. Recientemente se ha publicado el ensamblado híbrido del genoma de *Streptococcus agalactiae* (Hernández-Beeftink et al., 2018), un estreptococo β-hemolítico, grampositivo y anaerobio facultativo que forma parte de la flora normal del tracto gastrointestinal, que puede provocar infecciones generales e incluso complicar otras patologías (Fraile & López de Cueto, 2018). Su genoma es de interés clínico, ya que cerca del 50% se asocia a islas de patogenicidad, con genes de virulencia y elementos móviles, pero con una estructura general bastante conservada y al menos 69 regiones variables observadas. Estudiar en profundidad genomas de este tipo mejora las comparaciones entre genomas bacterianos y facilita el mapeo de dichos elementos móviles, así como sus consecuencias y relación con la adaptación bacteriana y la virulencia de estos microorganismos.

Las secuencias obtenidas a través del MinION provienen de la cepa SS1, aislada de una muestra biológica procedente de casos clínicos reales. Las lecturas se han analizado junto a lecturas de Illumina *MiSeq*, para realizar un ensamblado *de novo* del genoma de la bacteria. De esta forma podemos visualizar el ensamblado mediante la aplicación *bandage* (ver ilustración 11), que nos muestra aproximadamente la mitad del genoma como un solo *contig* o fragmento, secuenciando la gran mayoría de elementos móviles, aunque una pequeña región no se ha ensamblado correctamente debido a la presencia de secuencias repetitivas.

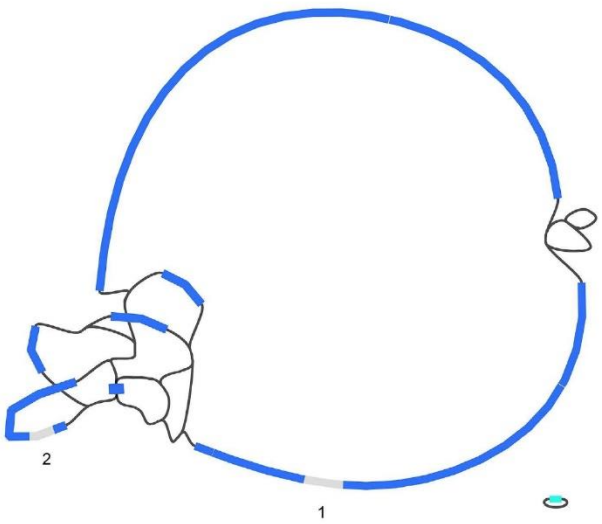

Ilustración 11: visualización del ensamblado del genoma de Streptococcus agalactiae *mediante* bandage (*Hernández-Beeftink et al., 2018).*

Con estos resultados podemos sacar conclusiones: a pesar de que las lecturas del MinION son de una calidad aproximada del 94-97% por sí solas (Tyler et al., 2018) (frente al máximo de 99,99% de los secuenciadores de Illumina (Tan, Opitz, Schlapbach, & Rehrauer, 2019)), resultan de utilidad a la hora de formar ensamblados híbridos (obtenidos mediante ambos secuenciadores) y complementan la información obtenida mediante un solo tipo de secuenciador.

Respecto a las mejoras realizadas en gráficos y estadísticos, basándonos en el código base del cuaderno de *NanoDJ*, hemos querido mostrar la versatilidad y la simplicidad del flujo de trabajo presentado, las enormes posibilidades que brinda y el nivel de adaptación que presenta el análisis de datos post-secuenciación; tanto para MinION en exclusiva, como para ensamblados híbridos.

Por este gran potencial, vale la pena seguir desarrollando herramientas que mejoren las lecturas y la posterior gestión en el análisis de los datos, ya que las facilidades físicas y funcionales de este pequeño secuenciador, como es el MinION, puede brindar grandes oportunidades en el futuro dentro del campo de la metagenómica.

Conclusiones

La evolución de las herramientas para la secuenciación y análisis de genomas completos ha supuesto una revolución en la investigación científica del área de la Genómica. A medida que se recogen más y más datos sobre los procesos genéticos, crece la necesidad de disponer de software más potente para agrupar, filtrar y organizar la información disponible, facilitando la interpretación y descripción de los procesos involucrados. El diseño y uso de estas herramientas define lo que es hoy día la bioinformática: un campo en continuo cambio y renovación, donde los programas que se utilizan son mayoritariamente libres y adaptables a cualquier tipo de experimento que se vaya a desempeñar. Este trabajo ha planteado el estudio del flujo de trabajo en la aplicación de técnicas de secuenciación y ensamblado, con el fin de conocer las estructuras de datos y herramientas usuales en bioinformática. En particular, además de conocer de cerca una tecnología concreta (implementada en el MinION, un sistema portátil y rápido con gran potencial), se ha realizado un análisis de datos obtenidos utilizando dicha tecnología, demostrando que la bioinformática es un campo de estudio dinámico, donde se van desarrollando poco a poco nuevas herramientas y nuevos enfoques a la hora de tratar con la cantidad ascendente de datos.

Con el gran número de posibilidades que existen actualmente, los investigadores se ven inmersos en una gran incógnita: los criterios para elegir un programa u otro dentro del flujo de trabajo, en función de sus necesidades y de conocimientos en el campo informático. En los últimos años se han empezado a publicar artículos científicos que evalúan la eficiencia, la calidad de los resultados y la facilidad de uso de distintos programas informáticos (Bohannan & Mitrofanova, 2019; Tan et al., 2019; Tyler et al., 2018), clasificándolos según las necesidades en cada tipo de secuenciador, o incluso teniendo en cuenta el tipo de muestra a secuenciar, comparando distintos softwares con funciones similares y valorando las ventajas e inconvenientes de cada uno. Sin embargo, continúa convirtiéndose en un caos a la hora de elegir los candidatos para obtener los mejores resultados. Estos conflictos se resolverían con el uso de cuadernos como el que hemos presentado en este trabajo, donde múltiples etapas del proceso de análisis de datos de la secuenciación sean fusionados e integrados en una única herramienta, que se podría utilizar para el mismo tipo de muestras y, además, se mantendría actualizada periódicamente, ya que permite su modificación para incluir nuevos programas de mejor rendimiento que se vayan desarrollando. Entre los varios aspectos estudiados en este trabajo se encuentra el análisis de las lecturas de secuencias genéticas y cómo mostrar dichos análisis en un contexto de control de calidad de las mismas: tal y como hemos mostrado en los resultados, es posible modificar y mejorar la visualización e interacción con los

resultados. De igual forma, es posible modificar el código de cualquiera de los programas que alberga el cuaderno sin modificar el resto de código de la aplicación.

Gracias al desarrollo de distintas estrategias a la hora de analizar los datos y el conocimiento de la enorme cantidad de opciones disponibles, los resultados de las investigaciones de secuenciación de genomas cada vez albergarán mayor calidad y mayor cantidad de datos, proporcionando mayor información sobre la biodiversidad, las causas de éstas a nivel metagenómico y sus posibles aplicaciones y potencial en muchos otros campos de estudio, como la biotecnología o la medicina.

Conclusions

The evolution of the tools for sequencing and analysis of complete genomes has meant a revolution in the scientific research in the genomics area. If more data is collected about genetic processes, the need for more powerful software to be available grows, so the information can be classified, filtered and organized to facilitate interpretation and description of those processes. The design and use of these tools determine what bioinformatics is nowadays – a field in continuous changes and renovation process, where software is mainly free and adaptable to any type of experiment to be performed. This project has also tried to study sequencing and assembly pipelines to get to know data structure and usual tools in biocomputing. Particularly, apart from approaching a specific technology (the one implemented in MinION, a fast and portable device with great potential), data analysis has been carried out using said technology, proving that biocomputing sciences is a dynamic field of study, where new tools and perspectives are continuously being developed to manage the increasing amount of data.

With the huge number of possibilities that exist, researchers find themselves lost into a big problem: criteria to choose a program or another inside the workflow, depending on their needs and knowledge in computer sciences. There has been a few articles published in recent years that assess efficiency, data quality and interface ease of different programs (Bohannan & Mitrofanova, 2019; Tan et al., 2019; Tyler et al., 2018), categorizing them by needs of each type of sequencer or even considering the kind of sample to use, comparing software with similar purposes and assessing advantages and disadvantages about each one. However, it is still a chaos when it comes to choosing the bioinformatic tools to obtain the best results. This conflict would be solved with notebooks such as the one presented in this project, where multiple steps of data analysis are integrated in only one tool, so it may be used with the same type of samples and also can stay regularly updated, as it can be modified to incorporate new high-performance software. Some of

the aspects that has been studied in this project were sequence reads analysis and data visualization in a quality control context. As showed in the results of this project, it is possible to improve data visualization and interaction, but it is also possible to change the code for any of the software from the notebook, without the need to change anything else.

Thanks to this development of free and available data management strategies and data knowledge, research results will have better quality and quantity of data, providing more information about biodiversity and their metagenomic causes as well as possible applications and potential in many other fields such as biotechnology or medicine.

Agradecimientos

Este Trabajo de Fin de Grado no habría sido posible sin la ayuda de mi tutor, que me ha orientado todo lo posible, me ha abierto puertas y en ocasiones ha cumplido una función complementaria, tranquilizando mi estrés y respondiendo a las mil preguntas que llevaba a tutorías. También me gustaría agradecer a la Unidad de Investigación del Hospital Universitario Nuestra Señora de Candelaria. Agradezco enormemente a todos, especialmente a Héctor, a Tamara y a Carlos Flores, el recibimiento que me dieron en la Unidad y la disponibilidad para resolver cualquier duda. El ambiente de trabajo y el aire innovador que se respira allí me ha hecho interesarme aún más, si cabía, en la bioinformática como gran campo de investigación que es. No hay nada que desee más que la utilidad de los trabajos que logre desempeñar en el futuro.

Así mismo quiero agradecer al Tribunal el haber dedicado su tiempo a leer este trabajo. A todos los profesores y profesoras que han impartido clase en el Grado en Biología durante los años que llevo cursándolo, tanto a mi grupo como al resto de grupos de mi curso. Cada uno de ellos ha dejado marca en mí, y gracias a ellos reviso los trabajos, las presentaciones y las bibliografías una y otra vez, para que contengan fuentes fiables y datos correctos. También quiero recordar a mi tutora de prácticas externas, con la que tuve la gran suerte de trabajar: me apoyó sin descanso y finalmente ha ganado mi máximo respeto y admiración.

Gracias por el apoyo de toda la gente que me mantiene a flote: a Raquel, a Gisela, a mi familia, a todos mis amigos y compañeros/as de clase por ayudarme tanto, especialmente a Javi, que siempre será mucho más que un compañero. Gracias a todas las personas que me han dedicado una palabra de ánimo, una sonrisa sincera, cualquier signo de apoyo o empatía, por pequeño que pudiera ser.

Por último, pero no menos importante, me gustaría darles las gracias a todas y cada una de mis divinas compañeras del gimnasio, incluida nuestra maravillosa monitora, que me ayudan a sobrepasar todos los obstáculos haciéndome reír. También han escuchado mil historias sobre el proceso de creación de este documento.

Todo lo que he vivido me ha dirigido hasta aquí, y estoy profundamente agradecida por todo lo que ha ocurrido en el proceso. Estoy orgullosa de haber elegido Biología como mi ruta de viaje.

Bibliografía

Bohannan, Z. S., & Mitrofanova, A. (2019). Calling Variants in the Clinic: Informed Variant Calling Decisions Based on Biological, Clinical, and Laboratory Variables. *Computational and Structural Biotechnology Journal*, *17*, 561-569. https://doi.org/10.1016/j.csbj.2019.04.002

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., & de Hoon, M. J. L. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422-1423. https://doi.org/10.1093/bioinformatics/btp163

De Coster, W., D'Hert, S., Schultz, D. T., Cruts, M., & Van Broeckhoven, C. (2018). NanoPack: Visualizing and processing long-read sequencing data. *Bioinformatics*, *34*(15), 2666-2669. https://doi.org/10.1093/bioinformatics/bty149

Dunn, T., Berry, G., Emig-Agius, D., Jiang, Y., Lei, S., Iyer, A., & Strömberg, M. (2019). Pisces: An accurate and versatile variant caller for somatic and germline next-generation sequencing data. *Bioinformatics*, *35*(9), 1579-1581. https://doi.org/10.1093/bioinformatics/bty849

Endrullat, C., Glökler, J., Franke, P., & Frohme, M. (2016). Standardization and quality management in next-generation sequencing. *Applied & Translational Genomics*, *10*, 2-9. https://doi.org/10.1016/j.atg.2016.06.001

Fraile, M. de la R., & López de Cueto, M. (2018). Streptococcus agalactiae. *Control Calidad SEIMC*, 3.

Front Line Genomics. (2017). Chapter 3: Analysis. *Clinical Genomics 101: 2017 Edition*, *1*. Recuperado de http://www.frontlinegenomics.com/magazine/10511/clinical-genomics-101-2017-edition/

Front Line Genomics. (2018a). Chapter 2: Turning ADN into data. *Genomic Data 101: 2018 Edition*. Recuperado de http://www.frontlinegenomics.com/magazine/21442/genomic-data-101-2018-edition/

Front Line Genomics. (2018b). Chapter 4: Variant Calling and Analysis. *Clinical Genomics 101: 2018 Edition*, *1*. Recuperado de http://www.frontlinegenomics.com/magazine/20409/clinical-genomics-101-2018-edition-free-download/

Griffiths, A. J. F., Wessler, S. R., Lewontin, R. C., Carroll, S. B., & Ayllón Gómez, F. (2008). *Genética*. España: Mcgraw-Hill Interamericana de España, S.A.U.

Gurevich, A., Saveliev, V., Vyahhi, N., & Tesler, G. (2013). QUAST: Quality assessment tool for genome assemblies. *Bioinformatics*, *29*(8), 1072-1075. https://doi.org/10.1093/bioinformatics/btt086

Hernández-Beeftink, T., Rodriguez-Perez, H., Usera, A., González-Montelongo, R., Lorenzo Salazar, J., Lorenzo-Díaz, F., & Flores, C. (2018). *Shallow MinION sequencing to assist de novo assembly of the Streptococcus agalactiae genome:* https://doi.org/10.1101/485029

Hughes, J. (2017). Exploring the FAST5 format [University of Glasgow]. Recuperado de Bioinformatics I/O: http://bioinformatics.cvr.ac.uk/blog/exploring-the-fast5-format/

Illumina (2011). *Quality Scores for Next-Generation Sequencing*. Recuperado de https://www.illumina.com/Documents/products/technotes/technote_Q-Scores.pdf

Klug, W. S., Cummings, M. R., Spencer, C. A., & Palladino, M. A. (2013). *Conceptos de genética*.

Klug, W. S., Cummings, M. R., Spencer, C. A., Palladino, M. A., & Killian, D. (2016). *Concepts of genetics*.

Kolmogorov, M. (2019). *Fast and accurate de novo assembler for single molecule sequencing reads: Fenderglass/Flye* [C++]. Recuperado de https://github.com/fenderglass/Flye

Koren, S., Walenz, B. P., Berlin, K., Miller, J. R., Bergman, N. H., & Phillippy, A. M. (2017). Canu: Scalable and accurate long-read assembly via adaptive *k*-mer weighting and repeat separation. *Genome Research*, *27*(5), 722-736. https://doi.org/10.1101/gr.215087.116

Li, H. (2019). *Ultrafast de novo assembly for long noisy reads (though having no consensus step): Lh3/miniasm* [TeX]. Recuperado de https://github.com/lh3/miniasm (Original work published 2015)

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

NCBI (2019). BLAST: Basic Local Alignment Search Tool. Recuperado el 27 de mayo de 2019, de https://blast.ncbi.nlm.nih.gov/Blast.cgi

Oxford Nanopore Technologies (2018). MinION. Recuperado el 5 de abril de 2019, de: http://nanoporetech.com/products/minion

Pierce, B. A. (2010). *Genética: Un enfoque conceptual*. Madrid: Editorial Médica Panamericana.

Project Jupyter (2019). Project Jupyter. Recuperado de https://www.jupyter.org

Rodríguez-Pérez, H., Hernández-Beeftink, T., Lorenzo-Salazar, J. M., Roda-García, J. L., Pérez-González, C. J., Colebrook, M., & Flores, C. (2019). NanoDJ: A Dockerized Jupyter notebook for interactive Oxford Nanopore MinION sequence manipulation and genome assembly. *BMC Bioinformatics*, *20*(1), 234. https://doi.org/10.1186/s12859-019-2860-z

Smedley, D., Jacobsen, J. O. B., Jäger, M., Köhler, S., Holtgrewe, M., Schubach, M., & Robinson, P. N. (2015). Next-generation diagnostics and disease-gene discovery with the Exomiser. *Nature Protocols*, *10*(12), 2004-2015. https://doi.org/10.1038/nprot.2015.124

Tan, G., Opitz, L., Schlapbach, R., & Rehrauer, H. (2019). Long fragments achieve lower base quality in Illumina paired-end sequencing. *Scientific Reports*, *9*(1), 2856. https://doi.org/10.1038/s41598-019-39076-7

Thermo Fisher Scientific. (2019). Ion Personal Genome Machine (PGM) System. Recuperado de https://www.thermofisher.com/order/catalog/product/4462921?SID=srch-srp-4462921

Tyler, A. D., Mataseje, L., Urfano, C. J., Schmidt, L., Antonation, K. S., Mulvey, M. R., & Corbett, C. R. (2018). Evaluation of Oxford Nanopore's MinION Sequencing Device for Microbial Whole Genome Sequencing Applications. *Scientific Reports*, *8*(1), 10931. https://doi.org/10.1038/s41598-018-29334-5

Vera, D. (2018). *Dockerfile for the Albacore basecaller from Oxford Nanopore: Dvera/albacore*. Recuperado de https://github.com/dvera/albacore (Original work published 2017)

Wick, R. (2018). *Add Guppy v0.5.1 and update to Nanopolish v0.9.0* · Recuperado de https://github.com/rrwick/Basecalling-comparison/commit/793b4ce980c2e156c3b132548036dcf5d2050f82

Wick, R. (2019a). *Adapter trimmer for Oxford Nanopore reads*. Recuperado de https://github.com/rrwick/Porechop

Wick, R. (2019b). *Reference-based long read assemblies of bacterial genomes: rrwick/Rebaler* [Python]. Recuperado de https://github.com/rrwick/Rebaler

Wick, R. R., Judd, L. M., Gorrie, C. L., & Holt, K. E. (2017). Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. *PLoS Computational Biology*, *13*(6). https://doi.org/10.1371/journal.pcbi.1005595

Wick, R. R., Schultz, M. B., Zobel, J., & Holt, K. E. (2015). Bandage: Interactive visualization of *de novo* genome assemblies: Fig. 1. *Bioinformatics*, *31*(20), 3350-3352. https://doi.org/10.1093/bioinformatics/btv383

Xu, C., Gu, X., Padmanabhan, R., Wu, Z., Peng, Q., DiCarlo, J., & Wang, Y. (2019). smCounter2: An accurate low-frequency variant caller for targeted sequencing data with unique molecular identifiers. *Bioinformatics*, *35*(8), 1299-1309. https://doi.org/10.1093/bioinformatics/bty790

Yang, C., Chu, J., Warren, R. L., & Birol, I. (2017). NanoSim: Nanopore sequence read simulator based on statistical characterization. *GigaScience*, *6*(4), 1-6. https://doi.org/10.1093/gigascience/gix010

Zhang, H. (2016). Overview of Sequence Data Formats. En E. Mathé & S. Davis (Eds.), *Statistical Genomics: Methods and Protocols* (pp. 3-17). https://doi.org/10.1007/978-1-4939-3578-9_1

Zimin, A. V., Marçais, G., Puiu, D., Roberts, M., Salzberg, S. L., & Yorke, J. A. (2013). The MaSuRCA genome assembler. *Bioinformatics*, *29*(21), 2669-2677. https://doi.org/10.1093/bioinformatics/btt476