Trabajo de Fin de Grado

Herramienta bioinformática usando Jupyter para el secuenciador de ADN MinION

Bioinformatic Jupyter notebook for the MinION DNA sequencer

Héctor Rodríguez Pérez

La Laguna, 5 de septiembre de 2016

D. **Marcos Colebrook Santamaría**, con N.I.F. 43.787.808-V, Profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **José L. Roda García**, con N.I.F. 43.356.123-L, Profesor Titular de Universidad adscrito al Departamento Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"Herramienta bioinformática usando Jupyter para el secuenciador de ADN MinION"

ha sido realizada bajo su dirección por D. **Héctor Rodríguez Pérez**, con N.I.F. 51.151.492-K.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 5 de septiembre de 2016.

Agradecimientos

A mi familia, novia y amigos.

A mis tutores, Dr. Marcos Colebrook y Dr. José L. Roda por su ayuda en todo el proceso de desarrollo y elaboración del trabajo.

Y en especial al Dr. Carlos Flores, por hacer posible este innovador proyecto, y que todos esperamos que tenga futuro y largo recorrido.

Licencia

Resumen

Este proyecto nace de la colaboración entre biólogos e ingenieros informáticos de la ULL. El principal objetivo ha sido desarrollar un cuaderno usando Jupyter para trabajar con un novedoso secuenciador de ADN, el MinION. Este cuaderno utiliza algunas de las herramientas disponibles y elaboradas por la comunidad para extraer datos a partir de una muestra secuenciada con el MinION. Los datos originales utilizados para trabajar en este cuaderno han sido aportados por el Dr. Carlos Flores.

El cuaderno, con finalidad docente y de investigación, permitirá a los usuarios del MinION, trabajar de forma sencilla y aprovechar las últimas herramientas de software libre disponibles para este secuenciador de ADN

Palabras clave: Bioinformática, Jupyter Notebook, MinION, ADN.

Abstract

This Project stems from the collaboration between ULL biologists and computer engineers. The main goal has been the development of a notebook using Jupyter which allows working with a new DNA sequencer, the MinION. This notebook uses some available tools developed by the community to extract data from a sequenced sample by the MinION. The original data used to work in this notebook has been provided by Dr. Carlos Flores.

The notebook, with a teaching and research purpose, will allow MinION users to work in an easy way and to take advantage of the latest free software tools available for this DNA sequencer

Keywords: *Bioinformatics, Jupyter Notebook, MinION, DNA.*

Índice General

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

1.1 Bioinformática

La bioinformática, es la aplicación de tecnologías computacionales a la gestión y análisis de datos biológicos. El término hace referencia a un campo de estudio multidisciplinar donde el ingeniero informático realiza tareas de análisis de datos o simulación de sistemas, todas ellas de índole biológica. Más concretamente, los problemas que ha afrontado la bioinformática desde sus inicios en los años 60 hasta hoy en día, suelen ser de nivel molecular.

La capacidad de procesamiento y almacenamiento de datos en la nube y la entrada con fuerza del Big Data en el mundo actual, colocan a la bioinformática como un campo de estudio con repercusión en el porvenir de la sanidad y la investigación. La comunidad científica, es consciente de estos avances y en la actualidad podemos observar el interés por introducir estos conceptos y métodos de trabajo en el ámbito docente, en vistas a un futuro donde los avances de la medicina van necesariamente de la mano del trabajo multidisciplinar con profesionales de la bioinformática.

Centrándonos en el campo de la Genómica (conjunto de disciplinas relacionadas con el estudio de los genomas y sus aplicaciones en terapia génica, biotecnología, etc.), la secuenciación del ADN consiste en la identificación y orden de los nucleótidos A, C, G, T que forman una secuencia de ADN. Este proceso es de gran utilidad para conocer cómo se estructura el ADN no solo humano, sino por ejemplo de bacterias relacionadas con patologías en las personas. Para llevar a cabo este proceso de secuenciación han ido surgiendo distintos métodos y herramientas que se han ido adaptando a las necesidades de los investigadores.

De cara al futuro, la secuenciación y ensamblado del ADN cobra una gran importancia debido a sus numerosas aplicaciones que van desde el diagnóstico de enfermedades de forma más precisa, ponderar el porcentaje que tiene una

persona de sufrir cierta enfermedad conociendo los genes implicados en ésta o el desarrollo de tratamientos específicos y personalizados entre otros usos.

Tomando como ejemplo una bacteria, la secuenciación de su ADN nos da como resultado el orden de sus bases para varios fragmentos, es decir, tenemos distintas lecturas o secuencias, pero aún no conocemos su genoma. El paso siguiente a la secuenciación es el ensamblado. Por medio de este, se toman todas estas secuencias y podemos llegar a obtener finalmente el genoma. Este proceso de ensamblado puede llevarse a cabo partiendo desde cero o también a partir de un genoma de referencia, que puede ser el de la especie que vamos a ensamblar o de una similar. La secuenciación y ensamblado son tareas con bastante actividad en el campo de la bioinformática ya que precisan de la aplicación de algoritmos, sistemas informáticos cada vez más complejos y en vistas al futuro, de una centralización y estructuración de datos masivos acorde con los retos que se plantea la medicina moderna gracias a los avances en el ámbito tecnológico.

1.2 Minion

En la primavera de 2014, la empresa Oxford Nanopore lanzaba el MinION [5], un secuenciador de ADN de coste bajo con respecto a lo que ya existía y más pequeño que un teléfono móvil de última generación, asemejándose bastante a un dispositivo USB.

Para su funcionamiento, el MinION utiliza una pieza desechable llamada *flowcell*, en la que colocamos una muestra de ADN previamente tratada con un químico específico. Una vez preparada la muestra, se conecta a un ordenador y con ayuda de un software, se pone en marcha. A medida que el MinION va obteniendo resultados, estos pueden ser consultados y tratados en tiempo real, otra ventaja frente a sus alternativas. Está también orientado al trabajo fuera del laboratorio, ya que la preparación de la muestra y el procesamiento de los datos que tiene como salida el MinION, tampoco requieren de una gran capacidad computacional gracias al postprocesamiento que se realiza en tiempo real en la nube.

El MinION utiliza una novedosa tecnología para identificar las bases en una secuencia. Su funcionamiento consiste en hacer pasar los distintos fragmentos

contenidos en la muestra de ADN por una serie de poros dispuestos en una membrana por la que circula una pequeña corriente eléctrica. Al pasar por estos poros, el MinION registra una diferencia de potencial por cada base que pasa por cada uno de ellos. El MinION por sí solo identifica nada más que estos cambios en la corriente, a los que llama eventos. Estos se almacenan en unos ficheros que luego deben pasar por una herramienta de nombrado de bases, que toma estos eventos e identifica las bases, elaborando finalmente el fichero con los detalles de la estructura de la secuencia de ADN.

Figura 1.1. Colocación de la muestra en el flowcell del MinION

Para el nombrado de bases, la empresa propone un software que funciona en la nube y a tiempo real, **Metrichor**, y a medida que el MinION trabaja, podemos contar poco a poco con los ficheros finales de salida de las distintas lecturas. Metrichor también filtra las lecturas que va haciendo el MinION y las separa según superen o no un umbral de calidad en dos directorios distintos dentro de la carpeta de salida. Sin embargo, al trabajar en la nube, Metrichor necesita de conexión a internet constante y su funcionamiento exacto es todavía desconocido por la comunidad.

Los ficheros de salida del MinION vienen en fomato **FAST5**, exclusivo de Oxford Nanopore Technologies, una variación del estándar HDF5, un formato de fichero que permite el almacenamiento y manejo de colecciones complejas de metadatos, por lo que al nombrado hay que sumarle un proceso de conversión.

Por tanto, es evidente que para explotar al máximo las posibilidades que ofrece el MinION, son necesarios distintos procesos y herramientas que

permitan trabajar con los datos de salida de una forma más cómoda, y no solo transformar estos en un formato más manejable y estándar como FASTA o FASTQ, sino también extraer tablas y gráficos de forma sencilla, abstrayendo al máximo al usuario del MinION de todo este proceso, ya que podría no contar con los conocimientos de informática necesarios para trabajar con la salida del secuenciador.

1.3 Jupyter

Jupyter [6] Notebook es el proyecto sucesor de **Ipython Notebook**, una aplicación web que permite la creación y distribución de documentos interactivos con código ejecutable y modificable, texto plano en *markdown*, ecuaciones en LaTeX, y visualizaciones de datos o imágenes. Ahora convertido en proyecto Jupyter, se está convirtiendo en una herramienta ampliamente usada en los campos de la ciencia de datos y computación científica debido a la posibilidad de usar otros lenguajes aparte de Python o incluso varios lenguajes simultáneos en un mismo notebook.

Elaborar un cuaderno de Jupyter permite ordenar y organizar por ejemplo una rutina basada en la limpieza y preprocesamiento de una serie de datos, para posteriormente elaborar un modelo estadístico predictivo. Una vez elaborado el cuaderno, podemos ejecutar individualmente o en conjunto las celdas de código que incluyamos, y modificar parámetros de este, viendo cómo se van reflejando los resultados de nuestras modificaciones o adiciones al código.

Los cuadernos de Jupyter también son una gran oportunidad en el campo docente ya que a menudo estas celdas de código van acompañadas de explicaciones a modo de tutorial o referencia. Actualmente, podemos encontrar en la página web oficial del proyecto cuadernos de ejemplo de distintas disciplinas, consistentes en tutoriales o lecciones académicas donde se mezcla teoría en forma de texto y ecuaciones acompañado de celdas de código que se pueden ejecutar o modificar libremente, dotando de un poder interactivo a la lección académica.

Estos cuadernos, están tomando protagonismo en el ámbito de la investigación como una buena forma de presentar resultados y acompañarlos del código correspondiente. Algunos descubrimentos recientes como las ondas

gravitacionales, fueron presentadas en forma de notebook de Jupyter [9]. También en el ámbito de la bioinformática, algunos proyectos actuales relacionados con el virus Zika, están siendo compartidos con la comunidad en forma de notebook [17].

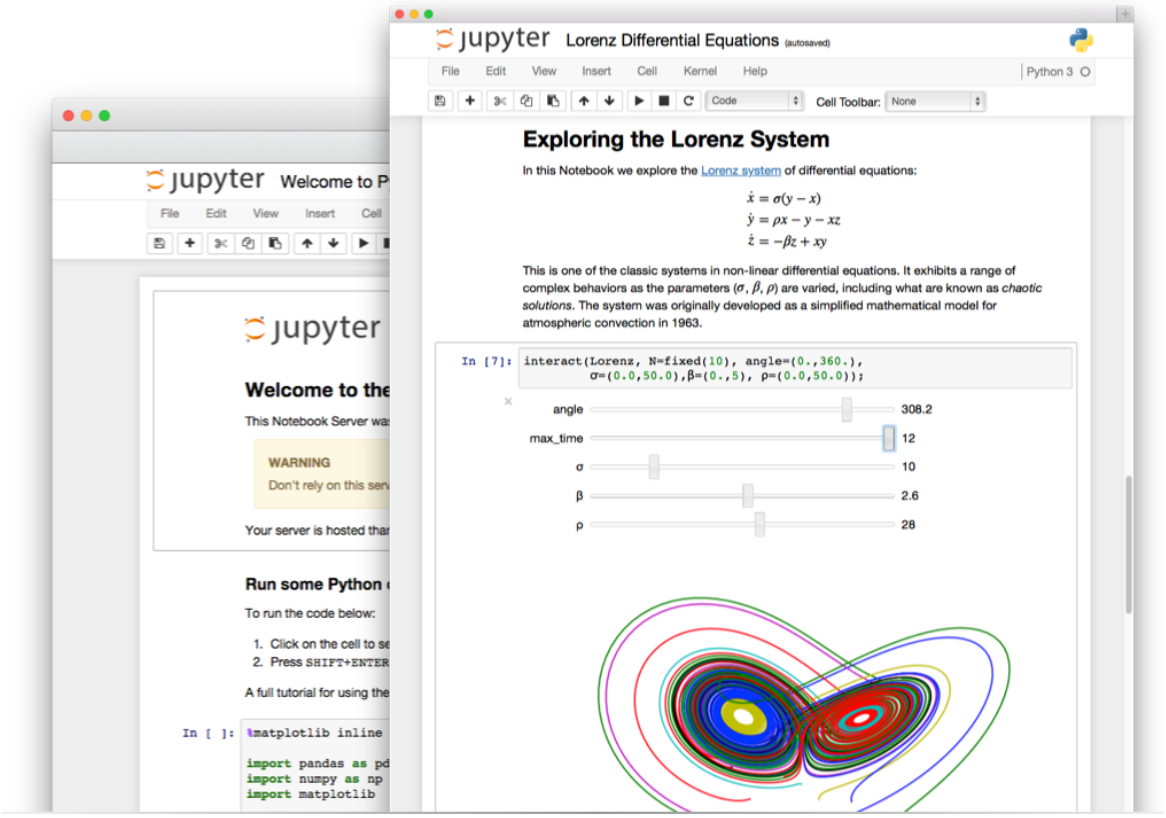

Figura 1.2. El proyecto Jupyter

Actualmente, Jupyter permite no solo el lenguaje Python para crear un cuaderno, sino otros lenguajes ampliamente utilizados en la ciencia de datos y computación científica como R, Scala, Julia y también el uso de *magics* para mezclar distintos lenguajes en un mismo cuaderno Jupyter. En su repositorio, se puede acceder a una lista de *kernels* de lenguajes de programación para la creación de cuadernos.

1.4 Objetivos y requisitos

El objetivo de este trabajo es elaborar un cuaderno (*notebook*) de Jupyter en el que se explique y se muestre cómo trabajar con el MinION de forma sencilla y guiada, utilizando las últimas herramientas desarrolladas por la comunidad. El objetivo es dotar a los usuarios del MinION en un entorno docente o de

investigación de una suite de trabajo en la que además de explicar el funcionamiento de las herramientas y guiar un proceso de análisis tras el uso del MinION, ejemplificarlo con unos datos originales y presentar nuestros resultados.

Utilizaremos una elección de herramientas de uso libre elaboradas por la comunidad específicamente para trabajar con MinION, y enfocadas a trabajar en un entorno libre y sin necesidad de conexión a internet. El cuaderno estará diseñado tanto como herramienta docente como para su uso como cuaderno de trabajo interactivo en entornos de investigación, en el que facilitaremos la personalización a la hora de que el usuario coloque los datos que ha extraído él mismo o elegir qué datos de entre los que podamos extraer se muestren en los gráficos y tablas resultantes del análisis.

El análisis de estos datos no solo se realiza a partir de la salida en crudo del MinION, sino que también incluiremos en nuestro notebook el ensamblado de la bacteria que analizaremos utilizando alguna herramienta de ensamblado por referencia de entre todas las que se ofrecen.

Capítulo 2.
Estado del arte

2.1 Herramientas para MinION

A partir de una muestra de ADN ya tratada y pasada por el MinION y Metrichor, obtenemos una cantidad limitada de información. Utilizando este software, la empresa Nanopore Technologies pasaría también a tener los datos de nuestra muestra, por lo que no serían exclusivamente del usuario. Metrichor devuelve junto a los ficheros de salida con sus bases nombradas un pequeño informe sobre los datos que se le enviaron, pero no tienen un gran valor analítico ni ilustran convenientemente aquellos aspectos que se desearían conocer.

Esta limitación, unida a que un usuario del MinION perteneciente al campo de la biología no tendría por qué tener unos conocimientos avanzados de informática, han impulsado a que la comunidad científica elabore una serie de herramientas para realizar desde las tareas más básicas como la conversión de los ficheros FAST5 a formatos más estandarizados y manejables por la comunidad como FASTA o FASTQ, crear visualizaciones de datos básicas, ensamblar y crear tablas y visualizaciones posteriores al proceso de ensamblado.

En este momento, tenemos a disposición varias herramientas que, aunque cada una esté elaborada en un lenguaje concreto y con algunas prestaciones específicas, tienen una finalidad similar. El objetivo es, por tanto, escoger qué herramientas vamos a utilizar en el cuaderno de entre las disponibles teniendo en cuenta la utilidad de las características exclusivas que tienen, su facilidad de uso, nivel de personalización mediante opciones y también cuya instalación no suponga un problema demasiado complejo debido a las dependencias y requisitos.

2.2 ¿Por qué Jupyter?

Los resultados del análisis de los datos con los que trabajaremos pueden presentarse en forma de tablas en un documento de texto, gráficos y un pequeño informe de cómo se ha llevado a cabo el análisis junto con los scripts correspondientes. Sin embargo, Jupyter se presenta como la plataforma perfecta para organizar todo ese trabajo y de paso, acompañar la parte de código y comandos con las imágenes y texto enriquecido a modo de descripción o guía.

De esta forma, el cuaderno no solo se convierte en una forma sencilla y accesible de presentar los datos que hemos obtenido con el MinION sino también como un documento de uso docente que puede ser utilizado en una clase o distribuido por internet, ya que los cuadernos de Jupyter se están convirtiendo poco a poco en una herramienta básica de trabajo en diversos campos multidisciplinares donde la informática toma protagonismo.

Los paquetes para MinION, vienen en forma de herramientas instalables en el sistema, e invocadas en línea de comandos o mediante código Python. No se trata en ningún caso de grandes cantidades de código en un script sino más bien de una rutina de trabajo en la que debemos ir ejecutando líneas de código determinadas o comandos de forma ordenada y con las opciones correctas o convenientes según las circunstancias de una serie de herramientas concretas que se han escogido para obtener el resultado deseado.

Por tanto, presentar los resultados en un cuaderno Jupyter mostrando el proceso y explicándolo para los usuarios interesados en emularlo, es la forma más elegante de afrontar este trabajo y convierte este TFG en una herramienta más de trabajo con el MinION o de docencia y con una gran facilidad para ser distribuido.

Para instalar y utilizar Jupyter, solamente es necesario tener instalado Python en nuestra máquina. En su documentación oficial, se recomienda instalar Anaconda, una distribución de Python y R que incluye más de 100 paquetes populares relacionados con la ciencia de datos en general incluyendo Jupyter Notebook. Sin embargo, instalar Anaconda provee al usuario de una cantidad de paquetes muy grande, que en muchas ocasiones no aprovechará al máximo, por lo que también se permite instalarlo manualmente.

2.3 Comparativa de herramientas disponibles

Tras una búsqueda de paquetes para MinION tanto en la comunidad oficial del producto como por publicaciones independientes, se han seleccionado las herramientas que a priori tienen bastante potencial para ser utilizadas en el cuaderno.

Se ha realizado esta elección teniendo en cuenta aquellos paquetes que ofrecen tanto funcionalidades esenciales que todo usuario de MinION podría requerir, como herramientas específicas que resuelven un problema concreto o se basan en una funcionalidad específica que ofrecen de forma casi exclusiva. De estas últimas, se escogerán finalmente las que presenten una dificultad de uso no muy grande y en cambio su utilidad se considere interesante.

2.3.1 Deep Nano

Podemos considerar que la mayor desventaja del MinION como secuenciador "portátil" y orientado a trabajar no solo en un laboratorio es su dependencia de conexión a internet para llevar a cabo el proceso de nombrado de bases. Mientras trabaja, el MinION vuelca los resultados de sus lecturas en ficheros FAST5 que solamente almacenan la información de los eventos, es decir, las diferencias de potencial registradas en los poros. Estos ficheros deben ser pasados a la herramienta Metrichor, un software que depende de conexión a internet para completar el trabajo.

Este hecho, sumado a la opacidad de la herramienta, ha contribuido a que una parte de la comunidad trabajase creando herramientas de nombrado de bases alternativas. Estas herramientas funcionan con los ficheros FAST5 que solo contienen los eventos.

Durante la búsqueda de herramientas, se han identificado dos cuya funcionalidad es exclusivamente resolver este problema. Publicadas ambas a finales de marzo de 2016, **Nanocall** [4] y **DeepNano** [3] abarcan este problema de distinta forma pero ambas sin la limitación de la conexión a internet y con la ventaja de ser de código abierto.

Nanocall fue la primera herramienta que realizaba un nombrado de bases alternativas. Bajo licencia MIT, Nanocall utiliza un modelo oculto de Markov

para resolver el problema, y finalmente conseguía un resultado del 68%, similar al de Metrichor.

DeepNano se presenta no solo como una herramienta de nombrado de bases no dependiente de la nube, sino también como una forma de reducir el error cerca de un 2% para lecturas de tipo 2D, y entre el 5% y 6% para lecturas de tipo *Template* y *Complement* frente a Metrichor. Para ello utiliza técnicas de aprendizaje profundo (*Deep Learning*), las cuales están tomando un importante papel en la actualidad gracias a su complejidad y capacidad de clasificación frente a otros modelos predictivos o de clasificación.

DeepNano utiliza una red neuronal recurrente (RNN) para renombrar las bases de los ficheros ya evaluados por Metrichor o para realizar la tarea de nombrado sobre los que extrae el MinION directamente, convirtiéndose en la alternativa más sólida de las dos a escoger. DeepNano está escrita en Python al igual que el resto de herramientas, y construye la red neuronal recurrente con una librería específica para esta tarea (Theano), haciendo uso de librerías para el manejo de ficheros FAST5 (h5py),

Su funcionamiento es sencillo ya que después de haber instalado las dependencias solo basta con invocar DeepNano desde su directorio e indicar dónde se encuentran los ficheros FAST5.

2.3.2 NanoOK

NanoOK [10] ha sido la herramienta elegida para la obtención de gráficas, y datos en general a partir de los ficheros del MinION. Previo a NanoOK, ya existían herramientas más ligeras y básicas que ésta, como **Poretools** o **PoRe**, las cuales realizan tareas básicas como la extracción de ficheros FASTA o FASTQ a partir de los FAST5 del MinION. Estos formatos son mucho más sencillos de manejar y hacían de estas herramientas las idóneas para usuarios con cierta experiencia en el campo de la bioinformática, ya que existen librerías populares para el manejo de este tipo de ficheros con distintas finalidades, y las herramientas de ensamblado funcionan a menudo a partir de ficheros de este tipo.

Sin embargo, NanoOK, incluye desde la funcionalidad de extracción de ficheros FASTA y FASTQ hasta facilidades para luego ensamblar nuestra muestra con distintas herramientas a nuestra elección, para posteriormente,

generar de forma sencilla un informe detallado en PDF con tablas y gráficos con información no solo de las lecturas del MinION sino del ensamblado posterior, permitiendo incluso hacer ensamblados con más de un referencia y volcar estos resultados en tablas.

NanoOK está escrito principalmente en Java, a diferencia de la gran mayoría de herramientas evaluadas, y presenta los resultados analizados en PDF a partir de LaTeX, con ficheros de datos y gráficas en formato de imagen representadas con R que también son accesibles de forma individual aparte del informe. A pesar de ser la herramienta más completa, se convierte no obstante en la que tiene mayor número de dependencias, aunque en la documentación oficial de la herramienta, podemos descargar una imagen de VirtualBox o de Docker, lo que simplifica al máximo el proceso de instalación.

El uso de NanoOK es bastante mecánico cuando se ha entendido su funcionamiento. Una vez ubicados en el directorio principal en el que tenemos los ficheros del MinION, debemos ir ejecutando comandos para primero extraer los ficheros FASTA/FASTQ, escoger el genoma para ensamblado por referencia, ensamblar, y generar el informe. Generar este documento con solo un comando es la principal ventaja de esta herramienta, ya que no hay que ir ejecutando un comando con sus opciones por cada tabla o por cada gráfica y además no hay otra que ofrezca tanta información

2.3.3 LAST

NanoOK ofrece varias herramientas a la hora de ensamblar, proceso necesario para generar el informe con toda la información. En su documentación se señalan las opciones disponibles, que son **LAST** [8], **BLASR** [2], **BWA-MEM** [11] y **MarginAlign** [12]. Independientemente de la que elijamos, el documento generado tiene las mismas secciones y se presupone un resultado similar o muy parecido. En este caso hemos escogido LAST que es el que se marca como ensamblador por defecto. La herramienta escogida, debe ser instalada previamente por separado a NanoOK.

A la hora de ensamblar, el usuario debe teclear un comando de NanoOK con algunas opciones y en caso de querer usar otro ensamblador que no sea LAST, debe indicarlo. Podemos decir entonces que para el proceso de ensamblado, NanoOK funciona simplemente como puente y abstrae al usuario del

funcionamiento específico de cada herramienta, es decir, en caso de tener más de una instalada y querer utilizar una determinada según la situación, no es necesario aprender a utilizar cada una de ellas, sino que simplemente hay que indicar cuál de ellas va a ensamblar. NanoOK hace el resto del trabajo.

LAST es un ensamblador por referencia cuyo funcionamiento se basa en encontrar regiones similares entre secuencias y ensamblarlas. En la documentación se destaca su capacidad para trabajar con conjuntos de datos grandes comparándolos entre sí (genomas de vertebrados o un gran número de lecturas de ADN).

Al contrario que un ensamblador *de novo*, que sin ninguna referencia basa su funcionamiento en encontrar secuencias cuyo final coincida con el principio de otra de forma que se puedan unir para formar fragmentos mayores hasta completar el genoma, LAST necesita de una referencia para ensamblar. Es decir, debemos indicarle previo al proceso de ensamblado, un genoma en formato FASTA a partir del cual LAST realiza las comparaciones entre secuencias y referencia, y luego ensambla.

En este tipo de ensamblado, el genoma de referencia no tiene por qué ser exactamente el mismo que se espera tras el ensamblado de nuestra muestra, pero sí conviene que sea similar, ya que de la elección depende la calidad del ensamblado final. NanoOK trabaja solamente con herrramientas de ensamblado por referencia, ya que permite generar el documento de análisis tras varios ensamblados con distintas referencias, aportando tablas comparativas y gráficas de los resultados de cada ensamblado.

LAST no funciona solamente siendo ejecutado vía NanoOK. También cuenta con un servicio online accesible desde su página web oficial, que realiza exactamente el mismo proceso que la aplicación instalable. Esta última, que ha sido la opción escogida y necesaria para funcionar con NanoOK, se puede descargar desde el mismo sitio web y se compila en nuestro equipo.

La siguiente Tabla 2.1 muestra un resumen de la comparación de las características más importantes de las herramientas analizadas. Como referencia, comentar que en este TFG se han utilizado DeepNano y NanoOK.

Funcionalidad	**NanoCORR**	**MinoTour**	**MarginAlign**	**DeepNano**	**Nanocall**	**Nanopolish**	**Poremap**	**NanoOK**	**Pore**	**Poretools**
Conversión a FASTA/FASTQ	x							x	x	x
Gráficas básicas								x	x	x
Generación de tablas		x						x		
Ensamblado	x					x	x	x		
Ensamblado multirreferencia								x		
Corrección de errores	x					x	x			
Nombrado de bases			x	x	x					

Tabla 2.1. Tabla comparativa de las herramientas para el MinION

Capítulo 3. Diseño y desarrollo de la solución

3.1 Requisitos para el diseño del notebook en Jupyter

Una vez vistas las herramientas disponibles para MinION y la evaluación y elección de estas, se han definido una serie de requisitos que debe cumplir el *notebook* de Jupyter.

Se deberá trabajar con los ficheros de salida del MinION en crudo, es decir, la misma estructura de directorios y ficheros que se obtienen como resultado de una rutina de trabajo con el MinION.

Aprovechando la posibilidad de poder usar un software libre y sin dependencia de la nube y aprovechando la mejoría en la precisión que esta herramienta ofrece frente a Metrichor, utilizaremos este método previo a cualquier análisis. Tendremos también a disposición del usuario del cuaderno, las lecturas nombradas con Metrichor, ofreciendo la posibilidad de, con pequeños cambios en el código, ver rápidamente los datos y gráficas de los análisis de las lecturas con Metrichor y DeepNano.

Aprovechando las características de Jupyter, acompañaremos la rutina de trabajo con las herramientas seleccionadas con explicaciones sobre el funcionamiento de las herramientas y sus diferentes opciones, convirtiendo el cuaderno en la suite de trabajo que señalábamos al principio, ofreciendo no solo una metodología de trabajo guiada sino también una referencia rápida para el usuario que desee modificar parámetros de las herramientas.

A pesar de que NanoOK genere un PDF con los resultados del análisis completo, se intentará que el usuario no tenga que salir del notebook para ver los resultados. Aparte del análisis en LaTeX y PDF, tras la ejecución del

comando que genera el documento, se crea una estructura de directorios que contiene los ficheros de datos utilizados para elaborar las tablas del documento y también las gráficas en PNG que se incluyen. Aprovecharemos esta situación para, con la colaboración del Dr. Carlos Flores, escoger las partes más interesantes y significativas del análisis que ofrece NanoOK para volcarlas en el cuaderno de nuevo de forma interactiva, permitiendo mostrar unos datos u otros según se hayan seleccionado los parámetros en las celdas de código.

3.2 Formato y estructura de los datos de salida del MinION

El análisis de datos de secuencias de nucleótidos, como las que genera el MinION suele realizarse sobre ficheros de formato FASTQ y FASTA. Estos ficheros suelen contener una o más entradas donde cada una corresponde a una secuencia.

En el formato **FASTQ**, cada entrada usa normalmente cuatro líneas. La primera como identificador, la segunda con los caracteres de la secuencia, la tercera con un caracter '+' seguido opcionalmente del identificador y descripción, y la cuarta con la codificación en símbolos de la calidad de la secuencia de la segunda línea (un símbolo ASCII por cada letra de la secuencia).

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

El formato **FASTA**, utilizado por NanoOK, consiste básicamente en entradas con una línea de cabecera seguida de las líneas con la secuencia. La cabecera se distingue de la secuencia ya que comienza con el símbolo '>' seguido de un identificador y a menudo con información adicional.

```
>AB000263 |acc=AB000263|descr=Homo sapiens mRNA for prepro
cortistatin like peptide, complete cds.|len=368
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCC
TGCCCCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGA
AAAGCAGCCTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCCCC
TCATAGGAGAGGAAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGC
GCCGGGACAGAATGCCCTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCC
CCATGAATGCTCACGCAAGTTTAATTACAGACCTGAA
```

Para afrontar la fase experimental, y debido a que en ese momento no se contaba con la muestra original que se utilizaría para ilustrar el cuaderno en el TFG, se necesitaba una muestra de prueba. En la documentación de NanoOK, se encuentra disponible para su descarga un conjunto de datos pertenecientes a una prueba realizada con una bacteria E. Coli DH10B. Tomamos esta muestra en su versión minimizada, consistente en 500 lecturas y con un tamaño de unos 415 MB, en comparación con los 3,6 GB que tenía la muestra original.

Entre los ficheros obtenidos tras una ejecución del MinION, destacan dos directorios, en los que encontramos los ficheros con las lecturas. Estos son:

- ***downloads***: En el interior de este directorio tenemos los ficheros FAST5 que se obtienen tras el nombrado de bases con Metrichor divididos en dos directorios **pass** y **fail**. Dentro de cada uno de estos, se encuentran los directorios **template**, **complement** y **2D**. Metrichor, aparte de realizar el nombrado de bases, divide las lecturas originales que va recibiendo según pasan o no un umbral de calidad. Con respecto a este umbral, no se ha encontrado nada de información acerca del criterio que usa para separar las lecturas.
- ***uploaded***: Este directorio no estaba incluido en los datos de prueba que ofrece NanoOK. En su interior se encuentran todos los ficheros FAST5 extraídos del MinION sin haber pasado por Metrichor. Por esta razón no están separados según su calidad como ocurre con el directorio *downloads*. Aparentemente estos datos no tendrían ninguna utilidad para el usuario si ya cuenta con las lecturas del directorio *downloads* a no ser que utilice una

herramienta de nombrado de bases alternativa para comenzar a analizar las lecturas.

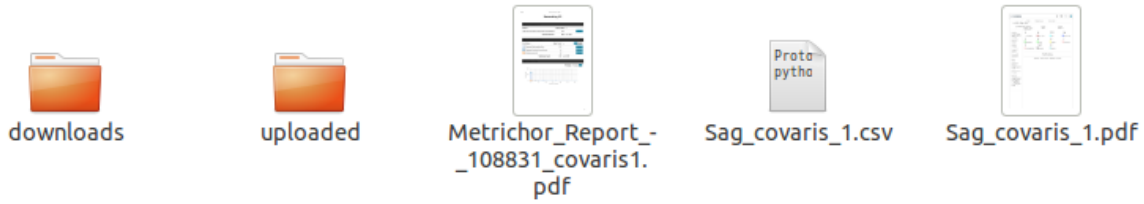

Figura 3.1. Estructura de los directorios y ficheros de una ejecución del MinION.

Los datos vienen organizados como lecturas en formato FAST5. El número de lecturas que encontramos en estos directorios depende del tiempo que ha estado el MinION trabajando. Desde la página oficial [5] se indica que puede estar en funcionamiento desde minutos hasta incluso dos días, dependiendo de la naturaleza del experimento a realizar.

Para comenzar a trabajar con estos datos, ya sea con herramientas específicas para MinION como con otras más generales de bioinformática, se hace necesario extraer las secuencias a ficheros FASTA o FASTQ. **Poretools** fue una de las primeras y más populares herramientas para este propósito. Aunque utilizaremos NanoOK, que ya incluye esta funcionalidad, en su documentación se explica también cómo realizar ensamblados y análisis para usuarios de Poretools. Esta herramienta toma cada uno de los ficheros FAST5 situados en el directorio indicado como ruta en la línea de comandos y genera un único fichero FASTA con las respectivas entradas a los ficheros. Por esta razón, al instalar NanoOK, se incluye un pequeño script que divide el fichero generando un FASTA por cada FAST5 a partir del cual se realizó la conversión. Esta organización de los ficheros FASTA es necesaria para realizar cualquier trabajo posterior con NanoOK.

En el *notebook* se permite trabajar con los datos procesados por Metrichor o con los que nombraremos con DeepNano variando parámetros en los comandos o fragmentos de código. Con los primeros no es necesario ningún preprocesamiento extra pero cuando decidimos hacer uso de DeepNano debemos reorganizarlos antes de seguir trabajando. La razón es que al ejecutar DeepNano con los datos del directorio *uploaded* y nombrar los FAST5 crudos del MinION, se genera un único fichero FASTA, al igual que ocurría con

Poretools. Como ventaja tenemos que no es necesaria la extracción de los FASTA con ninguna herramienta tras utilizar DeepNano, pero debemos dividirlo si queremos trabajar con NanoOK. Para esto aprovecharemos el *script* que se incluía como solución a los usuarios de Poretools.

3.3 Notebook generado

Finalmente, elaboramos el notebook con las herramientas escogidas y la organización correspondiente de los datos.

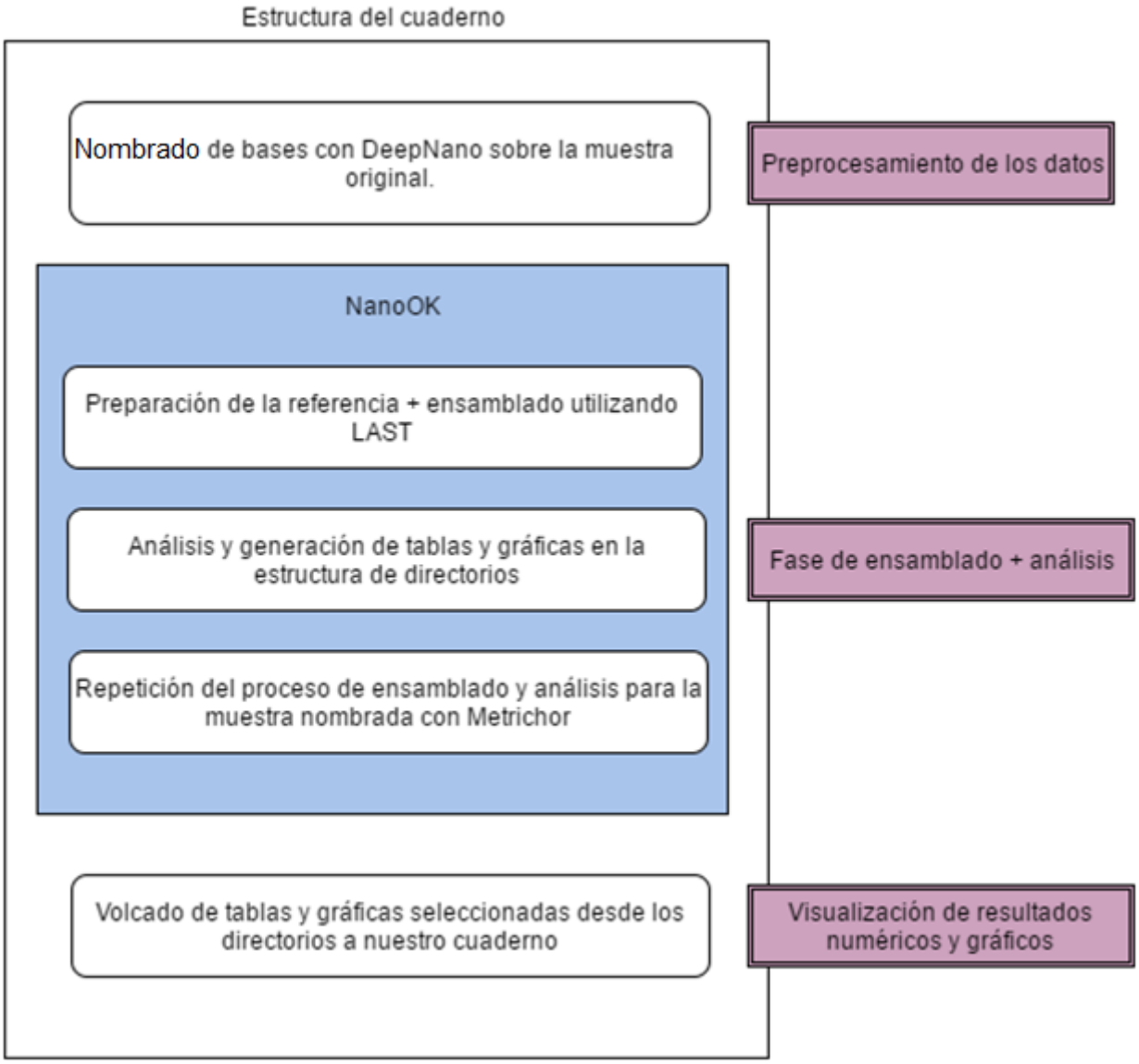

Figura 3.2. Esquema del notebook generado

El fichero del notebook se encuentra ubicado junto a dos directorios donde colocamos las lecturas de Metrichor y las lecturas sin nombrar. Los denominamos **metrichor_reads** y **reads**, respectivamente. En **metrichor_reads** tenemos el directorio FAST5 en el que encontramos las lecturas divididas en *pass* y *fail*, mientras que en **reads** tenemos el directorio *uploaded* con las lecturas sin nombrar. Esta estructura es únicamente orientativa y recomendable para una vez realizado los ensamblados y análisis, tener los ficheros de salida debidamente organizados para su consulta.

3.3.1 Nombrado de bases

El objetivo de esta fase es el de nombrar las bases de los ficheros de *reads*. Debido a que para el análisis necesitamos nombrar las secuencias *Template*, *Complement* y *2D* del ADN, debemos ejecutar DeepNano tres veces sobre los mismos ficheros, pero variando el tipo de secuencias que queremos con las opciones de la herramienta. Señalamos también el nombre del fichero FASTA de salida que luego necesitaremos dividir.

Una vez nombrados los ficheros, utilizamos el *script* **nanook_split_fasta** incluido en NanoOK sobre cada salida de DeepNano para generar un FASTA por cada secuencia incluida en el fichero. Es importante haber creado un árbol de directorios similar al que genera Metrichor (directorios *pass* y *fail* con subdirectorios *Template*, *Complement* y *2D*) y volcar la salida del script en la ubicación correspondiente. Esto se hace porque NanoOK está hecho para trabajar con esta estructura de directorios asumiendo que el usuario trae sus datos tras haber usado el MinION directamente.

Aunque pueda parecer un obstáculo crear esta estructura de ficheros si queremos trabajar con DeepNano, que NanoOK trabaje de esta manera se convierte en una ventaja gracias a todos los datos que luego extrae de nuestros ficheros con unos pocos comandos sencillos que por ejemplo nos ahorran conocer las instrucciones de uso de la herramienta de ensamblado. En el comando de llamada al *script*, observamos que volcamos los resultados siempre en el directorio *pass* de las lecturas, dejando *fail* vacío.

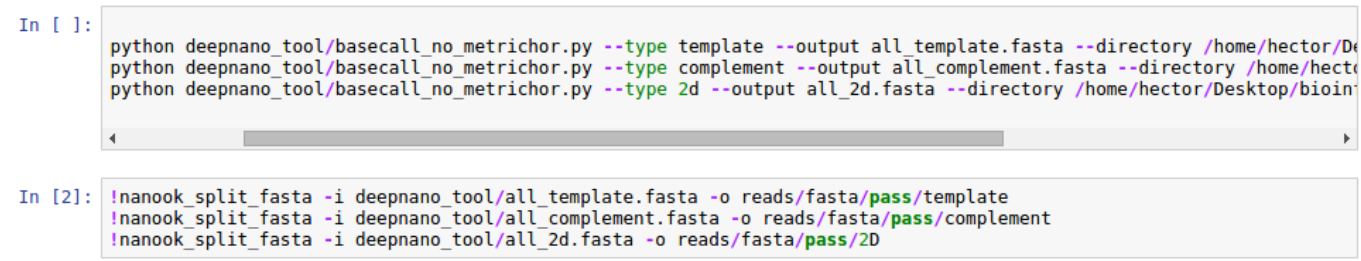

Figura 3.3. Ejemplo de ejecución usando DeepNano y NanoOK

Metrichor separaba las lecturas según su calidad pero al no estar utilizando este software, tratamos a todas las lecturas por igual y las tomaremos como "buenas". La única forma de respetar la clasificación que hace Metrichor, sería utilizar DeepNano sobre las lecturas ya separadas y nombradas por Metrichor. Esto también es posible ya que la herramienta tiene la opción de renombrar los ficheros FAST5 de salida de Metrichor aparte de nombrar la salida en crudo del MinION. Sin embargo, para este trabajo ilustramos el caso en el que solo se utilizan **herramientas de código abierto** y **sin necesidad de conexión a internet**.

3.3.2 Análisis de los datos con NanoOK

Una vez tenemos los ficheros nombrados con DeepNano y organizados debidamente podemos comenzar el análisis con NanoOK. Para generar el PDF junto con las gráficas y datos es necesario llevar a cabo un proceso de ensamblado previo. Como se indicó anteriormente, se ha instalado la herramienta LAST por separado, que es el ensamblador por defecto que utiliza NanoOK. Esto nos ahorra indicar mediante opciones la herramienta a utilizar, y solo necesitamos indicar la referencia y algunas opciones básicas para ensamblar.

Previo a este proceso de ensamblado, es necesario indexar la referencia. LAST necesita una referencia a partir de la cual ensambla y para indicar la que vamos a utilizar, es necesario hacerlo con un comando propio y no de NanoOK.

Los genomas de referencia pueden encontrarse en bases de datos accesibles para todo el mundo. Es importante escoger una buena referencia o al menos escoger de forma correcta aquella que deseamos utilizar, ya que el programa intentará ensamblar como pueda nuestras secuencias a partir de la referencia dada. En el caso de equivocarnos de referencia, seguimos obteniendo un

resultado con el que también se genera un documento, aunque con menos información útil o, en el peor de los casos, con información incorrecta.

Tras el período de pruebas utilizando un conjunto de datos minimizado obtenido en la página de documentación de NanoOK, trabajaremos con otro conjunto muestra esta vez original y facilitada por el Dr. Carlos Flores.

Los datos originales proporcionados corresponden a un aislado clínico de ***Streptococus agalactiae*** de la que se conoce hasta el momento que pertenece a un multilocus tipo ST1, un subtipo concreto de esa bacteria. Brevemente, dichas secuencias fueron obtenidas en el laboratorio del Dr. Carlos Flores en una carrera de 48 h utilizando un *flowcell* tipo R7 en un MinION mk 1.0 con la versión SQK-MAP006 de reactivos de secuenciación a partir de fragmentación de 40 ng de ADN generados mediante centrifugación en g-Tubes (Covaris) siguiendo protocolos recomendados por Oxford Nanopore Technologies. El Dr. Flores también indicó el lugar donde encontrar la referencia más cercana genéticamente en base a observaciones previas realizadas sobre el mismo aislado clínico (manuscrito en revisión).

En nuestro caso, ya teníamos nuestra referencia (el genoma en formato FASTA) en el directorio *reference*. Desde Jupyter no solo podemos invocar programas ya instalados en nuestra máquina sino que también podemos utilizar comando básicos como los de navegación entre directorios.

Una vez indexada la referencia ya podemos ensamblar. Aunque hayamos indexado la referencia, debemos volver a indicar mediante línea de comandos, dónde se encuentra. En el caso de la muestra, el proceso de ensamblar no consume demasiado tiempo. Aunque hayamos usado DeepNano y no tengamos las lecturas separadas según su calidad, la herramienta utiliza no solo aquellas lecturas dadas por buenas según Metrichor, sino que utiliza absolutamente todas las lecturas.

Este proceso indexar-ensamblar puede repetirse varias veces para la misma muestra, cambiando la referencia. NanoOK guarda todos los resultados con el fin de elaborar una comparativa en el análisis. Esta funcionalidad es única en esta herramienta, y una de las destacadas en el artículo con el que se presenta NanoOK [9].

```
In [35]: %cd /home/hector/Desktop/bioinformatics/agalactiae/reference
         !lastdb -Q 0 agalactiae_reference agalactiae_reference.fasta
         %cd ..

In [36]: !nanook align -s reads -r reference/agalactiae_reference.fasta
```

Figura 3.4. Comando de NanoOK de alineamiento de las lecturas usando la secuencia de referencia

Una vez finalizado el ensamblado, en el directorio desde el que lanzamos el *notebook* aparecen nuevos directorios y ficheros correspondientes al ensamblado. Sin embargo, para empezar a ilustrar los resultados, aprovecharemos la principal funcionalidad de NanoOK para extraer datos a partir del MinION.

La función ***analyse*** de NanoOK genera un documento PDF con un análisis de unas 9 páginas normalmente. El documento se crea primero en LaTeX y luego se renderiza a PDF. Para las gráficas, NanoOK utiliza R. Utilizando la opción ***-bitmaps***, generamos también los PNG de las gráficas renderizadas en el documento. Aunque podríamos terminar aquí, podemos aprovechar las funcionalidades de Jupyter para, aprovechando toda la información generada por NanoOK, visualizar la más importante en el cuaderno.

Para dotar al *notebook* de un poder interactivo extra, repetiremos el proceso de ensamblado y análisis pero esta vez con las lecturas de salida de Metrichor. Tendremos entonces dos análisis distintos para la misma muestra, y a continuación podremos ilustrar las mismas gráficas e información para uno y para otro solo cambiando parámetros en líneas de código.

```
In [38]: !nanook extract -s metrichor_reads
         !nanook align -s metrichor_reads -r reference/agalactiae_reference.fasta
         !nanook analyse -s metrichor_reads -r reference/agalactiae_reference.fasta -bitmaps
```

Figura 3.5. Proceso de ensamblado y análisis de NanoOK usando las lecturas de Metrichor

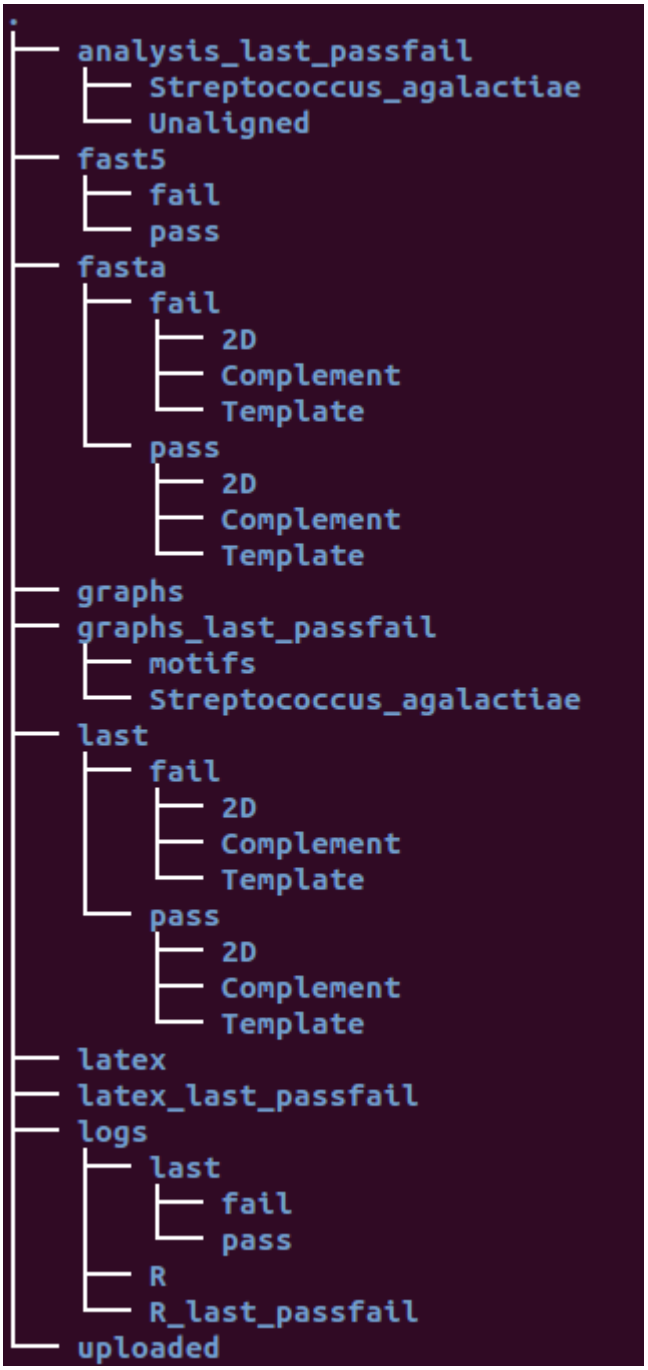

Figura 3.6. Estructura de los directorios después del proceso de análisis

3.3.3 Volcado de los resultados de forma interactiva en el notebook

Con ayuda del Dr. Carlos Flores, seleccionamos la información más significativa que se incluye en el documento. Una vez realizado el comando de análisis para una de las muestras, se crean junto al directorio que almacena los FASTA tres directorios con la información del análisis que son los que contienen tanto el documento PDF como los ficheros de texto con datos y los PNG de las

gráficas que se renderizan en este. Para volcar el contenido del análisis en el cuaderno, debemos conocer el contenido de cada uno de los tres directorios:

- **analysis_last_passfail**: Encontramos ficheros TXT con información tanto de las secuencias como del ensamblado realizado. Algunos de estos ficheros contienen tablas que se vuelcan directamente en el documento PDF. También hay un subdirectorio *unaligned* con información de las secuencias que no se ensamblaron y un directorio por cada ensamblado realizado con la misma muestra que tiene como nombre la referencia utilizada en cada caso.
- **latex_last_passfail**: Aquí se guarda el documento PDF que se genera como resultado. Aparte, se incluyen otros ficheros como el *.tex* (formato de LaTeX) a partir del cual se renderiza el PDF y el log del proceso.
- **graphs_last_passfail**: Este directorio tiene una estructura bastante similar al de **analysis_last_passfail** pero en lugar de contener ficheros TXT, contiene las gráficas en formato PNG. La mayoría de estas están ubicadas en los subdirectorios con los nombres de las referencias utilizadas para los ensamblados (en el caso de nuestra muestra, solo una). Además, estas visualizaciones están a tamaño completo y por tanto su nivel de detalle es mucho mayor con respecto al documento, ya que en este se incluyen todas las gráficas de este directorio, en muchos casos en unas proporciones muy reducidas.

Tomando como ejemplo el primer conjunto de datos que deseamos visualizar en el cuaderno, relacionado con las longitudes de las lecturas extraídas, se puede observar que por cada tipo de tipo de lectura (*Template*, *Complement* y *2D*) tenemos una tabla y una gráfica. Además, como la finalidad es poder visualizar los datos tanto de las lecturas de DeepNano como las de Metrichor, estas cifras se doblan.

Con el fin de permitir que el usuario escoja qué información ver, podemos insertar las imágenes y tablas no como texto plano en markdown sino como el resultado de la ejecución de código Python. Utilizando la librería de IPython (antiguo nombre de Jupyter Notebook) podemos mostrar imágenes que tengamos en nuestra máquina como si estuviéramos renderizando por ejemplo una visualización de datos en **matplotlib**, es decir, viendo la imagen como resultado de la ejecución del código Python. De esta forma, y señalando en el

notebook al usuario qué parámetros puede cambiar y qué opciones tiene, podemos visualizar en cualquier caso la tabla y gráfica correspondiente al tipo de lectura y herramienta de nombrado de bases que quiera.

Como el contenido de todas las celdas de un notebook de Jupyter están en el mismo ámbito, creamos al principio de la fase de muestra de resultados dos variables en las que seleccionamos las lecturas de Metrichor o DeepNano y los tipos *Template*, *Complement* o *2D*, respectivamente. Cuando ejecutamos todas las celdas de código desde esta declaración en adelante, mostramos siempre los datos pertenecientes a los tipos que colocamos en esas variables.

In [2]:
```
from IPython.display import Image

#sample puede ser "reads" (con el basecaller DeepNano) o "metrichor_reads" (basecaller Metrichor)
sample = "reads"
#read_type puede ser "Template", "Complement" o "2D"
read_type = "Complement"

f = open(sample + "/analysis_last_passfail/length_summary.txt")
print(f.read())

Image(filename = sample + '/graphs_last_passfail/all_' + read_type + '_lengths.png')
```

```
Nanotools report - reads

Length summary

Type       NumReads TotalBases Mean       Long     Short    N50      N50Count N90      N90Count
Template   551      2133402    3871.87    24904    2        6219     122      2491     327
Complement 243      1340416    5516.12    48832    67       6538     71       3244     180
2D         444      851139     1916.98    8978     0        4901     64       1650     170
```

Figura 3.7. Ejemplo de código para mostrar tablas y gráficas desde Jupyter

La tabla perteneciente al análisis del error del ensamblado de nuestra muestra es uno de los datos escogidos como significativos para mostrar en el notebook. Sin embargo, esta tabla no figura en formato TXT en el directorio de análisis generado. Se presupone entonces que esta tabla es el resultado de una serie de cálculos a partir de algunos de los ficheros que hay en el directorio y que no aparecen en el documento. La solución para mostrar estos datos desconociendo los cálculos a realizar sería extraer la tabla del documento PDF.

Para mostrar los datos en el cuaderno, es necesario hacer esta extracción con ayuda de algún módulo específico para Python, como **pyPDF** o **PDFMiner**. Tras haber probado estas dos herramientas, no se logró extraer correctamente la tabla. Aún seleccionando la página exacta donde está el contenido, el resultado es el texto de la tabla pero sin ningún formato, incluso ignorando espacios, tabulado o saltos de línea.

Sabiendo que el documento es un renderizado a partir del fichero LaTeX y que este fichero es accesible, se intentó entonces aprovechar la capacidad de Jupyter para escribir ecuaciones LaTeX. Localizado el fragmento que contiene la tabla en el fichero *.tex* no se logró tampoco que apareciera con el formato correcto ni con Jupyter ni tampoco con módulos de Python.

La solución ha sido finalmente elaborar un **script de Python** que a partir solo de expresiones regulares busca en el fichero LaTeX la tabla, luego almacena los números para, posteriormente, conociendo los identificadores de las filas y columnas, imprimir como resultado la tabla final. Este *script* funciona para cualquier muestra siempre que se le pase como argumento el directorio principal de ésta, y que además esté ubicado en el mismo directorio que el cuaderno.

```
In [14]: %run view_table.py metrichor_reads
```

	Template	Complement	2D
Overall base identity (excluding indels)	60.45%	65.36%	70.45%
Aigned base identity (excluding indels)	74.37%	76.55%	82.39%
Identical bases per 100 aligned bases (including indels)	60.80%	62.33%	70.20%
Inserted bases per 100 aligned bases (including indels)	4.18%	3.65%	4.58%
Deleted bases per 100 aligned bases (including indels)	14.07%	14.91%	10.22%
Substitutions per 100 aligned bases (including indels)	20.95%	19.10%	15.00%
Mean insertion size	1.51	1.46	1.51
Mean deletion size	1.81	1.87	1.67

Figura 3.8. Ejemplo de visualización de la tabla de resultados usando el *script* de expresiones regulares

Capítulo 4. Resultados

A continuación, se muestran los resultados que podemos visualizar en el notebook tal y como fueron obtenidos para el aislado *S. agalactiae*. El análisis realizado por la herramienta NanoOK también aporta más tablas y gráficas, que quedan almacenadas en los directorios que se crean tras el análisis, además de las seleccionadas con ayuda del Dr. Carlos Flores.

4.1 Resultados numéricos

4.1.1 Resultados de la muestra nombrada con DeepNano.

En estas primeras tablas, figuran los datos relacionados con las lecturas obtenidas del MinION. Para cada tipo, en cada columna podemos observar el número de lecturas (*NumReads*), el número total de bases (*TotalBases*), la longitud media (*Mean*), más larga (*Longest*), más corta (*Shortest*), el indicador de calidad del ensamblado N50, el número de secuencias con longitud mayor o igual a N50 (*N50Count*), el indicador N90, el número de secuencias con longitud mayor o igual a N90 (*N90Count*).

Type	NumReads	TotalBases	Mean	Longest	Shortest	N50	N50Count	N90	N90Count
Template	551	21334002	3871.87	24904	2	6219	122	2491	327
Complement	243	1340416	5516.12	48832	67	6538	71	3244	180
2D	444	851139	1916.98	8978	0	4901	64	1650	170

Tabla 4.1. Tabla con los *Read Lengths* (longitudes de las lecturas) con DeepNano

Las siguientes tablas contienen datos relacionados con el ensamblado de *S. agalactiae* utilizando como referencia un subtipo ST1 (Número de acceso en el NCBI: www.ncbi.nlm.nih.gov/nuccore/827409863), con información para cada tipo de lectura (*Template*, *Complement* y *2D*). Al realizar solo un ensamblado con una referencia, solo tenemos una fila (*Streptococcus agalactiae*) con información acerca del ensamblado.

Number of reads	551	
Number of reads with alignments	235	(42.65%)
Number of reads without alignments	316	(57.35%)

Tabla 4.2. Tabla con los *Template alignments* (alineamiento de las secuencias) con DeepNano

ID	**Size**	**Number of Reads**	**% of Reads**	**Mean read length**	**Aligned bases**	**Mean coverage**	**Longest Perf Kmer**
Streptococcus agalactiae	2092071	235	42.65	5965.70	1428884	0.68	38

Tabla 4.3. Tabla con los datos de la referencia del *S. agalactiae* (lecturas tipo *Template*) con DeepNano

Number of reads	243	1.
Number of reads with alignments	167	(68.72%)
Number of reads without alignments	76	(31.28%)

Tabla 4.4. Tabla con los datos de los *Complement alignments* (alineaciones de la secuencia complemento) con DeepNano

ID	Size	Number of Reads	% of Reads	Mean read length	Aligned bases	Mean coverage	Longest Perf Kmer
Streptococcus agalactiae	2092071	167	68.72	5903.52	862960	0.41	48

Tabla 4.5. Tabla con los datos de la referencia del *S. agalactiae* (lecturas tipo *Complement*) con DeepNano

Number of reads	444	
Number of reads with alignments	236	(53.15%)
Number of reads without alignments	208	(46.85%)

Tabla 4.6. Tabla con los *2D alignments* (alineamiento de secuencias *2D*) con DeepNano

ID	Size	Number of Reads	% of Reads	Mean read length	Aligned bases	Mean coverage	Longest Perf Kmer
Streptococcus agalactiae	2092071	236	53.15	2961.07	765164	0.37	66

Tabla 4.7. Tabla con los datos de la referencia del *S. agalactiae* (lecturas tipo *2D*) con DeepNano

2.	**Template**	**Complement**	**2D**
Overall base identity (excluding indels)	65.11%	55.87%	77.17%
Aligned base identity (excluding indels)	76.07%	76.32%	83.98%
Identical bases per 100 aligned bases (including indels)	63.88%	63.82%	70.48%
Inserted bases per 100 aligned bases (including indels)	5.01%	4.03%	2.83%
Deleted bases per 100 aligned bases (including indels)	11.01%	12.35%	13.24%
Substitutions per 100 aligned bases (including indels)	20.09%	19.80%	13.45%
Mean insertion size	1.47	1.35	1.32
Mean deletion size	1.64	1.73	1.69

Tabla 4.8. Tabla con los datos del análisis de alineamiento con respecto a la referencia del *S. agalactiae* con DeepNano

4.1.2 Resultados de la muestra nombrada con Metrichor.

A continuación, se muestran las mismas tablas que en el apartado anterior, pero referente a las lecturas y ensamblado de la muestra cuyas bases fueron nombradas por Metrichor.

Type	**NumReads**	**TotalBases**	**Mean**	**Longest**	**Shortest**	**N50**	**N50Count**	**N90**	**N90Count**
Template	517	1994959	3858.72	37409	11	6358	109	2583	295
Complement	297	1251282	4213.07	14212	16	6134	76	2901	186
2D	253	1280235	5060.22	17647	131	6630	71	3413	175

Tabla 4.9. Tabla con los *Read Lengths* (longitudes de las lecturas) con Metrichor

Number of reads	517	
Number of reads with alignments	193	(37.33%)
Number of reads without alignments	324	(62.67%)

Tabla 4.10. Tabla con los *Template alignments* (alineamiento de las secuencias) con Metrichor

ID	**Size**	**Number of Reads**	**% of Reads**	**Mean read length**	**Aligned bases**	**Mean coverage**	**Longest Perf Kmer**
Streptococcus agalactiae	2092071	193	37.33	5774.22	1108164	0.53	49

Tabla 4.11. Tabla con los datos de la referencia del *S. agalactiae* (lecturas tipo *Template*) con Metrichor

Number of reads	297	
Number of reads with alignments	155	(52.19%)
Number of reads without alignments	142	(47.81%)

Tabla 4.12. Tabla con los datos de los *Complement alignments* (alineaciones de la secuencia complemento) con Metrichor

ID	**Size**	**Number of Reads**	**% of Reads**	**Mean read length**	**Aligned bases**	**Mean coverage**	**Longest Perf Kmer**
Streptococcus agalactiae	2092071	155	52.19	5354.63	870262	0.42	46

Tabla 4.13. Tabla con los datos de la referencia del *S. agalactiae* (lecturas tipo *Complement*) con Metrichor

Number of reads	253	
Number of reads with alignments	184	(72.73%)
Number of reads without alignments	69	(27.27%)

Tabla 4.14. Tabla con los *2D alignments* (alineamiento de secuencias *2D*) con Metrichor

ID	**Size**	**Number of Reads**	**% of Reads**	**Mean read length**	**Aligned bases**	**Mean coverage**	**Longest Perf Kmer**
Streptococcus agalactiae	2092071	184	72.73	5651.75	1043595	0.50	113

Tabla 4.15. Tabla con los datos de la referencia del *S. agalactiae* (lecturas tipo *2D*) con Metrichor

3.	**Template**	**Complement**	**2D**
Overall base identity (excluding indels)	60.45%	65.36%	70.45%
Aligned base identity (excluding indels)	74.37%	76.55%	82.39%
Identical bases per 100 aligned bases (including indels)	60.80%	62.33%	70.20%
Inserted bases per 100 aligned bases (including indels)	4.18%	3.65%	4.58%
Deleted bases per 100 aligned bases (including indels)	14.07%	14.91%	10.22%
Substitutions per 100 aligned bases (including indels)	20.95%	19.10%	15.00%
Mean insertion size	1.51	1.46	1.51
Mean deletion size	1.81	1.87	1.67

Tabla 4.16. Tabla con los datos del análisis de alineamiento con respecto a la referencia del *S. agalactiae* con Metrichor

4.2 Resultados gráficos

4.2.1 Gráficas extraídas de la muestra nombrada con DeepNano

A continuación, se muestran, para cada tipo de lectura (*Template, Complement y 2D*), las gráficas que representan de forma visual las longitudes de las lecturas obtenidas para el aislado de *S. agalactiae*. Para cada tipo tenemos una gráfica que muestra para ciertas longitudes, el número de lecturas con esa longitud.

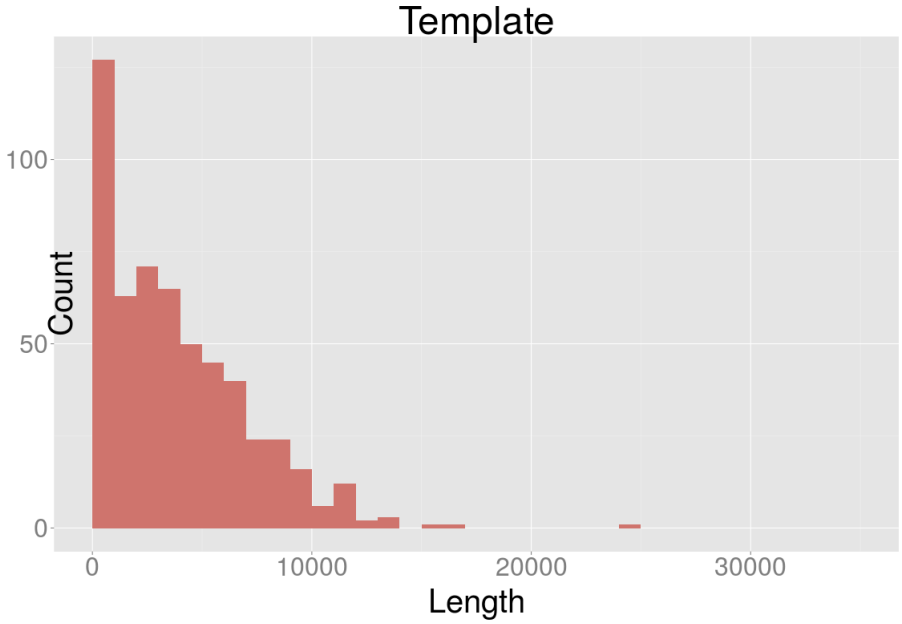

Figura 4.1. Gráfica de longitud de lecturas tipo *Template* usando DeepNano

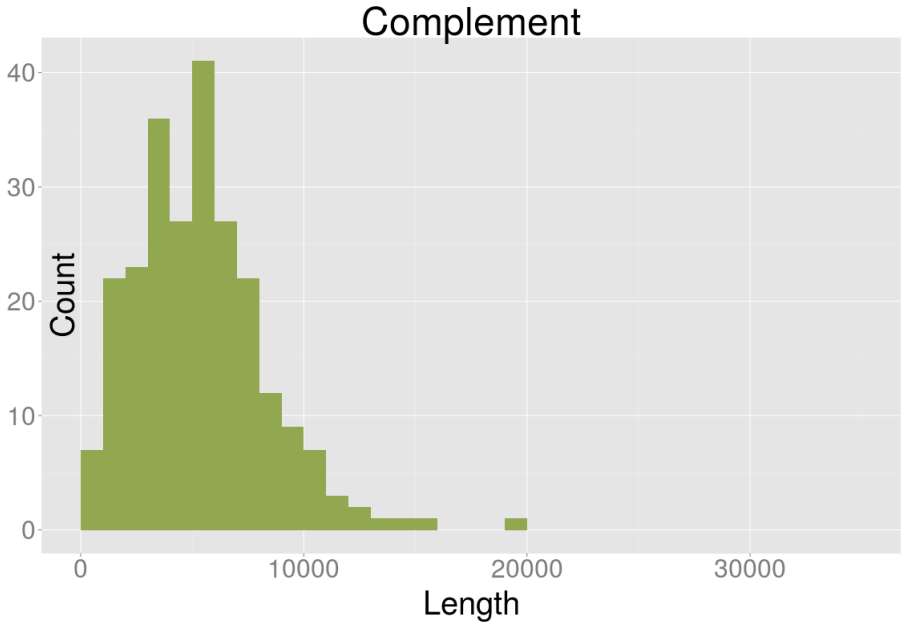

Figura 4.2. Gráfica de longitud de lecturas tipo *Complement* usando DeepNano

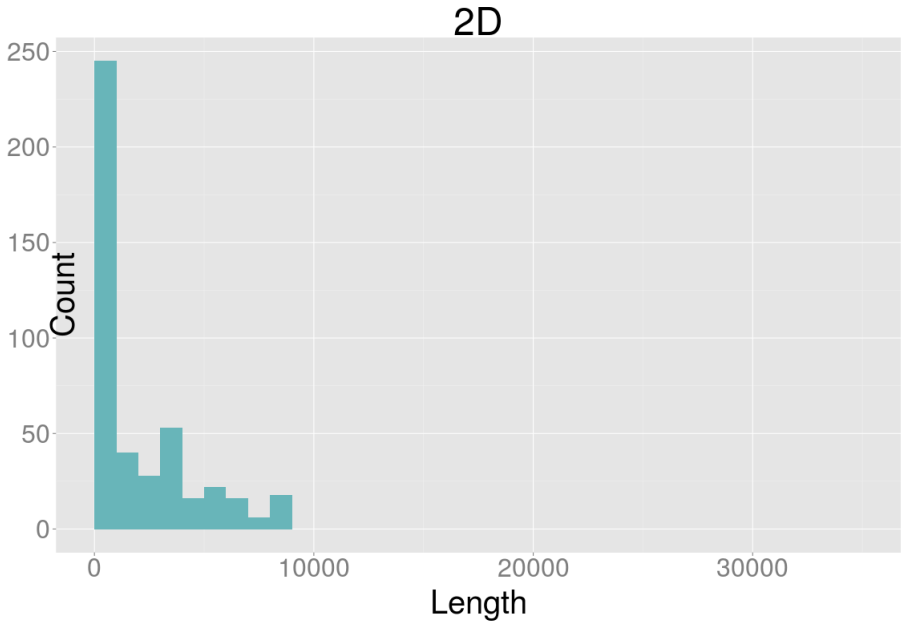

Figura 4.3. Gráfica de longitud de lecturas tipo *2D* usando DeepNano

De nuevo, para cada tipo de lectura, tenemos una gráfica que muestra el número de lecturas en la muestra con un porcentaje GC determinado. El porcentaje GC representa la cantidad de bases de tipo guanina o citosina en la molécula de ADN o genoma que está siendo investigado.

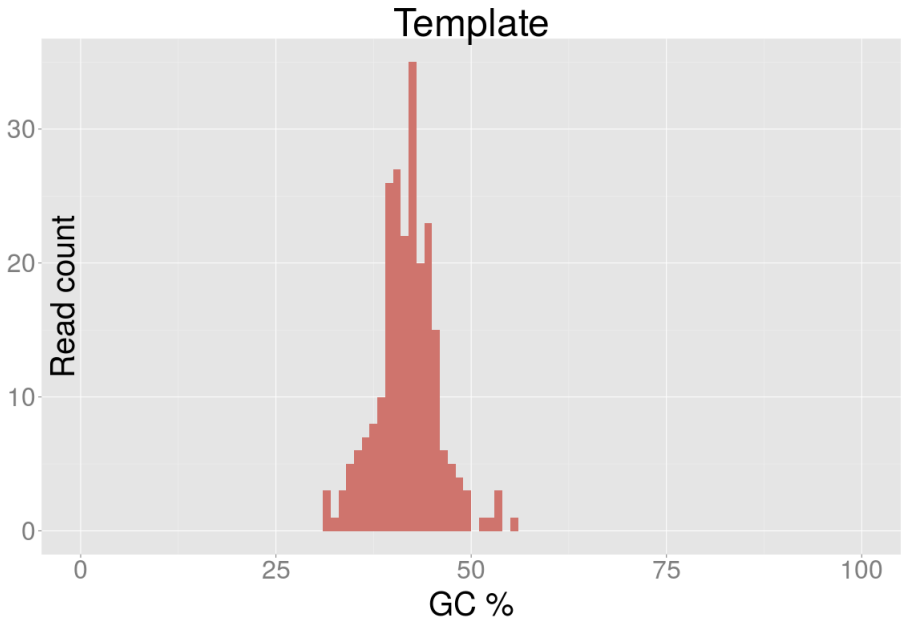

Figura 4.4. Gráfica de porcentaje GC de lecturas tipo *Template* usando DeepNano

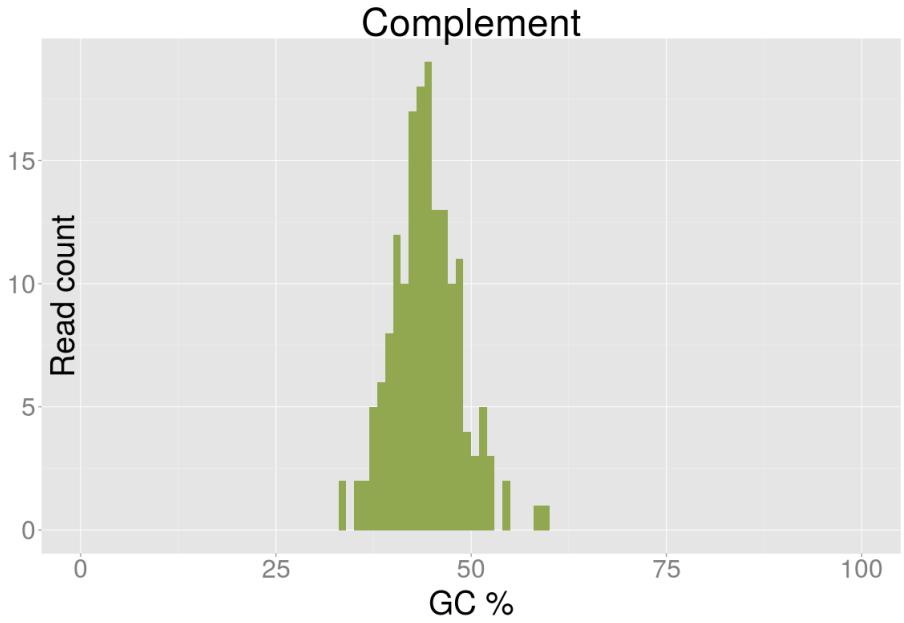

Figura 4.5. Gráfica de porcentaje GC de lecturas tipo *Complement* usando DeepNano

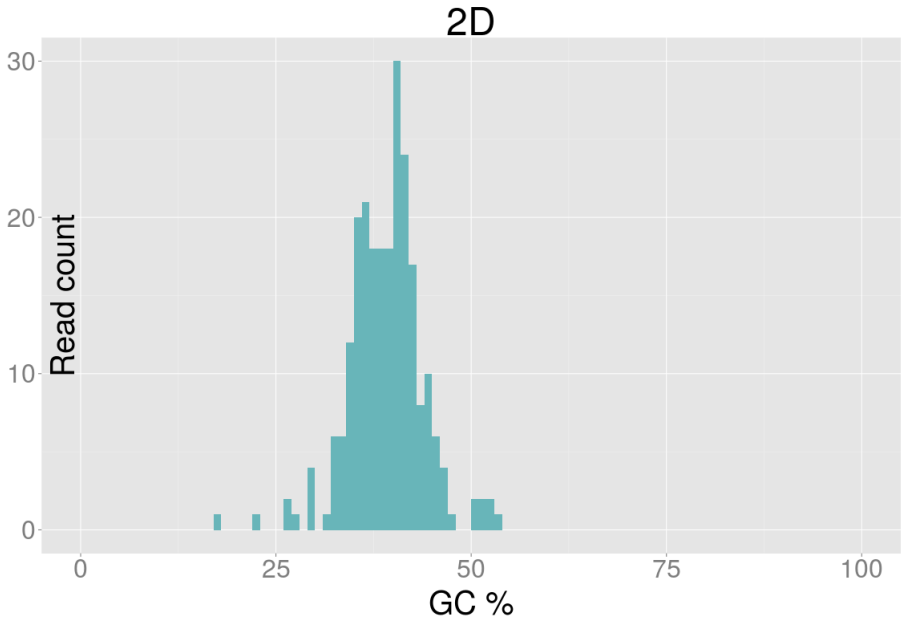

Figura 4.6. Gráfica de porcentaje GC de lecturas tipo *2D* usando DeepNano

4.2.2 Gráficas extraídas de la muestra nombrada con Metrichor.

A continuación vemos las mismas gráficas que en el apartado anterior pero referentes a los resultados obtenidos de las lecturas nombradas utilizando Metrichor.

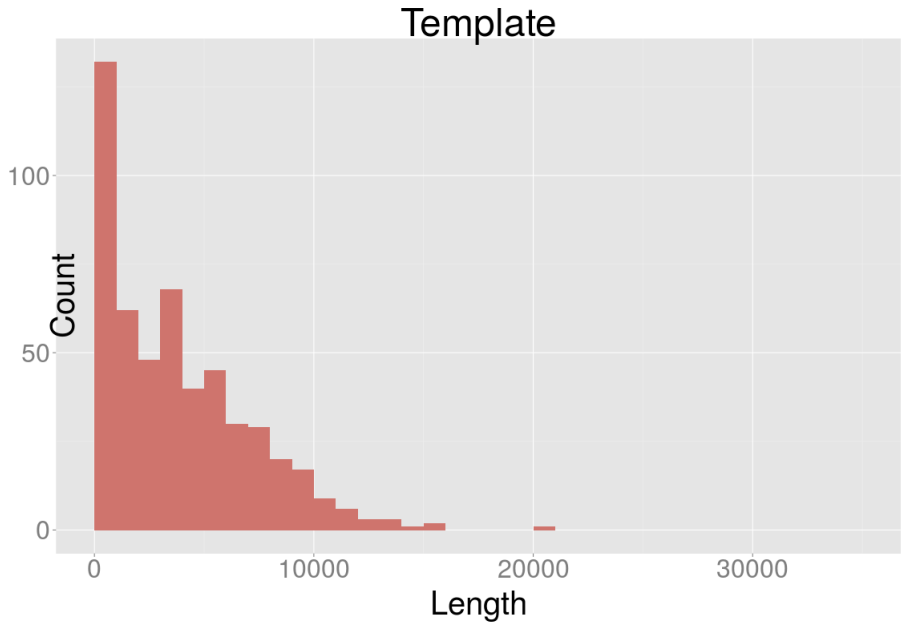

Figura 4.7. Gráfica de longitud de lecturas tipo *Template* usando Metrichor

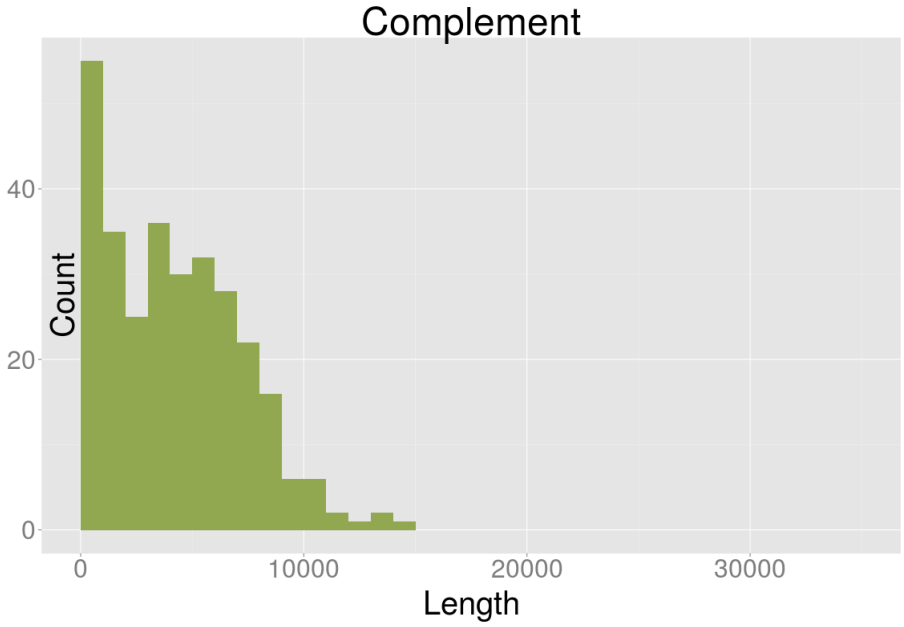

Figura 4.8. Gráfica de longitud de lecturas tipo *Complement* usando Metrichor

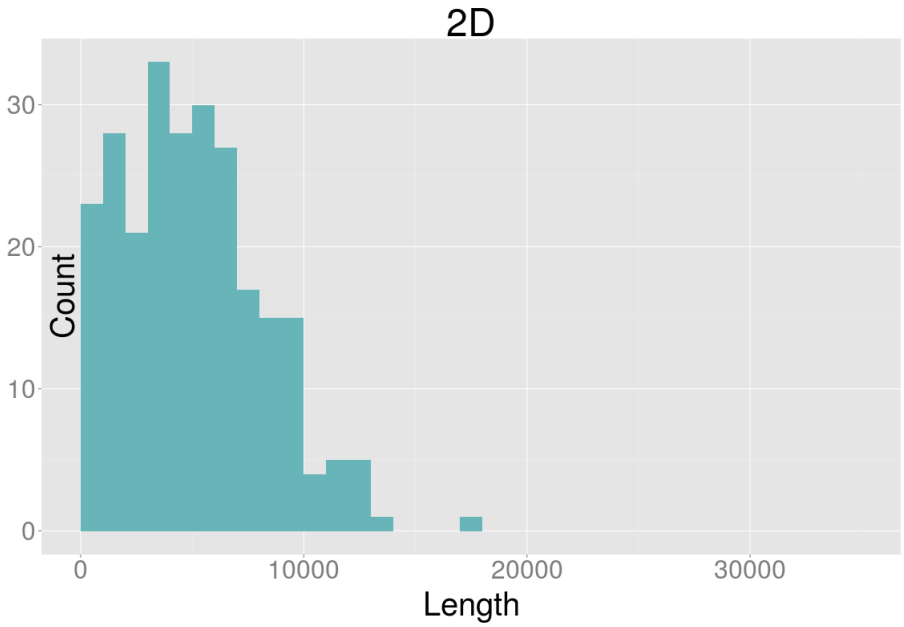

Figura 4.9. Gráfica de longitud de lecturas tipo *2D* usando Metrichor

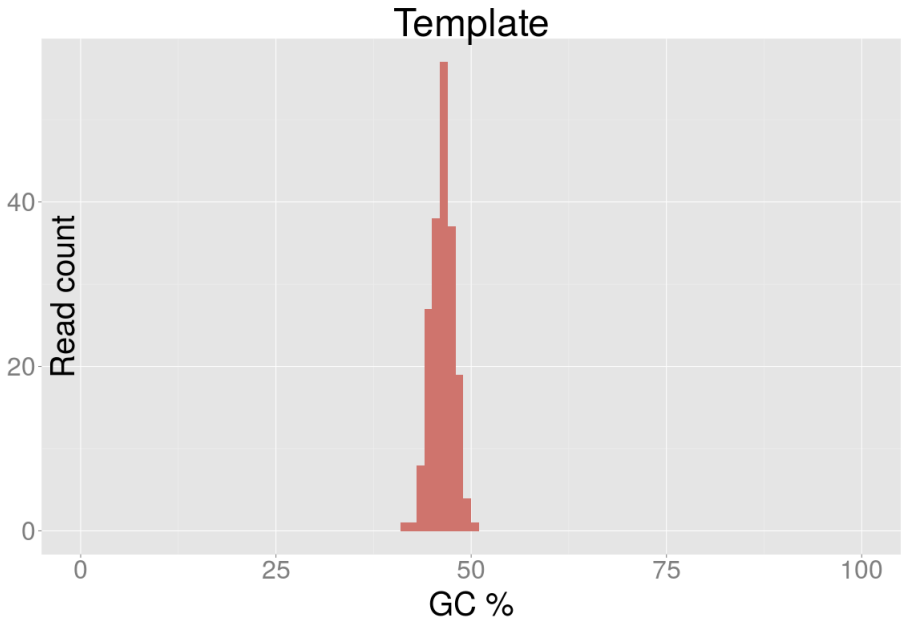

Figura 4.10. Gráfica de porcentaje GC de lecturas tipo *Template* usando Metrichor

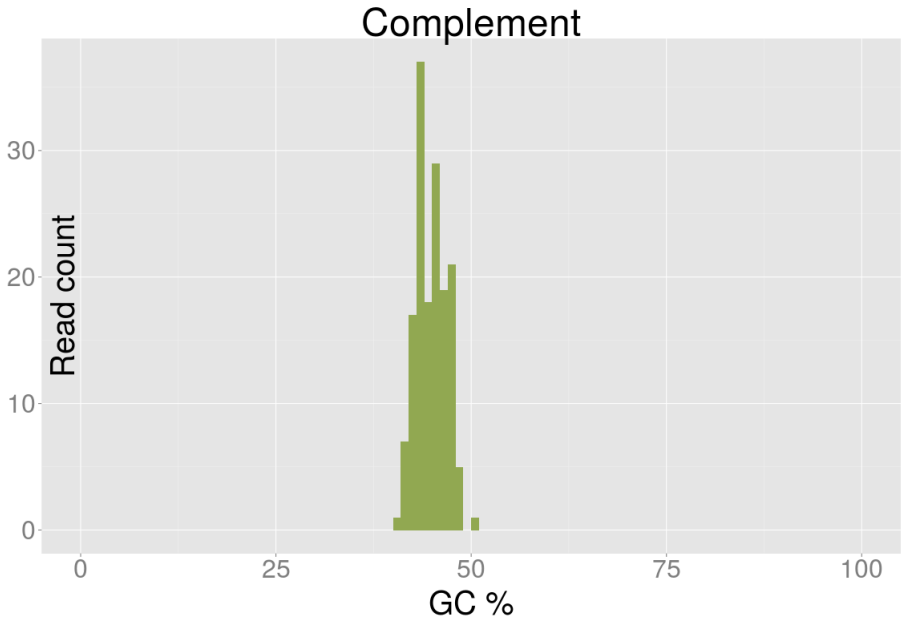

Figura 4.11. Gráfica de porcentaje GC de lecturas tipo *Complement* usando Metrichor

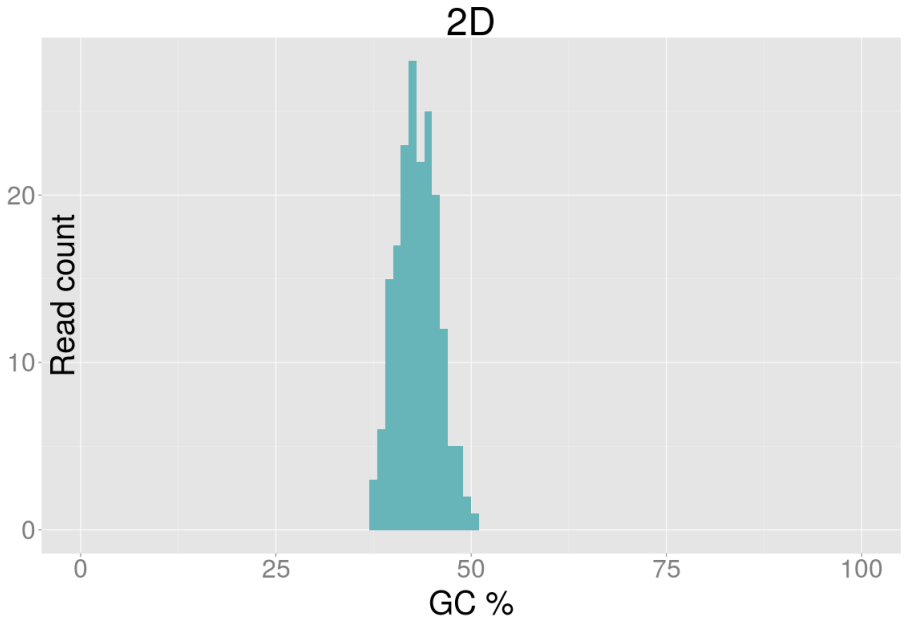

Figura 4.12. Gráfica de porcentaje GC de lecturas tipo *2D* usando Metrichor

Capítulo 5. Presupuesto

Para el presupuesto total del trabajo tendremos en cuenta tanto recursos humanos (ingeniero informático y biólogo especializado en genómica), como todo el material necesario. Las horas de trabajo asignadas incluyen también todo el proceso de documentación y aprendizaje, ya que tanto con el MinION como con las herramientas a utilizar, extraer unos resultados numéricos y gráficos supone conocer los procesos y herramientas implicadas y los contenidos abordados en este trabajo tienen una carga de investigación extra.

Debemos tener en cuenta también que el MinION, no solo asume unos conocimientos de genómica amplios para trabajar, sino que su puesta en marcha también requiere una obtención y preparación previa de las muestras, por lo que es necesario tener si no un laboratorio, unos materiales concretos.

5.1 Recursos humanos

5.1.1 Ingeniero informático

Tarea	Horas	Precio	Total
Fase de documentación	80	15 €/h	1.200 €
Implementación y resultados	40	15 €/h	600 €
Redacción de la memoria	20	8 €/h	160 €

Tabla 5.1. Resumen del presupuesto para el Ingeniero Informático

5.1.2 Biólogo

Tarea	Horas	Precio	Total
Fase de documentación	80	15 €/h	1.200 €
Preparación de muestras	10	15 €/h	150 €
Puesta en marcha del MinION	10	15 €/h	150 €

Tabla 5.2. Resumen del presupuesto para el Biólogo

5.2 Costes materiales

Material	Precio
Portátil Dell XPS 13	1.179 €
MinION (User Starter Pack)*	1.000 €
Material de laboratorio necesario (plásticos)	300 €

Tabla 5.3. Resumen del presupuesto para los materiales

El pack escogido del MinION viene con los reactivos y consumibles específicos de MinION necesarios para la realización de las primeras pruebas. En el caso de ya contar con el MinION o querer hacer más pruebas extra, se pueden comprar los materiales también en la página oficial.

5.3 Costes totales

Elemento	Coste
Tareas (Ingeniero Informático)	1.960 €
Tareas (Biólogo)	1.500 €
Material	2.479 €
Total:	5.939 €

Tabla 5.4. Resumen del presupuesto para los costes totales

Capítulo 6.
Resumen y Conclusiones

Abordar un trabajo de fin de grado de estas características ha supuesto una introducción y toma de contacto en el mundo de la investigación y la innovación científica en el ámbito de la Genómica. Es bastante destacable el hecho de que nuestra profesión es muchas veces necesaria y valorada en otras disciplinas totalmente distintas. Los equipos multidisciplinares son a menudo una forma de enriquecimiento profesional y personal, ya que nos abren los ojos de cara a otros mundos completamente diferentes y nos hacen conscientes de la importancia de la cooperación entre profesionales de distintos ámbitos para el desarrollo de la sociedad.

A pesar de que con el desarrollo de este cuaderno y con la extracción de datos propios a partir de una muestra nos hemos introducido de lleno en la Bioinformática, esta rama de nuestra profesión es muy amplia y abarca aspectos aún más difíciles de tratar como el desarrollo de algoritmos de ensamblado, técnicas de procesamiento de datos más rápidas o de plataformas de trabajo con datos biológicos esenciales para lo que la sociedad espera de la medicina moderna.

El MinION está siendo sin duda una herramienta prometedora de cara al presente y al futuro. Convertir la secuenciación de ADN en un proceso más simple y asequible, no solo motiva a los investigadores a trabajar con el MinION y a desarrollar herramientas con el fin de ayudar a la comunidad, sino que permite imaginar un futuro donde las ventajas de un aparato como este mejoran la calidad del sistema sanitario.

El papel del Ingeniero Informático en este ámbito no es solo el que se ha mostrado en este trabajo sino que va más allá. El Big Data ya ha llegado, y ahora nos toca a nosotros los profesionales, llevarlo más allá de los negocios o redes sociales. Las aplicaciones de herramientas como el MinION formarán parte de un sistema sanitario donde el diagnóstico y el tratamiento estarán personalizados. Tendremos más información que nunca sobre el estado de salud

de la población, el cáncer e incluso sobre las enfermedades que, por razones genéticas, cierta persona tendrá riesgo de padecer.

Por otra parte, las tecnologías como Jupyter, han llegado como pieza fundamental a la hora de generar entornos docentes o de trabajo. Ahora pasamos de manejar una lista de scripts o de utilizar herramientas de gestión de procesos para agilizar nuestro trabajo a tener a nuestro alcance una mejor forma de trabajar o presentar nuestras tareas. La metodología de trabajo de Jupyter ha permitido, por ejemplo, dotar a este trabajo de una capacidad extra para visualizar los procesos y resultados. Estamos dando al usuario un manual de trabajo con el MinION del que no tiene por qué salir para continuar trabajando con sus datos y mostrar solo aquellos datos que le interesan.

Algunas empresas importantes en el mundo del *Data Science* como DataBricks (de los creadores de Apache Spark), ya están utilizando este formato de cuaderno para los entornos de trabajo que ofrecen a sus clientes. En general, la computación científica y todas aquellas disciplinas en las que se desarrollan soluciones software que se organizan en forma de *pipelines*, se verán mejoradas gracias al proyecto Jupyter, que sigue en continuo desarrollo.

En este trabajo, se han extraído una serie de datos significativos haciendo uso solamente de herramientas de uso libre y bastante novedosas. Sin embargo, para nuestro cuaderno hemos escogido solo algunas de esas herramientas tras la fase de prueba, valorando su sencillez de uso y capacidad para obtener una buena cantidad de resultados sin utilizar un gran número de ellas, o repetir los mismos procesos una y otra vez para obtener resultados extra. Actualmente se siguen sacando nuevas herramientas, ya con usos más específicos. La forma de trabajar con todas ellas es similar a cómo se ha trabajado en el cuaderno, y la mejor manera de visualizar todo este trabajo, ha sido sin duda la utilización de Jupyter.

Por otra parte, la instalación de todas las herramientas y el tiempo que se debe emplear para utilizarlas correctamente y extraer unos buenos resultados, puede requerirle al profesional de la Biología unos conocimientos de informática que no tendrían por qué exigirse en su profesión, y éste es el principal obstáculo que se ha observado en la realización de este trabajo.

En resumen, trabajar en un entorno de investigación multidisciplinar, ha sido una experiencia de gran valor y a la vez una toma de contacto con el presente

y futuro de la secuenciación de ADN. Es una de las muchas salidas posibles que ofrece nuestra profesión y vuelve a recordarnos que estamos en continuo aprendizaje, un proceso que nos propone nuevos retos continuamente y que puede suponer a veces sacrificio, pero cuyas recompensas son importantísimas para el mundo actual.

Capítulo 7.
Summary and Conclusions

Working on a final degree project with this features, has been an introduction and a first contact with the research and innovation world. Its quite remarkable the fact that our profession is usually necessary and valued in other totally different fields. Multidisciplinary teams are often a way for professional and personal enrichment, because they open our eyes to totally different worlds and make us conscious about the value of the cooperation between professionals in different fields to society development.

Although the development of this notebook and the original data extraction using a sample have introduced us deeply on bioinformatics, this field of our profession is quite large and it encompasses even more difficult aspects of dealing like the alignment algorithms development, faster data processing technics or essential working platforms with biological data to what society expects of modern medicine.

The MinION is being, with no doubt a promising tool for the present and the future. Transforming DNA sequencing in a more simple and affordable process, not only motivates scientists to work with the MinION and develop tools with the purpose of help the community but allow us to imagine a future where the advantages of a gadget like this improve the quality of our sanitary system.

The role of a computer engineer in this field is not only the one that we have shown in this project. It goes beyond. Big Data has arrived and now is the turn for professionals to bring it from business or social media. The application of equipment like the MinION, will be part of a sanitary system where diagnosis or treatment will be personalized. We will have more information than ever about the population health, cancer o even about diseases that, because of genetic reasons, certain persons have the risk to suffer.

In the other hand, technologies like Jupyter, have arrived like a fundamental piece to generate learning and working environments. Now we turn to manage a list of scripts or use management tools to streamline our work to have a better way to present our work or tasks. The Jupyter workflow has allowed, for example, bringing this project an extra capability to visualize processes and results. We are providing the users with a manual for the MinION to keep on working with their own data and showing only the results that are of interest.

Some important companies like Databricks (Apache Spark developers), are now using this notebook format for working environments that offer to their clients. In general, scientific computation and all other fields that develop software solutions with pipelines, will be improved thanks to Jupyter, whose develop continues.

In this project, we have extracted some significative datasets using only open source and very new tools. Nevertheless, we have chosen only a few of these tools for our notebook after the testing process, valuing its simplicity and capability to obtain a good quantity of results without using a big number of tools or repeating processes. New tools are being released, now with more specific uses. The way to work with all of them is similar to the way we have seen in this project and the best way to visualize the results, has been Jupyter with no doubts. In the other hand, the installation of all of these tools and the time you have to spend learning to use them, can force the biology professionals to have some computer science knowledge that aren't related to their job and this is the main obstacle that we have been observed working on this project.

In summary, working on a multidisciplinary research environment, have been a high valued experience and a first contact with the present and future of DNA sequencing. It is one of many possible outputs that our profession offers and it remind us that we are always learning, a process that bring us new challenges all the time and that can sometimes assume sacrifice, but with important rewards to real world.

Bibliografía

[1] Anaconda. www.continuum.io/why-anaconda

[2] BLASR. github.com/PacificBiosciences/blasr

[3] V. Boža, B. Brejová, T. Vinař.DeepNano: Deep Recurrent Neural Networks for Base Calling in MinION Nanopore Reads. *arXiv*, 2016. Disponible en: arxiv.org/abs/1603.09195

[4] M. David, L.J. Dursi, D. Yao, P.C. Boutros, J.T. Simpson. Nanocall: An Open Source Basecaller for Oxford Nanopore Sequencing Data. *bioRxiv*, 2016. DOI: 10.1101/gr.113985.110.

[5] DNA Sequencing with MinION. www.nanoporetech.com

[6] Jupyter Notebook. jupyter.org

[7] S.M. Kiełbasa, R. Wan, K. Sato, P. Horton, M.C. Frith. Adaptive seeds tame genomic sequence comparison. Genome Research 21: 487-493, 2011. DOI: 10.1101/gr.113985.110.

[8] LAST. last.cbrc.jp

[9] LIGO Open Science Center. losc.ligo.org/tutorials

[10] R.M Leggett, D. Heavens, M. Caccamo, M.D. Clark, R.P. Davey NanoOK: multi-reference alignment analysis of nanopore sequencing data, quality and error profiles. *Bioinformatics* 32 (1): 142-144, 2016. DOI: 10.1093/bioinformatics/btv540.

[11] H. Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiV*. 2013. Disponible en: arxiv.org/abs/1303.3997

[12] MarginAlign. github.com/benedictpaten/marginAlign

[13] Metrichor. metrichor.com/s

[14] Nbviewer. nbviewer.jupyter.org

[15] Python. www.python.org

[16] R. www.r-project.org

[17] Z. Wang, A. Ma'ayan. An open RNA-Seq data analysis pipeline tutorial with an example of reprocessing data from a recent Zika virus study [version 1; referees: 3 approved]. *F1000Research* 2016, 5:1574. DOI: 10.12688/f1000research.9110.1