Desarrollo de una herramienta software de identificación de secuencias patógenas candidatas para el diseño de RNAs guía en el sistema SHERLOCK de diagnóstico.

Samantha López
Máster Universitario en Bioestadística y Bioinformática
TFM - Área 30 – Estadística y Bioinformática

Consultor: Amadís Pagès Pinós
Profesor responsable de la asignatura: Carles Ventura Royo

Entrega: 05/06/2018

FICHA DEL TRABAJO FINAL

Título del trabajo:	Desarrollo de una herramienta software de identificación de secuencias patógenas candidatas para el diseño de RNAs guía en el sistema SHERLOCK de diagnóstico.
Nombre del autor:	Samantha López Mochales
Nombre del consultor/a:	Amadís Pagès Pinós
Nombre del PRA:	Carles Ventura Royo
Fecha de entrega (mm/aaaa):	06/2018
Titulación::	Máster Universitario en Bioestadística y Bioinformática
Área del Trabajo Final:	TFM - Área 30 – Estadística y Bioinformática
Idioma del trabajo:	Castellano
Palabras clave	CRISPR, herramienta, patógeno.

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

El uso del Sistema CRISPR-Cas como método, en biología molecular, para la edición del DNA con alta especificidad, es revolucionario. Desde los primeros usos reportados en células de mamífero en 2013, ha sido ampliamente usado en la generación de organismos modelo modificados para enfermedades humanas (Musunuru, 2017).

SHERLOCK (Gootenberg et al., 2017) es una nueva plataforma basada en CRISPR-Cas de diagnóstico molecular para la detección de infecciones en Point-Of-Care, rápida y económica, que ofrece valores extremadamente altos de sensibilidad y especificidad. Está basada en el efecto colateral de la enzima Cas13a, que actúa como RNAsa promiscua cuando hibrida con su diana. Permite inferir la presencia de RNAs pertenecientes a determinados patógenos.

Este proyecto describe el desarrollo de una plataforma software que servirá para identificar regiones diana (secuencias cortas) en las secuencias genómicas de un patógeno, a través de las cuales se diseñarán crRNAs, para ser utilizados en la detección de dicho organismo mediante la construcción de un test SHERLOCK con este crRNA.

Los outputs obtenidos son secuencias, de longitud deseada, comunes entre todas las cepas y secuencias únicas para cada una de las cepas que no están presentes en el resto. La herramienta además comprueba la no-presencia de estas secuencias en el genoma del huésped y devuelve potenciales secuencias crRNA basadas en las originales.

Esta herramienta mejorará la eficiencia de actuación para los investigadores en la comunidad de CRISPR.

Abstract (in English, 250 words or less):

The use of the CRISPR-Cas system as a method, in molecular biology, for DNA edition with high specificity, is revolutionary. Since its first reported uses in mammal cells in 2013, it has been widely used in generation of modified model organisms for human diseases (Musunuru, 2017).

SHERLOCK (Gootemberg et al., 2017) is a new CRISPR-Cas-based platform of molecular diagnosis for detection of infections in Point-Of-Care, fast and economic, that offers extremely high values of sensitivity and specificity. It is based on the collateral effect of the Cas13a enzyme, that acts as a promiscuous RNAse when matching its target. It allows to infer the presence of RNAs belonging to certain pathogens.

This project describes the development of a software platform that will serve to identify target regions (short sequences) in the genomic sequence of a pathogen, through which crRNAs will be designed, to be used for the detection of this organism by building a SHERLOCK test with this crRNA.

The obtained outputs are both sequences, of desired length, that are common in all strains analyzed, and unique sequences for each strain that are not present on the other ones. The tool also proves the non-presence of these sequences in the host genome and gives back potential crRNA sequences based on the original ones.

It will improve the efficiency of performance for researchers on the CRISPR community.

Índice

Lista de ilustraciones

1. Introducción

1.1 Contexto y justificación del Trabajo

SHERLOCK (Gootenberg et al., 2017) es una novedosa plataforma de diagnóstico molecular basada en el sistema CRISPR-Cas, para la detección de patógenos en el "Point-of-care", rápida y barata, que ofrece unos valores extremadamente altos de sensibilidad y especificidad. Se basa en el efecto colateral de la enzima Cas13a, que actúa como RNAsa, rompiendo RNAs cercanos indiscriminadamente, cuando ha ocurrido una hibridación entre el crispRNA con el que forma complejo y la región diana de un patógeno.
Hasta ahora, han sido desarrollados y ampliamente utilizados múltiples métodos de edición genómica basados en enzimas de restricción de DNA, como las meganucleasas, ZFNs (*Zinc Fingers*) y TALEN (*transcription activator-like effector nucleases*). Sin embargo, ningún sistema ha demostrado tanta especificidad a la par que eficiencia como el sistema CRISPR-Cas, hecho que lo convirtió en ganador del Breakthrough Prize en Life Sciences en 2015.

La aplicación principal del sistema ha sido, desde su origen, la edición genómica para la generación de organismos modelo, como *knockout* de rutas metabólicas con intereses industriales, y en algunos casos como método de terapia génica. El desarrollo del método SHERLOCK abre un abanico de posibilidades de aplicación del sistema CRISPR-Cas como herramienta diagnóstica de alta precisión, que podría suponer una revolución en el diagnóstico de enfermedades infecciosas, especialmente en áreas con escasos recursos económicos, zonas rurales aisladas, en hospitales de campaña…
Es necesario que vaya acompañada de un método de identificar de forma rápida y eficaz las secuencias patógenas más aptas para ser detectadas en un potencial huésped, cuya precisión y especificidad sea acorde con la del método SHERLOCK.
Es por eso por lo que se ha desarrollado este trabajo: para proporcionar una herramienta de identificación de secuencias patógenas idóneas para generar RNAs guía que puedan ser implementados en un test diagnóstico SHERLOCK, y que aseguren la identificación de una cepa patógena particular, de una forma mucho más económica y sensible que las que se conocen hasta el momento.

1.2 Objetivos del Trabajo

El objetivo principal del trabajo es el desarrollo de una herramienta que permita identificar secuencias patógenas que sean idóneas para la generación de RNAs guía para el sistema diagnóstico SHERLOCK. Los objetivos de dicha herramienta informática son, a partir del genoma de varias cepas de determinado patógeno, y el genoma de determinado huésped, la obtención de secuencias cortas (de longitud determinada por el usuario) que sean comunes a todas las cepas y que no estén presentes en el huésped, y la obtención de secuencias de la misma longitud que sean particulares para cada una de las cepas (que no estén presentes en el resto de las cepas) y que tampoco estén presentes en el huésped. Además, obtener enlaces a una herramienta de diseño de secuencias para el sistema CRISPR-Cas, ya que el objetivo

final es que el usuario pueda diseñar sus propios RNA guía para aplicar con el método SHERLOCK.

Otro objetivo es la validación del funcionamiento de la herramienta con una serie de secuencias genómicas de distintas cepas de virus del papiloma humano (HPV) y la última versión del genoma de *H.sapiens* como huésped.

1.3 Enfoque y método seguido

El método que se ha llevado a cabo ha consistido en diseñar la herramienta a partir de los objetivos que se quería obtener. Los outputs, como se ha comentado anteriormente, son una serie de secuencias comunes a todas las cepas y una serie de secuencias particulares por cada cepa, asegurando que ninguna de ellas está presente en el genoma del huésped y, además, se ha trabajado para obtener también una serie de enlaces a la herramienta web de diseño de crispRNAs escogida. Partiendo de estos objetivos, se ha ido diseñando y construyendo la herramienta función a función. Finalmente, se han integrado todas las funciones en un *script* para ser ejecutado, y se ha probado su funcionamiento con determinadas secuencias de HPV y la última versión del genoma humano.

El lenguaje de programación con el que se han implementado las funciones es Python, ejecutado des de la línea de comandos de Linux, en una máquina virtual de sistema operativo Ubuntu (64-bit) mediante el programa Oracle VM Virtual Box.

1.4 Planificación del Trabajo

Los recursos necesarios para la ejecución del trabajo han sido las herramientas informáticas asociadas (nombradas en el apartado anterior), más las herramientas contenidas en el paquete Biopython para Python. También han sido empleados una serie de recursos bibliográficos para contextualizar sobre el sistema CRISPR-Cas y SHERLOCK, ya que la aplicación de la herramienta va enfocada al uso de este sistema.

Las principales tareas que se han llevado a cabo son la contextualización de ambos sistemas nombrados, el desarrollo de la herramienta en forma de código documentado (para justificar la implementación de cada función), la justificación del uso de secuencias de HPV para el test, y la ejecución del test de la herramienta con dichas secuencias para obtener resultados.

El calendario que se ha seguido para llevar a cabo todas las tareas se presenta en el Anexo 01 de este trabajo.

1.5 Breve sumario de productos obtenidos

Los productos obtenidos son una descripción teórica sobre los sistemas en los que la herramienta podrá ser aplicada (CRISPR-Cas y SHERLOCK), un archivo *script.py* con el código del programa desarrollado, y los resultados obtenidos de los test ejecutados con secuencias de HPV y *H.sapiens*.

1.6 Breve descripción de los otros capítulos de la memoria

Los capítulos que siguen son una breve introducción teórica sobre las secuencias CRISPR, el sistema CRISPR-Cas y el sistema SHERLOCK, una descripción de la metodología empleada para el desarrollo de la herramienta y una serie de resultados obtenidos a partir de una serie de secuencias de virus de HPV, justificando por qué se han escogido estas secuencias.

2. CRISPR

2.1 El sistema CRISPR-Cas

La tecnología **CRISPR** deriva del sistema inmune de algunas bacterias y arqueas como *Streptococcus pyrogenes* y *Staphilococcus epidermidis*. Se trata de una herramienta basada en la nucleasa Cas9 asociada a secuencias CRISPR (*Clustered Regularly Interspaced Short Palindromic Repeats*). Este sistema, in vivo, representa un método de inmunidad adaptativa para eliminar material genético invasor (ilustración 1).

El DNA invasor, al entrar en contacto con la bacteria, es cortado en pequeños fragmentos, y estos se integran en un *tandem array* en el locus de CRISPR del organismo huésped. Este se transcribe y se procesa hasta obtener un tipo de RNA denominado **crRNA** o **crispRNA**, que guiará las endonucleasas efectoras que se unirán al DNA invasor por complementariedad de secuencias (Jinek et al., 2012). A partir de aquí, los pasos a seguir son distintos según el tipo de sistema que posee la bacteria de entre los tres existentes.

En los tipos I y III se involucra un gran complejo de proteínas Cas. En el sistema tipo II, el más estudiado, el crRNA forma complejo con un **tracrRNA** (*trans activating CRISPR RNA)* y la Cas9 asociada; sólo es requerida esta nucleasa, que participa en el procesamiento del crRNA y en la destrucción del DNA diana (Jinek et al., 2012, Deltcheva et al., 2011). Ensamblado este sistema (Cas9 y dos RNAs) Cas9 cortará el DNA invasor que tenga una secuencia complementaria al crRNA de pocos nucleótidos (alrededor de 20) cuando el patógeno vuelva a atacar la bacteria. Para ello, es necesaria la presencia de una secuencia corta (de 2 a 5 nucleótidos) adyacente (inmediatamente después del extremo 3' de la secuencia diana) denominada PAM (*protospacer-associated motif*) (Jinek et al., 2012, Swarts et al., 2012).

Las aplicaciones *in vitro* de este sistema han sido, desde su descubrimiento, principalmente, la edición del genoma de forma mediada, pero también se ha usado como *knockout* de vías metabólicas competitivas, o para incrementar la tolerancia al estrés generado por determinados metabolitos, siendo estas dos últimas aplicables a la industria de generación de sustancias a partir de cultivos microbianos.
El uso de **CRISPR-Cas** como método en biología molecular para la edición de DNA con alta especificidad es un campo novedoso y en expansión, y está generando mucha atención desde los primeros usos reportados en células de mamífero en el año 2013, por su potencial impacto en investigación y, especialmente, en sus aplicaciones terapéuticas. Ha sido ampliamente utilizado y ha mejorado sustancialmente la capacidad de generar organismos modificados como modelos de enfermedades humanas (Musunuru, 2017).
Lo que ha permitido el uso *in vitro* de este sistema ha sido el desarrollo, en 2012, de un sistema simplificado donde se combina el trRNA y el crRNA en un solo RNA guía sintético, sgRNA (*single guide RNA*) (Jinek et al., 2012).

También se ha usado la variante de Cas9 sin actividad nucleasa, dCas9, que no puede cortar DNA pero sí unirse a él, para identificar ciertas regiones del genoma,

como herramienta de visualización (Jinek et al., 2012, Qi et al., 2013, Gasiunas et al., 2012, Chen et al., 2014). Ha sido usada unida a EGFP (*Enhanced Green Fluorescent Protein*) para visualizar secuencias de DNA, y así identificar genes en organismos y líneas celulares. Sin embargo, este sistema aún requiere la amplificación del material genético para detectar DNAs diana particulares.

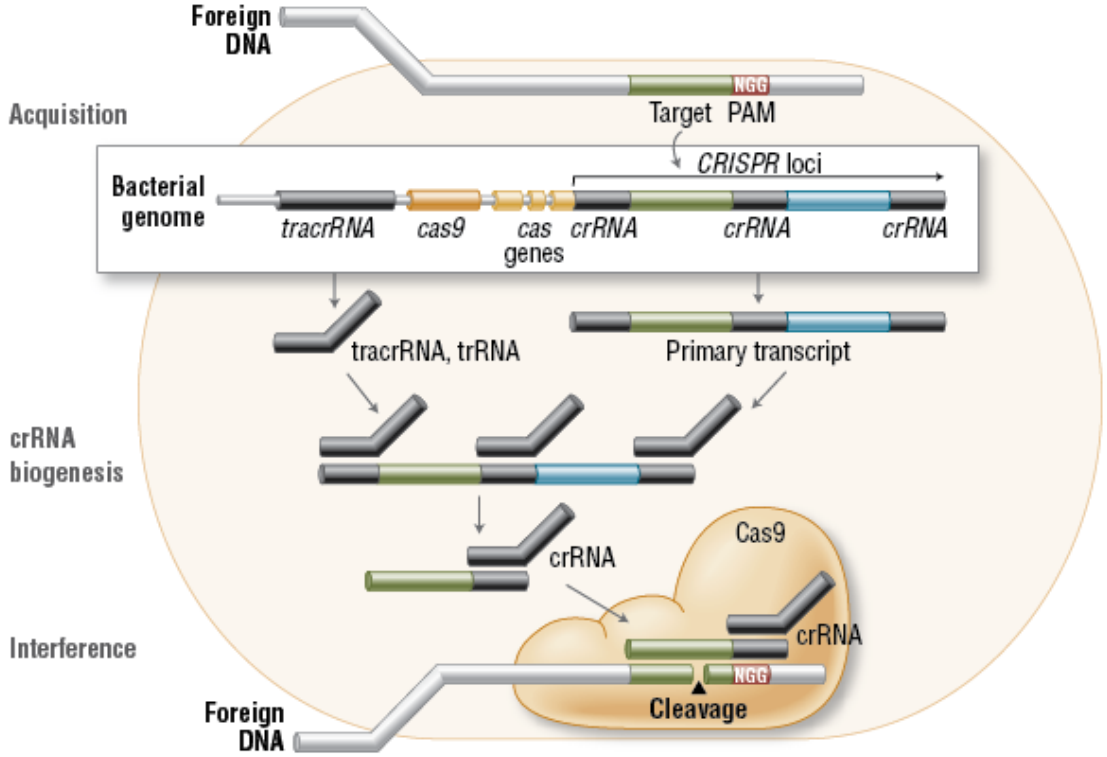

Ilustración 1 - Cas 9 in vivo: inmunidad adaptativa bacteriana. Biolabs, N. (2018).

2.2 SHERLOCK

La plataforma **SHERLOCK**, desarrollada en 2017, utiliza el sistema CRISPR-Cas13a, en lugar de Cas9, y especialmente el efecto colateral de **Cas13a** para detectar la presencia de determinados **RNA diana**. Puede ser utilizada para la detección de enfermedades de forma precisa, precoz, con muy poca cantidad de RNA y sin necesidad de amplificación del material genético (Gootenberg et al., 2017).
La nucleasa Cas13a es única en términos de estructura y función. Tiene dos dominios nucleasa tipo HEPN en vez de un HNH y un RuvC como la Cas9. Requiere unirse a un solo crRNA, y tolera diferencias puntuales, aunque, si hay 2, empieza a reducirse su capacidad de escindir. Requiere la presencia de una PFS (*protospacer flanking sequence*) homóloga a PAM, aunque en algunos casos, como la Cas13a de *Leptotrichia wadei* (LwaCas13a), esta puede mediar corte sin necesidad de la presencia de un PFS específico. Su mayor particularidad es su efecto colateral: una vez Cas13a ha reconocido y cortado su RNA diana, adopta un estado enzimático activo en vez de revertir a estado inactivo como hace Cas9. Entonces, corta RNAs adicionales sin necesidad de homología con el crRNA ni presencia de PFS (ilustración 2). Esto, in vivo, lleva a la muerte celular programada. Se ha visto que algunos ortólogos de Cas13a pueden funcionar en mamíferos y plantas.

SHERLOCK es la plataforma *in vitro* que explota este comportamiento de Cas13a, exponiendo una muestra fisiológica problema de la que se desea obtener diagnóstico para un patógeno determinado a la presencia de Cas13a unida a crRNAs para dicho patógeno, y a la vez añadiendo *beacons* fluorescentes que serán escindidos por la Cas13a sólo si esta hibrida con su diana, y que permitirán inferir la presencia del patógeno (Gootenberg et al., 2017).

Por ello, SHERLOCK es una novedosa plataforma de diagnóstico molecular para la detección de patógenos en el "Point-of-care", rápida y barata, que ofrece unos valores extremadamente altos de sensibilidad y especificidad, y que puede ser adaptada a cualquier patógeno mediante el diseño de los crRNAs correspondientes. Hasta el momento ha sido probada con el virus Zika, para genotipar DNA humano y para identificar la presencia de determinadas mutaciones tumorales.

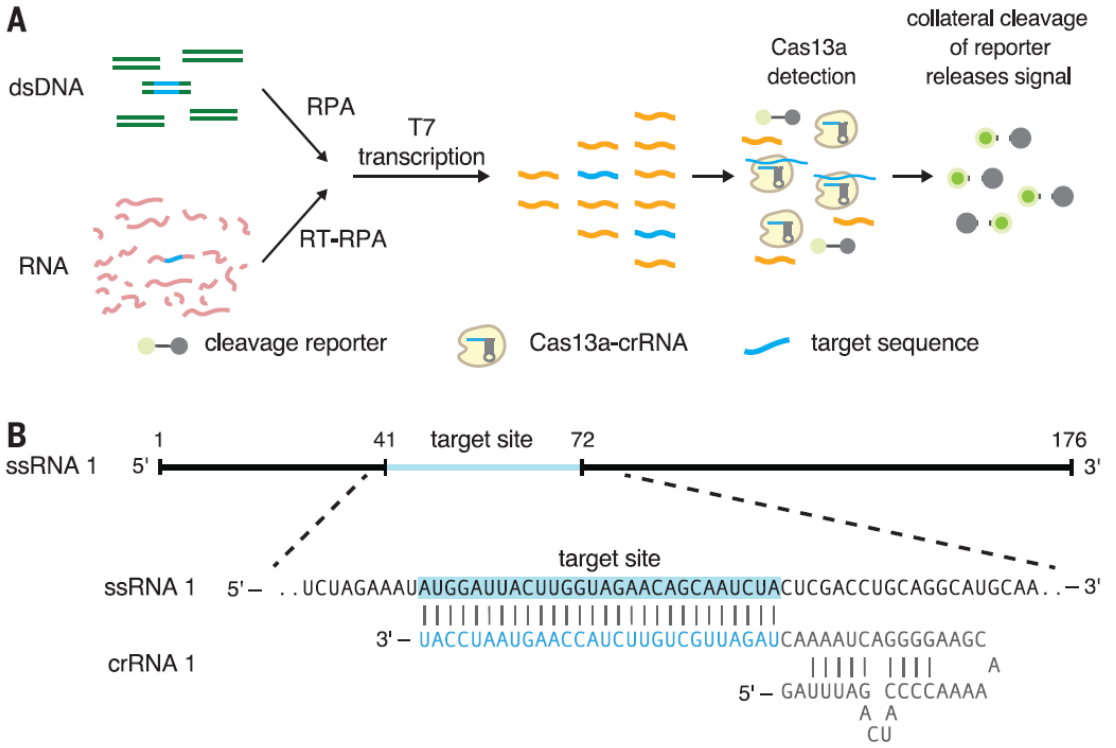

Ilustración 2 - A) Esquema del efecto colateral sobre los beacons (reportadores) de Cas13a al hibridarse el crispRNA con el que forma complejo con un RNA diana. B) Detalle de la hibridación entre RNA diana y crispRNA. (Gootenberg et al., 2017).

2.3 Herramientas bioinformáticas relacionades

El primer paso para usar el sistema de edición CRISPR-Cas es diseñar un RNA guia (sgRNA, o simplemente crRNA en el caso de Cas13a) que hibride con nuestro gen de interés. Hay muchas herramientas web que permiten diseñar sgRNAs de forma automática a partir de una secuencia que quiera usarse como diana, y cada una tiene sus ventajas y características.

Algunos ejemplos son *CasFinder*, de la Harvard Medical School, *CRISPR Design*, del MIT, o *CHOPCHOP* de la University of Bergen. Algunas permiten escoger las

características de la secuencia PAM deseadas, o modular con qué enzima de la familia Cas será implementado el sgRNA.

El desarrollo de la herramienta descrita en este trabajo permite obtener secuencias idóneas para ser introducidas en una de estas aplicaciones, para obtener, en nuestro caso, crRNAs óptimos.

3. Herramienta de búsqueda de secuencias guía para SHERLOCK

Los siguientes capítulos son destinados a describir el proceso de desarrollo de nuestra herramienta de interés, mediante el empleo de distintas metodologías para la obtención de nuestros resultados objetivo, y justificando en cada caso por qué son escogidas determinadas estrategias y no otras para llevar a cabo cada función.

Lo que esta herramienta permite es obtener de forma rápida secuencias de RNA guía para la enzima Cas13a, que podrán ser sintetizadas para combinarse con dicha enzima y el resto de los elementos requeridos por el sistema SHERLOCK, para obtener test diagnósticos precisos para cada una de las cepas patógenas de interés. Dada la alta especificidad de este sistema diagnóstico, es necesario identificar secuencias muy específicas para cada cepa, y es necesario que estas no estén presentes en el genoma del huésped ya que una de las ventajas del sistema SHERLOCK es que puede ser utilizado con muestras biológicas provenientes de un paciente del que se sospecha la presencia del patógeno, y de esta forma el sistema no dará falsos positivos por hibridación con el genoma del huésped.

3.1 Características

El objetivo de nuestra herramienta es la obtención de cuatro outputs distintos. El primero es una lista de secuencias comunes a todas las cepas patógenas introducidas como primer imput, que no están presentes en el huésped introducido como segundo imput. El segundo es una lista de secuencias particulares para cada una de las cepas (que no están presentes en el resto, y tampoco están presentes en el huésped) identificadas según la secuencia genómica (la cepa) de origen a la que pertenecen. El tercer output es un archivo con enlaces a una herramienta de diseño de secuencias de RNA guía para el sistema CRISPR-Cas13a (que es la nucleasa empleada en el sistema SHERLOCK) a partir de las secuencias comunes en todas las cepas. El cuarto y último es, de nuevo, un archivo con enlaces a la misma herramienta, pero partiendo de las secuencias particulares de cada una de las cepas (estos enlaces llevarán, en el archivo resultante, el mismo identificador que las secuencias cortas obtenidas, que corresponde a la cepa, para identificar siempre a qué cepa pertenece dicha secuencia particular).

3.2 Diseño

Todas las funciones de la herramienta han sido integradas en un archivo *script.py* (Anexo 03) para ser ejecutado des del terminal de Linux. Los archivos que requiere la herramienta como imput son un archivo con todas las secuencias patógenas a cotejar, en formato FASTA, que deberán pertenecer a distintas cepas del mismo organismo patógeno (idealmente, el genoma completo de cada una, para maximizar las probabilidades de encontrar subsecuencias de todos los tipos), y otro archivo también en formato FASTA con el genoma del huésped. Al llamar la herramienta des de la línea de comandos también se le indica la longitud deseada para las subsecuencias.

Los outputs son obtenidos en forma de archivos de texto, guardados en el directorio de trabajo, en el que deberán estar también los archivos imput y el *script*.

El primer paso es la búsqueda de secuencias comunes entre todas las cepas patógenas, y a continuación, se buscan las secuencias particulares de cada una. Se han probado distintas estrategias para estos pasos. Después, se cotejan todas estas subsecuencias con el genoma del huésped introducido, y se guardan como outputs aquellas que no estén presentes. Por último, se generan los enlaces a la herramienta de diseño de RNAs guía.

En la ilustración 3 se muestra un esquema general del funcionamiento de la herramienta, con referencias a los capítulos donde se detalla cada paso del desarrollo.

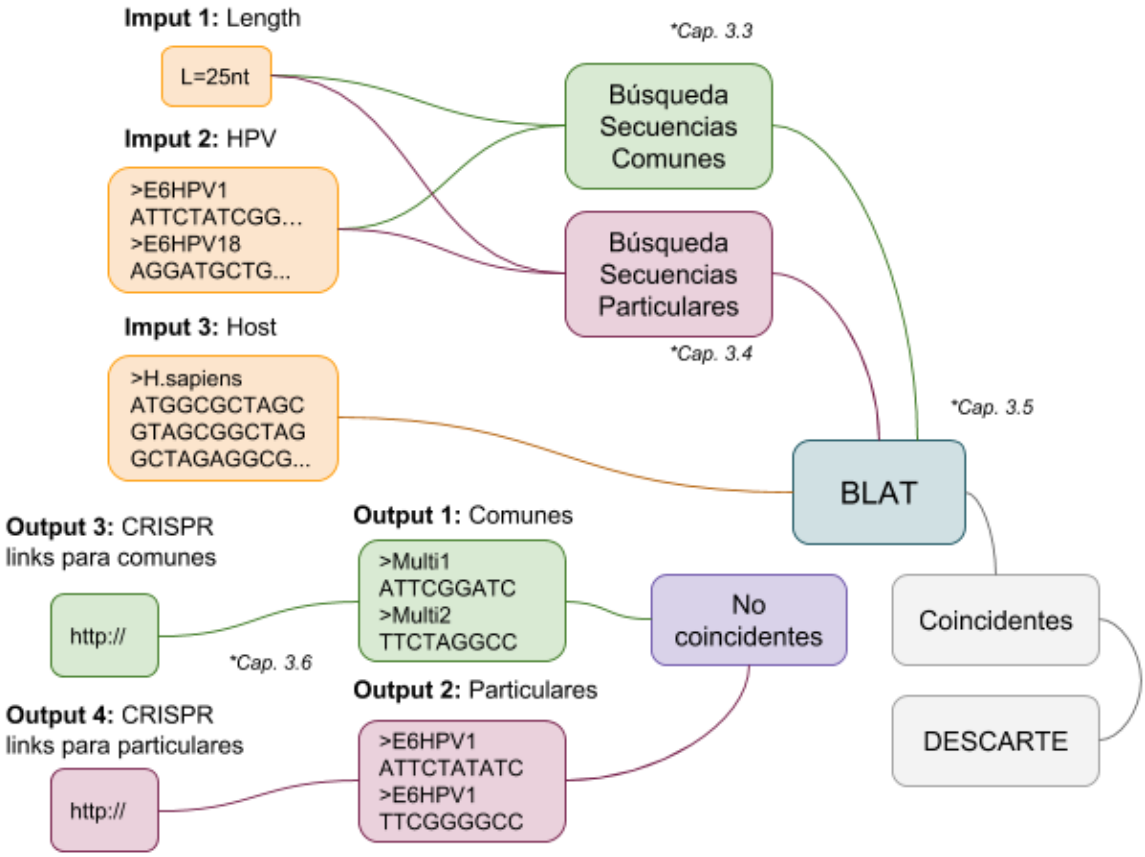

Ilustración 3 – Esquema general del funcionamiento de la herramienta.

3.3 Búsqueda de secuencias comunes

El objetivo de buscar secuencias comunes entre todas las cepas es la posibilidad de desarrollar un test multi-patógeno, que pueda identificar la presencia de un organismo sea cual sea la cepa presente en el huésped, si este es el interés del investigador o del personal médico.

3.3.1 Algoritmo de alineamiento múltiple: ClustalW

Hay varias formas de encontrar secuencias comunes entre un conjunto de múltiples secuencias. La primera estrategia que se probó fue un alineamiento múltiple mediante

el programa ClustalW ejecutado desde nuestro script. ClustalW es un método eficiente de alineamiento múltiple de secuencias de nucleótidos o proteínas. Es un método de alineamiento progresivo: alinea primero las secuencias más similares y luego, progresivamente, va alineando el resto, hasta tener un alineamiento global completo. Requiere tres o más secuencias como imput.
Este método nos permite obtener coincidencias entre los genomas de las cepas, para extraer las secuencias comunes a todas ellas (aquellas que presenten total alineamiento a lo largo de determinada longitud).

3.3.1.1 Características, ventajas e inconvenientes

En ClustalW las secuencias que se alineen con mayor puntuación (en un alineamiento pareado) son las primeras que serán alineadas. Esta puntuación se comprueba mediante una matriz de distancias entre cada par de secuencias. Después, se implementa el método *neighbor-joining*, que permite comprobar qué secuencia está más cerca a continuación, paulatinamente, a partir del primer alineamiento *pairwise*.

La principal ventaja de este método es que, al ser optimizado, no supone un coste computacional demasiado alto al alinear secuencias muy grandes. Sin embargo, al tratarse de un método de alineamiento global, en favor de permitir el alineamiento de todas las secuencias en conjunto, preservando el orden de los fragmentos de principio a fin, probablemente habrá secuencias, de la longitud que deseamos, que serían alineadas, pero no se plasmarán en el alineamiento, por tratarse de secuencias lejanas entre uno y otro genoma, por ejemplo.

3.3.1.2 Desarrollo

Ejecutamos el alineamiento múltiple con todas las secuencias de las cepas patógenas, y a continuación cortamos los fragmentos de longitud deseada que generen alineamiento total. Las secuencias que sí presentan alineamiento total de la longitud que habíamos determinado son guardadas en un objeto denominado “múltiples”. Esta es la parte más confusa a nivel conceptual del desarrollo de la herramienta, por lo que irá acompañada de algunos esquemas explicativos.

Antes del alineamiento, se define en un objeto “L” la longitud deseada para las secuencias a obtener, que será inmediatamente transformada en L+1 para tener en cuenta un nucleótido más que actuará como secuencia PFS a la hora de obtener nuestros RNA guía. Una vez definida la longitud, se procede al alineamiento.

Para llevar a cabo este paso, primero se ejecuta ClustalW, a partir de un archivo que contiene todas las secuencias patógenas en formato FASTA. ClustalW genera otro archivo de alineamiento donde las secuencias están alineadas en las posiciones que el programa ha considerado, y presentan “-“ donde el programa ha decidido introducir *gaps* para preservar el alineamiento global. Para ejecutarlo, primero se debe importar des del paquete Bio.Align.Applications, y para leer su output, se requiere el paquete AlignIO, que nos permite leer el archivo de alineamiento.

Una vez abierto, se genera un objeto de tipo lista de longitud igual a la del alineamiento, donde todos los elementos son “T” (*true*). A continuación, se muestra un esquema ejemplo:

```
...CAAGTTTTCTACTGATTTAGATCAA...
...CAAGTTGTCTACTGATTTAGATCAG...
...CAAACTATCCACAGATTTAGATCAG...

...TTTTTTTTTTTTTTTTTTTTTTTT...
```

El siguiente paso es indicar que, donde los elementos de la segunda hasta la última fila sean iguales que los de la primera, para una misma columna, se mantendrá el valor “T” en la lista que habremos creado, y donde no se cumpla esta condición (donde no haya alineamiento total) pondremos, en nuestra lista, una “F” (*false*). Se muestra también un esquema ejemplo:

```
...CAAGTTTTCTACTGATTTAGATCAA...
...CAAGTTGTCTACTGATTTAGATCAG...
...CAAACTATCCACAGATTTAGATCAG...

...TTTFFTFTTFTTFTTTTTTTTTTTF...
```

Después, se recorre la lista creada (con valores “T” y “F”), y se crea una nueva lista donde se establece el valor “T” si el elemento de la lista anteriormente creada, en la misma columna, y los L siguientes valores son todos “T”. Se establecerá “F” si el elemento en la lista anterior, correspondiente a la misma columna, es “F” o alguno de los N siguientes es “F”. Esto nos permite tener una lista que indica “T” si el nucleótido de esa columna en el alineamiento es el inicio de un conjunto de N nucleótidos alineados por completo, y por tanto puede ser considerado, dicho conjunto, como una secuencia común de longitud N para todas nuestras cepas. A continuación, se muestra también un esquema ejemplo.

```
...TTTFFTFTTFTTFTTTTTTTTTTF...
...FFFFFFFFFFFFFTTTTTTTFalse...

...TTTFFTFTTFTTFTTTTTTTTTTF...
...              True        ...
...               True       ...
...                True      ...
...                 True     ...
...                  True    ...
...                   True   ...
...                    True  ...
```

Los resultados son las secuencias, en el alineamiento, que corresponden a estas posiciones que hemos nombrado. Se muestra un ejemplo visual:

```
...CAAGTTTTCTACTGATTTAGATCAA...
...CAAGTTGTCTACTGATTTAGATCAG...
...CAAACTATCCACAGATTTAGATCAG...

               TTTTTTT

               GATTT
                ATTTA
                 TTTAG
                  TTAGA
                   TAGAT
                    AGATC
                     GATCA
```

Estas secuencias se guardan en un archivo que denominamos “multiSeqs”. Al final, añadimos una función que descarta aquellas secuencias identificadas cuyo nucleótido final sea “G”, ya que este no será válido como secuencia PFS para la herramienta de diseño de sgRNAs que vamos a emplear.

En el Anexo 02 se detallan todas las funciones empleadas en esta fase.

3.3.2 Algoritmo de búsqueda exhaustiva

A diferencia del alineamiento global múltiple que hemos llevado a cabo en el apartado anterior, ahora vamos a comentar la implementación de un método de búsqueda exhaustiva. Lo que hace es buscar cada subsecuencia de longitud deseada (que habremos introducido) de una de las cepas en el resto, para averiguar si está presente en todas y, por ello, puede ser considerada una secuencia común en todas las cepas.

3.3.2.1 Características, ventajas e inconvenientes

Esta estrategia no requiere la instalación de aplicaciones bioinformáticas; puede implementarse con funciones sencillas de Python, como veremos a continuación.
Esta puede ser considerada una ventaja, junto con el hecho de que nos aseguramos de que cada secuencia considerada como común mediante este método, realmente lo es, y las que no son consideradas, realmente no son comunes. Esto último no ocurría con el método anterior, ya que el alineamiento global, como hemos visto, obvia secuencias cortas con alineamiento total por el bien del alineamiento global, e introduce *gaps* en muchas ocasiones.
Sin embargo, el principal inconveniente es que el coste computacional para llevar a cabo este método será mayor.

3.3.2.2 Desarrollo

Para llevar a cabo esta estrategia se han probado dos métodos distintos. Ambos empiezan por “cortar” cada una de las secuencias patógenas en subsecuencias de longitud determinada. El primer método consiste en “contar” cuántas veces aparece una subsecuencia en el conjunto de las cepas. El segundo método itera la búsqueda

de cada una de las secuencias de la primera cepa en el resto de las cepas, una a una. En el Anexo 02 se encuentra el código documentado para la primera estrategia (ya que es código que no será empleado en el script final) y en el Anexo 03 se encuentra el código documentado correspondiente al segundo método, que sí se implementa en el script.

Para la primera estrategia, igual que para el alineamiento con ClustalW, primero definimos la longitud deseada, L, que se convierte en L+1 inmediatamente para incluir la secuencia PFS.
Esta vez, las secuencias son introducidas, des del archivo FASTA, en una lista que denominamos “strains”. Los identificadores correspondientes se guardan en una lista “IDs” conservando el orden de las cepas (la primera secuencia de la lista “strains” se corresponde con el primer identificador de la lista “IDs”, etc).
Escribimos una función que cogerá cada subsecuencia de L nucleótidos de la primera cepa y buscará su presencia en el resto de las cepas, sumando “1” por cada cepa donde encuentre esta subsecuencia. Si el número resultante es igual al número de cepas en nuestra lista, la secuencia es común a todas las cepas.

Para el segundo método se utiliza la función *re.search* (contenida en el paquete *re* que habrá de ser importado). Las secuencias son introducidas en el objeto “strains”, y los identificadores en el objeto “IDs”, de la misma forma que en el método anterior. Sin embargo, ahora, lo que hacemos es buscar cada subsecuencia de la primera cepa en el resto de las cepas, una a una, y sólo si tras todas las iteraciones (para todas las cepas) la secuencia se ha encontrado en todas ellas, es considerada secuencia común.

En ambos casos, al final, se añade la misma función de descarte de secuencias terminadas con el nucleótido “G” que hemos empleado con ClustalW.

3.3.3 Comparación de métodos

En nuestro script final sólo se ha integrado uno de estos tres métodos desarrollados. Para decidir cuál se implementa, lo que se hizo fue comparar el funcionamiento de cada uno de ellos y algunos pequeños resultados obtenidos.

3.3.3.1 Primeras pruebas con secuencias de HPV

Para llevar a cabo las primeras pruebas sin que estas supongan un elevado coste computacional (ya que deberán ser ejecutadas e iteradas en múltiples ocasiones) utilizamos tres secuencias cortas, correspondientes a un gen determinado (E6) en tres cepas distintas de HPV (las cepas 1, 18 y 30). Estas se muestran a continuación:

```
>e6hpv1
atggcgacaccaatccggaccgtcagacagctttccgaaagcctctgtatcccatatattgatgttttattgccttgtaattttt
gtaattatttttttgtctaatgctgagaagctgctttttgatcattttgatttgcatcttgtctggagagacaatttggtgtttgg
atgctgtcaagggtgtgctagaactgttagcctattggagtttgttttatattatcaggagtcttatgaggtaccggaaatagaa
gaaattttggacagacctttattgcaaattgaactccgttgtgttacatgcataaaaaaaactgagtgttgctgaaaaattggagg
ttgtgtcaaacggagaaagagtgcatagagttagaaacagacttaaagcaaagtgtagtttgtgtcgcttgtatgctatataa

>e6hpv18
```

```
atggcgcgctttgaggatccaacacggcgaccctacaagctacctgatctgtgcacggaactgaacacttcactgcaagacatag
aaataacctgtgtatattgcaagacagtattggaacttacagaggtatttgaatttgcatttaaagatttatttgtggtgtatag
agacagtataccgcatgctgcatgccataaatgtatagatttttattctagaattagagaattaagacattattcagactctgtg
tatggagacacattggaaaaactaactaacactgggttatacaatttattaataaggtgcctgcggtgccagaaaccgttgaatc
cagcagaaaaacttagacaccttaatgaaaaacgacgatttcacaacatagctgggcactatagaggccagtgccattcgtgctg
caaccgagcacgacaggaacgactccaacgacgcagagaaacacaagtataa

>e6hpv30
atggctttcaaatttgaaaatacaggcgagcgcccacgtactgtgcaccatctttgtgaggtacaagaaacatcgttgctggagc
tacagctacagtgtgtatattgcaagaaggaattatccagctcagaggtatataattttgcatgtaaagatttaagactggtata
tagggaggacagcccatatgcagtgtgcaatttctgtttattattttatagtaaagtaagaaagattagacattacaactattca
ttgtatggggcaagcctagtggcattaactaaaaaagagttatttgatttattaataaggtgctacagatgtcaacagccgttga
caccagaggaaaaacagttacactgtgaatataagaaacggtttcacagaatatcacgtacgtggaccgggttatgtctgcaatg
ctggagacacacaacgtccactgagacagcagtataa
```

Como hemos visto anteriormente, en el código empleado para cada estrategia, la longitud deseada que indicábamos según cada método era distinta: en el caso del alineamiento múltiple, la longitud indicada era de 5 nucleótidos, que eran transformados en 6 al añadir el que corresponderá a la secuencia PFS. Esto es porque este método no encontraba una secuencia común en todas las cepas de mayor longitud que 6 nucleótidos, aunque, como veremos a continuación, esta sí existía. La subsecuencia obtenida es la siguiente:

```
>gaaaaa
```

En cambio, en el caso de los dos métodos de búsqueda exhaustiva (que dan, ambos, el mismo resultado) indicábamos una longitud de 7 nucleótidos que, al añadir la PFS, acaban siendo 8. Esta era la longitud máxima obtenida para las secuencias comunes mediante esta estrategia, y con ambos métodos de búsqueda exhaustiva, el resultado era igual:

```
>tattgcaa
```

3.3.3.2 Comparación de metodologías

En la ilustración 4 se puede ver la ejecución de un alineamiento múltiple con la aplicación web de ClustalW. Como vemos, el resultado obtenido de secuencia común es el mismo que el obtenido ejecutando ClustalW desde nuestro script, cuya longitud es de 6 nucleótidos. Al alinearse las secuencias de forma global, como se ha comentado, no se conservan las secuencias comunes de mayor longitud en el alineamiento porque están presentes en regiones muy distantes entre un genoma y otro.

Ilustración 4 – Resultados del alineamiento múltiple mediante ClustalW a través de la aplicación web.

A pesar de que ahora hemos utilizado secuencias cortas, y el uso de secuencias genómicas completas va a suponer un mayor coste computacional, nos quedaremos con el método de búsqueda exhaustiva, para integrar en nuestro script. Esto es porque la longitud que deberemos introducir para obtener nuestras secuencias será mayor, y las subsecuencias obtenidas pasarán por muchos cuellos de botella por los que podrán ser descartadas (si no tienen la PFS adecuada, si están presentes en el genoma del huésped, etc.). Por eso, necesitamos maximizar el número de secuencias múltiples obtenidas.

Concretamente, el método de búsqueda exhaustiva que emplearemos es el segundo desarrollado (mediante la función *re.search*) ya que es más sencillo a nivel de implementación.

3.4 Búsqueda de secuencias particulares

El objetivo de buscar secuencias particulares para cada una de las cepas es el potencial desarrollo de test diagnósticos de alta precisión, capaces de detectar la presencia de una cepa particular del organismo de interés. En el caso de HPV, por ejemplo, el principal interés es detectar, en un paciente del que se sospecha la presencia de la infección vírica, si la cepa presente es oncogénica (generadora de cáncer cervical y otros cánceres genitales).

3.4.1 Algoritmo de búsqueda exhaustiva

Al igual que en los apartados anteriores en los que se ha implementado el método de búsqueda exhaustiva, exploramos los dos métodos: mediante “contaje” de cepas en las que aparece cada subsecuencia de cada una de las cepas, y mediante la iteración de la búsqueda de dicha subsecuencia, perteneciente a una cepa, en todas las demás cepas. diferencia del alineamiento global múltiple que hemos llevado a cabo en el apartado anterior, ahora vamos a implementar un método de búsqueda exhaustiva.

3.4.1.1 Diseño

Para el método de “contaje”, lo que se hace es cortar cada secuencia en subsecuencias de longitud determinada, y sumar 1 en un objeto cada vez que dicha subsecuencia sea encontrada en una cepa. El resultado deberá ser igual a 1 (la presencia de la subsecuencia en su cepa de origen) para que sea considerada particular de dicha secuencia original.

Para el método de iteración de la búsqueda, se corta de igual forma cada secuencia en subsecuencias de longitud deseada, y se itera la búsqueda de cada una de estas en las cepas que no son aquella a la que pertenece. Sólo si esta no es encontrada en ninguna de las demás cepas, será considerada particular de su secuencia de origen.

3.4.1.2 Desarrollo

Para el desarrollo del primer método, escribimos una función que, al igual que para las secuencias comunes, primero define cada subsecuencia de cada una de las cepas, de longitud L, como un elemento, y suma “1” cada vez que encuentra dicho elemento en alguna de las cepas analizadas. Si la suma final es igual a 1, correspondiente a la presencia del elemento en la cepa de origen de la que proviene, es una secuencia particular de esta cepa y se guarda en un objeto llamado “SingleS”.

Para el segundo método, escribimos una función que, de la misma forma, corta cada una de las secuencias en subsecuencias de longitud L, y busca su presencia, esta vez con la función *re.search* en el resto de cepas, una a una, a modo de iteración. Solo si la búsqueda da resultado nulo en todas las iteraciones se considera la secuencia como particular en su cepa de origen y se guarda en el objeto “SingleS”.

El código documentado y detallado para la primera estrategia se encuentra en el Anexo 02, y el de la segunda, en el Anexo 03 ya que es el que se implementará en el script final.

Dado que, como hemos dicho antes, la implementación resulta más sencilla y limpia, y para que el método de búsqueda de secuencias particulares sea acorde con el de búsqueda de secuencias múltiples, implementaremos la segunda estrategia (la de iteración de la búsqueda exhaustiva) en nuestro script final.

3.4.1.3 Pruebas con secuencias de HPV

Al probar esta búsqueda con las secuencias cortas de HPV nombradas anteriormente, estos son algunos de los resultados obtenidos (contenidos en el objeto SingleS). Se muestran los 10 primeros, correspondientes a la primera cepa (HPV tipo 1):

```
>atggcgac
>tggcgaca
>ggcgacac
>gcgacacc
>cgacacca
>gacaccaa
>acaccaat
>caccaatc
>accaatcc
>aatccgga
```

3.5 Alineamiento local con el huésped

El siguiente paso en el desarrollo de la herramienta es comprobar cuáles de las secuencias, comunes y particulares, que acabamos de obtener, están presentes en el huésped de interés, y cuáles no. Para ello, cotejamos cada una con el genoma de nuestro huésped (*H.sapiens*, en nuestro caso, cuyo genoma se utiliza como imput en forma de archivo en formato FASTA) y descartamos aquellas que sí estén presentes en este.

El interés de llevar a cabo este paso es que, potencialmente, nuestras secuencias serán implementadas en un test diagnóstico que podrá ser aplicado a muestras biológicas de un paciente para detectar la presencia del patógeno (por ejemplo, un frotis bucal). Es necesario que las secuencias a buscar no estén presentes en el genoma del huésped porque dichas muestras estarán altamente contaminadas por secuencias de dicho huésped. Es un método de evitar falsos positivos en nuestros test.

3.5.1 BLAT

El método que empleamos para evaluar la presencia de nuestras secuencias en el genoma huésped es un alineamiento mediante BLAT.

BLAT (*BLAST-like alignment tool*) es un método de alineamiento pareado local que permite cotejar secuencias cortas frente a un genoma de interés, para buscar homologías. Es un método rápido y óptimo, que, mediante la línea de comandos, permite buscar secuencias cortas determinadas en un genoma contenido en un archivo a nivel local.

3.5.2 Implementación

Los pasos que se deben seguir serán importar los paquetes necesarios para ejecutar BLAT, generar su imput y leer su output, guardar nuestras secuencias encontradas (comunes y particulares) en el formato que BLAT requiere como imput, ejecutarlo y transformar el output que genera al formato de nuestro interés.

Para ejecutar el alineamiento local mediante BLAT entre cada una de las subsecuencias encontradas y el genoma del huésped, importamos los paquetes Seq, SeqIO, SeqRecord y generic DNA, que son necesarios para generar e interpretar los formatos de entrada y salida de BLAT. También es necesario tener instalado el programa BLAT para poder ejecutarlo.
Primero, generamos un objeto de tipo "SeqRecord" para cada una de las secuencias comunes encontradas. Estos objetos contienen una secuencia y una serie de atributos, como un identificador, una descripción, etc. Es un paso intermedio para transformar nuestra lista de secuencias a formato FASTA.
A continuación, hemos transformado las secuencias a formato FASTA para poder ejecutar BLAT con ellas. Hemos ejecutado el programa mediante la función "subprocess", indicándole una serie de parámetros para acotar cómo queremos que sea nuestro alineamiento:

- *TileSize* indica el número mínimo de coincidencias que esperamos encontrar para considerar un alineamiento. Siendo de 8 nucleótidos la longitud de nuestras secuencias, indicamos "6" en este parámetro, ya que si hubiese coincidencia de menos de "6" nucleótidos consideraríamos que la secuencia es válida porque no es suficientemente similar al huésped.

- *StepSize* es el espacio entre *"tiles"*; por defecto es el mismo número que *"tileSize"* pero indicamos "1" porque nos interesa considerar alineamiento de cada subsecuencia empezando por cada nucleótido del huésped (no hacer una búsqueda de 6 en 6).

- *MinMatch* es el número mínimo de coincidencias en una *"tile"*; hemos indicado que el tamaño de match queremos que sea 6, así que como máximo habrá un match por *"tile"*; indicamos "1".

- *MinScore* es la puntuación mínima para considerar el alineamiento; considerados el resto de los parámetros lo podríamos obviar, pero por defecto el valor es 30, así que indicamos 1 para considerar aquellos de menor valor que 30. Por último, indicamos el formato de salida (*out*) que será *plsx*, que incluye en el archivo resultante las secuencias alineadas.

- Además, a BLAT le indicamos, en este orden, la base de datos (en nuestro caso, el archivo con el genoma del huésped), las *query* (nuestro archivo de secuencias múltiples o particulares) y el nombre del archivo de salida deseado, que se guardará en nuestro directorio.

A continuación, se ha escrito una función que busca, en el archivo resultante de BLAT (que contiene, además de los datos del alineamiento, las secuencias para las que ha encontrado alineamiento total con el huésped), la presencia de cada una de las secuencias, y guarda aquellas donde la búsqueda es nula: serán aquellas que no están presentes en el genoma del huésped.

Este proceso ha sido llevado a cabo de la misma forma con las secuencias particulares para cada una de las cepas, que han sido guardadas, además, con su identificador, para conservar siempre de qué secuencia original provienen.

Todas estas secuencias son nuestro primer output. El programa genera los dos primeros archivos de interés, que se guardan en nuestro directorio de trabajo: un archivo “MultResult.fa” con los resultados para las secuencias comunes, y un archivo “SingleResult.fa” con los resultados para las secuencias particulares de cada cepa y el identificador que indica de qué secuencia original proviene cada una.

En el Anexo 03 se presenta el código documentado que se ha empleado para esta fase del desarrollo.

3.5.3 Pruebas con secuencias de HPV y *H.sapiens*

Para ejecutar nuestras pruebas, en lugar de utilizar el genoma de *H.sapiens* completo todas las veces, lo que se ha hecho es utilizar como secuencia huésped un fragmento del cromosoma 10 humano de unos 10000 nucleótidos de longitud.

Tras aplicar este paso a las secuencias que habíamos obtenido en los pasos anteriores, conservamos el resultado de la única secuencia común (es decir, esta no estaba presente en el huésped introducido):

```
>tattgcaa
```

A continuación, se muestran los primeros 7 resultados para las secuencias particulares (estas pertenecen al HPV tipo 1) que no están presentes en el huésped introducido.

```
>aaaaaaac
>aaaaaact
>aaaactga
>aaacagac
>aaagcaaa
>aaagcctc
>aaagtgta
```

3.6 Acoplamiento a una herramienta CRISPR

Como se ha ido explicando reiteradamente, el objetivo de obtener estas secuencias es, principalmente, obtener potenciales RNAs guía para desarrollar test diagnósticos con el método SHERLOCK. Para ello es necesario transformar las secuencias obtenidas en CRISPR-RNAs que puedan acoplarse a la enzima Cas13a. Por esto, se utilizará una aplicación web mediante la cual podemos obtener CRISPR-RNAs idóneos para ser utilizados con dicha enzima, y el output que se va a obtener es una serie de links a la aplicación que devolverán secuencias RNA guía para SHERLOCK correspondientes a cada una de las subsecuencias que hemos obtenido anteriormente.

3.6.1 Herramienta CRISPR-RT

La herramienta que se ha escogido para acoplar a la desarrollada es CRISPR-RT (Zhu et. al., 2017). CRISPR-RT (CRISPR RNA-Targeting) es el primer servicio web per permite diseñar CRISPR-RNAs con especificidad de diana para el sistema CRISPR-C2c2 (o CRISPR-Cas13a).
Nos sirve para generar, a partir de nuestras secuencias, RNAs guía que puedan servir para el sistema SHERLOCK. La herramienta interpreta el último nucleótido de las secuencias introducidas como secuencia PFS, genera la complementariedad necesaria para el RNA guía, y añade las secuencias necesarias para que el investigador tenga claro cómo debe ser sintetizado. La ilustración 5 muestra una captura de pantalla de la interfaz de la herramienta.

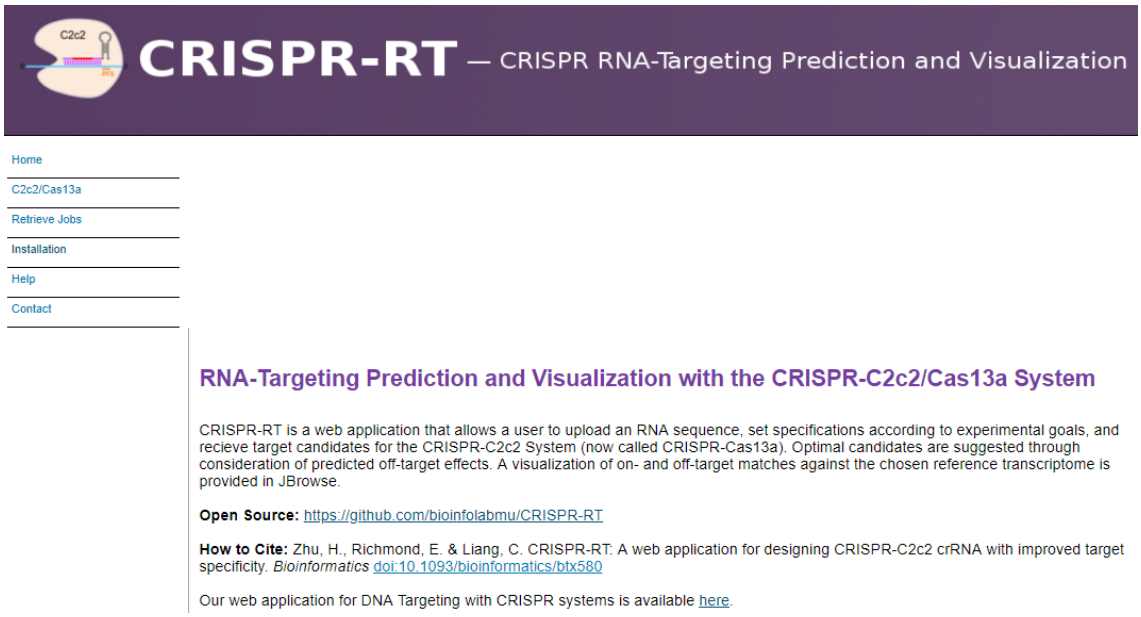

Ilustración 5 – Interfaz de la herramienta web CRISPR-RT.

3.6.2 Implementación

El primer paso para la implementación es transformar los nucleótidos “T” de las secuencias en “U” ya que CRISPR-RT requiere tránscritos como imput, y nosotros hemos trabajado con secuencias genómicas. También se han sustituido, en caso de estar las secuencias en minúsculas, todas las minúsculas por mayúsculas, ya que la herramienta CRISPR-RT también requiere este paso. Se lleva a cabo el mismo paso con las secuencias comunes y las particulares.

Se ha escrito una función que, a partir de las secuencias, genera enlaces a la herramienta que contienen dicha secuencia. Los enlaces deben contener en determinada posición, la secuencia completa, en otra posición, la PFS, y en otra, la secuencia sin la PFS; por eso, se han cortado las secuencias y se han introducido en los puntos clave en el enlace. Este es un ejemplo:

http://bioinfolab.miamioh.edu/CRISPR-RT/proc/crRNA_design.php?seq=UAUUGCAA&GRNA=UAUUGCA&PFS=A

Este paso genera nuestro segundo output: dos archivos de texto, que contienen los enlaces a la herramienta para las secuencias obtenidas. En el caso de las comunes, contiene sólo los enlaces, y en el caso de las particulares, contiene los enlaces y un identificador asociado a cada uno que corresponde al original de la secuencia introducida (a la cepa).

En el Anexo 03 se presenta el código documentado que se ha empleado para esta fase del desarrollo.

3.6.3 Pruebas con las secuencias obtenidas

Se muestran aquí algunos ejemplos de los resultados output para las secuencias comunes y particulares.

En el caso de las comunes, con las secuencias y la longitud que habíamos introducido sólo obteníamos una, cuyo enlace es:

http://bioinfolab.miamioh.edu/CRISPR-RT/proc/crRNA_design.php?seq=UAUUGCAA&gRNA=UAUUGCAA&PFS=A

En la ilustración 6 vemos una captura que corresponde a la interfaz del sitio web al que nos dirige dicho enlace.

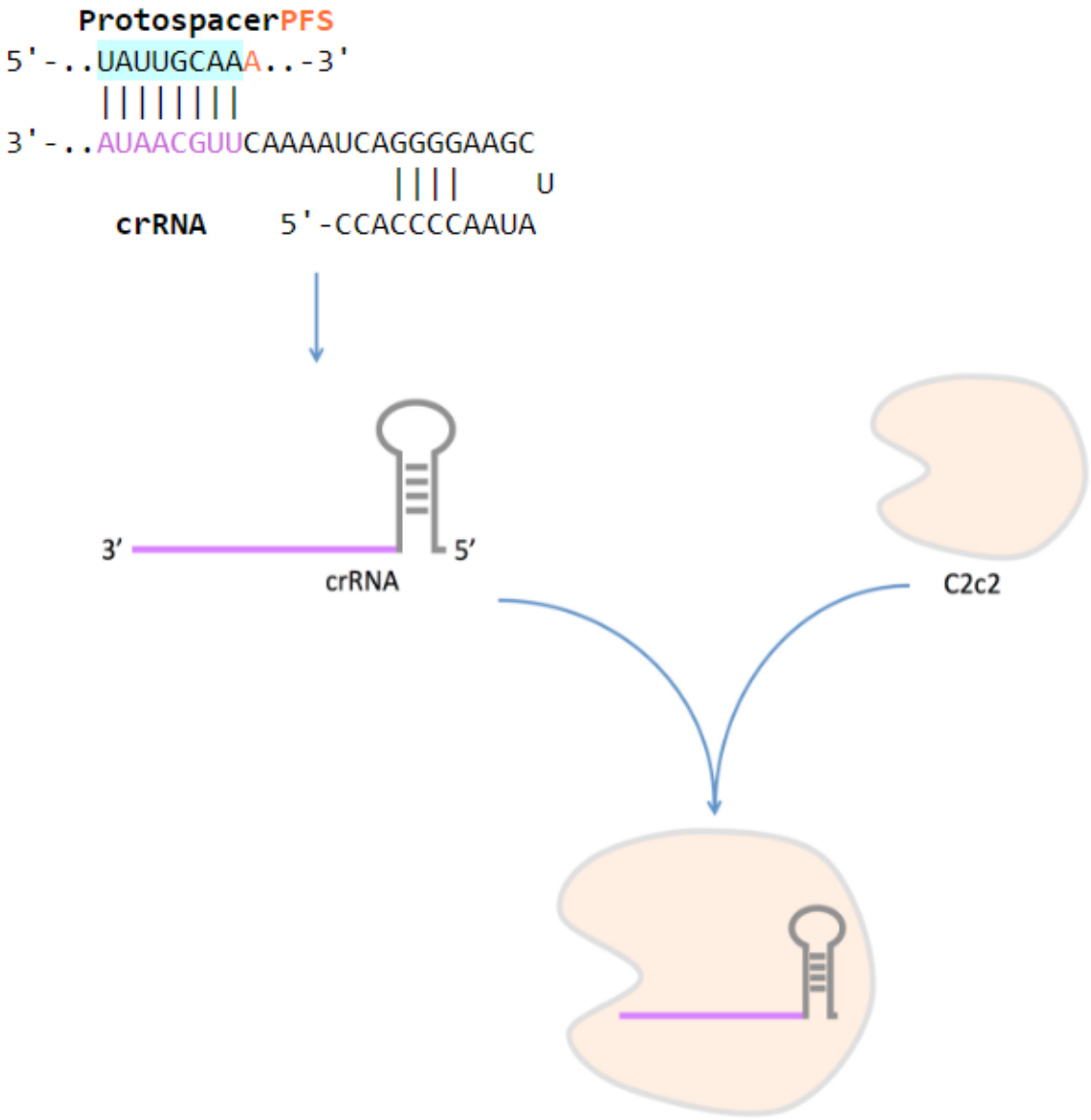

Ilustración 6 – Ejemplo de resultado del diseño de una secuencia RNA guía a través de CRISPR-RT a partir de una de las secuencias comunes identificadas.

En el caso de las particulares, a continuación, se muestra un ejemplo de enlace (correspondiente a la primera de las cepas introducidas, HPV tipo 1) y, en la ilustración 7, la captura correspondiente.

http://bioinfolab.miamioh.edu/CRISPR-RT/proc/crRNA_design.php?seq=AAAAAAC&gRNA=AAAAAAA&PFS=C

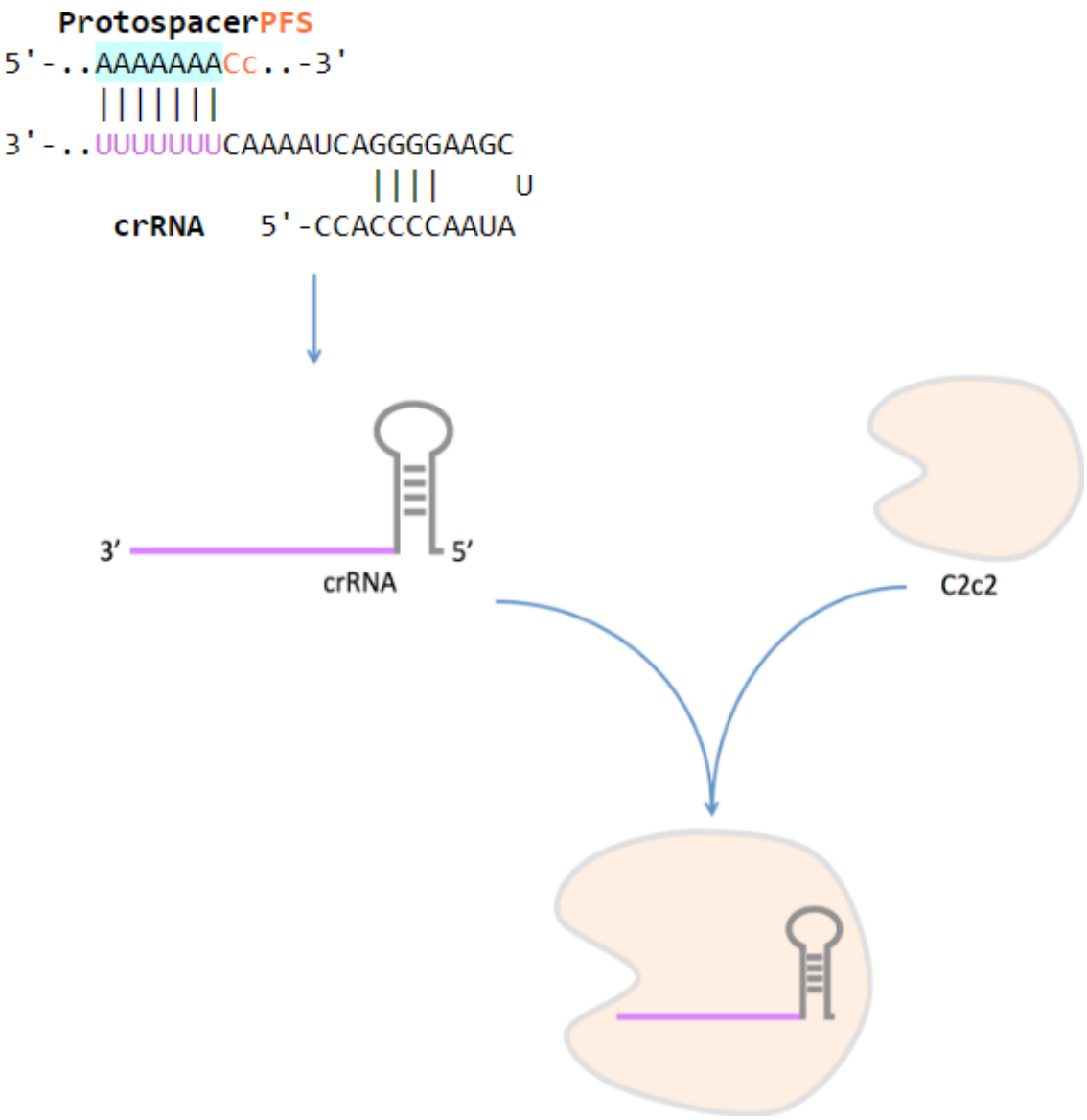

Ilustración 7 - Ejemplo de resultado del diseño de una secuencia RNA guía a través de CRISPR-RT a partir de una de las secuencias particulares identificadas.

4. Resultados en HPV y *H.sapiens*

4.1 HPV

Como se ha ido comentando, a pesar de que la herramienta es aplicable potencialmente a la detección de secuencias comunes y particulares entre cepas de cualquier patógeno, se ha aplicado a HPV y se ha testeado en secuencias de este. El principal interés es distinguir entre las cepas oncogénicas del virus y aquellas que no lo son, para identificar la presencia de determinada cepa en un paciente al que se le aplique el test diagnóstico desarrollado con el sistema SHERLOCK.

4.1.1 Virus del papiloma humano

El virus del papiloma humano (HPV, ilustración 8) son grupos de diversos virus de DNA pertenecientes a la familia de los *Papillomaviridae*. Representan una de las enfermedades de transmisión sexual más comunes. Se replican específicamente en el núcleo de las células epiteliales escamosas. Se transmiten por contacto piel a piel. Se conocen más de 100 tipos diferentes, que se clasifican según su patogenia oncológica en tipos de alto y bajo riesgo oncológico.

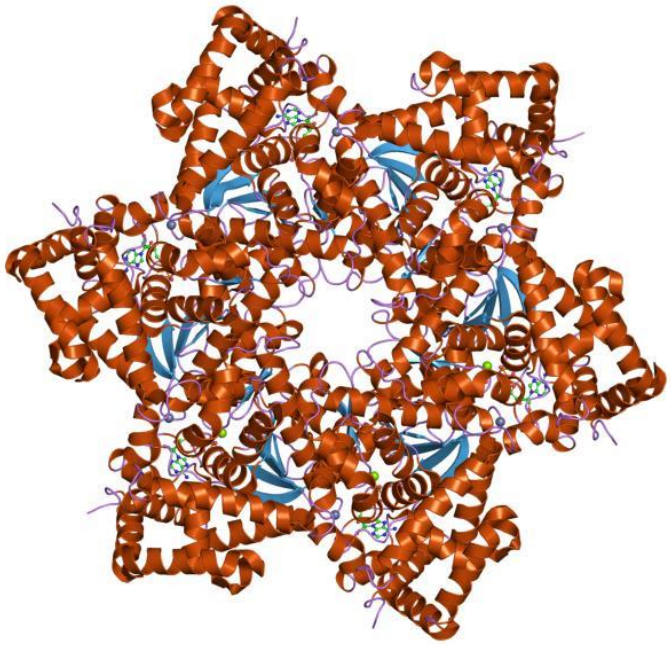

Ilustración 8 – Virus del Papilloma Humano.

4.1.2 Cepas oncogénicas

Los tipos 16, 18, 31, 33, 35, 39, 45, 51, 52, 56, 58, 59 y 66 son carcinógenos para los humanos (tipos de alto riesgo oncológico). Algunos pueden causar verrugas o condilomas, mientras que otros pueden generar infecciones subclínicas que pueden dar lugar a cáncer cervical, de vulva, de vagina y ano en mujeres, o cáncer de ano y pene en hombres.

4.2 Validación con secuencias completas de HPV y *H.sapiens*

4.2.1 Secuencias escogidas

Las secuencias escogidas para validar la herramienta desarrollada han sido, en el caso de HPV, el genoma completo de los tipos 4, 5, 9 y 16, y los genes E6, E7, E1, E2, E4, E5, L1 y L2 del tipo 18, ya que no está disponible en la base de datos el genoma completo de esta cepa, pero es de especial interés analizarla por su potencial oncogénico. Todas las secuencias han sido obtenidas de la base de datos Genome de NCBI. Los tipos 4, 5 y 9 no son de alto riesgo oncogénico, mientras que los tipos 16 y 18 sí. Como se ha comentado, una aplicación de la herramienta sería el diseño de test con el sistema SHERLOCK capaces de identificar la presencia de una de estas cepas oncogénicas, a partir de la hibridación de una secuencia que sólo se encuentre en el genoma del tipo 16 o del tipo 18 y no esté en ninguna otra cepa ni en el huésped. Para desarrollar este test con precisión, sin embargo, sería necesario cotejar entre sí todas las cepas distintas para asegurar que la secuencia encontrada realmente sólo pertenece a la cepa de interés. Se han escogido estas secuencias como muestra porque son tipos comunes.

La secuencia escogida como huésped, en esta fase de validación final, es el genoma completo de H.sapiens. Este ha sido descargado (des del terminal de Linux mediante el comando wget) para poder ejecutar el programa BLAT de forma local, del siguiente enlace de descarga del portal UCSC, donde se encuentra la última versión (hg38) del genoma humano completo:

http://hgdownload.cse.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz

4.2.2 Ejecución

Para ejecutar la herramienta des del terminal de Linux utilizamos un comando y una serie de parámetros indicando el script que se debe ejecutar (script.py es el nombre de nuestro archivo de código), la longitud deseada (que serán 20 nucleótidos), el archivo de entrada de las secuencias de HPV (que hemos guardado como HPV.fasta) y el archivo de entrada de la secuencia del genoma del huésped (que hemos guardado como Host.fasta). El comando es el siguiente:

```
python script.py -l 20 -p HPV.fasta -g Host.fasta
```

4.3 Resultados

A continuación, se muestran algunos ejemplos de los resultados obtenidos al ejecutar nuestro programa con las secuencias nombradas anteriormente. Con las secuencias testeadas, no se han encontrado secuencias comunes. Han sido probados varios sets de secuencias y varias longitudes, pero las probabilidades de encontrar secuencias comunes a todas las cepas de una longitud tan larga (en proporción a la longitud total del genoma de un virus) son pequeñas.

- Ejemplos de secuencias particulares encontradas para cada cepa:

```
>>JN565303.1 Human papillomavirus type 16 strain ZG01-258, complete genome
AAAAAAAAACAGGGGATGCTA
>>JN565303.1 Human papillomavirus type 16 strain ZG01-258, complete genome
AAAAAAAACAAATGAGTATGA
>>JN565303.1 Human papillomavirus type 16 strain ZG01-258, complete genome
AAAAAAAACAGGGGATGCTAT
…
>>M17463.1 Human papillomoavirus type 5, complete genome
AAAAAAAAATTGCATTTTAAT
>>M17463.1 Human papillomoavirus type 5, complete genome
AAAAAAAATTGCATTTTAATT
>>M17463.1 Human papillomoavirus type 5, complete genome
AAAAAAATTGCATTTTAATTT
…
>>X05015.1 Human papillomavirus type 18 E6, E7, E1, E2, E4, E5, L1 & L2 genes
AAAAAAAATTGTTTAGTATTT
>>X05015.1 Human papillomavirus type 18 E6, E7, E1, E2, E4, E5, L1 & L2 genes
AAAAAAACAGGAGATGTAATA
>>X05015.1 Human papillomavirus type 18 E6, E7, E1, E2, E4, E5, L1 & L2 genes
AAAAAAATTGTTTAGTATTTT
…
>>X70827.1 Human papillomavirus type 4 complete genome
AAAAAAAAACTGTATAGTTAT
>>X70827.1 Human papillomavirus type 4 complete genome
AAAAAAAACTGTATAGTTATT
>>X70827.1 Human papillomavirus type 4 complete genome
AAAAAAACTGTATAGTTATTC
…
>>X74464.1 Human papillomavirus type 9 genomic DNA
AAAAAAAAAAGAGGCTTAGTA
>>X74464.1 Human papillomavirus type 9 genomic DNA
AAAAAAAAAGAGGCTTAGTAA
>>X74464.1 Human papillomavirus type 9 genomic DNA
AAAAAAAAGAGGCTTAGTAAA
…
```

- Ejemplos de enlaces a la herramienta CRISPR para las secuencias particulares:

```
>JN565303.1 Human papillomavirus type 16 strain ZG01-258, complete genome
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAAACAGGGGAUGCUA&gRNA=AAAAAAAAACAGGGGAUGCU&P
FS=A
>JN565303.1 Human papillomavirus type 16 strain ZG01-258, complete genome
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAACAAAUGAGUAUGA&gRNA=AAAAAAAACAAAUGAGUAUG&P
FS=A
>JN565303.1 Human papillomavirus type 16 strain ZG01-258, complete genome
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAACAGGGGAUGCUAU&gRNA=AAAAAAAACAGGGGAUGCUA&P
FS=U
…
>M17463.1 Human papillomoavirus type 5, complete genome
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAAAUUGCAUUUUAAU&gRNA=AAAAAAAAAUUGCAUUUUAA&P
FS=U
>M17463.1 Human papillomoavirus type 5, complete genome
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAAUUGCAUUUUAAUU&gRNA=AAAAAAAAUUGCAUUUUAAU&P
FS=U
```

```
>M17463.1 Human papillomoavirus type 5, complete genome
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAUUGCAUUUUAAUUU&gRNA=AAAAAAAUUGCAUUUUAAUU&P
FS=U
…
>X05015.1 Human papillomavirus type 18 E6, E7, E1, E2, E4, E5, L1 & L2 genes
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAAUUGUUUAGUAUUU&gRNA=AAAAAAAAUUGUUUAGUAUU&P
FS=U
>X05015.1 Human papillomavirus type 18 E6, E7, E1, E2, E4, E5, L1 & L2 genes
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAACAGGAGAUGUAAUA&gRNA=AAAAAAACAGGAGAUGUAAU&P
FS=A
>X05015.1 Human papillomavirus type 18 E6, E7, E1, E2, E4, E5, L1 & L2 genes
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAUUGUUUAGUAUUUU&gRNA=AAAAAAAUUGUUUAGUAUUU&P
FS=U
…
>X70827.1 Human papillomavirus type 4 complete genome
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAAACUGUAUAGUUAU&gRNA=AAAAAAAAACUGUAUAGUUA&P
FS=U
>X70827.1 Human papillomavirus type 4 complete genome
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAACUGUAUAGUUAUU&gRNA=AAAAAAAACUGUAUAGUUAU&P
FS=U
>X70827.1 Human papillomavirus type 4 complete genome
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAACUGUAUAGUUAUUC&gRNA=AAAAAAACUGUAUAGUUAUU&P
FS=C
…
>X74464.1 Human papillomavirus type 9 genomic DNA
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAAAAGAGGCUUAGUA&gRNA=AAAAAAAAAAGAGGCUUAGU&P
FS=A
>X74464.1 Human papillomavirus type 9 genomic DNA
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAAAGAGGCUUAGUAA&gRNA=AAAAAAAAAGAGGCUUAGUA&P
FS=A
>X74464.1 Human papillomavirus type 9 genomic DNA
http://bioinfolab.miamioh.edu/CRISPR-
RT/proc/crRNA_design.php?seq=AAAAAAAAGAGGCUUAGUAAA&gRNA=AAAAAAAAGAGGCUUAGUAA&P
FS=A
```

En el Anexo 05 se presentan los archivos que el programa devuelve como output.

5. Aplicación del producto desarrollado

La rapidez, precisión, especificidad y bajo coste del sistema SHERLOCK como potencial método diagnóstico de infecciones podría ser revolucionario. En el caso explorado, de HPV, la detección rápida y precoz de una cepa particular del virus puede suponer una mejora en las probabilidades de curación del paciente y una forma de evitar el desarrollo de cánceres, en caso de tratarse de una cepa oncogénica. Puede suponer una inmensa ventaja especialmente en zonas subdesarrolladas, por su modo de uso sencillo y su bajo coste.

La herramienta desarrollada en este trabajo permite obtener de forma igualmente rápida las secuencias más idóneas para el diseño de test mediante SHERLOCK con alta especificidad y reduciendo al máximo la probabilidad de falsos positivos por contaminación del genoma del huésped. De esta forma, puede emplearse el test en muestras prácticamente sin purificar, provenientes de un frotis o una muestra de sangre. Permite, por lo tanto, el proceso asociado al diseño de estos test, para incrementar aún más su eficiencia.

6. Conclusiones

La mayor aportación de este trabajo a mi aprendizaje ha sido el desarrollo de un programa informático. Hasta cursar este plan de estudios, nunca había programado software, y ha sido un perfecto complemento para las asignaturas de programación que he realizado. He comprendido la dificultad de automatizar procesos que, de entrada, pueden parecer sencillos y, a la vez, los beneficios que supone.
Además, he podido aprender mucho sobre los métodos de edición genómica existentes, particularmente sobre el sistema CRISPR-Cas, y sobre la estrecha relación entre la informática y la biología molecular. Probablemente la aplicación del sistema CRISPR-Cas para la edición y detección de secuencias sea uno de los métodos más revolucionarios del siglo en el campo de las ciencias de la vida, y aportar una herramienta que tal vez pueda ser aplicada experimentalmente algún día es muy satisfactorio.

Los objetivos fijados al principio han sido cumplidos, aunque durante el desarrollo han surgido múltiples ideas de mejora, y no todas ellas han podido ser implementadas.
La planificación temporal no ha sido la más ajustada, dado que el propio desarrollo de la herramienta ha requerido mucho más tiempo el esperado, y esto ha hecho que no haya podido dedicar tanto tiempo a adquirir más conocimiento sobre los sistemas CRISPR-Cas y SHERLOCK, o sobre el HPV, o a la redacción final del trabajo. Sin embargo, la herramienta desarrollada y los resultados obtenidos son los hitos más importantes del proyecto.
La metodología ha incluido varias iteraciones en el desarrollo de cada una de las funciones de la herramienta. Esto ha retrasado el desarrollo y, probablemente, hubiese sido mejor evaluar des del inicio cuál era la mejor estrategia de diseño para cada una de las partes, para poder ejecutarlas una sola vez. Sin embargo, me ha aportado conocimiento sobre el funcionamiento de metodologías de alineamiento como, por ejemplo, ClustalW.

De cara al futuro, hay múltiples mejoras que pueden ser aplicadas a la herramienta para poder optimizarla y aplicarla experimentalmente.
De cara a la ejecución, deberían llevarse a cabo pruebas con un mayor set de secuencias patógenas, para asegurar la presencia de secuencias comunes entre ellas, y también para asegurar la especificidad de un test para una cepa concreta (ya que analizando cinco cepas, sólo nos aseguramos, por cada subsecuencia de una cepa, de la no-presencia en las otras cuatro, pero no en el conjunto completo).
Podrían establecerse distintos sistemas de puntuación que permitan conocer, de cada conjunto de secuencias, cuales son óptimas para el diseño de los test diagnósticos. Esto podría basarse en establecer, para las secuencias particulares, cuales son más distintas de una cepa a otra, y para todas, cuales difieren más del genoma del huésped.
También podrá desarrollarse una interfaz con la que el usuario pudiese interactuar cómodamente.
La validación final del método debería consistir en alguna prueba experimental: probar de desarrollar un test con el sistema SHERLOCK con algunas de las secuencias obtenidas y testearlo con muestras biológicas reales para evaluar el resultado.

7. Glosario

BLAT: *Blast-like alignment tool*, método de alineamiento pareado local.

Cas: Familia de enzimas nucleasas de escisión de doble cadena.

Cas9: Nucleasa de la familia Cas implicada en el sistema CRISPR-Cas9 de corte de DNA de doble cadena

Cas13a / C2c2: Nucleasa de la familia Cas implicada en el sistema CRISPR-Cas13a de corte de RNA, en el que se basa en sistema SHERLOCK.

ClustalW: Proveniente de "*cluster analysis of the pairwise alignments*", método de alineamiento multiple.

CRISPR: *Clustered regularly interspaced short palindromic sequences.* Secuencias implicadas en la inmunidad adaptativa bacteriana.

CRISPR-Cas: Conjunción entre secuencias CRISPR y nucleasas de la familia Cas, sistema de inmunidad adaptativa bacteriana.

crispRNA / crRNA: RNA obtenido de la transcripción de una secuencia CRISPR para guiar el corte de la enzima Cas sobre el DNA diana.

DNA: *Desoxiribonucleic acid.*

EGFP: *Enhanced Green Fluorescent Protein.*

FASTA: Formato de texto para representar secuencias nucleotídicas o proteínas.

***H.sapiens*:** Abreviatura de *Homo sapiens*.

HEPN: *Higher eukaryotes and prokaryotes nucleotide,* dominio de enzimas nucleasas.

HNH: Dominio de enzimas nucleasas con residuos histidina y asparagina.

HPV: *Human papiloma virus* (virus del papiloma humano).

LwaCas13a: Enzima Cas13a presente en el organismo *Leptotrichia wadei*.

PAM: *Protospacer associated motiv*. Secuencia corta adyacente al DNA diana en el sistema CRISPR-Cas, en varios tipos de Cas.

PFS: *Protospacer flanking sequence*. Secuencia corta adyacente al DNA diana en el sistema CRISPR-Cas13a.

RNA: *Ribonucleic acid.*

RNAsa: Enzima de escisión de RNA.

RuvC: Dominio nucleasa para la reparación del DNA en *E.coli*.

sgRNA: *Single-guided RNA*, secuencia guía para el sistema CRISPR-Cas resultante de la síntesis en una sola secuencia de crRNA y trRNA.

SHERLOCK: *Specific high sensitivity enzymatic reporter unlocking*. Método de detección de la presencia de secuencias basado en el sistema CRISPR-Cas13a.

TALEN: *Transcription activator-like effector nuclease.* Método de edición genómica por nucleasas.

tracrRNA / trRNA: *Trans-activating RNA*. Secuencia necesaria para la activación de Cas9 (y otros tipos de Cas) en conjunción con el crRNA.

ZFNs: *Zinc fingers*. Método de edición genómica por nucleasas.

8. Bibliografía

[1] Musunuru, K. (2017). The Hope and Hype of CRISPR-Cas9 Genome Editing. *JAMA Cardiology*, 2(8), p.914.

[2] Gootenberg, J., Abudayyeh, O., Lee, J., Essletzbichler, P., Dy, A., Joung, J., Verdine, V., Donghia, N., Daringer, N., Freije, C., Myhrvold, C., Bhattacharyya, R., Livny, J., Regev, A., Koonin, E., Hung, D., Sabeti, P., Collins, J. and Zhang, F. (2017). Nucleic acid detection with CRISPR-Cas13a/C2c2. *Science*, 356(6336), pp.438-442.

[3] Hsu, P., Lander, E. and Zhang, F. (2014). Development and Applications of CRISPR-Cas9 for Genome Engineering. *Cell*, 157(6), pp.1262-1278.

[4] Tian, P., Wang, J., Shen, X., Rey, J., Yuan, Q. and Yan, Y. (2017). Fundamental CRISPR-Cas9 tools and current applications in microbial systems. *Synthetic and Systems Biotechnology*, 2(3), pp.219-225.

[5] Ran, F., Hsu, P., Wright, J., Agarwala, V., Scott, D. and Zhang, F. (2013). Genome engineering using the CRISPR-Cas9 system. *Nature Protocols*, 8(11), pp.2281-2308.

[6] Fu, Y., Reyon, D. and Joung, J. (2014). Targeted Genome Editing in Human Cells Using CRISPR/cas Nucleases and Truncated Guide RNAs. *Methods in Enzymology*, 546, pp.21-45.

[7] McDade, J. (2018). *CRISPR 101: Targeting RNA with Cas13a (C2c2)*. [online] Blog.addgene.org. Available at: http://blog.addgene.org/crispr-101-targeting-rna-with-cas13a-c2c2 [Accessed 5 Mar. 2018].

[8] Broad Institute. (2018). *Sherlock: Detecting disease with CRISPR*. [online] Available at: https://www.broadinstitute.org/videos/sherlock-detecting-disease-crispr [Accessed 5 Mar. 2018].

[9] Clontech.com. (2018). *Tools for Guide RNA Design*. [online] Available at: http://www.clontech.com/US/Products/Genome_Editing/CRISPR_Cas9/Resources/Online_tools_for_guide_RNA_design [Accessed 5 Mar. 2018].

[10] Biolabs, N. (2018). *CRISPR/Cas9 and Targeted Genome Editing:A New Era in Molecular Biology | NEB*. [online] Neb.com. Available at: https://www.neb.com/tools-and-resources/feature-articles/crispr-cas9-and-targeted-genome-editing-a-new-era-in-molecular-biology [Accessed 5 Mar. 2018].

[11] Capecchi, M. (2005). Gene targeting in mice: functional analysis of the mammalian genome for the twenty-first century. *Nature Reviews Genetics*, 6(6), pp.507-512.

[12] Jinek, M., Chylinski, K., Fonfara, I., Hauer, M., Doudna, J. and Charpentier, E. (2012). A Programmable Dual-RNA-Guided DNA Endonuclease in Adaptive Bacterial Immunity. *Science*, 337(6096), pp.816-821.

[13] Deltcheva, E., Chylinski, K., Sharma, C., Gonzales, K., Chao, Y., Pirzada, Z., Eckert, M., Vogel, J. and Charpentier, E. (2011). CRISPR RNA maturation by trans-encoded small RNA and host factor RNase III. *Nature*, 471(7340), pp.602-607.

[14] Nishimasu, H., Ran, F., Hsu, P., Konermann, S., Shehata, S., Dohmae, N., Ishitani, R., Zhang, F. and Nureki, O. (2014). Crystal Structure of Cas9 in Complex with Guide RNA and Target DNA. *Cell*, 156(5), pp.935-949.

[15] Swarts, D., Mosterd, C., van Passel, M. and Brouns, S. (2012). CRISPR Interference Directs Strand Specific Spacer Acquisition. *PLoS ONE*, 7(4), p.e35888.

[16] Qi, L., Larson, M., Gilbert, L., Doudna, J., Weissman, J., Arkin, A. and Lim, W. (2013). Repurposing CRISPR as an RNA-Guided Platform for Sequence-Specific Control of Gene Expression. *Cell*, 152(5), pp.1173-1183.

[17] Gasiunas, G., Barrangou, R., Horvath, P. and Siksnys, V. (2012). Cas9-crRNA ribonucleoprotein complex mediates specific DNA cleavage for adaptive immunity in bacteria. Proceedings of the National Academy of Sciences, 109(39), pp.E2579-E2586.

[18] Chen, B., Gilbert, L., Cimini, B., Schnitzbauer, J., Zhang, W., Li, G., Park, J., Blackburn, E., Weissman, J., Qi, L. and Huang, B. (2014). Dynamic Imaging of Genomic Loci in Living Human Cells by an Optimized CRISPR/Cas System. *Cell*, 156(1-2), p.373.

[19] Zhu, H., Richmond, E. & Liang, C. CRISPR-RT: A web application for designing CRISPR-C2c2 crRNA with improved target specificity. *Bioinformatics.*

9. Anexos – Índice

- **Anexo 01**: Plan de trabajo. *Descripción de la planificación temporal a modo de calendario que se ha seguido para el desarrollo del trabajo. Archivo PDF.*
- **Anexo 02**: Código desarrollado no utilizado (carpeta comprimida).
    - Anexo 02.1: ClustalW. *Desarrollo de código y comentarios para el alineamiento global múltiple de las secuencias patógenas mediante ClustalW. Archivo de texto simple.*
    - Anexo 02.2: Búsqueda exhaustiva de secuencias comunes mediante contaje. *Desarrollo de código y comentarios para la búsqueda exhaustiva de secuencias comunes mediante el primer método explicado y no implementado en el script final. Archivo de texto simple.*
    - Anexo 02.3: Búsqueda exhaustiva de secuencias particulares mediante contaje. *Desarrollo de código y comentarios para la búsqueda exhaustiva de secuencias particulares mediante el primer método explicado y no implementado en el script final. Archivo de texto simple.*
- **Anexo 03**: Script.py. *Archivo Python, contenido en una carpeta comprimida, de código documentado, script final que integra todas las funciones empleadas en la herramienta y que deberá ser ejecutado desde la línea de comandos como se indica en este trabajo para obtener los outputs descritos.*
- **Anexo 04**: Secuencias HPV. *Secuencias de HPV empleadas en la validación de la herramienta. Carpeta comprimida, contiene un archivo FASTA.*
- **Anexo 05**: Resultados (carpeta comprimida).
    - Anexo 05.1: SingleResult. *Resultados obtenidos mediante la ejecución de la herramienta, de secuencias particulares para cada cepa, de la longitud determinada por el usuario, que no están presentes en el huésped. Archivo FASTA.*
    - Anexo 05.2: SingleCRISPR. *Enlaces a la herramienta CRISPR-RT asociados a las secuencias particulares encontradas. Archivo de texto simple.*