

# Trabajo Fin de Máster

Máster en Bioinformática

Universidad de Murcia

*Junio 2014*

## NWK exchanger: Un sistema flexible de integración y filtrado de secuencias de proteínas

**Autor:** Diego Bastida Hernández

**Tutores:**

- Prof. Álvaro Sánchez Ferrer
- Prof. Jesualdo Tomás Fernández Breis

# Índice de contenido

1. Resumen/Abstract.....	3
2. Introducción.....	4
3. Materiales y métodos.....	7
3.1. Obtención de secuencias de proteínas.....	7
3.2. Bases de datos usadas.....	7
3.2.1. UniProt.....	7
3.2.2. NCBI.....	8
3.2.3. Ensembl.....	8
3.2.4. PFAM.....	8
3.3. Homogenización de la información.....	8
3.4. Integración de la información.....	10
3.5. Filtrado de la información biológica.....	10
3.6. Herramientas de programación usadas.....	11
4. Resultados.....	12
4.1. Diseño de la solución.....	12
4.2. Descripción de la aplicación.....	12
4.2.1. Entradas y salidas.....	12
4.2.2. Editar.....	14
4.2.3. Filtrados.....	15
4.2.4. Opciones de configuración.....	16
4.3. Evolución del programa.....	16
4.4. Ejemplo de ejecución.....	17
4.4.1. A partir de un fichero FASTA.....	17
4.4.2. A partir de un fichero árbol en Newick.....	21
4.5. Evaluación del programa diseñado.....	22
5. Discusión.....	24
Bibliografía.....	25

# 1. Resumen/Abstract

Este proyecto está relacionado con el desarrollo de un programa para ayudar a los investigadores a encontrar secuencias de proteínas/genes de las bases de datos actuales para homogeneizar y categorizar la información disponible de una forma rápida y flexible, siendo el primero descrito en la bibliografía en hacer esto.

El flujo del programa comienza con la obtención del identificador de la proteína a partir de un fichero de entrada (FASTA o Newick) para continuar con la recuperación de los datos de diferentes bases de datos (UniProt, NCBI, PFAM y Ensembl) y obtener una tabla personalizada con la información más relevante para publicaciones científicas. Después de usar diferentes tipos de filtros, que incluyen confiabilidad, tamaño, similitud y organización de los dominios de la secuencia, el programa produce diferentes formatos de salida, incluyendo ficheros de texto necesarios para obtener figuras listas para su publicación, con la herramienta web iTOL (interactive Tree Of Life). Este tipo de figuras muestran la arquitectura de dominios de la proteína para cada secuencia en un árbol filogenético, siendo una herramienta poderosa para encontrar nuevas proteínas con diferentes propiedades bioquímicas o estructurales. Además, el programa permite el uso de árboles previamente publicados, para actualizarlos o usarlos como punto de inicio de nuevos experimentos de búsqueda, siendo esta la última característica el rol esencial de la bioinformática.

**Palabras clave:** proteínas, árboles filogenéticos, herramienta bioinformática, dominios estructurales.

This project is related to the development of a program to assist researchers in finding protein/gene sequences from the current databases in order to homogenize and categorize the information available in a fast and flexible way, being the first described in the bibliography to do so.

The workflow of the program starts with finding the protein ID from the input file (FASTA or Newick) and continue with the retrieval of the data from different databases (UniProt, NCBI, PFAM and Ensembl) to obtain a customized table with most relevant information for paper publication. After using different kind of filters, which includes reliability, size, similarity and domain organization of the sequence, the software produce different output formats, including the text files needed to obtain paper-ready figures with the web tool known as iTOL (interactive Tree Of Life). These kind of figures show the protein domain architecture of each sequence in a phylogenetic tree, becoming a powerful tool for finding new proteins with different biochemical or structural properties. In addition, the program allows the use of previously published trees, to update them or to be used as a starting point for new research experiments, being this last feature the essential role of bioinformatics.

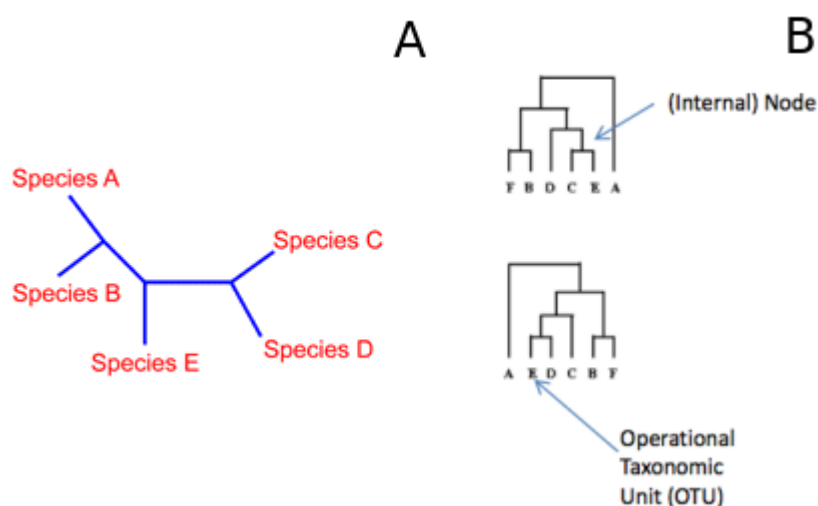
**Keywords:** proteins, filogenetic trees, bioinformatic tool, structural domains.

## 2. Introducción

El análisis comparativo de secuencias es una importante herramienta para bioquímicos y genéticos (Holder and Lewis, 2003). Minutos después de obtener una nueva secuencia, las búsquedas BLAST pueden descubrir indicios sobre las funciones y otras propiedades de un gen o de una proteína. La comparación de varias secuencias puede mostrar qué partes están cambiando rápidamente (y por tanto pueden estar menos limitadas funcionalmente), y qué residuos muestran evidencias de haber sido sometidos a selección natural.

Además, puede mostrar el ritmo y la direccionalidad de las mutaciones. Estos análisis comparativos se basan en la creación de un árbol filogenético (Liu et al., 2009), que es una representación que describe las relaciones ancestro-descendiente entre organismos o secuencias de genes/proteínas. Las secuencias son las hojas del árbol, mientras que las ramas del árbol conectan las hojas de sus secuencias ancestrales o raíces.

La Figura 1A muestra un árbol sin raíz (*unrooted*), mientras que la Figura 1B muestra un árbol en formato rectangular (*rooted*) con distintos nodos y unidades taxonómicas operacionales (OTUs).

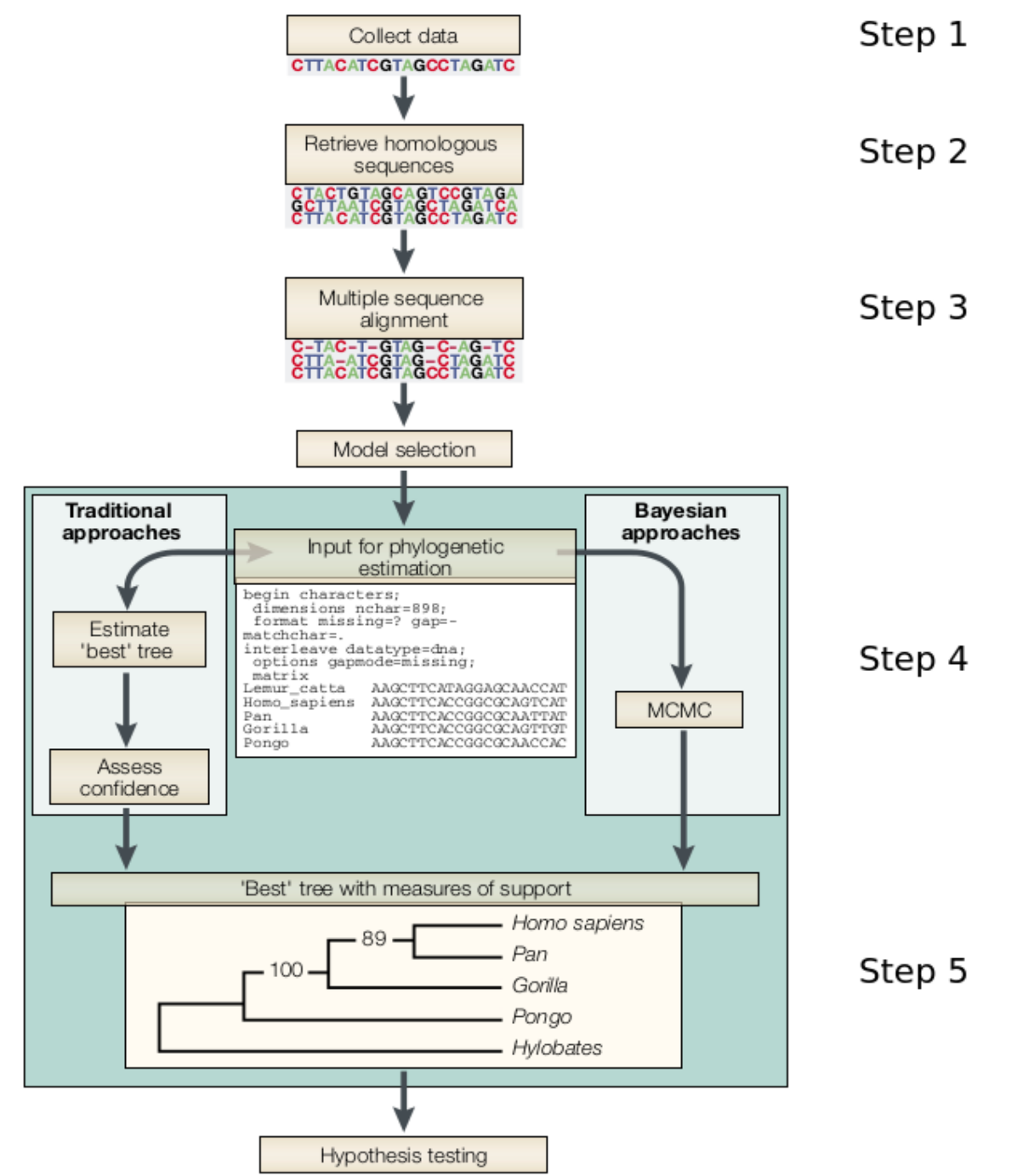


**Figura 1. Ejemplos de árboles filogenéticos.** A) Árbol sin raíz (*unrooted*). B) Árbol rectangular (*rooted*)

Inicialmente, el objetivo de los árboles filogenéticos fue estimar las relaciones entre especies en base a sus secuencias, pero hoy en día se ha ampliado su utilidad, estudiando la relación existente entre las secuencias sin considerar a la especie huésped, infiriendo la función de genes/proteínas que todavía no han sido estudiados experimentalmente (Whelan, 2008). Generar un árbol filogenético requiere de varios pasos. Para empezar, hay que localizar la información de los genes/proteínas homólogas, explorando diferentes bases de datos, ya que no todos los datos están aglutinados en el mismo sistema. Al obtener información de diferentes bases de datos (UniProt, NCBI), debido a que cada una va en un formato, el proceso se complica, pues se obtiene una información muy heterogénea que habrá que modificar manualmente mediante el uso de procesadores de texto para hacerla más uniforme. Este proceso es tedioso y sin valor añadido para un investigador.

Una vez se tiene toda esta información homogeneizada, llega una fase más compleja, en la que hay que filtrar la gran cantidad de información recibida, para ir reduciendo el número de proteínas/genes con el que se está trabajando, a un número manejable no solo por el investigador sino por los programas de las etapas posteriores. Un primer filtro a aplicar es comprobar que ninguna secuencia esté repetida, pero después se aplican otros filtros que pueden ir desde centrarse en un tipo de secuencias muy concretas, para subdividir los datos en varias agrupaciones, o por el contrario, se quiera tener una visión más global, y sólo centrarse en secuencias que sean sustancialmente diferentes entre sí (por ejemplo una de cada orden taxonómico).

El tercer paso es adaptar la información para el programa de alineamiento de secuencias, que requiere un formato concreto de datos de entrada (Hall, 2013). Tras este, hay que estimar un árbol a partir de las secuencias alineadas, y finalmente representar este árbol de tal manera que sea capaz de transmitir claramente la información relevante en él contenida. Estos pasos son mostrados esquemáticamente en la Figura 2. Además, este proceso no se realiza una sola vez, sino que, se repite en múltiples ocasiones, sobre todo el paso 4, hasta encontrar un árbol consenso que englobe no solo las secuencias iniciales sino otras provenientes de artículos científicos, o de web de proyectos de secuenciación de determinados organismos. Esto supone en muchos casos rehacer gran parte del proceso o incluso el proceso entero (Liu et al., 2009).



**Figura 2.** Esquema de creación de un árbol filogenético (Adaptado de Holder and Lewis, 2003)

Respecto a esto, hay que tener en cuenta el tiempo de computación necesario para la generación

del árbol. La Tabla 1 muestra una estimación de esos tiempos dependiendo del método usado en el paso 4.

**Tabla 1.** Estimación de los tiempos (Hall, 2011)

Datos	Unión de Vecinos	Máxima parsimonia	Máxima probabilidad	Inferencia bayesiana
<b>Pequeños</b>	1 s.	3 s.	6 s.	-
<b>Pequeños**</b>	9 s.	10 min.	1 h. 34 min.	29 min. 40 s.
<b>Grandes</b>	1 s.	22 s.	3 min. 29 s.	-
<b>Grandes**</b>	86 s.	10 h. 2 min.	58 h.	6 h. 33 min.

\* MacPro procesador dual (2.6 GHz); UV y MPar usando MEGA 4.0, MPro usando PHYML, e IB usando MrBayes

\* Pequeños: Menores a 100 secuencias; Grandes: A partir de 100 secuencias.

\*\* con estimación de fiabilidad

Como se ve, el proceso es bastante rápido cuando se trata de pocos datos, como es lógico, pero se dispara a varias horas y hasta días según el tipo de algoritmo y computador usado. Por lo tanto, sería muy interesante reducir el número de veces a generar este árbol, disponiendo de un mecanismo que te permita, previamente, analizar y filtrar bien toda la información, para quedarse con los datos más significativos.

El objetivo de este proyecto es automatizar y armonizar las tareas mecánicas de búsqueda y filtrado de la información asociada a secuencias de proteínas presentes en distintas bases de datos, para que sirva como soporte a la generación de árboles filogenéticos que permitan avanzar de forma más rápida las investigaciones en el campo de la bioquímica y de la biotecnología.

Así, se trata de disponer de una herramienta que permita, además de realizar una búsqueda de todas las proteínas deseadas en una única ejecución, a partir de varios formatos de datos (un árbol en formato Newick con códigos de distintas base de datos o bien un fichero FASTA), poder personalizar la información y exportarla también en varios formatos de salida, compatibles con distintas webs o con programas ya existentes que generen y/o muestren gráficamente árboles filogenéticos.

Así, la ardua **labor de semanas y meses** de buscar la información en las diferentes fuentes, integrarla y poder adaptarla a diferentes salidas se reduce **a unas horas**, el tiempo de realizar la conexión inicial a las distintas bases de datos para obtener la información, que se realiza en pocos minutos, y varias horas en las que el investigador decidirá qué datos desea generar. Una vez se tiene esta información, ya no es necesario volver a conectarse con la fuente de datos, que es lo que más tiempo penaliza, y se podrá trabajar y aplicar diferentes filtrados a la información. El programa permite guardar cualquier cambio realizado, para tener siempre un respaldo o *backup* en cualquier fase de la investigación, evitando tener que rehacerlo de nuevo si se ha tomado un camino equivocado. Además, si el usuario exporta sus datos en un formato concreto, y finalmente se da cuenta de que le ha faltado algo, o tiene algún campo que desea añadir nuevo, podrá volver a exportar los datos, con los cambios que desee, en un tiempo mínimo.

Por tanto, el objetivo del proyecto no es suplir sistemas ya existentes y consolidados para, por ejemplo, representar gráficamente un árbol, o crear árboles filogenéticos en base a una serie de secuencias, sino ofrecer un apoyo a la obtención, integración y gestión de la información que permita generar ficheros que puedan ser usados en programas ya existentes, acortando así el tiempo de publicación de los resultados derivados de los estudios bioinformáticos asociados a una proteína o enzima concreta.

### 3. Materiales y métodos

#### 3.1. Obtención de secuencias de proteínas

Las proteínas en las distintas bases de datos tienen una serie de códigos de identificación o identificadores, que son distintos según sean de UniProt (código corto y largo), NCBI (genID) o bien Ensembl. Estos identificadores forman parte de los ficheros FASTA y de árboles Newick.

Un fichero FASTA es un fichero de texto que consta de una cabecera y de un cuerpo, donde aparece la secuencia de una proteína/gen (Pearson and Lipman, 1988; Pearson, 2014). La cabecera puede ser variable, pero siempre debe contener el identificador de la proteína. Se hará uso de este identificador para después, por medio de las bases de datos biológicas, obtener más información sobre cada proteína. El formato se puede consultar en UniProt (<http://www.uniprot.org/help/fast-header>).

Los árboles en formato Newick son ficheros de texto que contienen la estructura de un árbol mostrando el código de la proteína/gen y un número que corresponde a la distancia evolutiva, todo ello separado por comas y paréntesis (Olsen G., 1990; Cardona et al., 2008). Por medio de estos ficheros se puede obtener una representación gráfica de la filogenia de las secuencias contenidas en él.

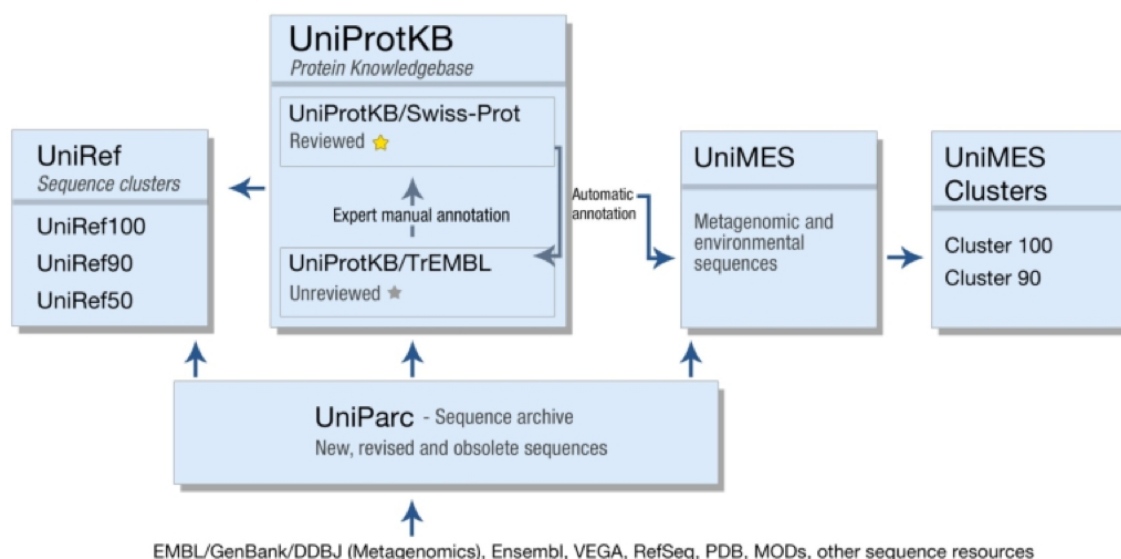
La técnica empleada para obtener la información será la de explorar el fichero FASTA o árbol Newick, en busca de la información relevante (identificadores de cada proteína en todos los ficheros, además de distancias en el caso de que el fichero sea un árbol Newick). Por medio de esta técnica, se dispondrá de un listado de todas las secuencias a analizar, y se desarrollará el proceso para obtener el resto de información de las proteínas.

#### 3.2. Bases de datos usadas

La información para cada proteína se ha obtenido de diferentes bases de datos de proteínas (UniProt, NCBI, Ensembl, PFAM), cada una especializada en una información concreta de las proteínas.

##### 3.2.1. UniProt

UniProt es un consorcio web que intenta aglutinar y anotar toda la información disponible sobre proteínas y sus secuencias. Está compuesta por varias bases de datos (UniRef, UniProtKB, UniParc y UniMES) (Figura 3) (UniProt Consortium, 2008).



**Figura 3.** Esquema del sistema UniProt y sus bases de datos (<http://www.uniprot.org/help/about>)

Esta base de datos puede considerarse la principal fuente de información del programa desarrollado, ya que es la más completa. A través de UniProt, es posible obtener otra serie de identificadores, como el código NCBI, que permitirá acceder a la base de datos del NCBI para descargar información sobre la taxonomía de la secuencia a analizar, y otros códigos como el NCBI NP, el NCBI GI, el Ensembl ID, el Gene ID y el Ensembl Gene.

Para el proyecto realizado, han sido de especial utilidad las bases de datos UniProtKB (<http://www.uniprot.org/help/uniprotkb>), sobre proteínas conocidas y estudiadas, y UniParc (<http://www.uniprot.org/help/uniparc>), de donde se obtienen proteínas UPI que están en vías de estudio y de las que aún no se dispone de gran información.

### 3.2.2. NCBI

El Centro Nacional para la Información Biotecnológica o National Center for Biotechnology Information (NCBI) es una potente base de datos con información de diverso tipo (NCBI Resource Coordinators, 2014). En este proyecto, se ha accedido a esta web para obtener la taxonomía de los organismos (<http://www.ncbi.nlm.nih.gov/taxonomy>), a partir de su apartado sobre taxonomías, y el identificador NCBI NP (<http://www.ncbi.nlm.nih.gov/guide/proteins/>), a través del cual conectar de nuevo a UniProt para obtener el resto de información. Para ello, se ha consultado el apartado sobre proteínas.

### 3.2.3. Ensembl

El proyecto trabaja también con identificadores Ensembl, un proyecto de investigación que desarrolla un sistema software para producir y mantener anotaciones automáticas en los genomas eucariotas seleccionados, integrar esta anotación con otros datos biológicos disponibles y disponer esta información de forma pública en la web (McLaren et al., 2010).

### 3.2.4. PFAM

La base de datos PFAM que describe las regiones funcionales de las proteínas, denominadas dominios, está conectada con UniProt, y por ello es posible acceder a ella por medio de los propios códigos UniProt, y es utilizada para obtener los dominios de cada proteína, así como el tamaño de su secuencia (Finn et al., 2014).

## 3.3. Homogeneización de la información

A partir de esta información de cada proteína en las distintas bases de datos, se obtendrá una tabla con los siguientes datos:

**Tabla 2.** Estructura de la información obtenida por el programa

<b>ID Original</b>	ID que tiene la proteína/gen en el fichero de entrada (FASTA o Newick), que podrá ser un identificador usado en cualquiera de las bases de datos del apartado 3.2.
<b>ID Principal</b>	ID que podrá generar el usuario para homogeneizar todos los datos, y que aunque en la entrada tengan diferente formato, al final todos se identifiquen de forma armonizada.
<b>UniProt Corto</b>	Identificador corto de 6 caracteres que usa la base de datos UniProt para almacenar las secuencias.
<b>UniProt Completo</b>	Identificador completo de la base de datos UniProt, que son 6 letras seguidas del código que define género y especie.
<b>NCBI NP</b>	Identificador de la base de datos NCBI de la proteína.
<b>NCBI GI</b>	Identificador GI (genID) de la base de datos NCBI.
<b>Ensembl ID</b>	Identificador en la base de datos Ensembl para las proteínas.
<b>NCBI ID Organismo</b>	Identificador taxonómico del organismo en la base de datos NCBI, que será usado para obtener la taxonomía del organismo que contiene la secuencia proteica.



<b>GeneID</b>	Identificador del Gen.	
<b>Ensembl Gene</b>	Identificador de la base de datos Ensembl para el gen.	
<b>Name</b>	Nombre de la proteína en UniProt.	
<b>OS</b>	Nombre del organismo en UniProt.	
<b>GN</b>	Nombre del gen en UniProt.	
<b>Observaciones</b>	Campo a rellenar por el usuario, que será rellenado automáticamente cuando alguna secuencia del fichero de entrada no cumpla ningún formato de los identificadores reconocidos.	
<b>PE</b>	Código de UniProt que muestra un nivel de evidencia que ofrece la existencia de la proteína, de 1 a 5. <ul style="list-style-type: none"> <li>• 1. Evidencia a nivel de proteína.</li> <li>• 2. Evidencia a nivel de transcripción.</li> <li>• 3. Inferida desde homología.</li> <li>• 4. Predicha.</li> <li>• 5. Incierto.</li> </ul>	
<b>SV</b>	Código de UniProt que refleja la versión de la secuencia.	
<b>MW (Da)</b>	Masa molecular de la proteína, medida en Dalton.	
<b>Length</b>	Tamaño de la secuencia.	
<b>Sequence</b>	Secuencia de la proteína	
<b>Campos del dominio</b>	Grupos de 4 campos que reflejan los dominios PFAM de la proteína, ofreciendo por cada dominio estos 4 campos comunes:	
	PFAM Domain	Nombre del dominio.
	Accession	Identificador de la familia del dominio.
	Start	Posición del aminoácido donde empieza el dominio.
	End	Posición del aminoácido final del dominio.
<b>Campos de la taxonomía del organismo</b>	Grupo variable de campos, en función de las secuencias de la ejecución, que muestran el rango y el nombre científico de cada elemento taxonómico del organismo. Un ejemplo podría ser el siguiente:	
	Superkingdom	Bacteria
	Phylum	Firmicutes
	Class	Bacilli
	Order	Bacillales
	Family	Bacillaceae
	Genus	Bacillus
	Species	Bacillus coagulans
<b>% Similitud</b>	Porcentaje de similitud entre cada secuencia y la tomada como patrón para esa familia o bien para ese género y especie.	
<b>Patrón Similitud</b>	Campo que muestra qué secuencia ha sido tomada como patrón para cada categoría de alineamiento creada.	
<b>ID Similitud</b>	ID de la secuencia que es patrón en ese género y especie o familia, contra la que se ha comparado la secuencia para realizar la similitud.	

La tabla 3 muestra un resumen de las bases de datos accedidas, y qué información se ha obtenido de cada una de ellas.

**Tabla 3.** Bases de datos accedidas e información obtenida

Base de datos origen	Dato obtenido
UniProt	UniProt corto UniProt completo NCBI GI Ensembl ID GeneID Ensembl Gene Name OS GN PE SV MW (Da) Sequence
PFAM	Length Dominios
NCBI	NCBI NP Taxonomía del organismo

### 3.4. Integración de la información

Como se ha comentado anteriormente, la información vendrá dada de diversas fuentes, y cada uno podrá tener un identificador diferente para la misma proteína. Por este motivo, la primera tarea de la aplicación es la de identificar qué clase de identificador se tiene en la secuencia a analizar, y buscar cómo está representada esta misma secuencia en otras bases de datos (UniProt corto, UniProt largo, NCBI NP, NCBI GI y Ensembl).

Esta equivalencia se obtiene a partir de la base de datos UniProtKB, que contiene, para cada secuencia con código UniProt, los distintos códigos en el resto de bases de datos. Existe una excepción, el código NCBI GI, por el que no se puede buscar en UniProt. En este caso, la información se obtiene a través del código NCBI NP, que podemos obtener, a partir del mencionado GI, en la base de datos del NCBI. Una vez se dispone de ese código NCBI NP, ya sí que UniProt es capaz de ofrecer la equivalencia con el resto de identificadores.

Esto es posible porque, aunque UniProt identifique todas las secuencias con su código corto, nos permite realizar búsquedas por el resto de códigos, tomando como entrada cualquier tipo de código mencionado (salvo el GI), y mostrándonos qué proteína es, en código UniProt, la deseada. Así, por medio de este código, será posible acceder a su página en UniProt, y obtener el resto de códigos y de información necesaria. Es posible que, al partir de distintos identificadores, se obtenga la misma secuencia varias veces. Para poder eliminar estas redundancias, en el apartado 3.8. se detallan los tipos de filtros disponibles, entre ellos uno para evitar estas posibles redundancias generadas.

### 3.5. Filtrado de la información biológica

Una de las principales problemas de los investigadores es el de filtrar toda la información obtenida. Dentro de los diferentes filtros que se pueden aplicar, para que el usuario pueda ir reduciendo la gran cantidad de secuencias a un número adecuado para poder trabajar con ellas, nos vamos a centrar en los siguientes:

- **Filtrar por duplicados:** Comprueba que no haya dos secuencias con el mismo ID Principal.
- **Filtrar por PE:** Permite establecer un valor de PE para eliminar las proteínas que tengan dicho valor.
- **Filtrar por tamaño de secuencia:** Permite al usuario elegir un rango de tamaño de secuencia y eliminar las secuencias que no cumplan dicho tamaño.

- **Filtrar por dominios:** Muestra el conjunto de dominios existentes y qué cantidad de secuencias están en cada uno de esos grupos, permitiendo eliminar los grupos que se desee (por ejemplo, si hay alguna secuencia aislada que tiene unos dominios diferentes, sería interesante por un lado identificarlo, y ya después estudiar si interesa quitarlo porque no es nada semejante al resto o al contrario, marcarlo como algo importante a tener en cuenta por ser “único”).
- **Filtrar por similitud (máxima o mínima):** Indicando un valor mínimo o máximo de similitud, se pueden eliminar las secuencias que no cumplan la restricción. Esto es útil para cuando se quieran ir eliminando secuencias similares. Esto se usa para quedarse únicamente con una secuencia de cada familia.

Para realizar el filtrado por similitud, se realiza un alineamiento de secuencias, para permitir quedarse con más de una secuencia por familia, si es bastante diferente al resto. Mediante un alineamiento de secuencias se pueden comparar diferentes secuencias, para buscar similitudes entre ellas y tratar de averiguar si existe alguna relación funcional o evolutiva entre ellas. Existen los alineamientos locales, que buscan similitudes en partes de la secuencia, y los globales, que comparan toda la secuencia, aunque haya zonas que no sean similares entre una secuencia y otra.

En este trabajo se han usado alineamientos globales, ya que el objetivo es comparar siempre la secuencia completa de cada proteína/gen para tener un porcentaje de similitud respecto a toda la secuencia, y no únicamente respecto a una zona concreta que haría que el porcentaje fuera más alto. El algoritmo usado para realizar los alineamientos es el descrito por Needleman-Wunsch (Needleman and Wunsch, 1970), ya que ofrece el mejor alineamiento independientemente del tamaño y la complejidad de las secuencias. La matriz de sustitución usada para realizar el alineamiento ha sido la BLOSUM62, por ser el estándar en este tipo de programas.

### **3.6. Herramientas de programación usadas**

Este trabajo ha sido realizado mediante una combinación del lenguaje de programación Java y una serie de librerías para la obtención, manipulación y presentación de los datos.

Para realizar la interfaz gráfica (ventanas con botones, tablas, etcétera), se ha usado la biblioteca de Swing (<http://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>) de Java, incluida en el paquete Javax.

Por otro lado, se ha hecho uso del Document Object Model (DOM), el paquete org.w3c.dom (<http://docs.oracle.com/javase/7/docs/api/org/w3c/dom/package-summary.html>), incluido en la API de Java, para el procesamiento XML. Este paquete ha sido útil para la lectura de datos en algunas bases de datos, de donde se obtenía la información en formato XML por medio de una petición a la base de datos para obtener la información en este formato. Se ha hecho uso también del paquete org.xml.sax, para la lectura de algunos datos.

Por último, para realizar el alineamiento entre secuencias, se ha partido de la librería JAligner (<https://github.com/ahmedmoustafa/JAligner>) (Gotoh, 1982), que ha sido adaptada a los requisitos requeridos por el programa diseñado.

## 4. Resultados

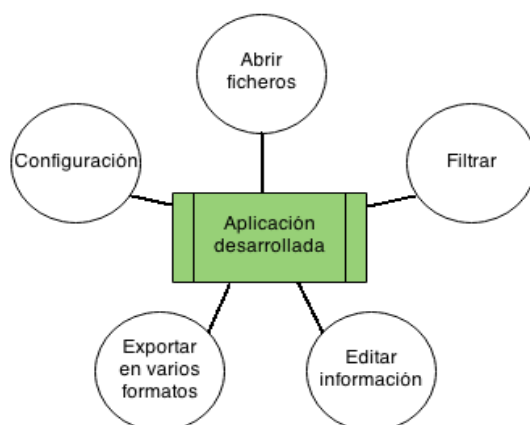
### 4.1. Diseño de la solución

Para que el programa diseñado sea sencillo para el usuario, el programa tomará como entrada una serie de códigos, que podrán ser UniProt, Ensembl o NCBI GI, y se encargará de explorar las distintas bases de datos para encontrar la equivalencia de cada código con el resto de códigos de otras bases de datos, y a partir de ellos ir buscando la información para rellenar los datos asociados a cada secuencia.

El diseño en Java se ha dividido en dos paquetes. En un paquete tenemos una serie de clases que almacenarán los datos obtenidos de las diversas bases de datos, que se complementan con otras clases que definen métodos de acceso, creación y manipulación de esos datos, mientras que en el otro paquete se disponen las clases encargadas de la representación visual de la información, por medio de la creación de ventanas. Por último, un tercer paquete contiene la librería externa JAligner que realiza el alineamiento, usado en el filtrado por similitud.

### 4.2. Descripción de la aplicación

El sistema cuenta con una serie de módulos, que permiten realizar toda la funcionalidad (obtener datos, trabajar con ellos, y exportarlos en el formato deseado, con la conectividad mostrada en la Figura 4).



**Figura 4.** Módulos desarrollados en la aplicación.

#### 4.2.1. Entradas y salidas

La tabla 4 muestra los tipos de entrada y salida compatibles con la aplicación.

**Tabla 4.** Tipos de entrada y salida disponibles.

Formatos de entrada soportados	Formatos de salida disponibles
Árbol Newick	Árbol Newick
Fichero FASTA	Fichero FASTA
Estado generado por el programa	Estado generado por el programa
	Árbol para iTOL
	Texto plano para Excel

## Entrada

La entrada a la aplicación se podrá realizar por medio de árboles Newick (Cardona et al., 2008), ficheros FASTA (Pearson, 2014) o archivos de estado (tipo de fichero que genera el propio programa para poder almacenar la información en algún estado intermedio y poder seguir trabajando con ella en el futuro, de forma *offline*, sin tener que volver a acceder a las bases de datos ni realizar los filtrados que ya se habían realizado).

La lectura de los ficheros tiene dos partes. Por un lado, el sistema explora el archivo FASTA o árbol Newick, para obtener los identificadores existentes en ellos. Los ficheros FASTA disponibles tienen siempre el formato FASTA de UniProt, así que se realiza una lectura automática, pero los árboles pueden contener el ID aislado, o acompañado por más información, por lo que es necesario, antes de realizar la lectura, pedir al usuario que identifique un ID de ejemplo para que el programa sea capaz de buscarlo. Así, lo primero que hace la aplicación cuando el usuario desea abrir un nuevo fichero árbol Newick, es mostrarle la información de la primera secuencia, y le pide que inserte cuál es el ID de ese elemento. Gracias a esto, el sistema aprende el formato de este fichero árbol concreto, y cómo almacena los IDs, siendo capaz de obtener el resto de los identificadores de ese fichero.

Una vez se dispone de un listado de los identificadores a analizar, el programa va rellenando la información asociada en las distintas bases de datos para cada uno de ellos. Esto requiere por parte del programa una comprobación de qué tipo de código es, para posteriormente conectarse a las diferentes bases de datos en función de esto, para obtener la equivalencia con el resto de códigos.

Cuando el programa obtiene todos estos códigos, se dedica a explorar las diferentes bases de datos para obtener el resto de información, y es entonces cuando le muestra al usuario la información. Llegado a este punto, el usuario podrá trabajar con los diferentes identificadores, pero se le da la opción de que genere un ID Principal, para homogeneizar los datos y que, independientemente de qué tipo de identificador hayan tenido en el fichero de entrada, tener ahora un identificador común (código UniProt corto por ejemplo), y que a la hora de exportar la información todos salgan con este identificador. A la hora de generar este ID Principal, el programa permite que el usuario marque los diferentes identificadores con una prioridad para que, en el caso de que el identificador elegido no se haya encontrado en alguna secuencia, esta sea denominada por otro identificador, de tal forma que no puedan existir secuencias sin ningún identificador. La integración entre los distintos tipos de identificadores se llevan a cabo a partir de la base de datos UniProt (UniProt Consortium, 2008), como se ha descrito en el apartado 3.4. Integración de la información. Todo esto es un proceso interno, queda el paso final de mostrar toda la información al usuario. Para ello, se hace uso de la librería Swing, y se pinta toda la información en una tabla, mostrando una secuencia por fila y su información asociada en distintas columnas, que se crean dinámicamente según, por ejemplo, el número de dominios PFAM (Finn et al., 2014) que contenga, que harán que se dispongan más o menos columnas con esta información. Hay un ejemplo de esta información integrada en la Figura 9.

## Salida

Las opciones de salida o de exportación del programa son diversas, pudiendo volver a generar un árbol Newick a partir de las distancias que tomó como entrada pero personalizando ahora la etiqueta que identifica a cada secuencia, pudiendo ser desde algún código hasta un listado de todos los dominios PFAM de cada secuencia. Los diferentes campos de la etiqueta podrán ser separados por el carácter que el usuario elija, lo que permite poder poner un espacio, un guión bajo (\_), o cualquier otro carácter, permitiendo flexibilidad para que la aplicación sea compatible con un gran número de programas existentes para la visualización o manipulación de árboles.

El estado actual de la tabla generada podrá ser guardado en un fichero de estado, y posteriormente abrirlo y tener la misma información que se había guardado y, de forma inmediata, sin tener que acceder a la red para descargarse de nuevo la información. Esta opción se lanza automáticamente por el programa antes de realizar cualquier filtro que elimine secuencias, creando de forma automática un fichero de estado que es nombrado como el fichero importado añadiendo un par de campos variables. Uno con cuándo se ha generado ese fichero (antes de filtrar por dominios, o de filtrar por tamaño de secuencia, por ejemplo), y otro con la fecha y hora de cuándo se generó, para que el usuario siempre sepa qué fichero es cada uno y cuándo se creó.

Otro tipo de fichero posible de generar es uno de formato FASTA, que permitirá al usuario elegir también los campos que desea ubicar en la primera línea de cabecera. Además, permite también elegir un trozo de la secuencia (dominio) para tomar como inicio, y otra para tomar como final, por si el usuario quiere exportar únicamente las proteínas que tengan unos u otros dominios.

Una opción muy interesante es la de exportar el árbol Newick pero acompañado de un fichero de configuración para la herramienta iTOL (Letunic and Bork, 2011), un visualizador de árboles que es capaz de generarte, además del árbol, una representación gráfica de los dominios de cada proteína. Mediante esta opción, se generarán 3 ficheros. El primero será el fichero árbol Newick habitual, que irá acompañado de un segundo fichero que, a partir de las características que el usuario establezca, indicará a la plataforma iTOL, para cada secuencia, los dominios que tiene, y una forma y un color para representar cada dominio (elegidos por el usuario), además de una etiqueta que podrá ser el nombre del dominio o lo que el usuario indique.

Además de estos dos ficheros, que son requeridos por la herramienta iTOL para dibujar el árbol con los dominios de cada elemento, se generará un tercer fichero, de información para el usuario, que indicará, para cada dominio, qué forma, color y nombre indicó, ya que sino, en el futuro, no habrá forma de saber la equivalencia entre forma, color y dominio, ya que la plataforma iTOL no indica, a veces, leyenda con las opciones elegidas.

Por último, se ofrece también una opción de exportación de los datos presentados en la tabla del programa a un fichero de texto plano, separando los campos por \$, para que se puedan importar en algún programa tipo Excel. Algo que quizás sea más cómodo para el usuario y con lo que está más acostumbrado a trabajar.

## **4.2.2. Editar**

### **ID Principal**

El sistema dispone de un ID Principal, elegido por el usuario, para homogeneizar todas las secuencias. A la hora de generar este ID, la aplicación pide al usuario qué tipo de ID desea elegir como principal, por medio de una prioridad entre todos los tipos existentes (UniProt Corto, UniProt Largo, NCBI NP, NCBI GI y Ensembl ID). De esta manera, si el usuario elige un ID NCBI GI pero alguna secuencia no tiene este ID, se mostrará como Principal la segunda elección del usuario (y así sucesivamente), asegurando que todas las secuencias tengan algún identificador.

La opción que permite generar este ID se encarga, tras obtener la prioridad de identificadores deseada por el usuario, de recorrer todas las secuencias e ir estableciendo como ID Principal el primero de la lista. Si no existe, va descendiendo en prioridad hasta encontrar uno que esté disponible para la proteína en cuestión.

### **Agregar datos**

Otra opción de entrada disponible en el programa es la de leer datos de un fichero FASTA o Newick pero no desde cero, sino completar los datos ya existentes con un nuevo fichero que contenga algunos datos nuevos. Por medio de esta opción, el programa hace una lectura normal del fichero, pero antes de cargar la información comprueba si ese identificador ya existe en el sistema, para evitar duplicados y no perder tiempo en recoger esa información. Así, únicamente inserta las secuencias que no existan. Además, es posible también añadir una nueva fila de forma manual, pidiendo al usuario un identificador e insertando una fila en blanco, que el usuario ya irá rellenando con los campos que desee, o bien automáticamente con la opción siguiente de actualizar datos.

### **Actualizar datos**

Por último, una opción interesante es la de actualizar la tabla, y que el programa complete la información de algunas secuencias que hemos indicado manualmente el identificador pero nos faltan aún campos, o sencillamente volver a conectar con la base de datos por si algo ha cambiado. Mediante esta opción, el sistema volverá a hacer una búsqueda por las diferentes bases de datos, obteniendo de nuevo todos los datos para cada proteína.

El usuario podrá también editar manualmente cualquier campo de la tabla, salvo el ID Principal (que

deberá ser generado automáticamente), el ID de entrada (que lógicamente viene fijado por qué ha leído el sistema del fichero de entrada), y datos relativos al alineamiento, como el porcentaje obtenido o el ID del patrón (que no tiene sentido modificarlos porque entonces se estarían falseando los datos).

### 4.2.3. Filtrados

#### Duplicados

El sistema es capaz de realizar un filtro por duplicados, eliminando registros que tengan el mismo identificador principal. Para ello, es necesario que el usuario genere este ID Principal, y es entonces cuando el programa podrá ir comprobando que no haya ninguno repetido.

#### Tamaño de secuencia

El tamaño de la secuencia es un filtro básico y sencillo, pero muy útil, ya que permite descartar fácilmente secuencias muy grandes o muy pequeñas (que posiblemente serán fragmentos, y no secuencias totales). El sistema pedirá al usuario un valor mínimo y máximo de secuencia, para después ir comprobando los elementos que se encuentran en ese valor.

#### PE

El filtro por PE simplemente recorre las secuencias comprobando este valor para eliminar las que cumplan el valor marcado por el usuario, que el sistema le habrá requerido previamente por medio de una ventana emergente.

#### Dominios

Un filtro interesante es el filtro por dominios. En este, el sistema realiza un recorrido por todas las secuencias, almacenando qué dominios tiene cada una, para después juntar todos los elementos que tengan los mismos dominios, y le muestra esta información al usuario. Así, este puede ver cómo están repartidas las secuencias en función de sus dominios, y realizar un filtrado basado solamente en las secuencias que contengan unos dominios u otros, o destacar los grupos de dominios que tienen muy pocas secuencias en ellos, para ver qué particularidades tienen esas proteínas.

#### Similitud

El filtrado por similitud es un filtrado más complejo, ya que requiere una gestión de todos los elementos y su pertenencia a un grupo u otro. Se puede realizar de dos formas, haciendo grupos por secuencias que pertenezcan a la misma familia o por secuencias que compartan género y especie. Esta decisión se le pide al usuario al iniciar la carga de datos, y después se puede volver a modificar por medio de las opciones del programa.














Para realizar este proceso de filtrado, el sistema recorre todas las secuencias, comprobando su familia o género y especie. Cada vez que encuentra una secuencia con una familia (o género y especie) nueva, la marca como patrón, y establece que tiene un 100% de similitud consigo mismo. Cuando encuentra una secuencia que ya tiene un patrón definido para esa familia, o género y especie, se encarga de realizar un alineamiento contra dicho patrón, usando un alineamiento mediante la librería JAligner (<https://github.com/ahmedmoustafa/JAligner>) (Gotoh, 1982), una implementación en código abierto del algoritmo de Needleman-Wunsch (Needleman and Wunsch, 1970) y Smith-Waterman para alineamiento de secuencias. En el sistema, se ha adaptado esta librería para, usando el algoritmo de Needleman-Wunsch (por realizar alineamientos globales como se deseaba), adaptarlo al resto del programa y coordinar el funcionamiento del sistema con el cálculo de similitud.

El programa permitirá, una vez obtenidas las similitudes entre cada elemento con su patrón (que se puede modificar por el usuario para que éste elija el patrón deseado), marcar un número mínimo o máximo de similitud, y que las secuencias que no lo cumplan sean propuestos para su eliminación. Estas secuencias no serán eliminadas automáticamente, sino que el usuario podrá revisar esta selección, por si quiere cambiar el criterio o, individualmente, eliminar o dejar en el sistema alguna secuencia concreta que tiene unas características interesantes para su estudio. Conviene destacar que la aplicación, antes de cualquier eliminación de secuencias, guarda automáticamente los datos

en un fichero propio de estado, por si el usuario no recordó hacerlo y finalmente se da cuenta de que ha realizado un filtro no deseado, perdiendo algún elemento importante para su investigación.

#### 4.2.4. Opciones de configuración

Cuando el usuario desea exportar a iTOL (Letunic and Bork, 2011), deberá elegir una figura y un color para representar cada dominio. Estas figuras están ya predefinidas en iTOL (Figura 5), y cada una debe ser codificada con un código específico de esta plataforma.

Code	Shape	Example
RE	rectangle	
HH	horizontal hexagon	
HV	vertical hexagon	
EL	ellipse	
DI	rhombus (diamond)	
TR	right pointing triangle	
TL	left pointing triangle	
PL	left pointing pentagram	
PR	right pointing pentagram	
PU	up pointing pentagram	
PD	down pointing pentagram	
OC	octagon	
GP	rectangle (gap)	

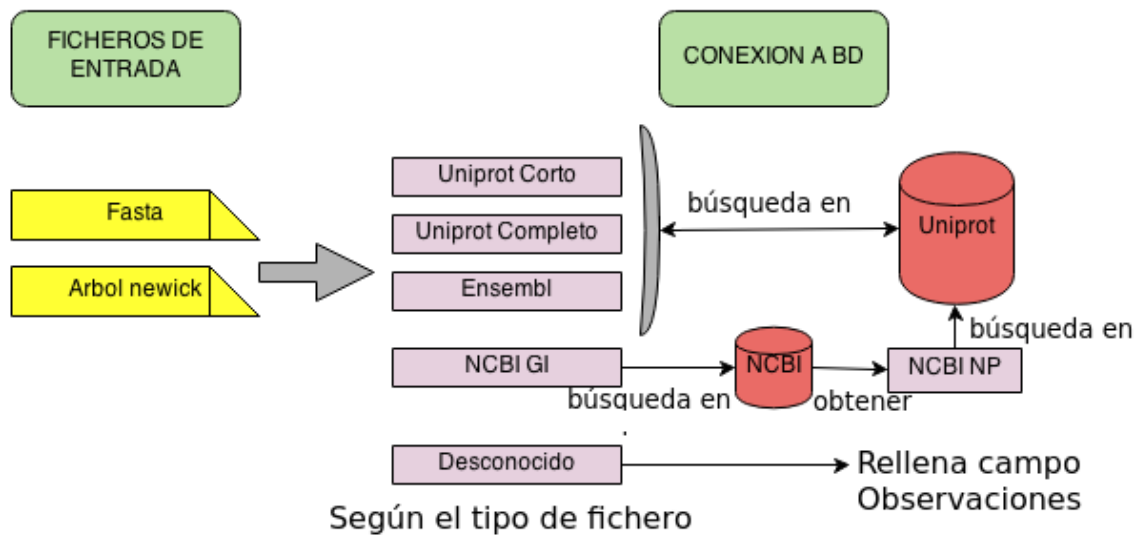
**Figura 5.** Códigos y formas usados por iTOL para representar dominios proteicos

El programa tiene por defecto estas figuras disponibles, con su correspondiente código, pero permite la opción de gestionar este listado y poder añadir nuevas, por si en el futuro la plataforma iTOL añade alguna; o modificarlas, por si cambia la codificación de alguna de ellas. Lo mismo sucede con los colores. Se han incluido una serie de colores básicos, pero el usuario puede agregar o suprimir cualquiera de ellos, simplemente indicando su código hexadecimal.

#### 4.3. Evolución del programa

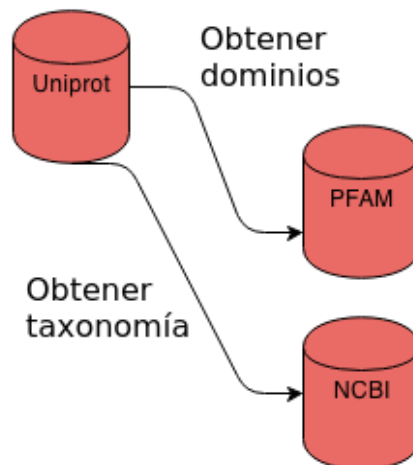
El primer paso del sistema es leer el fichero que indique el usuario, y tras obtener los códigos, identificar de qué tipo son para obtener su equivalencia con el resto de códigos en las diferentes bases de datos. En función de qué tipo de código se disponga, habrá que buscar la equivalencia con el resto de códigos en UniProt directamente, o pasar por NCBI para obtener el código NCBI NP, que ya sí que es admitido por UniProt.





**Figura 6.** Lectura e identificación de códigos por parte del programa desarrollado

Una vez se tienen todos los códigos, el sistema busca el resto de la información en las bases de datos de UniProt, PFAM y NCBI (Figura 7).



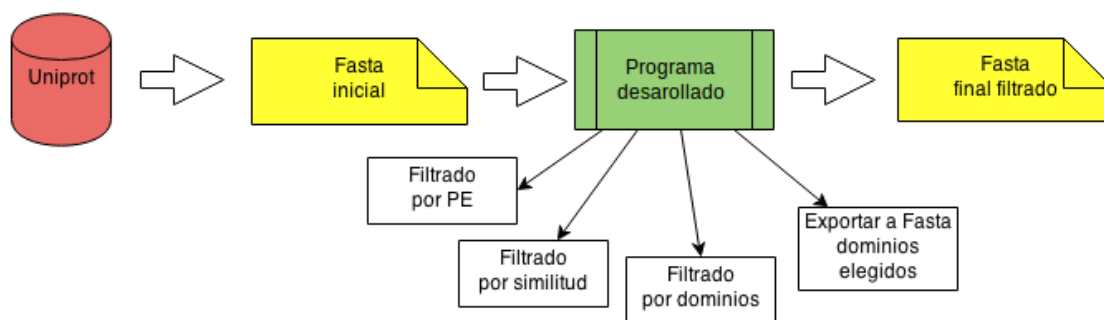
**Figura 7.** Flujo secundario de acceso a las bases de datos por parte del programa desarrollado

Los dominios se obtienen por medio del propio código UniProt, ya que PFAM funciona con estos identificadores, mientras que, para obtener la taxonomía desde el apartado Taxonomy de NCBI, es necesario obtener antes el código de la taxonomía del organismo a analizar, también disponible en UniProt.

## 4.4. Ejemplo de ejecución

### 4.4.1. A partir de un fichero FASTA

Un ejemplo de ejecución podría comenzar por un fichero FASTA (Pearson, 2014) que contenga cientos de secuencias, obtenido a través de la propia web de UniProt, y que nos interese filtrar para generar después otro FASTA pero ya con la información interesante para el estudio a desarrollar.



**Figura 8.** Ejemplo de ejecución con fichero FASTA

Se comenzaría a partir de un fichero FASTA inicial grande, que contiene más de 400 proteínas.

El fichero FASTA inicial (por ejemplo, de 400 secuencias), tendría que ser filtrado a mano por el investigador a partir de una combinación de Word y Excel. Gracias al programa desarrollado, todo esto se hará de forma automática y rápida, importándolo a la aplicación y visualizando, en pocos minutos, toda la información de cada secuencia, totalmente actualizada pues acaba de ser obtenida de las bases de datos.

ID Original	¿Eliminar	ID Principal	Uniprot Corto	Uniprot Completo	NCBI NP	NCBI GI	Ensembl
Q6P3L9	<input type="checkbox"/>	ENSDARP00000017123	Q6P3L9	Q6P3L9_DANRE	NP_955839.2	gi_41282194	ENSDARP000
Q47949	<input type="checkbox"/>	Q47949	Q47949	Q47949_9EURY			
Q977X9	<input type="checkbox"/>	Q977X9	Q977X9	Q977X9_9EURY			
Q9Y8I4	<input type="checkbox"/>	Q9Y8I4	Q9Y8I4	Q9Y8I4_PYRIS			
Q977U6	<input type="checkbox"/>	Q977U6	Q977U6	Q977U6_HALME			
Q9HKG4	<input type="checkbox"/>	gi_16081724	Q9HKG4	Q9HKG4_THEAC	NP_394107.1	gi_16081724	
Q9HK32	<input type="checkbox"/>	gi_16081843	Q9HK32	Q9HK32_THEAC	NP_394238.1	gi_16081843	
Q97Y81	<input type="checkbox"/>	gi_15898289	Q97Y81	Q97Y81_SULSO	NP_342894.1	gi_15898289	
Q97X22	<input type="checkbox"/>	gi_15898731	Q97X22	Q97X22_SULSO	NP_343336.1	gi_15898731	
Q97WS2	<input type="checkbox"/>	gi_15898835	Q97WS2	Q97WS2_SULSO	NP_343440.1	gi_15898835	
Q97AP0	<input type="checkbox"/>	gi_13541590	Q97AP0	Q97AP0_THEVO	NP_111278.1	gi_13541590	
Q97AN9	<input type="checkbox"/>	gi_13541591	Q97AN9	Q97AN9_THEVO	NP_111279.1	gi_13541591	
Q96YC6	<input type="checkbox"/>	gi_15922573	Q96YC6	Q96YC6_SULTO	NP_378242.1	gi_15922573	
Q8ZW33	<input type="checkbox"/>	gi_18313020	Q8ZW33	Q8ZW33_PYRAE	NP_559687.1	gi_18313020	

**Figura 9.** Información obtenida por el programa de las bases de datos

En la Figura 9 se muestran las primeras filas y columnas del FASTA abierto. Algunos identificadores no estaban disponibles para algunas secuencias, así que no se han podido completar esos campos. En este caso, el usuario debe generar un ID Principal, marcando como prioridad el Ensembl ID, después el NCBI GI y, si ninguno de los dos existiera, el UniProt Corto. Vemos como, en la primera secuencia, se ha establecido lo que el usuario quería, el identificador de Ensembl, pero en el resto se ha tenido que optar por el GI o el UniProt Corto, ya que no existían los demás (Figura 9).

El usuario puede entonces desplazarse por las columnas del programa, para ver el resto de la información de cada secuencia. Un primer filtro a aplicar puede ser el de PE, para lo que puede ordenar las secuencias por este valor y, tras ver qué cantidad hay de cada uno de ellos, realizar un filtrado y eliminar por ejemplo las que tengan un valor PE de 4 (Figuras 10 y 11).

NWK exchanger						
Archivo Editar Filtrar Configuración Ayuda						
	GN	Observaciones	PE ▾	SV	MW (Da	
	TC_0154		4	1	37253	
0824 / MB4)	GdhA4		4	1	39103	
	ldh		4	1	37081	
	DVU_0375		4	1	44125	
	GSTENG00009556001		4	1	67169	
	HCH_00319		4	1	37406	
			3	1	19211	
			3	1	46309	
	gdh-1		3	1	47959	
15155 / AMRC-C165)	Ta0635		3	1	47712	
15155 / AMRC-C165)	Ta0776		3	1	46033	
	gdhA-1		3	1	45510	
	gdhA-3		3	1	47351	
	gdhA-4		3	1	46063	

**Figura 10.** Secuencias ordenadas por PE

NWK exchanger						
Archivo Editar Filtrar Configuración Ayuda						
	GN	Observaciones	PE ▾	SV	MW (Da	
	TC_0154		4	1	37253	
0824 / MB4)	GdhA4		4	1	39103	
	ldh		4	1	37081	
			4	1	44125	
			4	1	67169	
			4	1	37406	
			3	1	19211	
			3	1	46309	
			3	1	47959	
15155 / AMRC-C165)			3	1	47712	
15155 / AMRC-C165)	Ta0776		3	1	46033	
	gdhA-1		3	1	45510	
	gdhA-3		3	1	47351	
	gdhA-4		3	1	46063	

**Eliminar PE** ✕

? Escriba el PE a eliminar

Aceptar
Cancelar

**Figura 11.** Secuencias eliminadas por PE igual a 4 (secuencia predicha y no comprobada)

Tras esto, el usuario observa que aún tiene excesivas secuencias, así que vuelve a realizar un filtro, pero por similitud, para eliminar secuencias que, siendo de la misma familia, tengan una similitud superior al 80%, para quedarse únicamente con una secuencia por familia salvo que haya varios en la misma familia con diferencias significativas (Figura 12).

**Similaridad máxima** ✕

? Introduzca su similitud máxima

Aceptar
Cancelar

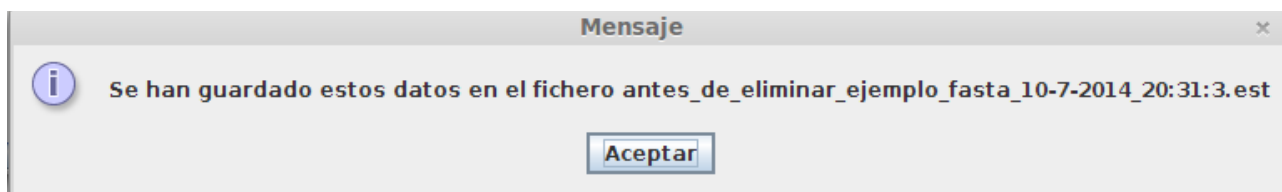
**Figura 12.** Filtrado de secuencias por similitud

El programa no eliminará las secuencias que cumplan la restricción, sino que las marcará para que el usuario valide esa selección. Si no está conforme con alguna eliminación, porque quizás sea alguna secuencia interesante o referente a nivel bibliográfico, puede desmarcarla para que no sea eliminada.

NWK exchanger								
Archivo Editar Filtrar Configuración Ayuda								
ID Original	¿Eliminar	ID Principal	Uniprot Corto	Uniprot Completo	NCBI NP	NCBI GI	Ensembl ID	NCBI ID C
Q3DQR9	<input type="checkbox"/>	Q3DQR9	Q3DQR9	Q3DQR9_STRAG				342613
Q3K0H7	<input type="checkbox"/>	gi_76786874	Q3K0H7	Q3K0H7_STRAL	YP_329976.1	gi_76786874		205921
Q9KG94	<input type="checkbox"/>	gi_15612781	Q9KG94	Q9KG94_BACHD	NP_241084.1	gi_15612781		272558
Q821T5	<input type="checkbox"/>	gi_29840607	Q821T5	Q821T5_CHLCV	NP_829713.1	gi_29840607		227941
Q7YZV0	<input type="checkbox"/>	Q7YZV0	Q7YZV0	Q7YZV0_SPIVO				58336
Q4KTJ5	<input type="checkbox"/>	gi_146322226	Q4KTJ5	Q4KTJ5_9LACT	YP_001174716.1	gi_146322226		1358
Q8XK85	<input checked="" type="checkbox"/>	gi_18310500	Q8XK85	Q8XK85_CLOPE	NP_562434.1	gi_18310500		195102
Q7M823	<input checked="" type="checkbox"/>	gi_34558218	Q7M823	Q7M823_WOLSU	NP_908033.1	gi_34558218		273121
Q5KYC0	<input checked="" type="checkbox"/>	gi_56420566	Q5KYC0	Q5KYC0_GEOKA	YP_147884.1	gi_56420566		235909
Q64Q81	<input checked="" type="checkbox"/>	gi_53714892	Q64Q81	Q64Q81_BACFR	YP_100884.1	gi_53714892		295405
Q4Q7X1	<input checked="" type="checkbox"/>	gi_157872056	Q4Q7X1	Q4Q7X1_LEIMA	XP_001684577.1	gi_157872056		5664
Q5L9X6	<input checked="" type="checkbox"/>	gi_60682871	Q5L9X6	Q5L9X6_BACFN	YP_213015.1	gi_60682871		272559
Q3IS94	<input checked="" type="checkbox"/>	gi_76801551	Q3IS94	Q3IS94_NATPD	YP_326559.1	gi_76801551		348780
Q30QE4	<input checked="" type="checkbox"/>	gi_78777707	Q30QE4	Q30QE4_SULDN	YP_394022.1	gi_78777707		326298

**Figura 13.** Secuencias marcadas para eliminar basadas en un patrón de similitud

Cuando se está seguro de las secuencias a eliminar, se ejecuta la función de eliminar seleccionadas. No obstante, el sistema siempre creará un archivo de estado automáticamente, por si el usuario ha borrado algo que no quería y se da cuenta tarde (Figura 14). Así, al igual que al filtrar previamente por PE se generaba un fichero de estado, ahora se generó otro nuevo, de tal forma que el usuario nunca pierda información si desea recuperarla.



**Figura 14.** Generación automática de fichero de estado tras eliminar los marcados por similitud

También se puede filtrar por dominios PFAM (Finn et al., 2014), y ver si hay muchas secuencias con el mismo patrón de dominios o no (Figura 15).

Selección de campos			
<input type="checkbox"/>	2	ELFV_dehydrog	ELFV_dehydrog_N Pfam-B_5374
<input type="checkbox"/>	292	ELFV_dehydrog	ELFV_dehydrog_N
<input type="checkbox"/>	8	ELFV_dehydrog	ELFV_dehydrog_N Pfam-B_302
<input type="checkbox"/>	1	ELFV_dehydrog	ELFV_dehydrog_N Pfam-B_39687
<input type="checkbox"/>	2	ELFV_dehydrog	ELFV_dehydrog_N Pfam-B_34265
<input type="checkbox"/>	0	Sin dominios	

**Figura 15.** Secuencias por grupos de dominios

En este caso, hay un dominio bastante numeroso (ELFV\_dehydrog + ELFV\_dehydrog\_N), mientras que otros pocas secuencias tienen, además de los dos dominios ELFV\_dehydrog y ELFV\_dehydrog\_N, algún otro dominio especial. El usuario podrá elegir únicamente las secuencias que forman el gran grupo que tienen dos dominios, o quedarse con el resto, para observar detalladamente qué particularidades tienen. En este caso, se desea eliminar todo lo que no tenga esos dos dominios, quedándonos únicamente con 292 (Figura 15).

De esta forma, se ha reducido el número de proteínas existentes, desde más de 400 que había inicialmente a las 292 que han quedado de estos dominios. Posteriormente, se genera un FASTA

personalizable, poniendo por ejemplo una cabecera con el código UniProt Corto, Completo y el tamaño de la proteína.

El sistema solicita también qué fragmento de la secuencia desea exportar, por si se desea generar un FASTA pero, únicamente, con los aminoácidos de los dominios deseados. En este caso, como se ha realizado un filtrado previo, solo tenemos disponibles los 2 dominios que tenían todos las 292 secuencias que quedan. Supongamos que se desea estudiar únicamente el dominio ELFV\_dehydrog, para ello indicaremos al programa que exporte un fichero FASTA, con los campos indicados previamente para la cabecera, y que la secuencia corresponda a este dominio (ELFV\_dehydrog).

	Inicio	Final
Aminoácido ini...	<input type="radio"/>	<input type="radio"/>
ELFV_dehydrog	<input checked="" type="radio"/>	<input checked="" type="radio"/>
ELFV_dehydro...	<input type="radio"/>	<input type="radio"/>
Aminoácido final	<input type="radio"/>	<input type="radio"/>

Aceptar

**Figura 16.** Ventana que permite la elección selectiva de dominios

El FASTA generado tendrá la cabecera como hemos indicado (UniProt corto, UniProt largo y tamaño de secuencia), y la secuencia del dominio marcado (ELFV\_dehydrog) (Figura 17).

```

1 > Q47949|Q47949_9EURY|176
2 GGS LGRNEATARGASYTIREAAKVLGWGDLKGKTI A IQGYGNAGYYLAKIMSEDYGMKVAVSDTKGGIYNPI
3 > Q977X9|Q977X9_9EURY|419|
4 GGS LGRGTATAQGAIFTIREAAKALGIDLKGKTI AVQGYGNAGYYTAKLAKEQLGMKVAVSDSQGGIYNPN
5 > Q977U6|Q977U6_HALME|441
6 GGSEGRDTAPGRSVAIIAREADYLSWDIEDTTVAVQGFSGVGA PAARLLDDYGANVVAVSDVNGAIYDPDGI
7 > Q9HKG4|Q9HKG4_THEAC|436
8 GGS LGRFDSTGKGMFVLREGAKKIGLDLSKARVAVQGFNGVQFAVKFVEEMFGAKVVAVSDIKGGIYSEN
9 > Q97Y81|Q97Y81_SULSO|419
10 GGIGVRLYSTGLGVA TIARDAANKFIGGIEGSRV I IQFGNVGFFTAKFLSEMGAKIIGVSDIGGGVINENG
11 > Q97AP0|Q97AP0_THEV0|435
12 GGS LGRFDSTGKGMFVLREGAKKIGLDLSKARVAVQGFNGVQFAVKFVEEMFGAKVVAVSDIKGGIYSEN

```

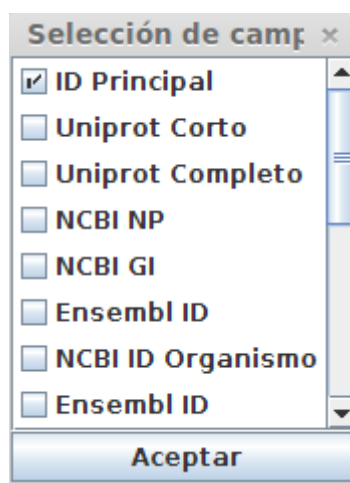
**Figura 17.** FASTA generado tras el filtrado por dominio

El programa diseñado permite que este mismo fichero pueda ser tomado como entrada de nuevo, si pasado un tiempo se desea reestudiar de nuevo. Además, si quisiéramos generar este mismo fichero pero con una cabecera distinta, sería un proceso sencillo y rápido, simplemente abriríamos el fichero de estado generado y volveríamos a exportar, pero eligiendo otros campos para la cabecera. Si no se dispone de ningún fichero de estado, se puede volver a cargar este fichero FASTA, para, ya sin realizar ningún filtrado, únicamente volver a exportarlo como FASTA pero con otra elección de los campos de cabecera.

#### 4.4.2. A partir de un fichero árbol en Newick

En otro caso, podemos tener un fichero de árbol en Newick (Cardona et al., 2008), ya filtrado y depurado, pero del que se quiere generar una representación gráfica por medio de iTOL con unos colores y figuras concretos para cada dominio. Habitualmente, para realizar esto, había que crear manualmente el fichero de configuración de iTOL, con sumo cuidado de establecer bien cada campo.

Gracias a la aplicación diseñada, este proceso se hará en pocos minutos. Tras abrir el fichero árbol Newick, el programa rellenará la tabla con los datos de las distintas bases de datos, y simplemente habrá que indicar que se desea exportar a iTOL, y el sistema solicitará los campos deseados como etiqueta del árbol (Figura 18).



**Figura 18.** Campos a exportar en el árbol

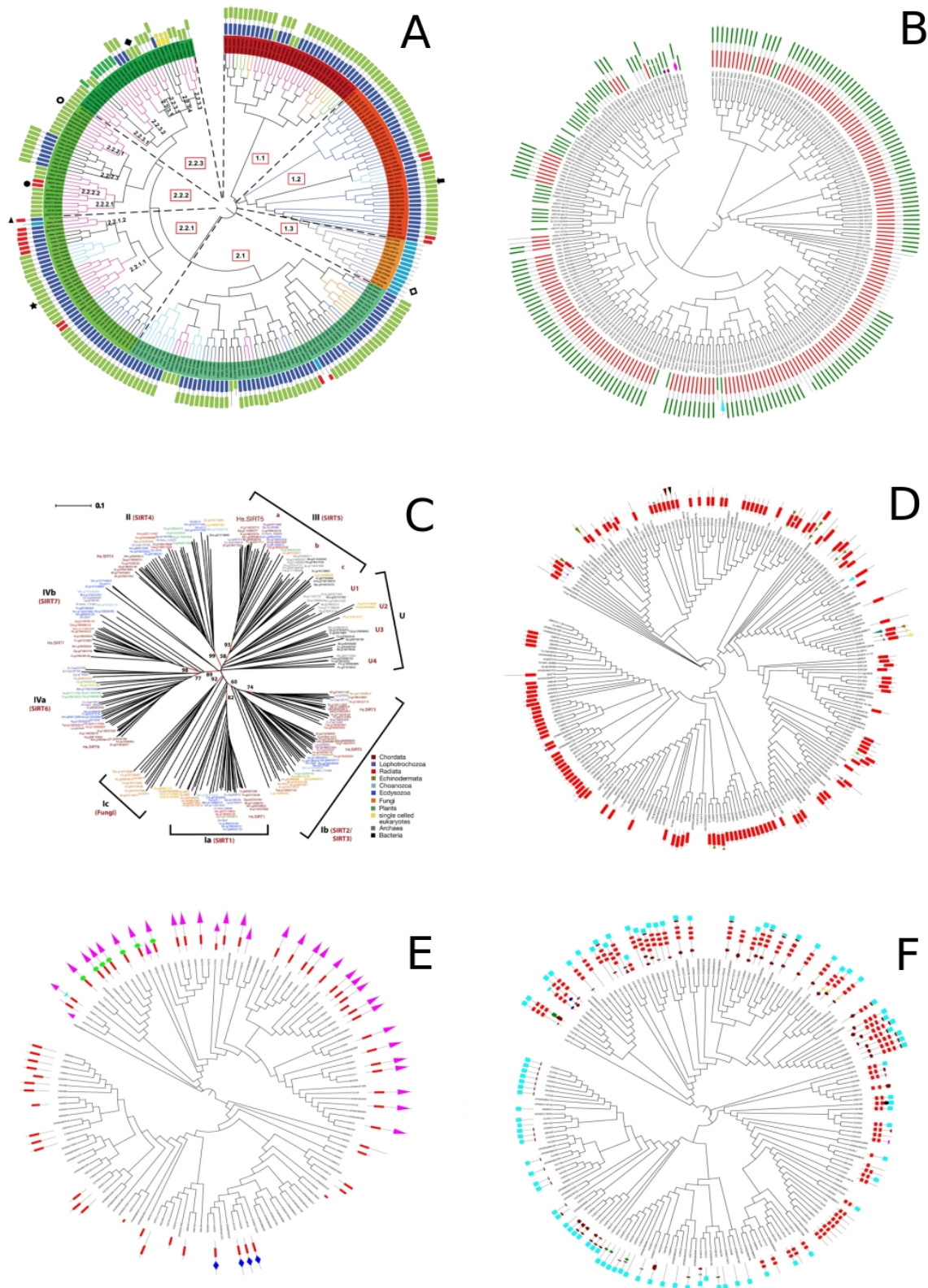
Se establecerá por ejemplo el ID Principal, para que haya unicidad de códigos, y entonces el sistema solicitará al usuario qué color y figura desea para cada dominio, así como un nombre significativo (que por defecto será el nombre del propio dominio). Una vez elegido esto, el usuario tendrá sus ficheros disponibles, para poder realizar, por medio de la plataforma iTOL, imágenes de su árbol. Si, una vez realizado, desea hacer cualquier cambio de figuras o colores, simplemente habría que volver a generar los ficheros, pero cambiando la elección, así que en unos segundos volvería a tener el árbol pintado con los nuevos colores y formas deseados.

#### **4.5. Evaluación del programa diseñado**

Además de todas las opciones detalladas en los apartados anteriores, con la herramienta diseñada también será posible realizar conversiones de árboles publicados en pocos minutos, como el mostrado en la Figura 19, que corresponden a los artículos publicados por Sánchez-Carrón et al. (2013) (Figura 19A) y Greiss and Gartner (2009) (Figura 19C), respectivamente. Asimismo, permite usar los árboles de webs como *TreeFam* (Schreiber et al., 2014) (Figura 19E) o *Genomicus* (Louis et al., 2013) (Figura 19F), pudiéndose observar no como códigos Ensembl, sino como códigos UniProt y con sus dominios PFAM.

Como se puede observar en la Figura 19B, el árbol obtenido con nuestro programa tiene la misma forma que el desarrollado en el artículo correspondiente a la Figura 19A, pero este estudio se realizó en 6 meses usando editores de texto y herramientas convencionales, y por medio del programa diseñado se podría haber realizado en pocos días. Asimismo, se puede observar en el caso de la Figura 19D que la información con el programa desarrollado en esta tesis de máster es mucho más completo y estandarizado (ya que se ha convertido a códigos UniProt, que dan más información sobre proteínas que NCBI o Ensembl) que el que se publicó en 2009 (Figura 19C), ya que muestra con mayor claridad la evolución de las distintas secuencias. Por último, las figuras 19E y 19F muestran una información mucho más potente en cuanto a representación e interpretación que la obtenida dentro de las webs de donde se han descargado los correspondientes árboles permitiendo, sin un conocimiento previo, la obtención de patrones distintos dentro de un mismo tipo de proteína, permitiéndole al investigador señalar a determinadas proteínas como posibles objetos de futuras investigaciones.





**Figura 19. Ejemplos de árboles obtenidos mediante el uso del programa desarrollado.** A) Árbol obtenido y publicado por Sánchez-Carrón et al. (2013). B) Árbol obtenido a partir de los mismos datos con el programa desarrollado y unificado a códigos UniProt. C) Árbol obtenido y publicado Greiss and Gartner (2009). D) Árbol obtenido a partir de los mismos datos con el programa desarrollado y unificado a códigos UniProt. E) Árbol obtenido de *TreeFam* (Schreiber et al., 2014). F) Árbol obtenido de *Genomicus* (Louis et al., 2013).

## 5. Discusión

El sistema diseñado permitirá que los investigadores accedan a la información de las secuencias a analizar de una forma homogénea, en una única herramienta, que sirva de unión de la multitud de datos existentes en diversas bases de datos biológicas.

Además de servir como nexo de unión de información existente en la web de forma dispersa, también posibilita que el usuario pueda reducir la gran cantidad de datos iniciales, cientos o miles de secuencias, aplicándole una serie de filtros, para quedarse con un número de elementos abordables. Esta reducción se podrá aplicar desde diversos puntos de vista, pudiendo el usuario optar por unos tipos de filtro u otros, en función de lo que le interese realizar en su investigación.

Por medio de la aplicación desarrollada, el investigador podrá desarrollar la tarea inicial de búsqueda de la información en pocos minutos, frente a los días o semanas, en función del número de secuencias, que llevaría realizar una recolección de datos si se hiciera de forma manual, ya que se requerirían de varias jornadas de trabajo únicamente para obtener todos los datos de las proteínas, mientras que ahora es un proceso automático y la obtención de esta información se realizará en unos minutos, independientemente del número de proteínas a buscar. Por lo tanto, la productividad aumenta considerablemente, y el usuario podrá dedicar todo ese tiempo a analizar más a fondo los datos.

Pero no solo facilita la obtención de datos, sino que toda la tarea de análisis y filtrado se podrá realizar también en pocos minutos, y tener varias vías de estudio paralelas, en función de diferentes criterios, en un tiempo mínimo. Realizar estas tareas manualmente requeriría un tiempo de varias semanas, y cualquier mínimo cambio conllevaría rehacer gran parte del trabajo, mientras que por medio de la herramienta diseñada se podrán realizar cambios (más o menos importantes) con un esfuerzo y un tiempo mínimo, de pocos minutos.

De esta forma, el usuario podrá trabajar con varios conjuntos de datos simultáneamente, de forma cómoda y sencilla, para investigar a fondo finalmente el que decida. O incluso realizar varios estudios, que parten de unos mismos datos, pero que después se dividen en varios subconjuntos.

Una opción inicial planteada fue la de almacenar en bases de datos locales toda la información relativa a las secuencias a estudiar por el usuario, pero se decidió realizar, cada vez que haya una nueva entrada de datos del usuario, conexiones *online* a las bases de datos, ya que de esta forma se tendría la garantía de que la información era actual. Una vez que el usuario tuviera esta información, el sistema ya sí permite trabajar de forma *offline*, así que de acuerdo al sistema diseñado solo hay que realizar la conexión a las bases de datos la primera vez (o más adelante si se quiere realizar un refresco de datos), y el tiempo necesario para obtener cientos de datos es de únicamente unos minutos.

El sistema podrá ser ampliado en el futuro, añadiendo nueva información incluso desde otras bases de datos. Para ello, habría que analizar cómo está almacenada esa información, y añadir al sistema nueva funcionalidad para que sea capaz de conectarse a las bases de datos, recoger la información y almacenarla en el sistema. Se podría también adaptar el programa y convertirlo en una herramienta web, para que sea más accesible a los usuarios.

Una vía de futuro del sistema puede ser su traducción a varios idiomas, inicialmente un sistema que permitiera inglés y español. Otra opción que se está estudiando es la de añadir una conciliación de los datos frente a árboles 16S, que permita cruzar la información obtenida con estos árboles.

Por último, cabe resaltar que el programa está siendo utilizado por el grupo de investigación de uno de los tutores en casos reales de su trabajo.



## Bibliografía

Cardona, G., Rosselló, F., and Valiente, G. (2008). Extended Newick: it is time for a standard representation of phylogenetic networks. *BMC Bioinformatics* 9, 532.

Finn, R.D., Bateman, A., Clements, J., Coghill, P., Eberhardt, R.Y., Eddy, S.R., Heger, A., Hetherington, K., Holm, L., Mistry, J., et al. (2014). Pfam: the protein families database. *Nucleic Acids Res.* 42, D222–230.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol.* 162, 705–708.

Greiss, S., and Gartner, A. (2009). Sirtuin/Sir2 phylogeny, evolutionary considerations and structural conservation. *Mol. Cells* 28, 407–415.

Hall, B.G. (2011). *Phylogenetic Trees Made Easy: A How To Manual*, Fourth Edition (Sunderland, Mass: Sinauer Associates, Inc.).

Hall, B.G. (2013). Building phylogenetic trees from molecular data with MEGA. *Mol. Biol. Evol.* 30, 1229–1235.

Holder, M., and Lewis, P.O. (2003). Phylogeny estimation: traditional and Bayesian approaches. *Nat. Rev. Genet.* 4, 275–284.

Letunic, I., and Bork, P. (2011). Interactive Tree Of Life v2: online annotation and display of phylogenetic trees made easy. *Nucleic Acids Res.* 39, W475–478.

Liu, L., Yu, L., Kubatko, L., Pearl, D.K., and Edwards, S.V. (2009). Coalescent methods for estimating phylogenetic trees. *Mol. Phylogenet. Evol.* 53, 320–328.

Louis, A., Muffato, M., and Roest Crollius, H. (2013). Genomicus: five genome browsers for comparative genomics in eukaryota. *Nucleic Acids Res.* 41, D700–705.

McLaren, W., Pritchard, B., Rios, D., Chen, Y., Flicek, P., and Cunningham, F. (2010). Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor. *Bioinforma. Oxf. Engl.* 26, 2069–2070.

NCBI Resource Coordinators (2014). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 42, D7–17.

Needleman, S.B., and Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48, 443–453.

Olsen G. (1990). Newick's 8:45" Tree Format Standard.

Pearson, W.R. (2014). BLAST and FASTA Similarity Searching for Multiple Sequence Alignment. In *Multiple Sequence Alignment Methods*, D.J. Russell, ed. (Humana Press), pp. 75–101.

Pearson, W.R., and Lipman, D.J. (1988). Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. U. S. A.* 85, 2444–2448.

Sánchez-Carrón, G., Martínez-Moñino, A.B., Sola-Carvajal, A., Takami, H., García-Carmona, F., and Sánchez-Ferrer, Á. (2013). New insights into the phylogeny and molecular classification of nicotinamide mononucleotide deamidases. *PLoS One* 8, e82705.

Schreiber, F., Patricio, M., Muffato, M., Pignatelli, M., and Bateman, A. (2014). TreeFam v9: a new website, more species and orthology-on-the-fly. *Nucleic Acids Res.* 42, D922–925.

UniProt Consortium (2008). The universal protein resource (UniProt). *Nucleic Acids Res.* 36, D190–195.

Whelan, S. (2008). Inferring trees. *Methods Mol. Biol.* Clifton NJ 452, 287–309.