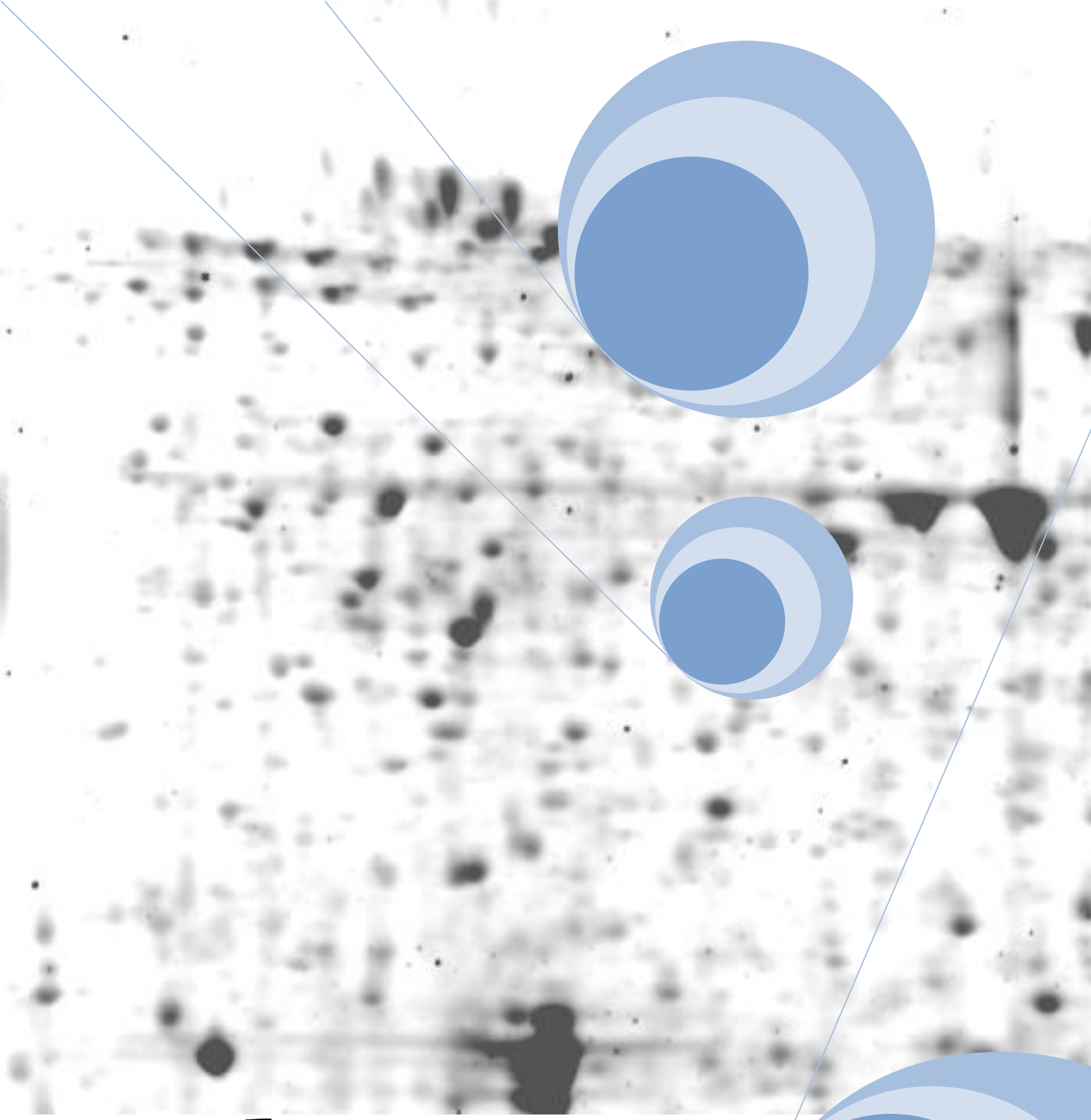
TÍTULO

BlasStorP
HERRAMIENTA NCBI-BLAST
A NIVEL LOCAL

AUTOR

Eduardo Chicano Gálvez

Esta edición electrónica ha sido realizada en 2010
Director A.J. Pérez Pulido y Juan Falgueras Cano
Curso **III Máster en Bioinformática**
ISBN 978-84-7993-191-9



BlaStorP

Herramienta NCBI-BLAST a nivel local

Proyecto Fin de Master Universidad Internacional de Andalucía

Eduardo Chicano Gálvez

Universidad Internacional de Andalucía

AGRADECIMIENTOS

En primer lugar me gustaría agradecer a mis directores de Tesis, Juan López Barea y José Alhama Carmona, el haberme dado la posibilidad de entrar en un mundo tan fascinante como la Proteómica.

Agradecer muy especialmente a Juan López Barea el apoyo incondicional que ha mostrado conmigo y mis proyectos durante la realización de la Tesis doctoral que ahora se encuentra en sus últimas fases de corrección y que próximamente será defendida. Sin el apoyo que me brindaste para el desarrollo de este Máster en Bioinformática nada de esto sería ahora realidad. De corazón, muchas gracias.

A mis tutores del Proyecto, Antonio J. Pérez Pulido y Juan Falgueras Cano. A Juan por abrirme los ojos con los proyectos Open Source y los sistemas basados en Unix (no todo es Windows) y especialmente a Antonio, por la paciencia que ha tenido con el bombardeo continuo de emails con mi desesperación ante los puntos donde me atascaba. ¡Muchísimas gracias a los dos!

A mis compañeros de "batalla": Dani Bonilla, Rafa Montes, Inma Osuna, Patricia, Bea, Amalia Vioque, Ricardo Fernandez, Chaparro, y todos aquellos que han pasado por nuestro laboratorio durante estos años: Eder da Costa, Jihene Ghedira, Pedro Costa, Sergio Fernández... por aguantar mis "paranoias" de programación.

Al "Canijo", a mi amigo del alma, Carlos Fuentes. Sobre todo por aconsejarme bien en aquellos momentos en los que me desesperaba y por su apoyo. ¡Que grande eres!

A Nieves, Julia, Pepa, Juan Jurado, Carmina y Carmen Pueyo… sois la leche, ¡seguid así!

A mis padres desde que me puse a estudiar Bioquímica no han dejado de apoyarme. Todos los proyectos que llevo hechos tienen una dedicatoria especial a ellos y el esfuerzo que han hecho para que hoy sea quien soy.

A mi mujer, Ana, sin la cual no habría podido hacer nada de nada en muchas parcelas de mi vida. Gracias a ella, comenzó esto. Todas las palabras del mundo no son suficientes para agracederte lo mucho que haces por mí todos los días. Te quiero.

”Lo que hacemos en la vida tiene su eco en la eternidad”

(Máximo Décimo Meridio, Gladiator, de Ridley Scott).

INTRODUCCIÓN

I. BIOINFORMATICA

La Bioinformática es, según una de sus definiciones más sencillas dada por el EBI (European Bioinformatics Institute), la aplicación de tecnología de computadores a la gestión y análisis de datos biológicos (http://www.ebi.ac.uk/2can/bioinformatics/bioinf_what_1.html). Según el NCBI (National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov/): "Bioinformática es un campo de la ciencia en el cual confluyen varias disciplinas tales como: biología, computación y tecnología de la información. El fin último de este campo es facilitar el descubrimiento de nuevas ideas biológicas así como crear perspectivas globales a partir de las cuales se puedan discernir principios unificadores en biología".

La Bioinformática es por lo tanto una disciplina científica que utiliza tecnología de la información para organizar, analizar y distribuir información biológica con la finalidad de responder preguntas complejas en biología. Es un área multidisciplinaria, la cual puede ser ampliamente definida como una interfase entre Biología y Computación.

Inicialmente, cuando comenzó esta rama de la ciencia, el concepto de Bioinformática se refería sólo a la creación y mantenimiento de bases de datos donde se almacena información biológica, tales como secuencias de nucleótidos y aminoácidos. Pero hoy día, la aplicación de la Bioinformática implica solucionar o investigar problemas sobre escalas de tal magnitud que sobrepasan las capacidades humanas usando para esto herramientas de sistemas y computación. Para ello se procede a la colección, organización, almacenamiento y recuperación de la información biológica que se encuentra en distintas bases de datos mediante el desarrollo de distintas técnicas: informática, matemática aplicada, estadística, ciencias de la computación, inteligencia artificial, química y bioquímica y simulación de sistemas, mecanismos todos ellos imprescindibles en Bioinformática y sin cuyo aporte no existiría esta rama de la ciencia.

Por otra parte, el desarrollo de bases de datos no solamente implica el diseño de las mismas, sino también el desarrollo de interfaces complejas donde los investigadores puedan acceder a los datos existentes y suministrar o revisar esos datos. Esas interfaces sirven después para que los investigadores puedan combinar toda esa información para formarse una idea de las situaciones experimentales normales o de referencia, de tal manera que puedan estudiar cómo estas situaciones pueden verse alteradas en los estados bajo estudio, por ejemplo, para localizar un gen dentro de una secuencia, predecir estructura o función de proteínas y agrupar secuencias de proteínas en familias relacionadas.

Las herramientas de software para Bioinformática van desde simples herramientas de línea de comandos hasta complejos programas gráficos y servicios web autónomos situados en compañías de bioinformática o instituciones públicas.

Los principales campos de la Bioinformática incluyen alineamientos de secuencias, predicción de genes, montaje de genomas, alineamientos estructurales de proteínas, predicciones de estructura de proteínas, predicciones de expresión génica, interacciones proteína-proteína, y modelado de la evolución.

Los servicios bioinformáticos básicos, de acuerdo a la clasificación del EBI (European Bioinformatics Institute), podrían clasificarse la siguiente manera:

- Servicios de obtención de información en línea (por ejemplo: Uniprot, http://www.uniprot.org/).
- Herramientas de análisis (por ejemplo EMBOSS, http://emboss.sourceforge.net/).
- Búsquedas de similitudes entre secuencias (como BLAST, http://blast.ncbi.nlm.nih.gov/Blast.cgi).
- Alineamientos múltiples de secuencias (por ejemplo ClustalW http://www.ebi.ac.uk/Tools/clustalw2/index.html).
- Análisis estructural (acceso a servicios de alineamiento estructural de proteínas como Swiss-Model, http://swissmodel.expasy.org/SWISS-MODEL.html).
- Servicios de acceso a literatura especializada y ontologías (PubMed, GO http://www.geneontology.org/).

II. PROTEÓMICA

La utopía que suponía hace 15 años la secuenciación de genomas completos es hoy en día toda una realidad. En base a los datos disponibles en febrero de 2009 se tiene constancia de la existencia de 4595 proyectos de secuenciación (completados ya 949), siendo en total 1016 de eucariotas (fuente: GOLD: www.genomesonline.org). Los grandes proyectos de secuenciación de genomas han generado una enorme cantidad de información y la necesidad de descifrar esta información genómica ha estimulado considerablemente el estudio directo y a gran escala de las proteínas.

El término proteoma, se usó por primera vez en 1996 para describir el conjunto de proteínas que se expresan a partir de un genoma. Se define como el conjunto de las proteínas expresadas en un organismo en un determinado momento bajo unas determinadas circunstancias.

El proteoma es un elemento muy dinámico, cuyos componentes varían entre organismos, tejidos, células u orgánulos como consecuencia de cambios en su entorno, situaciones de estrés, administración de drogas, efectores o señales bioquímicas o su estado fisiológico o patológico. Todos estos factores incrementan de forma considerable la complejidad del proteoma, como consecuencia de la activación o supresión de la expresión de genes, las alteraciones de las interacciones entre las proteínas, o los cambios en sus modificaciones postraduccionales. El objetivo principal en proteómica es la identificación de esas variaciones. Interesa conocer cuáles son las diferencias en los niveles de proteínas cuando un organismo está sometido a determinadas condiciones ambientales, por ejemplo, cuales son las diferencias en la expresión de determinadas proteínas cuando se compara un tejido sano con otro enfermo.

Para los análisis globales y separación de los componentes de un proteoma dado pueden utilizarse dos tipos de estrategias:

a) Convencional: se separan todas las proteínas de la muestra (mediante técnicas de electroforesis bidimensional generalmente) y posteriormente se tratan las proteínas de interés para su posterior identificación mediante espectrometría de masas.
b) De gran escala ("shot-gun proteomics"): la muestra es tratada con una enzima proteolítica y la mezcla de péptidos resultantes se separan mediante cromatografía líquida acoplada a un sistema de espectrometría de masas. Así se consigue la identificación masiva de todas las proteínas presentes en la mezcla.

Una vez que se ha realizado el análisis de imagen (en caso de haber utilizado la electroforesis bidimensional) o la cromatografía líquida (en el caso de las técnicas de análisis a gran escala) hay que identificar las proteínas de interés.

Debido a la gran capacidad de análisis (alto rendimiento), a su precisión y a la sensibilidad en la determinación de masas moleculares de las proteínas la tecnología analítica por excelencia en Proteómica para la identificación de proteínas es la espectrometría de masas. Un espectrómetro de masas es un instrumento analítico que convierte los componentes de una muestra en iones gaseosos (moléculas que están cargadas eléctricamente) y los separa en base a su relación masa/carga. En él se diferencian tres tipos de componentes: la fuente de ionización, el analizador y el detector.

a) FUENTES DE IONIZACIÓN

MALDI

La desorción de iones asistida por láser (MALDI: Matrix-assisted laser desorption ionization) es uno de los dos métodos de ionización "suaves" que actualmente se utilizan en espectrometría de masas. Este método fue desarrollado por Karas y Hillenkamp a finales de los años 80.

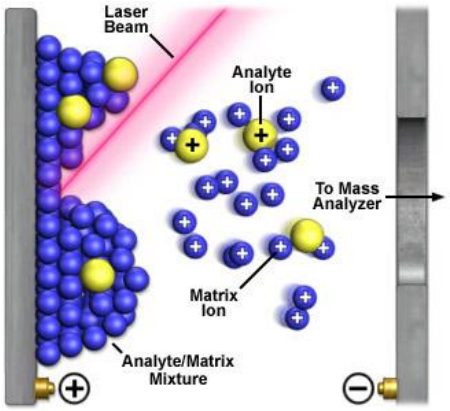


Figura 1. Esquema de funcionamiento de una fuente de ionización MALDI. (Fuente: www.magnet.fsu.edu).

Electrospray

La espectrometría de masas para biomoléculas basada en electrospray fue desarrollada por Fenn en 1989. Este método de ionización consiste en el uso de una aguja a través de la cual se bombean flujos muy reducidos de una disolución (normalmente una mezcla entre orgánico y agua, como metanol:agua, 50:50) que contiene al analito disuelto. En la punta de esa aguja se aplica un voltaje muy alto de forma que se produce una dispersión electrostática del fluido en pequeñas gotas que se evaporan y dan su carga a las moléculas de analito que se ionizan a presión atmosférica.

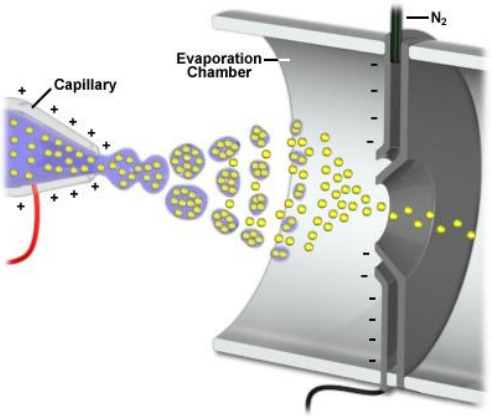


Figura 2. Esquema de funcionamiento de una fuente de ionización ESI. (Fuente: www.magnet.fsu.edu)

b) TIPOS DE ESPECTRÓMETROS DE MASAS

Los espectrómetros de masas miden la relación masa-carga (m/z) de analitos como proteínas, péptidos o fragmentos peptídicos. Existen tres principios básicos muy diferentes para realizar la separación según masas: separación en base al tiempo de vuelo (TOF MS), separación mediante campos eléctricos *cuadrupolo* generados por cilindros metálicos (Quadrupole MS) y separación mediante selección de iones desde una trampa magnética tridimensional (IT MS o FTIC MS).

Para la secuenciación de péptidos se llevan a cabo dos pasos de espectrometría de masas en tándem (MS/MS) que se pueden realizar empleando el mismo principio de separación o una combinación entre los tres anteriores.

Tanto MALDI como nESI se pueden acoplar a cualquiera de los tres métodos de separación. El hecho de que la fuente MALDI produzca pequeños pulsos de iones en el vacío y que la fuente nESI genere iones de forma continua a presión atmosférica ha hecho que la primera acople normalmente a analizadores tipo TOF y que la segunda se acople a *cuadrupolos* y trampas iónicas.

c) SECUENCIACIÓN DE PÉPTIDOS MEDIANTE ESPECTROMETRÍA DE MASAS

Una vez que el espectrómetro de masas ha determinado los valores m/z de todos los picos del espectro, se procede a la obtención de la estructura primaria o secuencia.

En la espectrometría de masas en tándem normalmente un ion en particular se aísla y se le imparte energía con un gas inerte en una cámara de colisión. De esta forma se produce la fragmentación mediante colisiones que hacen que el péptido se rompa en varios lugares (principalmente el enlace peptídico) y que se produzca un espectro de los fragmentos resultantes que se denomina “espectro de fragmentación”.

La forma de nombrar los distintos fragmentos depende de si contienen el extremo *N*-terminal o el *C*-terminal; así los fragmentos de tipo *a, b*, *c* contienen el extremo amino terminal mientras que los de tipo *x, y, z* contienen el carboxilo terminal. Los iones más comunes e informativos suelen ser los resultantes de la rotura del enlace peptídico y se llaman iones de “serie b” (con la carga en la parte aminoterminal del péptido) e iones de “serie y” (con la carga en la parte carboxiterminal).

Cada fragmento peptídico de una serie difiere de su vecino en un solo aminoácido. En principio es posible determinar la secuencia aminoacídica fácilmente, considerando solamente la diferencia de masas entre picos vecinos de una serie.

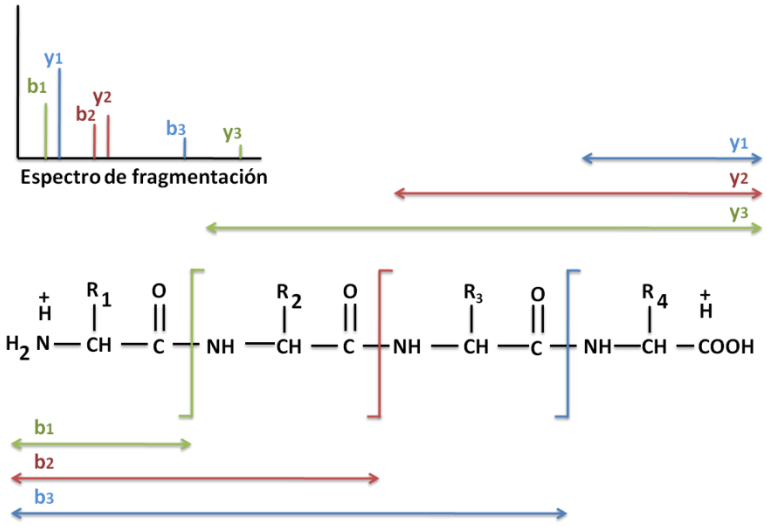


Figura 3. Esquema de fragmentación de un péptido teórico con las correspondientes series b/y. Se muestra también en la figura un espectro de masas teórico para dichas series.

Hoy día existen varios algoritmos informáticos que se encargan de la secuenciación *de novo* a partir de los datos de los espectros de fragmentación: Audens (http://www.ti.inf.ethz.ch/pw/software/audens/), Lutefisk(http://www.hairyfatguy.com/lutefisk/), NovoHMM, PepNovo, Peaks (http://www.bioinformaticssolutions.com) o DeNovoX (http://www.thermo.com) entre otros.

IDENTIFICACIÓN DE PROTEÍNAS EN BASES DE DATOS

Para la identificación en las distintas bases de datos existentes hay multitud de algoritmos informáticos disponibles en la web que ayudan a identificar proteínas en base a las secuencias peptídicas que se obtienen de un espectrómetro de masas. La identificación de estas proteínas se lleva a cabo mediante el análisis y correlación de los espectros experimentales con los espectros teóricos simulados a partir de los datos de secuencia contenidos en las distintas bases de datos o bien mediante la búsqueda de esas secuencias en las bases de datos. La estrategia de identificación de la proteína depende del tipo de espectrometría utilizada. Así encontramos tres grandes aproximaciones:

1. Identificación mediante huella peptídica o PMF: En ella se compara la lista de picos de masas peptídicas del espectro de masas (o fullscan) obtenido tras la digestión de una muestra con una lista de masas peptídicas calculadas de todas las entradas de una base de datos dada. Si ambas listas coinciden en varias masas se procesa la información obtenida y se le asigna una puntuación al estilo de la herramienta Blast. Su alta dependencia de la similitud con las

proteínas de las bases de datos utilizadas es una gran desventaja en este tipo de metodología de identificación.

2. Identificación mediante espectro de fragmentación o PFF: es una modificación del anterior con la diferencia de que los espectros que se utilizan no son de masas, sino espectros de fragmentación de péptidos generados a partir de la muestra que está pasando por el espectrómetro de masas. A partir de las listas de picos de los espectros de fragmentación se lleva a cabo la búsqueda en las bases de datos de espectros de fragmentación teóricos obtenidos a partir de las entradas existentes en la base de datos. La principal ventaja respecto a la identificación de proteínas por huella peptídica es su menor dependencia de las bases de datos, ya que requiere menor similitud con las proteínas presentes en estas bases y además se puede analizar mezclas complejas de proteínas sin necesidad de fraccionarlas previamente, aunque como desventaja tiene que sus niveles de sensibilidad son menores.

3. Secuenciación "*de novo*": En este tipo de identificaciones, se interpretan los espectros de fragmentación bien manualmente o bien de forma automatizada mediante software realizado para esta tarea (véase punto anterior).

Una vez obtenida la secuencia peptídica del espectro en cuestión se realiza una búsqueda en las bases de datos utilizando múltiples herramientas como por ejemplo Blast, las cuales nos darán una posible identificación (por similitud) de la proteína de la que procede el fragmento peptídico que hemos secuenciado, como es el caso que nos ocupa.

III. Búsqueda de similitud

Cuando se obtiene una secuencia nueva, normalmente, un investigador trata de identificar a que gen o proteína pertenece comparando la secuencia que ha obtenido experimentalmente contra una base de datos con secuencias previamente caracterizadas, para lo que se usa una herramienta de búsqueda de similitud.

BLAST (Basic Local Alignment Search Tool) es un programa informático de alineamiento local de secuencias, ya sea de ADN o de proteínas, que puede comparar una secuencia problema contra secuencias que se encuentren en una base de datos (como por ejemplo la base de datos de Swiss-Prot) encontrando las secuencias de la base de datos que tienen mayor parecido a la secuencia problema. Al utilizar un algoritmo heurístico no se nos garantiza la solución óptima pero podremos calcular la significación de los resultados, lo que nos dará un parámetro con el que valorar los resultados que se han obtenido tras la búsqueda.

BLAST es mantenido por el NIH a través de NCBI (National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov/) siendo de dominio público y de uso gratuito. Además de los servidores disponibles para BLAST en EEUU existen otros disponibles en Europa en el EBI, Sanger Institute y otros y también existe un paquete instalable en standalone, ofrecida por los distintos proyectos genómicos para analizar sus datos, etc.

La ventaja de utilizar BLAST a través del formulario que provee el NCBI es que el usuario no tiene que mantener al día las bases de datos y además la búsqueda se realiza muy rápidamente. Sin embargo, de esta manera no se pueden realizar búsquedas masivas ni personalizar la base de datos contra la que se desea comparar la secuencia en cuestión, ni poder manejar varios parámetros que en las búsquedas de NCBI están estandarizados (e-Value por ejemplo). Para estos casos está más indicada la aplicación local de BLAST, que provee una mayor flexibilidad para los usuarios avanzados.

BLAST utiliza el algoritmo Smith-Waterman que se basa en el uso de algoritmos de programación dinámica para garantizar que el alineamiento local encontrado es óptimo con respecto a un determinado sistema de puntuación utilizado como las matrices tipo BLOSUM o PAM.

El uso de este tipo de matrices permite a BLAST dar una puntuación a los alineamientos que realiza. Una matriz de este tipo contiene la puntuación (score) que se le da al alinear un nucleótido o un aminoácido X de la secuencia A con otro aminoácido Y de la secuencia B.

Existen distintos tipos de programas Blast para el análisis de secuencias tanto de nucleótidos (Blastn, BlastX, TBlastX) como de proteínas (Blastp, PSI-Blast, PHI-Blast, TBlastn, etc). En este

proyecto se utilizará por razones obvias a Blastp, que compara una secuencia de aminoácidos contra una base de datos del mismo tipo. Normalmente se usan las matrices BLOSUM o PAM para realizar los alineamientos.

IV. PERL

Perl es un lenguaje de programación creado y diseñado por Larry Wall en 1987. Es un acrónimo de *Practical Extracting and Reporting Language* lo que ya nos indica que se trata de un lenguaje de programación para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Perl es un lenguaje interpretado muy cercano aunque más fácil y de mayor nivel que los intérpretes de órdenes tipo UNIX como bash y tcsh, de los que provino su expansión. Así está inspirado en sh, awk y sed y finalmente C, denominador común de todos estos lenguajes de origen UNIX, pero está enfocado a ser más práctico y fácil que estos últimos.

En Perl las variables son prefijadas con signos @, $, % etc, para indicar su tipo. Aparte de su mucha mayor facilidad de creación de programas, con subrutinas, módulos y objetos, Perl se destaca por la facilidad con la que se manipulan las cadenas de letras. Además Perl tiene muchas funciones integradas para tareas comunes de UNIX y para acceder a los recursos del sistema.

Una diferencia fundamental de Perl con respecto a los otros lenguajes es la flexibilidad en el tamaño de los datos con los que trabaja ya que el límite lo pone la memoria que en ese momento se encuentre disponible en el sistema sobre el cual se encuentra. La memoria dinámica se maneja de forma transparente y es muy seguro construir estructuras muy complejas de datos sin que haya que preocuparse de punteros ni errores de acceso a direcciones de memoria.

Al ser un lenguaje de programación interpretado, el código de los guiones/programas en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Además es extensible a otros lenguajes, ya que desde Perl podremos hacer llamadas a subprogramas escritos en otros lenguajes y viceversa: desde otros lenguajes podremos ejecutar código Perl.

Este lenguaje se utiliza también frecuentemente para realizar “enlaces” entre sistemas e interfaces que no fueron diseñados para interaccionar entre ellos y para la minería de datos, convirtiendo o procesando grandes cantidades de datos para tareas como por ejemplo crear informes. También es ampliamente usado en Bioinformática donde es muy apreciado por su desarrollo rápido, tanto de aplicaciones como de despliegue, así como la habilidad de manejar grandes volúmenes de datos. Aun con todas estas propiedades su uso ha ido disminuyendo

(Perl ya ha superado con creces el objetivo para el que fue creado) a favor de nuevos lenguajes de programación en alza como Ruby (http://www.ruby-lang.org/es/) y proyectos asociados a esos lenguajes como BioRuby, una plataforma bioinformática basada en lenguaje Ruby (http://bioruby.org/)

Veamos por último un guión que cuenta los nucleótidos desde un fichero como un ejemplo de lenguaje Perl:

```
#!/usr/bin/env perl
$fichero1 = "nucleotidos.txt";

open (F1,"<$fichero1")||die "Error: no puedo abrir el archivo\n";

while (<F1>) {

        chomp;

        $conteo += length if ! /^>/;

}

print "Ud. tiene $conteo nucleotidos en el fichero";

close (F1);
```

V. OBJETIVOS DEL PROYECTO

El desarrollo del presente Proyecto de Fin de Máster viene motivado por la gran cantidad de problemas que el investigador novel en bioinformatica tiene a la hora de llevar a cabo la identificación masiva de secuencias peptídicas provenientes de secuenciación “de novo” mediante espectrometría de masas en las bases de datos más comúnmente utilizadas por los investigadores en Internet.

Estas bases de datos son muy complejas y contienen demasiada información que no siempre se usa en su totalidad. Por ejemplo, la página web de uniprot ofrece información muy extensa sobre la proteína que estemos buscando en varios subapartados: nombres y orígenes, atributos de la proteína, comentarios de la anotación, ontología, anotaciones sobre la secuencia, secuencia en formato fasta y propiedades como longitud y masa teóricas, referencias cruzadas, referencias bibliográficas, etc.

De forma general, la mayoría de los servicios basados en web tienen este formato, con demasiada información que resulta inabarcable para el investigador que mayoritariamente quiere restringir su búsqueda al nombre, función y las propiedades que le permitan identificar

por similitud la secuencia en cuestión (score y e-value), o sea, una pequeñísima parte de la información ofrecida por estos servicios.

Por otra parte, en el caso que nos ocupa, al hecho de trabajar en un campo relativamente nuevo como la proteómica se suma también la complicación de estar trabajando con organismos no secuenciados. Las secuencias de los organismos con los que se trabaja están escasamente representadas en las bases de datos debiendo realizar gran cantidad de trabajo de minería de datos y búsquedas para poder dar una mínima garantía de identificación de proteínas que creemos que están involucradas en las distintas situaciones experimentales con las que trabajamos en el laboratorio.

La creación de este programa podrá constituir una herramienta más eficaz para la realización de este trabajo y ayudará a ahorrar tiempo en la identificación y búsqueda de las funciones más comunes de las proteínas que estemos tratando de identificar, además de resultar una herramienta sencilla y disponible por la web en el caso de montar un servidor público o bien de forma local.

Además, se ha de tener en cuenta que gracias al aspecto local de esta herramienta será totalmente adaptable a las necesidades del investigador permitiendo encontrar similitud en secuencias muy cortas como las que generalmente se encuentran en las secuenciaciones "*de novo*" que nos ocupan.

Más adelante, se tratará de ampliar la información que ofrece el programa (o guión) mejorando la presentación de resultados y ampliando la información suministrada al usuario mediante la inclusión de nuevos módulos que nos puedan dar información de la estructura tridimensional de la proteína identificada o incluso un cálculo de su masa relativa y punto isoeléctrico teóricos.

MATERIALES Y MÉTODOS

Para el desarrollo del presente Proyecto de Fin de Master se ha utilizado como plataforma Microsoft Windows Vista. Todos los materiales que a continuación se describen se deben instalar en dicha plataforma tal como se indica en el Anexo 1. También se describen las posibles instalaciones de material necesario en otras plataformas como MacOS o Linux (se ha elegido la distribución Ubuntu Jaunty Jackalope 9.04 en este caso) en los anexos 2 y 3.

El programa final está enteramente escrito en Perl y ha sido implementado en un servidor web. El código fuente del mismo puede ser consultado en el fichero adjunto a esta memoria.

I. INTÉRPRETE PERL

El lenguaje Perl necesita un intérprete, el cual está escrito en C, junto con una gran colección de módulos, escritos tanto en Perl como en C. La distribución fuente tenía en 2005 un tamaño de unos 12 MB comprimida en un archivo tar. En dicha distribución existían 500 módulos en la distribución, sumando 200.000 líneas de Perl y unas 350.000 líneas adicionales de código C.

El intérprete tiene una arquitectura orientada a objetos. Todos los elementos del lenguaje Perl (escalares, listas, hashes, referencias a código, manejadores de archivos) están representados en el intérprete como estructuras C. Las operaciones sobre estas estructuras están definidas como una numerosa colección de macros, *typedef* y funciones.

II. BIOPERL

Además del intérprete de Perl, necesitaremos instalar una serie de módulos específicos para nuestro Programa. Estos módulos son suministrados por la librería BioPerl.

BioPerl es una enorme colección de módulos Perl que facilitan el desarrollo de scripts perl para aplicaciones bioinformáticas. También suministra interfaces para el análisis de secuencias con una gran variedad de programas externos (FASTA, BLAST, ClustalW, etc.) así como interfaces para trabajar con bases de datos remotas (GenBank, EMBL, etc.) o locales (MySQL, Bases de datos con ficheros planos, GFF, etc.) para el almacenamiento y recuperación de estas secuencias.

Bioperl suministra módulos para la mayoría de las tareas de uso común en la programación bioinformática. Estas tareas incluyen:

- Acceso a datos de secuencias desde bases de datos tanto locales como online
- Transformaciones en distintos formatos de bases de datos o ficheros
- Manipulación de secuencias individuales
- Búsqueda de secuencias similares
- Creación y manipulación de alineamientos de secuencias
- Búsqueda de genes y otras estructuras en DNA
- Desarrollo de anotaciones de secuencias comprensibles para computación

Los módulos a instalar son los siguientes son listados en la tabla 1:

Tabla 1. Módulos Bioperl que habrán de ser instalados para el correcto funcionamiento del programa

Name	perl 5.10
BioPerl-Regular Releases	http://bioperl.org/DIST
BioPerl-Release Candidates	http://bioperl.org/DIST/RC
Kobes	http://cpan.uwinnipeg.ca/PPMPackages/10xx/
Bribes	http://www.Bribes.org/perl/ppm
tcool	No Disponible

III. CGI

El CGI (por sus siglas en inglés “Common Gateway Interface”) es de las primeras formas de programación web dinámica. Cuando Internet inició su funcionamiento solo se podía ver texto, imágenes y enlaces, todo ello estático y nada dinámico. Gracias a la introducción de “plugins” en los navegadores, se permitió mayor interactividad entre el usuario y el cliente, aunque estaba limitado por la velocidad y la necesidad de tener que bajar e instalar cada “plugin” que se necesitara, por lo que estos se desarrollaron mayormente en áreas de vídeo, audio y virtualización.

El CGI supuso un cambio en la forma de manipular información en la web y es una importante tecnología que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web y que tiene la ventaja de correr en el servidor cuando el usuario lo solicita por lo que es dependiente del servidor y no de la computadora del usuario.

El CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME. Las aplicaciones que se ejecutan en el servidor reciben el nombre de CGIs.

Para el desarrollo del programa de este proyecto necesitaremos tener instalado el módulo CGI de Perl, lo que nos permite ejecutar el programa en el servidor web y devolver los resultados via HTML. Normalmente dicho módulo viene con la distribución estandar de PERL. Para saber si tenemos instalado el módulo basta con escribir en una terminal perl -e 'use Nombre_del_Módulo'. En caso de que este instalado aparecerá de nuevo el prompt de la consola, mientras que si no está instalado aparecerá un mensaje de error.

IV. APACHE

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation (http://www.apache.org/). Es un software servidor HTTP de código abierto para plataformas Unix, MS Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual y será utilizado como servidor local durante este PFM.
Durante el desarrollo de este PFM se utilizó la distribución 2.0.63 (http://ftp.udc.es/apache-dist/httpd/binaries/win32/apache_2.0.63-win32-x86-no_ssl.msi) para MS Windows aunque hoy día ya existe una nueva versión disponible (2.2.11). El archivo es suministrado vía ftp, comprimido en zip, con un tamaño de 7,9 MB. Apache es el que permite el uso del programa desarrollado en este proyecto.

V. BASES DE DATOS

Para trabajar por supuesto necesitaremos una base de datos local. (Véase Anexo I, Apartado IV: BASES DE DATOS UTILIZADAS)
En el desarrollo de este trabajo se ha utilizado la base de datos de Swiss-Port porque es la base de datos de proteínas con mayor calidad de anotaciones siendo incluso revisada manualmente por expertos en anotación proteica de forma continua, hecho que no ocurre por ejemplo, en las bases de datos pertenecientes al NCBI. Esta base de datos está depositada en el servidor de Uniprot.org (http://www.uniprot.org/downloads)

De este repositorio se descargará la opción UniProtKB/Swiss-Prot en formato FASTA. El archivo que nos vamos a descargar vía ftp viene comprimido como .gz y tiene un tamaño de 61,2 MB. Una vez descomprimido, tiene un tamaño de unos 188 MB, tratándose de un fichero plano en formato FASTA. Esta base de datos quedará con el nombre de uniprot_sprot.fasta.

Por otra parte, debido a necesidades que surgirán más adelante para mostrar los resultados, debemos descargar también la misma base de datos pero en formato .dat, ya que el script requerirá más adelante un fichero .idx donde se “indexa” toda la base de datos para buscar anotaciones de las fuentes: Keywords, Gene Ontologies y enlaces con Interpro. Dicho fichero se encuentra también en el servidor de Uniprot.org pero no está visible como en el

caso de la base de datos en formato FASTA, sino que hay que buscarlo en la zona ftp en la siguiente dirección: ftp://ftp.uniprot.org/pub/databases/uniprot_datafiles_by_format/flatfile/uniprot_sprot.dat.gz

Se descargará un archivo comprimido en .gz que tiene un tamaño aproximado de 317 MB y que tras ser descomprimido tiene un tamaño total aproximado de 1,7 GB. Este archivo se formateará a .idx como veremos más adelante.

VI. PAQUETE SOFTWARE BLAST

Para poder realizar búsquedas locales necesitaremos instalar el paquete de software BLAST disponible en el servidor de NCBI: http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=Download

La herramienta Formatdb (incluida dentro del paquete Blast) es una de las herramientas principales para la consecución de nuestro PFM. Esta herramienta se usa para formatear bases de datos, bien sean de nucleótidos o proteínas como es nuestro caso, y adecuarlas al uso de otras herramientas que utiliza Blast en sus búsquedas.

La base de datos que se formatea con Formatdb debe tener formato FASTA, como la que hemos comentado antes o bien ASN.1, siendo el más común el primero de estos dos casos. Una vez que se formatee la base de datos, Formatdb deja de ser necesario para el resto del PFM.

VII. FORMATO DE .DAT A .IDX

Como comentamos anteriormente, necesitaremos formatear también la base de datos de su extensión .dat a .idx para ofrecer al usuario más resultados tras la búsqueda. Para ello utilizaremos un pequeño guión en Perl que utiliza el módulo Swissprot de BioPerl. El código es el siguiente:

```
#!/usr/bin/env perl
# AJPerez, 2007-02-20
# Index a file with Swiss-Prot records
# Running: perl indexSwissprotFile.pl swissprot_file

use Bio::Index::Swissprot;
$ENV{BIOPERL_INDEX_TYPE} = "SDBM_File";

my $file = shift;
my $inx = Bio::Index::Swissprot->new(-filename => $file . ".idx", -write_flag => 1);
$inx->make_index($file);
```

VIII. EDITOR HTML

Necesitaremos un editor HTML cualquiera para componer la página de inicio y de resultados.

Para ello se puede escoger cualquier editor como KompoZer (el que se ha escogido para trabajar en este PFM) o Dreamweaver, Frontpage (aunque se desaconseja su uso por estar orientado a obtener páginas optimizadas solo para Internet Explorer)

RESULTADOS Y DISCUSIÓN

I. DESCRIPCIÓN DE LA HERRAMIENTA DESARROLLADA: BlaStorP

El programa desarrollado ha sido implementado en un servidor web al que llamo BlaStorP por la herramienta central del programa (Blast) y el sufijo del campo en el que es aplicado escrito al revés (PROTeómica). El servidor web es accesible desde http://jaguar.genetica.uma.es/blast y tiene una serie de parámetros de entrada necesarios para obtener la predicción final, los cuales serán descritos en las siguientes secciones.

I.A. Entradas

El programa acepta, como entrada, una secuencia proteica representada en el alfabeto de aminoácidos de una sola letra. La longitud máxima permitida es de 25 aminoácidos, para así bloquear posibles intentos de saturación del servidor mediante la introducción de secuencias largas que incrementarían en demasía el tiempo de realización de la búsqueda o bloquearían la correcta ejecución del Programa.

I.B. Matrices de puntuación

Para comparar las secuencias, BlaStorP utiliza matrices de puntuación BLOSUM como una forma de cuantificar la similitud entre la secuencia problema y las secuencias de la base de datos.

Para calcular la puntuación de cada alineamiento, es necesario dar una mayor puntuación a los aminoácidos idénticos que a las posiciones similares, y a éstas más que a las no similares. La manera de permitir flexibilidad a la hora de determinar qué puntuación se asigna para cada cambio, es la utilización de matrices de sustitución (PAM o BLOSUM), en las cuales la puntuación para cada posible reemplazamiento se encuentra predeterminada.

El hecho de haber utilizado matrices BLOSUM para el desarrollo de BlaStorP es debido a que entre las matrices PAM y BLOSUM existe una equivalencia (Tabla 2), la cual depende del porcentaje de identidad (%id), por lo que en caso de comparar puntuaciones con ambos tipos de matrices se debe tener presente este matiz. Así, no es comparable Blosum62 con PAM30 pero sí Blosum60 con PAM160 y un 40% Identidad. Por ello se deja a la elección del usuario el utilizar uno u otro tipo de matriz según esté buscando un mejor rendimiento en secuencias muy cercanas o secuencias divergentes. En el caso que nos ocupa, las matrices BLOSUM de identidad más alta (BLOSUM90, BLOSUM 80) suelen preferirse a las matrices BLOSUM con identidad más baja (62, 50, 45) para identificar fragmentos peptídicos tan cortos con los que se quiere buscar secuencias con un alto grado de identidad.

Tabla 2. Equivalencias entre Matrices PAM y BLOSUM dependiendo del % Identidad

PAM	**0**	**30**	**80**	**110**	**160**	**200**	**250**
BLOSUM	**100**	**99**	**95**	**85**	**60**	**52**	**45**
% ID	100	75	60	50	40	25	20

I.C.. Salidas

El Programa BlaStorP ofrece como resultado una página html que contiene los resultados tras el uso de Blast.

El programa devuelve los resultados en función de sus valores de E-value (número de resultados con igual puntuación que la obtenida en una búsqueda al azar, por lo que cuanto más pequeño es el E-value más significativo es el resultado), esto es, los resultados se muestran como sigue: en verde los mejor valorados (e-value con mayor significación), en naranja los llamados resultados "umbrales" (con valores de e-value con significación intermedia) y finalmente, en rojo los resultados cuyos valores no son recomendables (valores e-value con baja significación) y que son dados por si no existieran datos significativos con los que trabajar. Cada fila del informe contiene la siguiente información: en la primera columna el nombre de la proteína, en la segunda el "accesion number" de la base de datos UniprotKB y un link hacia la misma, en la tercera, cuarta y quinta tenemos E-value, score y % de identidad con la secuencia de la base de datos para dar una idea de la significación de cada resultado al usuario.

Por último, bajo los resultados obtenidos de cada bloque (verde/ambar/rojo), se muestra una lista con anotaciones y su frecuencia de aparición, cuyo criterio es usado para ordenarlas.

Las anotaciones incluidas en el resultado son : las Keywords de Uniprot con sus respectivos enlaces para obtener la descripción de cada termino, los términos Gene Ontologie también presentadas como en el caso anterior y finalmente una última lista donde tenemos disponibles las anotaciones InterPro.

Bajo todos los resultados generados y agrupados por los criterios anteriormente mencionados se muestra el resultado blast al completo haciendo uso del módulo Bio::SearchIO::Writer::HTMLResultWriter de BioPerl (Figura 4).

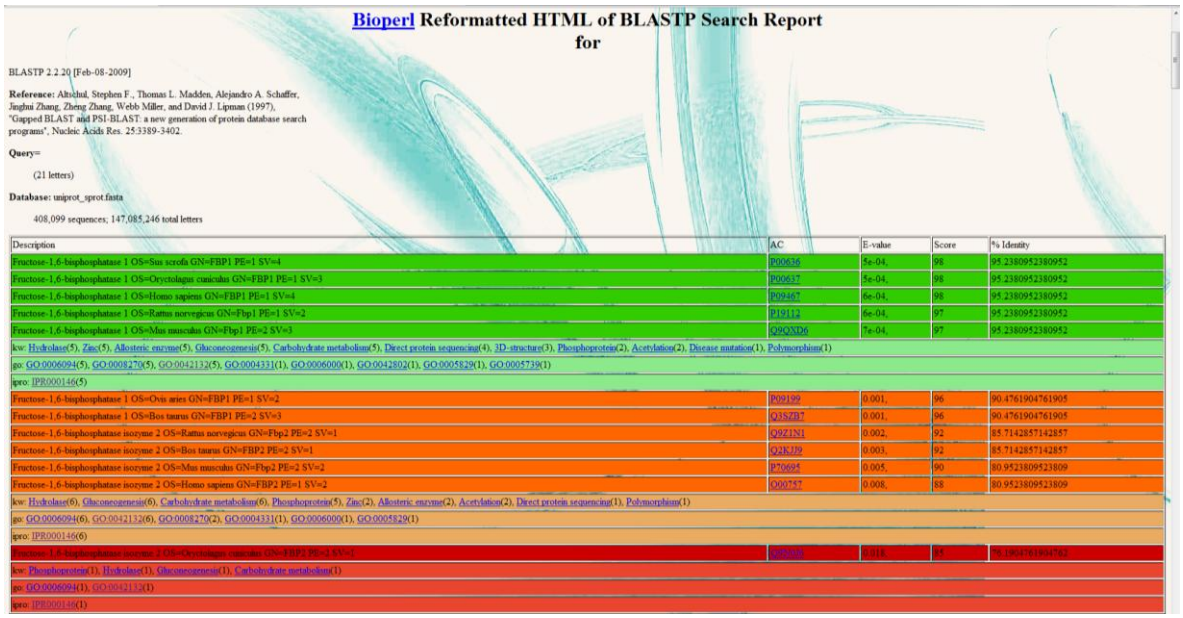


Figura 4. Captura de pantalla donde se observan los resultados obtenidos tras la ejecución del BlaStorP.

De esta manera, cuando el usuario comience a utilizar la herramienta encontrará un entorno amigable y que ofrece datos rápidos y organizados, en definitiva, datos "abarcables" de un solo vistazo y con enlaces a las bases de datos de mayor interés, para seguir buscando sólo la información que desee.

II. ENTRENAMIENTO DEL MÉTODO

Para comprobar la utilidad de la herramienta que se ha creado, se realizaron diversas pruebas con una serie de secuencias procedentes de experimentos reales (Tabla 3) llevados a cabo en los laboratorios del Departamento de Bioquímica y Biología Molecular de la Universidad de Córdoba y encuadrados dentro de la Tesis Doctoral titulada "Análisis proteómicos del desarrollo larvario del lenguado senegalés (*Solea senegalensis*) y anomalías durante su desarrollo" y que será defendida en los próximos meses.

Mediante la utilización de las secuencias citadas anteriormente, se comprobaron los efectos que tenían lugar en los resultados cuando se modificaban parámetros como el porcentaje de identidad a la hora de realizar la búsqueda, los valores de E-value como umbral de los resultados Blast, y las distintas matrices de puntuación. De ese modo se podrían ajustar los parámetros del programa para obtener los mejores resultados

Tabla 3. Secuencias utilizadas durante las pruebas de BlaStorP y descripción de las mismas

Secuencia	Descripción
SKQLEDDLVALQK	tpma; alpha-tropomyosin
IQLVEEELDR	tpma; alpha-tropomyosin
EQQEAIEHIDEVQNEIDR	RCJMB04_8c13, SET; SET translocation
AGFAGDDAPR	Actin, alpha cardiac related cluster
GAHQNIIPASTGAAK	GAPDH [Danio rerio]
VPVADVSVVDLTCR	GAPDH [Danio rerio]
ATDFVVDKPGKFK	Isocitrate dehydrogenase 2 (NADP+), mitochondrial [Danio rerio]
ATDFVVDKPGK	Isocitrate dehydrogenase 2 (NADP+), mitochondrial [Danio rerio]
DLPLAQGIKFE	Aldehyde dehydrogenase 7 family, member A1 [Danio rerio]
EGNGTVMGAELR	cardiac mysosin light chain 1 [Danio rerio]
VAYNQIADIMR	Zgc:77231 [Danio rerio]
AGIANLYGIAGSTNVTGDQVK	Fructose-1,6-bisphosphatase 1 [Danio rerio]
HKLEDGYPK	warm-temperature-acclimation-related-65kDa- protein wtap [Takifugu rubripes]
GECMADSVLFFK	warm-temperature-acclimation-related-65kDa- protein-like [Takifugu rubripes]
GGDDLDPNYVLSSR	Muscle creatine kinase related cluster
LSIEALNSLDGEFK	Muscle creatine kinase related cluster
GENCIAAGR	Formyltetrahydrofolate dehydrogenase related
GFTLPPHNSR	creatine kinase CKM3 [Danio rerio]
VLVTGAAGQIAYSLLYSIAK	cytosolic malate dehydrogenase A [Danio rerio]
ATDFVVDKPGKFK	Isocitrate dehydrogenase 2 (NADP+), mitochondrial [Danio rerio]

II.A. Pruebas con % Identidad

Las pruebas con porcentaje de identidad se llevaron a cabo modificando el valor de umbral de porcentaje de identidad (%id) para realizar los agrupamientos de los resultados en los colores verde, ambar y rojo. Se escogió un valor inicial del 75% y se utilizó la matriz BLOSUM62. Los resultados se muestran en la Tabla 4.

Con los resultados obtenidos se sugirieron como parámetros por defecto para dar como buena la identificación, aquellos que tienen menor valor de E-value y mayor de porcentaje de identidad.

En éste caso se escogió %ident = 66 y como intervalos para organizar los resultados se tomaron como componentes del intervalo verde los valores de E-value inferiores a 1e-05,

como componentes del intervalo ambar aquellos situados entre 1e-05 y 1e-03 y finalmente, como componentes del intervalo rojo aquellos superiores a 1e-03.

Tabla 4. Resultados de identificación tras utilizar los parámetros iniciales de búsqueda

Resultados con %ident=75	Anotada correctamente	Evalue	%ident
SKQLEDDLVALQK	SI	9,2	100
IQLVEEELDR	SI	273	100
EQQEAIEHIDEVQNEIDR	SI	0,017	100
AGFAGDDAPR	SI	14	100
GAHQNIIPASTGAAK	SI	0,34	100
VPVADVSVVDLTCR	SI	1,4	92,85
ATDFVVDKPGKFK	SI	106	76,92
ATDFVVDKPGK	SI	307	81,81
DLPLAQGIKFE	SI	17	90,9
EGNGTVMGAELR	SI	6	100
VAYNQIADIMR	SI	21	100
AGIANLYGIAGSTNVTGDQVK	SI	0,0005	95,23
HKLEDGYPK	NO		
GECMADSVLFFK	SI	22	66
GGDDLDPNYVLSSR	SI	0,22	100
LSIEALNSLDGEFK	SI	4,2	85,71
GENCIAAGR	SI	56	100
GFTLPPHNSR	SI	235	80
VLVTGAAGQIAYSLLYSIAK	SI	0,002	100
ATDFVVDKPGKFK	SI	106	76,92

II.B. <u>Pruebas con E-value</u>

Se realizaron diversas pruebas dejando fijo un valor para la variable E-value de Blast (resultados no mostrados) pero el valor de e-value puede variar según la longitud de la región de similitud, por lo que se optó por dejar que dicho valor fuera cambiando según la longitud de la secuencia introducida. Para ello se colocó dentro del código una parte donde se determinó como E-value inicial el valor 1000 y al cual, dependiendo de la longitud de la secuencia se le resta más o menos cantidad. De esta manera, todas aquellas secuencias cuya longitud no excediera de 15 aminoácidos no resultarían afectadas y tendrían un valor E-value para Blastall de 1000. Para aquellas secuencias que superasen ese umbral, se midió su longitud y dicha longitud se multiplicó por 20 obteniendo así el valor que se tenía que restar al E-value inicial y así poder ejecutar Blastall con este parámetro más adecuado a la longitud de dicha secuencia.

II.C. Pruebas con matrices de puntuación

También se llevaron a cabo distintas pruebas con varias matrices de puntuación: BLOSUM90, BLOSUM80, BLOSUM62 (la que posee Blast por defecto), BLOSUM50 y BLOSUM45. No se realizaron pruebas con las matrices PAM equivalentes ya que se espera que los resultados serían los mismos con las matrices equivalentes.

Los resultados pueden observarse en las Tablas 6, 7, 8, 9 y 10. Como se puede comprobar, los resultados difieren en función de la matriz que se utilice

Tabla 6. Resultados obtenidos tras utilizar la matriz BLOSUM90.

Resultados con BLOSUM90	Descripción	Anotada correctamente	Evalue	%id
SKQLEDDLVALQK	Tropomyosin alpha-1 chain OS=Danio rerio	SI	6,60E+00	100
IQLVEEELDR	Tropomyosin OS=Branchiostoma belcheri	SI	2,53E+02	100
EQQEAIEHIDEVQNEIDR	Protein SET OS=Homo sapiens	SI	7,00E-03	100
AGFAGDDAPR	Actin (Fragment) OS=Acetabularia cliftonii	SI	1,10E+00	100
GAHQNIIPASTGAAK	Glyceraldehyde-3-phosphate dehydrogenase OS=Oncorhynchus mykiss	SI	3,00E-03	100
VPVADVSVVDLTCR	Glyceraldehyde-3-phosphate dehydrogenase OS=Oncorhynchus mykiss	SI	5,00E-02	92,85
ATDFVVDKPGKFK	Isocitrate dehydrogenase [NADP], mitochondrial OS=Dictyostelium discoideum	SI	6,50E+01	69,23
ATDFVVDKPGK	Isocitrate dehydrogenase [NADP], mitochondrial OS=Dictyostelium discoideum	SI	4,70E+01	81,81
DLPLAQGIKFE	Alpha-aminoadipic semialdehyde dehydrogenase OS=Bos taurus	SI	1,40E+00	90,9
EGNGTVMGAELR	Myosin light chain 3 OS=Bos Taurus	SI	2,70E-01	100
VAYNQIADIMR	Myosin light chain 1, skeletal muscle isoform OS=Mugil capito	SI	1,10E+00	100
AGIANLYGIAGSTNVTGDQVK	Fructose-1,6-bisphosphatase 1 OS=Homo sapiens	SI	9,00E-07	95,23
HKLEDGYPK	Chaperone protein clpB OS=Corynebacterium efficiens	NO	4,70E+02	77,77
GECMADSVLFFK	Hemopexin OS=Pongo abelii	SI	4,00E+01	66,66
GGDDLDPNYVLSSR	Creatine kinase M-type OS=Bos Taurus	SI	4,00E-03	100
LSIEALNSLDGEFK	Creatine kinase M-type OS=Gallus gallus	SI	1,80E-01	85,71
GENCIAAGR	10-formyltetrahydrofolate dehydrogenase OS=Xenopus laevis	SI	6,50E+00	100
GFTLPPHNSR	Creatine kinase M-type OS=Sus scrofa	SI	4,70E+01	80
VLVTGAAGQIAYSLLYSIAK	Malate dehydrogenase, cytoplasmic OS=Gallus gallus	SI	2,00E-06	100
ATDFVVDKPGKFK	Isocitrate dehydrogenase [NADP], mitochondrial OS=Dictyostelium discoideum	SI	6,50E+01	69,23

Tabla 7. Resultados obtenidos tras utilizar la matriz BLOSUM80.

Resultados con BLOSUM80	Descripción	Anotada correctamente	Evalue	%id
SKQLEDDLVALQK	Tropomyosin alpha-1 chain OS=Liza aurata	SI	7,10E+00	100
IQLVEEELDR	Tropomyosin OS=Branchiostoma belcheri	SI	2,57E+02	100
EQQEAIEHIDEVQNEIDR	SET_RAT Protein SET OS=Rattus norvegicus	SI	9,00E-03	100
AGFAGDDAPR	Actin (Fragment) OS=Acetabularia cliftonii	SI	2,70E+00	100
GAHQNIIPASTGAAK	Glyceraldehyde-3-phosphate dehydrogenase OS=Oncorhynchus mykiss	SI	2,00E-02	100
VPVADVSVVDLTCR	Glyceraldehyde-3-phosphate dehydrogenase OS=Oncorhynchus mykiss	SI	1,50E-01	92,85
ATDFVVDKPGKFK	Isocitrate dehydrogenase [NADP], mitochondrial OS=Bos taurus	SI	9,40E+01	76,92
ATDFVVDKPGK	Isocitrate dehydrogenase [NADP], mitochondrial OS=Dictyostelium discoideum	SI	9,10E+01	81,81
DLPLAQGIKFE	Alpha-aminoadipic semialdehyde dehydrogenase OS=Homo sapiens	SI	3,30E+00	90,9
EGNGTVMGAELR	Myosin light chain 3 OS=Bos taurus	SI	8,60E-01	100
VAYNQIADIMR	Myosin light chain 1, skeletal muscle isoform OS=Mugil capito	SI	3,00E+00	100
AGIANLYGIAGSTNVTGDQVK	Fructose-1,6-bisphosphatase 1 OS=Rattus norvegicus	SI	8,00E-06	95,23
HKLEDGYPK	No ofrece resultados	NO		
GECMADSVLFFK	Hemopexin OS=Pongo abelii	SI	6,10E+01	66,66
GGDDLDPNYVLSSR	Creatine kinase M-type OS=Bos taurus	SI	1,40E-02	100
LSIEALNSLDGEFK	Creatine kinase M-type OS=Gallus gallus	SI	5,20E-01	85,71
GENCIAAGR	10-formyltetrahydrofolate dehydrogenase OS=Xenopus tropicalis	SI	1,40E+01	100
GFTLPPHNSR	Creatine kinase M-type OS=Sus scrofa	SI	8,90E+01	80
VLVTGAAGQIAYSLLYSIAK	Malate dehydrogenase, cytoplasmic OS=Gallus gallus	SI	2,00E-05	100
ATDFVVDKPGKFK	Isocitrate dehydrogenase [NADP], mitochondrial OS=Bos taurus	SI	9,40E+01	76,92

Tabla 8. Resultados obtenidos tras utilizar la matriz BLOSUM62.

Resultados con BLOSUM62	Descripción	Anotada correctamente	Evalue	%id
SKQLEDDLVALQK	Tropomyosin alpha-1 chain OS=Danio rerio	SI	9,20E+00	100
IQLVEEELDR	Tropomyosin OS=Branchiostoma belcheri	SI	2,73E+02	100
EQQEAIEHIDEVQNEIDR	Protein SET OS=Homo sapiens	SI	1,70E-02	100
AGFAGDDAPR	Actin (Fragment) OS=Acetabularia cliftonii	SI	1,40E+01	100
GAHQNIIPASTGAAK	Glyceraldehyde-3-phosphate dehydrogenase OS=Oncorhynchus mykiss	SI	3,40E-01	100
VPVADVSVVDLTCR	Glyceraldehyde-3-phosphate dehydrogenase OS=Oncorhynchus mykiss	SI	1,40E+00	92,85
ATDFVVDKPGKFK	Isocitrate dehydrogenase [NADP], mitochondrial (Fragment) OS=Sus scrofa	SI	1,06E+02	76,92
ATDFVVDKPGK	Isocitrate dehydrogenase [NADP], mitochondrial OS=Dictyostelium discoideum	SI	3,07E+02	81,81
DLPLAQGIKFE	Alpha-aminoadipic semialdehyde dehydrogenase OS=Homo sapiens	SI	1,70E+01	90,9
EGNGTVMGAELR	Myosin light chain 1, cardiac muscle OS=Gallus gallus	SI	6,00E+00	100
VAYNQIADIMR	Myosin light chain 1, skeletal muscle isoform OS=Mugil capito	SI	2,10E+01	100
AGIANLYGIAGSTNVTGDQVK	Fructose-1,6-bisphosphatase 1 OS=Sus scrofa	SI	5,00E-04	95,23
HKLEDGYPK	No ofrece resultados	NO		
GECMADSVLFFK	Hemopexin OS=Pongo abelii	SI	1,58E+02	66,66
GGDDLDPNYVLSSR	Creatine kinase M-type OS=Bos taurus	SI	2,20E-01	100
LSIEALNSLDGEFK	Creatine kinase M-type OS=Gallus gallus	SI	4,20E+00	85,71
GENCIAAGR	10-formyltetrahydrofolate dehydrogenase OS=Xenopus laevis	SI	5,60E+01	100
GFTLPPHNSR	Creatine kinase M-type OS=Sus scrofa	SI	2,35E+02	80
VLVTGAAGQIAYSLLYSIAK	Malate dehydrogenase, cytoplasmic OS=Gallus gallus	SI	2,00E-03	100
ATDFVVDKPGKFK	Isocitrate dehydrogenase [NADP], mitochondrial (Fragment) OS=Sus scrofa	SI	1,06E+02	76,92

Tabla 9. Resultados obtenidos tras utilizar la matriz BLOSUM50.

Resultados con BLOSUM50	Descripción	Anotada correctamente	Evalue	%id
SKQLEDDLVALQK	Tropomyosin alpha-1 chain OS=Danio rerio	SI	8,70E+00	100
IQLVEEELDR	Tropomyosin OS=Branchiostoma belcheri	SI	2,62E+02	100
EQQEAIEHIDEVQNEIDR	Protein SET OS=Homo sapiens	SI	2,00E-02	100
AGFAGDDAPR	Actin (Fragment) OS=Acetabularia cliftonii	SI	5,60E+01	100
GAHQNIIPASTGAAK	Glyceraldehyde-3-phosphate dehydrogenase 2 OS=Drosophila melanogaster	SI	3,20E+00	93,33
VPVADVSVVDLTCR	Glyceraldehyde-3-phosphate dehydrogenase OS=Oncorhynchus mykiss	SI	1,10E+01	92,85
ATDFVVDKPGKFK	Isocitrate dehydrogenase [NADP], mitochondrial (Fragment) OS=Sus scrofa	SI	1,21E+02	76,92
ATDFVVDKPGK	Probable proline racemase OS=Bos taurus	SI	5,08E+02	72,72
DLPLAQGIKFE	Alpha-aminoadipic semialdehyde dehydrogenase OS=Homo sapiens	SI	9,20E+01	90,9
EGNGTVMGAELR	Myosin light chain 3, skeletal muscle isoform OS=Mugil capito	SI	8,60E+00	100
VAYNQIADIMR	Myosin light chain 1, skeletal muscle isoform OS=Mugil capito	SI	1,68E+02	100
AGIANLYGIAGSTNVTGDQVK	Fructose-1,6-bisphosphatase 1 OS=Sus scrofa	SI	6,00E-04	95,23
HKLEDGYPK	No ofrece resultados	NO		
GECMADSVLFFK	Hemopexin OS=Pongo abelii	SI	4,27E+02	66,66
GGDDLDPNYVLSSR	Creatine kinase B-type OS=Oryctolagus cuniculus	SI	4,70E-01	100
LSIEALNSLDGEFK	Creatine kinase M-type OS=Gallus gallus	SI	3,00E+01	85,71
GENCIAAGR	10-formyltetrahydrofolate dehydrogenase OS=Xenopus laevis	SI	2,18E+02	100
GFTLPPHNSR	Creatine kinase B-type OS=Oryctolagus cuniculus	SI	5,50E+02	80
VLVTGAAGQIAYSLLYSIAK	Malate dehydrogenase, cytoplasmic OS=Gallus gallus	SI	4,60E-02	100
ATDFVVDKPGKFK	Isocitrate dehydrogenase [NADP], mitochondrial (Fragment) OS=Sus scrofa	SI	1,21E+02	76,92

Tabla 10. Resultados obtenidos tras utilizar la matriz BLOSUM45.

Resultados con BLOSUM45	Descripción	Anotada correctamente	Evalue	%id
SKQLEDDLVALQK	Tropomyosin alpha-1 chain OS=Danio rerio	SI	5,00E+00	100
IQLVEEELDR	Tropomyosin OS=Branchiostoma belcheri	SI	1,55E+02	100
EQQEAIEHIDEVQNEIDR	Protein SET OS=Rattus norvegicus	SI	8,00E-03	100
AGFAGDDAPR	Actin (Fragment) OS=Acetabularia cliftonii	SI	1,02E+02	100
GAHQNIIPASTGAAK	Glyceraldehyde-3-phosphate dehydrogenase OS=Pongo abelii	SI	1,80E+00	93,33
VPVADVSVVDLTCR	Glyceraldehyde-3-phosphate dehydrogenase, testis-specific OS=Macaca fascicularis	SI	1,20E+01	85,71
ATDFVVDKPGKFK	Isocitrate dehydrogenase [NADP], mitochondrial (Fragment) OS=Sus scrofa	SI	9,00E+01	76,92
ATDFVVDKPGK	No ofrece resultados	NO		
DLPLAQGIKFE	Alpha-aminoadipic semialdehyde dehydrogenase OS=Homo sapiens	SI	1,79E+02	90,9
EGNGTVMGAELR	Myosin light chain 3, skeletal muscle isoform OS=Mugil capito	SI	5,50E+00	100
VAYNQIADIMR	Myosin light chain 1, skeletal muscle isoform OS=Mugil capito	SI	3,66E+02	100
AGIANLYGIAGSTNVTGDQVK	Fructose-1,6-bisphosphatase 1 OS=Sus scrofa	SI	3,00E-04	95,23
HKLEDGYPK	No ofrece resultados	NO		
GECMADSVLFFK	Hemopexin OS=Pongo abelii	SI	6,79E+02	66,66
GGDDLDPNYVLSSR	Creatine kinase B-type OS=Oryctolagus cuniculus	SI	3,10E-01	100
LSIEALNSLDGEFK	Creatine kinase M-type OS=Gallus gallus	SI	8,00E+01	85,71
GENCIAAGR	10-formyltetrahydrofolate dehydrogenase OS=Xenopus tropicalis	SI	3,27E+02	100
GFTLPPHNSR	Creatine kinase B-type OS=Oryctolagus cuniculus	SI	4,30E+02	80
VLVTGAAGQIAYSLLYSIAK	Probable malate dehydrogenase 3 OS=Dictyostelium discoideum	SI	1,40E-01	70
ATDFVVDKPGKFK	Isocitrate dehydrogenase [NADP], mitochondrial (Fragment) OS=Sus scrofa	SI	9,00E+01	76,92

Se pudo observar que las identificaciones eran las mismas pero con distintos valores de E-value, Score y %id dependiendo de la matriz utilizada. En conjunto en todos los casos se pudieron observar identificaciones positivas a excepción de la secuencia HKLEDGYPK.El número de identificaciones correctas para BLOSUM90, 80, 62 y 50 fue de 19 de 20, y en BLOSUM45 se fuede 18 de 20 secuencias. Aun siendo identificaciones correctas, como se mencionó anteriormente, los valores de E-value, Score y %id difieren entre las matrices. El %id de aquellas que BlaStorP dio como correctas (intervalo verde) no bajó del 95% y el E-value máximo para estas secuencias fue de 2e-05. En los casos medianamente correctos (intervalo ambar) los resultados para %id fueron como mínimo del 95% y el mayor valor para el E-value de 9e-03.

Estos resultados mostraron que el método es sensible y específico, especialmente en los casos de identificaciones con valores límite (intervalo rojo), en los que la especificidad del programa a la hora de encontrar secuencias de la misma proteína con similitud a la secuencia diana en organismos homólogos, se mostró como una herramienta efectiva de identificación positiva a pesar de los altos valores de E-value.

Llevando los resultados anteriores a una gráfica donde se muestran los valores de E-value obtenidos para cada matriz, se pueden apreciar aun más las diferencias entre los usos de una matriz u otra (Figura 5). En aquellos casos en los que los E-value no son bajos como en la secuencia GENCIAAR (identificada como Creatina Quinasa), se obtienen siempre los valores más bajos utilizando BLOSUM90 e incluso en el caso de la secuencia HKLEDGYPK se llega a obtener un resultado, hecho que no ocurre con ninguna de las otras matrices de alineamiento. Otro ejemplo lo tenemos con la secuencia EGNGTVMAGELR, donde la matriz BLOSUM90 logra un valor E-value negativo mientras el resto de matrices tienen valores positivos.

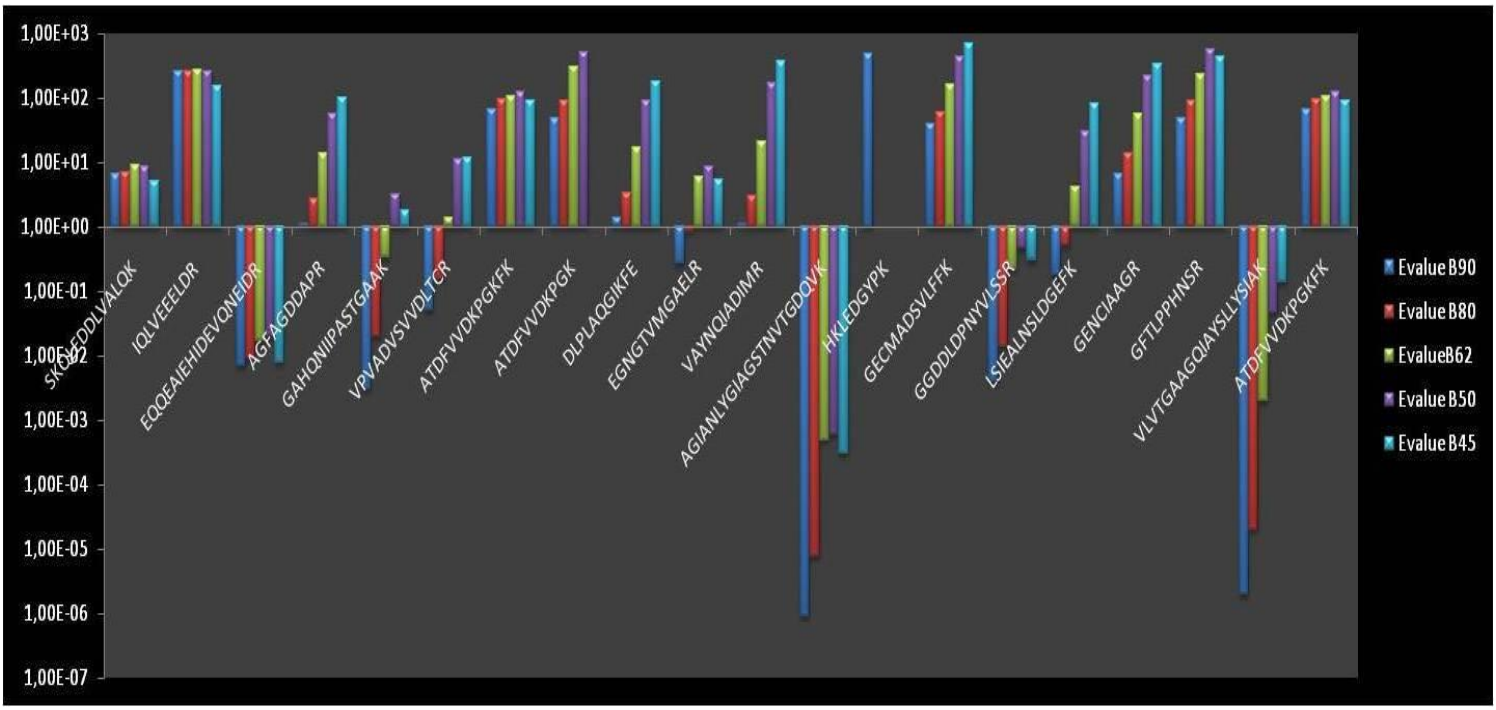


Figura 5. Resultados gráficos de los valores de E-value obtenidos cuando se utilizan las distintas matrices mencionadas anteriormente.

Para comprobar las diferencias, como se ha visto anteriormente (véase Tablas 6-10), se realizó un seguimiento del %id en cada caso con el fin de confirmar los parámetros introducidos por defecto (E-value dinámico), resultando una serie de perfiles que pueden observarse en la Figura 6. Llama la atención su similitud pero también el mencionado resultado con la secuencia HKLEDGYPK con una sola matriz (BLOSUM90), y que en otros casos incluso empeore (Secuencia ATDFVVDKPGKFK con BLOSUM45).

Por todos los motivos expuestos anteriormente, se fijó la matriz BLOSUM90 como matriz por defecto para realizar las búsquedas de fragmentos peptídicos cortos.

Por último, parece existir una tendencia a una mejor identificación cuanto mayor es la longitud de la secuencia diana. Así, si tenemos en cuenta la longitud de las secuencias que menor E-value ofrecen y hacemos un promedio nos da un resultado de 17 aminoácidos, una longitud medianamente larga si tenemos en consideración que la mayoría de las secuencias que se obtienen tras la secuenciación de novo de proteínas sometidas a digestión por tripsina (con cortes en K y R), suelen tener entre 10 y 15 aminoácidos de longitud. El Programa se encuentra por lo tanto en los límites superiores de identificación según la longitud de las secuencias.

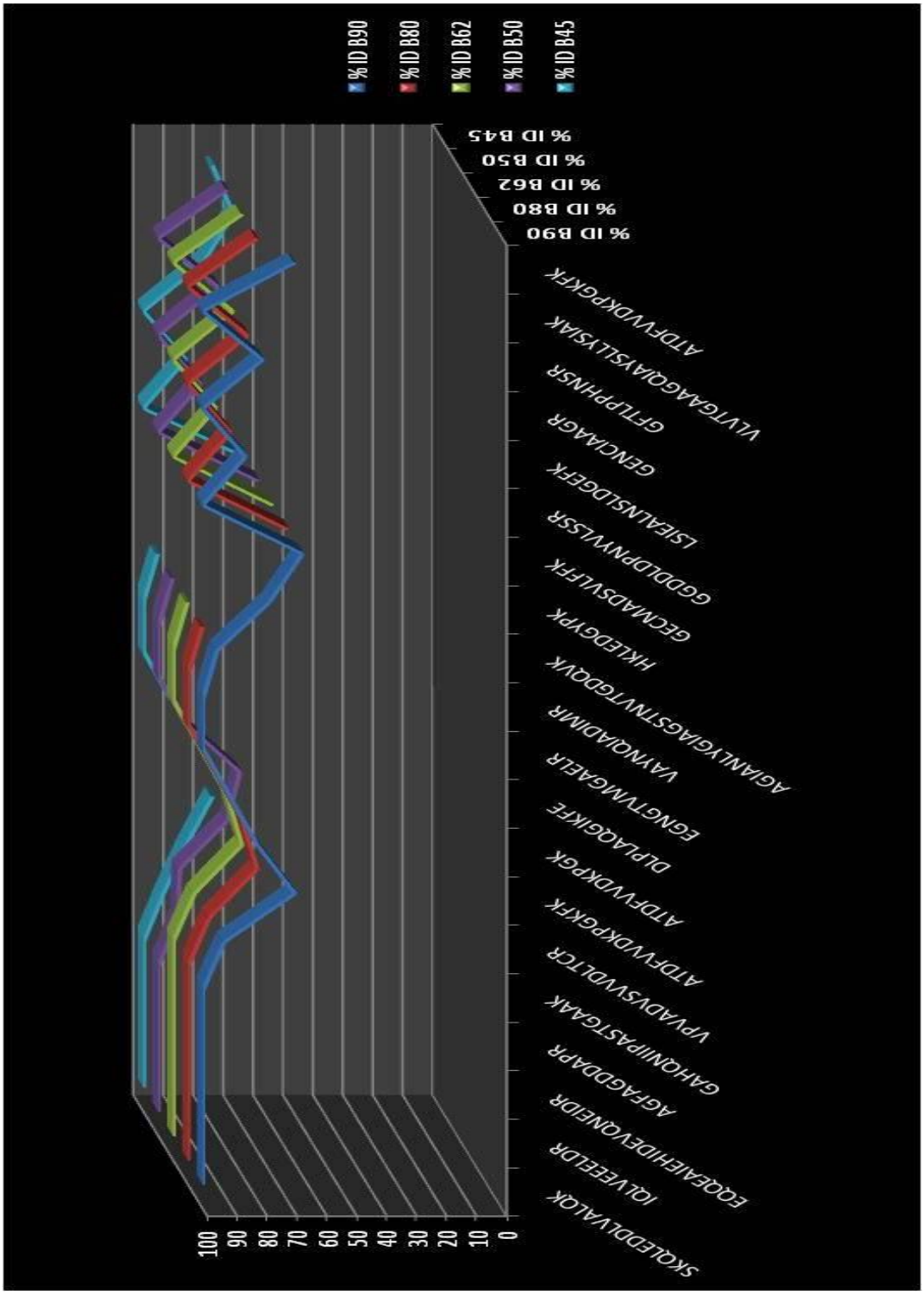


Figura 6. Resultados gráficos de los valores de %id obtenidos cuando se utilizan las distintas matrices

III. PRUEBA DEL MÉTODO

Para comprobar que los resultados obtenidos con los parámetros anteriores son los adecuados, se escogió un nuevo conjunto independiente de datos (Tabla 11), los cuales fueron sometidos a la búsqueda utilizando todos los parámetros definidos por defecto tras la etapa de Entrenamiento del Método.

Tabla 11. Conjunto de secuencias e identificaciones para la comprobación del método

Secuencia	Descripción
GDVVPKDVNSAIAAIK	Tubulin, alpha 8 like 3 [Danio rerio]
IAVGSDADLVIWDTDSIR	collapsin response mediator protein 4/Dihidropirimidinase [Danio rerio]
MVDVIVTTAGGIEEDLIK	Deoxyhypusine synthase [Danio rerio]
NPGLVLDLVEDIR	Dhps protein [Danio rerio]
WLNEGDALVAGGVSKTPSYLSCK	betaine homocysteine methyltransferase
VFDKEGNGTVMGAELR	cardiac mysosin light chain 1 [Danio rerio]
SQGGEPTYNVSVGK	Profilin-2 [Anoplopoma fimbria]
LGEEFDETTADDR	Fatty acid binding protein
NYNMDFYVGKEFEEDLSGVDDR	PREDICTED: (Cellular retinol-binding protein) (CRBP) [Danio rerio]
EFEEDLSGVDDRK	PREDICTED: (Cellular retinol-binding protein) (CRBP) [Danio rerio]

Tabla 12. Resultados obtenidos utilizando todos los parámetros de BlaStorP por defecto

Resultados con BLOSUM90	Descripción	Identificada correctamente	Evalue	%id
GDVVPKDVNSAIAAIK	Tubulin alpha chain OS=Oncorhynchus keta	SI	4,00E-04	100
IAVGSDADLVIWDTDSIR	Dihydropyrimidinase-related protein 2 OS=Gallus gallus	SI	7,00E-04	83,33
MVDVIVTTAGGIEEDLIK	Deoxyhypusine synthase OS=Nicotiana tabacum	SI	6,00E-05	94,44
NPGLVLDLVEDIR	Deoxyhypusine synthase OS=Homo sapiens	SI	3,30E-01	84,61
WLNEGDALVAGGVSKTPSYLSCK	Betaine--homocysteine S-methyltransferase 1 OS=Danio rerio	SI	7,00E-08	95,23
VFDKEGNGTVMGAELR	Myosin light chain 1, skeletal muscle isoform OS=Oryctolagus cuniculus	SI	1,10E-02	100
SQGGEPTYNVSVGK	Profilin-2 OS=Pongo abelii	SI	9,70E-02	85,71
LGEEFDETTADDR	Fatty acid-binding protein 9 OS=Homo sapiens	SI	5,20E-01	92,3
NYNMDFYVGKEFEEDLSGVDDR	Retinol-binding protein 1 OS=Homo sapiens	SI	5,00E-04	81,81
EFEEDLSGVDDRK	Retinol-binding protein 1 OS=Homo sapiens	SI	1,20E+01	84,61

Tras estos resultados, sólo quedaba observar el comportamiento de BlaStorP. Los valores de E-value quedaron todos, excepto un caso, en valores negativos y ninguna de las %id bajó del 80% (Figuras 7 y 8).

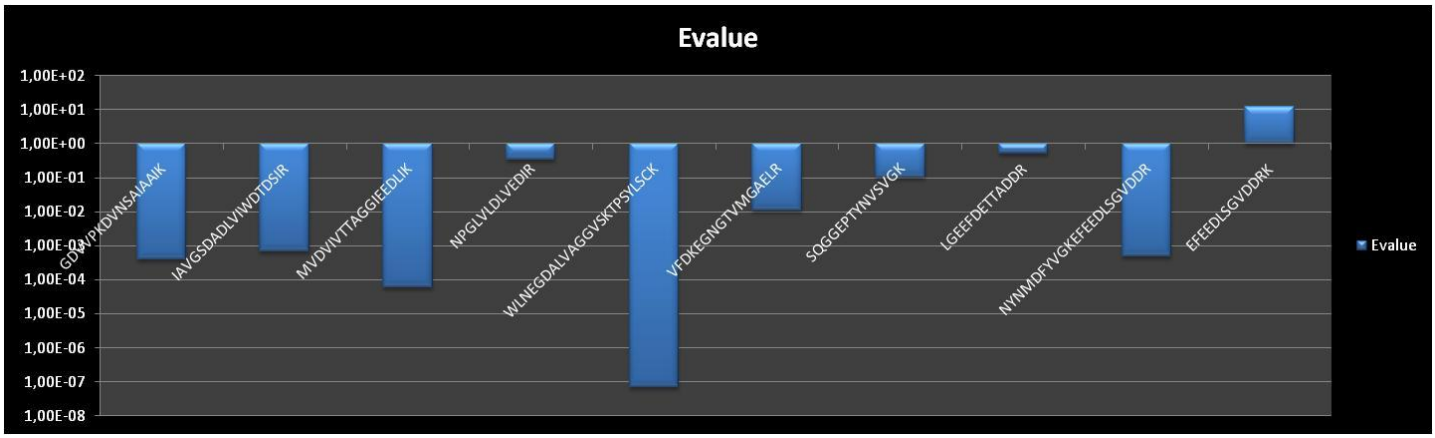


Figura 7. Resultados gráficos de los valores de E-value durante la prueba del método

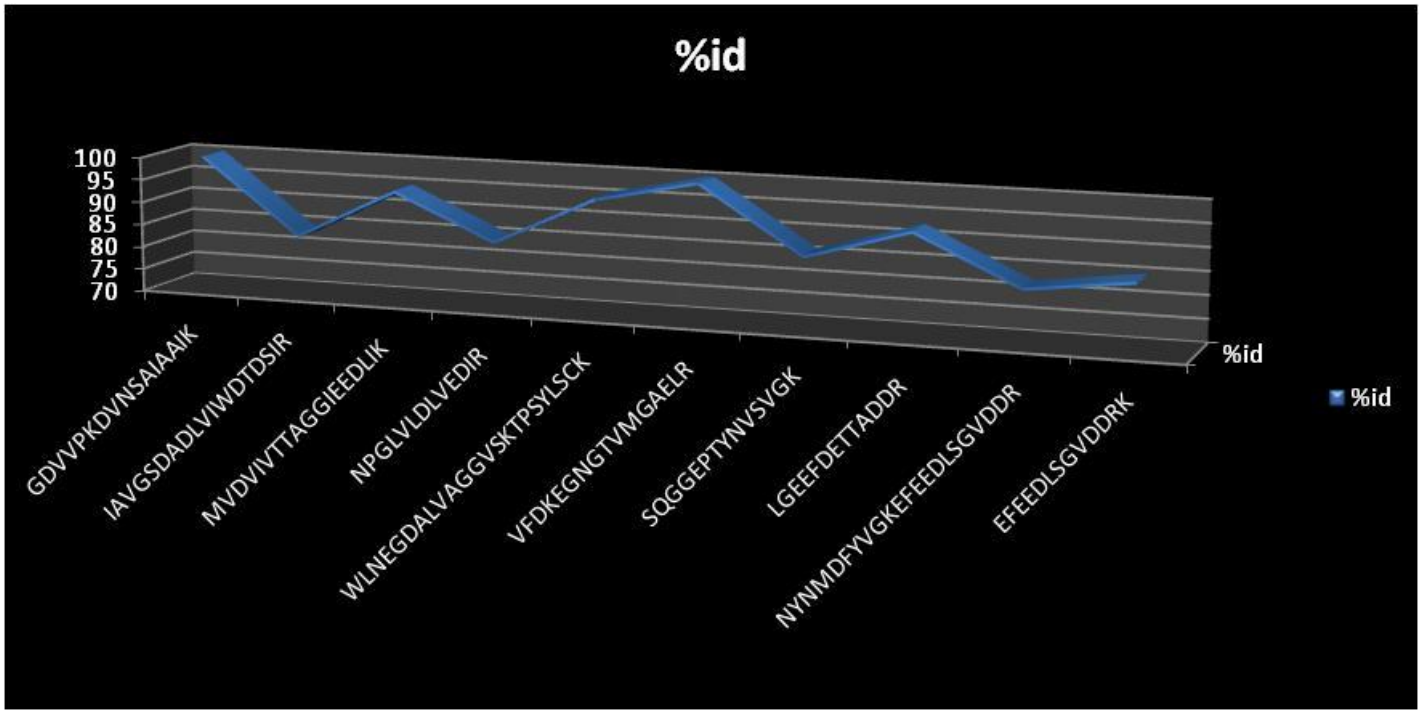


Figura 8. Resultados gráficos de los valores de %id tras la prueba del método

También se cuantificó la contribución de los distintos tramos verde/ambar/rojo en los resultados (Figura 9).

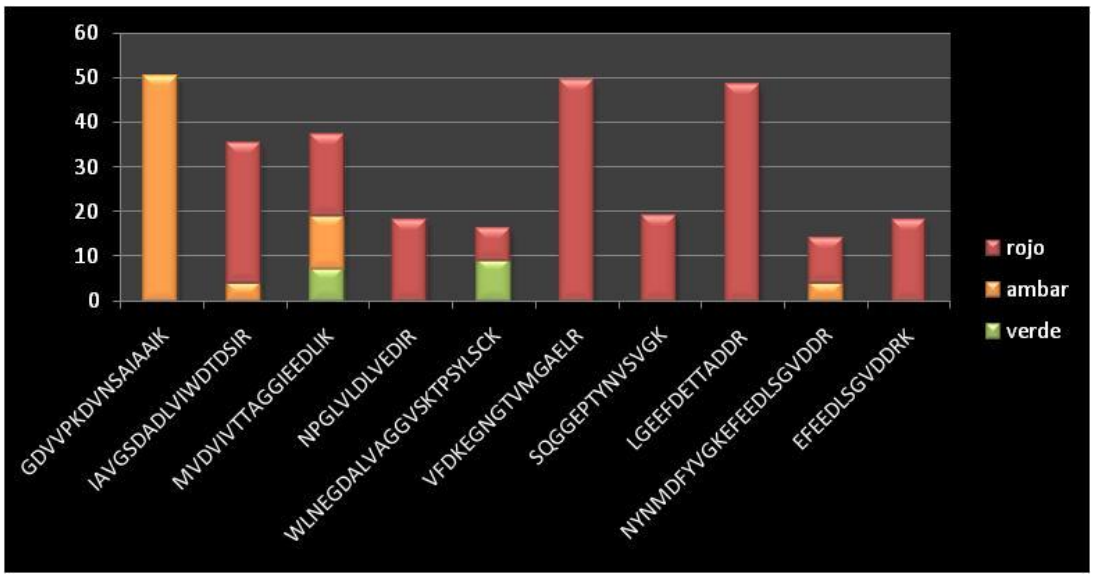


Figura 9. Contribuciones de cada tramo de color a cada resultado tras la prueba del método.

El programa, a pesar de encontrarse con secuencias procedentes de organismos no secuenciados, es capaz de encontrar una identificación correcta con una alta sensibilidad y especificidad y en las ocasiones en las que se encuentra en niveles no aceptables de identificación, llega a encontrar similitud en multitud de secuencias de la misma proteína en distintos organismos manteniendo la especificidad en la identificación como ocurre en el caso de las secuencias NPGLVLDLVEDIR y EFEEDLSGVDDRK. A pesar de que la contribución de resultados en la región rojo pueda parecer importante (Figura 9), en la mayoría de los casos, en este tipo de organismos la identificación basada en especificidad también puede ser muy útil (de hecho se lleva a cabo y después se confirman los resultados mediante Western blot) y puede dar lugar a identificaciones positivas en aquellos casos que no tienen resultados en otras herramientas bioinformáticas.

Por otra parte, las anotaciones que se muestran en el informe que da lugar BlaStorP (Tabla 13) se encuentran organizadas en función del número de veces que aparecen en los resultados de los tramos verde/ambar/rojo, lo que nos puede dar una información más precisa de la función que lleva a cabo la proteína identificada. Así, cuantas más veces aparezca una determinada Keyword, Gene Ontologie o Anotación Uniprot, más certeza tiene el usuario de la función que lleva a cabo dicha proteína.

Además, los enlaces a las distintas bases de datos son de mucha utilidad para el usuario, ya que tener estas referencias cruzadas a otras bases de datos es importante para indagar un poco más en aquellos aspectos en los que se encuentre más interesado (Tabla 13).

Tabla 13. Anotaciones de la secuencia MVDVIVTTAGGIEEDLIK, Deoxyhypusine synthase, organizadas según su frecuencia de aparición en los resultados de la identificación, lo que nos indica las funciones de dicha proteína (es una transferasa que consume NAD y participa en la biosíntesis de hipusina) y nos permite ir a las distintas bases de datos que contienen dicha información.

kw: Transferase(7), NAD(7), Hypusine biosynthesis(7), Phosphoprotein(3), Alternative splicing(1), Polymorphism(1), 3D-structure(1), Direct protein sequencing(1)
go: GO:0050983(7), GO:0008612(7), GO:0034038(7), GO:0005515(1), GO:0005829(1), GO:0006412(1), GO:0008284(1)
ipro: IPR002773(7)

IV. CONCLUSIONES

Se ha desarrollado una nueva herramienta bioinformática (BlaStorP) que simplifica las búsquedas de identificaciones por similitud realizadas con secuencias peptídicas cortas procedentes de secuenciación “de novo”, derivadas de experimentos en el emergente campo de la Proteómica.

Dicha herramienta ha sido probada extensamente con multitud de secuencias durante el período de entrenamiento del método y durante el período de prueba del método, lo que asegura su eficacia a la hora de identificar las secuencias peptídicas mediante similitud.

La utilidad del programa y su mayor baza es la simplicidad de uso así como una organización mejorada cuando se presentan los resultados, lo que da al investigador una visión global de la/s proteína/s ante las que se encuentra.

De esta forma, el investigador puede ir de una a otra base de datos gracias a los enlaces incluidos dentro del mismo informe que genera el Programa y a la organización basada en colores que le permite de un solo vistazo saber la calidad de las identificaciones ante las que se halla .

Otra de las ventajas añadidas gracias a su vertiente local es la posibilidad de personalizar la aplicación con multitud de posibles cambios, ya sean dentro del mismo guión, cambios de bases de datos, etc.

El futuro de éste Programa podría pasar por ampliar el contenido de la información con la inclusión de nuevos módulos siempre evitando la saturación por información excesiva.

Otro de los puntos a abordar para la mejora de las identificaciones podría ser la restricción por organismos en la base de datos que utilice el usuario.

Podemos resumir estas conclusiones en las siguientes Conclusiones Generales:

1. Es posible utilizar un blast con parámetros a medida, para encontrar similitud a secuencias peptídicas cortas procedentes de experiementos de proteómica.
2. Los mejores resultados se obtienen utilizando un %id=66 y un E-value "dinámico" durante la ejecución de Blast.
3. La matriz de puntuación que mejores resultados da para este tipo de secuencias es BLOSUM90.
4. Las secuencias más largas (de 15 a 25 aminoácidos) son más fáciles de identificar.
5. La separación de los resultados según su significación ayuda a la identificación de los péptidos.
6. En organismos no secuenciados se puede llegar a una identificación positiva, no sólo en base a la sensibilidad y especificidad del programa, sino también en base a la especificidad, solamente pudiendo confirmar estos resultados experimentalmente (Western blot).
7. El conjunto de anotaciones de Keywords, Gene Ontologie e Interpro y sus respectivos enlaces en el informe final facilitan la identificación de la función de la proteína y la consulta cruzada de distintas bases de datos.

ANEXO 1

Instalación de los materiales necesarios en MS Windows

Para permitir la reproducibilidad de instalación y uso de la herramienta desarrollada en este proyecto, se incluye este anexo donde se describen los pasos para su instalación en un sistema operativo MS Windows.

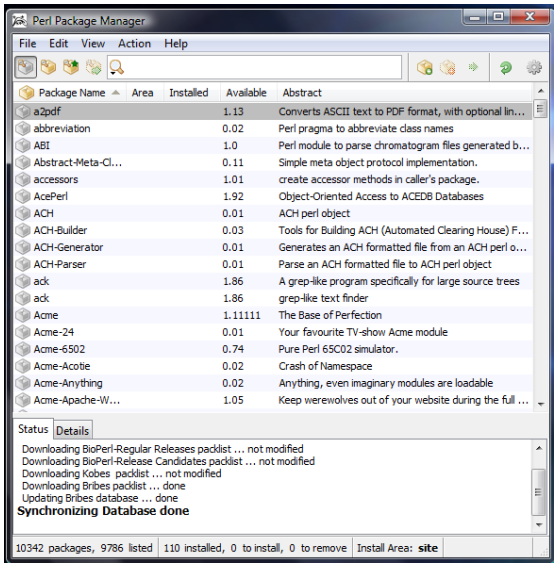
I. INTÉRPRETE PERL

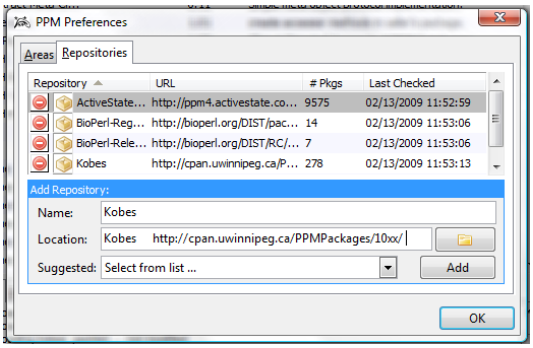
Para el desarrollo de este Proyecto de Fin de Máster (PFM en adelante) se tomó la distribución de ActiveState (http://www.activestate.com/) denominada ActivePerl, más concretamente ActivePerl v.5.10.0 Build 1003. El archivo ejecutable tiene un tamaño de 17 MB e instala todas las herramientas necesarias para que el intérprete Perl funcione así como la herramienta Perl Package Manager, para instalación/desinstalación de nuevos paquetes y módulos para el intérprete.

Esta actuación es imprescindible para el desarrollo del PFM si se quieren reproducir los resultados obtenidos en el, ya que el sistema operativo utilizado para el desarrollo del mismo ha sido MS Windows Vista Home Premium Edition, el cual no posee integrado un intérprete de Perl como en los casos de las distribuciones Linux (Ubuntu, Fedora, Mandriva, etc.) o sistemas Unix, aunque también existen versiones de ActivePerl para éstos casos (http://www.activestate.com/activeperl/downloads/).

II. BIOPERL

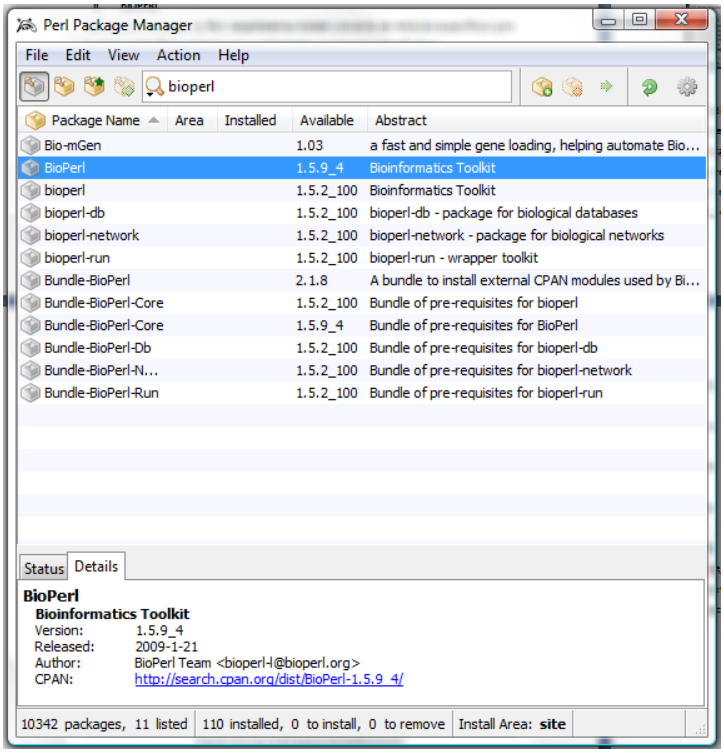
Para su instalación en MS Windows debemos abrir el gestor de paquetes de ActivePerl (Perl Package Manager) e ir al submenú Edit:



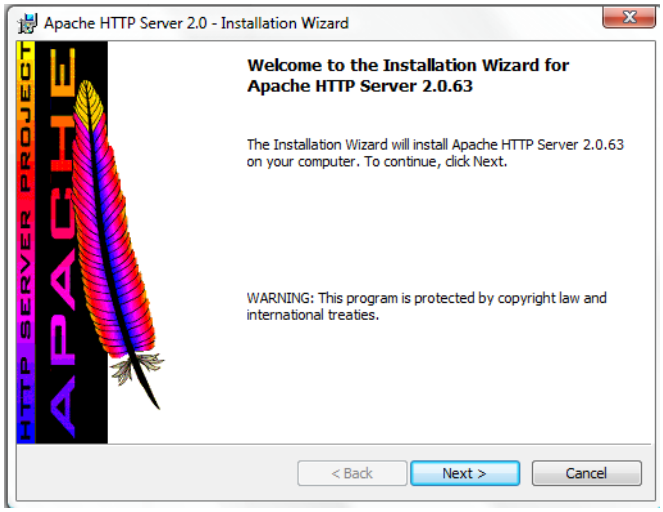


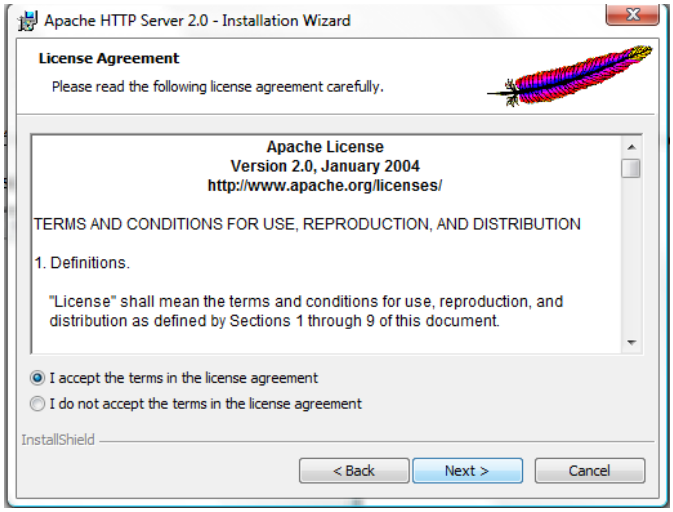
Una vez en éste menú debemos ir a Preferences y hacer click en Repositories tab y añadir los repositorios indicados anteriormente. Tras la adición de los repositorios debemos volver a la vista principal donde se encuentran todos los paquetes instalados (Select View: All Packages).Teclearemos en la opción de búsqueda bioperl y haremos click con el botón derecho sobre la última versión de BioPerl disponible eligiendo la misma para instalarla. Finalmente haremos click en la flecha verde situada en la esquina superior derecha (Run marked actions) para completar la instalación de los paquetes BioPerl.

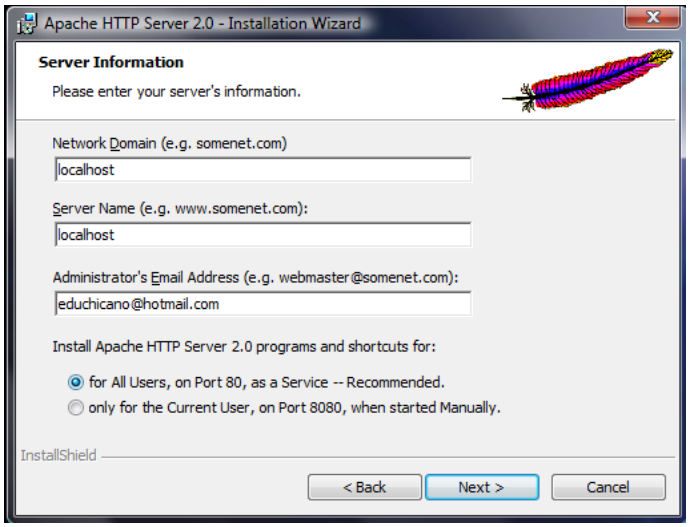
III.

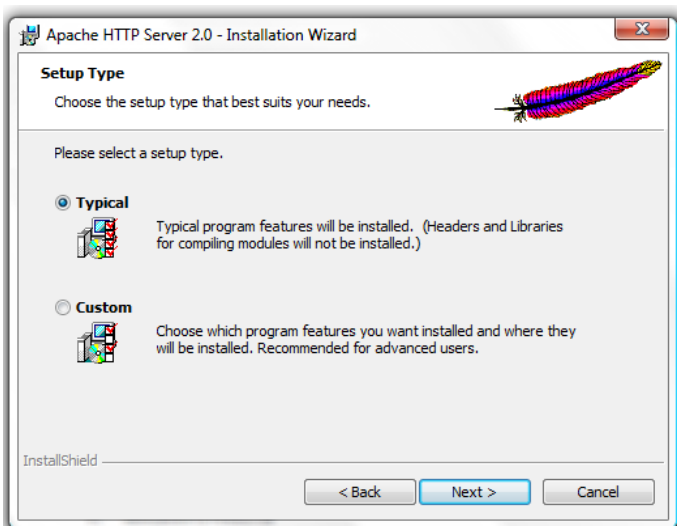


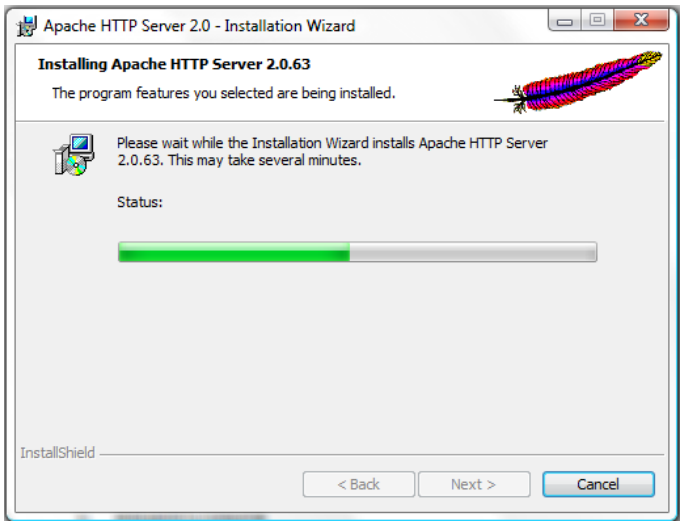
Su instalación es muy sencilla a través del ejecutable en formato .msi que nos descargamos. Vamos a describirlo de forma gráfica:

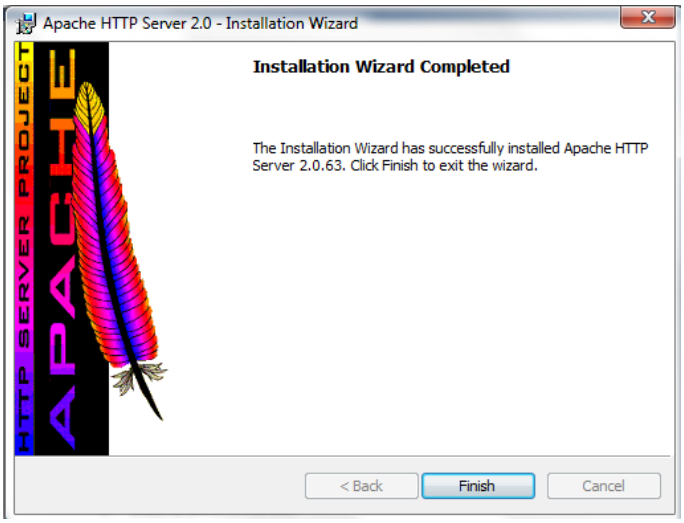












Una vez haya finalizado de instalar con los parámetros tal y como los hemos introducido en la ventana de opciones (Network: localhost, Server Name: localhost, Administrator's Address: educhicano@gmail.com), probaremos si realmente se ha instalado correctamente introduciendo la dirección 127.0.0.1 en nuestro navegador y nos dará como resultado la siguiente pantalla:

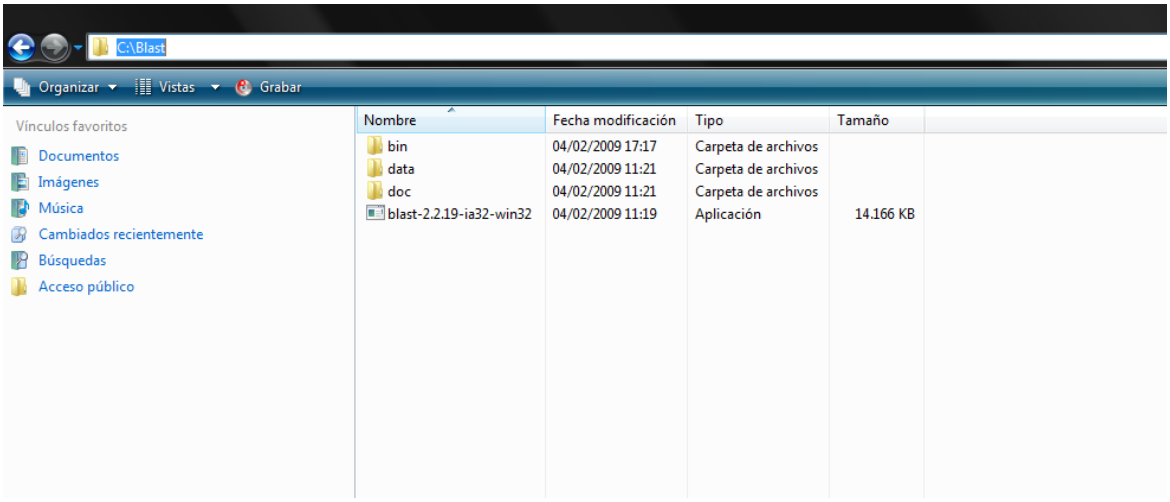
Nuestro script será depositado en MS Windows en la siguiente ruta C:\Program Files\Apache Group\Apache2\cgi-bin y el formulario en la siguiente: C:\Program Files\Apache Group\Apache2\htdocs.

IV. BASES DE DATOS UTILIZADAS

Como se dijo anteriormente en el apartado de Materiales y Métodos para instalar el el paquete de software BLAST debemos entrar en el servidor del NCBI en la siguiente url: http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=Download:

Tenemos que escoger en el caso de Windows la distribución win32-ia32 download que nos descargará en nuestro ordenador un archivo ejecutable de 13,2 MB de tamaño.

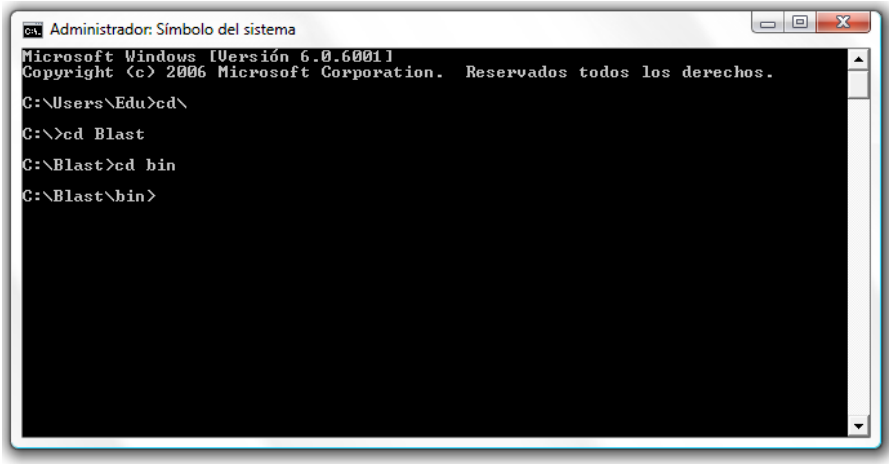
Este archivo ejecutable lo situaremos en una nueva carpeta denominada Blast en el archivo raíz de nuestro PC, de forma que quede dentro de la misma. La ruta sería por lo tanto C:\Blast. Una vez tengamos nuestro archivo ejecutable en esta carpeta, lo ejecutaremos y nos instalará automáticamente todo el software BLAST que necesitaremos para trabajar localmente y nos aparecerán 3 carpetas nuevas: bin, data y doc.



V. HERRAMIENTA FORMATDB

Para llevar a cabo el formateo de la base de datos uniprot_sprot.fasta debemos copiar el fichero FASTA de la base de datos a C:\Blast\bin. Una vez realizado este paso tenemos que iniciar la herramienta de MS Windows Símbolo del sistema, que se encuentra en Inicio/Accesorios.

Una vez estamos dentro de esta herramienta nos tendremos que situar en la ruta C:\Blast\bin:



Introduciremos la siguiente línea de comandos: **formatdb –i uniprot_sprot.fasta –p T – o T**. Esta línea de comandos le dice a Formatdb que ejecute los siguientes pasos sobre el fichero uniprot_sprot.fasta:

- -i : señala el fichero a tratar, por eso detrás de –i viene el nombre del fichero
- -p T: -p indica el tipo de fichero. –p tiene 2 opciones: T (la que hemos elegido) que indica que el fichero contiene proteínas y F, que indica que el fichero contiene nucleótidos.

- -o T: indica una de las opciones de parsing sobre el fichero. En este caso T indica a Formatdb que cree ejecute un parsing sobre las Id de las secuencias y cree índices. La otra opción sería F, que sería la opción contraria a la escogida: no realizar parsing y no crear índices.

En este PFM no son necesarias más instrucciones para Formatdb aunque si queremos más opciones de formateo podemos consultarlas en la carpeta C:\Blast\doc, donde viene toda la documentación para Formatdb y otras herramientas BLAST.

Una vez ha terminado de ejecutarse Formatdb, se crearán 5 archivos nuevos, todos con el mismo nombre (uniprot_sprot) pero distintas extensiones y tamaños: .phr (63,25 MB), .pin (3,11 MB), .psd (38,44 MB), .psi (926,7 KB) y .psq (140,66 MB).

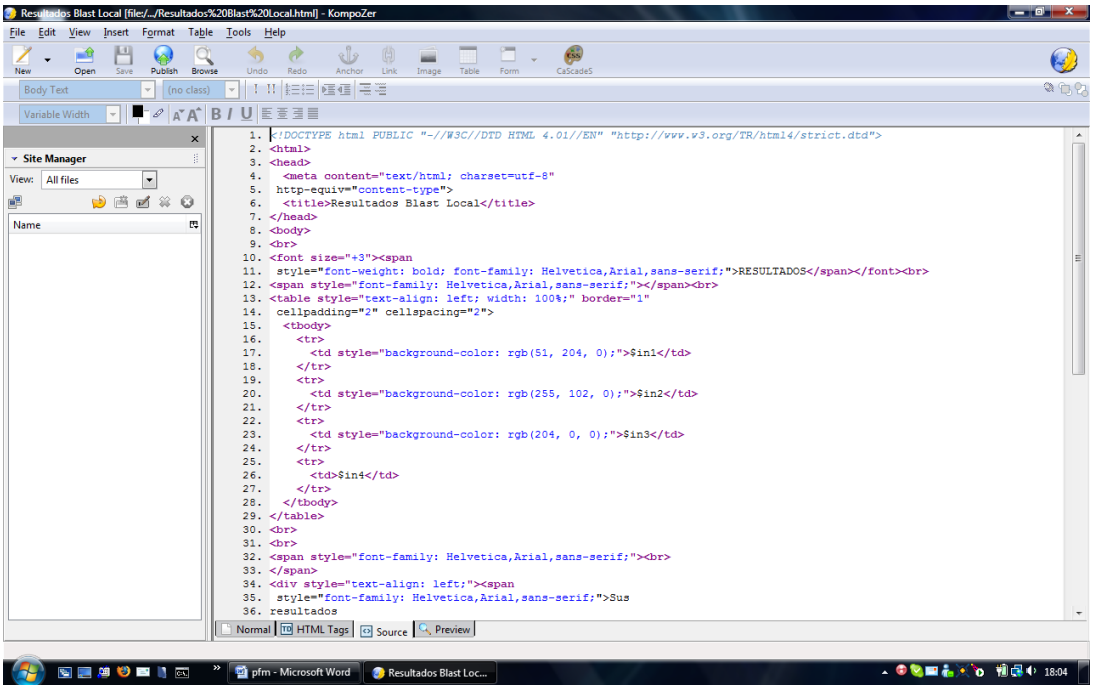
Estos archivos junto con el archivo FASTA de la base de datos conformarán la base de datos donde se buscarán las secuencias por parte de la herramienta Blastall.

VI. KOMPOZER

KompoZer es un editor HTML WYSIWYG (What You See Is What You Get (en inglés, "lo que ves es lo que obtienes")) basado en Nvu y muy similar al Dreamweaver de Adobe con la diferencia de que es un proyecto Open Source de licencia GNU/GPL 2.0.

Este editor se puede descargar libremente desde la siguiente dirección: http://kompozer.net/download.php donde escogemos la opción Win32 binary. Una vez instalado podremos diseñar el formulario y la página de resultados a nuestro antojo.

Abajo se muestra una captura del Kompozer conteniendo el código fuente de la página de resultados que se muestra tras la búsqueda con el script:



ANEXO 2

Instalación de los materiales necesarios en Linux (distribución Ubuntu 9.04 Jaunty Jackalope)

I. BIOPERL

Para la instalación de BioPerl en Linux se ha de estar en posesión de una versión de Perl superior a 5.6.1. Esto no es problema en la distribución Ubuntu 9.04, la cual trae la versión 5.10.0, pero debe tenerse en cuenta para distribuciones anteriores.

El primer paso para instalar BioPerl será actualizar CPAN para lo cual se tecleará en el terminal:

edu@ubuntu:~$ perl -MCPAN -e shell

cpan[1]>install Bundle::CPAN

Tras este paso, en la shell de cpan instalar o actualizar Module::Build como sigue:

cpan[1]>

cpan[1]>o conf prefer_installer MB

cpan[1]>o conf commit

Después de finalizar con la instalación del módulo buscar la versión más reciente de BioPerl en CPAN:

cpan[1]>d /bioperl/

E instalar la versión más reciente, en éste caso /CJFIELDS/BioPerl-1.6.0.tar.gz:

cpan[1]>install C/CJ/CJFIELDS/BioPerl-1.6.0.tar.gz

y seguir las instrucciones que irán apareciendo en el terminal. Dentro de estas opciones aparecerá una que da como opción testear online lo que se va instalando. Es muy recomendable hacer estos test para asegurarse de la correcta instalación de BioPerl.

II. APACHE

Si se desea instalar una nueva versión de Apache (Ubuntu lo trae por defecto instalado) se tecleará lo siguiente en el terminal:

edu@ubuntu:~$ sudo apt-get install apache2

Para comprobar que realmente se ha realizado una correcta instalación, se escribirá la dirección 127.0.0.1 en cualquier navegador de que se disponga (Firefox, Chrome, Opera…) y dará la siguiente respuesta: It Works!

III. BLAST

Para la instalación del paquete Blast se descargará de la página http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=Download el paquete correspondiente a Linux: Linux-ia32 (plataforma 32bits) de 30,6 Mb.

Una vez descargado, extraer en el lugar deseado y utilizar apt-get para instalar todas las herramientas necesarias:

edu@ubuntu:~$ sudo apt-get install blast2

IV. .DAT A .IDX

Ejecutar en el terminal el script reseñado anteriormente en el apartado Materiales y Métodos introduciendo en el terminal un comando para ejecutar un script Perl: perl script.pl base de datos swissprot.

ANEXO 3

Instalación de los materiales necesarios en Mac OS

La instalación de los posibles materiales necesarios en MacOS es exactamente igual a la descrita anteriormente para Ubuntu ya que la construcción de MacOS está también basada en los sistemas UNIX. Sustituir “apt-get” por “port”.