COLEGIO DE POSTGRADUADOS

INSTITUCIÓN DE ENSEÑANZA E INVESTIGACIÓN
EN CIENCIAS AGRÍCOLAS

CAMPUS MONTECILLO

POSTGRADO DE SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA
COMPUTO APLICADO

Base de datos para factores de transcripción de *Glycine max*, *Triticum aestivum* y *Zea maiz*

Obed Ramírez Sánchez

T E S I S

PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL
GRADO DE:

MAESTRO EN CIENCIAS

MONTECILLO,TEXCOCO, EDO. DE MÉXICO
2012

La presente tesis titulada: **Base de datos para factores de transcripción de *Glycine max*, *Triticum aestivum* y *Zea maiz***, realizada por el alumno: **Obed Ramírez Sánchez**, bajo la dirección del Consejo Particular indicado ha sido aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

MAESTRO EN CIENCIAS

SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA
COMPUTO APLICADO

CONSEJO PARTICULAR

CONSEJERO

Pérez Rdz.

Dr. Paulino Pérez Rodríguez

ASESOR

Dr. Juan Ricardo Bauer Mengelberg

ASESOR

Dr. Juan Andrés Burgueño Ferreira

Base de datos para factores de transcripción de *Glycine max*, *Triticum aestivum* y *Zea maiz*

Ramírez Sánchez Obed

Colegio de Postgraduados, 2012

Los factores de transcripción (FTs) son proteínas que incrementan o disminuyen la tasa transcripcional de uno o varios genes. Los FTs desempeñan un papel fundamental en el control de casi todos los procesos biológicos: crecimiento, metabolismo, respuesta a factores ambientales, etc. En plantas cultivadas, son de particular importancia aquellos relacionados con la respuesta al estrés (sequía, salinidad, bajas temperaturas, plagas, enfermedades, etc.). La identificación de FTs, su posterior anotación y la construcción de bases de datos públicas constituye un importante recurso para el estudio de los procesos que controlan la expresión génica. Se construyó la base de datos FT-MTS que contiene información de 13,975 genes identificados como FTs en los genomas de *Glycine max*, *Triticum aestivum* y *Zea maiz*. Para ampliar la información sobre dichos FTs, se hizo una completa anotación que incluye: descripción de cada familia, alineamientos múltiples, dominios de unión, arquitectura de dominios, estructura 3D de proteínas homologas y grupos de ortólogos. Además, se incluyeron referencias a bases de datos externas que contienen descripción de dominios (Pfam), descripción de estructuras 3D (Protein Data Bank) y literatura relacionada (PubMED). Finalmente, se construyó una interfaz web (http://174.123.176.26:3000) que permite a los usuarios realizar búsquedas en la base de datos, descargar las secuencias de cada FT, descargar los alineamientos múltiples de cada dominio y comparar sus secuencias con las de los FTs vía BLAST o HMMER.

Palabras clave: regulación transcripcional, anotación funcional, bioinformática.

Transcription Factor Database of *Glycine max*, *Triticum aestivum* and *Zea maiz*

Ramírez Sánchez Obed

Colegio de Postgraduados, 2012

Transcription factors (TFs) are proteins that enhance or decrease the transcriptional rate of one of several genes. TFs play key roles regarding the control of almost all biologic processes: growth, metabolism, response to environmental facors, etc. In crop plants, TFs related to response to stress are particularly important (drought, salinity, low temperatures, plagues, plant diseases, etc.). Identification of TFs, their subsequent annotation and the construction of public databases constitute a valuable resource to study the processes that control genetic expression. A database named FT-MTS was built containing information about 13,975 genes identified as TFs in the *Glycine max*, *Triticum aestivum* y *Zea maiz* genomes. In order to broaden the information about those TFs, other data was recorded: description of every family, multiple sequence alignments, binding domains, domain architectures, 3D structure of homologous proteins and ortholog clusters. References to external databases containing domain descriptions (Pfam), 3D structure description (Protein Data Bank) and related literature (PubMED) were also included. A web interface (http://174.123.176.26:3000) was built, so that users may search the data base, download the sequences for each TF and multiple sequence alignments of each domain, and compare their sequences with the TFs using BLAST or HMMER.

Key words: Transcriptional regulation, functional annotation, bioinformatics.

El presente trabajo fue hecho en colaboración
con el Dr. Diego Mauricio Riaño Pachón,
Universidad de los Andes, Colombia. Y se deriva
del proyecto Plant Transcription Factor Database
(http://plntfdb.bio.uni-potsdam.de).

Agradecimientos

Al Colegio de Postgraduados, por brindarme el apoyo y las herramientas para continuar mi formación académica.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el financiamiento brindado durante la realización de mis estudios de Maestría.

A los integrantes de mi consejo Particular:

Dr. Paulino Pérez Rodríguez, por su excepcional dirección, dedicación y amistad, que culminaron con la realización exitosa de este trabajo.

Dr. Juan Ricardo Bauer Mengelberg, por haber revisado este escrito y por sus inigualables clases y asesorías, que contribuyeron ampliamente a mi formación profesional.

Dr. Juan Andrés Burgueño Ferreira, por su valiosa asesoría y revisión de este trabajo.

Al Dr. Diego Mauricio Riaño Pachón, por haberme brindado la oportunidad de participar en el proyecto PlnTFDB, donde descubrí mi gusto por la Bioinformática.

Dedicatoria

A mi **Abuelita** Mere, por su comprensión y consejos.

A mis **padres**: Moisés y Rosalba, que me inculcaron con su ejemplo el valor del trabajo y la perseverancia.

A mis **hermanos**: Dalinda y Moisés René, que siguen luchando por realizar sus sueños.

A **Peq**, por estar conmigo durante este viaje inolvidable que comenzó en Chapingo.

A mis **compañeros de maestría**: Areli, Edna, Evelin, Hilda, Judith, Viridiana, Jorgue y Manuel, por brindarme su amistad sincera y su apoyo moral durante la escritura de este trabajo.

-La tierra es redonda como una naranja [les reveló José Arcadio Buendía].

Úrsula perdió la paciencia. «Si has de volverte loco, vuélvete tú solo -gritó-. Pero no trates de inculcar a los niños tus ideas de gitano.»

Cien Años de Soledad. Gabriel García Márquez

-Quiere decir -sonrió el coronel Aureliano Buendía cuando terminó la lectura- que sólo estamos luchando por el poder.

-Son reformas tácticas -replicó uno de los delegados-. Por ahora, lo esencial es ensanchar la base popular de la guerra. Después veremos.

...

Sin dejar de sonreír, [el coronel Aureliano Buendía] tomó los pliegos que le entregaron los delegados y se dispuso a firmar.

Cien Años de Soledad. Gabriel García Márquez

Índice

Índice de figuras

Índice de Tablas

Capítulo 1

Introducción

Los factores de transcripción (FTs) son proteínas que aumentan o disminuyen los niveles de expresión de uno o varios genes al unirse a secuencias específicas de ADN, conocidas como elementos-cis (Latchman, 1997, Yamaguchi-Shinozakia y Shinozaki, 2005). Los FTs tienen un papel clave dentro una compleja red de señales fisiológicas y bioquímicas, que van desde la percepción de estrés a nivel celular hasta la expresión de los genes que se activan como respuesta (Riechmann *et al.*, 2000, Tran *et al.*, 2007). Uno de los objetivos principales de la investigación actual en biotecnología es diseñar plantas tolerantes al estrés tanto biótico como abiótico (plagas, enfermedades, sequía, salinidad, etc.) a través de la manipulación de los genes que codifican a los FTs implicados en el proceso de respuesta (Hussain *et al.*, 2011). Por ejemplo, se ha conseguido mediante ingeniería genética controlar la expresión del gen que codifica al factor de transcripción DREB/CBG, perteneciente a la familia AP2, en varios cultivos (tabaco, trigo, arroz y cacahuate) confiriendo así mayor resistencia a sequía (Behnam *et al.*, 2006, Kasuga *et al.*, 2004, Mathur *et al.*, 2004, Oh *et al.*, 2005). Por lo tanto, la identificación de los TFs de diferentes plantas cultivadas representa un importante primer paso para el diseño de cultivos más resistentes al estrés.

En años recientes se han secuenciado por completo los genomas de diversos cultivos y se han almacenado en bases de datos de acceso libre. Por otra parte, debido a que la identificación de los FTs por métodos experimentales suele ser complejo y costoso debido a la redundancia

funcional existente entre diversas familias (Riechmann *et al.*, 2000), se han desarrollado métodos computacionales, basados en los modelos de Cadenas Ocultas de Markov (HMM por sus siglas en inglés: Hidden Markov Models), que permiten identificar, con cierto grado de precisión, la presencia de motivos relacionados con las familias de FTs (Durbin *et al.*, 1998, Polanksi y Kimmel, 2007). Gracias a estos recursos ha sido posible el desarrollo de bases de datos dedicadas que contienen los FTs de diversos cultivos así como información complementaria (descripciones, estructuras de dominios, árboles filogenéticos, alineamientos múltiples, etc.) útil para la comunidad científica (Guo *et al.*, 2008, Kummerfeld y Teichmann, 2007, Riaño-Pachón *et al.*, 2007, Yilmaz *et al.*, 2009).

En el presente trabajo se realizó la identificación y anotación de FTs en los genomas de soya (*Glycine max*), trigo (*Triticum aestivum*) y maíz (*Zea maiz*). Cultivos que tienen importancia mundial como fuente de proteína para la alimentación humana y por la gran cantidad de derivados agroindustriales que es posible obtener de ellos (Tran y Nguyen, 2009). Desde el punto de vista agronómico, este trabajo puede ser usado como punto de partida para un estudio más detallado, usando técnicas biotecnológicas, de aquellos genes implicados en el rendimiento, calidad de semilla, relaciones simbióticas (nodulación, micorrización, etc) y respuesta al estrés tanto biótico (plagas, enfermedades, competencia, etc) como abiótico (sequía, bajas temperaturas, salinidad, etc.).

Este documento está organizado de la siguiente manea: En el capítulo 2 se encuentran los objetivos de la tesis; El capítulo 3 se divide en dos partes. Primero se hace una revisión de los conceptos de biología realacionados al ADN, proteínas y factores de transcripción, así como el papel de la Bioinformática en el análisis de la información biológica. Posteriormente, se aborda el problema de identificación de FTs y el uso de modelos de Cadenas Ocultas de Markov como solución; En el capítulo 4 se distinguen los apartados de identificación de FTs, anotación y desarrollo de la interfaz web. En ellos se indican las fuentes de datos, métodos y software utilizados para la construcción de la base de datos FT-MTS; En el capítulo 5 se presentan los resultados. Primero se listan las familias y la cantidad de FTs encontrados en los tres genomas. En seguida están los resultados de la anotación, la identificación de estructura terciaria, los alineamientos múltiples de los dominios de unión y los grupos de

genes ortólogos. Posteriormente se describe la estructura de la base de datos. Finalmente, se describen las distintas secciones de la página web donde destaca el uso de Applets Java y las búsquedas en la base de datos FT-MTS utilzando el paquete HMMER3. En el capítulo 6 se encuentran las conclusiones. Por último, en el Anexo se encuentra el script utilizado para la clasificación de FTs, el modelo completo de la base de datos y los perfiles HMM.

Capítulo 2

Objetivos

- Identificar, con métodos bioinformáticos, los factores de transcripción presentes en los genomas de *Glycine max*, *Triticum aestivum* y *Zea maíz*.
- Crear una interfáz web que permita la búsqueda, visualización y descarga de la información generada.

Capítulo 3

Revisión de literatura

3.1. Ácidos nucleicos

El ácido desoxirribonucleico (ADN) es la molécula donde se encuentran codificadas todas las características biológicas de un ser vivo, también es el responsable de transmitir dichas características de una generación a otra a través de la herencia. El ADN se estructura en pequeños bloques, llamados genes, que contienen la información necesaria para la conformación y funcionamiento de todas las células (Lewin, 2004).

Existen cinco bases nitrogenadas que conforman los ácidos nucleicos: Adenina (A), Timina (T), Guanina (G), Citosina (C) y Uracilo (U). Tanto el ADN como el ARN están compuestos de cuatro bases. Para el primero dichas bases son: A, C, G y T; mientras que para el segundo son: A, C, G y U (Lehninger *et al.*, 2006).

El ADN y el ARN (ácido ribonucleico) son polinucleótidos, es decir, cadenas de nucleótidos unidos entre sí por enlaces fosfodiester entre el carbono $5'$ de un nucleótido y el carbono $3'$ del siguiente. Cada nucleótido está compuesto de un azúcar (dexosirribosa en ADN y ribosa en ARN), una base nitrogenada y un fosfato (Lehninger *et al.*, 2006).

El ADN está formado por dos cadenas de nucleótidos, unidas entre sí por enlaces covalentes conocidos como puentes de hidrógeno. Ambas cadenas forman una estructura de doble hélice (Figura 3.1) donde las bases se encuentran acomodadas hacia el eje, mientras que el azúcar

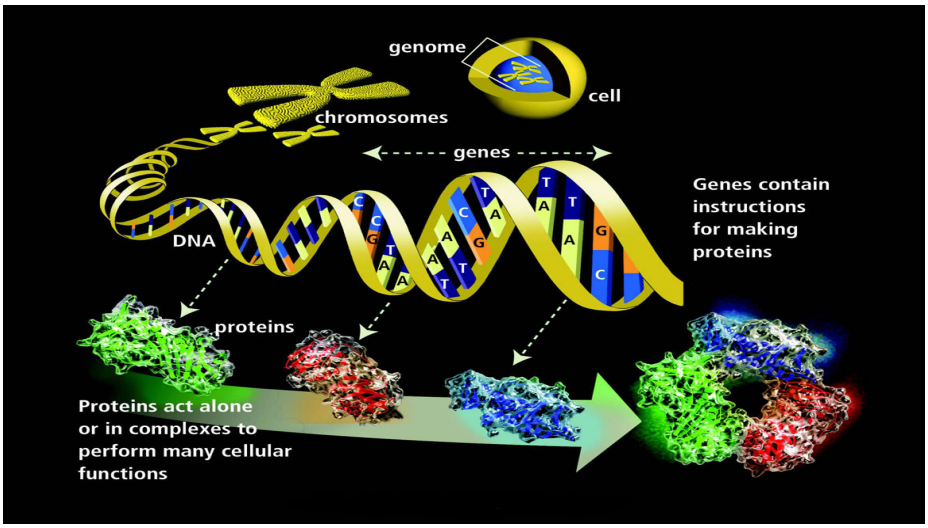

Figura 3.1: El ADN es una doble cadena de nucleótidos en forma de hélice. En él están codificados los genes, que a su vez codifican a las proteínas (Fuente: http://www.ornl.gov/sci/techresources/Human_Genome/publicat/primer2001/1.shtml).

y los fosfatos están orientados hacia el exterior de la molécula (Alberts *et al.*, 2004). Para que las cadenas se unan es necesario que cada base esté apareada sólo con otra que la complementa, esto es, A sólo se enlaza con T, y C sólo con G. A diferencia del ADN, el ARN no forma una doble cadena debido a la presencia de un oxígeno en la posición $2'$ de la ribosa.

3.2. Genes

De acuerdo con Lodish *et al.* (2003), un gen es una secuencia de nucleótidos necesaria para la síntesis de un producto funcional, normalmente una proteína (Figura 3.1) aunque algunas veces puede ser ARNr o ARNt.

De acuerdo con lo anterior, la estructura de un gen comprende no solo la secuencia codificante, también contiene la información necesaria para su expresión (Figura 3.2). La parte inicial está compuesta por una amplia zona de ADN no codificante donde se especifica el mecanismo para controlar la transcripción del gen. En dicha zona se encuentra el promotor, donde se acopla la ARN-polimerasa, además de regiones potenciadoras y silenciadoras, que activan o reprimen el proceso de transcripción. Posteriormente se encuentra una zona

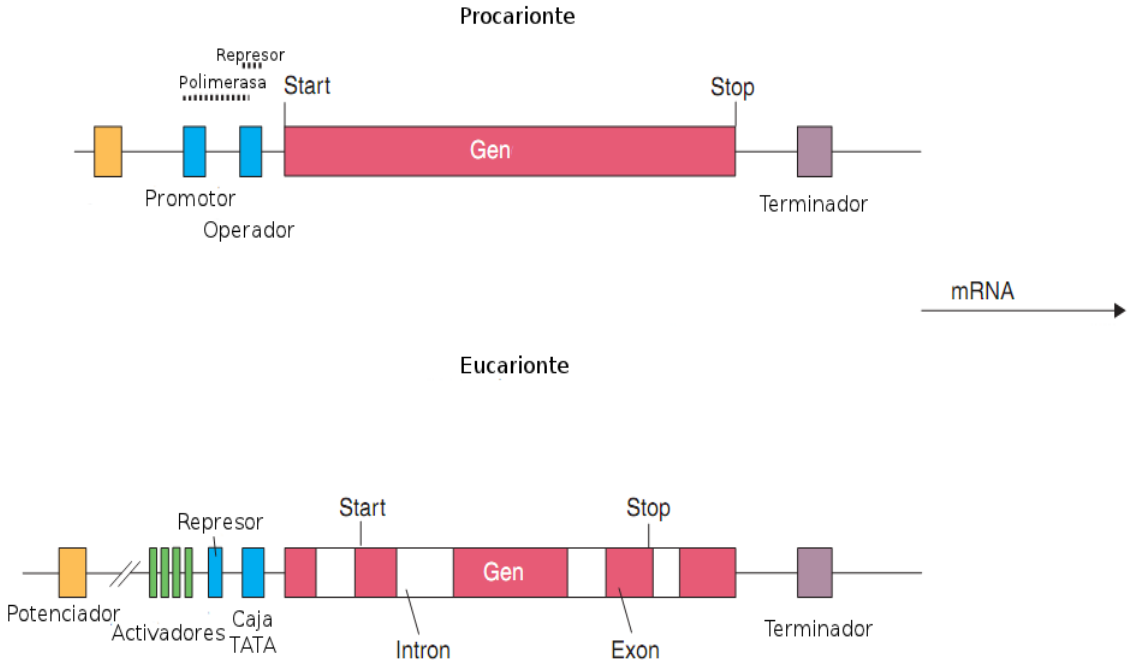

Figura 3.2: Arquitecturas típicas de un gen en organismos procariontes y en eucariontes. La principal diferencia entre genes de (a) procariontes y (b) eucariontes es la presencia de intrones en estos últimos (Reece, 2004).

codificante, que en eucariontes se divide en intrones y exones. Finalmente una zona rica en Timina señaliza el final del gen. En organismos eucariontes la estructura los genes es mucho más compleja que en procariontes (Lewin, 2004, Lodish *et al.*, 2003).

Los intrones son secuencias de AND no codificante que dividen a las secuencias que sí codifican (exones), sólo están presentes en los organismos eucariontes y son retiradas durante el proceso de corte y empalme, también llamado splicing (Lodish *et al.*, 2003). En muchos casos, los intrones constituyen una proporción mucho más grande que los exones, como en el caso del humano donde el 95 % de cada gen está compuesto por intrones (Lodish *et al.*, 2003).

3.3. Proteínas

Las proteínas son macromoléculas compuestas por una o varias cadenas polipetídicas, cada una de las cuales tiene una secuencia característica de aminoácidos unidos por un enlace

Figura 3.3: Los aminoácidos están codificados por grupos de tres nucleótidos llamados codones (Lodish *et al.*, 2003).

peptídico. Casi todo lo que ocurre en la célula requiere la intervención de las proteínas ya que realizan una enorme cantidad de funciones: estructurales, enzimáticas, de transporte, hormonales, reguladoras, señalizadoras, etc., organizándose en enormes redes funcionales de interacciones. En cada célula existen miles de proteínas diferentes, cada una codificada por un gen y encargada de realizar una tarea específica (Lehninger *et al.*, 2006, Zubay *et al.*, 1995).

3.3.1. Codones

Los aminoácidos están codificados por grupos de tres bases o tripletes llamados codones. Al ácido glutámico, por ejemplo, lo codifican los tripletes GAA y GAG. Existen 64 codones y, con excepción de cuatro, cada uno está encargado de codificar un aminoácido (Figura 3.3). Sin embargo, el codón AUG que codifica para metionina también sirve para indicar el inicio de la proteína y cualquiera de los tres codones UAA, UAG y UGG, marca el final de la transcripción (Lewin, 2004).

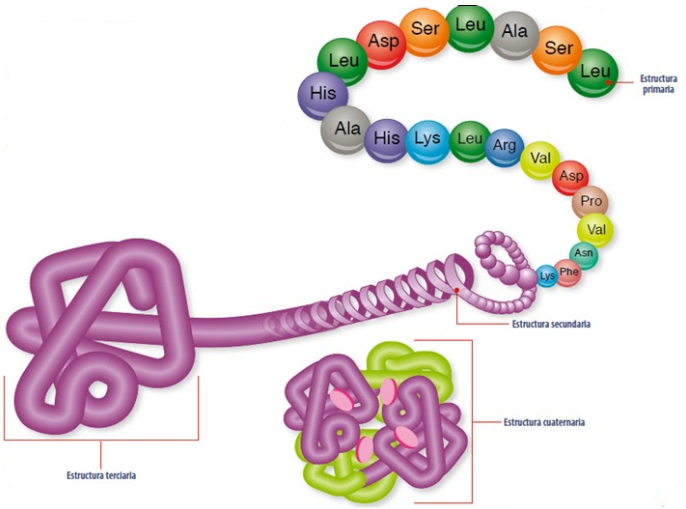

Figura 3.4: Niveles de estructura de las proteínas (Madigan *et al.*, 2004).

3.3.2. Estructura de las proteínas

La función biológica de una proteína está determinada por su estructura tridimensional (Petsko y Ringe, 2004). Esquemáticamente, se considera que hay cuatro niveles de estructuración de las proteínas. La hemoglobina (Figura 3.4), por ejemplo, se estructura de la siguiente forma: su estructura primaria comprende la secuencia de aminoácidos unidos por enlaces peptídicos covalentes. El polipéptido resultante se enrolla en forma de hélice alfa, una de las clases de estructura secundaria. La hélice se pliega para formar la estructura terciaria, que a su vez, forma parte de una de las subunidades que constituyen la estructura cuaternaria (Zubay *et al.*, 1995).

3.4. Transcripción y traducción de ADN

3.4.1. Transcripción

La transcripción es un proceso mediante el cual la información contenida en el ADN es copiada en el ARN mensajero (ARNm) gracias a un complejo de proteínas que interactúan entre sí. El proceso comienza cuando varios factores de transcripción junto con la ARN polimerasa se unen al promotor del gen (Figura 3.5). A continuación, se separa la doble

cadena de ADN y la ARN polimerasa comienza a sintetizar el ARNm tomando como molde la cadena $3'$ del ADN. El copiado continúa hasta que otra secuencia específica desestabiliza el complejo ADN-ARN y provoca la separación de la ARN polimerasa (Lodish *et al.*, 2003).

3.4.2. Traducción

Se le conoce como traducción al conjunto de procesos mediante los cuales el ARNm es usado para unir diferentes aminoácidos en una cadena de polipéptidos, es decir, el ensamblado de la proteína (Figura 3.6). Según Madigan *et al.* (2004), en este proceso tres tipos de ADN están involucrados:

- **ARN mensajero (ARNm)**. Se encarga de transportar la información transcrita del ADN hacia el exterior del núcleo, donde se encuentran los ribosomas. Además, contiene los codones que codifican a la proteína en cuestión.
- **ARN de transferencia (ARNt)**. Su función es transportar los aminoácidos hacia los ribosomas y es la clave para descifrar los codones del ARNm. El ARNt es en realidad un conjunto de ARNt, cada aminoácido se une a un ARNt específico, el cual sólo se une a su codón complementario en el ARNm.
- **ARN ribosomal (ARNr)**. Junto con un complejo de proteínas forma los ribosomas. Los cuales, catalizan el proceso de ensamblado de aminoácidos para formar las proteínas.

3.5. Factores de transcripción (FTs)

La regulación de la expresión génica es crítica para numerosos procesos biológicos en los organismos. Desde el desarrollo del organismo, el control de las respuestas fisiológicas a cambios ambientales o ante la invasión de microorganismos patógenos, hasta de los procesos neurológicos como la memoria y el aprendizaje. Dependiendo de las condiciones y de la fases

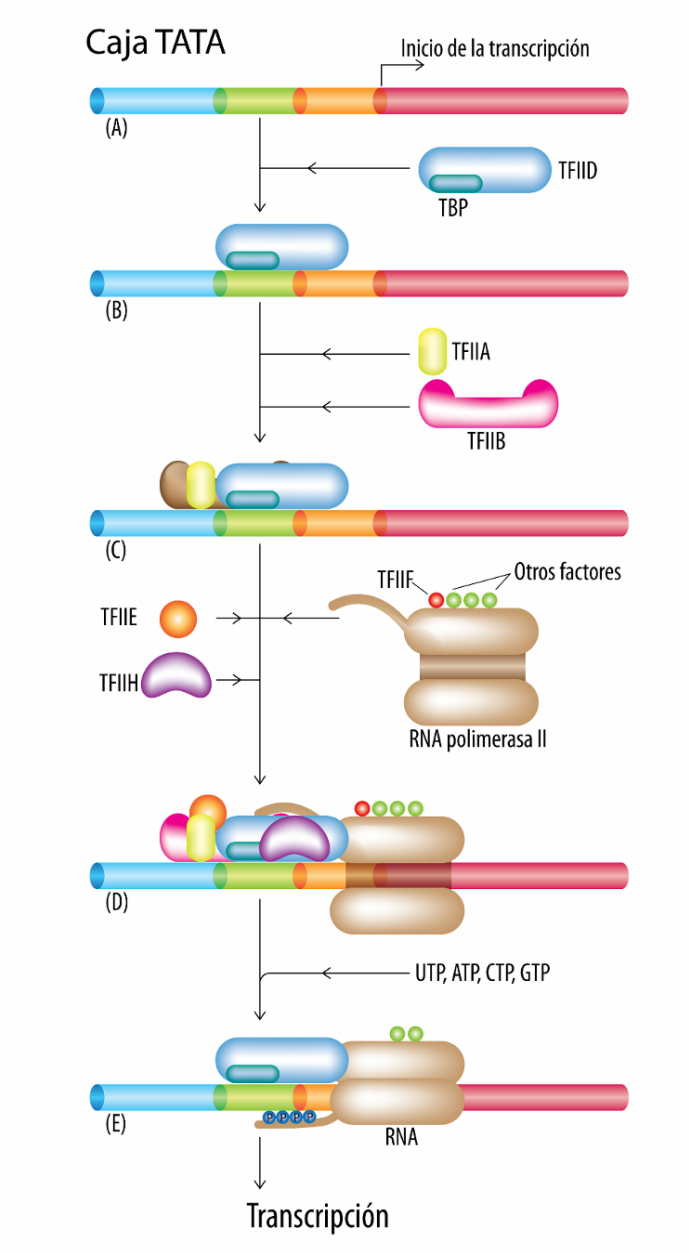

Figura 3.5: Complejo proteico para el inicio de la transcripción (Alberts *et al.*, 2004).

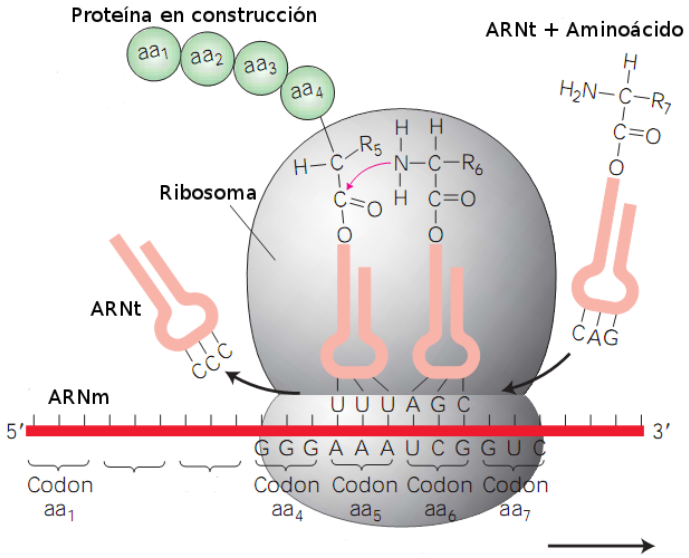

Figura 3.6: Proceso de ensamblado de una proteína (Traducción). El ARNm contiene la lista de aminoácidos, el ARNt se encarga de transportarlos y el Ribosoma se encarga de unirlos (Lodish *et al.*, 2003).

de desarrollo, las necesidades celulares respecto a un gen dado pueden variar ampliamente. Las proteínas maestras en la regulación de la expresión génica son conocidas como factores de transcripción (Lehninger *et al.*, 2006, Lodish *et al.*, 2003).

Los factores de transcripción son proteínas que, en interacción con la ARN-polimerasa y otras proteínas, regulan la expresión de uno o más genes (Figura 3.7). Estas proteínas, estimulan o reprimen la tasa transcripcional de los genes a los que regulan al unirse a regiones promotoras específicas (Latchman, 1997, Lodish *et al.*, 2003).

Existen dos tipos de factores de transcripción

- **Basales o generales.** Pertenecen a un conjunto mínimo de proteínas requeridas para la iniciación de la transcripción. Junto con la RNA polimerasa, forman el aparato transcripcional básico.
- **Regulatorios o específicos.** Afectan la iniciación de la transcripción al entrar en contacto con los componentes del complejo transcripcional.

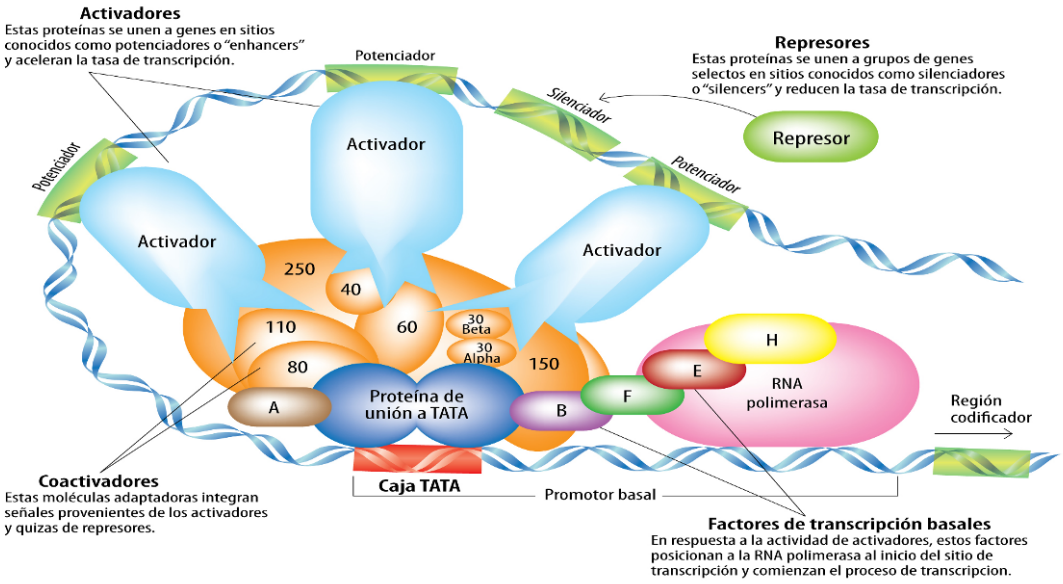

Figura 3.7: Complejo molecular transcripcional. Los factores de transcripción son indispensables para iniciar la transcripción del ADN (Tiessen, 2009).

3.5.1. Clasificación de los factores de transcripción

Los FTs pueden ser agrupados en diferentes familias de acuerdo a la estructura tridimensional de sus dominios de unión al ADN (Riechmann *et al.*, 2000).

En plantas se han identificado 68 familias de factores de transcripción, siendo las más numerosas: MYB-(R1), R2R3, AP2/EREBP, bHLH, NAC, C2H2 (Zn), HB, MADS, bZIP, WRKY (Zn), GARP, Dof (Zn), CO-like (Zn), y GATA (Zn) (Tiessen, 2009).

En seguida se resume la función de algunas familias:

- `AB13/VP1`: Junto con otros FTs como LEC1, LEC2 y FUS3 controla la maduración de las semillas y evita que los genes asociados a la germinación y el crecimiento se activen con antelación.

- `ARID`: Los genes que codifican estas proteínas están relacinados con diversos procesos biológicos, tales como desarrollo embrionario, regulación genética del linaje celular y control del ciclo celular.

- `AUX/IAA`: Estos genes muestran una rápida inducción en respuesta a la auxina ácido

indolacético. En arabidopsis afectan el crecimiento de la parte aérea y de la raíz.

- `C2C2-CO`: Controlan la floración en respuesta al fotoperiodo.
- `C2H2-ZPT2`: En arabidopsis incrementan la tolerancia al estrés retardando el crecimiento.
- `ERF`: Los miembros de esta familia se expresan en respuesta al estrés provocado por frío, salinidad y sequía.
- `CPP`: En leguninosas, el factor CPP1 participa en la regulación de los genes leghemoglobin del nódulo simbiótico.
- `NAC`: Estos FTs regulan los mecanismos de defensa contra patógenos. Por ejemplo, en arroz la expresión del gen OsNAC6 es ocacionada por estrés biótico y abiótico incluyendo ataques por hongos e insectos, frío, sequía y salinidad.
- `CH3CCCH`: En arroz y arabidopsis los genes de esta familia se expresan en distintos tejidos como respuesta respuesta al estrés tanto biótico como abiótico (salinidad, frío, hipoxia y estrés osmótico).
- `Whirly`: El gen *AtWhy1* de esta familia, es fundamental durante las respuestas de resistencia a enfermedades en plantas en una ruta dependiente del ácido salicílico.
- `WRKY`: Varios procesos fisiológicos únicos en plantas son regulados por genes de esta familia, incluyendo mecanismos de defensa contra patógenos, senescencia y desarrollo de tricomas.

3.6. Bioinformática

Inicialmente, el concepto de bioinformática solo hacía referencia a la creación y mantenimiento de bases de datos donde se almacenaban secuencias de nucleótidos y aminoácidos. También hacía referencia desarrollo de interfaces donde los investigadores pudieran acceder, localmente o por medio de la web, a los datos existentes (Mount, 2004). Sin embargo, pronto surgieron problemas más complejos, como por ejemplo, localizar un gen dentro de

una secuencia, predecir la expresión de los genes, alinear secuencias, predecir la estructura o función de proteínas, o poder agrupar secuencias de proteínas en familias relacionadas (Kanehisa y Bork, 2003).

La bioinformática es una disciplina que fuciona los campos de la biología, ciencias de la computación y tecnologías de la información. Con el objetivo de organizar, analizar datos (generalmente material genético) y simular sistemas o mecanismos de origen biológico (Kanehisa y Bork, 2003, Polanksi y Kimmel, 2007).

3.6.1. Bases de datos de secuencias

Las bases de datos de secuencias surgieron como respuesta a la gran cantidad de información biológica generada en los años 90's. Según Kanehisa y Bork (2003), dentro de las bases de datos de secuencias existen dos tipos: Primarias, que contienen información directa de la secuencia, estructura o patrón de expresión de ADN o proteínas, por ejemplo, Phytozome[1] y PlantGDB[2]; Secundarias, que guardan datos como relaciones evolutivas, mutaciones, agrupaciones, relación con enfermedades, etc., derivados del análisis de las bases de datos primarias. Un ejemplo de bases de datos secundarias son aquellas que proporcionan información exclusivamente sobre factores de transcripción, como PlnTFDB (Pérez-Rodríguez *et al.*, 2009), PlantTFDB (Zhang *et al.*, 2011), Grassius (Yilmaz *et al.*, 2009), PlanTAPDB (Kummerfeld y Teichmann, 2007), LegumeTFDB (Mochida *et al.*, 2009), entre otras.

3.7. Identificación de FTs

La función de una proteína está dada por su estructura tridimensional (Petsko y Ringe, 2004). Sin embargo, los métodos experimentales para su determinación suelen ser lentos y costosos debido a la redundancia funcional existente entre distintas familias de FTs (Khan, 2011, Tran *et al.*, 2009). Por ello, se han desarrollado métodos computacionales también llamados *in silico* (Baxevanis y Ouellette, 2001).

[1] http://www.phytozome.net/
[2] http://www.plantgdb.org/

El problema a resolver es el siguiente: Dada una secuencia de aminoácidos, ¿existe alguna manera de identificar de que proteína se trata?

Históricamente, la solución se basa en la idea de comparar una secuencia nueva, de la que se desconoce su función, con otras previamente caracterizadas (Baxevanis y Ouellette, 2001, Kanehisa y Bork, 2003), almacenadas en bases de datos como Pfam, y así poder inferir su función. En términos generales, se trata de encontrar secuencias que sean similares, dentro de algún parámetro cuantificable.

Para la realización de dichas comparaciones, se han desarrollado diferentes estrategias como (Mount, 2004): expresiones regulares y búsqueda de patrones, alinieamiento de secuencias, construcción de perfiles HMM, etc.

3.7.1. Alineamiento de secuencias

El alineamiento de secuencias es el procedimiento más común en bioinformática. El objetivo de éste es proporcionar un parámetro que permita decidir si dos secuencias muestran el sufiente grado de similitud como para ser consideradas homólogas. Aunque ambos términos se utilizan como sinónimos, existe una gran diferencia entre ambos. *Similitud* es una cantidad observable que puede ser expresada como, por ejemplo, porcentaje de identidad. *Homología*, por otra parte, es la conclución de que existe una relación evolutiva entre los genes comparados (Baxevanis y Ouellette, 2001).

Existen varios tipos de alineamientos:

- De acuerdo con el número de secuencias a alinear. Pareado y múltiple: para dos y para más de dos secuencias, respectivamente.
- De acuerdo a la región a alinear. Local, se utiliza una subregión de las secuencias. Global, se utilizan las secuencias completas.

En un alineamiento (Figura 3.8), los residuos (pueden ser bases o aminoácidos) que son alineados pero que no son idénticos representan *sustituciones*. En las zonas donde los residuos de una secuencia no tienen correspondencia con la otra se interpreta como *inserción* en la

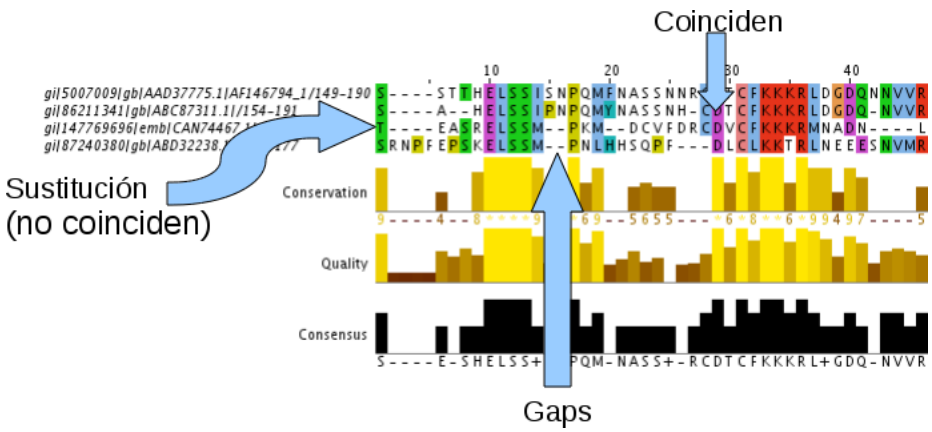

Figura 3.8: Alineamiento múltiple.

primera y *deleción* en la segunda. Estos huecos o *gaps* se representan normalmente como guiónes alineados con los residuos.

Cuando se realiza un alineamiento, se búsca maximizar la cantidad de residuos que coincidan y minimizar las inserciones/deleciones. Sin embargo, para la mayoría de los casos, la cantidad de posibles alineamientos es enorme. Por ejemplo, para encontrar el alineamiento óptimo, en el cual sean cosideradas todas las posibles sustituciones, inserciones y deleciones en proteínas con longitud de 300 bases, se necesitan 10^{88} comparaciones, lo que hace inviable su cálculo (Mount, 2004). Este problema es abordado con una estrategia conocida como programación dinámica que, en términos muy generales, consiste en obtener la solución de un problema complejo dividiendolo en subproblemas y combinando las soluciones de estos. En la aplicación este concepto se utiliza el algoritmo de Needleman-Wuncsh para alineamientos globales y el algoritmo de Smith-Waterman para alineamientos locales (Baxevanis y Ouellette, 2001).

El paquete BLAST (Basic Local Alignment Search, [3]) implementa el algoritmo de Smith-Waterman y desde hace más de 20 años ha sido ampliamente utilizado para la comparación de secuencias. Dado que el algoritmo es heurístico, no garantiza encontrar la solución correcta pero proporciona un parámetro estadístico (E-value) que permite juzgar el grado de significancia de los resultados obtenidos (Tiessen, 2009).

[3] `http://blast.ncbi.nlm.nih.gov/Blast.cgi`

3.7.2. Perfiles HMM

Como las posiciones específicas de los aminoácidos pueden no tener los mismos patrones de conservación bajo diferentes contextos, comparar las secuencias usando matrices de sustitución puede ser un método muy simplista. Es mejor buscar similitudes a nivel de dominio en lugar de hacerlo a nivel de secuencia.

Un dominio es una región espacialmente compacta de una estructura proteica que está relacionada con una determinada función y que es capáz de mantenerse estable independientemente del resto de la proteína (Petsko y Ringe, 2004).

Los FTs poseen dominios de unión al ADN que les permiten acoplarse sólo a ciertos genes al reconocer secuencias específicas en el ADN (Riechmann *et al.*, 2000). En consecuencia, para saber si una proteína es un factor de transcripción es necesario identificar en su secuencia la presencia de dichos dominios.

Existen varios paquetes de software empleados para determinar si existen patrones o dominios conservados en una secuencia de aminoácidos como, por ejemplo, HMMER[4] (Finn *et al.*, 2011) y SAM[5] . En HMMER una secuencia se compara con un perfil HMM y el puntaje resultante es una probabilidad de que la secuencia esté relacionada con el modelo dado (el perfil HMM).

Un perfil HMM es la implementación de un modelo estadístico conocido como: modelo de Cadenas Ocultas de Markov o HMM (por sus siglas en inglés: Hidden Markov Models). Su construcción se realiza a partir de un alineamiento múltiple de un dominio que es convertido en un sistema de puntaje de posiciones específicas (Eddy, 1998). La idea de usar perfiles HMM es comparar una secuencia con un modelo estadístico que describe a una familia o patrón de secuencias.

Un HMM es la generalización de una Cadena de Markov. Por esta razón, primero se exponen las bases teóricas de las Cadenas de Markov y en seguida dichos conceptos se extienden para los HMMs.

[4] `http://hmmer.janelia.org/`
[5] `http://compbio.soe.ucsc.edu/sam.html`

$$A = \{a_{ij}\} = \begin{bmatrix} 0,4 & 0,3 & 0,3 \\ 0,2 & 0,6 & 0,2 \\ 0,1 & 0,1 & 0,8 \end{bmatrix}$$

$$\pi_i = \frac{1}{3}, \quad 1 \leq i \leq 3$$

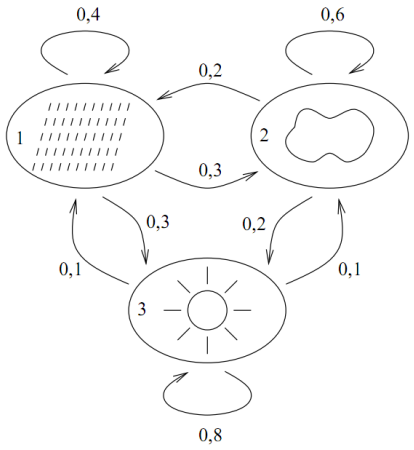

Figura 3.9: [
Cadena de Markov con tres estados (Polanksi y Kimmel, 2007).

3.7.2.1. Cadenas de Markov

Las cadenas de Markov se incluyen dentro de los denominados procesos estocásticos, esto es, procesos que evolucionan de forma no determinista a lo largo del tiempo en torno a un conjunto de datos.

Una cadena de Markov representa un sistema que varía su estado a lo largo del tiempo, siendo cada estado la caracterización de la situación en que se halla el sistema en un instante dado y cada cambio una transición del sistema. En éste sistema la probabilidad transición hacia el siguiente estado depende exclusivamente del estado anterior inmediato.

Una cadena de Markov puede ser representada mediante un grafo de estados de transición (Polanksi y Kimmel, 2007), como se muestra en la Figura 3.9. En el ejemplo los estados están representados por una condición climática (soleado, nublado y lluvioso) y las flechas indican las probabilidades de transición de un estado a otro.

La forma más cómoda de expresar las probabilidades de transición es mediante una matriz de probabilidades (Polanksi y Kimmel, 2007). La cual es cuadrada con tantas filas y columnas como estados tiene el sistema (Figura 3.9, izquierda). En ella, la posición de la fila representa el estado actual del sistema y la posición de la columna es el estado a donde el sistema transita. Del ejemplo anterior se puede observar que la probabilidad de pasar del estado lluvioso (fila 1) al soleado (columna 2) es 0.3.

Dicho de manera formal, de acuerdo con Rabiner (1989), una cadena de Markov es un proceso representado por N estados: Q=$(s_1, s_2, ..., s_N)$, que cambia, según cierta probabilidad, de un estado a otro en puntos de tiempo discretos (t=1,2,...,T). Se dice que en el tiempo t el sistema está en el estado q_t. A este proceso se le describe en términos probabilísticos de la siguiente manera:

$$P(q_t = s_j | q_{t-1} = s_i, q_{t-2} = k, ...) = P(q_t = s_j | q_{t-1} = s_i)$$

En otras palabras el estado actual en la cadena está determinado sólo por el estado anterior a él y es independiente de la evolución anterior del sistema (Durbin *et al.*, 1998). Ésta característica es conocida como *Propiedad Markoviana.*

Otra propiedad importante es que las probabilidades de transición de q_{t-1} a q_t, denotadas por a_{ij}, donde

$$a_{ij} = P(q_t = s_j | q_{t-1} = s_i) = P(s_j | s_i)$$

son constantes en el tiempo (probabilidades estacionarias). Además, deben satisfacer:

$$i) \quad 1 \leq i, j \leq N \qquad ii) \quad a_{ij} \geq 0, \quad \forall i, j \qquad iii) \quad \sum_{j=1}^{N} a_{ij} = 1, \quad \forall i$$

Por otra parte, la probabilidad del estado inicial está dada por:

$$\pi_i = (q_1 = i), \qquad \pi_i \geq 0, \qquad 1 \leq i \leq N, \qquad \sum_{i=1}^{N} \pi_i = 1$$

Finalmente, una vez especificadas las probabilidades iniciales y las probabilidades de transición es razobable preguntar, ¿cuál es la probabilidad de observar una determinada secuencia de estados($O = o_1, o_2, ..., o_T$)? Por ejemplo, dado que el clima en el en el dia 1 (t=1) se encuentra soleado (estado 3), ¿cuál es la probabilidad, de acuerdo con el modelo(Figura 3.9), de que los siguientes 3 días sean: soleado-lluvioso-nublado, es decir, $O = (s_3, s_3, s_1, s_2)$? Para calcular dicha probabilidad tenemos:

$$P(O|Modelo) = P(o_1, o_2, ..., o_T) = \pi_{o1} \prod_{t=1}^{T-1} a_{o_{t-1}o_t}$$

$$= P(q_1 = o_1)P(q_2 = o_2|q_1 = o_1)P(q_3 = o_3|q_2 = o_2)...P(q_T = o_T|q_{T-1} = o_{T-1})$$

Lo que para el ejemplo que se está utilizando es:

$$P(s_3, s_3, s_1, s_2) = P(s_3)P(s_3|s_3)P(s_1|s_3)P(s_2|s_1)$$

$$= \pi_3(p_{33})(p_{31})(p_{12})$$

$$= (1)(0.8)(0.3)(0.2) = 0.048$$

3.7.2.2. Modelos de Cadenas Ocultas de Markov

Un modelo de Cadenas Ocultas de Markov o HMM (por sus siglas en inglés: Hidden Makov Models), es una cadena de Markov cuyos estados no son observables, sólo es observable una secuencia de símbolos emitidos por los estados (Polanksi y Kimmel, 2007). Cada estado tiene una distribución de probabilidad sobre los posibles símbolos de salida. Consecuentemente, la secuencia de símbolos (observables) generada por un HMM proporciona cierta información acerca de la secuencia de estados (ocultos). Dicho de otra forma, un HMM tiene dos procesos estocásticos embebidos, uno no es observable (estados ocultos), pero puede ser inferido mediante el otro proceso (secuencia de observaciones).

En el siguiente ejemplo, tomado de Eddy (2004), se explica el concepto de HMM aplicado a la identificación del sitio $5'$ en una secuencia de ADN.

Se tiene una secuencia de ADN dividida en tres partes: exón, intrón y un sitio de empalme ($5'$) que los separa. Sin embargo, se desconoce la posición del sitio $5'$. El problema es identificar dicha posición.

Afortunadamente, se sabe que en cada parte las bases tienen una distribución estadística

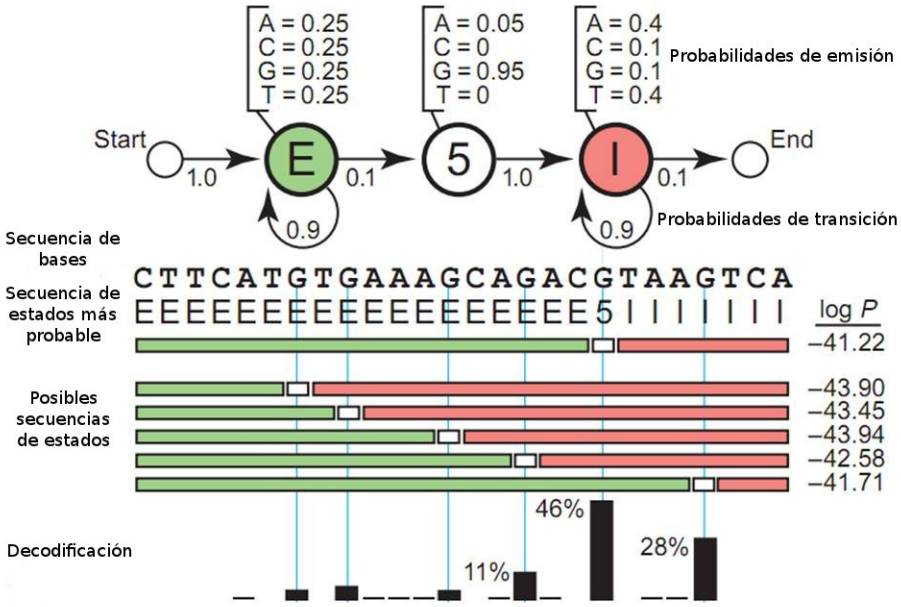

Figura 3.10: Identificación del sitio 5′ en una secuencia de ADN utilizando un HMM (Eddy, 2004).

diferente. En el exón las bases se distribuyen de manera uniforme (en proporción de un 25 % cada una), el intrón es rico en A/T (40 % para cada A/T y 10 % para cada C/G) y el sitio 5′ es casi siempre G (95 % G y 5 % A).

Con esta información se puede esquematizar un HMM con tres estados (Figura 3.10): E (exón), 5 (sitio 5′) e I (intrón). En cada estado las *probabilidades de emisión* de bases (símbolos de salida) son diferentes (arriba de los estados). Además, cada estado tiene sus *probabilidades de transición* (flechas). En este caso se espera que los cambios de un estado a otro ocurran de forma lineal, es decir, empezando por E el modelo puede continuar en él o transitar hacia el estado 5′ y luego transitar al estado I donde permanecerá hasta llegar al final de la secuencia.

El HMM construido con los parámetros anteriores funciona de la siguiente manera: Estando en un estado inicial, digamos E, el modelo genera una base de acuerdo a las probabilidades de emisión que le son propias (A 25 %, T 25 %, C 25 % y G 25 %). En seguida, en base a sus probabilidades de transición (90 % de probabilidades de que el siguiente estado siga siendo E y 10 % de que transite a 5′), el modelo transita hacia el siguiente estado, que puede ser él

mismo. Una vez que se llega al estado $5'$, el modelo solo puede generar A (95 %) o G (5 %) y la transición es automática hacia el estado I (la probabilidad de ir al estado I estando en $5'$ es del 100 %).

En este ejemplo, la parte observable del HMM es la *secuencia de bases* y la parte oculta, la cual se pretende inferir, es la *secuencia de estados* (E, $5'$ o I). De esta forma, la secuencia de estados es una cadena de Markov ya que el siguiente estado depende solamente del estado actual del modelo. Además, al solo contar con la información proporcionada por la secuencia de bases, la secuencia de estados está oculta y por lo tanto es un HMM.

Replanteando el problema inicial, se tiene una secuencia de bases de 26 nucleótidos y se quiere inferir la secuencia de estados que la generó. Dado que existen 14 secuencias de estados capaces de generar la misma secuencia de bases, lo que se busca es encontrar aquella secuencia de estados con mayor probabilidad.

Para calcular la probabilidad de una determinada secuencia de estados, se tienen que multiplicar todas las probabilidades de emisión y las probabilidades de transición utilizadas por dicha secuencia. Por ejemplo, considerando la secuencia de estados de la Figura 3.10 (en la parte superior a las demás secuencias), donde el sitio $5'$ se encuentra en la posición 19, se tienen 26 emisiones y 27 transiciones, y el cálculo sería el producto de dichas probabilidades:

$$\log\{(1.0\times0.25)(0.9\times0.25)(0.9\times0.25)\cdots(0.1\times0.95)(1.0\times0.4)(0.9\times0.4)\cdots(0.9\times0.4)\} = -41.22$$

donde cada paréntesis representa un estado y en ellos se agrupan las probabilidades de transición (izquierda) y la probabilidades de emisión (derecha) relacionadas con dicho estado. Además, se utiliza el logaritmo de la probabilidad por tratarse de números muy pequeños.

En la parte intermedia de la figura 3.10 se muestran las seis secuencias de estados con más alta probabilidad. De ellas, la que tiene la mayor probabilidad es la que fue utilizada para el cálculo en el ejemplo anterior, en el que se obtuvo un valor de -41.22. **Con esto podemos inferir que la quinta G en la secuencia es la posición más probable del sitio $5'$.**

Para formalizar las ideas expuestas anteriormente, un HMM, según Rabiner (1989), se caracteriza por:

1. N, el número de estados en el modelo. Denotados por S=$(s_1, s_2, ..., s_N)$, donde el estado del modelo en el tiempo t es q_t.

2. M, la cantidad de símbolos o emisiones por estado (son los sucesos observables) y se denota por V=$(v_1, v_2, ..., v_M)$; y a la observación en el tiempo t como O_t

3. Las probabilidades de transición para cada estado s_i hacia otro estado s_j. Es decir, A=$\{a_{ij}\}$, donde

$$a_{ij} = P(q_t = s_j | q_{t-1} = s_i) = p(s_j|s_i), \qquad 1 \leq i, j \leq N, \qquad 1 \leq t \leq T$$

$$\sum_{j=1}^{N} a_{ij} = 1$$

4. La distribución de probabilidades de emisión para cada estado, B=$\{b_j(v_k)\}$ donde

$$b_j(v_k) = P(O_t = v_k | q_t = s_j) = P(v_k|s_j),$$

$$b_j(v_k) \geq 0, \qquad 1 \leq j \leq N, \qquad 1 \leq k \leq M, \qquad ,1 \leq t \leq T$$

$$\sum_{k=1}^{M} b_j(v_k) = 1, \qquad \forall j$$

5. La distribución inicial de los estados $\pi = \{\pi_i\}$, donde

$$\pi_i = P(q_1 = s_i), \qquad \pi_i \geq 0, \qquad 1 \leq i \leq N$$

$$\sum_{i=1}^{N} \pi_i = 1$$

Así, para la especificación de un HMM se requiere de todos los parámetros mencionados anteriormente y para referirse al modelo se usará la notación $\lambda = (A, B, \pi)$.

De acuerdo con Rabiner (1989), existen tres problemas que deben resolverse para poder hacer aplicaciones de un HMM:

1. Dada una secuencia de observaciones O=$(O_1, O_2, ..., O_T)$ y el modelo $\lambda = (A, B, \pi)$,

calcular $P(O|\lambda)$. Dicho de otra manera, ¿cuál es la probabilidad de observar la secuencia O, dado el modelo λ? (Algoritmo forward-backward)

2. Dada una secuencia de observaciones O=$(O_1, O_2, ..., O_T)$ y el modelo $\lambda = (A, B, \pi)$, encontrar la secuencia de estados Q=$(q_1, q_2, ..., q_T)$ que mejor explique las observaciones. (Algoritmo de Viberbi)

3. Dada una secuencia de observaciones O=$(O_1, O_2, ..., O_T)$, estimar los parámetros del modelo $\lambda = (A, B, \pi)$. (Algoritmo de Baum-Welch)

Capítulo 4

Métodos

La construcción de la base de datos FT-MTS se divide en 4 fases generales (Figura 4.1):

- **Identificación de FTs.** Se descargaron las secuencias de los genomas y los perfiles HMM de cada familia. Algunos perfiles HMM no estaban disponibles así que se procedió a construirlos. A continuación, se procedió a realizar la identificación de dominios de unión utilizando el software HMMER v3.0[1] (Finn *et al.*, 2011). Aquellas proteínas cuyos dominios poseían un score mayor o igual al "punto de corte" definido por cada perfil HMM pasaron a la etapa de clasificación. Para clasificar una proteína dentro de una determinada familia de FTs se hizo uso de un conjunto de reglas basadas en la presencia o ausencia de los dominios de unión (ver más adelante: "Reglas de clasificación").

- **Anotación.** Se incluyó una gran cantidad de información que complementa la lista de FTs identificados en el paso anterior, así, para cada FT se tienen: arquitectura de dominios, estructuras 3D, marcadores de secuencia expresada (ESTs), secuencias (aminoácidos y bases) y grupos de genes ortólogos. Adicionalmente, cada familia cuenta con una descripción, alineamientos múltiples, dominios de unión y referencias a literatura.

- **Implementación de la base de datos.** Se utilizó el gestor MySQL y para llenar todas las tablas se utilizaron programas escritos en Perl.

[1] `http://hmmer.janelia.org/`

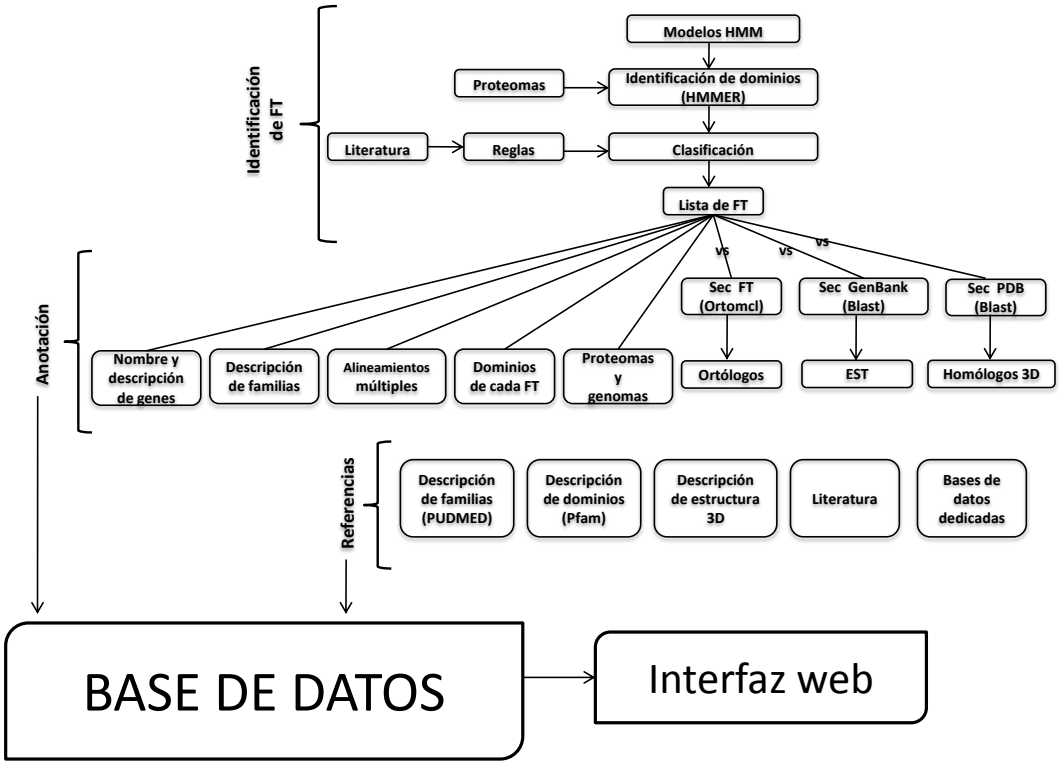

Figura 4.1: Fases de construcción de la base de datos FT-MTS. **vs** significa que los FTs se compararon contra las secuencias de otra base de datos. Por ejemplo, las de Protein Data Bank (PDB) para obtener los homólgos 3D.

- **Creación de la interfaz Web.** El sitio se construyó utilizando el Web-Framework Catalyst. Para mejorar el aspecto visual se añadieron elementos en JavaScript. Además, para visualizar algunos elementos como los alineamientos múltiples y la estructura 3D de los dominios se incorporaron Applets de Java.

Tabla 4.1: Obtención de genomas.

Especie	Fuente	Versión	Referencia
Glycine max	Núcleo	1.0	Schmutz *et al.* (2010)
	Cloroplasto		Saski *et al.* (2005)
Triticum aestivum	Núcleo	1.0	-
	Cloroplasto		Ogihara *et al.* (2001)
Zea maiz	Núcleo	5b.6	Schnable *et al.* (2009)
	Mitocondria		Clifton *et al.* (2004)
	Cloroplasto		Maier *et al.* (1995)

4.1. Identificación de FTs

4.1.1. Obtención de genomas

Las secuencias (genoma y proteoma) de *Glycine max* se obtuvieron del sitio Phytozome[2], mientras que las de *Triticum aestivum* se descargaron de PlantGDB[3] y las de *Zea mays* (v5b.6) de Maisequence[4]. Adicionalmente, para contar la mayor información posible se descargaron del sitio de NCBI[5] las secuencias pertenecientes a mitocondria y cloroplasto de las tres especies (Tabla 4.1).

4.1.2. Obtención y construcción de perfiles HMM

La mayoría de los perfiles HMM que se utilizaron para la identificación de dominios provienen de la base de datos de Pfam v25.0[6] (Finn *et al.*, 2010). Sin embargo, para algunas familias no existía un perfil adecuado en dicha base de datos, por lo que fue necesario construirlos (Dr. Diego Mauricio Riaño Pachón, comunicación personal). Para ello, se siguió la metodología utilizada por Pérez-Rodríguez *et al.* (2009), y que a continuación se describe brevemente:

- Utilizando una secuencia modelo de cada dominio se realiza una búsqueda PSI-BLAST (Position-Specific Iterative Basic Local Alignment Search Tool) contra la base de datos de proteínas del NCBI (National Center for Biotechnology Information)[7]. Dicha

[2] http://www.phytozome.net/soybean.php
[3] http://www.plantgdb.org/TaGDB/
[4] http://www.maizesequence.org/index.html
[5] http://www.ncbi.nlm.nih.gov/sites/entrez?Db=genome&Cmd=ShowDetailView&TermToSearch=19361,18974,15654,19401,10590
[6] ftp://ftp.sanger.ac.uk/pub/databases/Pfam/releases/Pfam25.0/Pfam-A.hmm.gz
[7] http://www.ncbi.nlm.nih.gov/

búsqueda permite identificar las regiones conservadas o motivos entre proteínas de organismos distantes y da como resultado un alineamiento múltiple de las secuencias similares encontradas.

- Con los alineamientos múltiples de cada dominio se construye el perfil HMM utilizando el software HMMER v3.0.

- Para definir el "punto de corte" de cada nuevo modelo se realiza una búsqueda de dominios, utilizando nuevamente el software HMMER y los nuevos modelos, en especies donde dichos dominios han sido caracterizados experimentalmente. Dicha búsqueda permite reconocer los falsos positivos de los verdaderos, de esta manera el "punto de corte" es establecido como la puntuación media entre ambos.

4.1.3. Reglas de clasificación

De acuerdo con Riechmann *et al.* (2000) los Fts y Trs pueden ser identificados y agrupados en familias a partir de la estructura de sus dominios de unión al ADN. Basado en esta idea, tras una revisión exaustiva de literatura, Riaño-Pachón *et al.* (2007) propuso un conjunto de reglas que posteriormente fueron ampliadas por Pérez-Rodríguez *et al.* (2009) para 110 familias. El presente trabajo utilizó una nueva versión con 134 reglas (Dr. Diego Mauricio Riaño Pachón, comunicación personal). En esta, se descartó el uso de 10 familias y se incluyeron 34 nuevas (BAF1, Bromodomain, CENP-B, Coactivator_p15, Copper_fist, CP2, DeoR, Fis, GATA-N, GntR, HTH, IclR, KILA, luxR, MED26, Mga, NDT80, NOT2_3_5, pipsqueak, Pseudo_ARR-B, SAND, SART-1, SGT1, SRF-TF, STE, TFIIS, Top, WhiA, Yl1, zf-A20, zf-BED, zf-LSD1, zf-MIZ, zf-NF-X1, ZnClus).

Las reglas toman en cuenta la presencia de ciertos dominios y, en algunos casos, la ausencia de otros. Una representación gráfica de estas se muestra en la Figura 4.1. En ella, los círculos representan a las familias de FTs y los cuadros representan a los diferentes dominios de unión: amarillos si los modelos fueron obtenidos de Pfam y verdes si fueron generados, como mencionó anteriormente. Las líneas verdes indican qué dominios caracterizan a una determinada familia (Por ejemplo, para que una proteína puede ser clasificada dentro de la

familia DBP de poseer los dominios de unión PP2C y DNC). En la mayoría de los casos es suficiente con la presencia de un solo dominio. Las líneas rojas aparecen en caso de que algún dominio NO deba de estar presente (Por ejemplo, un FT de la familia TRAF no puede presentar los dominios zf-TAZ, BACK, MATH y NPH3. Sin embargo, el dominio BTB sí debe de estar presente).

Varias proteínas presentaron una combinación de dominios que, de acuerdo con las reglas, no permitía clasificarlas dentro de ninguna familia. En estos casos, dichas proteínas fueron asignadas a la familia especial "huerfanos".

La reglas de clasificación fueron implementadas con un programa escrito en Perl (v5.12.4, Anexo A).

4.2. Anotación

4.2.1. Identificación de ortólogos

Los ortólogos son genes con cierto grado de similitud que están presentes en diferentes especies y que comparten un ancestro común pero que divergieron debido a la especiación (Fitch, 2000, Gogarten y Olendzenski, 1999). La identificación de ortólogos es importante para la anotación funcional de los genomas y ha sido ampliamente usada para facilitar estudios de genómica comparativa y en estudios de evolución de genes.

Existen varias alternativas para la identificación de ortólogos: RIO (Zmasek y Eddy, 2002), Orthostrapper (Hollich *et al.*, 2002), RSD (Wall *et al.*, 2003), OrthoMCL (Li *et al.*, 2003) e INPARANOID (Remm *et al.*, 2001). Sin embargo, para este trabajo se decidió utilizar OrthoMCL puesto que da mejores resultados cuando se comparan más de dos especies (Chen *et al.*, 2007).

OrthoMCL es un algoritmo de agrupamiento diseñado para identificar proteínas homologas (ortólogos y parólogos) a partir de secuencias similares. En términos generales consta de las siguientes etapas[8]:

[8]Ver algoritmo completo en: `http://orthomcl.org/common/downloads/software/v2.0/UserGuide.txt`

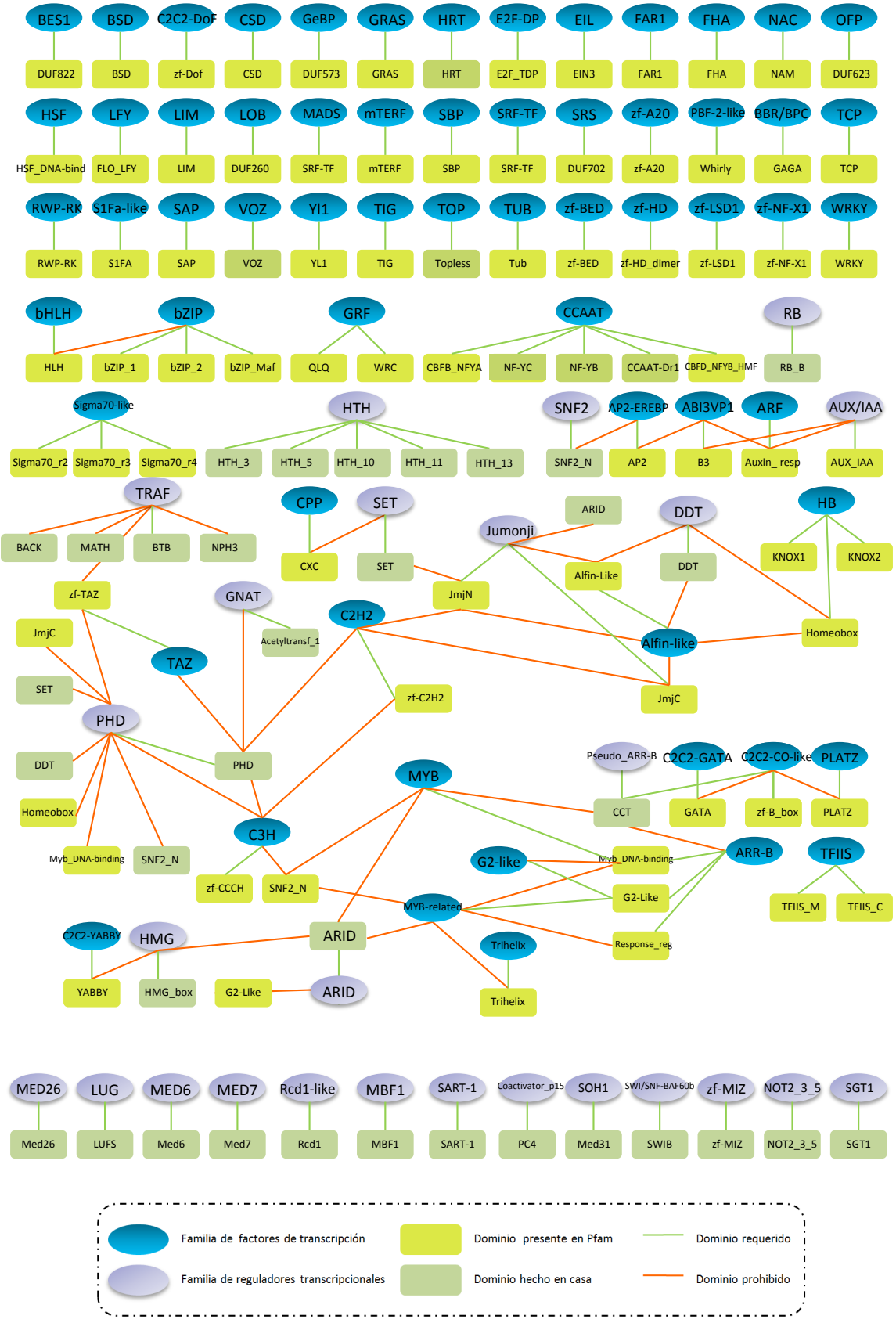

Figura 4.2: Reglas de clasificación de los FTs.

1. Búsqueda de secuencias similares mediante BLAST, utilizando los proteomas de cada especie y comparando cada secuencia contra todas las demás.

2. Identificación de potenciales ortólogos y parálogos a través de un grafo de similitud, usando el porcentaje de similitud de cada hit del BLAST.

3. Creación de grupos de ortólogos y parológos utilizando el algoritmo de Agrupamiento de Markov o MCL (por sus siglas en ingles: Markov Clustering algorithm) (Enright *et al.*, 2002).

4.2.2. Identificación de la estructura terciaria

Se realizó una búsqueda de similitud mediante BLAST, comparando la secuencia de cada FT con las de la base de datos de PDB, usando como parámetro un e-value de 1e-10. Posteriormente, de las secuencias que resultaron similares, se obtuvieron de la base de datos PDB los archivos de imagen y los archivos de estructura. Estos últimos, son archivos que describen la estructura tridimensional de la proteína incluyendo las coordenadas atómicas, las conexiones entre átomos, rotámeros laterales, etc.

4.2.3. Alineamientos múltiples

Se realizaron alineamientos múltiples por cada especie y dominio de unión. Por ejemplo, se realizó un alineamiento del dominio de unión HMG_box con las secuencias de *Zea maiz* pertenecientes a la familia HMG. Sólo se tomaron en cuenta aquellos dominios que, de acuerdo con las reglas de clasificación (Sección 4.1.3), deben estar presentes. Dichos alineamientos fueron construidos utilizando el software MAFFT v6.859[9] (Katoh y Toh, 2008).

4.2.4. Descripción de familias y referencias a bases de datos externas

Para cada una de las familias de FTs identificadas se incorporó a la base de datos una descripción y referencias a artículos. Dicha información se obtuvo mediante una búsqueda

[9] `http://mafft.cbrc.jp/alignment/software/`

manual en la literatura, utilizando como fuente principal la base de datos de PubMed[10]. Las bases de datos externas son Protein Data Bank (PDB)[11] y PubMed.

4.3. Implementación de la base de datos

Se decidió utilizar a MySQL v5.0[12] como programa gestor de base de datos, frente a otras alternativas como SQLServer, Oracle, PostgreSQL, etc. En seguida se anotan algunas razones:

- El uso que tendrá la base de datos FT-MTS será exclusivamente para consultas relativamente poco complejas. Y es precizamente en ese tipo de escenarios donde MySQL ha demostrado ser el manejador con mejor rendimiento.
- Se caracteriza por un bajo consumo de recursos, lo que lo hace ideal puesto que no se cuenta con un servidor dedicado.
- Usa licencia tipo GPL, la cual, entre otras cosas, permite usar el software sin ningún costo.
- Es multiplataforma, lo que permite instalarlo en cualquier servidor linux, como es nuestro caso.

4.4. Construcción de la interfaz web

Para desarrollar la interfaz web se utilizó Catalyst v5.8[13], un Web-Framework escrito en Perl (Antano, 2010). Soporta la arquitectura Modelo Vista Controlador (MVC) en la que se separan los datos de la aplicación (modelo), la interfaz de usuario (vista) y el manejador de eventos, usualmente peticiones de usuario (controlador). Catalyst implementa la arquitectura MVC de la siguiente manera:

[10] `http://www.ncbi.nlm.nih.gov/pubmed/`
[11] `http://www.rcsb.org/pdb/home/home.do`
[12] `http://www.mysql.com/`
[13] `http://www.catalystframework.org/`

- **Modelo.** A través del modulo *DBIx:Class* se hace una conversión de datos que crea una base de datos virtual orientada a objetos a partir de una base de datos relacional, utilizando una técnica de programación conocida como **mapeo objeto-relacional** o ORM (por sus siglas en inglés: Object-Relational mapping).

- **Vista.** Usualmente manejada por el sistema de procesamiento de plantillas **Template Toolkit**[14].

- **Controlador** Estos son escritos por el programador. Aunque es posible utilizar una gran cantidad de módulos de Perl que proporcionan funciones útiles para la programación web (Por ejemplo: *Catalyst::Plugin::FormValidator, Catalyst::Plugin::Prototype, Catalyst::Plugin::Account::AutoDiscovery*, etc.).

El estilo visual fue definido con Hojas de Estilo en Cascada o CSS (por sus siglas en inglés: Cascading Style Sheets). Para mejorar el estilo y dar dinamismo a las páginas se utilizó el lenguaje Javascript. También se incorporaron un par de Applets de Java:

- JmolApplet[15]. Un visor de código abierto para estructuras químicas en tres dimenciones, que permite abrir los archivos de estructura.

- Jalview[16] (Waterhouse *et al.*, 2009). Con este visor es posible obtener una representación visual de los alineamientos múltiples descritos en la sección 4.2.3.

Para asegurar al máximo posible la visualización correcta de la página en diferentes navegadores Web se utilizó el Servicio de Validación de CSS[17] y el Servicio de Validación de Marcado[18].

[14] `http://template-toolkit.org/`
[15] `http://www.jmol.org/`
[16] `http://www.jalview.org/`
[17] `http://jigsaw.w3.org/css-validator/`
[18] `http://validator.w3.org/`

Capítulo 5

Resultados

5.1. Identificación de FTs

5.1.1. Perfiles HMM

Se utilizaron en total 160 Perfiles HMM, 144 de los cuales fueron obtenidos de la base de datos de Pfam, los 16 restantes (Alfin-Like, CCAAT-Dr1, DNC, G2-Like, HRT, LUFS, NF-YB, NF-YC, NOZZLE, SAP, SW13, Topless, Trihelix, ULT, VIP3 y VOZ) se encuentran en el Anexo A.

5.1.2. Familias de FTs

De un total de 170756 proteínas, 13975 fueron clasificadas como factores de transcripción que se distribuyen en 93 familias de FTs. (Tabla 5.1). Una lista con la cantidad de FTs agrupados por familias y por especie se encuentra en la tabla 5.3.

En la tabla 5.2 se muestran las familias y la cantidad de FTs por especie que pueden ser considerados de importancia agronómica por el control que ejercen sobre genes que regulan la floración, la nodulación (sólo en el caso de la soya) y la resistencia al estrés (salinidad, sequía, plagas y enfermedades).

Tabla 5.1: Total de familas de FTs por especie.

Especie	No. proteínas	No. FTs	No. Fam
Glycine max	55870	5677	92
Triticum aestivum	51072	3410	91
Zea maiz	63814	4888	91
`TOTAL`	170756	13975	93

Tabla 5.2: FTs de importancia agronómica.

Familia	*Glycine max*	*Triticum aestivum*	*Zea maiz*
ABI3VP1	65	66	79
Alfin-like	27	11	36
AP2-EREBP	368	196	256
ARF	74	48	67
ARID	20	13	15
ARR-B	25	8	11
AUX/IAA	95	61	91
BBR/BPC	21	7	9
BES1	17	6	16
bHLH	334	182	261

Tabla 5.3: Cantidad de FTs por familia y por especie.

Familia	*Glycine max*	*Triticum aestivum*	*Zea maiz*
ABI3VP1	65	66	79
Alfin-like	27	11	36
AP2-EREBP	368	196	256
ARF	74	48	67
ARID	20	13	15
ARR-B	25	8	11
AUX/IAA	95	61	91
BBR/BPC	21	7	9
BES1	17	6	16
bHLH	334	182	261
BSD	19	11	14
bZIP	190	134	211
C2C2-CO-like	18	13	15
C2C2-Dof	85	37	51
C2C2-GATA	58	32	54

Familia	*Glycine max*	*Triticum aestivum*	*Zea maiz*
C2C2-YABBY	27	15	31
C2H2	65	43	70
C3H	110	77	129
CAMTA	10	5	6
CCAAT	121	76	116
Coactivator_p15	7	3	5
CPP	16	20	17
CSD	42	2	5
DBP	3	5	8
DDT	14	10	13
E2F-DP	23	11	24
EIL	13	11	9
FAR1	81	10	25
FHA	38	24	23
G2-like	6	2	4
GeBP	9	13	29
GNAT	85	40	60
GRAS	121	69	104
GRF	20	19	32
HB	260	138	210
HMG	44	17	30
HRT	1	1	49
HSF	60	38	1
HTH	1	0	0
Jumonji	41	16	34
LFY	3	2	4
LIM	40	16	31
LOB	88	39	60

Familia	*Glycine max*	*Triticum aestivum*	*Zea maiz*
LUG	5	7	3
MADS	0	0	1
MBF1	3	4	8
MED26	25	19	32
MED6	2	1	3
MED7	3	3	8
mTERF	57	41	41
MYB	304	124	198
MYB-related	284	152	253
NAC	192	172	190
NOT2_3_5	10	13	10
OFP	40	31	43
Orphans	707	475	649
PBF-2-like	10	2	6
PHD	76	49	61
PLATZ	35	21	16
Pseudo_ARR-B	13	14	5
RB	4	4	9
Rcd1-like	4	5	15
RWP-RK	23	15	23
S1Fa-like	4	2	5
SAP	13	12	23
SART-1	4	2	2
SBP	55	29	56
SET	83	48	87
SGT1	2	1	2
Sigma70-like	13	8	14
SNF2	76	52	65

Familia	*Glycine max*	*Triticum aestivum*	*Zea maiz*
SOH1	3	3	3
SRF-TF	189	98	136
SRS	24	6	11
SWI/SNF-BAF60b	35	11	30
TAZ	5	12	14
TCP	55	23	52
TFIIS	2	3	3
Tify	47	23	45
TIG	5	2	6
Top	98	53	85
TRAF	32	88	57
Trihelix	17	10	15
TUB	35	28	38
VOZ	7	2	9
WRKY	196	124	163
Yl1	1	1	1
zf-A20	32	20	14
zf-BED	4	18	12
zf-HD	51	14	26
zf-LSD1	19	13	23
zf-MIZ	5	3	2
zf-NF-X1	3	2	0

Tabla 5.3: Cantidad de FTs por familia y por especie.

Tabla 5.4: Estructura de algunos FTs.

Especie	Secuencia ID	PDB ID	e-value	score
Glycine max	Glyma01g02720.1	2ARO	7e-31	326
Glycine max	Glyma01g05000.1	1L3A	0	555
Glycine max	Glyma01g06150.2	1UT7	0	496
...	...	...	...	...
Triticum aestivum	Os01g40094.1	3RT0	0	1055
Triticum aestivum	Os01g42470.1	2PSW	2e-40	411
Triticum aestivum	Os01g44370.1	2CJJ	2e-15	192
...	...	...	...	...
Zea maiz	GRMZM2G009892_P04	1UT7	3e-35	366
Zea maiz	GRMZM2G013563_P01	2GNQ	0	715
Zea maiz	GRMZM2G025685_P02	2LDU	6e-13	172
...	...	...	...	...

5.2. Anotación

5.2.1. Estructura terciaria

Se identificaron 249 estructuras similares a las del PDB: 126 en soya, 40 en trigo y 126 en maíz. Además, se cuenta con dos archivos de cada estructura, uno contiene la imagen 2D y el otro un modelo 3D que puede ser visto con el Applet Jmol (Figura 5.6). Algunos resultados se presenta en la tabla 5.4. En esta, se muestra la especie, el identificador de la secuencia del FT y el identificador (PDB ID) de la estructura, así como los valores obtenidos con el BLAST.

Por ejemplo, se encontró que el FT de la soya, perteneciente a la familia CCAAT y codificado por el gen Glyma01g02720.1, presenta la estructura de la secuencia 2ARO (Figura 5.1) de la base de datos PDB.

5.2.2. Alineamientos múltiples

Se realizaron en total 314 alineamientos múltiples: 106 para soya, 104 para trigo y 104 para maíz. Estos fueron almacenados en la base de datos y pueden ser visualizados con el Applet Jalview. En la Figura 5.2 se muestra el alineamiento múltiple para el dominio de unión Auxin_resp de la familia ARF en maíz.

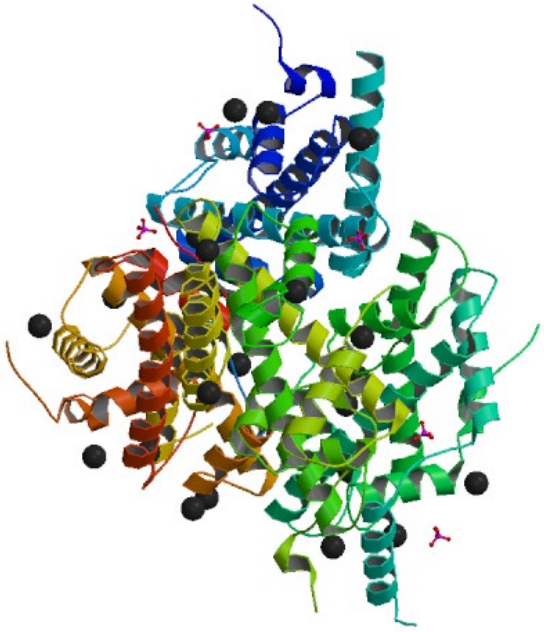

Figura 5.1: Estructura de un FT de la familia CCAAT.

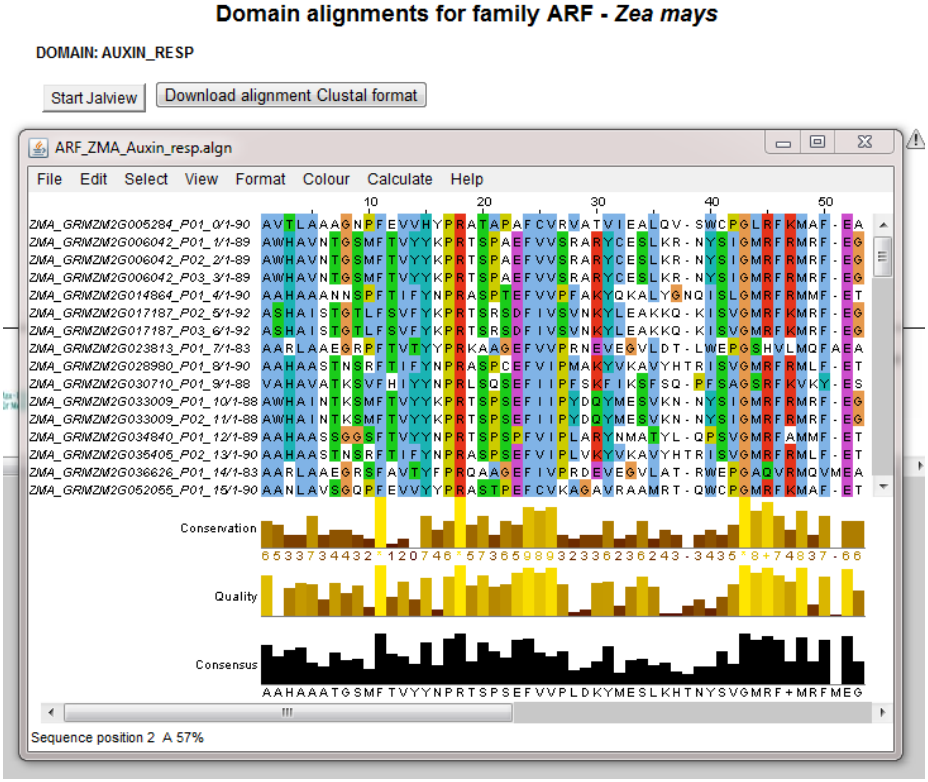

Figura 5.2: Vista de un alineamiento múltiple con el Applet Jalview.

5.2.3. Descripción de familias y referencias a bases de datos externas

Cada familia cuenta con su propia descripción y se tienen en total 269 referencias a artículos científicos. Por ejemplo, en la Figura 5.5 se muestra la descripción de la familia ARF.

En todos los dominios de unión se incorporó un enlace a la base de datos PDB donde se puede encontrar su descripción, imágenes, referencias bibliográficas, etc. En las referencias de literatura se incorporaron enlaces hacia PubMed, la cual contiene los resúmenes y una referencia hacia los artículos completos.

5.2.4. Grupos de ortólogos

OrthoMCL detectó 2347 grupos de ortólogos de FTs. Con las relaciones de ortología predichas por OrthoMCL se incorporaron referencias cruzadas entre genes que facilitaran comparaciones entre las tres especies de la base de datos.

5.3. Implementación de la base de datos

5.3.1. Estructura de la base de datos

La base de datos consta de 14 tablas que se pueden dividir, por la información que contienen, en tres grupos (Figura 5.3):

a) El primer grupo de tablas guarda información relacionada con las especies: nombre científico, clasificación taxonómica, la versión de la anotación de los genomas y el sitio web de donde fueron obtenidos.

b) Este grupo de tablas contiene información relativa a las familias de FTs. En la tabla "tf_families" están los nombres de las familias, su categoría y una breve descripción basada en la literatura, en "papers" se encuentran las referencias a artículos que contienen información relativa a las familias de FTs, en "tf_domains_alignments" se almacenan los alineamientos múltiples y "tf_families_motifs" contiene las reglas de clasificación.

c) En el tercer grupo se encuentran los datos provenientes del análisis de los genomas. En la tabla "tf" se listan los FTs hallados en cada especie, en "Present_domains" se encuentran sus respectivos dominios de unión e información relativa a éstos (su posición en el gen y su e-value), en "Sequences" se tiene la secuencia de ADN y de proteínas para cada gen que codifica para FTs, en "gene_names" está el nombre o nombres de dichos genes, y en "orthologs" están los grupos de genes ortólogos hallados en las diferentes especies de este trabajo.

El esquema completo de la base de datos se encuentra en el Anexo A.

5.4. Interfaz web

5.4.1. Home

La información contenida en FT-MTS pude ser accedida a través de varias vías (Figura 5.4). En la página principal el usuario dispone de un menú de especies que permite obtener conjuntos específicos de familias de FTs (1). Cada familia posee un hipervinculo hacia información más detallada (2), que será explicada en la siguiente sección. El usuario puede comparar sus secuencias con las almacenadas en FT-MTS vía BLAST o PHMMER (3). También, puede buscar información de un gen en particular ingresando el identificador de secuencia (4). Finalmente, el usuario puede descargar las secuencias de cada FT y los alineamientos de cada familia (5).

5.4.2. Anotación a nivel de familia

La información sobre cada familia, contenida en FT-MTS, que se muestra al usuario (Figura 5.5) está dividida en varias secciones:

1. Descripción basada en la revisión de literatura.

2. Los dominios de unión que caracterizan a la familia.

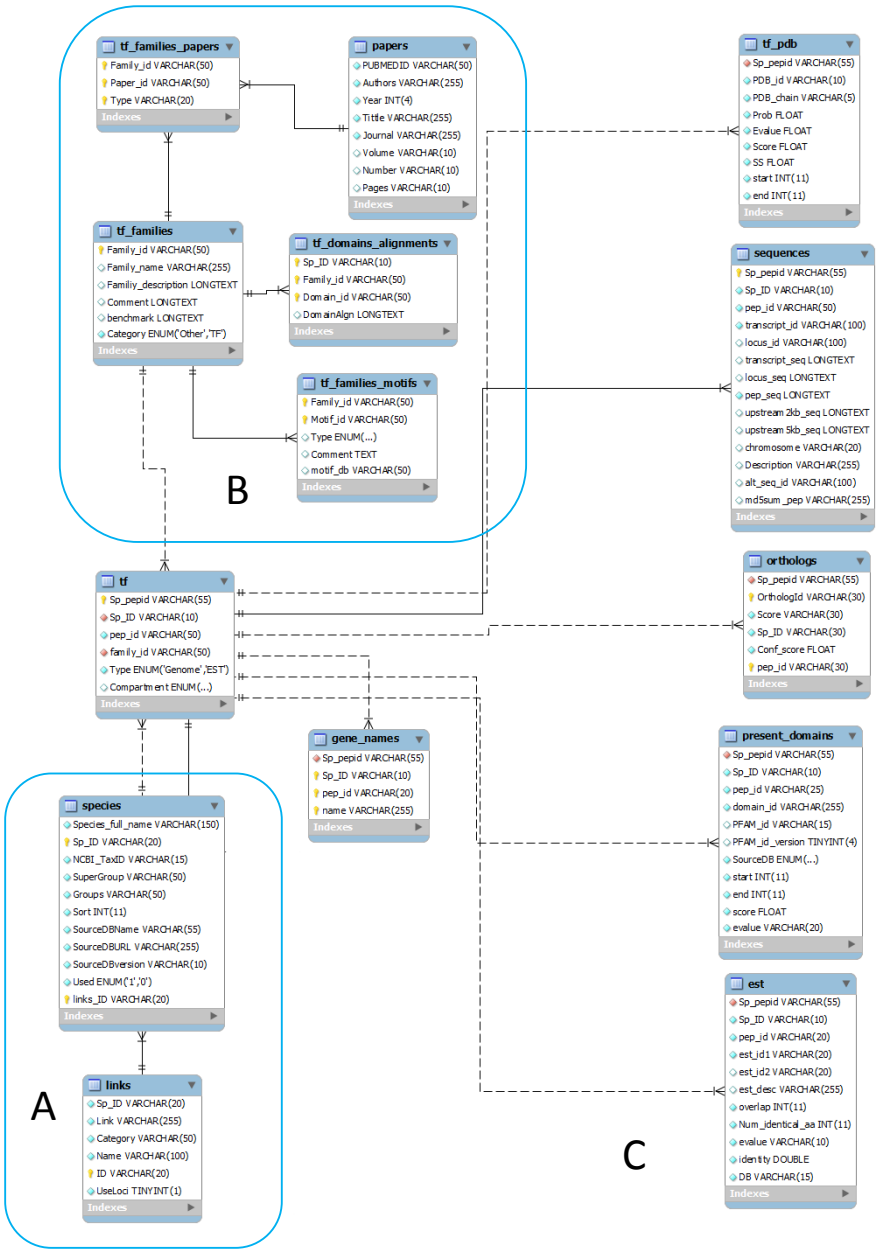

Figura 5.3: Modelo de la base de datos.

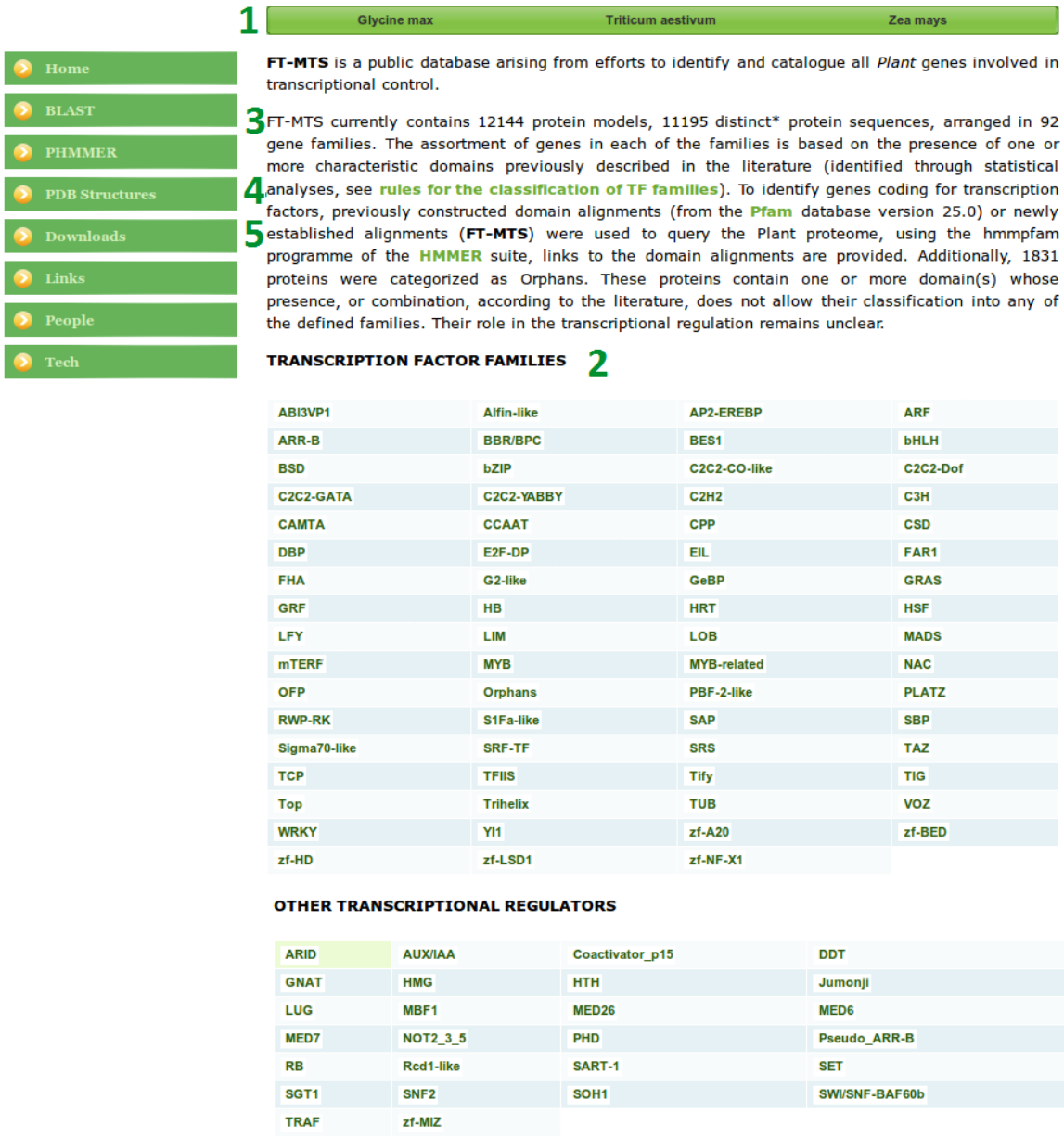

Figura 5.4: Página principal de la la base de datos FT-MTS.

3. Alineamientos múltiples. Éstos pueden ser descargados o visualizados gráficamente a través del Applet Jalview.

4. Referencias a literatura relacionada que incluye los datos relevantes de cada artículo: autor, año, título, nombre de la revista, número y páginas. Adicionalmente, por cada artículo, se incluye un enlace hacia la base de datos externa PubMed donde se encuentra su respectivo resumen.

5. Una tabla donde se listan todos los genes pertenecientes a la especie y familia consultadas. La primera columna contiene el identificador del gen, la segunda muestra su descripción (en caso de estar disponble) y en la última columna están los dominios identificados en dicho gen. Además, para cada dominio se proporciona un enlace a la base de datos de Pfam donde se encuentra una descripción del mismo.

5.4.3. Anotación a nivel de gen

Como en el apartado anterior, la información de cada gen que se muestra al usuario (Figura 5.6) se divide en varias secciones:

1. Datos de identificación que comprenden: especie, identificador de secuencia, familia de FT a la que pertenece y una lista de posibles estructuras terciarias. Cada estructura posee una referencia a la base de datos NCBI donde se muestra una descripción de la misma.

2. Se cuenta con una imagen para cada estructura.

3. De manera complementaria, también es posible visualizar dicha estructura a través del Applet Jmol, que permite hacer ciertas operaciones básicas, como zoom y rotación.

4. Expresión. Se proporciona una referencia a la base de datos NCBI, donde se pueden encontrar las diferentes partes de la planta donde se expresa dicho gen.

5. Arquitectura de dominios. Se listan los dominios encontrados con el paquete HMMER, así como su posición, el bit score y el e-value.

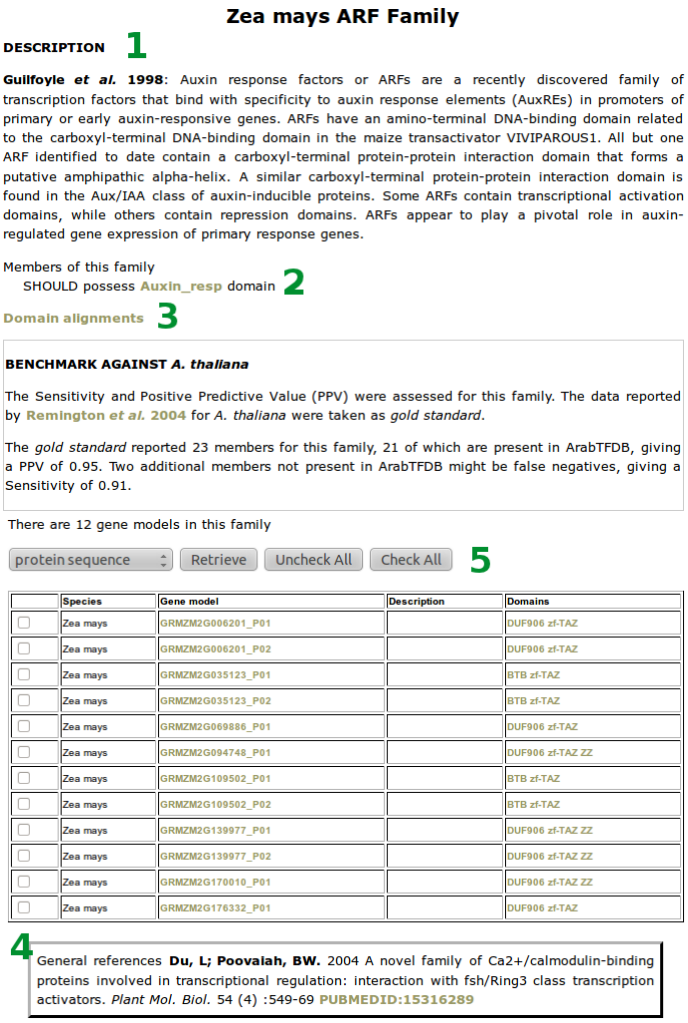

Figura 5.5: Descripción de familias.

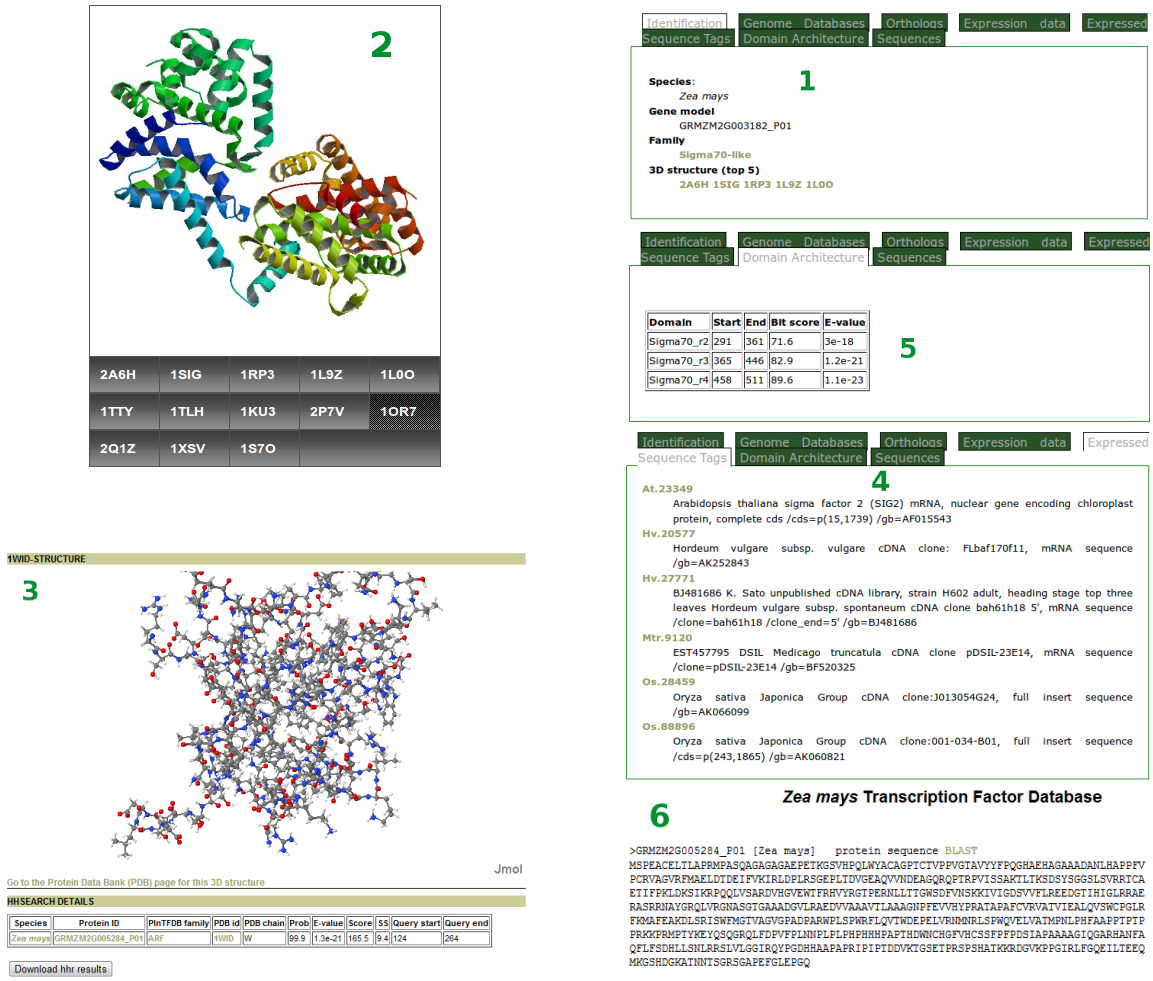

Figura 5.6: Descripción de genes.

6. Secuencias. Las secuencias del gen están disponibles como texto en formato fasta.

7. Una lista de genes ortólogos entre las tres especies dentro de la misma base de datos FT-MTS.

5.4.4. Búsquedas en la base de datos local

Un componente fundamental para cualquier base de datos es proporcionar a los usuarios herramientas que permitan extraer información de ella. En la página web se dispone de tres opciones.

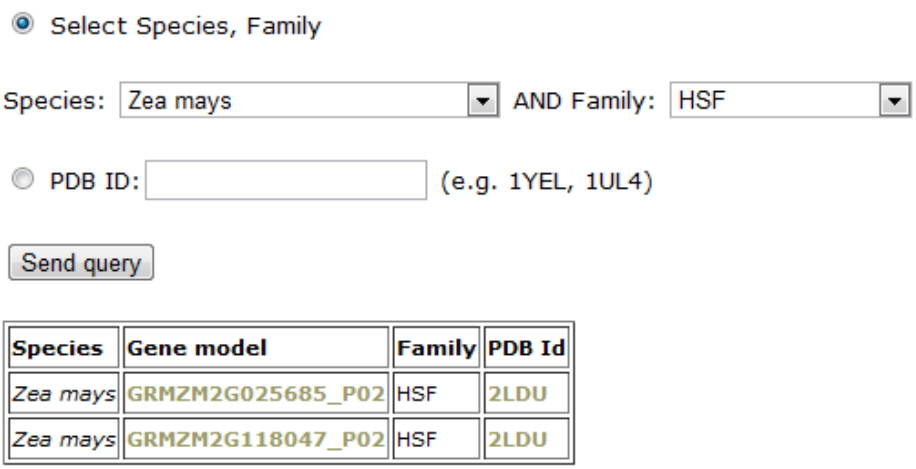

Figura 5.7: Interfaz para realizar búsquedas.

- **Búsqueda por espece y/o familia o por PDB Id**. El usuario puede optar por seleccionar una especie, una famila, o ambos y en seguida se despliega una tabla que muestra los Ids de las secuencias que coincidan como, por ejemplo, las secuencias que codifican para la familia HSF de la especie *Zea maiz* (Figura 5.7). También se tiene la opción de búscar a partir de un Id del PDB.

- **BLAST**. El usuario puede comparar sus secuencias con las almacenadas en la base de datos FT-MTS utilizando el software BLAST. Es posible configurar distintos parámetros como, por ejemplo, *blastp* si la secuencia problema es de proteínas o *blastx* si es de bases; la *matriz de sustitución* (se proporcionan las matrices estándar: Blosum62, Blosum45, Blosum80, PAM30 y PAM70), la presencia de huecos en el alineamiento y el *e-value* (Figura 5.8).

- **PHMMER**. Durante casi 20 años el paquete BLAST ha sido uno de los pilares en Bioinformática. Sin embargo, las bases teóricas del análisis de secuencias han mejorado bastante desde entonces, principalmente gracias métodos probabilísticos de modelación como los modelos de Cadenas Ocultas de Markov (HMM). La implementación de dichos modelos no es nueva pero el principal problema que presentan es un elevado costo computacional que hace inviable su implementación fuera de servidores de alto rendimiento. Recientemente, el paquete HMMER v3 ha destacado por realizar una implementación que es tan rápida como el BLAST pero además proporciona resul-

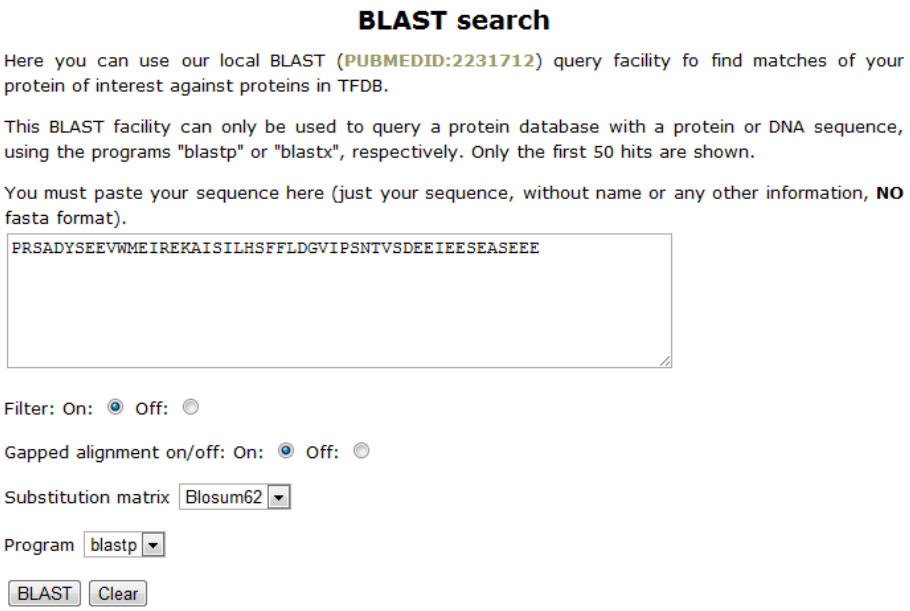

Figura 5.8: Comparación de secuencias utilizando BLAST.

tados más precisos. En la página Web se ha incorporado dicho paquete para que los usuarios puedan aprovechar sus características. Al igual que con el BLAST, también se pueden configurar distintos parámetros como las penalizaciones por huecos, la matriz de sustitución, y se puede elegir entre calcular los resultados utilizando el e-value o el bit-score (Para más detalle sobre el uso de los parámetros consultar el manual en: http://hmmer.janelia.org/)

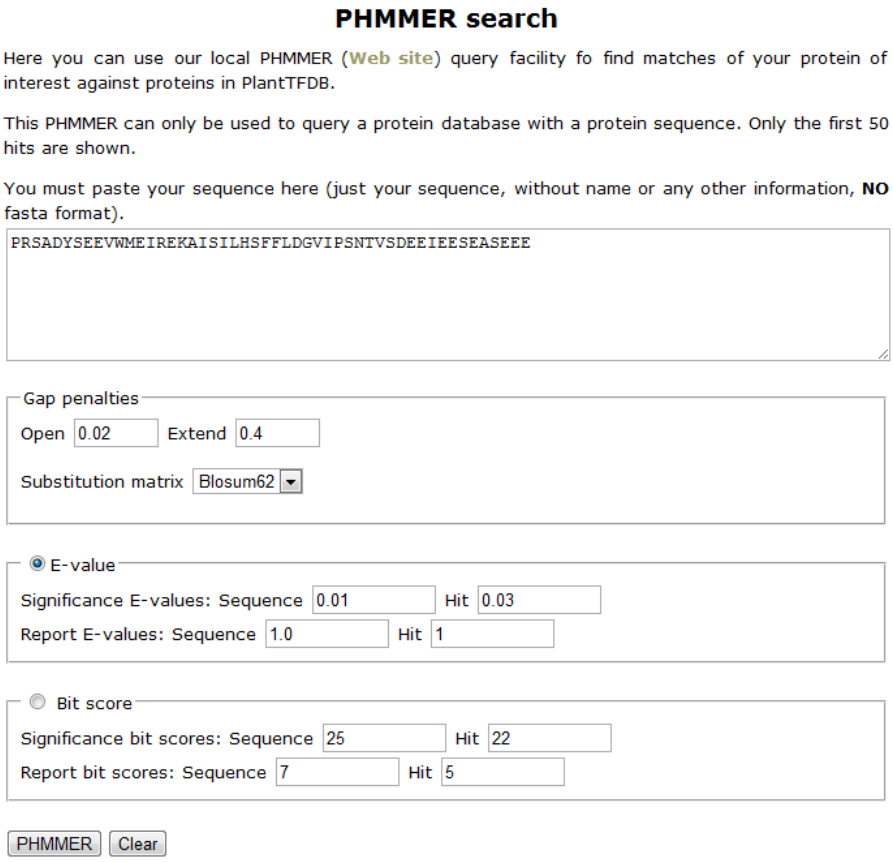

Figura 5.9: Comparación de secuencias utilizando PHMMER.

Capítulo 6

Conclusiones

FT-MTS es una base de datos de factores de transcripción, que fueron identificados con técnicas bioinformáticas, de los genomas de *Glycine max*, *Triticum aestivum* y *Zea maiz*. Además, contiene una anotación detallada a nivel de familia y de gen, incluyendo referencias a bases de datos externas. Cuenta también con una interfaz web que permite el acceso libre a los datos. Se espera que este trabajo sea una plataforma que pueda ser usada por la comunidad científica como un primer recurso en la identificación de genes que codifican FTs. Y así, facilitar el estudio de los procesos de regulación en la expresión génica en los cultivos durante las diferentes etapas fisiológicas y de desarrollo.

Debido a que la información utilizada para la construcción de la base de datos se encuentra en constante cambio (se liberan nuevas versiones de los genomas, se dispone de nuevos perfiles HMM o se actualizan los existentes, se describen nuevas familias en la literatura, etc.) es necesario actualizar constantemente los análisis, lo que permitirá contar con información más completa y precisa.

Bibliografía

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K. y Watson, J. D. (2004). *Biología molecular de la célula*. Omega.

Antano, J. (2010). *Catalyst 5.8. The Perl MVC Framework*. Packt Publishing, primera edición.

Baxevanis, A. D. y Ouellette, F. (2001). *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. JOHN WILEY and SONS, segunda edición.

Behnam, B., Kikuchi1, A., Celebi-Toprak, F., Yamanaka, S., Kasuga, M., Yamaguchi-Shinozaki, K. y Watanabe, K. N. (2006). The Arabidopsis DREB1A gene driven by the stress-inducible rd29A promoter increases salt-stress tolerance in proportion to its copy number in tetrasomic tetraploid potato (*Solanum tuberosum*). *Plant Biotechnology*, 23, 169–177.

Chen, F., Mackey, A. J., Vermunt, J. K. y Roos, D. S. (2007). Assessing Performance of Orthology Detection Strategies Applied to Eukaryotic Genomes. *PLoS ONE*, 2, 4, e383.

Clifton, S. W., Minx, P., Fauron, C. M., Gibson, M., Allen, J. O., Sun, H., Thompson, M., Barbazuk, B., Kanuganti, S., Tayloe, C., amd Richard K Wilson, L. M. y Newton, K. J. (2004). Sequence and Comparative Analysis of the Maize NB Mitochondrial Genome. *Plant Physiology*, 136, 3, 3486–3503.

Durbin, R., Eddy, S., Krogh, A. y Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Eddy, S. R. (1998). Profile hidden Markov models. *Bioinformatics*, 14, 9, 755–63.

Eddy, S. R. (2004). What is a hidden Markov model? *Nature Biotechnology*, 32, 1315–16.

Enright, A. J., Dongen, S. V. y Ouzounis, C. A. (2002). An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30, 7, 1575–84.

Finn, R. D., Clements, J. y Eddy, S. R. (2011). HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research*, 39, W29–W37.

Finn, R. D., Mistry, J., Tate, J., Coggil, P., Heger, A., Polligton, J. E., Gavin, O. L., Gunasekaran, P., Ceric, G., Forslund, K., Holm, L., Sonnhammer, E. L., Eddy, S. R. y Beteman, A. (2010). The Pfam protein families database. *Nucleic Acids Research*, 38, 1, D211–D222.

Fitch, W. M. (2000). Homology a personal view on some of the problems. *Trends in Genetics*, 16, 5, 227–31.

Gogarten, J. P. y Olendzenski, L. (1999). Orthologs, parologs and genome comparisons. *Current Opinion in Genetics & Development*, 9, 6, 630–6.

Guo, A.-Y., Chen, X., Gao, G., Zhang, H., Zhu, Q.-H. y Liu, X.-C. (2008). PlantTFDB: a comprehensive plant trascription factor database. *Nucleic Acids Research*, 36, D1114–D1117.

Hollich, V., Storm, C. E. V. y Sonnhammer, E. L. L. (2002). OrthoGUI, graphical presentation of Orthostrapper results. *Bioinformatics*, 18, 9, 1272–73.

Hussain, S. S., Kayani, M. A. y Amjad, M. (2011). Transcription factors as tools to engineer enhanced drought stress tolerance in plants. *Biotechnology Progress*, 27, 2, 297–306.

Kanehisa, M. y Bork, P. (2003). Bioinformatics in the post-sequence era. *Nature Genetics*, 33, 305–10.

Kasuga, M., Miura1, S., Shinozaki, K. y Yamaguchi-Shinozaki, K. (2004). A Combination of the Arabidopsis DREB1A Gene and Stress-Inducible rd29A Promoter Improved

Drought- and Low-Temperature Stress Tolerance in Tobacco by Gene Transfer. *Plant & Cell Physiology*, 45, 2, 346–350.

Katoh, K. y Toh, H. (2008). Recent devolepments in the MAFFT multiple sequence alignment program. *Briefings in Bioinformatics*, 9, 4, 286–298.

Khan, M. S. (2011). The role of DREB transcription factors in abiotic stress tolerance of plants. *Biotechnology & Biotechnological Equipment*, 25, 2433–42.

Kummerfeld, S. K. y Teichmann, S. A. (2007). DBD: a transcription factor prediction database. *Plant Physiology*, 143, 1452–1466.

Latchman, D. S. (1997). Transcription factors: an overview. *The International Journal of Biochemistry & Cell Biology*, 29, 12, 1305–12.

Lehninger, A. L., Cox, M. M. y Nelson, D. L. (2006). *Principios de Bioquimica*. Omega.

Lewin, B. (2004). *Genes VII*. Pearson Prentice Hall.

Li, L., Jr., C. J. S. y Roos, D. S. (2003). OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes. *Genome Research*, 13, 2178–2189.

Lodish, H., Berk, A., Matsudaira, P., Kaiser, C. A., Krieger, M., Scott, M. P., Zipursky, L. y Darnell, J. (2003). *Molecular Cell Biology*. W H Freeman, quinta edición.

Madigan, M. T., Martinko, J. N. y Parker, J. (2004). *Brock Biología de los Microorganismos*. Pearson Prentice Hall, décima edición.

Maier, R. M., Neckermann, K., Igloi, G. L. y Kössel, H. (1995). Complete Sequence of the Maize Chloroplast Genome: Gene Content, Hotspots of Divergence and Fine Tuning of Genetic Information by Transcript Editing. *Molecular Biology*, 251, 5, 614–628.

Mathur, P., Devi, M., R, S., Yamaguchi-Shinozaki, V, V. y K, S. K. (2004). Evaluation of transgenic groundnut lines under water limited conditions. *International Arachis Newletter*, 24, 33–35.

Mochida, K., Yoshida, T., Sakurai, T., Yamaguchi-Shinozaki, K., Shinozaki, K. y Tran, L.-S. P. (2009). LegumeTFDB: An integrative database of *Glycine max*, *Lotus japonicus* and *Medicago truncatula* transcription factors. *Bioinformatics*, 26, 2, 290–291.

Mount, D. W. (2004). *Bioinformatics: Sequence and Genome Analysis*. JOHN WILEY and SONS, segunda edición.

Ogihara, Y., Isono, K., Kojima, T., Endo, A., Hanaoka, M., Shiina, T., Terachi, T., Utsugi, S., Murata, M., Mori, N., Takumi, S., Ikeo, K., Gojobori, T., Murai, R., Murai, K., Matsuoka, Y., Ohnishi, Y., Tajiri, H. y Tsunewaki, K. (2001). Structural features of a wheat plastome as revealed by complete sequencing of chloroplast DNA. *Molecular Genetics and Genomics*, 266, 5, 740–746.

Oh, S.-J., Song, S. I., Kim, Y. S., Jang, H.-J., Kim, S. Y., Kim, M., Kim, Y.-K., Nahm, B. H. y Kim, J.-K. (2005). Arabidopsis CBF3/DREB1A and ABF3 in Transgenic Rice Increased Tolerance to Abiotic Stress without Stunting Growth. *Plant Physiology*, 138, 341–351.

Pérez-Rodríguez, P., Riano-Pachón, D. M., Correa, L. G. G., Resing, S. A., Kersten, B. y Mueller-Roeber, B. (2009). PlnTFDB: updated content and new features of the plant transcription factor database. *Nucleic Acids Research*, 38, 1, D822–D827.

Petsko, G. A. y Ringe, D. (2004). *Protein structure and function*. New Science Press Ltd., primera edición.

Polanksi, A. y Kimmel, M. (2007). *Bioinformatics*. Springer, primera edición.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, 2, 257–286.

Reece, R. (2004). *Analysis of Genes and Genomes*. JOHN WILEY and SONS, primera edición.

Remm, M., Storm, C. E. V. y Sonnhammer, E. L. L. (2001). Automatic Clustering of Orthologs and In-paralogs from Pairwise Species Comparisons. *J. Mol. Biol*, 314, 5, 1041–1052.

Riaño-Pachón, D. M., Ruzicic, S., Dreyer, I. y Mueller-Roeber, B. (2007). PlnTFDB: updated content and new features of the plant transcription factor database. *BMC Bioinformatics*, 8, D966–D969.

Riechmann, J. L., Heard, J., Martin, G., Reuber, L., Jiang, C. Z., Keddie, J., Adam, L., Pineda, O., Ratcliffe, O. J., Samaha, R. R., Creelman, R., Pilgrim, M., Broun, P., Zhang, J. Z., Ghandehari, D., Sherman, B. K. y Yu, G. L. (2000). *Arabidopsis* Transcription Factors: Genome-Wide Comparative Analysis Among Eukaryotes. *Science*, 290, 2105–2110.

Saski, C., Lee, S.-B., Daniell, H., Wood, T. C., Tomkins, J., Kim, H.-G. y Jansen, R. K. (2005). Complete chloroplast genome sequence of *Glycine max* and comparative analyses with other legume genomes. *Plant Molecular Biology*, 59, 309–322.

Schmutz, J., Cannon, S. B., Schlueter, J., Ma, J., Mitros, T., Nelson, W., Hyten, D. L., Song, Q., Thelen, J. J., Cheng, J., Xu, D., Hellsten, U., May, G. D., Yu, Y., Sakurai, T., Umezawa, T., Bhattacharyya, M. K., Sandhu, D., Valliyodan, B., Lindquist, E., Peto, M., Grant, D., Shu, S., Goodstein, D., Barry, K., Futrell-Griggs, M., Abernathy, B., Du, J., Tian, Z., Zhu, L., Gill, N., Joshi, T., Libault, M., Sethuraman, A., Zhang, X.-C., Shinozaki, K., Nguyen, H. T., Wing, R. A., Cregan, P., Specht, J., Grimwood, J., Rokhsar, D., Stacey, G., Shoemaker, R. C. y Jackson, S. A. (2010). Genome sequence of the palaeopolyploid soybean. *Nature*, 463, 178–183.

Schnable, P. S., Ware, D., Fulton, R. S., Stein, J. C., Wei, F., Pasternak, S., Liang, C., Zhang, J., Fulton, L., Graves, T. A., Minx, P., Reily, A. D., Courtney, L., Kruchowski, S. S., Tomlinson, C., Strong, C., Delehaunty, K., Fronick, C., Courtney, B., Rock, S. M., Belter, E., Du, F., Kim, K., Abbott, R. M., Cotton, M., Levy, A., Marchetto, P., Ochoa, K., Jackson, S. M., Gillam, B., Chen, W., Yan, L., Higginbotham, J., Cardenas, M., Waligorski, J., Applebaum, E., Phelps, L., Falcone, J., Kanchi, K., Thane, T., Scimone, A., Thane, N., Henke, J., Wang, T. y Ruppert, J. (2009). The B73 Maize Genome: Complexity, Diversity, and Dynamics. *Molecular Biology*, 326, 5956, 1112–1115.

Tiessen, A. (2009). *Fundamentos y metodologías innovadoras para el mejoramiento genético del maíz*. Fundación Ciencia Activa, primera edición.

Tran, L.-S. P., Nakashima, K., Shinozaki, K. y Yamaguchi-Shinozak, K. (2007). Plant gene networks in osmotic stress response: from genes to regulatory networks. *Methods Enzimol*, 428, 109–128.

Tran, L. S. P. y Nguyen, H. T. (2009). Nitrogen fixation in crop production. En W. D. Emerich y H. Krishnan, eds., *Future biotechnology of legumes*, 265–308. The American Society of Agronomy, Crop Science Society of America and Soil Science Society of America. Madison, WI, USA.

Tran, L. S. P., Quach, T. N., Guttikonda, S. K., Aldrich, D. L., Kumar, R., Neelakandan, A. y Nquyen, H. T. (2009). Molecular characterization of stress-inducible GmNAC genes in soybean. *Molecular Genetics and Genomics*, 281, 647–664.

Wall, D. P., Fraser, H. B. y Hirsh, A. E. (2003). Detecting putative orthologs. *Bioinformatics*, 19, 13, 1710–11.

Waterhouse, A. M., Procter, J. B., Martin, D. M. A., Clamp, M. y Barton, G. J. (2009). Jalview Version 2 - a multiple sequence alignment editor and analysis workbench. *Bioinformatics*, 25, 9, 1189–91.

Yamaguchi-Shinozakia, K. y Shinozaki, K. (2005). Organization of cis-acting regulatory elements in osmotic- and cold-stress-responsive promoters. *Trends in Plant Science*, 10, 2, 84–94.

Yilmaz, A., Nishiyama, M. Y., Garcia-Fuentes, B., Mendes-Souza, G., Janies, D., Gray, J. y Grotewold, E. (2009). GRASSIUS: A Platform for Comparative Regulatory Genomics across the Grasses. *Plant Physiology*, 149, 171–180.

Zhang, H., Jin, J., Tang, L., Zhao, Y., Gu, X., Gao, G. y Luo, J. (2011). PlantTFDB 2.0: update and improvement of the comprehensive plant transcription factor database. *Nucleic Acids Research*, 39, D1114–D1117.

Zmasek, C. M. y Eddy, S. R. (2002). RIO: Analyzing proteomes by automated phylogenomics using resampled inference of orthologs. *BMC Bioinformatics*, 3, 3:14.

Zubay, G. L., Parson, W. W. y Vance, D. E. (1995). *Principles of Biochemistry*. Wm. C. Brown, primera edición.

Anexos A

Programas y material complementario

Se incluye un CD con el programa usado para la clasificación de FTs, las reglas de clasificación, el esquema de la base de datos y los perfiles HMM.

A.1. Programa y reglas para la clasificación de FTs

A.1.1. Clasificación.pl

```
#!/usr/bin/perl
use strict;
use warnings;
use Getopt::Long;
my $version='3.1';
my $pfam_tbl=";
my $domain_list=";
my $species=";
my $family_out="";
my $license=";
my $rfile=";
```

```
my $help='';
GetOptions(
'pfam=s'=>\$pfam_tbl,
'species|s=s'=>\$species,
óut|o=s'=>\$family_out,
'rules|r=s'=>\$rfile,
'license|l'=>\$license,
'help|h|?'=>\$help
);
if ($help){
&usage();
exit(1); }
if ($license){
&license();
exit(1); }
if(!$pfam_tbl){
&usage();
print STDERR "You must provide the name of the PFAM domtblout file\n";
exit(1); }
if (!$family_out){
&usage();
print STDERR "You must provide the name of an output file to the family classification\n";
exit(1) }
if (!-s $rfile){
&usage();
print STDERR "Cannot find rules file [$rfile]\n\n";
exit(1) }
#opens familiy_out for writing
open OUTFAM, ">$family_out" or die "Cannot open $family_out for writing\n";
#open ftam_tbl (List of PFAM hits in tabular format) for reading
```

```
open PFAMTBL, "$pfam_tbl" or die "Cannot open file $pfam_tbl\n";
#It returns the entire file
my @tbl=<PFAMTBL>;
close PFAMTBL;
my @all_dom=();
open RULES, "$rfile" or die "Cannot open file $rfile\n";
my @rules=<RULES>;
my @rules1=@rules;
close RULES;
#Obtains all domains (required, forbidden) and also check
#the number of fields in each rule
my $semicolon_counter;
foreach my $item (@rules1) {
$semicolon_counter=0;
chomp($item);
$semicolon_counter ++ while $item=~/;/g;
if($semicolon_counter!=3) {
print "$item\n";
die "INCORRECT number of fields \nExpecting 3 fields Got: $semicolon_counter fields
\nEach field must be ended with ;\n"; }
$item=substr($item,index($item,";")+1,length($item)-index($item,";"));
$item=~s/;+|=/,/g;
$item=~s/:eq[0-9]+|:gt[0-9]+|:lt[0-9]+//g;
push @all_dom, split(',',$item); }
undef my %swx;
@swx{@all_dom} = ();
@all_dom = sort keys %swx;
#Some hashes
my %domains_per_gene;
my %gene_family;
```

```
my %evalues_per_gene;
my %evalues_gene_family;
foreach my $tbl(@tbl) {
next if $tbl=~/^#/;
next if $tbl eq '';
chomp $tbl; #delete \n from $tbl
#Each line in the PFAM results fiel shoudl havv the followng fields:
# — full sequence — ———— this domain ————- hmm coord ali coord env coord
# target name accession tlen query name accession qlen E-value score bias # of c-Evalue
i-Evalue score bias from to from to from to acc description of target
#———————- ———- —— ———————— ———- —— ——— —— —— —— —— ———
——— —— —— —— —— —— —— —— —— —- ——————
# my ($gen_id,$domain,$start,$end,$score,$evalue)=split(/\t/,$tbl);
my ($domain,undef,$domainlength,$gen_id,undef, $querylength,undef,undef,undef,undef,undef,undef,
$evalue,$score,undef,$hitalgnstart,$hitalgnstop, $start,$stop,undef,undef)=split(/\s+/,$tbl);
#next if($domainlength != ($hitalgnstop-$hitalgnstart+1));
#print $gen_id."\t$domain\n";
next unless defined $domain;
foreach my $dom(@all_dom){
chomp $dom; #delete \n from dom
if($domain eq $dom){
@{$gene_family{$gen_id}}=();
@{$evalues_gene_family{$gen_id}}=();
$domains_per_gene{$gen_id}{$domain}+=1;
$evalues_per_gene{$gen_id}{$domain}=$evalue; } } }
my @required_domains=();
my @or_required_domains=();
my @forbidden_domains=();
my @eqt_gt=();
my @eqt_gt_operator_operand=();
```

```
my $prod_required_domains;
my $prod_or_required_domains;
my $prod_forbidden_domains;
my $prod;
my $check_sum;
foreach my $gene(keys %domains_per_gene) {
foreach my $item (@rules) {
chomp($item);
my ($TFF, $required_domains, $forbidden_domains)=split(";",$item);
my @required_domains=split(",",$required_domains);
my @forbidden_domains=split(",",$forbidden_domains);
#Required domains
my $prod_required_domains=1;
my $sum_evalues=0;
foreach my $item_required (@required_domains) {
@or_required_domains=split("=",$item_required);
$prod_or_required_domains=0;
foreach my $item_or_required (@or_required_domains) {
#check requiered domains, we have three cases a) at least one (default), but we can specify
this by item_requiered:gt0
# b) exactly some_number, specify this by item_requiered:eqtsome_number, example
Myb_DNA-binding:eqt1 for MYB-related TFF
# c) at least some_other_number + 1, specify this by item_requiered:gtsome_number,
example Myb_DNA-binding:gt1 for MYB TFF
# d) at most some_other_number - 1, specify this by item_requiered:ltsome_number,
example AP2:lt3 for AP2-EREBP
if(!(($item_or_required=~/\:gt/) || ($item_or_required=~/\:eq/) || ($item_or_required=~/\:lt/)))
{
if(exists($domains_per_gene{$gene}{$item_or_required})) {
$prod_or_required_domains=1;
```

```
$sum_evalues+=$evalues_per_gene{$gene}{$item_or_required}; } } else {
my($case,$copynumber);
@eqt_gt=split(":",$item_or_required);
$item_or_required=$eqt_gt[0];
if($eqt_gt[1]=~/^(eq|gt|lt)(\d+)$/i){
$case=$1;
$copynumber=$2; } else{
die "FATAL:Number and case not allowed [@eqt_gt]!!!!!!" }
if(lc($case) eq éq') {
if(exists($domains_per_gene{$gene}{$item_or_required})) {
if($domains_per_gene{$gene}{$item_or_required}==$copynumber) {
$prod_or_required_domains=1;
$sum_evalues+=$evalues_per_gene{$gene}{$item_or_required}; } } } if(lc($case) eq 'gt')
{
if(exists($domains_per_gene{$gene}{$item_or_required})) {
if($domains_per_gene{$gene}{$item_or_required}>$copynumber) {
$prod_or_required_domains=1;
$sum_evalues+=$evalues_per_gene{$gene}{$item_or_required}; } } } if(lc($case) eq 'lt')
{
if(exists($domains_per_gene{$gene}{$item_or_required})) {
if($domains_per_gene{$gene}{$item_or_required}<$copynumber) {
$prod_or_required_domains=1;
$sum_evalues+=$evalues_per_gene{$gene}{$item_or_required}; } } } } } if($prod_or_required_domains=
{$prod_required_domains=0;} }
#Forbidden domains
$prod_forbidden_domains=1;
foreach my $item_forbidden (@forbidden_domains) {
if(exists($domains_per_gene{$gene}{$item_forbidden})) {$prod_forbidden_domains=0;}
}
$prod=($prod_required_domains)*($prod_forbidden_domains);
```

```
if($prod==1) {
push @{$gene_family{$gene}}, $TFF;
push @{$evalues_gene_family{$gene}},$sum_evalues; } } }
my %saw;
undef %saw;
my @genes = grep(!$saw{$_}++, (keys %domains_per_gene));
print STDERR "There are: ".scalar(keys %domains_per_gene)." genes with TF domains\n\n";
print STDERR scalar(keys %gene_family)." genes go into analysis from ".scalar(@genes)."\n\n";
print OUTFAM "Gene\tFamily\n";
foreach my $gene(keys %gene_family){
if (@{$gene_family{$gene}}>1) {
print STDERR "Gene $gene in more than two famlies @{$gene_family{$gene}} check
that\n";
#FIXME:Assuming that only two families are in conflict
my $fam1=(@{$gene_family{$gene}})[0];
my $fam2=(@{$gene_family{$gene}})[1];
if(!(($fam1=~/TFF/ && $fam2=~/TFF/) || ($fam1=~/OTR/ && $fam2=~/OTR/))) {
print STDERR "TFF or OTR?\t";
if($fam1=~/TFF/) {
print STDERR "$gene Assigned to $fam1\n";
if($fam1=~/_/) {print OUTFAM "$gene\t",(split("_",(split(":",$fam1))[0]))[1],"\n";}
else {print OUTFAM "$gene\t",(split(":",$fam1))[0],"\n";} } else {
print STDERR "$gene Assigned to $fam2\n";
if($fam2=~/_/) {print OUTFAM "$gene\t",(split("_",(split(":",$fam2))[0]))[1],"\n";}
else {print OUTFAM "$gene\t",(split(":",$fam2))[0],"\n";} } } else {
print STDERR "Using Evalues\n";
my $val1=(@{$evalues_gene_family{$gene}})[0];
my $val2=(@{$evalues_gene_family{$gene}})[1];
if($val1<$val2) {
print STDERR "$gene Assigned to $fam1\n";
```

```
if($fam1=~/_/) {print OUTFAM "$gene\t" ,(split("_",(split(" :",$fam1))[0]))[1],"\n";}
else {print OUTFAM "$gene\t",(split(":",$fam1))[0],"\n";} } else {
print STDERR "$gene Assigned to $fam2\n";
if($fam2=~/_/) {print OUTFAM "$gene\t",(split("_",(split(":",$fam2))[0]))[1],"\n";}
else {print OUTFAM "$gene\t",(split(":",$fam2))[0],"\n";} } } } elsif(@{$gene_family{$gene}}
== 1) {
my $gen_fam=(split(":",${$gene_family{$gene}}[0]))[0];
if($gen_fam=~/CHECK/) {print "$gene has additional domains, not considered in the optional domains and can not be classified $gen_fam\n";}
print OUTFAM "$gene\t",(split(":",${$gene_family{$gene}}[0]))[0],"\n";
} else {
print OUTFAM "$gene\tOrphans\n";
} }
sub usage{
print STDERR "$0 version $version, Copyright (C) 2005-2007 Diego Mauricio Riaño Pachón\n";
print STDERR "$0 version $version, Copyright (C) 2009 Paulino Perez\n";
print STDERR "$0 comes with ABSOLUTELY NO WARRANTY; for details type '$0 -l'.\n";
print STDERR "This is free software, and you are welcome to redistribute it under certain conditions;\n";
print STDERR "type '$0 -l'for details.\n";
print STDERR <<EOF;
NAME
$0 assign proteins into TF families based on PFAM hits.
USAGE
$0 -f file.pfam-tbl -r rules.txt
OPTIONS
–pfam List of PFAM hits in tabular format. REQUIRED
–out -o Output file where the classification into TF families will be stored. REQUIRED
–rules -r Rules file REQUIRED
```

–help, -h This help.

–license, -l License.

EOF }

sub license{

print STDERR <<EOF;

Copyright (C) 2005,2006,2007,2008,2009,2010 Diego Mauricio Riaño Pachón

e-mail: diriano\@gmail.com

Coypright (c) 2009 Paulino Perez Rodriguez

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA. EOF

exit; }

A.1.2. Reglas.txt

ABI3VP1:TFF;B3;AP2,Auxin_resp;

Alfin-like:TFF;Alfin-Like;DDT,Homeobox,JmjC,JmjN;

AP2-EREBP:TFF;AP2:lt3;;

ARF:TFF;Auxin_resp;;

ARID:OTR;ARID;;

ARR-B:TFF;Response_reg,Myb_DNA-binding=G2-Like;CCT;

AUX/IAA:OTR;AUX_IAA;Auxin_resp,B3;

BAF1:TFF;BAF1_ABF1;;

BBR/BPC:TFF;GAGA_bind;;

BES1:TFF;DUF822;;

bHLH:TFF;HLH;;

bHSH:TFF;TF_AP-2;;

Bromodomain:TFF;Bromodomain,Bromo_TP;;

BSD:TFF;BSD;;

bZIP:TFF;bZIP_1=bZIP_2=bZIP_Maf;HLH;

C2C2-CO-like:TFF;CCT,zf-B_box;PLATZ,GATA;

C2C2-Dof:TFF;zf-Dof;;

C2C2-GATA:TFF;GATA;;

C2C2-YABBY:TFF;YABBY;;

C2H2:TFF;zf-C2H2;PHD,JmjC,JmjN;

C3H:TFF;zf-CCCH;zf-C2H2,PHD,SNF2_N;

CAMTA:TFF;CG-1,IQ;;

CCAAT:TFF;CBFB_NFYA=NF-YC=NF-YB= CCAAT-Dr1=CBFD_NFYB_HMF;;

CENP-B:OTR;CENP-B_N;;

Coactivator_p15:OTR;PC4;;

Copper_fist:TFF;Copper-fist;;

CP2:TFF;CP2;;

CPP:TFF;CXC;;

CSD:TFF;CSD;;

CUT:TFF;CUT;;

DBP:TFF;PP2C,DNC;;

DDT:OTR;DDT;Homeobox,Alfin-Like;

E1A-like:TFF;Adeno_E1A;;

E2:TFF;PPV_E2_N;;

E2F-DP:TFF;E2F_TDP;;

EIL:TFF;EIN3;;

Ets-type:TFF;SAM_PNT=Ets;;

FAR1:TFF;FAR1;;

FHA:TFF;FHA;;

Fork_head:TFF;Fork_head;;

G2-like:TFF;G2-Like;Response_reg;

GATA-N:TFF;GATA-N;;

GeBP:TFF;DUF573;;

GNAT:OTR;Acetyltransf_1;PHD;

GntR:OTR;GntR;;

Grainyhead:TFF;CP2;;

GRAS:TFF;GRAS;;

GRF:TFF;QLQ,WRC;;

HB:TFF;Homeobox=KNOX1=KNOX2;;

HMG:OTR;HMG_box;ARID,YABBY;

HRT:TFF;HRT;;

HSF:TFF;HSF_DNA-bind;;

HTH:OTR;HTH_3=HTH_5=HTH_10=HTH_6=HTH_11=HTH_13;;

IRF:TFF;IRF;;

MED26:OTR;Med26;;

Jumonji:OTR;JmjC=JmjN;ARID,zf-C2H2,Alfin-Like;

KILA:TFF;KilA-N;;

LFY:TFF;FLO_LFY;;

LIM:TFF;LIM;;

LOB:TFF;DUF260;;

LUG:OTR;LUFS;;

luxR:TFF;GerE;;

MADS:TFF;SRF-TF;;

MED6:OTR;Med6;;

MED7:OTR;Med7;;

MBF1:OTR;MBF1;;

mTERF:TFF;mTERF;;

MYB:TFF;Myb_DNA-binding:gt1;Response_reg,ARID,SNF2_N;

MYB-related:TFF;Myb_DNA-binding:eq1; Response_reg,ARID,SNF2_N,G2-Like,Trihelix;

NAC:TFF;NAM;;

NDT80:OTR;NDT80_PhoG;;

NF-1:TFF;MH1=CTF_NFI;;

NOT2_3_5:OTR;NOT2_3_5;;

NOZZLE:TFF;NOZZLE;;

OFP:TFF;DUF623;;

P53:TFF;P53;;

PBF-2-like:TFF;Whirly;;

PCG:TFF;Hormone_recep;;

PcG-EZ:OTR;SANTA,SET;;

PHD:OTR;PHD;Alfin-Like,DDT,Homeobox,JmjC ,JmjN,Myb_DNA-binding,zf-CCCH;

PLATZ:TFF;PLATZ;;

POU:TFF;Pou;;

Pseudo_ARR-B:OTR;Response_reg,CCT;;

RB:OTR;RB_B;;

Rcd1-like:OTR;Rcd1;;

Rel:TFF;RHD;;

RF-X:TFF;RFX_DNA_binding;;

Runt:TFF;Runt;;

RWP-RK:TFF;RWP-RK;;

S1Fa-like:TFF;S1FA;;

SAP:TFF;SAP;;

SART-1:OTR;SART-1;;

SBP:TFF;SBP;;

SET:OTR;SET;PHD,zf-C2H2,CXC;

SGT1:OTR;SGT1;;

Sigma70-like:TFF;Sigma70_r2,Sigma70_r3,Sigma70_r4;;

Sin3:OTR;PAH,SIR2;WRKY;

SNF2:OTR;SNF2_N;PHD,AP2;

SOH1:OTR;Med31;;

SRF-TF:TFF;SRF-TF;;

SRS:TFF;DUF702;;

STAT:TFF;STAT_int=STAT_alpha=STAT_bind;;

STE:TFF;STE;;

SWI/SNF-BAF60b:OTR;SWIB;;

TAZ:TFF;zf-TAZ;PHD;

TCP:TFF;TCP;;

TEA:TFF;TEA;;

TFIIS:TFF;TFIIS_M,TFIIS_C;;

Tify:TFF;tify;GATA;

TIG:TFF;TIG;;

Top:TFF;Topless;;

TRAF:OTR;BTB;zf-TAZ,BACK,MATH,NPH3;

Trihelix:TFF;Trihelix;;

TUB:TFF;Tub;;

ULT:TFF;ULT;;

SAND:OTR;SAND;VARL;

VARL:TFF;VARL;;

VOZ:TFF;VOZ;;

WRKY:TFF;WRKY;;

Yl1:TFF;YL1;;

zf-A20:TFF;zf-A20;;

zf-BED:TFF;zf-BED;;

zf-C4:TFF;zf-C4;;

zf-HD:TFF;ZF-HD_dimer;;

zf-LSD1:TFF;zf-LSD1;;

zf-MIZ:OTR;zf-MIZ;;

zf-NF-X1:TFF;zf-NF-X1;;

ZnClus:TFF;Zn_clus;;

pipsqueak:TTF;HTH_psq;;

Fis:TFF;HTH_8;;

DeoR:TFF;HTH_DeoR;;

Mga:TFF;HTH_Mga;;

IclR:TFF;HTH_IclR;;

WhiA:TFF;HTH_WhiA;;

A.2. Esquema de la base de datos

– MySQL dump 10.11

– Server version 5.0.77

– Table structure for table EST

CREATE TABLE EST (

Sp_pepid varchar(55) NOT NULL,

Sp_ID varchar(10) NOT NULL,

pep_id varchar(20) NOT NULL,

est_id1 varchar(20) NOT NULL,

est_id2 varchar(20) default NULL,

est_desc varchar(255) default NULL,

overlap int(11) NOT NULL,

Num_identical_aa int(11) NOT NULL,

evalue varchar(10) NOT NULL,

identity double NOT NULL,

DB varchar(15) NOT NULL,

UNIQUE KEY Sp_pepid (Sp_pepid,est_id1),

CONSTRAINT EST_ibfk_1 FOREIGN KEY (Sp_pepid) REFERENCES tf (Sp_pepid)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

– Table structure for table Gene_names

CREATE TABLE Gene_names (

Sp_pepid varchar(55) NOT NULL,

Sp_ID varchar(10) NOT NULL,

pep_id varchar(20) NOT NULL,

```
name varchar(255) NOT NULL,
PRIMARY KEY (Sp_ID,pep_id,name),
KEY Gene_names_ibfk_1 (Sp_pepid),
CONSTRAINT Gene_names_ibfk_1 FOREIGN KEY (Sp_pepid) REFERENCES tf (Sp_pepid)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
– Table structure for table Links
CREATE TABLE Links (
Sp_ID varchar(20) NOT NULL,
Link varchar(255) NOT NULL,
Category varchar(50) NOT NULL,
Name varchar(100) NOT NULL,
ID varchar(20) NOT NULL,
UseLoci tinyint(1) NOT NULL,
PRIMARY KEY (ID)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
– Table structure for table Orthologs
CREATE TABLE Orthologs (
Sp_pepid varchar(55) NOT NULL,
OrthologId varchar(30) NOT NULL default '0',
Score varchar(30) NOT NULL,
Sp_ID varchar(30) NOT NULL,
Conf_score float NOT NULL default '0',
pep_id varchar(30) NOT NULL,
PRIMARY KEY (OrthologId,pep_id),
KEY pep_id (pep_id),
KEY Orthologs_ibfk_1 (Sp_pepid),
CONSTRAINT Orthologs_ibfk_1 FOREIGN KEY (Sp_pepid) REFERENCES tf (Sp_pepid)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
– Table structure for table Papers
CREATE TABLE Papers (
```

PUBMEDID varchar(50) NOT NULL,

Authors varchar(255) NOT NULL,

Year int(4) NOT NULL default '0',

Tittle varchar(255) NOT NULL,

Journal varchar(255) NOT NULL,

Volume varchar(10) default NULL,

Number varchar(10) default NULL,

Pages varchar(10) default NULL,

UNIQUE KEY PUBMEDID (PUBMEDID)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

– Table structure for table Present_domains

CREATE TABLE Present_domains (

Sp_pepid varchar(55) NOT NULL,

Sp_ID varchar(10) NOT NULL,

pep_id varchar(25) NOT NULL,

domain_id varchar(255) NOT NULL,

PFAM_id varchar(15) default NULL,

PFAM_id_version tinyint(4) default NULL,

SourceDB enum('PFAM','MueRoes') NOT NULL default 'PFAM',

start int(11) NOT NULL,

end int(11) NOT NULL,

score float NOT NULL,

evalue varchar(20) NOT NULL,

KEY Sp_ID (Sp_ID),

KEY Present_domains_ibfk_1 (Sp_pepid),

KEY PFAM_id (PFAM_id),

CONSTRAINT Present_domains_ibfk_1 FOREIGN KEY (Sp_pepid) REFERENCES tf

(Sp_pepid)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

– Table structure for table Sequences

```
CREATE TABLE Sequences (
Sp_pepid varchar(55) NOT NULL,
Sp_ID varchar(10) NOT NULL,
pep_id varchar(50) NOT NULL,
transcript_id varchar(100) NOT NULL,
locus_id varchar(100) default NULL,
transcript_seq longtext,
locus_seq longtext,
pep_seq longtext NOT NULL,
upstream2kb_seq longtext,
upstream5kb_seq longtext,
chromosome varchar(20) default ÑD',
Description varchar(255) default NULL,
alt_seq_id varchar(100) default NULL,
md5sum_pep varchar(255) default NULL,
PRIMARY KEY (Sp_pepid),
KEY pep_id (pep_id,Sp_ID),
CONSTRAINT Sequences_ibfk_1 FOREIGN KEY (Sp_pepid) REFERENCES tf (Sp_pepid)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
– Table structure for table Species
CREATE TABLE Species (
Species_full_name varchar(150) NOT NULL,
Sp_ID varchar(20) NOT NULL,
NCBI_TaxID varchar(15) NOT NULL,
SuperGroup varchar(50) NOT NULL,
Groups varchar(50) NOT NULL,
Sort int(11) NOT NULL default '0',
SourceDBName varchar(55) NOT NULL,
SourceDBURL varchar(255) NOT NULL,
SourceDBversion varchar(10) NOT NULL,
```

Used enum('1','0') NOT NULL,

PRIMARY KEY (Sp_ID),

UNIQUE KEY Species_full_name (Species_full_name),

UNIQUE KEY NCBI_TaxID (NCBI_TaxID)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

– Table structure for table tf

CREATE TABLE tf (

Sp_pepid varchar(55) NOT NULL,

Sp_ID varchar(10) NOT NULL,

pep_id varchar(50) NOT NULL,

family_id varchar(50) NOT NULL,

Type enum('Genome',ÉST') NOT NULL default 'Genome',

Compartment enum(ÑUC','MIT','CHL') default ÑUC',

PRIMARY KEY (Sp_pepid),

KEY Sp_ID (Sp_ID),

KEY family_id (family_id),

CONSTRAINT tf_ibfk_2 FOREIGN KEY (Sp_ID) REFERENCES Species (Sp_ID),

CONSTRAINT tf_ibfk_3 FOREIGN KEY (family_id) REFERENCES tf_families (Family_id) ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

– Table structure for table tf_PDB

CREATE TABLE tf_PDB (

Sp_pepid varchar(55) collate utf8_unicode_ci NOT NULL,

PDB_id varchar(10) collate utf8_unicode_ci NOT NULL,

PDB_chain varchar(5) collate utf8_unicode_ci NOT NULL,

Prob float NOT NULL,

Evalue float NOT NULL,

Score float NOT NULL,

SS float NOT NULL,

start int(11) NOT NULL,

end int(11) NOT NULL,

KEY Sp_pepid_2 (Sp_pepid),

KEY PDB_id (PDB_id),

KEY SppepidPDBid (Sp_pepid,PDB_id),

CONSTRAINT tf_PDB_ibfk_1 FOREIGN KEY (Sp_pepid) REFERENCES tf (Sp_pepid)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

– Table structure for table tf_domains_alignments

CREATE TABLE tf_domains_alignments (

Sp_ID varchar(10) NOT NULL,

Family_id varchar(50) NOT NULL,

Domain_id varchar(50) NOT NULL,

DomainAlgn longtext,

PRIMARY KEY (Sp_ID,Family_id,Domain_id),

KEY Family_id (Family_id),

CONSTRAINT tf_domains_alignments_ibfk_1 FOREIGN KEY (Family_id) REFEREN-

CES tf_families (Family_id) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Multiple Alignments';

– Table structure for table tf_families

CREATE TABLE tf_families (

Family_id varchar(50) NOT NULL,

Family_name varchar(255) default NULL,

Familiy_description longtext ,

Comment longtext,

benchmark longtext,

Category enum(Óther','TF') NOT NULL,

PRIMARY KEY (Family_id)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

– Table structure for table tf_families_motifs

CREATE TABLE tf_families_motifs (

Family_id varchar(50) NOT NULL,

Motif_id varchar(50) NOT NULL,

Type enum('REQUIRED','FORBID',ÓPTIONAL') default NULL,

Comment text,

motif_db varchar(50) default 'pfam',

PRIMARY KEY (Family_id,Motif_id),

CONSTRAINT tf_families_motifs_ibfk_1 FOREIGN KEY (Family_id) REFERENCES tf_families (Family_id) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

– Table structure for table tf_families_papers

CREATE TABLE tf_families_papers (Family_id varchar(50) NOT NULL,

Paper_id varchar(50) NOT NULL,

Type varchar(20) NOT NULL default 'General',

PRIMARY KEY (Family_id,Paper_id,Type),

KEY Paper_id (Paper_id),

CONSTRAINT tf_families_papers_ibfk_1 FOREIGN KEY (Family_id) REFERENCES tf_families (Family_id) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT tf_families_papers_ibfk_2 FOREIGN KEY (Paper_id) REFERENCES Papers (PUBMEDID)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

– Dump completed on 2011-06-12 20:21:17

–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.

Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes