Universitat Autònoma de Barcelona

escola d'enginyeria

4507-1 BIOINFORMÁTICA:

BASE DE DATOS DE MATRICES DE EXPRESION GÉNICA PARA SU ANÁLISIS VÍA WEB

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Daniel Sánchez Santolaya
i dirigit per
Mario Huerta y Jordi Gonzàlez i Sabaté
Bellaterra, 17 de Setembre de 2012

Universitat Autònoma de Barcelona

etse
Escola Tècnica Superior d'Enginyeria

El sotasignat, Jordi Gonzàlez i Sabaté

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Daniel Sánchez Santolaya

I per tal que consti firma la present.

Signat:

Bellaterra, 17 de setembre de 2012

Universitat Autònoma de Barcelona

etse
Escola Tècnica Superior d'Enginyeria

El sotasignat, Mario Huerta

De l'empresa, Institut de Biotecnologia i de Biomedicina de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat a l'empresa sota la seva supervisió mitjançant conveni amb la Universitat Autònoma de Barcelona.

Així mateix, l'empresa en té coneixement i dóna el vist-i-plau al contingut que es detalla en aquesta memòria.

Signat:

Bellaterra, 17 de setembre de 2012

Tabla de contenido

1. Introducción

1.1 Motivación personal

En los últimos años la informática, y más concretamente la bioinformática, se ha ido introduciendo en campos como la biología o la medicina. Hoy en día, muchos de los grandes avances científicos que se realizan en algunos de estos campos se consiguen en parte gracias a la ayuda que proporcionan las técnicas y herramientas que produce la bioinformática para la investigación científica.

Una fuente de motivación para la elección de este proyecto, y probablemente la más importante, ha sido el poder aplicar mis conocimientos para ayudar en investigaciones científicas en la biología o en la biomedicina, tratando temas tan importantes como la cura o tratamiento de enfermedades que pueden ayudar a mejorar la calidad de vida de los seres humanos. Por esta razón, en el momento de elección del proyecto sentí que podía ser una buena oportunidad para realizar una aplicación de utilidad, de manera que una vez realizado el proyecto los investigadores dispondrían de una herramienta que les facilitase su investigación en el estudio de los procesos biológicos y las enfermedades.

En segundo lugar, pensé que era una buena oportunidad para realizar un proyecto de una aplicación real y en un centro de investigación real, simulando de esta manera situaciones en las que me pueda encontrar en el futuro, algo que es una oportunidad realmente única hasta ahora.

Por otra parte, el proyecto me permitía adquirir conocimientos en el ámbito de la bioinformática y la biotecnología, unas disciplinas que actualmente se encuentran en crecimiento, por lo que los conocimientos adquiridos en mi proyecto pueden ser importantes para mi futuro. Además, me ha permitido poner en práctica los conocimientos que he ido adquiriendo durante la carrera además de aprender nuevos.

El proyecto se ha realizado en el Institut de Biotecnología y de Biomedicina(IBB)[1] de la Universidad Autónoma de Barcelona, formando parte de su línea de investigación de

análisis de microarrays. Al finalizar el proyecto he tenido la satisfacción de ver como he podido alcanzar todos los objetivos iniciales.

1.2 Estado del arte

Este proyecto está situado en el ámbito de la bioinformática, una disciplina científica que incluye varios campos como la biología, la computación y las tecnologías de la información. La bioinformática se dedica al estudio de los fenómenos biológicos desde un punto de vista computacional, como puede ser el alineamiento de secuencias, la predicción de genes, el estudio de la expresión génica, entre otros.

Concretamente, el actual proyecto está centrado en ofrecer herramientas a los investigadores para estudiar el comportamiento de la expresión de genes bajo diferentes condiciones experimentales. Los genes al expresarse sintetizan las diferentes proteínas. Son estas proteínas sintetizadas las que realizan las diferentes funciones de la célula. Por esta razón, al expresarse los genes determinan el estado celular y modificando esta expresión génica, se puede provocar un cambio celular. Esto es muy importante, ya que estos cambios celulares pueden provocar el llevar a una persona de la salud a la enfermedad o al contrario. Por lo tanto, el estudio de la expresión de los genes nos puede ayudar a entender las causas de ciertas enfermedades, o a encontrar una posible cura, de manera que puede ayudar a salvar muchas vidas o a mejorar la calidad de vida de las personas que padecen estas enfermedades.

Este estudio de la expresión génica se puede realizar mediante la tecnología de microarrays. La tecnología de microarrays nos permite observar el nivel de expresión de un número grande de genes bajo un número de circunstancias diferentes. De esta manera, permiten estudiar el comportamiento de los mismos genes en diversas condiciones. Los datos generados por las microarrays se pueden ver como una inmensa matriz, dónde las filas representan cada uno de los genes analizados, las columnas cada una de las condiciones experimentales a las que han sido sometidos los genes, y finalmente los valores de la matriz serán los niveles de expresión de cada gen bajo cada condición experimental.

Dependiendo de las condiciones experimentales, la microarray nos proporciona información de diferente tipo. Las condiciones experimentales pueden ser la respuesta génica a distintos fármacos, a variaciones de dosis, a diferentes fases en el progreso de una enfermedad, etc. Por ejemplo si las condiciones experimentales de la microarray son sobre

el cáncer de piel, la microarray nos mostrará los niveles de expresión de los genes para el cáncer de piel.

Las matrices resultantes de utilizar la tecnología de microarrays contienen una gran cantidad de información (miles de genes por cientos de condiciones experimentales), por lo que es necesario realizar análisis automáticos para extraer información útil para el investigador. En el servidor del IBB revolutionresearch.uab.es[2] se realizan algunos de estos análisis[3][4][5][6][7][8], como el agrupamiento o clustering de las condiciones muestrales, un grafo con las relaciones entre genes o la clasificación de las relaciones no lineales entre pares genes. Estos análisis son los que proporcionan información realmente útil a los investigadores.

En el servidor existe una base de datos de microarrays, así como las herramientas de análisis y la interfaz web para gestionarlas. Actualmente, estas microarrays han de ser obtenidas y subidas al servidor por los usuarios. El servidor también dispone de una base de datos de genes marcadores de microarrays. Los genes marcadores de microarrays son aquellos genes que se sobreexpresan en unas condiciones experimentales pero no en otras. Gracias a una aplicación del servidor se permite cruzar los genes marcadores de una microarray del usuario con los genes marcadores de esta base de datos, de manera que el usuario puede ver los genes marcadores que tiene en común con otras microarrays y extrapolar información de otras microarrays de origen muy diferente.

Un problema actual en el servidor es que para obtener nuevas microarrays que generen científicos de todo el mundo, el usuario es el encargado de buscar y subir esas microarrays. Esto produce que la cantidad de microarrays, y como consecuencia información, se reduzca ampliamente, ya que aparte de buscar las microarrays, el usuario tiene que ocuparse de introducirlas en un formato aceptado por el servidor. Por esta razón, es importante obtener microarrays de manera automatizada y que se dispongan en el servidor las últimas microarrays publicadas en todo el mundo. Por suerte, en Internet existen bases de datos que centralizan esta información y que se permiten el acceso a esta de manera gratuita.

El NCBI(National Center for Biotechnology Information)[9] que forma parte de la Biblioteca Nacional de Medicina de Estados Unidos, contiene numerosas bases de datos con información biológica. Una de estas bases de datos es GEO (Gene Expression Omnibus) Datasets[10][11], donde se publican las últimas microarrays en el panorama internacional. Esta base de datos fue creada en el 2000 y actualmente contiene más de 20.000 microarrays. Sin embargo, no todas las microarrays son de la misma utilidad. En la

base de datos GEO Datasets podemos encontrar desde microarrays con 2 condiciones experimentales hasta microarrays con 202. Cuantas más condiciones experimentales a las que se han sometido los genes tenga la microarray más análisis se podrán realizar y más información se podrá obtener, además, estas microarrays nos describen con mayor robustez y exactitud los diferentes cambios que se suceden en el fenómeno que está siendo estudiado en la microarray. Por lo tanto, obtener las microarrays con un número elevado de condiciones experimentales será especialmente útil para los investigadores.

1.3 Objetivos

El objetivo principal del proyecto es ampliar la base de datos local de microarrays a partir de las microarrays con un gran número de condiciones experimentales publicadas en el panorama internacional(microarrays públicas de gran tamaño), de manera que esta se vaya actualizando periódicamente para mantener la base de datos con las últimas microarrays publicadas, y permitir así, utilizar las aplicaciones web del servidor para el análisis de estas microarrays. De esta manera, al final del proyecto los usuarios de la aplicación web verán que la información disponible se ha ampliado sustancialmente, ya que dispondrán de todas las microarrays públicas de gran tamaño para realizar experimentos.

Para conseguir el objetivo principal del proyecto, este se ha dividido en dos:

- Actualización periódica y automática de la base de datos local de microarrays:
    - Obtener las nuevas microarrays de gran tamaño del NCBI.
    - Descargar y parsear los ficheros de las microarrays para que se adecuen al formato de las microarrays del servidor local. Tanto datos, como genes y condiciones experimentales.
    - Subir a la base de datos local las microarrays descargadas y parseadas.
    - Realizar la actualización de manera que si los genes de la microarray cambian de nombre, estos puedan ser actualizados por el robot actualizador de nombres de genes que actualmente hay en el servidor.
    - La actualización ha de ser robusta a posibles errores o a la caída del servidor durante el proceso.
    - Eliminación de directorios y ficheros temporales que no son necesarios tras finalizar el proceso de actualización.

- Realizar la actualización de manera periódica y sincronizada con el robot de actualización de genes marcadores de microarrays que existe actualmente en el servidor.

- Interfaz web para gestionar las nuevas microarrays:
  - Adaptar la aplicación web para que las operaciones que se hacían anteriormente con las microarrays subidas por los usuarios se puedan realizar también con las microarrays subidas por la actualización automática.
  - Crear Interfaz web que permita a los usuarios realizar búsquedas de las microarrays públicas de gran tamaño insertando el tema de la microarray y/o la especie sobre la que se realizaron los experimentos.
  - Mantener un listado de las microarrays públicas favoritas que el usuario considera de interés, de manera que le permitirá un acceso rápido a ellas.

Tras realizar el proyecto se proporcionará a los investigadores que trabajan con la aplicación la posibilidad de trabajar con microarrays con un gran número de condiciones experimentales descargadas automáticamente del NCBI.

1.4 Organización de la memoria

En este apartado expondré la planificación que he seguido para llevar a cabo los objetivos propuestos y la estructura que seguirá la memoria del trabajo.

En el siguiente apartado de la memoria, expondré los fundamentos teóricos necesarios para comprender los conceptos con los que he trabajado en el proyecto.

Tras esto, pasaré a explicar las distintas fases llevadas a cabo durante el proyecto. Estas fases han sido las siguientes:

- Adquirir los conocimientos necesarios sobre la bioinformÁtica y el ámbito del proyecto. Esta primera fase incluye adquirir los conocimientos de varios tipos:
  - Adquirir conocimientos necesarios sobre la bioinformática.
  - Adquirir conocimientos necesarios sobre la aplicación web y las bases de datos del servidor local.
  - Adquirir conocimientos necesarios sobre el entorno del NCBI.

- Crear robot de descarga de microarrays de gran tamaño del NCBI. Esta fase incluye todo el proceso realizado para crear la base de datos con las microarrays de gran tamaño del NCBI. Se divide en las siguientes sub-fases:
    - Identificar las microarrays a descargar.
    - Descargar las microarrays.
    - Parsear ficheros de las microarrays.
    - Subir microarrays a la base de datos local.
    - Control de errores y recuperación tras caída del servidor en el proceso de actualización.
    - Eliminación de directorios y ficheros temporales que no son necesarios tras finalizar el proceso de actualización.
- Sincronización con el robot de genes marcadores y programación periódica de ambos robots. En esta fase se explica el funcionamiento del robot de genes marcadores y como se ha conseguido la sincronización y la programación periódica de ambos robots.
- Aplicación web. Esta fase incluye las nuevas interfaces creadas para la búsqueda de microarrays de gran tamaño y para mostrar una lista de microarrays públicas favoritas para cada usuario. También incluye las modificaciones que se han realizado en la aplicación web para que funcionen todos los análisis y operaciones de gestión de estas microarrays.

Una vez expuestas las distintas fases del trabajo, se expondrá el Informe técnico. Esta sección incluye la descripción de todos los programas implementados, descripción de ficheros y bases de datos, y las estructuras de directorios utilizadas para facilitar la reusabilidad en posteriores implementaciones.

En el penúltimo apartado explicaré las conclusiones tras realizar el proyecto, detallando qué objetivos se han cumplido y las impresiones personales. Así como el trabajo futuro.

Finalmente, se podrá encontrar la bibliografía utilizada durante la realización del proyecto.

2. Fundamentos teóricos

2.1 Bioinformática

La Biología computacional o bioinformática es la ciencia dedicada al estudio de fenómenos biológicos desde un punto de vista computacional. Es un área de investigación multidisciplinaria, la cual puede ser ampliamente definida como la interfase entre dos ciencias: Biología y Computación. De esta manera tiene el objetivo de ofrecer métodos robustos para la comprensión, simulación y predicción de comportamientos biológicos observados en los seres vivos, así como organizar, analizar y distribuir información biológica, con el fin de responder preguntas complejas en biología. Está impulsada por la incógnita del genoma humano y la esperanza en que la investigación genómica puede ayudar a mejorar la condición y calidad de vida de los humanos.

Algunos de los beneficios que puede aportar la bioinformática son avances en la detección y tratamiento de enfermedades o la producción de alimentos genéticamente modificados. Combinada con las nuevas técnicas de biotecnología, la bioinformática permite identificar genes y proteínas causantes de enfermedades o de mecanismos de resistencia a antibióticos, de otra forma complicados a detectar.

Según la definición del NCBI, la bioinformática es un campo de la ciencia en el cual confluyen varias disciplinas tales como la biología, la computación y la tecnología de la información. El fin último de este campo es facilitar el descubrimiento de nuevas ideas biológicas así como crear perspectivas globales a partir de las cuales se puedan discernir principios unificadores en la biología. La bioinformática emplea una amplia gama de técnicas computacionales, como el diseño de bases de datos y data mining, la predicción de la estructura y función de proteínas, la búsqueda de genes, entre otros.

En sus inicios, el concepto de bioinformática se refería sólo a la creación y mantenimiento de base de datos donde se almacena información biológica. El desarrollo de este tipo de base de datos implicaba también el desarrollo de interfaces complejas donde los investigadores pudieran acceder a los datos i analizarlos.

Posteriormente toda esa información debía ser combinada para formar una idea lógica de las actividades celulares en estados normales, de tal manera que los investigadores

pudieran estudiar cómo estas actividades se veían alteradas en estados de una enfermedad. De allí viene el surgimiento del campo de la bioinformática y ahora el campo más popular es el análisis e interpretación de varios tipos de datos.

El proceso de analizar e interpretar estos datos es conocido como biocomputación y se puede dividir en otras dos sub-disciplinas. Por una parte, podemos considerar el desarrollo e implementación de herramientas que permitan el acceso, uso y manejo de varios tipos de información. Por otra parte, el desarrollo de algoritmos y estadísticos con los cuales se puede relacionar partes de un conjunto enorme de datos, como por ejemplo métodos para localizar un gen dentro de una secuencia o métodos para agrupar secuencias de proteínas en familias relacionadas.

La Medicina Molecular y la Biotecnología constituyen dos áreas prioritarias científico tecnológicas como desarrollo e Innovación Tecnológica. El desarrollo de ambas está estrechamente relacionado. En las dos se pretende potenciar la investigación genómica y postgenómica así como de la bioinformática, herramienta imprescindible para el desarrollo de estas, ya que el número y la complejidad de los datos aumentan a un ritmo más alto que la capacidad de los ordenadores que los harán procesar, por lo que se hace necesario un desarrollo teórico y práctico en el entorno de las herramientas bioinformáticas. Debido al extraordinario avance de la genética molecular y la genómica, la Medicina Molecular se constituye como arma estratégica del bienestar social del futuro inmediato. Se pretende potenciar la aplicación de las nuevas tecnologías y de los avances genéticos para el beneficio de la salud.

Las tecnologías de la información tendrán un papel fundamental en la aplicación de los desarrollos tecnológicos en el campo de la genética a la práctica médica. La aplicación de los conocimientos en genética molecular y las nuevas tecnologías son necesarios para el mantenimiento de la competitividad del sistema sanitario no sólo paliativo sino preventivo. La identificación de las causas moleculares de las enfermedades junto con el desarrollo de la industria biotecnológica en general y de la farmacéutica en particular permitirán el desarrollo de mejores métodos de diagnóstico, la identificación de dianas terapéuticas y desarrollo de fármacos personalizados y una mejor medicina preventiva.

2.2 Microarrays

Un chip de ADN(del inglés DNA microarrays)[1] es una superficie sólida a la cual se unen una serie de fragmentos de ADN. Las superficies empleadas para fijar el ADN son muy variables y pueden ser vidrio, plástico e incluso chips de silicio. Los arreglos de ADN son utilizados para averiguar la expresión de genes, monitorizándose los niveles de miles de ellos de forma simultánea.

La técnica consiste en extraer el RNAm de una célula buena y de otra experimental mediante isolation RNA, una vez extraído los dos RNAm se marca cada uno de ellos con un color distinto y se combinan los dos. A continuación se vierte el combinado en la superficie del chip de tal modo que cada RNAm se unirá o no a los cDNA de cada gen del chip. Finalmente, aplicando técnicas de análisis de imágenes es posible generar una matriz de datos numéricos, a partir de los patrones de intensidades de cada celda y discriminando la señal informativa de ruido que pudiera haber en segundo plano. Estos datos numéricos corresponderán al nivel de expresión de cada gen expuesto a una serie de condiciones experimentales. Por ejemplo, si el ARN de la célula experimental se coloreó de color rojo, los genes con un color más cercano al rojo tienen un nivel de expresión más elevado en las células experimentales.

Por lo tanto, las microarrays son una potente fuente de obtención de perfiles de expresión de genes sometidos a diferentes condiciones. Identificar los patrones de los niveles de expresión será muy útil para compararlos y poder estudiar las respuestas de los genes.

Aplicando una serie de procesos experimentales y computacionales sobre la microarray se obtiene una matriz numérica bidimensional que consta de los genes de poblaciones distintas como individuos y de las condiciones experimentales a las que se expusieron las células como variables en el caso que se quiera estudiar a los genes, o a la inversa, si es que se quiere realizar un estudio comparativo de las condiciones a que se somete. Cada uno de los valores de la matriz representa el nivel de expresión de un determinado gen bajo una cierta condición experimental. Es posible que en algunos casos se produzcan errores en el proceso y se generen huecos.

Estas matrices son de grandes dimensiones puesto que existen una gran cantidad de genes y de condiciones experimentales. En la figura 2.1 se puede observar el modelo de

1 http://es.wikipedia.org/wiki/Chip_de_ADN

representación de las microarrays. Cada fila de la microarray representa un gen, el cual debe ser indicado. Cada columna de la microarray representa una condición experimental de la microarray, cuyo nombre también debe ser indicado. Los valores de la matriz son los niveles de expresión de los genes para las condiciones experimentales.

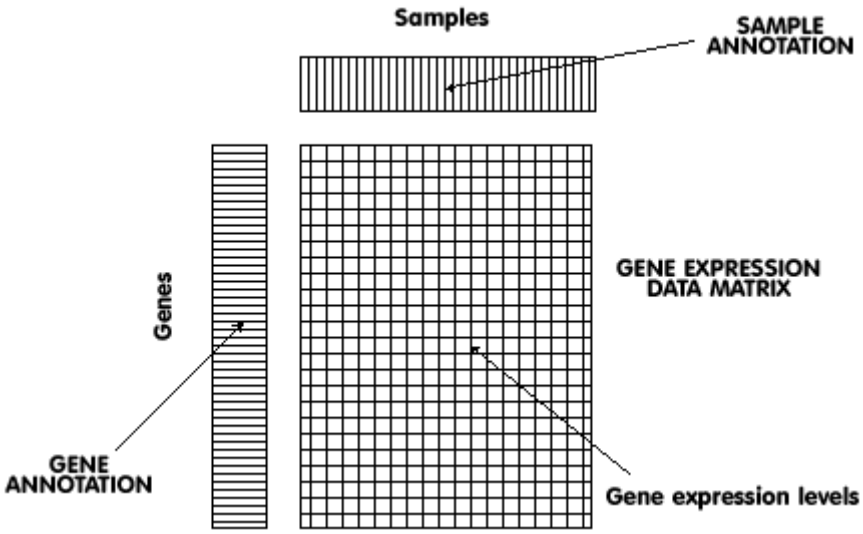

Figura 2.1 Microarray

Modelo de representación de las microarrays. Cada fila representa un gen, de los cuales se tiene que anotar su nombre. Cada columna representa una condición experimental, cuyo nombre debe anotarse también. Los valores de la matriz son los niveles de expresión de los genes para las condiciones experimentales.

Dado que realizar un análisis de estas matrices de grandes dimensiones es una tarea prácticamente imposible, se hace necesarias técnicas computacionales que permitan analizar todos estos datos y entonces realizar el análisis biológico.

Actualmente existen diferentes bases de datos a nuestro alcance a través de Internet que unifican y facilitan toda esta información genética además de ofrecer diversas herramientas para el análisis de esta gran cantidad de información. Algunas de estas bases de datos por ejemplo son las que hay en el EMBL (European Molecular Biology Laboratory), el SIB (Swiss Institute of Bioinformatics), el EBI (European Bioinformatics Institute) o el NCBI (National Center for Biotechnology Information). El EBI y el NCBI son los que más información contienen y por lo tanto los más utilizados.

2.2.1 Análisis de microarrays

Como se ha comentado, analizar la matriz de la microarray directamente no es una opción viable, por lo que la técnica consiste en utilizar algoritmos o técnicas con esta matriz de manera que posteriormente lo que se analizará serán los resultados de estos algoritmos y no la matriz entera.

Uno de las técnicas utilizadas consiste en emplear algoritmos de agrupamiento (clustering en inglés) a las microarrays. Estos algoritmos consisten en encontrar un grupo(clúster) de un conjunto de individuos de tal forma que los clústeres resultantes sean homogéneos y/o estén bien separados. El punto clave es reducir la gran cantidad es reducir la cantidad de datos caracterizándolos en grupos más pequeños de individuos similares. Esto implica que los individuos pertenecientes a un mismo clúster son lo más similares posibles entre ello, mientras que los individuos de clústeres distintos son lo más disimilares posibles.

Otra técnica utilizada consiste en observar la relación de expresión entre genes. Esto consiste en analizar si dos genes tienen dependencia observando que ocurre con sus niveles de expresión a medida que varían los niveles de expresión del otro. En la figura 2.2 se puede observar un ejemplo de relación entre genes. En el eje horizontal podemos observar los valores de expresión del gen ZER1 y en el eje vertical los valores del gen RETSAT. Podemos observar que a medida que sube el nivel de expresión del gen ZER1 también se aumenta el nivel de expresión del gen RETSAT. La captura de pantalla ha sido sacada de la aplicación web del servidor revolutionresearch.uab.es[2]

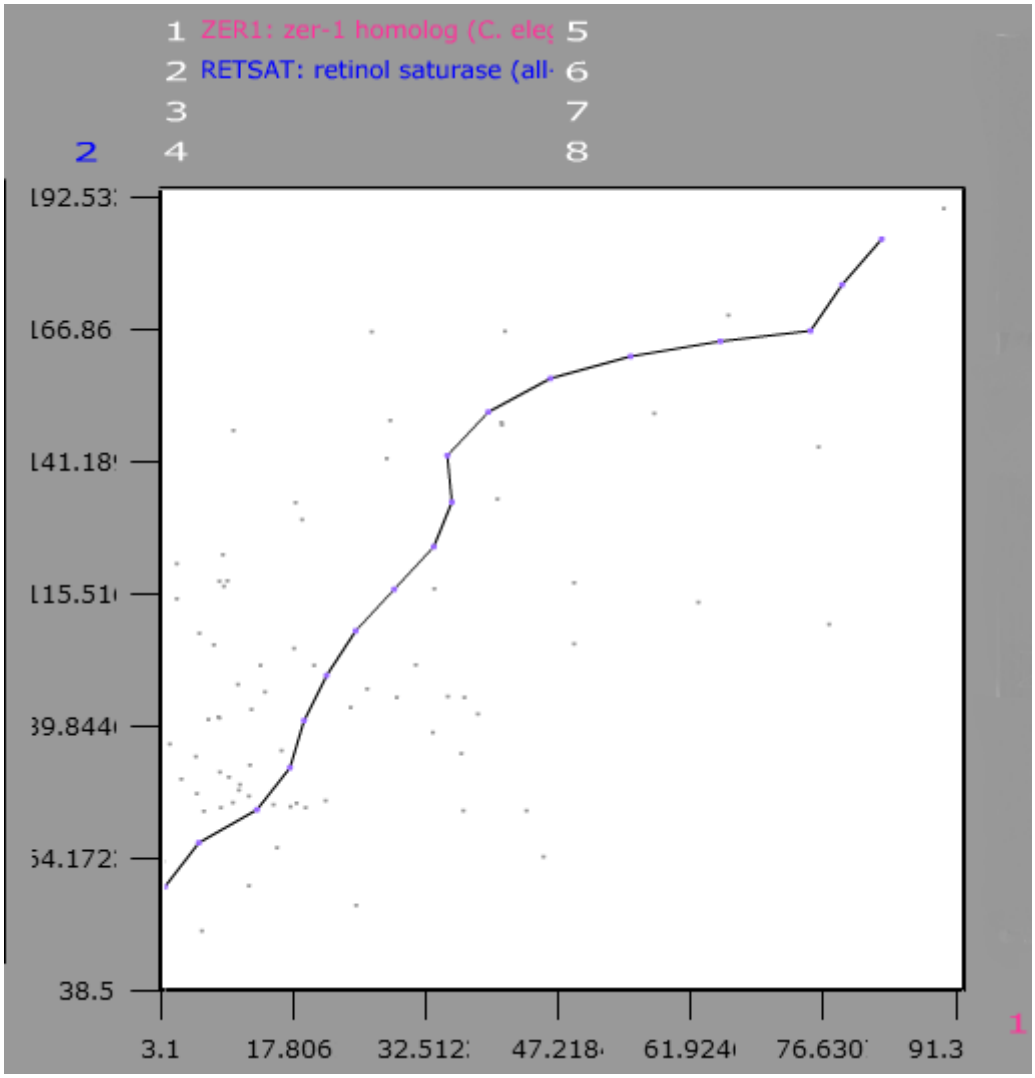

Figura 2.2 Ejemplo de una relación entre expresiones de genes realizado con la aplicación web del servidor revolutionrsearch.uab.es[2]

Ejemplo de gráfica de relación de expresión entre genes. En el eje horizontal podemos observar los valores de expresión del gen ZER1 y en el eje vertical los valores del gen RETSAT. Podemos observar que a medida que sube el nivel de expresión del gen ZER1 también se aumenta el nivel de expresión del gen RETSAT.

Otro tipo de análisis que se realiza en las microarrays consiste en identificar aquellos genes que se sobreexpresan en unas condiciones experimentales pero no en otras, y por extensión, los genes que se sobreexpresan en unos clústeres de condiciones experimentales pero no en otros. Estos genes son llamados genes marcadores. Los genes marcadores se caracterizan por destacar el comportamiento de la microarray.

Estas y otro tipo de técnicas son utilizadas para analizar las microarrays y sacar información en ellas. Estas herramientas de anàlisis están disponibles para los usuarios en el servidor revolutionresearch.uab.es[2][3][4][5][6][7][8].

2.3 NCBI y base de datos GEO

La base de datos GEO del NCBI contiene gran cantidad de microarrays públicas y perfiles de expresión de los genes de las microarrays que son subidas por investigadores de todo el mundo. En esta base de datos también hay disponibles herramientas de análisis de las microarrays.

En la figura 2.3 podemos ver una de estas herramientas. Podemos ver la agrupación en clusters de distintos genes y condiciones experimentales de la microarray. En la izquierda podemos observar las agrupaciones de los genes, y en la parte superior las agrupaciones de las condiciones experimentales. El color verde indica que los genes tienen un nivel bajo de expresión, mientras que el color lila indica que los genes tienen un nivel alto de expresión.

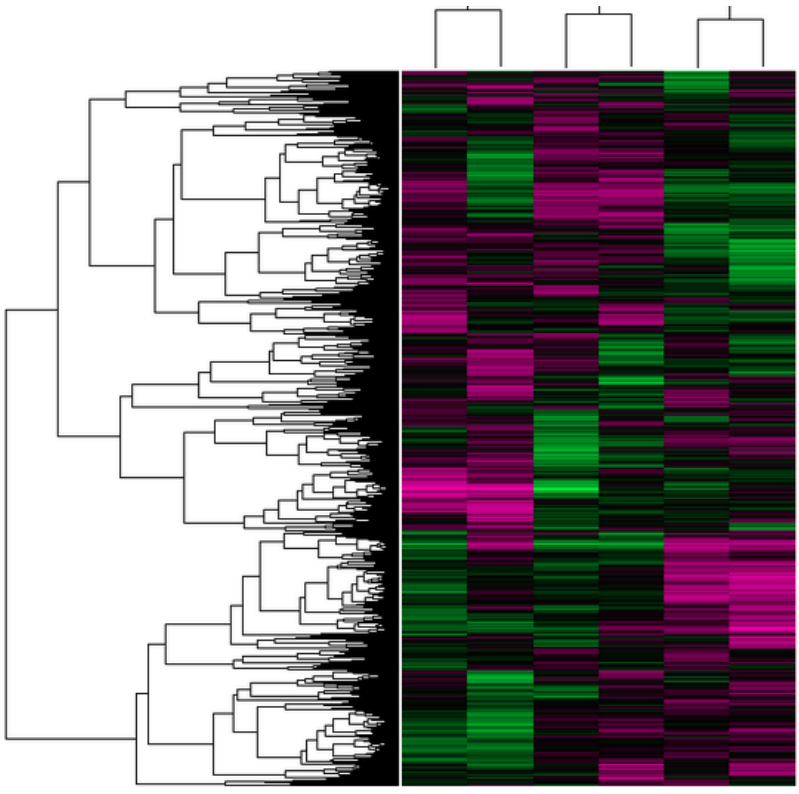

Figura 2.3 Imagen de los clusters de una microarray realizada en el NCBI Podemos ver la agrupación en clusters de distintos genes y condiciones experimentales de la microarray. En la izquierda podemos observar las agrupaciones de los genes, y en la parte superior las agrupaciones de las condiciones experimentales. El color verde indica que

los genes tienen un nivel bajo de expresión, mientras que el color lila indica que los genes tienen un nivel alto de expresión.

En la figura 2.4 podemos ver otra de estas herramientas de análisis. Esta herramienta nos permite buscar los genes marcadores de una microarray, así como ver su nivel de expresión en la microarray en sus diferentes condiciones experimentales. En este caso se ha podido encontrar como el gen zgc:73230 es un gen marcador de la microarray GDS3719 y en la imagen podemos ver como el gen tiene un nivel alto de expresión en condiciones Ovo1 morphant y un nivel bajo de expresión en condiciones de control.

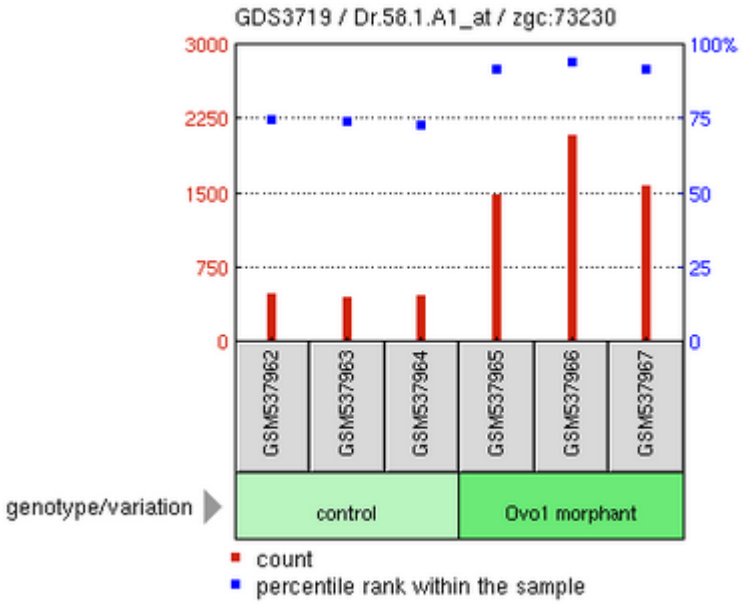

Figura 2.3 Imagen del perfil de expresión de un gen marcador de una microarray de la herramienta de análisis del NCBI

Esta herramienta nos permite buscar los genes marcadores de una microarray, así como ver su nivel de expresión en la microarray en sus diferentes condiciones experimentales. En este caso se ha podido encontrar como el gen zgc:73230 es un gen marcador de la microarray GDS3719 y en la imagen podemos ver como el gen tiene un nivel alto de expresión en condiciones Ovo1 morphant y un nivel bajo de expresión en condiciones de control.

3. Fases

A continuación, se muestra una comparativa entre la planificación inicial y la planificación final llevaba a cabo durante el trabajo:

<table>
<tr><th>Planificación inicial</th><th>Planificación final</th></tr>
<tr><td colspan="2">Adquirir conocimientos sobre la bioinformática y el ámbito del proyecto</td></tr>
<tr><td>Crear robot de descarga de microarrays de gran tamaño del NCBI.
- Identificar las microarrays a descargar.
- Descargar las microarrays.
- Parsear las microarrays.
- Subir microarrays a la base de datos local.
- Eliminación de directorios y ficheros temporales que no son necesarios tras finalizar el proceso de actualización.
- Controlo de errores y recuperación tras la caída del servidor en el proceso de actualización.</td><td>Crear robot de descarga de microarrays de gran tamaño del NCBI.
- Identificar las microarrays a descargar.
- Descargar las microarrays.
- Parsear las microarrays.
- Subir microarrays a la base de datos local.
- Eliminación de directorios y ficheros temporales que no son necesarios tras finalizar el proceso de actualización.
- Controlo de errores y recuperación tras la caída del servidor en el proceso de actualización
- Crear tabla con la información de los códigos y nombres de las especies dado que no estaba creada para las microarrays subidas por el usuario.
- Cambios en el robot actualizador de nombres de genes para adaptarlo a las nuevas microarrays</td></tr>
<tr><td colspan="2">Sincronización con el robot de genes marcadores y programación periódica de ambos robots.</td></tr>
<tr><td colspan="2">Crear interfaces web para la búsqueda de microarrays públicas de gran tamaño en el servidor local y para gestionar la lista de microarrays públicas favoritas de cada usuario.
Realizar las modificaciones necesarias para que funcionen todos los análisis y operaciones con las microarrays públicas de gran tamaño.</td></tr>
</table>

Los planificación inicial resultó ser bastante acertada y tan solo se han producido cambios a la hora de crear el robot de descarga de microarrays de gran tamaño del NCBI, donde se ha añadido la fase de descargar los códigos y nombres de las especies y una fase

para adaptar el robot actualizador de nombres de genes para adaptarlo a las nuevas microarrays.

3.1 Adquirir conocimientos sobre la bioinformática y el ámbito del proyecto

Antes de empezar a realizar el proyecto, tuve que realizar una etapa de adquisición de los conocimientos que me permitiesen desarrollar el proyecto con éxito. Estos conocimientos han sido básicamente sobre la bioinformática, sobre la aplicación y bases de datos que actualmente hay en el servidor revolutionresearch y sobre el entorno del NCBI, lugar escogido como fuente de microarrays que descargaremos.

3.1.1 Adquirir conocimientos sobre la bioinformática

El primer paso en la adquisición de conocimientos fue adentrarse en el mundo de la bioinformática, por lo cual tuve que aprender los conceptos fundamentales con la intención de comprender en su plenitud el proyecto escogido. Los conceptos adquiridos fueron en referencia a la bioinformatica en general y al análisis de microarrays. Primero adquirí conocimientos generales de la bioinformatica. A continuación me centré en el concepto de la microarray en sí, y posteriormente, aprendí algunos de los análisis que se pueden hacer sobre las microarrays para que se pueda obtener información útil de ellas. Estos conceptos se muestran en la sección anterior llamada fundamentos teóricos.

3.1.2 Adquirir conocimientos necesarios sobre la aplicación web y las bases de datos del servidor local

Comprender la aplicación web así como entender su funcionamiento interno y las bases de datos que utiliza es importante y un trabajo previo que hay que realizar, ya que es el lugar donde se integra mi trabajo.

La aplicación web es una herramienta para el estudio de microarrays. En esta herramienta, los usuarios pueden subir y compartir microarrays. Cada usuario puede acceder a las microarrays que él ha subido o que alguien ha compartido con ellos. Una vez un usuario puede acceder a una microarray este puede acceder a los análisis que se han

hecho en el servidor con esa microarray, además de poder realizar sus propios experimentos o ver sus genes.

Internamente, esta aplicación gestiona la información de los usuarios, microarrays, genes, etc. a partir de la base de datos de microarrays local. En la figura 3.1 podemos ver el diseño de esta base de datos.

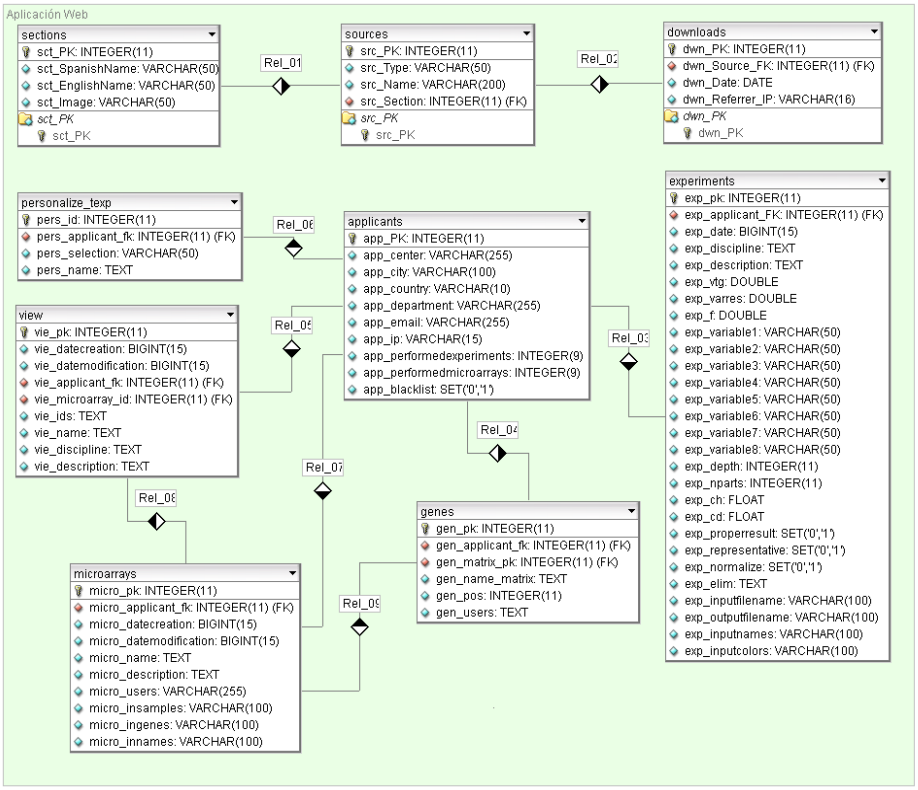

Figura 3.1 Diseño de la base de datos de microarrays.
Esta base de datos es utilizada para gestionar las microarrays subidas al servidor

Las tablas sections, sources y downloads no están relacionadas con la aplicación que he de desarrollar. A continuación, vamos a ver en qué consiste cada una de las demás tablas:

- Applicants: Contiene toda la información sobre los usuarios que utilizan la aplicación.
- Personalize_texp: se ocupa de guardar las preferencias de variables que tiene cada usuario a la hora de visualizar las tablas de experimentos en la aplicación. Por lo tanto, está vinculada con la tabla Applicants, ya que por cada una de las preferencias guardaremos al usuario que le pertenece.
- Experiments: Esta tabla se encargará de almacenar todos los datos de los experimentos que realice cualquier usuario. Está relacionada con la tabla

Applicants, ya que por cada experimento guardaremos el usuario que lo ha realizado.

- Microarrays: esta tabla se encargará de almacenar todas las microarrays que hayan sido subidas por cualquier usuario. Está vinculada con la tabla Applicants ya que por cada microarray guardamos el usuario que la ha subido mediante el campo *micro_applicant_fk*. También se puede destacar el campo *micro_users*, en el cual se guardan los usuarios que tienen acceso a la microarray. Además, recientemente se añadió el campo *micro_tax_id*, el cual almacena la especie de la microarray.
- View: se encarga de guardar las vistas creadas por el usuario. Está vinculada con las tablas Applicants i Microarrays, ya que guarda el usuario y la microarray a la que pertenece la vista.
- Genes: se ocupa de guardar todos los genes de las microarrays que estén dadas de alta. Por cada microarray, guardaremos todos sus genes. Está vinculada con la tabla applicants, ya que en el campo *gen_applicant_fk* almacenamos el usuario que ha subido la microarray que contiene el gen. En el campo *gen_users* tenemos todos los usuarios que pueden acceder a los genes. También está vinculada con la tabla de microarrays ya se guarda a que microarray pertenece cada gen.

Además cada microarray tendrá asociada una tabla Experiments específica. En esta tabla es donde se guardarán los datos y resultados de los experimentos realizados a partir de los datos de esa microarray.

Esta base de datos sirve para gestionar las microarrays, sin embargo, los datos numéricos de las microarrays son guardados en ficheros aparte. Entender como están representadas las micrarrays en el servidor es una parte crucial para poder desarrollar el trabajo. Como se ha explicado en la sección de fundamentos teóricos, una microarray está formada por genes, condiciones experimentales y los valores de expresión de los genes para estas condiciones. En el servidor, una microarray está dividida en tres ficheros.

En primer lugar, tenemos el fichero con los nombres de genes actualizados(genes). En este fichero se guarda todos los nombres de genes de la microarray, cada uno en una línea del fichero. A continuación, tenemos el fichero con los nombres de las condiciones experimentales(snames). En este fichero tenemos los nombres de todas las condiciones experimentales de la microarray, cada una separada en una línea del fichero. Finalmente, tenemos el fichero con los valores de expresión(samples), donde tenemos los valores de

expresión de los genes para cada condición experimental. Estos ficheros deben de estar ordenados de manera que concuerden los valores de expresión con los genes y las condiciones experimentales. De esta manera, en el fichero con valores de expresión(samples), cada línea representará los valores de expresión de un gen, cuyo nombre estará en esa misma línea en el fichero de los nombres de genes. Para separar los valores de expresión de diferentes condiciones experimentales se usan tabulaciones.

Los genes no tienen siempre un nombre especifico, si no que a veces tienen un código de secuencia o un nombre alfa-numérico llamado código Unigene asignado hasta la definición definitiva de uno. Mientras que estos códigos de secuencia o Unigene no vuelven a cambiar, es posible que sí que lo haga el nombre del gen. Por esta razón, en las microarrays distinguiremos dos ficheros de genes: el fichero con los nombres de genes(genesorig) y el fichero con los nombres de genes actualizados(genes). El fichero genesorig contiene los códigos de secuencia o Unigene del gen (si se conocen). El fichero con los nombres de genes actualizados(genes) contiene los nombres definitivos del gen. Entonces, la idea es generar el fichero con los nombres de genes actualizados(genes) a partir del fichero con los nombres de genes(genesorig). En el servidor local existe actualmente un robot que hace esta función, el robot actualizador de nombres de gen. Este

robot mantiene actualizada una base de datos con la correspondencia de códigos de secuencia y Unigene con los nombres de gen. Entonces, a partir del fichero con los nombres de los genes(genesorig) y de esta base de datos, el robot genera el fichero con los nombres de los genes actualizados(genes) que contiene los nombres de cada gen en ese momento. De esta manera, si un gen inicialmente no se le había asignado un nombre, en cuanto le sea asignado el robot actualizador de nombres de gen se encargará de actualizarlo, de igual manera que si cierto gen cambia de nombre, el robot dejará su nuevo nombre en el fichero con los nombres de los genes actualizados(genes). El objetivo será por lo tanto que al descargar las microarrays del NCBI los ficheros con los nombres de los genes(genesorig) queden en el formato adecuado para que posteriormente el robot actualizador de nombres de gen pueda actualizar los nombres de los genes.

Otro robot importante para mi trabajo en el servidor es un robot encargado de mantener actualizada una base de datos de genes marcadores. Este robot se encarga de descargar los genes marcadores de las microarrays del NCBI, agrupándolos por la microarray a la que pertenecen. En este caso, se deberá sincronizar el robot de descarga de genes marcadores para que se ejecute secuencialmente con mi actualización de la base de datos de microarrays y de manera que no tengan ningún directorio o fichero en común.

3.1.3 Adquirir conocimientos necesarios sobre el entorno NCBI

Como se ha comentado anteriormente, como fuente de datos de microarrays utilizaré el National Center for Biotechnology Information (NCBI). El motivo de utilizar el NCBI es porque es una de las mayores fuentes de información biotecnológica del mundo y constantemente se actualiza, además tienen disponibles todas sus bases de datos de manera gratuita y muy accesible.

El NCBI contiene diversas bases de datos[2]. En mi trabajo me interesa especialmente la base de datos Gene Expression Omnibus (GEO) Datasets[3]. Esta base de datos está formada por una gran cantidad de microarrays, accesibles de diferentes maneras y en diferentes formatos, además de incluir imágenes[4] análisis de clusters realizados. Más adelante analizaré diferentes formatos y métodos de descarga de las microarrays.

2 http://www.ncbi.nlm.nih.gov/guide/all/
3 http://www.ncbi.nlm.nih.gov/gds
4 http://www.ncbi.nlm.nih.gov/geo/gds/analyze/analyze.cgi?ID=GDS3719

3.1.3.1 Página web (GEO Datasets)

El primer paso que realicé para entender los datos con los que tenía que trabajar fue familiarizarme con la página web de la base de datos. La misma página web proporciona ayuda[5]. En la figura 3.2 se puede ver el aspecto de la página web. Tenemos disponible una entrada de texto (A) la cual nos permite hacer búsquedas por palabras clave o frases. En esta búsqueda se pueden usar varios términos, como palabras clave, el tipo de datos, o los autores. Se permite configurar(B) el formato de visualización, el número de elementos por página y el método de ordenación. Por cada uno de los resultados(C) de la búsqueda muestra el tipo de datos de que se trata. Estos pueden ser Datasets(GDS), Serie(GSE), Platform(GPL) o Samples(GSM). Más adelante profundizaré más en estos tipos de datos. También aparece el identificador de los datos, el título y la especie. A continuación (D) aparece una breve descripción del conjunto de datos, el tipo de datos, si aparecen subconjuntos dentro del conjunto de datos y el número de condiciones experimentales. En la parte derecha(E) de cada resultado de la búsqueda encontramos links hacia otras bases de datos. Más a la derecha(F) aun tenemos la opción de filtrar la búsqueda según el tipo de datos. Por cada conjunto de datos también tenemos una imagen de los clusters(G). Clicando sobre la imagen podemos verla en mayor tamaño. Finalmente tenemos la opción de buscar datos relacionados en otras bases de datos(H) .

5 http://www.ncbi.nlm.nih.gov/geo/info/datasets.html#d

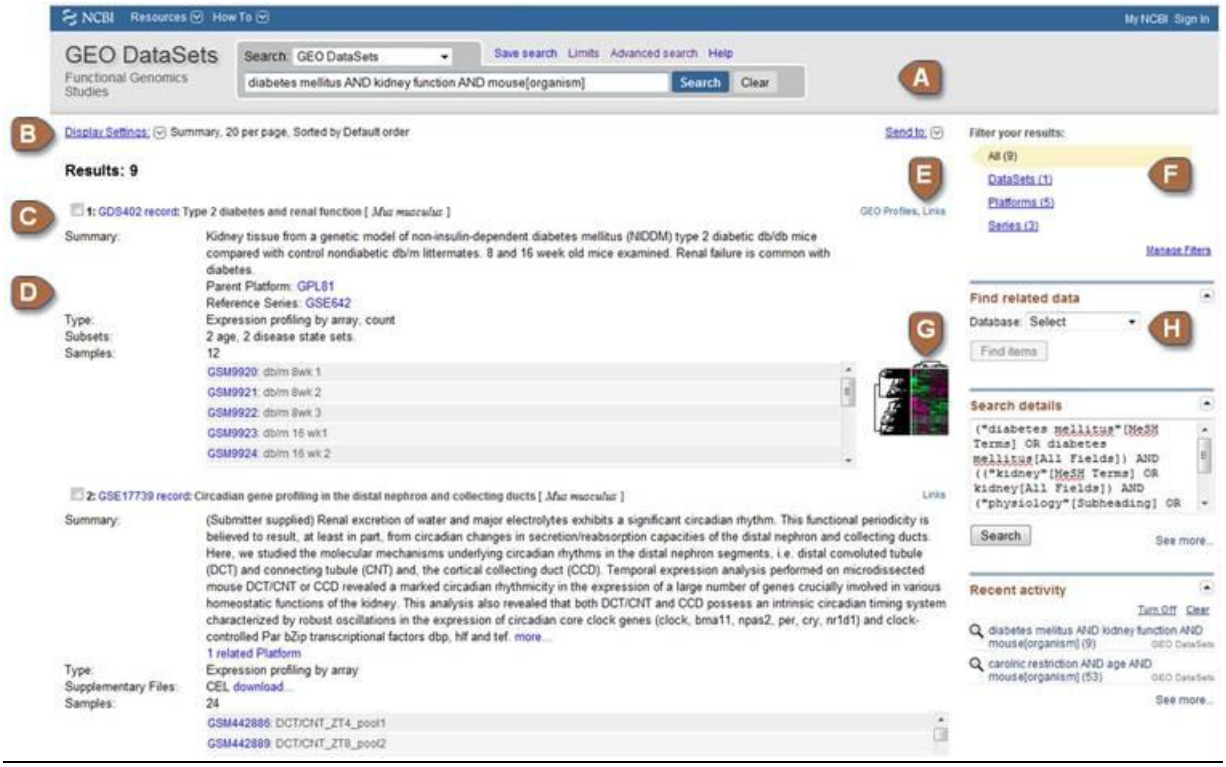

Figura 3.2 Aplicación web GEO Datasets del NCBI

Tenemos la opción de hacer una búsqueda insertando palabras clave o frases en la caja de texto(A). Podemos cambiar las opciones de visualización de la búsqueda(B). Por cada resultado de la búsqueda(C) podemos ver su tipo de datos, título y especie. También se muestra(D) otra información como la descripción o el número de condiciones experimentales, así como links a otras bases de datos(E). Tenemos la opción de filtrar la búsqueda según el tipo de datos(F). Se nos muestra una imagen de los clusters realizados en los análisis del NCBI(G). Finalmente está la opción de hacer la búsqueda de datos relacionados en otras bases de datos(H).

Si clicamos uno de los resultados de la búsqueda aparece en el navegador la información de ese conjunto de datos. Podremos ver una imagen como la de la figura 3.3. Podemos observar información sobre el conjunto de datos(I), como el título o la descripción. Podemos ver de nuevo la imagen de los clusters(J). También se nos muestra diversas opciones de descarga del conjunto de datos. En la parte inferior(L) tenemos información sobre las herramientas de análisis que proporciona el NCBI.

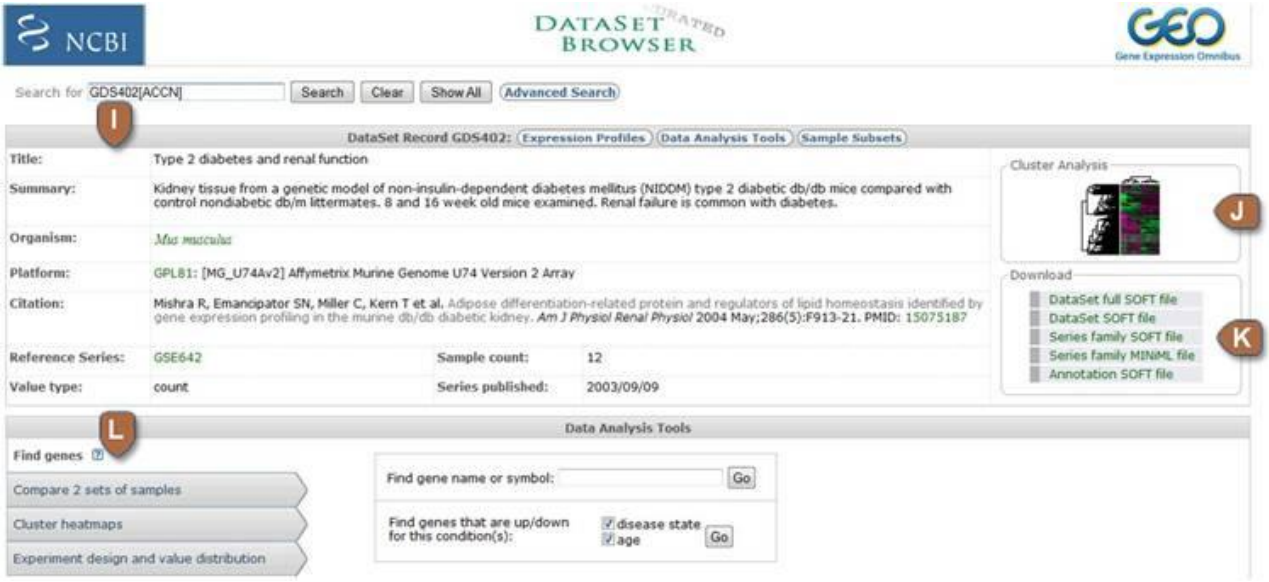

Figura 3.3 Ejemplo de Dataset Record de la base de datos GEO Datasets del NCBI Se puede observar información sobre el conjunto de datos(I), una imagen de los clusters(J), las diversas opciones para descargar el conjunto de datos(K) y herramientas de análisis del NCBI(L).

3.1.3.2 Analizar los formatos y los métodos de descarga de las microarrays

Una vez comprendido el funcionamiento de la página web de la base de datos GEO datasets, el siguiente paso es conocer como podré descargar la información, así como los tipos de datos disponibles.

Como hemos visto en la anterior sección, en la base de datos GEO Datasets nos podemos encontrar con diferentes tipos de datos:

- Platform(GPL): Estos registros se componen de una breve descripción de la microarray y una tabla de datos por cada condición experimental de la microarray. En la figura 3.4 se puede observar más detalladamente el formato de las GPL. Al inicio podemos encontrar información de la microarray, como el título, la especie o las muestras que contiene. A continuación tenemos información acerca de los genes de los cuales se ha obtenido los valores de expresión. Posteriormente, por cada muestra encontramos los valores de expresión de los genes. Una de las razones que

dificulta el tratamiento de estos registros es que no en todas aparece la misma información sobre los genes.

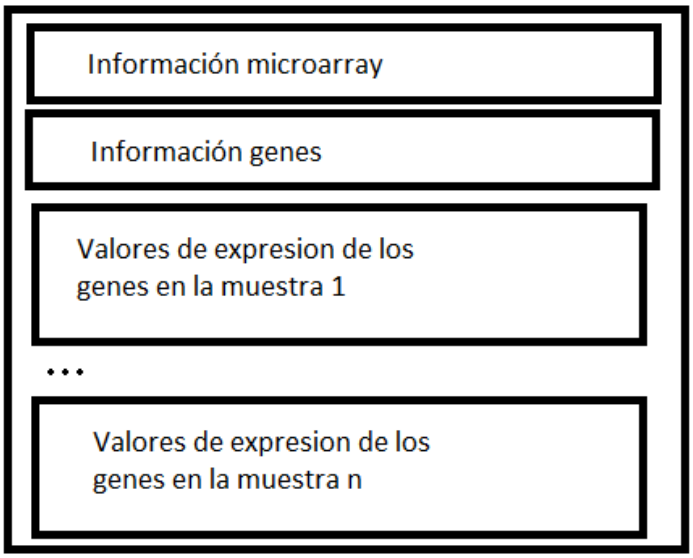

Figura 3.4 Formato de una Platform(GPL) de la base de datos GEO Datasets del NCBI

En primer lugar tenemos información descriptiva sobre la microarray. A continuación aparece información sobre los genes de la microarray. Finalmente, tenemos los valores de expresión para cada una de las condiciones experimentales de la microarray.

- Samples(GSM): Estos registros describen una condición experimental y incluyen los valores de expresión de los genes en esa condición experimental. Este tipo de datos no me será de mucha utilidad ya que simplemente son condiciones experimentales sueltas con sus valores de expresión.
- Series(GSE): Estos registros contienen los valores de expresión de una serie de muestras relacionadas y una descripción sobre el estudio. Tienen un formato muy similar a las GPL y al igual que ellas, no siempre aparece la misma información de los genes.
- Datasets(GDS): Estos registros, a diferencia de los anteriores, son creados por el NCBI a partir de los anteriores. Los datasets representan una colección de muestras biológicas y estadísticas comparables. Los datos están normalizados y son la base para las herramientas de análisis de la base de datos. Por otra parte hay varios formatos de descarga de los ficheros de las GDS. Por una parte, tenemos el DataSets SOFT file. En la figura 3.5 podemos ver el formato de este tipo de ficheros. Vemos que en la

parte superior aparece información sobre la microarray y sobre las muestras que esta contiene. A continuación, tenemos los valores de expresión para cada gen para cada una de las muestras.

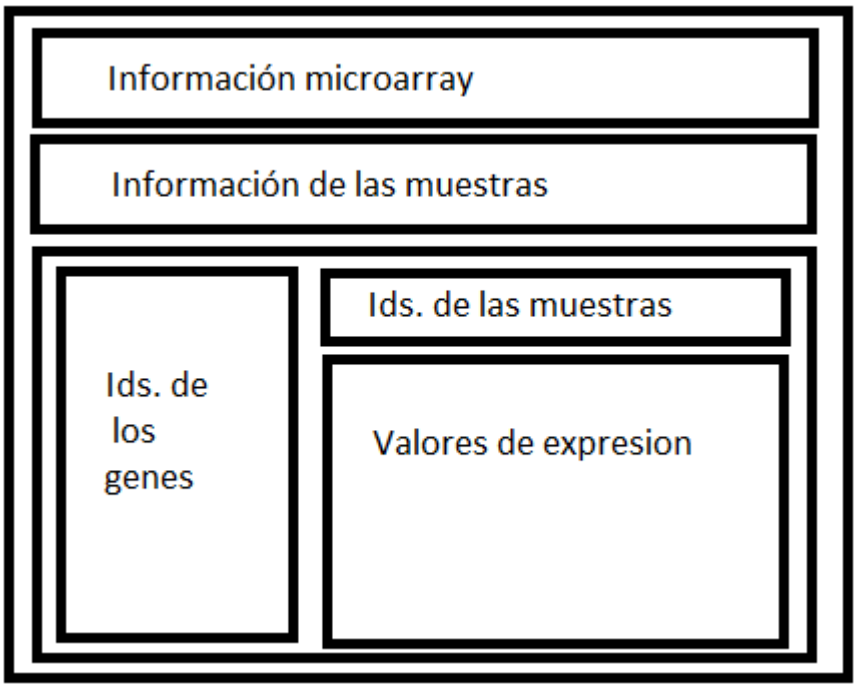

Figura 3.5 Formato de un DataSets(GDS) SOFT file *de la base de datos GEO Datasets del NCBI*

En el fichero GDS SOFT tenemos en la parte superior la información sobre la microarray. A continuación tenemos información sobre las muestras. Finalmente tenemos todos los valores de expresión para cada gen y para cada muestra.

Los GDS SOFT pueden descargarse mediante los links que aparecen en la página web o por FTP. Por otra parte, podemos descargar otro tipo de archivo de la GDS que tan solo incluye los identificadores de genes, los identificadores de muestras y los valores de expresión. Además, en este tipo de archivo los genes vienen ordenados según los análisis de los clusters realizados en el NCBI, lo que puede suponer una gran ventaja a la hora de

analizar la microarray. Este tipo de ficheros(A partir de ahora GDS clustering) puede descargarse mediante HTTP[6]. Finalmente, disponemos del fichero llamado DataSet full SOFT file(a partir de ahora GDS Full). Como se puede ver en la figura 3.6 los GDS Full contienen un formato muy similar a los DataSet SOFT file. La diferencia, es que este último contiene información adicional, como puede ser los códigos los códigos de secuencia de los genes o los códigos Unigene, lo cual nos podrá ser especialmente útil a la hora de adaptar nuestros ficheros con los nombres de genes(genesorig) para futuras actualizaciones. En este caso, cabe destacar que aunque en algunos genes nos encontramos que cierta información está vacía, la cabecera siempre es la misma. Este formato también puede ser descargado mediante los links de la página web o por FTP.

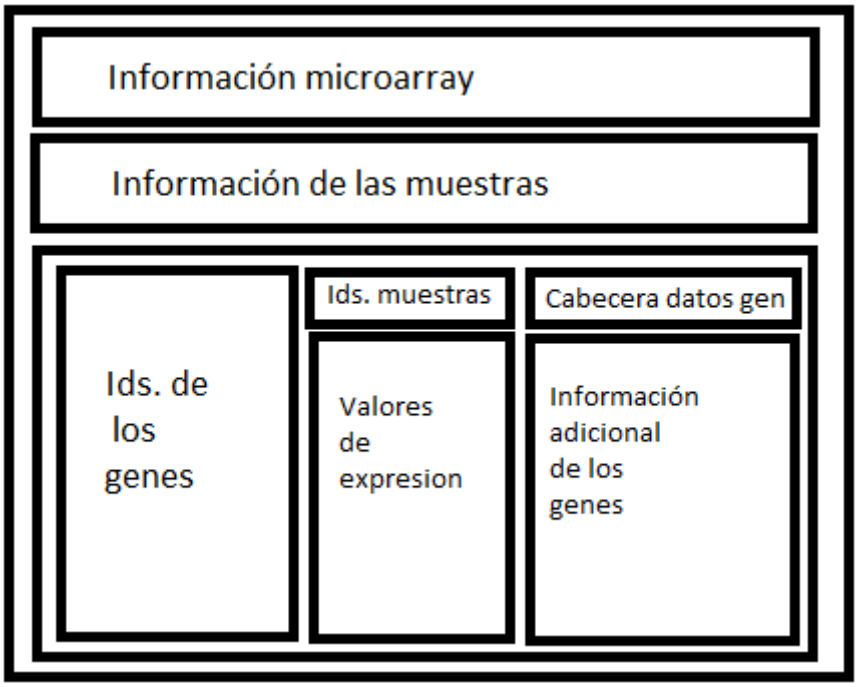

Figura 3.6 Formato de una GDS Full *de la base de datos GEO Datasets del NCBI*

Al inicio tenemos información de la microarray, seguida posteriormente de información de las muestras de la microarray. A continuación tenemos una tabla con los identificadores de los genes, sus valores de expresión y datos adicionales de cada gen de la microarray.

De todos los formatos que hemos visto, el único que nos puede servir son los GDS. Concretamente, hemos visto que en las GDS Full aparece una amplia información de los genes, por lo que nos podrán ser muy útiles a la hora de formar los ficheros con los nombres de genes. También hemos observado que se pueden obtener unos ficheros de las GDS en

6 http://www.ncbi.nlm.nih.gov/geo/gds/analyze/getData.cgi?ID=GDS3715

que los genes están ordenados por el análisis de los clusters realizado por el NCBI(GDS Clustering), lo cual puede ser de ayuda para los investigadores. Aunque los GDS Clustering tan solo contienen un identificador del gen dentro de la microarray, si hacemos uso de los dos podemos llegar a obtener los genes ordenados además de obtener toda la información necesaria del gen obtenida en el fichero GDS Full.

3.1.3.3 E-Utils

E-Utils es una herramienta que facilita el acceso a los datos del NCBI fuera de la interfaz web mediante consultas, lo cual puede ser útil para obtener datos a partir de una búsqueda.

Para acceder a la herramienta, se construye una URL y se envía a los servidores del NCBI. En los servidores se procesa la consulta y se envía la respuesta en formato XML. Esta URL está formada por cuatro partes:

- Una primera parte, la cual es común a todas las consultas y permite el acceso a Eutils: http://eutils.ncbi.nlm.nih.gov/entrez/eutils.
- La segunda parte determina la aplicación a utilizar con E-Utils. Puede ser cualquiera de las siguientes:
  - eSearch: a partir de una consulta responda con la lista de identificadores únicos(UID) dada una base de datos.
  - eSummary: a partir de una lista de UIDs responde con los registros.
  - ePost: acepta una lista de UIDs guardándolos como conjunto en el historial del servidor y responde con la correspondiente “query key” y “web environment”. Estos valores hacen referencia al conjunto o consulta asociada, así el servidor recuerda el conjunto o consulta anteriormente realizada. Las opciones también sirven para el resto de aplicaciones.
  - eLink: a partir de una lista de UIDs de una base de datos permite relacionar dicha lista con los IDs correspondientes a otra base de datos.
  - eInfo: proporciona el número de registros indexados en un campo de una base de datos, la fecha de la última actualización y los enlaces que están disponibles con otras bases de datos.
- En la tercera parte se indica a que base o bases de datos se realiza la consulta.

- En la última parte se introducen las condiciones de la búsqueda. Consiste en el término de entrada a la base de datos, opciones opcionales al término, opciones opcionales a la consulta y otras opciones obligatorias. Por ejemplo, en esta parte de la consulta se podría indicar que se quieren recuperar microarrays que contengan un número de condiciones experimentales mayor a 70.

Hay que tener en cuenta, que para no sobrecargar los servidores, el NCBI imponen una serie de condiciones para utilizar la herramienta E-Utils. Estas condiciones son:

- Ejecutar programas los fines de semana o entre las 21:00 y 5:00 del Este durante la semana para una serie de más de 100 solicitudes.
- No hacer más de tres peticiones por cada segundo.
- Utilizar el parámetro email y tool para el software distribuido, de manera que puedan analizar el proyecto y ponerse en contacto si hubiese algún problema.

En caso de que no se cumpla alguna de estas condiciones, puede producirse un corte en la conexión con los servidores.

3.2 Crear robot de descarga de microarrays públicas de gran tamaño del NCBI.

Una vez adquiridos los conocimientos necesarios para poder realizar el proyecto, inicié la fase de crear el robot de descarga de microarrays públicas de gran tamaño del NCBI. La idea es descargar periódicamente las microarrays del NCBI que contienen 70 o más condiciones experimentales aprovechando la base de datos y programas expuestos en las secciones anteriores. Para ello, programé un robot que se encarga de la actualización.

Para afrontar el problema, este lo dividí en varias partes. Primero, traté de resolver la manera de saber que microarrays debo descargar. Estas microarrays son aquellas que contienen 70 o más condiciones experimentales y que son nuevas desde nuestra última actualización. Posteriormente traté de encontrar el mejor método de descarga de las microarrays. A continuación hay que parsear los ficheros de las microarrays descargados de manera que se generen ficheros que se adecuen al formato de las microarrays del servidor y se obtenga la información necesaria de la microarray. Una vez generados los ficheros habrá que subir a la base de datos las microarrays descargadas. Finalmente, también será necesario un método que haga que el robot sea robusto a posibles errores a mitad de la

actualización, así como un procedimiento para borrar todos los archivos y directorios temporales creados.

En la figura 3.7 podemos ver un diagrama del proceso de actualización de las microarrays del NCBI. Cuando se inicia la ejecución el primer proceso que se realiza es el de obtener las microarrays a descargar. Este proceso ha de generar un fichero con los identificadores únicos del NCBI de las microarrays que se tendrán que descargar. A continuación, se ejecuta el proceso de descargar los ficheros de microarrays. Este proceso utilizará el fichero con los identificadores únicos del NCBI de las microarrays que se han de descargar generado en la etapa anterior. Una vez acabado, se tendrán en el servidor los ficheros de las microarrays del NCBI que necesitamos y pasaremos a la siguiente etapa. En esta, parsearemos los ficheros descargados, de manera que generaremos los ficheros con los valores de expresión(samples), con los nombres de las condiciones experimentales(snames), con los nombres de genes actualizados(genes) y con los nombres de genes(genesorig) en el formato adecuado para el servidor. También se insertaran en la base de datos local los registros de la microarray. Esta etapa se realiza una vez por cada microarray. Finalmente se realiza un proceso para eliminar todos los ficheros y directorios temporales que ya no necesitamos.

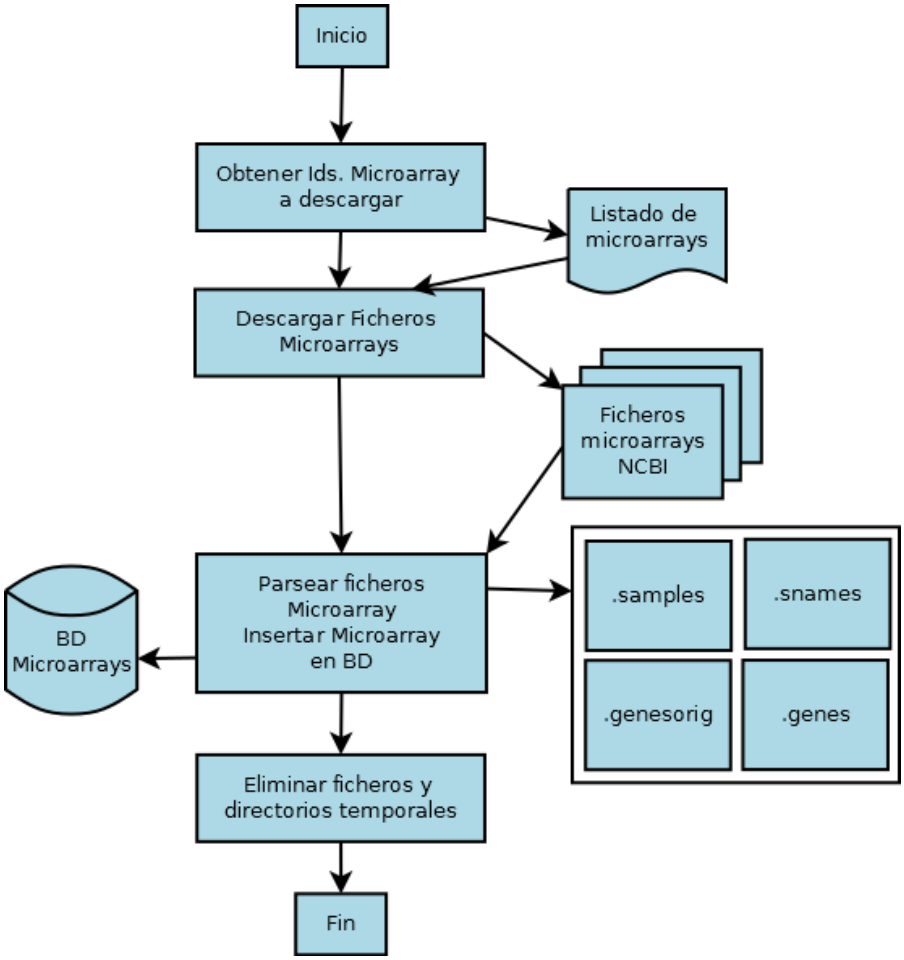

Figura 3.7 Diagrama proceso actualización base de datos de microarrays públicas de gran tamaño del NCBI

En la primera etapa se obtienen los identificadores de las microarrays a descargar y se escriben en el fichero con los identificadores únicos del NCBI de las microarrays a descargar. Este fichero es utilizado por la siguiente etapa, dónde se descargan los ficheros de las microarrays, dejando en el servidor los ficheros de las microarrays que necesitamos. Tras esto se procede a la etapa de parsear los ficheros de la microarray, donde se generan los ficheros con los valores de expresión(samples), con los nombres de las condiciones experimentales(snames), con los nombres de genes actualizados(genes) y con los nombres de genes(genesorig) en el formato adecuado para el servidor. También se suben los registros a la base de datos local. Finalmente, se eliminan los ficheros y directorios temporales que se han creado.

En las siguientes secciones, se explicaré más detalladamente cómo he realizado cada una de las etapas.

Por otra parte, todo el proceso ha sido implementado en Perl[12]. El motivo de la elección ha sido porque contiene funciones que facilitan parsear ficheros, además que permite realizar consultas E-Utils fácilmente, algo que como veremos en la siguiente sección será necesario. A pesar de no ser tan rápido como otros lenguajes como el C, esto no es realmente importante ya que al no ser un programa online no es tan necesaria la velocidad.

3.2.1 Identificar las nuevas microarrays públicas de gran tamaño a descargar del NCBI

Para actualizar la base de datos, lo primero en lo que me centro es en cómo puedo obtener las microarrays que se han de descargar. Las microarrays a descargar serán aquellas que contienen 70 o más condiciones experimentales que son nuevas desde la última actualización realizada, y además, que son del tipo Datasets(GDS), que como hemos visto en la sección 3.1.3.2 son los que nos permitirán obtener la información necesaria para ser analizadas en el servidor web. En el NCBI, los DataSets (GDS) contienen un identificador único. El objetivo por lo tanto será obtener estos identificadores únicos de las GDS del NCBI que se han de descargar al servidor local.

Una manera para obtener los identificadores de las microarrays es realizar una consulta E-Utils. De esta manera en la consulta podremos realizar el filtro para obtener solo los identificadores de las microarrays que necesitamos descargar.

Las primera condición en que pienso sobre las microarrays a descargar, es que ha de tener 70 o más condiciones experimentales. Para ello construyo la siguiente consulta:
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=gds&term=70:5000[Number+of+Samples]&retmax=5000
La aplicación utilizada es esearch. La parte db=gds de consulta se indica que se realizará sobre la base de datos Geo Datasets. A continuación, en el término, indicamos que queremos aquellas muestras desde 70 a 5000 condiciones experimentales. El tope de 5000 se ha introducido ya que es un número muy superior al máximo de condiciones experimentales que tiene una microarray del NCBI y por lo tanto no quedará ninguna microarray con un número superior. La última parte indica que se recuperaran hasta 5000 identificadores en esta consulta. Esto se introduce porque por defecto se recuperan hasta 20 identificadores, lo cual podría suponer realizar más de una consulta, mientras que de esta manera nos aseguramos que en una sola consulta obtendremos todas los identificadores. El problema de esta consulta es que no indicamos el tipo de conjunto de datos que queremos, por lo que nos puede devolver identificadores de GPL o GSE. Por este motivo, modifiqué la consulta quedando de esta manera:
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=gds&term=70:5000[Number+of+Samples]gds[Entry+Type]&retmax=5000
En este caso, mediante la opción Entry Type filtramos los identificadores a aquellos que solo sean Datasets(GDS).

Una vez obtenidos los identificadores de las microarrays con 70 o más condiciones experimentales y que sean del tipo GDS, faltaba ocuparse de cómo distinguiría que microarrays ya se poseen en la base de datos de anteriores actualizaciones y cuáles no. La primera opción que pensé fue en guardar la fecha de la última actualización en nuestro servidor en un fichero y posteriormente, al hacer las actualizaciones, introducir esta fecha en la consulta E-Utils como opción de Update Date de E-Utils. Al final, descarté esta opción debido a que la fecha de las microarrays en E-Utils podía no coincidir con la fecha de subida de los archivos de las microarrays en el NCBI. La solución que adopté finalmente fue crear un campo nuevo en la tabla microarrays de la base de datos para guardar los identificadores de las microarrays en el NCBI. Este campo se llama micro_gds_id y en aquellas microarrays que no sean del NCBI valdrá NULL. De esta manera a la hora de descargar podemos hacer

la comprobación si ya tenemos la microarray buscando el identificador en el campo micro_gds_id. Esta opción además nos proporcionará una ventaja a la hora de implementar la aplicación web, ya que nos permitirá distinguir fácilmente entre las microarrays subidas por los usuarios y las que son del NCBI, además de poder usar estos identificadores para usar links al NCBI. Por lo tanto, nos proporcionará una mayor operatividad web.

El resultado de la consulta E-Utils es un XML que contiene los identificadores encontrados según la consulta. Para obtener estos identificadores es necesario parsear el XML que nos devuelve la consulta.

Tras realizar la consulta deberemos hacer consultas locales a nuestra base de datos para saber que microarrays ya tenemos y cuáles no. En la figura 3.8 vemos como queda finalmente esta etapa. Inicialmente hacemos la consulta E-Utils para obtener las microarrays con 70 o más condiciones experimentales y que sean del tipo GDS. Se obtiene un XML como respuesta. Parseando este XML obtenemos los identificadores de las microarrays del NCBI con 70 o más condiciones experimentales. Con estos identificadores hacemos consultas a la tabla de microarrays y obtenemos los identificadores de las microarrays anteriores que no tenemos y que deberemos descargar. Estos identificadores se escribirán en un fichero de identificadores únicos del NCBI de las microarrays a descargar.

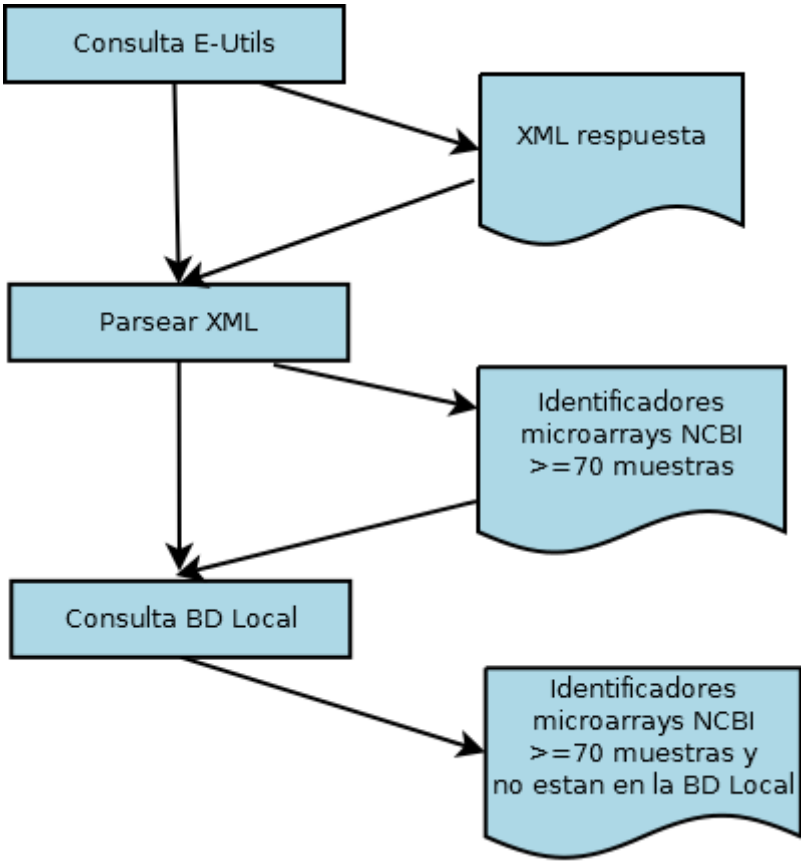

Figura 3.8 Esquema proceso de obtención de los identificadores únicos de las microarrays a descargar del NCBI

Inicialmente hacemos la consulta E-Utils para obtener las microarrays con 70 o más condiciones experimentales y que sean del tipo GDS. Se obtiene un XML como respuesta. Este XML se parsea obteniendo las microarrays del NCBI con 70 o más condiciones experimentales. Con estos identificadores hacemos consultas a la tabla de microarrays y obtenemos los identificadores de las microarrays anteriores que no tenemos y que deberemos descargar.

3.2.2 Descargar los ficheros de las microarrays nuevas del NCBI al servidor

Esta etapa se encarga de descargar los ficheros necesarios de las microarrays del NCBI. Como se ha visto en la sección 3.1.2.2 los ficheros escogidos serán los siguientes:

- Ficheros GDS Full para obtener toda la información de los genes que permita posteriormente realizar la actualización de nombres de gen.
- Ficheros GDS Clustering para obtener los genes, sus valores de expresión y las condiciones experimentales según el orden en que aparecen en los análisis de clusters realizados en el NCBI.

Los primeros pueden ser descargados por FTP mientras que los segundos se pueden descargar por HTTP.

Para saber que microarrays hay que descargar se leerá el fichero de identificadores únicos del NCBI de las microarrays a descargar generado en la etapa anterior.

3.2.3 Parsear los ficheros de las microarrays descargadas obteniendo la información que describe la microarray y generando los ficheros con los nombres de las condiciones experimentales(snames), con los valores de expresión(samples), con los nombres de genes(genesorig) y con los nombres de genes actualizados(genes)

Una vez tenemos los ficheros de las microarrays, ya podemos procesar a parsearlos con el fin de obtener los ficheros de las microarrays en el formato en el que los tenemos en el servidor local.

En la figura 3.9 podemos ver un esquema general del proceso. Lo primero que se hace es parsear el fichero GDS Full. Al hacer este parsing generamos lo siguiente:

- Fichero con la información descriptiva de la microarray(Info microarray en la figura). Es fichero con la información de la microarray que necesitamos, como el título y la descripción de la microarray.
- Fichero con los identificadores de los genes en la microarray y la información de los genes sin el orden de los clusters del NCBI(genes sin orden en la figura). Un fichero en el que tenemos una lista donde cada elemento es un identificador del gen en la microarray y toda la información necesaria del gen. En este fichero los genes están ordenados según el orden del archivo GDS Full.
- Fichero con las condiciones experimentales de la microarray sin el orden de los clusters del NCBI(cond. sin orden en la figura). Tendremos una lista donde cada elemento es el identificador de la condición muestral y su descripción. En este fichero las condiciones experimentales están en el orden del archivo GDS Full.

Posteriormente, se utilizan los ficheros de GDS Clustering, y los de genes y condiciones experimentales creadas anteriormente para generar los ficheros con los valores de expresión(samples), con los nombres de las condiciones experimentales (snames) y con los nombres de genes(genesorig). Una vez obtenido el fichero con los nombres de genes(genesorig) se crea el fichero con los nombres de genes actualizados (genes). Aunque en este proceso ya se crean los ficheros de la microarray, aquí aun no han sido movidos al directorio real de la microarray.

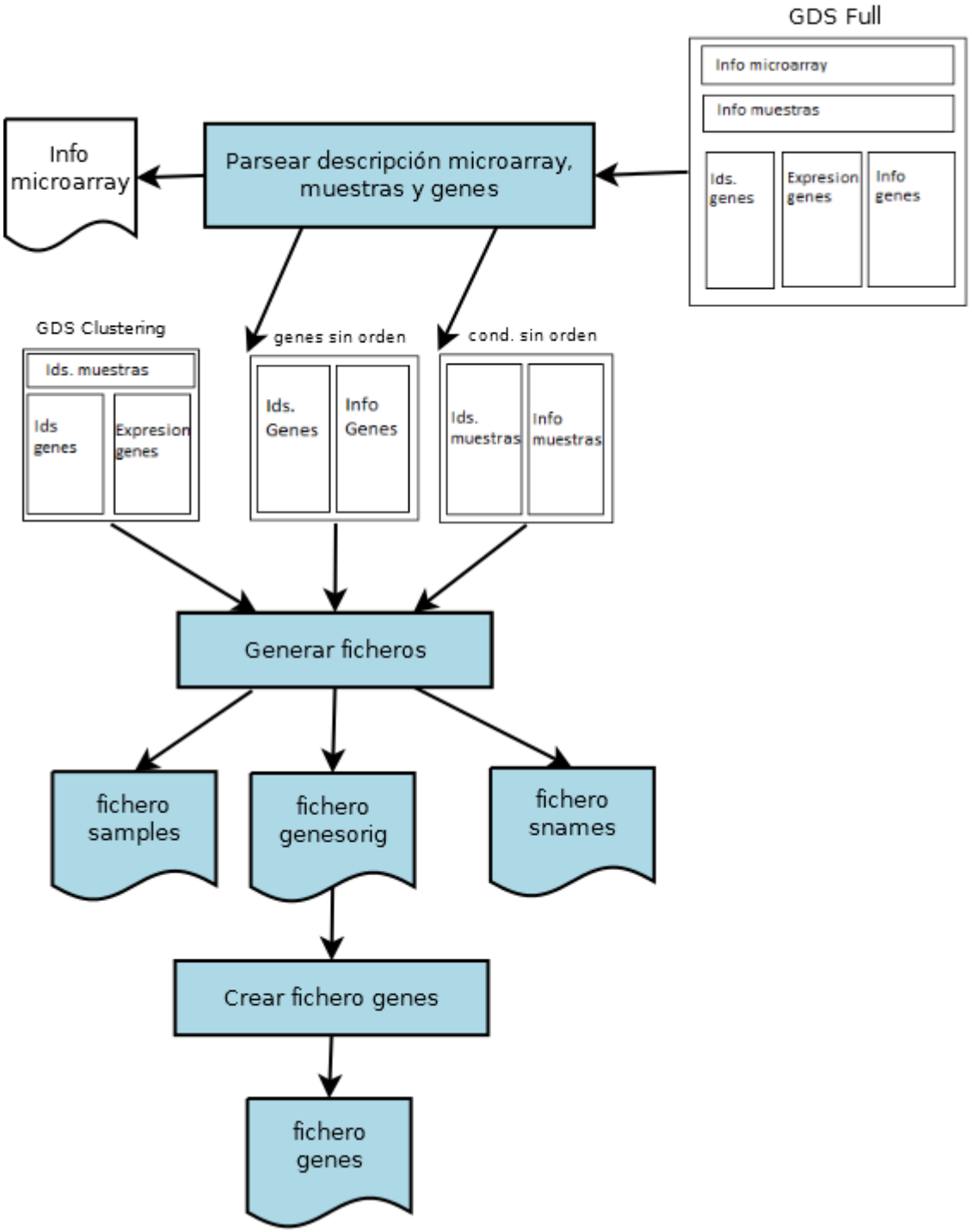

Figura 3.9 Esquema del parsing de los ficheros de las microarrays descargadas.

Primero parseamos la GDS Full, para obtener la información de la microarray,de los genes y de las condiciones experimentales que necesitamos, creando un fichero con la información descriptiva de la microarray, un fichero con los identificadores de los genes en la microarray y la información de los genes sin el orden de los clusters del NCBI, y otro fichero con las condiciones experimentales de la microarray sin el orden de los clusteres del NCBI. A continuación, a partir de estos ficheros y del GDS Clustering, generamos los ficheros con los valores de expresión(samples), con los nombres de genes(genesorig) *y* con los nombres de

las condiciones experimentales(snames). *Finalmente, a partir del fichero con los nombres de gen(genesorig) generamos el fichero con los nombres de gen actualizados (genes).*

3.2.3.1 Parsear la información que describe la microarray

A la hora parsear el fichero GDS Full se ha de obtener la información que describe la microarray. Esta información se puede obtener de la parte superior del fichero GDS Full. La información que se obtiene es el título de la microarray, su descripción, la especie, el identificador de la microarray en el NCBI y el número de muestras. Todos estos campos son fáciles de obtener parseando, a excepción de la especie.

Aunque en la microarray tenemos el nombre de la especie el problema viene dado porque en la base de datos local almacenamos el código de la especie y no el nombre. En la base de datos local, no tenemos ninguna tabla con la información de los códigos de las especie, aunque con el nombre de la especie obtenido en la GDS Full, podríamos obtener su código con una consulta E-Utils. Sin embargo, pensando que en el futuro en la implementación web se tenía pensado añadir la opción de hacer búsquedas de microarrays por especie decidí analizar mejor la situación. Si obteníamos el código de la especie haciendo una consulta E-Utils guardaríamos el código de la especie en la microarray. Entonces al no disponer en la base de datos local ninguna relación entre códigos y nombres, cuando un usuario introduce una búsqueda por especie se debería de transformar la especie que ha introducido el usuario a código. Este código podría ser nuevamente obtenido por una consulta E-Utils[7]. El problema surge al hacer el listado de las microarrays que se han encontrado. Al mostrar el resultado de las búsquedas, se desea mostrar entre otra información el nombre de la especie de la microarray, tanto si la búsqueda se ha hecho por especie o se ha hecho por palabra clave. En la tabla de microarrays sólo aparece el código de la microarray, y por lo tanto, si queremos mostrar microarrays con y hay especies distintas se deberían de realizar varias consultas E-Utils. Estas consultas, al hacerse seguidas, tendrían muchas probabilidades de no cumplir una de las normas del NCBI, que es no hacer más de 3 consultas E-Utils por segundo. Esto podría producir que se cortase la conexión.

7 http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=taxonomy&term=Mus%20musculus[ScientificName]

Tras hacer esta observación decidí que la mejor idea era mantener una tabla en la base de datos con la relación del código de la especie con su nombre. La decisión que tomé fue crear inicialmente una tabla con dos campos, uno para los códigos de las especies y otro para los nombres. Esta tabla se crearía en un programa de actualización de base de datos de especies antes e independientemente de la actualización de microarrays. Para crear esta tabla, se hace la descarga desde el FTP del NCBI del archivo taxdmp.zip[8]. Este es un archivo comprimido. Al descomprimirlo utilizaré el archivo names.dmp. En este archivo, por cada especie, tenemos el código de la especie y su nombre, entre otra información. El programa recorrerá este archivo insertando en la tabla todas las especies, de esta forma, tendremos todos los códigos de las especies.

Volviendo al parsing de los ficheros, al tener en la base de datos local una tabla con los códigos y nombres de cada especie, en lugar de consultar el código de la especie por E-Utils se hará una consulta a la base de datos local. En caso de que la especie no encontrase el código en la base de datos local porque es nueva se haría una consulta E-Utils, y se insertaría la nueva especie en la base de datos local.

Aunque el programa de actualización de base de datos de especies sólo se ejecuta una vez, este está preparado para que se ejecute periódicamente si se necesitase mantener todas las especies actualizadas, ya que tras finalizar cada nueva ejecución guarda en la base de datos local todas las especies existentes en el NCBI. La razón por la que no se ha programado el robot para que se actualice periódicamente es porqué el ritmo que aparecen nuevas especies, y aparece luego nueva información génica sobre estas nuevas especies es muy lento, ya que la información génica que se investiga es sobre las especies ya conocidas.

Por otra parte, las condiciones experimentales de una microarray se pueden dividir en varias clases. Cada clase, agrupa condiciones experimentales de cierto tipo. Por ejemplo, en una microarray, se pueden agrupar en una clase las condiciones experimentales que son del tipo “en estado de enfermedad” y en otra clase las condiciones experimentales que son del tipo “en estado de salud”. Esta información está disponible en los ficheros GDS Full de las microarrays. Para los investigadores, podría ser útil realizar búsquedas introduciendo el tipo de condición experimental que desea. Sin embargo, en las descripciones de las microarrays ya aparece esta información, por lo que no ha sido necesario añadir información adicional en la base de datos local.

8 ftp://ftp.ncbi.nih.gov/pub/taxonomy/

3.2.3.2 Parsear las condiciones experimentales de la microarray

Tras obtener toda la información de la microarray, se pasa a obtener todas las condiciones experimentales de la microarray. Se creará un fichero en que tengamos los identificadores de las muestras y su descripción separados por puntos. Hay que destacar que los nombres de las muestras no tienen realmente tanta importancia ya que muchas veces se trabaja más con clusters.

3.2.3.3 Parsear los genes de la microarray de manera que posteriormente se puedan actualizar por el robot actualizador de nombres de gen

Una vez parseadas las condiciones experimentales, se pasa a parsear los genes que aparecen en la GDS Full. La idea es guardar la información del gen, de manera que posteriormente el robot actualizador de nombres de gen pueda actualizar los nombres de los genes de la gds si cambian. Por lo tanto, para realizar esta parte he tenido que fijarme como funciona actualizador de nombres de gen.

Este robot lee el fichero con los nombres de genes(genesorig) y a partir de él crea el fichero con los nombres de genes actualizados(genes). Para esto, se tiene una base de datos de información de genes en la base de datos local en la que podemos destacar las tablas que aparecen en la figura 3.10. En primer lugar, podemos ver la tabla gene_info. En esta tabla podemos destacar el campo GeneID, que contiene un identificador del gen en el NCBI, el Symbol, que contiene el símbolo del gen del NCBI(Gene Symbol en el NCBI) y Description, que contiene la descripción del gen(Gene Title en el NCBI). A continuación tenemos la tabla unigene. En esta tabla tenemos dos campos, el UniGene_cluster, que contiene el código Unigene de un gen, y el GeneID, que contiene el código del gen para ese código Unigene. Los códigos Unigene son unos códigos del NCBI que se asignan a los genes cuando todavía no han sido identificadas las funciones. Por lo tanto, es posible que exista un código Unigene de un gen el cual todavía no tiene GeneID, símbolo y descripción, o incluso que de un código Unigene del que tengamos GeneID, símbolo y descripción estos cambien. Por lo tanto, con el código Unigene nos servirá para obtener el GeneID, Symbol y Description de un gen. La última tabla, la tabla sequences, contiene los campos UniGene_cluster y sequence. El Unigene_cluster contiene el código Unigene de un gen, el

mismo que en la tabla Unigene_cluster. El campo Sequence contiene un código de seqüencia de un gen. Si tenemos un código de secuencia de un gen, podremos obtener el código Unigene del gen. Con este código Unigene podremos obtener el código GeneID, y con este último podremos tener la información del gen.

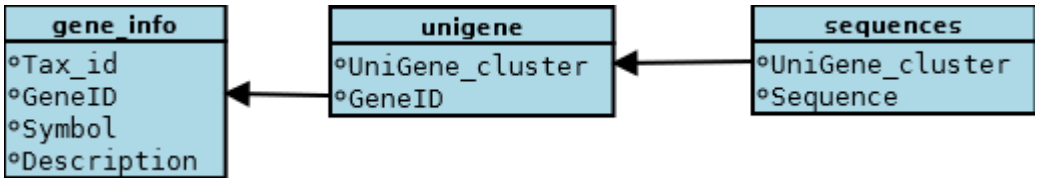

Figura 3.10 Tablas relevantes de la base de datos local de información de genes
En la tabla gene_info contenemos la información sobre el gen. En la tabla unigene tenemos la relación entre los códigos de los genes y los códigos Unigene. En la tabla sequences tenemos la relación entre los códigos unigene y los códigos de secuencia.

Todas estas tablas son actualizadas periódicamente por un robot. Al actualizar las tablas, el robot accede a todos los ficheros con los nombres de genes(genesorig) de las microarrays. En estos ficheros mira si tenemos códigos unigene o códigos de secuencia en cada gen, y a partir de ellos, hace consultas en la base de datos local y deja el símbolo y la descripción del gen en el fichero con los nombres de los genes actualizados(genes). En caso de no encontrar información en la base de datos en algún gen, deja en el fichero con los nombres de los genes actualizados(genes) la misma información que aparece en el fichero genesorig.

Un ejemplo de código de gen es el “Hs.2231”. La primera parte indica la especie del gen, en este caso “Hs” indica “Homo sapiens”. La segunda parte es un código que distingue los genes de una misma especie. Este código es detectado por el robot mediante una expresión regular, que detecta en la primera parte un código alfanumérico, un punto a continuación, y finalmente un código numérico. Cuando detecta el código de secuencia lo busca en la tabla unigene. Si encuentra un GeneID para este código Unigene lo buscara en la tabla gene_info y obtendrá el símbolo y la descripción, que es lo que se escribirá en el fichero con los nombres actualizados (genes).

Un ejemplo de códigos de secuencia tal como se guardan en el fichero con los nombres de genes(genesorig) es el “[5’:NM_003144, 3’:NM_003145]”. En este caso, realmente tenemos dos códigos de secuencia, el NM_003144 y el NM_003145, ya que un mismo gen puede tener dos códigos de secuencia. Podría ocurrir el caso de que tan solo tuviésemos un código de secuencia y este podría estar en el 5 o en el 3. El robot detecta

estos códigos de secuencia mediante una expresión regular, detectando el 5', un posible código a continuación, el 3' y otro posible código a continuación. Tras detectar estos códigos, se queda con el código o los códigos de secuencia y los busca en la tabla sequences. Si encuentra alguno, obtiene el UniGene_cluster y lo busca en la tabla unigene. Si en esta tabla obtenemos el GeneID, este lo buscaremos en la tabla gene_info para encontrar el símbolo y la descripción del gen. Por otra parte, aunque el 5' y el 3' tienen un significado biológico, a efectos del programa no tiene influencia que código aparece en uno y cual en otro, ya que primero realizará la búsqueda en la base de datos de un código y si no lo encuentra buscará el otro.

Una vez entendido el funcionamiento del robot que actualiza los nombres de gen, tuve claro que la información que tenía que obtener de los genes eran los códigos Unigene, los de secuencia, los simbolos del gen y su descripción. Observando los campos que aparecen en las GDS Full, observo que los que pueden ser de interés son los siguientes:

- Gene title: contiene la descripción del gen.
- Gene symbol: contiene el símbolo del gen.
- Unigene ID: contiene el código unigene del gen.
- GenBank Accession: contiene el código de secuencia del gen.

Es importante saber que no para todos los genes aparece esta información. Para cada gen, el criterio para saber qué datos coger es el siguiente:

Si no aparece ni Gene symbol ni GenBankAccesssion ni Unigene
No se coge el gen
Sino
Si aparece el GenBankAccession
Si aparece el Gene title y el Gene symbol
Guardamos el Gene symbol, el Gene Title y el GenBankAccession
Sino
Guardamos el GenBankAccession
Fin si
Sino
Si aparece el Unigene ID
Guardamos el Unigene ID
Sino
Guardamos el Gene symbol y el Gene Title
Fin si
Fin si

Fin si

De esta manera, siempre que tengamos en un gen el código de secuencia o Unigene se podrá ir actualizando con el robot. En caso de no tener estos códigos se guardan el símbolo y la descripción si se tiene. En caso de no tener información para poder identificar el gen, no se cogerá ese gen. Existe otro caso en el que no se coge los genes de la microarray. En algunas de las microarrays puede haber unos genes denominados de control. Estos genes tienen unos valores de expresión que no están normalizados, por la cual cosa no serán útiles para los análisis. Para detectar estos genes se mira el identificador de los genes en esa microarray. Estos siempre empiezan por “AFFX”, por lo tanto, cuando se detecta que un identificador empieza de esta manera no se coge el gen.

Cada vez que guardamos un gen, en la misma línea guardamos también el identificador del gen en esa microarray, el cual podemos encontrar también en el archivo GDS Full. Como estos identificadores aparecen también en el fichero GDS Clustering, este identificador servirá para ordenar los genes según el fichero GDS Clustering.

3.2.3.4 Generar los ficheros con los valores de expresión(samples), con los nombres de condiciones experimentales(snames) y con los nombres de genes(genesorig)

Cuando ya hemos parseado el GDS Full, ya podremos generar los ficheros con los valores de expresión(samples), con los nombres de las condiciones experimentales(snames) y con los nombres de genes(genesorig). Para ello utilizaremos los siguintes ficheros generados en la fase anterior:

- Fichero con los identificadores de los genes en la microarray y la información de los genes sin el orden de los clusters del NCBI.
- Fichero con las condiciones experimentales de la microarray sin el orden de los clusters del NCBI

Además también se utilizará el GDS Clustering que hemos descargado del NCBI.

El primer paso será crear el fichero con los nombres de las condiciones experimentales(snames). En la primera línea del fichero GDS Clustering encontramos todos los identificadores de las condiciones experimentales según el orden de los análisis de clustering realizados. Lo que haremos es leer uno a uno estos identificadores. Cada

identificador, lo buscaremos en el fichero con las condiciones experimentales de la microarray sin el orden de los clusters del NCBI que hemos creado a partir del GDS Full. De esta manera, podremos encontrar por cada identificador su descripción, y podremos escribir en un fichero todas las condiciones según el orden del fichero GDS Clustering. Este fichero que escribimos será el fichero con los nombres de las condiciones experimentales(snames) de la microarray.

A continuación, en el fichero GDS Clustering tenemos todos los identificadores de los genes de la microarray y sus valores de expresión. Por cada línea haremos lo siguiente:

- Buscaremos el identificador del gen en el fichero con los identificadores de los genes en la microarray y la información de los genes sin el orden de los clusters del NCBI, dónde tenemos el identificador del gen en la microarray y la información de ese gen. Si encontramos el identificador, copiaremos la información del gen en el fichero que contendrá los nombres de genes(genesorig), según el orden del fichero GDS Clustering.
- Si hemos encontrado el identificador del gen, pasamos a guardar los valores de expresión, que se encuentran a continuación de cada gen del fichero GDS Clustering. Estos valores de expresión vienen separados por tabuladores que es como se guardan también en el fichero con los valores de expresión(samples). Por lo tanto, podremos guardarlos tal cual como nos los encontramos en el fichero GDS Clustering, a excepción de aquellos valores nulos. Los valores nulos en el archivo GDS Clustering son representados mediante la cadena “nan”, mientras que en los ficheros con los valores de expresión(samples) se representa con tres espacios vacios. Por lo tanto, habrá que comprobar si tenemos algún valor “nan”, y si es así, substituirlo por tres espacios.

Una vez llegamos al final del fichero GDS Clustering ya habremos creado los ficheros con los valores de expresión(samples), con los nombres de genes(genesorig) y con los nombres de las condiciones experimentales(snames) con el orden de los análisis de clusters realizados en el NCBI.

3.2.3.5 Generar los ficheros con los nombres de genes actualizados(genes)

Cuando ya hemos terminado de parsear los ficheros GDS Full y GDS Clustering, lo que haremos será crear el fichero con los nombres de los genes actualizados(genes) a partir del fichero con los nombres de los genes(genesorig). Lo que hice fue utilizar una parte del robot que actualiza los nombres de los genes, concretamente la parte en que se genera el fichero genes a partir del fichero genesorig.

Como he explicado anteriormente, el robot actualizador de nombres de gen mira si en el fichero con los nombres de los genes(genesorig) tenemos códigos Unigene o códigos de secuencia mediante expresiones regulares. Sin embargo, el programa estaba realizado de una manera que no me interesaba. El primer paso que se hacía, era comprobar si en los 10 primeros genes encontraba algún código de secuencia o algún código Unigene. Si encontraba un código de secuencia, consideraba que en el fichero con los nombres de los genes(genesorig) solo habría códigos de secuencia, entonces hacía el recorrido del fichero buscando tan solo códigos de secuencia. Si encontraba un código Unigene, consideraba que en el fichero solo habría códigos Unigene, entonces hacía el recorrido del fichero buscando tan solo códigos Unigene. En caso de no encontrar ningún código en los 10 primeros, dejaba el fichero con los nombres de los genes actualizados(genes) igual que el fichero con los nombres de los genes(genesorig). Por lo tanto, de esta manera, consideraba que en un fichero con los nombres de los genes(genesorig) solo habría códigos de un tipo. Esto no interesaba, ya que en una GDS Full del NCBI nos podríamos encontrar microarrays en las que hubiese tanto códigos de secuencia como Unigene. Por lo tanto cambié esta parte del programa para que buscase en cada línea del fichero con los nombres de los genes(genesorig) si tenemos un código de secuencia o tenemos un código Unigene, de esta manera, podremos tener códigos de distintos tipos en un mismo fichero con los nombres de los genes(genesorig).

Esta parte cambiada la modifiqué para que el robot que actualiza los nombres de los genes funcionase de esta manera y a la vez la utilicé para generar el fichero con los nombres de genes actualizados(genes) en el momento de parsear los ficheros.

3.2.4 Subir las microarrays a la base de datos local

Una vez tenemos los ficheros generados y toda la información de la microarray, pasaremos a introducir los registros de la microarray en la base de datos local de microarrays y a mover los ficheros de la microarray a los directorios correspondientes.

En la figura 3.11 podemos ver un diagrama de flujo del proceso de subida de las microarrays a la base de datos local. Como se puede observar, principalmente se realizan operaciones sobre la base de datos de microarrays. En primer lugar, se inserta el registro de la microarray en la tabla de microarrays. A continuación, se insertan los registros de todos los genes de la microarray en la tabla de genes. Posteriormente, se realiza un update sobre la tabla de los usuarios de la aplicación(tabla applicants), concretamente sobre el usuario que sube la microarray, para indicar que tiene acceso a una microarray más. El siguiente paso es crear una tabla para los experimentos(tabla experiments) para la microarray que se está subiendo. Finalmente, se mueven los ficheros al directorio de la microarray correspondiente. Este proceso se realiza para todas las microarrays

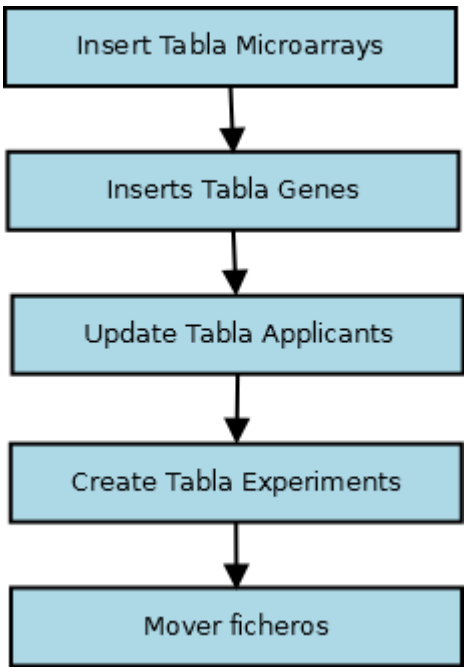

Figura 3.11Diagrama de flujo de la parte subir microarrays a la base de datos Inicialmente, se inserta el registro de la microarray en la tabla de microarrays. A continuación, se insertan todos los genes de la microarray en la tabla de genes. Posteriormente, se realiza un Update en la tabla applicants para indicar que el usuario que sube la microarray tiene acceso a una microarray más. El siguiente paso es crear una tabla

experiments para la microarray que se está subiendo. Finalmente movemos todos los ficheros de la microarray a su directorio correspondiente.

Uno de los asuntos previamente a tratar era que usuario se utilizaría para subir las microarrays. Este usuario, tendría que ser uno que no perteneciese a ningún usuario real para que no interfiriese en la aplicación web. Los usuarios son almacenados en la tabla applicants. El campo que identifica únicamente a un usuario es el campo app_PK, que es un valor numérico y es la clave primaria de la tabla. En principio pensé en utilizar un usuario con el valor app_PK igual a cero el cual aun no había sido creado. Posteriormente, se pensé en utilizar el usuario con el valor app_PK igual a uno que aunque estaba creado, este no pertenecía a un usuario real si no a uno de pruebas que ya no era de uso. Al final, opté por la opción de usar el usuario con el valor app_PK uno, ya que si se migrase la aplicación a otro servidor con otra base de datos podría dar problemas al poner una clave primaria a cero. De esta manera, augmenta la portabilidad entre servidores.

Al insertar la información en la tabla de microarrays se inserta la siguiente información:

- El identificador de la microarray en la base de datos en el campo micro_pk. Este identificador será el número siguiente al de la última microarray que se ha insertado.
- El usuario que ha subido la microarray en el campo micro_applicant_fk. Como hemos comentado, este será el usuario 1.
- La fecha de creación de la microarray en los campos micro_datecreation y micro_datemodification. Esta fecha la obtenemos del fichero GDS Full en la etapa parsear ficheros.
- El título de la microarray en el campo micro_name y la descripción en el campo micro_description. Esta información la obtenemos del fichero GDS Full en la etapa parsear ficheros.
- Los usuarios que tienen acceso a la microarray en el campo micro_users. Inicialmente, solo será el usuario 1.
- El nombre del fichero con los valores de expresión(samples) en el campo micro_insamples, el nombre del fichero con los nombres de genes actualizados(genes) en el campo micro_ingenes y el nombre del fichero de nombres con los nombres de las condiciones experimentales(snames) en el campo micro_innames. Estos nombres serán el identificador de la microarray seguido de un punto y el tipo de fichero que es.
- El código de la especie en el campo micro_tax_id. Este código se ha obtenido en la etapa parsear ficheros.

- El identificador de la microarray en el campo micro_gds_id.
- El número de condiciones experimentales en la microarray. Para esto se ha creado un nuevo campo llamado micro_number_samples. Esto permitirá mostrar el número de condiciones experimentales cuando listemos las microarrays en la interfaz web.

El título, la descripción, la fecha y la especie se leen del fichero con la información descriptiva de la microarray que se crea durante la etapa de parsear ficheros.

Tras insertar los datos en la tabla de microarrays, se recorren todos los genes del fichero con los nombres de genes actualizados(genes) y se van insertando en la tabla de genes. La información que se introduce en la tabla es la siguiente:

- El identificador del gen en la tabla en el campo gen_pk. Este identificador se obtiene sumándole a uno al identificador del último gen introducido.
- El usuario que ha introducido el gen en el campo gen_applicant_fk. En este caso, es el mismo de la microarray, es decir, el usuario 1.
- El identificador de la microarray a la que pertenece el gen en el campo gen_matrix_pk.
- El nombre del gen en el campo gen_name_matrix. Como nombre se introduce toda la información que aparece en el fichero con los nombres de genes actualizados(genes).
- La posición númerica del gen en la microarray en el campo gen_pos.
- Los usuarios que pueden acceder al gen, en el campos gen_users. En este caso, será el usuario 1.

Inicialmente, contemplé la posibilidad de realizar cambios en la tabla de genes. Estos cambios incluían compartir los mismos registros para los genes que apareciesen en varias microarrays, guardando las microarrays y las posiciones en que se encuentran en cada uno de ellos. Finalmente, al ver que el número de genes no crecía excesivamente, opté por dejar la tabla tal como estaba.

En el siguiente paso, se actualiza la tabla applicants. Concretamente, lo que se hace es sumar uno al campo app_performedmciroarrays. Este campo indica el número de microarrays que tiene acceso el usuario.

Es necesario también crear una tabla Experiments para la nueva microarray, debido que en estas tablas se guardarán los experimentos realizados por los usuarios para cada microarray.

Finalmente se mueven los ficheros de las microarrays, es decir, los ficheros con los nombres de los genes actualizados(genes), con los nombres de genes(genesorig), con los nombres de las condiciones experimentales(snames) y con los valores de expresión(samples) a su directorio correspondiente de la microarray. Este directorio es un directorio donde están las mismas microarrays que suben los usuarios. Se ha optado por poner las microarrays en este mismo directorio ya que la aplicación web y las herramientas de análisis acceden a este directorio para encontrar las microarrays. De esta manera, no se han de cambiar para que busquen las microarrays en dos carpetas diferentes.

3.2.5 Control de errores y recuperación tras caída del servidor en el proceso de actualización

Debido a que durante la actualización se podría producir una caída del servidor y producir que la base de datos quedase en un estado incorrecto, es necesario realizar un proceso que compruebe y arregle si es necesario el estado de la base de datos local tras una caída del servidor.

Este proceso de comprobar y arreglar el estado de la base de datos local se realizará en el momento de arrancar el servidor, de esta manera se hará tras haberse producido una caída del servidor. Este proceso sólo comprobará y arreglará cualquier problema en la base de datos, pero no seguirá con la ejecución del robot, si no toda la parte que no se haya actualizado se hará en la siguiente actualización programada. Al inicio, pensé también en hacer la comprobación cada vez que se ejecutaba la actualización, pero de esta manera durante el transcurso entre la caída del servidor y la próxima actualización los usuarios podrían a llegar a acceder a microarrays que no están correctamente subidas, por lo tanto, me decanté por realizar la comprobación tras arrancar el servidor.

El primer paso del programa de comprobación de errores será detectar si la base de datos está en estado correcto. Para eso, se hará una consulta a la base de datos para saber cuál es la última microarray que contiene el campo el campo micro_gds_id diferente de nulo. Es decir, consultamos cual es la última microarray que ha subido a la base de datos el

proceso de actualización. Una vez tenemos el identificador de esta microarray, se comprueba si en el directorio de la microarray están sus 4 ficheros, es decir, los ficheros con los nombres de genes actualizados (genes), los ficheros con los nombres de genes(genersorig), con los valores de expresión(samples) y con los nombres de condiciones experimentales(snames).

En caso de que estén todos, es que la última microarray se subió correctamente, por lo tanto no deberemos arreglar nada. Las microarrays que no se han subido en esa actualización se subieran en la siguiente ya que como se ha comentado en la sección 3.2.1 en la etapa de obtener los identificadores de las microarrays obtendremos los identificadores que no tenemos en la base de datos y cumplan las condiciones necesarias.

En caso de que falte algún fichero, quiere decir que la última microarray no se subió correctamente, ya que nos habremos quedado en algún paso intermedio entre el paso de insertar la microarray en la tabla de microarrays y mover los ficheros al directorio correspondiente. En este caso, el programa realizará lo siguiente:

- Eliminará, si se ha creado, el directorio de la microarray y todos sus ficheros.
- Eliminará, si se ha creado, la tabla de experiments de esa microarray.
- Eliminará todos los registros de genes que se encuentren para esa microarray.
- Eliminará el registro de esa microarray en la tabla de microarrays.

3.2.6 Eliminación de los directorios y ficheros temporales que no son necesarios tras finalizar el proceso de actualización

Durante cada actualización de la base de datos se generan ficheros y directorios temporales. Para no ocupar memoria del servidor de manera innecesaria, estos ficheros y directorios se eliminarán tras haber realizado toda la actualización.

3.3 Sincronización con el robot de genes marcadores y programación periódica de ambos robots

Tras haber creado el robot de actualización de microarrays públicas de gran tamaño, el objetivo es que este robot y el robot de actualización de genes marcadores que hay en el servidor se ejecuten periódicamente de manera sincronizada, es decir, que se ejecuten secuencialmente y de manera que sus sistemas de directorios no se solapen.

3.3.1 Comprensión del funcionamiento del robot de genes marcadores y sincronización con el robot de actualización de microarrays

Para sincronizar el robot de descarga de genes marcadores con el de descarga de microarrays públicas de gran tamaño tuve que entender el funcionamiento del robot de descarga de genes marcadores, así como comprobar que el robot se ejecutaba correctamente y generaba todos los ficheros bien. En esta etapa de comprobación me encontré un problema.

El robot de descarga de genes marcadores accede a la de datos GEO Profiles[13] del NCBI para obtener los genes marcadores, y a la base de datos GEO Datasets para descargar las microarrays y obtener los valores de expresión de los genes marcadores. Estos valores de expresión se utilizan para generar imágenes de los genes marcadores. El problema encontrado es que hay microarrays que son eliminadas de la base de datos GEO Datasets[9], sin embargo, los genes marcadores no son eliminados de la base de datos GEO Profiles[10]. Esto produce que en casos como este, se generen los ficheros de los genes marcadores, pero que posteriormente, al intentar generar las imágenes, al bajarse por el FTP la microarray esta no se encuentre, provocando que el programa se aborte.

Para solucionar el problema se modificó el robot, haciendo que si no se encuentra la microarray en la base de datos GEO Datsets salte el paso de generar las imágenes de ese gen marcador siga con el siguiente sin abortar la ejecución.

A la hora de realizar la sincronización, el robot encargado de actualizar las microarrays se ejecuta en un directorio interior al robot de genes marcadores, por lo tanto, no existirán problemas a la hora de evitar el solapamiento de directorios.

3.3.2 Programación periódica de la ejecución de los robots

La programación periódica de ambos robots es necesaria para mantener las últimas microarrays y genes marcadores que son subidos en los servidores del NCBI.

9 http://www.ncbi.nlm.nih.gov/gds?term=gds1971
10 http://www.ncbi.nlm.nih.gov/geoprofiles?term=gds1971

Para programar la ejecución periódica de los robots se ha utilizado el Cron de Linux. El Cron es en programa que permite a usuarios de Linux/Unix ejecutar automáticamente comandos o scripts a una hora o fecha específica y de manera periódica si se desea.

Por lo tanto, se utilicé el Cron para programar las ejecuciones de ambos robots cada tres meses, ejecutándose el día uno del mes a las 00:00. 2 meses serian suficientes para ejecutar ambos robots pero se ha extendido a 3 para que se ejecute conjunto un robot de actualización de las bases de datos.

El Cron también permite programar ejecuciones en el momento de arranque del servidor. Esta opción la utilicé a la hora de programar las comprobaciones de si ha habido una caída del servidor en el momento de las actualizaciones. De esta forma, al iniciarse el servidor se ejecutará primero la comprobación de errores y recuperación del robot de descarga de microarrays públicas de gran tamaño. Si ha habido algún error mientras se ejecutaba este, se solucionará dejando la base de datos en buen estado, mientras que si no ha habido ningún problema realizará la comprobación sin modificar nada. Una vez ejecutada la comprobación del robot de descarga de microarrays se realiza la comprobación de errores del robot de genes marcadores. En este caso se comprueba si la actualización ha acabado correctamente y si no es así limpia los restos y vuelve a ejecutar toda la actualización. Esto se puede realizar de esta manera ya que el robot de genes marcadores, al contrario que el de descarga de microarrays públicas de gran tamaño, no afecta a la base de datos.

Tras la primera ejecución de los robots el tiempo de ejecución de cada uno es el siguiente:

- El robot de descarga de microarrays públicas de gran tamaño aproximadamente está 6 horas en ejecución para descargar 89 microarrays.
- El robot de descarga de genes marcadores está aproximadamente 10 horas en ejecución.

3.4 Aplicación web

Una vez creada y actualizada la base de datos de microarrays pasé a crear y adaptar la aplicación web para poder gestionar las nuevas microarrays. Esta etapa la dividí en dos:

- Crear nueva interfaz para la búsqueda y acceso de microarrays públicas de gran tamaño.

- Realizar todos los cambios necesarios para que la aplicación web actual funcione con las microarrays públicas de gran tamaño de la misma manera que funcionan actualmente con las microarrays subidas por los usuarios.

En los siguientes puntos veremos cómo he realizado estas dos etapas.

3.4.1 Crear nueva interfaz para la búsqueda de microarrays

Al tener disponibles una gran cantidad de microarrays, se hace necesario añadir una interfaz para facilitar la búsqueda y el acceso a ellas.

Actualmente los usuarios tienen disponible una lista donde aparecen las microarrays subidas por los usuarios a las que tienen acceso. La idea, es realizar otra lista donde aparezcan las microarrays públicas de gran tamaño del NCBI que el usuario ha decidido añadir en su lista, es decir, una lista de microarrays públicas favoritas del usuario. Para ellos, se les proporcionará un sistema de búsqueda para que puedan encontrar las microarrays que desean investigar.

Tras realizar la implementación se puede ver las listas de microarrays que aparecen en la figura 3.12. En la parte superior podemos ver una lista donde aparece una microarray. Esta lista es la que actualmente ya tenía la aplicación y aparecen las microarrays subidas por los usuarios a las que tiene acceso el usuario. A continuación, podemos ver una lista de dos microarrays. Esta es la lista de microarrays públicas favoritas que he añadido para poder acceder a las microarrays públicas de gran tamaño que el robot de actualización de microarrays descarga. En la lista aparecen las microarrays que el usuario ha decidido añadirse para realizar análisis. Para poder buscar las microarrays en la parte inferior se dispone de la opción de búsqueda. Esta búsqueda se puede realizar introduciendo la especie de la microarray(taxonomy), introduciendo una palabra clave(topic), o ambas a la vez. Si no se introduce ninguna de las microarrays entonces se nos muestrearán todas las microarrays que ha descargado el robot.

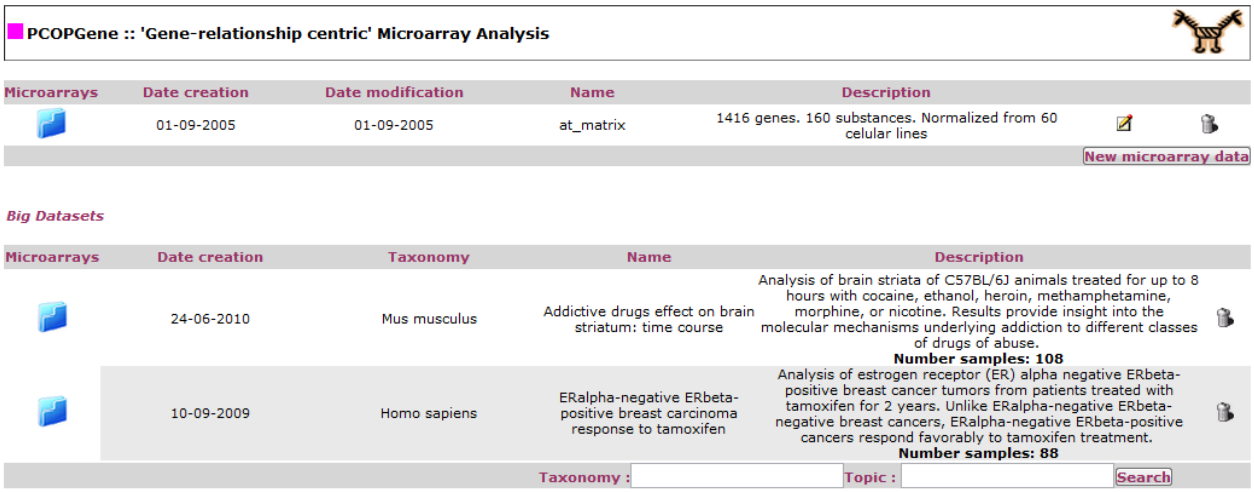

Figura 3.12 Listados de microarrays<<más ifno>>. Incluye la lista de microarrays subidas por los usuarios a las que el usuario tiene acceso y la lista de microarrays públicas favoritas.

En la parte superior aparece el listado de microarrays subidas por los usuarios. A continuación, aparece el listado de microarrays publicas favoritas del usuario, es decir, las descargadas por el robot y que el usuario ha decidido añadir en la lista. En la parte inferior tenemos la herramienta de búsqueda de microarrays.

Los campos que aparecen en la lista de microarrays públicas favoritas son los siguientes:

- Microarrays: Aparece un link que al clicar entramos dentro de la microarray para poder realizar análisis.
- Date creation: Aparece la fecha de creación de la microarray. Esta es la fecha de subida de la microarray en la base de datos del NCBI.
- Taxonomy: Aparece el nombre de la especie de la microarray.
- Name: Aparece el título de la microarray.
- Description: Aparece la descripción y el número de condiciones experimentales de la microarray.
- En la última posición disponemos de un icono que clicándola nos permite eliminar la microarray de la lista. Al hacer la eliminación lo único que hacemos es que el usuario no tenga acceso, pero no eliminamos la microarray de la base de datos, ni los experimentos que ha realizado el usuario con esa microarray. De esta forma, si el usuario desea volver a analizar la microarray conservará todos los análisis realizados.

Respecto a la lista de microarrays subidas por los usuarios no se ha añadido el campo Date Modification y el botón de editar la microarray. La razón es que al ser microarrays externas no tiene sentido editarlas y modificarlas ya que los datos vienen del NCBI.

Para mostrar el listado de microarrays públicas favoritas se realiza una consulta SQL para traer aquellas microarrays que tienen el campo micro_gds_id diferente a null y que el usuario está en el campo micro_users. Se ha protegido la aplicación para que no se pueda entrar con el usuario 1, ya que de esta manera tendría acceso a todas las microarrays, ya que es el que sube las microarrays.

Al pulsar el botón Search se nos muestra una pantalla como la que podemos ver en la figura 3.13. En este ejemplo, tenemos el resultado de la búsqueda tras introducir breast cáncer como palabra clave. Tenemos dos listados. El primero, titulado GDS found, se nos muestran todas las microarrays que se han encontrado con las condiciones de búsqueda introducidas y que el usuario no las tiene añadidas en su lista de microarrays favoritas. En esta lista el usuario puede añadir la microarray que le interesa a su lista pulsando sobre el botón que aparece en la parte derecha en forma de flecha de la microarray que desea. El segundo listado, titulado como GDS found in user list se nos muestran todas las microarrays que se han encontrado con las condiciones de búsqueda insertadas y que el usuario ya las tiene incluidas en su lista de microarrays públicas favoritas. En esta lista, como el usuario ya tiene las microarrays añadidas no se incluye la opción de añadir.

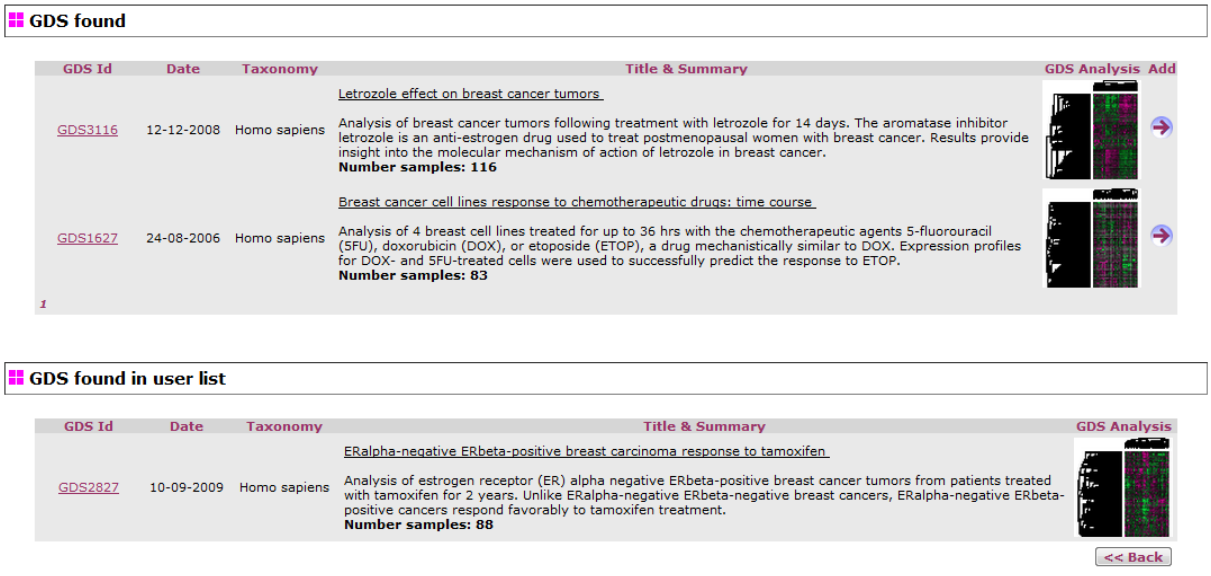

Figura 3.13 Resultado de búsqueda de microarrays con la palabra clave "breast cáncer".

Se puede observar dos listas. En la primera aparecen las microarrays que cumplen las condiciones de búsqueda y que el usuario aun no ha añadido en su lista de microarrays públicas favoritas. En la segunda lista aparecen las microarrays que cumplen las condiciones de búsqueda y que el usuario ya posee en su lista de microarrays públicas favoritas.

Los campos que aparecen en las listas de microarrays encontradas son los siguientes:

- GDS ID: Aparece el identificador de la microarray en el NCBI. Este identificador hace de link de la información de la microarray en el NCBI.
- Date: Aparece la fecha de subida de la microarray en el NCBI.
- Taxonomy: Aparece la especie de la microarray.
- Title & Summary: Aparece el título, la descripción y el número de condiciones experimentales de la microarray.
- GDS Analysis: Aparece una imagen donde podemos apreciar los clusters de la microarray en el análisis que se ha hecho en el NCBI. Clicando sobre la imagen nos lleva al sitio web donde podremos ver la imagen en mayor tamaño.
- Add: Solo aparece en la lista de microarrays que no posee el usuario en su lista de microarrays públicas favoritas. Aparece un botón que al clicarlo añadimos la microarray a la lista de microarrays públicas favoritas.

Para buscar las microarrays que cumplen las condiciones que ha insertado el usuario se realiza una consulta SQL utilizando los campos micro_tax_id, micro_name y micro_description de la tabla microarrays. Para buscar la especie, se busca previamente el código de la especie en la tabla taxonomy. Una vez se ha obtenido el código, se busca en el campo micro_tax_id de la tabla de microarrays. Para buscar la palabra clave introducida por el usuario se busca que aparezca en los campos micro_name o micro_description de la tabla microarrays. En la búsqueda por palabra clave se busca la palabra tal cual como la introduce el usuario. Es decir, si el usuario introduce “breast cancer”, se buscará en los campos micro_name y micro_description el texto “breast cancer”, en lugar de buscar “breast” y “cáncer” por separado. Esto se realiza de esta manera porque ofrece una mayor precisión a la hora de realizar una búsqueda. Además, no existe una cantidad excesiva de microarrays como para hacer búsquedas por más de una palabra separada, y en caso de que se deseara hacer, se podría introducir una palabra y posteriormente usar el buscador del navegador para encontrar lo que realmente se desea.

Cuando un usuario clica sobre el botón de añadir-se las microarrays esta será añadida a su lista de microarrays públicas favoritas. Internamente, se añade el código del usuario al campo micro_users de la microarray añadida.

Inicialmente, a la hora de tratar en cómo distinguir las microarrays que ya poseía el usuario en su lista de microarrays públicas favoritas pensé en realizar una sola lista y deshabilitar el botón de añadir las microarrays en aquellas que el usuario ya las posee en su lista de microarrays públicas favoritas. Posteriormente, pensé en la opción de añadir dos listas, una con las microarrays que no tiene el usuario en su lista de microarrays públicas favoritas y otra con las que ya posee. Mientras que en la primera opción, había que comprobar para cada microarray que cumplía las condiciones de búsqueda si el usuario estaba en el campo micro_users, en la segunda se puede realizar 2 consultas SQL, una recuperando aquellas microarrays que cumplen las condiciones y que el usuario ya posee, y otra recuperando aquellas microarray que cumplen las condiciones y el usuario no posee en su lista de microarrays públicas favoritas. Además, introduciendo dos listas se aprecia de manera rápida las microarrays que ya tiene el usuario en su lista de microarrays públicas favoritas y las que no. Por estas razones, decidí implementar la segunda opción.

No se ha incluido la opción de ordenar las microarrays encontradas por sus campos. La razón, es que las microarrays aparecen ordenadas según su fecha, las microarrays más nuevas aparecen en las primeras opciones y por otros campos no tenía mucho sentido hacer la ordenación, a excepción de realizar una agrupación por especie, pero esto ya se permite realizar mediante una búsqueda por especie.

Si se ha incluido la paginación de los resultados de la búsqueda, ya que en búsquedas dónde pueda aparecer un número considerable de microarrays podría ser necesario desplazar demasiado la página hacía abajo. Por esta razón, aparecen un máximo de 25 microarrays por página.

3.4.2 Modificaciones en la aplicación web para que funcione correctamente con las microarrays públicas de gran tamaño del NCBI

Para que la aplicación web funcione correctamente, realicé varios cambios para que se adaptase a las nuevas microarrays. Para ello, realicé todo tipo de pruebas y comprobando dónde podía haber problemas.

En primer lugar, observé que tras haber insertado las nuevas microarrays públicas de gran tamaño podrían ocurrir problemas en la lista de microarrays subidas por los usuarios que aparece en la figura 3.12. La razón es que para mostrar esta lista, simplemente se realizaba una búsqueda SQL de las microarrays a las que tiene acceso el usuario, por lo tanto, si un usuario se añadía una microarray públicas de gran tamaño también aparecería en esta lista de microarrays subidas por los usuarios a parte de su lista de microarrays públicas favoritas. Para ello, modifiqué la sentencia SQL de manera que considerase también que el campo micro_gds_id de las microarrays fuese null.

Un problema más complicado de arreglar ocurrió a la hora de probar de realizar experimentos con las relaciones entre las expresiones de genes de las microarrays. Esta opción, esta al entrar dentro de una microarray. Lo que se realiza, es mostrar un gráfico con la relación entre los genes de la microarray escogidos, como se ha visto en la figura 2.2. El problema que ocurría es que para algunas microarrays el experimento no se realizaba, mostrando la pantalla de la aplicación web en blanco sin mostrar ningún error. Tras hacer varias pruebas, observé que el problema solo ocurría para las microarrays que tenían mayor tamaño. Finalmente descubrí que el problema se producía que en una parte del código se leía todo el fichero con los valores de expresión(samples) de la microarray, lo que provocaba que se pasase el límite de memoria que permitía el PHP en el servidor. Observando el código comprobé que realmente no era necesario hacer la lectura de todo el fichero con los valores de expresión(samples), sino que simplemente era necesario leer los valores de expresión de los genes que se había escogido para realizar el experimento de relación entre genes. Tras realizar los cambios, comprobé que ahora si los experimentos se realizaban también para las microarrays de mayor tamaño correctamente.

A la hora de realizar las relaciones entre las expresiones de genes también se mostraba la opción para normalizar los datos. Al probar esta opción observé que esta no funcionaba. Esto es debido a que el programa que se encargaba de realizar la normalización en las relaciones entre genes no estaba realizado para hacerlo entre genes de una misma microarray. Por esta razón, eliminé la opción de normalizar tanto en las microarrays públicas de gran tamaño descargadas por el robot, como para las microarrays subidas por los usuarios, ya que en ninguna de las dos funcionaba.

También han sido necesarios realizar cambios en la subida manual de microarrays. Cuando un usuario sube una microarray manualmente, se comprobaba que el fichero con los valores de expresión(samples) no estubiese ya en el servidor, ya que si ya tenemos un

fichero con los valores de expresión(samples) igual quiere decir que ya tenemos la microarray en el servidor. Cuando ocurre esto, simplemente se le da acceso al usuario a la microarray y a sus genes y no se vuelve a subir, lo que facilita que se ahorre espacio en estos casos. El problema surgió que para comprobar los ficheros con los valores de expresión(samples) estos se cargaban en memoria con el código php, y al igual que pasaba con la creación de experimentos, cuando se cargaba un fichero con los valores de expresión(samples) muy grande, se excedía el límite de la memoria permitida por el php, provocando que se parara la ejecución y no se pudiesen subir microarrays. En este caso cambié la forma de comprobar los archivos. En vez de cargar el fichero con los valores de expresión(samples) en memoria desde el php, realicé llamadas a los comandos Linux cmp y stat para comprobar los archivos. De esta manera se solucionó este problema.

En la misma subida de microarrays manualmente hice otra modificación. Como añadí un campo para introducir el número de condiciones experimentales de la microarray decidí que en el momento de subir las microarrays manualmente se llenara este campo con el número de condiciones experimentales de la microaray en lugar de dejarlo a null.

4. Informe técnico

En este apartado se describirá de forma técnica los todos los programas implementados en las diferentes fases del proyecto y la estructura de directorios que se utiliza.

4.1 Estructura de directorios

La estructura de directorios que sigue el proyecto es la siguiente:

- Directorio /var/www/cgi-bin/gds/BigData: En este directorio se encuentra todos los programas utilizados por el robot de descarga de microarrays públicas de gran tamaño del NCBI. Internamente, también posee los siguientes directorios temporales que utiliza el robot:
    - tmp: Se guardan datos temporales, como los XML devueltos en las consultas E-Utils, o el listado de microarrays que se ha de descargar.
    - FTP: Se guardan los archivos descargados de las microarrays del NCBI.

- MicroarraysFiles: Se guardan los archivos parseados de las microarrays. Los archivos de genes, genesorig, samples y snames se dejan en este directorio antes de ser movidos al directorio /var/www/cgi-bin/pcop/microarray.

En este directorio también se encuentra el programa de descarga de la base de datos de especies del NCBI. Este programa utiliza un directorio interno llamado Taxonomy, donde se guardan los archivos temporales necesarios para descargar la base de datos de especies.

- Directorio /var/www/cgi-bin/gds: En este directorio se encuentra todos los programas utilizados por el robot de descarga de genes marcadores. Internamente, posee los siguientes directorios:
  - GDSs: Se guardan los archivos que contienen los datos de los genes marcadores, agrupados por las microarrays a las que pertenecen.
  - GIFS: Se guardan las imágenes generadas de los genes marcadores que posteriormente utiliza la aplicación web.
  - FTP: Se guardan los archivos de las microarrays descargados del NCBI que se utilizan para obtener los valores de expresión de los genes marcadores.

  También encontramos el de archivos de llamada a los robots.
- Directorio /var/www/html/applic/gexp: En este directorio se encuentra toda la aplicación web del servidor. En el directorio interior microarray se encuentran todos los programas que se utilizan en la aplicación web para el análisis de microarrays.
- Directorio /var/www/cgi-bin/pcop/microarray: En este directorio se encuentran todas las microarrays del servidor. Por cada microarray tenemos un subdirectorio donde se guardan sus archivos. Estos archivos de las microarrays son los de samples, genes, genesorig y snames, y además, otros archivos utilizados para experimentos de las microarrays.

4.2 Programas utilizados para crear el robot de descarga de microarrays públicas de gran tamaño del NCBI

Para crear el robot de descarga de microarrays del NCBI se han implementado varios programas en Perl. Cada uno de estos programas representa una fase de las que hemos podido ver en el apartado 3.2. El programa que lanza el robot se llama RobotActualizacionMicroarrays.pl. Este programa hace las siguientes llamadas:

- ObtenerIDMicroarrays.pl: Este programa se encarga de realizar la consulta E-Utils para obtener los identificadores únicos del NCBI de las microarrays que tengan 70

muestras o más. Se parsea el XML devuelto creando un fichero donde aparecen los identificadores obtenidos. Posteriormente se hace la consulta a la base de datos local, concretamente a la tabla de microarrays para saber que identificadores son los que no tenemos y debemos descargar. Se crea un fichero llamado microarrays.txt con estos últimos identificadores y se guarda en el directorio tmp. El fichero "microarrays.txt" contiene un identificador por línea del fichero.

- DescargarMicroarraysFTP.pl: Este programa lee el fichero con los identificadores de las microarrays microarrays.txt, y por cada identificador descarga del NCBI los ficheros GDS Full y GDS Clustering. Los ficheros descargados los guarda ene l directorio FTP.
- GenerarFicherosMicroarrays.pl: Este programa es el encargado de realizar todo el proceso mostrado en la figura 3.9. Por cada microarray leída del fichero microarrays.txt obtiene toda la información general de la microarray, los nombres necesarios de los genes, las condiciones muéstrales, y los valores de expresión. De esta manera, inicialmente y considerando que IDMicroarray es el identificador de la microarray en el NCBI se crean los siguientes ficheros en el directorio MicroarraysFiles:
    - El fichero IDMicroarray.genesorig, donde cada línea contiene el identificador del gen en la microarray y la información del gen, separados por una tabulación y ordenados según el fichero GDS Full.
    - El fichero IDMicroarray.snames, que contiene la siguiente estructura el identificador de la muestra en la microarray y la descripción de la muestra, separados por dos puntos y ordenados según el fichero GDS Full.
    - El fichero llamado DatosMicroarrayTemp.txt que contiene el título, la descripción, el código de la especie, el identificador de la microarray en el NCBI, la fecha de publicación y el número de condiciones experimentales de la microarray. Para obtener la especie se consulta la tabla taxonomy y si no se encuentra se realiza una consulta por E-Utils.

  Una vez se obtienen estos ficheros, usándolos conjuntamente con el GDS Clustering se obtienen los siguientes ficheros en el directorio MicroarraysFiles:
    - El fichero IDMicroarray.Samplesordenados, que contiene los valores de expresión de los genes según el orden del fichero GDS Clustering.
    - El fichero IDMicroarray.Genesordenados, que contienen la información de los genes según el orden del fichero GDS Clustering.
    - El fichero IDMicroarray.Snamesordenados, que contiene los nombres de las muestras según el orden del fichero GDS Clustering.

Una vez se obtienen estos ficheros, se hacen los siguientes cambios de nombres en los ficheros en el directorio MicroarraysFiles:

- El fichero IDMicroarray.Samplesordenados pasa a ser el fichero IDMicroarray.samples
- El fichero IDMicroarray.Genesordenados pasa a ser el fichero IDMicroarray.genesorig.
- El fichero IDMicroarray.Snamesordenados pasa a ser el fichero IDMicroarray.snames.

Por lo tanto, en este punto ya tenemos los ficheros gensorig, snames y samples según el orden del fichero GDS Clustering. Para generar el fichero genes a partir del fichero genesorig el programa GenerarFicherosMicroArrays.pl hace llama al siguiente programa:

- CrearFicheroGenes2.pl: Este programa genera el fichero genes, pasándole como argumento un fichero genesorig en el directorio MicroarrayFiles. De esta manera en el fichero genes aparecerán los nombres de los genes actualizados. Para actualizar los nombres, se acceden a las tablas sequence, unigene y gene_info de la base de datos local.

Finalmente, por cada microarray se realiza la llamada:

- SubirMicroarraysBD.pl: Este programa recibe como parámetro los nombres de los ficheros samples, genes y snames y sube a la base de datos local la microarray introduciendo los datos de la microarray en la tabla microarrays, los datos de los genes en la tabla genes, y suma uno al campo app_performedmicroarrays de la tabla applicants. Los datos que describen la microarray los lee del archivo DatosMicroarrayTemp.txt. También renombra con el identificador que tendrá la microarray en el servidor local y se encarga de mover los ficheros de la microarray al direcotrio /var/www/cgi-bin/pcop/microarray.

- LimpiezaArchivos.pl: Este programa se encarga de eliminar todos los ficheros y directorios creados en la actualización que ya no se volverán a utilizar. Elimina los directorios FTP, MicroarraysFiles y tmp y todos los ficheros que se encuentran dentro.

A parte de estos programas, el robot de descarga de microarrays también tiene un programa Perl, llamado “ComprobacionCrashRobotBigData.pl” que se ejecuta al ponerse en marcha el servidor. Este programa comprueba que no se haya dejado algún error en una actualización por culpa de una caída del servidor. En caso de que detecte algún error,

soluciona los problemas tal como se ha explicado en la sección 3.2.5. Para detectar si ha habido algún error, recupera el identificador de la última microarray subida, y comprueba si se han subido todos sus ficheros al directorio /var/www/cgi-bin/pcop/microarray.

Por otra parte tenemos el programa “RobotTaxonomy.pl” que es el encargado de crear la tabla taxonomy de la base de datos local y descargar todos los datos actuales del NCBI respecto a las especies. Para descargar todas los nombres y códigos de las especies, realiza la descarga por FTP del fichero taxdmp.zip que se encuentra en el directorio Taxonomy del FTP del NCBI. El fichero se descarga en el directorio temporal Taxonomy y se descomprime. Una vez se descomprime se utiliza el fichero names.dmp. Este fichero en cada línea tiene una especie y por cada especie aparece diversa información por el carácter ‘|’. La información que se obtiene es el código de la especie y el nombre científico de la especie(scientific name). Estos se insertan en la tabla taxonomy de la base de datos local y no vez recorrido todo el fichero names.dmp se elimina. Si se desease mantener actualizada totalmente la tabla taxonomy con las nuevas especies del NCBI, se podría programar la ejecución del programa RobotTaxonomy.pl con el Cron, ya que cada vez que se ejecute dejará en la base de datos local todas las especies que en ese momento existen en el NCBI.

En los programas Perl, para utilizar algunas funciones es necesario añadir ciertos módulos. Uno de los módulos que debí utilizar, el modulo Date::Format, me encontré que no estaba instalado en el servidor. Para instalarlo, utilicé la consola de CPAN que permite instalar módulos fácilmente. Para ello, primero se lanza como root el siguiente comando para entrar en la consola de CPAN:

Perl –MCPAN –e Shell

Una vez dentro de la consola de CPAN, se instala el módulo requerido de la siguiente manera:

cpan> install Date::Format

Una vez instalado, el módulo ya se podrá usar en los programas Perl.

4.3 Modificaciones realizadas en la base de datos local

Durante el proyecto ha sido necesario crear una nueva tabla, añadir nuevos campos en la tabla de microarrays y realizar alguna modificación en la base de datos local.

En la sección fases, se ha comentado que el usuario que subiría las microarrays durante la ejecución del robot de descarga de microarrays públicas de gran tamaño seria el usuario con el campo app_PK igual a 1. Inicialmente, ya existía un usuario con el campo app_PK igual a 1, pero este era de pruebas. Antes de la primera ejecución del robot lo que hice poner los campos app_performedexperiments y app_performedmicroarrays a 0. Además comprobé que en las tablas experiments, genes, microarrays y view no apareciese el usuario 1 en ningún campo.

La única tabla añadida ha sido la tabla taxonomy, que contiene los códigos y los nombres de las especies. Esta tabla tiene los siguientes campos:

Tabla taxonomy				
Campo	Tipo	Nulo	Predeterminado	Descripción
tax_id	varchar(10)	No		Código de la especie
tax_ScientificName	varchar(255)	Sí	Null	Nombre científico de la especie

Por otra parte, los siguientes dos campos a la tabla de microarrays:

Tabla microarrays				
Campo	Tipo	Nulo	Predeterminado	Descripción
micro_gds_id	int(12)	Sí	Null	Identificador de la microarray en el NCBI
micro_number_samples	int(12)	Sí	Null	Número de condiciones experimentales de la microarray

4.4 Programas utilizados por el robot de descarga de genes marcadores y la sincronización y programación periódica con el robot de descarga de microarrays públicas de gran tamaño

El robot de descarga de genes marcadores se ejecuta mediante la llamada al programa Perl principal.pl situado en el directorio /var/www/cgi-bin/gds. El robot de genes marcadores guarda los datos de los genes marcadores en el directorio GDSs. En este

directorio guarda tanto ficheros XML con los genes marcadores de las microarrays como ficheros XML con información de las microarrays.

La estructura de los ficheros XML que contienen los genes marcadores de una microarray es la siguiente:

```
<gds>
    <gen>
        <geneid></geneid>
        <uid></uid>
        <geneName></geneName>
        <alias></alias>
        <geneDesc></geneDesc>
        <idref></idref>
    </gen>
    <gen>
        <geneid></geneid>
        <uid></uid>
        <geneName></geneName>
        <alias></alias>
        <geneDesc></geneDesc>
        <idref></idref>
    </gen>
    …
</gds>
```

Todos estos genes marcadores pertenecen a la misma microarray. El identificador de la microarray aparece en el nombre del fichero y es el identificador de la microarray en el NCBI. Por otra parte, la estructura de los ficheros XML que contienen la información referente a las microarrays es:

```
<gds>
    <taxid></taxid>
    <titlee></titlee>
    <summary></summary>
    <taxon></taxon>
    <count></count>
</gds>
```

El robot de genes marcadores también genera imágenes del os perfiles de expresión de los genes marcadores. Estas imágenes son almacenadas en el directorio GIFS.

Finalmente también se guardan los archivos de las microarrays descargados del NCBI en el directorio FTP.

En el directorio /var/www/cgi-bin/gds también podemos encontrar el programa bash CronRobots.bsh. Este programa realiza la llamada de los dos robots. Primero, hace la llamada al robot de descarga de microarrays públicas de gran tamaño. Una vez ha finaliza la ejecución de este robot, realiza la llamada al robot de descarga de genes marcadores.

La llamada del fichero CronRobots.bsh se realiza cada dos meses mediante el Cron. Para programar la ejecución con el Cron se ha modificado el fichero crontab del directorio etc/. Este fichero contiene las ejecuciones que se programan para que se ejecuten en cierto momento. Para añadir la programación de ejecución, se ha añadido la siguiente línea al final del fichero:

0 0 1 */2* * root /bin/bash /var/www/cgi-bin/gds/CronRobots.bsh

Los tres primeros números indican minuto, hora y día de ejecución respectivamente. La ejecución se programa para que se hagan los día 1 de mes a las 00:00. El cuarto parámetro indica el mes que se ha de ejecutar. Al escribir */2* indicamos que se ejecute cada dos meses. El quinto parámetro es el día de la semana en que se ha de producir la ejecución. En este caso, como realmente no importa en qué día de la semana se ejecute, se escribe un asterisco que indica que se puede ejecutar cualquier día. El sexto parámetro indica el usuario que realiza la ejecución, en este caso será el usuario root. Finalmente, el séptimo y último parámetro es la orden a ejecutar. En este caso, ejecutamos el archivo CronRobots.bsh, que realizará las llamadas de los robots. Por lo tanto, tras introducir esta línea en el fichero, se ejecutarán ambos robots cada dos meses.

Para controlar si no hay errores en las ejecuciones de los robots se ha creado el fichero LlamadasCrashRobots.bsh, que también se encuentra en el directorio /var/www/cgi-bin/gds. Este programa realiza la llamada al programa ComprobacionCrashRobotBigData.pl, que comprueba los errores durante la actualización de microarrays públicas de gran tamaño, y al programa act.arranke.pl que realiza la comprobación de errores del robot de descarga de genes marcadores. Este archivo también se ejecuta mediante el Cron, pero en este caso, al iniciarse el servidor. Esto se realiza añadiendo la siguiente línea al fichero crontab:

@reboot root /bin/bash /var/www/cgi-bin/gds/LlamadasCrashRobots.bsh

En este caso, estamos indicando que al iniciarse la máquina se ejecuta el archivo LlamadasCrashRobots.bsh.

4.5 Programas utilizados en la aplicación web

A la hora de crear las nuevas interfaces web y adaptar el actual aplicativo web para las nuevas microarrays se han creado y modificado distintos programas. Para estos programas se ha usado PHP[14] y HTML[15].

Para añadir la lista de microarrays públicas favoritas de gran tamaño y la búsqueda se han utilizado los siguientes programas del directorio /var/www/html/applic/gexp/microarray:

- listapplic.phtml: este programa mostraba el listado de microarrays a las que tiene el acceso el usuario. Se ha modificado para que se muestre otro listado más, con las microarrays favoritas de gran tamaño del usuario. Este programa tiene los siguientes parámetros:
    - applicantID: es el identificador del usuario que accede a la aplicación.
    - cad_micro: contiene los identificadores de las microarrays que se han de mostrar en la lista de microarrays que tiene acceso el usuario. Se calculan accediendo a la base de datos local en el programa redirection.php que se accede antes de entrar en listapplic.phtml.
    - f1, col, order: son parámetros utilizados para la ordenación del listado de microarrays a las que tiene acceso el usuario.
    - gestor: indica si se está utilizando la aplicación de gestión de microarrays o la aplicación de gestión de experimentos.
- searchGDS.php: este programa se ha creado para realizar y mostrar la búsqueda de las microarrays públicas de gran tamaño. Muestra dos listados, uno con las microarrays que el usuario no tiene en su lista de microarrays públicas favoritas y otro con las que ya posee. Permite añadir las microarrays que no posee en el listado de microarrays públicas favoritas. Los parámetros del programa son:
    - applicantID: usuario que está usando la aplicación.
    - topic: es texto que ha introducido el usuario para la búsqueda. Este texto se busca en los campos micro_name y micro_description de la tabla de microarrays.
    - taxonomy: es el nombre de la especie que ha introducido el usuario. Este nombre se busca en la tabla taxonomy y se obtiene el código correspondiente. El código obtenido se busca en la tabla de microarrays en el campo micro_tax_id.

Tras crear estas interfaces, se han modificado los siguienes programas:

- redirection.php: en este programa se hace la consulta de las microarrays a las que tiene acceso el usuario para posteriormente pasarlas por parámetro al programa listapplic.phptml. Se ha modificado para que en la consulta no se recuperen microarrays descargadas del NCBI. Este programa tiene como parámetros:
    - applicantID: es el código del usuario que ha accedido a la aplicación.
    - gestor: indica si se está utilizando la aplicación de gestión de microarrays o la aplicación de gestión de experimentos.
- newmatrix.php: este programa se utiliza cuando los usuarios suben microarrays manualmente. Se ha adaptado para que funcione correctamente, evitando los problemas de exceso de memoria al comprobar con las microarrays existentes, y se ha añadido la inserción en la base de datos del número de muestras de la microarray subida. Este programa utiliza los ficheros samples, snames y genes que ha introducido el usuario para subir la microarray además del parámetro applicantID que permite saber que usuario está subiendo la microarray.
- experimentform.phtml y appl_experimentform.php: este programa es el utilizado para mostrar el formulario antes de realizar un experimento de relaciones entre genes de una microarray. Se ha modificado para no permitir la opción de normalizar. Este programa tiene los siguientes parámetros:
    - cad_gen: contiene los identificadores de los genes sobre los cuales se hacen los experimentos.
    - gestor: indica si se está utilizando la aplicación de gestión de microarrays o la aplicación de gestión de experimentos.
    - id_matrix: es el identificador de la microarray sobre la cual se hace el experimento.
    - name_matrix: es el nombre de la microarray sobre la cual se hace el experimento.
    - id_view: es el identificador de la vista sobre la cual se realiza el experimento.
    - name_view: es el nombre de la vista sobre la cual se realiza el experimento.
- newexperiment.php y appl_newexperiment.php: este programa se utilizada a la hora de realizar un experimento de relaciones entre genes de una microarray. Se ha modificado para que no se produzcan errores a la hora de realizar el experimento con microarrays muy grandes, ya que se producía un exceso de carga de memoria. Este fichero contiene los siguientes parámetros:
    - cad_gen: contiene los identificadores de los genes sobre los cuales se hacen los experimentos.

- gestor: indica si se está utilizando la aplicación de gestión de microarrays o la aplicación de gestión de experimentos.
- id_matrix: es el identificador de la microarray sobre la cual se hace el experimento.
- name_matrix: es el nombre de la microarray sobre la cual se hace el experimento.
- id_view: es el identificador de la vista sobre la cual se realiza el experimento.
- name_view: es el nombre de la vista sobre la cual se realiza el experimento.

Los últimos programas pertenecen directorio /var/www/html/applic/gexp/microarray, a excepción del programa redirection.php, que se encuentra en el directorio /var/www/html/applic/gexp/.

5. Conclusiones

Tras realizar todas las fases programadas puedo decir que todos los objetivos del trabajo han sido alcanzados satisfactoriamente. A continuación, expongo la situación actual de los objetivos propuestos al inicio del proyecto.

- Objetivos relacionados con la actualización periódica y automática de la base de datos local de microarrays:
  - Se obtienen microarrays de gran tamaño publicadas en la base de datos GEO Datasets del NCBI. Estas microarrays contienen un mínimo de 70 condiciones experimentales.
  - Se descargan y se parsean los ficheros de las microarrays del NCBI de manera que se adaptan al formato de las microarrays del servidor local, tanto los datos de los valores de expresión como los nombres de los genes.
  - Las nuevas microarrays se insertan en la base de datos una vez han sido descargadas y parseadas.
  - La actualización se realiza de manera que si los genes de las microarrays cambian de nombre, estos serán actualizados por el robot actualizador de nombres, de manera que los nombres de los genes estarán siempre actualizados.

- La actualización es robusta a posibles errores o a la caída del servidor. Si se cae el servidor, se lanza un proceso que comprueba si la base de datos ha quedado en mal estado. En ese caso se soluciona dejando la base de datos en correcto estado y se dejan las microarrays que han quedado pendientes para la siguiente actualización.
- La actualización se realiza de manera periódica y sincronizada con el robot de actualización de genes marcadores de microarrays. Cada dos meses, se produce la actualización. Primero se realiza la actualización de microarrays públicas de gran tamaño, y una vez finalizado, se produce la actualización de genes marcadores.

- Objetivos relacionados con la interfaz web para gestionar las nuevas microarrays:
  - Se ha adaptado la aplicación web actual. Las operaciones que anteriormente se realizaban con las microarrays subidas por los usuarios ahora se pueden realizar también con las nuevas microarrays descargadas del NCBI automáticamente. De esta manera, se pueden realizar análisis para un gran número de microarrays públicas de gran tamaño.
  - Se ha creado una interfaz web para poder realizar búsquedas de las microarrays públicas insertando el tema de la microarray y/o la especie para la que se realizaron los experimentos. De esta manera, el usuario puede buscar las microarrays según sus preferencias.
  - Se ha creado un listado dinámico con las microarrays públicas que el usuario considera de interés. De esta manera, el usuario puede acceder a ellas de manera rápida.
  - La interfaz web se ha realizado de manera que el usuario puede acceder a sus operaciones de manera cómoda y sencilla maximizando la operabilidad, la usabilidad, la comprensión, la atracción, y la facilidad cognoscitiva.

Por lo tanto, todos los objetivos iniciales han sido cumplidos y al final los investigadores dispondrán de una aplicación donde podrán realizar diferentes análisis con todas las microarrays públicas de gran tamaño. Estos análisis de microarrays de gran tamaño pueden ayudar a los investigadores a encontrar causas o tratamientos de patologías.

Para facilitar el uso a los investigadores, esta aplicación proporciona un aplicativo altamente usable, entendible y con un alta operativilidad. Además, la base de datos y la

aplicación web están diseñadas con un alto grado de portabilidad, de manera que podrían migrarse a otros servidores sin muchas dificultades.

5.1 Impresiones personales

Personalmente, puedo decir que he disfrutado durante la realización del proyecto y me siento satisfecho con el trabajo que he realizado. Además, este ha sido de gran utilidad para mi formación, permitiéndome ganar experiencia en la elaboración de proyectos y en otros campos.

Cuando tengo que pensar en la parte más complicada del proyecto me viene a la mente los primeros pasos cuando tuve que hacer el esfuerzo de entender los tipos de datos con los que trabajaría, ya que las microarrays, los genes, los diversos códigos que estos pueden tener, etc. han sido conceptos muy técnicos y muy nuevos para mí. A pesar de la dificultad, logré entender todos los conceptos y superar las distintas fases del proyecto, todo esto trabajando con datos reales, algo que lo ha convertido en una experiencia de enorme utilidad. Esta utilidad viene por el hecho de que hasta el momento en la universidad principalmente había trabajado con datos no reales adaptados a las practicas que he debido realizar en mis años de formación, mientras que en esta ocasión, he tenido que ser yo mismo el que ha adaptado los programas a los datos con los que tenía que trabajar. Además, esto ha supuesto el crear una aplicación preparada para un uso real.

Durante el proyecto, he realizado la planificación y la organización junto con el director del proyecto Mario Huerta. Entre ambos, hemos ido planificando las diferentes fases del proyecto hasta conseguir llegar al objetivo final.

En el proyecto he podido poner en práctica mis propias capacidades que he ido adquiriendo durante la formación universitaria. He podido utilizar todos mis conocimientos de base de datos aprendidos, tanto a nivel de diseño como a nivel de programación(SQL). He aprovechado mis conocimientos y habilidades a la hora de crear algoritmos y utilizar lenguajes de programación. También he tenido la oportunidad de ampliar mis conocimientos en programación web. Finalmente, me ha servido para adquirir nuevos conocimientos más allá de la informática, ya que me ha permitido entrar en el ámbito de la bioinformática y la biotecnología.

Creo que el trabajo realizado ha sido muy bueno y he ido superando los problemas que han surgido de manera adecuada. Los resultados de la aplicación final son muy buenos,

permitiendo de esta manera una herramienta fiable a los investigadores, por lo tanto, puedo decir que el trabajo realizado ha sido útil tanto para mí, como para el centro para el que he realizado el trabajo, el IBB, pero debería de ser útil también para la comunidad científica internacional, así como para la sociedad, y por último para la humanidad si logra conseguirse algún avance en la investigación médica gracias al trabajo realizado.

5.2 Trabajos futuros

Una vez finalizado el trabajo se abre la posibilidad de realizar nuevos proyectos para ampliar o mejorar la utilidad de la aplicación web.

El principal proyecto que se podría realizar en el futuro seria aprovechar el preproceso que se realiza en el servidor local con las microarrays subidas por los usuarios para realizar análisis más profundos de las microarrays públicas de gran tamaño. Este preproceso permite agrupar clusters en las microarrays, mostrar un grafo de las interacciones entre los genes de una microarray, mostrar el factor de correlación de los genes de la microarray con los demás genes de la mismo microarray, entre otros. De esta forma, los investigadores obtendrían mayor información útil de las microarrays y les permitiría tener más información para investigar enfermedades.

Este preproceso se debería lanzar para las microarrays públicas de gran tamaño, pero a la vez debería de permitir que se lanzase el preproceso para las microarrays que se suben manualmente. El problema es que el preproceso de una microarray muy grande puede tardar semanas o incluso hasta un mes, por lo tanto, se debería de gestionar la ejecución de este preproceso.

6. Referencias

[1] Instituto de Biotecnología y de Biomedicina (IBB) de la Universidad Autónoma de Barcelona. http://ibb.uab.es/ibb/

[2] http://revolutionresearch.uab.es: Web server for on line microarray analysis supported by the Institute of Biotechnology and Biomedicine of the Autonomous University of Barcelona (IBB-UAB).

[3] Delicado, P.(2001) Another look at principal curves and surfaces. Journal of Multivariate Analysis, 77, 84-116 .

[4] Delicado, P. and Huerta, M. (2003): 'Principal Curves of Oriented Points: Theoretical and computational improvements'. Computational Statistics 18, 293-315.

[5] Cedano J, Huerta M, Estrada I, Ballllosera F, Conchillo O, Delicado P, Querol E. (2007) A web server for automatic analysis and extraction of relevant biological knowledge. Comput Biol Med. 37:1672-1675.

[6] Huerta M, Cedano J, Querol E. (2008) Analysis of nonlinear relations between expression profiles by the principal curves of oriented-points approach. J Bioinform Comput Biol. 6:367-386.

[7] Cedano J, Huerta M, Querol E. (2008) NCR-PCOPGene: An Exploratory Tool for Analysis of Sample-Classes Effect on Gene-Expression Relationships Advances in Bioinformatics, vol. 2008

[8] Huerta M, Cedano J, Peña D, Rodriguez A, Querol E. (2009) PCOPGene-Net: holistic characterisation of cellular states from microarray data base on continuous and non-continuos analysis og gene-expression relationships. BMC Bioinformatics 2009 May 9;10:138.

[9] La página Web oficial del National Center for Biotechnology Information (NCBI) que ofrece de manera pública todas sus bases de datos. http://www.ncbi.nlm.nih.gov/

[10] Página web oficial de la base de datos Geo Datasets del NCBI. http://www.ncbi.nlm.nih.gov/gds

[11] NCBI GEO: archive for functional genomics data sets—10 years on

[12] http://www.perl.org : Home of the Perl programming language.

[13] http://www.ncbi.nlm.nih.gov/geoprofiles

[14] http://php.net : Sitio oficial de PHP con gran cantidad de recursos en ingles, noticias, descargas,
documentación, calendario de eventos relacionados.

[15] http://www.w3.org/MarkUp/ XHTML2 Working Group Home Page

Firmado: Daniel Sánchez Santolaya
Bellaterra, 17 de septiembre de 2012

Resumen

El IBB ha desarrollado un servidor de aplicaciones: http://revolutionresearch.uab.es para el análisis de microarrays. Estas microarrays las obtienen y las suben a la base de datos local los usuarios de la aplicación.

En la presente memoria se detalla el proceso realizado para automatizar la subida de microarrays públicas a la base de datos local. Estas microarrays se obtendrán del NCBI. El proceso de descarga de microarrays se realizará cada dos meses y estará sincronizado con un proceso de descarga de genes marcadores de microarrays del NCBI.

En la memoria también se describen las fases realizadas para crear la interfaz web para gestionar estas microarrays públicas y las modificaciones realizadas sobre el aplicativo web para permitir realizar análisis con estas microarrays.

Resum

El IBB ha desenvolupat un servidor d'aplicacions: http://revolutionresearch.uab.es per l'anàlisi de microarrays. Aquestes microarrays les obtenen y les pugen a la base de dades local els usuaris de l'aplicació.

En la present memòria es detalla el procés realitzat para automatitzar la pujada de microarrays públiques a la base de dades local. Aquestes microarrays s'obtindran del NCBI. El procés de descarrega de microarrays es realitzarà cada dos mesos y estarà sincronitzat amb un procés de descarrega de gens marcadors de microarrays del NCBI.

A la memòria també es descriuen las fases realitzades per crear la interfície web para gestionar aquestes microarrays públiques y las modificacions realitzades sobre l'aplicatiu web per permetre realitzar anàlisis amb aquestes microarrays.

Summary

The IBB center has developed a application server: http://revolutionresearch.uab.es to analyze the microarrays. These microarrays are obtained and uploaded to the local database by the application users.

This report details the process undertaken in order to automate the public microarrays upload to the local database. These microarrays will obtain from the NCBI. The download process will perform every two months and will be synchronized with a download process of gene markers of the NCBI.

The report also describes the steps taken to create the web interface to manage these public microarrays and the changes made on the web application to allow these microarray analysis.