Universitat Autònoma de Barcelona

INGENIERÍA INFORMÁTICA

2657 BIOINFORMÁTICA:

Interfaz web para mostrar los genes según las relaciones que mantienen sus expresiones

Memoria del Proyecto de Final de Carrera
de Ingeniería Informática
realizado por
Luis Alberto Hernandez Gracia
y dirigido por
Jordi Gonzàlez Sabaté

y Mario Huerta
Bellaterra, 21 de Junio de 2011

Universitat Autònoma de Barcelona

Escola Tècnica Superior d'Enginyeria

El sotasignat, Jordi Gonzàlez 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 Luis Alberto Hernandez Gracia.

I per tal que consti firma la present.

Signat: .....

Bellaterra, ...de.....de 2011

Universitat Autònoma de Barcelona

etse Escola Tècnica Superior d'Enginyeria

El sotasignat, Mario Huerta

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

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Luis Alberto Hernandez Gracia.

I per tal que consti firma la present.

Signat: .....

Bellaterra, ...de.....de 2011

Agradecimientos

En primer lugar me gustaría agradecer a mi tutor del proyecto, Mario Huerta, la colaboración, paciencia y confianza depositada en mí y en mi trabajo. Sin su guía y sus consejos este proyecto no habría sido posible.

Me gustaría dedicar este proyecto a mi pareja, mi amiga y mi compañera, que siempre me brindó su apoyo y comprensión durante el desarrollo de este proyecto. Gracias Marta.

Índice

1. Introducción

1.1 Motivación del proyecto

En el momento de comenzar mi formación universitaria me debatía entre la Informática y la Biotecnología. Eran los dos campos de la ciencia que me atraían con más fuerza y consecuentemente mis dos primeras opciones en la lista de preinscripción universitaria. Aunque me decliné por el mundo de la informática, la Biotecnología sigue siendo aún una de mis inquietudes.

Es por ello que la principal motivación que me ha llevado a escoger este proyecto, a parte de mi interés previo en la Biotecnología, ha sido la posibilidad de aplicar los conocimientos, técnicas y herramientas de la informática a la investigación relacionada con la medicina y la genética. Desde el primer momento como este proyecto, una vez finalizado, sería utilizado por investigadores en el campo de la biotecnología para comprender la relación que tienen la expresión de algunos genes con la aparición de enfermedades como por ejemplo el cáncer.

Otro motivo para decidirme fue que este proyecto lo ofrecía el Institut de Biotecnología i BioMedicina (IBB) de la Universitat Autònoma de Barcelona. Esta entidad, debido a su naturaleza multidisciplinar, plantea soluciones que combinan los avances y técnicas de Bioinformática, Genómica, Biología celular y otras ciencias frente a los importantes desafíos y problemas biológicos que se plantean en sus investigaciones.

1.2 Estado del arte

Mi proyecto se engloba dentro del campo de la Bioinformática, ciencia dedicada a la aplicación de la tecnología de computadores a la gestión y el análisis de datos biológicos. Más concretamente, el área de investigación dentro de la Bioinformática en el que se adscribe el presente proyecto es el del análisis de la expresión génica mediante microarrays.

Las microarrays son superficies solidas donde se unen fragmentos de ADN y se observa el nivel de expresión de los genes bajo diferentes condiciones experimentales.

Cuando los genes se expresan se sintetiza la proteína correspondiente a cada gen.
Las proteínas sintetizadas, actuando independiente o conjuntamente con otras proteínas sintetizadas, dan lugar a las diferentes funciones celulares. En algunos casos desarrollarán las funciones específicas del tejido y de esta manera llevarán a cabo las funciones de cada tejido que hará funcionar al organismo.

Un análisis típico de los datos generados por microarrays es el análisis de las relaciones de expresión entre los genes de la microarray. La relación de expresión entre dos genes describe el nivel de expresión de un gen respecto al nivel de expresión del otro gen. Las relaciones de expresión pueden ser de diferentes tipos.

Los genes que están coexpresados son genes que se expresan a la vez, es decir, tienen una relación de expresión lineal. Al expresarse a la vez, sus proteínas se sintetizan a la vez. Al sintetizarse sus proteínas simultáneamente, estas pueden interactuar entre ellas. La interacción entre proteínas es un proceso muy importante ya que determina tanto procesos biológicos totalmente funcionales como procesos patológicos. Por ejemplo en la investigación de nuevos tratamientos contra el cáncer se ha visto que al bloquear la interacción entre las proteínas eIF4E y eIF4G se logra frenar el desarrollo de algunas células cancerosas y se provoca su muerte [1].

La relación de expresión entre dos genes puede ser también no lineal. Una relación no lineal de expresión entre dos genes implica que la relación entre los niveles de expresión de ambos genes describe una curva. Por ejemplo una relación de expresión entre dos genes de tipo ln(X) implica que un gen ha de sobreexpresarse para que pase a expresarse el otro gen.

A partir de los datos de microarray pueden extraerse tanto las relaciones de expresión lineales como las relaciones de expresión no lineales que se existen entre los genes de la microarray para las diferentes condiciones muestrales de la microarray.

El método estadístico mediante el cual nosotros obtenemos la relación no lineal entre las expresiones génicas son las *Principal Curves of Oriented Points* (PCOP) definidas por *Delicado* (2001) [2]. En *Delicado and Huerta* (2003) [3], se presenta el método algorítmico para realizar el cálculo de las PCOP. En *Cedano et al* (2007) [4] se propone una aplicación web que permite el uso del método de las PCOP para el análisis de relaciones no lineales entre variables en el campo de la biomedicina. En *Huerta, Cedano and Querol* (2008) [5] se describe como el cálculo de las PCOP se puede utilizar para el análisis masivo de las relaciones no lineales en la expresión génica a partir de los datos de microarrays.

Los investigadores del IBB han creado entonces, un conjunto de herramientas y aplicativos web para el análisis de microarrays cuya base es el cálculo de las PCOP. Dichos aplicativos están disponibles en el servidor revolutionresarch.uab.es [6]. En dicho servidor se encuentra la aplicación web PCOPGene-Net. PCOPGene-Net se presenta en Huerta *et al* (2009) [7].

PCOPGene-Net tiene como objetivo estudiar las relaciones de expresión entre genes a partir de los datos obtenidos con la técnica de microarrays. La aplicación web propicia dicho análisis de forma muy visual e interactiva mediante un grafo interactivo generado para cada microarray analizada, así como por un conjunto de interfaces gráficas que se lanzan desde el grafo el grafo interactivo.

PCOPGene-Net incorpora funcionalidades de NCR-PCOPGene, una aplicación web anterior que también facilita el análisis masivo de microarrays usando PCOP para realizar clases muestras de la microarray y estudiar su efecto sobre las fluctuaciones en las relaciones de

las expresiones génicas. NCR-PCOPGene está descrita en *Cedano, Huerta and Querol* (2008)[8].

Los nodos del grafo interactivo son los genes de la microarray y las aristas son las relaciones de expresión entre los diferentes genes. En numerosos manuales [9] dentro del servidor se explica paso a paso como utilizar la herramienta web PCOPGene-Net.

PCOPGene-Net está basado en los lenguajes de programación PHP, CGIs en C++, Java i Flash. Para representar el grafo de relaciones entre genes se utiliza una librería Java open-source llamada JUNG (Java Universal Network/Graph Framework) [10]. Esta librería proporciona un lenguaje extensible para el modelado, análisis y visualización de datos representados por grafos o redes. JUNG dispone de un entorno de trabajo para la visualización que facilitará la exploración interactiva del grafo. También dispone de mecanismos de filtrado para tratar partes especificas del grafo.

En el servidor del IBB, existe un preproceso que se ejecuta cuando una nueva microarray es subida al servidor. Cuando los resultados del preproceso han sido calculados la aplicación online PCOPGene-Net queda accesible al usuario.

En el preproceso los datos de las microarrays a analizar que se han subido a nuestro servidor son tratados para calcular e identificar los tipos de relaciones de expresión entre los genes de la microarray.

Dentro del preproceso también se realiza una agrupación en clústeres de los genes pertenecientes al microarray basada en el tipo de relaciones de los genes. De esta forma, la agrupación de genes en clústeres permitiría identificar grupos de genes que mantienen el mismo tipo de relación de expresión con el resto de genes de la microarray.

Para el proceso de agrupación de genes en clústeres basada en el tipo de relaciones de los genes se hace uso de CLUTO [11]. CLUTO es una herramienta de clustering que permite calcular estos clústeres de genes y extraer información tanto estadística como gráfica de los clústeres formados.

Actualmente se calculan los clústeres de genes pero no se dispone de una interfaz web con la que el usuario pueda identificar qué genes forman parte de cada uno de los diferentes clústeres.

Por otro lado, PCOPGene-Net permite identificar sobre el grafo interactivo los diferentes tipos de relaciones de expresión no lineales que existen entre los genes de la microarray analizada. La identificación se consigue asociando colores a los diferentes tipos de relaciones de expresión no lineales, de forma que las aristas que representan las relaciones de expresión quedan coloreadas con el color del tipo. Dicha herramienta sufre algunos problemas que es necesario solucionar y necesita incorporar nuevas funcionalidades que permitan mejorar la experiencia de usuario del investigador que hace uso de PCOPGene-Net.

Algunos de los problemas a solucionar en PCOPGene-Net son:

- pintado incorrecto o pintado por duplicado de las aristas entre dos genes. Pintado por duplicado de sus correspondientes grados de correlación.
- La aplicación no recuerda la última asignación de colores a cada uno de los diferentes tipos de relaciones de expresión no lineales.
- Es necesaria la optimización de algunos procesos de pintado y coloreado del grafo que ralentizan la interactividad entre el usuario y el applet PCOPGene-Net.

1.3 Objetivos

El principal objetivo de este proyecto es facilitar la tarea de investigación a los usuarios del aplicativo PCOPGene-Net. Este objetivo principal se conseguirá permitiendo que en la aplicación se pueda:

1. Identificar los genes de una microarray según la relación de expresión que mantienen con el resto de genes.
2. Mejorar el proceso de identificación de los diferentes tipos de relaciones de expresión entre los genes de una microarray.

Para cumplir satisfactoriamente el propósito general de este proyecto he definido los siguientes objetivos:

1. **Generar una gráfica con la relación entre los clústeres de genes por los tipos de relaciones de expresión de cada gen con los diferentes tipos de relación de expresión, para mostrarla en PCOPGene-Net:**
   Poder presentar al usuario de PCOPGene-Net de manera gráfica e intuitiva el resultado del proceso de clustering de genes basado en los tipos de relaciones de expresión de cada gen.

   Para presentar dicha gráfica es necesario adaptar el preproceso existente de clustering de genes por los tipos de relaciones de cada gen.

   a. Adaptaré el preproceso para generar de manera automática una representación gráfica de la relación de los diferentes clústeres de genes con los diferentes tipos de relaciones de expresión.

   b. Será necesario tratar las imágenes generadas para que puedan ser visualizadas e interpretadas fácilmente por el applet de PCOPGene-Net. El procesado de las imágenes ha de tratar correctamente microarrays de

cualquier tamaño (las hay de 1000 a 20000 genes) y cualquier número de clústeres.

2. **Identificación sobre el grafo interactivo de los genes que pertenecen cada clúster en PCOPGene-Net:**
   Permitir al usuario de PCOPGene-Net identificar de manera gráfica los genes que forman parte de cada clúster calculado, así podrán ver todos los genes que mantienen una relación de expresión parecida con el resto de genes de la microarray.

   Para facilitar la identificación de "a qué clúster pertenece cada gen" en el grafo interactivo de PCOPGene-Net es necesario añadir una nueva interfaz gráfica que mediante un menú permita asignar diferentes colores a cada uno de los clúster obtenidos en el preproceso. Será necesario también modificar el proceso de coloreado de nodos en el grafo interactivo para que los genes de cada clúster en concreto tomen el color que se le ha asignado a dicho clúster.

3. **Mejorar la identificación de los diferentes tipos de relaciones de expresión no lineales en PCOPGene-Net:**
   Permitir al usuario de PCOPGene-Net identificar de manera gráfica los diferentes tipos de relaciones de expresión no lineales entre los genes de la microarray analizada.

   Para facilitar la identificación de las diferentes relaciones de expresión no lineales entre los genes en el grafo interactivo corregiré los errores que contiene la interfaz gráfica que permite colorear las aristas del grafo en función del tipo de relación de expresión que representen. También incorporaré a esta interfaz gráfica nuevas funcionalidades que mejoren la experiencia de usuario. Será necesario modificar y corregir el proceso de coloreado de las aristas para pintar correctamente cada arista a mostrar y que se pinte con el color que se le haya asignado.

4. **Mejora de rendimiento y recursos de PCOPGene-Net:**

   La implementación de las nuevas funcionalidades tiene que ser lo más óptima posible, ya que las microarrays sobre las que se ejecuta PCOPGene-Net pueden tener miles de genes. La cantidad de genes incide directamente en el rendimiento y la interactividad, que son aspectos críticos en esta aplicación. Es por esto que durante el desarrollo de las nuevas funcionalidades tengo que optimizar todo el código al máximo, igual que al diseñar los algoritmos, introduciendo alternativas, si así fuera necesario, a la metodología usada actualmente en PCOPGene-Net.

1.4 Organización de la memoria

Para poder documentar correctamente las interfaces gráficas creadas, el conjunto de adaptaciones y optimizaciones realizadas sobre el aplicativo web PCOPGene-Net he dividido la memoria en ocho partes, siendo esta introducción la primera. En el resto de partes podemos encontrar:

- En la segunda parte explico los fundamentos teóricos básicos que el lector debe conocer para situarse en el contexto adecuado para entender lo aplicado.
- En la tercera describo el planteamiento que se ha desarrollado para cumplir los objetivos propuestos, así como la planificación seguida para llevarlo a cabo.
- En la cuarta expongo los resultados obtenidos durante el desarrollo de las fases planificadas.
- En la quinta describo el software creado y el software utilizado durante el desarrollo del proyecto. Esto incluye, cómo los programas son llamados, los parámetros y ficheros de entrada, los ficheros de salida, la estructura de estos archivos, la descripción y uso de las interfaces gráficas creadas, etc.
- En la sexta parte, expongo las conclusiones finales e impresiones personales sobre el proyecto realizado.
- En la séptima parte se encuentran las referencias bibliográficas utilizadas a lo largo de la memoria
- Por último, en la octava parte se encuentra un resumen breve del proyecto.

2. Fundamentos teóricos

2.1 Microarrays

La tecnología de las microarrays es una de las diversas aproximaciones al análisis comparativo de patrones de expresión de fragmentos de ADN cuyo fin es colocar en una microarray, una superficie sólida con pequeñas casillas dispuestas en forma de matriz, cada uno de los genes de un genoma cuyos niveles de expresión pueden ser cuantificados. Para ello se sintetiza los fragmentos de ADN y se insertan de forma automática en una capa de cristal, silicio o plástico, colocándose en las casillas que actúan a modo de tubo de ensayo.

Después se hibrida y se elimina todas las cadenas que no se han unido mediante lavados (sólo las moléculas que hibridan permanecerán en el microarray), y se procede al revelado mediante un escáner óptico o con microscopía láser confocal. Después el trabajo lo realiza la informática analizando la imagen obtenida para obtener un patrón de intensidades en cada casilla.

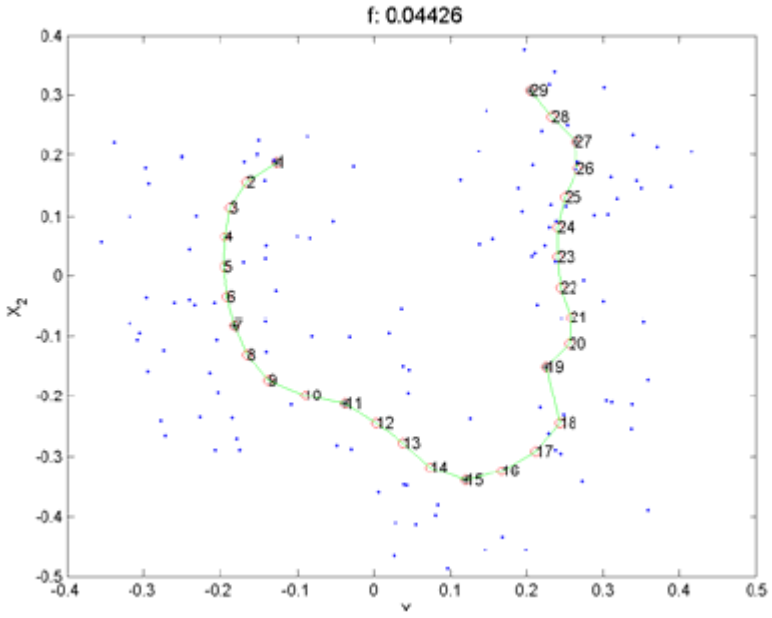
Las microarrays suelen utilizarse para identificar genes con una expresión diferencial bajo condiciones distintas. Por ejemplo, para detectar genes que producen ciertas enfermedades mediante la comparación de los niveles de expresión entre células sanas y células que están desarrollando ciertos tipos de enfermedades.

2.2 PCOP

Las curvas principales se usan para extraer patrones de comportamientos no lineales en el análisis de datos multivariable. Este método describe las relaciones entre variables independientes mediante una curva continua que pasa a través de una nube de puntos discretos, en nuestro caso microarrays de ADN.

Las PCOP se definen como una generalización a nivel local de la siguiente propiedad de las componentes principales: *"Para una distribución normal multivariante X, si X se proyecta sobre un hiperplano, la varianza total de la proyección es mínima cuando el hiperplano es ortogonal al primer componente principal"*. Los vectores locales de las componentes principales serán los vectores tangentes a las PCOP.

En la figura se puede observar la curva principal calculada para esos dos genes y la densidad y varianza de la nube de puntos durante su trayectoria. La curva muestra la relación no lineal entre los dos genes. En la parte superior de la figura puede observarse el grado de correlación de la relación, la cual es proporcionada por el cálculo de la PCOP.



PCOP de relación entre dos genes

Los diferentes tipos de curvas con las que se trabaja en PCOPGene-Net son los siguientes:

A26-L1_-CERR-CERR-AALT
A26-L1_-CERR-CERR-ABAJ
A26-L1_-CERR-ORTO-AALT
A26-L1_-ORTO-CERR-AALT
A26-L2_-CERR-CERR-AALT
A26-L2_-CERR-CERR-ABAJ
A26-L2_-CERR-ORTO-AALT
A26-L2_-ORTO-CERR-AALT

A26-___-ORTO-ORTO-AALT
A48-L1_-CERR-CERR-AALT
A48-L1_-CERR-CERR-ABAJ
A48-L1_-CERR-ORTO-AALT
A48-L1_-ORTO-CERR-AALT
A48-L2_-CERR-CERR-AALT
A48-L2_-CERR-CERR-ABAJ
A48-L2_-CERR-ORTO-AALT
A48-L2_-ORTO-CERR-AALT
A48-___-ORTO-ORTO-AALT
A51-L1n-ABIE-CERR
A51-L1n-CERR-CERR
A51-L1n-CERR-ORTO
A51-L1n-ORTO-CERR
A51-L1p-CERR-CERR
A51-L1p-CERR-ORTO
A51-L1p-ORTO-CERR
A51-L2_-CERR-CERR
A51-L2_-ORTO-CERR
A51-___-ORTO-ORTO
ACIRCULAR

Su nombre viene determinado por las características de la curva donde:

- Orientación de la horizontal
  - *A26*
  - *A48*
  - *A51*

- Posición de los puntos que forman la curva respecto a la horizontal.
  - *L1_*
  - *L1p*
  - *L1n*
  - *L2_*
- Angulo de sus extremos
  - *ORTO*
  - *CERR*
  - *ABIE*
- Amplitud máxima
  - *ABAJ*
  - *AALT*
- Caso especial: Curva circular
  - *ACIRCULAR*

2.3 Minimum Spanning Tree

Un grafo ponderado es aquel grafo en el que sus aristas tienen un peso asociado. Un grafo no dirigido es aquel grafo donde el par de vértices que representa una arista no está ordenado. Por lo tanto, los pares (v1, v2) y (v2, v1) representan a la misma arista.

Dado un grafo ponderado y no dirigido, definimos el árbol de expansión mínima o *Minimum Spanning Tree* (MST) como el conjunto de aristas que no forman ciclos y que conectan todos los vértices del grafo con una suma de pesos de aristas mínima. El MST cumple que su número de aristas es igual al número de vértices del grafo menos uno.

Existen varios algoritmos para encontrar el MST de un a partir de un grafo ponderado y no dirigido: el algoritmo de Kruskal y el algoritmo de Prim.

El algoritmo de Prim consiste en:

1. Se marca un nodo cualquiera, será el nodo de partida.
2. Seleccionamos la arista de menor valor incidente en el nodo marcado anteriormente, y marcamos el otro nodo en el que incide.
3. Repetir el paso 2 siempre que la arista elegida enlace un nodo marcado y otro que no lo esté.
4. El proceso termina cuando tenemos todos los nodos del grafo marcados.

El algoritmo de Kruskal consiste en:

1. Se marca la arista con menor valor. Si hay más de una, se elige cualquiera de ellas.
2. De las aristas restantes, se marca la que tenga menor valor, si hay más de una, se elige cualquiera de ellas.
3. Repetir el paso 2 siempre que la arista elegida no forme un ciclo con las ya marcadas.
4. El proceso termina cuando tenemos todos los nodos del grafo en alguna de las aristas marcadas, es decir, cuando tenemos marcados *n-1* arcos, siendo *n* el número de nodos del grafo.

3. Fases

Para el desarrollo del proyecto planifiqué el trabajo a realizar en diferentes fases secuenciales y dependientes una de otra. Por lo tanto, cada fase tenía que ser finalizada para poder pasar al desarrollo de la siguiente. Esto remarca la necesidad de definir una serie de hitos en el trabajo a realizar en el presente proyecto, limitando así la posibilidad de desviarse tanto en tiempo como esfuerzo en el desempeño de cada una de las diferentes fases.

El trabajo a realizar se puede dividir en tres partes bien diferenciadas. Por una parte la identificación en PCOPGene-Net de los genes de la microarray agrupados por los tipos de relaciones de expresión que mantienen con el resto de genes de la microarray, o sea identificar el clúster al que pertenecen después del proceso de clustering. Por otra parte es necesario mejorar el proceso de identificación en PCOPGene-Net de los diferentes tipos de relaciones de expresión que mantienen los genes de la microarray analizada. Por último, es necesaria la aplicación de todas las optimizaciones posibles para mejorar el rendimiento de PCOPGene-Net.

La identificación de los genes de la microarray según al clúster al que pertenezcan requiere dividir el trabajo a realizar en dos fases diferentes. Primero es necesario adaptar el preproceso existente de clustering que permite agrupar los genes según su los tipos de relaciones de expresión que mantiene el gen con el resto de genes de la microarray. Esta adaptación tiene como objetivo generar una gráfica que muestre la relación entre los clústeres y los diferentes tipos de relación de expresión no lineal. Seguidamente se ha de crear la interfaz gráfica que permitirá mostrar en PCOPGene-Net los resultados del preproceso de clustering.

La mejora en la identificación en PCOPGene-Net de los diferentes tipos de relaciones de expresión ha sido planteada como una única fase. El proceso de optimización de rendimiento de la aplicación web PCOPGene-Net lo he planificado como una fase aparte.

En el último apartado de esta sección comparo la planificación inicial del trabajo a desarrollar comparándola con el tiempo real dedicado a cada fase.

A continuación procedo a detallar las diferentes fases anteriormente definidas.

3.1 Generar una gráfica con la relación entre los clústeres de genes por los tipos de relaciones de expresión de cada gen con los diferentes tipos de relación de expresión, para mostrarla en PCOPGene-Net.

Dentro del conjunto de preprocesos que forman parte de PCOPGene-Net existe el preproceso de clustering de genes por los tipos de relaciones de expresión cada gen, pero este preproceso se limita al cálculo de los clústeres propiamente dichos, carece de alguna forma de representar los resultados de dicho proceso de clustering. El preproceso de clustering de genes por los tipos de relaciones de expresión de cada gen se realiza con la herramienta de clustering CLUTO.

CLUTO, además de realizar procesos de clustering también permite mostrar el resultado de dichos procesos de dos formas, mediante estadísticas o bien de manera gráfica. Por lo tanto se hace necesario modificar el preproceso existente para mostrar dichos resultados.

Como PCOPGene-Net es una herramienta de investigación genómica esencialmente visual, he decidido que lo más coherente es que entre las diferentes formas de representación de los resultados de clustering que ofrece CLUTO, elija la representación gráfica.

Para obtener la representación gráfica de la relación entre los clústeres de genes por los tipos de relaciones de cada gen con los diferentes tipos de relación de expresión, es necesario modificar el preproceso de clustering existente. Por otra parte las imágenes generadas tendrán que ser tratadas para que se puedan integrar en la nueva interfaz gráfica de identificación sobre el grafo interactivo de los genes que forman cada clúster en PCOPGene-Net.

3.1.1 Modificación del preproceso de clustering de genes por los tipos de relaciones de expresión de cada gen

La herramienta de clustering CLUTO puede usarse de diferentes formas. Una de ellas es mediante llamadas a funciones de clustering implementadas en C++ disponibles en las librerías que contiene CLUTO. También puede utilizarse CLUTO como una aplicación de clustering independiente que puede llamarse mediante la consola de comandos.

Después de consultar el manual de uso de CLUTO he visto que para poder generar las imágenes que representan los resultados de clustering es necesario llamarlo por consola de comandos.

El preproceso existente de clustering de genes por los tipos de relaciones de expresión de cada gen, lo realiza un programa implementado en C++ que usa las funciones de clustering de las librerías proporcionadas por CLUTO. Este programa existente se llama Gencluster. Gencluster no solo realiza el proceso de clustering sino que también calcula el número óptimo de clústeres en el que se ha dividir el conjunto de genes de la microarray. Este cálculo de número de clústeres en los que dividir el conjunto de genes de la microarray es imprescindible ya que CLUTO lo necesita como parámetro de entrada.

La necesidad de conocer el número de clústeres en los que dividir el conjunto de genes de la microarray hace indispensable mantener la ejecución de gencluster dentro del preproceso de PCOPGene-Net. Por lo tanto después de la ejecución de gencluster tendrá que ejecutarse la llamada a CLUTO por consola de comandos para generar la imagen de resultados.

El preproceso existente de clustering de genes por los tipos de relaciones de expresión de cada gen realiza tres procesos de clustering diferentes:

1. Usando datos de entrada normalizados los genes.
2. Usando datos de entrada normalizados las clases de relación de expresión.
3. Usando datos de entrada normalizados los genes y las clases de relación de expresión.

Entonces para cada microarray analizada por PCOPGene-Net se tendrá que ejecutar una llamada a CLUTO por consola de comandos diferente, una para cada tipo de datos.

Uno de los archivos de resultados de Gencluster es el número de clústeres óptimo encontrado para los tres procesos de clustering, utilizaré el contenido de dicho archivo para especificar el número de clústeres en los que se dividirán los genes de la microarray.

La estrategia a seguir para solucionar esta primera parte de la fase es la siguiente:

1. Anotar todos los parámetros de entrada utilizados en las llamadas a funciones de clustering de CLUTO dentro del preproceso Gencluster.
2. Construir las llamadas a CLUTO por consola de comandos con los parámetros de entrada anotados en el paso anterior. Una por cada tipo de clustering.
3. Añadir al preproceso de PCOPGene-Net las tres llamadas a CLUTO por consola de comando justo después de la ejecución de Gencluster.
4. Comparar resultados de clustering obtenidos con la llamada por consola de comandos a CLUTO con los resultados de clustering de Gencluster.

3.1.2 Tratamiento de las imágenes obtenidas

Una vez obtenidas las imágenes que muestran la relación entre los clústeres de genes por los tipos de relaciones de expresión de cada gen con los diferentes tipos de relación de expresión, se tiene que diseñar un sistema que prepare dichas imágenes. Las imágenes tienen que ser procesadas para poder ser usadas en la nueva interfaz gráfica de identificación sobre el grafo interactivo de los genes que pertenecen a cada clúster en PCOPGene-Net.

La imagen generada se compone por tres áreas: el árbol jerárquico de los diferentes clústeres formados, la leyenda con los diferentes tipos de relaciones de expresión no lineales y la matriz gráfica que representa los genes agrupados en clústeres según la relación de expresión, donde la intensidad del color es el grado de correlación entre el clúster y tipo

de relación y el alto de las filas es proporcional al número de elementos que componen cada clúster. Se puede observar un ejemplo en la figura 1 que presento a continuación.

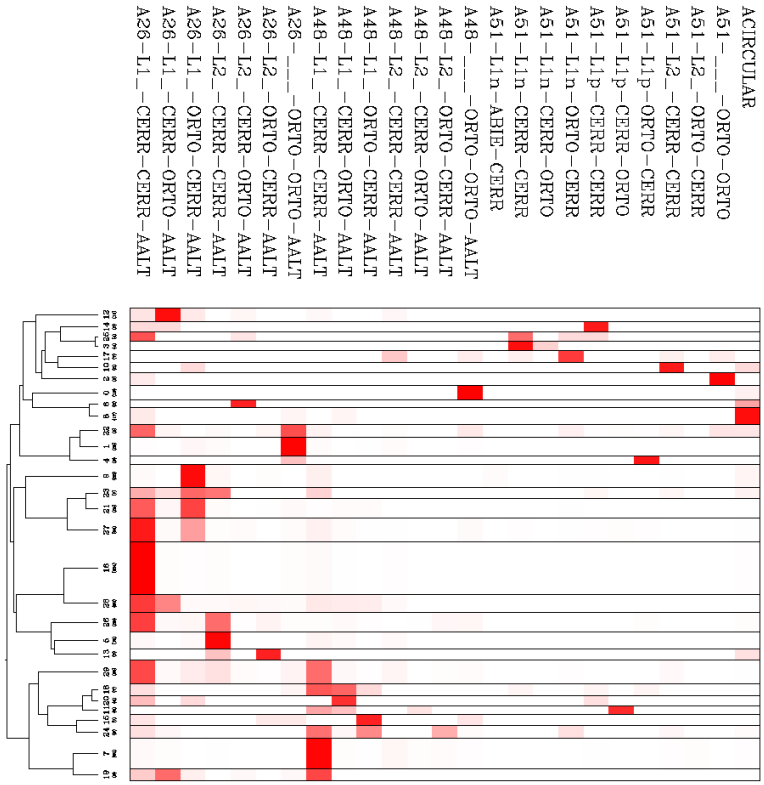


Figura 1. Ejemplo de imagen que muestra la dependencia entre los clústeres de genes por los tipos de relaciones de expresión de cada gen con y los diferentes tipos de relación de expresión. En el eje horizontal se pueden apreciar los diferentes tipos de relaciones de expresión no lineales encontrados para la microarray analizada. En eje vertical se observan los diferentes clústeres creados formando un árbol jerárquico. La intensidad del color es el grado de correlación entre el clúster y tipo de relación, y el alto de las filas es proporcional al número de elementos que componen cada clúster.

Tal como son generadas las imágenes por CLUTO las diferentes partes de la imagen no pueden ser visualizadas con facilidad por la orientación en que se presentan dichas partes. Estas tres áreas tendrán que ser separadas, rotadas y posteriormente unidas por un proceso de tratamiento de imagen para su correcta presentación en la nueva interfaz gráfica de identificación sobre el grafo interactivo de los genes que forman cada clúster en PCOPGene-Net.

Para poder rotar la leyenda y que sus elementos sean coherentes con la matriz gráfica se le pasará a la llamada de CLUTO la lista de clases de relación de expresión en orden inverso. Al introducir los elementos de la leyenda en orden inverso después de una rotación de 90 grados en sentido antihorario de la leyenda y 90 grados en sentido horario de la matriz gráfica, cada elemento de la leyenda estará alineado con la fila de la matriz gráfica que realmente le corresponde. La lista de clases de relación de expresión que conforman la leyenda está disponible en un fichero de resultado de un preproceso previo de PCOPGene-Net. La inversión de las clases de relación de expresión se realizará mediante un script sobre el fichero de clases de relaciones de expresión.

Como resultado se obtendrán dos archivos de imagen por separado para cada proceso de clustering. El primer archivo de imagen será el árbol de jerarquía de clústeres, ajustado al ancho de la matriz gráfica que representa los genes agrupados en clústeres según la relación de expresión. El segundo archivo de imagen será la composición de la matriz gráfica junto con la leyenda de las diferentes clases de relación de expresión a su derecha.

En el servidor del IBB, existe un repositorio de diferentes leyendas de tipos de relación de expresión no lineales. Las leyendas del repositorio se diferencian de las generadas por CLUTO en que cada tipo de relación de expresión no lineal está representado por el dibujo de la curva de relación de expresión en vez de por la descripción de los componentes de la curva. El dibujo de la curva de relación de expresión es mucho más descriptivo que la definición de los componentes de la curva de relación de expresión. En cada procesado de imagen se analizará si se puede usar alguna de las leyendas del repositorio del servidor. En la figura 2 se puede observar un ejemplo de una de las leyendas disponibles en el servidor.

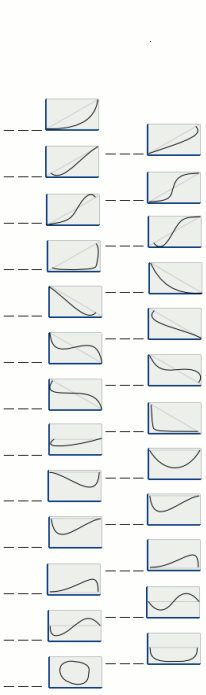


Figura 2. Ejemplo de leyenda existente en el repositorio del servidor. Esta leyenda muestra los diferentes tipos de relación de expresión no lineal mediante una imagen de la curva que los describe. Esta leyenda se acoplará al gráfico que muestra la correspondencia entre clústeres de genes por tipos de correlación y los tipos de correlación, de forma que los tipos de correlación queden descritos gráficamente.

Para el tratamiento de la imagen será necesario crear un script que mediante el uso de las instrucciones de procesado de imagen disponibles en el entorno UNIX se realicen los recortes, rotaciones y composiciones necesarias para la obtención de las imágenes en el formato deseado.

El procesado de la imagen tendrá que tener en cuenta los diferentes tamaños de microarrays posibles ya que el tamaño de leyenda y de la matriz de representación gráfica de la imagen generada por CLUTO depende de dicho tamaño. Por lo tanto los tamaños de recorte tienen que variar según el tamaño de la microarray que ha sido utilizada. Por el contrario el tamaño del árbol de jerarquía de los clústeres es fijo y conocido.

El procesado de la imagen tendrá que hacerse para las imágenes generadas para los tres tipos de clustering: datos normalizados por genes, por clases y por último, por clases y genes a la vez.

La estrategia para el procesado de las imágenes que se implementa en el script es la siguiente:

1. Obtención del tamaño de la microarray.
2. Obtención de las dimensiones de la imagen generada por CLUTO.
3. Recorte y rotación de 90 grados en sentido horario de la parte del árbol de jerarquía de los clústeres.
4. Recorte de la parte de la leyenda según el tamaño de la microarray obtenido.
5. Rotación de la leyenda 90 grados en sentido antihorario.
6. Rotación de la matriz gráfica 90 grados en sentido horario.
7. Elección entre la leyenda generada por CLUTO y las existentes en el servidor.
8. Composición de matriz gráfica de clustering junto con la leyenda.

3.2 Identificación sobre el grafo interactivo de los genes que pertenecen a cada clúster en PCOPGene-Net.

Dado que ya se dispone del resultado del proceso de clustering de los genes según la relación de expresión con el resto de genes de la microarray, ya se puede crear la interfaz gráfica que permita identificar los genes que pertenecen a cada clúster sobre el grafo interactivo de PCOPGene-NET. Para identificar los genes se tiene que permitir escoger un color para cada clúster generado y el tipo de clustering que el usuario desea visualizar.

Una vez seleccionado un tipo de clustering y los colores deseados para los clústeres, el proceso de coloreado de los genes sobre el grafo interactivo tiene que pintar dichos genes con el color que les corresponda. Para el correcto coloreado se hace necesario personalizar el proceso de coloreado de vértices del grafo interactivo que proporciona la librería JUNG.

3.2.1 Creación de la interfaz gráfica de identificación de los genes que pertenecen a cada clúster.

El primer paso es decidir qué elementos tiene que contener la interfaz gráfica de identificación de los genes que pertenecen a cada clúster para poder satisfacer el objetivo propuesto. Se tiene que destinar una parte de la interfaz gráfica a la asignación de un color para cada clúster, esto se hará mediante un menú de coloreado de los genes de cada clúster. Para cada microarray analizada por PCOPGene-Net tenemos tres resultados de clustering distintos, uno por cada tipo de clustering. Cada tipo de clustering de genes por las relaciones de expresión de cada gen, tiene un número de clústeres que puede ser diferente al de los otros tipos de clustering. Se requiere por lo tanto de una estructura dinámica que permita representar en un espacio fijo diversas cantidades de clústeres.

Además de ser dinámica tiene que ser interactiva. Cuando un usuario de PCOPGene-Net elige un color para uno de los clústeres tiene que poder hacerlo de la manera más intuitiva posible. Una vez escogido el color para el clúster se tiene que poder ver en todo momento que el color está asignado al clúster.

Dado que se utiliza la librería gráfica Java Swing para la construcción de las interfaces gráficas de PCOPGene-Net, después de un periodo de investigación, he decidido que la estructura de tabla dinámica e interactiva JTable de Java Swing cumple con las necesidades requeridas:

- Es una tabla que se compone de las celdas que forman sus filas y columnas, ideal para mostrar el número de clúster y su color asociado de forma clara.
- Para un tamaño absoluto en pantalla puede contener diversas cantidades de filas y columnas.
- Al clicar sobre ella es capaz de detectar en que celda se ha clicado y se puede definir la acción a realizar al ser clicada una de sus celdas.
- Su visualización es completamente personalizable.

Una vez que está clara la estructura que se usará también es necesario definir de qué manera se seleccionará el color para cada clúster. Dado que con la implementación actual del preproceso de clustering de genes por los tipos de relaciones de expresión de cada gen podemos tener un máximo de hasta 40 clústeres, es necesario disponer de una amplia paleta de colores para que no sean muy parecidos entre ellos y así poder diferenciarlos de manera clara sobre el grafo interactivo.

Después de barajar diferentes opciones, he decidido usar JColorChooser, otro componente de Java Swing que tal como se aprecia en la figura 3, permite escoger un color de entre diferentes paletas de colores como RGB, HSB, etc. JColorChooser es altamente personalizable, permite escoger que paletas de colores se quieren utilizar y que acciones se tienen que realizar una vez escogido un color de la paleta.

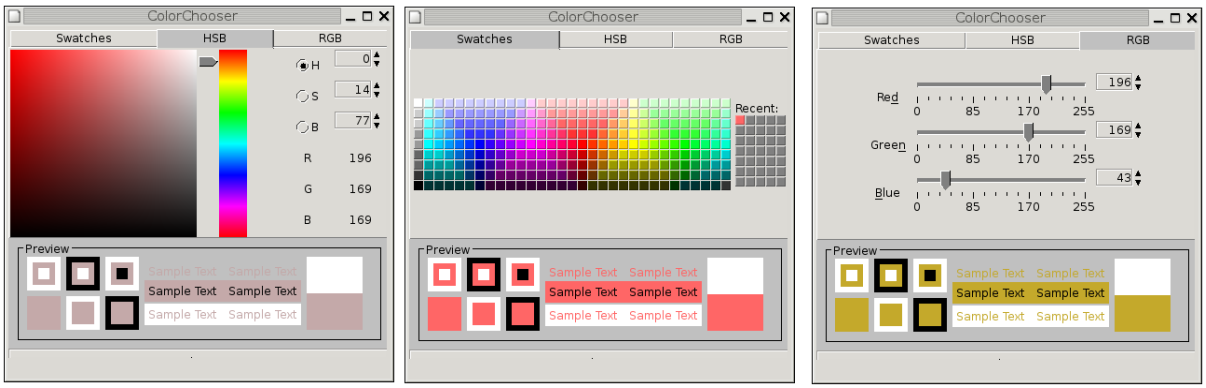


Figura 3. Ejemplo del componente JColorChooser de Java Swing que utilizaré en la interfaz gráfica del menú para asociar un color a cada clúster de genes que pertenecen a cada clúster. Este componente personalizable permite escoger un color de entre diferentes paletas de colores estándar.

Con el objetivo de ofrecer al usuario de PCOPGene-Net una visión global del resultado del proceso de clustering de genes por los tipos de relaciones de expresión de cada gen quiero presentar en la interfaz gráfica las imágenes que muestran la relación entre los clústeres de genes con los tipos de relación de expresión que definen cada clúster. Estas imágenes ya vendrán cortadas en dos trozos y sus partes estarán perfectamente orientadas gracias al procesado que se ha hecho sobre ellas en el preproceso.

Dado que tenemos tres tipos de clustering procesados sobre los genes de la microarray con sus respectivos clústeres generados, la interfaz gráfica de identificación de los genes de cada clúster tiene que permitir escoger sobre qué tipo clustering se van a colorear los clústeres. Para seleccionar el tipo de clustering deseado es necesario añadir a la interfaz gráfica las diferentes opciones disponibles y que estas sean seleccionables. Como solo procesamos tres tipos de clustering, no es necesario agruparlos en un desplegable o componente de agrupación.

Por lo tanto he decidido mostrar los tipos de clustering seleccionables por separado, cada uno de ellos con su correspondiente botón o casilla de selección. Cada vez que se seleccione un tipo de clustering tendrá que cargarse la tabla interactiva que representa los clústeres generados para ese tipo de clustering y se tendrá que cargar la imagen que presenta los resultados del proceso de clustering para el tipo de clustering seleccionado.

Aunque se cierre el menú de coloreado de los genes de cada clúster o se cambie de tipo de clustering, se tiene que recordar las colores escogidos para los diferentes clústeres. Dado que he optado por usar 3 tablas dinámicas interactivas, una por cada tipo de clustering no existe ningún problema en cumplir este requisito.

La estrategia a seguir para cumplir todos los requerimientos descritos es la siguiente:

- Creación de tres tablas dinámicas interactivas, una para cada tipo de clustering aplicado en el preproceso.
- Creación del desplegable que presenta la paleta de colores.
- Creación y carga de las tres parejas de imágenes que representan el resultado de los diferentes tipos de clustering realizados en el preproceso.
- Creación de las casillas de selección de los tres tipos de clustering.
- Creación del botón de cierre del menú de coloreado de genes de cada clúster y del botón de reseteo de las selecciones de colores.

3.2.2 Modificación del proceso de coloreado de vértices del grafo interactivo

Los vértices del grafo interactivo de PCOPGene-Net representan los genes del microarray analizado. Para poder identificar los genes según el clúster al que pertenecen es necesario poder colorear los vértices según el color asignado por el usuario a cada clúster. Es necesario por tanto personalizar el proceso de coloreado de vértices del grafo interactivo.

Para saber a qué clúster pertenece cada gen de la microarray analizada por PCOPGene-Net se tiene que consultar los archivos de resultados del preproceso de clustering. Los archivos de resultado del preproceso de clustering relacionan cada gen de la microarray con el clúster al cuál ha sido asignado.

La librería JUNG permite la personalización de muchos procesos de visualización del grafo interactivo. La personalización es posible creando una nueva clase Java que implemente la clase abstracta de coloreado de vértices de JUNG, VertexPaintFunction. PCOPGene-Net ya

contiene una clase Java, MyVertexPaintFunction, que implementa la clase abstracta VertexPaintFunction.

El proceso de pintado del grafo interactivo que proporciona JUNG hace una llamada a la función getFillPaint de MyVertexPaintFunction para cada vértice que se esté visualizando en ese momento. La función getFillPaint retorna el color con el que se ha de pintar el vértice.

La implementación existente de getFillPaint colorea todos los vértices de color rojo, con la excepción de cuando un vértice está seleccionado. En caso de que un vértice esté seleccionado lo pinta de color amarillo.

Es necesario añadir a la implementación existente el caso en que el usuario desee colorear los vértices según el clúster al que pertenecen. Para decidir el color con el que se ha de pintar cada vértice he decidido seguir la siguiente estrategia:

SI el coloreado de genes según el clúster al que pertenecen está activado ENTONCES:
    Consultar que tipo de clustering está seleccionado para mostrar.
    Consultar el archivo respectivo de resultado del preproceso de clustering para saber que clúster le corresponde al gen que se pretende pintar.
    Consultar que color ha asignado el usuario a ese clúster en la tabla dinámica interactiva correspondiente.
    Retornar dicho color.
SINO SI el coloreado de genes según el clúster al que pertenecen está desactivado ENTONCES:
    SI el vértice está seleccionado ENTONCES
        Retornar color amarillo.
    SINO
        Retornar color rojo.
    FIN SI.
FIN SI.

3.3 Mejorar la identificación de los diferentes tipos de relaciones de expresión no lineales en PCOPGene-Net

Para permitir al usuario de PCOPGene-Net poder identificar correctamente sobre el grafo interactivo los diferentes tipos de relaciones de expresión no lineales existentes entre los genes de la microarray es necesario corregir algunas funcionalidades de la interfaz gráfica que permite asignar colores a cada uno de dichos tipos de relaciones de expresión no lineales.

Para que la selección tenga efecto sobre el coloreado de las aristas del grafo interactivo, que son las que representan las relaciones de expresión entre los genes, será necesario corregir el proceso personalizado de coloreado de aristas que existe en PCOPGene-Net.

La funcionalidad de mostrar el grado de correlación de las relaciones de expresión, que es el peso de cada arista, también tendrá que ser modificado acorde con las correcciones que se apliquen en el pintado de las aristas. Solo se mostraran los grados de correlación de aquellas relaciones de expresión que se estén pintando en ese momento.

3.3.1 Corrección de la interfaz gráfica de coloreado de las relaciones de expresión no lineales.

Actualmente PCOPGene-Net contiene una interfaz gráfica (ver figura 4) que permite escoger el color con el que se quiere colorear cada uno de los 29 diferentes tipos de relaciones de expresión no lineales. Como este número no es alto hay espacio suficiente para representar cada una de ellos por separado dentro de la interfaz gráfica mediante el dibujo de la curva que describen.

Como las líneas de las aristas que se dibujan en el grafo interactivo tienen que poder verse bien no sirve cualquier color, ya que los colores claros sería muy difícil distinguirlos. Así que en vez de ofrecer al usuario todos los colores de una paleta estándar, permite escoger entre

una serie de colores predeterminados que son suficientemente intensos para que se puedan distinguir correctamente.

En PCOPGene-Net las relaciones de expresión no lineales están filtradas por su factor de correlación. Por defecto se permite colorear aquellas relaciones de expresión no lineales que te tengan un factor de correlación < 0.05, aunque si el usuario lo decide puede colorear las relaciones de expresión que cumplan 0.05 < factor de correlación <0.08. Por este motivo se filtra mediante un checkbox con nombre “Less restrictive filter” que tipo de relaciones no lineales de expresión se quiere visualizar.

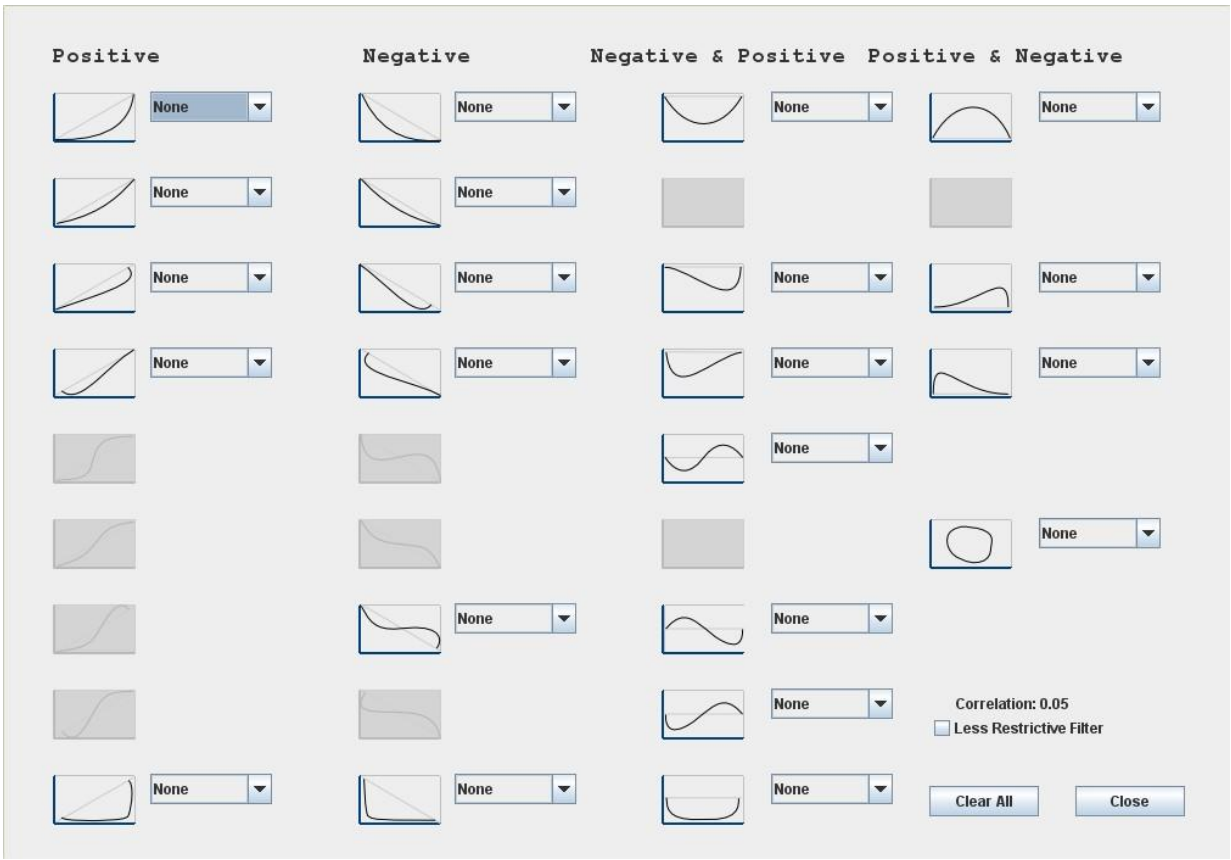


Figura 4. Menú de coloreado de los tipos de relaciones no lineales de PCOPGene-Net. En el menú se muestran los diferentes tipos de relaciones de expresión no lineales representados por una imagen de la curva que describen y acompañados por un desplegable donde escoger el color. También se pueden apreciar la casilla de filtro de correlación y los botones de reseteo de los colores asociados y cierre del menú.

Solo se presentan activos los tipos de relaciones de expresión no lineales que están presentes en la microarray analizada y que cumplan con el valor del filtro de correlación que este activo en ese momento.

Recordar las selecciones de colores:

Una vez seleccionados los colores deseados para los tipos de relaciones de expresión no lineales, actualmente no recuerda cual era esa selección una vez se cierra el menú o se cambia el filtro de relaciones no lineales. Por lo tanto es necesario guardar la relación de las últimas selecciones realizadas por el usuario. Esta selección de colores tiene que recordarse por separado cuando el filtro "Less restrictive filter" está activado y cuando no lo está.

Para guardar las dos selecciones por separado será necesario recordar el color seleccionado para cada una de las 29 relaciones de expresión no lineales y por duplicado, color seleccionado con el filtro "Less restrictive filter" desactivado y color seleccionado con el filtro "Less restrictive filter" activado.

Para cumplir este objetivo he decidido que guardaré en variables estáticas estas elecciones de color para cada una de las 29 relaciones de expresión no lineales, una versión para el filtro "Less restrictive filter" desactivado y otra versión si está activado. Tiene que ser variables estáticas para que así sean accesibles desde las clases encargadas del coloreado de aristas del grafo interactivo mediante llamadas estáticas.

Reseteo de las selecciones de colores:

Cuando el usuario quiere limpiar todas las selecciones de colores hechas hasta el momento, el botón de reseteo no borra correctamente las selecciones hechas. Además el borrado tiene que afectar solo a las selecciones guardadas para el estado actual del filtro "Less restrictive filter", activado o desactivado.

Para poder corregir esta funcionalidad es necesario definir una función para cada estado posible de "Less restrictive filter" que permita resetear todas las elecciones de colores pero

solo para la versión de las 29 variables estáticas asociada al estado actual del filtro “Less restrictive filter”.

3.3.2 Corrección y personalización del proceso de coloreado de aristas del grafo interactivo.

Las aristas del grafo interactivo de PCOPGene-Net representan las relaciones de expresión entre los genes de la microarray que está siendo analizada. Para el coloreado de las aristas del grafo se tiene que implementar la función abstracta AbstractEdgePaintFunction que proporciona JUNG. Actualmente PCOPGene-Net ya incorpora la función MyEdgePaintFunction donde está codificado el coloreado de las aristas del grafo, o sea de las relaciones de expresión.

Sin embargo MyEdgePaintFunction no proporciona un coloreado del todo correcto para las aristas del grafo que representan las relaciones de expresión no lineales. Concretamente se han detectado los siguientes errores:

- Entre dos genes del grafo interactivo se pintan dos relaciones de expresión a la vez, esta situación no debería darse.
- Al iniciar PCOPGene-Net se pintan relaciones de expresión que no deberían verse. Al iniciar PCOPGene-Net solo se tendrían que pintar las relaciones de expresión que forman el árbol de expansión mínima (Minimum Spanning Tree o MST), cosa que actualmente no se cumple.

Para solucionar estos errores será necesario corregir el código existente en MyEdgePaintFunction, donde la principal fuente de errores son comprobaciones erróneas o que no tienen en cuenta todas las opciones del pintado del grafo interactivo que el usuario de PCOPGene-Net ha activado.

En MyEdgePaintFunction no solo se tendrá que tener en cuenta el color escogido y el estado del filtro de correlación, también es necesario controlar otras opciones de pintado del grafo interactivo que están fuera de la interfaz gráfica de coloreado de relaciones de expresión no

lineales. Las opciones a controlar son: si el usuario ha desactivado el pintado del MST, si ha activado la opción de degradar el color escogido según la correlación de cada arista y si ha activado la opción de forzar el pintado de aristas aunque no formen parte del MST.

Para el pintado del peso de las aristas, que corresponde al grado de correlación de las relaciones de expresión, PCOPGene-Net tiene implementada la función MyEdgeStringer. MyEdgeStringer implementa la función abstracta EdgeStringer que proporciona la librería JUNG para el pintado de los pesos de las aristas.

Si la opción de pintar los grados de correlación de las relaciones de expresión está activada se tendrán que mostrar dichos grados solo para aquellas relaciones de expresión que se estén pintando en ese momento. Para ello será necesario aplicar los mismos cambios en el control de pintado a MyEdgeStringer que se apliquen a MyEdgePaintFunction ya que solo se ha de mostrar el peso de las aristas que se estén pintando.

Para el problema concreto del pintado por duplicado de relaciones de expresión no lineales y sus correspondientes grados de correlación, he visto que es debido a que las relaciones de expresión entre genes son tratadas de manera independiente. En la inicialización del grafo interactivo el aplicativo PCOPGene-Net puede insertar a la vez tres aristas de la relación de expresión entre dos genes: la que forma parte del MST, la que corresponde a un grado de correlación menor a 0.05 y la que corresponde a un grado de correlación entre 0.05 y 0.08.

No existe actualmente ninguna estructura de control que permita saber qué aristas existen para cada par de genes que estén relacionados. Si existiera esta estructura de control, antes de pintar la relación de expresión entre los dos genes, se podría comprobar cuál de las existentes es la que realmente se ha de pintar.

Así que es necesario implementar una estructura de control donde dados dos genes podamos saber qué aristas existen entre los dos genes. Después de investigar encontré la colección Java del paquete Common-Collections de Apache, MultiKeyMap, que amplía la típica relación clave-valor a (clave1, clave2)-valor.

En este caso las 2 claves serán los identificadores de los vértices que representan a los genes y el valor será una clase que contenga los 3 tipos de aristas: la que representa a la relación de expresión presente en el MST, la que cumple el filtro de correlación < 0.05 y la que cumple el filtro de correlación entre 0.05 y 0.08. Aunque el uso de esta estructura implicará más consumo de memoria, aportará el control necesario para el pintado correcto de las relaciones de expresión.

3.4 Mejora de rendimiento y uso de recursos de PCOPGene-Net

PCOPGene-Net trata con miles de genes para cada microarray y por lo tanto con miles de relaciones de expresión, representados en el grafo interactivo por miles de vértices y un mayor número de aristas. El pintado que proporciona JUNG se ejecuta para cada uno de los genes representados por los vértices y cada una de las relaciones de expresión representadas por las aristas. Esto implica que todas las comprobaciones y cálculos que se realizan dentro de las funciones de pintado, que se ejecutan continuamente, se han de multiplicar por la cantidad de genes y relaciones de expresión que contenga la microarray.

La consecuencia directa es que si la codificación no es suficientemente óptima el aplicativo web PCOPGene-Net consume muchos recursos de sistema reduciendo la interactividad y la correcta ejecución del mismo.

Las optimizaciones que se pueden implementar vienen limitadas en gran parte por las estructuras de datos que se utilizan para guardar la información de los elementos del grafo interactivo y los resultados del preproceso con los que se inicializa dicho grafo. Si dichas estructuras de datos se utilizan a lo largo de todo el aplicativo web la implantación de una optimización que afecte al uso de dicha estructura será difícil o no factible ya que requerirían recodificar la implementación de demasiadas partes del aplicativo.

Así que las optimizaciones que realmente puedo aplicar son las que afecten mayoritariamente a las partes de PCOPGene-Net de pintado de los vértices que representan los genes y pintado de relaciones de expresión.

3.4.1 Eliminación del uso de pares de cadenas de tipo String que actúan como listas relacionadas por la posición de sus elementos.

Me he encontrado que PCOPGene-Net hace un uso intensivo de pares de cadenas de tipo String que actúan como listas relacionadas por la posición de sus elementos para relacionar elementos de la microarray con elementos internos del grafo interactivo. Principalmente me he encontrado dos cadenas donde la primera contiene los identificadores de los genes y la otra los identificadores de vértices que JUNG usa internamente. Actualmente para saber que gen corresponde a un vértice del grafo interactivo se busca mediante el uso de StringTokenizer que posición ocupaba el identificador del vértice dentro de la segunda cadena y después se buscaba el identificador de gen que ocupaba esa posición en la primera cadena.

Esta implementación es muy cara en términos de uso de CPU y de creación de variables temporales que ocupan memoria. Como la traducción gen-vértice y vértice-gen se hace continuamente durante la ejecución de PCOPGene-Net la búsqueda de una alternativa más eficiente es muy necesaria.

Como solución más eficiente me planteo usar estructuras de tipo Map que permiten relacionar dos valores y que proporcionan herramientas de búsqueda y obtención de valores óptimas. Tendré que modificar las funciones de inicialización de PCOPGene-Net pero no supone ningún problema añadido.

3.4.2 Eliminación de bucles que usan StringTokenizer

Para aquellas cadenas String que actúan como listas y que por su uso extendido en multitud de partes del aplicativo web PCOPGene-Net no puedo substituir por estructuras de tipo Map, me planteo al menos optimizar las funciones que ya están definidas para buscar elementos dentro de estas cadenas.

Dichas funciones hacen uso de StringTokenizer, que es costoso en términos de uso de CPU y de consumo de memoria por la creación de variables String temporales. Para reducir estos costes computacionales he encontrado que el método split de la clase String permite separar los elementos de una cadena que estén separados por un delimitador concreto.

Además retorna un Array de Strings que es una estructura más sencilla de usar y más fácilmente eliminable por el Garbage Collector de Java que multitud de Strings independientes.

Entonces modificaré las funciones de búsqueda de elementos de las cadenas String que se utilizan como listas para evitar el uso de StringTokenizer y que pasen a utilizar el método split de la clase String.

3.4.3 Declaración de variables estáticas en el proceso de coloreado.

Teniendo en cuenta que:

- Los procesos de coloreado de vértices y aristas proporcionados por JUNG se ejecutan de manera secuencial, vértice a vértice y arista a arista.
- Dependiendo de las opciones escogidas por el usuario en PCOPGene-Net se van creando nuevas instancias de las clases encargadas de los procesos de coloreado pero siempre existiendo un único objeto de cada clase, siempre el mismo.
- Las variables estáticas de una clase son alojadas en memoria una vez por objeto, no una vez por instancia.

Como combinación de estos tres hechos me he propuesto definir como variables estáticas las variables miembro de las clases que implementan el proceso de coloreado de vértices y aristas del grafo interactivo, así evito definir continuamente variables temporales ahorrando espacio en memoria y trabajo al Garbage Collector de Java.

3.5 Planificación inicial y tiempo real de las fases

A continuación se pueden ver las diferencias entre la planificación que se hizo en el momento de entregar el informe previo del proyecto y el tiempo real que se ha necesitado para cumplir los objetivos.

La mayor desviación se ha producido en la mejora de rendimiento de la aplicación PCOPGene-Net. La desviación de tiempo es debida a que los cambios que se realizaban afectaban de manera transversal a muchas partes y funcionalidades ya implementadas del applet PCOPGene-Net.

	Planificación previa	Tiempo real
Octubre 2010 Noviembre 2010 Diciembre 2010 Enero 2011 Febrero 2011 Febrero 2011 Marzo 2011 Abril 2011 Mayo 2011	Análisis del preproceso y de la obtención de los parámetros que se usaban en el programa de agrupación en clústeres ya existente. Generación y tratamiento de las imágenes. Creación del menú para asociar un color a cada clúster de genes por tipo de relaciones. Implementación del coloreado de genes que pertenecen a cada clúster. Corrección del menú de coloreado de los tipos de relaciones de expresión no lineales y del proceso de coloreado de aristas que representan las relaciones de expresión. Optimizaciones sobre PCOPGene-Net. Optimizaciones sobre PCOPGene-Net.	Modificación del preproceso de clustering de genes por los tipos de relaciones de expresión de cada gen. Tratamiento de las imágenes. Creación del menú para asociar un color a cada clúster de genes por tipo de relaciones. Modificación del proceso de coloreado de vértices del grafo interactivo para colorear los genes de cada clúster. Corrección del menú de coloreado de las relaciones de expresión no lineales. Corrección y personalización del proceso de coloreado de aristas del grafo interactivo. Corrección y personalización del proceso de coloreado de aristas del grafo interactivo para colorear los diferentes tipos de relaciones de expresión no lineal. Optimizaciones sobre PCOPGene-Net. Optimizaciones sobre PCOPGene-Net.

4. Resultados

A continuación presento los resultados obtenidos al aplicar las estrategias descritas en las diferentes fases del apartado anterior.

Gran parte de estos resultados solo pueden ser analizados de manera visual ya que son las imágenes generadas por el proceso de clustering de genes por los tipos de relaciones de expresión de cada gen, las interfaces gráficas de coloreado de genes que pertenecen a cada clúster y coloreado de los tipos de relaciones de expresión no lineal y el pintado del grafo interactivo del applet PCOPGene-Net tras hacer uso de dichas interfaces gráficas.

4.1 Resultados de obtención de gráfica que muestre la relación entre los clústeres y los tipos de relación de expresión para mostrarla en PCOPGene-Net:

4.1.1 Modificación del preproceso de clustering

He construido las llamadas a CLUTO por consola de comandos que obtienen los mismos resultados del preproceso de agrupación de genes en clústeres por los tipos de relaciones de expresión de cada gen de la microarray. Las llamadas a CLUTO que he construido para datos normalizados por genes, datos normalizados por clases y datos normalizados por genes y clases son respectivamente:

./vcluster -clmethod=rbr -crfun=i2 -cstype=best -rowmodel=none -colmode=IDF -colprune=1.0 -ntrials=10 -niter=10 -seed=985794 -plotformat=gif -plotclusters=datos_finales_normGen.gif -clabel=clasesi datos_finales_normGen.mat numClusters

./vcluster -clmethod=rbr -crfun=i2 -cstype=best -rowmodel=none -colmode=IDF -colprune=1.0 -ntrials=10 -niter=10 -seed=985794 -plotformat=gif -plotclusters=datos_finales_normClass.gif -clabel=clasesi datos_finales_normClass.mat numClusters

./vcluster -clmethod=rbr -crfun=i2 -cstype=best -rowmodel=none -colmode=IDF -colprune=1.0 -ntrials=10 -niter=10 -seed=985794 -plotformat=gif -plotclusters=datos_finales_normClass_normGen.gif -clabel=clasesi datos_finales_normClass_normGen.mat numClusters

Donde los últimos parámetros, numClusters, son el número de clústeres proporcionado por el preproceso que existía anteriormente, Gencluster.

Los resultados obtenidos después de la ejecución de la llamada a CLUTO por consola de comandos son los mismos que los obtenidos mediante el preproceso de clustering existente anteriormente. Para comprobarlo he comparado los ficheros de resultados del proceso de clustering existente en la parte de preproceso de PCOPGene-Net con los generados por la llamada a CLUTO por consola de comandos.

Al comparar dichos ficheros se ha visto la misma agrupación de los genes en clústeres con la única diferencia de los identificadores de clúster en uno y otro proceso. Esta diferencia no es importante ya que obedece simplemente a una diferencia de nomenclatura de los clústeres debida a que en la llamada a CLUTO por consola de comandos los clústeres obtienen un identificador en orden creciente según su relación ISIM-ESIM. Esto es, que los clústeres cuyos elementos sean muy parecidos entre ellos y muy diferentes respecto a los elementos de los demás clústeres obtienen un identificador menor.

Como resultado adicional a los archivos de resultado del clustering, con la llamada a CLUTO por consola de comandos se ha obtenido la gráfica que muestra la relación entre los clústeres de genes y los tipos de relación de expresión no lineal, que era el objetivo principal de esta fase.

Se puede observar un ejemplo en la figura 5, que es la gráfica que muestra la relación entre los clústeres de genes y los tipos de relación de expresión que definen cada clúster mediante la llamada a CLUTO por consola de comandos. En dicha figura en el eje horizontal se pueden apreciar los diferentes tipos de expresión no lineales existentes para la microarray analizada. En eje vertical se observan los diferentes clústeres creados. La intensidad del color es el grado de correlación y el alto de las filas es el número de elementos que componen cada clúster.

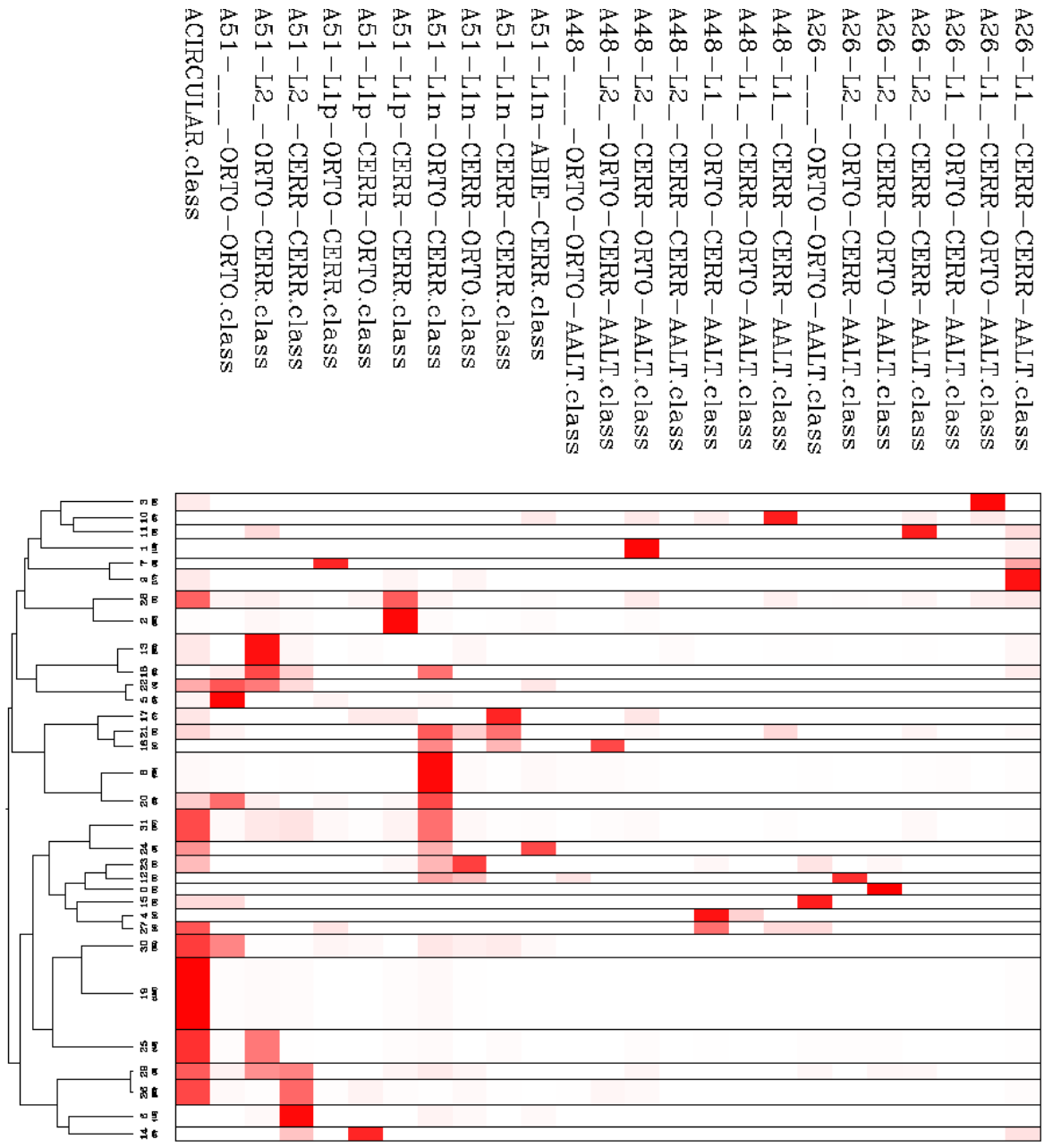


Figura 5. Representación gráfica que muestra la dependencia entre los clústeres y los tipos de relación de expresión no lineal. En el eje horizontal se pueden apreciar los diferentes tipos de expresión no lineales encontrados en la microarray analizada. En el eje vertical se observan los diferentes clústeres creados. La intensidad del color es el grado de correlación y el alto de las filas es el número de elementos que componen cada clúster.

4.1.2 Recortado de imágenes

Una vez obtenida la imagen que representa el resultado del proceso de agrupación de genes en clústeres según su relación de expresión con el resto de genes de la microarray, se ha tratado mediante las operaciones descritas en el apartado 3.1.2. Como resultado de dicho procesado de imagen obtenemos las siguientes imágenes.

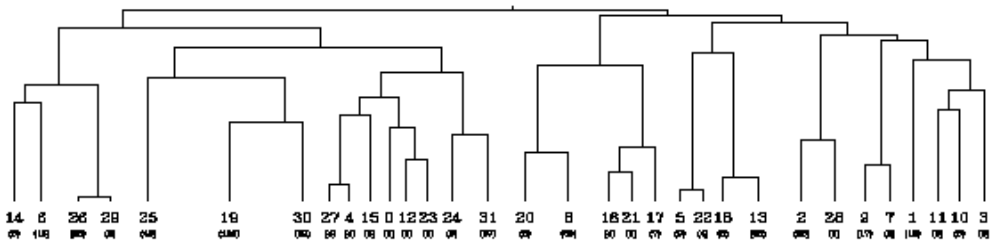


Figura 6. Árbol jerárquico de los clústeres de genes por los tipos de relaciones de expresión de cada gen calculados. Esta imagen es uno de los resultados del recorte de la figura 5 mediante el procesado de imagen descrito en el apartado 3.1.2

En la figura 6 se puede ver el árbol jerárquico de clústeres obtenido después del recortado aplicado en el procesado de imagen descrito en el apartado 3.1.2. Esta es una de las dos imágenes que se mostrarán posteriormente en el menú de coloreado de los genes de cada clúster.

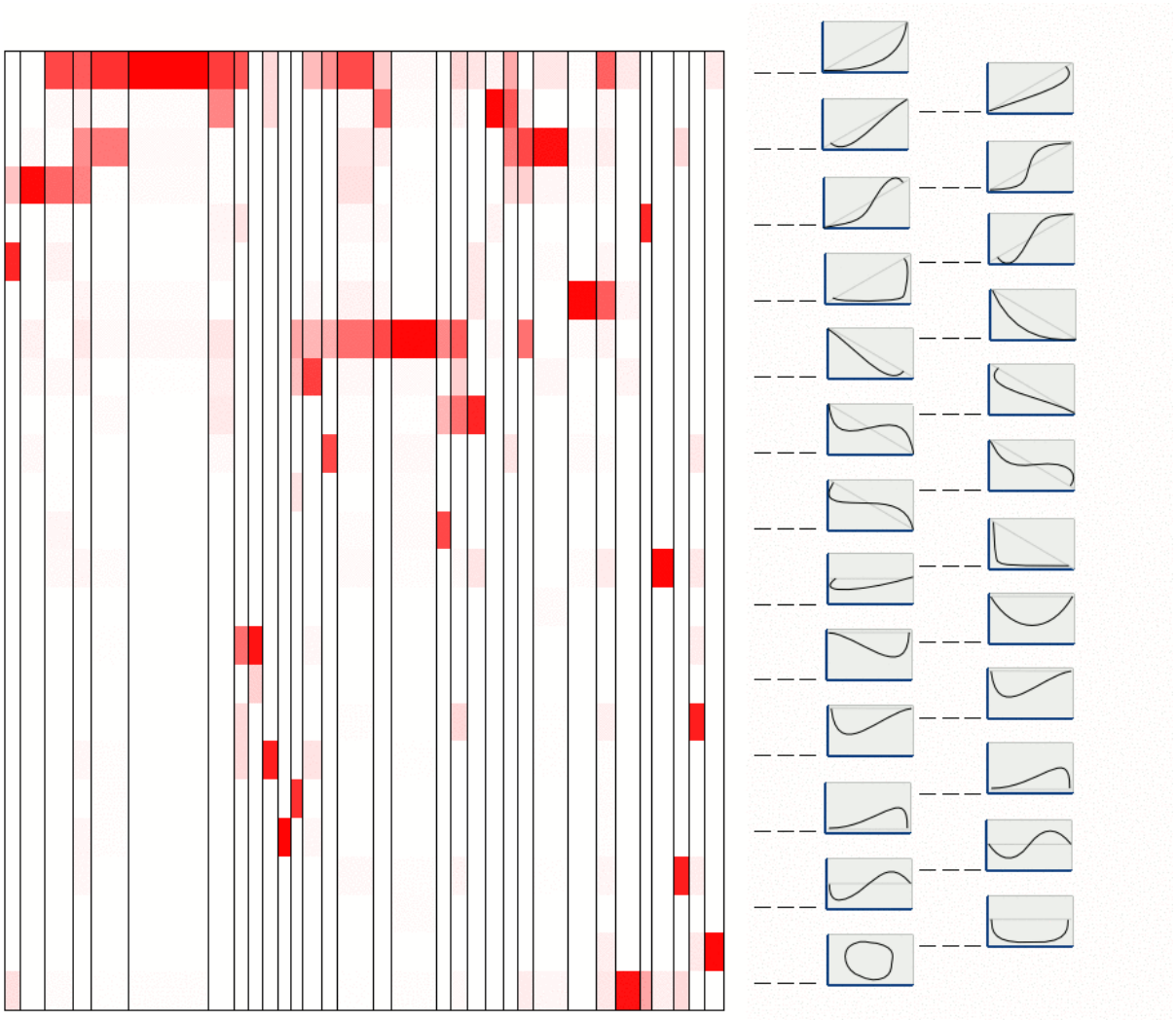


Figura 7. Composición de la de la matriz de dependencia entre los clústeres y los tipos de relación que los definen, junto con una de las leyendas de tipos de relaciones disponibles en el servidor. La matriz gráfica es una de las partes de la imagen generada en el proceso de clustering. Esta composición se ha conseguido mediante el procesado de imagen descrito en el apartado 3.1.2. Solo se puede asignar una leyenda a una matriz, si esta leyenda se corresponde con los tipos de relaciones encontrados en la microarray de la que se obtuvo la matriz. De no ser así, se mostrarían los tipos de curva con un literal obtenido directamente de CLUTO junto a la matriz.

En la figura 7 se puede ver la composición de la matriz de dependencia entre los clústeres y los tipos de relación que los definen con la leyenda disponible en el servidor para ese listado de tipos de relación de expresión. La composición la ha realizado el procesado automático de imágenes descrito en el apartado 3.1.2. Esta imagen se usará en el menú de coloreado de los genes de cada clúster.

4.2 Resultados de la identificación sobre el grafo interactivo de los genes que forman cada clúster en PCOPGene-Net

Los resultados de identificación sobre el grafo interactivo de los genes que forman cada clúster en PCOPGene-Net son dos: el menú en PCOPGene-Net que permite colorear los genes de cada clúster y por otra parte el coloreado de los genes del grafo interactivo de PCOPGene-Net una vez usado el menú de coloreado de genes de cada clúster.

4.2.1 Menú de coloreado de genes de cada clúster en PCOPGene-Net

En la figura 9 se puede ver el menú creado para la identificación sobre el grafo interactivo de los genes que forman cada clúster en PCOPGene-Net. En esta interfaz gráfica se puede observar la tabla dinámica interactiva en la parte superior, debajo de esta tabla están las dos imágenes que representan la dependencia de los clústeres de genes y los tipos de relación de expresión. En la parte superior derecha están situados los botones que permiten elegir el tipo de clustering. En la parte superior izquierda se encuentran situados los botones de reseteo de la selección de colores y de cierre de la interfaz gráfica. También se puede ver el selector de color que se abre una vez clicado el identificador de clúster en la tabla dinámica.

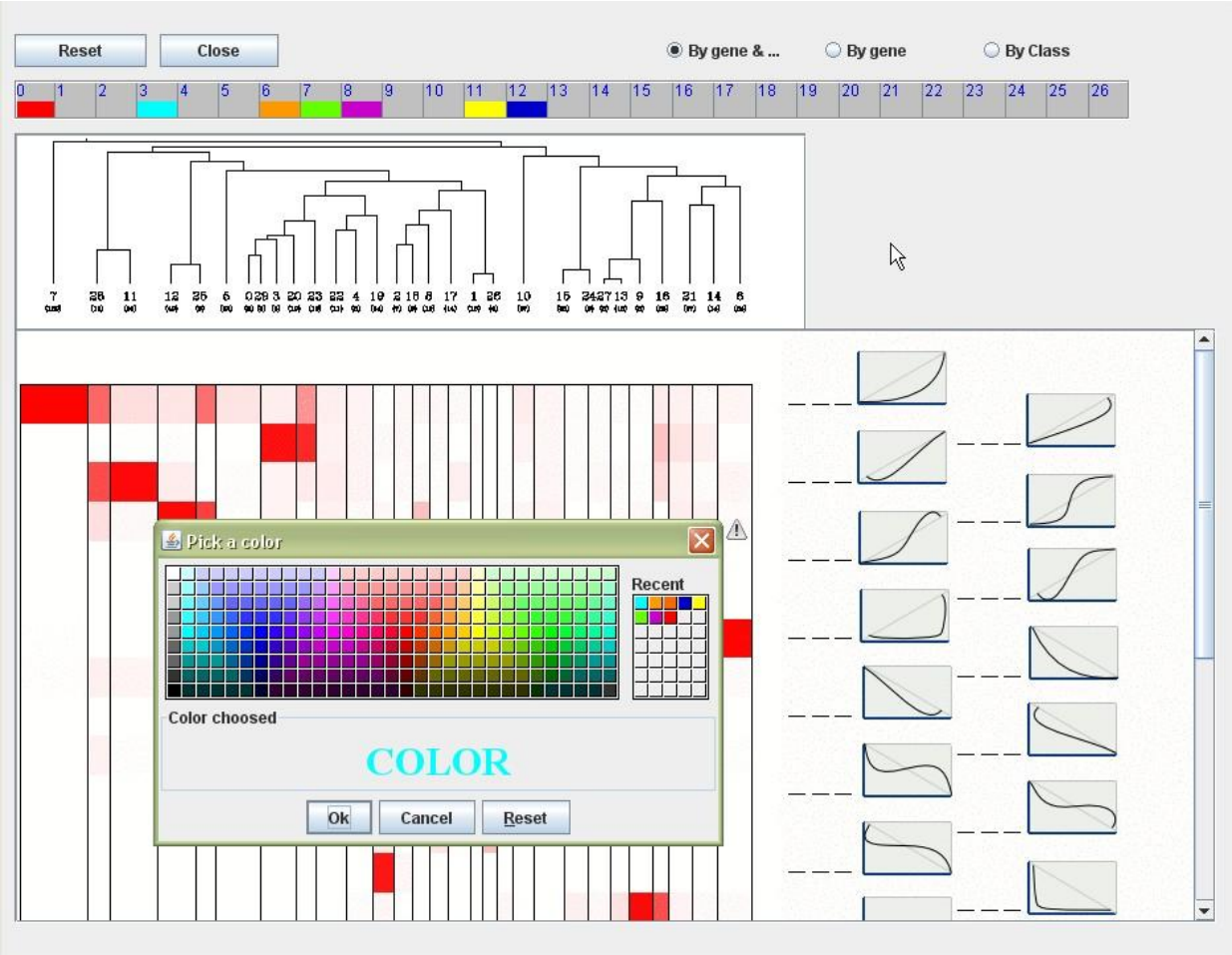


Figura 9. Menú creado para el coloreado de los genes de cada clúster. Se puede observar ciertos colores elegidos ya asociados a algunos clústeres en la tabla dinámica. También se aprecia el selector de colores que se ha implementado y que se abre al clicar una posición (un clúster) de la tabla dinámica.

4.2.2 Coloreado de los genes de cada clúster sobre el grafo interactivo

En la figura 10 se puede apreciar cómo se colorean los genes del gráfico interactivo con los colores escogidos en el menú de coloreado de los genes de cada clúster. Los genes de los clústeres que no tienen un color asignado se pintan de color gris claro. Este coloreado de los genes se realiza una vez activada la casilla correspondiente etiquetada “Show clústeres”. A su izquierda se encuentra el botón que abre el menú que se muestra en la figura 9. Dicho botón está etiquetado como “Gene Cluster”.

En caso que “Show Clústeres” esté desactivado todos los genes del microarray se pintan de color rojo, y de color amarillo aquellos que estén seleccionados.

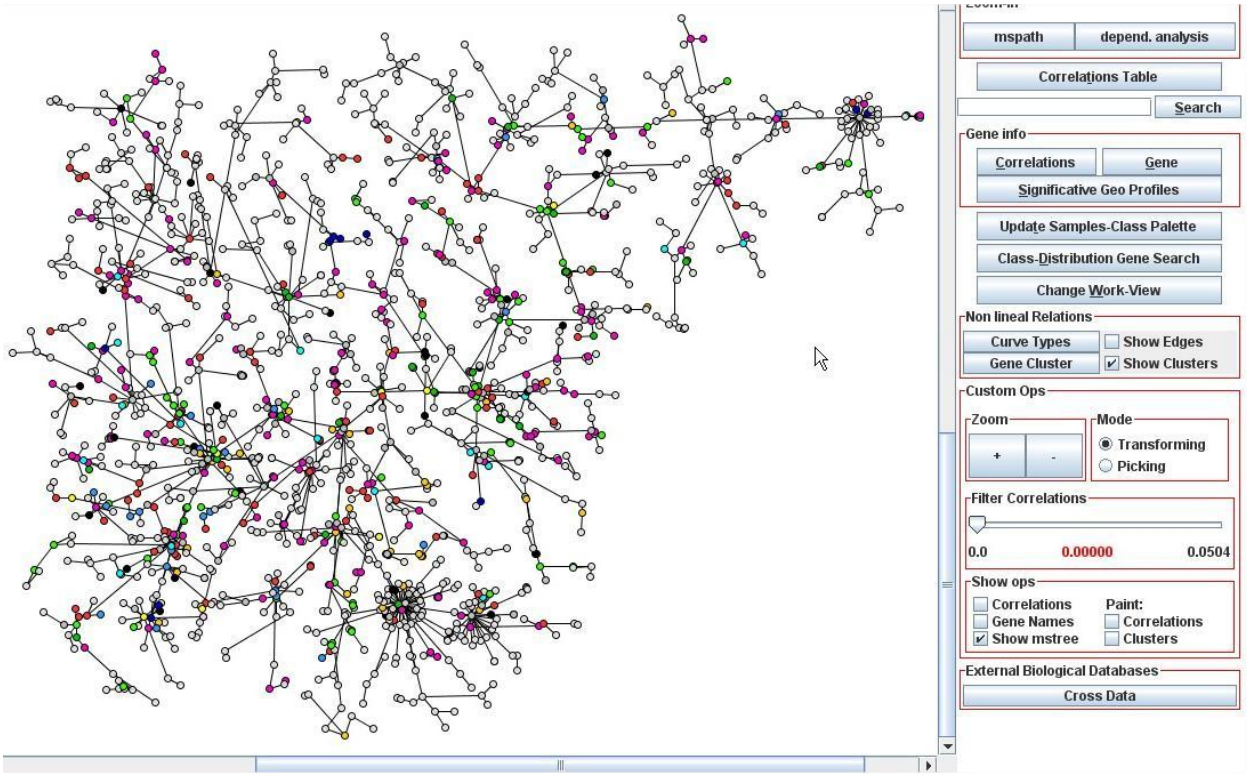


Figura 10. Coloreado de los genes por clúster sobre el grafo interactivo de PCOPGene-Net. Los genes se pintan del color que tiene asociado el clúster al que pertenecen. Cada color indicará que esos genes mantienen un tipo de relación específica con el resto de genes en (aunque esas relaciones no se muestren en el grafo). Los genes cuyo clúster no tiene asociado ningún color se pintan de color gris claro.

4.3 Resultado de mejorar la identificación de relaciones de expresión no lineales

Como resultado del proceso de mejora de la identificación de relaciones de expresión no lineales he obtenido un menú de coloreado de las relaciones de expresión no lineales con capacidad de recordar la última selección de colores. El menú incorpora el botón “Clear All” que ahora realiza un reseteo de las selecciones de los colores hechas por el usuario. En la figura 11 se puede observar el aspecto del menú de coloreado de las relaciones de expresión no lineales.

Figura 11. Menú de coloreado de los tipos de relaciones de expresión no lineales. En el menú se muestran los diferentes tipos de relaciones de expresión no lineales representados por una imagen de la curva que describen y acompañados por un desplegable donde escoger el color. También se pueden apreciar la casilla de filtro de correlación y los botones de reseteo de la asociación de colores y cierre del menú.

Una vez seleccionados los colores para los tipos de relaciones de expresión no lineales que se quieren identificar, las aristas del grafo del applet PCOPGene-Net se pintan tal como muestra la figura 12.

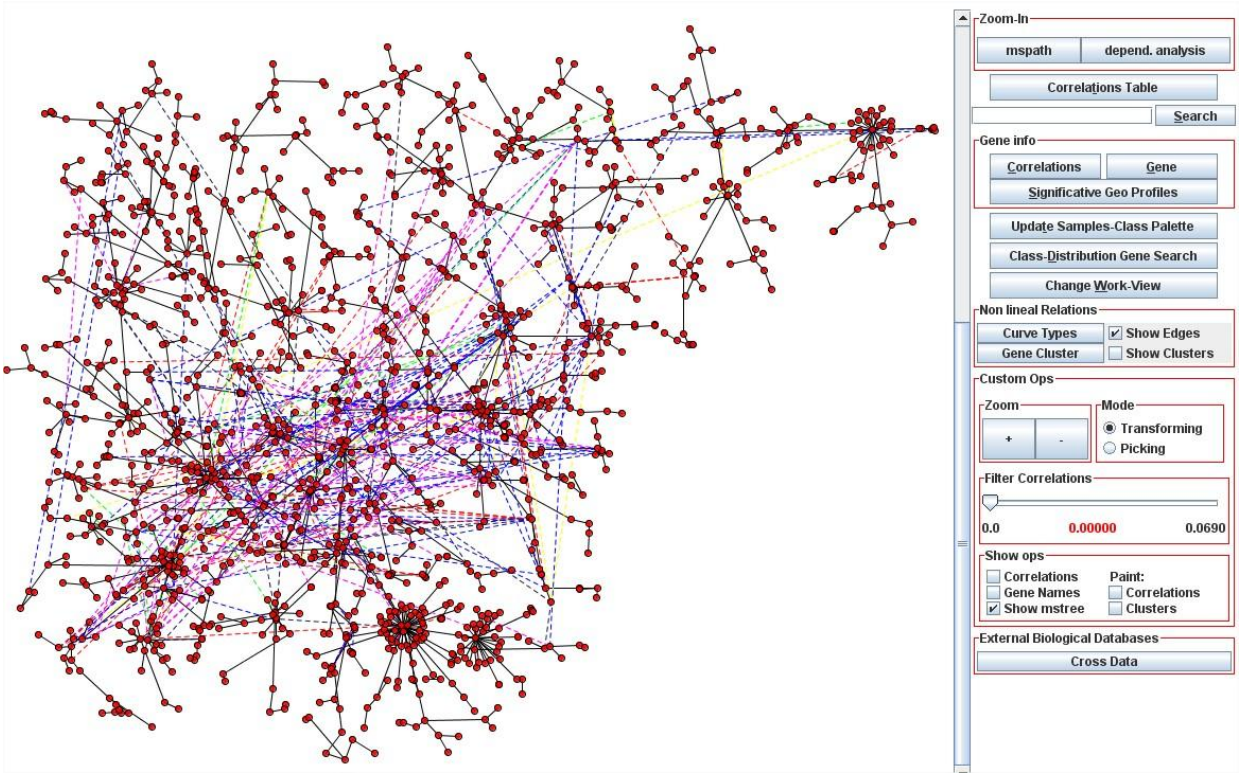


Figura 12. Coloreado de las aristas por tipo de relación de expresión no lineal sobre el grafo interactivo de PCOPGene-Net. Las aristas se pintan con los colores seleccionados por el usuario en el menú de coloreado de los tipos de relaciones de expresión no lineales. Cada color indicará que los genes mantienen un tipo de relación específica entre ellos. Se puede observar cómo al estar activada la casilla “Show Edges”, las aristas que representan tipo de relación de expresión con un color asignado se pintan también aunque no formen parte del MST. Estas aristas coloreadas que no forman parte del MST se pintarán con líneas discontinuas.

Como se puede ver en la figura 13, he solucionado el error de pintado por duplicado de la relación de expresión no lineal entre dos genes y el pintado por duplicado de los correspondientes grados de correlación. También he aumentado el tamaño de la fuente de los grados de correlación para que se puedan leer más fácilmente.

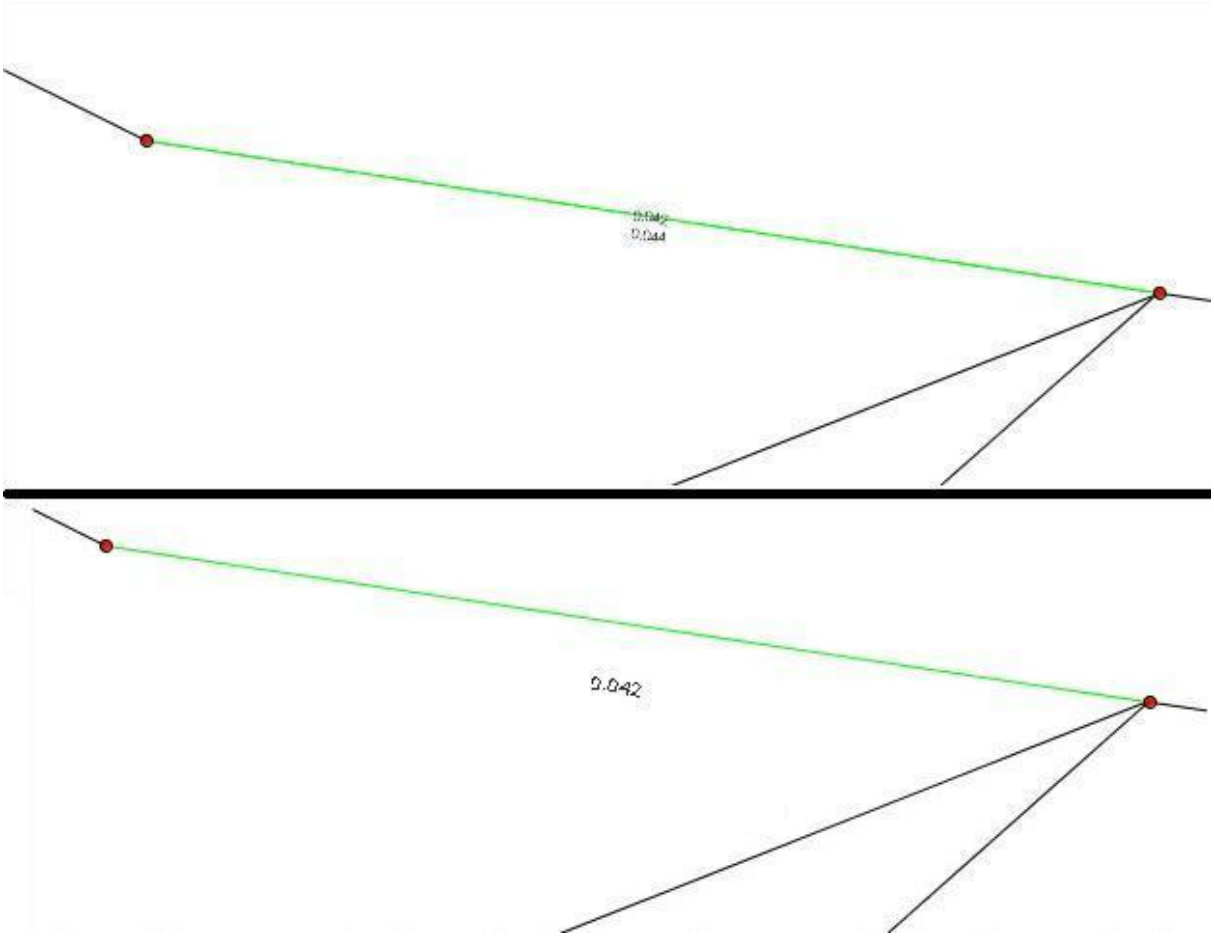


Figura 13. En la imagen superior se puede observar el error de duplicación de aristas y de grados de correlación que se producía en el applet PCOPGene-Net. En la imagen inferior se observa cómo tras mis modificaciones se soluciona el error de duplicación y se ha aumentado el tamaño de la fuente con la que se muestran los grados de correlación.

Las primeras aristas que muestra el applet PCOPGene-Net nada más iniciarse, son las que forman el árbol de expansión mínima o MST. Existía un error de pintado en que una vez iniciado PCOPGene-Net, si se desactivaba el pintado del MST continuaban pintándose algunas aristas que no deberían visualizarse al no estar activa ninguna opción de pintado de aristas. Como se muestra en la figura 14 he solucionado ese error de pintado.

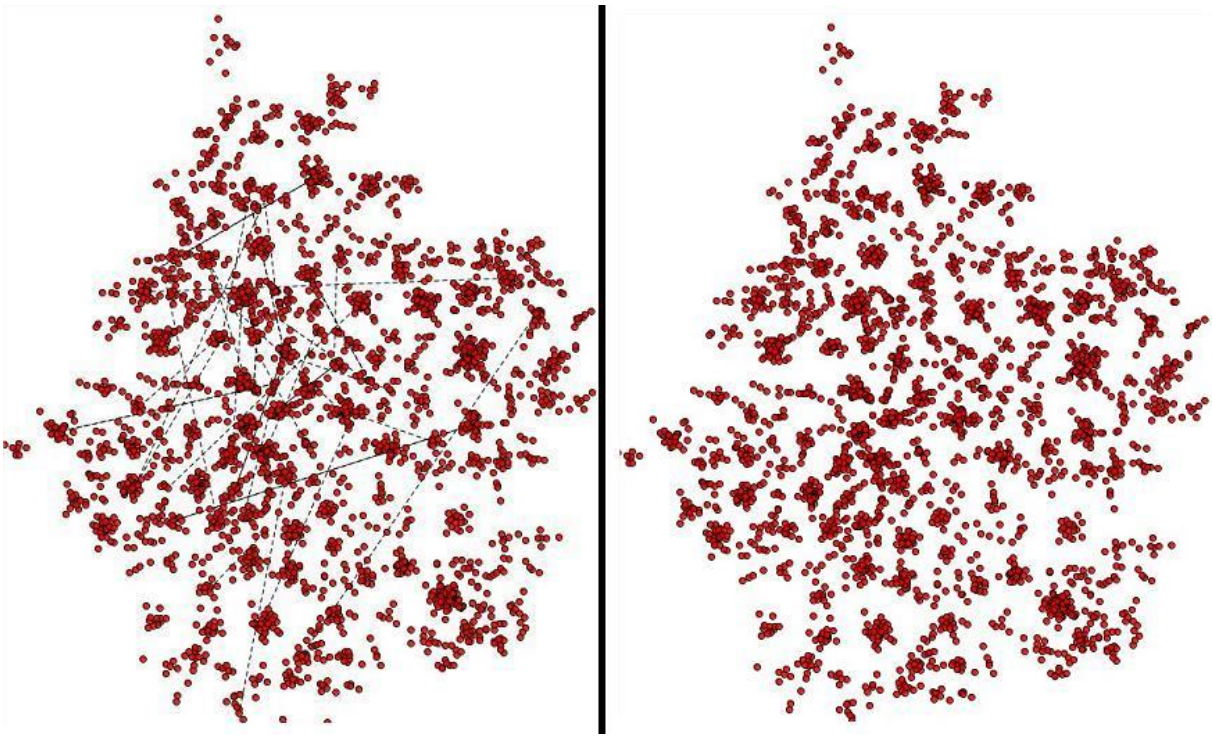


Figura 14. En la imagen de la izquierda se observa el error de pintado de aristas que se producía cuando se desactivaba el pintado de las aristas del MST, se mostraban aristas que no tenían que visualizarse en ese momento al no estar activa ninguna opción de pintado de aristas. En la imagen de la derecha se observa que tras las correcciones que he aplicado ya no se visualizan dichas aristas.

4.4 Resultado de mejora de rendimiento y uso de recursos de PCOPGene-Net

Los resultados de las optimizaciones los represento mediante una serie de mediciones en la versión original del applet y en la nueva versión: uso de memoria que consume el applet, tiempo de carga del applet, uso de CPU.

Estas mediciones se han realizado con la microarray m17 que contiene 1416 genes y un ordenador con procesador Intel Core2 Duo T7500 de 2,20 GHz y 2 GB de RAM corriendo sobre Windows XP SP3.

Consumo de memoria:

Versión de PCOPGene-Net anterior a mis modificaciones:

- Al cargar pero sin realizar ninguna acción sobre el applet: 32 MB. No aumenta con el tiempo.
- Al pintar tipos de relaciones no lineales y grados de correlación: 67 MB aumentando este valor continuamente de manera proporcional al tiempo estabilizándose sobre los 150 MB.

Versión de PCOPGene-Net con mis modificaciones:

- Al cargar pero sin realizar ninguna acción: 72 MB. No aumenta con el tiempo.
- Al pintar tipos de relaciones no lineales y grados de correlación: 76 MB. No aumenta con el tiempo.
- Al pintar aristas, grados de correlación y clústeres de genes: 82 MB. No aumenta con el tiempo.

Consumo de CPU:

Versión de PCOPGene-Net anterior a mis modificaciones:

- Al cargar pero sin realizar ninguna acción: 48% uso de CPU
- Al pintar aristas y grados de correlación: 50 % uso de CPU

Versión de PCOPGene-Net con mis modificaciones:

- Al cargar pero sin realizar ninguna acción: 48% uso de CPU
- Al pintar aristas y grados de correlación: 50% uso de CPU
- Al pintar aristas y grados de correlación: 51% uso de CPU

Se puede observar que no ha sido posible mejorar el uso de CPU por la incorporación de la estructura de control de aristas entre genes para evitar duplicaciones

Tiempo de carga del applet:

Con imágenes del servidor que usa el applet ya descargadas en el ordenador que ejecuta el applet. Este tiempo se ha medido calculando la diferencia de el valor retornado por la llamada System.currentTimeMillis() al inicio y al final del proceso de carga del applet.

Versión de PCOPGene-Net anterior a mis modificaciones: 25093 milisegundos
Versión de PCOPGene-Net con mis modificaciones: 24678 milisegundos

Las optimizaciones por la substitución de cadenas String que se recorrían continuamente con bucles con String Tokenizer y generaban miles de variables temporales de tipo String, por estructuras Map son las responsables de impedir que el consumo de memoria aumente proporcionalmente al tiempo. Esta optimización está explicada en el apartado 3.4.1.

La incorporación de las estructuras Map y de estructuras de control que guardan todas las aristas existentes entre cada par de genes ha provocado que una vez cargado el applet ocupe más memoria. A cambio se ha obtenido un pintado correcto de las aristas y no perder la interactividad con el applet al no consumir demasiada memoria.

Se ha conseguido reducir el uso de CPU eliminando los bucles de recorrido de las cadenas String que actuaban como listas de vértices y genes. Pero al añadir las comprobaciones a la estructura de control del pintado de aristas se obtiene un uso de CPU parecido al de la versión anterior del applet.

La reducción de tiempo de carga es poco significativa. Esto es debido a que en el proceso de inicialización del applet se ha substituido la obtención con bucles de StringTokenizer de los elementos que forman las cadenas String, por el método split de la clase String explicado en el apartado 3.4.2.

5. Informe técnico

5.1 Estructura de archivos

Los archivos de los programas del preproceso asociado a PCOPGene-Net, archivos que se encuentran en el servidor del IBB, son los siguientes:

Carpeta	Archivo / Subcarpeta	Descripción
/fullcorrelations	/compile	Carpeta donde se encuentran los archivos de código fuente de los programas aquí descritos
	lanzadora	Programa lanzadora que va llamando a todos los programas y scripts del preproceso asociado a PCOPGene-Net
	gencluster	Programa existente anteriormente que realiza la agrupación de los genes en clústeres por el tipo de relaciones de expresión con el resto de genes y calcula el número de clústeres a formar.
	generaim	Script creado por mí. Llama a invertclass y generagifs, luego realiza el procesado de las imágenes generadas.
	invertclass	Script creado por mí. Realiza la inversión de orden de los tipos de relaciones de expresión que luego se utilizarán como leyenda de las imágenes generadas.
	generagifs	Script creado por mí. Realizada las llamadas a vcluster descritas en el apartado 4.1.1. Una llamada para cada normalización de los datos, para generar las imágenes de cada tipo de clustering.
	vcluster	Programa stand-alone de CLUTO. Es el programa que permite generar las imágenes del clustering.

En el servidor del IBB están los archivos de datos que PCOPGene-Net utiliza como datos de entrada, son los siguientes:

Carpeta	Archivo / Subcarpeta	Descripción
/microarray	/mXX	Carpeta que contiene los datos de los genes y las relaciones de expresión entre ellos de la microarray XX. Yo he utilizado con m17, m22, m2501, m2503 en mi proyecto
	/curveimages	Carpeta que sirve de repositorio de leyendas con las miniaturas de las relaciones no lineales.

La ejecución del applet web PCOPGene-Net se hace mediante 2 archivos en el servidor del IBB que se encuentran en:

Carpeta	Archivo / Subcarpeta	Descripción
applic/gexp/microarray	BioProject.jar	Contenedor donde está el código fuente y las clases compiladas del applet web PCOPGene-Net
	BioProject.html	Archivo HTML que se abre el usuario en su navegador web y que ejecuta el applet PCOPGene-Net llamando a BioProject.rar

5.2 Descripción y uso de los programas

5.2.1 Preproceso

Para la ejecución de los programas creados en el preproceso se ha añadido la llamada al script generaim en el programa lanzadora que va llamando en orden a los diferentes programas y scripts que forman el preproceso.

Para empezar el preproceso se ha de llamar al programa lanzadora, se le pasa como parámetros la id de la microarray y el número de genes de esta. Para el caso de la microarray m17 la llamada es la siguiente:

./lanzadora 17 1416

Dentro de lanzadora la llamada a generaim, que no tiene parámetros de entrada, es la siguiente:

```
./generaim
```

Dentro de generaim, encontramos las siguientes llamadas:

```
`./invertclass`
```

Invertclass realiza la inversión de la lista de relaciones de expresión no lineales que hay en clases.txt. Dicha lista ya ha sido generada por programas llamados con anterioridad por lanzadora en el preproceso.

Después de la ejecución de invertclass, generaim realiza la llamada al programa generagifs pasándole por parámetro el número de clústeres por gencluster para cada tipo de clustering:

```
`./generagifs $aux1 $aux2 $aux3`
```

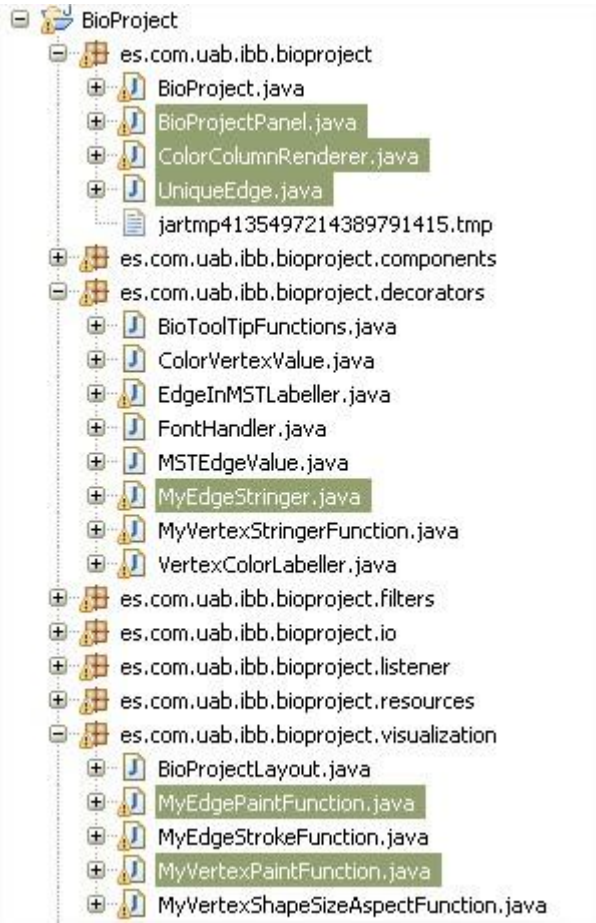
En generagifs se hacen una a una las llamadas a vcluster expuestas en el apartado 4.1.1, obteniendo así las gráficas que relacionan los clústeres de genes por las relaciones de expresión de cada gen con los diferentes tipos de relaciones de expresión.

Una vez generadas las imágenes, generaim continúa su ejecución recortando, rotando y componiendo las diferentes partes de las imágenes. Al finalizar su ejecución se obtienen las siguientes imágenes que corresponden el árbol de jerarquía de clústeres y la composición de matriz gráfica y leyenda respectivamente:

- Clustering con datos normalizados por clases:
    - datos_finales_normClass.gif y datos_finales_normClass_cluster.gif
- Clustering con datos normalizados por genes:
    - datos_finales_normGen.gif y datos_finales_normGen_cluster.gif
- Clustering con datos normalizados por clases y genes:
    - datos_finales_normClass_normGen.gif y datos_finales_normClass__normGen cluster.gif

5.2.1 Applet web PCOPGene-Net

El applet PCOPGene-Net está compuesto por los siguientes paquetes:



Las clases Java seleccionadas han sido modificadas o creadas por mí a lo largo del desarrollo. A continuación hago una pequeña descripción de cada una de ellas:

- BioProjectPanel.java: Clase Java que contiene el proceso de inicialización de PCOPGene y el código que implementa todas las interfaces gráficas de PCOPGene-Net.
- ColorColumnRenderer.java: Clase Java que permite mostrar en la tabla dinámica el color seleccionado para cada clúster en el menú de coloreado de genes de cada clúster.
- UniqueEdge: Clase Java donde se guardan la información de todas las aristas entre dos genes concretos. La nueva estructura de control de pintado de aristas se forma a base de objetos de esta clase.
- MyEdgeStringer: Clase Java que implementa la clase abstracta de pintado de pesos de aristas proporcionada por JUNG. Está implementada para pintar el grado de correlación de cada relación de expresión.
- MyEdgePaintFunction: Clase Java que implementa la clase abstracta de coloreado de aristas proporcionada por JUNG. Modificada para pintar cada relación de expresión no lineal del color que se le haya asignado en el menú de coloreado de los tipos de relaciones de expresión no lineales.
- MyVertexPaintFunction: Clase Java que implementa la clase abstracta de coloreado de vértices proporcionada por JUNG. Modificada para pintar cada gen del color que se le haya asignado en el menú de coloreado de los genes de cada clúster.

Para ejecutar el applet web PCOPGene-Net, tan solo se tiene que llamar a la URL de BioProject.html desde el navegador web. Un ejemplo de la URL a la que llamar :

http://revresearch.phpwebhosting.com/applic/gexp/microarray/BioProject.html

6. Conclusiones

A lo largo del desarrollo de este proyecto, y aplicando la metodología descrita en el apartado 3, he podido cumplir todos los objetivos que me había propuesto.

PCOPGene-Net realiza un preproceso de clustering que agrupa los genes de la microarray en clústeres según los tipos de relación de expresión de dichos genes con el resto de genes de la microarray. He conseguido adaptar exitosamente el preproceso de clustering existente para generar las imágenes que luego se utilizarán en la interfaz web cuando se analice la microarray en cuestión. El procesado de esas imágenes ha quedado finalmente adaptado con éxito para microarrays de cualquier tamaño.

He creado satisfactoriamente la interfaz web que facilita de forma intuitiva mediante un menú, escoger los colores con los que pintar los genes de cada clúster, para cada uno de los tres tipos de clustering. También he creado satisfactoriamente el selector de color que permite asociar un color a los genes de cada clúster. Este menú también está adaptado para microarrays de cualquier tamaño y cualquier número de clústeres. He habilitado el aplicativo de PCOPGene-net para el correcto coloreado de los genes en función del clúster al que pertenecen y el color asignado a dicho clúster. La personalización del método originario de JUNG para el coloreado de vértices así como el parsing de los ficheros donde se le asigna un clúster a cada gen, las he realizado con éxito y de manera óptima.

He logrado que PCOPGene-Net disponga de una interfaz gráfica para la identificación de los diferentes tipos de relaciones de expresión no lineales que, mediante un menú de coloreado de los tipos de relaciones de expresión no lineales, permite pintar correctamente cada una de las relaciones de expresión no lineal del color elegido por el usuario. He añadido a la interfaz gráfica la función de recordar la última asignación de colores a cada tipo de relación de expresión no lineal.

En la modificación y corrección del proceso de coloreado de las aristas me he encontrado con diversos problemas y errores que han requerido cambiar la lógica del coloreado y las estructuras de datos, pero la estrategia finalmente adoptada ha conseguido un correcto y eficiente coloreado de las relaciones de expresión.

En la búsqueda del mejor rendimiento posible ha sido necesario aplicar numerosas optimizaciones, tanto durante la implementación de las funcionalidades añadidas como en funcionalidades ya implementadas anteriormente. Las optimizaciones aplicadas han conseguido reducir de manera drástica la cantidad de memoria que requiere el aplicativo web PCOPGene-Net en su ejecución.

Debido a la necesidad de añadir comprobaciones y cálculos nuevos en el proceso de coloreado de aristas y nodos del grafo no se ha podido reducir el uso de CPU durante la ejecución del applet PCOPGene-Net.

Como consecuencia de los diferentes problemas que han surgido durante en el coloreado de aristas y a que las optimizaciones aplicadas sobre PCOPGene-Net requerían aplicar cambios en funcionalidades anteriores de PCOPGene, he tenido que priorizar esos cambios a la posibilidad de implementar nuevas herramientas que habrían complementado las funcionalidades ya añadidas para el coloreado de aristas (relaciones de expresión).

Trabajo Futuro

Como trabajo futuro yo propongo añadir nuevas herramientas a PCOPGene-Net que permitan mostrar las relaciones de expresión entre genes de cada tipo, pero no ya de manera visual sobre el grafo interactivo, sino con un listado. Se podrían listar todas las relaciones de expresión detectadas para un tipo de curva concreto. También se podría identificar gráficamente el tipo de relación mantenida en al actual listado de genes ordenados por correlación respecto a un gen seleccionado de la microarray. Estos listados facilitarían aún más la investigación con PCOPGene-Net.

Con el propósito de reducir el uso de CPU por parte de PCOPGene-Net aplicaría algunos cambios que afectan de manera transversal al funcionamiento de PCOPGene-Net:

- Cambiaría la forma en que se guarda la información del binomio arista - tipo de relación de expresión. Actualmente se guarda como atributo de la propia arista en forma de String. Sobre el tipo de relación de expresión de cada arista se aplican multitud de comprobaciones de comparación y pertenencia, con lo que es necesario optimizar el tratamiento. Yo guardaría la información en estructuras de tipo Map que permiten un manejo mucho más óptimo. Por ejemplo, se podría tener una estructura Map para cada tipo de relación de expresión no lineal que contendría todas las aristas que representan ese tipo de relación de expresión no lineal. Esta implementación reduciría el uso de CPU al no tener que estar manejando y comprobando tantos objetos String continuamente.
- Actualmente entre dos genes podemos llegar a tener varias aristas a la vez, una por el tipo de relación de expresión para el filtro de alta correlación, otra por el tipo de relación de expresión para el filtro por baja correlación y una tercera si la arista entre los dos genes forma parte del MST. Yo recomendaría modificar la generación de aristas para que entre dos genes solo pueda existir una sola arista. Para ello se necesitaría añadir nuevas estructuras de datos que relacionen los diferentes tipos de relación de expresión o diferentes categorías de arista con cada par de genes entre los que aparezca cualquier tipo de arista.
- Para mejorar aún más el rendimiento de PCOPGene-Net, propongo recodificar las partes que no he optimizado ya del applet de PCOPGene-Net para evitar el uso de cadenas de tipo String para cargar y guardar la información asociada con los genes, las relaciones de expresión y demás información de la microarray. Guardar la información en listas del tipo String requiere recorrerlas con bucles que realizan computación innecesaria y generan variables temporales que ocupan memoria. Como alternativa propongo el uso de estructuras de tipo Map o List, dependiendo de si se necesita una relación clave-valor o una lista simple, puesto que estas no requieren recorrer sus elementos con bucles para buscar elementos o para obtener su valor. Concretamente las cadenas a optimizar son: las 29 cadenas de aristas de relaciones de expresión no lineales de bajo factor de correlación y las 29 cadenas de

alta correlación, las cadenas de listas temporales de aristas (yesIn y notIn) del proceso de inserción de aristas al grafo.

Impresiones personales

Durante el desarrollo de este proyecto la cantidad de desafíos a los que me he tenido que enfrentar creo que han resultado muy importantes en mi formación como ingeniero informático. Pese a la escasa documentación y no demasiados trabajos previos en el uso de la librería JUNG he podido plantear soluciones válidas a los problemas que han ido surgiendo durante el desarrollo de este proyecto.

La parte del desarrollo que más me ha gustado y ha resultado más gratificante ha sido la búsqueda e introducción de alternativas en el código para optimizar el rendimiento del applet. Como el objetivo de este proyecto era desarrollar interfaces para modificar el coloreado del grafo interactivo, con cada fase implementada podía observar los resultados de manera visual inmediatamente sobre el grafo. Esto resultaba especialmente motivador para continuar con el desarrollo restante ya que podía ver de manera directa el resultado de los cambios que iba añadiendo.

La parte que menos he disfrutado ha sido la adaptación del preproceso previo de PCOPGene-Net que realizaba el proceso de clustering de los genes por los tipos de relaciones de expresión de cada gen de la microarray. Consistía en utilizar la herramienta de clustering CLUTO de manera distinta a como se utilizaba en la versión anterior del preproceso para obtener los mismos resultados de clustering pero generando además una imagen que representa gráficamente estos resultados del proceso de clustering. Como ya he comentado me ha resultado mucho más atractivo trabajar en el desarrollo y optimización del applet.

El trabajo desarrollando todas estas herramientas para el análisis de microarrays ha contado con la dificultad añadida de tener que comprender los complejos conceptos teóricos de la genómica y la biología molecular. Aunque tras el esfuerzo inicial de aprendizaje y

comprensión dicha dificultad se ha convertido en reto y ha acrecentado aún más mi interés por el campo de la Biotecnología.

7. Referencias

[1] **Interlandi, Geneen.(2007)** *Small Synthetic Molecule Curbs Cancer Growth.* Focus Magazine. Harvard University.
<http://archives.focus.hms.harvard.edu/2007/020907/biochemistry.shtml>.

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

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

[4] 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.

[5] 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.

[6] http://revolutionresearch.uab.es : *Web server for online microarray analysis supported by the Institute of Biotechnology and Biomedicine of the Autonomous University of Barcelona (IBB-UAB).*
<http://revolutionresearch.uab.es>

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

[8] 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

[9] **PCOPGene-Net Tour.**
<http://revolutionresearch.uab.es/downloads/PCOPGene/PCOPGene.htm>.

[10] **JUNG -** *Java Universal Network/Graph Framework*
<http://sourceforge.net/apps/trac/jung/wiki/JUNGManual>

[11] **CLUTO** - *Software for Clustering High-Dimensional Datasets* | Karypis Lab.
<http://glaros.dtc.umn.edu/gkhome/fetch/sw/cluto/manual.pdf>.

8. Resumen

Resum

La investigació entre les relacions dels nivells d'expressió dels gens aporta molta informació sobre els processos biològics i patològics. Mitjançant la tècnica de les microarrays es possibilita la investigació de les relacions d'expressió de milers de gens a la vegada.
La finalitat d'aquest projecte es fent ús de l'aplicatiu web PCOPGene-Net, permetre la identificació dels gens per les relacions d'expressió no lineals que tenen amb la resta de gens i permetre també la identificació de les relacions d'expressió no lineals entre els gens d'una microarray.

Resumen

La investigación de las relaciones entre los niveles de expresión de los genes aporta mucha información sobre el desarrollo de los procesos biológicos y patológicos. Mediante el uso de la técnica de las microarrays se posibilita la investigación de la relaciones de expresión de miles de genes a la vez.
La finalidad de este proyecto es mediante el aplicativo web PCOPGene-Net, permitir la identificación de los genes por las relaciones de expresión no lineales que mantienen con el resto de genes y la identificación de las relaciones de expresión no lineales entre los genes de una microarray analizada.

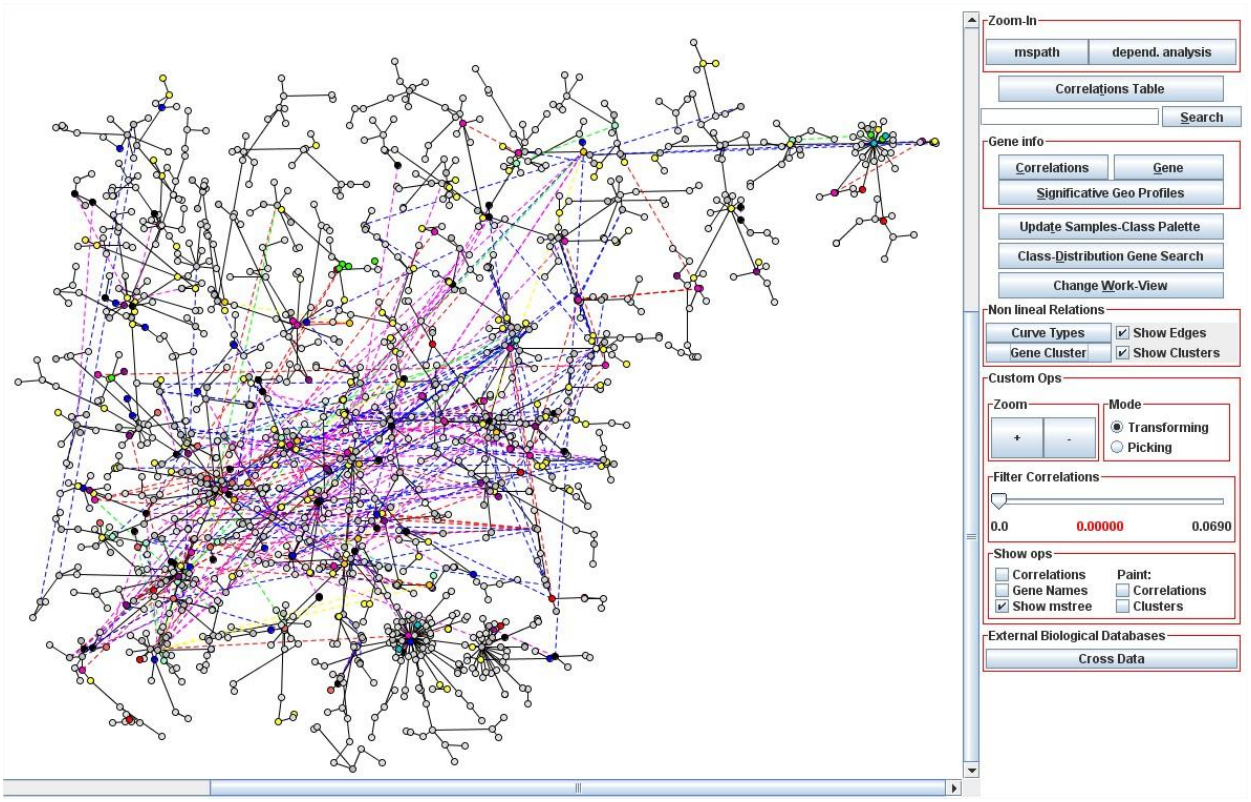
Abstract

Research on relations between gene expression levels provides a lot of information about the development of biological and pathological processes. Microarray technique allows the research on expression relations over thousands of genes simultaneously.
The aim of this project is to make possible, using the web applet PCOPGene-net, the identification of genes by their non-lineal expression relations with the rest of genes as well as the identification of non-lineal expression relations between genes of the microarray that is being analyzed by PCOPGene-Net.

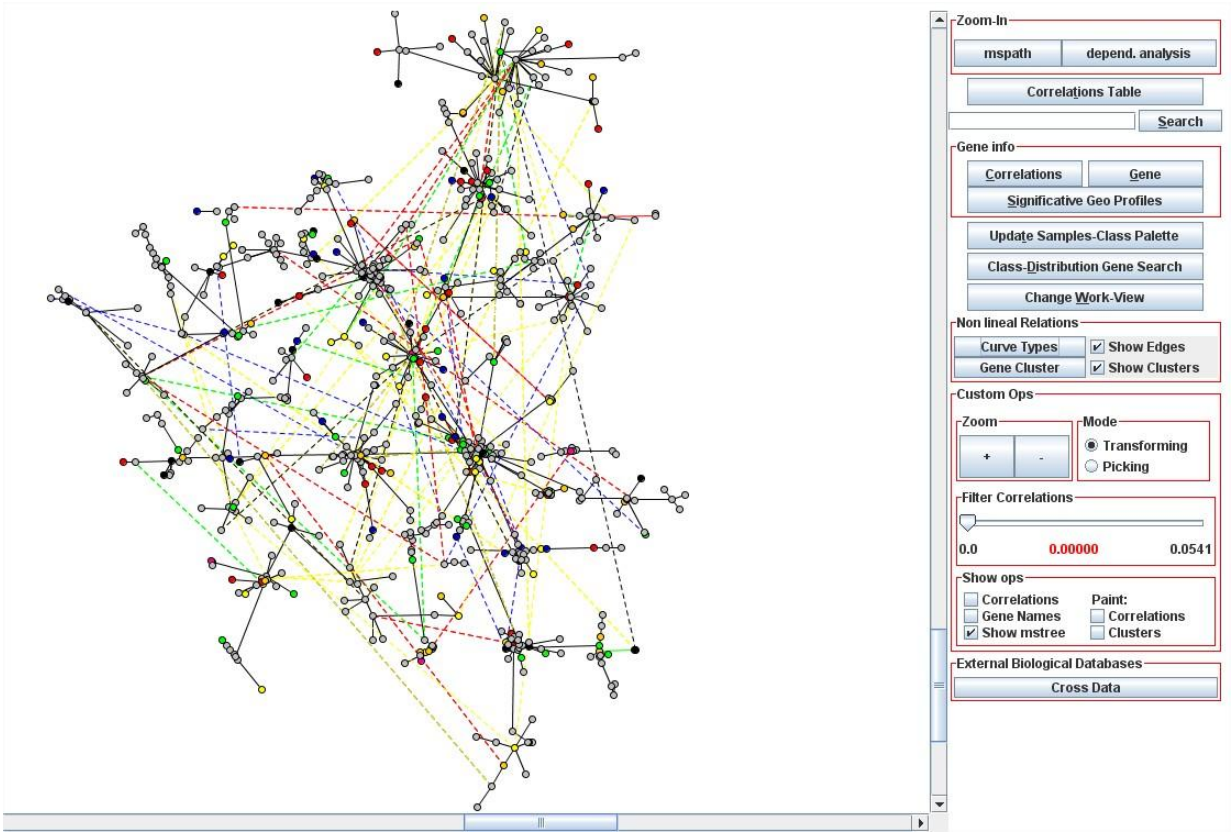
Anexos

A continuación presento más ejemplos del pintado del gráfico interactivo de PCOPGene-Net usando las interfaces gráficas desarrolladas en este proyecto para las diferentes microarrays disponibles en el servidor del IBB.

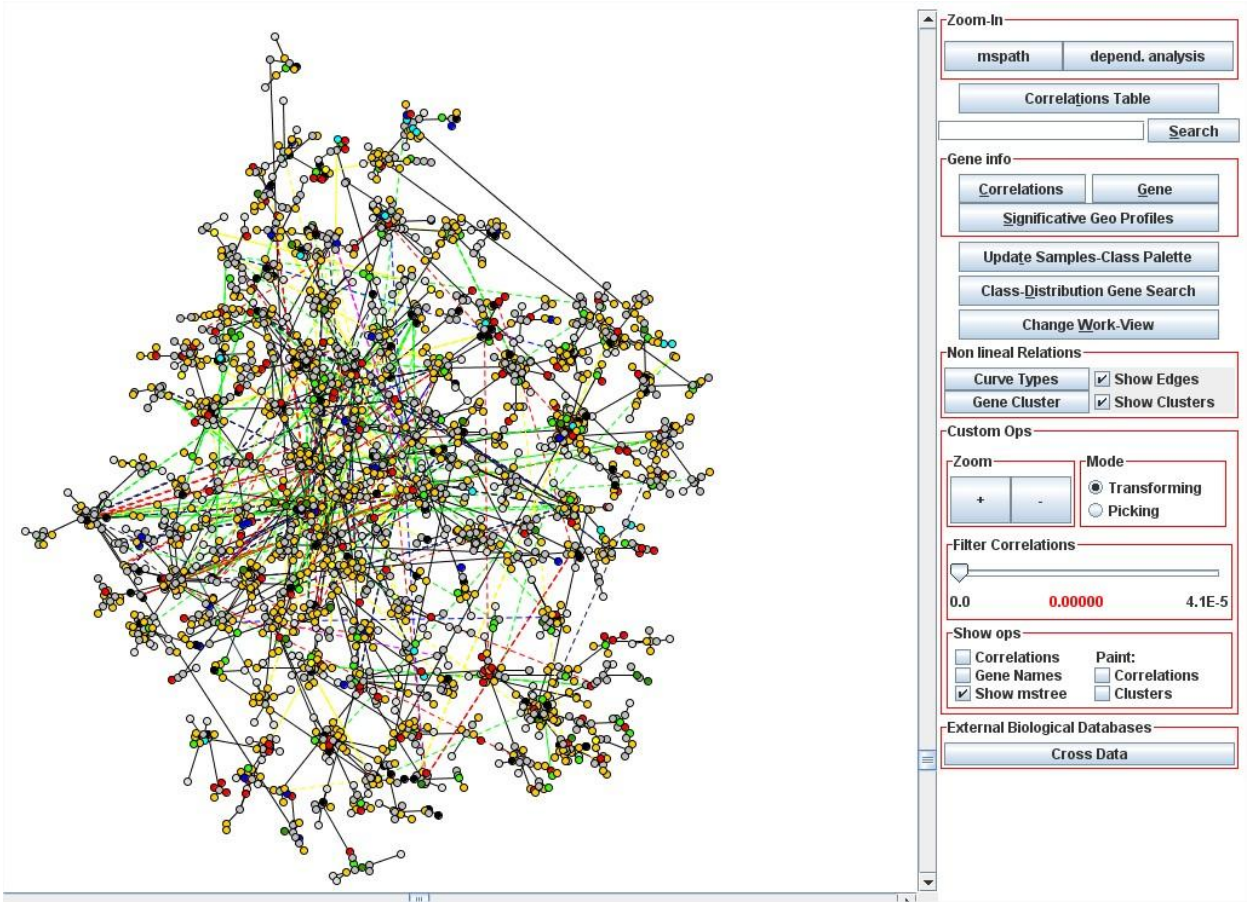
La primera microarray analizada es m17 que contiene 1416 genes.



La siguiente microarray es m22, contiene 664 genes.



Microarray m2501, contiene 3000 genes.



Microarray m2503, contiene 5000 genes.