UNIVERSIDAD PABLO DE OLAVIDE DE SEVILLA

Área de Lenguajes y Sistemas Informáticos
Escuela Politécnica Superior

Similitud Funcional de Genes basada en Conocimiento Biológico

Tesis Doctoral

Norberto Díaz Díaz

Sevilla, Enero de 2012

UNIVERSIDAD PABLO DE OLAVIDE DE SEVILLA

Similitud Funcional de Genes basada en Conocimiento Biológico

MEMORIA QUE PRESENTA
Norberto Díaz Díaz

PARA OPTAR AL GRADO DE DOCTOR POR LA UNIVERSIDAD PABLO DE OLAVIDE DE SEVILLA

DIRECTOR
Jesús S. Aguilar-Ruiz

Área de Lenguajes y Sistemas Informáticos
Escuela Politécnica Superior

Enero de 2012

D. Jesús S. Aguilar Ruiz, profesor Titular de Universidad adscrito al área de Lenguajes y Sistemas Informáticos de la Universidad Pablo de Olavide de Sevilla,

CERTIFICA QUE:

D. Norberto Díaz Díaz, Ingeniero Informática por la Universidad de Sevilla, ha realizado bajo su supervisión el trabajo de investigación titulado:

SIMILITUD FUNCIONAL DE GENES BASADA EN CONOCIMIENTO BIOLÓGICO

Una vez revisado, autoriza la presentación del mismo como tesis doctoral en la Universidad Pablo de Olavide de Sevilla y estima oportuna su presentación al tribunal que habrá de valorarlo. Dicha tesis ha sido realizada dentro del programa de doctorado *Tecnología e Ingeniería del Software,* con mención de calidad MCD2005-00261, del Departamento de Leguajes y Sistemas Informáticos de la Universidad de Sevilla.

Sevilla, Enero de 2012.

D. Jesús S. Aguilar Ruiz, profesor adscrito al área de Lenguajes y Sistemas Informáticos de la Universidad Pablo de Olavide de Sevilla, como director de la tesis titulada

SIMILITUD FUNCIONAL DE GENES BASADA EN CONOCIMIENTO BIOLÓGICO,

propone la siguiente composición del tribunal titular, a fin de que la Comisión de Doctorado designe al tribunal encargado de juzgar la tesis doctoral.

PRESIDENTE:	Dr. D. José C. Riquelme Santos
VOCALES:	Dr. D. José Clemente Litrán
	Dr. D. Marco Masseroli
	Dr. Dña Rocio C. Romero Zaliz
SECRETARIO:	Dr. Dña. Alicia Troncoso Lora
SUPLENTES:	Dr. D. Raúl Giráldez Rojo
	Dr. Dña. Cristina Rubio Escudero

A María

Agradecimientos

Son decenas las veces que he pensado cómo realizar este apartado ya que son muchas las personas, que de una forma u otra, me han ayudado a realizar este arduo trabajo. Sin duda, sería imposible nombrar de una en una a todas éstas aunque no puedo dejar escapar la oportunidad de mencionar a unas personas concretas.

En primer lugar, me gustaría comenzar por todos aquellos investigadores que me ayudaron y, en especial, a los miembros de mi grupo de investigación, a quien considero mi familia investigadora. A Pepe, el abuelo de la familia, que no por su edad sino por su conocimiento. Gracias por su tiempo, consejos y atención. A Jesús, por darme la oportunidad de formar parte de esto y por concederme total libertad para realizar esta línea de investigación. A quien considero mis hermanos de fatiga; Isa y Domingo. Ellos, mejor que nadie, entiende el trabajo que hay detrás de esta tesis, sin su ayuda y discusiones esto hubiera sido aún más difícil. En especial, a Domingo, por mantenerme los pies en el suelo e incentivarme a finalizar esta tesis. A mi "bio"–hermanastro, Juan Antonio, y a mis primos investigadores, Paco, Jorge y Bea. A mis tíos investigadores; a Federico por sus valiosas opiniones y por ayudarme a encaminar mi primera estancia; a Raúl y Alicia, por enseñarme el camino; y a Roberto, quien me ayudó como ninguno en mi labor docente, sabiendo entenderme en todo momento. Y como no, al resto de doctorando del grupo, quien me contagiaron sus ganas por comerse el mundo.

Por otro lado, agradecer a mi familia. A mis padres, Norberto y Ana, por concederme todo su apoyo, por abrirme todas las posibilidades de futuro, por todo. A mi hermana, con quien he compartido tantas y tantas cosas; por alentarme, por demostrarme lo que es el tesón. A mis cuñados, Diego, Gema, Ñoño y Belén, por esos ratos de diversión que me sirvieron para evadirme de los problemas investigadores. A mis suegros, por su comidas llenas de energía para seguir adelante.

En último lugar y más importante, gracias a mi mujer, María. Sin duda, se ha convertido en la persona más importante en mi vida y, de igual forma, de esta tesis. Sin ella esto no hubiera sido posible. Gracias por apoyarme en los momentos difíciles, por alentarme cuando parecía que todo se volvía en mi contra, por estar junto a mi donde y como fuera, por entenderme, por sufrirme, por aguantarme, por todo, gracias. Esta tesis es por y para ti.

Tesis Doctoral parcialmente subvencionada por el Ministerio de Educación y Ciencia con el proyecto TIN2007-68084-C-02-00

MEC
TIN2007-68084-C-02-00

Índice general

Índice de figuras

Índice de tablas

Índice de algoritmos

Parte I

Introducción

Capítulo 1

Introducción

Seis honrados servidores me enseñaron cuanto sé; sus nombres son cómo, cuándo, dónde, qué, quién y por qué.

RUDYARD KIPLING.

1.1. Planteamiento

En muchas áreas del saber, el conocimiento se ha venido obteniendo por el clásico método hipotético-deductivo de la ciencia positiva. En él es fundamental el paso inductivo inicial: a partir de un conjunto de observaciones y de unos conocimientos previos, la intuición del investigador le conduce a formular la hipótesis. Esta "intuición" resulta inoperante cuando no se trata de observaciones aisladas y casuales, sino de millones de datos almacenados en soporte informático. En el fondo de todas las investigaciones sobre inducción en bases de datos subyace la idea de automatizar ese paso inductivo.

Las técnicas de análisis estadístico, desarrolladas hace décadas, permiten obtener ciertas informaciones útiles, pero no inducir relaciones cualitativas generales, o leyes, previamente desconocidas. Por ello se hizo necesaria la aparición de un procesamiento automático que extraiga conocimiento.

El proceso global de búsqueda de nuevo conocimiento a partir de un conjunto de datos se denomina *KDD* (*Knowledge Discovery in Data bases*). Como se muestra en la figura 1.1, este proceso comprende diversas etapas, que van desde la obtención de los datos hasta la aplicación del conocimiento adquirido en la toma de decisiones. Entre esas etapas, se encuentra la que puede considerarse como el núcleo del proceso KDD y que se denomina Minería de Datos o Data Mining (DM). Esta fase es crucial para la obtención de resultados apropiados, pues durante la misma se aplica el algoritmo de aprendizaje automático encargado de extraer el conocimiento inherente a los datos, además de que sea refinado y validado. No

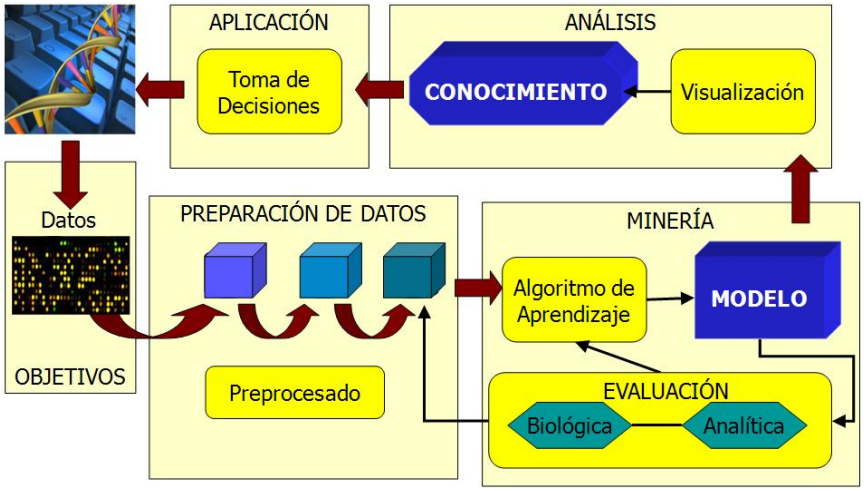


Figura 1.1: Proceso KDD sobre datos biomédicos.

obstante, esta fase se ve influida en gran medida por la calidad de los datos que llegan para su análisis desde la fase previa.

Las técnicas de minería de datos han encontrado una gran aplicación en el campo de la bioinformática. La gran y diversa expansión de la cantidad de datos producidos por la biología molecular moderna han generado una necesidad de algoritmos precisos de predicción y clasificación. La precisión de los algoritmos de clasificación puede verse afectada por una diversidad de factores, algunos de ellos considerados genéricos en cualquier algoritmo de aprendizaje automático y, por tanto, aplicables a los distintos campos de investigación. Son estos factores genéricos los que han recibido atención por la comunidad de *aprendizaje automático* y *reconocimiento de patrones* durante un gran número de años. Por el contrario, la aplicación de técnicas de minería a grandes volúmenes de datos biomédicos es un suceso relativamente nuevo.

En la mayoría de aplicaciones bioinformáticas la naturaleza de los datos requiere novedosas modificaciones de los algoritmos y procesos existentes, e incluso en algunas ocasiones de técnicas de análisis totalmente nuevas. Igualmente, las modificaciones en los algoritmos y, sobre todo, en la tipología de los datos biomédicos, provocan la necesidad de desarrollar nuevas metodologías y técnicas que contrasten y validen el nuevo conocimiento extraído.

En este contexto, el punto de partida de esta investigación es el nuevo marco de trabajo que se abre para las ciencias de la información, con la necesidad del diseño de técnicas automáticas de validación de conocimiento capaz de procesar modelos sobre bases de datos enormes en tiempos comprensiblemente reducidos.

1.2. Objetivos

Este trabajo se centra en la validación biológica de los modelos generados por la aplicación de técnicas de minería de datos a información proporcionada por tecnología de microarrays a través del enriquecimiento y similitud funcional de los genes que lo componen. El objetivo principal de esta tesis se subdivide en los siguientes sub–objetivos.

- Estudiar las medidas de similitud funcional de grupos de genes actuales y desarrollar una nueva medida que resuelva sus carencias. Las medidas actuales utilizan, indistintamente, todas las funcionalidades en las que interviene un gen para ponderar la coherencia de los datos de entrada; siendo un problema cuando se analizan genes que intervienen en diferentes procesos. La propuesta a desarrollar tendría el objetivo de seleccionar una única funcionalidad por cada gen y estudiar su coherencia a través de ella. La funcionalidad debería ser la más común y específica según el contexto completo de genes.

  Igualmente, en la literatura actual no existe ninguna representación que permita interpretar gráficamente la similitud de uno o varios grupos de genes. Así, otro objeto será generar una representación espacial que proporcione una visualización conjunta de la cohesión funcional de varios grupos de genes y que posibilite la comparación gráfica de diferentes técnicas de minería de datos.

- Analizar las herramientas de análisis de enriquecimiento funcional actuales y diseñar e implementar una nueva aproximación. Las herramientas actuales suelen basarse en el mismo conjunto de información biológica; Gene Ontology. La carencia de no utilizar otros conjuntos de datos contrastados es la base para plantearnos como objetivo el de diseñar e implementar una herramienta software de análisis de enriquecimiento para la validación de resultados a través de repositorios biológicos públicos. Esta herramienta se utilizaría para analizar la coherencia biológica de grupos de genes proporcionados como resultados de una técnica computacional de inferencia a partir de microarray. El repositorio público a utilizar sería *Kyoto Encyclopedia of Genes and Genomes* (KEGG), y dicha herramienta analizaría el enriquecimiento de rutas metabólicas a partir de grupos de genes mediante procedimientos de contraste de hipótesis múltiple.

1.3. Principales contribuciones

1. ***Artículos publicados en revistas:***

   - GO-based Functional Dissimilarity of Gene Sets. BMC Bioinformatics, vol 12, pp. 360, 2011.
   - CarGene: Characterization of Sets of Genes based on Metabolic Pathways Analysis. International Journal of Data Mining and Bioinformatics. vol 5, Num. 5, pp. 558, 2011.

2. ***Artículos publicados en congresos internacionales:***

   - Gene–Gene Interaction based Clustering method for Microarray Data (ISDA'11). IEEE press, In press.
   - Gene Networks Validation based on Metabolic Pathways. IEEE International Conference on Bioinformatics and Bioengineering (BIBE'11). ISBN: 978-1-61284-975-1, pp. 9–14, IEEE press, 2011.
   - Pattern recognition in biological time series (CAEPIA'11). Lecture Note in Computer Science 7023. ISBN 978-3-642-25273-0, pp 164-172, 2011.
   - Gene Regulatory Networks Validation Framework based in KEGG (HAIS'11). Lecture Notes in Artificial Intelligence 6679, pp 279-286, 2011.
   - Discovering alpha expression patterns from gene expression data (IDEAL'07). Lecture Notes in Computer Science. ISSN 0302–9743 , vol 4881, pp. 831-839, Springer–Verlag, 2007.
   - A Deterministic Model to Infer Gene Networks from Mircroarray Data (IDEAL'07). Lecture Notes in Computer Science. ISSN 0302–9743 , vol 4224, pp. 850–859, Springer–Verlag, 2007.
   - Neighborhood-based Clustering of Gene-Gene Interactions (IDEAL'06). Lecture Notes in Computer Science. ISSN 0302–9743, vol 4224, pp. 1111–1120, Springer–Verlag, 2006.
   - A measure for data set editing by ordered projections. Applications on Data Mining (IEA/AIE'06). Lecture Notes in Artificial Intelligence. ISSN 0302–9743, vol 4031, pp. 1339–1348, Springer–Verlag, 2006.
   - Classification with VNS. In Proceeding of the 18th Mini Euro Conference on VNS. ISBN 84–689–5679–1, Num 18, pp. 1–2, 2005.

- Feature Selection based on bootstrapping. Advanced Computing in Biomedicine (CIMA'05). IEEE Conference Proceeding. ISBN 1–4244–0020–1, 2005.
- Analysis of Feature Rankings for Classification. 6th International Symposium on Intelligent Data Analysis (IDA'05). Lecture Notes in Computer Science. ISSN 0302-9743, vol 3646, pp. 362–372, Springer-Verlag, 2005.
- An Approach to Reduce the Cost of Evaluation in Evolutionary Learning. Computational Intelligence and Bioinspired Systems: 8th International Workshop on Artificial Neural Networks (IWANN'05). Lecture Notes in Computer Science. ISSN 0302-9743, vol 3512, pp. 804–811, Springer-Verlag, 2005.

3. ***Artículos publicados en congresos nacionales:***

- Uso de Rutas Metabólicas para la Validación de Redes Genéticas (EVaBio'11). Actas CAEPIA'11, Pp 753–760, 2011.
- Caracterización de un Conjunto de Genes basado en el Análisis de Pathways Metabólicos. VI Taller Nacional de Minería de Datos y Aprendizaje (TAMIDA'10). Pp 251–264, 2010.
- Software y técnicas de validación de conocimiento en bioinformática. I Workshop Español sobre Extracción y Validación de Conocimiento en Bases de Datos Biomédicas (EvaBio'07). ISBN-13:978-84-611-8854-3, pp. 75–84, 2007.
- Algoritmo de inferencia de Redes de Genes a partir de tecnología Microarray . I Workshop Español sobre Extracción y Validación de Conocimiento en Bases de Datos Biomédicas (EvaBio'07). ISBN-13:978-84-611-8854-3, pp. 75-84, 2007.
- InterClus: Clustering basado en la Vecindad de la Interacción Gen-Gen. V Taller Nacional de Minería de Datos y Aprendizaje (TAMIDA'07). ISBN: 84-7643-872-3, pp. 121–130, 2007.
- Aprendizaje Semisupervisado basado en VNS. V Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'07). ISBN: 978-84-690-3470-5, pp. 549–554, 2007.
- Análisis de Datos de Expresión Genética. XVII Jornadas de Automática. ISBN: 84-689-9417-0, Num 27, pp. 911–918, 2006.
- Un algoritmo heurístico para Clasificación Semisupervisada. XXIX Congreso Nacional de Estadística e Investigación Operativa (SEIO'06). ISBN: 84-689-8553-8, Num 29, pp. 653–653, 2006.

- Biclustering de Datos de Expresión Genómica con Computación Evolutiva Multiobjetivo. IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'05). ISBN: 84-9732-440-4, pp. 229–236, 2005.
- Selección de atributos relevantes basada en bootstrapping. III Taller Nacional de Minería de Datos y Aprendizaje (TAMIDA05). ISBN: 84-9732-449-8, pp. 21–30, 2005.

1.4. Organización

El contenido de esta memoria de investigación se encuentra dividido en nueve capítulos diferentes. En el capítulo 2 se ha realizado una introducción a la bioinformática, además de centrar la labor de la minería de datos en esta nueva ciencia interdisciplinar y exponer los conceptos biológicos clave en los que se basa esta tesis.

En el capítulo 3, se realiza un estudio sobre las técnicas más relevantes para el análisis de datos de expresión génica basadas en microarrays; clustering, biclustering y redes genéticas reguladoras.

En el capítulos 4 se expone la necesidad de la fase de validación para medir la calidad de los modelos generados por cualquier técnica de análisis. Concretamente aquellas metodologías basadas en estudios analíticos/matemáticos de los datos de entrada son expuestas en el capítulo 4. Éstas son divididas en medidas externas e internas, atendiendo a si se basan en el conocimiento de las etiquetas correctas de la clase o si sólo usan la información intrínseca de los datos. Así mismo, son presentadas diferentes metodologías para realizar una validación gráfica de grupos de genes.

La necesidad de la validación de los resultados de técnicas de análisis de microarrays desde un punto de vista biológico es presentado en el capítulo 5. Nótese que la interpretación biológica del conocimiento extraído es aún una etapa esencial en cualquier estudio microarray. Esta necesidad no es cubierta por las técnicas de validación analítica, siendo necesarias metodologías que usen el conocimiento biológico disponible. En este sentido, serán expuestos los diferentes repositorios de datos biológicos existentes en primer lugar, haciendo hincapié en las bases de datos especializadas Gene Ontology y KEGG. Posteriormente, las dos aproximaciones más relevantes para el análisis de la coherencia funcional (herramientas de enriquecimiento y medidas de similitud funcional) son detalladas. Para las herramientas de enriquecimiento, se actualizará el estudio realizado por Khatri et al. [183] sobre herramientas de análisis funcional basadas, principalmente, en GO. Mientras que las medidas de similitud serán estudiadas diferenciando entre métricas entre

GO–terms, gene–products y genes.

En el capítulo 6 se desarrolla la primera propuesta y aportación de esta tesis. Concretamente es tratado la carencia existen sobre herramientas que estudien en enriquecimiento funcional sobre las rutas metabólicas existente. A continuación presentamos una herramienta para el análisis de enriquecimiento basado en la información de pathways almacenada en KEGG.

En el capítulo 7 se presenta la principal propuesta de esta tesis. En primer lugar, se presenta una metodología para evaluar la similitud funcional de un conjunto de genes. Esta medida, denominada GFD, se basa en la información biológica en Gene Ontology (GO) y en una organización en árbol (GOTREE) de tal conocimiento. GFD se centra en encontrar la función más cohesiva (común y específica) del conjunto de genes de entrada para evaluar la similitud funcional de tales genes. Posteriormente es propuesta una heurística, basada en Diagramas de Voronoi, para reducir el coste computacional de la búsqueda tal funcionalidad. Por último, es expuesta una poderosa representación, GOGRAM , para interpretar gráficamente la evaluación obtenida para los tres puntos de vistas biológico u ontologías que contiene GO.

En el capítulo 8 un análisis del comportamiento de la aproximación heurística frente a la exhaustiva para el cálculo de GFD es llevado a cabo. Primeramente, es realizado un estudio de la eficacia de la heurística. Para ello se realiza una comparación de los resultados generados por ambas aproximaciones al ser aplicadas a árboles con diferentes topologías. En segundo lugar, se compara la eficacia de la heurística realizando una comparación de los espacios de búsqueda usados por las aproximación exhaustiva y heurística. El estudio realizado en este capítulo demuestra que el algoritmo heurístico consigue reducir considerablemente el coste computacional sin afectar significativamente a la calidad de los resultados.

En el capítulo 9 es evaluada la utilidad de la medida de similitud propuesta. En primer lugar, un estudio riguroso sobre la fase S del ciclo celular fue llevada a cabo con el objetivo de validar el sentido de la medida GFD. Tras probar la validez de la propuesta, es comparada con tres medidas de similitud existentes mediante un análisis ROC. Posteriormente, un estudio de robustez de GFD será presentando para estudiar su comportamiento ante la aleatoriedad. Y, por último, se expondrá la utilidad de la representación gráfica GOGRAM para interpretar gráficamente las similitudes obtenidas para las tres ontologías.

Finalmente, en el capítulo 9 se muestran las conclusiones y trabajos futuros de esta tesis. Y por último, siguen un apéndice y la bibliografía utilizada en este documento de tesis.

Capítulo 2

Bioinformática: una nueva ciencia interdisciplinar

If you can't do Bioinformatics, you can't do Biology.
J.D. TISDALL.

2.1. Introducción

En este capítulo se introducen algunos conceptos que son relevantes para el resto de esta tesis. Se comenzará describiendo qué es la bioinformática y cuáles son sus finalidades. A continuación, y con el fin de centrar el marco de trabajo de este documento, los cometidos de la bioinformática se asocian a las diferentes técnicas de aprendizaje que pueden ser usadas para su resolución.

Una vez establecido el objeto de esta tesis, se introducirán brevemente las nociones biológicas necesarias para el entendimiento del contenido de ésta y se presentará la tecnología de microarrays.

2.2. ¿Qué es la Bioinformática?

El desarrollo de la ingeniería genética y las nuevas tecnologías de la información durante la última década del siglo XX, ha condicionado el surgimiento de una disciplina que ha generado vínculos indisolubles entre la ***Informática*** y las ***Ciencias Biológicas*** [238]: la Bioinformática.

La Bioinformática se encuentra en la intersección de las ciencias de la vida y las ciencias de la información. Es un campo científico interdisciplinario que se propone investigar y desarrollar sistemas que faciliten la comprensión del flujo de información desde los genes a las estructuras moleculares, su función bioquímica,

su conducta fisiológica y finalmente su influencia en las enfermedades y la salud [219].

Entre los principales factores que han favorecido el desarrollo de esta disciplina, se encuentra el impresionante volumen de datos sobre secuencias generadas por los distintos proyectos genoma (tanto el humano como los de otros organismos); los nuevos enfoques experimentales, basados en biochips que permiten obtener datos genéticos a gran velocidad, bien de genomas individuales (mutaciones, polimorfismos), o de enfoques celulares (expresión génica); así como el desarrollo de Internet y la World Wide Web, que permite el acceso mundial a las bases de datos de información biológica.

El término ***Bioinformática*** es relativamente reciente. Apareció en la literatura a principios de 1990, cuando comenzaba a estructurarse el llamado "Proyecto Genoma Humano" y el *National Center for Biotechnology Information* de los Estados Unidos, daba sus primeros pasos. En los primeros momentos, su vinculación con la Informática Médica se debió meramente a la similitud semántica, así como al indispensable uso de las computadoras por parte de ambas disciplinas [137].

La ***Informática Médica*** como disciplina que se preocupa por el análisis y la diseminación de datos médicos mediante la aplicación de las computadoras a varios aspectos de la práctica sanitaria, incluye sistemas automatizados de diagnóstico, terapia y comunicación de información de salud. Se relaciona con casi todas las especialidades médicas y configura un sector multidisciplinario con ramificaciones en la epidemiología, evaluación de la tecnología, economía, gestión sanitaria y ética médica. Tiene más de 40 años de existencia, y se ha convertido en una ciencia médica básica que comprende los aspectos teóricos y prácticos relacionados con el procesamiento y la comunicación de información derivada de procesos médicos y relacionados con la salud [222].

La Bioinformática, por su parte, surge como respuesta a la respuesta a la avalancha de datos biológicos. Mientras que hace unos años, los resultados de los experimentos podían interpretarse sobre el cuaderno de laboratorio, hoy se necesitan bases de datos y técnicas de visualización sólo para almacenarlos y comenzar a estudiarlos. Esta nueva disciplina pasa de ser un conjunto de técnicas a una verdadera ciencia, al aportar el componente de análisis para entender los procesos de la vida e integrar los datos que permitan crear modelos predictivos para los sistemas biológicos [219, 222].

No obstante, con el advenimiento del nuevo milenio, el estudio de procesos celulares (bioinformática) y de la información clínica (informática médica) amenaza con fundir ambas disciplinas en una sola, lo que algunos autores han definido como Informática Biomédica. En la medida que se genera información sobre el genoma humano y ésta se vincula con el conocimiento médico de las enfermedades, esta definición ha comenzado a hacerse realidad. Los datos que maneja la bioinformá-

tica tienen cada vez más presencia en la práctica médica; por tanto, conseguir la unificación de la información clínica con la información molecular representa el desafío más importante de esta disciplina durante el presente siglo [185].

2.3. Minería de Datos en Bioinformática

El crecimiento ingente de datos biológicos disponibles en la actualidad ha provocado dos problemas: por un lado, el almacenamiento y manejo eficiente de información y, por otro, la extracción de información útil a partir de dichos datos. El segundo de ellos es uno de los principales desafíos en la biología computacional, el cual requiere el desarrollo de herramientas y métodos capaces de transformar todos esos datos heterogéneos en conocimiento biológico sobre los mecanismos subyacentes [46]. Estas herramientas y métodos deben proporcionarnos una descripción más allá de los datos y el conocimiento suministrado en forma de modelo testeable. A partir de esta abstracción simplificada que constituye un modelo, es posible obtener predicciones de sistemas.

Existen distintos dominios biológicos donde las técnicas de minería de datos son aplicadas a la extracción de conocimiento. La figura 2.1 muestra un esquema de los principales problemas biológicos en donde los métodos computacionales están siendo aplicados. Estos problemas han sido clasificados por Larrañaga et al. [193] en seis dominios diferentes: genómicos, proteómicos, microarrays, biología de sistemas, evolución y minería de textos o *text mining*. La categoría denominada "otras aplicaciones" agrupa al resto de problemas. Estas categorías deberían ser entendidas de una forma general, especialmente la genómica y la proteómica, las cuales podrían ser consideradas como el estudio de cadenas de nucleótidos y proteínas, respectivamente.

La ***genómica*** es uno de los dominios más importantes en la bioinformática. El número de secuencias disponibles se incrementa exponencialmente haciendo que estos datos necesiten ser procesados para obtener información útil. A partir de la secuencias del genoma, se pueden extraer la localización y estructuras de genes [220]. Recientemente, la identificación de elementos reguladores [3, 52, 309] y genes no codificadores de ARN [69] son también abordados desde un punto de vista computacional. La información secuencial es también usada para la predicción de funciones genéticas y de la estructura secundaria del ARN.

Si los genes contienen la información, las proteínas son los trabajadores que transforman esta información en vida. Las proteínas juegan un papel muy importante en los procesos de la vida, y su estructura 3D es una característica fundamental en su funcionalidad. En el dominio de la ***proteómica***, la principal aplicación de los métodos computacionales es la predicción de la estructura de proteínas. Las

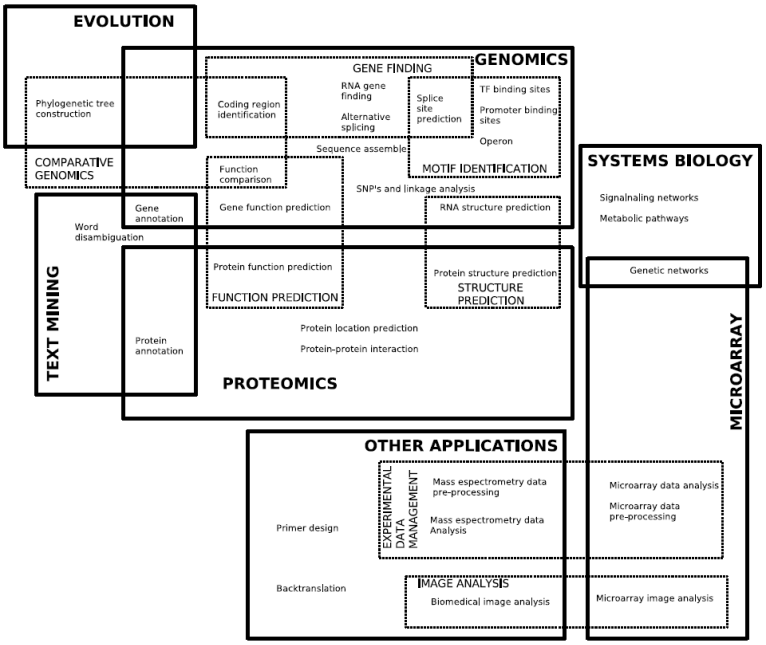


Figura 2.1: Clasificación de los problemas bioinformáticos según la aplicación de métodos de aprendizaje.

proteínas son macromoléculas muy complejas con miles de átomos y ligaduras. Por ello, el número de posibles estructuras es inmenso. Esto hace que la predicción de estructura de proteínas sea un problema computacional muy complicado donde las técnicas de optimización son requeridas. En la proteómica, como en el caso de la genómica, las técnicas de minería de datos son aplicadas a la predicción de la función proteínica.

Otra aplicación interesante de los métodos computacionales en biología es el manejo de datos experimentales complejos. Los ***microarrays*** son los dominios más conocidos donde este tipo de información es almacenada. Estos datos experimentales provocan dos problemas diferentes. Primero, los datos necesitan ser preprocesados, es decir, modificados para ser usados adecuadamente por algoritmos de aprendizaje automático. Segundo, al análisis de los datos, que a su vez, depende de lo que estemos buscando. En el caso de los microarrays, la aplicación más típica es la identificación de patrones, clasificación e inducción de redes reguladoras genéticas.

La ***biología de sistemas*** es otro dominio donde la biología y la minería de datos trabajan en conjunto. Es muy complejo modelar los procesos de la vida que tienen lugar dentro de la célula. Por ello, las técnicas computacionales son extremadamente prácticas cuando se desea modelar redes biológicas [57], especialmente

redes genéticas o rutas metabólicas[1].

La ***evolución***, y especialmente la reconstrucción de árboles filogenéticos, también aprovechan las técnicas de minería de datos. Los árboles filogenéticos son representaciones esquemáticas de la evolución de los organismos. Tradicionalmente, éstos eran construidos de acuerdo con diferentes características (morfológica, metabólica, etc.) pero, actualmente, con el gran crecimiento de secuencias de genomas disponibles, los algoritmos de construcción de árboles filogenéticos están basados en la comparación entre diferentes genomas [24]. Esta comparación es realizada mediante el alineamiento de secuencias múltiples, donde las técnicas de optimización son muy útiles.

El efecto de la aplicación de técnicas computacionales al incremento de datos se ve reflejado en el aumento de publicaciones disponibles. Esto provee una nueva fuente de información valiosa, donde las técnicas de ***minería de textos*** son requeridas para la extracción de conocimiento. De este modo, la minería de textos se está haciendo más y más interesante en la biología computacional, y está siendo aplicada en anotaciones funcionales, predicción de localización celular y el análisis de interacción entre proteínas [187]. Una revisión de la aplicación de las técnicas de minería de datos en biología y biomedicina pueden ser encontradas en [13].

Además de estas aplicaciones, las técnicas computacionales son usadas para resolver otros problemas, tales como el análisis de imágenes biológicas o el preprocesado de datos provenientes de la espectrometría.

Como se describió al comienzo de este documento, la minería de datos es la fase crucial del proceso *KDD* y consiste en el desarrollo de algoritmos computacionales que optimicen un cierto criterio usando ejemplos o experiencias pasadas. El criterio de optimización puede ser la precisión de un determinado modelo para un problema de modelado, o el valor de la función de evaluación para uno de optimización.

En un ***problema de modelado***, el término "aprendizaje" se refiere a la ejecución de un programa computacional que induzca un modelo basándose en datos de entrenamiento y experiencias pasadas. La minería de datos, a veces, usa teoría estocástica para construir modelos computacionales, ya que el objetivo es realizar inferencias a partir de ejemplos. Las dos principales etapas en este proceso son, inducir el modelo procesando la gran cantidad de datos, y representar el modelo y realizar eficientes inferencias. Nótese que la eficiencia del algoritmo de aprendizaje, al igual que sus espacios, complejidad y su transparencia e interpretabilidad, pueden ser tan importantes como su precisión predictiva. El proceso de transformación de datos a conocimiento (KDD) es iterativo e interactivo. La fase iterativa se divide en varias subfases. La primera de ellas tiene el objetivo de integrar y com-

[1] Serie de reacciones químicas que ocurren dentro de la célula.

binar fuentes de información diferente en un único formato. El uso de técnicas de *data warehouse* soluciona la detección y resolución de *outliers* e inconsistencia. Por otro lado, en la segunda subfase (preprocesado) es necesario seleccionar, limpiar y transformar los datos. Para llevar a cabo esta tarea, se eliminan o corrigen los datos incorrectos, además de decidir la estrategia que trate los datos omisos. En esta etapa también se selecciona las variables relevantes y no redundantes. La tercera subfase, denominada minería de datos, tiene por objetivo seleccionar el análisis de datos más apropiados y su estudio posterior. En esta etapa, el tipo de paradigma de aprendizaje, supervisado o no supervisado, debería ser seleccionado y a partir de él inducir el modelo en base a los datos. Una vez obtenido el modelo, éste debe ser evaluado e interpretado –ambos desde un punto de visto estadístico y biológico– y, si fuera necesario, retornar al paso anterior para una nueva iteración. Chequeado satisfactoriamente el modelo y descubierto el nuevo conocimiento, éste es usado para resolver el problema de partida.

El ***problema de optimización*** puede ser planteado como el problema de encontrar una solución óptima en un espacio de soluciones múltiples. La selección del método de optimización es una parte crucial para solucionar este tipo de problema. Las diferentes técnicas de optimización para problemas biológicos pueden ser clasificadas según el tipo de solución encontrada: métodos exactos o aproximados. Los métodos exactos obtienen como resultado soluciones exactas cuando se consigue la convergencia. Sin embargo, éstos no convergen necesariamente en todos los casos. Los algoritmos de aproximación siempre proporcionan una solución candidata, pero no se garantiza que sea la óptima.

La optimización es también una tarea fundamental en los problemas de modelado. De hecho, los procesos de aprendizaje pueden ser considerados como la búsqueda del mejor modelo que describa los datos. En esta búsqueda en el espacio de modelos cualquier tipo de heurística puede ser usada. Por tanto, los métodos de optimización pueden ser considerados como una parte del modelado.

La finalidad de este documento es realizar un estudio sobre las diferentes metodologías de evaluación de modelos genéticos actuales y proponer una novedosa técnica que permita la evaluación de éstos desde un punto de vista puramente biológico. Por ello, esta investigación se enmarca dentro de la genómica y sería aplicable a los resultados obtenidos en el análisis de microarrays.

2.4. Datos de expresión génica

2.4.1. ADN, ARN, genes y expresión genética

La característica fundamental del *ADN* (ácido desoxiribonucleico) es su capacidad de almacenar y transmitir la información genética a las células hijas, gracias a su capacidad para duplicarse (***replicación***). Esto es posible gracias a la complementariedad de las bases nitrogenadas que componen su estructura y que fueron definidas en el trabajo pionero de Watson y Crick en la década de los cincuenta (Adenina, Guanina, Citosina y Timina). Dichos investigadores describieron la estructura básica del ADN como una estructura en doble hélice.

La replicación se lleva a cabo en dos etapas básicas: la separación de las hebras de la doble hélice y la síntesis de hebras complementarias gracias a la acción de la ADN polimerasa.

El proceso de ***descodificación*** de la información es conceptualmente simple, sin embargo su grado de complejidad aumenta de manera progresiva cuanto más evolucionado sea el organismo. Básicamente, la información contenida en el ADN es "leída", una vez se haya producido la separación de la doble hélice en hebras individuales mediante la enzima ARN polimerasa, en un proceso llamado ***transcripción***; generándose una molécula de *ARN* por cada *gen*. En el proceso de ***traducción*** este ARN, denominado mensajero, abandona el núcleo e interacciona en los ribosomas con otras moléculas de ARN, ya sean estructurales (ARN ribosomal) o portadoras de aminoácidos (ARN de transferencia), dando origen al producto final: una **proteína**.

Este proceso, representado en la Figura 2.2, está basado en los paradigmas de la complementariedad de las bases nitrogenadas (las purinas, adeninas, y guaninas se unen siempre con las pirimidinas, timidina [o uracilo en el ARN] y citosina respectivamente) y del código genético. Una combinación específica de tres bases, llamada codón, da origen a un aminoácido específico.

Se estima que la información genética del ser humano comprende entre 20000 y 30000 genes. Conceptualmente, se puede definir a un **gen** como una porción discreta de ADN que codifica una determinada proteína. La estructura básica de un gen se muestra en la figura 2.3.

En los genes existen fragmentos de ADN (denominados intrones) que, tras la transcripción, son eliminados durante el procesamiento del ARN, por lo que no se encuentran en el ARN mensajero maduro. Las secuencias que finalmente codifican la información para sintetizar una determinada proteína se denominan exones. Además, existen zonas reguladoras que no codifican información, sino que unen proteínas capaces de modificar, positiva o negativamente, la transcripción del gen. Estas zonas se conocen, genéricamente, con el nombre de *promotor* y se localizan,

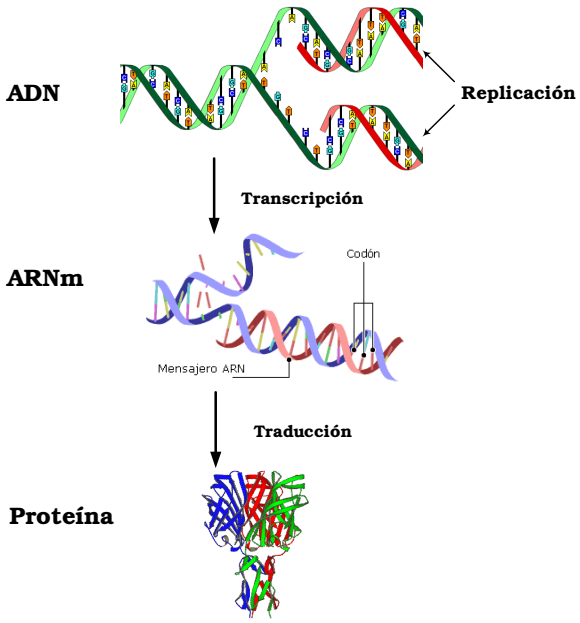


Figura 2.2: Procesos básicos de la expresión de los genes.

por regla general, antes del comienzo de la transcripción. El inicio de la transcripción suele estar dado por una secuencia estándar de bases nitrogenadas a la cual le siguen otras (secuencias) no codificantes que son importantes para la estabilidad del ARN mensajero y que no son traducidas finalmente. Del mismo modo, existen zonas no traducidas en el extremo distal del ADN y cuya adición está medida por una secuencia o señal de poliadenilación. Dicho extremo, correspondiente a una secuencia continua de adeninas, facilitan la maduración de éste y el término de la transcripción.

Los procesos de replicación, transcripción y traducción están regulados de un modo complejo. La información disponible a este respecto es abundante, pero sin duda parcial e incompleta. Desde un punto de vista conceptual, el modo en que se regula la expresión de los genes es fascinante si se considera que, de la totalidad de los genes existentes, la mayoría de ellos están destinados a la regulación de la expresión de otros genes y, sólo el 10 ó 20 % están destinados a la síntesis de proteínas estructurales y/o funcionales. La transcripción de un gen es probablemente el proceso más crítico y regulado. La existencia de proteínas capaces de unirse a secuencias específicas del ADN e influir en la velocidad con que los genes son transcritos es uno de los factores más relevantes en este sentido. Dichas proteínas reciben el nombre genérico de **factores de transcripción** y pueden actuar como activadores o represores de la transcripción de un gen determinado. La mayoría de los lugares de unión para factores de transcripción se encuentran ubicados en la

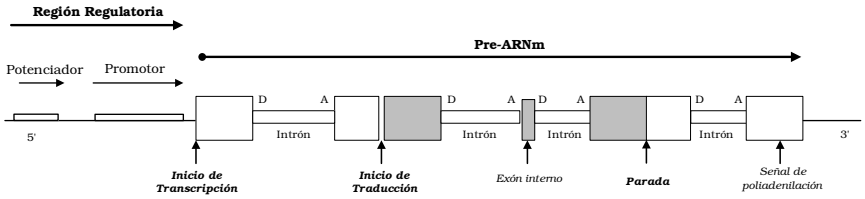


Figura 2.3: Representación esquemática de un gen eucariota.

región del promotor de los genes (ver Figura 2.3).

Estas proteínas realizan cambios conformacionales en el ADN durante la interacción ADN-factor de transcripción, modificando así la capacidad de dicho ADN de ser transcrito por la ARN polimerasa. Es interesante que un factor de transcripción pueda influir en la transcripción de varios genes a la vez, ya que, desde esta perspectiva, los factores de transcripción son obvios candidatos a ser genes responsables de enfermedades complejas si el gen que los codifica presenta alguna mutación. De hecho, esto ha sido descrito en enfermedades tales como la talasemia, algunas endocrinopatías y leucemias. Por último, se puede señalar que existen, también, mecanismos de control post-transcripcionales (relacionados fundamentalmente con el control de la estabilidad del ARN) y de traducción que, sólo recientemente, se han comenzado a dilucidar.

En resumen, la expresión genética se puede entender como la actividad que lleva a cabo un gen en una célula y en un cierto momento. Este concepto es importante por diferentes motivos. Primero, el conocer qué proteínas han sido producidas en una célula concreta, nos ayudaría a distinguir entre diferentes tejidos, pudiendo incluso discernir tejidos tumorales. Segundo, si podemos medir cuáles y cuántos genes están expresados en una célula en un cierto instante o bajo unas condiciones concretas, la comparación de estas medidas puede decirnos si la célula está sana o no. De esta forma, diferenciando o averiguando el tipo de tumor de un tejido o diagnosticando a un paciente, son obvias las aplicaciones para la expresión genética.

2.4.2. Microarrays

Debido al aumento de información biológica almacenada, se necesitaban técnicas que permitieran el análisis simultáneo de la expresión de un gran número de genes, estableciéndose métodos de estudios en serie (análisis directo de un gran número de ADNs) y en paralelo (hibridación con ADNs que estaban fijados en distintos sustratos). La necesidad de conseguir un estudio simultáneo de la expresión de cientos de genes con un bajo coste económico llevó a la creación de técnicas de detección que permitieran disponer de una gran densidad de sondas colocadas en

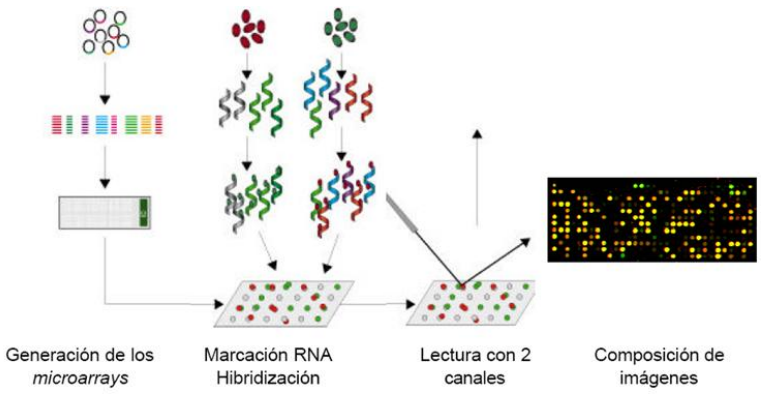


Figura 2.4: Creación de un microarry.

un área muy pequeña, por ejemplo 10.000 ADNs en una superficie de unos pocos centímetros.

El poder estudiar cómo se expresan cientos de genes en un solo ensayo permite realizar comparaciones entre tejidos, estados del desarrollo, patológicos o las diferentes respuestas que a lo largo del tiempo produce el ambiente en el que se encuentra una célula. Como consecuencia de ello, el número de datos a ser valorado crece exponencialmente. Además, estos datos precisan de la información existente en grandes bases de datos genómicas, como por ejemplo, la información relativa a genes cuya función es conocida. Por otro lado, se necesita un sistema para almacenar los datos (y salvaguardarlos) y potentes herramientas de análisis estadístico y tratamiento de imágenes. En definitiva, el conocimiento acumulado en los últimos años es necesario para entender el significado biológico de la información que se obtiene de los patrones de hibridación.

La tecnología MicroArray (biochip) [271, 263] es una de las diferentes aproximaciones al análisis comparativo de patrones de expresión de ARNs, cuyo fin sería colocar en una micromatriz cada uno de los genes de un genoma cuyos niveles de expresión pueden ser cuantificados [28]. Para ello se sintetiza el material genético y se insertan de forma automática en una capa de cristal, silicio o plástico, colocándose en unas 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 biochip), y se procede al revelado mediante un escáner óptico o con microscopia láser confocal. El resumen de todo este proceso es mostrado en la figura 2.4.

Dado el estado actual de las tecnologías, los experimentos microarrays no son perfectos provocando que los datos resultantes posean una gran cantidad de ruido. El primer paso en el análisis de estos datos se denomina análisis a bajo nivel y

consiste en reducir el ruido introducido en el proceso experimental. Este análisis incluye análisis de imágenes, normalización, manejos de valores omisos, selección de atributos, etc. En la literatura existen multitud de trabajos que estudian esta cuestión, tales como [153, 266, 249, 307] donde el problema de la normalización es tratado o en [295] donde se examina el problema de la estimación de valores omisos.

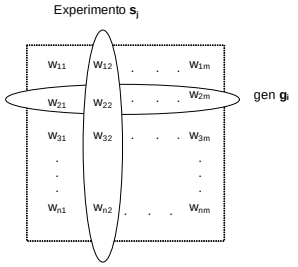


Figura 2.5: Matriz de expresión genética.

El análisis a bajo nivel transforma la matriz de datos original a una nueva matriz denominada ***matriz de expresión***. Esta nueva matriz, por tanto, contendrá a un conjunto de datos de expresión genética provenientes de un microarray y que serán representados por valores reales en m filas y n columnas (ver figura 2.5):

$$M = w_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m$$

donde las filas ($G = \vec{g_1}, ..., \vec{g_n}$) representan a los genes, las columnas ($S = \vec{s_1}, ..., \vec{s_m}$) a los pacientes/condiciones, y cada celda w_{ij} al nivel de expresión medido para el gen i en la condición j.

Una vez obtenida la matriz de expresión basada en la tecnología microarray, es el momento donde la informática, y más en particular la Bioinformática, entra en acción.

2.5. Resumen

En este capítulo se ha realizado una introducción a la bioinformática, presentándose los motivos de su aparición y los objetivos que se plantea. Seguidamente, se ha centrado la labor de la minería de datos en esta nueva ciencia interdisciplinar. Con tal fin, se han clasificado los problemas bioinformáticos, según la aplicación de métodos de aprendizaje, en seis dominios diferentes: genómica, proteómica, microarrays, biología de sistemas, evolución y minería de texto.

Por otro lado, los conceptos biológicos clave en los que se basa esta tesis han sido introducidos. Conceptos como los de gen o expresión genética que son usados

en la descripción de la tecnología de microarray. Finalmente, se describe la matriz de expresión como resultado del usado de técnicas de microarrays y que será la representación de los datos usada en este documento.

Parte II

Estado del Arte

Capítulo 3

Análisis de Datos de Expresión Génica basados en Microarrays

El conocimiento viene, la sabiduría se queda.
ALFRED TENNYSON.

3.1. Introducción

La extracción de información útil a partir de grandes volúmenes de datos, o *data mining*, es una disciplina en la que confluyen técnicas que provienen de una gran variedad de áreas como la estadística, el aprendizaje automático, la inteligencia artificial, la gestión de bases de datos o el reconocimiento de patrones. Desde cada una de estas áreas se aborda el problema del análisis de los datos bajo una perspectiva diferente, siempre con el objetivo de encontrar modelos que expliquen los datos de entrada.

El explosivo crecimiento del ritmo de producción de datos biológicos ha generado una brecha difícil de salvar entre la capacidad de recoger y almacenar los datos, y las limitaciones para analizar y comprender estas grandes colecciones. Como respuesta a dicho problema se comenzaron a usar técnicas de minería de datos para que aportaran algún tipo de conocimiento sobre tal ingente cantidad de información. Poco a poco se fue observando que estas técnicas no eran lo suficientemente específicas como para poder extraer todo el conocimiento subyacente en bases de datos biológicas o biomédicas, con lo que se comenzó a desarrollar técnicas de aprendizaje basadas en el paradigma genético. En la actualidad existen multitud de cuestiones biológicas que pueden ser solucionadas mediante el análisis de datos de expresión génica almacenada en microarrays. Cada cuestión requiere un tipo específico de análisis, donde la dificultad del problema varía de uno a otro [122, 160].

Una de las características fundamentales del análisis de experimentos microarrays es el estudio de la expresión de múltiples genes en paralelo e identificar grupos de genes que muestren un patrón de comportamiento similar. Para descubrir genes con un patrón de expresión similar se requiere dividir el conjunto de datos en subconjuntos según su proximidad. La distancia o similitud puede ser calculada de multitud de formas, dependiendo de la medida de proximidad seleccionada.

Basándonos en las medidas de proximidad entre genes, que serán descritas en el apartado 3.2, éstos pueden ser clasificados según la semejanza de sus expresiones. La clasificación puede ser supervisada o no supervisada. Un ***análisis supervisado***, aunque también engloba técnicas de regresión, suele referirse a la clasificación de los datos en un conjunto de categorías predefinidas denominadas *clase*. Por ejemplo, dependiendo del propósito del experimento, los datos pueden ser clasificados en clases de "enfermo" o "sano". Por el contrario, el ***análisis no supervisado*** no asume clases predefinidas, sino que identifica categorías en los datos según sus patrones de similitud.

Uno de las investigaciones más exitosas en el análisis supervisado es la agrupación de tumores en clases basada en el nivel de expresión [133]. Este tipo de estudios permiten, por ejemplo, clasificar los tipos de tumores, y pueden ser usados como un diagnóstico o una herramienta terapéutica. Dentro de este tipo de análisis existen diferentes aproximaciones: *predicción de clases* [84, 237], que tras la clasificación permiten que nuevos pacientes puedan ser diagnosticados (prognosis); *selección de atributos* [260, 261], cuyo resultado puede ser usado para encontrar genes significativos y descartar entre enfermedades; y el *descubrimiento de clases* [231], que recupera tipos tumores conocidos o encuentra otros nuevos.

Sin embargo, debido a que muchos de los datos biomédicos son obtenidos de manera colateral, sin una hipótesis previa que guíe al experimento, o que involucre genes que por ahora no tienen ninguna función asociada para organismos modelo, el aprendizaje no supervisado está tomando una mayor importancia dentro de la comunidad bioinformática. Aunque el análisis de descubrimiento de clases puede también ser agrupado dentro de la clasificación no supervisada, las técnicas más importantes que se aplican en este análisis son las de clustering, biclustering y redes regualadores de genes, las cuales son expuestas en los apartados 3.3, 3.4 y 3.5, respectivamente.

3.2. Medidas de Proximidad

El primer paso para la clasificación de genes es definir una medida de distancia o de similitud entre genes que convierta la matriz de expresión en una matriz de proximidades.

Seguidamente se expondrán las medidas más relevantes para el cálculo de proximidades de dos genes x e y representados por sus vectores de expresión genética $\vec{x}$ e $\vec{y}$, respectivamente . Para tal exposición se asumirá que el nivel de expresión de cada gen g ha sido medido en n condiciones diferentes (el vector posee n valores) y que g_i denota el nivel de expresión para el gen g en la condición i ($g_i = \vec{g}(i)$).

Nótese que las medidas que se presentan en este apartado miden la proximidad de dos genes atendiendo únicamente a sus valores de expresión y no al conocimiento biológico previo que se tiene de éstos. Actualmente, en el campo de la bioinformática, está teniendo una gran importancia el desarrollo de nuevas medidas que recojan el comportamiento de los genes desde un punto de vista más biológico. Uno de los trabajos que mejor representa esta vertiente es [158], donde Huang et al. presentan una metodología para incorporar el conocimiento de la función genética actual al desarrollo de una nueva medida de distancia. Consecuentemente, existen multitud de trabajos donde se describen nuevas medidas de proximidad basadas en un conocimiento biológico previo, las cuales son descritas detalladamente en la sección 5.4.

3.2.1. Medidas de distancia o disimilitud

Las medidas de distancias son capaces de ponderar la diferencia entre dos genes. Así, valores pequeños implicará que los genes son parecidos, y altos lo contrario.

Dentro de estas medidas podemos destacar las medidas basadas en la distancia de Minkowski o las medidas cuadráticas. Seguidamente son expuestas estas distancias y sus variantes, así como su uso en el campo de la bioinformática.

Familia de distancias basadas en Minkowski

La distancia entre dos genes x e y según Minkowski vendría determinada por la siguiente ecuación:

$$d_{Mikowski}(x, y) = \left(\sum_{i=1}^{N} (x_i - y_i)^p \right)^{1/p} \tag{3.1}$$

Variando p tendríamos diferentes casos particulares, generando una familia de medidas basadas en la ***distancia de Minkowski***. Esta familia de distancia ha sido

usada, por ejemplo, como medida de proximidad en el análisis de datos de expresión genética por Hathaway et al. en [146].

Para $p = 2$ estaríamos en el caso particular de la ***distancia Euclídea*** (ver ecuación 3.4). Esta medida, aunque muy utilizada [316], no es capaz de diferenciar el comportamiento de los distintos genes cuando las variaciones de la expresión de éstos es muy baja, provocando que no sea capaz de identificar fenómenos de desplazamiento y escalado [302]. Este problema ha sido tratado en diferentes trabajos [290, 90, 275], los cuales proponen una estandarización, con media 0 y desviación 1, del vector de expresión de cada gen.

$$d_{Euclidea}(x, y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{3.2}$$

Por otro lado, la ***distancia de Manhattan o City–Block***, es determinada para un valor $p = 1$. Su formulación, presentada en la ecuación 3.3, ha sido usada como medida de proximidad en el campo de la bioinformática en trabajos como el de Carpenter et al. [71].

$$d_{Manhattan}(x, y) = \sum_{i=1}^{n}|x_i - y_i| \tag{3.3}$$

Por último, para un valor de $p = \infty$, tendríamos la distancia de Chebychef, la cual viene determianda por la formula:

$$d_{Chebychef}(x, y) = max_{1\leq i\leq N}|x_i - y_i| \tag{3.4}$$

Medidas cuadráticas

Considerando que $Q = (Q_{ij})$ es una matriz cuadrada de $N \times N$ definida positiva de pesos entre los vectores $\vec{x}$ e $\vec{y}$, la distancia cuadrática entre los genes x e y viene determinada por:

$$d_{Cuadratica}(x, y) = \sqrt{(\vec{x} - \vec{y})^T Q(\vec{x} - \vec{y})} = \sqrt{\sum_{i=1}^{N}\sum_{J=1}^{N}(x_i - y_i)Q_{ij}(x_j - y_j)} \tag{3.5}$$

En el caso concreto en que $Q = V^{-1}$, siendo V^{-1} la matriz de covarianza entre $\vec{x}$ e $\vec{y}$, la medida es conocida con el nombre de ***distancia Mahalanobis***. Esta distancia ha sido usada para el análisis de datos de expresión genética por Anagnostopoulos et al. [12] o por Mao et al. [215].

3.2.2. Medidas de similitud

Estas medidas, al contrario que las expuestas anteriormente, cuantifican el nivel de parecido de dos objetos. Dentro de éstas podemos remarcar las medidas de correlación, llamadas también medidas de separación angular o de producto interno normalizado.

Uno de los ***coeficientes de correlación*** más usados es el ***de Pearson***, el cual mide la similitud existente entre las formas de dos patrones de expresión. En la ecuación 3.6 presentamos la formulación para calcular este coeficiente, donde $\bar{x}$ e $\bar{y}$ representan el valor medio para los vectores $\vec{x}$ e $\vec{y}$, respectivamente.

$$d_{Pearson}(x,y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{3.6}$$

El coeficiente de Pearson es ampliamente usado y bastante efectivo como medida de similitud entre los datos de expresión genética [171, 318]. Sin embargo, estudios empíricos han demostrado que esta medida no es robusta frente a outlier [164], provocando que existan potenciales *falsos positivos* que asignen una alta similitud a pares de patrones no similares. Este hecho generó la aparición de una medida mejorada denominada ***correlación de Jackknife*** [151, 166]. La medida es definida según la ecuación 3.7, donde $d^{(l)}_{Pearson}(x,y)$ representa el coeficiente de correlación de Pearson para los genes x e y sin tener en cuenta la condición situada en la posición l.

$$d_{Jackknife}(x,y) = min\{d^{(1)}_{Pearson}(x,y), \ldots, d^{(l)}_{Pearson}(x,y), \ldots, d^{(N)}_{Pearson}(x,y)\} \tag{3.7}$$

Otro problema que presenta el coeficiente de Pearson es que asume una aproximación de la distribución Gaussiana para los puntos, por lo que no es robusto para distribuciones no Gaussianas [47]. Este problema quedó resuelto con la aparición de la ***correlación de Spearman***, la cual utiliza valores medidos a nivel de una escala ordinal. Es decir, cada valor numérico de expresión g_i es sustituido por la posición que ocupa dicho valor entre el resto. Así, el valor de g_i será modificado, por ejemplo, a tres, si este nivel de expresión es el tercer valor más grande entre los valores de $\vec{g}$. El coeficiente de Spearman, por tanto, no requiere que la distribución Gaussiana sea asumida y es más robusto frente a los outliers. Sin embargo, a consecuencia del ranking de valores, la pérdida de información es su punto débil.

3.3. Técnicas de Clustering

El aprendizaje basado en **clustering** consiste en agrupar los datos en conjuntos disjuntos, llamados *clusters*, de forma que los objetos que estén en un mismo cluster sean muy similares, mientras que aquéllos que no se encuentren en el mismo cluster sean muy diferentes [273, 169].

Esta técnica de aprendizaje es de gran ayuda para entender las funciones genéticas, la regulación de genes, los procesos celulares y los subtipos de células. Los genes con patrones de expresión similar (*genes co-expresados*) pueden ser agrupados juntos en funciones celulares similares. Tal enfoque permite comprender con mayor claridad las funciones de muchos genes desconocidos hasta la fecha [290, 111]. Además, los genes co-expresados en el mismo cluster estarán, probablemente, envueltos en el mismo proceso celular y una fuerte correlación de patrones de expresión entre esos genes indica *co–regulación*.

Actualmente, un microarray contiene de 10^3 a 10^4 genes, mientras que el número de condiciones (experimentos) es, generalmente, menor que 100. Una de las características de los datos de expresión genómica es su utilidad para agrupar tanto genes como condiciones. Por un lado, los genes co-expresados pueden ser agrupados en clusters basados en su patrón de expresión. En tal proceso de clustering, conocido como ***gene-based clustering*** [169, 34, 111], los genes son tratados como los objetos, mientras que los experimentos son los atributos. Por otro lado, los experimentos pueden ser particionados en grupos homogéneos, donde cada grupo corresponde con algún fenotipo macroscópico particular (tipos de cancer [133], síndromes clínicos u otros). Este proceso de clustering es conocido como ***sample-based clustering*** [169, 203, 133], los cuales consideran a los experimentos como los objetos y a los genes como atributos.

En la literatura se tratan ampliamente ambas vertientes de clustering, encontrándose algunos algoritmos, como K-means o enfoques jerárquicos, que pueden ser usados para agrupar genes o condiciones, indistintamente. En cualquier caso, en el campo de la bioinformática, el clustering de genes ha sido el más desarrollado. Por ello, seguidamente se pasa a describir las técnicas más representativas de este tipo de clustering.

3.3.1. Clustering jerárquico

El clustering jerárquico (***Hierarchical Clustering***) [166, 179] se basa en descomponer jerárquicamente el conjunto de datos de entrada, donde la solución es representada por un dendograma (ver figura 3.1). Este tipo de clustering se divide en:

- *Métodos aglomerativos (algoritmos boottom-up)* [111]. Comienzan con ca-

da objeto en un grupo separado. Sucesivamente, estos objetos/grupos se van uniendo hasta que todos se agrupen en uno (el nivel más alto de la jerarquía), o hasta que se cumpla una condición de parada.

- *Métodos divisores (algoritmos top-down)* [8]. Comienzan con todos los objetos en un mismo cluster. En cada iteración, los clusters se separan en clusters más pequeños hasta que, finalmente, cada objeto esté en un cluster o se cumpla la condición de finalización.

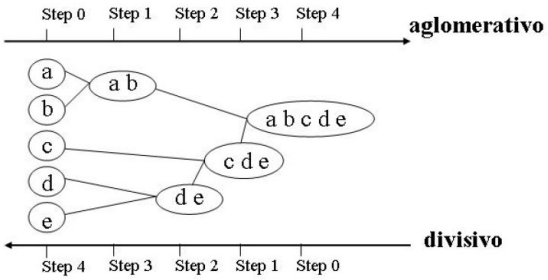


Figura 3.1: Dendograma generado por los dos tipos de estrategias en un procesamiento ficticio.

Algoritmo 3.1 HIERARCHICAL CLUSTERING

INPUT D: matriz de distancias ($D = (d_{ij})$)
OUTPUT C: Clustering jerárquico
begin
1. Encontrar los dos grupos más cercanos, $i - j$, en D, y unir los clusters i y j
2. Modificar D borrando las filas y columnas i, j y añadiendo una nueva fila y columna $i \cup j$, con sus distancias actualizadas
3. Si hay más de un cluster, volver al paso 1.

end

Centrándonos en los algoritmos de clustering jerárquico aglomerativo, el método de clustering más antiguo y popular es el propuesto por Cormack en 1971 [79]. Éste comienza de una partición inicial de los datos en cluster unitarios, a los que se le aplica una etapa de combinación hasta que todos los elementos pertenezcan al mismo cluster. Cada etapa de unión corresponde a la unión de dos clusters. El esquema general para una técnica de clustering de estas características es presentado en el algoritmo 3.1. En el esquema se pueden destacar dos pasos que generan diferentes vertientes de esta técnica de clustering jerárquica. Por un lado, según sea actualizada la matriz de distancias (paso 2 del algoritmo 3.1) tendríamos diferentes aproximaciones para algoritmos aglomerativos [167]. Por otro, la selección de los dos grupos más cercanos (paso 1 del algoritmo 3.1) pueden generar diferentes vertientes, dentro de las que se destacan [226]: *Single–Link*, *Complete–Link*,

Average–Link y *Centroide*. Éstas consideran la distancia más pequeña, más grande o distancia media entre todos los elementos de los grupos para las tres primeras aproximaciones, y la distancia entre los centroides (punto medio) de los grupos para la aproximación Centroide.

Las técnicas aglomerativas presentan un problema de escalabilidad debido a su coste computacional $O(n^3)$, siendo n el número de elementos a tratar. El coste del cálculo de la matriz de similitud es de $O(n^2)$. Tras este paso hay $n-1$ iteraciones que envuelven a los descritos en el algoritmo 3.1. Si suponemos una búsqueda lineal para realizar la búsqueda en la matriz de distancias, entonces para la iteración i-ésima, el primer paso requiere un tiempo $O((n-i+1)^2)$. El segundo paso, supondría un tiempo de $O(n-i+1)$. Si ninguna modificación, el tiempo de computacional final sería $O(n^3)$, pero si suponemos que las distancias de cada cluster al resto se encuentran almacenadas en una lista ordenada, el tiempo sería reducido a $O(n^2 \log n)$.

3.3.2. Clustering basado en particiones (K–Means)

Los métodos de clustering basados en particiones, como su propio nombre indica, se basan en particionar el conjunto de datos de entrada, donde cada partición representa un grupo o cluster.

La técnica de clustering más representativa dentro de este grupo es el algoritmo K-medias ***(k–means)*** [25, 210]. Éste supone que el número de cluster K es conocido, y consiste en minimizar las distancias entre los elementos y el centroide[1] del cluster al que pertenecen.

Sea M una matriz de expresión $n \times m$, con n genes y m experimentos. Una partición P de elementos en $\{1, ..., k\}$, indicando por $P(i)$ el cluster asignado al gen i, y por $c(j)$ el centroide del cluster j. Sea $d(g_1, g_2)$ una medida de distancia entre los genes g_1 y g_2 (ver apartado 3.2.1). K–means intenta encontrar una partición P que minimice el error E_p establecido por la siguiente función: $E_p = \sum_{i=1}^{n} d(i, c(P(i)))$. Esta idea se encuentra reflejada en el algoritmo 3.2, donde cada iteración de k–means modifica la partición actual, comprobando todas las posibles modificaciones de la solución. En éste, un elemento es movido a otro cluster si dicho cambio reduce la función de error. La complejidad de este algoritmo es de $O(n^{mk+1} log(n))$, donde k es el número de cluster, n el número de entidades a agrupar (genes) y m las dimensión del espacio a tratar (experimentos).

El algoritmo de aprendizaje K–medias ha sido utilizado ampliamente en el campo de la Minería de Datos por su simplicidad y rapidez. Estudios empíricos realizados en [169] muestran que k–means suele converger en un número peque-

[1] Vector representante del cluster, calculado como la media de los elementos pertenecientes al cluster

Algoritmo 3.2 K–MEANS

INPUT M: matriz de expresión (*Genes*,*Condiciones*)
k: número de cluster
OUTPUT *P*: Partición en k cluster
begin
1. Comenzar con una partición aleatoria P de M en k clusters
2. Para cada elemento i y el cluster $j \neq P(i)$, siendo $E_p^{i,j}$ el coste de una solución en donde i se ha movido al cluster
2.1. **Si** $E_p^{i*,j*} = min_{ij} E_p^{i,j} < E_p$ **entonces** mover $i*$ al cluster $j*$ y repetir el paso 2
2.2. **En otro caso** Paro
end

ño de iteraciones. Sin embargo, presenta diferentes inconvenientes como algoritmo de clustering genético. Primero, el número de clusters no suele conocerse a priori. Para detectar el número óptimo de éstos, el algoritmo deberá ser ejecutado repetidamente con diferentes valores de k, comparando los resultados obtenidos. Segundo, los datos de expresión genética suelen contener una enorme cantidad de ruido; sin embargo, k–means fuerza a cada gen a ser incorporado en un cluster, lo cual hace que el algoritmo sea sensible al ruido [90, 276]. Para solventar esta problemática se han presentado diferentes estudios [150, 186, 90], los cuales suelen usar algún parámetro global para controlar la *calidad* de los clusters resultantes (e.g., el radio máximo de un cluster y/o la distancia mínima entre clusters).

3.3.3. Self Organizing Map

Los mapas autoorganizados o **SOM** (*Self-Organizing Map*) [186] , también llamados redes de Kohonen, son un tipo de red neuronal no supervisada competitiva [328]. Este tipo de red está distribuida de forma regular en una rejilla de, normalmente, dos dimensiones y tiene como fin el de descubrir la estructura subyacente de los datos introducidos en ella.

Los SOMs son construidos seleccionando, primeramente, una topología de "nodos", por ejemplo una rejilla 3×2. Seguidamente, los nodos son mapeados a un espacio k–dimensional, incialmente aleaotrio y que es ajustado iterativamente. En cada iteración es seleccionado un dato P de manera aleatoria y se mueven los nodos en dirección de P. El nodo más cercano a P (N_P) es el más movido, mientras que los otros son modificados de menor manera dependiendo de su distancia a N_P en la geometría inicial. De esta forma, los puntos vecinos en la geometría inicial tiende a ser mapeados a puntos vecinos en el espacio k–dimensional.

Tamayo et al [283] fueron los pioneros en usar este tipo de mapas para el clustering de datos de expresión génica. Ellos consideraban que los SOMs tienen unas características que le hacen especialmente apropiadas para este tipo de estudios; permiten estructuras parciales en los clusters (en contraste con la rigidez del clus-

tering jerárquico o la destructuración de K–medias) y proporcionan una fácil visualización e interpretación. Con tal fin, estos autores propusieron GeneCluster para generar y mostrar SOMs de datos de expresión génica.

Algoritmo 3.3 SELF ORGANIZING MAP

INPUT D: matriz de expresión ($V = (v_{ij})$)
OUTPUT C: conjunto de clusters
Establecer arbitrariamente los vectores referencia $f_1(v) \in \Re^k$ para cada nodo v
for $i = 1$ hasta que la localización de los nodos no cambie en más de ϵ **do**
Seleccionar aleatoriamente un elemento P
Calcular el nodo N_P con los vectores referencias $f(N_P)$ más cercanos a p
Actualizar todos los vectores referencias: $f_{i+1}(N) = f_i(N) + \tau(d(N, N_P), i)(P - f_i(N))$
end for
Asignar cada elemento a un cluster con el vector referencia más cercano

En el algorimto 3.3 se describe el proceso con el que Tamayo et al. hacen uso de SOM para el estudio de datos génicos. El algoritmo presenta un coste computacional de $O(n^3)$, donde n es el número de patrones a agrupar (genes en este caso) y se basa en la idea de que un SOM tiene un conjunto de nodos con una topología simple (por ejemplo, una matriz bidimensional) y una función de distancia $d(N_1, N_2)$. Los nodos son iterativamente mapeados en un espacio "de expresión génica" k–dimensional, donde la coordenada de posición i representa el nivel de expresión para el ejemplo i. La posición del nodo N en la iteración i es denotado por $f_i(N)$. El mapeo inicial f_0 es aleatorio. En las iteraciones posteriores, se selecciona un punto P de forma aleatoria y se identifica el punto más cercano a P (N_P). El mapeo de los nodos es ajustado moviendo los puntos hacia P según la fórmula:

$$f_{i+1}(N) = f_i(N) + \tau(d(N, N_P), i)(P - f_i(N))$$

τ representa el rango de aprendizaje, el cual decrece con la distancia del nodo N a N_P y con el número de iteraciones. Concretamente, esta función viene determina da por:

$$\tau(x, i) = \begin{cases} 0{,}02T/(T + 100i), & \text{x=p(i);} \\ 0, & \text{en otro caso.} \end{cases}$$

Siendo T el número máximo de iteraciones y el radio $p(i)$ decreciente linealmente con i ($p(0) = 3$).

Por otro lado, además de GenCluster, existen otras propuestas que usan SOM para agrupar datos de expresión genética, destacando el trabajo presentado por Toronen et al.[294].

3.3.4. Clustering basado en teorías de grafos

Las técnicas de clustering basadas en teoría de grafos (***graph–theoretical approaches***) [34] son presentadas explícitamente en términos de grafos, de este modo el problema de clustering se traduce en un problema de teoría de grafos basado en encontrar el corte mínimo o "clique" (grafo conexo) máximo en el grafo de proximidad G.

Dado un conjunto de datos X, se puede construir una *matriz de similitud* P, donde $P[i, j] = proximidad(O_i, O_j)$, y un grafo ponderado $G(V, E)$, denominado *grafo de similitud*, donde cada elemento del conjunto de datos corresponde con un vértice. Para algunos métodos de clustering, cada par de objetos se conecta por una arista con un peso acorde al valor de proximidad entre los objetos [275, 313]. Para otros métodos, la proximidad es expresada sólo como 0 ó 1 según un umbral, y las aristas sólo existen entre dos objetos i y j si $P[i, j] = 1$ [145].

Seguidamente se presentan las técnicas de clustering más representativas basadas en grafos:

CLICK (CLuster identification via Connectivity Kernels)[274, 275][2] se basa en un modelo estadístico. El modelo da un significado probabilístico a los pesos de las aristas en el grafo de similitud y al criterio de parada. La principal suposición probabilística de CLICK es que el valor de similitud entre parejas de elementos sigue una distribución normal: los valores de similitud entre *mates*[3] sigue una distribución normal con media μ_T y varianza σ_T^2, mientras que los valores de similitud entre *no-mates* siguen una distribución normal con media μ_F y varianza σ_F^2, donde $\mu_T > \mu_F$. Esta suposición se cumple en datos reales, y puede ser asintóticamente justificada [275].

El algoritmo usa tanto los valores μ_T, μ_F, σ_T y σ_F, como la probabilidad p_{mates} de elegir dos elementos que sean *mates* aleatoriamente. Estos parámetros pueden ser calculados directamente a partir de una solución conocida o estimados usando el algoritmo de clustering EM (*Expectation Maximization*), asumiendo el modelo probabilístico descrito anteriormente para valores de similitud (ver, por ejemplo, [223], sección 3.2.7).

Sea S la matriz de similitud del conjunto de datos de entrada M (matriz de expresión $n \times m$), donde S_{ij} es el producto escalar entre los genes i y j, i.e., $S_{ij} = \sum_{k=1}^{m} M_{ik} M_{jk}$. CLICK representa los datos de entrada como un grafo de similitud ponderado $G = (V, E)$, siendo $V(G)$ los vértices del grafo G. En este grafo los vértices corresponden a los genes y el peso de las aristas es derivado de los valores de similitud. El peso w_{ij} de la arista (i, j) refleja la

[2] disponible en `http://ww.math.tau.ac.il/ rshamir/click/click.html`

[3] Elementos que pertenecen al mismo cluster

probabilidad de que i y j sean *mates*, y es calculado a partir de la función de similitud de la normal $f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ y del Teorema de Bayes:

$$
\begin{aligned}
w_{ij} &= log\left(\frac{p_{mates} f(S_{ij}|i, j \texttt{ son mates})}{(1 - p_{mates}) f(S_{ij}|i, j \texttt{ son no mates})}\right) \\
&= log\left(\frac{p_{mates}\sigma_F}{(1 - p_{mates})\sigma_T} + \frac{(S_{ij} - \mu_F)^2}{2\sigma_F^2} - \frac{(S_{ij} - \mu_T)^2}{2\sigma_T^2}\right)
\end{aligned}
$$

El algoritmo clásico de CLICK usa el esquema recursivo presentado en el algoritmo 3.4 para formar grupos o kernels. En cada paso el algoritmo maneja las componentes conectadas del subgrafo inducido por elementos no agrupados. Si el componente sólo contiene un vértice (v), entonces este vértice es considerado como *singleton* y es tratado separadamente. De lo contrario, se comprueba un criterio de parada (que será descrito después). Si el componente satisface tal criterio, es declarado como *kernel*. En otro caso, el componente es dividido según un peso de corte mínimo. La salida del algoritmo es una lista de kernels que sirven de base para clusters eventuales. En el algoritmo 3.4 se asume que $MinWeightCut(G)$ calcula el peso de corte mínimo del grafo G y devuelve una partición de G en dos subgrafos H y $\overline{H}$ según este corte.

Algoritmo 3.4 CLICK–BÁSICO

INPUT G: Grafo de similitud
OUTPUT R: Conjunto de singletons
L: conjunto de kernels
begin
  if $V(G) = v$ **then**
    $R = R \cup v$
  else if G es un Kernel **then**
    $L = L \cup V(G)$
  else
    $(H, \bar{H}) \leftarrow CortePesoMinimo(G)$
    CLICK–BASICO(H)
    CLICK–BASICO($\bar{H}$)
  end if
end

Actualmente, el algoritmo básico de CLICK omite todos los vertices del grafo con valor menor a un umbral predefinido no negativo, realizando el corte mínimo sobre dicho grafo simplificado, y corrigiendo la solución para las aristas perdidas.

La complejidad de este algoritmo viene determinado, en gran medida, por el algoritmo de corte mínimo. Con tal fin, Sharan y Shamir usaron una su

propia implementación basada en el algoritmo de Hao y Horlin [143] para conjuntos de entrada inferiores a 1000 elementos y que tiene un coste de $O(n^2\sqrt{m})$. Para tamaños superiores usaron una aproximación basada en el algoritmo de Dinic [114], que presenta un coste de $O(nm^{2/3})$.

CAST Ben–Dor et al.[34] desarrollaron un algoritmo polinomial bajo el siguiente modelo estocástico de los datos: la estructura esencial del cluster correcto se presenta por un grafo de uniones disjuntas de cliques (grafos conexos), y los errores son posteriormente introducidos en el grafo, eliminando y añadiendo aristas entre pares de vertices con una probabilidad α. Si todos los clusters tienes al menos un tamaño cn, para alguna constante $c > 0$, el algoritmo resuelve el problema con la precisión deseada con una alta probabilidad.

CAST usa como entrada la matriz de similitud S del conjunto de datos de entrada $M(n \times m)$. La *afinidad* de un elemento v a un cluster C es $a(v) = \Sigma_{i \in C} S_{iv}$). El algoritmo polinomial motivaba el uso de la afinidad para desarrollar una heurística más rápida denominada CAST (Clustering Affinity Search Technique) [34]. Este algoritmo usa un único parámetro t y genera los clusters uno a uno. El siguiente cluster comienza con un único elemento, y los elementos son añadidos o borrados del cluster si su afinidad relativa es mayor o menor que t, respectivamente, hasta que el proceso se estabilice. CAST es mostrado en el algoritmo 3.5 ($O(n^2)$), el cual finaliza con una heurística adicional (*cluster cleaning*) que consiste en comprobar que cada elemento está en el cluster con el que tiene mayor afinidad.

Algoritmo 3.5 CAST

INPUT S: Matriz de similitud
OUTPUT C: Conjunto de clusters
begin
  while haya elementos no agrupados **do**
    Selecciona un elemento no agrupado para empezar un nuevo cluster C
    while No ocurran cambios **do**
      ADD: añade un elemento no agrupado v con una afinidad máxima a C si $a(v) > t|C|$
      REMOVE: elimina un elemento u de C con afinidad mínima si $a(u) \leq t|C|$
    end while
    Añade C a la lista final de clusters
  end while
end

El gran inconveniente de esta técnica es su dependencia al parámetro t. Bellaachia et al. solventaron este problema al proponer en [32] que el umbral fuera calculado dinámicamente basándose sólo en los elementos que aún no han sido asignados a ningún cluster (U'). Así, el parámetro t es computado antes de que cada cluster sea deducido, mientras que el umbral dinámico es

calculado atendiendo a la similitud de los nodos que quedan por ser agrupados:

$$T = \frac{\sum_{i,j \in U' \wedge S(i,j) \geq 0,5} S(i,j) - 0,5}{|u : u \in U' \wedge a(u) \geq 0,5|} + 0,5$$

3.4. Técnicas de Bi–Clustering

Tanto las técnicas de clustering basada en genes como las basadas en ejemplos, buscan exclusivamente y exhaustivamente particiones de objetos que compartan el mismo espacio de atributos. Sin embargo, estudios actuales en biología molecular han demostrado que sólo un pequeño subconjunto de genes participan en cualquier proceso celular de interés y que un proceso celular tiene lugar sólo en un subconjunto de ejemplos [169, 285]. Por tanto, un gen puede estar co-expresado con otro en un cierto número de experimentos pero no en todos.

Supongamos que tenemos un gen g_1 que se co-expresa con otro g_2 en cinco experimentos de ocho experimentos totales, y con g_3 en los tres restantes. Una técnica de clustering no podría unir g_1 con g_2 y a su vez con g_3, puesto que no comparten el mismo patrón de comportamiento en todos los experimentos en cuestión. Por tanto, podríamos afirmar que, en este caso, el clustering no es un método aprendizaje lo suficientemente potente, puesto que diría que los tres genes no están relaciones, aunque biológicamente sí que lo estén: g_1 con el g_2 y g_1 con g_3.

Las técnicas de ***biclustering*** contemplan este tipo de situaciones, y consisten, al igual que el clustering, en agrupar los datos en conjuntos. La gran diferencia radica en que los genes y los experimentos son tratados simétricamente, es decir, tanto los genes como los experimentos pueden ser considerados como objetos o como atributos. Es más, los grupos generados (*bicluster*) pueden tener diferentes espacios de características, siendo posible encontrarnos bicluster con un cierto número de genes y experimentos, y otros con diferentes genes/experimentos o número de ellos. Además, los bicluster, al contrario que los cluster, pueden solaparse [211].

En resumen, un bicluster puede entenderse como una submatriz con un conjunto de genes y otro de atributos, donde no hay ninguna limitación a-priori en la organización de biclusters y, en particular, los genes o condiciones pueden ser parte de más de un bicluster o de ninguno. La carencia de una limitación estructural aporta una mayor libertad pero provocan, por consiguiente, que los biclusters sean más vulnerable al sobreajuste. Por ello, los algoritmos de biclustering deben garantizar que los biclusters obtenidos sean significativos.

Seguidamente se presentan algunos de los modelos y algoritmos de biclustering más representativos [285] desarrollados para el análisis de expresión génica.

Tabla 3.1: Matriz de datos de expresión génica.

	Condición 1	...	Condición j	...	Condición m
Gen 1	a_{11}	...	a_{1j}	...	a_{1m}
Gen ...	...	...	...	...	...
Gen i	a_{i1}	...	a_{ij}	...	a_{im}
Gen ...	...	...	...	...	...
Gen n	a_{n1}	...	a_{nj}	...	a_{nm}

3.4.1. Definiciones, notaciones y formulación del problema

A lo largo de la siguiente exposición de técnicas de biclustering se asume que se tiene una matriz de expresión de genes n por m, donde cada elemento a_{ij} será, en general, un valor real y representará el nivel de expresión del gen i bajo la condición j. En la tabla 3.1 se ilustra esta nomenclatura.

Dada la matriz A un *cluster de filas* es un subconjunto de filas que exhiben el mismo comportamiento en todas las columnas. El cluster de filas $A_{IY} = (I, Y)$ es un subconjunto de filas definidas sobre el conjunto total de columnas Y. Donde $I = \{i_1, ..., i_k\}$ es un subconjunto de filas ($I \subseteq X$ y $k \leq n$), siendo X el conjunto de todas las filas de la matriz original. Por tanto, éste puede ser definido como una submatriz $k \times m$ de la matriz de datos A. Similarmente, un *cluster de columnas* es un subconjunto de columnas que exhiben un comportamiento similar a lo largo de todas las filas. Un cluster $A_{XJ} = (X, J)$ es un subconjunto de columnas definidas sobre todas las filas X, donde $J = \{j_1, ..., j_s\}$ es un subconjunto de columnas ($J \subseteq Y$ y $s \leq m$), siendo Y el conjunto de todas las columnas de la matriz original. Con lo que un cluster de columnas (X, J) puede ser definido como una submatriz de $n \times s$ de la matriz de datos A.

Un *bicluster* es un subconjunto de filas que exhiben el mismo comportamiento en un subconjunto de columnas, y viceversa. El bicluster $A_{IJ} = (I, J)$ es un subconjunto de filas y columnas donde $I = \{i_1, ..., i_k\}$ es un subconjunto de filas ($I \subseteq X$ y $k \leq n$), y $J = \{j_1, ..., j_s\}$ es un subconjunto de columnas ($J \subseteq Y$ y $s \leq m$). Por tanto, un bicluster (I, J) puede ser definido como una submatriz $k \times s$ de la matriz de datos A.

El problema de biclustering, dada una matriz A, consiste en identificar un conjunto de biclusters $B_k = (I_k, J_k)$ tal que cada bicluster B_k satisfaga algún criterio específico de homogeneidad (función objetivo). Debido a que cada una de las técnicas de biclustering que se expondrán utiliza un criterio propio, la función objetivo será descrita como parte de la descripción de cada uno de los métodos.

Para ello, se denotará por a_{iJ} a la media de fila i–ésima en el bicluster, a_{Ij} la media de la columna j–ésima en el bicluster y a_{IJ} a la media de todos los elementos del bicluster. Estos valores son definidos por:

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}$$

$$a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}$$

$$a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij} = \frac{1}{|I|} \sum_{i \in I} a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{Ij}$$

3.4.2. Algoritmo de Cheng y Church

Cheng y Church fueron los primeros en introducir el biclustering para el análisis de expresión de genes [76]. Ellos definieron un bicluster como un conjunto de filas y uno de columnas con una alta similitud. El valor de similitud introducido y denominado ***residuo cuadrado medio***, H, fue usado como medida de coherencia de las filas y las columnas del bicluster. Dado una matriz de datos $A = (X, Y)$ un bicluster era definido como una submatriz uniforme (I, J) con un residuo bajo. Una submatriz (I, J) se considera un δ–bicluster si $H(I, J) < \delta$ para algún $\delta \geq 0$. En particular, ellos perseguían encontrar bicluster grandes con una puntuación menor que un cierto umbral δ.

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

Filas Constantes

1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

Columnas Constantes

1	2	5	0
2	3	6	1
4	5	8	3
5	6	9	4

Valores coherentes

Figura 3.2: Ejemplo de Biclusters perfectos.

En un δ–bicluster *perfecto* cada fila/columna o ambas, filas y columnas, exhibe una coherente tendencia absoluta ($\delta = 0$) (los biclusters mostrados en la figura 3.2 son ejemplos de este tipo de biclusters perfectos). Esto significa que el valor de cada fila o columna puede ser generado desplazando los valores de otras filas o columnas por un valor común. Cuando éste es el caso, $\delta = 0$ y cada elemento a_{ij} puede ser definido únicamente por la media de su fila, a_{iJ}, la media de su columna, a_{Ij}, y la media del bicluster, a_{IJ}. La diferencia $a_{Ij} - a_{iJ}$ es la influencia relativa de la columna j con respecto a las otras columnas en el δ–bicluster. El mismo razonamiento aplicado a las filas nos conduce a la definición que, en un δ–bicluster perfecto, el valor de un elemento, a_{ij}, es dado por una fila, una columna y un valor constantes:

$$a_{ij} = a_{iJ} + a_{Ij} - a_{IJ}$$

Desafortunadamente, debido al ruido de los datos, los δ–biclusters pueden no

ser siempre perfectos. El concepto de *residuo* es, de este modo, introducido para cuantificar la diferencia entre el valor actual de un elemento a_{ij} y su valor predicho esperado procedente de la media de la fila, columna y bicluster correspondientes.

El residuo de un elemento a_{ij} en el bicluster (I, J) es definido como sigue:

$$r(a_{ij}) = a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}$$

Asumiendo la posibilidad de que exista un residuo, el valor de un elemento a_{ij} perteneciente a un bicluster no–perfecto sería:

$$a_{ij} = r(a_{ij}) + a_{iJ} + a_{ij} - a_{IJ}$$

El valor del residuo es, por tanto, un indicador de la coherencia de un valor relativo a los valores restantes en el bicluster dada la relevancia de las filas y columnas. A menor residuo, mayor coherencia. Para valorar la calidad global de un δ–bicluster se usa el *residuo cuadrado medio* (MSR), H, de un bicluster (I, J) como la suma de los residuos cuadrados:

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} r(a_{ij})^2$$

donde $H(I, J) = 0$ indica que los valores de la matriz fluctúan al unísono.

El algoritmo de Cheng y Church, resumido en 3.6, toma como entrada la matriz original de datos ($A = (X, Y)$; X conjunto de genes, Y conjunto de condiciones) y un valor umbral de residuo. Éste tiene un coste computacional de $O(MN \times (M+N) \times k)$ para descubrir k biclusters, y puede ser entendido como un algoritmo de búsqueda local. En él se pueden diferenciar dos fases principales. La primera de borrado, donde el algoritmo elimina filas y columnas de la matriz original; y la segunda, donde se lleva cabo una adición.

En la fase borrado, donde la submatriz en construcción consta de I filas y J columnas, el algoritmo examina el conjunto de posibles movimientos. Para cada fila, se calcula el valor de su residuo, $d(i) = \frac{1}{|J|} \sum_{j \in J} r(e_{ij})$, y lo mismo para cada columna, $d(j) = \frac{1}{|I|} \sum_{i \in I} r(e_{ij})$. En el siguiente paso se selecciona la fila o columna con mayor valor de residuo y eliminarla de la submatriz. El proceso finaliza cuando el residuo de la submatriz es H(I,J)$< \delta$.

En la segunda fase del algoritmo, son añadidas filas y columnas a la submatriz resultante de la anterior fase. Con tal fin se sigue un esquema similar al anterior, pero esta vez buscando las filas y columnas de la matriz original con menores valores de residuo. El proceso finaliza cuando un posible aumento del tamaño de la submatriz hace que el valor de su residuo MSR supere el umbral fijado como parámetro de entrada. Finalmente, el resultado obtenido es la submatriz máxima

Algoritmo 3.6 CHENG Y CHURCH

INPUT A: Matriz de expresión génica (A=(X,Y))
δ: umbral (residuo cuadrado medio máximo)
OUTPUT I,J: conjunto de filas y columnas, respectivamente
begin
Inicializar un bicluster (I, J) con $I = X, J = Y$

Fase de Borrado
while $H(I, J) > \delta$ **do**
Calcular para cada $i \in I$, $d(i) = \frac{1}{|J|} \sum_{j \in J} r(e_{ij})$
Calcular para cada $j \in J$, $d(j) = \frac{1}{|I|} \sum_{i \in I} r(e_{ij})$
if $max_{i \in I} d(i) > max_{j \in J} e(j)$ **then**
$I = I \backslash argmax_i(d(i))$
else
$J = J \backslash argmax_j(d(j))$
end if
end while

Fase de Adición
Asignar $I' = I, J' = J$
while $H(I', J') < \delta$ **do**
Asignar $I' = I, J' = J$
Calcular para cada $i \in X \backslash I$, $d(i) = \frac{1}{|J|} \sum_{j \in J} r(e_{ij})$
Calcular para cada $j \in Y \backslash J$, $d(j) = \frac{1}{|I|} \sum_{i \in I} r(e_{ij})$
if $max_{i \in I} d(i) < max_{j \in J} e(j)$ **then**
$I' = I \cup argmax_i(d(i))$
else
$J' = J \cup argmax_j(d(j))$
end if
end while
end

que cumple la condición de no sobrepasar el valor umbral del MSR.

La medida de residuo cuadrado medio usada en este algoritmo asume que no hay valores omisos en la matriz de datos. Para garantizar esta precondición, Cheng y Church sustituyeron dichos valores por números aleatorios durante la etapa de preprocesado.

Yang et al. [318, 319] generalizaron la definición de un δ–bicluster, solventando el problema generado por los valores omisos y la interferencias causados por el relleno aleatorio de Cheng y Church. Ellos definieron un δ–bicluster como un subconjunto de filas y uno de columnas que exhiben valores coherentes en valores específicos (no omisos) de las filas y columnas consideradas. Esta aproximación es conocida como FLOC, FLexible Overlapped biClustering.

El verdadero problema del algoritmo propuesto por Cheng y Church, y de todas sus modificaciones, es que el residuo depende sólo de la varianza de escalado (scaling) y no de la varianza del desplazamiento (shifting). Esto fue probado matemáticamente por Aguilar en [4], donde afirmó que era un factor crítico, ya que una alta varianza de escalado nos conduciría a perder aquellos biclusters que proveen un residuo cuadrado medio mayor que δ.

3.4.3. Coupled Two-way Clustering

Coupled two-way clustering (CTWC) o "emparejamiento de clustering en dos dimensiones", introducido por Getz et al. en [129], define un esquema genérico para transformar un algoritmo de clustering unidimensional en uno de biclustering. El algoritmo depende de un algoritmo de clustering unidimensional (estándar) que pueda descubrir clusters significativos (denominados *estable* en [129]). CTWC aplicará recursivamente el algoritmo unidimensional a submatrices, consistiendo en encontrar subconjuntos de genes que aumenten los clusters significativos de condiciones y, similarmente, encontrar subconjuntos de condiciones que aumenten los clusters significativos de genes. Las submatrices definidas por tal emparejamiento son denominadas ***submatrices estables*** y corresponden a un bicluster. El proceso, el cual es mostrado en el algoritmo 3.7 con un coste exponencial, opera sobre un conjunto de genes $\mathcal{X}$ y uno de condiciones $\mathcal{Y}$. Inicialmente $\mathcal{X}_1 = \{X\}$, $\mathcal{Y}_\infty = \{Y\}$; y $\mathcal{X} = \emptyset$, $\mathcal{Y} = \emptyset$. Seguidamente, el algoritmo selecciona iterativamente un subconjunto de genes $X' \in \mathcal{X}$ y uno de condiciones $Y' \in \mathcal{Y}$ y aplica el algoritmo de clustering unidimensional dos veces, agrupando X' y Y' en la submatriz $X' \times Y'$ ($A_{X',Y'}$). Si se detectan clusters estables, sus conjuntos de genes/condiciones son añadidos al conjunto respectivo $\mathcal{X}$, $\mathcal{Y}$. Este proceso se repite hasta que no se encuentren nuevos clusters estables. La implementación se asegura que cada pareja de subconjuntos no se encuentra más de una vez.

Nótese que el éxito de esta técnica depende de las prestaciones del algoritmo

Algoritmo 3.7 COUPLED TWO–WAY CLUSTERING

INPUT A: Matriz de expresión génica (A=(X,Y))
ALG: Algoritmo de clustering unidimensional. Recibe como entrada una matriz y genera clusters significativos (estables) de filas y columnas
OUTPUT $\mathcal{X}, \mathcal{Y}$: conjunto de filas y columnas, respectivamente
begin
Inicializar una tabla de pesos
Inicializar $\mathcal{X}_1 = \{X\}$, $\mathcal{Y}_1 = \{Y\}$
Inicializar $\mathcal{X} = \emptyset$, $\mathcal{Y} = \emptyset$
Inicializar los conjuntos de tablas jerárquicas H_X almacenando para clusters de genes los subconjuntos de condiciones para generarlos
Inicializar los conjuntos de tablas jerárquicas H_Y almacenando para cada clusters de condiciones los subconjuntos de genes usados para generarlos
while $\mathcal{X}_1 \neq \emptyset \vee \mathcal{Y}_1 \neq \emptyset$ **do**
Inicializar $\mathcal{X}_2 = \emptyset$, $\mathcal{Y}_2 = \emptyset$
for all $(X', Y') \in (\mathcal{X}_1 \times \mathcal{Y}_1) \cup (\mathcal{X}_1 \times \mathcal{Y}) \cup (\mathcal{X} \times \mathcal{Y}_2)$ **do**
Ejecuta $ALG(A_{X',Y'})$ para agrupar los genes en X':
Añade los conjuntos de genes estables a $\mathcal{X}_2$
Ejecuta $ALG(A_{X',Y'})$ para agrupar las condiciones en Y':
Añade los conjuntos de condiciones estables a $\mathcal{Y}_2$
end for
Asignar $\mathcal{X} = \mathcal{X} \cup \mathcal{X}_1$, $\mathcal{Y} = \mathcal{Y} \cup \mathcal{Y}_1$
Asignar $\mathcal{X}_1 = \mathcal{X}_2$, $\mathcal{Y}_1 = \mathcal{Y}_2$
end while
end

de clustering unidimensional elegido. Muchos de los algoritmos de clustering más populares, vistos en el apartado 3.3 (e.g. K–means, Hierarchical, SOM), no pueden ser utilizados "tal cual" en el mecanismo de CTWC, ya que éstos no distinguen clusters significativos de los que no los son o, hacen a priori, una suposición en el número de clusters.

3.4.4. Algoritmo de signatura iterativa

En el algoritmo de signatura iterativa (ISA, Iterative Signature Algorithm)[163, 41] el concepto de bicluster significativo se define intrínsecamente en los bicluster de genes y condiciones – las condiciones de un bicluster definen únicamente los genes y viceversa. La idea es que los genes de un bicluster estén co–regulados y, así, para cada condición la media de expresión génica sobre todo los genes del bicluster deberían ser extremas (excepcionalmente grandes o pequeñas) y para cada gen la media de expresión génica sobre todas las condiciones del bicluster sean sorprendentes (inusualmente grandes o pequeñas). Esta idea se formaliza usando un modelo lineal simple para cada expresión génica, asumiendo que los niveles de expresión siguen una distribución normal para cada gen o condición como se muestra seguidamente.

El proceso, representado en el algoritmo 3.8, usa dos copias normalizadas de

la matriz de expresividad original. La matriz E^G tiene normalizadas las filas con media 0 y varianza 1, y E^C con las columnas normalizadas de la misma forma. Se denota $e^G_{uV'}$ como el nivel de expresión medio de los genes de V' en la condición u y $e^C_{U'v}$ al nivel medio de expresión para el gen v en las condiciones de U'. Un bicluster $B = (U', V')$ debe tener:

Algoritmo 3.8 ISA

INPUT E: Matriz de expresión génica
V: conjunto de genes
U: conjunto de condiciones
V_{in}: Conjunto inicial de genes
T_G, T_C: umbrales de genes y condiciones, respectivamente
m, ϵ: criterios de parada
OUTPUT U', V': conjunto de filas y columnas, respectivamente
begin
Construir una matriz de columnas estandarizadas E^C
Construir una matriz de filas estandarizadas E^G
Inicializar los contadores $n = 0, n' = 0$
Inicializar el conjunto de genes actuales $V' = V_{in}$
Inicializar un conjunto vacío de condiciones U'
while $n - n' < m$ **do**
Calcular $e^C_{uV'} = \frac{1}{|V'|} \sum_{v \in V'} e^C_{uv}$ para $u \in U$
$U' = \{u \in U : |e^C_{uV'}| > \frac{T_C}{\sqrt{|V'|}}\}$
Calcular $e^G_{U'v} = \frac{1}{|U'|} \sum_{u \in U'} e^G_{uv}$ para $v \in V$
$V'' = V'$
$V' = \{v \in V : |e_{U'v}| > \frac{T_G}{\sqrt{|U'|}}\}$
if $\frac{|V' \setminus V''|}{V' \cup V''} < \epsilon$ **then**
$n' = n$
end if
$n = n + 1$
end while
end

$$U' = \{u \in U : |e^C_{uV'}| > T_C \alpha_C\}, V' = \{v \in V : |e^G_{U'v}| > T_G \alpha_G\}$$

donde T_G es un parámetro de entrada que define un umbral y α_G es la desviación estándar de las medias $e^G_{U'v}$ donde todos los posibles rangos de genes y U' son fijados. Similarmente, T_C, α_C son los parámetros correspondientes al conjunto de columnas V'. La idea es que si los genes de V' están regulados ascendente o descendentemente en las condiciones U', entonces su expresión media debe ser significativamente diferente a sus valores esperados en las matrices aleatorias (la cual es 0, ya que la matriz está estandarizada). Un argumento similar puede ser aplicado para las condiciones U'. La desviación estándar puede ser calculada como $\frac{1}{\sqrt{|U'|}}, \frac{1}{\sqrt{|V'|}}$.

En resumen, el algoritmo comienza con un conjunto arbitrario de genes $V_0 = V_{in}$, el cual puede ser generado aleatoriamente o a partir de algún conocimiento

anterior. Seguidamente, el algoritmo aplica repetidamente la ecuación de actualización:

$$U_i = \{u \in U : |e^C_{uV_i}| > T_C\alpha_C\}, V_{i+1} = \{v \in V : |e^G_{U_iv}| > T_G\alpha_G\}$$

Las iteraciones finalizan en el paso n satisfaciendo:

$$\frac{V_{n-i} \backslash V_{n-i-1}}{V_{n-i} \cup V_{n-i-1}} < \epsilon$$

para todo i menor que algún m. Por tanto, se puede afirmar que ISA converge a un punto fijo aproximado, el cual debe ser considerado como un bicluster. Este punto depende tanto del conjunto inicial V_{in} como de los umbrales T_C, T_G.

3.4.5. El algoritmo SAMBA

El algoritmo SAMBA (Statistical–Algorithmic Method for Bicluster Analysis)[284] usa un modelado probabilístico de los datos y técnicas de teoría de grafos para identificar subconjuntos de genes que *respondan conjuntamente* a lo largo de un subconjunto de condiciones, donde un gen es calificado *respondedor* para algunas condiciones si su nivel de expresión cambia significativamente en dichas condiciones con respecto a su nivel normal.

SAMBA modela los datos de expresión como un grafo bipartito. Cada una de las partes corresponden a condiciones y genes, respectivamente, con aristas en los cambios significativos de expresividad. Cada pareja de vertices del grafo tiene asignado un peso según el modelo probabilístico, tal que los subgrafos ponderados corresponden a biclusters con una gran probabilidad. El descubrimiento del bicluster más significativo en los datos reduce, bajo este esquema ponderado, el encontrar el subgrafo mejor valorado (con más peso) dentro del modelo de grafo bipartito. SAMBA emplea una heurística para buscar estos subgrafos (*heavy graph*).

El algoritmo SAMBA se basa en la representación del conjunto de datos de expresión de entrada como un grafo bipartido $G = (U, V, E)$. En este grafo, U es el conjunto de condiciones, V el de genes, y $(u, v) \in E$ si el nivel de expresión de v cambia significativamente en la condición u. Un bicluster correspondería a un subgrafo $H = (U', V', E')$ de G y representaría un subconjunto de genes V' que están corregulados bajo un subconjunto de condiciones U'. El *peso* de un subgrafo (o bicluster) es la suma de los pesos de los pares gen–condición. Este peso será dependiente del modelo estadístico usado para representar el bicluster final. Los autores presentan dos de estos modelos y muestran cómo asignar los pesos a cada pareja, estén (*edge*) o no unidos (*non-edge*).

Una vez ponderado el grafo, se procede a encontrar los k subgrafos más pesa-

dos en el grafo. Este problema NP–completo, fue abordado por SAMBA mediante el uso de una heurística reduciendo su coste computacional a $O(n2^d \log k)$. La aproximación usa como semilla grandes *bicliques*[4]. La generación de estas semillas se describe seguidamente, donde se supone que el grado[5] de cada gen es limitado a d.

Dado el grafo bipartido $G = (U, V, E)$ con $n = |V|$ genes. Dada la función de ponderación $w : U \times V \rightarrow \mathcal{R}$. Para una pareja de los subconjuntos $U' \subseteq U$, $V' \subseteq V$, se denota por $w(U', V')$ al peso del subgrafo inducido sobre $U' \cup V'$, i.e., $w(U', V') = \sum_{u \in U', v \in V'} w(u, v)$. La vecindad de un vértice v, denotado por $N(v)$, es el conjunto de vértices adyacentes a v en G.

El pseudocódigo de SAMBA se muestra en el algoritmo 3.9, el cual puede ser agrupado en dos fases. Una primera, donde se genera el modelo de grafo bipartito y se calculan los pesos de cada pareja de vértices. Otra segunda, donde se buscan diferentes *heavy graph* a lo largo de cada vértice del grafo. Esto se realiza comenzando con un conjunto de vértices y expandiéndolos usando una búsqueda local. Para ahorrar tiempo y espacio, el algoritmo ignora aquellos genes cuyo grado no supere un umbral d, y usando por cada gen el conjunto de sus vecinos ($N(v)$)de tamaño entre N_1 y N_2. El procedimiento de mejora local se aplica iterativamente al bicluster actual (añadiendo o borrando un vértice) hasta que no exista mejora posible. El proceso voraz está limitado a buscar alrededor de un *biclique* sin realizar cambios que eliminen vértices de él o creando vértices redundantes. Para evitar bicluster similares con un conjunto de vértices ligeramente diferentes, se lleva a cabo una etapa final que filtra biclusters similares con un solape mayor que un $L\,\%$.

[4]Un biclique es un grafo bipartito conexo

[5]El grado de un gen es el número de vecinos de éste

Algoritmo 3.9 SAMBA

INPUT V: conjunto de genes
U: conjunto de condiciones
E: aristas del grafo, w:pesos de aristas/no–aristas
d: umbral del grado de un gen
N_1, N_2: umbrales de tamaño del conjunto de vecinos.
K: nº de biclusters máximo por gen/condición
L: umbral de solape
begin
Inicializar una tabla de *pesos*
for all $v \in V$ con $|N(v)| \leq d$ **do**
for all $S \subseteq N(v)$ con $N_1 \leq |S| \leq N_2$ **do**
$peso[S] \leftarrow peso[S] + w(S, v)$
end for
end for
for each $v \in V$ **do**
Establece $best[v][1...k]$ los k conjuntos mejores valorados S tal que $v \in S$
end for
for each $v \in X$ y $K \in S = best[v][i]$ **do**
$V' \leftarrow \bigcap_{u \in S} N(u)$
$B \leftarrow S \cup V'$
while exista mejora **do**
$a = argmax_{x \in V \cup U}(w(B \cup x))$
$b = argmax_{x \in B}(w(B \setminus x))$
if $w(B \cup a) < w(B \setminus b)$ **then**
$B = B \cup a$
else
$B \setminus b$
end if
end while
Almacenar B
end for
Filtrar solapamiento de biclusters según L
end

3.5. Redes Reguladoras de Genes

En ocasiones, los usuarios de microarrays no sólo están interesados en los grupos de genes, sino que también les interesa la relación entre los clusters, y la relación entre los genes del mismo cluster. Es decir, conocer qué influencias tiene un gen sobre el resto, y de qué forma son afectados.

Las técnicas de aprendizaje que nos aportan este tipo de conocimiento se denominan "redes reguladoras de genes" (***gene regulatory network (GRN)***). Éstas tienen la finalidad de imitar las redes biológicas en algún nivel de abstración y posibilitan un mejor entendimiento del sistema biológico subyacente. En este tipo de técnicas es crítico tener un modelo de arquitectura y un procedimiento de inferencia fiable [278]. Seguidamente son detallados los modelos y los procesos de inferencia más relevantes [147].

3.5.1. Modelos de arquitecturas de red

Antes de inferir un GRN, se debe elegir el tipo apropiado de modelo de arquitectura de red. El modelo es una función matemática parametrizada que describe el comportamiento general de un componente objetivo basado en la actividad de los componentes reguladores. En los últimos años, numerosos modelos de arquitecturas han sido propuestos [147]. Éstos varían en el grado de simplificación y reflejan distintas suposiciones del mecanismo molecular subyacente.

En general, los nodos de las redes representan compuestos de interés; genes, proteínas o conjuntos de éstos. Según van Someren et al. [298], los modelos pueden distinguirse por:

1. *La representación del nivel de actividad de los componentes de la red*. La contentración o la actividad de un componente puede ser repersenatdo por valores lógicos (e.g. 'activo'–'no activo', 'presente'–'ausente'), discretos (e.g. etiquetas del cluster), dispersos (e.g. 'bajo', 'medio', 'alto') o continuos.

2. *El tipo de modelo* (estocástico o determinista, estático o dinámico).

3. *El tipo de relaciones entre las variables* (dirigido o no dirigido; funciones lineales o no lineales)

Los cuatro modelos de arquitectura principales se encuentran representados en la figura 3.3. En la figura, el objetivo es inferir la interacción regulativa entre los tres genes (la gráfica de GRN arriba a la derecha) basada en los datos de expresión de estos genes para unos experimentos concretos (matriz de expresión arriba a la izquierda). Para el modelado se usas los datos de expresión y otra información biológica disponible. Las cuatro arquitecturas de modelado reflejan la misma GRN

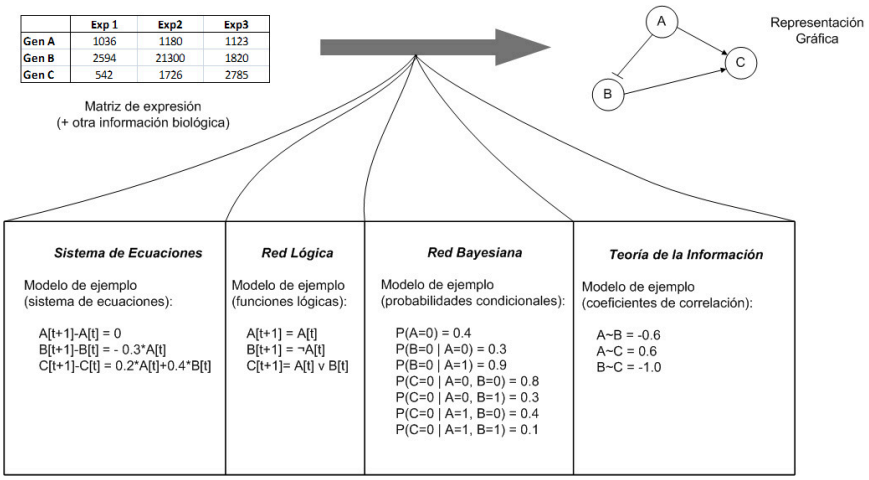


Figura 3.3: Ejemplo global de los cuatro modelos de arquitectura principales de GRN.

de diferentes maneras. Cada modelo de arquitectura, que es descrito seguidamente, es ilustrado por un ejemplo simple y típico de una posible realización de su formalismo.

Ecuaciones Diferenciales Las ecuaciones diferenciales describen los cambios del nivel de expresión de un gen como una función de expresión de otros genes y factores ambientales. De este modo, éstas son adecuadas para modelar el comportamiento dinámico de las GRNs de multitud de formas. Su flexibilidad permite incluso describir relaciones complejas entre componentes. Un modelado dinámico de expresión génica podría aplicar, por ejemplo, ecuaciones diferencias ordinarias (ODE):

$$\frac{dx}{dt} = f(x, p, u, t)$$

donde $x(t) = (x_1(t), ..., x_n(t))$ es el vector del nivel de expresión para los genes $1, ..., n$ en el momento t; f es la función que describe la tasa de cambio de las variables de estado x_i con respecto a los parámetros p del modelo; y u las señales de perturbación externas. Así, la inferencia de redes supone la identificación de la función f y los parámetros p de las señales medidas x, u y t.

En general, existen múltiples soluciones, es decir, el sistema ODE no es únicamente identificable a partir de los datos a mano. De este modo, la identificación de la estructura y los parámetros del modelo requieren la especificación de la función f y forzar la representación de conocimiento previo me-

diante simplificaciones o aproximaciones. Por ejemplo, la función f puede ser lineal o no lineal (para más detalle se remite al lector a la sección 3.3. de [147]), aunque, evidentemente, los procesos reguladores son caracterizados por dinamismos no lineales complejos. Sin embargo, muchos trabajos sobre inferencia de GRN basados en ecuaciones diferenciales consideran modelos lineales o están limitados a tipos muy específicos de funciones no lineales [89, 300].

Redes Lógicas. Las redes lógicas son redes dinámicas discretas. Fueron propuestas por Kauffan [176, 177, 178] y desde su aparición han sido intensamente investigadas para el modelado de la regulación de genes [56, 291]. Éstas usan variables binarias $x_i \in \{0, 1\}$ que definen el estado de un gen i representado por un nodo de la red como 'off' o 'on' (inactivo o activo). Por ello, antes de inferir un red lógica, el nivel de expresión de un gen (valores continuos) tiene que ser transformado a valores binarios. Esta discretización puede ser realizada, por ejemplo, usando la regresión del vector soporte [218]. Las redes lógicas pueden ser representadas como un grado dirigido, donde las aristas son simbolizadas por funciones lógicas creadas de operaciones lógicas simples: Y ($\wedge$), O ($\vee$) y NO ($\neg$) (una descripción más completa para generar este tipo de red puede encontrarse en [277]). El desafío de estas redes es encontrar una función lógica para cada gen de la red tal que los datos observados (discretizados) sean explicados por el modelo. En la actualidad existen varios algoritmos para la inferencia de redes lógicas, i.e. REVEAL (*REVerse Engineering ALgorithm*)[204].

Las redes lógicas son limitadas por definición, ya que la expresión genética no puede ser descrita adecuadamente en sólo dos estados. De todas formas, este tipo de redes son fácilmente interpretables y, al ser dinámicas, pueden ser usadas para simular los eventos reguladores del gen.

Redes Bayesianas. Las redes Bayesianas (BNs) hacen uso de las reglas de Bayes para reflejar la naturaleza estocástica de la regulación génica. Éstas asumen que los valores de expresión pueden ser descritos por variables aleatorias que sigan distribuciones probabilísticas. Al representar relaciones reguladoras mediante probabilidades, las BNs están pensadas para modelar la aleatoriedad y el ruido como características inherentes de los procesos reguladores de los genes [123]. Además, las BNs proporcionan un marco muy flexible para combinar diferentes tipos de datos y conocimiento previo en los procesos de inferencia de GRN para deducir una estructura de red adecuada [305]. Igualmente, este modelo de red posee numerosas características que lo hacen candidato para el modelado de GRN, tales como su habilidad para

evitar el sobreajuste y manejar datos ruidoso e incompletos. Los métodos de aprendizaje de redes Bayesianas han sido estudiados en detalle en los trabajos de Heckerman [148] y Needham et al. [228]. En resumen, hay tres partes esenciales en el aprendizaje de una BN:

- *Selección del modelo.* Define un grafo acíclico dirigido (DAG) como grafo candidato de relaciones.
- *Parámetro adecuado.* Dado un grafo y unos datos experimentales, encontrar la mejor probabilidad condicional para cada nodo.
- *Valoración de capacidad.* Puntuar cada modelo candidato. Cuanto mayor sea la puntuación, mejor es el modelo de red para representar los datos.

La selección del modelo es la etapa crítica. La aproximación naïve es simplemente enumerar todos los posibles DAGs para un número de nodos dado. El problema es que el número de DAGs con N nodos crece de manera exponencial. Por ello, al igual que para otros tipos de modelo, es necesaria una heurística para aprender eficientemente un BN.

Las redes Bayesianas han sido usadas extensamente para la reconstrucción de GRNs. Como ejemplo, Rangel et al. [251] infirieron un modelo lineal de 39 genes sobre la activación de *T–cell* a partir de series temporales de expresión génica. Digno de mención es BANJO [144], una aplicación software para inferir redes Bayesianas y redes Bayesianas dinámicas.

Modelos de Teoría de la Información. Una de las arquitecturas de red más simples es la red de correlación, la cual puede ser representada por un grafo no dirigido con aristas ponderadas según el coeficiente de correlación. Con tal fin, dos genes interactuarán si el coeficiente de correlación de sus niveles de expresión es superior a algún umbral establecido. Cuanto mayor sea este umbral, más dispersa será la red inferida.

Para detectar dependencias reguladoras entre genes se pueden emplear multitud de medidas de proximidad (ver apartado 3.2), como por ejemplo distancias Euclídeas o puntuaciones provenientes de la teoría de la información o de la información mutua [280]. Los algoritmos de inferencia RELNET (*RELevance NETworks*) [63], ARACNE (*Algorithm for the Reverse engineering of Accurate Cellular NEtworks*) [216] o su ampliación TimeDelay–ARACNE [332], y CLR (*Context Likelihood of Relatedness*) [115] aplican esquemas en los que las aristas son ponderadas por valores estadísticos provenientes de la información mutua.

Las mayores ventajas que aportan los modelos basados en teoría de la información son la simplicidad y el bajo coste computacional. Por ello, son muy apropiados para estudiar propiedades globales de sistemas reguladores de gran escala. En comparación con otros formalismos, un inconveniente de estos modelos es que no tienen en cuenta la intervención de varios genes en una regulación. Aunque una mayor desventaja de éstos es que son estáticos.

Otros modelos de arquitectura de red. No todas las técnicas de modelado de GRN puede ser asignadas a una de las cuatro categorías descritas anteriormente. Para completar este apartado, tres de esas propuestas son presentadas a manera de ejemplo. Segal et al. [268] identificaron modelos reguladores en *S. cerevisiae* mediante árboles de regresión. Por otro lado, Erns et al. introdujeron el algoritmo DREM (*Dynamic Regulatory Events Miner*) [113] que usaba modelos de Markov para la identificación y anotación de puntos de bifurcación en expresión genética. Recientemente, Mordelet y Vert [225] descompusieron la inferencia de GRN en un gran número de problemas de clasificación binaria enfocados a separar los genes objetivo de los que no lo son para cada factor de transcripción.

3.5.2. Algoritmos de aprendizaje para la inferencia de redes

En general, la reconstrucción de una red es llevada a cabo aplicando un algoritmo de aprendizaje que cumpla la salida del modelo matemático provisto por los datos experimentales. La elección de un algoritmo apropiado esta influenciado, en gran medida, por el modelo de arquitectura seleccionado, además de por la calidad y cantidad de datos disponibles. Además, si está disponible el conocimiento previo de las interacciones regulativas de los genes, el algoritmo debería ser capaz de incorporar tal conocimiento al modelo final.

En la inferencia de redes, se deben distinguir dos tareas fundamentales: (1) la estimación de la estructura del modelo y (2) la valoración de los parámetros del modelo. La optimización de la estructura corresponde al problema de encontrar la conectividad o topología de la red que mejor represente los datos observados y que cumpla las restricciones del conocimiento disponible, i.e. que tenga en cuenta los requisitos de dispersión de la red [117]. La estimación de los parámetros concierne al problema de identificar los parámetros del modelo una vez la estructura de éste esté determinada. Para calcular la dispersión de una red, la mayoría de estudios de inferencia intentan reducir el grado de cada nodo. En muchos de estos enfoques la estructura es optimizada explícitamente y la optimización de parámetros se convierten en un tarea embebida de la optimización de la estructura. Alternativamente, existen muchas aproximaciones donde la estructura es implícitamente determinada

durante la estimación de los parámetros (para más detalle ver la sección 5.2.1 de [147]).

Optimización de parámetros La optimización de los parámetros de un modelo está relacionada con el modelo de arquitectura seleccionado y con la función de puntuación a optimizar. Esta función siempre contiene un término que cuantifica la adecuación de las salidas del modelo predichas con respecto a los datos de expresión. Dependiendo de la distribución de ruido predicha, las medidas para este criterio son, por ejemplo, la suma de errores al cuadrado [298]. Para cada tipo de arquitectura existe un gran número de técnicas de optimización de parámetros, tales como el presentado por Pollisetty et al. [241] para los modelos GMA (*Generalized Mass Action*) o por Vilela et al. [299] para los *S–systems*.

Optimización de la estructura Deducir la estructura del modelo o la conectividad entre los nodos es un desafiante problema de optimización combinatorio. Para cada nodo, se deben encontrar las combinaciones de reguladores más probables. El número total de posibles combinaciones para cada nodo es $2^N - 1$, donde N representa al número de nodos en la red. Consecuentemente, incluso para redes pequeñas, probar todas las posibles estructuras de red no es factible. Sin embargo, el número puede ser decrementado significativamente asumiendo un límite de conectividad entre los genes.

Una regla general en la reconstrucción de redes es que cuanto mayor sea el grado de conectividad del grafo, mejor cuadrará el modelo con los datos. Sin embargo, cuanto más conectado esté el grafo más número de dificultades existen [188]. Primero de todo, se supone que los genes están regulados por un número limitado de reguladores [17]. Segundo, la fiabilidad de la estimación de los parámetros decrece cuanto mayor número de éstos haya. Por ello, se debe encontrar un equilibrio entre la calidad del modelo y su complejidad. Una visión general de métodos de optimización de estructura aplicados al modelado experimental fue realizada por Nelles en [229], o más recientemente por Gregorcic en [134].

3.6. Resumen

En este capítulo se ha realizado un estudio sobre las diferentes técnicas de análisis de datos de expresión génica basadas en microarrays. En primer lugar, se destacada el papel de la minería de datos en el estudio de datos ómicos. Posteriormente, se presenta la diferencia entre aprendizaje superviado y no supervisado, así como su relevancia en datos biomédicos.

En segundo lugar, se muestran las medidas de proximidad más relevantes. Estas medidas son mostradas como la base de cualquier análisis de datos para poder comparar la similitud o disimilitud entre genes. Con tal fin, las medidas son distinguidas entre medidas de distancia y medidas de similitud, así valoren la diferencia o parecido de dos genes.

Una vez descritas las medidas de proximidad que se pueden encontrar en la literatura de análisis de microarrays, son detalladas las tres técnicas de aprendizaje más relevantes en el aprendizaje no supervisado: clustering, biclustering y redes genéticas reguladoras. La exposición de técnicas de clustering es agrupadada en clustering jerárquico, basado en particiones, mapas autoorganziados y basados en teorías de grafos. Para las técnicas de biclustering, un apartado de definiones y notaciones ha sido necesario. Este apartado es la base para poder presentar el algoritmo de Cheng y Church, la aproximación CTWC y las técnicas ISA y SAMBA. Por último, las redes reguladoras son expuestas. Con tal fin, se presenta, en primer término, los diferentes modelos de arquitectura de red existentes y, en segundo, los algoritmos de aprendizaje para inferir redes genéticas.

Capítulo 4

Validación Analítica

Razonar y convencer, ¡qué difícil, largo y trabajoso! ¿Sugestionar? ¡Qué fácil, rápido y barato!

SANTIAGO RAMÓN Y CAJAL.

4.1. Introducción

En la sección anterior se han presentado numerosas técnicas de aprendizaje automático que, dependiendo del grupo en que se encuentren, representan el conocimiento extraído de una forma u otra. El siguiente paso es comprobar si el modelo resultante es correcto para los datos procesados, y poder así establecer si un método concreto es válido e incluso elegir la mejor técnica (bajo unas determinadas condiciones) entre varias de éstas ya contrastadas. Por ello, la etapa de validación puede entenderse como el proceso para determinar la calidad y fiabilidad de los modelos generados, siendo una fase crucial en cualquier proceso de aprendizaje automático.

En la literatura de Minería de Datos se encuentran numerosos trabajos enmarcados en la etapa de validación. En general, son divididas en dos grupos dependiendo de si los datos están (*supervised learning*) o no etiquetados (*unsupervised learning*). En Bioinformática se suele realizar una distinción muy similar, con la salvedad de que las técnicas de validación basadas en conocimiento previo pueden trabajar, también, con datos no etiquetados. En muchas ocasiones, el conjunto de datos de partida, particularmente, datos de expresión genómica, no dispone de un atributo que nos informe en qué proceso y bajo qué condiciones intervienen unos genes determinados. Sin embargo, no implica que se desconozca totalmente la funcionalidad o la estructura de éstos. Por ejemplo, si un gen no puede ser catalogado dentro de un proceso biológico particular (etiqueta), no implica que sea un gen totalmente desconocido, puesto que se podría saber sobre qué otros genes

influye o con qué otros presenta un comportamiento similar. En resumen, la validación en Bioinformática puede ser agrupada en técnicas que no se basan en ningún conocimiento previo y técnicas que sí lo hacen, con la distinción que para estas últimas el conocimiento puede encontrarse en un atributo clase o en una base de datos biológica.

Con el objetivo de presentar las técnicas de validación en Bioinformática se hace una novedosa distinción de éstas en dos grupos. En un primer grupo (***validación analítica***) se incluirán todas aquéllas que se basan en estudios analítico/matemáticos de los datos, utilicen o no una información adicional para ello. Y otro segundo (***validación biológica***), donde se encuentran aquéllas basadas en la comparación del conocimiento obtenido con algún otro extraído de forma biológica experimental. Nótese, que el segundo grupo, presentado en el capítulo 5, es un caso particular de evaluación basada en conocimiento previo y nos aporta una estimación del modelo extraído desde el punto de vista biológico. Por otro lado, las técnicas pertenecientes al primer grupo, presentadas en este capítulo, nos aportan una valoración desde un punto de vista estadístico, muchas de las cuales provenientes de la Minería de Datos.

Las técnicas de validación analítica y biológica existentes en la actualidad serán expuestas en el capitulo que nos ocupa y el siguiente, respectivamente. La visión que se realiza sobre éstas no es exhaustivo, y se centra en aquellos métodos que evalúan los modelos generados por técnicas de clustering. La razón de esto es que, además de que el análisis de genes diferentemente expresados genera grupos de genes, las técnicas de biclustering y GRN, como primera aproximación, suelen evaluarse como si de grupos de genes se tratara:

- Un cluster puede ser entendido como un caso particular de bicluster en donde un conjunto de genes se comportan igual en todos y cada uno de los experimentos. Debido a la escasez de técnicas especializadas en la validación de biclusters, esta equiparación es comúnmente usada con tal fin.

- Un cluster es un caso particular de una red genética reguladora, donde todos los elementos del cluster están relacionados con todos de la misma forma. Por tanto, una posible validación de redes es utilizar técnicas de validación de cluster, en donde la entrada sean los elementos de la red (nodos) y se obvien las relaciones entre ellos. Ciertamente, en la actualidad existen técnicas específicas para la validación y comparación de redes pero, debido a su dependencia de la teoría de grafos, quedarán fueran del alcance de este documento.

Concretamente, esta sección comienza mostrando las diferentes medidas a tener en cuenta para evaluar un resultado de clustering; dividiéndolas en externas e

internas. Y, posteriormente, se presentan las técnicas de visualización como complemento a la medidas anteriores.

4.2. Medidas de validación Analítica

La literatura de minería de datos proporciona diferentes técnicas de validación, las cuales pueden ser divididas en medidas de validación externas o internas [139]. Estos dos grupos de técnicas difieren fundamentalmente en sus enfoques, y encuentran aplicación en distintos marcos experimentales. Las medidas de validación externas comprenden a todos los métodos que evalúan los resultados de un cluster basándose en el conocimiento de las etiquetas correctas de la clase. Evidentemente, ésta es de gran utilidad al permitir una evaluación y comparación totalmente objetiva de los algoritmos de clustering sobre un conjunto de datos, para los cuales las etiquetas de la clase corresponden con la estructura verdadera de cluster. En el caso donde no se disponga de esta clase, o la clase sea dudosa, se deberá utilizar una medida de validación interna. Las técnicas de validación interna no usan un conocimiento adicional, sino que sólo se basan en la información intrínseca de los datos.

4.2.1. Medidas Externas

- ***Medidas Unarias***: Las medidas de validación externa toman a un único resultado de clustering como entrada, y lo comparan con un conjunto de etiquetas conocidas (la "base de verdad" o "regla de oro") para valorar el grado de consenso entre ambos. Tradicionalmente, este conocimiento previo sería completo y único, de forma que la exactitud de una etiqueta es provista para cada elemento de los datos, y que ésta está definida inequívocamente. Por tanto, una partición puede ser evaluada con respecto a la ***pureza*** del cluster individual, o mediante la ***completitud*** de los clusters. Aquí, la pureza denota la fracción del cluster que ocupa su etiqueta predominante, mientras que completitud es la fracción de los elementos en esta clase predominante que es agrupada en el cluster en cuestión. O dicho de otra manera, la pureza es la relación de genes bien clasificados en relación al número de genes del cluster a tratar, mientras que la completitud es la proporción de genes bien etiquetados con respecto al número total de genes de entrada que comparten esa etiqueta, y por tanto, deberían haber estado en dicho cluster. Evidentemente, estos aspectos sólo proporcionan una cantidad de información limitada, y soluciones triviales de ambos existen. Por ejemplo, un cluster simple presentaría una puntuación máxima según su pureza, mientras que una solución de un único cluster tendría máxima puntuación según su completitud. A fin de

obtener una valoración de una partición de acuerdo con la base de verdad, es importante tener en cuenta tanto la pureza como la completitud. Las medidas como *F–measure* [297] suministran una base para evaluar ambas y son, de esta forma, preferibles sobre una única técnica. Dichas medidas aportan un medio para valorar la calidad de los resultados de un cluster al nivel de la partición entrante, y no sólo para clusters individuales. En principio, tales medidas también pueden ser adaptadas para usarlas con una "etiquetación parcial" (i.e. conjunto donde la información de las etiquetas sea incompleta) aplicando la medida a los datos etiquetados y a sus respectivos cluster. Esto puede aportar una mejor comprensión a la hora de asignar la calidad de los cluster que el realizar el cálculo del nivel de significatividad del "enriquecimiento" [126, 290, 293] de un cluster concreto.

- ***Medidas Binarias***: Además de las medidas basadas en la pureza y la completitud, la literatura de minería de datos también proporciona otras que permiten calcular el consenso entre una partición y el "conocimiento base" basada en la tabla de contingencia de la asignación de pares de elementos de los datos. La mayoría de estas nuevas medidas son simétricas y, por tanto, son apropiadas para usarlas como medidas binarias, es decir, para asignar la similitud entre dos resultados de clustering diferentes.

  Dentro de estas medidas, la más conocida es el ***Rand Index*** [250]. Ésta determina la similitud entre dos particiones como una función de concordancias positivas y negativas en las parejas de los clusters. Actualmente existen numerosas variaciones de esta medida, en particular el ***Rand Index ajustado*** [161], el cual introduce una normalización estadística para que los valores sean cercanos a cero en particiones aleatorias. Otra medida relacionada es el ***coeficiente de Jaccard*** [165], la cual aplica una definición un tanto estricta de correspondencia en la que sólo se recompensan las concordancias positivas. Igualmente, podemos encontrarnos una métrica similar a *Rand Index* [10] para calcular la consistencia entre clusters. La medida, denominada ***weighted–kappa***, asigna valores entre -1 y $+1$ en la comparación de dos clusters, de forma que un valor alto indica que los clusters comparados son similares, mientras que un valor bajo indica que éstos son diferentes.

  Todas las medidas expuestas anteriormente son simétricas al estar basadas en tablas de contingencia. Como medida asimétrica (i.e. $M(U,V) \neq M(V,U)$ para dos particiones U y V) podemos encontrar la puntuación de ***Minkowski*** [168], aunque este tipo de medidas pierden utilidad para asignar similitudes entre resultados de clustering.

4.2.2. Medidas Internas

Las medidas internas toman todos los grupos resultantes de la aplicación de una técnica de clustering y los datos de partida como entrada, y usan la información intrínseca de los datos para asignar la calidad de la técnica de clustering en cuestión. Existen diferentes propiedades válidas para que puedan ser atribuidas a una buena partición, pero éstas, en parte, están en conflicto y son generalmente difícil expresarlas en términos de funciones objetivos. A pesar de ello, existen criterios/algoritmos de clustering que las agrupan en las siguientes categorías [141]:

- ***Homogeneidad***: El primer grupo comprende las medidas de validación que valoran la compactación y homogeneidad del cluster. Entre ellas, las más representativas son la varianza dentro del cluster, el criterio de varianza de la suma del error cuadrado mínimo o la optimización local usando el algoritmo de k–medias (sección 3.3.2). Actualmente existen numerosas variaciones de estas medidas, destacando las variantes de la medida de homogeneidad intra–cluster que son calculadas como la media o el máximo de las distancias entre pares del cluster, media o el máximo de las similitudes basadas en el centroide o el uso de técnicas basadas en grafos [45].

- ***Conectividad***: El segundo tipo de técnica de validación interna intenta valorar cómo de buena es una partición basándose en el concepto de conectividad, i.e., en qué grado, en una partición dada, se observan densidades locales y grupos de items junto a sus vecinos más cercanos en el espacio de datos. Un ejemplo representativo de estas técnicas puede ser la *consistencia* [96] o *conectividad* [141] de los k vecinos más cercanos, las cuales cuentan las infracciones de las relaciones de los vecinos más cercanos.

- ***Separación***: El tercer grupo incluye a aquellas medidas que cuantifican el grado de separación entre los clusters individualmente. Por ejemplo, una valoración global para una partición puede ser definida como la media de las distancias inter–clusters, donde la distancia entre dos clusters puede ser calculada como la distancia de los centroides de los clusters, o como la distancia mínima entre dos elementos que se encuentran en clusters diferentes. Alternativamente, la separación del cluster en una partición puede, por ejemplo, ser calculada como la separación mínima observada entre clusters individuales en la partición.

- ***Combinaciones***: La literatura provee numerosas técnicas que combinan medidas de diferente tipo. En esta línea, las combinaciones de las medidas de *homogeneidad y separación* son las más populares, ya que ambas medidas exhiben tendencias opuestas: mientras que la homogeneidad intra–cluster

mejora con un incremento en el número de clusters, la distancia entre clusters tiende a deteriorarse. De esta forma, diferentes técnicas calculan la homogeneidad intra–cluster y la separación inter–cluster, y calculan una medida final como la combinación lineal o no lineal de estas dos medidas. Un ejemplo de una combinación lineal es *SD-validity Index* [139] donde SD se refiere al hecho de que este índice mide la dispersión y la distancia de los clusters. Como ejemplo de combinación no lineal serían el *índice de Dunn* [102], *índices basados en el de Dunn* [45], *Davies-Vouldin Index* [88] o *Silhouette Width* [259].

Mientras que los métodos anteriores son relativamente populares, las combinaciones lineales o no lineales de las medidas producen, inevitablemente, una cierta pérdida de información que puede que nos lleven a algunas conclusiones incorrectas. Un camino alternativo para realizar una validación usando N medidas simultáneamente es usar el Pareto óptimo [236]: un resultado de clustering es considerado dominante de otro si éste es igual o superior bajo todas las medidas, y es estrictamente superior bajo al menos una medida. Un estudio reciente usando el Pareto óptimo para validar un resultado de clustering usando las medias de homogeneidad y separación puede ser encontrado en [142].

- ***Estabilidad y predicción***: Las técnicas de validación calculan el poder predictivo o de estabilidad de una partición formando una clase especial de medidas de validación interna. Evidentemente, estas medidas no son medidas externas, ya que no usan una información etiquetada. Sin embargo, son bastante diferentes de las medidas internas tradicionales ya que su uso requiere un acceso adicional al algoritmo de clustering. Las medidas de este tipo perturban los datos originales y reagrupan los datos. La consistencia de los nuevos resultados aportan una estimación del significado del cluster obtenido de los datos originales.

  Los métodos descrito en [36, 49, 60, 121, 182, 192, 200, 202, 221, 292] emplean el concepto de auto–consistencia, es decir, la idea de que un algoritmo de clustering debería producir resultados consistentes cuando sea aplicado a datos probados de la misma fuente. A fin de calcular el grado de estabilidad de una partición, diferentes artículos [36, 200] extraen subejemplos solapados repetidamente del mismo conjunto de datos (cada subejemplo individual es extraído sin reemplazamiento). Cada subejemplo es agrupado individualmente, y posteriormente, las particiones resultantes son comparadas aplicando un índice de validación externa a la partición parcial obtenida por superposición del conjunto de puntos compartidos.

Una propuesta diferente ha sido dada en [60, 121, 192, 292]. Aquí, los datos son divididos repetidamente en datos de entrenamiento y de test (típicamente con el mismo tamaño y sin solape), y ambas particiones son agrupadas (se les aplica una técnica de clustering). La partición en conjuntos de entrenamiento es empleada para sacar un clasificador para predecir todas las etiquetas de la clase para los conjuntos de test. El desacuerdo entre la predicción y la partición en el conjunto de test puede ser por tanto calculado usando un índice de validación externo binario. Obviamente, el clasificador usado para la predicción tiene un gran impacto sobre la actuación de este método y debería cumplir con la suposición del modelo creado por el algoritmo de clustering. Lange et al. [192] recomendaron el uso de un clasificador de vecino más cercano para vínculos simples, y clasificadores basados en centroides para algoritmos como k–means, los cuales asumen clusters con forma esférica.

Finalmente, la estabilidad de un resultado de clustering puede se calculado comparando la partición obtenida sobre los datos perturbados [49, 182, 202]. En este supuesto, se generan diferentes conjuntos de datos a partir de los originales usando bootstrap: usando un modelo simple de error [49] o un método más avanzado como ANOVA [182], en donde se añade ruido a cada elemento del conjunto de datos. Las bases de datos resultantes son sometidas a un análisis cluster. Por tanto, la partición obtenida puede ser comparada directamente usando índices binarios externos.

- ***Conformidad entre una partición y la información de distancia***: Un camino alternativo para calcular la calidad de una técnica de clustering es estimar directamente el grado en que la información de distancia en los datos originales es conservada en una partición determinada de éstos. Para este propósito, una partición de los datos es representada por la media de su matriz *cophenetic* C [258], donde C es una matriz simétrica de tamaño $N \times N$ y N el tamaño del conjunto de datos. En una partición determinada, la matriz C contiene sólo ceros y unos, donde cada entrada $C(i, j)$ indica si dos elementos i y j han sido asignados al mismo cluster o no. Para la evaluación de un cluster jerárquico (apartado 3.3.1), estas matrices también pueden ser construidas para reflejar la estructura del dendograma. Para ello, una entrada $C(i, j)$ representa el nivel donde los elementos (items) i y j son asignados por primera vez al mismo cluster.

  Por tanto, la matriz *cophenetic* puede ser comparada con la matriz de disimilitud original usando el Γ estadístico de Hubert (esencialmente es el producto escalar entre dos matrices), el Γ estadístico normalizado, o una medida de correlación como la correlación de Pearson [107] o el rango de correlación de Spearman [197]. La correlación entre dos matrices es comúnmente referen-

ciado como correlación *cophenetic*, matriz de correlación o el estandarizado *Mantel Statistic* [139]. Por otro lado, la correlación *cophenetic* también puede ser usada como un índice binario para calcular la conservación de distancias bajo diferentes funciones de distancia o características de espacios, o para comparar los dendogramas obtenidos por diferentes algoritmos.

- ***Medidas especializadas para datos muy correlados***: Esta última categoría de medidas de validación interna incluyen a numerosas técnicas que explotan explícitamente redundancias y correlaciones, tales como aquéllas inherentes a datos genómicos. El primero de éstos, la figura de méritos [321], es motivado por el trabajo desarrollado por Efron y Tibshirani en [108]. Para un conjunto de datos de n genes y a atributos, clasificado en k clusters, la figura de mérito (***FOM***) de Yeung et al. [320, 321] requiere la computación de a particiones, cada una de ellas basada en $a - 1$ de los a atributos. Por tanto, para cada partición e, su figura de mérito es calculada como la media de la varianza intra–cluster dentro del atributo no usado:

$$FOM(e,k) = \sqrt{\frac{1}{n}\sum_{i=1}^{k}\sum_{g\in C_i}(R(g,e) - \overline{R}_i(e))^2} \tag{4.1}$$

Siendo C_i el conjunto de genes en el cluster i, $R(g,e)$ el nivel de expresión del gen g bajo la condición e en la matriz inicial, y $R_i(e)$ el nivel de expresión medio en la condición e para los genes en el cluster i.

La figura de mérito agregada, $FOM(k) = \sum_{e=1}^{a} FOM(e,k)$, es una estimación del poder de predicción total del algoritmo sobre todas las condiciones para los k clusters de un conjunto datos.

Datta [85] amplió este enfoque a la computación de una figura de mérito mediante diferentes índices internos de validez; específicamente, una medida basada en el par separación–homogeneidad de clusters.

La segunda propuesta, análisis sobre-abundante [35, 49], calcula la frecuencia de varianza discriminatoria para una partición dada. Es decir, identifica aquellas variables que muestran diferencias significativas entre los clusters identificados. Las frecuencias observadas son comparadas con el modelo nulo para valorar la importancia de la partición.

Claramente, ambas propuestas son aplicables únicamente a conjunto de datos con variables correladas (dependientes), aunque probablemente sean aplicables a otros tipos de datos genómicos [141].

4.3. Visualización de Clusters

Aunque las computadoras son de gran utilidad para realizar análisis estadísticos a grandes cantidades de datos, éstas no alcanzan la gran habilidad cognitiva del ser humano. Los usuarios expertos son capaces de identificar rápidamente tanto las correlaciones como las irregularidades si los datos o los resultados están visualizados apropiadamente [180]. Tareas que para las computadoras son inherentemente dificultosas, como la parametrización; o que son subjetivas, como la redundancia o la relevancia para un fin particular, pueden ser abordadas mediante la interacción del hombre y la máquina. La visualización juega un papel fundamental en dicha interacción, ya que ésta genera interfaces entre la salida automatizada y el usuario. Para la validación de conocimiento la visualización juega un rol complementario a las medidas presentadas en el apartado anterior, siendo una etapa crucial en el proceso KDD [140].

En la literatura existen diferentes técnicas de visualización que se encuentran resumidas en trabajos como [68, 75]. Según [180] y como se muestra la Figura 4.1, estas técnicas pueden ser clasificadas según tres criterios: los datos a ser visualizados, el método de visualización usado o la técnica de interacción empleada.

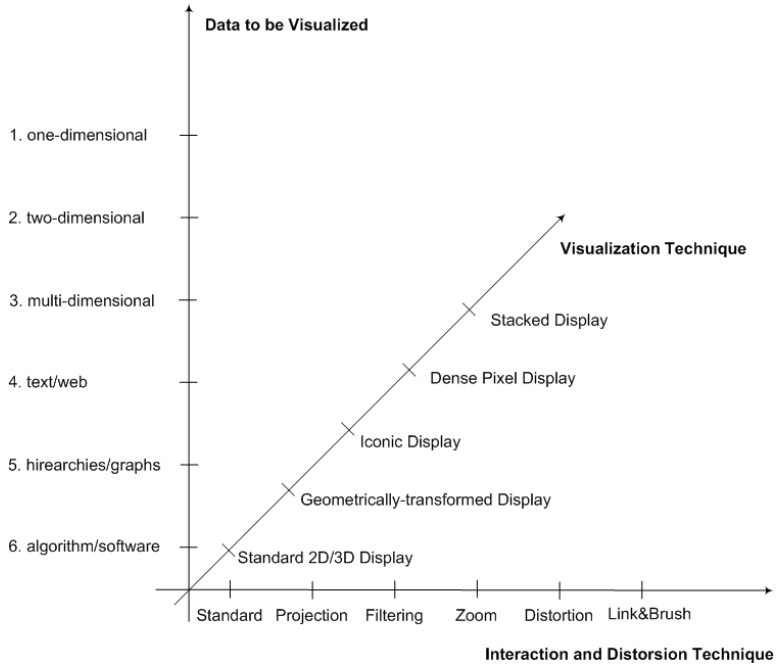


Figura 4.1: Clasificación de técnicas de visualización de información.

En el caso del estudio concreto que nos ocupa, existen métodos de visualización para validar resultados de técnicas de clustering que cubren cada una de las

clasificaciones. Dependiendo del método de clustering usado para dividir al conjunto de datos, podríamos encontrar, por ejemplo, técnicas de visualización de grafos jerárquicos para aproximaciones como *hierarchical clustering* (apartado 3.3.1) o bidimensionales para métodos como *k–means* (apartado 3.3.2). Por otro lado, debido al fundamento particional de los métodos de clustering, las técnicas basadas en proyecciones son de gran utilidad para explorar los resultados obtenidos.

Hoy en día, existen diversas herramientas que proporcionan diferentes aproximaciones de visualización. Entre las más usadas se encuentra EXPANDER [272], una herramienta desarrollada por Shamir et al. para el análisis de datos de expresión genética. Entre sus diferentes utilidades se encuentra la visualización de microarrays, representación de tendencia de cluster y análisis de componente principal; las cuales son descritas seguidamente.

4.3.1. Visualización de microarrays

Como se expuso en el apartado 2.4.2, el análisis de bajo nivel sobre datos microarrays tiene el objetivo de reducir el ruido producido en el proceso de creación de éstos. Tras la reducción de ruido, las señales pueden ser transformadas a números. De esta forma, se convierte la matriz de datos original en una matriz de expresión donde cada celda $W_{i,j}$ representa el nivel de expresión para el gen i bajo la condición j. Esta matriz puede ser representada gráficamente codificando el nivel de expresión en un color de diferente intensidad. Comúnmente se usa una degradación del verde al rojo, donde las tonalidades cercanas al verde indican que los datos presentan una expresividad baja, mientras que las cercanas al rojo representan datos con gran nivel de expresión.

La visualización de microarrays es un herramienta útil para ver gráficamente la homogeneidad y separación de los resultados obtenidos por una técnica de clustering. En la figura 4.2 se puede observar cómo los genes que se encuentran bajo el mismo cluster (genes cuyo nombre poseen el mismo color) siguen un patrón similar, mientras que la comparación de los diferentes patrones de los cluster muestra un comportamiento distinto entre genes de diferentes clusters.

Sobre este tipo de visualización han surgido diferentes tipos de estudios. Por ejemplo, Garber et al. demostraron en [124] que la representación basada en ranking es más eficiente que la basada en saturación para detectar ruidos en los cluster. Más recientemente Hibbs et al. [152] propusieron dos nuevas técnicas de visualización basadas en la cohesión, para analizar la calidad general de los cluster y detectar outliers u otras anomalías en éstos. Un ejemplo ilustrativo del trabajo de Hibbs es mostrado en la figura 4.3, donde tres clusters son visualizados tanto de la forma tradicional (izquierda) como por la propuesta por Hibbs et al. (derecha). En la nueva representación aparece una barra superior que muestra la media del

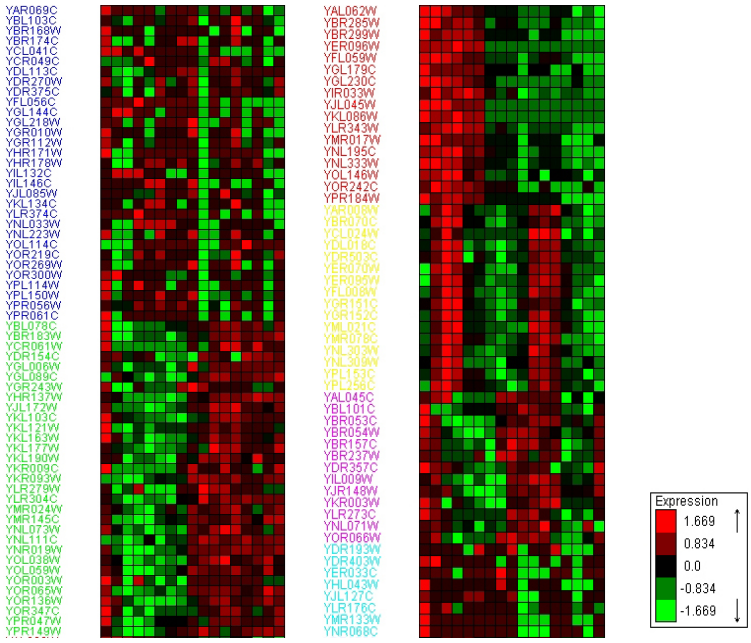


Figura 4.2: Ejemplo de visualización de microarrays.

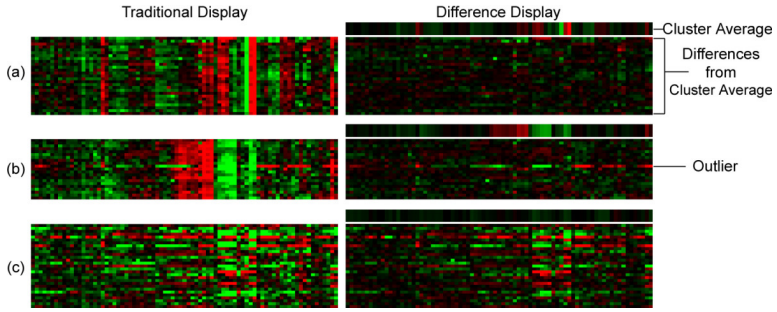


Figura 4.3: Representación visual de diferencias de expresividad.

cluster; cada gen es representado como la diferencia entre su expresividad y la media del cluster. Nótese que el verde indica una infra–expresión con respecto a la media, el rojo muestra que está más expresado, mientras que el negro significa que presenta una expresión equitativa.

4.3.2. Tendencia de cluster

Este tipo de representación tiene el fin de aportar información sobre el patrón de comportamiento de los cluster resultantes. Para ello cada cluster es representado de forma individual mediante una línea que simboliza el patrón de comportamiento medio de los genes del cluster en cuestión. Dicha línea es pintada sobre los ejes de coordenadas x–y, donde el eje abscisas representa la condición y el eje de coordenadas el nivel medio de expresividad de los genes del cluster en dicha condición. Además, se incluye la desviación media de expresividad que existe en cada condición de cada cluster para identificar el error cometido en el cálculo de comportamiento medio.

Nótese, que la gráfica resultante (Figura 4.4) posee la suficiente potencia para ver gráficamente la homogeneidad y separación que existe en cada cluster. Siendo por tanto, una herramienta visual complementaria a la expuesta en la sección 4.2.2.

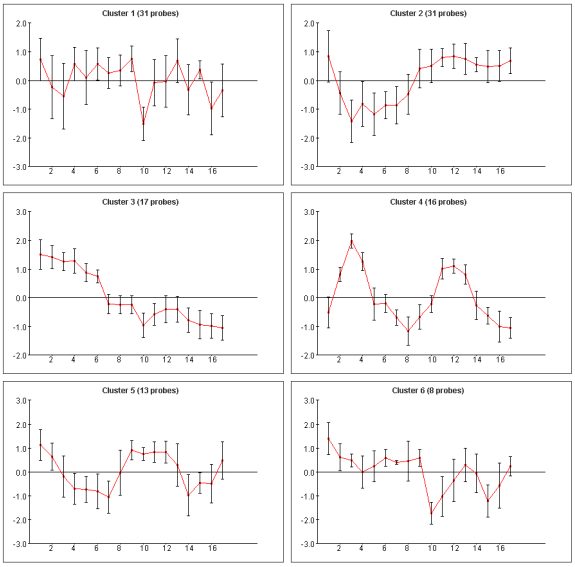


Figura 4.4: Representación de la tendencia de los cluster de la figura 4.2.

En la figura 4.4 se representa el patrón de comportamiento y el error cometido en su cálculo de los cluster expuestos en la figura 4.2. En tal representación, se observa que el patrón de comportamiento de los diferentes clusters es muy diferente uno de otro. Igualmente, según el error cometido, se puede apreciar que los

cluster 3 (genes de color rojo) y 4 (genes de color amarillo) presentan una mayor homogeneidad que el resto de clusters. Ambas afirmaciones, son corroboradas por la figura 4.2 y por la medida de homogeneidad[1] presentada por Expander (ver tabla 4.1) .

Tabla 4.1: Resumen analítico presentado por Expander.

Cluster (color)	*Tamaño*	*Homogeneidad*
1 (azul)	31	0,612
2 (verde)	31	0,78
3 (rojo)	17	0,912
4 (amarillo)	16	0,917
5 (rosa)	13	0,727
6 (cian)	8	0,757

4.4. Resumen

En este apartado se presenta la utilidad de la fase de validación para medir la calidad de los modelos generados por cualquier técnica de análisis. Así mismo, se ha realizado una división de las técnicas de validación existentes atendiendo a si usan o no conocimiento biológico existente. Concretamente, aquellas metodologías basadas en estudios analíticos/matemáticos de los datos de entrada (validación analítica) son estudiadas en esta sección, mientras que las basadas en la comparación del conocimiento extraído con algún otro extraído y contrastado de forma experimental (validación biológica) serán analizadas en el próximo capítulo.

Las medidas analíticas son divididas en externas e internas. Las medidas externas comprenden a todos los métodos que se basan en el conocimiento de las etiquetas correctas de la clase (supervised learning). Mientras que dentro de las medidas internas se encuentran aquellas que tan sólo usan la información intrínseca de los datos.

Posteriormente, se detallan diferentes técnicas de visualización de grupos de genes para realizar una validación gráfica de tales modelos. Con tal fin, se presentan las técnicas de visualización de microarrays y las basadas en el patrón de comportamiento del nivel de expresión de los genes de un cluster.

[1] Homogeneidad calculada como la media de los valores de similaridad de cada elemento

Capítulo 5

Validación Biológica

No saber lo que ha sucedido antes de nosotros es como ser incesantemente niños

CICERÓN.

5.1. Introducción

La aplicación de una técnica de análisis de microarray genera un modelo de los datos de entrada, que, de forma general, pueden ser entendidos como una lista de genes. Estas listas son objeto de estudio por la comunidad científica para determinar su relevancia biológica, obteniendo de esta forma un mejor entendimiento del fenómeno biológico subyacente.

Con tal fin, algunas medidas de validación analítica han sido usadas para validar los modelos obtenidos mediante información biológica externa. Por ejemplo, *weighted–kappa* es usado en [282] para realizar un estudio entre diferentes técnicas de clustering de expresión genética. Priness et al. [247] usan los conceptos de homogeneidad y separación para asignar la calidad de cluster de datos de expresión genética. Brohee et al. [62] aplican los conceptos de sensibilidad, separación o robustez para evaluar la capacidad de varias técnicas de clustering para inferir complejos de proteínas a partir de redes de interacción proteínica. Incluso, como se indica en [141], las medidas analíticas de distancia interna pueden ser usadas en bioinformática para estimar el número de clusters óptimo en que se divide el conjunto de datos de entrada. Por ejemplo, podemos encontrarnos este tipo de estudios en [53, 54, 121, 192, 292, 315]

Sin embargo, y a pesar de que estas adaptaciones se encuentran disponibles y pueden ser fácilmente desarrolladas, la interpretación biológica del conocimiento extraído es aún una etapa esencial en este tipo de estudios [2]. En este sentido, multitud de trabajos de investigación han sido llevados a cabo para desarrollar me-

todologías, herramientas o medidas de validación biológica.

Los procesos de validación biológica, que consisten en la comparación del modelo de conocimiento generado con la información biológica real, pueden ser divididos en dos fases. Una primera, donde es necesario obtener y procesar los datos biológicos que nos servirán como conocimiento real. Y otra segunda, en donde se compara la solución obtenida y la solución real. En base a esta distinción, esta sección comenzará presentando el conocimiento biológico actual y las bases de datos relacionadas; posteriormente, se estudiarán las herramientas de enriquecimiento desarrolladas hasta el momento; y para finalizar, se presentarán las medidas biológicas y de similitud funcional propuestas hasta la fecha.

5.2. Bases de Datos Biológicas

Aunque el conocimiento actual sobre datos biológicos está muy lejos de ser completo, es impresionante el tamaño y el crecimiento extremadamente rápido de éste. Muchos científicos trabajan en generar datos y llevar a cabo investigaciones de análisis sobre éstos. Actualmente, hay un flujo fluido y abundante de resultados obtenidos en bancos de laboratorios para ser almacenados. Hoy en día existen multitud de bases de datos biológicas que pueden ser divididas en tres categorías [314]: bases de datos *primarias*, *secundarias* o *especializadas*.

Las ***bases de datos primarias*** contienen datos biológicos originales, cuya información es la de secuencia completas o datos estructurales generadas por la comunidad científica. Entre las más relevantes encontramos *GenBank* [40] como la colección más completa y general de datos de secuencias de ácidos nucléicos, *EMBLBank* [281] generado por el Laboratorio de Biología Molecular de Europa (*European Molecular Biology Laboratory*) o *DDBJ* [287] suministrado por el Banco de Datos de ADN de Japón (*DNA Data Bank of Japan*).

Las ***bases de datos secundarias*** contienen información procesada computacionalmente o revisada de forma manual, basada en la información original almacenada en las bases de datos primarias. En esta categoría se encuentran las bases de datos de secuencias de proteínas que contienen anotaciones funcionales. Ejemplos pueden ser *UniProtKG/Swiss-Prot* [78] o *Protein Information Resources* (*PIR*) [311].

Por último, las ***bases de datos especializadas*** son aquellas que satisfacen un interés de investigación particular. Por ejemplo, *FlyBase* [296], la base de datos de secuencia *HIV* [190] o *Ribosomal Database Project* [233] son bases de datos que están especializadas en un organismo o tipo de datos particular.

En la tabla 5.1 se presentan las bases de datos más relevantes en la era postgenómica, aunque en [59] podemos encontrar casi todo el conocimiento biológico

actual. Las bases de datos son organizadas según la información contenida, donde cada una de estas clasificaciones es detallada en los siguientes apartados.

Tabla 5.1: Bases de Datos Biológicas más relevantes disponibles via Web.

Bases de Datos y Sistemas de Recuperación	**Resumen del Contenido**	**URL**
AceDB	BD genómica para *Caenorhabditis elegans*	www.acedb.org
DDBJ	BD primaria de secuencia de nucleótidos en Japón	www.ddbj.nig.ac.jp
EMBL	BD primaria de secuencia de nucleótidos en Europa	www.ebi.ac.uk/embl/index.html
Entrez	Portal del NCBI para una variedad de BD biológicas	www.ncbi.nlm.nih.gov/gquery/gquery.fcgi
ExPASY	BD proetómica	http://us.expasy.org
FlyBase	Una BD del genoma de la *Drosophila*	http://flybase.bio.indiana.edu/
FSSP	Estructuras secundarias de proteínas	www.bioinfor.biocenter.helsinki.fi:8080/dali/index.html
GenBank	BD primaria de secuencia de nucleótidos in el NCBI	www.ncbi.nlm.nih.gov/Genbank
HIV databases	Datos de secuencias e información de inmunología relacionada con el VIH	www.hiv.lanl.gov/content/index
Microarray gene expression database	Datos de ADN en microarrays y herramientas analíticas	www.ebi.ac.uk/microarray
OMIM	Información genética de enfermedades humanas	www.ncbi.nlm.nih.gov/entrez/quiery.fcgi?db=OMIM
PIR	Secuencias de proteínas anotadas	http://pir.georgetown.edu/pirwww/pirhome3.shtml
Pubmed	Literatura Biomédica	www.ncbi.nlm.nih.gov/PubMed
Ribosomal database project	Secuencias del ARN del ribosoma y árboles filogenéticos derivados de las secuencias	http://rdp.cme.nsu.edu/html
SRS	Sistema de recuperación de secuencias generales	http://srs6.ebi.ac.uk
Swiss–Prot	BD de secuencias de proteínas	www.ebi.ac.uk/swissprot/acces.html
TAIR	BD sobre la *Arabidopsis*	www.arabidopsis.org

Por último, y para finalizar esta sección, se presentan las dos bases de datos más relevantes para la clasificación y asignación de funciones proteicas y génicas: Gene Onotolgy y KEGG. En tales secciones se realizará un profundo análisis de ambas aproximaciones, además de exponer la solución adoptada por Gene Ontology para resolver el problema de la ambigüedad de términos.

Bases de datos de secuencias de ácidos nucleicos

Como se expuso anteriormente, el conjunto de datos de secuencias de ácido nucleico son suministrados por una triple sociedad: *National Center for Biotech-*

nology (*NCBI*) de EEUU; *EMBL Nucleotide Database* o *EMBLBank*, del European Bioinformatics Institute (EBI, de Reino Unido); y el *DDBJ* del *National Institute of Genetics* en Japón. Estos proyectos archivan y distribuyen las secuencias de ADN y ARN recopiladas a partir de proyectos genoma, publicaciones científicas y postulaciones de patentes. Los grupos intercambian información diariamente, de manera que los datos son idénticos. Sin embargo, el formato con el que éstos son presentados y la naturaleza de las anotación varía según la base de datos.

Las bases de datos de secuencias de ácidos nucleicos son colecciones de entradas. Cada entrada es representada por un fichero que contiene los datos y anotaciones para un secuencia concreta. Muchas entradas son recopiladas a partir de diferentes publicaciones reportando fragmentos solapados de una secuencia completa.

En la figura 5.1 se muestra un ejemplo de la información almacenada en el EMBL Nucelotide Database. Concretamente, se muestra una secuencia de nucleótidos de un serotipo[1] de la gripe aviar (H5N1); *Influenza A virus* (EMBL-Bank: DQ659324.1),

Bases de datos de genomas

Las bases de datos generales de ácidos nucleicos se centran en coleccionar secuencias individuales. Sin embargo, podemos encontrar bancos de datos con la información de secuencias de genomas enteros obteniendo base de datos con toda la información molecular disponible sobre una especie particular.

La finalidad principal de estas bases de datos es almacenar y anotar toda la información disponible sobre secuencias genómicas de ADN, enlazarla con la secuencia del genoma original y hacerla accesible a los científicos que examinarán los datos desde diferentes puntos de vista y requerimientos. Con este fin, además de almacenar y organizar la información, se han realizado multitud de esfuerzos para desarrollar infraestructuras computacionales, incluyendo el establecimiento de convenciones de nomenclaturas adecuadas. No es trivial idear un esquema para mantener los identificadores de los datos de forma estable ya que éstos no sólo sufrirán un crecimiento sino que además están en continua revisión. El resultado más visible de este esfuerzo está en la web; es extremadamente fácil extraer información general o centrarse en detalle en alguna de éstas.

Ensembl [119, 120] es considerada como la fuente de información universal para el genoma humano y otros. Éste es un proyecto conjunto del *EBI* y del *The Sanger Centre*, que se enmarca como un proyecto abierto para animar a la contribución externa. La información que se encuentra en Ensembl incluye genes, re-

[1] Un serotipo es un tipo de microorganismo infeccioso clasificado según los antígenos que presentan en su superficie celular.

```
ID   DQ659324; SV 1; linear; viral cRNA; STD; VRL; 1398 BP.
XX
AC   DQ659324;
XX
DT   13-JUN-2006 (Rel. 88, Created)
DT   16-DEC-2008 (Rel. 98, Last updated, Version 2)
XX
DE   Influenza A virus (St Jude H5N1 influenza seed virus 163243) neuraminidase
DE   (NA) gene, complete cds.
XX
KW   .
XX
OS   Influenza A virus (St Jude H5N1 influenza seed virus 163243)
OC   Viruses; ssRNA negative-strand viruses; Orthomyxoviridae; Influenzavirus A.
XX
RN   [1]
RP   1-1398
RA   Hoffmann E., Webby R.J., Webster R.G.;
RT   "Availability of new H5N1 prototype strain for influenza pandemic vaccine
RT   development";
RL   Unpublished.
XX
RN   [2]
RP   1-1398
RA   Hoffmann E., Webby R.J., Webster R.G.;
RT   ;
RL   Submitted (26-MAY-2006) to the INSDC.
RL   Department of Infectious Diseases, St.Jude Children's Research Hospital,
RL   332 N. Lauderdale Street, Memphis, TN 38105, USA
XX
FH   Key             Location/Qualifiers
FH
FT   source          1..1398
FT                   /organism="Influenza A virus (St Jude H5N1 influenza seed
FT                   virus 163243)"
FT                   /serotype="H5N1"
FT                   /mol_type="viral cRNA"
FT                   /note="derived from A/Whooping swan/Mongolia/244/2005
FT                   (H5N1)"
FT                   /db_xref="taxon:388040"
FT   gene            21..1370
FT                   /gene="NA"
FT   CDS             21..1370
FT                   /codon_start=1
FT                   /gene="NA"
FT                   /product="neuraminidase"
FT                   /db_xref="GOA:Q195D7"
FT                   /db_xref="InterPro:IPR001860"
FT                   /db_xref="InterPro:IPR011040"
FT                   /db_xref="UniProtKB/TrEMBL:Q195D7"
FT                   /protein_id="ABF93438.1"
FT                   /translation="MNPNQKIITIGSICMVIGIVSLMLQIGNMISIWVSHSIQTGNQRQ
FT                   AEPISNTKFLTEKAVASVTLAGNSSLCPISGWAVYSKDNSIRIGSRGDVFVIREPFISC
FT                   SHLECRTFFLTQGALLNDKHSNGTVKDRSPHRTLMSCPVGEAPSPYNSRFESVAWSASA
FT                   CHDGTSWLTIGISGPDNGAVAVLKYNGIITDTIKSWRNNILRTQESECACVNGSCFTVM
FT                   TDGPSSGQASYKIFKMEKGKVVKSVELDAPNYHYEECSCYPDAGEITCVCRDNWHGSNR
FT                   PWVSFNQNLEYQIGYICSGVFGDNPRPNDGTGSCGPVSPNGAYGVKGFSFKYGNGVWIG
FT                   RTKSTNSRSGFEMIWDPNGWTGTDSSFSVKQDIVAITDWSGYSGSFVQHPELTGLDCIR
FT                   PCFWVELIRGRPKESTIWTSGSSISFCGVNSDTVSWSWPDGAELPFTIDK"
XX
SQ   Sequence 1398 BP; 411 A; 253 C; 356 G; 378 T; 0 other;
     agcaaaagca ggagttcaaa atgaatccaa atcagaagat aataaccatc ggatcaatct        60
     gtatggtaat tggaatagtt agcttaatgt tacaaattgg gaacatgatc tcaatatggg       120
     tcagtcattc aattcagaca gggaatcaac gccaagctga accaatcagc aatactaaat       180
     ttcttactga gaaagctgtg gcttcagtaa cattagcggg caattcatct ctttgcccca       240
     ttagcggatg ggctgtatac agtaaggaca acagtataag gatcggttcc agggggggatg      300
     tgtttgttat aagagagccg ttcatctcat gctcccactt ggaatgcaga actttctttt       360
     tgactcaggg agccttgctg aatgacaagc actccaatgg gactgtcaaa gacagaagcc       420
     ctcacagaac attaatgagt tgtcctgtgg gtgaggctcc ctccccatat aactcaaggt       480
     ttgagtctgt tgcttggtca gcaagtgctt gccatgatgg caccagttgg ttgacaattg       540
     gaatttctgg tccagacaat ggggctgtgg ctgtattgaa atacaatggc ataataacag       600
     acaccatcaa gagttggagg aacaacatac tgagaactca agagtctgaa tgtgcatgtg       660
     taaatggctc ttgctttact gtaatgactg atggaccaag tagtgggcag gcatcatata       720
     agatcttcaa aatggaaaaa gggaaagtgg ttaaatcagt cgaattggat gctcctaatt       780
     atcactatga ggagtgctcc tgttatcctg atgccggcga aatcacatgt gtgtgcaggg       840
     ataattggca tggctcaaat aggccatggg tatctttcaa tcaaaatttg gagtatcaaa       900
     taggatatat atgcagtgga gttttcggag acaatccacg ccccaatgat ggaacaggta       960
     gttgtggtcc ggtgtcccct aacggggcat atggggtaaa agggttttca tttaaatacg      1020
     gcaatggtgt ttggatcggg agaaccaaaa gcactaattc caggagcggc tttgaaatga      1080
     tttgggatcc aaatgggtgg actggaacgg acagtagctt ttcggtgaag caagatatcg      1140
     tagcaataac tgattggtca ggatatagcg ggagttttgt ccagcatcca gaactgacag      1200
     gattagattg cataagacct tgtttctggg ttgagttaat cagagggcgg cctaaagaga      1260
```

Figura 5.1: Entrada del *EMBL Nucleotide Database* para un serotipo de la gripe aviar (EMBL-Bank: DQ659324.1).

peticiones y homologías. Los genes pueden ser conocidos experimentalmente o deducidos a partir de las secuencias. Debido a que el soporte experimental para las anotaciones del genoma humano es muy variable, Ensembl registra y presenta las pruebas para la identificación y anotación de cada gen. La información disponible es complementada mediante enlaces a otras bases de datos con información relacionada, tales como *Online Mendelian Inheritance in Man* ($OMIM^{TM}$) o bases de datos de expresión.

Bases de datos de secuencia de proteínas

En la actualidad, las tres bases de datos de secuencias de proteínas más importantes son: *The Protein Information Resource* (*PIR*) [26] del *National Biomedical Research Foundation* de la Universidad Médica de Georgetown en Washington, como bases de datos de secuencia pionera; y *SWISS–PROT* y *TrEMBL* [23], del *Swiss Institute of Bioinformatics* en Ginebra y del *EBI* en Hinxton. Éstas, aunque comparten su información para establecer el consorcio UniProt, siguen ofreciendo su información de forma separada además de proporcionar herramientas para su acceso.

TrEMBL contiene la traducción de los genes identificados en la secuencia de ADN almacenadas en la base de datos de nucleótidos del EMBL. Las entradas de TrEMBL son consideradas como preliminares, y son convertidas en entradas maduras del SWISS–PROT.

Hoy, la mayoría de la información de secuencia de aminoácidos surgen de la traducción de secuencias de genes. Sin embargo, la secuencia de aminoácidos de una proteína no es, en general, inferible a partir de la secuencia de genes. La razón principal, en los eucariotas, es la ambigüedad en las uniones. Además, la información sobre ligandos[2] o los efectos de editado del ARNm no están disponibles para secuencias de ADN. Los bancos de datos de secuencias coleccionan esta información adicional de la literatura y proporcionan anotaciones adecuadas.

Bases de datos de familia de proteínas

Las relaciones evolutivas son esenciales para establecer el sentido de los datos biológicos. La evolución proporciona un marco para una apreciación de las propiedades de las moléculas y los procesos, y sus similitudes y diferencias en varias especies. Conociendo sólo una secuencia o estructura concreta es difícil entender el significado de propiedades particulares [199]. Los patrones de conservación identifican características que la naturaleza ha encontrado necesaria retener. Por tanto, el reto es comprender el porqué.

[2]Un ligando es una sustancia capaz de unir y formar una estructura con una biomolécula para servir a un propósito biológico.

Estudios de patrones de evolución deben comenzar reuniendo un conjunto de homologías. Nótese que la homología hace referencia a un descendente de un ancestro común, la cual será una propiedad si o no; mientras que la similitud es alguna medida cuantitativa de la diferencia entre dos objetos. La similitud puede ser siempre medida, pero es difícil observar homologías directamente. De esta forma, en la mayoría de los casos, la homología es una inferencia a partir de la similitud.

R. Doolittle sugirió una calibración general de similitud entre pares de secuencias para la detección de homologías. Dos secuencias completas de proteínas que tienen un porcentaje de residuos idénticos mayor o igual a 25 % en un alineamiento óptimo deben ser relacionadas. Por debajo de un 15 % no hay razones para creer que las secuencias están relacionadas, aunque podría ser. Doolittle definió el rango entre 18 y 25 % de identidad como "*the twilight zone*", donde habría posibilidad de relación, aunque no haya grandes evidencias para ello.

La estructura de proteínas cambia de forma más conservadora que la secuencia de aminoácidos. Por ello, la inferencia de homogeneidad a partir de similitud estructural puede vincular mayor número de parientes relativos que la similitud de secuencias. En el caso de encontrarnos en la zona *twilight* donde la similitud de secuencias es sugerida pero no convincente, la similitud estructural es el último recurso.

Es común referirse a un grupo de proteínas relacionadas como una familia. En la actualidad, existen multitud de bases de datos que realizan esta clasificación, incluyendo bases de datos orientadas a secuencias como *InterPro* [162], *Pfam* [30] y *COG* [288, 289]; y bases de datos orientadas a estructuras como *SCOP* [14, 227] y *CATH* [83, 234]. Por otro lado, podemos encontrar herramientas para la búsqueda de alineamientos locales, entre las que se destaca *BLAST* [11, 66] como la más potente y usada para encontrar regiones de similitud entre secuencias.

Bases de datos de estructura de proteínas

Las bases de datos de estructura de proteínas almacenan, anotan y distribuyen conjuntos de coordenadas atómicas. La mayor base de datos de estructuras biológicas macromoleculares es la *world–wide Protein Data Bank* (*wwPDB*) [42], que es un trabajo conjunto del *Research Collaboratory for Structural Bioinformatics* (RCSB); *Molecular Structure Database*, del EBI; y el *Protein Data Bank Japan*. El wwPDB contiene estructuras de proteínas, ácidos nucleicos y varios carbohidratos. Además, su página web proporciona enlaces a datos de la propia wwPDB y a tutoriales para facilitar la inclusión de nuevas entradas y la búsqueda especializada de estructuras.

Además, podemos encontrar multitud de sitios web que ofrecen una clasificación jerárquica de todas las proteínas con estructura conocida según su patrón

de plegado: *Structural Classification of Proteins* (SCOP); *Class Architecture Topology Homology* (CATH); basadas en la extracción de similitud de estructuras a partir de matrices de distancias (DALI) o una base de datos de alineamiento estructural (CE).

5.2.1. Gene Ontology (GO)

Los científicos actuales emplean mucho tiempo y esfuerzo en buscar en toda la información disponible sobre un area de investigación concreta. Esto es un obstáculo mayor debido a la amplia variedad en la terminología, lo que imposibilita búsquedas efectivas tanto para las computadoras como para los investigadores. Por ejemplo, si estuviéramos interesados en buscar nuevos objetivos para los antibióticos, querríamos encontrar todos los *gene–products*[3] que están implicados en la síntesis de proteínas bacteriales, y que presenten secuencias o estructuras significativamente diferenciadas en los humanos. Si una base de datos describe esas moléculas como implicadas en la "traducción", mientras que otra usa la frase "síntesis de proteínas", será difícil encontrar términos equivalentes funcionalmente.

El proyecto Gene Ontology (GO) [18] es un trabajo de colaboración centrado en dirigir la necesidad de descripciones consistentes de gene–products en diferentes bases de datos. El proyecto comenzó en 1998 como una colaboración entre tres bases de datos de organismos: *FlyBase*, the *Saccharomyces Genome Database* (*SGD*) y *Mouse Genome Database* (*MGD*). Desde entonces, el consorcio de Gene Ontology ha incluido nuevas bases de datos, incluyendo los mayores repositorios de plantas, animales y genomas microbianos.

El proyecto GO ha desarrollados tres vocabularios controlados y estructurados (ontologías) que describen los gene–products en términos de su procesos biológicos, componentes celulares y funciones moleculares asociadas de una manera independiente. Existen tres aspectos independientes para tal esfuerzo: primero, el desarrollo y mantenimientos de la ontología en sí misma; segundo, la anotación de gene–products, que realiza asociaciones entre ontologías y los genes y gene–products en las bases de datos colaboradoras; y tercero, el desarrollo de herramientas que faciliten la creación, mantenimiento y uso de ontologías.

El uso de término GO (GO–term) por las bases de datos colaboradoras facilita la búsqueda de respuestas a través de GO. El vocabulario controlado es estructurado de forma que se puedan realizar cuestiones a diferente nivel: por ejemplo, se puede usar GO para encontrar todos los gene–products del genoma del ratón que estén envueltos en la traducción, o para realizar un estudio concreto en todos los receptores del "tirosine kinases". Este estructura nos ofrece además anotaciones para

[3]Gene product es el material bioquímico, ARN o proteína, resultante de la expresión de un gen.

asignar propiedades a los genes o *gene–products* a diferentes niveles, dependiendo del grado de conocimiento sobre esa entidad.

5.2.1.1. Términos en Gene Ontology (GO–term)

Gene Ontology está basado en términos. Cada entrada en GO tiene un identificador numérico único de la forma *GO:nnnnnn*, y un nombre, e.g. *cell*, *fibroblast growth factor receptor binding* o *signal transduction*. Cada término es asignado a una de las tres ontologías; función molecular, componente celular o proceso biológico.

La mayoría de términos tienen una definición textual con una referencia a la fuente de dicha definición. Si es necesaria cualquier clarificación de la definición o algún comentario acerca del término usado se incluye un campo de comentario separado.

Muchos de los GO–terms tienen sinónimos; GO usa el concepto "sinónimo" de forma general, ya que el nombre del campo sinónimo puede no significar exactamente lo mismo que el término con el que está ligado. En cambio, un sinónimo GO puede ser más amplio o más reducido que el string del término; puede ser una frase relacionada; puede ser una expresión, ortografía o uso alternativo de un sistema diferente de nomenclatura; o puede ser un sinónimo auténtico. Esta flexibilidad permite a los sinónimos GO servir como valiosa ayuda para la búsqueda, además de ser útil para aplicaciones tales como *text mining* y *semantic matching*.

El alcance de Gene Ontology se solapa con varias bases de datos, y en los casos donde un GO–term es idéntico en significado que un objeto en otra bases de datos, se añade al término una referencia a la base de datos externa.

Además GO controla los ***términos específicos de especies***. Gene Ontology tienen el cometido de proveer un vocabulario controlado que pueda ser usado para describir cualquier organismo; sin embargo, muchas funciones, procesos y componentes no son comunes a todas las formas de vida. La convención es incluir cualquier término que pueda aplicarse a más de un clase taxonómica de organismo. Para especificar la clase de organismo con el que el término es aplicable, GO usa la designación `in the sense of`; por ejemplo, *trichome differentitation (sensu Magnoliophyta)* representa la diferenciación de las células de crecimiento de cabello (trichomes).

Ocasionalmente, un término puede ser considerado que está fuera del alcance de GO, ya sea porque está nombrado o definido de forma errónea, o describe un concepto que podría ser representado mejor de otra manera. Estos términos (***términos obsoletos***), mejor que ser borrados, son censurados o convertidos en obsoletos. El término y el ID siguen existiendo en la base de datos GO, aunque el término es marcado como obsoleto, además de ser añadido un comentario que ex-

prese la razón de su censura. Usualmente, se sugiere un término que lo reemplaza.

5.2.1.2. Ontologías

Las tres organizaciones principales de GO son ***componente celular***, ***proceso biológico*** y ***función molecular***. Un *gene–product* puede ser asociado o estar localizado en uno o más componentes celulares; ser activo en uno o más procesos biológicos, durante los cuales lleva a cabo una o más funciones moleculares. Por ejemplo, el gene–product "cytochrome c" puede ser descrito por el término de función molecular *oxidoreductase activity*, por los términos de procesos *oxidative phosphorylation* e *induction of cell death*, y por los términos de componente celular *mitochondrial matrix* y *mitochondrial inner membrane*.

Cellular Component. Un componente celular es justo eso, un elemento de una célula pero con la condición de que sea parte de algún objeto mayor, el cual pudiera ser una estructura anatómica (e.g. retículo o núcleo endoplásmico rugoso) o un grupo de gene–products (e.g. ribosoma, proteasoma o dímero de proteínas).

La ontología componente celular describe localizaciones, al nivel de estructuras subcelulares y complejas macromoléculas (ver figura 5.2). Ejemplos de componentes celulares puede ser *nuclear inner membrane*, con el sinónimo *inner envelope*, y el *ubiquitin ligase complex*, con multitud de subtipos de esta representación compleja.

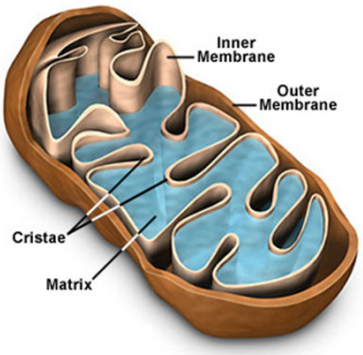
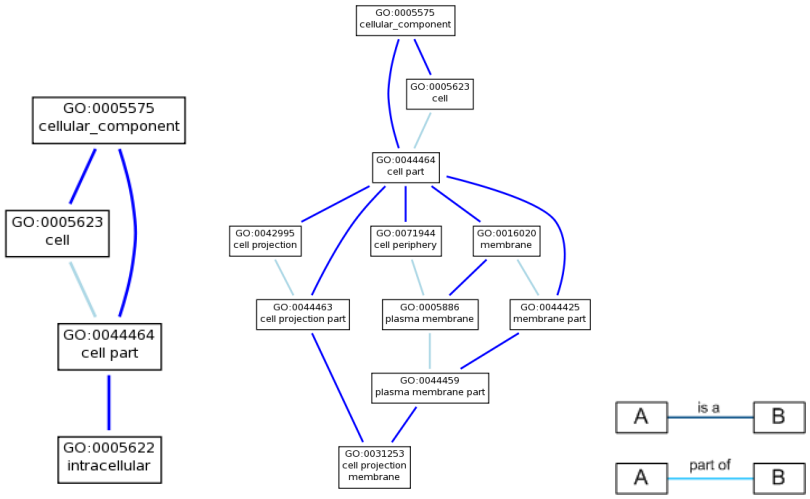


Figura 5.2: Diferentes localizaciones donde un *gene–product* puede actuar.

Generalmente, un gene–product esta *'localizado en'* o es un *'subcomponente de'* un componente celular concreto. La ontología componente celular incluye encimas multi–subunidad y otros complejos de proteínas, pero no proteínas o ácidos nucleicos individuales.

Uno de los conceptos básicos en esta ontología es la *célula*, la cual es definida en GO como todos los componentes incluidos en la membrana y cualquier estructura de encapsulación externa, tales como la pared celular o la envoltura celular. *Intracellular* (GO:0005622) es definido como el contenido de la célula excluyendo la mebrama y cualquier estructura fuera de ésta. Por esta razón, *cell projection* (GO:0042995) es un hijo de *cell* (GO:0005623) y *cell projection membrane* (GO:0031253) es `parte de` *cell projection* (GO:0042995) y de *plasma membrane*(GO:0005886). En la figura 5.3 se muestra un ejemplo de esta ontología.



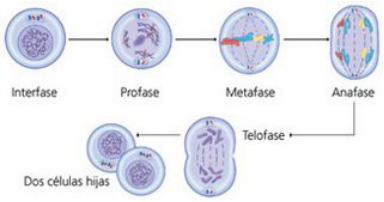
(a) *Intracellular* (b) *Cell projection membrane* (c) Leyenda

Figura 5.3: Estructura de la ontología Cellular Component para diferentes términos GO.

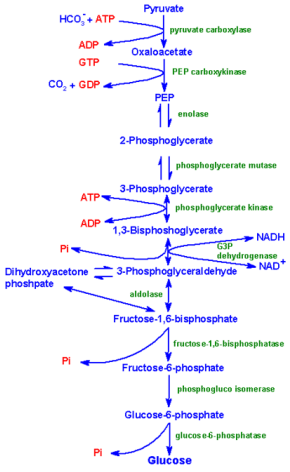
Biological Process Un proceso biológico es una serie de eventos acompañados por una o más funciones moleculares. Como ejemplo, podemos encontrar los procesos fisiológicos celulares (ver figura 5.4(a)), o en términos más concretos, la regulación de la glucogénesis (ver figura 5.4(b)). Puede ser difícil distinguir entre procesos biológicos y funciones moleculares, pero la regla general es que un proceso debe tener más de una etapa diferente.

Un proceso biológico no es equivalente a un pathway, aunque existan GO–terms que describan pathways. GO no captura específicamente o no intenta representar ningún dinamismo o dependencia que fuera requerido describir en un pathway.

La ontología Biological Process incluye términos que codifican colecciones



(a) División Celular (Mitosis).



(b) Regulación de la glucogénesis.

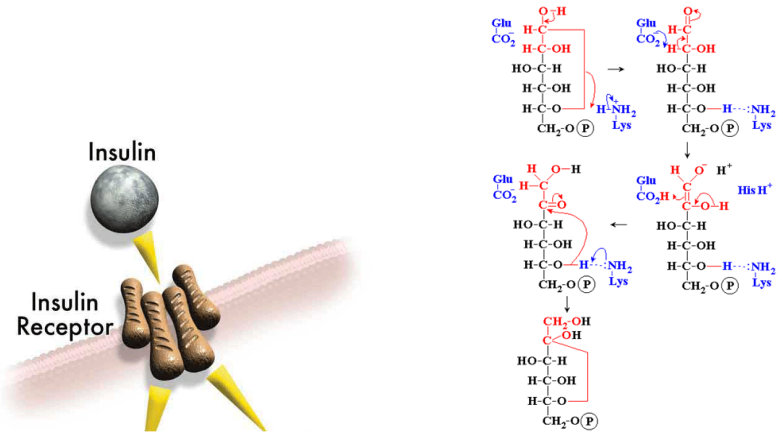
Figura 5.4: Ejemplos de procesos biológicos.

de procesos además de términos que representan un proceso completo específico. Generalmente, el primer término tendrá, en su mayoría, hijos con una relación `is_a`, y los siguientes tendrán hijos con relación `part_of` que representan subprocesos (ver 5.2.1.3).

Uno de los procesos biológicos más relevantes es el *cell cycle* (GO:0007049). Este proceso es situado bajo el término *cellular process* y es dividido en dos tipos de ciclos celulares (mitóticos y meióticos) y fases (G1 phase, S phase, G2 phase and M phase), más un término de regulación.

Molecular Function La función molecular describe actividades a un nivel molecular. Los términos GO en esta ontología representan actividades que más que entidades (moléculas o compuestos) son las acciones llevadas a cabo, y no especifican dónde o cuándo, o en qué contexto, ocurre la acción. Generalmente, las funciones moleculares corresponden a actividades realizadas por genes individuales, aunque algunas de éstas pueden ser soportadas por compuestos de genes. Ejemplos en términos funcionales generales son: actividad catalítica (*catalytic activity*), actividad transportadora (*transporter activity*) o *binding* (ver figura 5.5(a)); y en términos particulares: *adenylate cyclase activity*, *Toll receptor binding* o *isomerase activity* (ver figura 5.5(b)).

Las funciones de un gene–product son los trabajos que realiza o las "habilidades" que tiene, incluyendo el transporte, la fijación o el cambio de una cosa



(a) Insulin binding.

(b) Glucose-6-phosphate isomerase activity.

Figura 5.5: Ejemplos de funciones moleculares.

a otra. Es fácil confundir estas actividades con su función molecular, y por esta razón muchas funciones moleculares de GO poseen la palabra "actividad". Conteptualmente, para distinguir ambos vertientes se podría considerar una analogía con una compañía. Los individuos (gene–products) tienen diferentes habilidades o cometidos (funciones) y éstos trabajan conjuntamente para conseguir diferentes metas (procesos).

5.2.1.3. Estructura de las ontologías

- ***Topología***. Las ontologías están estructuradas como grafos acíclicos dirigidos (DAG), las cuales son similares a las estructuras jerárquicas con la diferencia de que un término más especializado (hijo) puede estar relacionado con más de un término menos especializado (padre). Por ejemplo, el termino biológico *hexose biosynthesic process* tiene dos padres, *hexose metabolic process* y *monosaccharide biosynthetic process*, debido a que el proceso biosintético es un tipo de proceso metabólico y que un *hexose* es un tipo de monosacárido. Esto provoca que cuando se anota un gen al término *hexose biosynthesic process*, éste es automáticamente anotado tanto a *hexose metabolic process* como a *monosaccharide biosynthetic process*.

- ***Relación entre GO–terms***. Los GO–terms pueden ser relacionados según cinco tipos de relación: `is_a`, `part_of`, `regulates`, `positively_regulates` y `negatively_ _regulates`.

La relación `is_a` se trata de una relación simple clase–subclase, donde *A* `is_a` *B* significa que *A* `es_una` subclase de *B*; por ejemplo, la recombinación de ADN (*DNA recombination*, GO:0006310)' `es_un` proceso metabólico de ADN (*DNA metabolic process*, GO:0006259).

La relación `part_of` es un poco más compleja; *C* `part_of` *D* significa que siempre que *C* este presente, es siempre una `parte_de` *D*, pero *C* no siempre tiene que estar presente. Un ejemplo puede ser: *periplasmic flagellum* (GO:0055040) es `parte_de` *periplasmic space*(GO:0042597). Cuando *periplasmic flagellum* está presente es siempre `parte_de` *periplasmic space*. Sin embargo, cada *periplasmic space* no tiene, necesariamente, un *periplasmic flagellum* (ver figura 5.6).

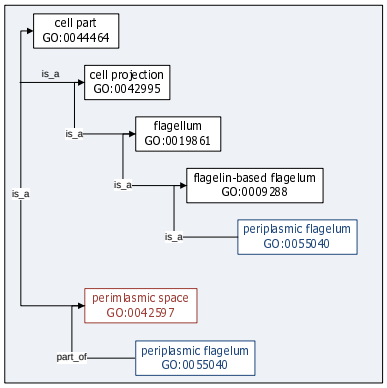


Figura 5.6: Ejemplo de relación 'part_of' entre GO–terms.

La relaciones `regulates`, `positively_regulates` and `negatively_regulates` describen interacciones entre procesos biológicos y otros procesos biológicos, funciones moleculares o cualidades biológicas. Cuando un proceso biológicos *E* `regula` a una función o un proceso *F*, modula la ocurrencia de *F*. Si *F* es una cualidad biológica, entonces *E* modula el valor de *F*. Un ejemplo de la regulación de un proceso biológico sería el término de la regulación de transcripción (*transcription regulation*), ya que cuando éste ocurre siempre altera la tasa, la extensión o la frecuencia con la que el gen es transcrito.

- ***Transitividad de las relaciones***. Las relaciones `is_a` y `part_of` son transitivas, lo que significa que las relaciones son propagadas desde los términos hijos hacia los términos padres.

  En la figura 5.7 se muestra un ejemplo de cada una de las transitividades. Concretamente, la figura 5.7.a representa un ejemplo de transitividad de la relación `is_a`, denotando que todos los *nuclear chromosomes* deben ser

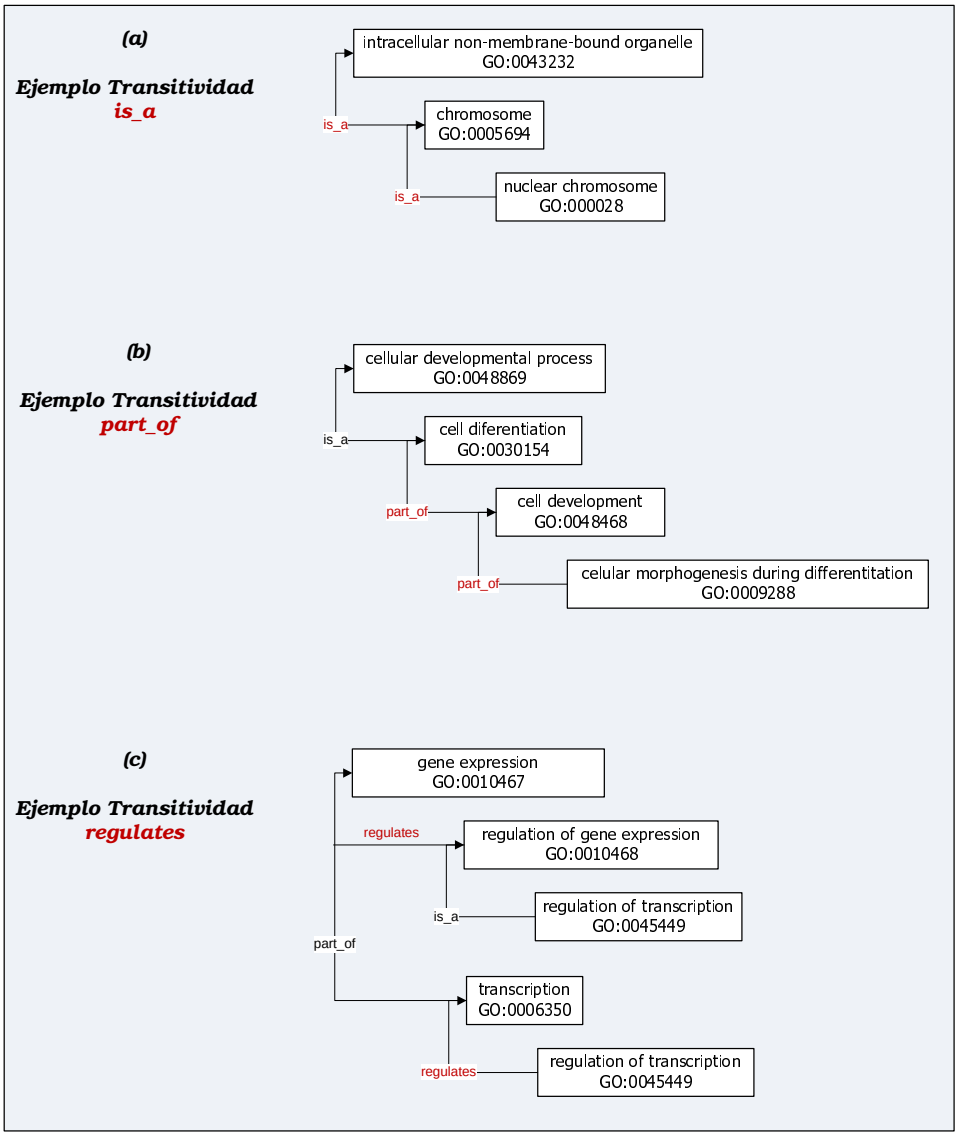


Figura 5.7: Ejemplo de transitividad de relaciones entre GO–terms.

intracellular non-membrane-bound organelles. La transitividad en la relación `part_of` es representada en la figura 5.7.b, donde cada ocurrencia de *cellular morphogenesis during differentiation* debe ser parte de una ocurrencia de *cell differentiation*.

La relación `regulates` es transitiva con respecto a las relaciones `is_a` y `part_of`. Con respecto a esta última relación, si el proceso *Y* existe en la ontología *biological proces*s y es `parte_del` hijo del proceso X entonces cualquier proceso que `regule` al proceso *Y* también `regula` a *X*. Por ejemplo, *regulation of transcription* `regula` *transcription* la cual es `parte_de` *gene expression*. Por tanto, *regulation of transcription* también `regula` a *gene expression* (ver figura 5.7.c).

Con respecto a la transitividad de la relación `regulates` sobre la relación `is_a`, si el proceso *B* existe en la ontología *biological process* y `es_un` hijo del proceso *A* entonces cualquier proceso que `regule` el proceso *B* también `regula` al proceso *A*. Por ejemplo, *regulation of transcription* `es_una` forma de *regulation gene expression*, la cual `regula` a *gene expression*. Por tanto, *regulation of transcription* también `regula` a *gene expresssion* (ver figura 5.7.c).

5.2.2. The Kyoto Encyclopedia of Genes and Genomes (KEGG)

La Enciclopedia de Genes y Genomas de Kyoto (KEGG) almacena genomas individuales, *gene–product* y sus funciones, aunque su principal propósito es la integración de información bioquímica y genética. KEGG, primeramente presentado en [172, 232] y posteriormente ampliado en [174, 173], es un recurso bioinformático para la comprensión general del significado funcional y la utilidad de una célula u organismo a partir de su información genómica. KEGG integra el conocimiento en redes de interacción molecular (ver figura 5.8), tales como pathways (*PATHWAY database*), información de genes y proteínas generada en experimentos genómicos y proteínicos (*GENES/SSDB/KO databases*), y la información sobre componentes químicos y reacciones que son relevantes en procesos celulares (*LIGAND database*). Además, KEGG proporciona facilidades para inferir funciones de alto nivel a partir del nivel de información molecular (*BRITE database*).

Esta información es organizada en cinco tipos de datos, generando un sistema completo: (1) Catálogo de componentes químicos en las células; (2) Catálogo de genes; (3) Mapas de genomas; (4) Mapas de pathways; (5) Tablas de homologías. El catálogo de compuestos químicos y de genes (items 1 y 2) contiene información sobre moléculas o secuencias particulares. El item 3, mapas de genomas, incorpora genes del anterior según su aparición en los cromosomas. En algunos casos, el

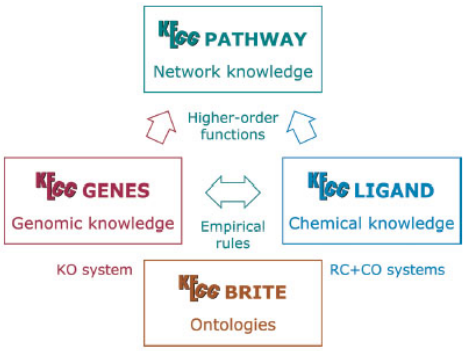


Figura 5.8: Estructura general de KEGG.

conocimiento de que un gen aparezca en un operón[4] concreto puede suministrar pistas de su función.

El ítem 4, mapa de pathways, describe redes potenciales de actividades moleculares, ya sean metabólicas o reguladoras. Un pathway metabólico de KEGG es una idealización correspondiente a un gran número de posibles cascadas metabólicas. Éstas pueden generar un pathway metabólico real de un organismo particular emparejando las proteínas del organismo con las encimas del pathway referenciado. En la figura 5.9 se muestra un mapa general de los pathways metabólicos almacenados en KEGG, donde cada nodo (círculo) identifica a un compuesto químico y cada línea que conecta dos nodos simboliza a una serie de reacciones. Uno de los mapas representados es el pathway "Citrate Cycle" (map:00020), el cual se muestra en detalle en la figura 5.10.

[4]Un operón se define como una unidad genética funcional formada por un grupo o complejo de genes capaces de ejercer una regulación de su propia expresión por medio de los sustratos con los que interaccionan las proteínas codificadas por sus genes.

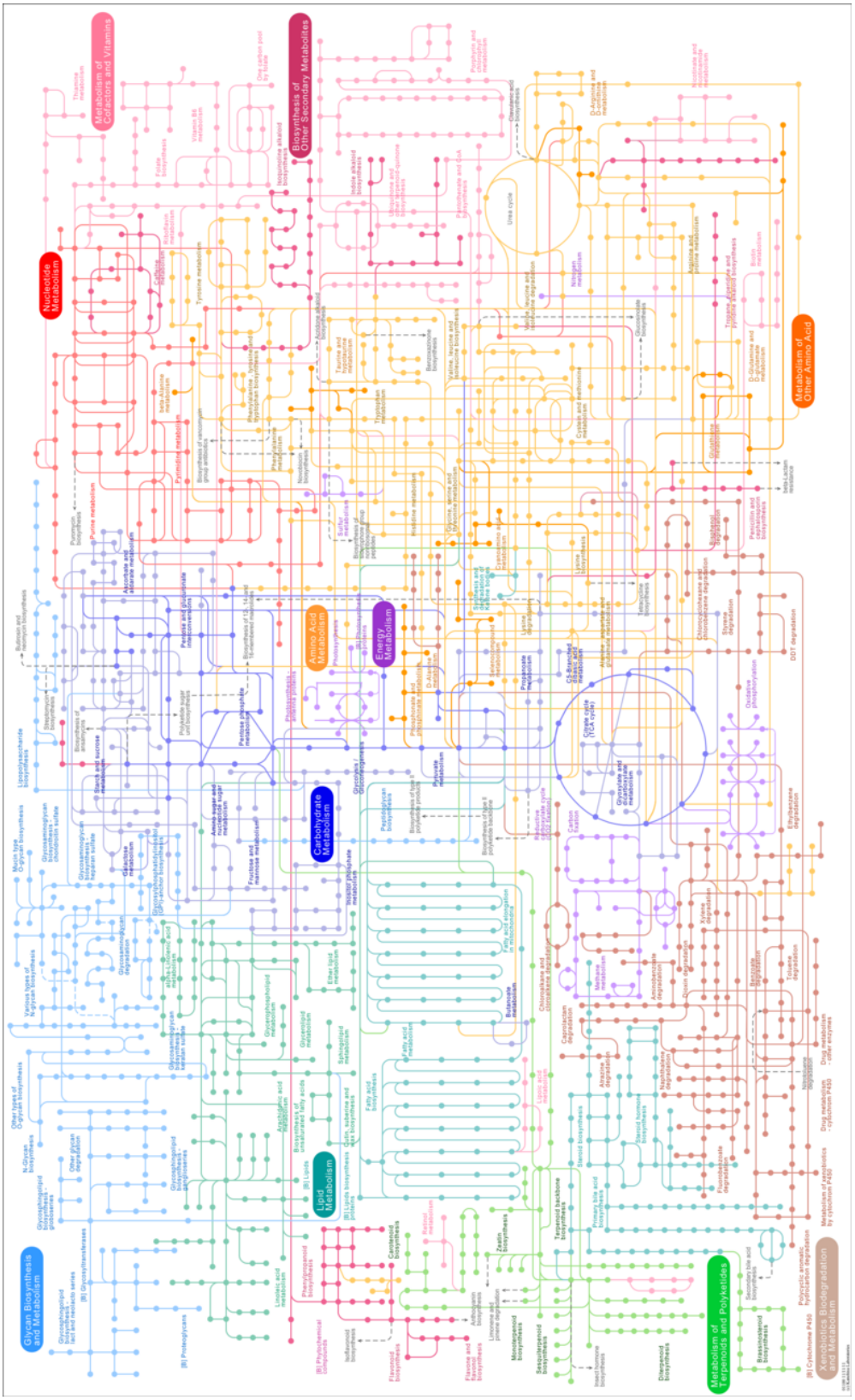


Figura 5.9: Representación general de los mapas de KEGG.

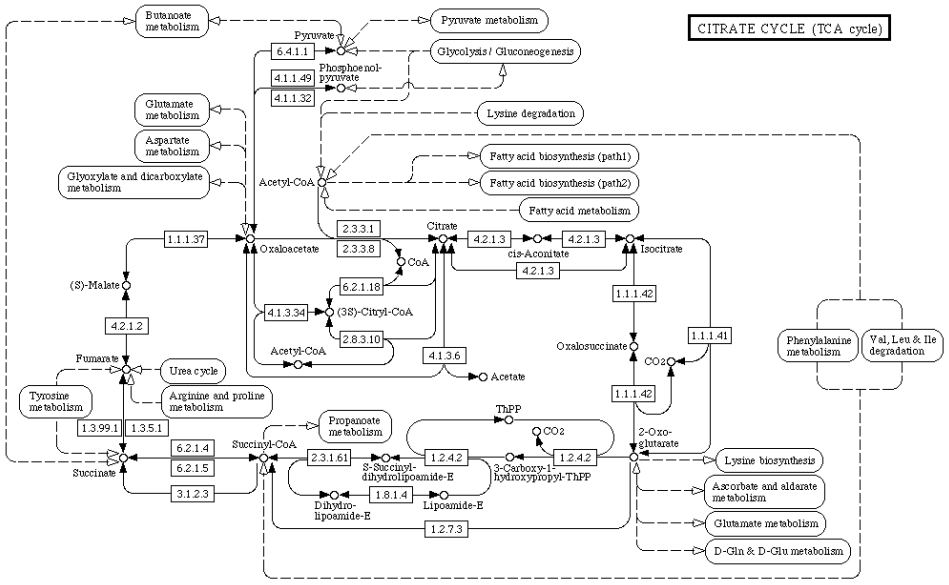


Figura 5.10: Representación del pathway Citrate Cycle (map:00020).

Una enzima en un organismo estaría referenciada en KEGG en sus tablas de homologías, ítem 5, las cuales vinculan el enzima a diferentes enzimas relacionados de otros organismo. Esto permite el análisis de relaciones entre pathways metabólicos de diferentes organismos.

5.3. Herramientas basadas en conocimiento previo

A partir de la aparición de GO, se han desarrollado una gran cantidad de herramientas basadas en el análisis funcional de dicha ontología.

A finales de 2001, Khatri et al. propusieron la primera herramienta de análisis automático basado en GO [235]. Desde 2003 hasta 2005 se propusieron otras 13 herramientas, las cuales, junto con esta primera, fueron estudiadas en [183]. El estudio, basado en la comparación de las características de las distintas herramientas, ponía de manifiesto que aunque todas éstas tengan el mismo enfoque general, se diferencian en muchos aspectos que influyen en la forma de analizar los resultados.

Tras el estudio realizado en 2005, han aparecido 22 herramientas relevantes sobre análisis estadístico de listas de genes basado en GO (ver figura 5.11). El año con mayor número de herramientas desarrolladas fue 2005 – 06 con 7, decreciendo en los siguiente años. Este decremento viene debido al cambio de enfoque que están experimentados las herramientas. Primeramente estaban diseñadas para traducir listas de genes diferentemente expresados a un perfil funcional capaz de ofrecer una idea del mecanismo celular en unas condiciones dadas, mientras que en los útimos años, aproximaciones como las realizadas por Prifti et al. [245, 246], tienen el objetivo de analizar las interacciones gen–gen. Nótese, que este último tipo de herramientas quedarían fuera del objetivo de este trabajo.

Seguidamente se presenta un estudio de comparación similar al desarrollado en [183] pero ampliado con las herramientas más relevantes desarrolladas hasta la fecha. El estudio se encuentra resumido en las tablas 5.2 y 5.3, y los criterios de comparación usados son descritos en detalle posteriormente.

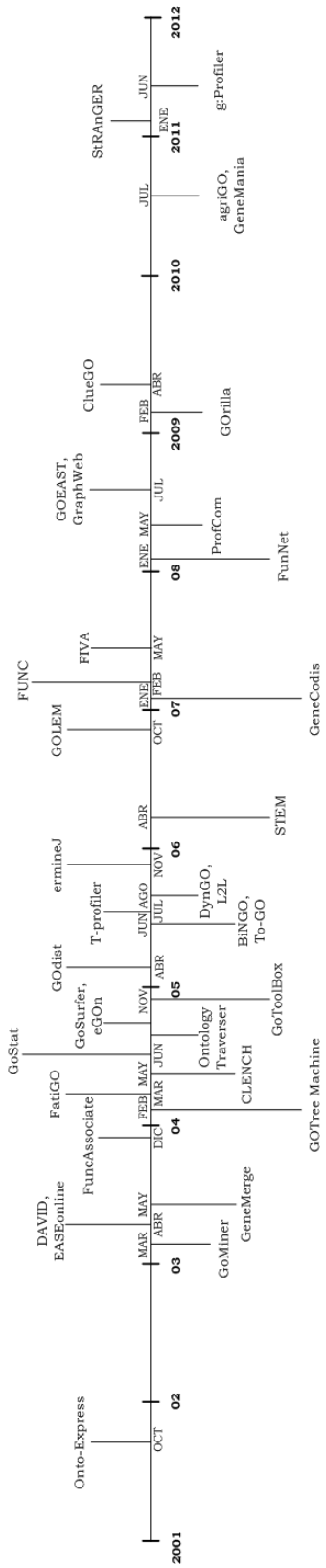


Figura 5.11: Evolución histórica de herramientas para el análisis funcional de expresión genética/microarrays basadas en GO.

Herramienta	Interfaz de de usuario	Tipo de aplicación	Plataforma	Ámbito del análisis[a]	Nivel de abstracción[b]	IDs soportados
Onto-Express [235]	Java GUI	Web-based	Cualquiera	Todas las categorías GO	Muy flexible; diferentes niveles de abstracción en los diferentes árboles GO	GenBank, UniGene, Entrez Gene, Affymetriz, Gene symbol
GoMiner [325]	Java GUI	Stand-alone	sólo Windows	Todas las categorías GO	Análisis global fijo	IDs específicos de GO
DAVID [91]	HTML GUI	Web-based	Cualquiera	Todas las categorías GO	Sólo niveles más bajos	GenBank, UniGene, Entrez Gene, Affymetrix, refSeq, UniProt, PIR
EASEonline [103]	HTML GUI	Ambos	Cualquiera	Todas las categorías GO	Sólo niveles más bajos	Affymetrix, GenBank, UniGene, Entrez Gene
GeneMerge [72]	HTML GUI	Ambos	Cualquiera	Una categoría	Sólo niveles más bajos	IDs específicos de GO
FuncAssociate [43]	HTML GUI	Web-based	Cualquiera	Todas las categorías GO	Sólo niveles más bajos	MODB
GOTM [327]	HTML GUI	Web-based	Cualquiera	Todas las categorías GO	Seleccionado por el usuario; nivel y análisis global fijo	Affymetrix, UniGene, ENSEMBL, Swiss-Prot, Entrez Gene
FatiGO [7]	HTML GUI	Web-based	Cualquiera	Una categoría	Análisis global	Affymetrix, GenBank
CLENCH [270]	(input) linea de comandos (output) HTML	Stand-alone	sólo Windows	Todas las categorías GO	Análisis global fijo	A.thaliana MIPS IDs
GOstat [31]	HTML GUI	Web-based	Cualquiera	Todas las categorías GO	Usuario selecciona nivel	GenBank, Unigene, gene symbol, IDs específicos de GO
GOToolBox [217]	HTML GUI	Web-based	Cualquiera	Todas las categorías GO	Usuario selecciona nivel	IDs específicos de GO
GoSurfer [331]	C/C++ GUI	Stand-alone	sólo Windows	Todas las categorías GO	Sólo niveles más bajos	Affymetrix, UniGene, Entrez Gene
Ontology Traverser [322]	HTML GUI	Web-based	Cualquiera	Una categoría	Sólo niveles más bajos	Affymetrix
eGOn [109]	HTML GUI	Web-based	Cualquiera	Una categoría	Sólo niveles más bajos	GenBank, UniGene, Clone
GOdist [37]	MATLAB GUI	Stand-alone	Cualquiera	Todas las categorías GO	Definidos por el usuario	Affymetrix, UniGene
BiNGO [212]	JAVA GUI	Stand-alone	Cualquiera	Una categoría	Sólo niveles más bajos	Entrez Gene ID, LocusTag

[a] ***Ámbito del análisis:*** se refiere al número de categorías GO que pueden analizarse simultáneamente.

[b] ***Nivel de abstracción:*** hace referencia a la profundidad de la anotación GO asociada a los datos de entrada. Nótese que algunas herramientas (e.g. GoMiner) permiten al usuario expandir y plegar el resultado, pero el análisis se realiza sólo una vez. Esto es descrito como un análisis global. El resto de columnas son auto-explicativas.

Herramienta	Interfaz de de usuario	Tipo de aplicación	Plataforma	Ámbito del análisis[a]	Nivel de abstracción[b]	IDs soportados
TO–GO [324]	JAVA GUI	Java Web Start	Cualquiera	Todas las categorías GO	Sólo los niveles más bajos de GO	Nomenclatura GO (id, nombre, definiciones, gene–products...) y expresiones regulares de ésta
T–profiler [55]	HTML GUI	Web-based	Cualquiera	Todas las categorías GO	Análisis global	Gene symbol de los organismos S.cerevisiae y C.albicans
DynGO [207]	Java GUI	Stand alone	Cualquiera	Todas las categorías GO	Sólo niveles más bajos	identificadores GO
L2L [230]	HTML GUI	Ambas	Cualquira	Una categoría	Sólo niveles más bajos	Affimetrix, Entrez Gene, Unique probe, HUGO name
ermineJ [194]	Java GUI	Stand alone	Cualquiera	Todas las categorías GO	Análisis global	Identificadores GO, Affymetrix y refSeg
STEM [112]	Java GUI	Stand alone	Cualquiera	Todas las categorías GO	Sólo niveles más bajos	Gene Symbol
GOLEM [267]	Java GUI	Ambas	Cualquiera	Todas las categorías GO	Definidos por el usuario	Identificadores GO
GENECODIS [70]	HTML GUI	Web-based	Cualquiera	1 ó varias categorías	Sólo nivel más bajo	Ensembl, EntezGene, RefSeq, UniProt/SwitProt, GeneName, GeneId, UniGene, HGNC ...
FUNC [248]	Línea de comandos y HTML GUI	Ambas	UNIX y GNU Linux para stand-alone y cualquiera para web–service	1 ó varias categorías	Seleccionado por el usuario	RefSeq con GO ID
FIVA [51]	Java GUI	Stand alone	Cualquiera	Todas las categorías GO	Sólo los niveles más bajos	Locus tags, Gene Symbol, SwitProt, InterPro, Identificadores GO
FunNet [149]	HTML GUI	Web-based	Cualquiera	1 ó varias categorías	Sólo niveles más bajos	EntrezGene
ProfCom [15]	HTML GUI	Web-based	Cualquiera	Todas las categorías	Sólo niveles más bajos	Gene Symbol, Ensembl, LocusTag, RefSeq, UniProt/Swiss-Prot, UniGene y Affimetrix
GOEAST [330]	HTML GUI	Web-based	Cualquiera	Todas las categorías GO	Sólo niveles más bajos	Affymetrix, target name, probe ID, gene/protein ID (NCBI, RefSeq, Ensembl ...)

Herramienta	Interfaz de de usuario	Tipo de aplicación	Plataforma	Ámbito del análisis[a]	Nivel de abstracción[b]	IDs soportados
GraphWeb [254]	HTML GUI	Web-based	Cualquiera	Todas las categorías GO	Sólo niveles más bajos	geneId de Ensembl
GOrilla [106]	HTML GUI	Web-based	Cualquiera	1 ó todas las categorías	Análisis global	gene symbol, RefSeq, Uniprot, Unigene y Ensembl
ClueGO [48]	Java GUI	Stand-alone	Cualquiera	1 ó varias categorías	Análisis global	AffymetrixID, AccessionID, SymbolID
agriGO [100]	HTML GUI	Web-based	Cualquiera	Todas las categorías GO	Análisis global	Affymetrix ,Agilent, BGI
GeneMANIA [304]	HTML GUI	Web-based	Cualquiera	Biological Process	Análisis global	Ensembl, Entrez, UniProtKB y RefSeq
StRAnGER [74]	Java GUI	Web-based	Cualquiera	Todas las categorías GO	Análisis Global	Ensembel
g:Profiler [252, 253]	HTML GUI	Web-based	Cualquiera	Todas las categorías GO	Diferentes niveles de abstracción	Cientos de Identificadores

Tabla 5.2: Herramientas sobre el análisis Ontológico de datos de expresión genómica (1).

Herramienta	Modelo Estadístico	Corrección para múltiples experimentos	Visualización GO [c]	Microarrays soportados	Cometido principal
Onto-Express	χ^2, binomial, hipergeométrica, test de Fisher	*Šidák*, Holm, Bonferroni, FDR	Flat, Tree	172 arrays comerciales (Afymetrix, SuperArray, Sigma-Genonsys, ClonTech, PerkinElmer, Operon, Takara, NIA); también soporta listas definidas por el usuario	Buscar en las bases de datos públicas y devolver una tablas que correlacionan la expresividad con la localizaciones de genes citogenéticos, la función molecular y bioquímica, el proceso biológico, componente celular y roles celulares en la translación de proteinas
GoMiner	test de Fisher	Enriquecimiento relativo	Tree, DAG	Suministrado por el usuario	Interpretación biológica de listas de genes relevantes
DAVID	Ninguno	Ninguno	No disponible	No aplicable	Anotación y análisis de bases de datos genómicas.
EASEonline	test de Fisher	Bonferroni	No disponible	27 arrays (Afymetrix), también soporta listas definidas por el usuario	Interpretación biológica y descubrir enriquecimientos biológicos a partir de listas de genes
GeneMerge	Hipergeométrica	Bonferroni	Flat	Suministrado por el usuario	Aportar información genómica funcional y proporcionar ranking estadísticos para sobre–representaciones de categorías o funciones particulares.
FuncAssociate	test de Fisher	Ninguno	No disponible	Suministrado por el usuario	Devolver listas de atributos GO sobre/infra–representados a partir de una lista de genes
GOTM	Hipergeométrica	Ninguno	Tree	37 arrays (Afymetrix), listas definidas por el usuario	Análisis y visualización de listas de genes relevantes basada en la jerarquía GO
FatiGO	Porcentaje	Step–down MinP, FDR[38], FDR[39]	Flat, Tree	Suministrado por el usuario	Asignar información funcional representativa a un conjunto de genes.
CLENCH	Hipergeométrica, χ^2, binomial	Ninguno	DAG	Suministrado por el usuario	Recuperar anotaciones GO a partir de TAIR y calcular el enriquecimiento de GO–terms.

[c] ***Visualización GO:*** '*flat*' indica que la herramienta no representa la estructura jerárquica de GO cuando muestra el resultado, '*tree*' indica que la herramienta sí muestra dicha estructura como un árbol, mientras que '*DAG*' indica que la herramienta muestra el resultado como un grafo dirigido acíclico.

Herramienta	Modelo Estadístico	Corrección para múltiples experimentos	Visualización GO [c]	Microarrays soportados	Cometido principal
GOstat	χ^2,test de Fisher	FDR,Holm	No disponible	Suministrado por el usuario	Determinar significatividad sobre o infra representado de categorías de GO estadísticamente.
GOToolBox	Hipergeométrica, binomial, test de Fisher	Bonferroni, Holm, Hochberg, Hommel, FDR	No disponible	Suministrado por el usuario	Determinar significatividad sobre o infra representado de términos GO de forma estadística.
GoSurfer	χ^2	q-value	DAG	22 arrays (Afymetrix), listas definidas por el usuario	Análisis de genes obtenidos en computaciones de genomas, análisis de microarrays o cualquier otro método similar.
Ontology Traverser	Hipergeométrica	FDR	No disponible	5 arrays (Afymetrix), listas definidas por el usuario	Herramienta de enriquecimiento de lista de genes de micorarrays.
eGOn	Binomial	Ninguno	Tree	Suministrado por el usuario	Mapea datos de microarrays a la estructura GO; posee técnicas de visualización, filtrado, definición de anotaciones GO, análisis estadístico, enlace a bases de datos externas
GOdist	Hipergeométrica, test de Fisher	Ninguno	No disponible	Suministrado por el usuario	Analiza datos de expresión mediante el método de Kolmogorov–Smirnov
BiNGO	Hipergeométrica, binomial	Bonferroni, FDR	DAG	GOSlim ontologies, Suministrado por el usuario	Determinar las categorías GO sobre–representadas a partir de un conjunto de genes.
TO–GO	Ninguno	Ninguno	Tree	Suministrado por el usuario	Mostrar conocimiento biológico de un conjunto de genes a partir de la estructura GO
T–profiler	t–value	Bonferroni (E–value)	No disponible	153 *motif group* (secuencias reguladoras de C. albicans y elemento regulador de S.cerevisiae); también soporta listas suministradas por el usuario	Calcular t–test para puntuar cambios en la actividad media en los grupos predefinidos de genes

Herramienta	Modelo Estadístico	Corrección para múltiples experimentos	Visualización GO [c]	Microarrays soportados	Cometido principal
DynGO	ninguno	Ninguno	Tree	Suministrada por el usuario	Calcular similitud semántica entre GO–terms
L2L	Binomial	Bonferroni, FDR, Simulación	Ninguno	Suministrada por el usuario	Descubrir significatividad biológica oculta en datos microarray
ermineJ	pValue sobre análisis ORA, GSR y ROC	FDR, Bonferroni, Resampling	Tree	Definida por el usuario, 30 arrays de ratón, hombre y rata; y los ficheros genéricos de refSeq	Análisis de grupos de genes
STEM	Binomial, Hipergeométrica	Bonferroni	No disponible	Definida por el usuario	Agrupamiento, comparación y visualización de series temporales pequeñas sobre datos de expresión genética
GOLEM	Hipergeométrica	Bonferroni, FDR	DAG	Definida por el usuario y ficheros de anotación de GO	Visualización del DAG de GO y búsqueda de GO–terms enriquecidos
GENECODIS	Hipergeométrica, χ^2	FDR y permutaciones	No disponible	Suministrado por el usuario	Análisis funcional de lista de genes
FUNC	Hipergeométrica, Wilconxon Rank Text, Binomial, 2x2 contingency	FWER, FDR	No disponible	Suministrado por el Usuario	Analizar datos de expresividad asociados con genes o gene–products.
FIVA	Test Fisher	Bonferroni, Bonferroni Step Down, FDR, Benjamini	No disponible	Suministrado por el usuario o ficheros de anotaciones genómicas o transcriptomas (e.g. Ensembl ó GeneBank)	Identificar procesos biológicos relevantes a partir del *transcriptome analysis*
FunNet	Test Fisher	FDR	No disponible	Arrays del tejido adiposo blanco	Analizar redes de genes coexpresados
ProfCom	Hipergeométrico, binomial y χ^2	Ninguno	No Disponible	Suministrado por el usuario	Interpreta funcionalmente a un conjunto de datos empleando GO y funciones complejas extraídas de los términos GO disponibles.
GOEAST	Hipergeométrica, Fisher, χ^2	FDR, Hochberg, Bonferroni, Hommel, Yekutieli	DAG	Affymetrix, Illumina, Agilent y definidas por el usuario	Análisis de resultados experimentales sobre microarrays de hibridación

Herramienta	Modelo Estadístico	Corrección para múltiples experimentos	Visualización GO [c]	Microarrays soportados	Cometido principal
GraphWeb	Test exacto de Fisher	Simulación (proporcionada en g:SCS)	No disponible	PPI de IntAt, HPRD y redes reguladoras de S.Cerevisiae	Detección de módulos a partir de redes biológicas, heterogéneas y multiespecies, y la interpretación de los modulos detectados usando GO
GOrilla	mHG p–value [105]	multiHG [104]	DAG	Listas ordenadas suministradas por el usuario	Identificar términos GO enriquecidos en una lista de genes empleando un límite estadístico flexible
ClueGO	Hipergeométrica, mid–P–values y *doubling* [257]	Bonferroni, Bonferroni step-down and Benjamini–Hochberg	No Disponible	Suministrado por el usuario	Es un plugin de Cytoscape para visualizar términos biológicos no redundantes a partir de grandes listas de genes en una red funcional.
agriGO	Hipergeométrico, Fisher y χ^2	FDR, q-value, Holm	DAG	292 arrays provenientes de 45 especies agrícolas	Herramienta para el análisis de datos agrícolas que contiene cuatro modulos: SEA (Singular enrichment analysis), PAGE (Parametric Analysis of Gene set Enrichment), BLAST4ID (Transfer IDs by BLAST) y SEACOMPARE (Cross comparison of SEA)
GeneMania	Ninguno	FDR	No Disponible	S.cerevisiae, C. elegans, D. melanogaster, M. musculus, A. thaliana y H. sapiens	Predice la funcionalidad de un conjunto de genes usando, para ello, un enorme conjunto de datos sobre asociación funcional (interacciones de proteínas y genes, pathways, co-expressiones, co-localizaciones y dominios de similitud de proteinas)
StRAnGER	Hipergeométrico, Fisher y χ^2	FDR	Tree	Suministrado por el usuario	Análisis funcional de grandes bases de datos genómicas usando la información almacenada en GO y KEGG

Herramienta	Modelo Estadístico	Corrección para múltiples experimentos	Visualización GO [c]	Microarrays soportados	Cometido principal
g:Profiler	Hipergeométrico	Bonferroni, SCS, FDR	Tree	85 especies de Ensembl y Ensembl genomes, mammals, fungi, insects, plants ...	Conjunto de herramientas web para el análisis funcional de listas de genes. Se destacan los módulos *g:GOSt* y *g:Cocoa* que combinan métodos para interprestar lista de genes en el contexto de ontologías biomédicas, pathways, factores de transcripción e interacciones proteína-proteína; *g:Convert* para mapeo de ID; *g:Sorter* como buscador similitudes de expresión genética y *g:Orth* enfocado a encontrar homologías entre genes.

Tabla 5.3: Herramientas sobre el análisis Ontológico de datos de expresión genómica (2).

5.3.1. El modelo estadístico

El análisis ontológico puede realizarse con diferentes modelos estadísticos incluyendo el hipergeométrico [77], binomial, X^2 (chi-cuadrado)[118], y el test de Fisher [214]. La probabilidad de que una cierta categoría se encuentre x veces en la lista de genes independientes puede ser modelado por una distribución hipergeométrica. Sin embargo, el cálculo de dicha distribución es más costoso cuando se refiere a grandes conjuntos de datos (ej. Affymetrix HGU133A). No obstante, la distribución hipergeométrica tiende a una distribución binomial cuando el número de genes es grande. De esta forma, el modelo binomial puede ser usado cuando el conjunto de datos es numeroso. Un enfoque alternativo puede ser el uso de X^2 o el test de Fisher. En la mayoría de los casos, las diferencias entre los modelos no serán drásticas. Estos tests son tratados en detalle en la literatura [97, 98, 99].

TO–GO y DynGO no generan modelos sino que están especialmente diseñado para la exploración del árbol GO, aunque este último aporta una medida de similitud entre GO–terms basada en el trabajo realizado por Lord et al. [208](ver apartado 5.5). FatiGO tampoco usa un modelo estadístico como tal, pero calcula porcentajes con respecto a los genes anotados en GO o a todos los conocidos en un organismo. GeneMania tampoco aporta un modelo estadístico concreto sino que calcula la cobertura de los genes de entrada para cada Go term y calcula su FDR. T–profiler genera el t–value(t_G), el cual es calculado según la ecuación 5.1, donde μ_G y $\mu_{G'}$ son la media de las expresiones de los N_G y $N_{G'}$ genes en los grupos G y G' respectivamente.

$$t_G = \frac{\mu_G - \mu_{G'}}{s\sqrt{\frac{1}{N_G} + \frac{1}{N_{G'}}}}; s = \sqrt{\frac{(N_G - 1) \times s_G^2 + (N_{G'} - 1) \times s_{G'}^2}{N_G + N_{G'} - 2}} \tag{5.1}$$

La propuesta ermineJ calcula el p–value a partir de una hipótesis nula concreta en cada una de los tipos de análisis que propone (ORA, GSR y ROC). GOrilla calcula el p–value usando una caracterización de la distribución de los datos, denominada mHG [105], que permite el análisis estadístico de miles de genes y términos GO en segundos. ClueGO, además de proporcionar diferentes test basados en distribución hipergeométrica, aporta la posibilidad de calcular mid–P–values y *doubling* para test bilaterales con el fin de lidiar con los efectos del discrecionismo y conservadurismo según Rivals et al. [257].

Por otro lado, GoMiner, EASEonline, GeneMerge, FuncAssociate, GOTree Machine (GOTM), GOSurfer, Ontology Traverser, eGOn, L2L, GOLEM, FIVA y GraphWeb sólo soportan un test estadístico, aunque este último aporta un nuevo análisis denominado iGA [61]. FunNet, al igual que las herramientas anteriores, sólo suministra un análisis estadístico (test de Fisher), aunque éste puede ser apli-

cado a tres tipos de análisis diferentes: análisis funcional convencional, análisis funcional de redes de transcripción y la estimación del umbral de coexpresión. GOstat, GOdist, GENECODIS, BiNGO y STEM proporciona al usuario elegir entre dos tests (X^2 y Fisher en GOstat; Hipergeométrico y Fisher para GOdist; Hipergeométrico y X^2 en GENECODIS, y Binomial e Hipergeométrico para BiNGO y STEM). CLENCH y GOToolBox permiten una elección entre tres tests (X^2, hipergeométrico y binomial para CLENCH e hipergeométrico, binomial y Fisher para GOToolBox); agriGO y StRAnGER proporcionan Hipergeométrico, Fisher y χ^2 ; mientras que Onto-Express implementa los cuatro tests (X^2, hipergeométrico, binomial y Fisher). FUNC soporta Hipergeométrico y Binomial, además de ofrecer dos nuevos test de análisis: *Wilcoxon Rank Test* y *2x2 contingency*. Por último, se destaca la herramienta g:Profiler que, además de aportar el test Hipergeométrico, presenta en su módulo g:Cocoa para realizar análisis comparativos de múltiples listas de genes.

5.3.2. El conjunto de genes de referencia

Una consideración importante cuando identificamos el significado de los términos GO estadísticamente es la elección de la lista de referencia de genes, los cuales son usados para el cálculo de los p–values [270] de cada uno de los términos.

Numerosas herramientas, tales como GOToolBox, GOSTAT, GoMiner, FatiGO, GOLEM, GeneCodis, GOTM[5] y FIVA, usan el total de los genes de un genoma como entrada [217, 31, 325, 327] o el conjunto de genes con anotación GO [7, 267]. Sin embargo, estas herramientas realizan un estudio erróneo cuando la entrada es una lista de genes independientes extraídos de un experimento con microarrays; los genes que no estén presentes en un microarray no tienen ocasión de ser seleccionados.

La idea fundamental es asignar una importancia a varias categorías funcionales comparando el número de genes observados en una categoría específica con el número de genes que aparecerían en la misma categoría si fueran seleccionados aleatoriamente. Si a la hora de seleccionar aleatoriamente los genes se considera el genoma completo, estaríamos incluyendo todos los genes del genoma. Sin embargo, si no partimos del genoma completo y sí de un subconjunto de él, los genes no tendrían la misma probabilidad de ser elegidos aleatoriamente ya que un gen que no está en un array no puede ser seleccionado. Esto representa una flagrante contradicción de la suposición de los modelos estadísticos usados.

[5]GOTM también permite al usuario seleccionar su propia lista de genes o usar uno de los 37 arrays en formato affymetrix que tiene almacenados como conjunto de genes de referencia.

5.3.3. Corrección de múltiples experimentos

Otro factor crucial en el cálculo de una categoría funcional es la corrección para múltiples experimentos (ver capítulo 9 de [97]). Este tipo de corrección debe ser realizada en todas las situaciones donde la categoría funcional no es seleccionada *a priori* y se consideran muchas categorías simultáneamente.

Algunas de las herramientas revisadas no llevan a cabo tal corrección: GoMiner, DAVID, GOTM, CLENCH, eGOn, GOdist, DynGO y ProfCom. GoMiner provee un 'enriquecimiento relativo' calculado estadísticamente como $R_e = (n_f/n)/(N_f/N)$, donde n y N son el número de genes seleccionados y referenciados respectivamente, y n_f y N_f son el número de genes seleccionados y referenciados, respectivamente, que se encuentran en la categoría funcional estudiada ([325]). Sin embargo, esta medida no puede usarse en ningún caso como una corrección de múltiples experimentos[6]. En realidad, esta magnitud debe usarse para decidir si una categoría es significativa o no, más que como un 'enriquecimiento relativo'.

El resto de herramientas proponen algún tipo de solución al problema de comparación múltiple. EASEonline, GeneMerge y T–profiler soportan la corrección de Bonferroni, donde éste último la usó para generar el estadístico E–value. Las correcciones de Bonferroni y *Šidák* [1] son muy apropiadas en diferentes situaciones, en particular, cuando no se consideran muchas categorías funcionales (e.g., menos que 50). Sin embargo, estas técnicas de correcciones son conocidas por ser demasiado conservativas si se consideran más categorías ([97]). Una familia de métodos que permiten un menor ajuste conservativo de los p-values es el grupo de métodos descendentes de Holm ([154, 155, 157]).

Las medidas de Bonferroni, *Šidák*, y Holm [156] son procedimientos estadísticos que presuponen que las variables son independientes, lo cual se sabe que es falso en este tipo de análisis[7]. Cuando se sabe que existen dependencias, los métodos como el descubrimiento de la tasa de error (FDR) son más apropiados ([38, 39, 97]).

Las herramientas que ofrecen más de un método de corrección permiten al investigador adaptar el análisis al número de categorías y al grado de dependencias conocidas entre ellas. Bonferroni y *Šidák* son adecuados si las categorías consideradas no están relacionadas directamente. Si la mayoría de las categorías no relacionadas son seleccionadas, Holm sería una buena solución. Si hay muchas categorías funcionales claramente relacionadas, FDR es, probablemente, la mejor elección. Si las dependencias son muy fuertes (e.g. muchos subprocesos que forman parte de un proceso superior), debería seleccionarse las técnicas de bootstrap

[6] Nótese que esta estadística no considera el número de experimentos realizados en paralelo.

[7] La mayoría de la jerarquía de GO muestra en este tipo de análisis que muchas categorías biológicas están fuertemente relacionadas, algunas veces como hijos del mismo nodo del nivel superior.

o Monte-Carlo para capturar esas dependencias.

Las dos herramientas que sobresalen respecto de este criterio son FuncAssociate y GOrilla. FuncAssociate usa una simulación muy original de Monte-Carlo, mientras que en GOrilla es usada una puntuación hipergeométrica multidimensional (multiHG) presentada por Egen en [104]. GraphWeb procura una corrección basada en la simulación proporcionada en g:SCS [253]. FatiGO y GOstat implementan las correcciones de Holm y FDR. BiNGO y GOLEM suministran Bonferroni y FDR. L2L y ermineJ aportan, además de las dos correcciones anteriores, una nueva corrección basada en la simulación de diferentes ejecuciones. L2l también suministra el cálculo del p–value ajustado a partir de las frecuencias de las ocurrencias en los resultados aleatorios. Fiva aporta Bonferroni, Bonferroni step-downn y FDR. ClueGO soporta Bonferroni, Bonferroni step-down y Benjamini-Hochberg. OntoExpress ofrece Bonferroni, *Šidák*, Holm y FDR. Mientras que GOToolBox procura las correcciones de FDR, Bonferroni, Holm, Hochberg y Hommel; y GOEAST, además de los test FDR, Hochberg, Bonferroni y Hommel, proporciona uno basado en FDR bajo dependencias (Yekutieli).

5.3.4. Ámbito del Análisis

Un factor importante en la valoración de la utilidad de una herramienta es su habilidad en proporcionar una imagen completa del fenómeno estudiado. En términos del estudio funcional usando GO, un análisis completo debería incluir las tres categorías principales de GO: función molecular, proceso biológico y componente celular, además de otra información disponible. Entre las herramientas revisadas, eGOn, FatiGO, GeneMerge, Ontology Traverser, BiNGO y L2L sólo analizan una categoría en cada ejecución. Las otras herramientas permiten analizar las tres categorías simultáneamente. Además de estas características GeneMerge, OntoExpress, GENECODIS, FunNet y StRAnGER muestran información de pathways usando Kegg, siendo para estas dos últimas una característica fundamental en su análisis de redes de genes coexpresados. GraphWeb, para permitir un análisis más completo, está asociado a Kegg y a Reactome. GO–TO añade información de los gene–products al estar enlazado con TrEMBLM; DynGO está asociado con PIRSF [310], un sistema de clasificación de familias de proteínas; FIVA puede usar información funcional de pathways metabólicos (Kegg), clases COG, interacciones reguladoras, claves Uniprot e InterPro; y ClueGO emplea la información almacenada en GO, Kegg y BioCarta.

5.3.5. Capacidad de Visualización

Como se comentó anteriormente, GO está organizado como un grafo dirigido acíclico (DAG). A diferencia de las estructuras de árbol, un DAG permite que un

nodo tenga diferentes padres. Sin embargo, la estructura DAG no es la mejor elección para la navegación. Un camino alternativo para visualizar estructuras de grafos es representarlo y visualizarlo en un árbol diferentes veces, cada vez bajo un padre. Cualquier herramienta basada en GO debe ser capaz de representar las relaciones jerárquicas entre las diferentes categorías funcionales, ya que permiten al usuario un mejor entendimiento del fenómeno estudiado. Además, el análisis funcional puede ser continuado y refinado explorando ciertos subgrafos de la estructura GO.

Entre las herramientas revisadas, DAVID, EASEonline, FuncAssociate, GOstat, GOTollBox, OntologyTraverser, GOdist, T–profiler, L2L, STEM, GENECODIS, FUNC, FIVA, FunNet, ProfCom y GraphWeb no muestran el resultado en el contexto de la estructura jerárquica de GO. Mientras que Onto-Express, eGOn, FatiGO, CLENCH, GoMiner, GOTM y GOrilla sí lo hacen. Además, Onto-Express, GOMiner y GOTM permiten al usuario la opción de expandir y contraer los nodos, e, incluso Onto-Express es capaz de realizar operaciones de ordenación y búsqueda en la estructura que, de forma automática y si es necesario, expande y/o contrae los nodos pertinentes. TO–GO es capaz de realizar búsquedas en el árbol de GO incorporando la posibilidad de introducir expresiones regulares de cualquier nomenclatura usada en GO (identificador, nombre o definición de GO–term y nombre o evidencia de gene–product). BiNGO y GOLEM, además de mostrar la información sobre el grafo de GO, dan la opción de buscar y señalar conceptos concretos de la ontología mostrada.

ClueGO, GeneMANIA y g:Profiler son las únicas herramientas de las revisadas que proponen una nueva forma de representar el resultado obtenido. Éstas, en vez de usar la estructura de GO, genera una red genética basada en el análisis funcional llevado a cabo.

5.3.6. Nivel de abstracción

Como ya hemos visto, los genes que componen la estructura jerárquica de GO son anotados en diferentes niveles de abstracción (figura 5.12). Por ejemplo, ‘induction of apoptosis by hormones’ (inducción de la apoptosis con hormonas) es un tipo de ‘induction of apoptosis’ el cual a su vez es parte de ‘apoptosis’. La apoptosis representa una mayor nivel de abstracción, más general, mientras que la inducción de la apoptosis con hormonas representa a un menor nivel de abstracción, más específico. Cuando los genes se anotan usando la terminología GO se realiza el esfuerzo de anotar los genes con el mayor nivel de detalle posible, el cual corresponde con el menor nivel de abstracción. Por ejemplo, si se sabe que un gen induce a la apoptosis como respuesta a unas hormonas, éste será anotado con el término ‘induction of adoptosis by hormones’ y no simplemente como un término de mayor nivel, tal como ‘induction of apoptosis’ o ‘apoptosis’.

Un aspecto muy valorable de una herramienta que estudia la relevancia biológica es que permita al usuario seleccionar un nivel de abstracción. Desde este punto de vista, las herramientas revisadas se encuentran en una de las tres siguientes categorías:

- La primera categoría incluye las herramientas capaces de realizar el análisis sólo con términos específicos asociados a cada gen. Esto corresponde a un análisis que escoge los menores niveles posibles de abstracción o los más específicos (ver línea discontinua, punto-raya, en la figura 5.12). Este tipo de análisis es esencialmente una búsqueda particular de la base de datos usada y no puede ser usado como respuesta a una pregunta biológica ni ser refinado de ninguna forma.

- La segunda categoría incluye aquellas herramientas que posibilitan seleccionar una profundidad determinada, o el nivel de abstracción en GO. Una vez el nivel esté seleccionado, este tipo de herramientas considerarán cualquier gen que se encuentre debajo del nivel escogido. Esta situación es ilustrada por la linea discontinua, raya-raya, en la figura 5.12. Se debe tener en cuenta que cada categoría es propagada ascendentemente por todos sus padres, siguiendo la estructura DAG y no la estructura de árbol usada como visualización. La capacidad de elegir una profundidad determinada permite refinar el análisis con sucesivas repeticiones usando diferentes niveles de abstracción. Esto fuerza a varios términos específicos a ser agrupados en uno más general, y, quizás, en categorías más informativas. Cuando esto se realiza, muchos genes que son asociados con categorías muy específicas (e.g. inducción de la apoptosis) son ahora agrupadas juntas bajo una categoría más general, tal como 'positive regulation of apoptosis'. Incluso es la causa de que cada categoría específica no parezca ser significativa ya que hay sólo unos pocos genes asociados con ella, mientras que categorías más generales llegan a tener mayor relevancia una vez que todos los genes asociados con la subcategoría más específica son analizados juntos como representantes de la categoría superior. Las herramientas que tienen esta capacidad aportan un análisis más complejo y detallado que puede ser utilizado para plantear cuestiones biológicas específicas.

- Finalmente, la tercera categoría incluye herramientas que permiten especificar con mayor exactitud el nivel de abstracción, permitiendo diferentes niveles en diferentes direcciones. Si el análisis es realizado mezclando profundidades del nivel 9, por ejemplo, el análisis puede distinguir entre varios subtipos de inducción de apófisis: por hormonas, por señales extracelulares, por señales intracelulares, etc (figura 5.12). Sin embargo, para una profundidad

fija, el mismo análisis será también realizado sobre otras miles de categorías funcionales situadas en el mismo nivel. Si los resultados son presentados en diagramas de barras, las categorías interesantes serán abarrotados por todas las categorías extras que se encuentran en la misma profundidad de GO, aunque, incluso, no sean interesantes para el investigador. Además, otro fenómeno sería erróneo, ya que el nivel escogido debe ser también específico para aquellas categorías de GO. Una herramienta que permita un control completo del nivel de abstracción es más potente, ya que permitirá al usuario realizar un análisis a diferentes profundidades en varias partes de la jerarquía de GO. Esto es ilustrado en la figura 5.12 con una línea continua.

La mayoría de las herramientas existentes sólo permiten un análisis en los niveles más bajos de GO y no permiten más refinamiento. Entre las herramientas revisadas, FatiGO, EASEonline, GOToolgBox, GOstat, ermineJ, STEM, FUNC o FIVA permiten seleccionar un nivel específico de abstracción enviando, previamente, la lista de genes de entrada. FatiGO, CLENCH, GOMiner y GOrilla también calculan un p-value para todos los nodos de GO. Esto corresponde a un análisis estadístico global en donde todos los genes bajo un cierto nodo son consideraros para ser asociados con ese nodo. BiNGO, permite realizar el estudio sobre un subconjunto de los nodos GO (GOslim). T–profiler realiza el análisis usando grupos de genes que se encuentren en categorías con más de seis genes asociados.

Onto-Express, GOdist y GOLEM son, en este momento, las únicas herramientas que permites un análisis completo, permitiendo un análisis sobre un nodo concreto. GOdist y GOLEM permite análisis hipergeométricos sobre categorías específicas o globales. Mientras que Onto-Express permite que cualquier nodo sea contraído o expandido en GO. Contraer un nodo es equivalente a reasignar a este nodo todos los genes asociados con cualquiera de sus descendientes. El p-value calculado a partir de un nodo contraído en Onto-Express corresponde con el p-value calculado en el análisis estadístico global de GoMiner y FatiGO. Expandiendo un nodo se distinguirá entre genes asociados con el propio nodo y los genes asociados con cualquiera de sus descendientes. El p-value de un nodo expandido estará basado sólo en los genes asociados directamente con él. Este p-value no es proporcionado por ninguna otra herramienta de las revisadas. Un inconveniente en Onto-Express es que si un usuario desea realizar el análisis sobre una profundidad fija de GO, el usuario es requerido para expandir manualmente los nodos superiores a ese nivel.

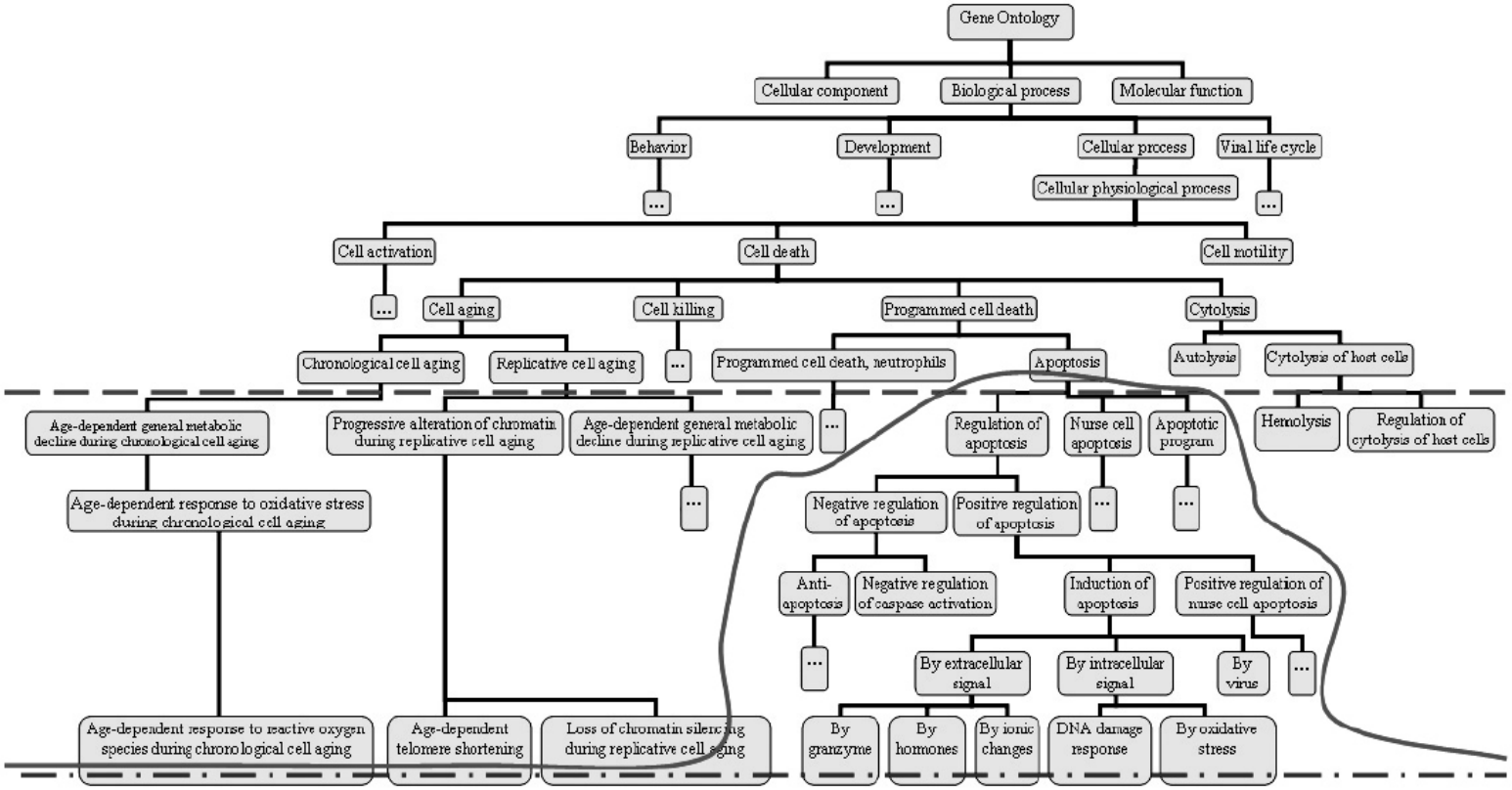


Figura 5.12: Nivel de abstracción. *El análisis puede ser realizado en los menores niveles de abstracción (linea discontinua; punto-raya), en un nivel intermedio elegido por el usuario (linea discontinua; raya-raya) o por un nivel predeterminado que pueda ir a diferentes profundidades en varios subárboles de GO (línea continua).*

5.3.7. Prerrequisitos e instalación

Otro factor importante es la cantidad de pasos necesarios para instalar y usar una herramienta. Las herramientas basadas en servicios web permiten a los experimentos biológicos una solución conveniente para esquivar los problemas usualmente asociados con la instalación local de un programa ([327]). Por otra parte, las herramientas disponibles en la web deberían estar inicialmente bloqueadas por razones de seguridad. Por ejemplo, si la herramienta usa un puerto TCP/IP específico y el investigador está tras un firewall, el puerto requerido deber estar abierto en el firewall antes de usar la herramienta.

Las herramientas que se ejecutan en local *stand-alone*, tales como CLENCH, GoMiner, GoSurfer o GOdist fuerzan al usuario a entender el complejo proceso de instalación (nótese que los usuarios de estas herramientas suelen ser biólogos con una habilidad informática no muy desarrollada). Por ejemplo, los prerrequisitos de CLENCH incluyen una instalación previa de distintos módulos de Perl. Otro ejemplo puede ser GoMiner, que requiere una instalación de un plug-in de Adobe para mostrar gráficamente los resultados, aunque hay que decir que esta herramienta puede funcionar sin los plug-in. BiNGO o ClueGO que son un plug-in de Cytoscape [19], o GOdist, que basa sus componentes en una instalación previa de Matlab.

En principio, las herramientas que puede ser ejecutadas online (Onto-Express, EASEonline, DAVID, GeneMerge, Ontology Traverser, GOTM, FuncAssociate, FatiGO, TO–GO, T–profiler DynGO, GraphWeb o GOrilla) sólo necesitan que el usuario posea un navegador web y una conexión a internet. En la práctica, este tipo de herramientas también necesitan algún tipo de compatibilidad con las plataformas usadas. Por ejemplo, la *máquina virtual de microsoft* incluida en el navegador *internet explorer* no es compatible 100 % con el estándar de Java [206]. En consecuencia, algunas herramientas basadas en Java (e.g. Onto-Express), requerirá la instalación de *Sun Java Runtime Environment*.

Otra cuestión es relativa a la disponibilidad de la herramienta y la necesidad de una conexión a internet. Las herramientas online pueden ser usadas en cualquier computadora, pero no pueden ser usadas sin conexión a Internet. Las herramientas *stand-alone* requieren una instalación local en todos los ordenadores que vayan a usarla, pero en principio, pueden ser usadas sin conexión a la red. En la práctica, dentro de las herramientas *stand-alone* estudiadas, la única que no necesita conexión para ser ejecutada es GoSurfer, aunque existen otras que usan servidores de bases de datos que pueden ser instalados en local (GoMiner, EASE, DAVID).

El problema más importante en esta categoría es el control de versiones. Desde este punto de vista, las herramientas online son muy superiores ya que los investigadores están seguros de utilizar siempre la última versión. Las actualizaciones de

programas o bases de datos son siempre realizadas sobre el servidor por el equipo que realizó la herramienta. Sin embargo, el problema del control de versiones en las herramientas 'stand-alone' suele recaer sobre el usuario, el cual es el encargado de mantener actualizada la aplicación. Una vez conseguidas estas actualizaciones, los usuarios también serían los encargados de reinstalarlas. En principio, este tipo de herramientas deberían suministrar un programa que sea el encargado de estas actualizaciones.

5.3.8. Conjunto de datos

La mayoría de las herramientas disponibles usan una anotación similar a alguna base de datos públicas. Éstas tienen la ventaja de que sus anotaciones se actualizan simultáneamente a la actualización de la base de datos usada. La desventaja es que una única base de datos no ofrece una información completa. Para las principales anotaciones de GO, la base de datos de GO es un conjunto de datos detallado y actualizado puesto que las actualizaciones de ésta se hacen directamente sobre ella. Otras fuentes, tales como Entrez Gene, reciben sus datos de la base de datos de GO, por ello hay un pequeña ventaja desde el punto de vista de las anotaciones de GO por derivar de distintas fuentes. Sin embargo, un segundo análisis tratado aquí es la mayor potencia si más tipos de datos son integrados de una forma coherente. Una base de datos con anotación dedicada que integra varios tipos de datos provenientes de varias fuentes (e.g., pathways de KEGG) es potencialmente más útil que cualquier base de datos única. El inconveniente es que tales bases de datos son: (1) difíciles de diseñar y (2) necesitan ser actualizadas continuamente. Esto es una tarea ardua que tendría que realizar el equipo de mantenimiento de la herramienta. Teniendo en cuenta esto, es comprensible que la mayoría de las herramientas usen sólo una de las bases de datos de anotaciones disponibles. EASEonline, GOSurfer, eGOn, GOTM y BiNGO usan Entrez Gene, mientras que GeneMerge, GoMiner y TOToolBox usan las notaciones GO. T–profiler usa los datos de Saccharomyces Cerevisiae y Candida Albicans. GOLEM puede emplear tanto las anotaciones de GO como las de KEGG o InterPro. ProfCom emplea las anotaciones de GO, Interpro y FunCat; FIVA permite el uso de anotaciones de Emsembl o GeneBank; y ClueGO usa anotaciones de GO, KEGG y BioCarta. Onto-Express usa su propio conjunto de datos (Onto-Tools), siendo una de las única tentativas de integrar diferentes anotaciones. Actualmente, Onto-Tools usa datos provenientes y enlazados con: GenBank, dbEST, UniGene, Entrez Gene, RefSeq, GO y KEGG. Onto-Tools también usa datos de NetAffx y Wormbase sin estar enlazados con ellos. En la misma línea se encuentra L2L, la cual integra multitud de datos sobre cancer, inflamaciones, inmunidad/virus, hipoxia, transcripción, cromatina, ARN y otros. Y, especialmente, GeneMANIA la cual integra información de multitud de bases de

datos disponibles: datos de co–expresión de Gene Expression Omnibus (GEO); interacciones físicas y genéticas de BioGRID; interacciones de proteínas de I2D; e información de interacciones moleculares y pathways provenientes de Pathway Commons, el cual contiene datos de BioGRID, Memorial Sloan-Kettering Cancer Center, Human Protein Reference Database, HumanCyc, Systems Biology Center New York, IntAct, MINT, NCI-Nature Pathway Interaction Database and Reactome.

Por otro lado, agriGO, por su objetivo puramente agrícola, incorpora 292 tipos de datos de 45 especies diferentes.

5.3.9. Identificadores de genes soportados

Cada prueba almacenada en un microarray identifica una secuencia de nucleótidos, los cuales identifican a su vez un gen específico. Típicamente las anotaciones de las bases de datos usan genes para proporcionar una anotación funcional. Por eso, para crear una reseña funcional para una lista de genes expresados diferentemente, uno primero necesita convertir la lista de identificadores (IDs) investigados a una lista de genes. Una condición similar existe también cuando la anotación funcional proporcionada usa proteínas, donde uno necesitar además mapas de genes y/o de proteínas. Una herramienta de análisis de ontologías que soporta más de un tipo de IDs como entrada será más útil puesto que ayudará al usuario en la traducción de un tipo de IDs (e.g., conjunto de IDs Affymetrix) a uno IDs concreto (e.g., IDs de genes).

Onto-Express, GoMiner, GeneMerge y GrapWeb suministran herramientas separadas (Onto-Translate, MathMiner, el Convertidor de Nombres de Genes y g:Profiler [253], respectivamente) que posibilitan al usuario convertir a partir de otro tipo de ID (e.g., posición dentro de GenBank, IDs de RefSeq, etc.) a otro tipo/s de ID usado por la aplicación. Aunque en principio estas herramientas soportan más tipos de IDs como entrada, este diseño añade un paso separado al análisis puesto que el usuario debe realizar manualmente la fase de traducción. Por supuesto, esta comparación ha sido realizada considerando sólo la capacidad de análisis de la ontología por sí misma.

GoMiner, GeneMerge, GOToolBox y FUNC sólo permiten enviar IDs específicos de organismos usados en la bases de datos usadas como entrada. FuncAssociate, Ontology Traverse, CLENCH y FunNett sólo soportan un tipo de ID; MODB, Affymetrix *A.thaliana* MIPS y EntrezGene, respectivamente. FatiGO soporta Affymetrix y GenBank, GOToolBox, GenBank y símbolos de genes, GOdist, Affimetrix y UniGene, y BiNGO, EntrezGene y LocusTag. Onto-Express, DAVID, EASEonline y GOTM soportan la mayoría de tipos de IDs: Affymetrix, GenBank, UniGene y Entrez Gene. Además, Onto-Express y GOTM también soportan símbolos de ge-

nes, y DAVIS permite GenPept, PIR y la identificación de proteínas UniProt y RefSeq. FIVA además de soportar diferentes identificadores (Locus tags, GeneSymbol, SwisProt o InterPro) permite la conversión de identificadores InterPro a GO. Por otro lado, L2L suministra la funcionalidad de traducir diferentes identificadores de entrada, tales como Affymetrix o Entrez Gene, a Unique probe. ClueGO, por su lado, permite el uso de identificadore de Affymetrix, Accession y GeneSymbol. ProfCom soporta una gran variedad de identificadores del NCBI y affymetrix. GOEAST soporta Affymetrix, target name, probe Ids de Illumina y Agilent microarrays e identificadores de genes/proteínas de varias bases de datos, tale como NCBI, RefSeq, Esembl, UniProtKB, Sanger GeneDB, MGI, RGD, FlyBase, WormBase, TAIR, Gramene, etc. GOrilla además de permitir una gran variedad de identificadores (RefSeq, Uniprot, Unigene y Ensembl) permite la conversión de otros identificadores mediante el uso de la herramienta WebGestalt [101, 326]. Mientras que GENECODIS y g:Profiler son de las herramientas más completas en este concepto. GENECODIS permite todas o la mayoría de las identificaciones existentes por cada orgamisno; por ejemplo, para homo sapiens soporta: Ensembl Gene Id, EntrezGene, RefSeq, UniProt/SwisProt ID, UniProt/SwisProt Accession, Protein Id, Gene Symbol, UniGene y HGNC. g:Profiler, por su lado, además de soportar más de cien identificadores diferente, posee un módulo (g:Convert) capaz de reconocer diferentes identificadores en el mismo conjunto de datos de entrada.

5.4. Medidas Biológicas

En el capítulo 4 se expusieron diferentes medidas analíticas para la evaluación de grupos de genes divididas en medidas internas y externas. Dentro de esta última se enmarcaron las técnicas basadas en datos externos.

En el campo de la bioinformática, para evaluar grupos de genes según la función genética conocida, es común el uso de medidas externas basadas en conocimiento biológico previo. Dentro de este enfoque encontramos diferentes aproximaciones que usan modelos estadísticos para comparar los resultados obtenidos con la información existente. Las herramientas descritas anteriormente son un ejemplo de ello. Como se detalló, todas las propuestas, con la salvedad de DAVID y DynGO, aportan uno o más modelos estadísticos para anotar y organizar valores de expresión de experimentos microarrays en familias relacionadas por las características biológicas de los genes o de las proteínas codificadas. Además de estas herramientas, se han desarrollado diferentes estudios basados, igualmente, en usar test estadísticos o complejas ecuaciones matemáticas. Entre éstos, es de destacar el trabajo realizado por Gibbson et al. [131], donde es presentado una particular figura de méritos, z–score, basada en la información mutua entre el cluster a evaluar y las anotaciones genéticas conocidas. La medida, basada en la suma de información mutua entre los clusters y cada atributo individual de GO, fue usada para estudiar el problema de la selección del número de cluster más óptimo según la función genética conocida hasta el momento. Más recientemente encontramos estudios con el mismo enfoque. Por ejemplo, en [175] Kaplan et al. usan el concepto de falsos positivos y ciertos positivos para asignar similitud entre clusters; Lewing y Grieven realizan en [201] una modificación en la organización de GO para mejorar la eficiencia del test de Fisher; Cai et al. [65] presentan una metodología para comparar genomas usando la información de GO mediante tests estadísticos; o [181], donde es propuesto un nuevo algoritmo que elimina los problemas de los tests no paramétricos.

Por otro lado Datta et al. [86, 87] presentaron dos medidas complementarias para evaluar un conjunto de genes de una forma diferente: ambas medidas estaban basadas en ponderar tanto la estabilidad estadística como la congruencia biológica, donde para esta última se usa un conjunto de genes de entrenamiento (anotados) con función biológica conocida. Dichas medidas evalúan el resultado de un algoritmo de clustering según su habilidad de producir clusters biológicamente significativos. La primera medida, denominada ***índice de homogeneidad biológica*** (*BHI*), como su propio nombre indica, mide la homogeneidad biológica del cluster. Ésta puede ser empleada para cuantificar la habilidad de un algoritmo de clustering en agrupar genes para un conjunto de datos particular y, también, para comparar las actuaciones de distintas técnicas de clustering sobre una misma entrada. La se-

gunda medida, ***índice de estabilidad biológica*** (*BSI*), fue propuesta para medir la consistencia de un algoritmo de clustering para producir clusters biológicamente significativos cuando es aplicado a datos similares. Según Datta et al., un buen algoritmo de clustering debe tener un alto BHI y un moderado o alto BSI.

Sin embargo, es el cálculo de la similitud funcional la que está tomando un papel principal en la evaluación de genes. Estas medidas, descritas en el apartado 5.5, son capaces de asignar un valor numérico de similitud entre dos conceptos biológicos cualesquiera, lo que permite una comparación mucho más precisa que las aproximaciones anteriores.

5.5. Medidas de Similitud funcional

En esta sección se presentarán las medidas de similitud funcional que existen en la literatura, suponiendo una actualización del trabajo presentado por Pesquita et al. [240]. Todas estas aproximaciones se basan fundamentalmente en el conocimiento almacenado en Gene Ontology (ver apartado 5.2.1). Debido a ello, los trabajos son diferenciados según presenten una similitud entre GO–terms, gene–products o genes.

5.5.1. Similitud entre GO–terms

En esta sección se presentarán las distintas aproximaciones que existen hasta la actualidad para calcular la similitud entre dos términos GO. Para ello nos basaremos en la nomenclatura usada en [303] donde proponían que un GO–term A puede ser representado como el grafo $DAG_A = (A, T_A, E_A)$, donde T_A representa al conjunto de términos involucrados (i.e. el conjunto de nodos que forman DAG_A, incluyendo el término A y todos sus términos ancestros en el grafo GO), y E_A es el conjunto de aristas que conectan los términos de DAG_A.

Por otro lado, nótese que las medidas de similitud entre términos GO no son organizadas de la misma forma que en Pesquita et al. [240]: *node–based*, *edge–based* y *hybrid*. En esta ocasión se ha optado por una descripción más cronológica. De todas formas, al final de esta subsección se ha incluido un apartado catalogando las medidas descritas según la distinción de Pesquita et al.

Similitudes de Lin, Jiang y Conrath, y Resnik

Las medidas más relevantes y usadas como base en multitud de trabajos son las propuestas por Lin [205], Jiang y Conrath [170] y Resnik [255], respectivamente. Estas medidas fueron originalmente descritas para el análisis de cualquier recopilación de texto y posteriormente adaptadas para el uso en GO por Lord et al. [208].

Concretamente, estas medidas basan la comparación entre términos buscando el ancestro común más bajo (*Lowest Common Ancestor*, LCA) dentro de la jerarquía GO y son descritas en detalle seguidamente:

Supóngase que la información contenida por un GO–term A es:

$$IC(A) = -\log(p(A)) \tag{5.2}$$

Donde $p(A)$ es la probabilidad de que un término ocurra en el conjunto de anotaciones bajo consideración:

$$p(A) = freq(A)/freq(root) \tag{5.3}$$

"Root" representa al término raíz de una de las tres ontologías y $freq(root)$ es el número de veces que un gen es anotado con algún término de la ontología. Mientras que $freq(A)$ es dado por:

$$freq(A) = |annot(A)| + \sum_{c \in children(A)} |annot(c)| \tag{5.4}$$

Siendo $children(A)$ el conjunto de todos los términos hijos del termino A. Es decir, el conjunto de todos los términos para los que A es un término padre, ya sea directa o indirectamente.

Resnik calcula la similitud entre dos términos usando sólo la información contenida (IC) del LCA compartido entre dos términos A y B:

$$sim_{Res}(A, B) = IC(LCA(A, B)) \tag{5.5}$$

Por otro lado, la medida de similitud de **Lin** toma en cuenta los valores de IC para cada uno de los términos A y B, además del LCA compartido por los dos términos:

$$sim_{Lin}(A, B) = \frac{2 \times IC(LCA(A, B)))}{IC(A) + IC(B)} \tag{5.6}$$

Mientras que **Jiang y Conrath** propusieron un IC basado en distancia semántica, la cual puede ser transformada en la siguiente medida de similitud:

$$sim_{Jiang}(A, B) = \frac{1}{IC(A) + IC(B) - 2 \times IC(LCA(A, B)) + 1} \tag{5.7}$$

Para cada una de estas medidas, cuanto mayor sea el valor obtenido mayor similitud semántica presentan los dos términos. El menor valor posible es el 0, mientras que el mayor valor es 1.

Guo, Sevilla, Wang et al. evaluaron estas medidas en tres trabajos diferentes

[136, 269, 301]. Ellos mostraron que la medida de Resnik es mejor que los otros métodos en términos de correlación con la similitud de secuencia de genes y el nivel de expresión genético.

Basadas en los de conceptos ancestros comunes e información contenida

Debido a la estrutura de gráfo acíclico dirigido de GO, es posible que un GO–term presente diferentes padres con lo que se podría dar el caso que dos términos puedan compartir ancestros en diferentes caminos hacia el nodo raíz de la ontología.

Lord et al. [208], además de adaptar las medidas de Jiang, Lin y Resnik, fueron los pioneros en proponer una nueva medida de similitud semántica teniendo en cuenta esta posibilidad de GO. Esta medida, que denominaremos sim_{Lord}, se basa en encontrar el ancestro común más informativo (*probability of the minimum subsumer*, p_{ms}):

$$p_{ms}(A, B) = min_{c \in S(A,B)} p(c) \tag{5.8}$$

Donde $S(A, B)$ es el conjunto de términos ancestros compartidos por A y B. Quedando la similitud entre dos términos según Lord et al. como:

$$sim_{Lord}(A, B) = -ln(p_{ms}(A, B)) \tag{5.9}$$

Una variación de esta medida fue propuesta por **Liu et al.** en [207], en donde en vez de seleccionar un ancestro del conjunto de todos los ancentros comunes, se basaban en un subconjunto de éstos (LCA) para su elección:

$$sim_{Liu}(A, B) = -ln(min_{t \in LCA(A,B)}\{p(t)\}) \tag{5.10}$$

Por otro lado, **Couto et al.** [81] también abordaron el problema de similitud entre dos GO–terms teniendo en cuenta la posibilidad de que un término pueda tener diferentes padres. La medida, que denominaremos sim_{Couto}, considera todos los ancestros comunes y no sólo uno como la medida propuesta por Lord.

Concretamente, la aproximación de Couto se basa en el concepto de información contenida de un término GO definida en la ecuación 5.2, aunque en este caso $p(A)$ fue definido como el número de ocurrencias de A dividido por el número total de ocurrencias de los términos de T_A. De esta forma, y debido a que GO presenta una estructura jerárquica, la nueva definición de $p(t)$ ($p'(t) = \frac{annot(t)}{\sum_{i \in T_t} annot(i)}$) tiende a incrementarse cuanto más ascendamos en la jerarquía, es decir, cuanto más cerca esté t de la raíz de la ontología en cuestión. Por tanto, cuando un GO–term t_1 subsume a otro t_2, y t_2 ocurre entonces se considera que t_1 también ocurre. Así, $p'(t_1)$ siempre será mayor o igual que $p'(t_2)$, i.e. $IC'(t_1) \leq IC'(t_2)$.

Basándose en la nueva definición de información contenida (IC'), Couto et al. presentaron la distancia entre t_1 y t_n, considerando que t_1 subsume a t_n y que la secuencia de términos GO $t_0, \ldots, t_n$ representa el camino de t_0 a t_n con una longitud n, como:

$$\Delta(t_0, t_n) = \sum_{i=0}^{n-1} D(t_i) \times E(t_i) \times (IC'(t_{i+1}) - IC'(t_i)) \tag{5.11}$$

Donde $D(t)$ y $E(t)$ representan la profundidad y la densidad del factor de distancia conceptual para el GO–term t, respectivamente:

$$D(t) = \left(\frac{d(t)+1}{d(t)}\right)^{\alpha}; E(t) = (1-\beta) \times \frac{\bar{E}}{e(t)} + \beta \tag{5.12}$$

Siendo $d(t)$ la profundidad de t en la ontología, $e(t)$ la densidad local de t (i.e. el número de aristas que comienzan en t) y $\bar{E}$ la densidad media en la ontología (i.e. el número de aristas dividido por el número términos GO involucrados). Mientras que los parámetros α y β controlan el grado de contribución de la profundidad y la densidad en la ecuación 5.11, respectivamente. Un valor de $\alpha = 0$ ó $\beta = 1$ indica que el grado de contribución es nulo para el concepto que representan.

Sin embargo, en el caso en que los dos términos t_1 y t_2 no presenten una relación de subsumisión, la distancia semántica es calculada mediante la suma de sus distancias semánticas al LCA. Nótese que el LCA subsume a ambos términos, y por tanto, es aplicable la ecuación 5.11:

$$\Delta(t_a, t_b) = \Delta(LCA, t_a) + \Delta(LCA, t_b) \tag{5.13}$$

A partir de estas definiciones de distancia semántica, Couto et al. propusieron la medida de similitud entre dos términos GO A y B como:

$$sim_{Couto}(A, B) = 1 - min\{1, \frac{\Delta(A, B)}{IC'(t_0)}\} \tag{5.14}$$

Donde t_0 es un término que sólo ocurre una vez, es decir, $IC'(t_0)$ representa la máxima información contenida.

Esta medida, que toma valores entre 0 y 1, y al contrario que las expuestas en el apartado anterior, cuanto mayor sea peor similitud presentan los términos estudiados.

Por otro lado, **Schlicker et al.** [265] propusieron una medida (sim_{Rel}) que combina la medida de Lin y la de Resnik. Esta aproximación trata de tener en cuenta la cercanía de los términos a su LCA y cómo de detallado está dicho ancestro común. El nivel de detalle es reflejado mediante la probabilidad de que ocurra

el LCA.

$$sim_{Rel}(A,B) = \frac{2 \times IC(LCA(A,B))}{IC(A) + IC(B)}(1 - p(LCA(A,B))) \tag{5.15}$$

Posteriormente, en 2007, **Wang et al.** [303], también abordando el problema de similitud entre términos GO teniendo en cuenta sus ancestros comunes. En este caso propusieron una medida de combinación de ancestros comunes que, según demostraron en el mismo trabajo, es más consistente que la medida de Resnik en estudios génicos humanos.

La medida, que denominaremos sim_{Wang}, se basa en calcular un valor S (S–value) de los diferentes términos involucrados en la ontología:

$$sim_{Wang}(A,B) = \frac{\sum_{t \in T_A \cap T_B}(S_A(t) + S_B(t))}{SV(A) + SV(B)} \tag{5.16}$$

Donde $S_A(t)$ y $S_B(t)$ son los S–values del término t relacionado con el término A y B respectivamente. El valor S de un término t_i ($S_A(t_i)$) se determina a través de los valores S de sus términos hijos y el tipo de relación que guarde con ellos (ver apartado 5.2.1). Concretamente el factor de contribución semántica de estas relaciones viene determinado por el valor w_e, cuyos valores propuestos por Wang et al. son 0,8 y 0,6 para las relaciones `is_a` y `part_of`, respectivamente.

$$S_A(t_i) = \begin{cases} 1, & t_i = A; \\ max\{w_e * S_A(t') | t' \in childrenof(t_i)\}, & t_i \neq A. \end{cases} \tag{5.17}$$

Mientras que $SV(A)$ y $SV(B)$, representan al valor semántico de los términos relacionados con A y B:

$$SV(A) = \sum_{t \in T_A} S_A(t) \tag{5.18}$$

Por otro lado, **Couto et al.** [82] propusieron una medida que tenía en cuenta los ancestros no comunes. Particularmente, la aproximación (***GraSM***), es calculada como la media aritmética de la información contenida de los ancestros comunes disjuntos (DCA):

$$sim_{GraSM}(A,B) = \frac{\sum_{t \in DCA(A,B)} IC(t)}{|DCA(A,B)|} \tag{5.19}$$

Dos ancestros son disjuntos si poseen rutas independientes desde dichos ancestros hasta el término o concepto. Donde dos rutas independientes son aquellas que usan al menos un concepto de la ontología no usado por la otra. De esta forma, dos ancestros disjuntos de un término representan dos interpretaciones distintas de dicho concepto. Más concretamente, GraSM considera que a_1 y a_2 representan ancestros disjuntos (DA) de A si existe una ruta desde a_1 hasta A que no pase por a_2 y una ruta desde a_2 hasta A que no pase por a_1:

$$DA(A) = \{(a_1, a_2)| \\ (\exists p_1 : (p_1 : Paths(a_1, A)) \wedge (a_2 \notin p_1)) \wedge \\ (\exists p_2 : (p_2 : Paths(a_2, A)) \wedge (a_1 \notin p_2))\} \tag{5.20}$$

donde $Paths(a, A)$ representa al conjunto de rutas diferentes desde a hasta A en el DAG.

Dados dos términos A y B, sus ancestros disjuntos comunes son los ancestros comunes más informativos de A y B, i.e. c_1 es un ancestro común disjunto de A y B si para cada ancestro c_2 más informativo que c_1, c_1 y c_2 son ancestros disjuntos de A y B:

$$DCA(A, B) = \{c_1| \\ c_1 \in CA(A, B) \wedge \forall c_2 : \\ (c_2 \in CA(A, B) \wedge (IC(c_1) \leq IC(c_2)) \\ \Rightarrow ((c_1, c_2) \in DA(A) \cup DA(B))\} \tag{5.21}$$

Más recientemente, en 2011, el propio Couto junto con Silva [80] propusieron una mejora de la media GraSM. Concretamente presentan una nueva definición del concepto DCA (**DiShIn**) basada en identificar los ancestros disjuntos a través del número de rutas diferentes de los términos a sus ancestros comunes:

$$DCA_{DiShIn}(A, B) = \{c : \\ c \in CA(A, B) \wedge \\ \forall_{c_x \in CA(A,B)} PD(A, B, c) = PD(A, B, c_x) \\ \Rightarrow IC(A) > IC(c_x)\} \tag{5.22}$$

donde CA representa a los ancestros comunes y PD a la diferencia entre el número de rutas de dos términos a sus ancestros:

$$PD(A, B, c) = |Paths(A, c) - Paths(B, c)| \tag{5.23}$$

Basadas en la representación del DAG como GO–tree

En 2004, **Lee et al.** [196], abordaron una medida de similitud semántica basada en una novedosa variación de la estructura de GO. La variación, denominada GO–tree, consiste en mapear la estructura de grafo de GO a un árbol. Para ello, un GO–term en la estructura GO–tree sólo podrá tener un padre aunque éste puede estar representado en diferentes partes del árbol. De esta forma, un GO–term aparecerá tantas veces en el GO–tree como caminos ascendentes diferentes tenga éste en el

DAG para acceder al nodo raíz de la ontología en cuestión. En la figura 5.13 se puede observar un ejemplo de este mapeo, donde se presentan un DAG con nueve términos GO de los cuales dos ($GoT3$, $GoT9$) poseen más de un camino para llegar a la raíz y, por tanto, aparecen repetidos en el GO–tree.

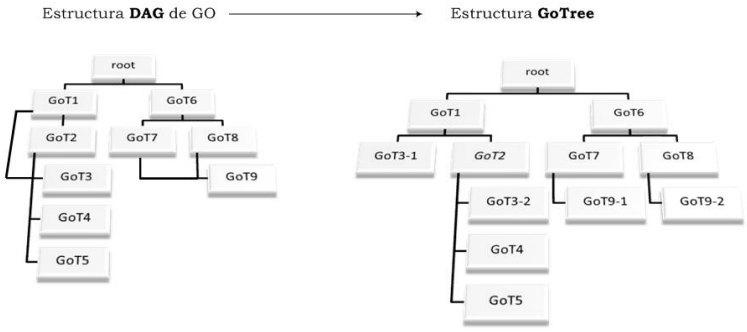


Figura 5.13: Ejemplo de construcción de un GO–tree a partir de la estructura DAG de GO.

A partir de esta estructura, la medida de similitud, que denominaremos sim_{Lee}, se calcula como el peso del LCA de los términos que le ocupa:

$$sim_{Lee}(A,B) = \begin{cases} 0, & A = B; \\ W(LCA(A,B)), & \text{en otro caso} \end{cases} \tag{5.24}$$

Donde el peso viene determinado por la siguiente ecuación:

$$W(A) = H \times 10 - 10(d(A) - 1) \tag{5.25}$$

Siendo H un parámetro que indica la profundidad máxima que se tendrá en cuenta en el GO–tree. Este parámetro fue usado con un valor de $H = 15$ en el estudio presentado en [196].

Basadas en la combinación de conocimiento

En 2008 **Pozo et al.** [243] propusieron una nueva vertiente para calcular la similitud entre GO–terms. El método consiste en analizar la información contenida en la ontología *Molecular Function* y las entradas de Interpro.

Concretamente, por cada GO–term en la ontología *Molecular Function* se crea un vector que representa la presencia o ausencia en diferentes entradas de InterPro. Posteriormente y tras una fase de filtrado, es construida una matriz de ocurrencias simultáneas (***co–occurrence matrix***) de términos MF–GO en las entradas Interpro. Las ocurrencias son acumuladas para todas los perfiles, quedando descrito cada

vector de ocurrencias simultáneas (***co–occurrence vector***) como un término MF–GO en relación con el resto de términos MF–GO. Una vez determinados cada unos de estos vectores, la medida de similitud entre dos GO–terms, que denominaremos sim_{Pozo} puede ser calculada como:

$$sim_{Pozo}(A, B) = \cos(\overrightarrow{V}(A), \overrightarrow{V}(B)) = \frac{\overrightarrow{V}(A) * \overrightarrow{V}(B)}{|\overrightarrow{V}(A)||\overrightarrow{V}(B)|} \tag{5.26}$$

Donde $\overrightarrow{V}(A)$ y $\overrightarrow{V}(B)$ representan los *co–occurrence vectors* de los términos A y B, respectivamente.

Resumen de similitudes entre términos GO: *edge–based*, *node–based* y *hybrid*

En este apartado, y más concretamente, en la tabla 5.4, se realiza una organización de las similitudes descritas anteriormente atendiendo a la organización llevada a cabo por Pesquita et al. [240]. En dicho trabajo, las medidas fueron organizadas en: *edge–based*, en donde la principal fuente son las aristas y tipo de éstas; *node–based*, cuya fuente son los nodos y sus propiedades (mayoritariamente basadas en información contenida); y *hybrid*, que son una mezcla de ambas. Para las similitudes basadas en aristas, se asume que la especificidad de un término puede ser directamente inferido de su profundidad en el grafo; mientras que para las basadas en nodos, es la información contenida la que generalmente informa de la especificidad de un término.

Tabla 5.4: Resumen de similitudes entre términos.

Aproximación	Similitud		Aproximación	Similitud	
Node–based	Lin	[205]	*Edge–Based*	Couto	[81]
	Jiang	[170]		Wang	[303]
	Resnik	[255]	*Hybrid*	Lee	[196]
	Lord	[208]		Pozo	[243]
	Liu	[207]			
	Shlicker	[265]			
	GraSM	[82]			
	DiShIn	[80]			

Sobre esta distinción cabe destacar la afirmación realizada por el propio Pesquita en un artículo anterior [239], donde afirmaban que las medidas basadas en IC son más adecuadas que las basadas en aristas.

5.5.2. Similitud entre Gene–Products o proteínas

Las medidas expuestas en el apartado anterior tienen la intención de medir la similitud entre dos GO–terms, y deben ser extendidas para comparar gene–products

o proteínas. Un gene–product posee uno o más términos GO asociados, con lo que una aplicación directa de las medidas expuestas anteriormente no sería posible. Al conjunto de GO–terms anotados o asociados a un gene–product o proteína "gp" lo denominaremos ***anotaciones*** ($Anot_{gp}$).

Basadas en combinación de similitudes de GO–terms (*Pairwise–based measure*)

Lord et al. [209] fueron los pioneros en proponer una solución al problema anterior. Éstos presentaron la similitud entre dos gene–products A, B ($simP(A, B)$) como la combinación de las similitudes de los distintos GO–terms asociados a éstos. Cada término asociado a A es comparado con todos los términos asociados a B, obteniendo por cada pareja un valor de similitud entre GO–terms. Estos valores son usados para producir una medida final de similitud entre pares de gene–products. Concretamente, Lord et al. propusieron la ***media aritmética*** como combinación de las similitudes de todos las posibles parejas de GO–terms.

$$simP_{avg}(A, B) = \frac{\sum_{t_A \in Anot_A \wedge t_B \in Anot_B} sim(t_A, t_B)}{|Anot_A| \times |Anot_B|} \tag{5.27}$$

Posteriormente, en la literatura existen diferentes aproximaciones basadas en el mismo concepto de combinar las diferentes similitudes entre parejas de GO–terms. Estas combinaciones son:

- ***Máximo***($simP_{max}$) [196]: Selecciona el máximo nivel de similitud encontrado como similitud de los gene–products.

$$simP_{max}(A, B) = max\{sim(t_A, t_B) | t_A \in Anot_A \wedge t_B \in Anot_B\} \tag{5.28}$$

- ***Suma***($simP_{sum}$) [198]: Es calculado como la suma de las similitudes de todas las parejas de GO–terms.

$$simP_{sum}(A, B) = \sum_{t_A \in Anot_A \wedge t_B \in Anot_B} sim(t_A, t_B) \tag{5.29}$$

- ***Match*** [198]: Engloba a las combinaciones que sólo tiene en cuenta aquellas parejas de GO–terms que son iguales. En esta categoría se podrían encontrar las combinaciones de media aritmética ($simP_{avg\_M}$), máximo ($simP_{max\_M}$) y suma ($simP_{sum\_M}$).

$$simP_{avg\_M}(A, B) = \frac{\sum_{t \in [Anot_A \cap Anot_B]} sim(t, t)}{|Anot_A \cap Anot_B|} \tag{5.30}$$

$$simP_{max\_M}(A, B) = max\{sim(t, t)|t \in [Anot_A \cap Anot_B]\} \quad (5.31)$$

$$simP_{sum\_M}(A, B) = \sum_{t \in [Anot_A \cap Anot_B]} sim(t, t) \quad (5.32)$$

- ***Best Pairs*** [198]: Engloba a aquellas combinaciones que sólo tienen en cuenta aquellas parejas de GO–terms que maximizan la similitud sin repetición de GO–term. De esta forma, se selecciona todos los términos de aquel soporte que contenga menos GO–terms, y éstos son asociados con el GO–term del soporte del otro gene–product que maximice la similitud entre ambos. Además, cada GO–term, independientemente del soporte al que pertenezca, sólo puede usado una vez en el calcula de similitud de términos GO. En esta categoría se encuentran tanto la media aritmética ($simP_{avg\_BP}$) como la suma ($simP_{sum\_BP}$).

- ***Best Match Average(BMA)***($simP_{avg\_BM}$) [239]: Se determina como la media aritmética de las mejores similitudes. Con tal fin, se escoge el soporte con menor número de GO–terms y por cada uno de los términos en éste es seleccionado un término del soporte del otro gene–product que maximice la similitud entre ambos. Al contrario que en *Best Pairs*, en este caso es posible usar un GO–term en diferentes mediciones de similitud.

Estas combinaciones han sido usadas en diferentes trabajos mediante el uso de una o varias similitudes de GO–terms. En la tabla 5.5 se presenta un resumen de esta información, donde se indica el trabajo donde se recoge el estudio junto a los autores de éste; la similitud de GO–terms usada y la combinación empleada en el estudio. Además, los nombre de las similitudes de GO–terms o gene–products que sean propuestos y usadas en el mismo artículo aparecerán remarcados en negrita. Por ejemplo, Pesquita et al. describieron en [239] sim_{GraSM} como medida de similitud entre dos GO–terms, la cual fue combinada usando la media aritmética para calcular la similitud de dos gene–products. Además, en el estudio que propusieron combinaron mediante el máximo, media y BMA las medidas de Resnik, Lin y Jiang para calcular las similitudes de dos gene–products que poseen un soporte mayor que uno.

Por otro lado, en la tabla 5.5 se usan dos conceptos que no han sido definidos previamente; la similitud entre dos GO–terms *ExactMatch* y la combinación empleada por Couto et al. [81] (MAX + IC de términos). En primer lugar, Lei et al. [198] usaron, entre otros, *ExactMatch* como medida de similitud entre dos GO–terms, la cual toma valores igual a 1 en caso de que los dos GO–terms comparados sean idénticos, y 0 en otro caso. Por otro lado, la combinación propuesta por Couto et al. se basa en la idea de que dos gene–products no sólo son similares cuando son anotados con términos funcionales similares, sino que además lo son cuando estos

Tabla 5.5: Resumen de similitudes de gene–products basadas en pares.

Autor		*sim* GO term	*sim* Gene–Product
Lord et al.	[209]	Lord	AVG
Couto et al.	[81]	***Couto***	***MAX + IC de términos***
Lee et al.	[196]	***Lee***	MAX, AVG
Liu et al.	[207]	***Liu***	AVG
Azuaje et al.	[22]	Resnik, Lin	AVG
Lei et al.	[198]	Lord, ***ExactMatch***	MAX, AVG, SUM, ***Match***(max,avg,sum), ***BestPairs***(avg,sum)
Bramier and Wiuf	[58]	Resnik	MAX
Guo et al.	[136]	Resnik, Lin, Jiang	MAX
Faria et al.	[116]	Resnik, Lin, Jiang, Rel	***Best Matching***
Wang et al.	[303]	Wang	MAX
Pesquita et al.	[239]	Resnik, Lin, Jiang	MAX, AVG, ***BMA***
		GraSM	AVG
Guo et al.	[135]	Resnik, Lin, Jiang, Rel	MAX, AVG
Mistry et al.	[224]	Resnik, Lin, Jiang	MAX, AVG
Couto y Silva	[80]	DiShIn	BMA

términos tienen una información contenida significante. De esta forma, la ecuación de la combinación máximo queda definida como:

$$simP_{max\_IC}(A,B) = max\{sim(t_A,t_B) \times IC(t_A) \times IC(t_B) | t_A \in Anot_A \wedge t_B \in Anot_B\} \tag{5.33}$$

Finalmente, Xu et al. [317] realizaron una comparación de la diferentes aproximación basadas en pares, y concluyeron que BMA consistentemente da el mejor rendimiento para todos los tests que llevaron a cabo.

Basadas en modelos de espacio vectorial

Desde un punto de vista biológico, existen limitaciones a las aproximaciones de similitud de proteínas que combinan medidas de similitud de términos. Usar la media no es apropiado para gene–products con muchos términos compartidos o similares. Por ejemplo, dos gene–products funcionalmente similares que comparten los términos antioxidant activity y *binding* presentan una similitud del 50 %, y no del esperado 100 %, ya que las similitudes son calculadas entre todos los posibles pares de términos de dos gene–products. Este mismo problema es presentado por la combinación de suma, ya que también tiene en cuenta todas las posibles parejas. Por contrario, el usar el máximo no presenta esta problemática aunque para algunos autores como Pesquita et al. o Popescu et al. [242] presenta otra al ser indiferente al número de términos no relacionados entre los gene–products. Por ejemplo, un gene-product con los términos antioxidant activity y *binding* y un segundo gene–product con sólo uno de esos términos tendría una similitud del 100 %, cuando funcionalmente no son totalmente iguales.

Para solucionar esta problemática, **Popescu et al.** [242] propusieron una medi-

da basada en el espacio vectorial que denominaron ***Cosine similarity***. Esta medida requiere primero el cálculo de una matriz de anotaciones $m \times n$, donde m es el número de gene–products en el estudio y n el número total de GO–terms. Cada fila de la matriz representa las anotaciones de un gene–product, de forma que cada vector es ponderado binariamente, donde 1 representa la presencia del GO–term en las anotaciones del gene–product y 0 su ausencia. A partir de este concepto, Popescu et al. plantearon la siguiente medida de similitud entre dos gene–products:

$$simP_{cos}(A, B) = \frac{v_A \cdot v_B}{|v_A||v_B|} \tag{5.34}$$

Una variación a la aproximación *Cosine* fue propuesta por **Chabalier et al.** [73]. La nueva medida, que fue denominada como ***Weighted Cosine***, genera un peso, w_t, para cada GO–term basado en la frecuencia de sus apariciones. Estos pesos, que son calculados mediante la ecuación 5.35, reemplazan los valores que no son cero en el vector binario usado por la ecuación 5.34.

$$w_t = log(m/n_t) \tag{5.35}$$

Donde m es el número total de gene–products en el estudio y n_t el número de gene–products anotados con el término t.

Finalmente, **Huang et al.** [159] también propusieron una medida de similitud basada en vectores, la cual es integrada en ***DAVID*** *Gene Functional Classification Tools*. Esta medida, para un par de gene–products dados, extrae primeramente los vectores binarios como se indicó anteriormente. Tras ello, es usada la estadística Kappa [64] para medir la coocurrencia de las anotaciones entre pares de gene–products.

Basadas en Grafos y el solape de términos

Siguiendo la idea de solventar la problemática que presentan la media y el máximo como combinación de medida de similitud de GO–terms, **Gentleman et al.** [127] presentaron una nueva medida de similitud de gene–products basada en la similitud de dos grafos integrado en Bioconductor [128]. De esta forma, la similitud de dos gene–products A y B viene determinada por los grafos de cada una de las anotaciones de A y B (DAG'_A, DAG'_B), respectivamente.

$$DAG'_A = \bigcup_{t \in Anot_A} DAG_t \tag{5.36}$$

Como medidas de similitud entre dos grafos, Gentleman et al. propusieron dos aproximaciones diferentes: una primera basada en la ruta compartida más

larga (*longest shared path*, sim_{LP}), y otra en la unión de intersecciones (*union–intersection*, sim_{UI}),

La primera medida adopta la profundidad de la ruta más larga compartida por los grafos DAG'_A y DAG'_B como medida de similitud. Nótese, que esta definición podría cambiarse a términos de LCA de gene–products, entendiendo por este nuevo concepto el LCA de todas las posibles parejas de GO–terms:

$$simP_{LP}(A, B) = d(LCA'(A, B)) \tag{5.37}$$

donde d(*) tiene el mismo significado que el usado en la ecuación 5.12

La segunda propuesta, sim_{UI}, usa el número de nodos que comparten los grafos DAG'_A y DAG'_B divido por el número de nodos en los dos grafos. El resultado de similitud se encuentra entre 0 y 1, de modo que cuanta mayor similitud posean más cercano al 1 será.

$$simP_{UI}(A, B) = \frac{|T'_A \cap T'_B|}{|T'_A| + |T'_B|} \tag{5.38}$$

Estas medidas han sido usadas por diferentes trabajos para ponderar la similitud entre proteínas. Así por ejemplo, Guo et al. [136] realizaron un estudio de $simP_{LP}$ y $simP_{UI}$ junto con la combinación máxima de las similitudes de Resnik, Lin y Jiang; o Wolting et al. [308] usaron $simP_{UI}$ como base para evaluar la similitud entre clusters de proteínas. Además, estas medidas, aunque fueron pensadas para evaluar gene–products, han sido empleadas por Lei et al. [198] como medida de similitud entre dos GO–terms, entendiendo que un GO–term es un gene–product con una única anotación.

Según **Pesquita et al.** [239], las medida de similitud $simP_{LP}$ y $simP_{UI}$, ponderan todas los términos de igual forma y, por tanto, no tiene en cuenta términos específicos. Para subsanar esta limitación, propusieron una nueva medida basada en $simP_{UI}$, que denominaremos $simP_{GIC}$, con la variación de que cada término es ponderado según su información contenida:

$$simP_{GIC}(A, B) = \frac{\sum_{t \in \{Anot_A \cap Anot_B\}} IC(t)}{\sum_{t \in \{Anot_A \cup Anot_B\}} IC(t)} \tag{5.39}$$

Por otro lado, Mistry et al. [224] propusieron un nuevo enfoque para generar medidas de similitud. Este enfoque, es una variación a la medida presentada por **Lee et al.** en [195] la cual propone la similitud entre dos gene–products como el número de términos solapados entre los soportes de ambos gene–products:

$$simP_{TO}(A, B) = |Anot_A \cap Anot_B| \tag{5.40}$$

Al igual que otras medidas, cuanto mayor sea el valor obtenido mayor es la

similitud entre la pareja de gene–products. El menor valor, correspondiente al caso en que no hay ningún solape entre los términos, es cero, mientras que no hay un tope superior. Esto hizo que **Mistry et al.** propusieran una variante que normalizara el solape de términos (*NTO*):

$$simP_{NTO}(A,B) = \frac{sim_{TO}(A,B)}{min(|Anot_A|,|Anot_B|)} \tag{5.41}$$

Medidas de similitud tradicionales basadas en la cardinalidad, tales como **Jaccard** y **Dice** [242] son calculadas de forma similar a $simP_{NTO}$, pero usan la unión o suma, respectivamente, del tamaño de los conjuntos de términos anotados como factor de normalización. Mientras que la distancia **Czekanowski-Dice**, propuesta en el módulo de análisis funcional de GoToolBox [217], calcula la similitud normalizando el número de diferencias simétricas entre los dos conjuntos soporte con la suma de las intersecciones y uniones de los conjuntos. De esta forma, la escala para la distancias es inversa a $simP_{NTO}$, presentando una valor igual a 1 en caso de que no posean ningún término en común y cercanos al 0 si poseen una funcionalidad muy relacionada.

De igual forma, **Langaas et al.** [191] o **Wu et al.** [312] usaron una variante de estas medidas que también normalizan el solape de términos. En este caso, la similitud vendría dada como el número de términos de la intersección partido por el tamaño del conjunto de la unión.

Otras medidas de similitud

Además de las vertientes presentadas anteriormente existen diferentes trabajos en los que se extrapolan las medidas de similitud presentadas en el la sección 5.5.1 para generar nuevas medidas de similitud funcional entre proteínas.

En primer lugar, **Haiying Wang et al.** [301] propusieron una medida basada en sim_{Lin} que puede ser calculada según la siguiente ecuación:

$$simP_{HWang}(A,B) = \frac{1}{|Anot_A| \times |Anot_B|} \times \sum_{t_A \in Anot_A, t_B \in Anot_B} sim_{Lin}(t_A,t_B) \tag{5.42}$$

Por otro lado, **Shlicker et al.** [265] presentaron una nueva medida que consistía en combinar las evaluaciones de similitudes funcionales obtenidas en las ontologías *Biological Process* y *Molecular Function* para dos gene–products A y B. Concretamente, se basa en el cálculo previo de una matriz por cada ontología de tamaño $|Anot_A| \times |Anot_B|$. Esta matriz contiene todos las similitudes de pares de GO–terms posibles, de forma que las filas y columnas representan dos comparaciones direccionales diferentes; los vectores filas corresponden con la comparación de A con B, mientras que los vectores columnas de B con A. Observando esta ma-

triz, se puede extraer el valor máximo, y a partir de la fila y columna donde se encuentre (i,j) calcular lo que denominaron Schlicker et al. [265] el GO_{score}. Esta puntuación viene determinada por el máximo entre la media de los valores del vector de la fila i-ésima y la media entre los valores de la columna j-ésima para una ontología dada. Quedando la ecuación de similitud entre dos gene–products como:

$$simP_{funSim}(A,B) = \frac{1}{2} \cdot [(\frac{BP_{score}(A,B)}{max(BP_{score}(A,B))})^2 + (\frac{MF_{score}(A,B)}{max(MF_{score}(A,B)})^2] \quad (5.43)$$

Donde BPscore y MFscore denotan la puntuación para las ontologías *Biological Process* y *Molecular Function*, respectivamente. Además, esta puntuación, GO_{score}, fue calculada por Schlicker et al. [265] atendiendo tanto a la medida de similitud propuesta en el mismo trabajo (sim_{Rel}), como la presentada por Lord et al. [208](sim_{Lord}). Igualmente, el concepto de GO_{score} fue integrado por Guo [135] en el paquete ***SemSim*** de Bioconductor. Además, la medida $funSim$ está actualmente disponible, junto con la medida de Popescu et al. [242] ($simP_{Gic}$), en la herramienta ***funSimMat*** desarrollada por Schlicker y Albrecht [264].

El concepto de GO_{score} también fue usado por **Bastos et al.** [29] pero en su caso con un significado diferente. Concretamente, propusieron tres medidas diferentes: $GO_{ocurrence}$, para medir la coherencia funcional de una lista de gene–products (media de la frecuencia de anotaciones por cada GO–term); GO_{score} para indicar cómo de bueno ha sido la caracterización funcional de un grupo de genes–products; y GO_{center} que mide cuántas anotaciones funcionales del clusters son capturadas por la proteína central del cluster. En dicho trabajo, un conjunto de gene–products es tratado como un conjunto de todos los GO–terms relacionados con cada gene–product. De esta forma, $GO_{ocurrence}$ es definido como la media de las frecuencia de ocurrencia de cada GO–term; GO_{score} como el máximo de la información contenida por la frecuencia de anotación del GO–term; y GO_{center} como la fracción de términos el cluster que están anotados con la proteína central del cluster.

Otra aproximación diferente fue presentada en 2007 por **James Z. Wang et al.** [303], en donde extrapolaron la medida de similitud que propusieron en el mismo trabajo (sim_{Wang}) para calcular similitudes funcionales entre grupos dos grupos de GO–terms:

$$simP_{JWang}(A,B) = \frac{\sum_{t_A \in Anot_A} sFun(t_A,B) + \sum_{t_B \in Anot_B} sFun(t_B,A)}{|Anot_A| + |Anot_B|} \quad (5.44)$$

Donde $sFun(t,X)$ denota la similitud funcional entre el GO–term t y el gene–product X:

$$sFun(t,X) = max_{t_X \in Anot_X}(sim_{Wang}(t,t_X)) \quad (5.45)$$

En el mismo año, **Tao et al.** [286] propusieron una extrapolación de sim_{Lin} ba-

sada en grupos de conceptos. Para ello, el grupo de conceptos de un gene–product son aquellos términos GO que están asociados con dicho gene–product, o dicho de otra manera, las anotaciones de éste. A partir de ahí, definieron una medida de similitud que tenía en cuenta sólo aquellas similitudes de pares de GO–terms que superaban un umbral límite t:

$$simP_{Tao}(A,B) = \frac{2 \times \sum_{t_A \in Anot_A \wedge t_B \in Anot_B \wedge sim(t_A,t_B)>t} sim(t_A,t_B)}{|Anot_A| + |Anot_B|} \tag{5.46}$$

Más actualmente, en 2008, ***Pozo et al.*** [243] extrapolaron la similitud funcional de dos GO–terms que propusieron en el mismo trabajo (sim_{Pozo}). Concretamente usaron la distancia de Hausdorff, la cual es definida como el valor máximo entre cualquier punto en un conjunto y el punto más cercano del otro conjunto. Formalmente, la distancia de los gene–products A a B viene determinada como:

$$D^{A \to B}_{hausdorff} = max_{a \in Anot_A}\{min_{b \in Anot_B}(D(a,b))\} \tag{5.47}$$

Como la distancia de Hausdorff no es simétrica, Pozo et al. propusieron la siguiente medida similitud:

$$simP_{Hausdorff}(A,B) = max(D^{A \to B}_{hausdorff}, D^{B \to A}_{hausdorff}) \tag{5.48}$$

Usualmente, la distancia de Hausdorff es evaluada según el espacio Euclídeo, aunque Pozo et al. usaron dos medidas diferentes: (a) la medida de similitud propuesta en el mismo artículo (sim_{Pozo}); (b) la medida de similitud de Lord (sim_{Lord}).

Por otro lado, ***Zheng y Lu*** [329] propusieron un enfoque diferente basado en la literatura biomédica. En tal trabajo, desarrollaron una medida para determinar la coherencia funcional general de un grupo de proteínas a través de la literatura biomédica asociada con las proteínas.

5.5.3. Similitud entre genes

Es conocido que un gen codifica a una o varias proteínas, y que por tanto su comportamiento viene determinado por las funcionalidades de las proteínas con las que está asociado. De esta forma, las medidas de similitud funcional de varios genes deben atender a la similitud de las diferentes proteínas que codifica.

Esta distinción no es tenida en cuenta como tal en multitud de trabajos que se encuentran en la literatura. Trabajos como los de H. Lee et al. [195] o S. Lee et al. [196], u otros más recientes como los de Wang et al. [303], Tao et al. [286], Chabalier et al. [73] o Mistry et al. [224], usan indistintamente los concepto de gen y gene–product. Particularmente, estos trabajos no tratan las diferentes funcionalidades de un gen de forma independiente, sino que un gen es asociado a un conjunto

de GO–terms. Este conjunto vendría dado por la unión de los diferentes conjuntos soporte de las proteínas que el gen codifica. Debido a esto, el problema de similitud funcional de diferentes genes es abordado como una comparación entre grupos de GO–terms, siendo ésta la razón por la que dichas aproximaciones hayan sido detalladas en la sección anterior. Además, siguiendo la misma idea, el resto de propuestas presentadas en tal sección también podrían emplearse como medidas de similitud entre genes.

No es hasta 2009 cuando se presenta el primer trabajo para medir eficientemente la similitud de un conjunto de genes. En ese trabajo Ruths et al. [262] propusieron una medida, denominada GS^2, para cuantificar la similitud de un conjuntos de genes mediante el promedio de su contribución individual. Cada gen es comparado con el resto calculando cómo de cerca están las anotaciones del gen con la del resto. Así, GS^2, para un conjunto de genes $\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$, vendría dado por:

$$GS^2(\mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{g_i \in \mathcal{G}} Comp(g_i, \mathcal{G} - \{g_i\}) \tag{5.49}$$

donde

$$Comp(g_i, \mathcal{H}) = \frac{1}{|G_i|} \sum_{j \in G_i} \frac{1}{|A_{\{j\}}|} \sum_{K \in A_{\{j\}}} \frac{Rank_{\mathcal{H}(k)}}{|\mathcal{H}|} \tag{5.50}$$

Siendo G_i el conjunto de términos GO con los que el gen g_i es anotado; $A_{\{j\}}$ el conjunto de términos que comprenden el camino desde el término j al nodo raíz de la ontología; y $Rank_H(k)$ es el número de genes del conjunto H anotados directa o indirectamente con el término k. Formalmente:

$$Rank_{\mathcal{H}}(k) = |\{g_j \in H : k \in A_{G_j}\}| \tag{5.51}$$

GS^2 promedia los valores de comparación para cada gen con el resto. Esta comparación de un gen (*target*) con el resto (*source*) devuelve un valor de 1 cuando los genes del *target* y el *source* comparten todos las mimas anotaciones, y, consecuentemente, todos los términos ancestros de tales anotaciones. Así, cuanto más similares sean los genes de entrada más cercano a 1 será el valor genrado por GS^2. De la misma forma, valores cercanos a 0 indicarán baja similitud.

Posteriormente, Richards et al. [256] propusieron otras medidas basadas en GO para evaluar la coherencia funcional de un conjunto de genes. Esas medidas están basadas en las propiedades topológicas del grafo compuesto por los genes y sus anotaciones GO, y consideran el enriquecimiento de las anotaciones y las relaciones entre anotaciones para determinar la significatividad de la coherencia funcional.

5.6. Resumen

En esta sección se ha analizado la utilidad de la validación de los resultados de técnicas de análisis de microarrays desde un punto de vista biológico. Esta validación, basada en la información biológica contrastada existente, es llevada a cabo mediante el contraste de información.

En primer lugar se han detallado los diferentes repositorios de datos biológicos existentes. Con tal fin, se ha realizado una distinción entre bases de datos de secuencias de aminoácidos, de genomas, de secuencias de proteínas, de familias de proteínas y de estructuras de proteínas. Posteriormente, las bases de datos especializadas GO y KEGG han sido estudiadas en detalle.

Posteriormente se han estudiado en profundiad las dos aproximaciones más relevantes para el análisis de la coherencia funcional de un conjunto de genes: herramientas de enriquecimiento y medidas de similitud semántica. Para la primera aproximación, se ha actualizado el estudio realizado por Khatri et al. [183] sobre herramientas de análisis funcional basadas, principalmente, en GO. Mientras que las medidas de similitud fueron divididas en similitud entre GO–terms, gene–products y genes.

Parte III

Propuestas

Capítulo 6

Herramienta de enriquecimiento basada en KEGG

No basta con adquirir sabiduría, es preciso además saber usarla.
CICERÓN.

Como se expuso en el capítulo 5, la etapa final del análisis de micorarray debería ser una fase donde se mida la significatividad biológica de los resultados generados. Esta verificación biológica engloba un amplio espectro de tareas que van desde el contraste de resultados con información biológica en repositorios conocidos hasta la verificación en laboratorio de las conclusiones obtenidas. El contraste de información puede considerarse una acción previa al laboratorio ya que ayuda a obtener conclusiones sobre la hipótesis de partida. Para llevar a cabo dicho contraste de información se ha diseñado una herramienta software denominada CARGENE, la cual permite medir y comparar el enriquecimiento de varios conjuntos de genes de manera simultánea según la información almacenada en KEGG y ofreciendo una interfaz gráfica amigable.

6.1. Evaluación por contraste de información

Como destaca Azuaje [21] o Halkidi [138], la mayoría de las técnicas de evaluación cuantifican la calidad de grupos de genes en términos de métricas que no consideran la información a priori en Biología (ver sección 4 para más detalle). Al evaluar los resultados de un algoritmo, el objetivo es conocer si éstos están provistos o no de significado biológico. En este contexto, la gran cantidad de información sobre genes recopilada en repositorios públicos es un elemento clave para acometer dicho objetivo y por ello la integración de herramientas de búsquedas para evaluar automáticamente grupos de genes frente a estos repositorios son de gran relevan-

cia (ver sección 5). En la sección 5.3 se detallaron las herramientas software más representativas que acometen dicho objetivo. Tales herramientas se basan, principalmente, en la base de datos Gene Ontology (GO) [18] y proporcionan un análisis de enriquecimiento (test de significatividad) a partir de un grupo de genes. Otras herramientas existentes se basan en el repositorio Kyoto Encyclopedia of Genes and Genomes (Kegg) [173, 172], tales como KCaM [16], PathwayVoyager [9] y KGML-ED [184] que ayudan en la visualización de rutas metabólicas de genomas concretos. En este caso existen menos herramientas que realicen un análisis de enriquecimiento o test de significatividad biológica. En este punto presentamos aportación y resultados de una herramienta para el cálculo de enriquecimiento de pathways Kegg dado un grupo de genes.

Medir el enriquecimiento de los genes con respecto a una base de datos concreta implica calcular las coincidencias de esos genes con la información almacenada y, posteriormente, dar una medida de la relevancia de esas coincidencias utilizando para ello cálculos estadísticos. Un contraste de hipótesis o test de significatividad es una técnica de inferencia estadística utilizada para juzgar si una propiedad que se supone cumple una población es compatible con lo observado en una muestra de dicha población.

6.2. CarGene

CarGene [5, 6] es una herramienta software diseñada para extraer y procesar información obtenida a partir de Kegg, para así evaluar (por contraste con información biológica) diversos grupos de genes obtenidos a partir de técnicas de inferencia. Con técnicas de inferencia nos referimos a técnicas computacionales que realicen análisis de microarray y que proporcionen listas de genes, clusters, biclusters o redes de genes, es decir, grupos de genes (ver sección 3 para más detalle). Una de las principales características de CarGene es la posibilidad de comparar y analizar estadísticamente, así como suministrar la información relacionada con diversos grupos de genes. Estas tareas las realiza tanto de forma visual como textual. Además, incluye un navegador propio para explorar la información relacionada con dichos genes y extraída de Kegg.

La funcionalidad de CarGene será descrita como sigue: en primer lugar, extracción de información y consulta del repositorio Kegg; en segundo lugar, los tests estadísticos implementados en la herramienta; en tercer lugar, cómo interpretar los resultados visuales así como usar el navegador; finalmente, mencionaremos algunos detalles de implementación de la herramienta.

6.2.1. Extracción de información biológica

CARGENE es una herramienta software que permite al usuario extraer información biológica de KEGG por medio de dos tipos de consultas diferentes. El primer tipo de consulta se realiza a partir de un organismo dado, proporcionando el listado de genes y rutas metabólicas o pathways asociados a dicho organismo. En el repositorio KEGG podemos encontrar la información genómica de una gran cantidad de organismos. Esta información se actualiza diariamente y se incrementa en unos diez organismos por mes. Utilizando este tipo de consultas, el usuario puede conocer el listado de genes involucrados en un cierto pathway o en qué pathways metabólicos participa un gen concreto.

El segundo tipo de consulta se basa en la verificación del grado de coherencia de un grupo de genes con respecto a la participación de los mismos en los pathways metabólicos almacenados en KEGG. Para utilizar esta funcionalidad, el usuario puede hacer referencia a un grupo de genes de dos maneras diferentes. La primera de ellas se basa en la utilización de un fichero para almacenar cada grupo de genes. Dicho fichero tiene el siguiente formato: el nombre tipo ORF de los genes organizados en una sola columna. La segunda de ellas permite almacenar varias grupos de genes en un solo fichero. El formato de este fichero es similar al anterior, sólo que junto a cada gen añadimos el número del grupo al que pertenece. Con este último tipo de fichero, CARGENE se adapta al formato de salida de la herramienta *Expander* [272], uno de los recursos bioinformáticos más utilizados para aplicar técnicas de análisis de micorarrays. Finalmente, indicar que en ambos formatos de fichero no hay restricciones sobre el nombre o la extensión del mismo y que se aceptan los comentarios, precediendo con el símbolo % cada línea que quiera ser comentada.

El usuario puede seleccionar grupos de ficheros de manera individual ó incluso tiene la posibilidad de seleccionar una carpeta entera. Una vez seleccionados los grupos de genes a validar, CARGENE consulta la base de datos KEGG, extrae información acerca de los pathways metabólicos en los que cada gen está involucrado y devuelve una medida estadística que nos informa acerca del nivel de enriquecimiento de cada grupo de genes con respecto a los datos consultados.

6.2.2. Medidas estadísticas

En cada ejecución, CARGENE utiliza la información extraída sobre pathways metabólicos para medir el nivel de enriquecimiento de los grupos de genes bajo análisis. Para llevar a cabo esta medición, CARGENE acepta o rechaza la siguiente hipótesis nula: "pertenecer a un grupo de genes Q es independiente de pertenecer a una ruta metabólica o pathway P". CARGENE calcula la probabilidad de encontrar como mínimo y como máximo m genes de una consulta Q de longitud q en la ruta

P si la hipótesis nula es cierta. La probabilidad (p–value) se calcula por medio del Test de Fisher [257], que emplea la probabilidad hipergeométrica exacta. Si el valor p–value asociado es igual o menor que un nivel de significatividad α, entonces la hipótesis nula se rechaza, es decir, el grupo de genes analizado Q está relacionado en cierto grado con la ruta metabólica P.

Escoger un valor $\alpha = 0{,}05$ implica contar con una probabilidad del 95 % de que la hipótesis nula sea cierta. Sin embargo, esta probabilidad decrece con el número de hipótesis a probar. Es decir, si yo pruebo dos hipótesis nulas independientes, entonces la probabilidad de encontrar que las dos son ciertas disminuye de la siguiente forma: $0{,}95 \times 0{,}95 = 0{,}90$. En el caso de CarGene, por cada consulta Q tenemos que probar esta hipótesis para todas las rutas P detectadas, es decir, estamos ante un problema de test de múltiples hipótesis. Para evitar el decrecimiento de esta probabilidad utilizamos correcciones para ajustar los p–values calculados.

La corrección de Bonferroni [50] es un procedimiento que ajusta el nivel de significatividad mediante la ecuación $\alpha' = \frac{\alpha}{k}$, siendo k el número de hipótesis. Tras esta corrección, una hipótesis nula será rechazada cuando el p–value sea menor a α'. Diversas herramientas utilizan como método de ajuste el proporcionado por Bonferroni ó Holm. Sin embargo, sí existen dependencias entre las hipótesis, en nuestro caso concreto entre los pathways de KEGG, se recomienda utilizar otros procedimientos de ajuste [44, 130]. A diferencia de las correcciones de Bonferroni, en el que cada valor p–value se corrige de manera independiente, las correcciones proporcionadas por Westfall y Young [306] tienen en cuenta la dependencia entre los pathways por medio de simulaciones en las que se permutan los genes que pertenecen a cada pathway. Este procedimiento estima el p–value como resultado de 1000 simulaciones de consultas Q utilizando el procedimiento Monte Carlo [189]. Tanto la corrección de Bonferroni como la de Westfall están implementadas en CarGene.

6.2.3. Representación de la información

Tanto la información extraída de Kegg como los resultados del análisis de enriquecimiento se presentan en CarGene de manera visual. El objetivo es proporcionar al usuario una forma gráfica de comparar el nivel de enriquecimiento de cada grupo de genes, de manera que se pueda rechazar fácilmente aquellos grupos no relevantes desde el punto de vista biológico. El usuario tienen la posibilidad de visualizar de manera individual, *single view mode*, o simultánea, *global view mode*, los resultados obtenidos para cada grupo de genes. Además, la herramienta proporciona un informe completo de todas la ejecuciones con cada grupo de genes de manera textual.

Vista global

El objetivo de esta vista es mostrar en una única ventana todas las ejecuciones realizadas para diferentes grupos de genes, resultado de diversos métodos computacionales. En la figura 6.1 observamos tres partes diferenciadas. A la izquierda, el árbol de ejecución, en el que observamos los resultados de tres algoritmos distintos de clustering y biclustering sobre la misma base de datos (CLICK [275], BIMAX [244] y OPSM [33]). Cada técnica ha producido 6, 5, y 5 grupos de genes respectivamente. El usuario puede elegir qué grupo de genes quiere analizar en detalle, teniendo en cuenta que éstos aparecen ordenados en la lista por orden de relevancia (en cuanto al análisis de enriquecimiento se refiere). En el caso de la figura, el usuario ha seleccionado los dos primeros resultados de cada técnica. La vista global de la herramienta proporciona una comparación visual, organizada y rápida de los grupos de genes seleccionados. Para cada ejecución de un método computacional, se muestran los pathways con su p–value asociado de manera creciente, desde el pathway menos significativo hacia el pathway más significativo, estadísticamente hablando. Así por ejemplo, en la figura 6.1 y para el algoritmo CLICK, el grupo de genes denominado cluster 3 tiene un p–value menor que el cluster 5.

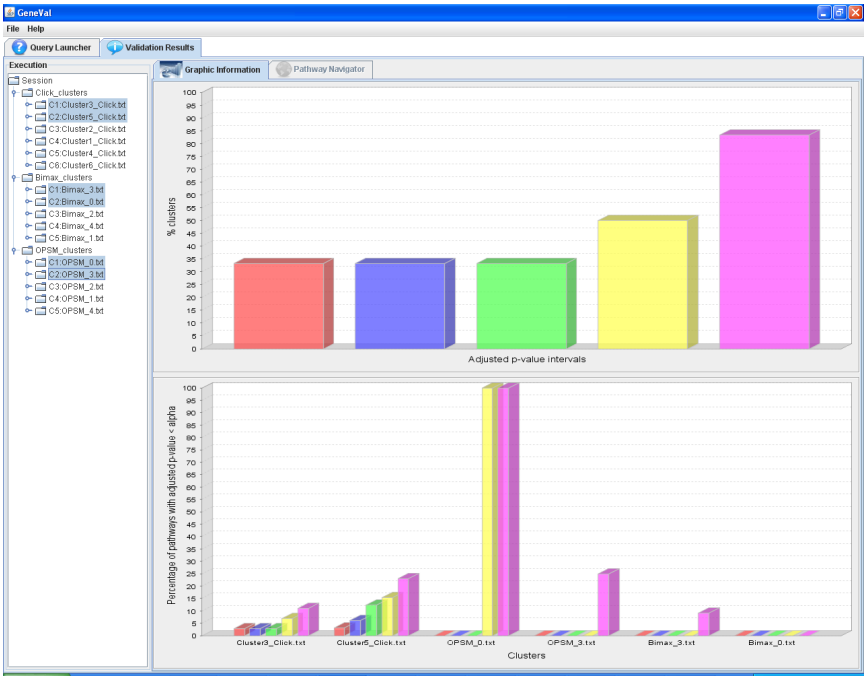


Figura 6.1: Vista global del análisis de enriquecimiento de pathways para los dos mejores grupos de genes proporcionados por tres métodos computacionales.

La gráfica superior derecha muestra el porcentaje de grupos de genes que contienen algún pathway con un p–value ajustado en los siguientes intervalos: [0,0.001),[0, 0.01),[0, 0.1), [0,1) y [0,5). Cada barra muestra el porcentaje de grupos cuyo p–value es menor al α del intervalo correspondiente. Así, por ejemplo, el 33 % de los grupos seleccionados por el usuario contienen pathways significativos para un ($\alpha \leq 0{,}001$). Las tres primeras barras muestran el mismo porcentaje, es decir, el 33 % de los grupos tienen al menos un pathway con un p–value ajustado menor que 0.1.

La gráfica inferior derecha muestra, por cada grupo de genes, el porcentaje de pathways con un p–value ajustado en alguno de los intervalos mencionados anteriormente. Por ello, cada grupo de genes tiene asociado cinco barras y los grupos que muestra son los seleccionados por el usuario en el árbol de ejecución.

CarGene proporciona más información sobre cada ejecución, además de la que se observa en la figura 6.1. Asociado con cada uno de los grupos de genes, tenemos un menú con opciones de visualización tales como la visualización en pantalla completa (porcentaje de genes por cada pathway, porcentaje de genes por cada pathway en grupos, porcentaje de pathways con un p–value ajustado $\leq \alpha$, porcentaje de conjuntos que contienen un p–value en el intervalo correspondiente) y finalmente un resumen textual.

Vista única

Esta vista proporciona dos tipos de información referente a un único grupo de genes. El primer tipo de información es referente a los p–values de cada pathway detectado en el grupo de genes. En este caso, se representa el porcentaje de pathways en el grupo de genes con un p–value ajustado en cada uno de los intervalos mencionados anteriormente. La herramienta también proporciona un resumen textual que incluye el nombre de los pathways, sus genes asociados, el p–value exacto y ajustado en cada caso, la fecha de ejecución, etc.

Con respecto al segundo tipo de información, *CarGene* proporciona un navegador para que el usuario pueda explorar la información relativa a cada uno de esos pathways que proporciona KEGG. Por ejemplo, la figura 6.2 representa el mapa asociado a la ruta metabólica que produce proteínas de Ribosoma. Este pathway aparece para el grupo de genes que forman el cluster 20 obtenido a partir del algoritmo CLICK. *CarGene* proporciona el mapa que representa al pathway seleccionado. En dicho mapa se marcan los genes pertenecientes al grupo de genes analizados en color rojo. Además, el usuario puede interactuar navegando por dicho mapa. Así, por ejemplo, el usuario puede acceder a la descripción detallada sobre el gen L21e (marcado en rojo en la Figura 6.2) como es el nombre del gen, motif, posición, secuencia, etc.

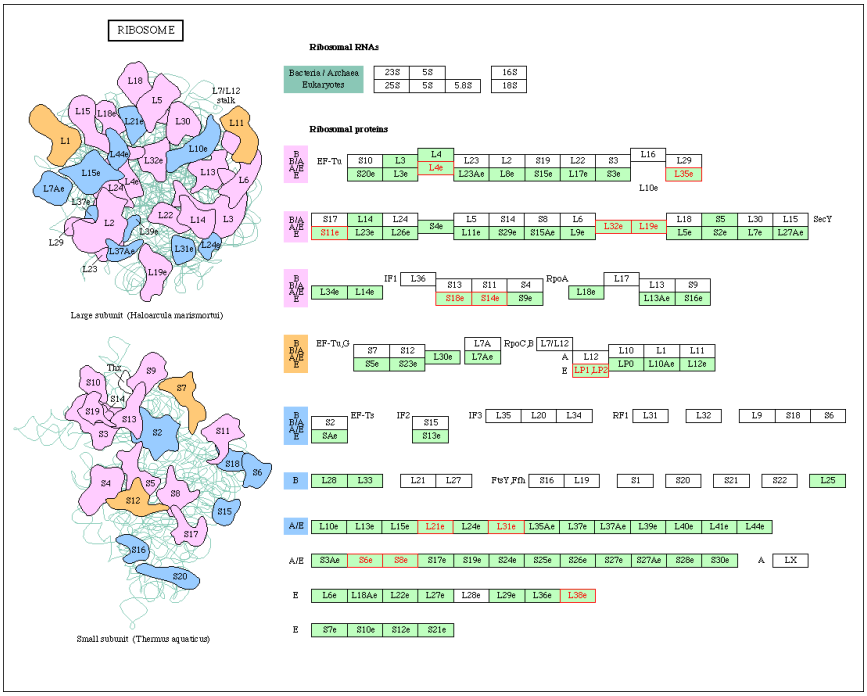


Figura 6.2: Representación de la proteína Ribosoma generada por KEGGY proporcionado por CARGENE.

Resumen estadístico

CARGENE ofrece la posibilidad de comparar diferentes resultados, en relación con determinados clusters o conjunto de ejecuciones es una característica muy útil de CARGENE. Por otra parte, un usuario puede comparar todas las ejecuciones desde el explorador CARGENE por medio de un informe textual que contiene un resumen estadístico de ellos. En este informe, para cada ejecución, se proporciona la siguiente información: a) Número de pathways diferentes detectados en todos los clusters (esto muestra la variedad de procesos biológicos detectados en las ejecuciones); b) Porcentaje de clusters que presentan un p–value ajustado dado (esto mide la calidad de los clusters en relación con los procesos biológicos).

6.2.4. Detalles de implementación

Java es el lenguaje de programación utilizado en el desarrollo de CARGENE. Se ha llevado a cabo una implementación de tipo multihilo y para el intercambio de información estructurada XML se ha utilizado el Simple Object Access Protocol (SOAP). De esta forma, el usuario puede lanzar diversos análisis de resultados al

mismo tiempo, pudiendo trabajar con la herramienta mientras ésta se encuentra ejecutando otras tareas. Por lo cual, es una herramienta especialmente pensada para plataformas multiprocesador. El núcleo estadístico está desarrollado en Java y utiliza JNI (Java Native Interface) para realizar llamadas a método nativos escritos en lenguaje C.

6.3. Conclusión

Este capítulo nos centramos en la necesidad de verificar la importancia biológica de los resultados obtenidos a partir de técnicas computacionales. Centrándonos en técnicas de enriquecimiento, la validación consistiría en medir el grado de coherencia de los miembros de un mismo grupo llevando a cabo un análisis a partir de información biológica ya conocida. Esta información biológica aparece en numerosos e importantes repositorios online que permiten un acceso rápido a una gran cantidad de datos. Una de esas bases de datos es KEGG, la cual almacena información sobre las rutas metabólicas en las que los genes de los organismos se ven involucrados.

El objetivo de la herramienta CarGene es el de evaluar el enriquecimiento de un grupo de genes a partir de la información almacenada en Kegg. Aparte de su interfaz gráfico amigable, uno de los mayores atractivos de CarGene es la posibilidad de comparar de manera simultánea la relevancia biológica de varios grupos de genes, tanto de manera visual como textual. Son varias las medidas estadísticas (test de Fisher, corrección de Bonferroni y corrección de Westfall-Young) las que se utilizan para medir la bondad de las relaciones de los genes de un grupo concreto con los pathways almacenados en Kegg.

La herramienta desarrollada analiza la relevancia de la coherencia de conjuntos de genes según la participación de éstos en procesos metabólicos, permitiendo obtener conclusiones de diferentes conjuntos de datos obtenidos por cualquier técnica de agrupamiento.

Capítulo 7

Disimilitud funcional de conjuntos de genes basada en GO

Es mejor saber después de haber pensado y discutido que aceptar los saberes que nadie discute para no tener que pensar.

FERNANDO SAVATER.

En este capítulo se describe la metodología para medir la disimilitud de un conjunto de genes ponderando la funcionalidad más cohesiva (común y específica), aportación principal de esta tesis. En primer lugar, se presenta la problemática de las medidas de similitud existentes y se justifica la investigación llevada a cabo. A continuación, la metodología propuesta es ampliamente descrita, utilizando para ellos dos ejemplos; uno ficticio y otro real. Posteriormente, la aproximación heurística de la medida propuesta es justificada y detallada, aplicando para ello los mismos ejemplos que para la aproximación exhaustiva. Finalmente, y tras exponer el concepto de GoGram para una representación gráfica de los resultados, las conclusiones más relevantes son presentadas.

7.1. Problemática y Justificación

En el capítulo 4 se presentaron diferentes aproximaciones para validar, analíticamente, el comportamiento de técnicas de aprendizaje automáticas empleadas para agrupar genes funcionalmente parecidos. En la sección posterior (5) se expusieron los problemas presentados por tales medidas y se describieron las alternativas existentes para validar la funcionalidad de grupos de genes desde un punto de vista biológico. En términos generales, dos aproximaciones fueron destacadas para el análisis de anotaciones genéticas: medidas de enriquecimiento (5.3) y medidas de similitud semántica (5.4).

Las herramientas de enriquecimiento, aunque han sido usadas en estudio de análisis de datos microarrays y dar un nivel de significatividad para un enriquecimiento concreto, sólo informan sobre la distribución de los datos y no aportan información sobre la relaciones inherentes, siendo un factor crucial para la comparación de conjuntos de genes.

Para solventar este problema, fueron presentadas las medidas de similitud semántica. Estas medidas han sido empleadas en diferentes aplicaciones [240], tales como la comparación de gene–products con diferentes funcionalidades o la predicción funcional de gene–products. Sin embargo, todas ellas, muestran una gran limitación cuando se enfrentan a genes involucrados en varias funcionalidades. Para tales genes, las aproximaciones actuales ponderan de igual forma a todas las funciones biológicas y no es posible escoger la más relevante considerando el contexto del resto de genes involucrados [183].

7.2. Propuesta de Metodología

En esta tesis doctoral se presenta una nueva alternativa para el cálculo de la similitud de un conjunto de genes de entrada: GFD (Gene Functional Dissimilarity) [92]. El objetivo es medir la disimilitud de un conjunto de genes ponderando la funcionalidad más cohesiva (común y específica) basada en el comportamiento global de todo el conjunto de genes de entrada. Debido al coste computacional que una aproximación exhaustiva conllevaría, se propone una aproximación heurística basada en Diagramas de Voronoi para su cálculo. La comparación realizada entre la aproximación exhaustiva y la heurística indican que la reducción del coste computacional no afecta significativamente a la calidad de los resultados.

GFD está basada en *Gene Ontology* (ver 5.2.1) y asigna valores numéricos en el intervalor (0, 1) para un conjunto de genes para cada una de las tres ontologías GO. Con el objetivo de visualizar gráficamente la similitud de varios conjuntos de datos se propone el concepto de GoGRAM ; representación en $3D$ de las tres ontologías GO bajo unos ejes isométricos.

Finalmente, la experimentación llevada a cabo confirma que la elección de la funcionalidad más cohesiva es más robusto que las aproximaciones existentes.

7.3. Descripción

A lo largo de esta sección se describirá en detalle la medida GFD. Con tal fin, se mostrará la metodología de la medida propuesta usando un ejemplo ficticio y, posteriormente, será aplicada a un caso real a modo de ejemplo.

7.3.1. Metodología

GFD está basada en una adaptación de la estructura GO–tree presentada en [196]. La estructura es usada para desarrollar una novedosa medida de disimilitud de términos GO que será empleada en calcular la disimilitud de un conjunto de genes. Ésta sería el primer estudio de una medida que evalúa a un conjunto de genes teniendo en cuenta la funcionalidad más cohesiva encontrada en el conjunto. El método comprende una búsqueda de la funcionalidad más específica para cada gen que, además, es similar a las otras funcionalidades encontradas en el conjunto de genes.

Esta metodología es esbozada en la figura 7.1, la cual presenta un ejemplo ficticio de un conjunto de cuatro genes. El cálculo de GFD comprende cinco pasos consecutivos que son descritos en las siguientes subsecciones.

Primera fase: *Identificación de genes*

El primer paso consiste en encontrar el representante de cada gen de entrada en GO. Asumamos que GO contiene Θ genes para un organismo específico y que A es el conjunto de genes a ser evaluado. Cada gen $g \in A$ es buscado en Θ, y si la búsqueda es insatisfactoria, g es transformado en un sinónimo g' usando la información de genes sinónimos dada en GOA [27]. Es decir, el conjunto inicial de genes $A = \{g_1, \ldots, g_n\}$ es transformado en $A' = \{g'_1, \ldots, g'_{n'}\}$, donde cada g' está presente en Θ. Sin embargo, el gen es eliminado si no existe ningún sinónimo. Por ejemplo, en la figura 7.1, los genes $g1$ y $g3$ son encontrados en GO, el $g2$ fue transformados a un sinónimo ($g2'$), y el gen $g4$ no fue encontrado.

Segunda fase: *Identificación de función génica*

La segunda fase corresponde a la identificación de la funcionalidad de los genes en el conjunto. Cada gen es transformado en las diferentes proteínas codificadas por cada gen (***gene–product***) según la base de datos *Entrez Gene* [213]. De este modo, un conjunto de gene–products $\mathcal{H}(i) = \{g_i p_1, \ldots, g_i p_m\}$ es asociado con cada $g'_i \in A'$. Continuando con el ejemplo anterior, g_1 codifica a la proteína $g_1 p_1$, g_2 a $g_2 p_1$, y g_3 a las proteínas $g_3 p_1$, $g_3 p_2$ y $g_3 p_3$.

$$\mathcal{H}(i) \begin{cases} \mathcal{H}(g_1) = \{g_1 p_1\} \\ \mathcal{H}(g_2) = \{g_2 p_1\} \\ \mathcal{H}(g_3) = \{g_3 p_1, g_3 p_2, g_3 p_3\} \end{cases} \tag{7.1}$$

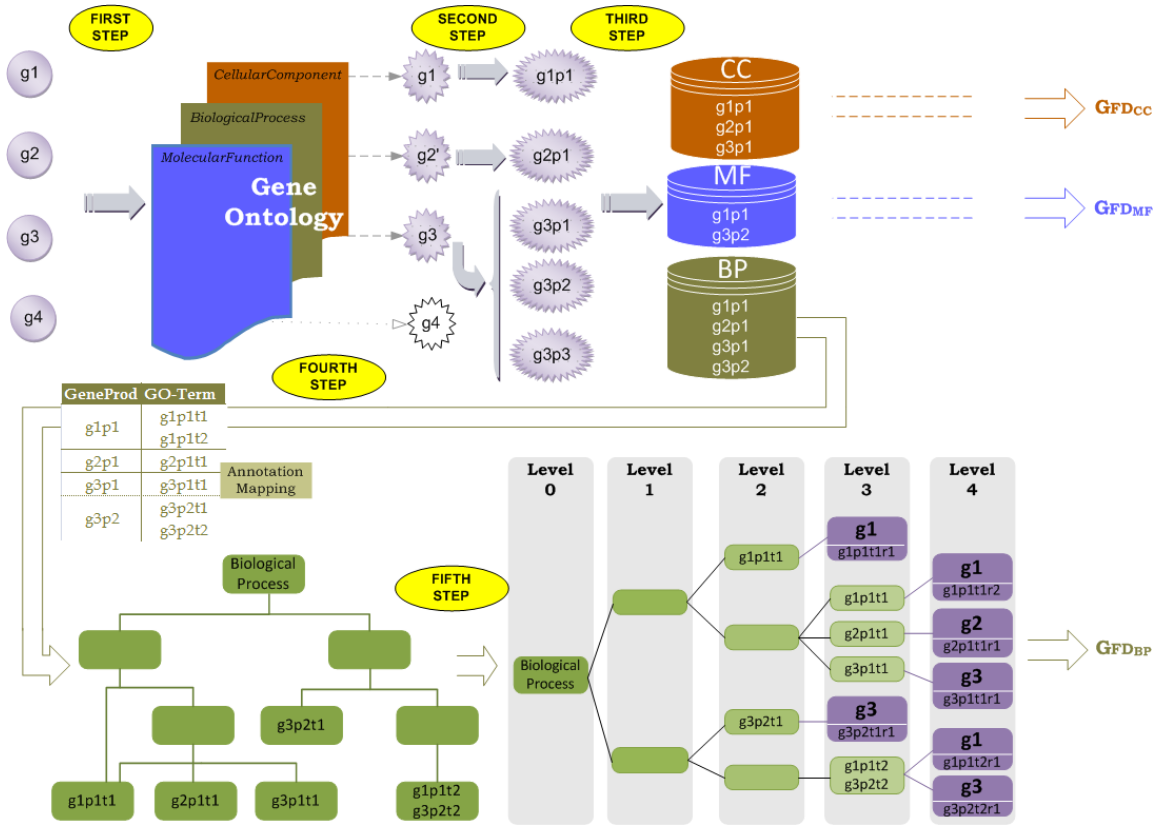


Figura 7.1: Esquema global de la metodología para calcular la disimilitud funcional de un conjunto de genes según GFD. Las tres primeras etapas son comunes a las tres ontologías. Las dos últimas son ilustradas sólo para la ontología *Biological Process*.

Tercera fase: *Filtrando gene–products*

En esta fase, cada funcionalidad génica es filtrada para los tres dominios de GO. Las proteínas seleccionadas en la etapa previa son eliminadas de aquellos dominios en los que no estén involucradas, y son seleccionadas en otra caso. Así, en la figura 7.1, la proteína g_1p_1, codificada por el gen g_1, forma parte de todas las ontologías; la proteína codificada por el gen g_2 (g_2p_1) está presente en las ontologías *Biological Process* y *Cellular Component*; y el gen g_3 es representado por la proteína g_3p_2 en la ontología *Molecular Function*, por las proteínas g_3p_1 y g_3p_2 en *Biological Process*, y por la proteína g_3p_1 en la ontología *Cellular Component*. Una vez los genes de entrada han sido transformados en sus funcionalidades biológicas y éstas han sido filtradas por cada dominio, el siguiente paso debe ser repetido por cada una de las tres ontologías, suministrando tres resultados diferentes (uno por cada ontología). Por cuestiones de claridad, sólo será considerada al ontología *Biological Process* en las siguientes descripciones y ejemplos.

Cuarta fase: *Búsqueda de anotaciones de gene–product*

Para cada ontología, son examinadas las anotaciones de cada ontología. Una proteína concreta puede estar asociado con o localizada en una o más componentes celulares, es activa en uno o más procesos biológicos, durante los cuales puede llevar a cabo varias funciones moleculares. Esta característica es tenida en cuenta en GO: cada anotación funcional es identificada por un único termino GO ***GO–term***.

En la figura 7.1, las anotaciones para la ontología *Biological Process* son representadas. Por cada g_ip_j es obtenido un conjunto de GO–terms, i.e., $\mathcal{H}(i,j) = \{g_ip_jt_1, \ldots, g_ip_jt_q\}$. Por ejemplo, la proteína g_1p_1 tiene dos términos diferentes en el dominio *Biological Process*. Ambas funcionalidades son representadas por lo términos $g_1p_1t_1$ y $g_1p_1t_2$. Aplicando este concepto a las diferentes proteínas codificadas por los genes de entrada sería:

$$\mathcal{H}(i,j)\begin{cases} g_1 \rightarrow \mathcal{H}(g_1,p_1) = \{g_1p_1t_1, g_1p_1t_2\} \\ g_2 \rightarrow \mathcal{H}(g_2,p_1) = \{g_2p_1t_1\} \\ g_3 \rightarrow \begin{cases} \mathcal{H}(g_3,p_1) = \{g_3p_1t_1\} \\ \mathcal{H}(g_3,p_2) = \{g_3p_2t_1, g_3p_2t_2\} \end{cases} \end{cases} \tag{7.2}$$

Quinta fase: *Funcionalidad de gene–product*

En esta etapa, cada anotación funcional en GO es identificada. Para ello, el grafo acíclico dirigido de GO (DAG) es usado, pero sólo considerando las anotaciones "is a". Esta aproximación no usa las relaciones 'part of" por tres razones:

a) nos gustaría comparar los resultados de las tres ontologías, y la ontología *Molecular Function* no posee relaciones "part of" ; b) la relación "part of" es usada en la ontología *Biological Process* cuando un nodo hijo es una instancia de sólo una porción del proceso padre; c) las tres ontologías son ahora "is a" completas, implicando que cada término tiene una ruta hasta el nodo raíz que sólo pasa por relaciones "is a". En la figura 7.1, el término $g_1p_1t_2$ es idéntico al término $g_3p_2t_2$. Por consiguiente, ambas funciones están localizadas en el mismo nodo del DAG de GO. Seguidamente, la información recopilada es transformada en una estructura árbol (*GO–tree*). Un nodo estará presente en el GO–tree tantas veces como rutas diferentes haya en el DAG desde tal nodo a la raíz. Por ejemplo, en la figura 7.1, $g_1p_1t_1$ tiene dos caminos para llegar al nodo raíz de la ontología *Biological Process*, así este nodo es duplicado en el GO–tree resultante.

Una vez la estructura árbol es construida, los genes de entrada son añadidos como nodos hoja en el GO–tree. La posición de tales nodos designa la anotación funcional encontrada entre el conjunto de genes. Cada posición es determinada según los términos GO y la proteína producto del gen. Nótese que un gen puede estar presente en diferentes hojas, las cuales son diferentes ***representaciones*** del gene para diferentes dominios. Cada término GO $g_ip_jt_k$ tendrá un número de representaciones en GO (ruta desde el termino GO a la raíz), ya que puede estar presente en diferentes lugares del GO–tree. Este conjunto de representaciones es denotado por $\mathcal{H}(i,j,k) = \{g_ip_jt_kr_1, \ldots, g_ip_jt_kr_s\}$, donde $r_1 \ldots r_s$ denota las representaciones del término $g_ip_jt_k$.

Después de ser construido el GO–tree, los genes de entrada pueden ser evaluados. En este punto, la información inicial $A = \{g_1, \ldots, g_4\}$ ha sido transformada en tres representaciones del g_1, una representación de g_2, y tres de g_3.

$$\mathcal{H}(i,j,k)\begin{cases} g_1 \rightarrow g_1p_1 \rightarrow \begin{cases} \mathcal{H}(g_1,p_1,t_1) = \{g_1p_1t_1r_1, g_1p_1t_1r_2\} \\ \mathcal{H}(g_1,p_1,t_2) = \{g_1p_1t_2r_1\} \end{cases} \\ g_2 \rightarrow g_2p_1 \rightarrow \mathcal{H}(g_2,p_1,t_1) = \{g_2p_1t_1r_1\} \\ g_3 \rightarrow \begin{cases} g_3p_1 \rightarrow \mathcal{H}(g_3p_1t_1) = \{g_3p_1t_1r_1\} \\ g_3p_2 \rightarrow \begin{cases} \mathcal{H}(g_3p_2t_1) = \{g_3p_2t_1r_1\} \\ \mathcal{H}(g_3p_2t_2) = \{g_3p_2t_2r_1\} \end{cases} \end{cases} \end{cases} \tag{7.3}$$

Cada una de estas representaciones está localizada en una estructura que por sí sola provee información útil. La medida funcional para un conjunto de genes GFD, la cual es descrita en detalle abajo, está basada en la similitud de las representaciones de cada gen y es soportada por la estructura GO–tree.

7.3.2. Disimilitud funcional de representaciones de genes ($\mathcal{R}$)

Dada dos representaciones génicas r_α y r_β, la disimilitud entre ambas esta dada por:

$$\mathcal{R}(r_\alpha, r_\beta) = \frac{length(r_\alpha, r_\beta)}{depth(r_\alpha) + depth(r_\beta)} \tag{7.4}$$

donde $length(r_\alpha, r_\beta)$ denota el número mínimo de nodos que separa r_α de r_β en el GO–tree (i.e., el número de nodos de la ruta desde r_α hasta r_β) y $depth$ indica el nivel de la representación en el GO–tree.

Desde un punto de vista biológico, *length* indica la relación funcional entre ambos GO–terms, mientras que $depth$ indica el nivel de especificidad de la representación. De este modo, la medida penaliza a aquellos pares de representaciones génicas que están muy separadas, y premia la especialización. Esta medida provee valores entre 0 y 1, donde valores cercanos al 0 significa "similar", y valores cercano al 1 significa "distinto".

Dos representaciones génicas, r_α y r_β, presentan la mejor similitud cuando comparten los mismos padres ($length(r_\alpha, r_\beta) = 1$) y sus profundidades son máximas ($depth(r_\alpha) = depth(r_\beta) = k$). En este caso, su disimilitud funcional sería:

$$\mathcal{R}(r_\alpha, r_\beta) = \frac{1}{k + k} \approx 0$$

En cambio, el peor caso ocurriría cuando las dos representaciones están abajo del árbol ($depth(r_\alpha) = depth(r_\beta) = k$), y no comparten ningún nodo ancestro mas que el nodo raíz ($length(r_\alpha, r_\beta) = k + k - 1$):

$$\mathcal{R}(r_\alpha, r_\beta) = \frac{2k - 1}{2k} \approx 1$$

Por ejemplo,

$$\mathcal{R}(g_1 p_1 t_2 r_1, g_3 p_2 t_2 r_1) = \frac{1}{4 + 4} = 0{,}125$$

$$\mathcal{R}(g_1 p_1 t_1 r_2, g_3 p_2 t_2 r_1) = \frac{7}{4 + 4} = 0{,}875$$

Para este conjunto de genes ($\{g_1, g_3\}$), los valores mínimos y máximos de $\mathcal{R}$ son 0,125 y 0,875, respectivamente.

7.3.3. Medida de Disimilitud Funcional: GFD

La disimilitud funcional está basada en la disimilitud de representaciones génicas descrita anteriormente. Esta aproximación extrapola esta medida de disimilitud para evaluar la homogeneidad de conjuntos de genes.

Dado un conjunto de genes $A = \{g_1, g_2, \ldots, g_n\}$, el conjunto de representacio-

nes para un gene g_i es dado por $\mathcal{T}(g_i)$, como muestra la ecuación 7.5, (ver $\mathcal{H}(i, j, k)$ en la quinta fase).

$$\mathcal{T}(g_i) = \bigcup_{\substack{j \in \mathcal{H}(i) \\ k \in \mathcal{H}(i,j)}} \mathcal{H}(i, j, k) \tag{7.5}$$

El producto cartesiano $\mathcal{P}(A) = \mathcal{T}(g_1) \times \mathcal{T}(g_2) \times \cdots \times \mathcal{T}(g_n)$ define el conjunto de todos los posibles conjuntos de representaciones. La disimilitud S de un conjunto de representaciones $p \in P$ es dado por la ecuación 7.6, donde R es la dismilitud de dos representaciones génicas calculada según la ecuación 7.4. Nótese que $|p| = |A|$.

$$\mathcal{S}(p) = \frac{1}{\begin{pmatrix} |p| \\ 2 \end{pmatrix}} \sum_{\forall \delta, \gamma | 0 < \delta < \gamma \leq |p|} \mathcal{R}(p[\delta], p[\gamma]) \tag{7.6}$$

Finalmente, la disimilitud funcional basada en GO, GFD, es la mínima disimilitud para todas los posibles conjuntos de representaciones para un conjunto de genes dado A.

$$\text{GFD}(A) = \min_{p \in \mathcal{P}(A)} \mathcal{S}(p) \tag{7.7}$$

En la figura 7.1 existen siete representaciones diferentes para el conjunto de genes de entrada A; tres para $\mathcal{T}(g_1)$, una para $\mathcal{T}(g_2)$, y tres para $\mathcal{T}(g_3)$:

$$\mathcal{T}(g_i \in A) \begin{cases} \mathcal{T}(g_1) = \{g_1p_1t_1r_1, g_1p_1t_1r_2, g_1p_1t_2r_1\} \\ \mathcal{T}(g_2) = \{g_2p_1t_1r_1\} \\ \mathcal{T}(g_3) = \{g_3p_1t_1r_1, g_3p_2t_1r_1, g_3p_2t_1r_2\} \end{cases} \tag{7.8}$$

Esta representaciones pueden generar nueve posibles conjuntos de representaciones $3 \times 1 \times 3$), con lo que $|\mathcal{P}(A)| = 9$:

$$\begin{aligned} \mathcal{P}(A) = \; & \{\{g_1p_1t_1r_1 \;,\; g_2p_1t_1r_1 \;,\; g_3p_1t_1r_1\} \\ & \{g_1p_1t_1r_1 \;,\; g_2p_1t_1r_1 \;,\; g_3p_2t_1r_1\} \\ & \{g_1p_1t_1r_1 \;,\; g_2p_1t_1r_1 \;,\; g_3p_2t_1r_2\} \\ & \{g_1p_1t_1r_2 \;,\; g_2p_1t_1r_1 \;,\; g_3p_1t_1r_1\} \\ & \{g_1p_1t_1r_2 \;,\; g_2p_1t_1r_1 \;,\; g_3p_2t_1r_1\} \\ & \{g_1p_1t_1r_2 \;,\; g_2p_1t_1r_1 \;,\; g_3p_2t_1r_2\} \\ & \{g_1p_1t_2r_1 \;,\; g_2p_1t_1r_1 \;,\; g_3p_1t_1r_1\} \\ & \{g_1p_1t_2r_1 \;,\; g_2p_1t_1r_1 \;,\; g_3p_2t_1r_1\} \\ & \{g_1p_1t_2r_1 \;,\; g_2p_1t_1r_1 \;,\; g_3p_2t_1r_2\}\} \end{aligned} \tag{7.9}$$

Existen dos representaciones óptimas para g_1 ($g_1p_1t_1r_2$ y $g_1p_1t_2r_1$) y otras dos para g_3 ($g_3p_1t_1r_1$ y $g_3p_2t_2r_1$), que generan cuatro configuraciones óptimas. Sin embargo, sólo existe una combinación funcional óptima según la función cohesiva para todos los genes. Seleccionando de forma aleatoria, podríamos obtener el peor caso $S(g_1p_1t_1r_1, g_2p_1t_1r_1, g_3p_2t_2r_1) = 0{,}768$, a diferencia del mejor caso $S(g_1p_1t_1r_2, g_2p_1t_1r_1, g_3p_1t_1r_1) = 0{,}428$.

Es importante remarcar que esta aproximación no selecciona el mejor término GO para cada gene individualmente; sino que busca la función más común y específica para el conjunto completo de genes. De esta forma, GFD es bastante diferente a las medidas especialmente diseñadas para el análisis funcional de grupos de genes (ver sección 5.5.3): GS^2y las propuestas por Richard et al. [256]. GFD sólo selecciona una funcionalidad por cada gene (la funcionalidad más cohesiva globalmente), mientras que GS^2 y las propuestas por Richard et al. consideran todas las funciones para cada gen con igual importancia.

7.3.4. Un ejemplo real: ABC transporter

Con el fin de mostrar la metodología propuesta, es seleccionado los genes de la ruta metabólica *"ABC transporter"* del organismo *Saccharomyces Cerevisiae* almacenado en KEGG (*sce02010*).

La ruta *ABC (ATP-binding cassette) transporter* forma uno de las familias de proteínas más conocidas, y que están presentes en bacterias, arqueas[1] y eucariotas. Estas proteínas se acoplan al ATP hidrólisis para activar el transporte de una gran variedad de substratos, tales como iones, azúcares, lípidos, esteroles, péptidos o proteínas.

Concretamente, el pathway *sce02010* está compuesto de tres genes:

$$A = \{YDR011W, YMR301C, YOR153W\}$$

Éstos genes no poseen ninguna anotación en GO aunque sí existen sinónimos de ellos que las poseen. Así, el conjunto de datos es transformado a A'={SNQ2, ATM1,PDR5}. Cada uno de estos genes codifica a una proteína que a su vez posee varias anotaciones en las diferentes ontologías (ver tabla 7.1). Por ejemplo, el sinónimo SNQ2 codifica al gene–product con identificador 9215296 ($\mathcal{H}(i)$) y cuya descripción está relacionada con el *ABC transporter*. Esta proteína, a su vez, contiene 6 anotaciones en la ontología MF, 3 en BP y 5 en CC ($\mathcal{H}(i, j)$). En el caso concreto de la ontología BP, los términos en los que está anotada son GO:0000304,

[1]Las arqueas son un grupo de microorganismos unicelulares procariotas que carecen de núcleo o cualquier otro orgánulo dentro de la célula.

Tabla 7.1: Información de ABC transporter extraída de GO.

Gene	**Sinónimo**	$\mathcal{H}(i)$ **Gene–Product** identificador	descripción	$\mathcal{H}(i,j)$ **GO–term** ont	identificador	descripción	$\lvert\mathcal{H}(i,j,k)\rvert$ **Rep.**
YDR011W	SNQ2	9215296	Plasma membrane ATP-binding cassette (ABC) transporter, multidrug transporter involved in multidrug resistance and resistance to singlet oxygen species		*GO:0000166*	nucleotide binding	1
					GO:0042626	ATPase activity, coupled to transmembrane movement of substances	4
				MF	*GO:0017111*	nucleoside-triphosphatase activity	1
					GO:0005524	ATP binding	5
					GO:0008559	xenobiotic-transporting ATPase activity	6
					GO:0016887	ATPase activity	1
					GO:0000304	response to singlet oxygen	3
				BP	***GO:0006810***	transport	1
					GO:0042493	response to drug	1
					GO:0005886	plasma membrane	1
					GO:0016020	membrane	1
				CC	***GO:0005739***	mitochondrion	4
					GO:0016021	integral to membrane	1
					GO:0005624	membrane fraction	1
YMR301C	ATM1	9211626	Mitochondrial inner membrane ATP-binding cassette (ABC) transporter, exports mitochondrially synthesized precursors of iron-sulfur (Fe/S) clusters to the cytosol		*GO:0000166*	nucleotide binding	1
					GO:0042626	ATPase activity, coupled to transmembrane movement of substances	4
				MF	*GO:0017111*	nucleoside-triphosphatase activity	1
					GO:0005524	ATP binding	5
					GO:0016887	ATPase activity	1
					GO:0006810	transport	1
					GO:0006811	ion transport	1
				BP	GO:0006826	iron ion transport	1
					GO:0006879	cellular iron ion homeostasis	7
					GO:0055085	transmembrane transport	2
					GO:0016020	membrane	1
					GO:0005739	mitochondrion	4
				CC	*GO:0016021*	integral to membrane	1
					GO:0005743	mitochondrial inner membrane	8
YOR153W	PDR5	9214093	Plasma membrane ATP-binding cassette (ABC) transporter, multidrug transporter actively regulated by Pdr1p		*GO:0000166*	nucleotide binding	1
					GO:0042626	ATPase activity, coupled to transmembrane movement of substances	4
				MF	*GO:0017111*	nucleoside-triphosphatase activity	1
					GO:0005524	ATP binding	5
					GO:0008559	xenobiotic-transporting ATPase activity	6
					GO:0016887	ATPase activity	1
					GO:0006810	transport	1
					GO:0042493	response to drug	1
				BP	GO:0015893	drug transport)	1
					GO:0046677	response to antibiotic	1
					GO:0046898	response to cycloheximide	1
					GO:0005886	plasma membrane	1
					GO:0016020	membrane	1
				CC	***GO:0005739***	mitochondrion	4
					GO:0016021	integral to membrane	1

GO:0006810 y GO:0042493, que contienen tres, una y una representación, respectivamente.

En la tabla 7.1 además de la información almacenada en GO para los genes involucrados en el pathway *sce02010*, son escritos en cursiva los identificadores de aquellos términos GO en los que están anotados todos los genes de entrada, y en negrita el término concreto que describe la funcionalidad usada por GFD para medir la similitud del conjunto de datos de entrada.

Para la ontología MF los genes YDR011W y YOR153W comparten sus seis anotaciones, y para el gen YMR301C, aunque también comparte todas, posee una menos. Así, los términos GO:0000166, GO:0042626, GO:0017111, GO:0005524 y GO:0016887, están anotados para todos los genes. De entre todas las funcionalidades en las que se anotan los genes de entrada, es el término compartido GO:0042626 el seleccionado por GFD como funcionalidad más cohesiva al conjunto total de entrada. Ésta es seleccionadA tras explorar 3888 combinaciones posibles de las 48 (18 + 12 + 18) representaciones de los términos en los que se anotan los genes en estudio, resultando una disimilitud de 0,0455.

En la ontología BP, los genes presentan una funcionalidad más dispersa, compartiendo tan sólo el término GO:0006810 que describe la funcionalidad de *transporte*. La metodología propuesta, de forma automática y tras explorar las 280 combinaciones posibles, selecciona al único término compartido como funcionalidad más cohesiva. Nótese que la metodología es capaz de obviar proceso que no están directamente relacionados con el transporte, como podría ser los procesos GO:0005886 o GO:0006879. A partir de este proceso común y específico, la valoración obtenida por GFD es 0,1667.

El comportamiento de los genes para CC es similar a MF, ya que éstos son anotados en diferentes términos compartidos. Concretamente, se trata de las localizaciones GO:0016020, GO:0005739 y GO:0016021, siendo la segunda de ellas el término seleccionado por GFD para medir la disimilitud según su localización. El valor obtenido tras explorar 784 posibles combinaciones es de 0,0833.

Por otro lado, recuérdese que GFD se basa en el estudio de las representaciones de cada término GO ($\mathcal{H}(i, j, k)$). De manera que para cada ontología se ha seleccionado una representación concreta de todas las que tiene el término elegido como más cohesivo. Así, para la ontología BP es seleccionado la única representación del término GO:0006810, mientras que para MF y CC la representación seleccionada es una de las 4 que posee los términos GO:0042626 y GO:0005739, respectivamente.

Con el fin de realizar una descripción más detallada sobre la metodología propuesta y la selección de las representaciones usadas por GFD para evaluar a los genes que componen el pathway *sce02010* es mostrado el GO–tree generado para la ontología MF (ver figura 7.2). Ésta ha sido seleccionada como ejemplo porque,

como se explicó anteriormente, el término usado por GFD posee diferentes representaciones y los genes están anotadas en una gran variedad de funcionalidades.

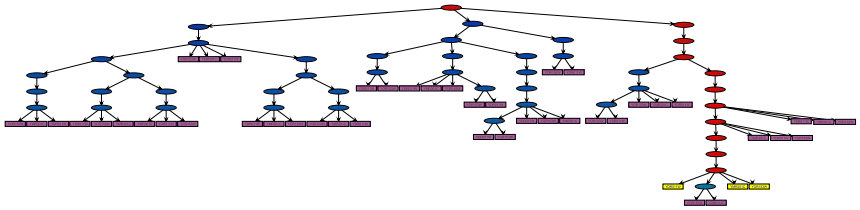
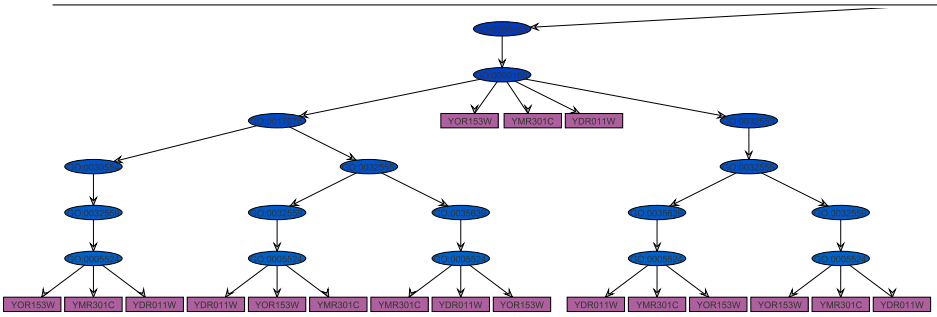


Figura 7.2: GO–tree para *ABC transporter*.

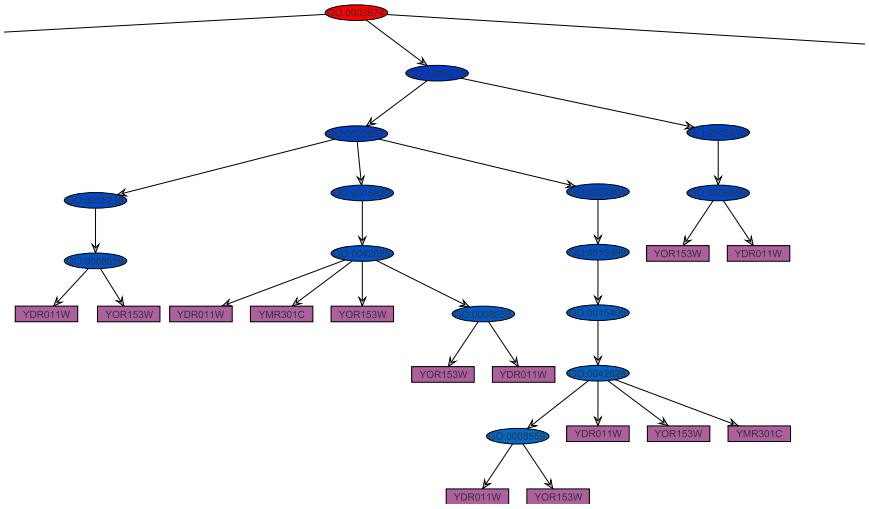
En la figura 7.2, se muestra con una elipse los diferentes términos que componen el GO–tree, mientras que son representados con un rectángulo los genes. Igualmente, los términos que están relacionados con la representación seleccionada por GFD para cada gen son coloreados en rojo, y en amarillo la representación del gen escogida. Para una mejor visualiación el árbol representado en la figura 7.2 ha sido dividido en tres subárboles en la figura 7.3.

De esta manera, por ejemplo, vemos que el gen YDR011W aparece representado en el árbol en 18 ocasiones; una anotada con el término GO:0000166 (subárbol izquierdo), cuatro con GO:00042626 (subárbol central y derecho), una con GO:0017111 (subárbol derecho), cinco con GO:0005524 (subárbol izquierdo y central), seis con GO:0008559 (subárbol central y derecho) y una con GO:0016887 (subárbol derecho). De entre todas las representaciones es selecciona la que está asociada con el término GO:00042626 (subárbol derecho) y cuyo camino al nodo raíz está en rojo.

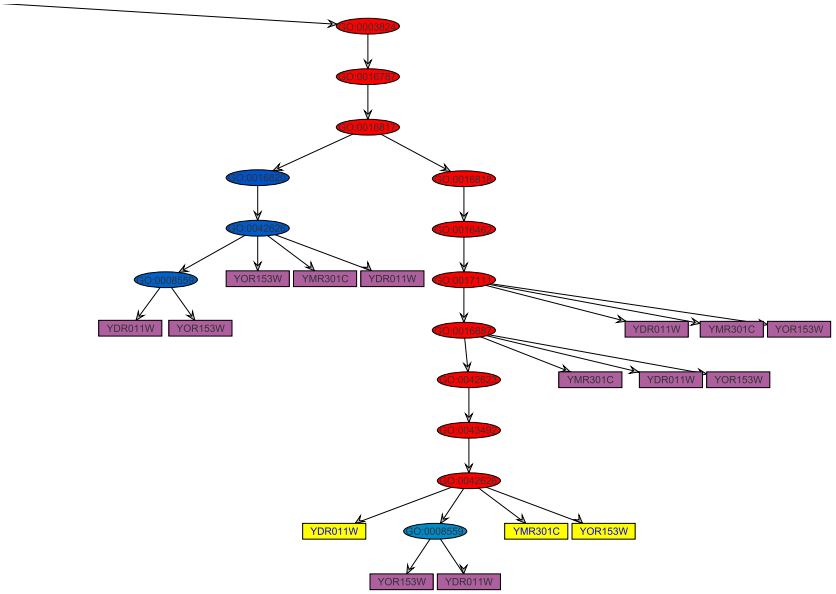
De una forma visual, podemos observar como la metodología, de entre todas las posibles representaciones de cada gen, ha seleccionado la más común y específica haciéndose notoria cómo la representación del término GO:0008559 (hijo de la representación seleccionada) no ha sido elegida por no se la más común aunque sea la más específica para los genes YDR011W y YOR153W.



(a) *Subárbol izquierdo*



(b) *Subárbol central*



(c) *Subárbol derecho*

Figura 7.3: Subárboles para *ABC transporter*

7.4. Aproximación Heurística

7.4.1. Justificación

La medida de coherencia funcional de un conjunto de genes propuesta está basada en el cálculo de la disimilitud de todas las posibles combinaciones de las representaciones de los genes de entrada. Considerando que el conjunto de entrada tiene n genes, que cada gen codifica p gene–products, cada gene–product está anotado en t GO–terms en cada ontología, y el número medio de representaciones de cada término GO es r, el orden computacional de la medida de similitud sería:

$$T(n) \in \Theta((p \times t \times r)^n \times n^2) = \Theta(K^n \times n^2)$$

Donde $K = p \times t \times r$ simboliza la cantidad de representaciones por cada gen; K^n al número de todos los posibles conjuntos de representaciones (ver ecuación 7.3.3);y n^2 al número de pares de representaciones a evaluar por cada combinación(ver ecuación 7.3.3). Consecuentemente, el cálculo exhaustivo de GFD tiene una complejidad computacional alta, haciéndola *intratable* para grandes conjuntos de datos.

Debido a que la medida debería ser capaz de calcular la homogeneidad de cualquier conjunto de datos de una forma eficiente, se hace plausible la necesidad de incorporar una técnica heurística para explorar el espacio de búsqueda. Con tal fin, se propone el uso del concepto de diagramas de Voronoi [20] para reducir la complejidad del algoritmo descrito.

7.4.2. Descripción

La aproximación heurística propuesta se basa en usar el mismo GO–tree formado por el conjunto de genes de entrada A como espacio de búsqueda. Para cada individuo del espacio, nodo del árbol, se calcularía un valor representate que vendría dado por la disimilitud del conjunto formado por la representación más cercana de cada gen al nodo en cuestión. Finalmente, el menor valor calculado es el seleccionado como solución.

Dado el conjunto de genes de entrada $A = \{g_1, g_2, \ldots, g_n\}$ y el GO–tree generado por tal conjunto, el valor de similitud representante para cada nodo Δ sería:

$$\mathcal{S}_\Delta(A) = \mathcal{S}(p_\Delta) \tag{7.10}$$

Donde $\mathcal{S}$ simboliza la disimilitud de un conjunto de representaciones (ver ecuación 7.3.3) y p_Δ al conjunto formado por la representación más cercana de cada gen $g_i \in A$ al nodo Δ. Cada representación vendría determinada por:

$$\mathcal{T}_\Delta(g_i) = \{r \in \mathcal{T}(g_i) : min(\mathcal{R}(r, r_\Delta))\} \tag{7.11}$$

Siendo $\mathcal{T}(g_i)$ el conjunto de representaciones del gen g_i (ver ecuación 7.5), $\mathcal{R}$ la disimilitud entre dos representaciones (ver ecuación 7.4) y r_Δ la representación del nodo Δ.

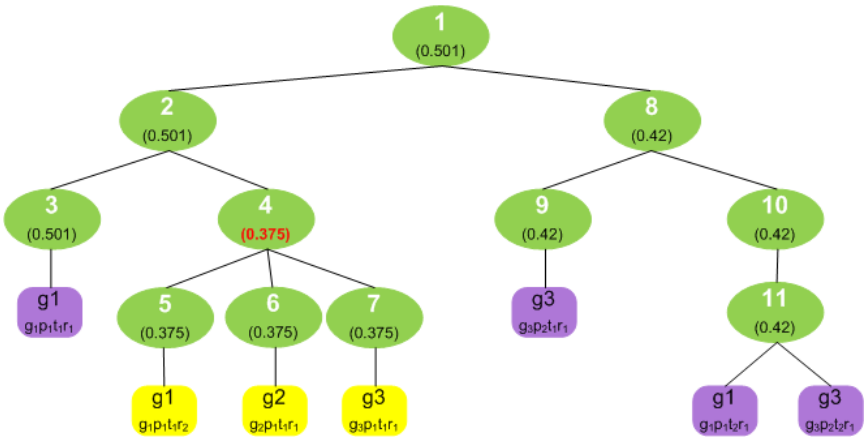


Figura 7.4: Representación del espacio de búsqueda para la aproximación heurística.

En la figura 7.4 se encuentra representado el espacio de búsqueda para la aproximación heurística sobre el ejemplo ficticio representado en la figura 7.1. En ésta se pueden observar un espacio con 11 nodos diferentes. Cada uno de ellos tiene representado el índice del nodo en cuestión y su valor representante entre paréntesis. Estos valores fueron obtenidos en base a la representación más cercana de los genes de entrada $A = \{g_1, g_2, g_3\}$ a cada uno de ellos. Por ejemplo, para el nodo 2, las representaciones seleccionadas son $\mathcal{T}_2(g_1) = \{g_1p_1t_1r_1\}$, $\mathcal{T}_2(g_2) = \{g_2p_1t_1r_1\}$ y $\mathcal{T}_2(g_3) = \{g_3p_1t_1r_1\}$. Nótese que para el g_1 es seleccionada entre tres candidatos, entre uno para g_2 y tres para g_3:

$$\mathcal{T}_2(g_i)\begin{cases} g_1 & \begin{cases} \mathcal{R}(\boldsymbol{g_1p_1t_1r_1}, \Delta_2) = \frac{2}{2+2} = 0{,}5 \\ \mathcal{R}(g_1p_1t_1r_2, \Delta_2) = \frac{3}{3+2} = 0{,}6 \\ \mathcal{R}(g_1p_1t_2r_1, \Delta_2) = \frac{6}{5+2} = 0{,}86 \end{cases} \\ g_2 & \left\{ \mathcal{R}(\boldsymbol{g_2p_1t_1r_1}, \Delta_2) = \frac{3}{3+2} = 0{,}6 \right. \\ g_3 & \begin{cases} \mathcal{R}(\boldsymbol{g_3p_1t_1r_1}, \Delta_2) = \frac{3}{3+2} = 0{,}5 \\ \mathcal{R}(g_3p_2t_1r_1, \Delta_2) = \frac{4}{3+2} = 0{,}8 \\ \mathcal{R}(g_3p_2t_2r_1, \Delta_2) = \frac{5}{4+2} = 0{,}83 \end{cases} \end{cases} \tag{7.12}$$

A partir de estas representaciones es posible calcular el valor representante del

nodo Δ_2:

$$\mathcal{S}_2(A) = \mathcal{S}(\{g_1p_1t_1r_1, g_2p_1t_1r_1, g_3p_1t_1r_1\}) = \frac{\frac{4}{3+4} + \frac{4}{3+4} + \frac{3}{4+4}}{3} = 0{,}501$$

El resto de valores, mostrados entre paréntesis en la figura 7.4, son calculados de igual forma para el resto de nodos. Entre ellos, es destacado en rojo el valor mínimo encontrado, el cuál será el resultado de la medida GFD. Así mismo, las representaciones usadas para obtener tal valor son destacadas en amarillo. Nótese como el valor mínimo se encuentra en los nodos 4, 5, 6 y 7; siendo algo común para esta aproximación.

En resumen, la aproximación heurística se basa en el recorrido de todos los nodos del GO–tree en vez de explorar todas las posibles combinaciones.

7.4.3. Un ejemplo real: ABC transporter

En este subsección será descrito la aplicación de la aproximación heurística a un ejemplo real. Concretamente, será empleada para medir la similitud, en la ontología *Molecular Function*, al conjunto de genes que forma el pathway ABC transporter y que fue previamente evaluado por el enfoque exhaustivo en 7.3.4.

Recuérdese que el conjunto de datos está compuesto por los genes YDR011W, YMR301C y YOR153W. Cada uno de ellos contiene 18, 10 y 18 representaciones asociadas a 6, 5 y 6 diferentes términos GO, respectivamente. Esto genera un espacio de búsqueda de $10^{3,5897}$, resultado del número de combinaciones de representaciones posibles. Tras explorar exhaustivamente este espacio, el término más cohesivo elegido es GO:0042626, el cual está asociado directamente con, al menos, una representación de cada gen.

En el caso de la aproximación heurística, el espacio de búsqueda es bastante menor. El GO–tree, representado en las figuras 7.2 y 7.3, contiene sólo 46 nodos, lo que implica un decremento importante en el espacio.

Siendo justos, el coste de la exploración de cada individuo del espacio de búsqueda exhaustivo no es equivalente al del heurístico. En el caso de la heurística, para cada individuo (nodo del árbol) se debe seleccionar la representación más cercana de cada gen, empleando la función $\mathcal{R}$ (ver ecuación 7.4), y después calcular su similitud según $\mathcal{S}$ (ecuación 7.6), la cual está basada, a su vez, en la función $\mathcal{R}$. Mientras que para el exhaustivo se aplica directamente la ecuación $\mathcal{S}$. Así, para el ejemplo que nos ocupa, por cada individuo del espacio heurístico, se realizan 40 $(18+12+18)$ cálculos de $\mathcal{R}$ para seleccionar la representación más cercana de cada gen al nodo; y $3 = \binom{3}{2}$ cómputos de $\mathcal{R}$ para el cálculo de $\mathcal{S}$. Esto hace un total de

43 cálculos de $\mathcal{R}$ por cada individuo del espacio heurístico, y 3 del exhaustivo.

Esta diferencia, aunque parece importante, resulta insignificante debido al tamaño de los espacios de búsquedas; $10^{3,5897}$ y 46, para exhaustivo y heurístico, respectivamente. Así, $\mathcal{R}$ es calculada un total de 11663 ($10^{3,5897} \times 3$) y 1978 (43×46) para ambas aproximaciones. Nótese que estos espacios han sido generados a partir de un conjunto de entrada con tres genes, lo que hace plausible la necesidad de la heurística y la eficiencia de ésta (ver sección 8 para una comparación detallada entre ambas aproximaciones).

Para el ejemplo que nos ocupa, existe más de un individuo en el espacio exhaustia con valor representante mínimo (0,0455). Concretamente, se tratan de los nodos GO:0042623, GO:0043492 y GO:0042626, recalcadas en rojo en las figuras 7.2 y 7.3, y cuyas representaciones seleccionadas, remarcadas en amarillo, son idénticas.

Así, la aproximación heurística selecciona las mismas representaciones y genera el mismo valor que la aproximación exhaustiva reduciendo notablemente el espacio de búsqueda a explorar.

7.5. GoGram : Visualización ontológica

7.5.1. Descripción

La validación y comparación de técnicas de aprendizaje automático es un concepto ampliamente usado y estudiado en la literatura. Éste resulta imprescindible para demostrar la validez de un método ante otros o la realización de un ranking de técnicas por su comportamiento ante unos datos concretos.

Como se ha detallado durante este documento, la validación de técnicas de análisis de micorarrays puede llevarse a cabo usando los propios datos usados o mediante datos externos. Es ésta última, usando datos biológicos conocidos y contrastados, la que en el mundo de la bioinformática está siendo usada con tal fin [86].

Dentro de la validación a partir de bases de datos biomédicas contrastadas, las medidas de similitud son las más demandadas en la actualidad [240]. Esta vertiente, aunque puede medirse la similitud funcionalidad de un conjunto de genes, carece de una representación gráfica que informe sobre la evaluación de diferentes conjuntos de datos en una única figura atendiendo a diferentes conceptos biológicos.

Con el objetivo de resolver tal carencia, se propone una descriptiva representación, denominada GoGram , para interpretar gráficamente la evaluación obtenida para los tres puntos de vistas biológicos, u ontologías, en las que se subdivide Gene Ontology (ver 5.2.1). Esta representación ofrece una visión conjunta de la similitud

funcional en cada par de ontologías de las tres simultáneamente. De esta manera, se hace posible la comparación de diferentes conjuntos de genes.

Un GoGram es una representación gráfica $3D$ donde cada dimensión denota la similitud funcional obtenida en uno de los tres dominios diferentes de GO. En la figura 7.5, a modo de ejemplo, se muestra el GoGram resultante de representar a único conjunto de genes. En este caso se ha supuesto que los valores de similitud generados son 0,2 en BP, 0,5 en CC y 0,8 en MF.

Como medida de similitud funcional es elegido el método propuesto en esta tesis, Gfd. Tal selección ha sido llevada a cabo atendiendo a la comparación entre medidas de similitud existentes realizada a lo largo del apartado 9. No obstante, cualquier otra que genere valores para las tres ontologías de GO sería válida.

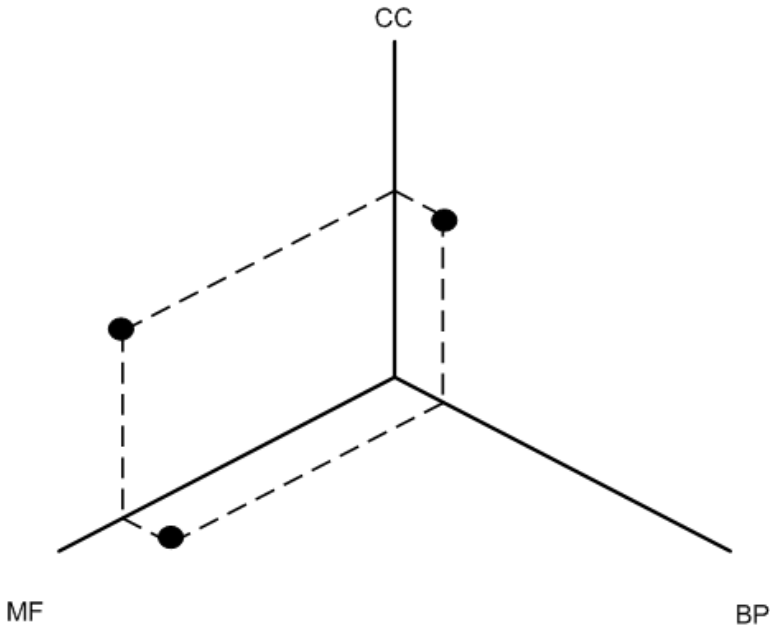


Figura 7.5: Ejemplo de un GoGram para una única entrada.

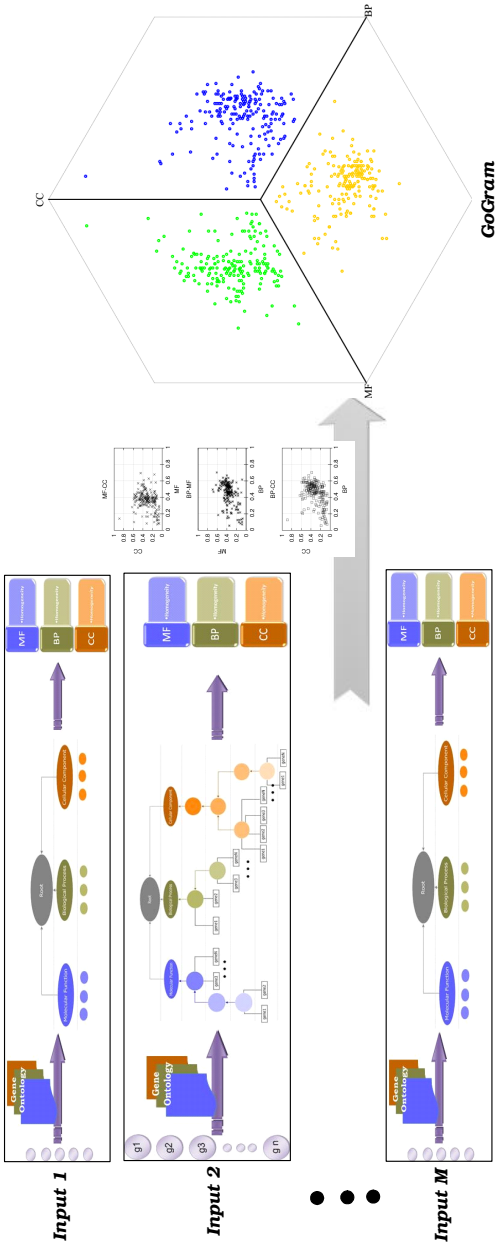


Figura 7.6: Proceso global para la generación de un GoGram . Primeramente, es medida la similitud funcional de cada entrada, obteniendo tres valores diferentes. Posteriormente, cada resultado es representado en el GoGram .

La figura 7.6 muestra el proceso completo para generar un GoGram a partir de M conjuntos de datos diferentes y usando Gfd. El gráfico muestra la evaluación de cada conjunto de genes por un punto en los tres posibles planos. Así, la similitud obtenida para un conjunto de entrada concreto es representado por tres puntos, donde cada punto es localizado en uno de los planos creados por las ontologías CC–BP, CC–MF y MF–BP. De esta forma, se ofrece la similitud de un conjunto de genes, desde tres puntos de vistas biológicos, de una manera única. Recuérdese que Gfd asigna valores cercanos a 0 si el conjunto evaluado presenta un comportamiento biológico similar y cercano a 1 en caso contrario. Así, aquellos puntos cercanos al origen $(0, 0, 0)$ representan entradas con una alta similitud biológica, mientras que el máximo valor posible en una dimensión es 1.

La potencia de un GoGram no está sólo limitada al estudio particular de un conjunto de datos sino que además permite la comparación de varios grupos de genes de un único gráfico. Por ejemplo, en la figura 7.6, el GoGram es usado para mostrar la evaluación de M conjuntos de entrada. Por ello, cada plano del GoGram presenta M puntos.

7.5.2. Transformaciones espaciales

Un GoGram es una representación gráfica en $3D$ sobre unos ejes isométricos, donde cada eje representa la similitud calculada sobre las ontologías Biological Process (BP), Molecular Function (MF) y Cellular Component (CC). Este tipo de ejes tiene la particularidad de formar tres ángulos iguales, de 120°.

Aunque la representación simbolice un espacio tridimensional ($BP-MF-CC$), éste ha sido generado en un espacio bidimensional ($X - Y$), lo que ha implicado una transformación de $2D$ a $3D$ (ver figura 7.7). Tras un estudio geométrico, las transformación llevadas a cabo son:

Plano I (BP-CC)

$$x = BP \times \cos\left(\frac{\Pi}{6}\right)$$
$$y = CC - BP \sin\left(\frac{\Pi}{6}\right)$$

Plano II (MF-CC)

$$x = (-1) \times MF \times \cos\left(\frac{\Pi}{6}\right)$$
$$y = CC - MF \sin\left(\frac{\Pi}{6}\right)$$

Plano III (MF-BP)

Si $MF > BP$:

$$x = (BP - MF) \times \cos\left(\frac{\Pi}{6}\right)$$
$$y = (-1) \times |BP - MF| \times \sin\left(\frac{\Pi}{6}\right) + BP$$

| otros:

$$x = (BP - MF) \times \cos\left(\frac{\Pi}{6}\right)$$
$$y = |BP - MF| \times \sin\left(\frac{\Pi}{6}\right) + MF$$

7.6. Resumen y conclusiones

En este capítulo se ha presentado una metodología, denominada Gfd, para el cálculo de la similitud funcional de un conjunto de genes. La medida, basada en

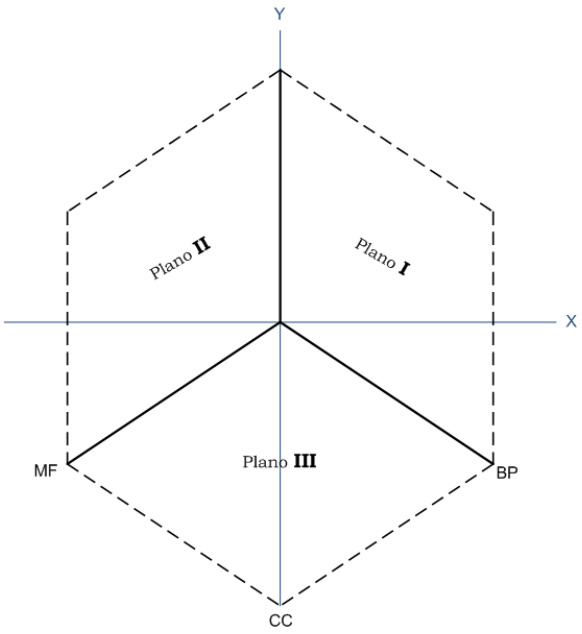


Figura 7.7: Transformación de $2D$ a $3D$ para la representación de un GoGram .

Gene Ontology, selecciona la funcional más cohesiva (común y específica) del conjunto de entrada para cada ontología. Los valores generados en cada ontología están en el rango $(0, 1)$, donde 0 indica comportamiento similar y 1 disimilar. Posteriormente, se presenta una aproximación heurística para solventar el coste computacional de la selección de la funcionalidad más cohesiva. La aproximación, basada en Diagramas de Voronoi, reduce significativamente el coste computacional sin que afecte a la calidad de los resultaos. Por último, es expuesta una poderosa representación, GoGram , para interpretar gráficamente la evaluación obtenida para los tres puntos de vistas biológico u ontologías de GO.

Parte IV

Resultados

Capítulo 8

Aproximación exhaustiva vs heurística

Saber y saberlo demostrar es valer dos veces.
Baltasar Gracián.

El objetivo principal de esta parte de la investigación consiste en analizar el comportamiento de la aproximación heurística de la medida Gfd bajo diversas circunstancias y compararla con la exhaustiva. Para ello, se han realizado dos tipos de experimentaciones. En primer lugar, se ha diseñado un algoritmo de generación de árboles aleatorios a partir de los cuales analizar la eficacia de la aproximación heurística. En segunda lugar, se realiza un estudio de su eficacia, centrándonos para ello en los espacios de búsqueda empleados para evaluar conjuntos de datos reales.

8.1. Comparación entre los valores de similitud: Eficacia

En esta sección se realiza una comparación entre el método exhaustivo y el heurístico para el cálculo de Gfd. Para ello, se ha tenido en cuenta el valor obtenido por ambas aproximaciones al evaluar diferentes árboles representantes de un GO–tree. Estos árboles han sido generados de forma totalmente aleatoria, cuya descripción es detallada seguidamente. Posteriormente, son expuestos los resultados de tal comparación.

8.1.1. Generación de árboles aleatorios

Un árbol aleatorio estará compuesto por al menos un nodo, que denominaremos raíz. A este nodo raíz se le añadirán hijos de una forma aleatoria cuyo pseudocódigo está recogido en algoritmo 8.1. En él se especifican cuatro parámetros de entrada: *n*, que simboliza al nodo al que estamos insertando hijos de forma aleatoria; y tres

valores que simbolizan diferentes propiedades del árbol; profundidad máxima del árbol (p), número de hijos por nodo (h) y número de genes usados (g).

En primer lugar se calcula la probabilidad de que el nodo n tenga hijos ($P(\mathcal{H}(n))$). Esta probabilidad viene determinada por la relación entre la profundidad del nodo n y la profundidad máximo del árbol (p): $P(\mathcal{H}(n)) = \frac{depth(n)}{p}$.

La probabilidad para el nodo n es usada como umbral para determinar si éste tendrá, o no, nodos hijos. Con tal fin, se genera un número aleatorio 0 y 1. Si la probabilidad es mayor o igual al valor generado implicará que el nodo n contendrá hijos. El número de hijos a insertar, superior a 0 y menos o igual que h, sería determinado aleatoriamente. Posteriormente, cada uno de los hijos es tratado por separado y de forma recursiva para determinar si, a su vez, poseerán nodos hijos.

En caso de que el nodo n no posea hijos, será insertado una representación génica. Para ello, se seleccionará un gen de forma aleatoria atendiendo al número máximo de genes g. Nótese que cabe la posibilidad de que uno o varios genes no hayan sido escogidos como nodo hoja en ninguna parte del árbol, entendiéndose en ese caso que tales genes no forman parte del conjunto de entrada.

Algoritmo 8.1 Inserción aleatoria de nodos

```
INPUT n: nodo del árbol a generar
  p: profundidad máxima del árbol
  h: número de hijos máximo por nodo
  g: número de genes máximo a insertar en el árbol
begin
  if P(H(n)) ≥ random(1) then
    for i := 1 → random(h − 1) + 1 do
      N.addChild(n_i)
      insertaHijos(n_i)
    end for
  else
    gen:=random(g-1)+1
    N.addChild(gen)
  end if
end
```

8.1.2. Resultado de la comparación

Para generar diferentes muestras de árboles, los valores de codificación del árbol fueron variados. Concretamente, el número máximo de genes (g) fue variado desde 3 hasta 10, mientras que el número máximo de hijos por nodo (h) tomó los valores 3 y 4. Por otro lado, la profundidad máxima, atendiendo a la profundidad media (6,8) y su desviación (±2,4) en GO, fue fijada a 9. Además, para cada una de las configuraciones posibles, se generaron 100 árboles diferentes. Así, 1600 árboles diferentes fueron usados para analizar la aproximación con y sin heurística.

Tabla 8.1: Error relativo cometido por la aproximación heurística para las diferentes configuraciones de árboles aleatorios. La columna h indica el número máximo de hijos por nodo; g el máximo de genes; Dif. la cantidad de evaluaciones diferentes obtenidas para la configuración dada; y $\overline{E_r}$ el Error relativo medio para los 100 experimentos con tal configuración.

h	g	Dif.	$\overline{E_r}$
3	3	1	0,00192
	4	1	0,00143
	5	1	0,00048
	6	3	0,00020
	8	3	0,00020
	9	5	0,00090
	10	2	0,00020
4	3	1	0,00002
	5	1	0,00003
	6	1	0,00018
	7	9	0,00100
	8	7	0,00180
	9	5	0,00030

Nótese que valores superiores a los seleccionados no ha sido posibles evaluarlos por la aproximación exhaustiva debido a su coste computacional.

En la tabla 8.1 se muestran los resultados obtenidos, donde las columnas h y g representan el número máximo de hijos por nodo y de genes, respectivamente; la columna Dif el número de valoraciones diferentes obtenidas por la aproximación heurística; mientras que en la última columna se encuentras los Errores relativos medios para los 100 experimentos generados según la configuración $h - g$ correspondiente. Así, por ejemplo, para $h = 3$ y $g = 3$, sólo se ha obtenido una única diferencia en los 100 experimentos realizados con tal configuración, generando un error relativo medio de 0,00192. Nótese que en la tabla sólo se representan aquellas configuraciones que presentan al menos una experimentación con valores diferentes para ambas aproximaciones, siendo, por tanto, idénticas las aproximaciones en tales configuraciones no representadas.

En resumen vemos que el uso de la heurística produces pequeñas diferencias en tan sólo el 2,5 % de los casos (40 árboles), y que la media del error relativo es de 0,00054. De esta forma podríamos afirmar que el uso del método heurístico no afecta a la calidad de los resultados obtenidos.

8.2. Comparación entre los espacios de búsqueda: Eficiencia

En esta sección se compararán los espacios de búsqueda que emplean las aproximaciones heurística y exhaustiva, así como el número de computaciones de $\mathcal{R}$ (sección 7.3.2). Para ello se usarán diferentes conjuntos de datos reales que se describen a continuación.

8.2.1. Pathways de KEGG como múltiples conjuntos de entrada

Los conjuntos de datos usados para la comparación de espacios de búsqueda son extraídos del conocimiento almacenado en KEGG (ver 5.2.2). KEGG [173] es una base de datos de sistemas biológicos que integra información de funciones genómica, química y sistémica. Esta base de datos ofrece información genómica de multitud de organismos, de entre el que hemos elegido *Saccharomyces Cerevisiae* (SCE) para este estudio. Para este organismo, son seleccionados todos los pathways y son tratados como una entrada. Así, los genes que intervienes en cada pathway son elegidos como conjunto de entrada. Originalmente se consideraron 100 conjuntos de genes, pero finalmente fueron reducidos a 98 tras eliminar aquéllos que sólo contienen un gen (Nótese que la versión de KEGG y GO usada es de Marzo del 2011).

8.2.2. Análisis computacional

Para explorar el efecto de la heurística en el coste computacional, es presentada la tabla 8.2. Ésta muestra información sobre espacios de búsqueda usados por las aproximaciones exhaustiva y heurística para evaluar cada conjunto para la ontología *Biological Process*. Concretamente se muestran los diez conjuntos con mayor número de genes conocidos en GO (de los 98), junto con información relevante sobre el número de anotaciones y representaciones. Asimismo, contiene información sobre el número de individuos que poseen cada espacio de búsqueda y el número de evaluaciones de $\mathcal{R}$ (ver apartado 7.3.2) que son calculados por ambas aproximaciones. Recuérdese que el coste de computar cada individuo es diferente para ambas aproximaciones y que vendría medido en número de ecuaciones $\mathcal{R}$ calculadas (ver 7.4.3). Además, para la aproximación heurística es mostrado su tiempo de cálculo. Por último, y para una información completa para todas las ontologías y grupos de genes, se remite al lector al apéndice A.

Analizando el bien conocido pathway del ciclo celular (*sce:04111*) podemos ver que para analizar este conjunto, que contiene 125 genes que todos son conocidos (columna “Syn”), es necesario considerar sus 426 anotaciones. Esas anotacio-

Tabla 8.2: Análisis Computacional

Pathway	**Genes**	**Syn**	**Anot**	**Rep**	*Exhaustivo* (log_{10})		*Heurístico* (log_{10})		
					Individuos	$\mathcal{R}$'s	Individuos	$\mathcal{R}$'s	Tiempo
sce01100	645	645	2960	4132	467,69	473,01	3,04	8,37	26,33
sce01110	235	235	1152	1444	170,67	175,11	2,67	7,13	1,64
sce03008	159	157	253	253	25,09	29,18	1,36	5,46	0,14
sce04113	127	127	514	821	80,27	84,17	2,39	6,34	0,44
sce04111	125	125	426	664	65,66	69,55	2,36	6,29	0,47
sce00230	93	93	474	615	68,97	72,60	2,32	6,01	0,24
sce03010	93	91	256	413	43,12	46,74	2,19	5,84	0,20
sce03013	83	83	251	378	40,22	43,76	2,24	5,82	0,22
sce04141	79	79	250	369	42,39	45,88	2,25	5,79	0,22
sce00190	76	76	284	710	66,47	69,92	2,27	5,82	0,25
sce03040	76	76	219	307	30,84	34,29	1,89	5,39	0,16

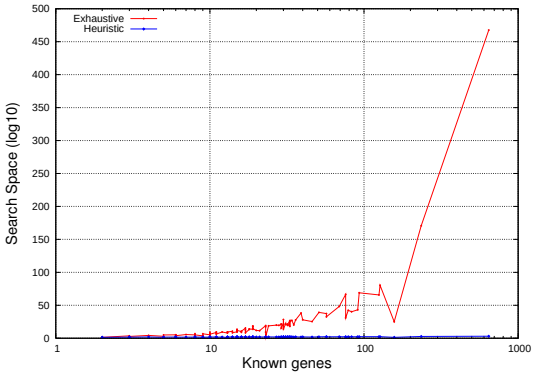
nes originan 664 representaciones, que producen $10^{65,66}$ combinaciones ó individuos en el espacio exhaustivo. Este número de individuos es de $10^{2,36}$ aplicando la heurística, lo que significa una reducción bastante notoria. Así mismo, el número de $\mathcal{R}$ a calcular sería $10^{69,55}$ y $10^{6,29}$, respectivamente. Por último, el tiempo empleado por la aproximación heurística es de 0,47 segundos, mientras que el tiempo estimado de la exhaustiva sería de $10^{62,94}$ segundos.

Con el fin de realizar un estudio más completo del coste computacional de ambas aproximaciones es presentada la figura 8.1. En ella se encuentras 6 figuras diferentes correspondientes a la relación del número de individuos a evaluar por número de genes (parte izquierda); y el número de ecuaciones $\mathcal{R}$ a computar (parte derecha). Este estudio es llevado a cabo para las tres ontologías, generando las seis gráficas mostradas. Nótese que en todas ellas el eje x, que indica el tamaño del conjunto de entrada, es mostrado en una escala logarítmica mientras que el eje y presentan los logaritmos de los tamaños de los espacios o el número de $\mathcal{R}$'s.

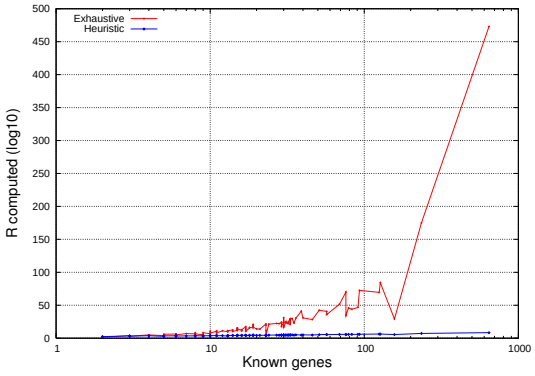
El comportamiento que se extrae de estas figuras es idéntico para todas las ontologías: el tamaño del espacio exhaustivo es muy superior al del espacio heurístico. Además, el espacio exhaustivo es incrementado de forma exponencial conforme el número de genes de entrada crece, mientras que el heurística no sufre apenas modificación. Igualmente, podemos realizar la misma afirmación para el número de similitudes entre dos representaciones a computar ($\mathcal{R}$). De forma que, quedaría probado que la aproximación heurística es capaz de reducir notablemente el espacio de búsqueda exhaustivo.

Comparando las representaciones del tamaño del espacio de búsqueda con las del número de $\mathcal{R}$ a calcular, podemos seguir afirmando el mismo comportamiento para todas las ontologías: la diferencia del coste de evaluar cada individuo de los espacios de búsqueda resulta insignificante ante la diferencia en sus tamaños.

Molecular Function

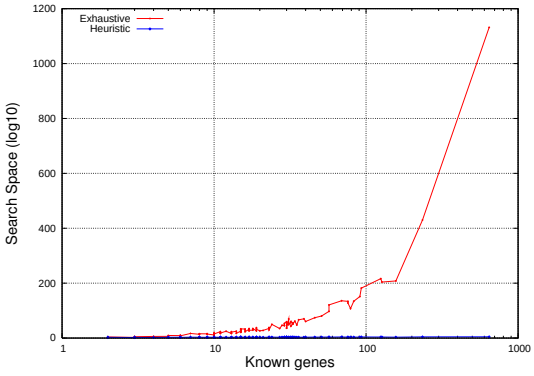


(a) *Tamaño Espacios de Búsqueda (MF)*

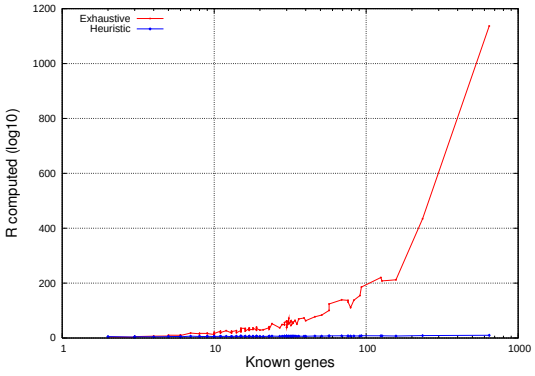


(b) *Número de $\mathcal{R}$ computados (MF)*

Biological Process

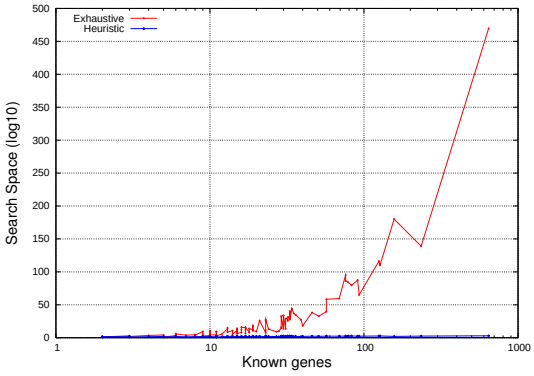


(c) *Tamaño Espacios de Búsqueda (BP)*

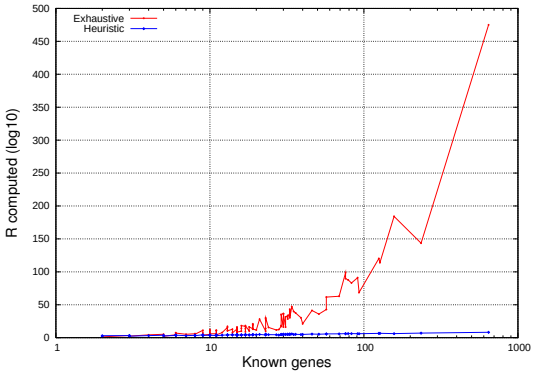


(d) *Número de $\mathcal{R}$ computados (BP)*

Cellular Component



(e) *Tamaño Espacios de Búsqueda (CC)*



(f) *Número de $\mathcal{R}$ computados (CC)*

Figura 8.1: Espacios de búsqueda heurísticos y exhaustivos.

8.3. Conclusiones

En este capítulo se ha llevado a cabo un análisis del comportamiento de la aproximación heurística frente a la exhaustiva para el cálculo de GFD. Primeramente, y con el objetivo de estudiar la eficacia de la heurística, han sido generados 1600 árboles con topología diferente y evaluados por ambas aproximaciones, obteniendo un error relativo medio de 0,00192. En segundo lugar, los espacios de búsqueda fueron comparadas, así como el número de cómputos de $\mathcal{R}$ llevados a cabo. Los resultados mostraron una reducción drástica del espacio de búsqueda y, como se afirmó en la sección donde se proponía la heurística (7.4), la diferencia del coste de evaluación de los individuos de cada espacio resulta insignificante frente a su diferencia en tamaño.

En resumen, podemos afirmar que el algoritmo heurístico aporta una reducción considerable del coste computacional sin afectar a la calidad de los resultados.

Capítulo 9

Experimentación con GFD

Duda siempre de ti mismo, hasta que los datos no dejen lugar a dudas.

LOUIS PASTEUR.

En este apartado se analizará el comportamiento de la metodología propuesta para evaluar la similitud funcional de un conjunto de genes. Para el cálculo de GFD se ha usado una implementación multihilo en java de la aproximación heurística, ya que como es demostrado en la sección 8.1, ésta genera resultados equivalentes reduciendo drásticamente el coste computacional

Primeramente, con el fin de mostrar la validez de GFD en términos biológicos, se realiza un estudio en 9.1 sobre la fase S del Cell Cycle por ser un proceso extensamente estudiado en la literatura . Posteriormente, en la sección 9.2 es comparado el comportamiento de GFD con las medidas de similitud más representativas mediante un análisis ROC. Mientras que un análisis de robustez de GFD es llevado a cabo en las sección 9.3.

9.1. Histone Cluster

En el trabajo [279], Spellman et al. agruparon los genes de la levadura envueltos en el proceso del ciclo celular (*cell cycle*) en ocho grupos usando *Eisen Clustering* [110] y llevaron a cabo un estudio meticuloso de cada uno de los clusters obtenidos. ***Histone cluster*** es uno de ellos, y agrupa a nueve genes (*YPL127C*, *YBL002W*, *YBL003C*, *YBR009C*, *YNL030W*, *YNL031C*, *YDR224C*, *YBR010W*, *YDR225W*) que presetan su máxima expresión durante la fase de síntesis (*S phase*), en donde ocurre la replicación de ADN.

Basándonos en GO y estudiando independientemente cada gene del *Histone cluster*, podemos observar que forman parte de diferentes procesos biológicos, pre-

sentando más de una funcionalidad. Esas funcionalidades génicas, resumidas en la tabla 9.1, están directamente relacionadas con la replicación del ADN.

La funcionalidad más común y específica es "Nucleosome assembly", identificada por el término GO:0006334. Este proceso biológico es asociado a "histone chaperone" (encontrado con la herramienta web AmiGO [67]), el cual está directamente relacionado con la fase S, ya que representa la agregación, disposición y unión de un conjunto de nucleosomas[1]. Consecuentemente, cualquier medida que trate de evaluar la similitud de estos genes debería usar tal funcionalidad, ya que ellos fueron agrupados en el mismo grupo por desarrollar, a la vez, esa funcionalidad.

Tabla 9.1: Descripción de la funcionalidad, a partir de la ontología *Biological Process*, de los genes envueltos en *Histone Cluster*.

Genes	Anotaciones	Descripción
YPL127C	GO:0006334	**Nucleosome assembly**
	GO:0006355	Regulation of transcription, DNA-dependent
	GO:0045910	Negative regulation of DNA recombination
YBL002W	GO:0006333	Chromatin assembly or disassembly
	GO:0006334	**Nucleosome assembly**
YBL003C	GO:0006281	DNA repair
	GO:0006333	Chromatin assembly or disassembly
	GO:0006334	**Nucleosome assembly**
	GO:0006974	response to DNA damage stimulus
YBR009C	GO:0006333	Chromatin assembly or disassembly
	GO:0006334	**Nucleosome assembly**
	GO:0034729	Histone H3-K79 methylation
YNL030W	GO:0006333	Chromatin assembly or disassembly
	GO:0006334	**Nucleosome assembly**
	GO:0034729	Histone H3-K79 methylation
YNL031C	GO:0006281	DNA repair
	GO:0006333	Chromatin assembly or disassembly
	GO:0006334	**Nucleosome assembly**
	GO:0006412	translation
	GO:0006417	regulation of translation
	GO:0006974	response to DNA damage stimulus
YDR224C	GO:0006333	Chromatin assembly or disassembly
	GO:0006334	**Nucleosome assembly**
	GO:0006301	postreplication repair
	GO:0045816	Negative regulation of transposition from RNA polymerase II promotor, global
YBR010W	GO:0006333	Chromatin assembly or disassembly
	GO:0006334	**Nucleosome assembly**
	GO:0006281	DNA repair
	GO:0006412	translation
	GO:0006417	regulation of translation
	GO:0006974	response to DNA damage stimulus
	GO:0010526	negative regulation of transposition, RNA mediated
YDR225W	GO:0006281	DNA repair
	GO:0006333	Chromatin assembly or disassembly
	GO:0006334	**Nucleosome assembly**
	GO:0006974	response to DNA damage stimulus
	GO:0045816	negative regulation of transposition from RNA polymerase II promotor, global

En la figura 9.1 se encuentra representada la parte del GO–tree en la que está

[1]El nucleosoma es una estructura que constituye la unidad fundamental y esencial de la cromatina, que es la forma de organización del ADN en los eucariotas.

localizada la representación génica seleccionada por GFD para medir la disimilitud del *Histone cluster*. Tales representaciones son la misma para todos los genes del *Histone*, "Nucleosome assembly" (GO:0006334), que es asociada con el montaje del ADN, un proceso crítico de la fase S. De esta manera, se concluye que la funcionalidad génica seleccionada para cada gene es la más adecuada según el conjunto de entrada, como se mencionó anteriormente. Además, nótese que la representación génica seleccionada no es la más específica, ya que los genes *YBR009C* y *YNL030W* presentan un comportamiento común en el nivel 11: 'Histone H3-K79 methylation" (GO:0034729) (ver tabla 9.1) en lugar del nivel 8 al que pertenece el término GO:0006334.

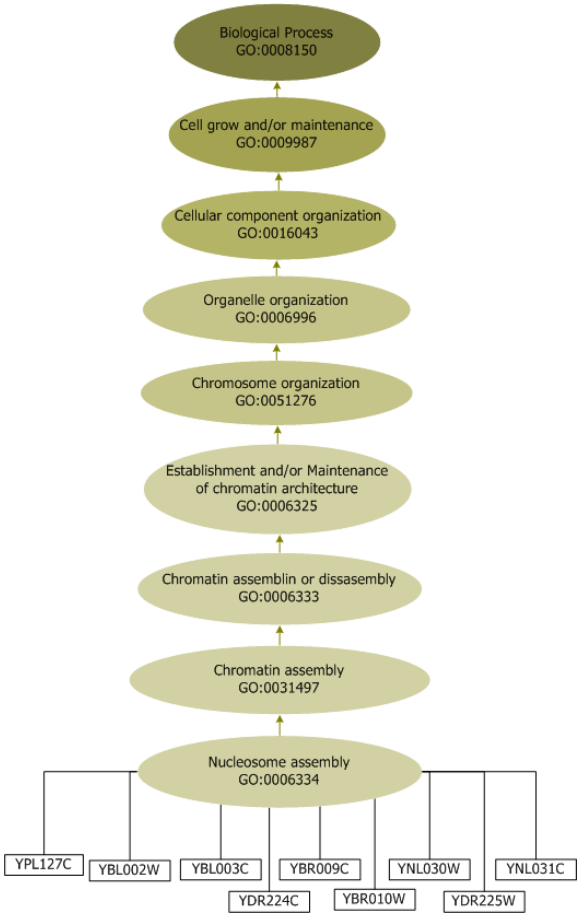


Figura 9.1: Información automáticamente recuperada de GO para aplicar GFD a Histone Cluster.

La disimilitud funcional encontrada por GFD para el *Histone cluster* es de 0,055. Este valor, generado a partir de una de las $|P| = 10^{18,2}$ representaciones posibles, caracteriza que los genes pertenecientes a tal cluster son muy similares

según su proceso biológico.

9.2. Comparación con otras medidas de similitud

9.2.1. Justificación de medidas seleccionadas

El comportamiento de la principal propuesta de esta tesis es testeado mediante la comparación con tres medidas de similitud diferentes.

En primer lugar es elegida una medida basada en información contenida, ya que, según Pesquita [239], este tipo de medidas son más adecuadas que las basadas en aristas. Como medida representante se ha elegido la similitud de Resnik [255]. Esta elección ha sido atendiendo a los estudios realizados por Guo, Sevilla, Wang et al. [136, 269, 301], donde afirmaban que esta medida es la que mejor comportamiento presenta en estudios de similitud de genes y niveles de expresión. Por otro lado, la medida híbrida de Wang [303] es también seleccionada ya que, según el estudio desarrollado por sus propios creadores, ésta es más consistentes que Resnik en estudios humanos. Por último, la medida de GS^2 [262] (ec. 5.5.3) ha sido empleada en este estudio por ser la primera medida que evalúa eficientemente un conjunto de genes en lugar de pares de genes o términos GO.

Recuérdese que las medidas de Resnik (ec. 5.5.1) y Wang (ec. 5.5.1) evalúan la similitud de dos términos GO, lo que hace necesaria una extrapolación de ellas. Tal extrapolación es realizada mediante la aplicación de la aproximación *best–match average* (BMA) (ver sección 5.5.2). Ésta es un técnica basada en pares de terminos GO, y que fue seleccionada por Xu et al. [317] como la aproximación que mejores resultados genera.

Por último, cabe destacar que las medidas de Resnik y Wang, para la evaluación de similitud de términos GO, fueron calculadas usando sus implementaciones en Bioconductor [323] usando lenguaje R. Mientras que el código en Python de GS^2 fue descargado de la página web referenciada en [262].

9.2.2. Conjunto de datos con y sin significatividad biológica

La comparación del comportamiento de las medidas de similitud ha sido llevado a cabo atendiendo a dos conjuntos de datos: conjuntos con y sin coherencia funcional.

Ambos conjuntos de datos fueron generados según la información almacenada en KEGG. Así, todos los pathways metabólicos de la levadura (SCE) fueron usados como ejemplos de conjuntos de genes con coherencia funcional. El conjunto de datos sin coherencia funcional fue generado con el mismo tamaño que los de coherencia funcional, pero seleccionando aleatoriamente los genes del mismo cluster.

Así, por cada pathway, tenemos dos cluster de genes de tamaño k: uno con genes involucrados en el mismo pathway y otros con genes generados aleatoriamente.

Concretamente, en Marzo del 2011, KEGG contaba con 100 pathways en el organismo SCE. Sobre estos, tras eliminar 2 pathways con tan sólo un gen, se generaron 98 cluster con significatividad biológica y otros 98 generados de forma pseudoaleatoria. Nótese que los cluster generados con coherencia funcional son los mismos que los usados en la experimentación recogida en la sección 8.2.

Debido a que el conjunto de entrada es del 03/2011, esta es la versión de GO usada para el desarrollo de este capítulo.

9.2.3. Análisis ROC

El análisis ROC ha sido ampliamente usado en la literatura [317] ya que permite calificar el comportamiento de clasificadores como una relación entre sensitividad o rango de ciertos positivos (*True Positive rate*, TPR) y rango de falsos positivos (*False Positive rate*, FPR). Además, el área bajo la curva ROC también ha sido usada, ya que provee información sobre el nivel de aleatoriedad de la aproximación.

En el ejemplo concreto que nos ocupa, el análisis ROC es realizado sobre las tres ontologías de GO. En particular, los métodos GFD, Resnik y Wang son comparados para las tres ontologías, mientras que GS^2 es sólo usado para la ontología *Biological Process* ya que ésta no provee valores para las otras dos ontologías. La curva ROC es pintada sobre los intervalos $[0, 1]$ con incrementos de 0,01, como es ilustrado en las figuras 9.2, 9.3 y 9.4 para las ontologías BP, CC y MF, respectivamente. En todas ellas, el área bajo la curva (AUC) es incluida entre paréntesis. Se remite al lector al apéndice B para una información completa sobre el valor de similitud generado por las medidas en cada ontología.

Las figuras 9.2, 9.3 y 9.4 muestran que GFD presenta un comportamiento similar y satisfactorio para las tres ontologías. Los métodos de Resnik y Wang actúan de forma diferente. Para la ontología *Biological Process*, sólo la propuesta de Wang se comporta peor de lo esperado, debido a su rango de falsos positivos. El AUC es superior al 0,90 para la mayoría de medidas, excepto para la de Wang, que parece ser aleatoria (por debajo de 0,5). Para la ontología *Molecular Function*, la aproximación propuesta es excelente, con un AUC de 0,98, el cual es mucho mayor que la de Resnik (0,65) o la de Wang (0,17). En el caso de la ontología *Cellular Component*, el comportamiento de las tres medidas en similar.

Aunque un proceso biológico no es equivalente a un pathway, estos conceptos están muy relacionados. Por ejemplo, el pathway *Cell cycle* (sce:04111) está directamente relacionado con el GO–term "mitotic cell cycle" según la información almacenada en KEGG. De este modo, los genes dentro del mismo pathway deben

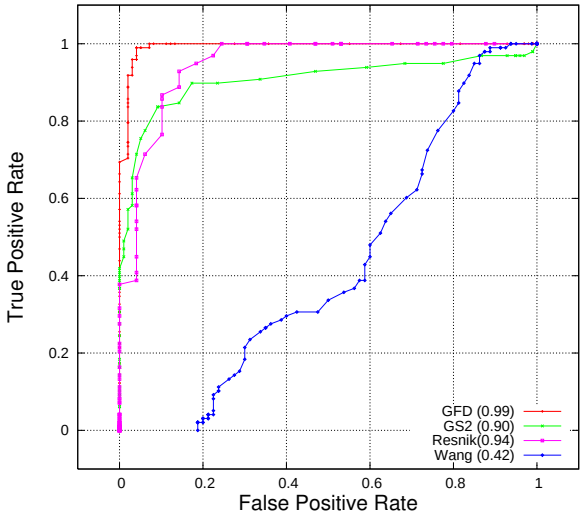


Figura 9.2: Análisis ROC para la ontología **Biological Process**.

ser similares para la ontología *Biological Process* (ver figura 9.2).

Sin embargo, estos genes no tiene que ser similares para todos los casos bajo la ontología *Cellular Component*, ya que pueden estar localizados en diferentes lugares de la célula para el mismo pathway. Por ejemplo, los genes del Ciclo Celular (*Cell Cycle*) relacionados con la transcripción (*transcription*) está localizados en el núcleo mientras que aquellos relacionados con la traducción (*translation*) están en el ribosoma. Por ello, los resultados obtenidos para esta ontología no son suficientemente consistentes para comparar el comportamiento de las diferentes enfoques (ver figura 9.3).

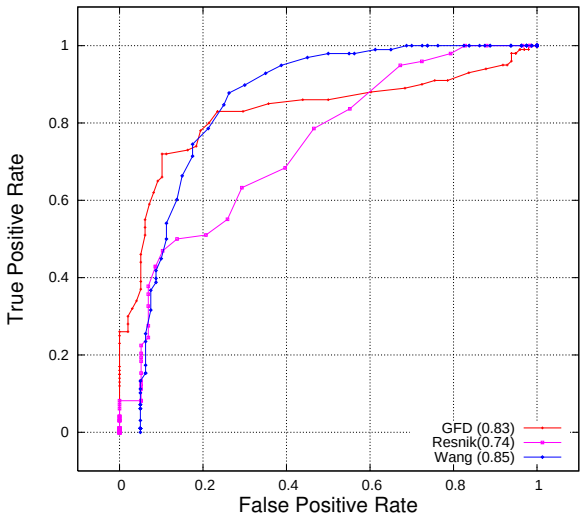


Figura 9.3: Análisis ROC para la ontología **Cellular Component**.

Finalmente, los genes en el mismo pathway también deben ser similares en la ontología *Molecular Function*. Esta ontología describe tipos de actividades, alguna de las cuales están presentes en una ruta metabólica describiendo el proceso. Esto es crucial para este estudio ya que la aproximación que se propone selecciona la funcionalidad más cohesiva entre los genes. En contraste, las aproximaciones de Wang y Resnik están basadas en BMA donde la misma funcionalidad no tiene que ser seleccionada necesariamente para calcular la similitud general del conjunto de genes. Esto causa un alto nivel de falsos positivos (FPR) (ver figura 9.4). GFD sólo usa un término GO para evaluar la similitud de un gen en relación al resto, mientras que las otras propuestas pueden seleccionar diferentes términos para medir la similitud de un gen con respecto a los otros genes. Esta es la principal razón para el pobre comportamiento de Resnik en la ontología *Molecular Function* en comparación con *Biological Process*.

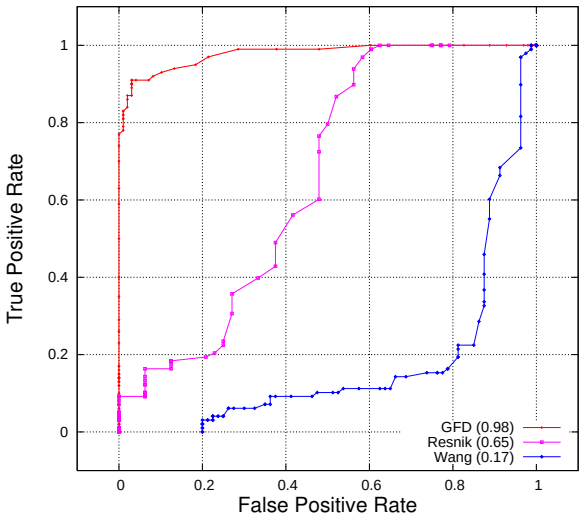


Figura 9.4: Análisis ROC para la ontología **Molecular Function**.

9.3. Análisis de robustez ante aleatoriedad

En esta sección se realiza un estudio de la robustez de GFD ante la aleatoriedad. Con tal fin se ha realizado una experimentación que muestra el comportamiento de GFD (ec. 7.3.3), GS^2 (ec. 5.5.3)y las aproximaciones de Wang et al. (ec. 5.5.1) y Resnik (ec. 5.5.1) et al. con respecto a la presencia de datos aleatorios.

Para ello se ha replicado la experimentación que llevaron a cabo Ruth et al. [262] para probar la robustez de GS^2. Esta experimentación se basaba en la evaluación de diferentes grupos de genes, partiendo de un conjunto con significatividad biológica y variando el porcentaje de genes aleatorios. Tal experimentación fue llevada a cabo para la ontología *Biological Process*. Esta ontología también será la única seleccionada ya que GS^2 sólo aporta valores para tal ontología y, como se explicó en la sección 9.2, los resultados de la ontología *Cellular Component* no son concluyente y el comportamiento de las aproximaciones de Wang y Resnik era penalizado en *Molecular Function*.

Para este caso concreto, se ha partido del pathway *Ribosome* de la levadura (usado previamente en la sección 9.2) que contiene 159 genes. De esos 159 genes, 157 son conocidos en GO, y ellos generan una similitud, según GFD, de 0,22 para *Biologica Process*. A partir de este conjunto, se realizaron diez estudios diferentes variando la aleatoriedad en incrementos de un 10 por ciento, generando un total de 100 conjuntos de datos. Por ejemplo, 10 % significa que fueron eliminado $15-16$ genes e introducidos otros $15-16$ de manera aleatoria. Para eliminar cualquier tipo de distorsión, se ha realizado este estudio 100 veces obteniendo como resultado la media de cada una de las configuraciones. Igualmente, para que el número de genes conocidos no varíe en cada estudio, los genes aleatorios fueron selecciondos del conjunto de genes de la levadura (*saccharomyces cerevisiae*) conocidos por GO.

En la figura 9.5 se representa el resultado obtenido, mostrando para la aproximación propuesta el valor $\overline{\text{GFD}} = 1 - \text{GFD}$. A partir de ésta podemos afirmar que el comportamiento (silueta) de GFD y GS^2 es muy similar. Cuanto mayor número de genes aleatorios sean introducidos, y el porcentaje incremente, la similitud decrece. El decrecimiento es mayor al comienzo, y casi ninguno después del 90 % de aleatoriedad. El comportamiento de la aproximación de Resnik es también similar, aunque la curva no tiene la misma forma. Esta parece más estable (en el sentido que la aleatoriedad no afecta al valor de la similitud más allá del 60 %), aunque este comportamiento no es apropiado. El método de Wang et al. [303] es sorprendente, ya que los valores de similitud incrementan cuando son introducidos datos aleatorios en el nivel de 100 %. Nótese que el análisis ROC (sección 9.2) para este método ya anunciada algo extraño en el rango de falsos positivos. En particular, este experimento fue repetido varias veces y los resultados siempre fueron similares.

En resumen, el análisis de aleatoriedad muestra que GFD y GS^2 presentan un

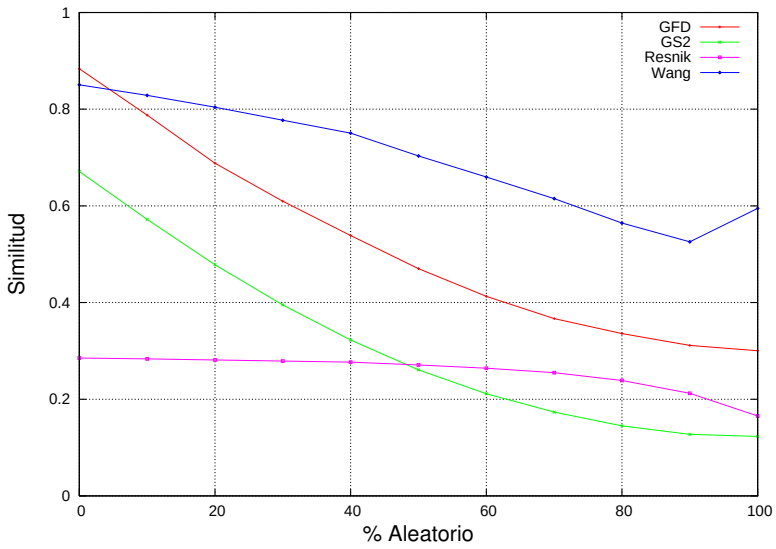


Figura 9.5: Análisis aleatorio (Ribosome).

comportamiento robusto con respecto a la aleatoriedad.

9.4. GoGRAM : Representación gráfica de la similitud funcional en SCE

La similitud funcionalidad de los 98 pathways asociados con el organismo SCE, usados en la sección anterior como conjuntos de datos con significatividad biológica, son representadios en la figura 9.6 aplicando el concepto de GoGRAM . De esta forma, cada uno de los planos del GoGRAM (CC–BP, CC–MF, MF–BP) contiene 98 puntos que representan la similitud de cada pathway de SCE bajo un par de puntos de vista biológicos. Adicionalmente, es insertado un hexágono no regular en el GoGRAM para identificar los límites de la evaluación aleatoria. Así, aquellos puntos que no están dentro del hexágono representarían entradas que han sido generadas de forma aleatoria.

Los límites aleatorios fueron obtenidos evaluando múltiples conjuntos de entrada sin coherencia funcional. Concretamente, para cada pathway, se generaron 100 entradas diferentes con el mismo número de genes pero seleccionados de forma aleatoria. El promedio del valor encontrado por GFD para los 1000 conjuntos en cada ontología fue seleccionado como el valor límite para tal dominio. Siendo igual a 0,69 para las ontologías BP y MF, y 0,45 en CC.

En el GoGRAM representado en la figura 9.6 podemos observar que en el plano MF–BP tan sólo existe un punto fuera del hexágono, mientras que existen algunos más para los planos que comparten la dimensión CC. Este comportamiento es si-

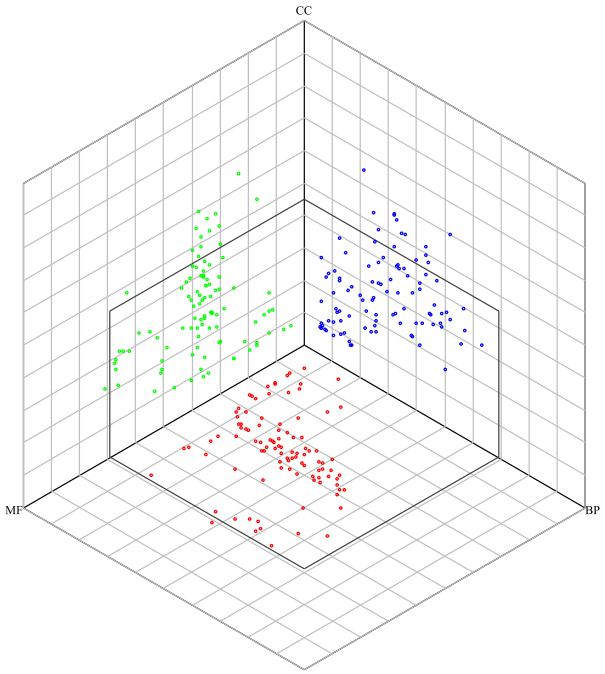


Figura 9.6: GoGram para pathways del organismo SCE

milar al descrito mediante el análisis ROC (sección 9.2). En la ontología BP el área encontrada era muy cercana a uno (0.99), lo que queda recogido gráficamente al encontrarse todos los puntos signficativamente por detrás del límite aleatorio. Para el dominio MF, tan sólo existe una entrada por encima del límite aleatorio. Esta entrada, correspondiente al pathway "*Protein Export*" (sce03060) y que denota al transporte de proteínas desde el citoplasma al exterior de la célula, fue evaluada por un valor (0,71) muy cercano al límite del ortógono. De esta forma podría explicarse el pequeño decremento en el AUC encontrado (0,98). Por último, es la ontología CC la que obtenía un AUC inferior al resto (0,83) lo que cabría explicarse por estar situados el 15 % de los puntos por encima del límite aleatorio.

Por consiguiente podemos afirmar que el GoGram recoge gráficamente el comportamiento que previamente fue analizado, donde el plano MF–BP recoge el mejor comportamiento. Recuérdese que éstos resultados son los esperados, ya que el conjunto de datos de entrada, pathways metabólicos, presentan conjuntos de genes con similitud en su funcionalidad o en los procesos biológicos en los que intervienen y no en su localización celular.

9.5. Conclusiones

Para evaluar la utilidad de la medida de similitud propuesta, GFD, se han realizados diferentes test con datos biológicamente significativos y pseudoaleatorios.

En primer lugar, se presenta un estudio riguroso sobre la fase S del ciclo celular con el objetivo de validar el sentido de la medida GFD. Concretamente, se analizaron los genes involucrados en el Histone Cluster generado por Spellman et al. [279] atendiendo a la información contenida en GO. Posteriormente, y con ayuda de GFD, fue medida la similitud funcional de tal conjunto de genes para la ontología BP. Así mismo, se examinó la funcionalidad génica escogida por GFD para realizar tal valoración, siendo la funcionalidad deseada según el análisis previo. La evaluación generó un valor de 0,055 indicando, según lo esperado, que estos genes son funcionalmente similares al compartir su funcionalidad en un proceso crítico de la fase S: el montaje del ADN.

Tras probar la validez de la propuesta, fue comparada con tres medidas de similitud existentes. Como primera cuestión se recoge la justificación de la elección de tales medidas. Fundamentalmente, las medidas elegidas son las que presentan mejores comportamientos según estudios realizados por terceros. En concreto, dos medidas basadas en similitud de términos GO, Resnik como representante de medidas basadas en información contenida y Wang de las híbridas, que fueron extrapoladas a similitud de grupos de genes según la técnica de combinación de pares *best match average*. Posteriormente, según la información recogida por KEGG para el organismo de la levadura, se generaron dos conjunto de entrada. Los conjuntos, con y sin significatividad biológica, fueron usados como entrada de las diferentes medidas de similitud. Los resultados obtenidos para las tres ontologías GO se examinaron mediante un análisis ROC con el fin de analizar el poder discriminatorio de la medida de disimilitud propuesta y su sensibilidad. Tal análisis reveló que la idea de seleccionar tan sólo una funcionalidad por cada gen, en vez de promediadas, para evaluar la similitud de un conjuntos de genes genera resultados superiores al resto de medidas, siendo especialmente interesante en estudios de grupos de genes que intervienen en diferente funciones biológicas.

En tercer lugar, se presentó un estudio de robustez de la medida propuesta. En esta ocasión el objetivo era analizar la robustez de GFD ante la inclusión de aleatoriedad en los datos. Para ello, su comportamiento se comparó al del resto de medidas de similitud seleccionadas anteriormente. En tal estudio, se seleccionaron los genes que intervienen en la ruta *Ribosome* como conjunto de genes sin aleatoriedad. Posteriormente, en incrementos de 10 %, es añada aleatoriedad en los datos, hasta llegar a un total de 100 %. Este proceso se aplicó 100 veces, y los diferentes cluster generados fueron evaluados por tales medidas. Finalmente, este estudio demostró que GFD presenta un comportamiento robusto ante la aleatoriedad.

En último lugar, se generó un GoGRAM para representar gráficamente las evaluaciones obtenidas por GFD al computar la similitud el conjunto de datos con coherencia biológica. Así mismo, se explicó la utilidad de tal propuesta gráfica basándonos en el análisis ROC realizado previamente. Demostrando así la potencia de esta representación para interpretar gráficamente los resultados obtenidos para las tres ontologías.

Parte V

Conclusiones

Capítulo 10

Conclusiones y Trabajos Futuros

Unos dicen lo que saben, y otros saben lo que dicen.
ANÓNIMO.

En principal objetivo de esta tesis ha sido la creación de una metodología para medir la similitud funcional de un conjunto de genes. La medida propuesta hace uso de la información biológica almacenada en Gene Ontology y de una organización en árbol de tal conocimiento.

La motivación de esta propuesta es solventar la importante limitación de la medidas de similitud actuales para ponderar el parecido funcional de un conjunto de genes atendiendo a una única funcionalidad. Tales medidas usan el conjunto completo de todas las funcionalidades de los genes bajo estudio, ponderando de igual forma todas sus funcionalidades e, incluso, usando diferentes funcionalidades de un mismo gen para medir su similitud con el resto. Esta limitación es especialmente importante al medir la similitud de genes que intervienen en diferentes funciones biológicas pero que fueron agrupados acorde a una de ellas.

Aunque el resultado final de esta tesis se sintetiza en una medida para calcular la similitud de un conjunto de genes basada en la funcionalidad más cohesiva, han sido varias las contribuciones de esta tesis importantes:

- El estudio del enriquecimiento funcional constituye una poderosa herramienta para la validación de la coherencia biológica de un grupos de genes. Las técnicas actuales se basan, fundamentalmente, en el uso de la información almacenada en Gene Ontology. Para ampliar este tipo de herramientas es desarrollada la herramienta CARGENE[5, 6]. Esta nueva herramienta analiza la relevancia de la coherencia de conjuntos de genes según la participación de éstos en procesos metabólicos, permitiendo extraer conclusiones de diferentes conjuntos de datos obtenidos por cualquier técnica de agrupamiento.

- Se ha realizado un estudio en profundidad de las técnicas de validación, realizando un especial esfuerzo en las basadas en conocimiento biológico previo [94, 125, 132] para contrastar técnicas que generan conjuntos de genes [93, 95]. Este estudio puso de manifiesto la necesidad de desarrollar una medida que seleccionara la funcionalidad más relevante considerando el contexto de todos los genes de entrada según [183].

  Con tal fin es propuesta la medida GFD [92], la cual analiza la similitud funcional de un conjunto de genes atendiendo a su funcionalidad más común y específica. La experimentación llevada a cabo demostró obtener mejores resultados que las medidas más relevantes hasta la fecha para todas las ontologias de GO, siendo especialmente interesante en estudios de grupos de genes que intervienen en diferente funciones biológicas

- La búsqueda de la funcionalidad más relevante para un conjunto de genes conlleva una espacio de búsqueda intratable. Para solventar tal coste computacional, se ha propuesto una aproximación heurística basada en Diagramas de Voronoi[92]. Esta aproximación aporta una reducción considerable del coste computacional sin afectar a la calidad de los resultados.

- Actualmente, no existe ningún tipo de representación diseñada para visualizar la similitud obtenida por varios grupos de genes de una forma simultánea y para todas las ontologías. Para solventar esta carencia es presentada una novedosa y descriptiva representación que denominamos GoGRAM . Esta representación ofrece una visión conjunta de la similitud funcional en cada par de ontologías, haciendo posible la comparación de diferentes conjuntos de genes.

A partir de los estudios realizados y la medida propuesta se plantean nuevos objetivos:

- Analizar y comparar el comportamiento de las técnicas de análisis de microarray atendiendo a la similitud funcional de los modelos de genes obtenidos. Para este estudio sería necesario el uso de GoGRAMS.

- Generar una nueva aproximación al desarrollo de técnicas de aprendizaje mediante la integración de distancias mixtas. Estas distancias deberían incluir el conocimiento biológico existente a medidas de distancia tradicionales y usarla como base de emparejamiento de genes o grupos de ellos.

- Desarrollar un nuevo repositorios de conocimiento biológico que integren y unifiquen el conocimiento actual. A partir de este podría realizarse estudios

más completos de validación e incluso detectar nuevas anotaciones. Anotaciones que podrían venir dadas, por ejemplo, al examinar la información almacenada de diferentes organismos para el funcionalidades equivalentes.

- Estudiar la relación entre la similitud funcional de un conjunto de proteínas y el parecido de sus secuencias de aminoácidos. Es sabido que la funcionalidad de una proteína viene determinada por la su forma estructural. Ésta, a su vez, está fuertemente influenciada por la secuencia de aminoácidos que la forman. Así, a partir de este nuevo conocimiento se podrían enriquecer los estudios de estructura de proteínas actuales así cómo ampliar las anotaciones que existen al respecto.

- Extrapolar las medida de similitud de conjuntos de genes a redes genéticas. En la actualidad, la validez de una redes genética es contrastada mediante el uso de Benchmarks o tratándolas como conjuntos de genes. Con esta aproximación solventaríamos tal carencia y permitiríamos el análisis funcional de las interacciones gen–gen.

Parte VI

Apéndices

Apéndice A

Espacio Exhaustivo y Heurístico

Las tablas A.2, A.1 y A.3 recogen la información sobre el número de anotaciones y representaciones para evaluar los conjuntos de genes con significatividad biológica para las aproximaciones exhaustiva y heurística. Así mismo, contiene información sobre el número de individuos que poseen cada espacio de búsqueda (número de combinaciones y nodos del GOTREE para exhaustivo y heurístico, respectivamente) y el número de evaluaciones de $\mathcal{S}$ que es llevado a cabo por ambas aproximaciones. Además, para la aproximación heurística es mostrado su tiempo de cálculo.

Pathway	Genes	Syn	Anot	Rep	Molecular Function				
					Exhaustivo		*Heurístico*		
					Individuos	$\mathcal{R}$'s	Individuos	$\mathcal{R}$'s	Tiempo
sce01100	645	645	2960	4132	467,69	473,01	3,04	8,37	26,33
sce01110	235	235	1152	1444	170,67	175,11	2,67	7,13	1,64
sce03008	159	157	253	253	25,09	29,18	1,36	5,46	0,14
sce04113	127	127	514	821	80,27	84,17	2,39	6,34	0,44
sce04111	125	125	426	664	65,66	69,55	2,36	6,29	0,47
sce00230	93	93	474	615	68,97	72,60	2,32	6,01	0,24
sce03010	93	91	256	413	43,12	46,74	2,19	5,84	0,20
sce03013	83	83	251	378	40,22	43,76	2,24	5,82	0,22
sce04141	79	79	250	369	42,39	45,88	2,25	5,79	0,22
sce00190	76	76	284	710	66,47	69,92	2,27	5,82	0,25
sce03040	76	76	219	307	30,84	34,29	1,89	5,39	0,16
sce00240	69	69	327	418	48,21	51,58	2,24	5,68	0,19
sce04011	57	57	243	399	37,41	40,62	2,30	5,60	0,20
sce03018	57	57	218	310	32,58	35,78	2,14	5,42	0,14
sce00010	51	51	274	345	39,24	42,34	2,16	5,37	0,14
sce04120	46	46	158	223	25,25	28,27	1,77	4,87	0,11
sce00500	40	40	195	225	27,94	30,83	2,00	5,00	0,13
sce00970	315	39	224	376	38,16	41,03	1,79	4,84	0,11
sce03420	36	36	190	283	27,75	30,55	2,16	5,12	0,13
sce03050	35	35	127	177	20,21	22,99	1,80	4,69	0,11
sce04145	34	34	108	302	27,11	29,86	2,20	5,13	0,14
sce00020	33	33	180	231	26,65	29,38	2,14	5,02	0,13
sce04144	37	33	108	162	18,02	20,74	2,06	4,90	0,14
sce00620	33	33	168	227	25,38	28,11	2,16	5,04	0,13
sce03022	32	32	143	189	19,46	22,15	2,11	4,95	0,14
sce03015	32	32	128	180	20,56	23,26	2,04	4,87	0,13
sce04146	32	32	130	181	21,42	24,11	2,05	4,88	0,11
sce00270	31	31	139	174	22,12	24,79	2,09	4,89	0,13
sce00564	31	31	130	167	20,86	23,52	2,08	4,88	0,13
sce00250	30	30	145	191	21,67	24,31	2,06	4,86	0,13
sce00330	30	30	139	174	20,90	23,54	2,11	4,90	0,11
sce03030	30	30	195	303	28,46	31,09	2,13	5,00	0,13
sce00510	30	30	101	107	15,81	18,45	1,84	4,57	0,11
sce00513	30	30	90	96	13,99	16,62	1,63	4,36	0,11
sce00520	29	29	136	174	20,87	23,48	2,05	4,81	0,11
sce00680	29	29	142	189	21,71	24,32	2,02	4,80	0,13
sce03020	29	29	112	112	15,25	17,86	1,40	4,11	0,11
sce00030	28	28	129	171	19,60	22,18	1,88	4,61	0,11
sce00051	27	27	130	180	19,83	22,37	1,95	4,68	0,11
sce00260	24	24	121	153	18,80	21,24	2,06	4,70	0,11
sce00052	23	23	132	170	19,10	21,50	1,88	4,50	0,11
sce00480	23	23	103	142	17,37	19,77	1,97	4,57	0,13
sce00563	23	23	62	70	9,19	11,59	1,62	4,13	0,11
sce04130	23	23	32	32	1,95	4,36	1,26	3,71	0,11
sce03060	21	21	61	97	11,51	13,83	1,94	4,43	0,11
sce00650	20	20	80	87	11,79	14,07	1,85	4,29	0,11
sce00561	19	19	83	113	13,62	15,85	1,98	4,43	0,11
sce03440	19	19	127	205	17,86	20,09	2,10	4,68	0,13
sce03430	19	19	120	201	18,69	20,92	2,04	4,61	0,11
sce00380	19	19	94	122	14,50	16,73	2,01	4,48	0,11
sce00350	18	18	93	105	13,26	15,44	1,83	4,24	0,11
sce00290	18	18	89	116	14,08	16,27	1,91	4,34	0,11

Pathway	Genes	Syn	Molecular Function						
			Anot	Rep	*Exhaustivo*		*Heurístico*		
					Individuos	$\mathcal{R}$'s	Individuos	$\mathcal{R}$'s	Tiempo
sce03410	17	17	113	157	15,39	17,53	2,06	4,52	0,11
sce00071	17	17	91	109	13,15	15,29	1,83	4,22	0,11
sce00400	17	17	83	94	11,36	13,49	1,87	4,23	0,11
sce04140	17	17	49	78	8,37	10,50	1,85	4,18	0,11
sce00910	16	16	65	83	10,14	12,22	1,84	4,15	0,11
sce00860	16	16	60	72	8,84	10,92	1,90	4,19	0,11
sce00100	16	16	68	84	10,59	12,67	2,03	4,33	0,11
sce00630	15	15	68	73	9,72	11,74	1,73	3,98	0,09
sce00340	15	15	65	82	9,96	11,98	2,00	4,27	0,11
sce00562	15	15	90	146	13,41	15,43	2,04	4,44	0,13
sce00670	15	15	69	84	10,11	12,13	1,79	4,07	0,11
sce04070	15	15	87	138	13,22	15,24	1,96	4,35	0,11
sce00920	15	15	75	101	11,86	13,88	1,95	4,26	0,13
sce00310	14	14	67	94	10,54	12,50	2,00	4,27	0,11
sce00740	14	14	54	69	8,30	10,26	1,92	4,12	0,11
sce00514	13	13	50	53	7,84	9,73	1,08	3,20	0,11
sce00770	13	13	59	76	9,39	11,29	1,86	4,05	0,11
sce00600	13	13	56	75	8,85	10,74	1,88	4,07	0,11
sce00900	13	13	59	76	9,46	11,35	1,82	4,01	0,11
sce00450	12	12	52	75	9,39	11,21	1,81	3,96	0,11
sce00300	11	11	62	75	8,81	10,55	1,93	4,05	0,11
sce00640	11	11	60	95	9,53	11,27	1,93	4,11	0,11
sce00280	11	11	50	54	7,31	9,05	1,74	3,78	0,11
sce00750	11	11	38	51	6,24	7,98	1,72	3,74	0,11
sce00591	10	10	34	40	5,11	6,76	1,74	3,67	0,11
sce03450	10	10	75	112	9,14	10,79	2,00	4,19	0,11
sce00040	10	10	39	45	6,08	7,73	1,76	3,72	0,11
sce01040	9	9	28	29	3,58	5,14	1,65	3,47	0,09
sce00460	9	9	24	26	3,86	5,41	1,40	3,19	0,09
sce00360	9	9	49	55	6,73	8,28	1,60	3,56	0,11
sce00410	8	8	31	37	5,04	6,48	1,67	3,49	0,11
sce00790	8	8	42	63	6,91	8,35	1,85	3,80	0,11
sce04122	8	8	34	52	5,05	6,50	1,77	3,67	0,11
sce00760	7	7	36	48	5,61	6,93	1,66	3,50	0,11
sce00590	6	6	33	47	5,11	6,29	1,82	3,61	0,11
sce00780	6	6	35	46	4,95	6,13	1,79	3,57	0,11
sce00565	6	6	29	37	4,46	5,64	1,78	3,49	0,11
sce00130	6	6	26	28	3,84	5,02	1,57	3,20	0,09
sce00903	5	5	20	20	2,88	3,88	1,34	2,82	0,11
sce00730	5	5	30	52	4,96	5,96	1,68	3,47	0,11
sce00061	4	4	38	52	4,22	4,99	1,92	3,69	0,11
sce02010	3	3	17	48	3,59	4,07	1,66	3,37	0,13
sce00592	3	3	16	22	2,48	2,95	1,67	3,07	0,09
sce00785	3	3	18	24	2,68	3,16	1,60	3,03	0,09
sce00072	2	2	9	10	1,32	1,32	1,26	2,30	0,09
sce00430	2	2	9	10	1,32	1,32	1,30	2,34	0,09

Tabla A.1: Análisis Computacional completo para la ontología MF

Pathway	Genes	Syn	Biological Process						
			Anot	Rep	*Exhaustivo*		*Heurístico*		
					Individuos	$\mathcal{R}$'s	Individuos	$\mathcal{R}$'s	Tiempo
sce01100	645	645	2544	72354	1131,73	1137,05	4,48	9,93	1160,59
sce01110	235	235	1005	26745	430,06	434,50	4,19	8,92	127,44
sce03008	159	157	312	4090	208,11	212,20	3,13	7,34	3,38
sce04113	127	127	884	8850	204,08	207,99	3,84	8,07	19,70
sce04111	125	125	909	10410	216,56	220,45	3,83	8,09	20,47
sce00230	93	93	385	15953	181,97	185,60	4,06	8,37	38,08
sce03010	93	91	396	5969	151,11	154,72	3,49	7,49	5,56
sce03013	83	83	451	4759	134,24	137,77	3,55	7,47	5,75
sce04141	79	79	323	2869	107,35	110,83	3,60	7,38	5,06
sce00190	76	76	350	16114	133,81	137,26	3,42	7,69	8,73
sce03040	76	76	321	4116	127,71	131,16	3,20	7,05	2,11
sce00240	69	69	299	10111	135,83	139,20	4,07	8,17	24,66
sce04011	57	57	373	4954	97,39	100,59	3,80	7,62	10,98
sce03018	57	57	438	13389	120,87	124,07	3,61	7,79	14,63
sce00010	51	51	225	3071	80,44	83,54	3,46	7,10	3,05
sce04120	46	46	365	3033	74,23	77,24	3,62	7,23	4,49
sce00500	40	40	169	1867	60,25	63,14	3,15	6,57	0,99
sce00970	315	39	152	2602	70,08	72,95	3,08	6,60	1,02
sce03420	36	36	264	3185	66,69	69,48	3,28	6,86	1,73
sce03050	35	35	114	1116	47,82	50,60	3,04	6,27	0,77
sce04145	34	34	205	5187	61,48	64,23	3,42	7,18	3,33
sce00020	33	33	167	1659	51,53	54,25	3,20	6,54	1,06
sce04144	37	33	229	2473	55,71	58,43	3,52	6,99	2,77
sce00620	33	33	149	1740	51,86	54,58	3,33	6,69	1,45
sce03022	32	32	199	2643	60,04	62,73	3,05	6,55	0,94
sce03015	32	32	166	2431	56,67	59,36	3,33	6,79	1,75
sce04146	32	32	131	1056	43,06	45,76	3,22	6,41	0,97
sce00270	31	31	137	6465	69,44	72,11	3,54	7,38	4,39
sce00564	31	31	117	1424	48,94	51,61	3,07	6,35	0,72
sce00250	30	30	121	3732	59,40	62,04	3,63	7,25	4,63
sce00330	30	30	117	2927	56,67	59,31	3,34	6,86	1,64
sce03030	30	30	242	3413	58,29	60,93	3,30	6,88	1,97
sce00510	30	30	70	615	36,09	38,73	2,62	5,64	0,19
sce00513	30	30	63	647	36,31	38,94	2,78	5,82	0,34
sce00520	29	29	135	1911	46,93	49,54	3,38	6,75	1,72
sce00680	29	29	93	1438	44,78	47,39	3,18	6,45	0,86
sce03020	29	29	105	2182	53,49	56,09	3,00	6,42	0,67
sce00030	28	28	108	1785	46,43	49,01	3,22	6,55	1,13
sce00051	27	27	99	906	34,73	37,28	3,06	6,16	0,66
sce00260	24	24	131	3805	49,96	52,40	3,44	7,05	2,27
sce00052	23	23	112	1272	35,33	37,73	3,16	6,35	0,78
sce00480	23	23	83	1394	38,49	40,89	3,32	6,54	1,38
sce00563	23	23	50	1048	36,83	39,24	2,59	5,70	0,22
sce04130	23	23	120	520	29,99	32,39	2,50	5,39	0,14
sce03060	21	21	78	510	27,68	30,00	2,81	5,67	0,38
sce00650	20	20	70	777	26,44	28,71	2,98	5,96	0,52
sce00561	19	19	73	566	26,09	28,32	2,88	5,75	0,41
sce03440	19	19	189	1968	37,78	40,01	3,06	6,39	0,75
sce03430	19	19	148	1664	34,97	37,21	3,10	6,37	0,77
sce00380	19	19	70	1687	31,89	34,12	3,30	6,56	1,16
sce00350	18	18	73	1043	28,71	30,89	3,19	6,27	0,80
sce00290	18	18	73	1646	34,51	36,70	3,02	6,27	0,70

Pathway	**Genes**	**Syn**	**Biological Process**						
			Anot	**Rep**	*Exhaustivo*		*Heurístico*		
					Individuos	$\mathcal{R}$'s	Individuos	$\mathcal{R}$'s	Tiempo
sce03410	17	17	126	1713	32,07	34,20	3,17	6,44	0,83
sce00071	17	17	77	626	25,95	28,09	2,87	5,75	0,38
sce00400	17	17	72	2325	34,42	36,56	3,11	6,50	0,81
sce04140	17	17	142	808	28,26	30,39	2,81	5,78	0,41
sce00910	16	16	69	2033	32,88	34,96	3,20	6,54	1,06
sce00860	16	16	56	1300	29,07	31,15	2,98	6,13	0,45
sce00100	16	16	76	572	23,31	25,39	2,54	5,38	0,14
sce00630	15	15	74	941	25,29	27,31	3,25	6,27	0,89
sce00340	15	15	55	1138	27,03	29,05	3,04	6,13	0,59
sce00562	15	15	57	677	21,48	23,50	3,20	6,09	0,81
sce00670	15	15	66	3551	31,73	33,75	3,61	7,17	3,70
sce04070	15	15	61	456	20,95	22,97	2,83	5,58	0,39
sce00920	15	15	74	3222	33,52	35,54	2,96	6,49	0,64
sce00310	14	14	76	1529	24,77	26,73	3,36	6,57	1,30
sce00740	14	14	26	495	17,33	19,29	2,94	5,70	0,44
sce00514	13	13	28	333	16,76	18,66	2,58	5,20	0,14
sce00770	13	13	33	888	23,17	25,06	2,76	5,74	0,36
sce00600	13	13	47	316	17,62	19,51	2,64	5,24	0,14
sce00900	13	13	73	713	21,54	23,43	2,76	5,66	0,36
sce00450	12	12	51	1880	24,86	26,68	3,02	6,31	0,69
sce00300	11	11	59	1834	23,33	25,07	2,91	6,19	0,47
sce00640	11	11	41	380	16,38	18,12	2,87	5,51	0,36
sce00280	11	11	58	836	19,30	21,04	3,08	6,03	0,61
sce00750	11	11	45	726	18,26	20,01	2,89	5,79	0,41
sce00591	10	10	38	308	11,32	12,97	2,72	5,27	0,38
sce03450	10	10	82	1158	19,65	21,30	3,05	6,13	0,59
sce00040	10	10	43	449	14,78	16,43	2,99	5,69	0,44
sce01040	9	9	41	487	15,25	16,80	2,63	5,35	0,14
sce00460	9	9	30	829	15,75	17,31	2,88	5,82	0,39
sce00360	9	9	36	1077	14,63	16,19	3,26	6,31	0,84
sce00410	8	8	26	474	13,85	15,29	2,86	5,56	0,41
sce00790	8	8	31	665	14,18	15,62	2,90	5,74	0,42
sce04122	8	8	43	872	15,62	17,06	2,84	5,79	0,38
sce00760	7	7	33	2003	16,71	18,03	3,33	6,64	1,41
sce00590	6	6	24	295	8,87	10,05	2,93	5,42	0,39
sce00780	6	6	13	208	8,50	9,68	2,42	4,77	0,13
sce00565	6	6	27	325	10,22	11,40	2,79	5,32	0,38
sce00130	6	6	18	140	7,98	9,16	2,37	4,57	0,13
sce00903	5	5	20	164	6,33	7,33	2,56	4,80	0,14
sce00730	5	5	15	427	9,30	10,30	2,94	5,58	0,41
sce00061	4	4	19	128	6,01	6,79	2,15	4,28	0,13
sce02010	3	3	13	22	2,48	2,95	1,81	3,21	0,14
sce00592	3	3	17	185	5,37	5,85	2,56	4,83	0,14
sce00785	3	3	8	145	4,72	5,20	2,36	4,53	0,13
sce00072	2	2	8	68	3,03	3,03	1,96	3,80	0,11
sce00430	2	2	7	205	3,98	3,98	2,75	5,06	0,36

Tabla A.2: Análisis Computacional completo para la ontología BP

Pathway	Genes	Syn	Cellular Component						
			Anot	Rep	*Exhaustivo*		*Heurístico*		
					Individuos	$\mathcal{R}$'s	Individuos	$\mathcal{R}$'s	Tiempo
sce01100	645	645	1858	5745	469,73	475,05	2,88	8,21	19,33
sce01110	235	235	545	1502	139,05	143,49	2,48	6,94	1,19
sce03008	159	157	804	2352	180,09	184,18	1,99	6,15	0,30
sce04113	127	127	434	1283	110,21	114,11	2,65	6,62	0,70
sce04111	125	125	431	1377	116,26	120,15	2,56	6,52	0,63
sce00230	93	93	223	743	65,10	68,73	2,38	6,08	0,34
sce03010	93	91	360	1006	87,52	91,14	2,37	6,08	0,33
sce03013	83	83	289	873	79,53	83,06	2,55	6,18	0,36
sce04141	79	79	355	1268	84,23	87,72	2,60	6,24	0,36
sce00190	76	76	385	1520	95,11	98,57	2,51	6,15	0,28
sce03040	76	76	335	1210	86,24	89,70	2,32	5,93	0,25
sce00240	69	69	186	673	59,51	62,88	2,32	5,80	0,22
sce04011	57	57	221	409	39,64	42,84	2,32	5,63	0,20
sce03018	57	57	256	722	58,45	61,66	2,34	5,70	0,19
sce00010	51	51	138	345	32,73	35,84	2,10	5,31	0,14
sce04120	46	46	148	415	38,31	41,33	2,40	5,56	0,16
sce00500	40	40	94	167	18,23	21,12	1,97	4,94	0,13
sce00970	315	39	86	259	27,33	30,20	1,78	4,78	0,11
sce03420	36	36	118	365	34,85	37,65	2,14	5,13	0,13
sce03050	35	35	187	446	37,00	39,77	1,85	4,86	0,11
sce04145	34	34	212	729	44,36	47,11	2,56	5,67	0,17
sce00020	33	33	107	461	33,10	35,82	2,24	5,24	0,13
sce04144	37	33	188	560	40,57	43,29	2,52	5,56	0,17
sce00620	33	33	91	313	27,53	30,25	2,22	5,15	0,13
sce03022	32	32	102	312	30,34	33,03	2,14	5,05	0,13
sce03015	32	32	97	247	25,63	28,33	2,14	5,01	0,13
sce04146	32	32	107	416	31,00	33,69	2,32	5,28	0,16
sce00270	31	31	56	118	13,43	16,10	1,93	4,70	0,11
sce00564	31	31	129	378	28,90	31,57	2,28	5,20	0,14
sce00250	30	30	60	127	13,49	16,13	1,99	4,74	0,13
sce00330	30	30	62	185	18,17	20,81	1,95	4,75	0,11
sce03030	30	30	120	424	33,28	35,92	2,19	5,12	0,13
sce00510	30	30	139	455	34,06	36,69	2,17	5,12	0,13
sce00513	30	30	146	470	33,46	36,10	2,14	5,09	0,13
sce00520	29	29	76	170	14,49	17,10	2,23	4,99	0,13
sce00680	29	29	67	161	15,96	18,57	2,15	4,90	0,13
sce03020	29	29	93	415	32,76	35,37	1,86	4,77	0,11
sce00030	28	28	50	83	9,83	12,41	1,43	4,10	0,11
sce00051	27	27	46	88	9,08	11,63	1,83	4,47	0,11
sce00260	24	24	49	132	13,18	15,62	2,03	4,64	0,13
sce00052	23	23	39	72	7,56	9,96	1,61	4,12	0,11
sce00480	23	23	51	121	11,52	13,92	2,08	4,66	0,13
sce00563	23	23	111	362	26,26	28,66	2,12	4,91	0,13
sce04130	23	23	141	436	27,75	30,15	2,45	5,29	0,14
sce03060	21	21	109	377	25,78	28,10	2,21	4,98	0,13
sce00650	20	20	37	111	9,36	11,63	2,00	4,48	0,11
sce00561	19	19	48	108	11,39	13,63	1,94	4,39	0,11
sce03440	19	19	52	178	17,24	19,48	2,04	4,58	0,11
sce03430	19	19	63	209	18,79	21,02	2,02	4,60	0,11
sce00380	19	19	37	128	10,71	12,95	2,06	4,54	0,11
sce00350	18	18	31	74	8,09	10,27	1,85	4,20	0,11
sce00290	18	18	41	150	14,08	16,27	1,84	4,32	0,11

Pathway	**Genes**	**Syn**	**Anot**	**Rep**	**Cellular Component**				
					Exhaustivo		*Heurístico*		
					Individuos	$\mathcal{R}$'s	Individuos	$\mathcal{R}$'s	Tiempo
sce03410	17	17	43	144	15,13	17,26	1,83	4,28	0,11
sce00071	17	17	37	121	11,53	13,67	2,04	4,45	0,13
sce00400	17	17	29	58	6,69	8,82	1,72	4,01	0,11
sce04140	17	17	92	215	15,94	18,08	2,15	4,69	0,11
sce00910	16	16	37	66	7,84	9,92	1,76	4,03	0,11
sce00860	16	16	41	136	12,05	14,12	1,88	4,29	0,11
sce00100	16	16	63	185	16,07	18,14	1,82	4,30	0,11
sce00630	15	15	33	119	11,09	13,11	1,93	4,28	0,11
sce00340	15	15	26	64	6,16	8,18	1,83	4,06	0,11
sce00562	15	15	54	162	12,31	14,33	2,26	4,68	0,13
sce00670	15	15	24	52	7,14	9,16	1,48	3,67	0,11
sce04070	15	15	66	191	14,28	16,30	2,28	4,75	0,13
sce00920	15	15	27	54	6,50	8,53	1,79	3,99	0,11
sce00310	14	14	37	140	10,89	12,84	2,15	4,51	0,11
sce00740	14	14	25	57	5,46	7,42	1,98	4,15	0,13
sce00514	13	13	55	156	13,67	15,56	1,76	4,13	0,11
sce00770	13	13	27	84	9,49	11,39	1,88	4,08	0,11
sce00600	13	13	72	220	15,30	17,19	2,06	4,53	0,13
sce00900	13	13	31	86	8,42	10,31	1,86	4,08	0,11
sce00450	12	12	24	49	5,96	7,78	1,73	3,79	0,11
sce00300	11	11	17	37	4,25	5,99	1,32	3,29	0,11
sce00640	11	11	31	101	9,59	11,33	2,01	4,20	0,11
sce00280	11	11	27	100	9,04	10,78	2,02	4,21	0,11
sce00750	11	11	15	23	2,41	4,15	1,15	3,04	0,11
sce00591	10	10	16	34	3,43	5,08	1,79	3,68	0,11
sce03450	10	10	39	139	10,64	12,30	2,04	4,31	0,13
sce00040	10	10	18	46	4,25	5,91	1,75	3,71	0,11
sce01040	9	9	38	139	9,52	11,08	2,13	4,37	0,11
sce00460	9	9	27	52	5,47	7,03	1,91	3,85	0,11
sce00360	9	9	13	27	2,42	3,98	1,69	3,49	0,09
sce00410	8	8	16	46	4,56	6,00	1,73	3,60	0,11
sce00790	8	8	19	50	4,36	5,81	1,81	3,70	0,11
sce04122	8	8	14	30	3,85	5,29	1,32	3,09	0,11
sce00760	7	7	13	28	3,79	5,11	1,38	3,07	0,09
sce00590	6	6	15	33	3,26	4,44	1,73	3,41	0,11
sce00780	6	6	15	50	3,80	4,97	1,91	3,73	0,11
sce00565	6	6	29	89	6,38	7,56	1,94	3,96	0,13
sce00130	6	6	19	75	5,78	6,95	1,92	3,87	0,11
sce00903	5	5	11	49	4,20	5,20	1,91	3,68	0,11
sce00730	5	5	7	12	1,15	2,15	1,30	2,64	0,11
sce00061	4	4	13	32	3,50	4,28	1,59	3,17	0,11
sce02010	3	3	13	29	2,89	3,37	1,67	3,18	0,11
sce00592	3	3	7	31	2,77	3,25	1,76	3,29	0,11
sce00785	3	3	4	13	1,90	2,38	1,20	2,41	0,11
sce00072	2	2	3	6	0,90	0,90	1,23	2,08	0,11
sce00430	2	2	6	20	1,28	1,28	1,69	3,01	0,11

Tabla A.3: Análisis Computacional completo para la ontología CC

Apéndice B

Valores de similitud obtenidos

En la tabla B.1 se muestran los valores de similitud generados por las medidas GFD, GS^2, Resnik y Wang en las tres ontologías de GO para los conjuntos de genes con y sin coherencia funcional extraídos de KEGG (la primera fila representa los conjuntos de genes obtenidos de los pathways metabólicos, y la segunda fila muestra los conjunto de genes seleccionados aleatoriamente). Por razones de claridad, se mostrarán los resultados de $\overline{\text{GFD}} = 1 - \text{GFD}$ para la medida propuesta y -1 para aquellas evaluaciones que no generen resultado alguno.

	Biological Process				Molecular Function			Cellular Component		
Identificador	$\overline{\text{GFD}}$	GS^2	Resnik	Wang	$\overline{\text{GFD}}$	Resnik	Wang	$\overline{\text{GFD}}$	Resnik	Wang
sce00010	0,55	0,46	0,32	0,48	0,64	0,18	0,31	0,68	0,14	0,76
	0,30	0,15	-1	0,88	0,28	0,22	0,65	0,55	0,04	0,47
sce00020	0,75	0,54	0,31	0,52	0,62	0,21	0,33	0,81	0,22	0,78
	0,31	0,09	0,08	0,60	0,28	0,07	0,54	0,53	0,07	0,51
sce00030	0,75	0,22	0,37	0,44	0,60	0,18	0,35	0,83	0,10	0,91
	0,32	0,12	0,14	0,44	0,29	0,33	0,58	0,63	0,12	0,57
sce00040	0,67	0,37	0,30	0,38	0,62	0,19	0,33	0,45	0,08	0,58
	0,29	0,17	-1	-1	0,28	-1	-1	0,49	-1	-1
sce00051	0,57	0,40	0,27	0,45	0,66	0,24	0,38	0,48	0,13	0,68
	0,28	0,08	-1	0,58	0,32	-1	1,00	0,56	-1	0,33
sce00052	0,73	0,38	0,45	0,53	0,63	0,22	0,37	0,51	0,10	0,67
	0,32	0,09	-1	0,48	0,27	-1	0,56	0,58	0,11	0,38
sce00061	0,94	0,55	0,59	0,92	0,68	0,34	0,57	0,92	0,19	0,81
	0,26	0,26	-1	1,00	0,47	-1	0,36	0,38	-1	0,45
sce00071	0,72	0,37	0,29	0,42	0,70	0,26	0,43	0,59	0,15	0,64
	0,35	0,10	0,08	0,34	0,30	0,00	0,44	0,48	0,00	0,60
sce00072	0,94	0,39	0,54	1,00	0,83	0,40	0,36	0,70	0,11	0,63
	0,64	0,03	-1	-1	0,17	-1	-1	0,78	-1	-1
sce00100	0,87	0,46	0,48	0,78	0,65	0,20	0,31	0,86	0,28	0,89
	0,26	0,11	-1	0,50	0,26	-1	1,00	0,54	-1	0,36
sce00130	0,78	0,41	0,44	0,70	0,66	0,17	0,33	0,79	0,19	0,75
	0,47	0,11	-1	-1	0,31	-1	-1	0,68	-1	-1
sce00190	0,50	0,36	0,26	0,50	0,50	0,32	0,43	0,82	0,29	0,78
	0,30	0,14	0,28	0,63	0,28	0,29	0,67	0,49	0,14	0,58

Identificador	Biological Process				Molecular Function			Cellular Component		
	$\overline{\text{GFD}}$	GS^2	Resnik	Wang	$\overline{\text{GFD}}$	Resnik	Wang	$\overline{\text{GFD}}$	Resnik	Wang
sce00230	0,61	0,16	0,22	0,44	0,59	0,18	0,36	0,64	0,16	0,62
	0,28	0,12	0,12	0,69	0,31	0,22	0,75	0,49	0,06	0,60
sce00240	0,68	0,22	0,24	0,49	0,60	0,21	0,40	0,75	0,22	0,63
	0,32	0,11	0,09	0,50	0,30	0,14	0,58	0,53	0,09	0,54
sce00250	0,73	0,13	0,31	0,48	0,64	0,17	0,33	0,61	0,09	0,66
	0,31	0,11	-1	0,68	0,26	-1	1,00	0,54	-1	0,60
sce00260	0,74	0,22	0,37	0,59	0,70	0,16	0,30	0,68	0,11	0,74
	0,33	0,11	0,21	0,41	0,32	0,10	0,44	0,48	0,14	0,60
sce00270	0,83	0,26	0,39	0,59	0,67	0,17	0,35	0,74	0,09	0,78
	0,31	0,12	0,14	0,58	0,30	0,33	0,67	0,46	0,14	0,61
sce00280	0,67	0,25	0,26	0,43	0,68	0,24	0,38	0,67	0,13	0,70
	0,28	0,08	-1	1,00	0,25	-1	1,00	0,53	-1	1,00
sce00290	0,79	0,26	0,33	0,58	0,62	0,16	0,32	0,76	0,15	0,75
	0,31	0,07	-1	-1	0,27	-1	-1	0,41	-1	-1
sce00300	0,91	0,57	0,48	0,77	0,67	0,18	0,31	0,72	0,09	0,81
	0,31	0,18	-1	1,00	0,38	-1	1,00	0,47	-1	0,24
sce00310	0,55	0,29	0,22	0,38	0,61	0,22	0,31	0,68	0,13	0,67
	0,27	0,09	-1	0,48	0,33	-1	0,56	0,45	0,08	0,55
sce00330	0,73	0,15	0,32	0,52	0,64	0,15	0,32	0,67	0,12	0,76
	0,36	0,10	0,37	0,54	0,32	0,43	0,60	0,59	0,35	0,63
sce00340	0,68	0,22	0,29	0,54	0,62	0,18	0,31	0,54	0,05	0,69
	0,28	0,12	-1	0,48	0,29	-1	1,00	0,62	-1	0,56
sce00350	0,59	0,20	0,29	0,47	0,65	0,24	0,38	0,62	0,08	0,74
	0,22	0,15	-1	1,00	0,31	-1	1,00	0,57	-1	1,00
sce00360	0,67	0,21	0,35	0,55	0,66	0,25	0,43	0,58	0,07	0,73
	0,31	0,20	-1	1,00	0,39	-1	1,00	0,51	-1	0,45
sce00380	0,53	0,21	0,27	0,37	0,61	0,18	0,33	0,56	0,12	0,67
	0,32	0,18	0,20	0,64	0,30	0,05	0,62	0,57	0,08	0,52
sce00400	0,90	0,52	0,42	0,72	0,69	0,20	0,37	0,79	0,08	0,89
	0,30	0,09	0,09	0,38	0,27	0,00	0,61	0,56	0,09	0,65
sce00410	0,76	0,15	0,31	0,43	0,65	0,18	0,37	0,72	0,08	0,78
	0,33	0,07	-1	0,23	0,25	-1	0,34	0,63	0,08	0,80
sce00430	0,75	0,00	0,27	0,26	0,60	0,08	0,25	0,56	0,08	0,53
	0,50	0,21	-1	-1	0,50	-1	-1	0,75	-1	-1
sce00450	0,71	0,26	0,26	0,57	0,63	0,20	0,39	0,76	0,09	0,65
	0,34	0,13	-1	0,23	0,27	-1	0,34	0,49	-1	0,38
sce00460	0,68	0,19	0,35	0,47	0,68	0,36	0,56	0,44	0,20	0,57
	0,28	0,07	-1	-1	0,30	-1	-1	0,58	-1	-1
sce00480	0,54	0,26	0,29	0,38	0,66	0,23	0,38	0,64	0,09	0,75
	0,33	0,08	-1	-1	0,32	-1	-1	0,57	-1	-1
sce00500	0,64	0,37	0,34	0,44	0,63	0,26	0,39	0,48	0,09	0,59
	0,29	0,13	0,07	0,51	0,30	0,01	0,63	0,53	0,08	0,56
sce00510	0,78	0,46	0,45	0,70	0,70	0,36	0,48	0,84	0,36	0,76
	0,32	0,10	0,08	0,51	0,29	0,07	0,44	0,53	0,03	0,53
sce00513	0,76	0,40	0,49	0,74	0,69	0,50	0,66	0,84	0,28	0,66
	0,29	0,16	0,08	0,64	0,36	0,06	0,57	0,47	0,11	0,70
sce00514	0,94	0,02	0,61	0,80	0,88	0,60	0,78	0,90	0,21	0,71
	0,25	0,12	-1	-1	0,32	-1	-1	0,54	-1	-1
sce00520	0,63	0,26	0,28	0,36	0,63	0,18	0,32	0,60	0,09	0,65
	0,27	0,19	0,00	0,71	0,33	-1	0,91	0,42	0,07	0,60
sce00561	0,57	0,28	0,26	0,41	0,62	0,20	0,31	0,57	0,12	0,68
	0,35	0,10	-1	0,48	0,27	-1	0,56	0,54	0,11	0,43

	Biological Process				Molecular Function			Cellular Component		
Identificador	$\overline{\text{GFD}}$	GS^2	Resnik	Wang	$\overline{\text{GFD}}$	Resnik	Wang	$\overline{\text{GFD}}$	Resnik	Wang
sce00562	0,67	0,08	0,36	0,44	0,64	0,24	0,34	0,59	0,11	0,62
	0,28	0,16	-1	0,42	0,35	-1	0,64	0,58	-1	0,52
sce00563	0,87	0,62	0,54	0,79	0,53	0,22	0,46	0,90	0,30	0,69
	0,35	0,14	0,03	0,32	0,27	0,29	0,52	0,52	0,05	0,56
sce00564	0,77	0,30	0,42	0,53	0,65	0,19	0,30	0,64	0,12	0,66
	0,30	0,14	-1	0,74	0,29	-1	0,78	0,54	0,03	0,58
sce00565	0,85	0,24	0,39	0,47	0,64	0,18	0,29	0,71	0,17	0,70
	0,30	0,07	-1	-1	0,27	-1	-1	0,56	-1	-1
sce00590	0,57	0,18	0,20	0,41	0,60	0,24	0,43	0,51	0,07	0,60
	0,26	0,29	0,18	1,00	0,32	0,19	1,00	0,37	0,11	0,50
sce00591	0,79	0,52	0,31	0,52	0,73	0,48	0,69	0,36	0,09	0,60
	0,31	0,13	-1	0,68	0,31	-1	0,75	0,58	0,12	0,74
sce00592	0,88	0,23	0,67	0,60	0,57	0,19	0,35	0,73	0,32	0,62
	0,64	0,00	0,37	1,00	0,59	0,43	1,00	0,77	0,35	1,00
sce00600	0,86	0,18	0,53	0,70	0,63	0,19	0,33	0,89	0,27	0,78
	0,27	0,09	0,00	0,44	0,33	0,00	0,44	0,53	0,04	0,42
sce00620	0,60	0,40	0,29	0,48	0,66	0,18	0,31	0,73	0,16	0,74
	0,32	0,14	0,15	0,55	0,34	0,22	0,60	0,50	0,16	0,62
sce00630	0,69	0,37	0,35	0,50	0,65	0,21	0,35	0,64	0,18	0,64
	0,33	0,08	0,28	0,94	0,29	0,00	0,23	0,47	0,35	0,53
sce00640	0,61	0,33	0,23	0,43	0,66	0,25	0,36	0,76	0,16	0,80
	0,31	0,12	-1	0,61	0,39	-1	0,67	0,60	0,08	0,58
sce00650	0,65	0,35	0,23	0,45	0,70	0,27	0,45	0,48	0,10	0,61
	0,29	0,16	-1	1,00	0,25	-1	1,00	0,51	-1	0,52
sce00670	0,59	0,21	0,34	0,46	0,64	0,32	0,44	0,60	0,13	0,71
	0,31	0,12	0,03	0,30	0,26	0,00	0,43	0,57	0,03	0,35
sce00680	0,51	0,29	0,27	0,43	0,63	0,14	0,30	0,66	0,14	0,74
	0,32	0,09	0,03	0,42	0,28	0,00	0,28	0,62	0,03	0,48
sce00730	0,75	0,43	0,46	0,60	0,83	0,28	0,52	0,47	0,13	0,63
	0,29	0,05	-1	-1	0,38	-1	-1	0,34	-1	-1
sce00740	0,48	0,22	0,32	0,44	0,62	0,20	0,37	0,40	0,05	0,51
	0,29	0,13	-1	1,00	0,38	-1	1,00	0,53	-1	0,63
sce00750	0,68	0,13	0,36	0,52	0,65	0,10	0,47	0,44	0,09	0,70
	0,24	0,12	-1	1,00	0,37	-1	1,00	0,46	-1	1,00
sce00760	0,92	0,09	0,54	0,59	0,64	0,23	0,40	0,75	0,10	0,90
	0,40	0,10	-1	-1	0,35	-1	-1	0,71	-1	-1
sce00770	0,68	0,27	0,35	0,51	0,63	0,14	0,31	0,74	0,13	0,78
	0,33	0,07	0,04	0,28	0,26	0,00	0,38	0,54	0,10	0,53
sce00780	0,48	0,20	0,26	0,47	0,55	0,13	0,33	0,66	0,11	0,70
	0,31	0,17	-1	1,00	0,34	-1	1,00	0,61	-1	0,45
sce00785	0,94	0,52	0,83	1,00	0,83	0,54	0,63	0,92	0,21	1,00
	0,39	0,03	-1	-1	0,22	-1	-1	0,81	-1	-1
sce00790	0,76	0,24	0,39	0,51	0,65	0,14	0,31	0,73	0,10	0,78
	0,33	0,15	-1	-1	0,34	-1	-1	0,60	-1	-1
sce00860	0,66	0,19	0,42	0,63	0,58	0,18	0,32	0,59	0,14	0,68
	0,31	0,08	-1	-1	0,30	-1	-1	0,56	-1	-1
sce00900	0,76	0,37	0,36	0,55	0,67	0,20	0,37	0,68	0,13	0,71
	0,37	0,15	0,03	0,21	0,32	0,29	0,61	0,66	0,03	0,40
sce00903	0,58	0,33	0,42	0,49	0,70	0,34	0,49	0,54	0,18	0,56
	0,25	0,13	-1	-1	0,23	-1	-1	0,80	-1	-1
sce00910	0,87	0,13	0,29	0,51	0,64	0,22	0,38	0,61	0,11	0,63
	0,30	0,10	0,13	0,59	0,34	0,18	0,60	0,62	0,13	0,66

Identificador	Biological Process				Molecular Function			Cellular Component		
	$\overline{\text{GFD}}$	GS^2	Resnik	Wang	$\overline{\text{GFD}}$	Resnik	Wang	$\overline{\text{GFD}}$	Resnik	Wang
sce00920	0,85	0,47	0,38	0,70	0,66	0,18	0,35	0,64	0,08	0,66
	0,28	0,09	0,28	0,44	0,27	0,43	0,56	0,51	0,14	0,75
sce00970	0,94	0,01	0,46	0,73	0,89	0,47	0,69	0,83	0,13	0,84
	0,32	0,10	0,13	0,38	0,36	0,33	0,61	0,52	0,10	0,51
sce01040	0,89	0,18	0,50	0,62	0,47	0,18	0,37	0,71	0,21	0,67
	0,29	0,19	-1	-1	0,25	-1	-1	0,55	-1	-1
sce01100	0,43	0,16	0,20	0,35	0,56	0,13	0,28	0,61	0,11	0,64
	0,29	0,12	0,16	0,57	0,29	0,19	0,65	0,52	0,11	0,56
sce01110	0,49	0,21	0,26	0,41	0,62	0,14	0,30	0,64	0,11	0,71
	0,28	0,11	0,15	0,61	0,29	0,22	0,68	0,54	0,11	0,56
sce02010	0,83	0,36	0,20	0,38	0,95	0,62	0,92	0,92	0,19	0,76
	0,17	0,03	-1	0,23	0,34	-1	0,34	0,34	-1	0,27
sce03008	0,88	0,67	0,29	0,85	0,83	0,39	0,98	0,84	0,50	0,87
	0,30	0,13	0,09	0,74	0,27	0,07	0,73	0,53	0,11	0,59
sce03010	0,51	0,20	0,27	0,49	0,36	0,11	0,40	0,70	0,26	0,65
	0,28	0,11	0,21	0,62	0,29	0,08	0,73	0,52	0,14	0,58
sce03013	0,43	0,21	0,22	0,41	0,41	0,12	0,41	0,67	0,21	0,60
	0,33	0,11	-1	0,57	0,28	-1	0,66	0,47	-1	0,63
sce03015	0,58	0,21	0,26	0,46	0,61	0,15	0,42	0,72	0,20	0,64
	0,30	0,10	0,05	0,27	0,28	0,12	0,35	0,55	0,07	0,39
sce03018	0,66	0,19	0,35	0,54	0,51	0,13	0,42	0,76	0,21	0,67
	0,29	0,10	0,01	0,62	0,30	0,00	0,88	0,56	0,09	0,47
sce03020	0,93	0,47	0,39	0,80	0,93	0,47	0,72	0,92	0,62	0,90
	0,30	0,14	-1	0,81	0,32	-1	0,73	0,60	0,07	0,46
sce03022	0,93	0,35	0,41	0,67	0,83	0,45	0,86	0,91	0,49	0,84
	0,31	0,10	0,12	0,47	0,29	0,29	0,63	0,54	0,17	0,53
sce03030	0,91	0,37	0,41	0,71	0,79	0,24	0,40	0,88	0,37	0,73
	0,29	0,14	0,15	0,47	0,26	0,06	0,51	0,54	0,23	0,57
sce03040	0,84	0,35	0,39	0,71	0,49	0,18	0,57	0,83	0,37	0,78
	0,31	0,14	0,18	0,69	0,31	0,27	0,67	0,58	0,11	0,58
sce03050	0,90	0,12	0,37	0,75	0,47	0,15	0,58	0,88	0,42	0,81
	0,30	0,14	-1	1,00	0,29	-1	0,82	0,56	-1	0,67
sce03060	0,62	0,31	0,44	0,62	0,29	0,14	0,43	0,78	0,36	0,66
	0,31	0,08	0,00	0,36	0,28	0,07	0,28	0,62	0,03	0,43
sce03410	0,90	0,48	0,45	0,81	0,66	0,24	0,38	0,92	0,24	0,77
	0,25	0,14	-1	1,00	0,29	-1	0,57	0,50	-1	0,45
sce03420	0,84	0,29	0,35	0,65	0,60	0,17	0,38	0,91	0,32	0,70
	0,28	0,11	0,17	0,42	0,29	0,14	0,48	0,54	0,09	0,66
sce03430	0,92	0,27	0,41	0,74	0,75	0,23	0,40	0,92	0,31	0,74
	0,28	0,14	0,00	0,30	0,29	0,00	0,33	0,48	-1	0,71
sce03440	0,84	0,29	0,39	0,69	0,63	0,23	0,41	0,92	0,22	0,76
	0,28	0,09	0,05	0,23	0,34	-1	0,67	0,51	0,08	0,58
sce03450	0,94	0,42	0,41	0,69	0,81	0,16	0,44	0,92	0,24	0,83
	0,31	0,24	-1	1,00	0,34	-1	1,00	0,63	-1	0,38
sce04011	0,56	0,16	0,28	0,40	0,37	0,14	0,33	0,52	0,10	0,59
	0,30	0,14	0,24	0,60	0,32	0,25	0,73	0,55	0,19	0,53
sce04070	0,65	0,10	0,34	0,44	0,59	0,23	0,33	0,59	0,11	0,57
	0,33	0,13	-1	1,00	0,33	-1	1,00	0,56	-1	0,45
sce04111	0,59	0,28	0,27	0,40	0,34	0,12	0,40	0,79	0,19	0,65
	0,29	0,13	0,10	0,60	0,30	0,14	0,69	0,57	0,10	0,62
sce04113	0,51	0,26	0,21	0,35	0,34	0,10	0,36	0,69	0,15	0,61
	0,30	0,12	0,19	0,77	0,31	0,20	0,74	0,54	0,09	0,64

	Biological Process				Molecular Function			Cellular Component		
Identificador	$\overline{\text{GFD}}$	GS^2	Resnik	Wang	$\overline{\text{GFD}}$	Resnik	Wang	$\overline{\text{GFD}}$	Resnik	Wang
sce04120	0,80	0,23	0,33	0,56	0,71	0,39	0,77	0,75	0,26	0,69
	0,30	0,15	0,08	0,72	0,28	0,16	0,65	0,53	0,06	0,59
sce04122	0,74	0,32	0,38	0,55	0,50	0,21	0,44	0,80	0,11	0,83
	0,31	0,12	-1	-1	0,32	-1	-1	0,68	-1	-1
sce04130	0,75	0,48	0,44	0,71	0,88	0,59	0,95	0,78	0,27	0,68
	0,38	0,13	-1	-1	0,33	-1	-1	0,58	-1	-1
sce04140	0,87	0,54	0,48	0,72	0,33	0,14	0,35	0,74	0,25	0,68
	0,30	0,07	0,00	0,58	0,31	0,00	0,71	0,67	0,12	0,67
sce04141	0,44	0,15	0,22	0,43	0,32	0,11	0,36	0,72	0,22	0,63
	0,29	0,10	0,23	0,61	0,29	0,27	0,62	0,51	0,08	0,55
sce04144	0,54	0,24	0,23	0,43	0,38	0,15	0,48	0,71	0,21	0,62
	0,29	0,10	0,15	0,39	0,27	0,19	0,49	0,48	0,15	0,56
sce04145	0,65	0,35	0,20	0,41	0,33	0,22	0,37	0,71	0,29	0,62
	0,30	0,11	0,22	0,56	0,29	0,22	0,54	0,50	0,14	0,63
sce04146	0,37	0,20	0,20	0,32	0,45	0,12	0,34	0,68	0,25	0,69
	0,26	0,11	-1	0,73	0,26	-1	0,81	0,55	0,05	0,50

Tabla B.1: Similitudes generadas por GFD, GS^2, Resnik y Wang en las tres ontologías GO

Bibliografía

[1] H. Abdi. *Bonferroni and Šidák corrections for multiple comparisons*. 2007.

[2] B. Adryan and R. Schuh. Gene-ontology-based clustering of gene expression data. *Bioinformatics*, 20:2851–2852, 2004.

[3] S. Aerts, P. Van Loo, Y. Moreau, and B. De Moor. A genetic algorithm for the detection of new cis-regulatory modules in sets of coregulated genes. *Bioinformatics*, 20(12):1974–1976, 2004.

[4] J. S. Aguilar-Ruiz. Shifting and scaling pattern from gene expression data. *Bioinformatics*, 21(20):3840–3845, 2005.

[5] J. S. Aguilar Ruiz, D. S. Rodriguez Baena, N. Diaz-Diaz, and I. A. Nepomuceno Chamorro. CarGene: Characterisation of sets of genes based on metabolic pathways analysis. *Int. J. Data Min. Bioinformatics*, 5(5):558–573, Oct. 2011.

[6] J. S. Aguilar Ruiz, D. S. Rodriguez Baena, N. Diaz-Diaz, I. A. Nepomuceno Chamorro, and F. Gomez Vela. Caracterización de un conjunto de genes basado en el análisis de pathways metabólicos. In *Actas del V Simposio de Teoria y Aplicaciones de Minería de Datos (TAMIDA)*, volume 5, pages 558–573, Inderscience Publishers, Geneva, SWITZERLAND, Oct. 2010. Inderscience Publishers.

[7] F. Al-Shahrour, R. Díaz-Uriarte, and J. Dopazo. Fatigo: a web tool for finding significant associations of gene ontology terms with groups of genes. *Bioinformatics*, 20(4):578–580, 2004.

[8] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96:6745—-6750, 1999.

[9] E. Altermann and T. Klaenhammer. Pathwayvoyager: pathway mapping using the kyoto encyclopedia of genes and genomes (kegg) database. *BMC Genomics*, 6(1):60–67, 2005.

[10] D. G. Altman. *Practical Statistics for Medical Research (2nd edition)*. Chapman & Hall, 2008.

[11] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, 1990.

[12] G. C. Anagnostopoulos and M. Georgiopoulos. Ellipsoid art and artmap for incremental unsupervised and supervised learning. In K. L. Priddy, P. E. Keller, and P. J. Angeline, editors, *Applications and Science of Computational Intelligence IV*, volume 4390, pages 293–304. SPIE, 2001.

[13] S. Ananiadou and J. Mcnaught. *Text Mining for Biology And Biomedicine*. Artech House, Inc., Norwood, MA, USA, 2005.

[14] A. Andreeva, D. Howorth, J.-M. M. Chandonia, S. E. Brenner, T. J. Hubbard, C. Chothia, and A. G. Murzin. Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Research*, 36(suppl 1):D419–D425, Jan. 2008.

[15] A. V. Antonov, T. Schmidt, Y. Wang, and H. W. Mewes. ProfCom: a web tool for profiling the complex functionality of gene groups identified from high-throughput data. *Nucleic Acids Research*, 36(suppl 2):W347–W351, July 2008.

[16] K. Aoki, A. Yamaguchi, N. Ueda, T. Akutsu, H. Mamitsuka, S. Goto, and M. Kanehisa. Kcam (kegg carbohydrate matcher): a software tool for analyzing the structures of carbohydrate sugar chains. *Nucleic Acids Res*, 32:267–272, 2004.

[17] M. I. Arnone and E. H. Davidson. The hardwiring of development: organization and function of genomic regulatory systems. *Development*, 124(10):1851–1864, 1997.

[18] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, and et al. Gene ontology: tool for the unification of biology. The Gene Ontology. *Nature Genetics*, 25:25–29, 2000.

[19] M. Ashkenazi, G. D. D. Bader, A. Kuchinsky, M. Moshelion, and D. J. J. States. Cytoscape esp: simple search of complex biological networks. *Bioinformatics*, 24(12):1465–1466, 2008.

[20] F. Aurenhammer and R. Klein. *Voronoi Diagrams*. In J.-R. Sack and J. Urrutia,editors, Handbook of Computational Geometry. Elsevier Science Publishers B.V., Amsterdan, 1999.

[21] F. Azuaje. A cluster validity framework for genome expression data. *Bioinformatics*, 18(2):319–320, 2002.

[22] F. Azuaje, F. Al-Shahrour, and J. Dopazo. Ontology-driven approaches to analyzing data in functional genomics. *Methods in Molecular Biology*, 316:67–86, 2005.

[23] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Research*, 28(1):45–48, 2000.

[24] P. Baldi and S. a. Brunak. *Bioinformatics: The Machine Learning Approach, Second Edition (Adaptive Computation and Machine Learning)*. The MIT Press, 2 edition, 2001.

[25] G. Ball and D. Hall. A clustering technique for summarizing multivariate data. *Behaviorial Sciences*, 12(2):153–155, 1967.

[26] W. C. Barker, J. S. Garavelli, H. Huang, P. B. Mcgarvey, B. C. Orcutt, G. Y. Srinivasarao, C. Xiao, L.-S. L. Yeh, R. S. Ledley, J. F. Janda, F. Pfeiffer, H.-W. Mewes, A. Tsugita, and C. Wu. The protein information resource (pir). *Nucleic Acids Research*, 28(1):41–44, 2000.

[27] D. Barrell, E. Dimmer, R. P. Huntley, D. Binns, C. O'Donovan, and R. Apweiler. The GOA database in 2009–an integrated Gene Ontology Annotation resource. *Nucleic acids research*, 37(Database issue):D396–403, Jan. 2009.

[28] P. R. Barrero. Aplicaciones de la técnica de microarrays en ciencias biomédicas: presente y futuro. *Química Viva*, 3(4):1–10, 2005.

[29] H. Bastos, D. Faria, C. Pesquita, and A. O. Falcão. Using GO terms to evaluate protein clustering. *BioOntologies SIG at ISMB/ECCB - 15th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 2007.

[30] A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. R. Eddy, S. Griffiths-Jones, K. L. Howe, M. Marshall, and E. L. L. Sonnhammer. The Pfam Protein Families Database. *Nucleic Acids Research*, 30(1):276–280, Jan. 2002.

[31] T. Beissbarth and T. Speed. GOstat: find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics*, 20:1464–1465, 2004.

[32] A. Bellaachia, D. Portnoy, Y. Chen, and A. Elkahloun. E-cast: A data mining algorithm for gene expression data. In *In Workshop on Data Mining in Bioinformatics*, pages 49–54, 2002.

[33] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: The order preserving submatrix problem. In *6th International Conference on Computacional Biology, RECOMB*, pages 49–57, 2002.

[34] A. Ben-Dor, R. Shamir, and Y. Z. Clustering gene expression pattern. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[35] F. M. Ben-Dor, A. and Z. Yakhini. Overabundance analysis and class discovery in gene expression data. Technical report, Agilent Laboratories, Palo Aeto, 2002.

[36] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing.*, pages 6–17, 2002.

[37] Y. Ben-Shaul, H. Bergman, and H. Soreq. Identifying subtle interrelated changes in functional gene categories using continuous measures of gene expression. *Bioinformatics*, 21(7):1129–1137, 2005.

[38] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[39] Y. Benjamini and D. Yekutieli. Contolling the false discovery rate in multiple testing under dependency. *Ann.Stat.*, 29:1165–1188, 2001.

[40] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. Genbank. *Nucleic Acids Research*, 36(Database issue):D25–D30, 2008.

[41] S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large–scale gene expression data. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 67(3 Pt 1):0319020, 2003.

[42] H. M. Berman, T. Battistuz, T. N. Bhat, W. F. Bluhm, P. E. Bourne, K. Burkhardt, Z. Feng, G. L. Gilliland, L. Iype, S. Jain, P. Fagan, J. Marvin, D. Padilla, V. Ravichandran, B. Schneider, N. Thanki, H. Weissig, J. D. Westbrook, and C. Zardecki. The Protein Data Bank. *Acta Crystallogr D Biol Crystallogr*, 58(Pt 6 No 1):899–907, June 2002.

[43] G. Berriz, O. King, B. Bryan, C. Sander, and F. Roth. Characterizing gene sets with funcassociate. *Bioinformatics*, 19:2502–2504, 2003.

[44] G. Berriz, O. King, B. Bryant, C. Sander, and F. Roth. Characterizing gene sets with FuncAssociate. *Bioinformatics*, 19(18):2502–2504, 2003.

[45] J. Bezdek and N. Pal. Some new indexes of cluster validity. *IEEE Trans. Syst. Man Cybernet.*, 28:301–315, 1998.

[46] H. Bhaskar, D. C. Hoyle, and S. Singh. Machine learning in bioinformatics: A brief survey and recommendations for practitioners. *Computers in Biology and Medicine*, 36(10):1104–1125, 2006.

[47] D. R. Bickel. Robust cluster analysis of microarray gene expression data with the number of clusters determined biologically. *Bioinformatics*, 19(7):818–824, 2003.

[48] G. Bindea, B. Mlecnik, H. Hackl, P. Charoentong, M. Tosolini, A. Kirilovsky, W.-H. Fridman, F. Pagès, Z. Trajanoski, and J. Galon. ClueGO: a Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. *Bioinformatics*, 25(8):1091–1093, Apr. 2009.

[49] M. e. a. Bittner. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, 406:536–540, 2000.

[50] J. Bland and D. Altman. Multiple significance tests - the bonferroni method. *Brittish Medical Journal*, 310:170, 1995.

[51] E.-J. Blom, D. W. J. Bosman, S. A. F. T. van Hijum, R. Breitling, L. Tijsma, R. Silvis, J. B. T. M. Roerdink, and O. P. Kuipers. Fiva: Functional information viewer and analyzer extracting biological knowledge from transcriptome data of prokaryotes. *Bioinformatics*, 23(9):1161–1163, 2007.

[52] J. Bockhorst, M. Craven, D. Page, J. Shavlik, and J. Glasner. A bayesian network approach to operon prediction. *Bioinformatics*, 19(10):1227–1235, 2003.

[53] N. Bolshakova and F. Azuaje. Cluster validation techniques for genome expression data. *Signal Process.*, 83(4):825–833, 2003.

[54] N. Bolshakova, F. Azuaje, and P. Cunningham. An integrated tool for microarray data clustering and cluster validity assessment. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 133–137, New York, NY, USA, 2004. ACM.

[55] A. Boorsma, B. C. Foat, D. Vis, F. Klis, and H. J. Bussemaker. T-profiler: scoring the activity of predefined groups of genes using gene expression data. *Nucleic Acids Res*, 33(Web Server issue):W592–W595, July 2005.

[56] S. Bornholdt. Boolean network models of cellular regulation: prospects and limitations. *Journal of the Royal Society Interface*, 5:S85–S94, 2008.

[57] J. M. Bower and H. Bolouri. *Computational Modeling of Genetic and Biochemical Networks (Computational Molecular Biology)*. The MIT Press, 2004.

[58] M. Brameier and C. Wiuf. Co-clustering and visualization of gene expression data and gene ontology terms for Saccharomyces cerevisiae using self-organizing maps. *J. of Biomedical Informatics*, 40:160–173, Apr. 2007.

[59] M. D. Brazas, D. S. Yim, J. T. Yamada, and B. F. F. Ouellette. The 2011 bioinformatics links directory update: more resources, tools and databases and features to empower the bioinformatics community. *Nucleic Acids Research*, 39(suppl 2):W3–W7, July 2011.

[60] J. Breckenridge. Replicating cluster analysis: method, consistency and validity. *Multivar. Behav. Res.*, 24:147–161, 1989.

[61] R. Breitling, A. Amtmann, and P. Herzyk. Iterative group analysis (iga): A simple tool to enhance sensitivity and facilitate interpretation of microarray experiments. *BMC Bioinformatics*, 5(1):34, 2004.

[62] S. Brohee and J. van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7:488+, November 2006.

[63] A. J. Butte and I. S. Kohane. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 418–429, 2000.

[64] T. Byrt, J. Bishop, and J. Carlin. Bias, prevalence and kappa. *J Clin Epidemiol*, 46(5):423–9, 1993.

[65] Z. Cai, X. Mao, S. Li, and L. Wei. Genome comparison using gene ontology (go) with statistical testing. *BMC Bioinformatics*, 7:374+, August 2006.

[66] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. Madden. BLAST+: architecture and applications. *BMC Bioinformatics*, 10(1):421+, 2009.

[67] S. Carbon, A. Ireland, C. J. J. Mungall, S. Shu, B. Marshall, S. Lewis, the AmiGO Hub, and the Web Presence Working Group. Amigo: online access to ontology and annotation data. *Bioinformatics*, 25(2):288–289, November 2008.

[68] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[69] R. Careter, I. Dubchak, and S. Holbrook. A computational approach to identify genes for functional rnas in genomic sequence. *Nucelic Acids Research*, 29(19):3928–3938, 2001.

[70] P. Carmona-Saez, M. Chagoyen, F. Tirado, J. M. Carazo, and A. Pascual-Montano. Genecodis: a web-based tool for finding significant concurrent annotations in gene lists. *Genome Biology*, 8:R3, 2007.

[71] G. A. Carpenter, S. Grossberg, David, and B. Rosen. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Network*, 4:759–771, 1991.

[72] C. Castillo-Davis and D. Hartl. Genemerge-post-genomic analysis,data mining, and hypothesis testing. *Bioinformatics*, 19:891–892, 2003.

[73] J. Chabalier, J. Mosser, and A. Burgun. A transversal approach to predict gene product networks from ontology-based similarity. *BMC Bioinformatics*, 8:235+, July 2007.

[74] A. A. Chatziioannou and P. Moulos. Exploiting Statistical Methodologies and Controlled Vocabularies for Prioritized Functional Analysis of Genomic Experiments: the StRAnGER Web Application. *Frontiers in neuroscience*, 5:8, 2011.

[75] C. Chen. *Information Visualization*. Springer, July 2004.

[76] Y. Cheng and G. Church. Biclustering of expression data. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology ; ISMB*, pages 93–103, 2000.

[77] R. Cho and et al. Transcriptional regulation and funciton dring the human cell cycle. *Nat.Genet.*, 27:48–54, 2001.

[78] T. U. Consortium. Ongoing and future developments at the Universal Protein Resource. *Nucleic Acids Research*, 39(suppl 1):D214–D219, Jan. 2011.

[79] R. M. Cormack. A review of classificiation. *Journal Royal Statistical Society, Series A,*, 134:321–367, 1971.

[80] F. Couto and M. Silva. Disjunctive shared information between ontology concepts: application to Gene Ontology. *Journal of Biomedical Semantics*, 2(5), 2011.

[81] F. Couto, M. Silva, and P. Coutinho. Implementation of a functional semantic similarity measure between gene-products. DI/FCUL TR 03–29, Department of Informatics, University of Lisbon, November 2003.

[82] F. Couto, M. Silva, and P. Coutinho. Measuring semantic similarity between Gene Ontology terms. *Data and Knowledge Engineering*, 61(1):137–152, 2007.

[83] A. L. Cuff, I. Sillitoe, T. Lewis, A. B. Clegg, R. Rentzsch, N. Furnham, M. Pellegrini-Calace, D. Jones, J. Thornton, and C. A. Orengo. Extending CATH: increasing coverage of the protein structure universe and linking structure with function. *Nucleic Acids Research*, 39(suppl 1):D420–D426, Jan. 2011.

[84] F. d'Alché Buc. *Bioinformatics using Computational Intelligence Paradigms*, volume 176 of *Studies in Fuzziness and Soft Computing*, chapter Class Prediction with Microarray Datasets, pages 93–117. Springer, 2005.

[85] S. Datta and S. Datta. Comparison and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, 19:459–466, 2003.

[86] S. Datta and S. Datta. Evaluation of clustering algorithms for gene expression data. *BMC bioinformatics*, 7(Supplement 4):S17, 2006.

[87] S. Datta and S. Datta. Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. *BMC bioinformatics*, 7:397, 2006.

[88] D. Davies and D. Bouldin. A cluster separation measure. *IEEE Trans. Pattern Anal. Machine Intell.*, 1:224–227, 1979.

[89] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[90] F. De Smet, J. Mathys, K. Marchal, G. Thijs, B. De Moor, and Y. Moreau. Adaptive quality-based clustering of gene expression profiles. *Bioinformatics*, 18(5):735–746, 2002.

[91] G. J. Dennis, B. Sherman, D. Hosack, J. Yang, W. Gao, H. Lane, and R. Lempicki. DAVID: Database for annotation, visualization, and integrated discovery. *Genome Biol.*, 4:3, 2003.

[92] N. Diaz-Diaz and J. Aguilar Ruiz. GO-based Functional Dissimilarity of Gene Sets. *BMC Bioinformatics*, 12(1):360+, Sept. 2011.

[93] N. Diaz Diaz, F. Gomez Vela, J. Garcia Gutierrez, and J. S. Aguilar Ruiz. Gene–gene interaction based clustering method for microarray data. *International Conference on Intelligent Systems Design and Application (ISDA)*, page In Press, 2011.

[94] N. Diaz-Diaz, F. Gomez Vela, D. S. Rodriguez Baena, and J. S. Aguilar Ruiz. Gene Regulatory Networks Validation Framework based in KEGG. *Lecture Notes in Artificial Intelligence*, pages 279–286, 2011.

[95] N. Diaz Diaz, D. Rodriguez Baena, I. Nepomuceno, and J. S. Aguilar Ruiz. Neighborhood-based Clustering of Gene-Gene Interactions. *Lecture Notes in Computer Science*, pages 1111–1120, 2006.

[96] C. Ding and C. He. K-nearest neighbor consistency in data clustering: incorporating local information into global optimization. *In Haddad,H.M. et al. (eds), Proceedings of the 2004 ACM Symposium on Applied Computing*, ACM Press, New York:584–589, 2004.

[97] S. Draghici. Data Analysis Tools for DNA Microarrays. *Chapman and Hall-CRC press*, 2003.

[98] S. Draghici and et al. Global functional profiling of gene expression. *Genomics*, 81:98–104, 2003.

[99] S. Draghici and et al. Onto-Tools, the toolkit of the modern biologist: onto-Express, Onto-Compare, Onto-Design and Onto-Translate. *Nucelic Acids Res.*, 31:3775–3781, 2003.

[100] Z. Du, X. Zhou, Y. Ling, Z. Zhang, and Z. Su. agriGO: a GO analysis toolkit for the agricultural community. *Nucleic Acids Research*, 38(suppl 2):W64–W70, July 2010.

[101] D. Duncan, N. Prodduturi, and B. Zhang. WebGestalt2: an updated and expanded version of the Web-based Gene Set Analysis Toolkit. *BMC Bioinformatics*, 11(Suppl 4):P10+, 2010.

[102] J. Dunn. Well separated clusters and fuzzy partitions. *J. Cybernet.*, 4:95–104, 1974.

[103] EASEonline. http://david.niaid.nih.gov/david/ease.htm.

[104] E. Eden. *Discovering Motifs in Ranked Lists of DNA Sequences*. PhD thesis, Israel Institute of Technology, 2007.

[105] E. Eden, D. Lipson, S. Yogev, and Z. Yakhini. Discovering Motifs in Ranked Lists of DNA Sequences. *PLoS Comput Biol*, 3(3):e39+, Mar. 2007.

[106] E. Eden, R. Navon, I. Steinfeld, D. Lipson, and Z. Yakhini. GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC Bioinformatics*, 10(1):48+, Feb. 2009.

[107] A. Edwards. The Correlation Coefficient. *W.H. Freeman*, 18:33–46, 1967.

[108] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman and Hall, 1993.

[109] eGOn. http://nova2.idi.ntnu.no/egon.

[110] M. B. Eisen and P. O. Brown. DNA arrays for analysis of gene expression. *Methods Enzymol*, 303:179–205, 1999.

[111] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95(25):14863–14868, December 1998.

[112] J. Ernst and Z. B. Joseph. Stem: a tool for the analysis of short time series gene expression data. *BMC Bioinformatics*, 7:191, 2006.

[113] J. Ernst, O. Vainas, C. T. Harbison, I. Simon, and Z. Bar-joseph. Reconstructing dynamic regulatory maps. *Molecular Systems Biology*, 3:74, 2007.

[114] S. Even. *Graph Algorithms*. Computer Science Press, RockVille, Maryland, 1979.

[115] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol*, 5(1):e8+, 2007.

[116] D. Faria, C. Pesquita, F. M. Couto, A. O. Falcão, D. Faria, F. M. Couto, C. Pesquita, and A. O. Falcão. Proteinon: A web tool for protein semantic similarity, 2007.

[117] V. Filkov. *Handbook of Computational Molecular Biology*. CRCPress, Chapman & Hall, 2005.

[118] L. Fisher and G. van Belle. Biostatistics: A methodology for health sciences. *John Wiley and Sons, Inc. New York*, 1993.

[119] P. Flicek, B. L. L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, T. Down, S. C. C. Dyer, T. Eyre, S. Fitzgerald, J. Fernandez-Banet, S. Gräf, S. Haider, M. Hammond, R. Holland, K. L. L. Howe, K. Howe, N. Johnson, A. Jenkinson, A. Kähäri, D. Keefe, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, K. Megy, P. Meidl, B. Overduin, A. Parker, B. Pritchard, A. Prlic, S. Rice, D. Rios, M. Schuster, I. Sealy, G. Slater, D. Smedley, G. Spudich, S. Trevanion, A. J. J. Vilella, J. Vogel, S. White, M. Wood, E. Birney, T. Cox, V. Curwen, R. Durbin, X. M. M. Fernandez-Suarez, J. Herrero, T. J. P. J. Hubbard, A. Kasprzyk, G. Proctor, J. Smith, A. Ureta-Vidal, and S. Searle. Ensembl 2008. *Nucleic Acids Res*, 36(Database issue):D707–D714, November 2007.

[120] P. Flicek, M. R. Amode, D. Barrell, K. Beal, S. Brent, Y. Chen, P. Clapham, G. Coates, S. Fairley, S. Fitzgerald, L. Gordon, M. Hendrix, T. Hourlier, N. Johnson, A. Kähäri, D. Keefe, S. Keenan, R. Kinsella, F. Kokocinski, E. Kulesha, P. Larsson, I. Longden, W. McLaren, B. Overduin, B. Pritchard, H. S. Riat, D. Rios, G. R. S. Ritchie, M. Ruffier, M. Schuster, D. Sobral, G. Spudich, Y. A. Tang, S. Trevanion, J. Vandrovcova, A. J. Vilella, S. White, S. P. Wilder, A. Zadissa, J. Zamora, B. L. Aken, E. Birney, F. Cunningham, I. Dunham, R. Durbin, X. M. Fernández-Suarez, J. Herrero, T. J. P. Hubbard, A. Parker, G. Proctor, J. Vogel, and S. M. J. Searle. Ensembl 2011. *Nucleic Acids Research*, 39(suppl 1):D800–D806, Jan. 2011.

[121] J. Fridlyand and S. Dudoit. Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method. Technical report, Department of Statistics, Berkeley, 2001.

[122] N. Friedman and N. Kamisnki. Statistical methods for analyzing gene expression data for cancer research. *Ernst Schering Res Found Workshop*, 38:109–131, 2002.

[123] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.

[124] M. E. Garber, O. G. Troyanskaya, K. Schluens, S. Petersen, Z. Thaesler, M. Pacyna-Gengelbach, M. van de Rijn, G. D. Rosen, C. M. Perou, R. I. Whyte, R. B. Altman, P. O. Brown, D. Botstein, and I. Petersen. Diversity of gene expression in adenocarcinoma of the lung. *Proc Natl Acad Sci U S A 2001*, 98(24):13784–9, 2001.

[125] J. García Gutiérrez, N. Díaz Díaz, D. S. Rodríguez Baena, and F. Martínez Álvarez. Software y técnicas de validación de conocimiento en bioinformática. In *Extracción y Validación de Conocimiento en Bases de Datos Biomédicas*, pages 75–84. ISBN-13:978-84-611-8854-3, 2007.

[126] I. Gat-Viks, R. Sharan, and R. Shamir. Scoring clustering solutions by their biological relevance. *Bioinformatics*, 19(18):2381–2389, 2003.

[127] R. Gentleman. Visualizing and distances using go. 2004.

[128] R. Gentleman, V. Carey, D. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Yang, and J. Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80+, 2004.

[129] G. Getz, E. Levine, and E. Domany. Copule two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Sciences*, 97(22):12079–84, 2000.

[130] F. Gibbons and F. Roth. Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Res.*, 12:1574–1581, 2002.

[131] F. D. Gibbons and F. P. Roth. Methods judging the quality of gene expression-based clustering methods using gene annotation. *Genome Research*, 12:1574–1581, 2002.

[132] F. Gómez Vela, N. Diaz Diaz, and J. S. Aguilar Ruiz. Gene Networks Validation based on Metabolic Pathways. *IEEE International Conference on Bioinformatics and Bioengineering*, pages 9–14, 2011.

[133] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield,

and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, October 1999.

[134] G. Gregorcic and G. Lightbody. Nonlinear system identification: From multiple-model networks to gaussian processes. *Engineering Applications of Artificial Intelligence*, 21(7):1035–1055, 2008.

[135] X. Guo. Gene ontology-based semantic similarity measures. World Wide Web electronic publication, 2008.

[136] X. Guo, R. Liu, C. D. Shriver, H. Hu, and M. N. Liebman. Assessing semantic similarity measures for the characterization of human regulatory pathways. *Bioinformatics*, 22(8):967–973, 2006.

[137] J. Hagen. The origin of bioinformatics. *Nature reviews. Genetics*, 1(3):231–236, 2001.

[138] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001.

[139] M. e. a. Halkidi. On clustering validation techniques. *IJ. Intell. Inform. Syst.*, 17:107–145, 2001.

[140] J. Han and M. Kamber. *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, September 2000.

[141] J. Handl, J. Knowles, and D. B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[142] J. Handl and J. D. Knowles. Exploiting the trade-off - the benefits of multiple objectives in data clustering. In *EMO*, pages 547–560, 2005.

[143] J. Hao and J. B. Orlin. A faster algorithm for finding the minimum cut in a directed graph. *Journal of Algorithms*, 17:424–446, 1994.

[144] A. Hartemink, D. Gifford, T. Jaakkola, and R. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Proceeding of the Pacific Symposium on Biocomputing*, volume 6, pages 422–433, 2001.

[145] E. Hartuv and R. Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4–6):175–181, 2000.

[146] R. Hathaway and J. Bezdek. Fuzzy c-means clustering of incomplete data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 31(5):735–744, 2001.

[147] M. Hecker, S. Lambeck, S. Toepfer, E. van Someren, and R. Guthke. Gene regulatory network inference: Data integration in dynamic models – a review. *Biosystems*, 96(1):86–103, April 2009.

[148] D. Heckerman. A tutorial on learning with bayesian networks. Technical report, Microsoft Research, MSR–TR–95–06, 1995.

[149] C. Henegar, J. Tordjman, V. Achard, D. Lacasa, I. Cremer, M. Guerre-Millo, C. Poitou, A. Basdevant, V. Stich, N. Viguerie, D. Langin, P. Bedossa, J.-D. Zucker, and K. Clement. Adipose tissue transcriptomic signature highlights the pathological relevance of extracellular matrix in human obesity. *Genome Biology*, 9:R14+, January 2008.

[150] R. Herwig, A. J. Poustka, C. Müller, C. Bull, H. Lehrach, and J. O'Brien. Large-scale clustering of cdna-fingerprinting data. *Genome Research*, 9(11):1093–1105, November 1999.

[151] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9(11):1106–1115, 1999.

[152] M. A. Hibbs, N. C. Dirksen, K. Li, and O. G. Troyanskaya. Visualization methods for statistical analysis of microarray clusters. *BMC Bioinformatics*, 6:115, 2005.

[153] A. Hill, E. Brown, M. Whitley, G. T. Kellogg, C. Hunter, and D. Slonim. Evaluation of normalization procedures for oligonucleotide array data based on spiked crna controls. *Genome Biology*, 2(12):research0055, 2001.

[154] Y. Hockger and A. Tamhane. *Multiple Comparison Procedures*. John Wiley and Sons,Inc., 1987.

[155] B. Holland and M. Copenhaver. An improved sequentially rejective bonferroni test procedure. *Biometrica*, 43:417–423, 1987.

[156] S. Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scand J Statist*, 6:65–70, 1979.

[157] S. Holm. A simple sequentially rejective multiple test procedure. *Scand. J. Stat.*, 6:65–70, 1979.

[158] D. Huang and W. Pan. Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Bioinformatics*, 22(10):1259–1268, 2006.

[159] D. W. Huang, B. T. Sherman, Q. Tan, J. R. Collins, G. W. Alvord, J. Roayaei, R. Stephens, M. W. Baseler, C. H. Lane, and R. A. Lempicki. David

gene functional classification tool: A novel biological module-centric algorithm to functionally analyze large gene list. *Genome Biology*, 8:R183+, September 2007.

[160] W. Huber, A. von Heydebreck, H. Sültmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18 Suppl 1:S96–S104, 2002.

[161] A. Hubert. Comparing partitions. *J. Classif.*, 2:193–198, 1985.

[162] S. Hunter, R. Apweiler, T. K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bork, U. Das, L. Daugherty, L. Duquenne, R. D. Finn, J. Gough, D. Haft, N. Hulo, D. Kahn, E. Kelly, A. Laugraud, I. Letunic, D. Lonsdale, R. Lopez, M. Madera, J. Maslen, C. McAnulla, J. McDowall, J. Mistry, A. Mitchell, N. Mulder, D. Natale, C. Orengo, A. F. Quinn, J. D. Selengut, C. J. A. Sigrist, M. Thimma, P. D. Thomas, F. Valentin, D. Wilson, C. H. Wu, and C. Yeats. Interpro: the integrative protein signature database. *Nucl. Acids Res.*, 37(suppl_1):D211–215, January 2009.

[163] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. *Nature Genetics*, 31(4):370–377, 2002.

[164] V. R. Iyer, M. B. Eisen, D. T. Ross, G. Schuler, T. Moore, J. C. Lee, J. M. Trent, L. M. Staudt, J. Hudson, M. S. Boguski, D. Lashkari, D. Shalon, D. Botstein, and P. O. Brown. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283(5398):83–87, 1999.

[165] S. Jaccard. Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat.*, 44:223–270, 1908.

[166] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall Advanced Reference Series, 1998.

[167] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[168] N. Jardine and R. Sibson. *Mathematical Taxonomy.* John Wiley and Sons., 1971.

[169] D. Jiang, C. Tang, and A. Zhang. Cluster Analysis for Gene Expression Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1370–1386, 2004.

[170] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33, 1997.

[171] S. Jonnalagadda and R. Srinivasan. Nifti: An evolutionary approach for finding number of clusters in microarray data. *BMC Bioinformatics*, 10(1):40+, January 2009.

[172] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, 28:27–30, 2000.

[173] M. Kanehisa, S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa. KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Research*, 38(suppl 1):D355–D360, Jan. 2010.

[174] M. Kanehisa, S. Goto, M. Hattori, K. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res.*, 34:D354–357, 2006.

[175] N. Kaplan and M. Linial. Automatic detection of false annotations via binary property clustering. *BMC Bioinformatics*, 6:46, 2005.

[176] S. Kauffman. Homeostasis and differentiation in random genetic control networks. *Nature*, 224:177–178, 1969.

[177] S. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467, 1969.

[178] S. Kauffman and K. Glass. The logical analysis of continuous, nonlinear biochemical control networks. *Journal of Theoretical Biology*, 39:103–129, 1973.

[179] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics)*. Wiley-Interscience, March 2005.

[180] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[181] A. Keller, C. Backes, and H. P. Lenhof. Computation of significance scores of unweighted gene set enrichment analyses. *BMC Bioinformatics*, 8(1):290, 2007.

[182] M. Kerr and G. Churchill. Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. *Proceedings of the National Academy of Sciences of the United States of America*, 98:8961—-8965, 2001.

[183] P. Khatri and S. Drăghici. Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, 21(18):3587–3595, September 2005.

[184] C. Klukas and F. Schreiber. Dynamic exploration and editing of KEGG pathway diagrams. *Bioinformatics*, 23(3):344, 2007.

[185] I. Kohane. Bioinformatics and clinical informatics: the imperative to collaborate. *Journal of the American Medical Informatics Association*, 7(5):512–516, 2000.

[186] T. Kohonen. *Self–Organizing Maps*. Spring Verlag, Berlin, 1997.

[187] M. Krallinger, R. A. Erhardt, and A. Valencia. Text–mining approaches in molecular biology and biomedicine. *Drug Discovery Today*, 10(6):439–445, 2005.

[188] A. Krishnan, A. Giuliani, and M. Tomita. Indeterminacy of reverse engineering of gene regulatory networks: The curse of gene elasticity. *PLoS ONE*, 2(6):e562+, 2007.

[189] D. Kroese, T. Taimre, and Z. Botev. *Handbook of Monte Carlo Methods*. John Wiley and Sons, 2011.

[190] C. Kuiken, B. Foley, T. Leitner, C. Apetrei, B. Hahn, I. Mizrachi, J. Mullins, A. Rambaut, S. Wolinsky, and Korber, editors. *HIV Sequence Compendium 2010*. Published by Theoretical Biology and Biophysics Group, Los Alamos National Laboratory, NM, LA-UR 10-03684.

[191] M. Langaas. Statistical hypothesis testing of associa-tion between two reporter lists within the go-hierarchy. Technical report, Department of Mathematical Sciences, Norwegian University of Science and Technology, 2005.

[192] T. e. a. Lange. Stability-based validation of clustering solutions. *Neural comput.*, 16:1299–1323, 2004.

[193] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, and V. Robles. Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112, 2006.

[194] H. K. Lee, W. Braynen, K. Keshav, and P. Pavlidis. Erminej: tool for functional analysis of gene expression data sets. *BMC Bioinformatics*, 6:269, 2005.

[195] H. K. Lee, A. K. Hsu, J. Sajdak, J. Qin, and P. Pavlidis. Coexpression analysis of human genes across many microarray data sets. *Genome Research*, 14:1085–1094, 2004.

[196] S. G. Lee, J. U. Hur, and Y. S. Kim. A graph-theoretic modeling on go space for biological interpretation of gene clusters. *Bioinformatics*, 20(3):381–388, 2004.

[197] E. Lehmann and H. D'Abrera. *Nonparametrics: Statistical Methods Based on Ranks.* Prentice-Hall, 1998.

[198] Z. Lei and Y. Dai. Assessing protein similarity with gene ontology and its use in subnuclear localization prediction. *BMC Bioinformatics*, 7:491, 2006.

[199] A. M. Lesk. *Introduction to Bioinformatics, Third Edition*. Oxford University Press, 2008.

[200] E. Levine and E. Domany. Resampling method for unsupervised estimation of cluster validity. *Neural Comput.*, 13:2573–2593, 2001.

[201] A. Lewin and I. C. Grieve. Grouping gene ontology terms to improve the assessment of gene set enrichment in microarray data. *BMC Bioinformatics*, 7:426+, October 2006.

[202] C. Li and W. Wong. Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biol.*, 2:1–11, 2001.

[203] L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method. *Bioinformatics*, 17(12):1131–1142, 2001.

[204] S. Liang, S. Fuhrman, and R. Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *In Proceeding of the Pacific Symposium on Biocomputing*, pages 18–19, 1998.

[205] D. Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[206] T. Lindholm and F. Yellin. *The Java Virtual Machine Specification*. Addison-Wesley Pofessional. 2nd edition, 1999.

[207] H. Liu, Z. Z. Hu, and C. H. Wu. Dyngo: a tool for visualizing and mining of gene ontology and its associations. *BMC bioinformatics*, 6:201, 2005.

[208] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283, July 2003.

[209] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. Semantic similarity measures as tools for exploring the gene ontology. *Pac Symp Biocomput*, pages 601–612, 2003.

[210] J. MacQueen. Some methods for classification and analysis f multivariate observations. In *Proceeding of the 5th Berkeley Symposium on Mathematial Statistics and Probability*, pages 281–297, 1965.

[211] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[212] S. Maere, K. Heymans, and M. Kuiper. Bingo: a cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. *Bioinformatics*, 21(16):3448–3449, 2005.

[213] D. Maglott, J. Ostell, K. Pruitt, and T. Tatusova. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, 33:D54–D58, 2005.

[214] M. Man and et al. Power-sage: comparing statistical tests for sage experiments. *Bioinformatics*, 16:953–959, 2000.

[215] J. Mao and A. Jain. A self-organizing network for hyperellipsoidal clustering (hec). *IEEE Transactions on Neural Networks*, 7(1):16–29, 1996.

[216] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera, and A. Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics*, 7(Suppl 1):S7, 2006.

[217] D. Martin, C. Brun, E. Remy, P. Mouren, D. Thieffry, and B. Jacq. GOToolBox: functional analysis of gene datasets based on Gene Ontology. *Genome Biology*, 5(12):R101, 2004.

[218] S. Martin, Z. Zhang, A. Martino, and J.-L. Faulon. Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics*, 23(7):866–874, 2007.

[219] F. Martín Sánchez, G. López Campos, and N. Ibarrola de Andrés. Impacto de la bioinformática en las ciencias biomédicas. *Servicios de Salud: ¿estrategias o tecnologías?. Madrid: Editorial Médica Panamericana*, 1999.

[220] C. Mathé, M.-F. Sagot, T. Schlex, and P. Rouzé. Current methods of gene prediction, their strengths and weaknesses. *Nucelic Acids Research*, 30(19):4103–4117, 2002.

[221] L. e. a. McShane. Methods for assessing reproducibility of clustering patterns observed in analyses of microarray data. *Bioinformatics*, 18:1462–1469, 2002.

[222] P. Miller. Opportunities at the intersection of bioinformatics and health informatics: a case study. *Journal of the American Medical Informatics Association*, 7(5):431–438, 2000.

[223] B. Mirkin. *Mathematical Classification and Clustering*. Kluwer, 1996.

[224] M. Mistry and P. Pavlidis. Gene ontology term overlap as a measure of gene functional similarity. *BMC Bioinformatics*, 9:327+, August 2008.

[225] F. Mordelet and J.-P. Vert. Sirene: supervised inference of regulatory networks. *Bioinformatics*, 24(16):i76–82, 2008.

[226] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359, 1983.

[227] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, Apr. 1995.

[228] C. J. Needham, J. R. Bradford, A. J. Bulpitt, and D. R. Westhead. A primer on learning in bayesian networks for computational biology. *PLoS Comput Biol*, 3(8):e129+, 2007.

[229] O. Nelles. Nonlinear system identification. *Measurement Science and Technology*, 13(4):646, 2002.

[230] J. Newman and A. Weiner. L2l: a simple tool for discovering the hidden significance in microarray expression data. *Genome Biology*, 6(9):R81, 2005.

[231] M. O'Connell. Differential expression, class discovery and class prediction using s-plus and s+arrayanalyzer. *SIGKDD Explor. Newsl.*, 5(2):38–47, 2003.

[232] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, 27:29–34, 1999.

[233] G. J. Olsen, R. Overbeek, N. Larsen, T. L. Marsh, M. J. McCaughey, M. A. Maciukenas, W. M. Kuan, T. J. Macke, Y. Xing, and C. R. Woese. The Ribosomal Database Project. *Nucleic acids research*, 20 Suppl:2199–2200, May 1992.

[234] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. CATH–a hierarchic classification of protein domain structures. *Structure (London, England : 1993)*, 5(8):1093–1108, Aug. 1997.

[235] K. P and et al. Profiling gene expression using Onto-Express. *Genomics*, 79:266–270, 2002.

[236] V. Pareto. *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*. Manual of Political Economy, 1971 Translation of 1927 Edition., 1971.

[237] P. Pavlidis and P. Poirazi. Individualized markers optimize class prediction of microarray data. *BMC Bioinformatics*, 7:345–359, 2006.

[238] L. Perezleo Solorzano, R. Arencibia Jorge, C. Conill González, G. Achón Veloz, and J. A. Araújo Ruiz. Impacto de la bioinformática en las ciencias biomédicas. *ACIMED*, 11(4), 2003.

[239] C. Pesquita, D. Faria, H. Bastos, A. E. Ferreira, A. O. Falcão, and F. M. Couto. Metrics for GO based protein semantic similarity: a systematic evaluation. *BMC bioinformatics*, 9 Suppl 5:S4, 2008.

[240] C. Pesquita, D. Faria, A. O. Falcão, P. Lord, and F. M. Couto. Semantic similarity in biomedical ontologies. *PLoS Comput Biol*, 5(7):e1000443+, July 2009.

[241] P. Polisetty, E. Voit, and E. Gatzke. Identification of metabolic system parameters using global optimization methods. *Theoretical Biology and Medical Modelling*, 3(1):4+, 2006.

[242] M. Popescu, J. M. Keller, and J. A. Mitchell. Fuzzy measures on the gene ontology for gene product similarity. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 3(3):263–274, 2006.

[243] A. D. Pozo, F. Pazos, and A. Valencia. Defining functional distances over Gene Ontology. *BMC Bioinformatics*, 9:50, 2008.

[244] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, May 1 2006.

[245] E. Prifti, J.-D. Zucker, K. Clement, and C. Henegar. FunNet: an integrative tool for exploring transcriptional interactions. *Bioinformatics*, 24(22):2636–2638, Nov. 2008.

[246] E. Prifti, J.-D. Zucker, K. Clément, and C. Henegar. Interactional and functional centrality in transcriptional co-expression networks. *Bioinformatics*, 26(24):3083–3089, Dec. 2010.

[247] I. Priness, O. Maimon, and I. Ben-Gal. Evaluation of gene-expression clustering via mutual information distance measure. *BMC Bioinformatics*, 8:111+, March 2007.

[248] K. Prufer, B. Muetzel, H.-H. Do, G. Weiss, P. Khaitovich, E. Rahm, S. Paabo, M. Lachmann, and W. Enard. Func: a package for detecting significant associations between gene sets and ontological annotations. *BMC Bioinformatics*, 8:41, 2007.

[249] J. Quackenbush. Microarray data normalization and transformation. *Nature Genetics*, 32:496–501, 2002.

[250] W. Rand. Objective criteria for the evaluation of clustering methods. *O. J. Am. Stat. Assoc.*, 66:846–850, 1971.

[251] C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani. Modeling t-cell activation using gene expression profiling and state-space models. *Bioinformatics*, 20(9):1361–1372, 2004.

[252] J. Reimand, T. Arak, and J. Vilo. g:Profiler–a web server for functional interpretation of gene lists (2011 update). *Nucleic Acids Research*, 39(Web Server issue):W307–W315, June 2011.

[253] J. Reimand, M. Kull, H. Peterson, J. Hansen, and J. Vilo. g:profiler–a web-based toolset for functional profiling of gene lists from large-scale experiments. *Nucleic Acids Res*, 35(Web Server issue):W193–W200, July 2007.

[254] J. Reimand, L. Tooming, H. Peterson, P. Adler, and J. Vilo. GraphWeb: mining heterogeneous biological networks for gene modules with functional significance. *Nucleic acids research*, 36(Web Server issue):W452–W459, July 2008.

[255] P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[256] A. J. Richards, B. Muller, M. Shotwell, L. A. Cowart, B. Rohrer, and X. Lu. Assessing the functional coherence of gene sets with metrics based on the Gene Ontology graph. *Bioinformatics*, 26(12):i79–i87, June 2010.

[257] I. Rivals, L. Personnaz, L. Taing, and M.-C. Potier. Enrichment or depletion of a GO category within a class of genes: which test? *Bioinformatics*, 23(4):401–407, Feb. 2007.

[258] H. Romesburg. *Cluster Analysis for Researchers.* Belmont, 1984.

[259] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20:53–65, 1987.

[260] R. Ruiz, J. S. Aguilar-Ruiz, J. C. Riquelme, and N. Diaz-Diaz. Analysis of feature rankings for classification. *Lecture Notes in Computer Science, Spriner Verlag*, 3646:362–372, 2005.

[261] R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz. Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recogn.*, 39(12):2383–2392, 2006.

[262] T. Ruths, D. Ruths, and L. Nakhleh. GS2: an efficiently computable measure of GO-based similarity of gene sets. *Bioinformatics (Oxford, England)*, 25(9):1178–1184, May 2009.

[263] M. Schena, D. Shalon, R. Davis, and P. Brown. Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA microarray. *Science*, 70:467–470, 1995.

[264] A. Schlicker and M. Albrecht. Funsimmat: a comprehensive functional similarity database. *Nucl. Acids Res.*, 36(Database issue):D434–D439, October 2007.

[265] A. Schlicker, F. S. Domingues, J. Rahnenfuhrer, and T. Lengauer. A new measure for functional similarity of gene products based on gene ontology. *BMC Bioinformatics*, 7:302+, June 2006.

[266] J. Schuchhardt, D. Beule, A. Malik, E. Wolski, H. Eickhoff, H. Lehrach, and H. Herzel. Normalization strategies for cdna microarrays. *Nucleic Acids Research*, 28(10):E47, 2000.

[267] R. S. G. Sealfon, M. A. Hibbs, C. Huttenhower, C. L. Myers, and O. G. Troyanskaya. Golem: an interactive graph-based gene-ontology navigation and analysis tool. *BMC Bioinformatics*, 7:443+, October 2006.

[268] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–176, May 2003.

[269] J. L. Sevilla, V. Segura, A. Podhorski, E. Guruceaga, J. M. Mato, L. A. Martinez-Cruz, F. J. Corrales, and A. Rubio. Correlation between gene expression and go semantic similarity. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 2(4):330–338, 2005.

[270] N. H. Shah and N. V. Fedoroff. CLENCH: a program for calculating Cluster ENriCHment using the Gene Ontology. *Bioinformatics*, 20(7):1196–1197, 2004.

[271] D. Shalon, S. Smith, and P. Brown. A DNA microarray system for analyzing complex dna samples using two-color fluorescent probe hybridization. *Genome Res*, 6:639–645, 1996.

[272] R. Shamir, A. Maron-Katz, A. Tanay, C. Linhart, I. Steinfeld, R. Sharan, Y. Shiloh, and R. Elkon. EXPANDER–an integrative program suite for microarray data analysis. *BMC Bioinformatics*, 6:232, 2005.

[273] R. Shamir and R. Sharan. Algorithmic approaches to clustering gene expression data. In T. Jiang, T. Smith, Y. Xu, and M. Q. Zhang, editors, *Current Topics in Computational Biology*, pages 269–300. MIT press, 2001.

[274] R. Sharan, A. Maron-Katz, and R. Shamir. CLICK and EXPANDER: A System for Clustering and Visualizing Gene. *Bioinformatics*, 19(14):1797–1799, 2003.

[275] R. Sharan and R. Shamir. CLICK: A clustering algorithm with applications to gene expression analysis. In *Proceddings of the eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 307–316. AAAI Press, Menlo Park, CA, 2000.

[276] G. Sherlock. Analysis of large-scale gene expressiond data. *Curr Opin Immunol*, 12(2):201–205, 2000.

[277] I. Shmulevich, R. Dougherty, S. Kim, and W. Zhang. Probabilistic boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

[278] C. Sima, J. Hua, and S. Jung. Inference of gene regulatory networks using time–series data: A survey. *Current Genomics*, 10:416–429, 2009.

[279] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, 9(12):3273–97, Dec 1998.

[280] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig. The mutual information: Detecting and evaluating dependencies between variables. *Bioinformatics*, 18(supplement 2):S231–240, 2002.

[281] G. Stoesser, W. Baker, A. van den Broek, E. Camon, M. Garcia-Pastor, C. Kanz, T. Kulikova, R. Leinonen, Q. Lin, V. Lombard, R. Lopez, N. Redaschi, P. Stoehr, M. A. Tuli, K. Tzouvara, and R. Vaughan. The embl nucleotide sequence database. *Nucl. Acids Res.*, 30(1):21–26, January 2002.

[282] S. Swift, A. Tucker, V. Vinciotti, N. Martin, C. Orengo, X. Liu, and P. Kellam. Consensus clustering and functional interpretation of gene-expression data. *Genome Biology*, 5:R94, 2004.

[283] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences of the United States of America*, 96(6):2907–2912, March 1999.

[284] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18:136–144, 2002.

[285] A. Tanay, R. Sharan, and R. Shamir. *Biclustering Algorithms: A Survey*. In Handbook of Computational Molecular Biology, Edited by Srinivas Aluru, Chapman & Hall/CRC, Computer and Information Science Series, 2005.

[286] Y. Tao, L. Sam, J. Li, C. Friedman, and Y. A. Lussier. Information theory applied to the sparse gene ontology annotation network to predict novel gene function. *Bioinformatics*, 23(13):i529–538, July 2007.

[287] Y. Tateno, S. Miyazaki, M. Ota, H. Sugawara, and T. Gojobori. Dna data bank of japan (ddbj) in collaboration with mass sequencing teams. *Nucl. Acids Res.*, 28(1):24–26, January 2000.

[288] R. Tatusov, N. Fedorova, J. Jackson, A. Jacobs, B. Kiryutin, E. Koonin, D. Krylov, R. Mazumder, S. Mekhedov, A. Nikolskaya, B. S. Rao, S. Smirnov, A. Sverdlov, S. Vasudevan, Y. Wolf, J. Yin, and D. Natale. The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, 4(1):41+, Sept. 2003.

[289] R. L. Tatusov, M. Y. Galperin, D. A. Natale, and E. V. Koonin. The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Research*, 28(1):33–36, Jan. 2000.

[290] S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.

[291] R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563–585, December 1973.

[292] R. Tibshirani, G. Walther, D. Botstein, and P. Brown. Cluster validation by prediction strength. Technical report, Department of Statistics, Stanford University, CA., 2001.

[293] P. Törönen. Selection of informative clusters from hierarchical cluster tree with gene classes. *BMC Bioinformatics*, 5:32, 2004.

[294] P. Toronen, M. Kolehmainen, G. Wong, and E. Castren. Analysis of gene expression data using self–organizing maps. *FEBS Letters*, 451:142–146, 1999.

[295] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[296] S. Tweedie, M. Ashburner, K. Falls, P. Leyland, P. McQuilton, S. Marygold, G. Millburn, D. Osumi-Sutherland, A. Schroeder, R. Seal, H. Zhang, and T. F. Consortium. FlyBase: enhancing Drosophila Gene Ontology annotations. *Nucleic Acids Research*, 37(suppl 1):D555–D559, Jan. 2009.

[297] C. van Rijsbergen. *Information Retrieval.* 2nd edn. Butterworths, 1979.

[298] E. P. van Someren, B. L. T. Vaes, W. T. Steegenga, A. M. Sijbers, K. J. Dechering, and M. J. T. Reinders. Least absolute regression network analysis of the murine osteoblast differentiation network. *Bioinformatics*, 22(4):477–484, 2006.

[299] M. Vilela, I.-C. C. Chou, S. Vinga, A. T. T. Vasconcelos, E. O. Voit, and J. S. Almeida. Parameter optimization in s-system models. *BMC Systems Biology*, 2(1):35+, 2008.

[300] E. O. Voit. *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. Cambridge Universisty Press, Cambridge, New York, 2000.

[301] H. Wang, F. Azuaje, O. Bodenreider, and J. Dopazo. Gene expression correlation and gene ontology-based similarity: An assessment of quantitavie retaionships. In *Proceedings of the 2004 IEEE Sumposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 25–31, San Diego, CA, 2004.

[302] H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *ACM SIGMOD International Conference on Management of Data*, pages 394–405, 2002.

[303] J. Z. Z. Wang, Z. Du, R. Payattakool, P. S. S. Yu, and C.-F. F. Chen. A new method to measure the semantic similarity of go terms. *Bioinformatics*, 23(Issue 10):1274–1281, March 2007.

[304] D. Warde-Farley, S. L. Donaldson, O. Comes, K. Zuberi, R. Badrawi, P. Chao, M. Franz, C. Grouios, F. Kazi, C. T. Lopes, A. Maitland, S. Mostafavi, J. Montojo, Q. Shao, G. Wright, G. D. Bader, and Q. Morris. The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Research*, 38(suppl 2):W214–W220, July 2010.

[305] A. V. Werhli and D. Husmeier. Reconstructing gene regulatory networks with bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, 6(Issue 1):Article 15, 2007.

[306] P. Westfall and S. Young. p Value Adjustments for Multiple Tests in Multivariate Binomial Models. *Journal of the American Statistical Association*, 84(407):780–786, 1989.

[307] D. L. Wilson, M. J. Buckley, C. A. Helliwell, and I. W. Wilson. New normalization methods for cDNA microarray data. *Bioinformatics*, 19(11):1325–1332, July 2003.

[308] C. Wolting, C. McGlade, and D. Trithchler. Cluster analysis of protein array results via similarity of gene ontology annotation. *BMC Bioinformatics*, 7(1):338, 2006.

[309] K.-J. Won, A. Prügel-Bennett, and A. Krogh. Training hmm structure with genetic algorithm for biological sequence analysis. *Bioinformatics*, 20(18):3613–3619, 2004.

[310] C. H. Wu, A. Nikolskaya, H. Huang, L. S. Yeh, D. A. Natale, C. R. Vinayaka, Z. Z. Hu, R. Mazumder, S. Kumar, P. Kourtesis, R. S. Ledley, B. E. Suzek, L. Arminski, Y. Chen, J. Zhang, J. L. Cardenas, S. Chung, J. Castro-Alvear, G. Dinkov, and W. C. Barker. PIRSF: family classification system at the Protein Information Resource. *Nucleic Acids Res*, 32(Database issue):D112–D114, January 2004.

[311] C. H. Wu, L.-S. L. Yeh, H. Huang, L. Arminski, J. Castro-Alvear, Y. Chen, Z. Hu, P. Kourtesis, R. S. Ledley, B. E. Suzek, C. R. Vinayaka, J. Zhang, and W. C. Barker. The Protein Information Resource. *Nucleic Acids Research*, 31(1):345–347, Jan. 2003.

[312] H. Wu, Z. Su, F. Mao, V. Olman, and Y. Xu. Prediction of functional modules based on comparative genome analysis and gene ontology application. *Nucleic Acids Research*, 33(9):2822–2837, 2005.

[313] E. P. Xing and R. M. Karp. Cliff: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 17(supplement 1):S306–315, 2001.

[314] J. Xiong. *Essential Bioinformatics*. Cambridge University Press, New York, NY 10011-4211, USA, 2006.

[315] D. Xu, V. Olman, L. Wang1, and Y. Xu. EXCAVATOR: a computer program for efficiently mining gene expression data. *Nucleic Acids Research*, 31(19):5582–5589, 2003.

[316] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[317] T. Xu, L. Du, and Y. Zhou. Evaluation of GO-based functional similarity measures using S. cerevisiae protein interaction and expression profile data. *BMC Bioinformatics*, 9(1):472+, 2008.

[318] J. Yang, W. Wang, H. Wang, and P. Yu. δ–clusters: Capturing subspace correlation in a large data set. In *Proceedings of the 18th IEEE International Conference on Data Engineering*, pages 517–528, 2002.

[319] J. Yang, W. Wang, H. Wang, and P. Yu. Enhanced biclustering on expression data. In *Proceedings of the 3rd IEEE Conference on Bioinformatics and Bioengineering*, pages 321–327, 2002.

[320] K. e. a. Yeung. Model-based clustering and data transformation for gene expression data. *Bioinformatics*, 17:977–987, 2001.

[321] K. e. a. Yeung. Validating clustering for gene expression data. *Bioinformatics*, 17:309–318, 2001.

[322] A. Young, N. Whitehouse, J. Cho, and C. Shaw. Ontologytraverser: an r package for go analysis. *Bioinformatics*, 21(2):275–276, 2005.

[323] G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu, and S. Wang. GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics*, 26(7):976–978, Apr. 2010.

[324] U. Yu, Y. Choi, J. Choi, and S. Kim. To-go: a java-based gene ontology navigation environment. *Bioinformatics*, 21:3580–3581, 2005.

[325] B. R. Zeeberg, W. Feng, G. Wang, M. D. Wang, A. T. Fojo, M. S. andSudarshan Narasimhan, D. W. Kane, W. C. Reinhold, S. Lababidi, K. J. Bussey, J. Riss, J. C. Barrett, and J. N. Weinstein. GoMiner: a resource for biological interpretation of genomic and proteomic data. *Genome Biology*, 4:R28, 2003.

[326] B. Zhang, S. Kirov, and J. Snoddy. WebGestalt: an integrated system for exploring gene sets in various biological contexts. *Nucleic Acids Research*, 33(suppl 2):W741–W748, July 2005.

[327] B. Zhang, D. Schmoyer, S. Kirov, and J. Snoddy. GOTree Machine (GOTM): a web-based platform for interpreting sets of interesting genes using Gene Ontology hierarchies. *BMC Bioinformatics*, 5:16, 2004.

[328] G. P. Zhang. Neural networks for classification: a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30(4):451–462, 2000.

[329] B. Zheng and X. Lu. Novel metrics for evaluating the functional coherence of protein groups via protein semantic network. *Genome Biology*, 8(7):R153+, July 2007.

[330] Q. Zheng and X.-J. J. Wang. Goeast: a web-based software toolkit for gene ontology enrichment analysis. *Nucleic acids research*, 36(Web Server issue):W358–W363, May 2008.

[331] S. Zhong, L. Tian, C. Li, K.-F. Storch, and W. H. Wong. Comparative Analysis of Gene Sets in the Gene Ontology Space under the Multiple Hypothesis Testing Framework. In *IEEE Computational Systems Bioinformatics Conference.*, pages 425–435, 2004.

[332] P. Zoppoli, S. Morganella, and M. Ceccarelli. Timedelay-aracne: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC Bioinformatics*, 11(1):154+, March 2010.