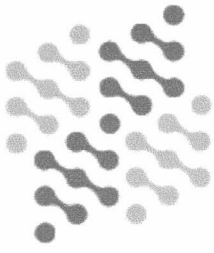
Tesis de Licenciatura:

GO-GPS: un método computacional basado en clustering conceptual para la identificación, explicación y predicción de perfiles de expresión genética

Diciembre de 2005



DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Alumno: Juan Pablo Tomás Grassi
jgrassi@dc.uba.ar

Director: Dr. Igor Zwir
zwir@borcim.wustl.edu

Co-directora: Dra. Rocío Celeste Romero Zaliz
rromero@dc.uba.ar

Agradecimientos

A lo largo del tiempo que llevó la realización de este trabajo y de mi carrera, hubo mucha gente que se preocupó por mí y me ayudó. Seguramente olvidándome de alguien, quisiera agradecer profundamente a las siguientes personas:

A mis directores Igor y Rocío, sin cuya predisposición, arduo trabajo y desinteresada ayuda, esta tesis nunca hubiera podido concretarse.

A Irene Loiseau, por ser la persona que siempre me ha escuchado y ayudado cada vez que tuve alguna duda o necesité ayuda.

A mis compañeros de facultad Emanuel, Gaby Barbuto, Patricio, Chirlo, Gaby Gasser, Ale, Claudia, Spike, Pablo y Marto por afrontar conmigo este camino.

A Natalia, por estar siempre a mi lado apoyándome y alentándome, por compartir conmigo todos estos años y por enseñarme tantas cosas importantes en la vida.

A mis padres, por apoyarme en este largo camino y preocuparse en todo momento por mí.

A Eugenio, Andrea, Gisela, Jhonatan, Joel, Nadia, José, Tania, Juan José, Panchi, Alejandra, Nashka, María y Jorge, por comprenderme.

A mis amigos Christian, Fari, Raulo, Facha, Piedi y Simo por estar siempre desde hace tanto tiempo.

Y a todas aquellas personas que he olvidado de nombrar, pero que se han preocupado por la evolución de este trabajo en algún momento.

Índice general

Índice de cuadros

Índice de figuras

Resúmen

Los avances en biología molecular y en técnicas computacionales permiten el estudio sistemático de los procesos moleculares complejos que subyacen en los sistemas biológicos. Particularmente, la tecnología de Microarray ha revolucionado la investigación biomédica moderna por su capacidad para monitorear cambios en la abundancia de ARN relativa para miles de genes simultáneamente.

La creciente disponibilidad de estos conjuntos de datos que contienen representaciones de productos de genes como series de tiempo de Microarray, redes regulatorias o caminos metabólicos ha permitido el acceso a una gran cantidad de datos. Sin embargo, la subyacente caracterización de estos conjuntos de datos aún constituye un desafío. El proyecto *Gene Ontology* (GO) ha sido uno de los más interesantes enfoques dedicado a proveer una descripción uniforme de productos de genes, organizándolos por sus procesos biológicos, funciones moleculares o componentes celulares en una base de datos estructurada. Sin embargo, las herramientas y técnicas actuales para examinar el contenido de estas extensas bases de datos estructuradas aún están obstaculizadas por su inhabilidad de realizar búsquedas basadas en criterios que sean significativos para los usuarios de estos repositorios, quienes usualmente no logran obtener los resultados esperados al analizar estos extensos conjuntos de datos. En particular - y a pesar del reciente renovado interés en técnicas de descubrimiento del conocimiento (o data mining) - hay una carencia de métodos de análisis de datos diseñados para facilitar el entendimiento de los objetos representados y sus sistemas relacionados por sus características más representativas y aquellas relaciones derivadas de estas características.

En esta tesis se propone un método de clustering conceptual llamado GO-GPS por *Gene Ontology Grouping, Prototyping and Searching*. Este método trata el problema de descubrir conjuntos de características de genes basadas en la base de datos de GO que pueden describir y predecir perfiles de expresión genética. GO-GPS utiliza técnicas de optimización multiobjetivo que permiten describir perfiles de genes de diferentes ángulos y de esta manera codificar la incertidumbre que caracteriza generalmente a los experimentos de Microarray. El método propuesto es aplicado a un problema derivado de perfiles longitudinales de expresión en sangre de humanos voluntarios tratados con una endotoxina intravenosa comparados contra un placebo. Este problema es parte de un proyecto de investigación colaborativo de gran escala respaldado por el *National Institute of General Medical Sciences* de Estados Unidos (www.gluegrant.org). El análisis del conjunto de perfiles de expresión genética obtenidos de este experimento es complejo, dado el número de muestras tomadas y la variación debido al tratamiento, el tiempo y el fenotipo del paciente. Por lo tanto, creemos que este problema es un punto de referencia para el análisis de datos de Microarray.

Abstract

Advances in molecular biology and computational techniques permit the systematical study of complex molecular processes that underlie biological systems. Particularly, microarray technology has revolutionized modern biomedical research by its capacity to monitor changes in relative RNA abundance for thousands of genes simultaneously.

The increased availability of these datasets containing gene product representations as microarray time series, regulatory networks or metabolic pathways has permitted access to vast amounts of data. However, the underlying characterization of these datasets still constitutes a challenge. The *Gene Ontology* Project (GO) has been one of the most interesting approaches devoted to provide uniform descriptions of gene products, organizing them by their biological process, molecular function or cellular components into structured databases. However, current tools and techniques to examine the content of these large structural databases are still hampered by their inability to support searches based on criteria that are meaningful to users of those repositories, who usually get trapped when attempt to mine into vast datasets. In particular -and in spite of the recent renewed interest in knowledge-discovery techniques (or data mining)- there is a dearth of data analysis methods intended to facilitate understanding of the represented objects and related systems by their most representative features and those relationship derived from these features.

In this thesis we propose a conceptual clustering method termed GO-GPS for *Gene Ontology Grouping, Prototyping and Searching*. This method addresses the problem of uncovering sets of gene features based on GO database that can describe an predict gene expression profiles. GO-GPS uses multiobjective optimization techniques that allow to describe gene profiles from different angles and thus encoding the uncertainty that usually characterize the microarray experiments. We applied our proposed method to a problem derived from longitudinal blood expression profiles of human volunteers treated with intravenous endotoxin compared to placebo, as part of a Large-scale Collaborative Research Project sponsored by the *National Institute of General Medical Sciences* (www.gluegrant.org). Analysis of the set of gene expression profiles obtained from this experiment is complex, given the number of samples taken and variance due to treatment, time, and patient phenotype. Therefore, we believe this problem is a benchmark for the analysis of microarray data.

Capítulo 1

Introducción

Durante los últimos años la biología molecular ha experimentado importantes avances, especialmente con proyectos como el del Genoma Humano, en los cuales se ha logrado determinar experimentalmente las secuencias de ADN de muchos organismos. La gran cantidad de datos generados ha dado origen a un nuevo campo de estudio, la *biología computacional* ó *bioinformática*, que consiste en la utilización de técnicas computacionales y matemáticas para modelar y resolver problemas de biología molecular.

Uno de los problemas que se estudia en bioinformática es el de *expresión genética*. Las células de un organismo utilizan subconjuntos de sus genes, denominados activos o expresados, para dirigir la producción de las moléculas que aseguran la supervivencia del organismo. Toda célula de un organismo multicelular contiene un conjunto idéntico de genes. Sin embargo, células especializadas en diferentes funciones, por ejemplo cardíacas o de la piel, se comportan de manera distinta debido al subconjunto de genes activos en cada una de ellas a lo largo del tiempo, es decir, la expresión de los genes está regulada. Es importante estudiar la expresión y la regulación genética para lograr un mejor entendimiento del comportamiento celular. Un ejemplo de cómo afecta los mecanismos de regulación genética se puede observar en enfermedades tales como el cáncer. La células cancerígenas se multiplican en condiciones donde sus contrapartes no lo hacen, dado que en parte ciertos genes están activados, o no, cuando en células sanas esto no ocurre.

Para estudiar la expresión genética se pueden aplicar nuevas tecnologías como los Microarrays de ADN, los cuales permiten medir la expresión de miles de genes simultáneamente. Convertir este gran catalogo de información en conocimiento es un gran desafío y recién se están dando los primeros pasos.

Cuando se analizan datos de expresión genética existen situaciones que son muy difíciles de distinguir, puesto que las expresiones pueden parecerse mucho entre sí. Para citar un ejemplo, las enfermedades de Parkinson y Esquizofrenia se parecen mucho en sus síntomas ya que al principio las expresiones de los genes son muy parecidas. De esta manera surge la necesidad de proveer información externa a la expresión con el fin de encontrar conceptos que se puedan utilizar para clasificar, explicar y distinguir situaciones que a priori parecen indistinguibles.

La presente tesis tiene como objetivo plantear un método que se pueda utilizar para obtener conocimiento a partir de datos de expresión genética obtenidos de experimentos de Microarrays. El método propuesto, llamado GO-GPS por *Gene Ontology Grouping, Prototyping and Searching*, encuentra grupos cohesivos de genes que comparten procesos biológicos, funciones moleculares y componentes celulares (*Grouping*), formando así modelos (*Prototyping*) para poder utilizarlos como patrones de búsqueda

en la clasificación de nueva información (*Searching*).

Este método fue desarrollado con la finalidad de proveer información para un nuevo problema, derivado del programa de investigación *Inflamación y respuesta del huésped a estímulos externos (Inflammation and the Host Response to Injury)*, cuyo objetivo es investigar los pasos que ocurren en el sistema inmunológico como respuesta a lesiones traumáticas. La identificación de estos eventos moleculares y los factores genéticos que los originan ayudarán a los médicos a predecir los resultados de la recuperación de pacientes con lesiones graves.

La presente tesis está estructurada en cinco capítulos. El contenido de cada uno de los capítulos se comentará brevemente a continuación.

El capítulo 2 comienza explicando los conceptos básicos de biología molecular necesarios para comprender qué son los Microarrays de ADN y como pueden utilizarse para medir la expresión de miles de genes simultáneamente. Luego se presenta en detalle el experimento biológico objeto de nuestro estudio y la base de datos *Gene Ontology* sobre la cual está basado el método propuesto en la presente tesis.

En El capítulo 3 se presentan los diferentes métodos utilizados para extraer conocimiento de base de datos. Se estudia por un lado la herramienta FatiGO, la cual utiliza específicamente *Gene Ontology* para el análisis de datos de expresión. Por otro lado, se presentan diversos métodos automáticos mencionando los problemas que surgen al aplicarlos a nuestro domino. Se hace hincapié en clustering conceptual y optimización multiobjetivo, técnicas utilizadas en el método que se presenta en esta tesis

En el capítulo 4 se presenta el método GO-GPS, el cual es un método exhaustivo que se utiliza para el análisis de datos provenientes de estudios de expresión genética. Adicionalmente, se realiza un análisis de distintas funciones objetivo para este dominio en particular.

En el capítulo 5 se presenta el método GO-GPS-GA, que es una adaptación de un algoritmo genético multiobjetivo para su utilización en análisis de datos de expresión genética, utilizanso los objetivos estudiados en el capítulo anterior. Se presentan los resultados obtenidos, mostrando como pueden utilizarse para explicar perfiles de expresión genética utilizando información de *Gene Ontology*. Estos resultados son comparados con los obtenidos por otros dos métodos, uno de ellos un algoritmo genérico de data mining APRIORI, y el otro la herramienta FatiGO, presentada en el capítulo 3.

En el capítulo final se presentan las conclusiones de la presente tesis y algunas líneas de investigación posibles para trabajos futuros.

Capítulo 2

El problema biológico: identificación de perfiles de expresión genética

Los seres vivos están formados por células que comparten una maquinaria común para sus funciones más básicas. Si bien por fuera los seres vivos son infinitamente diversos, su estructura celular es muy similar. En este capítulo se introducen diversos conceptos biológicos y se presenta detalladamente el experimento biológico que se estudia en la presente tesis.

En la sección 2.1 se presentan los conceptos biológicos básicos involucrados en el proceso de expresión genética. En la sección 2.2 se explica brevemente en qué consiste un experimento de Microarray y cómo puede ser utilizado para estudiar la expresión genética. En la sección 2.3 se presenta el experimento biológico objeto de nuestro estudio, el cual constituye un esfuerzo conjunto por explicar el sistema inflamatorio en seres humanos. Finalmente, la sección 2.4 contiene una descripción detallada de la base de datos del proyecto *Gene Ontology*, que contiene información sobre varias características de genes y producto de genes. Esta información será utilizada para explicar los perfiles de expresión genética derivados del experimento biológico.

2.1. Conceptos biológicos básicos

En todos los organismos vivientes existen unas moléculas esenciales que realizan prácticamente todas las funciones celulares, las *proteínas*. Estas macromoléculas están formadas por moléculas más pequeñas denominadas *aminoácidos* y son producidas en la célula gracias a un complejo sistema que comienza con la información contenida en los *ácidos nucleicos*. Los ácidos nucleicos son las macromoléculas en las cuales se almacena y procesa toda la información genética de los sistemas biológicos. En las células de los organismos existen dos tipos de ácidos nucleicos: el *ácido desoxirribonucleico (ADN)* y el *ácido ribonucleico (ARN)*. El ADN y el ARN están formados por moléculas más simples denominadas *nucleótidos* [AJL+03].

Los nucleótidos son moléculas formadas por *bases nitrogenadas*, *ácido fosfórico* y un azúcar como la *ribosa* o la *desoxirribosa*. Las bases nitrogenadas que forman el ADN son *Adenina*, *Timina*, *Guanina* y *Citosina*, y la secuencia de estas bases en los *cromosomas* constituye el código genético (ver Figura 2.1). El *Uracilo* es otro tipo de base nitrogenada que se caracteriza por formar parte del ARN sustituyendo a la Timina. Además de esta diferencia, también se caracteriza porque su esqueleto, en

lugar de estar formado por fosfato y desoxirribosa, se compone de fosfato y ribosa. El ADN presenta una estructura de doble hélice, donde una larga hebra de ácido nucleico está enrollada alrededor de otra hebra formando un par entrelazado. Las bases de nucleótidos de una hebra se juntan con las bases de nucleótidos de la otra, en el sentido de que la Adenina siempre se junta con la Timina (A$\longleftrightarrow$T) y la Guanina siempre se junta con la Citosina (G$\longleftrightarrow$C) (Figura 2.2).

Un *gen* es una secuencia específica de bases de nucleótidos que lleva la información requerida para la construcción de una proteína. La secuencia de bases presente en el gen determina la secuencia de aminoácidos de la proteína por medio del código genético. A, T, G, y C son las "letras" del código genético y representan las bases nitrogenadas Adenina, Timina, Guanina y Citosina, respectivamente. En cada gen se combinan las cuatro bases en diversas formas, para crear palabras de 3 letras *(codón)* que especifican qué aminoácido es necesario en cada paso de la elaboración de la proteína. Por ejemplo, *Alanina* es uno de los aminoácidos que forman las proteínas de los seres vivos y, en ARN, codifica como GCU, GCC, GCA o GCG.

Existen aproximadamente veinte aminoácidos distintos que se denominan aminoácidos esenciales, los cuales poseen especial importancia porque son los requeridos por los organismos para construir proteínas. En muchas especies de organismos sólo una pequeña fracción del total de la secuencia del genoma codifica proteínas. La función del resto es desconocida.

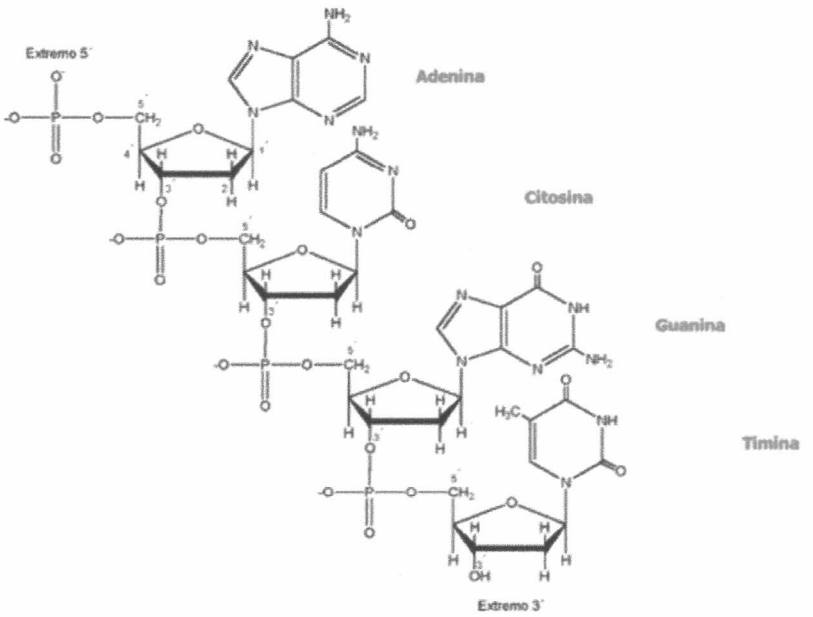


Figura 2.1: Estructura química de las bases nitrogenadas que forman el ADN: Adenina, Citosina, Guanina y Timina.

Existen dos tipo de células, *procariotas* y *eucariotas*. La célula procariota es un organismo vivo cuyo núcleo celular no está envuelto por una membrana, en contraposición con los organismos eucariotas, que presentan un núcleo verdadero o rodeado de membrana nuclear. El proceso de expresión genética que se describe a continuación es el que se produce en los organismos eucariotas.

Se dice que un gen está *activo* o *expresado* si este produce la proteína que codifica. Si la cantidad de proteínas producidas es alta, el gen está expresado; en caso de no

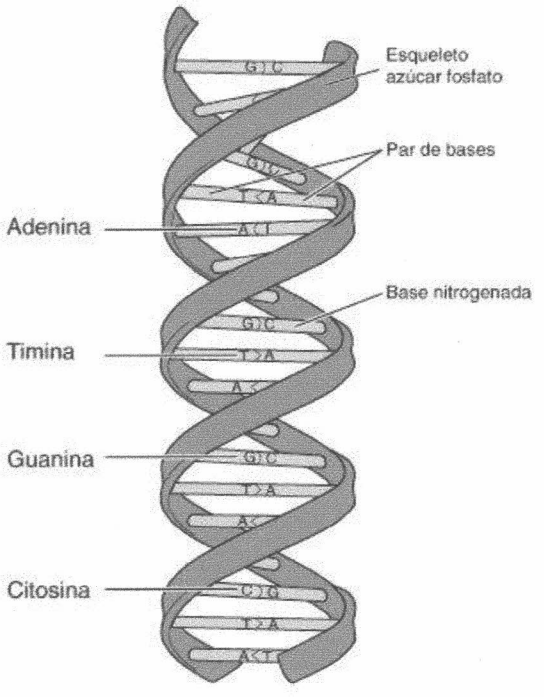


Figura 2.2: Estructura de doble hélice del ADN.

producir proteínas, el gen no está expresado [Dra03, Lew01].

El primer paso del proceso de expresión genética es leer la información contenida en el ADN y convertirla en una secuencia de ARN. Este proceso se llama *transcripción* y es realizado por una enzima llamada *ARN polimerasa*. Esta enzima se *hibrida* a una secuencia especial de nucleótidos llamada *promotor*, luego comienza a moverse a través del gen y va construyendo una secuencia de ARN utilizando ribonucleótidos libres. La secuencia de ARN construída es complementaria a la secuencia de ADN leída por la ARN polimerasa. Este proceso continúa hasta que la enzima encuentra una secuencia especial que le indica el final del gen. La secuencia de ARN construída contiene la misma información que el gen y se denomina *ARNm (ARN mensajero)*.

Una vez formado, el ARNm deja el núcleo de la célula hacia el *citoplasma* para unirse a una estructura celular llamada *ribosoma*. Aquí la información contenida en el ARNm se agrupa en tripletas de nucleótidos (codones) que codifican un determinado aminoácido. Los aminoácidos son llevados al ribosoma por moléculas especiales de ARN llamadas *ARNt (ARN de transferencia)*. Hay moléculas de este tipo específicas para cada tipo de aminoácido. Con estos aminoácidos se forma la proteína codificada por el gen.

Generalmente existe una correspondencia entre la cantidad de ARNm producido por el gen y la cantidad de proteína producida. Entonces la cantidad de ARNm puede ser utilizada para medir el nivel de expresión de un determinado gen [Dra03].

Es conocido que anomalías en la expresión de los genes pueden llevar a disfunciones celulares, provocando graves enfermedades como el cáncer, entre muchas otras. Por esto es sumamente importante estudiar la expresión de genes. En este sentido, desde mediados de los años noventa existe la técnica de los Microarrays de ADN, que permite

monitorizar simultáneamente el nivel de expresión de miles de genes.

2.2. Experimentos de Microarrays de ADN

Los Microarrays de ADN son una herramienta que permite realizar análisis genéticos diversos, por ejemplo, medir simultáneamente el nivel de expresión de miles de genes. Las mejoras tecnológicas han perfeccionado la calidad y han ampliado el espectro de aplicaciones, de manera que los Microarrays se han consolidado como herramientas útiles en investigación genética con aplicaciones en medicina.

El funcionamiento de los Microarrays de expresión se basa en la capacidad de las moléculas complementarias de ADN de hibridar entre sí. Pequeñas cantidades de ADN, correspondientes a diversos genes cuya expresión se desea medir, son depositadas en una base de cristal. Para ello se utilizan robots de precisión que usan agujas especiales para obtener las moléculas de sus recipientes y depositarlas en las coordenadas adecuadas. Estas muestras de ADN depositadas en el Microarray se denominan *dianas*. En un Microarray típico, una superficie de 2 x 2 cm puede contener más de 10.000 dianas en forma de pequeños puntos separados adecuadamente. De las células que se quiera medir su expresión se obtiene una muestra de ARN que se traducirá en ADN complementario (ADNc) y se marcará con una molécula fluorescente. A esta muestra marcada se la denomina *sonda* y será utilizada para buscar posibles correspondencias con las dianas del Microarray. Cada molécula de ADNc marcada de la sonda se moverá por difusión hacia la diana que contenga su molécula complementaria para hibridarse con ella y quedar fijada allí. Después de un tiempo para que la mayoría de las cadenas complementarias hibriden, el Microarray se lava y se procede a hacer una medición relativa de la cantidad de ADN de la sonda que ha quedado fijada en cada diana [Dra03, AJL+03].

Existe otra tecnología que emplea oligonucleótidos (secuencias cortas de ADN, de unas 15-30 bases). Estos oligonucleótidos, en lugar de ser depositados en el soporte mediante un robot, son sintetizados directamente sobre el soporte mediante una técnica denominada fotolitografía que es similar a la empleada para confeccionar circuitos microelectrónicos sobre silicio. Esta tecnología requiere una infraestructura muy sofisticada y su empleo por el momento está limitado a unas pocas empresas especializadas entre las que destaca Affymetrix Inc®. Para detectar la expresión de un gen se emplea una serie amplia de oligonucleótidos, alrededor de 30, por lo que estos Microarrays contienen muchas más dianas, lo que es factible porque la fotolitografía permite obtener mayores densidades.

El proceso completo de un experimento de Microarray es complicado y además interviene software específico en diferentes etapas. El objetivo de un experimento de Microarray es medir y comparar los niveles relativos de expresión de miles de genes en una sola muestra simultáneamente. Típicamente se comparan diferentes estados del ciclo celular, tipos de células, células sanas y enfermas y diferentes tratamientos.

El resultado de un experimento de Microarray es un listado de genes con su nivel de expresión. Si se realizan varios experimentos con distintas condiciones de estudio se obtiene una matriz de expresión donde las filas corresponden a los genes y las columnas a los distintos experimentos. A modo de ejemplo, consideremos un experimento en el cual se suministra cierta sustancia a un paciente y se mide la expresión de sus genes en 6 instantes de tiempo diferentes. Cada uno de estos 6 instantes de tiempo es un experimento de Microarray. Los resultados podrían ser los presentados en la Tabla 2.1 (solo se muestran 5 genes por simplicidad, pero en un experimento real se cuenta con varios miles de genes). En la Figura 2.3 se muestra el gráfico correspondiente a estos datos de expresión. Este gráfico permite observar la evolución de los niveles de

expresión para los 5 genes del experimento.

gen	*instante 1*	*instante 2*	*instante 3*	*instante 4*	*instante 5*	*instante 6*
gen 1	1434.79	412.453	600.224	592.453	702.107	710.675
gen 2	3857.15	1863.75	2223.38	2050.35	2166.63	3432.44
gen 3	6686.68	1829.78	2178.51	1681.9	1871.06	7489.39
gen 4	12932.5	8255	6202.36	6158.49	6501.12	15140.4
gen 5	5806.01	5875.25	5461.45	4781.4	5113.52	5439.97

Tabla 2.1: Resultado de varios experimentos de Microarray. Se mide el nivel de expresión de 5 genes en 6 instantes de tiempo. Cada una de las columnas corresponde a un experimento de Microarray. Por implicidad solo se muestran 5 genes, en un experimento real se mide el nivel de expresión de miles de genes simultáneamente.

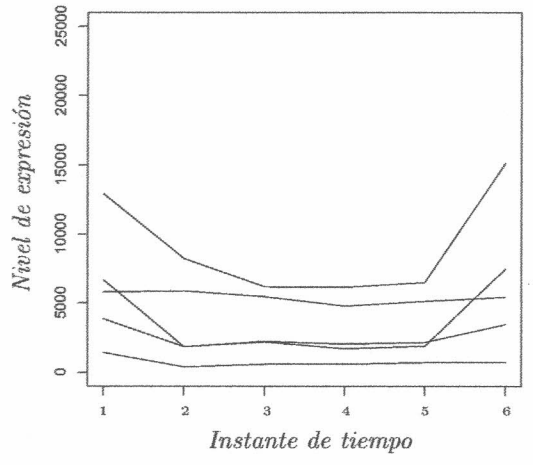


Figura 2.3: Gráfica de la serie temporal correspondiente a la expresión de un conjunto de genes en el tiempo. El eje X corresponde a los 6 instantes de tiempo en que se toma la medición, mientras que el eje Y corresponde al nivel de expresión detectado.

2.3. Presentación del experimento biológico

El problema biológico que se estudiará constituye un esfuerzo conjunto dentro del programa de investigación *Inflamación y respuesta del huésped a estímulos externos (Inflammation and the Host Response to Injury)*. Este programa está respaldado por el *National Institute of General Medical Sciences (NIGMS)*, una división del *National Institutes of Health* de Estados Unidos y su objetivo es descubrir las razones biológicas por las cuales diferentes pacientes pueden llegar a tener respuestas muy distintas tras sufrir una herida traumática. Este programa interdisciplinario a gran escala constituye el primer intento por explicar las reacciones de los seres humanos ante quemaduras o inflamaciones seguidas a un trauma importante. Este proyecto reúne a instituciones médicas e investigadores de las áreas de cirugía, genómica, proteómica, bioestadística, bioinformática, biología computacional y genética, para estudiar la biología molecular de las inflamaciones.

El cuerpo humano utiliza la inflamación para protegerse de una infección o una lesión y para indicar a los tejidos que comiencen a curarse. Sin embargo la inflamación excesiva puede conducir a una enfermedad llamada *sepsis*, en la cual se presenta una caída de la presión sanguínea que produce shock, conduciendo a que los sistemas orgánicos principales, incluyendo los riñones, hígado, pulmones y sistema nervioso central, dejen de funcionar normalmente. La sepsis con frecuencia es potencialmente mortal.

El objetivo del programa de investigación *Inflamación y respuesta del huésped a estímulos externos* es investigar los pasos que ocurren en el sistema inmunológico como respuesta a lesiones traumáticas. La identificación de estos eventos moleculares y los factores genéticos que los originan ayudarán a los médicos a predecir los resultados de la recuperación de pacientes con lesiones graves.

En el contexto de este programa de investigación se han realizado varios experimentos de Microarrays de ADN con humanos voluntarios. Concretamente se han analizado ocho pacientes, cuatro tratados con una endotoxina intravenosa (pacientes 1 a 4) y cuatro con placebo (pacientes 5 a 8). La endotoxina intravenosa suministrada a voluntarios normales es un estímulo bien definido que, de forma reproducible, induce síntomas similares a la gripe, cuya resolución es en 24 horas [RRF+89]. Tales síntomas están asociados, en el modelo, con cambios importantes en los perfiles de expresión de genes de Leucocitos circulantes [CXR+05]. Los datos fueron obtenidos en diferentes instantes de tiempo a las 0, 2, 4, 6, 9 y 24 horas y procesados con GeneChips® HG-U133A v2.0 de Affymetrix Inc®.

Para extraer conocimiento de este experimento se puede estudiar la expresión de genes a través del tiempo. Se ha utilizado el algoritmo k-means [Mit97] para clasificar la expresión de los genes a través del tiempo en 24 clusters. El conjunto de perfiles de expresión genética encontrados en este experimento es uno de los más complejos que se pueden encontrar en estos días [EZ05].

En la Figura 2.4 se presentan los 24 clusters de expresión. En cada gráfico están representados los cuatro pacientes tratados con endotoxina intravenosa. A su vez, para cada paciente están los datos a las 0, 2, 4, 6, 9, y 24 horas, como se muestra en la Figura 2.5. Notemos que la expresión de cada gen ha sido representada con 24 puntos: paciente 1 hora 0, paciente 1 hora 2, ... , paciente 1 hora 24, paciente 2 hora 0, ..., paciente 2 hora 24, paciente 3 hora 0, ..., paciente 3 hora 24, paciente 4 ahora 0, ..., paciente 4 hora 24. La principal ventaja de esta representación es que se puede examinar el comportamiento de los genes para cada paciente, haciendo posible reconocer diferencias en el comportamiento de un determinado gen en distintos pacientes. Estas diferencias pueden ayudar a reconocer condiciones que de otra manera estarían ocultas, como por ejemplo sexo o edad. Un ejemplo de comportamiento diferente de un mismo gen en distintos pacientes se da en los cluster 11 y 18 (ver Figura 2.4). Aquí se ve claramente que el paciente 1 se comporta de manera diferente que el resto.

El objetivo principal de este trabajo es encontrar una explicación biológica para estos clusters de expresión, puesto que de esta manera se tendrían características comunes en genes que se han expresado de la misma manera a través del tiempo. La base de datos de *Gene Ontology* constituye una herramienta útil para este propósito, y será utilizada para poder explicar biológicamente estos clusters o perfiles de expresión.

2.4. Gene Ontology

Muchos investigadores sostienen que existen una gran cantidad de genes y proteínas compartidas por muchos organismos [Con00]. Es por ello que el conocimiento de la función del rol biológico de un determinado gen en un organismo puede ser utilizado para inferir su rol en otros organismos. El problema que se tiene actualmente es que

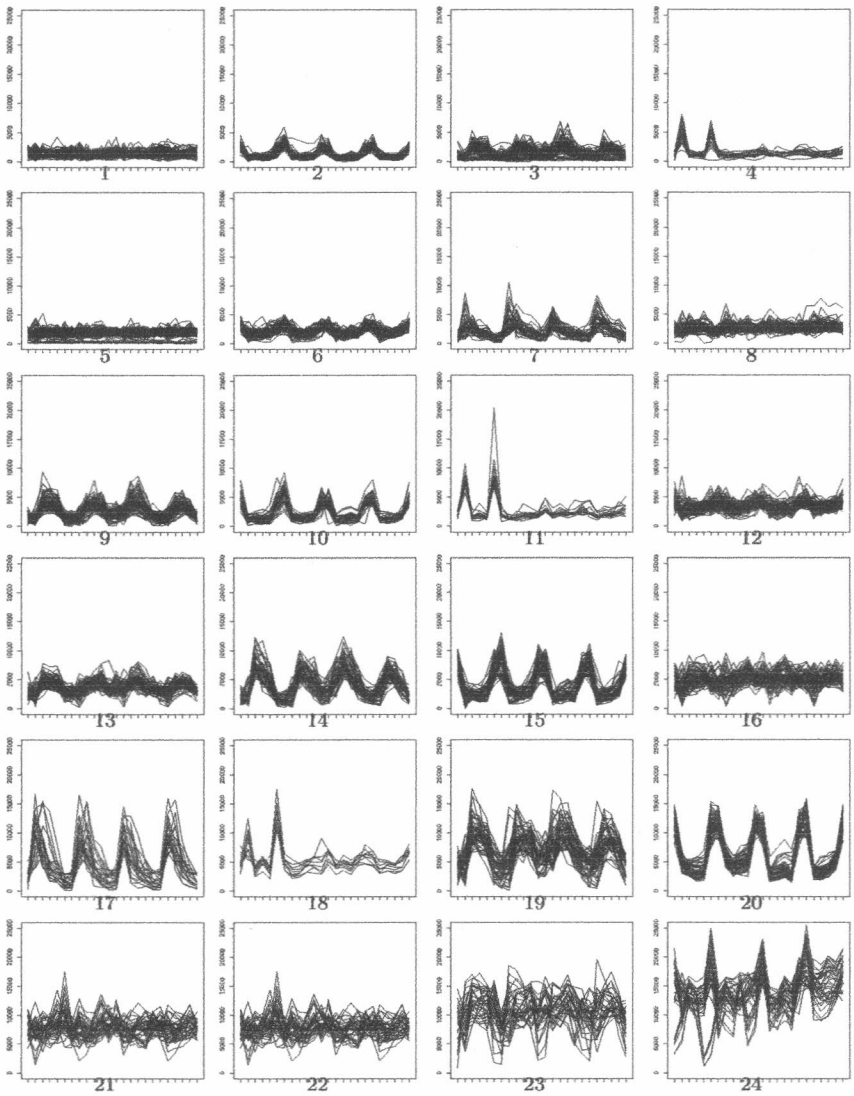


Figura 2.4: Clusters de los datos de expresión. Este agrupamiento ha sido realizado utilizando el algoritmo k-means con $k = 24$.

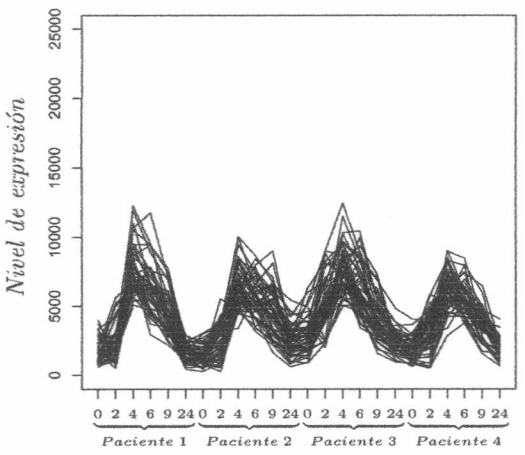


Figura 2.5: Gráfico de la serie temporal correspondiente a la expresión de un conjunto de genes en el tiempo. El eje X corresponde a la hora en que se toma la medición, mientras que el eje Y corresponde al nivel de expresión detectado. Notar que solamente se representa la información de los cuatro pacientes a los cuales se les ha inyectado la endotoxina en forma sucesiva, obteniendo una gráfica con una distribución regular.

las bases de datos existentes utilizan diferentes vocabularios y diferentes formatos para describir anotaciones funcionales.

Una manera de abordar la cuestión de la heterogeneidad en la descripción de genes es desarrollar una ontología para genes. De esta manera nace *Gene Ontology (GO)* [Con00], un vocabulario controlado que se utiliza para describir uniformemente productos de genes. El uso de los términos de GO por varias bases de datos facilita la uniformidad de las consultas. Los vocabularios están estructurados de tal manera que se puede consultar a diferentes niveles. Por ejemplo, se puede usar GO para buscar todos los genes en el genoma del ratón que estén involucrados en la *transducción de señales*[1], o se puede buscar, en forma más específica, todos los receptores *tirosina-kinasa*[2]. Esta estructura también permite realizar anotaciones sobre propiedades de genes en diferentes niveles, dependiendo del conocimiento que se tenga de los mismos.

Es importante notar que GO no es una base de datos de secuencias de genes, como puede ser GenBank [BKML+03] o EMBL [KAA+05], y tampoco un catálogo de genes. Por el contrario, GO describe cómo se comportan los genes en un contexto celular. Por otro lado, GO no intenta describir cada aspecto de la biología. Por ejemplo, la estructura de dominio, la estructura 3D, la evolución y la expresión genética de una proteína, no están contenidas en la base de datos de GO.

Esta base de datos está compuesta de tres vocabularios (ontologías) estructurados y controlados para describir los genes en términos de sus procesos biológicos asociados, componentes celulares y funciones moleculares, independientemente de las especies

[1] Conjunto de procesos o etapas que ocurren de forma concatenada por el que una célula convierte una determinada señal o estímulo exterior, en otra señal o respuesta específica.

[2] Una tirosina-kinasa es una enzima que puede transferir un grupo fosfato a una tirosina en una proteína. Estas enzimas son una subclase de una clase de proteínas kinasa más amplia. La fosforilación es una función importante en la transducción de señales para regular una actividad enzimática.

biológicas. Un gen puede tener una o más funciones moleculares, ser utilizado en uno o más procesos biológicos y estar asociado a uno o más componentes celulares. Por ejemplo, el gen *citocromo c* puede describirse por los términos "matriz mitocondrial" y "membrana mitocondrial interna".

2.4.1. Las diferentes ontologías

Como ya se ha mencionado, las tres organizaciones principales de GO son función molecular, proceso biológico y componente celular. A continuación se analizan en detalle cada una de ellas:

- *Función Molecular:* esta ontología abarca aquellas tareas desarrolladas por genes individuales.

  La función molecular describe actividades, tales como actividades catalíticas o de *binding*, a nivel molecular. Los términos de funciones moleculares de GO representan actividades y no entidades, como moléculas o complejos, que realizan acciones y no especifican dónde, cuándo, ni en qué contexto se lleva a cabo la acción. Las funciones moleculares generalmente corresponden a actividades que pueden ser realizadas por genes individuales, pero algunas actividades son realizadas por complejos ensamblados de genes. Ejemplos de términos de funcionalidad general son las actividades "catalíticas", actividades de "transporte", o "*binding*". Ejemplos de términos funcionales más específicos son la actividad de adenilato ciclasa o el *binding* del receptor *toll*.

- *Proceso biológico:* esta ontología comprende objetivos biológicos, tales como la mitosis o el metabolismo de purinas, que son realizados por funciones moleculares.

  Un proceso biológico es llevado a cabo por uno o más ensamblajes ordenados de funciones moleculares. Ejemplos de términos de procesos biológicos de amplio espectro son "crecimiento y mantenimiento" o "transducción de señales". Ejemplos de términos más específicos son "metabolismo de pirimidinas" o "transporte de alfa-glucosidasa". Puede ser difícil distinguir entre proceso biológico y función molecular, pero la regla general es que los procesos deben tener más de un paso distintivo. Esto no debe confundirse con un camino o *pathway*.

- *Componente celular:* esta ontología cubre estructuras subcelulares, localizaciones y complejos macromoleculares.

  Un componente celular es simplemente lo que su nombre indica, un componente de la célula, a condición de que sea parte de algún objeto mayor, el cual puede ser una estructura anatómica (retículo endoplasmático rugoso o núcleo) o un grupo de genes (ribosoma, proteasoma o dímero proteico). Ejemplos de términos de esta ontología son "núcleo" y "telómero".

2.4.2. Estructura de las ontologías

Los términos GO están organizados en estructuras llamadas *grafos dirigidos acíclicos (GDAs)* [AHU82], que son diferentes a una jerarquía en que un "hijo", es decir, un término más especializado, puede tener varios "padres", es decir, términos menos especializados[3]. Por ejemplo, el proceso biológico "biosíntesis de hexosa" tiene dos padres, "metabolismo de hexosa" y "biosíntesis de monosacáridos". Esto es debido a que es

[3]Si bien un GDA no es una jerarquía en sentido estricto, en esta tesis se utilizan estos términos como sinónimos.

un subtipo de metabolismo y a que una hexosa es un tipo de monosacárido. Cualquier gen involucrado en la biosíntesis de hexosa anotado con este término, también es automáticamente anotado tanto a "metabolismo de hexosa" como con "biosíntesis de monosacáridos". Esto es debido a que cada término de GO obedece la *regla del camino verdadero*[4]: si el término hijo describe un producto de un gen, también todos sus términos padre deben aplicar a ese producto. Se dice entonces que un nodo de GO, es decir un nodo del GDA, se refiere al término indicado en particular y a todos sus padres.

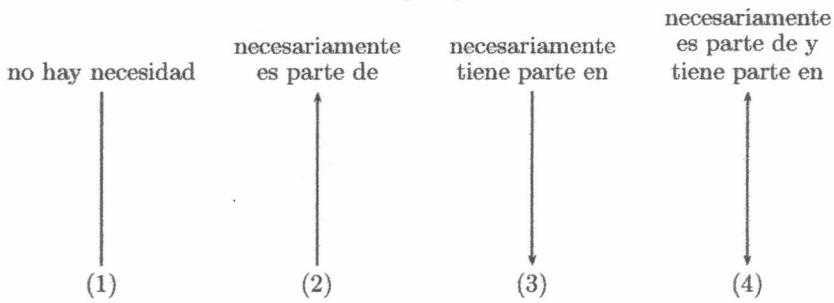
Las tres ontologías están unidas entre sí por el término de GO 0003673, llamado "Gene_Ontology". Todo término de GO tiene como ancestro a éste término, sea cual sea su tipo (componente celular, función molecular o proceso biológico). Este término no representa ningún concepto biológico real y ha sido declarado como obsoleto y reemplazado por un nodo artificial, que no pertenece a ninguna de las ontologías, llamado "all" (todo), el cual constituye el término más general posible.

Un término hijo puede tener una de dos posibles relaciones con su/s padre/s: "es_un" o "es_parte_de". Un mismo término puede tener diferentes relaciones con diferentes padres.

La relación "es_un" significa que un término es una subclase de sus padres. Por ejemplo, "ciclo de una célula mitótica" es_un "ciclo de una célula". No debe confundirse con una *instancia*, la cual es un ejemplo específico. La relación "es_un" es transitiva, lo cual significa que si un término GO A es una subclase del término GO B, y el término GO B es una subclase del término GO C, entonces el término GO A es también una subclase del término GO C.

La relación "es_parte_de" es más compleja. Existen cuatro niveles básicos de restricciones para una relación "es_parte_de":

Concepto padre



Concepto hijo

- El primer tipo (1) no tiene restricciones. Esto es, no se puede realizar otra inferencia de la relación entre padre e hijo aparte de que el padre pueda o no tener al hijo como parte de él, y que el hijo pueda o no ser parte del padre.
- El segundo tipo (2) significa que, cuando el hijo existe, obligatoriamente es parte del padre. Por ejemplo, "bifurcación de una réplica" es parte de "cromosoma". Por lo tanto, cuando una "bifurcación de una réplica" ocurre, entonces es parte de "cromosoma"; pero "cromosoma" no necesariamente tiene una "bifurcación de una réplica".

[4]La regla del camino verdadero define que "el camino desde un término hijo hasta su/s padre/s en el nivel más alto debe ser verdadero".

- El tercer tipo (3) es exactamente la inversa del tipo (2): cuando un padre existe, tiene al hijo como parte, pero no necesariamente el hijo es parte del padre. Por ejemplo, "núcleo" siempre es parte de "cromosoma", pero "cromosoma" no es necesariamente parte de "núcleo".
- El cuarto tipo (4) es la combinación de los casos (2) y (3). Un ejemplo de éste caso es "membrana nuclear" que es parte de "núcleo". Por ello, "núcleo" siempre tiene parte en "membrana nuclear" y "membrana nuclear" siempre es parte de "núcleo".

La relación "es_parte_de" utilizada en GO es generalmente del segundo tipo (2), "necesariamente_es_parte_de". Notar que los tipos (1) y (3) no se utilizan en GO, ya que violarían la regla de camino verdadero. Al igual que "es_un", "es_parte_de" es transitiva, de tal manera que si un término GO A es parte del término GO B, y el término GO B es parte del término GO C, entonces el término GO A es también es parte del término GO C.

2.4.3. Formato de las anotaciones utilizando GO

Cada gen se anota con uno o varios códigos de GO de una o varias de sus ontologías. Adicionalmente, cada uno de estos términos tiene asociado un código de evidencia, que determina de qué manera se ha obtenido la relación entre el término y el gen. Ejemplos de evidencia pueden ser IC (inferido por un curador) y TAS (declaración detectable de un autor, lo cual puede ser a través de un experimento publicado en un artículo de revista o en un libro de texto o diccionario). En la Figura 2.6 se puede ver una porción de una entrada de la base de datos de GenBank de un producto de un gen donde se pueden apreciar las anotaciones de GO asociadas. Por ejemplo, esta proteína tiene la función molecular "actividad oxidoreductasa", que corresponde al código de GO 0016491, y la evidencia IEA (inferido de una anotación electrónica, lo que quiere decir que no ha sido verificada por un curador).

2.5. Observaciones finales

En este capítulo se han presentado los conceptos biológicos básicos involucrados en el proceso de expresión genética, como así también en qué consiste un experimento de Microarray y como puede utilizarse para realizar estudios de expresión genética.

El experimento biológico objeto de nuestro estudio constituye un esfuerzo conjunto por explicar el sistema inflamatorio en seres humanos, y consiste en varios estudios de Microarrays realizados en distintos instantes de tiempo a un grupo de humanos voluntarios. Los resultados del estudio han sido agrupados utilizando el algoritmo k-medias con el cual se obtienen 24 perfiles de expresión.

Finalmente, se ha presentado la base de datos del proyecto *Gene Ontology*, la cual brinda información sobre los procesos biológicos, funciones moleculares y componentes celulares de los genes y producto de genes. Esta información será utilizada para explicar los perfiles de expresión genética derivados del experimento biológico.

```
LOCUS       NP_001082               751 aa            linear   PRI 26-OCT-2004
DEFINITION  amiloride binding protein 1 precursor [Homo sapiens].
ACCESSION   NP_001082
VERSION     NP_001082.1  GI:4501851
DBSOURCE    REFSEQ: accession NM_001091.1
KEYWORDS    .
SOURCE      Homo sapiens (human)
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Euarchontoglires; Primates; Catarrhini;
            Hominidae; Homo.
REFERENCE   1  (residues 1 to 751)
  AUTHORS   Olive,M., Unzeta,M., Moreno,D. and Ferrer,I.
  TITLE     Overexpression of semicarbazide-sensitive amine oxidase in human
            myopathies
  JOURNAL   Muscle Nerve 29 (2), 261-266 (2004)
   PUBMED   14755492
  REMARK    GeneRIF: Semicarbazide-sensitive amine oxidase is a source of
            oxidative stress in diseased human skeletal muscle; it contributes
            to oxidative stress-induced damage in various inflammatory and
            other myopathies.
...
FEATURES             Location/Qualifiers
     source          1..751
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
                     /chromosome="7"
                     /map="7q34-q36"
     Protein         1..751
                     /product="amiloride binding protein 1 precursor"
                     /EC_number="1.4.3.6"
                     /note="diamine oxidase; Amiloride-binding protein-1"
     sig_peptide     1..19
     mat_peptide     20..751
                     /product="amiloride-binding protein 1"
     CDS             1..751
                     /gene="APB1"
                     /coded_by="NM_001091.1:72..2327"
                     /note="go_component: peroxisome [goid 0005777] [evidence
                     NAS] [pmid 1356107];
                     go_function: drug binding [goid 0008144] [evidence NR];
                     go_function: heparin binding [goid 0008201] [evidence
                     IEA];
                     go_function: copper ion binding [goid 0005507] [evidence
                     IEA];
                     go_function: oxidoreductase activity [goid 0016491]
                     [evidence IEA];
                     go_function: amine oxidase activity [goid 0008131]
                     [evidence TAS] [pmid 8144586];
                     go_process: metabolism [goid 0008152] [evidence NR]"
                     /db_xref="GeneID:26"
                     /db_xref="MIM:104610"
ORIGIN
        1 mpalgwavaa ilmlqtamae pspgtlprka gvfsdlsnqe lkavhsflws kkelrlqpss
       61 tttmaknvvf liemllpkky hvlrfldkge rhpvrearav iffgdqehpn vtefavgplp
      ...
      721 ngpnyvqrwi pedrdcsmpp pfsyngtyrp v
//
```

Figura 2.6: Ejemplo de anotación de GO.

Capítulo 3

El problema computacional: análisis y explicación de perfiles de expresión genética

Para el análisis y explicación de perfiles de expresión genética derivados de experimentos de Microarrays, es necesario utilizar información externa a la expresión. El objetivo es encontrar conceptos que puedan ser utilizados para explicar y distinguir situaciones que parecen indistinguibles si solo se observan los datos de expresión [RZCRE+].

En este sentido, existen varias herramientas y técnicas de *data mining* que pueden utilizarse [HK00]. En la sección 3.1 se presentan los conceptos de data mining, haciendo hincapié en las características de las bases de datos biológicas que las hacen propicias para el empleo de estas técnicas. Dentro de las herramientas de data mining que utilizan la información de *Gene Ontology* para el análisis de perfiles de expresión genética se encuentra FatiGO [FASD04] (sección 3.2), que es una herramienta web desarrollada en la unidad de bioinformática del CNIO *(Centro Nacional de Investigaciones Oncológicas, Madrid, España)*. En la sección 3.3 se explican las técnicas de data mining genéricas que se utilizan para la identificación de patrones y los conceptos involucrados en optimización multiobjetivo.

El capítulo muestra detalladamente la aplicación de FatiGO a un ejemplo, destacando los aspectos del funcionamiento de esta herramienta que pueden ser mejorados.

3.1. Data Mining en biología

Minería de Datos o Data mining, también conocido como KDD (Knowledge Discovery in Databases), se puede definir como la actividad de realizar "extracción no trivial de información implícita, desconocida previamente, y potencialmente útil desde los datos", y consiste en el conjunto de técnicas avanzadas para la extracción de información oculta en grandes bases de datos [HK00].

Uno de los campos que ha experimentado un gran avance en los últimos tiempos y en el cual existen varios problemas para los cuales estas técnicas resultan valiosas es la biología. En este sentido el comienzo de la utilización de Microarrays de ADN ha dado como resultado una gran cantidad de datos a la espera de ser analizados eficazmente para la obtención de información relevante.

En la actualidad los laboratorios de bioinformática se encuentran en la etapa de desarrollo de herramientas computacionales capaces de ayudar a los biólogos a analizar la vasta cantidad de información generada por diversos experimentos.

Existen muchas bases de datos biológicas, sin embargo no existe consenso a la hora de identificar los genes o producto de genes que se encuentran almacenados. Por lo tanto, la manipulación de información resulta una tarea complicada, especialmente para el desarrollo de métodos computacionales capaces de resolver problemas biológicos.

Asimismo, la utilidad de estas bases de datos se vuelve parcialmente limitada debido a la incapacidad de poder ser utilizadas para búsquedas teniendo en cuenta la experiencia de sus usuarios. En particular, hay una carencia de métodos de representación adecuados para facilitar el entendimiento de los objetos representados y sus sistemas relacionados. Generalmente las estructuras provistas para organizar y buscar información reflejan la conveniencia de los implementadores de bases de datos. La tendencia de estos es basarse en técnicas para introducir una enorme cantidad de información, sin embargo las mismas no proveen un mecanismo para hacer inferencias en nuevas hipótesis y poder, de esta manera, realizar predicciones en base a estas. En otras palabras, las técnicas actuales están más ligadas a la eficiencia computacional que a proveer mecanismos que permitan entender las características estructurales y funcionales de los problemas biológicos.

Por lo anterior la biología, en particular los problemas de interpretación de expresión genética derivados de experimentos de Microarray, son un campo propicio para el empleo de estas técnicas y el desarrollo de nuevas herramientas que se adapten mejor a este dominio [ZSK+05, ZHG05]. Actualmente existen varias herramientas que se utilizan para analizar estos datos, algunas utilizando la información de GO. En la próxima sección se presentará una de estas herramientas, la cual ha sido tomada como punto de partida para la presente tesis.

3.2. FatiGO: data mining utilizando *Gene Ontology*

FatiGO [FAT] es una herramienta web que utiliza la base de datos estructurada de *Gene Ontology* para realizar un análisis estadístico de uno o dos grupos de genes.

Con un único grupo de genes de entrada, la herramienta realiza una búsqueda en diversas bases de datos para encontrar los términos GO de cada gen y luego realiza un histograma con la frecuencia de aparición de cada término en el grupo de genes. Los resultados se pueden visualizar ordenados por genes o por términos GO. Con dos grupos de genes de entrada, el objetivo es extraer términos GO relevantes en un grupo de genes con respecto a otro conjunto de genes de referencia. Para esto primeramente se calcula la frecuencia de cada término GO en cada uno de los grupos y luego, para considerar los términos relevantes, se aplica un test exacto de Fisher [FASD04].

FatiGO puede trabajar con miles de genes de diferentes organismos (humano, ratón, levadura, gusano, etc.). Para resolver el problema de las diferentes maneras de nombrar genes utilizadas por los fabricantes de Microarrays, se utilizan códigos Xref EBI para relacionar identificadores GenBank, ENSEMBL y Unigene a SwissProt/TREMBL, que luego son asociados a GO por medio de las tablas annotations@EBI [FASD04]. Para algunas bases de datos se usan tablas que tienen directamente la correspondencia entre genes y GO. Este es el caso de affymetrix, que justamente es el código en el cual están anotados los genes del experimento biológico que se analizará en este trabajo.

A fin de analizar detalladamente el funcionamiento de esta herramienta, se utilizará un ejemplo sencillo. Supongamos que se quiere analizar el conjunto de genes:

```
G = {200017_at, 200018_at, 200025_s_at, 200029_at, 200038_s_at,
     200061_s_at, 200081_s_at, 200088_x_at, 200092_s_at,
     200099_s_at, 200725_x_at, 200763_s_at}
```

y se quiere encontrar si existe alguna relación relevante en las funciones moleculares en las cuales están involucrados estos genes. Para operar con FatiGO se debe elegir qué

ontología utilizar, en este caso *función molecular (molecular function)*, y a qué nivel se desea trabajar. Para este ejemplo se trabajará a nivel 3.

La primera tarea que realiza FatiGO es asociar términos GO con cada gen de entrada, realizando una búsqueda en diversas bases de datos. La herramienta es muy potente es este sentido.

Una vez encontrados para cada gen los correspondientes términos GO, estos pueden no pertenecer al nivel en el cual se está trabajando, entonces la herramienta sube en la jerarquía por todas las ramas posibles hasta llegar al nivel deseado y extraer los términos de este nivel (ver Figura 3.1).

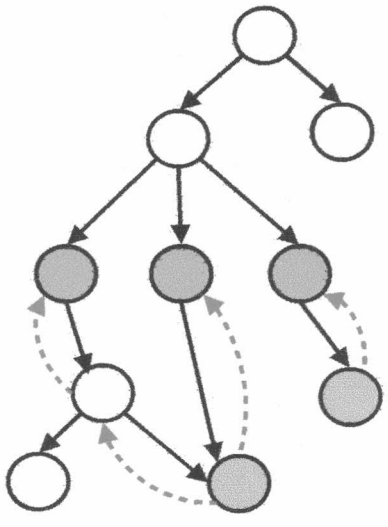


Figura 3.1: Ejemplo de búsqueda de términos GO a nivel 3 en FatiGO. Los círculos celeste son los términos que le corresponderían al gen en cuestión y los naranja son los términos que FatiGO encuentra para ese gen a nivel 3. Las flechas rojas indican como la herramienta sube por las diferentes ramas en la jerarquía hasta el nivel buscado.

Luego de realizar la búsqueda en la base de datos, las referencias a funciones moleculares a nivel 3 que encontró FatiGO se muestran en la Tabla 3.1.

Funciones moleculares	**Genes**
GO:0003676: *nucleid acid binding*	200088_x_at, 200763_s_at, 200029_at, 200038_s_at
GO:0003735: *structural constituent of ribosome*	200088_x_at, 200061_s_at, 200763_s_at, 200029_at 200017_at, 200025_s_at, 200081_s_at, 200092_s_at 200038_s_at, 200099_s_at, 200018_at, 200725_x_at
GO:0016491: *oxidoreductase activity*	200038_s_at

Tabla 3.1: Referencias a funciones moleculares a nivel 3 encontradas por FatiGO.

La herramienta devuelve una tabla similar a la anterior y además crea un histograma teniendo en cuenta la cantidad de genes que tiene asociado cada término GO (Figura 3.2).

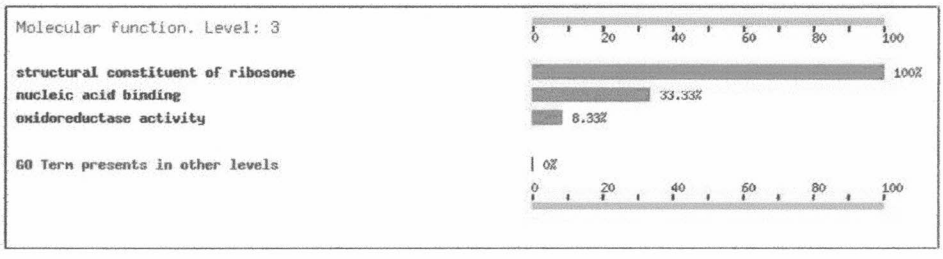


Figura 3.2: Histograma producido con FatiGO.

Mejoras a FatiGO

Existen dos aspectos en FatiGO que fueron tomados como punto de partida para intentar mejorarlos y de esta manera proponer un nuevo método:

- En FatiGO se tiene que elegir a qué nivel trabajar y los resultados se calculan a este nivel. Supongamos, por ejemplo, que en un conjunto de genes se quiere ver si existe algún patrón interesante en sus términos GO, se podría elegir trabajar a nivel 3, pero a priori no se sabe si existe alguna relación interesante a nivel 2 o 4, o también pueden existir grupos interesantes que combinen niveles. Estos últimos grupos nunca podrán ser detectados con FatiGO porque no existe manera de combinar niveles de la jerarquía.

- En FatiGO es necesario elegir con qué ontología trabajar. Se tiene que ejecutar tres veces FatiGO para poder obtener resultados en las tres ontologías. Cuando se analizan los términos GO puede que existan genes que tienen algún patrón interesante si se miran las tres ontologías al mismo tiempo. Esto es bastante difícil de ver utilizando FatiGO sin la utilización en alguna herramienta adicional para cruzar los resultados.

El método propuesto en este trabajo tiene como uno de sus objetivos mejorar estos aspectos de FatiGO.

3.3. Preliminares

En esta sección se presentarán los conceptos fundamentales que se utilizan como base para el método propuesto en esta tesis. En la primer sección se presenta brevemente el estado del arte en el tema de identificación de patrones en base de datos. Luego se introducen los conceptos involucrados en optimización multiobjetivo que, junto con clustering conceptual, constituyen las técnicas principales utilizadas en esta tesis.

3.3.1. Métodos utilizados para identificación de patrones

Diversos métodos se utilizan en la actualidad como técnicas de *data mining* para extraer conocimiento a partir de objetos almacenados en bases de datos. La tarea

consiste en clasificar estos objetos, es decir, asignar cada uno a determinada clase basándose en algún tipo de información de los mismos.

Los **métodos supervisados**, aunque bastante utilizados, presentan el problema de que necesitan una fuente de información previa obtenida por expertos del área para luego poder utilizar la misma en la clasificación de los objetos. Ejemplos de estos métodos son los árboles de decisión y las redes neuronales supervisadas.

Debido al inconveniente de los métodos supervisados surgen los **métodos no supervisados**. Dentro de este tipo de métodos diversas técnicas de *clustering* (ej: *k-means*, *hierarchical clustering* [Mit97, WF99]) fueron desarrolladas y aplicadas a una gran variedad de dominios [AK88, Ras92, LB95]. La idea del *clustering* o agrupamiento es la clasificación de objetos de acuerdo a la similitud entre ellos. Un cluster, por lo tanto, es un grupo de objetos que son más similares entre sí en comparación a cada uno de los miembros de otros clusters. La "similitud" se debe entender como un concepto matemático medido formalmente utilizando una función definida sobre las propiedades de las entidades que están siendo comparadas.

Muchas de las técnicas de clustering conocidas presentan inconvenientes para trabajar con datos estructurados, donde no solo existen objetos sino también relaciones entre las características estos. Por otro lado, también agrupan objetos basándose únicamente en medidas numéricas de similitud entre ellos, como por ejemplo la distancia; no se tiene en cuenta ninguna propiedad global o concepto para caracterizar las clases. Consecuentemente, las clases resultantes no tienen una descripción conceptual clara y pueden ser difíciles de interpretar.

Para presentar un ejemplo, observemos la Figura 3.3. La mayoría los métodos de clustering tradicionales ubicarían en el mismo cluster a los objetos A y B, puesto que están muy cerca uno del otro con respecto a los otros puntos. Sin embargo, se puede notar que los puntos se deberían agrupar formando dos elipses, luego A y B no estarían en el mismo cluster.

Estos inconvenientes dan origen al **clustering conceptual** [RR84]. Desde el punto de vista de esta técnica, la similitud entre dos objetos, llamada *cohesividad conceptual*, depende no solo de estos objetos y de los demás objetos que los rodean, sino también de un lenguaje compuesto por "conceptos" que describen los objetos que se están estudiando. Se pueden observar dos ventajas de este tipo de clustering. Por un lado, los objetos serán agrupados de acuerdo a conceptos de un lenguaje externo. Por otro lado permite obtener como resultado no solo clusters de objetos, sino también información que describe conceptualmente a los objetos de dichos clusters. Por lo tanto el proceso de interpretación de los clusters obtenidos mediante está técnica resulta más sencillo.

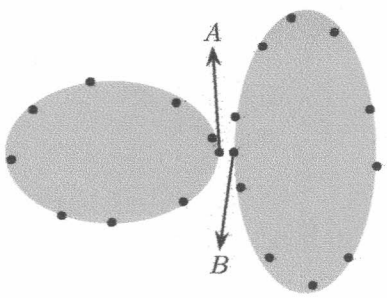


Figura 3.3: Diferencia entre cercanía y cohesión conceptual.

3.3.2. Optimización multiobjetivo

Existen numerosos problemas de optimización en mundo real con varios objetivos contrapuestos que deben satisfacerse al mismo tiempo. En este contexto, la noción de *óptimo* debe ser redefinida debido a que no se busca una única solución óptima, sino un conjunto de soluciones de compromiso entre todos los objetivos a considerar [ZZR04, CVL02].

Formalmente un problema de optimización multiobjetivo puede definirse como: encontrar el vector de variables de decisión $x^* = [x_1^*, x_2^*, ..., x_n^*]$ que satisfaga las restricciones de desigualdad:

$$g_i \geq 0 \quad i = 1, 2, ..., m \tag{3.1}$$

las restricciones de igualdad:

$$h_j = 0 \quad j = 1, 2, ..., p \tag{3.2}$$

y optimice la función objetivo:

$$f(x) = [f_1(x), f_2(x), ..., f_k(x)] \tag{3.3}$$

La definición más común de optimalidad utilizada es la *optimalidad de Pareto* [CVL02]. Decimos que un vector de variables de decisión x^* es *óptimo* si no existe otro vector x tal que $f_i(x) \geq f_i(x^*)\ \forall i = 1, 2, ..., k$ y $f_j(x) > f_j(x^*)$ para al menos un j. Coloquialmente, decimos que un vector de variables de decisión es óptimo si no existe otro vector en el espacio de posibles soluciones que incremente algún criterio sin causar una disminución en al menos uno de los otros criterios.

Este concepto de optimalidad raramente nos devuelve una única solución, más bien hay un conjunto de soluciones que se llama *conjunto Pareto*. Las soluciones que pertenecen a este conjunto se dice que son *no dominadas* entre sí. El resto de las soluciones son *dominadas*. La siguiente notación se utilizará para la *dominancia* entre soluciones:
Se dice que una solución b *domina* a otra solución a (también escrito como $a \prec b$) sii

$$\begin{array}{l} \forall i \in \{1, \ldots, n\} : f_i(a) \leq f_i(b) \quad \wedge \\ \exists j \in \{1, \ldots, n\} : f_j(a) < f_j(b) \end{array}$$

Adicionalmente, se dice que b cubre a $(a \preceq b)$ sii $a \prec b$ o $f(a) = f(b)$.

3.4. Observaciones finales

Para el análisis y explicación de perfiles de expresión genética es necesario utilizar información externa que permitan distinguir situaciones que parecen indistinguibles si solo se observan los datos de expresión. En el presente capítulo se han presentado diversas técnicas de data mining que pueden utilizarse para este fin. Dentro de las herramientas de data mining que utilizan la información de *Gene Ontology* para el análisis de perfiles de expresión genética se ha presentado FatiGO

FatiGO ha sido aplicado a un pequeño ejemplo. Se han señalado dos aspectos en los cuales esta herramienta puede ser mejorada. Por un lado, el problema de elegir un nivel adecuado para trabajar y de no utilizar combinación de niveles. Por otro lado, el problema de no considerar las ventajas que puede brindar la combinación de las tres ontologías para encontrar grupos de genes que compartan términos GO. Asimismo, se han considerado varias técnicas tradicionales de data mining. Algunas de estas técnicas presentan inconvenientes a la hora de manejar datos estructurados, mientras que otras no producen información conceptual o cualitativa sobre los clusters obtenidos.

Finalmente, se han presentado los conceptos de clustering conceptual y optimización multiobjetivo, dado que estas técnicas son utilizadas en el presente trabajo de tesis.

Capítulo 4

GO-GPS: Un nuevo método para el análisis de perfiles de expresión genética

Como resultado de un experimento de Microarray se pueden obtener perfiles de expresión genética [EZ05]. Por otro lado, utilizando la información de *Gene Ontology* se pueden encontrar características biológicas comunes en grupos de genes y esta información puede resultar útil para explicar estos perfiles de expresión. Existen herramientas de data mining que podrían ser utilizadas para analizar este tipo de información. Sin embargo, muchos de estos métodos necesitan información supervisada o tienen inconvenientes al tratar datos estructurados. Otra desventaja de estos métodos es que los grupos resultantes no están acompañados de ningún tipo de información adicional que facilite su interpretación. Otras herramientas como FatiGO [FASD04] utilizan *Gene Ontology*, pero pueden ser mejoradas en cuanto a la calidad y cantidad de términos GO de cada cluster de genes obtenido.

En el presente capítulo se presentará el método GO-GPS (sección 4.1) por *Gene Ontology Grouping, Prototyping and Searching*, el cual encuentra grupos cohesivos de genes que comparten procesos biológicos, funciones moleculares y componentes celulares (*Grouping*), formando así modelos (*Prototyping*) para poder utilizarlos como patrones de búsqueda en la clasificación de nueva información (*Searching*). GO-GPS es un algoritmo exhaustivo basado en clustering conceptual [RR84] y en el uso de herramientas de optimización multiobjetivo [ZZR04, CVL02].

GO-GPS se aplica a un subconjunto de 62 genes del conjunto de datos principal del experimento biológico, utilizando la base de datos estructurada de *Gene Ontology*. En la sección 4.2 se analizarán distintas funciones objetivo con el fin de encontrar la función con la cual se pueda obtener un clustering que represente nuestras expectativas.

4.1. El método GO-GPS

El método propuesto en este trabajo, GO-GPS (*Gene Ontology Grouping, Prototyping and Searching*), recibe como entrada un conjunto de genes y la base de datos estructurada de GO. Basado en técnicas de optimización multiobjetivo, realiza *clustering conceptual* en el conjunto de genes (*Grouping*) utilizando técnicas de optimización multiobjetivo. Estos grupos comparten procesos biológicos, funciones moleculares y componentes celulares extraídos de la base de datos de GO y de esta manera sirven

como modelos (*Prototyping*) que pueden ser utilizados como patrones de búsqueda en la clasificación de nueva información (*Searching*).

Los resultados obtenidos utilizando optimización multiobjetivo principalmente exclusivamente de la definición de la función objetivo, y esta, a su vez, depende del problema particular que se desea resolver. Por este motivo, en esta sección solo se estudiará el algoritmo principal del método, mientras que la sección siguiente está dedicada íntegramente al estudio de funciones objetivo para el problema biológico que se pretende resolver.

4.1.1. Datos de entrada

El algoritmo recibe como entrada dos fuentes principales de datos. Una de estas fuentes es una base de datos estructurada (en nuestro problema esta base es GO) que es representada como un grafo dirigido acíclico (GDA). Esta base de datos tiene la particularidad de que está dividida en tres ontologías: *proceso biológico (biological process), función molecular (molecular function) y componente celular (cellular component)* las cuales están unidas por un nodo en común denominado *all* [Con00]. En nuestro método se tratará a cada una de estas ontologías como un GDA separado.

La segunda fuente de datos es el conjunto de *genes* entre los cuales se desea encontrar grupos cohesivos ó clusters. Además se conocen los términos GO que tienen asociados estos genes en las tres ontologías.

4.1.2. Algoritmo principal

La idea principal del algoritmo es utilizar la información contenida en los datos estructurados de GO para generar modelos o conceptos que luego puedan ser utilizados para agrupar las instancias que cubren. En otras palabras, los modelos son subgrafos del grafo GO y las instancias cubiertas son aquellos genes que tienen todos los términos GO que componen el modelo.

Una de las tareas que debe realizar el algoritmo es ir generando modelos utilizando GO. Cada uno de estos modelos junto con las instancias que cubre es una *potencial solución* la cual, formalmente, es una tupla $< Genes, BP, MF, CC >$ donde:

- *Genes:* es el conjunto de genes asociados con esta potencial solución particular
- *BP:* son los términos de *GO* correspondientes a la ontología *proceso biológico* que todos los genes tienen en común.
- *MF:* son los términos de *GO* correspondientes a la ontología *función molecular* que todos los genes tienen en común.
- *CC:* son los términos de *GO* correspondientes a la ontología *componente celular* que todos los genes tienen en común.

Un ejemplo de una potencial solución es el siguiente:

```
<{202649_x_at, 213414_s_at}, {GO:0003015}, {GO:0005488},
                                {GO:0005622, GO:0004322}>
```

esta tupla indica que los genes `202649_x_at` y `213414_s_at` comparten el proceso biológico `GO:0003015`, la función molecular `GO:0005488` y los componentes celulares `GO:0005622, GO:0004322`.

En una potencial solución únicamente se presentan en forma explícita aquellos términos *GO* más específicos dentro de la jerarquía, con lo cual, ninguno de estos términos

pueden encontrarse en la misma rama. Sin embargo, los genes de la potencial solución no solo comparten estos términos sino también todos sus ancestros. Esto surge naturalmente puesto que el grafo *GO* representa una jerarquía donde los términos se van haciendo más específicos a medida que la profundidad en el grafo aumenta (ver Figura 4.1).

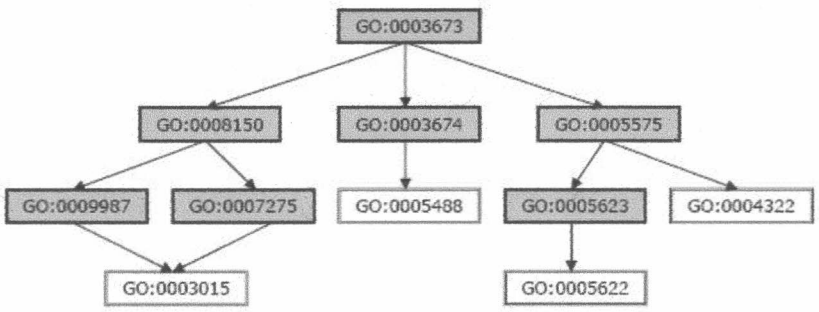


Figura 4.1: subárbol de la jerarquía *GO* para los términos GO:0003015, GO:0005488, GO:0005622 y GO:0004322.

El objetivo del algoritmo de optimización es encontrar el *conjunto Pareto* entre todas las posibles soluciones dada una configuración de entrada. GO-GPS es un método exhaustivo que va formando todos los modelos posibles con los datos estructurados, encuentra las instancias cubiertas por cada modelo y luego calcula los valores de la *función objetivo* para conservar únicamente aquellas soluciones *no dominadas* entre sí.

Las tareas que el algoritmo debe realizar son las siguientes:

- Calcular todos los subgrafos de cada ontología de GO, notemos a estos BP^j (subgrafo j de BP), MF^k (subgrafo k de MF), CC^m (subgrafo m de CC). Los subgrafos con los que se trabajará cumplen con la propiedad de que si incluyen a un nodo, entonces también incluyen a todos los ancestros de este. Luego, cada uno de estos subgrafos queda totalmente caracterizado por las hojas que contiene, puesto que conociendo las hojas el resto del subgrafo puede deducirse desde el grafo completo de GO.
- Para cada combinación de subgrafos BP^j, MF^k, CC^m (el grafo que queda formado es el modelo), calcular las instancias (*Genes*) que están cubiertas por este modelo. De este modo se obtiene la potencial solución S:
  $S =< Genes, BP^j, MF^k, CC^m >$
- Calcular los valores de la función objetivo para cada potencial solución S.
- Conservar únicamente aquellas soluciones que sean *no dominadas* entre sí, en otras palabras, encontrar el conjunto Pareto.

El algoritmo 4.1 es utilizado para la optimización de Pareto. Este algoritmo utiliza la función *domina_a* para decidir si una solución domina a otra dada o no[1].

[1] En el algoritmo se expone el pseudocódigo. La implementación actual del método utiliza programación dinámica [CLRS01, AHU82] y tiene en cuenta varias optimizaciones que pueden hacerse al trabajar con el problema biológico en cuestión.

Algoritmo 4.1 Cálculo del conjunto Pareto

entrada $Genes, BP, MF, CC$
salida conjunto pareto: $Pareto$

$resp \leftarrow \emptyset$
$L_1 \leftarrow$ listado de subgrafos de BP
$L_2 \leftarrow$ listado de subgrafos de MF
$L_3 \leftarrow$ listado de subgrafos de CC

para cada elemento g_1 de L_1 **hacer**
  para cada elemento g_2 de L_2 **hacer**
    para cada elemento g_3 de L_3 **hacer**
      $Genes_s \leftarrow \{x \in Genes / g_1$ *cubre a* $x \wedge g_2$ *cubre a* $x \wedge g_3$ *cubre a* x
      $s \leftarrow < Genes_s, g_1, g_2, g_3 >$ //armamos la potencial solución
      $esNoDominada \leftarrow$ **true**
      $ARemover \leftarrow \emptyset$ //contiene las soluciones que serán removidas de $Pareto$
      si $Pareto = \emptyset$ **entonces**
        $Pareto \leftarrow Pareto \cup \{s\}$
      si no
        para cada elemento r de $Pareto$ **hacer**
          mientras $esNoDominada$ sea **true hacer**
            si r *domina_a* s **entonces**
              $esNoDominada \leftarrow$ **false**
            fin si
            si $esNoDominada$ es **true y** s *domina_a* r **entonces**
              $ARemover \leftarrow ARemover \cup \{r\}$
            fin si
          fin mientras
          si $esNoDominada$ es **true entonces**
            si $ARemover \neq \emptyset$ **entonces**
              $Pareto \leftarrow Pareto - ARemover$
            fin si
            $Pareto \leftarrow Pareto \cup \{s\}$
          fin si
        fin para
      fin si
    fin para
  fin para
fin para
retornar $ParetoSet$

Función *domina_a*:

$$domina\_a(r,s) = true \iff$$
$$(sensitividad(r) \geq sensitividad(s) \wedge especificidad(r) \geq especificidad(s)) \quad \wedge$$
$$(sensitividad(r) > sensitividad(s) \vee especificidad(r) > especificidad(s))$$

Esta función es la utilizada para determinar si una solución domina a otra o no, teniendo en cuenta la definición de *optimización de Pareto*. Los objetivos que se están maximizando son *sensibilidad* y *especificidad*, los cuales serán definidos en la siguiente sección.

La complejidad del algoritmo 4.1 depende directamente del subgrafo de GO determinado por los genes de entrada. Llamemos a este grafo $G = (V, E)$ donde V es el conjunto de nodos y E es el conjunto de ejes. El conjunto de nodos está formado por los términos GO de las tres ontologías V_{BP}, V_{MF} y V_{CC}, luego $V = V_{BP} \cup V_{MF} \cup V_{CC}$. Por lo tanto, la cantidad de nodos n del subgrafo G está dada por la expresión: $n = \#(V_{BP}) + \#(V_{MF}) + \#(V_{CC})$.

La implementación actual del algoritmo calcula, en el paso k, la cantidad de subgrafos de k nodos que cubren un conjunto no vacío de genes. Luego, en el paso $k+1$ utiliza las soluciones del paso anterior para combinarlas con todos los nodos y calcular los nuevos modelos. La complejidad temporal en peor caso es:

$$O(2^{\#(V_{BP})} \times 2^{\#(V_{MF})} \times 2^{\#(V_{CC})}) = O(2^{\#(V_{BP})+\#(V_{MF})+\#(V_{CC})}) = O(2^n)$$

El algoritmo almacena en cada paso las soluciones que tienen cubrimiento no vacío, por lo tanto en peor caso, la complejidad espacial es $O(2^n)$. Esta complejidad puede reducirse a $O(1)$ si no se almacenan resultados intermedios, pero el algoritmo tarda mucho más tiempo de ejecución.

Si bien estos valores son una cota para el peor caso del algoritmo, el tiempo de procesamiento y el espacio requerido disminuyen notablemente realizando las siguientes acciones en cada paso:

- Las nuevas potenciales soluciones que no cubren genes son desechadas, puesto que no sirven como modelo y ningún nuevo modelo puede generarse a partir de las mismas.
- Las nuevas soluciones son generadas a partir de las soluciones del paso anterior, combinándolas con cada uno de los nodos. Si alguno de los nodos de la potencial solución es ancestro del nuevo nodo que se combina, entonces esta nueva solución es desechada puesto que es un caso que ya se ha tenido en cuenta anteriormente.

4.2. Análisis de funciones objetivo para el problema biológico

Como se ha dicho anteriormente, el mayor desafío a la hora de realizar optimización multiobjetivo es encontrar la función objetivo adecuada con el fin de obtener los resultados deseados. En esta sección se presentan los estudios realizados con el subconjunto de 62 genes del cluster de expresión 20 a fin de obtener una función objetivo que luego será aplicada al conjunto completo de genes.

Existen características que permiten evaluar cualitativamente un clustering conceptual [JCH01]. Una propiedad deseable es que el clustering tenga la mayor cobertura

con el mínimo número posible de clusters, lo cual implica que los clusters sean lo suficientemente generales para cubrir todos los datos, pero a su vez continúen definiendo conceptos individuales. Otra propiedad deseable es obtener descripciones con varias características para cada cluster aumentando de esta manera el poder de inferencia. La tercer propiedad que se busca es obtener el mínimo solapamiento entre los clusters. Generalmente estas tres propiedades son conflictivas, cuanto más características tienen las descripciones, mas alta es la probabilidad de que dos clusters compartan estas características y por lo tanto se solapen.

Encontrar los objetivos para obtener un clustering conceptual de alta calidad es dificultoso, por este motivo se han analizado varias funciones objetivo distintas. Para cada una de las funciones estudiadas se explicará la manera en que es calculado cada objetivo y se mostrarán y analizarán los resultados obtenidos al aplicar la función al conjunto de 62 genes.

4.2.1. Cálculo de objetivos para la evaluación de clusters

En el presente trabajo se evalúa la calidad de los clusters de una potencial solución $< Genes, BP, MF, CC >$ en base a dos criterios u objetivos diferentes, su *sensibilidad* y *especificidad*. Las definiciones preliminares de estos objetivos se presentan en las fórmula 4.1 a y 4.1 b:

$$sensibilidad = \frac{\#(Genes)}{\#(GenesTotales)} \tag{4.1 a}$$

$$especificidad = \frac{\frac{\sum_{e \in BP} \ell(e)}{h_{BP_{GO}}} + \frac{\sum_{e \in MF} \ell(e)}{h_{MF_{GO}}} + \frac{\sum_{e \in CC} \ell(e)}{h_{CC_{GO}}}}{\#(BP) + \#(MF) + \#(CC)} \tag{4.1 b}$$

El objetivo *sensibilidad* mide la cantidad de genes que están cubiertos por el modelo GO del cluster. Esta cantidad se divide por el total de genes de la instancia del problema para normalizarla ($\#(GenesTotales)$).

El objetivo *especificidad* mide cuan profundo es el modelo dentro de la jerarquía *GO*. Para hacer esto tenemos que basarnos en el nivel $\ell(e)$ de cada término e en la jerarquía[2]. Además, al nivel de cada término lo dividimos por la profundidad total de la ontología a la cual pertenece ($h_{BP_{GO}}$ para BP, $h_{MF_{GO}}$ para MF y $h_{CC_{GO}}$ para CC) para así normalizarlo. Los valores actuales para estas cantidades son: $h_{BP_{GO}} = 17$, $h_{MF_{GO}} = 14$ y $h_{CC_{GO}} = 15$. Luego de obtener estos valores realizamos un promedio sobre todos los términos de la solución.

Estos objetivos son contrapuestos debido a que cuando se aumenta uno de ellos el otro disminuye. A medida que una solución se hace más específica, necesariamente pierde sensibilidad puesto que la cantidad de genes cubiertos disminuye. Por el contrario, las soluciones poco específicas cubren a una mayor cantidad de genes.

Aplicación y análisis de resultados

Con la aplicación de los objetivos *sensibilidad* y *especificidad* se han obtenido 8 clusters (ver Tabla 4.1). Cada uno de estos clusters tiene distintos tipos de términos a distintos niveles (e.g. el cluster 1 tiene el proceso biológico `GO:0006444` que se encuentra a niveles 10 y 9, y el cluster 6 tiene el componente celular `GO:0005737` cuyo nivel es 1). Estos resultados son posibles gracias a que el método explora la jerarquía GO en

[2] En el grafo de *GO* los términos pueden llegar a estar en más de un nivel. En estos casos $\ell(e)$ es el promedio de todos los niveles en los que el término aparece.

forma completa (i.e. en todos los niveles de las tres ontologías) a diferencia de otras herramientas como FatiGO (sección 3.2), la cual explora únicamente una ontología a un determinado nivel.

Cluster	*BP*	*MF*	*CC*	*sensibilidad*	*especificidad*
1	GO:0006444			0.02	0.58
2	GO:0044267			0.92	0.35
3	GO:0006414			0.05	0.52
4	GO:0009987			0.97	0.12
5	GO:0006412			0.85	0.41
6			GO:0005737	0.95	0.27
7		GO:0003674		0.98	0.07
8			GO:0005842	0.29	0.47

Tabla 4.1: Clusters obtenidos con los objetivos *sensibilidad* y *especificidad*. La tabla presenta los términos GO de procesos biológicos (BP), funciones moleculares (MF) y componentes celular (CC) de cada cluster junto con los valores de los objetivos *sensibilidad* y *especificidad*.

La Figura 4.2 presenta los clusters en el espacio de los objetivos utilizando los valores de *sensibilidad* y *especificidad*. En el Pareto únicamente se encuentran las soluciones no dominadas, por ejemplo los clusters 1 y 6. El cluster 1 tiene valores de *sensibilidad* 0.02 y de *especificidad* 0.58, en el cluster 6 estos valores son 0.95 y 0.27 respectivamente, luego se observa que estas soluciones son no dominadas entre sí ya que ninguna logra aumentar un objetivo sin causar una disminución en el otro.

El Pareto anterior presenta los clusters obtenidos cualitativamente, pero no permite visualizar el tipo de información de GO (i.e. los objetivos engloban a todos los términos ocultando el tipo de ontología y los niveles). La Figura 4.3, en cambio, presenta los clusters en el espacio de variables. Cada cluster se representa con una esfera cuyo diámetro es proporcional a la cantidad de genes del cluster y sus coordenadas indican el promedio de los niveles de los términos GO del cluster en cada una de las tres ontologías. Aquí se pueden apreciar claramente los tipos de términos GO y como están distribuidos en los niveles.

Como se observa en la Tabla 4.1, cada cluster tiene un único término GO asociado. Recordando las propiedades cualitativas de los clusters presentadas en la sección 4.2, no se obtienen descripciones de alta calidad y por lo tanto, el poder de inferencia es bajo. Por otro lado, debido a que las descripciones son muy generales, se han obtenido un número pequeño de clusters lo cual era una propiedad deseable. Se observa entonces que estas propiedades son conflictivas: a mayor generalidad en las descripciones, menor número de clusters. Una mejora sobre estos resultados sería lograr que cada cluster tenga más de un término GO en sus descripciones.

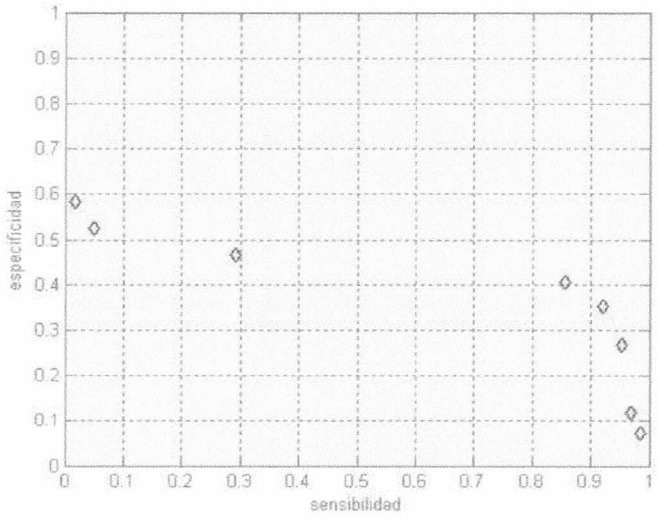


Figura 4.2: Gráfico del conjunto Pareto en el espacio de los objetivos *sensibilidad* y *especificidad*. Este gráfico se corresponde con los valores de la Tabla 4.1.

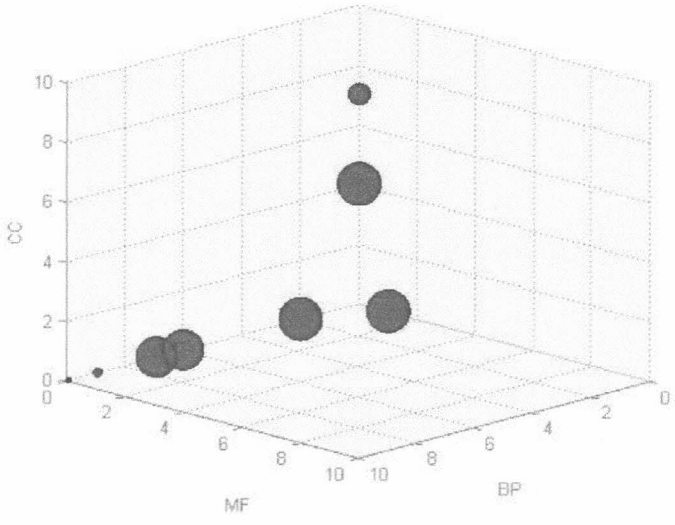


Figura 4.3: Gráfico del conjunto Pareto en el espacio de variables para la función con dos objetivos: *sensibilidad* y *especificidad*. Cada cluster está representado con una esfera cuyo diámetro es proporcional a la cantidad de genes del cluster y sus coordenadas indican el promedio de los niveles de los términos GO del cluster en cada una de las tres ontologías.

4.2.2. Incorporación del objetivo *complejidad*

En los objetivos *sensibilidad* y *especificidad*, utilizados hasta el momento para comparar por dominancia las potenciales soluciones, se tiene en cuenta cuantos *genes* tiene cada una y también cuan específico es el modelo que la describe (profundidad dentro de la jerarquía GO). En este nuevo estudio de objetivos se ha decidido incorporar alguna medida que permita tener en cuenta la cantidad de información de *Gene Ontology* de cada potencial solución. Este nuevo objetivo es la *complejidad* y se mide como la cantidad de términos GO de cada modelo, es decir, la cantidad de nodos del subgrafo de GO de cada potencial solución. De esta manera, los nuevos objetivos quedan definidos con las fórmulas 4.2 a, 4.2 b y 4.2 c.

$$sensibilidad = \frac{\#(Genes)}{\#(GenesTotales)} \tag{4.2 a}$$

$$especificidad = \frac{\frac{\sum_{e \in BP} \ell(e)}{h_{BP_{GO}}} + \frac{\sum_{e \in MF} \ell(e)}{h_{MF_{GO}}} + \frac{\sum_{e \in CC} \ell(e)}{h_{CC_{GO}}}}{\#(BP) + \#(MF) + \#(CC)} \tag{4.2 b}$$

$$complejidad = \frac{\#(BP) + \#(MF) + \#(CC)}{\#(BP_{max}) + \#(MF_{max}) + \#(CC_{max})} \tag{4.2 c}$$

La *complejidad* se calcula como la cantidad de términos GO de la potencial solución divido la máxima cantidad de términos posibles (el máximo entre todas las potenciales soluciones $max = \#(BP_{max}) + \#(MF_{max}) + \#(CC_{max})$ con el fin de normalizar el valor.

Aplicación y análisis de resultados

Los clusters obtenidos tras incorporar la *complejidad* a los objetivos estudiados anteriormente se presentan en la Tabla 4.2. La mayoría de estos grupos presentan términos GO en más de una ontología (e.g. el cluster 1 tiene el proceso biológico `GO:0006412`, la función molecular `GO:0003723` y el componente celular `GO:0005840`). Este hecho también puede observarse en la Figura 4.5, donde se aprecian esferas que no se encuentran sobre alguno de los tres ejes puesto que sus coordenadas en BP, MF y CC son distintas de cero. En la Figura 4.4 se presenta el Pareto correspondiente con los valores de la Tabla 4.2. Como se puede apreciar se ha obtenido como resultado un conjunto numeroso de clusters.

Con respecto a la definición anterior de objetivos, se ha mejorado notablemente la calidad de las descripciones de los clusters. Por ejemplo, el cluster 13 de esta función tiene procesos biológicos `GO:0006412` y `GO:0007516`, funciones moleculares `GO:0003723` y `GO:0003735` y componente celular `GO:0005843`. Comparando con cualquier cluster de la definición de objetivos anterior (ver Tabla 4.1) se observa una mejora sustancial. Es claro que se ha mejorado una de las propiedades que se pretenden de un clustering conceptual, la calidad de las descripciones, causando un aumento del número de clusters al perder generalidad. Con respecto a la otra propiedad deseable, mínimo solapamiento entre clusters, se profundizará a continuación.

Observando la Tabla 4.2 se encuentra que algunos clusters tienen el mismo valor de *sensibilidad*. Este hecho puede apreciarse en la Figura 4.6 donde se dibuja el Pareto pero únicamente con los objetivos *sensibilidad* y *especificidad*. A modo de ejemplo, se puede observar en la Tabla 4.2 en los clusters 19 y 22 que los términos *GO* para dichos clusters son iguales, excepto por la función molecular `GO:0003735`, la cual se

Cluster	*BP*	*MF*	*CC*	*sensibilidad*	*complejidad*	*especificidad*
1	GO:0006412	GO:0003723	GO:0005840	0.52	0.65	0.34
2			GO:0005737	0.95	0.10	0.27
3	GO:0006412		GO:0005737 GO:0043234	0.84	0.44	0.27
4	GO:0006412		GO:0005830	0.48	0.60	0.40
5	GO:0006412		GO:0043229	0.82	0.44	0.32
6	GO:0043123 GO:0006412	GO:0003723	GO:0015934	0.05	0.94	0.37
7	GO:0006412		GO:0005730 GO:0005842	0.05	0.73	0.42
8	GO:0006412	GO:0003723 GO:0003735	GO:0005843	0.13	0.81	0.34
9	GO:0044237 GO:0044238	GO:0003674	GO:0005737	0.92	0.27	0.20
10	GO:0009987			0.97	0.06	0.12
11		GO:0003674		0.98	0.04	0.07
12			GO:0005842	0.29	0.33	0.47
13	GO:0006412 GO:0007516	GO:0003723 GO:0003735	GO:0005843	0.03	0.98	0.34
14	GO:0006414	GO:0003723	GO:0005842	0.02	0.77	0.43
15	GO:0006378	GO:0008143		0.02	0.50	0.50
16	GO:0006414			0.05	0.38	0.52
17	GO:0043123 GO:0006412	GO:0003723 GO:0003735 GO:0004871	GO:0015934	0.05	1.00	0.31
18	GO:0006412 GO:0007516		GO:0005843	0.03	0.85	0.40
19	GO:0006412	GO:0003723 GO:0003735	GO:0005830	0.47	0.73	0.33
20	GO:0044237 GO:0044238			0.95	0.17	0.24
21	GO:0006412		GO:0005842	0.29	0.65	0.44
22	GO:0006412	GO:0003723	GO:0005830	0.47	0.69	0.37
23	GO:0006412		GO:0005843	0.13	0.69	0.42
24	GO:0043123 GO:0006412		GO:0015934	0.05	0.85	0.40
25	GO:0006412	GO:0003735	GO:0005840	0.81	0.63	0.32
26	GO:0006444			0.02	0.40	0.58
27	GO:0044267			0.92	0.25	0.35
28	GO:0006412			0.85	0.33	0.41
29	GO:0006412		GO:0005737	0.84	0.42	0.34
30	GO:0006412	GO:0003674	GO:0005737 GO:0043234 GO:0043229	0.82	0.50	0.22
31	GO:0006412		GO:0005840	0.81	0.56	0.37
32	GO:0006412	GO:0003735	GO:0005830	0.48	0.67	0.34
33	GO:0006414		GO:0005842	0.02	0.69	0.50
34	GO:0044237		GO:0005737	0.94	0.23	0.25
35	GO:0006412		GO:0005737 GO:0043229	0.82	0.46	0.30
36	GO:0044267		GO:0005737	0.90	0.33	0.31
37	GO:0043037			0.06	0.35	0.47
38	GO:0006412		GO:0005737 GO:0043234 GO:0043229	0.82	0.48	0.26
39	GO:0006412	GO:0003723	GO:0005843	0.13	0.77	0.38
40	GO:0006412	GO:0003723 GO:0003735	GO:0005842	0.27	0.77	0.34
41	GO:0006412	GO:0003735	GO:0005842	0.29	0.71	0.36
42	GO:0044237 GO:0044238		GO:0005737	0.94	0.25	0.25
43	GO:0043123 GO:0006412	GO:0003723 GO:0003735	GO:0015934	0.05	0.98	0.34
44	GO:0006412	GO:0003723 GO:0003735	GO:0005840	0.52	0.69	0.31
45	GO:0044267	GO:0003674	GO:0005737	0.89	0.35	0.23
46	GO:0006412	GO:0003723	GO:0005842	0.27	0.73	0.39
47	GO:0006412 GO:0007516	GO:0003723	GO:0005843	0.03	0.94	0.37

Tabla 4.2: Términos *GO* de los clusters obtenidos con los objetivos *sensibilidad*, *especificidad* y *complejidad*.

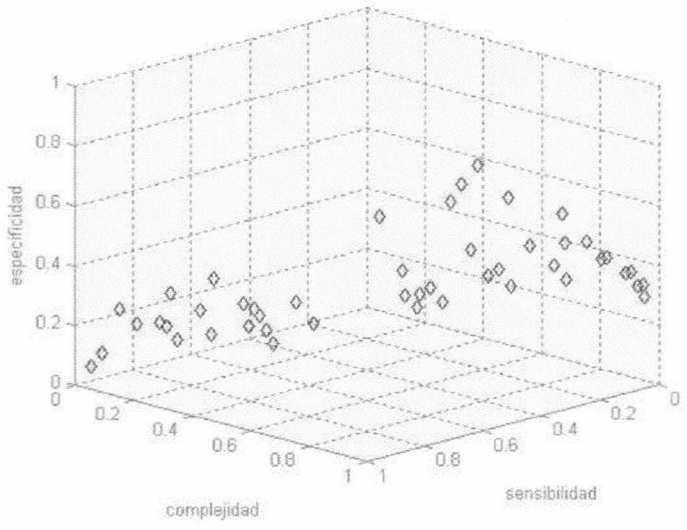


Figura 4.4: Gráfico del conjunto Pareto en el espacio de objetivos con la incorporación de la *complejidad* en la optimización. Este gráfico se corresponde con los valores de la Tabla 4.2.

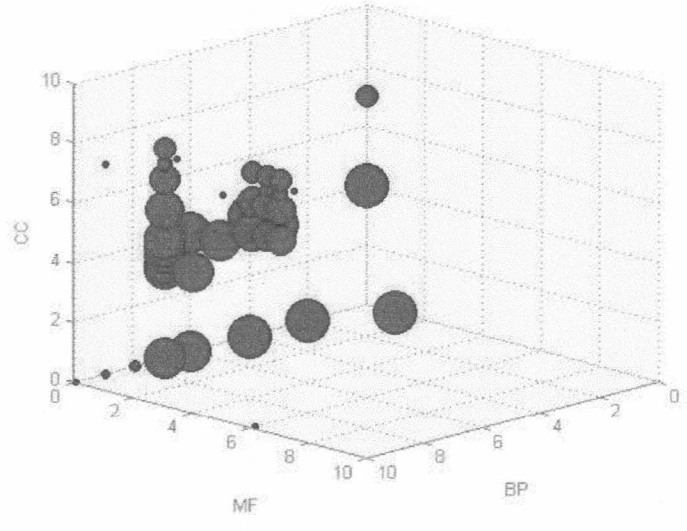


Figura 4.5: Gráfico del conjunto Pareto en el espacio de variables para los objetivos *sensibilidad*, *complejidad* y *especificidad*. Cada cluster está representado con una esfera cuyo diámetro es proporcional a la cantidad de genes del cluster y sus coordenadas indican el promedio de los niveles de los términos GO del cluster en cada una de las tres ontologías.

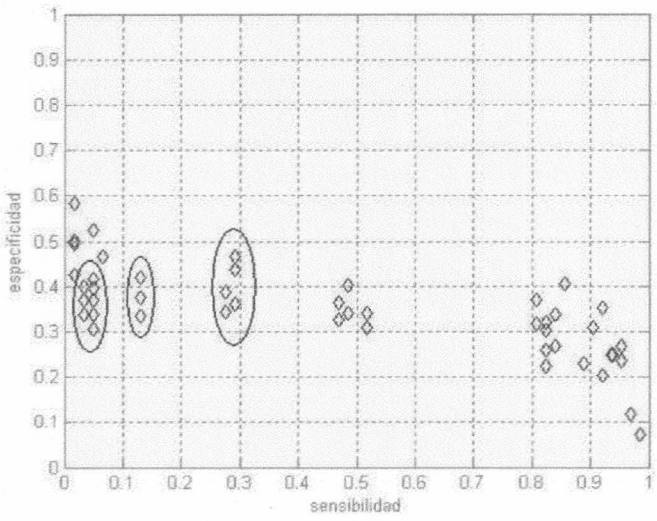


Figura 4.6: Gráfico de 2 dimensiones de los objetivos *sensibilidad* y *especificidad* correspondientes a los datos de la Tabla 4.2. Se puede observar que para algunos valores de sensibilidad existen varios clusters distintos.

encuentra en el cluster 19 pero no en el 22. En la Figura 4.7 se presenta el subgrafo de *GO* que contiene todos estos términos en forma conjunta. Por otro lado, los genes que tienen estos dos clusters son exactamente los mismos, es decir, se solapan totalmente.

Lo que se deduce de estos hechos es que el cluster 22 no aporta información al resultado final, y por lo tanto debería haber sido subsumido por el cluster 19.

Al comparar los objetivos se puede apreciar que en *complejidad* el cluster 19 tiene un valor mayor, lo cual coincide con lo esperado, pero en *especificidad* el valor es menor al del cluster 22. Este problema se aprecia claramente en la Figura 4.7 y surge de que el término que tiene el cluster 19 hace que la *especificidad* total disminuya, debido a que se la calcula como un promedio y el término `GO:0003735` tiene un nivel más bajo con respecto a los otros términos GO.

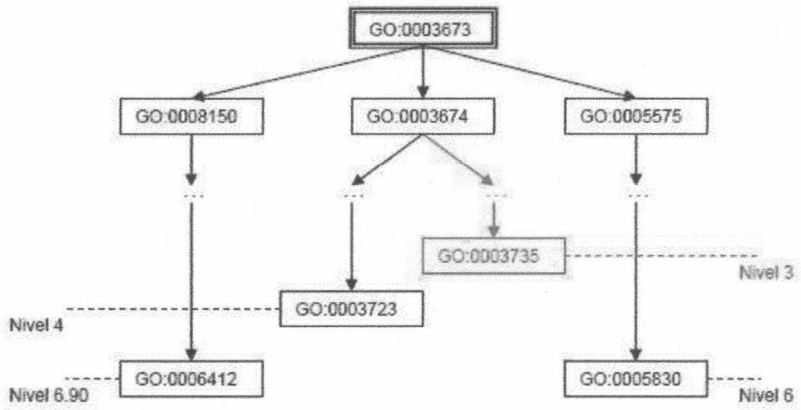


Figura 4.7: Subgrafo de *GO* con los términos GO de los clusters 19 y 22 de la Tabla 4.2. El término GO:0003735 se encuentra en rojo porque es el único que diferencia a estos clusters. Los niveles están calculados como el promedio entre todos los niveles en el grafo *GO* de cada término.

4.2.3. Redefinición del objetivo *especificidad*

El objetivo *especificidad* mide cual es la profundidad de una solución dentro del grafo de *GO*. Una manera de calcular este valor es hacer un promedio entre las alturas de cada uno de los términos GO del cluster, pero puede ocurrir que agregar términos GO a una solución disminuya el valor de *especificidad* (esto ocurre cuando se agregan términos que tienen un nivel bajo comparado a los otros y entonces el promedio disminuye) como se ha visto en la sección anterior. Por otro lado, un término *GO* puede presentarse en distintos niveles de la jerarquía. Hasta ahora este problema se ha resuelto representando el nivel de un término *GO* como el promedio de todos los niveles en los cuales aparece.

En esta última función objetivo se mantiene la definición anterior de *sensibilidad* y *complejidad*, pero se redefine el objetivo *especificidad* de la siguiente manera:

$$especificidad = \text{máx}\left(\max_{e\in BP}\left(\frac{\omega(e)}{h_{BP_{GO}}}\right), \max_{e\in MF}\left(\frac{\omega(e)}{h_{MF_{GO}}}\right), \max_{e\in CC}\left(\frac{\omega(e)}{h_{CC_{GO}}}\right)\right)$$

donde $\omega(e)$ es el máximo nivel en el que se encuentra el término e en la jerarquía de *GO*. Al igual que en la fórmula anterior de *especificidad*, el nivel de cada término es dividido por la profundidad total de la ontología a la cual pertenece ($h_{BP_{GO}}$ para BP, $h_{MF_{GO}}$ para MF y $h_{CC_{GO}}$ para CC).

Aplicación y análisis de resultados

Como resultado de la redefinición del objetivo *especificidad* se han obtenido 23 clusters (ver Tabla 4.3). En la Figura 4.8 se presenta el Pareto en el espacio de objetivos donde se puede apreciar la reducción de la cantidad de clusters con respecto al resultado con los objetivos anteriores. A su vez, se puede observar que los clusters no están alineados en el objetivo *sensibilidad* y no tienen genes repetidos. En la Figura 4.9 se presenta el Pareto en el espacio de variables y permite visualizar que el número de clusters con respecto a los objetivos anteriores ha disminuido, debido a la eliminación de clusters que no aportaban información. Este hecho hace que los nuevos resultados sean de una

calidad superior a los anteriores, asimismo, se puede observar que la distribución de las esferas en el espacio resulta similar.

Esta definición de función objetivo representa un balance entre las tres propiedades deseables del clustering conceptual. La calidad de las descripciones no ha disminuido significativamente con respecto a la función objetivo anterior y se ha logrado disminuir el solapamiento eliminando los clusters repetidos que no aportaban información. El número de clusters obtenidos es un punto medio entre las dos funciones anteriores. Esto es así porque se busca balancear las propiedades ya que, al ser conflictivas, no es posible obtener la mejor calidad en todas ellas.

Cluster	*BP*	*MF*	*CC*	*sensibilidad*	*complejidad*	*especificidad*
1			GO:0005737	0.95	0.10	0.27
2	GO:0006412		GO:0005737 GO:0043234	0.84	0.44	0.41
3	GO:0006378	GO:0005515 GO:0008143	GO:0005737	0.02	0.60	0.59
4	GO:0006412	GO:0003735	GO:0005840	0.81	0.63	0.41
5	GO:0044267			0.92	0.25	0.35
6	GO:0006412			0.85	0.33	0.41
7	GO:0006412	GO:0003674	GO:0005737 GO:0043234 GO:0043229	0.82	0.50	0.41
8	GO:0006412	GO:0003735	GO:0005830	0.48	0.67	0.47
9	GO:0006412	GO:0003723 GO:0003735	GO:0005843	0.13	0.81	0.53
10	GO:0044237 GO:0044238	GO:0003674	GO:0005737	0.92	0.27	0.27
11	GO:0009987			0.97	0.06	0.12
12		GO:0003674		0.99	0.04	0.07
13	GO:0044267		GO:0005737	0.90	0.33	0.35
14	GO:0006412	GO:0003723 GO:0003735	GO:0005730 GO:0005842	0.05	0.85	0.53
15	GO:0006412	GO:0003723 GO:0003735	GO:0005842	0.27	0.77	0.53
16	GO:0006412 GO:0007516	GO:0003723 GO:0003735	GO:0005843	0.03	0.98	0.53
17	GO:0006412	GO:0003735	GO:0005842	0.29	0.71	0.53
18	GO:0044237 GO:0044238		GO:0005737	0.94	0.25	0.27
19	GO:0006412	GO:0003723 GO:0003735	GO:0005840	0.57	0.69	0.41
20	GO:0043123 GO:0006412	GO:0003723 GO:0003735 GO:0004871	GO:0015934	0.05	1.00	0.53
21	GO:0044267	GO:0003674	GO:0005737	0.89	0.35	0.35
22	GO:0006412	GO:0003723 GO:0003735	GO:0005830	0.47	0.73	0.47
23	GO:0044237 GO:0044238			0.95	0.17	0.24

Tabla 4.3: Términos *GO* de los clusters obtenidos con la redefinición del objetivo *especificidad*, medido este último como la longitud de la rama más larga del subgrafo GO de cada cluster.

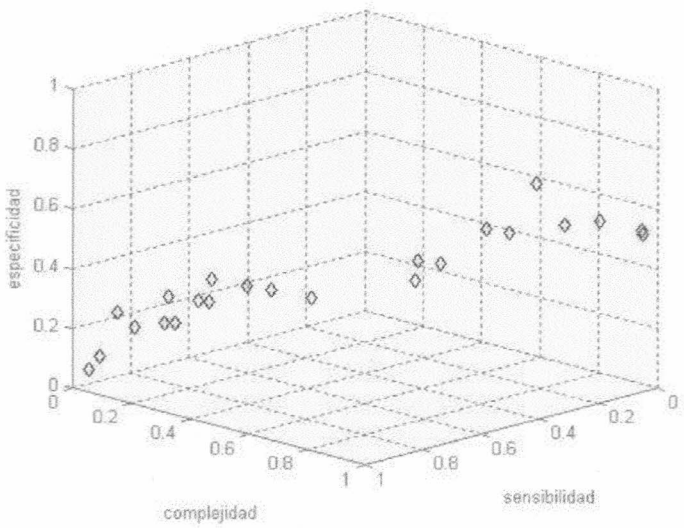


Figura 4.8: Gráfico del conjunto Pareto para los objetivos *sensibilidad*, *complejidad* y *especificidad*, medido este último como la longitud de la rama más larga del subgrafo GO de cada cluster. Este gráfico se corresponde con los valores de la Tabla 4.3.

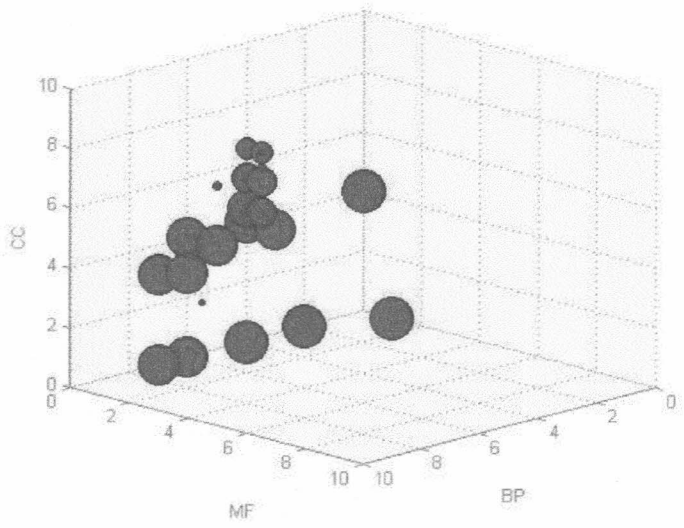


Figura 4.9: Gráfico del conjunto Pareto en el espacio de variables para los objetivos *sensibilidad*, *complejidad* y *especificidad*, medido este último como la longitud de la rama más larga del subgrafo GO de cada cluster. Cada cluster está representado con una esfera cuyo diámetro es proporcional a la cantidad de genes del cluster y sus coordenadas indican el promedio de los niveles de los términos GO del cluster en cada una de las tres ontologías.

4.2.4. Comparación de los resultados

Un conjunto Pareto debería estar compuesto por un número grande de soluciones no dominadas, presentar una buena distribución de las soluciones que lo componen y cubrir la mayor amplitud de valores posibles. En la literatura se pueden encontrar varias métricas cuantitativas para medir la calidad de los Paretos [ZDT00, ZT99]. Si bien estas métricas están pensadas para medir la calidad de los Paretos generados por metaheurísticas, en esta sección se utilizarán como medidas de comparación de los Paretos óptimos encontrados por las distintas funciones objetivo estudiadas anteriormente. Se utilizarán dos de estas métricas en esta sección, la métrica $\mathcal{M}_2^*$ y $\mathcal{M}_3^*$ definidas en [ZT99].

Sea Y' el conjunto de los vectores objetivo que corresponden a un Pareto y $\sigma^* > 0$, $\|\cdot\|$ una métrica de distancia, se define:

- La función $\mathcal{M}_2^*$ tiene en cuenta la distribución de las soluciones del conjunto Pareto con respecto al número de soluciones no dominadas que lo componen ($|Y'|$):

$$\mathcal{M}_2^*(Y') = \frac{1}{|Y'-1|} \sum_{p' \in Y'} |\{q' \in Y'; \|p'-q'\| > \sigma\}| \tag{4.1}$$

  Nótese como, para cada solución p' del conjunto Y', se contabiliza cuántas de las soluciones restantes están a una distancia mayor de σ de ella. Finalmente, se calcula el valor medio de la suma de la cuenta correspondiente a cada solución. De este modo, el valor de $\mathcal{M}_2^*$ está definido en $[0, |Y'-1|]$ y el Pareto generado será tanto mejor cuanto mayor sea dicho valor para un parámetro de vecindad adecuado. Por ejemplo, el valor $\mathcal{M}_2^* = |Y'-1|$ indica que, para cada solución del Pareto, no existe ninguna otra solución a una distancia menor de σ de ella.

- Por otro lado, la distribución de las soluciones a lo largo del Pareto puede estar aglomerada en una sola zona del espacio de búsqueda, como puede verse en la Figura 4.10, lo que no es deseable. Por ello, se calcula la función $\mathcal{M}_3^*$ que considera la extensión del frente descrito por Y':

$$\mathcal{M}_3^*(Y') = \sqrt{\sum_{i=1}^{M} max\{\|p_i' - q_i'\|; p', q' \in Y'\}} \tag{4.2}$$

  Así, $\mathcal{M}_3^*$ mide la distancia máxima en cada dimensión para determinar el área que ocupa el Pareto.

Se han aplicado las métricas anteriormente descritas a los Paretos de la Figura 4.11, utilizando distintos valores de σ para la métrica $\mathcal{M}_2^*$. Como se ve en la Tabla 4.4, para distintos valores de σ se mantiene la tendencia de que la segunda función es mayor a la tercera y esta, a su vez, es mayor a la primer función objetivo. Esto indica que la segunda función objetivo es la que tiene la mejor distribución de las soluciones, seguida por la tercer función, y bastante más abajo, la primera (este hecho también se puede observar visualmente en los gráficos (a), (b) y (c) de la Figura 4.11). A pesar de que la segunda función objetivo es la mejor con respecto a esta métrica, recordemos que esta función tiene la desventaja de generar clusters repetidos.

Los resultados con la métrica $\mathcal{M}_3^*$ se presentan en la Tabla 4.5. Aquí se ve que la tercer función objetivo es la que mejor se comporta con respecto a la aglomeración de soluciones en zonas del Pareto. Nótese que prácticamente no hay diferencia en esta métrica entre la segunda y tercer función objetivo.

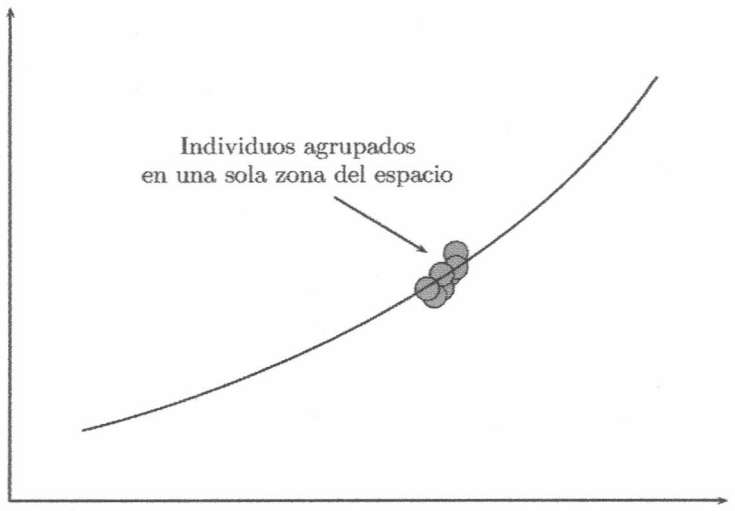


Figura 4.10: Soluciones agrupadas en una zona del Pareto óptimo.

σ	*primer función*	*segunda función*	*tercer función*
0.1	6.86	44.17	21.45
0.2	6	39.35	18.82
0.3	4.86	34.70	16.09
0.4	4.29	29.78	13.91
0.5	4.29	24.96	12.27
1	1.14	5.39	3.91

Tabla 4.4: Valores de la métrica $\mathcal{M}_2^*$ en distintos valores de σ para los Paretos de las tres funciones objetivo estudiadas.

función objetivo	$\mathcal{M}_3^*$
primer función	1.22
segunda función	1.561
tercera función	1.562

Tabla 4.5: Valores de la métrica $\mathcal{M}_3^*$ para los Paretos de las tres funciones objetivo estudiadas.

Otro tipo de comparación puede hacerse desde el punto de vista de las propiedades deseables del clustering que generan los Paretos. Como se ha presentado en la sección 4.2, estas propiedades son: descripciones de los clusters con numerosas características para obtener mejor poder de inferencia, pequeño numero de clusters para obtener un cubrimiento extenso, y mínimo solapamiento entre clusters. Se ha visto que estas propiedades son conflictivas puesto que al aumentar alguna de ellas disminuyen las otras. Los clusters que generan los Paretos de Las Figuras 4.11 (a) y (a') presentan un pequeño número de clusters, lo cual es positivo. Sin embargo, las descripciones de estos clusters son muy generales y por lo tanto se pierde poder de inferencia. La segunda función objetivo aumenta la calidad de las descripciones de los clusters (Figuras 4.11 (b) y (b')), aumentando significativamente el número de clusters. Sin embargo, se puede apreciar que se forman Paretos locales y esto se debe a que ha aumentado

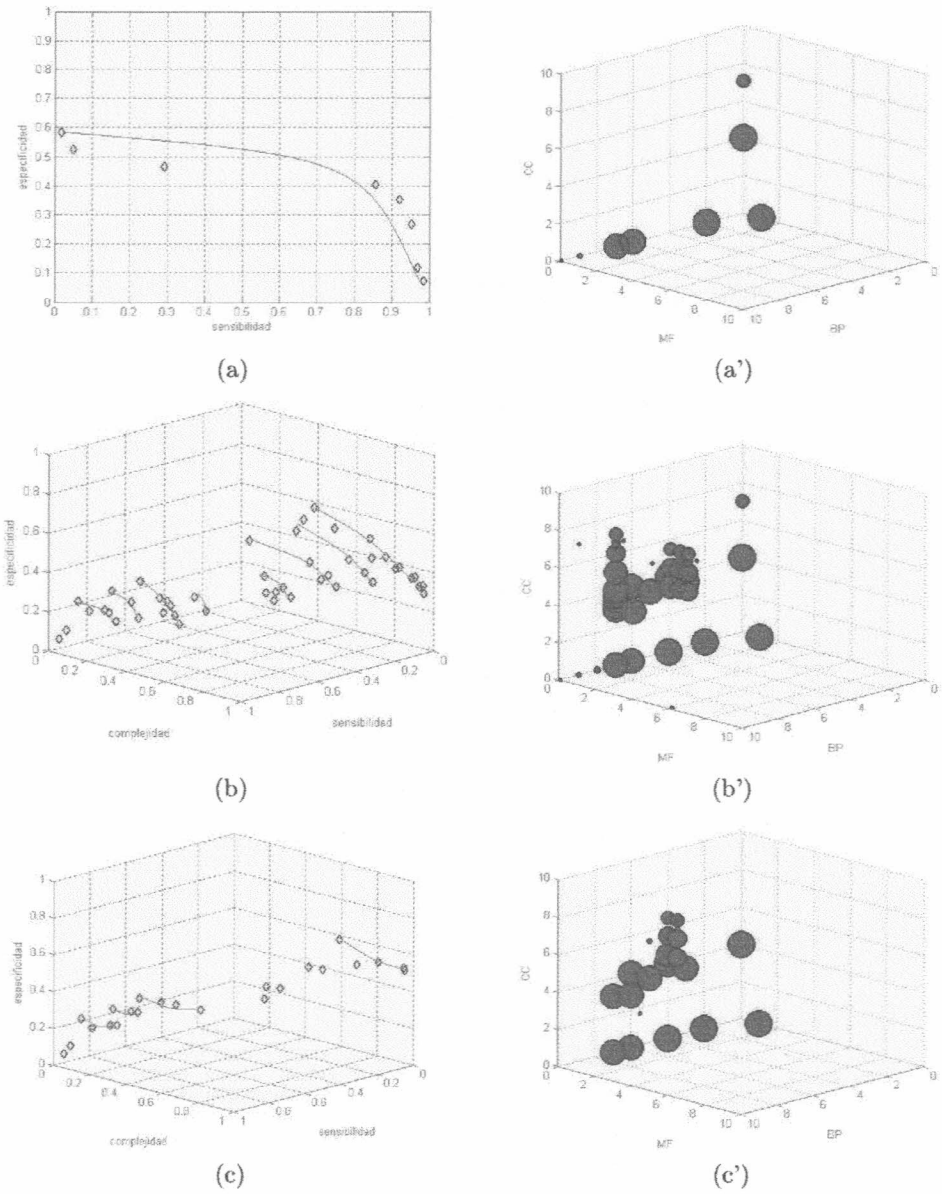


Figura 4.11: Gráficos de los Paretos con las distintas funciones objetivo estudiadas. (a) y (a') corresponden al Pareto en el espacio de objetivos y en el espacio de variables respectivamente para los objetivos *sensibilidad* y *especificidad*. (b) y (b') corresponden a la función objetivo que incorpora el objetivo *complejidad*. Finalmente, (c) y (c') corresponden a la función objetivo que redefine el objetivo *especificidad* a fin de obtener soluciones sin clusters repetidos.

significativamente el solapamiento. La última función objetivo (Figuras 4.11 (c) y (c')) es el mejor balance que se pudo lograr entre las tres propiedades conflictivas. El poder descriptivo no ha disminuido significativamente con respecto a la segunda función objetivo (Figuras (b') y (c')), el número de clusters ha disminuido y esta en un valor medio entre las otras dos funciones, y el solapamiento ha disminuido con respecto a la segunda función (Figuras (b) y (c)). Esto último se ve reflejado en el hecho que con esta última función objetivo no hay clusters repetidos (solapamiento total) como en el caso de la segunda función.

4.3. Observaciones finales

La base de datos del proyecto *Gene Ontology* presenta información útil para explicar perfiles de expresión genética provenientes de experimentos de Microarray. En este capítulo se ha presentado un nuevo método para el análisis de este tipo de información basado en clustering conceptual y optimización multiobjetivo. La componente más importante en optimización multiobjetivo es la *función objetivo*, puesto que la calidad de los clusters generados depende fuertemente de esta función. En este capítulo se ha analizado el comportamiento de distintas funciones objetivo aplicadas a un conjunto de 62 genes tomado del conjunto original de 1776.

Existen características que permiten evaluar cualitativamente un clustering conceptual. Una propiedad deseable es que el clustering tenga la mayor cobertura con el mínimo número posible de clusters, por otro lado, es deseable obtener descripciones con varias características para cada cluster aumentando de esta manera el poder de inferencia. La tercer propiedad que se busca es obtener el mínimo solapamiento entre los clusters. Estas tres propiedades son conflictivas puesto que el aumento en una de ellas causa una disminución en las otras. Luego del estudio de funciones objetivo, GO-GPS logra encontrar un balance entre estas tres propiedades deseables como se ha mostrado en los resultados presentados.

La implementación actual del algoritmo utiliza programación dinámica. Si bien los resultados obtenidos en cuanto a tiempo de ejecución son aceptables, debido a la cantidad de resultados intermedios que deben ser almacenados en memoria, la ejecución con los 1776 genes no ha podido concretarse.

Capítulo 5

Método *GO-GPS-GA*

Las heurísticas son una alternativa a los algoritmos exhaustivos cuando estos demoran mucho tiempo en retornar la solución exacta de algún problema (e.g. *Tabu Search, GRASP*) [MF00, Osm95]. GO-GPS es un método exhaustivo que realiza clustering conceptual en conjuntos de genes utilizando información de *Gene Ontology*. GO-GPS analiza exhaustivamente todas las posibles soluciones para encontrar el Pareto óptimo, por lo tanto, a medida que crece el tamaño de entrada se hace evidente la necesidad de utilizar heurísticas.

En el presente capítulo se presenta GO-GPS-GA, un método basado en la metodología *CC-EMO (Clustering conceptual basado en evolución multiobjetivo)* [RZCRE+] que realiza clustering conceptual en conjuntos de genes utilizando los mismos objetivos definidos en GO-GPS. En la sección 5.1 se explican los conceptos básicos de algoritmos evolutivos, principalmente algoritmos genéticos. En las secciones 5.2 y 5.3 se presentan los algoritmos evolutivos multiobjetivo NSGA-II [DAPM00] y CC-EMO respectivamente. El algoritmo evolutivo utilizado en CC-EMO constituye la base de GO-GPS-GA.

GO-GPS-GA es aplicado al conjunto de 62 genes para comparar sus resultados con los de GO-GPS (ver sección 5.4). Luego, en la sección 5.5, se analizan los resultados obtenidos de la aplicación de GO-GPS-GA al conjunto completo de 1776 genes. Finalmente, en la sección 5.6 se compara GO-GPS-GA con APRIORI, un método computacional de machine learning para búsqueda de patrones [AIS93], y con FatiGO, un método que utiliza específicamente información de *Gene Ontology* [FASD04].

5.1. Algoritmos evolutivos

La *Computación Evolutiva* (CE) se basa en el empleo de modelos de procesos evolutivos para el diseño e implementación de sistemas de resolución de problemas. Los distintos modelos computacionales que se han propuesto dentro de esta filosofía suelen recibir el nombre genérico de *Algoritmos Evolutivos* (AEs) [BFM97]. Existen cuatro tipos de AEs bien definidos que han servido como base a la mayoría del trabajo desarrollado en el área: los *Algoritmos Genéticos* (AGs), las *Estrategias de Evolución* (EEs), la *Programación Evolutiva* (PE) y la *Programación Genética* (PG).

Un AE se basa en mantener una población de posibles soluciones del problema a resolver, llevar a cabo una serie de alteraciones sobre las mismas y efectuar una selección para determinar cuáles permanecen en generaciones futuras y cuáles son eliminadas. Aunque todos los modelos existentes siguen esta estructura general, existen algunas diferencias en cuanto al modo de ponerla en práctica. Los AGs se basan en técnicas que tratan de modelar los operadores genéticos existentes en la naturaleza, como el

cruce y la mutación, los cuales son aplicados a los individuos que codifican las posibles soluciones. En cambio, las EEs y la PE aplican transformaciones basadas en mutaciones efectuadas sobre los padres para obtener los hijos, lo que permite mantener una línea general de comportamiento del individuo en su descendencia. Finalmente, la PG representa las soluciones al problema en forma de programas, habitualmente codificados en una estructura de árbol, y adapta dichas estructuras empleando operadores muy específicos.

Cada individuo de la población recibe un valor de una medida de adaptación que representa su grado de adecuación al entorno. La selección hace uso de estos valores y se centra en los individuos que presentan mayor valor en la media. Los operadores de recombinación y/o mutación alteran la composición de dichos individuos, guiando heurísticamente la búsqueda a través del espacio. Aunque simples desde un punto de vista biológico, este tipo de algoritmos son suficientemente complejos para proporcionar mecanismos de búsqueda adaptativos muy robustos. Los mismos procedimientos pueden ser aplicados a problemas de distintos tipos sin necesidad de hacer muchos cambios [Gol89].

En la metodología *CC-EMO* [RZCRE+] general el algoritmo evolutivo que se utiliza es del tipo PG. En nuestra adaptación al dominio biológico de Gene Ontology utilizaremos un AG. En la siguiente sección se explicarán con más detalle los AGs.

5.1.1. Algoritmos genéticos

Los AGs [Gol89] son una técnica *metaheurística* para la solución de problemas de optimización. Se basan en una analogía de la teoría biológica de la evolución de las especies y toman esta idea para buscar una o más soluciones óptimas entre un conjunto de posibles soluciones.

> La *Teoría de la Evolución* explica el origen y la transformación de los seres vivos como el producto de la acción de dos principios fundamentales: la selección natural y el azar. La selección natural regula la variabilidad de la recombinación y mutación aleatorias de los genes: toda la variedad que observamos en la naturaleza se basa en la capacidad de los seres vivos de producir copias de sí mismos, en que el proceso de reproducción actualiza muchas variantes, y en que, en la interacción con el ambiente, algunas de ellas son seleccionadas para sobrevivir y producir las copias subsiguientes [Dar59].

Los AGs generan descendientes realizando repetidas mutaciones y cruces de las mejores soluciones de un conjunto. A cada paso, una colección de soluciones es actualizada reemplazando una fracción de la población por la descendencia de las soluciones más adaptadas al medio. Entonces se puede ver que estas soluciones serán las que tendrán mayor probabilidad de pasar a la próxima generación.

Para explicar como funcionan los AGs es necesario primero comprender la terminología utilizada. En las Figuras 5.1, 5.2 y 5.3 se pueden observar los conceptos en forma gráfica tomando como ejemplo el problema de encontrar el número máximo entre 1 y 15 usando una representación binaria.

Población. Se denomina población al conjunto de individuos que representan las soluciones a optimizar.
Cromosoma - Gen. Se denomina cromosoma a cada individuo de la población. A su vez, se conoce con el nombre de gen a cada parte del cromosoma que tiene significado por sí misma.

Genotipo - Fenotipo. En la naturaleza, un genotipo es la información genética que, al desarrollarse, crea un fenotipo o ser vivo. En el ámbito de los AGs, se denomina genotipo al conjunto de parámetros representado por un cromosoma particular que contiene toda la información necesaria para construir una solución (organismo), a la cual se la denomina fenotipo.
Generación. Se denomina generación a cada iteración del algoritmo.

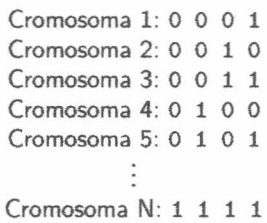


Figura 5.1: Población.

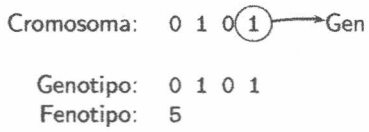


Figura 5.2: Genotipo vs. Fenotipo.

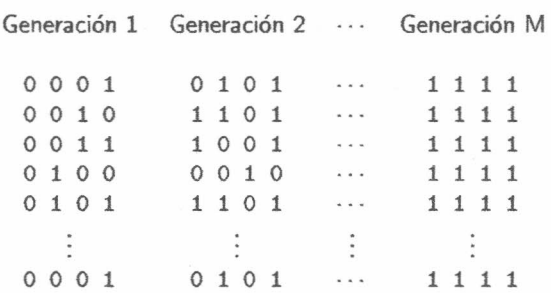


Figura 5.3: Generaciones.

5.1.2. Algoritmo genético básico

Los AGs exploran un espacio de soluciones candidatas en busca de la mejor solución. Cuando decimos la mejor solución, nos referimos a aquella que optimice una cierta función relevante para el problema tratado, a la que se conoce con el nombre de función de *fitness* o función de aptitud. A pesar que existen clases muy diversas de AGs, todas mantienen una estructura en común:

> En cada iteración, todos los miembros de la población se evalúan de acuerdo a la función de fitness. Se genera una nueva población seleccionando de forma probabilística los mejores individuos de la población actual. Algunos de estos individuos pasan a la próxima generación automáticamente,

mientras que otros se utilizan para procrear nuevas soluciones por medio de cruces de dos individuos, o bien se mutan antes de pasar a la próxima generación.

El algoritmo itera hasta cumplir con un criterio de parada. Éste puede ser la cantidad de generaciones o evaluaciones de la función de fitness realizadas o la obtención de una solución que esté dentro de un cierto umbral de aceptación. El pseudocódigo del algoritmo básico para un algoritmo genético simple se muestra en la Figura 5.1 [Gol89].

Algoritmo 5.1 Pseudocódigo para un AG básico.

entrada: f función de fitness,
f_{umbral} criterio de terminación,
p tamaño de una población,
p_c probabilidad de cruce,
p_m probabilidad de mutación
salida: individuo con mayor f

$P \leftarrow$ Generar p individuos al azar
para todos $h \in P$ **hacer**
  Calcular $f(h)$
fin para
mientras $(|\text{máx}_h\, f(h)] < f_{umbral})$ **hacer**
  $P_S \leftarrow \emptyset$
  Seleccionar probabilísticamente $(1 - r)p$ miembros de P para agregar a P_S
  Seleccionar con probabilidad p_c pares de individuos y producir descendencia aplicando el operador de cruce y agregarlos a P_S
  Seleccionar con probabilidad p_m individuos de P_S. Para cada uno utilizar el operador de mutación y agregar el individuo modificado a P_S
  $P \leftarrow P_S$
  para todos $h \in P$ **hacer**
    Calcular $f(h)$
  fin para
fin mientras
Retornar individuo con mayor f

Representación de los cromosomas

La representación de los cromosomas depende del problema a tratar e influye directamente sobre los resultados del AG. Existen distintos esquemas generales de codificación entre los que destacan los siguientes:

1. La *codificación binaria*: Es la más antigua de todas las existentes. La representación de los cromosomas está definida como cadenas de bits de modo que, dependiendo del problema, cada gen puede estar formado por una subcadena de uno o varios bits.

2. La *codificación real*: La codificación binaria presenta algunos inconvenientes cuando se trabaja con problemas que incluyen variables definidas sobre dominios continuos: excesiva longitud de los cromosomas, falta de precisión, etc. Una posible manera de evitar estos inconvenientes es considerar un esquema de representación real. Aquí, cada variable del problema se asocia a un único gen que toma

un valor real dentro del intervalo especificado, por lo que no existen diferencias entre el genotipo y el fenotipo.

3. La *codificación basada en orden*: Este esquema está diseñado específicamente para problemas de optimización combinatoria en los que las soluciones son permutaciones de un conjunto de elementos determinando.

Estos ejemplos de posibles representaciones nos dan una idea genérica del tipo de esquemas que se utilizan más comúnmente. Esto no implica que sean los únicos, o que no se puedan crear esquemas propios, que no tengan relación alguna con los comentados, si es que se adaptan mejor a un problema en particular.

Mecanismo de Selección

El mecanismo de selección es el encargado de seleccionar la población intermedia de individuos la cual, una vez aplicados los operadores de cruce y mutación, formará la nueva población del AG en la siguiente generación. De este modo, el mecanismo de selección se encarga de obtener una población intermedia formada por copias de los cromosomas de la población original como se muestra en la Figura 5.4.

Dos son los métodos de selección más comunes:

- *La Ruleta.* Consiste en crear una ruleta en la que cada cromosoma tiene asignada una fracción proporcional a su aptitud. Esta ruleta se gira varias veces para determinar qué individuos se seleccionarán. Debido a que a los individuos más aptos se les asignó un área mayor de la ruleta, se espera que sean seleccionados más veces que los menos aptos.
- *El torneo.* La selección por torneo se ha popularizado debido a que sólo utiliza información local para elegir a los mejores candidatos, por lo tanto se reduce la complejidad de cálculo en poblaciones de gran tamaño. El torneo consiste en seleccionar un conjunto de individuos de la población al azar, dependiendo del tamaño del torneo, para luego comparar entre sí dichos individuos y elegir aquél con mejor valor de aptitud. Este proceso se realiza tantas veces como elementos existan en la población. En el caso de un torneo binario, la competencia se realiza entre dos individuos, seleccionando aquél con mejor función de aptitud.

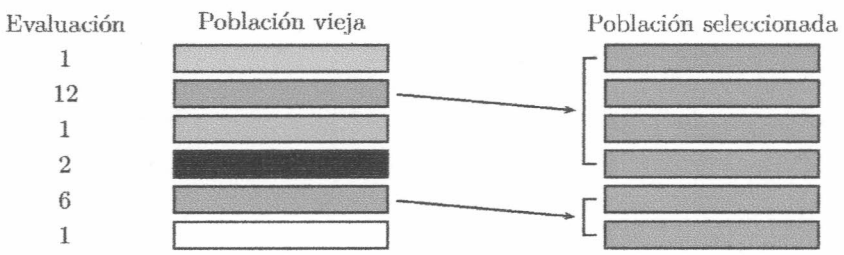


Figura 5.4: Ejemplo de aplicación del mecanismo de selección.

Operadores genéticos

El operador de cruce constituye un mecanismo para compartir información entre cromosomas. Combina las características de dos cromosomas padre para obtener dos descendientes, con la posibilidad de que los cromosomas hijo, obtenidos mediante la recombinación de sus padres, estén mejor adaptados que éstos. No suele aplicase a todas

las parejas de cromosomas de la población intermedia, sino que se lleva a cabo una selección aleatoria en función de una determinada probabilidad de aplicación, la *probabilidad de cruce*, p_c.

El operador de cruce cumple un papel fundamental en el AG. Su tarea consiste en **explotar** el espacio de búsqueda combinando las soluciones obtenidas hasta el momento mediante la recombinación de las buenas características que presenten.

Un ejemplo del cruce más conocido, llamado *cruce simple en un punto*, se muestra de forma gráfica en la Figura 5.5. El cruce simple se basa en seleccionar aleatoriamente un punto de cruce e intercambiar el código genético de los cromosomas padre a partir de dicho punto para formar los dos hijos.

Figura 5.5: Ejemplo de aplicación del operador de cruce simple de un punto.

La mutación, en cambio, pretende **explorar** el espacio de búsqueda alterando una de las componentes del código genético de un individuo. La mutación altera localmente el genotipo esperando obtener un individuo mejor. Debido al efecto de la selección, se sabe que sólo serán elegidas las buenas soluciones para pasar a la próxima generación, mientras que las malas soluciones serán eliminadas.

5.1.3. Algoritmos genéticos para funciones multimodales: *Nichos*

Como el nombre sugiere, las funciones multimodales tienen múltiples soluciones óptimas, de las cuales varias pueden ser óptimos locales. Como se ha comentado, los AGs son conocidos por su capacidad para llevar a cabo procesos de búsqueda en espacios complejos. A pesar de ello, un AG clásico puede no trabajar de modo adecuado cuando el espacio de búsqueda es multimodal y presenta varios óptimos locales. En estos casos, los AGs simples se caracterizan por converger hacia la zona del espacio donde se encuentran los mejores óptimos locales, abandonando la búsqueda en las zonas restantes (proceso conocido con el nombre de "deriva genética") [Gol89].

Se han propuesto varios métodos para el tratamiento de funciones multimodales en AGs. Sin embargo, la forma más habitual de diseñar AGs multimodales se basa en los conceptos de *nicho* y *especie*. Ambos conceptos fueron introducidos con el objeto de mantener múltiples óptimos. Una de las formas más habituales para provocar la formación de especies y la creación de nichos se basa en el esquema de *sharing* o proporción [GR87]. El proceso de sharing permite mantener en la población de cada generación una cantidad proporcional de individuos en distintas zonas del espacio de búsqueda, manteniendo de esta manera una buena diversidad de soluciones. Cada zona del espacio de búsqueda estará representado en el algoritmo como un *nicho*. En cada una se encontrará un conjunto de soluciones cercanas de acuerdo a una cierta distancia.

Como ocurre en la naturaleza, los individuos de cada nicho comparten la recompensa asociada a dicho nicho entre ellos. Para esta tarea se define una función conocida como *fitness sharing* (función de proporción). Este método permite distribuir la población sobre diferentes picos (máximos o mínimos locales dependiendo del tipo de función de aptitud utilizada) del espacio de búsqueda, donde la cantidad de individuos que recae en cada pico es proporcional a la calidad del pico si se trata de optimizar una función.

Entonces, la selección de individuos debe permitir que elementos pertenecientes a distintos nichos sean mantenidos en las futuras generaciones. Para poder hacerlo, se genera un torneo. Dos individuos compiten entre sí para determinar cual de los dos pasará a la próxima generación. Para decidir cual de los dos competidores será el ganador, se calcula la cantidad de elementos que existen en los nichos a los cuales pertenecen. La cantidad de elementos en cada nicho se define como:

$$nicho(x_i) = \sum_{x_j \in P} sh(d(x_i, x_j)) \tag{5.1}$$

donde d es la medida de distancia entre dos elementos, P es el conjunto de individuos de la población y sh es la función de sharing. Esta función se define por lo general como:

$$sh(v) = \begin{cases} 1 - v/\sigma_{share} & v \leq \sigma_{share} \\ 0 & v > \sigma_{share} \end{cases} \tag{5.2}$$

En este caso, σ_{share} es el radio del nicho que debe ser especificado por el usuario y determina la mínima separación deseada entre picos. La función de proporción (función de fitness modificada) sobre un individuo x_i se define entonces como $f(x_i)/nicho(x_i)$, donde f es el valor de su función de fitness. Este proceso permite evitar que toda la población converja a una única solución y, en lugar de ello, permite que los individuos de cada nicho converjan independientemente. Es deseable obtener nichos igualmente poblados en las distintas generaciones de forma tal que:

$$\frac{f(x_i)}{nicho(x_i)} = \frac{f(x_j)}{nicho(x_j)} \quad \forall x_i, x_j \quad individuos \tag{5.3}$$

5.1.4. Elitismo

El ciclo de nacimiento y muerte de los individuos está muy relacionado con el manejo de la población. El tiempo de vida de un individuo es típicamente de una generación, aunque en algunos AGs puede ser mayor. La estrategia de elitismo relaciona la vida de los individuos con su aptitud. Estas estrategias son técnicas utilizadas para mantener las buenas soluciones más de una generación. Una estrategia de elitismo habitualmente utilizada en AGs consiste en mantener una copia del mejor individuo encontrado hasta el momento en cada generación. Esto se realiza porque en el cruce los padres suelen ser reemplazados por sus hijos y, por ello, no hay seguridad de que los individuos con mayor aptitud sobrevivan a la próxima generación.

5.2. Algoritmos genéticos multiobjetivo

Una diferencia entre los métodos de búsqueda y optimización clásicos y los AGs es que, en estos últimos, se procesa una población de soluciones en cada iteración. Esta característica, por sí sola, le da a los AGs una ventaja para su uso en problemas de optimización con múltiples objetivos.

Los problemas de optimización multiobjetivo son aquellos donde hay por lo menos dos objetivos contrapuestos que deben satisfacerce al mismo tiempo (ver sección 3.3.2 para una explicación detallada).

En la siguiente sección se describirá el funcionamiento de un AGMO (*Algoritmo Genético MultiObjetivo*) muy utilizado, el NSGA-II [DAPM00].

5.2.1. El algoritmo NSGA-II

El algoritmo NSGA-II [DAPM00] es una versión mejorada del *nondominated sorting genetic algorithm (NSGA)*. La idea subyacente al algoritmo NSGA [SD94] es la utilización de un método de selección basado en ranking para enfatizar las buenas soluciones y un método de nichos para una buena diversidad de soluciones en las sub-poblaciones.

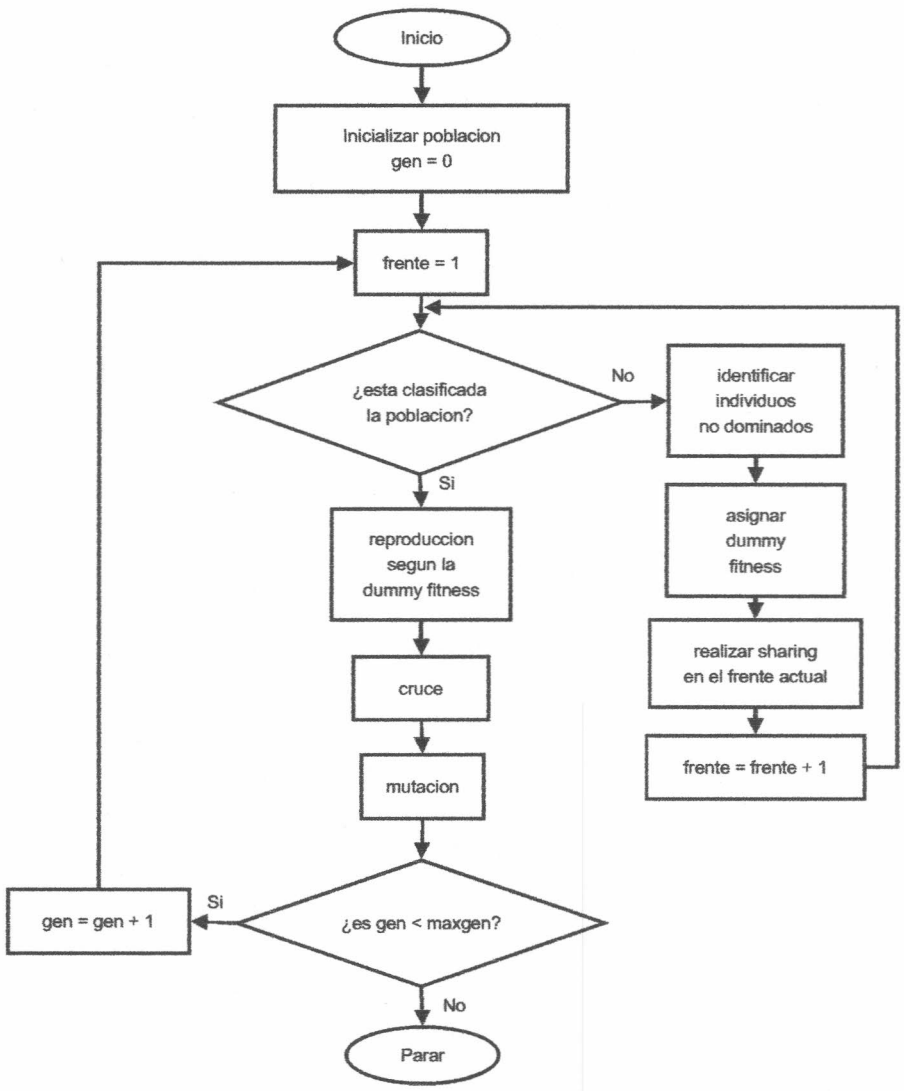


Figura 5.6: Diagrama de flujo del algoritmo NSGA.

El algoritmo NSGA trabaja como un AG clásico con algunos pasos adicionales para poder obtener un frente de Pareto diverso (ver Figura 5.6). Antes de la etapa de selección, se ordena la población en base a la no dominancia de cada individuo. Se asigna entonces un valor de aptitud dummy suficientemente alto a todas las soluciones no dominadas para darle igual potencial reproductivo a todas ellas. Con el fin de mantener la diversidad de la población, el valor de aptitud de estos individuos se calcula

proporcionalmente al número de individuos a los que domina. Este procedimiento se conoce como *sharing*. Luego del sharing, se ignora temporalmente a los individuos no dominados para así procesar al resto de la población de la misma manera. Los individuos resultantes conformarán el segundo frente de Pareto. A este nuevo conjunto de soluciones se les asigna un nuevo valor dummy de aptitud, el cual se mantiene estrictamente menor que el mínimo valor de aptitud del frente de Pareto previo. Este proceso continúa con el resto de la población hasta clasificar a todos los individuos en varios frentes de Pareto.

La población es entonces seleccionada de acuerdo a sus valores de fitness *dummy*, para luego aplicarles los operadores genéticos de la misma manera que en un AG clásico. Dado que los individuos que se encuentran en el primer frente tienen el valor de fitness máximo, serán siempre los que obtengan un mayor número de copias que el resto de la población, resultando en una rápida convergencia de la población hacia las regiones no dominadas mientras que el *sharing* ayuda a distribuirlas uniformemente en esta región.

El algoritmo NSGA tiene algunos problemas que la extensión realizada por el NSGA-II intenta solucionar:

- Alta complejidad computacional en la determinación del orden de las soluciones no dominadas.
- Falta de elitismo.
- Necesidad de especificar los parámetros del sharing.

El algoritmo NSGA-II incluye las modificaciones necesarias para superar estos problemas. Primero, se reduce la complejidad computacional reescribiendo el código original de la ordenación de una manera más eficiente, guardando los datos temporales en cada paso para su posterior reutilización. Segundo, se agrega un procedimiento de elitismo que compara la población actual con la población anterior de las mejores soluciones no dominadas en cada generación del algoritmo. Finalmente, para evitar la necesidad de parámetros en el proceso clásico de sharing, se utiliza un nuevo procedimiento de sharing.

Para conseguir una estimación de la densidad de soluciones que rodean a una solución particular de la población, se calcula la distancia promedio de dos soluciones x_b y x_c a cada lado de esta solución en cada uno de los objetivos. Estos dos puntos se seleccionan de acuerdo al siguiente procedimiento: el cómputo de la distancia de crowding requiere reordenar la población de acuerdo a cada uno de los objetivos en orden descendente. Luego, para cada objetivo, se asigna a aquellas soluciones en los bordes (soluciones con un valor máximo o mínimo) un valor de distancia infinito. Entonces, se asigna a todas las soluciones intermedias un valor de distancia igual a la diferencia absoluta normalizada de los valores en el objetivo en cuestión de las dos soluciones adyacentes (las soluciones x_b y x_c).

El valor de distancia global de crowding se calcula finalmente como la suma de los valores de distancia individual correspondientes a cada objetivo. Cada función objetivo es normalizada antes de calcular el valor de distancia. Una solución con un valor de distancia menor está, en algún sentido, más rodeada de otras soluciones.

Luego de calcular el valor de distancia a cada solución de la población, podemos comparar dos soluciones según su grado de proximidad con otras soluciones. Gracias a ello, se define un operador de comparación de crowding $\prec_c$ para guiar este proceso de selección:

$$x_i \prec_c x_j \quad \text{si} \quad (\mathrm{x}_{\mathrm{i}_{\mathrm{orden}}} < \mathrm{x}_{\mathrm{j}_{\mathrm{orden}}}) \vee ((\mathrm{x}_{\mathrm{i}_{\mathrm{orden}}} = \mathrm{x}_{\mathrm{j}_{\mathrm{orden}}}) \wedge (\mathrm{x}_{\mathrm{i}_{\mathrm{distancia}}} > \mathrm{x}_{\mathrm{j}_{\mathrm{distancia}}})) \quad (5.4)$$

donde $x_{i_{orden}}(x_{j_{orden}})$ es el *orden*−ésimo frente donde la solución $x_i(x_j)$ está ubicado, e $x_{i_{distancia}}(x_{j_{distancia}})$ es la distancia de crowding de $x_i(x_j)$. Esto significa que preferimos, entre dos soluciones con diferentes rangos de no dominancia, aquella con el rango más bajo. Si ambas soluciones pertenecen al mismo frente, entonces preferiremos aquella solución que esté ubicada en la región menos densa. El esquema general de funcionamiento del algoritmo NSGA-II se grafica en la Figura 5.7.

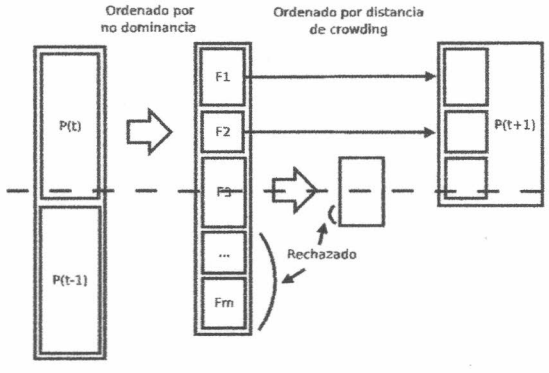


Figura 5.7: Esquema del algoritmo NSGA-II.

5.3. El método GO-GPS-GA

El método GO-GPS-GA (*GO-GPS Genetic Algorithm*) está basado en la metodología llamada *Clustering Conceptual basado en Evolución MultiObjetivo (CC-EMO)* [RZCRE+]. Esta metodología realiza clustering conceptual de las instancias de un repositorio de datos estructurado mediante el uso de técnicas de optimización multiobjetivo, identificando patrones comunes representados por subestructuras. Es importante destacar que CC-EMO es una metodología compuesta de varios pasos que van desde la adaptación de los datos a modelos estructurados hasta la compactación de resultados, permitiendo en base a estos hacer predicciones con nuevas instancias no incorporadas en los datos originales. El método GO-GPS-GA hace uso de Algoritmos Evolutivos [BFM97], que han demostrado obtener buenos resultados en este campo [Deb01, CVL02]. La implementación actual del algoritmo GO-GPS-GA está basada en CC-EMO, que a su vez está basado en el algoritmo NSGA-II [DAPM00] descrito en la Sección 5.2.1.

Mediante el uso de un enfoque multiobjetivo, el algoritmo obtiene el mejor conjunto de subestructuras que sean conjuntamente óptimas en los objetivos utilizados. Por lo tanto, dada una base de datos estructurada, el algoritmo GO-GPS-GA generará, en una sola ejecución, un conjunto Pareto de subestructuras de un repositorio dado. Este conjunto Pareto estará compuesto de varias subestructuras que representarán diferentes conceptos, cada uno de ellos cubriendo un subconjunto de elementos de la base de datos. Para ello, cada cluster tiene un concepto inherente que agrupa un conjunto de instancias, las cuales están determinadas por su semántica.

En la metodología CC-EMO original se utilizan dos objetivos para optimizar: *soporte* y *especificidad*. En GO-GPS-GA estos objetivos se cambian por los que han sido estudiados en el capítulo anterior.

Los componentes principales del método GO-GPS-GA se describen a continuación:

Representación de los cromosomas. En su versión general, el algoritmo CC-EMO utiliza una representación interna de cromosomas en forma de árbol, lo cual lo convierte en un algoritmo de Programación Genética (PG) [Koz92, BPKF98]. En su adaptación al dominio del problema biológico, sería viable mantener esta representación. Sin embargo, es mucho más eficiente codificar los cromosomas como un vector de códigos GO almacenando aquí únicamente los términos más específicos y manteneniendo la jerarquía almacenada en forma externa.

Para la representación de un cromosoma se utilizará un vector de nodos, los cuales podrán ser de tres tipos: nodos tipo 1 (correspondientes a los posibles nodos de la jerarquía de GO de procesos biológicos), nodos tipo 2 (correspondientes a función molecular) y nodos tipo 3 (correspondientes a componente celular). Cada uno de estos nodos tiene asociada una etiqueta, la cual restringirá la aplicación de los operadores genéticos. Al utilizar una representación lineal, en contraposición con la representación jerárquica de la versión original, trabajaremos con un AG en lugar de un algoritmo de PG.

Operadores genéticos. Sobre los cromosomas que componen la población se aplican los operadores de *cruce* y *mutación.*

El cromosoma *cruce* se obtiene al elegir una cantidad al azar de nodos tanto del primer cromosoma como del segundo, en cualquiera de las tres ontologías. Estos nodos son los que forman el cromosoma cruce, como muestra la Figura 5.8.

Los operadores de *mutación* utilizados son los siguientes:

- *Eliminación de un nodo:* Se selecciona aleatoriamente un nodo del vector de cualquiera de las tres ontologías y se elimina (ver Figura 5.9). Conceptualmente se está eliminando una hoja del subgrafo de GO que queda determinado por las hojas contenidas en el vector.
- *Modificación de un nodo:* Se selecciona aleatoriamente un nodo del vector y es reemplazado por otro perteneciente al conjunto posible de nodos de reemplazo. Este conjunto está formado por los padres y los hijos del nodo a cambiar (ver Figura 5.10).
- *Agregación de un nodo:* Se elige aleatoriamente un nodo del grafo de GO y se agrega al vector (ver Figura 5.11).

Optimización multiobjetivo. Los objetivos que se procuran maximizar son: *sensibilidad* que es la cantidad de instancias (genes) cubiertas por el modelo, *complejidad* que es la cantidad de nodos del subgrafo de GO que queda determinado por las hojas contenidas en el vector del cromosoma y *especificidad* que mide la profundidad del subgrafo de GO como el máximo nivel de cada hoja relativo a la profundidad de la ontología a la cual pertenece la hoja. Estos objetivos han surgido del estudio del capítulo anterior y se calculan de la siguiente manera:

$$sensitividad = \frac{\#(Genes)}{\#(GenesTotales)}$$

$$especificidad = \frac{\frac{\sum_{e \in BP} \ell(e)}{h_{BP_{GO}}} + \frac{\sum_{e \in MF} \ell(e)}{h_{MF_{GO}}} + \frac{\sum_{e \in CC} \ell(e)}{h_{CC_{GO}}}}{\#(BP) + \#(MF) + \#(CC)}$$

$$especificidad = \text{máx}\left(\text{máx}_{e \in BP}\left(\frac{\omega(e)}{h_{BP_{GO}}}\right), \text{máx}_{e \in MF}\left(\frac{\omega(e)}{h_{MF_{GO}}}\right), \text{máx}_{e \in CC}\left(\frac{\omega(e)}{h_{CC_{GO}}}\right)\right)$$

donde $\omega(e)$ es el máximo nivel en el que se encuentra el término e en la jerarquía de *GO*. Al igual que en la fórmula anterior de *especificidad*, el nivel de cada término es dividido por la profundidad total de la ontología a la cual pertenece ($h_{BP_{GO}}$ para BP, $h_{MF_{GO}}$ para MF y $h_{CC_{GO}}$ para CC).

En GO-GPS-GA se utiliza una técnica de *nichos* calculada en el espacio de instancias a fin de no caer en óptimos locales. Dos soluciones son comparadas por dominancia si y solo si tienen al menos un 50 % de instancias en común, esto es, si el valor del *Coeficiente de Jaccard* [Jac12] (ver ecuación 5.5, donde X e Y son los conjuntos de instancias cubiertas por cada solución) entre ellos es mayor que 0,5. En caso contrario, se dice que las soluciones son no dominadas.

$$jaccard(X, Y) = \frac{X \cap Y}{X \cup Y} \tag{5.5}$$

Con esta restricción, dos soluciones serán comparadas únicamente si la intersección de las instancias que cubren es mayor al 50 % de su soporte.

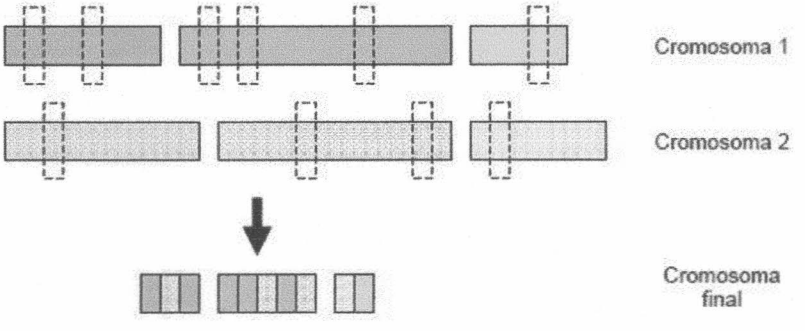


Figura 5.8: Operación de cruce utilizada en el algoritmo GO-GPS-GA. Los diferentes colores corresponden con las tres ontologías de GO. El cromosoma final se forma eligiendo nodos al azar de los dos cromosomas originales.

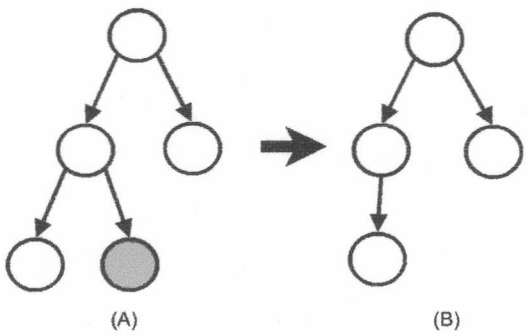


Figura 5.9: Mutación: borrado de una hoja. (A) es el árbol original con el nodo naranja elegido para eliminar, (B) es el árbol mutado.

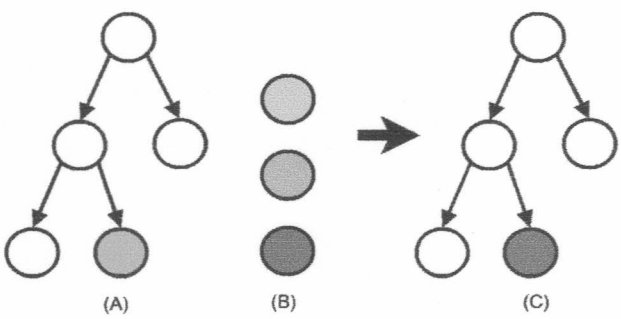


Figura 5.10: Mutación: modificación de un nodo. (A) es el árbol original con el nodo naranja elegido para modificar, (B) son las posibilidades entre los cuales se elige uno al azar, (C) es el árbol mutado.

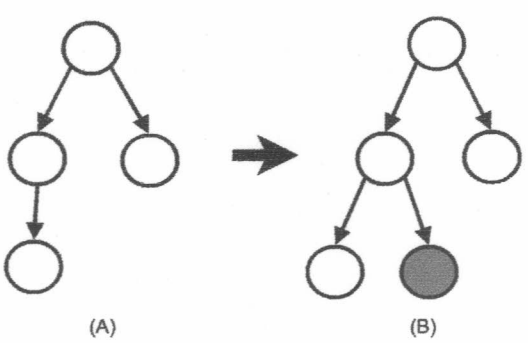


Figura 5.11: Mutación: agregación una hoja. (A) es el árbol original, (B) es el árbol mutado con el nodo celeste elegido para agregar.

5.4. Comparación de GO-GPS-GA y GO-GPS

En esta sección se muestran los resultados de aplicar el método GO-GPS-GA a los datos de los 62 genes del cluster 20 que fueron utilizados para el estudio de funciones objetivo del capítulo anterior. De esta manera, se comparará el resultado obtenido con la metaheurística GO-GPS-GA contra el algoritmo exhaustivo GO-GPS, es decir, se comparará la aproximación del Pareto obtenida con GO-GPS-GA contra el Pareto óptimo obtenido con el método exhaustivo.

Para la ejecución de GO-GPS-GA la población inicial está formada por un 50 % de subárboles elegidos aleatoriamente de la base de datos y un 50 % de instancias completamente aleatorias generadas a partir de la jerarquía de GO. Este procedimiento particular resultó necesario debido a que no todos los términos de GO aparecían en los datos de entrada, haciendo más difícil que el algoritmo encontrara buenas soluciones.

Los parámetros del algoritmo GO-GPS-GA para este dominio se muestran en la Tabla 5.1.

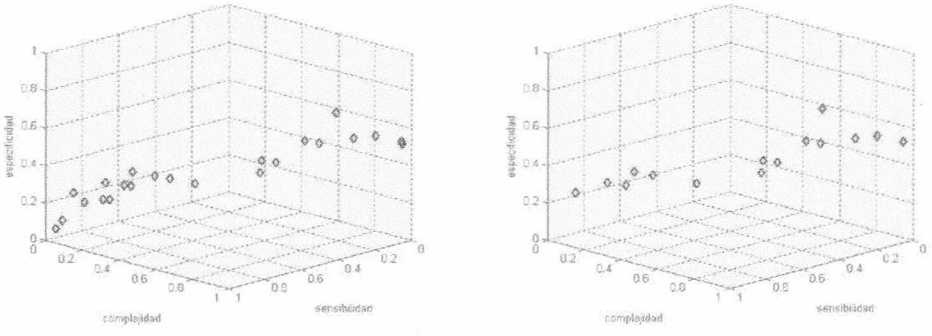
Parámetro	Valor
Tamaño de la población	200
Número de evaluaciones	20000
Probabilidad de cruce	0,6
Probabilidad de mutación	0,2

Tabla 5.1: Parámetros del algoritmo GO-GPS-GA para el dominio *Gene Ontology*.

En el contexto de las investigaciones sobre convergencia al frente del Pareto óptimo, una de las métricas que se utiliza es la función $\mathcal{M}_1^*$ que se define en |ZT99|. La función $\mathcal{M}_1^*$ proporciona la distancia media del Pareto aproximado Y' al Pareto óptimo $\bar{Y}$. Las soluciones del método aproximado pueden estar en el Pareto óptimo o fuera de él. En el caso de estar dentro del frente óptimo del Pareto, su distancia al mismo será cero; en caso contrario, se calcula la distancia a la solución del Pareto más cercana y se toma este valor como indicador de la calidad de esa solución:

$$\mathcal{M}_1^*(Y') = \frac{1}{|Y'|} \sum_{a' \in Y'} min\{\|a' - \bar{a}\|; \bar{a} \in \bar{Y}\}$$

En la Figura 5.12 se presentan los dos Paretos obtenidos. El gráfico (a) corresponde al Pareto obtenido en el capítulo anterior con el método exhaustivo y el (b) corresponde al Pareto aproximado obtenido con el método GO-GPS-GA. Se puede observar que los Paretos son muy parecidos, con lo cual el GO-GPS-GA no pierde mucha información y obtiene una buena aproximación al Pareto óptimo. Para corroborar esta observación se utiliza la métrica definida, cuyo valor $\mathcal{M}_1^* = 0{,}037$ resulta cercano a cero confirmando la similitud de ambos Pareto. Por lo tanto, GO-GPS-GA obtiene una buena aproximación al Pareto óptimo.



(a) (b)

Figura 5.12: el gráfico (a) corresponde al Pareto óptimo obtenido con el método exhaustivo GO-GPS y el gráfico (b) corresponde al Pareto aproximado obtenido con GO-GPS-GA.

5.5. Validación y explicación de perfiles de expresión genética

Como resultado de la aplicación del método GO-GPS-GA al conjunto de 1776 genes extraídos del experimento biológico *Inflamación y respuesta del huésped a estímulos externos* introducido en la Sección 2.3, se ha obtenido un total de 156 clusters (Apéndice A). Las Figuras 5.13 y 5.14 presentan el Pareto obtenido con el algoritmo en el espacio de objetivos y el espacio de variables respectivamente. Estos Paretos serán utilizados en los análisis comparativos realizados en la sección 5.6.

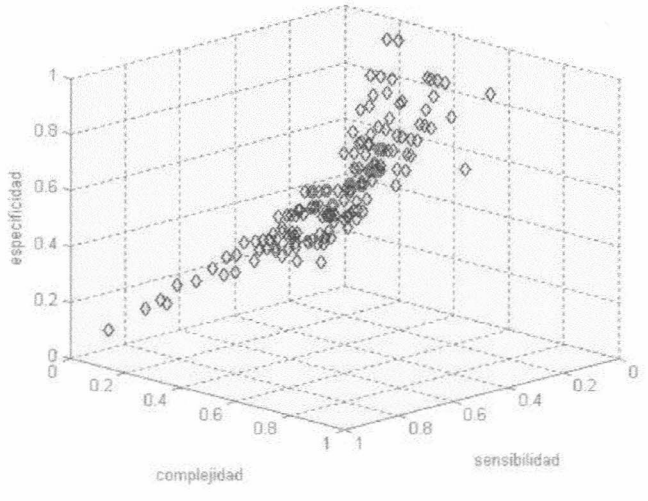


Figura 5.13: Gráfico del conjunto Pareto en el espacio de objetivos obtenido con el método GO-GPS-GA utilizando como entrada el conjunto de 1776 genes.

El principal objetivo en el presente trabajo es utilizar la información de *Gene Ontology* para analizar los perfiles de expresión de genes en sangre de humanos voluntarios

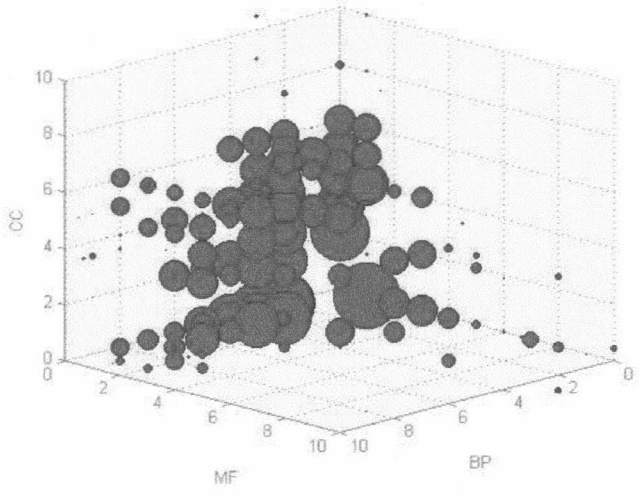


Figura 5.14: Gráfico del conjunto Pareto en el espacio de variables obtenido con el método GO-GPS-GA utilizando como entrada el conjunto de 1776 genes.

tratados con endotoxina intravenosa comparado contra placebo. Para extraer conocimiento de este experimento se estudió la expresión de genes a través del tiempo. Utilizando el algoritmo k-medias se han clasificado los genes de acuerdo a su expresión a través del tiempo en 24 clusters. En la sección 2.3 se ha presentado el experimento en mayor profundidad y se han mostrado los 24 clusters de expresión obtenidos. A partir de este experimento se tiene dos agrupamientos diferentes del mismo conjunto de genes, los perfiles de expresión por un lado, y los clusters obtenidos utilizando GO-GPS-GA por otro. Luego se pueden utilizar los clusters conceptuales obtenidos junto con su información de contexto (términos GO) para encontrar explicación biológica sobre los genes que se han expresado de la misma manera. La comparación se hace calculando la intersección entre todos los clusters de expresión y los clusters obtenidos por GO-GPS-GA. Para cada intersección de cluster de expresión (*Clustersexpresion*) y cluster de GO-GPS-GA (*ClustersGO*) calculamos el p-value utilizando la siguiente expresión [THC+99]:

$$P(cluster\ expresion, clusterGO) = 1 - \sum_{i=0}^{k-1} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}} \tag{5.6}$$

donde f es el número total de genes que pertenecen al cluster de expresión, n es el tamaño del cluster GO, k es el número de genes pertenecientes a la intersección de ambos clusters, y g es el número total de genes de todos los clusters. Esta fórmula indica la probabilidad de observar al menos k elementos del cluster de expresión en el cluster GO. Cuando más cercano a 1 es el valor del p-value, mayor es la probabilidad de que la intersección sea simplemente por azar y no sea relevante. En contraposición, cuanto menor sea este valor, más relevante será la intersección. Se definirá entonces un valor de umbral δ que permitirá decidir a partir de qué valor de p-value se considerará que las intersecciones obtenidas no son aleatorias. Se utilizará el valor $\delta = 3, 10^{-4}$.

Los resultados se presentan en la Figura 5.15. La escala de colores se utilizan para indicar los p-values de la intersección. El color rojo intenso es el p-value más bajo encontrado y el verde intenso es el más alto, siempre dentro del umbral δ. El radio

de los círculos indican la cantidad de genes en la intersección (a mayor radio, mayor cantidad de genes).

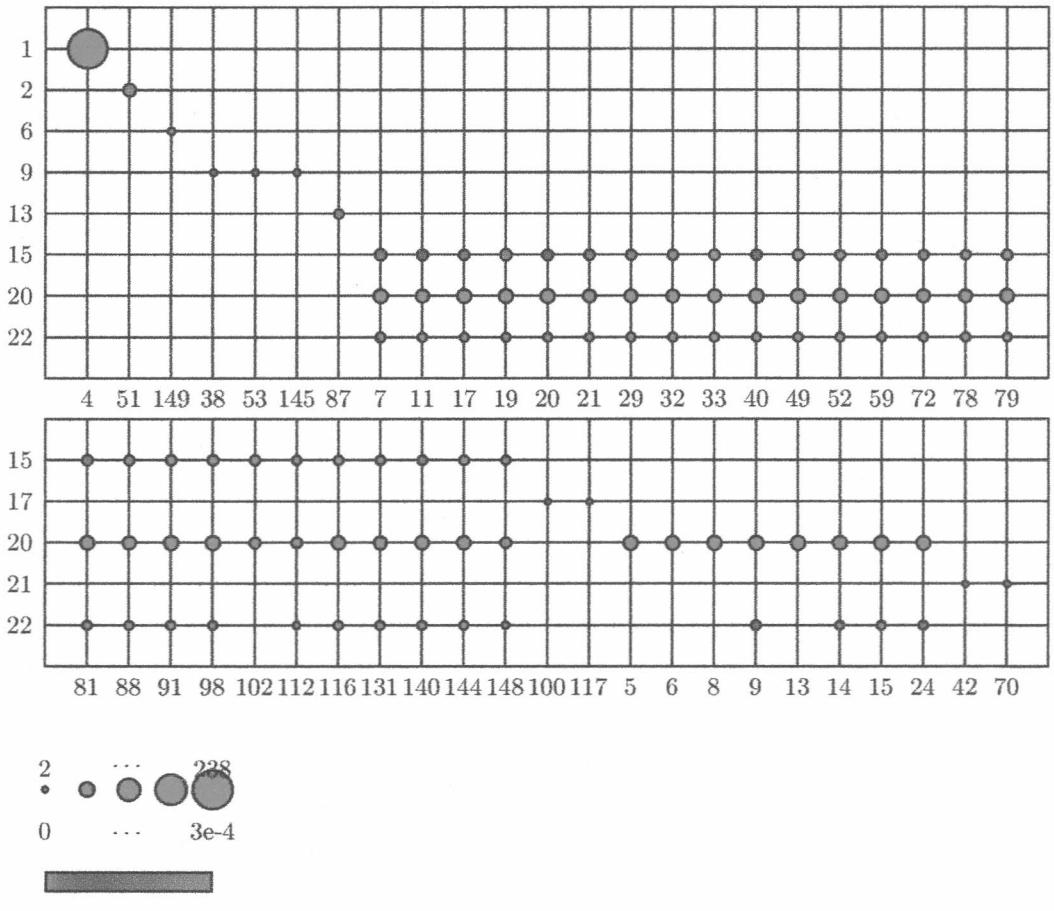


Figura 5.15: Resultados de la intersección de los clusters de expresión con los clusters obtenidos con el método GO-GPS-GA con Jaccard (clusters GO). Las filas son los clusters de expresión y las columnas los clusters GO.

A modo de ejemplo, se analizarán los clusters de expresión 9, 17 y 20. Como se puede apreciar, los clusters 9 y 17 intersecan con pocos clusters GO. Lo contrario pasa con el cluster 20, el cual interseca con numerosos clusters GO.

Cluster 9: Los clusters GO que intersecan con el cluster de expresión 9 son el 38, el 53 y el 145. En la Figura 5.16 se muestran los gráficos de la expresión de los genes de estos clusters GO y puede verse como dichos clusters dividen al cluster de expresión en 3 grupos distintos, que pueden ser no disjuntos. En la Tabla 5.2 se muestran los términos de estos clusters GO. Se puede observar que se encuentran términos en procesos biológicos y en funciones moleculares a niveles 7 y 5 respectivamente, en ambos casos bastante específicos. La herramienta *AmiGO* [AMI] permite visualizar la jerarquía de términos de GO como se observa en la Figura 5.17, que es un gráfico esquemático con el resultado que arrojó esta herramienta a una consulta realizada con los términos GO obtenidos para el perfil de expresión 9.

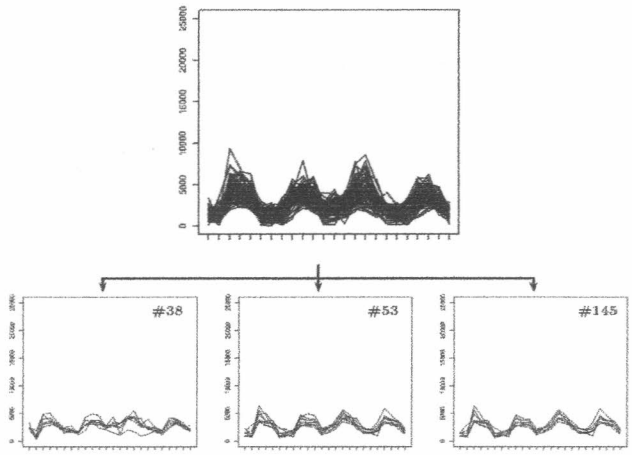


Figura 5.16: Expresión del cluster 9 y su relación con los Clusters GO con los cuales interseca. Los Clusters GO representados corresponden a la intersección entre éstos y el cluster 9 de expresión.

cluster GO	BP	MF	CC
38		GO:0008234 cysteine-type peptidase activity (level: 5)	
53	GO:0006917 induction of apoptosis (level: 7)	GO:0003674 molecular function (level: 1)	
145	GO:0006917 induction of apoptosis (level: 7)	GO:0004871 signal transducer activity (level: 5)	

Tabla 5.2: Términos GO de los clusters de GO-GPS-GA que intersecan con el cluster de expresión 9.

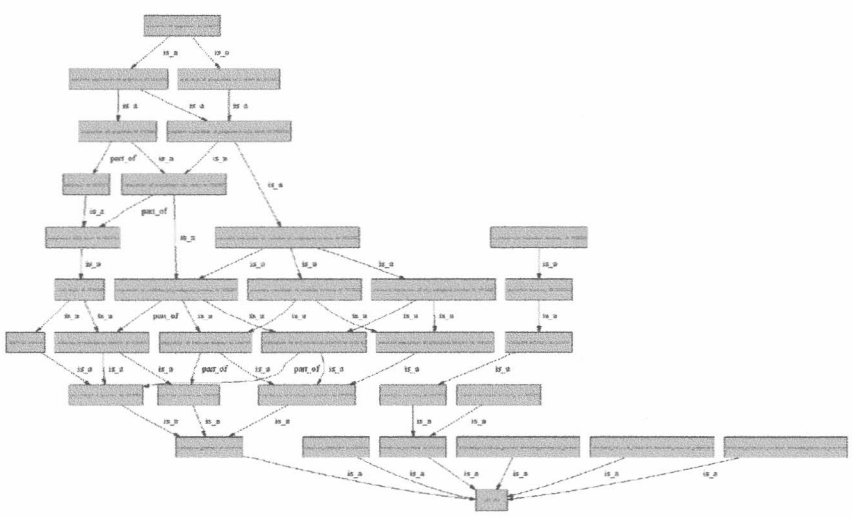


Figura 5.17: subgrafo de GO obtenido con *AmiGO* al realizar una consulta introduciendo los términos de los clusters que intersecan con el cluster de expresión 9.

Cluster 17: En la Tabla 5.3 se muestran las características de los clusters GO que intersecan con este perfil de expresión. En la Figura 5.18 se presenta el gráfico esquemático del grafo de GO correspondiente a la unión de todos estos términos encontrados. Nuevamente aquí se observan términos bastante específicos, como se puede apreciar en la densidad del grafo. Comparando este grafo con el del cluster 9 (ver Figura 5.17) se observa una mayor especificidad y densidad de términos presentes.

cluster GO	BP	MF	CC
100	GO:0007253 cytoplasmic sequestering of NF-kappaB (level: 8)		GO:0005622 intracellular (level: 4)
117	GO:0045449 regulation of transcription (level: 8) GO:0044267 cellular protein metabolism (level: 7)	GO:0003677 DNA binding (level: 4)	GO:0005634 nucleus (level: 5)

Tabla 5.3: Términos GO de los clusters de GO-GPS-GA que intersecan con el cluster de expresión 17.

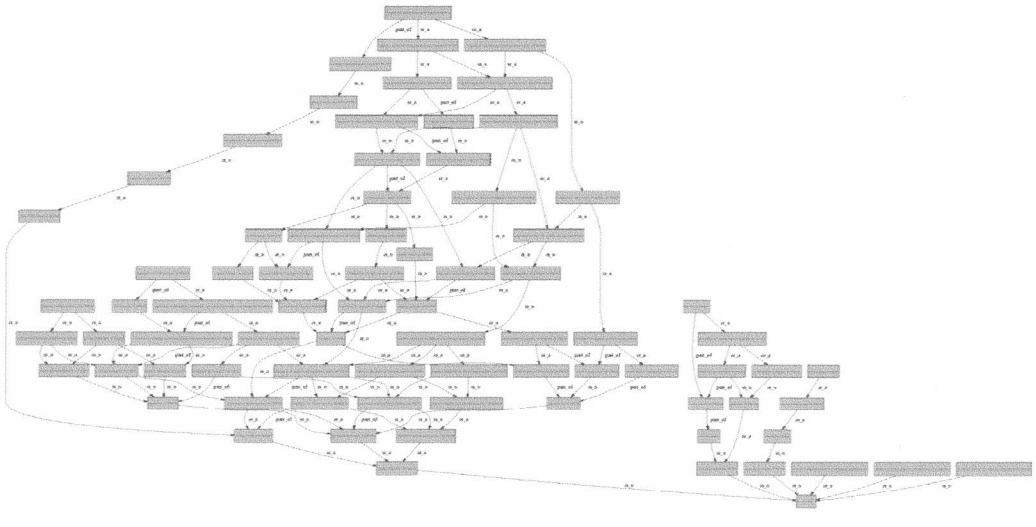


Figura 5.18: subgrafo de GO obtenido con los términos de los clusters que intersecan con el cluster de expresión 17.

Cluster 20: Con respecto al cluster de expresión 20, no se presentará la tabla con el detalle de las características de los ClustersGO que lo intersecan por motivos de espacio (ver apéndice A). En la Figura 5.19 se presenta el gráfico esquemático del grafo que queda determinado por todos los ClustersGO que intersecan con el cluster de expresión 20. Se puede observar que este perfil de expresión interseca con numerosos clusters, sin embargo, si se compara el grafo con los correspondientes a los perfiles 9 y 17, este es menos profundo y menos denso, lo cual indica que la información de *Gene Ontology* es menos específica.

Si se observa nuevamente la Figura 5.15, el perfil de expresión 1 no pasa desapercibido, puesto que es el que mayor intersección tiene en cuanto al número de genes y su

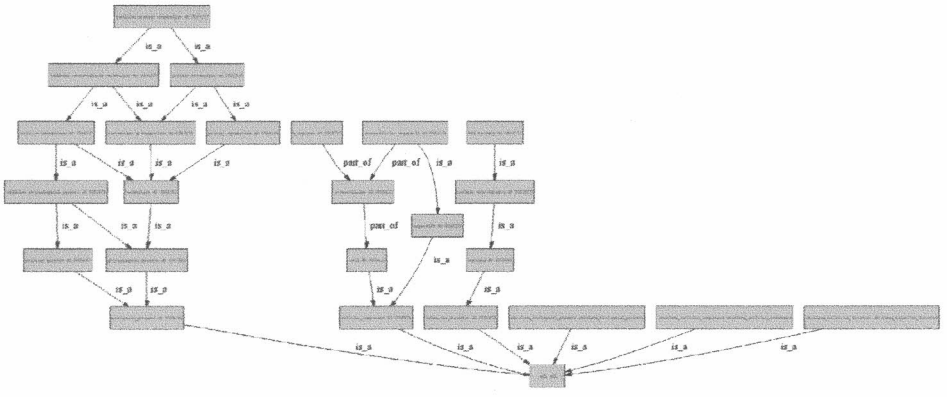


Figura 5.19: subgrafo de GO con los términos de los clusters que intersecan con el perfil de expresión 20.

p-value es bajo, como lo indica el color rojo intenso. El único cluster GO con el cual interseca es el 4. Este cluster tiene un único término GO, a saber: "*molecular function*" lo cual indica que los genes de la intersección tienen alguna función molecular anotada. Esta información no especifica cuales son las funciones de estos genes, por lo tanto, este es un caso en el cual la intersección de genes es alta pero la información de GO es escasa.

Los clusters analizados anteriormente fueron tomados como ejemplo para mostrar que el método permite obtener grupos de genes utilizando la información de *Gene Ontology*. Luego estos grupos se pueden utilizar para explicar perfiles de expresión. Para cada intersección posible (que sea estadísticamente significativa) de clusters GO con perfiles de expresión se puede deducir que "*se tienen grupos de genes cuya expresión ha sido similar a través del tiempo y a su vez se tiene una descripción conceptual basada en Gene Ontology que presenta características biológicas de estos grupos*". Estos resultados son un aporte dentro del programa de investigación *Inflamación y respuesta del huésped a estímulos externos*. Científicos de la biología se encuentran actualmente analizando estos resultados junto con otros experimentos dentro del programa.

5.6. Comparación con otros métodos

En esta sección se analizará el comportamiento del método GO-GPS-GA comparándolo con otros dos métodos, APRIORI [AIS93] y FatiGO [FASD04], para lo cual se utilizará como datos de entrada la base de datos completa de 1776 genes. Por un lado, se aplicará el método computacional de machine learning APRIORI y se analizarán las soluciones no dominadas obtenidas como resultado comparándolas con GO-GPS-GA. Por otro lado, se utilizará la herramienta FatiGO presentada anteriormente en la sección 3.2. Teniendo en cuenta el procedimiento utilizado en esta última herramienta no se aplicará no dominancia a las soluciones, ya que siempre se obtienen grupos con un único término GO y en una determinada ontología, por lo cual el objetivo *complejidad* pierde sentido (siempre vale 1). Sin embargo, se realizará un análisis ejecutando FatiGO sucesivas veces, comparando la calidad de la información de *Gene Ontology* brindada por esta herramienta y GO-GPS-GA.

Ya se ha mencionado en el capítulo anterior que en clustering conceptual se desea obtener descripciones profundas de los clusters. En problemas como los de expresión

genética, donde se hace mining sobre algo desconocido, se hace más importante este hecho. La diversidad de soluciones sirve para poder explicar mejor la expresión con datos externos. En las comparaciones realizadas en esta sección se tendrá en cuenta la diversidad de soluciones obtenidas por cada método.

5.6.1. Comparación con APRIORI

El algoritmo APRIORI consiste en un proceso sencillo de dos etapas: generar y combinar. La primera etapa genera conjuntos de elementos (*itemsets*) más frecuentes de tamaño k y luego, durante la segunda etapa, se combinan para generar *itemsets* de tamaño $k+1$. Solamente luego de explorar todas las posibilidades de asociación conteniendo k elementos, se consideran aquellos conjuntos de $k+1$ elementos. El pseudocódigo del algoritmo APRIORI se muestra en el Algoritmo 5.2.

Algoritmo 5.2 APRIORI

APRIORI (D base de datos)
  $L_1 \leftarrow$ {1-*itemsets* más frecuentes}
  $k \leftarrow 2$
  repeat
    $C_k \leftarrow$ k-*itemsets* generados a partir de L_{k-1}
    para todos $t \in D$ **hacer**
      Incrementar el contador de todos los candidatos de C_k que estén cubiertos por t
    fin para
    $L_k \leftarrow$ Todos los candidatos de C_k con soporte mínimo
    $k \leftarrow k+1$
  until $L_{k-1} = \emptyset$

En la Figura 5.20 se visualizan los Paretos obtenidos con APRIORI y GO-GPS-GA. Se puede observar que APRIORI obtiene un número limitado de soluciones, las cuales a su vez no resultan muy específicas y están aglomeradas en un extremo del Pareto. Esto último se puede visualizar en la Figura 5.20 (b), en la cual las soluciones presentan valores de especificidad inferiores a 0.3. Asimismo, en la Figura 5.20 (b') se aprecia que las esferas son de gran tamaño, lo cual significa que hay numerosos genes en esos clusters, sin embargo los niveles para los términos en las ontologías resultan inferiores a 1. GO-GPS-GA tiene mucha más diversidad en sus soluciones que APRIORI, y este hecho es positivo a la hora de explicar pérfiles de expresión.

Para comparar cuantitativamente ambos paretos se utilizarán dos métricas de comparación. La primera medida propuesta utiliza una función que mapea un par de soluciones (X_1,X_2) a un intervalo [0,1] [ZDT00]:

$$\mathcal{C}(X_1, X_2) = \frac{|\{a_2 \in X_2; \exists a_1 \in X_1 : a_2 \preceq a_1|}{|X_2|} \tag{5.7}$$

El valor extremo $\mathcal{C}(X_1, X_2) = 1$ significa que todas las soluciones en X_2 están dominadas o son iguales que las soluciones de X_1. El valor extremo $\mathcal{C}(X_1, X_2) = 0$, por su parte, representa la situación en la cual ninguna de las soluciones de X_2 está cubierta por el conjunto X_1. Es necesario notar que tanto $\mathcal{C}(X_1, X_2)$ como $\mathcal{C}(X_2, X_1)$ tienen que ser considerados ya que $\mathcal{C}(X_1, X_2) \neq 1 - \mathcal{C}(X_2, X_1)$.

La segunda medida propuesta utiliza una función que mapea un par de soluciones (X_1,X_2) a un intervalo [0,1] [ZZR04]:

$$\mathcal{ND}(X', X'') = |\{a' \in X' \wedge a' \notin X'' : (\forall a'' \in X'' : a' \not\preceq a'')\}| \tag{5.8}$$

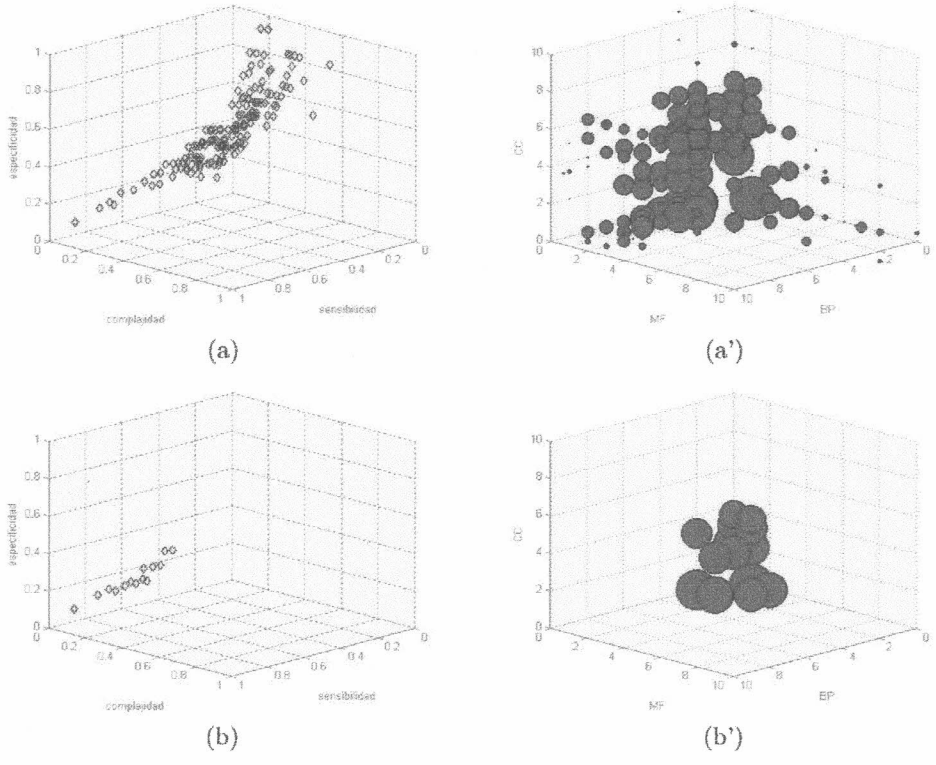


Figura 5.20: Gráficos de los Paretos obtenidos con GO-GPS-GA y APRIORI. Las figuras (a) y (a') corresponden al Pareto en el espacio de objetivos y en el espacio de variables respectivamente para el método GO-GPS-GA y (b) y (b') para el método APRIORI.

La medida $\mathcal{ND}(X', X'')$ compara dos conjuntos de soluciones no dominadas y devuelve el número de soluciones de X' que no son iguales y no son dominadas por ningún miembro de X''. Una vez más, ambas medidas, $\mathcal{ND}(X', X'')$ y $\mathcal{ND}(X'', X')$, deben ser tenidas en cuenta. Existe una diferencia clara entre las medidas $\mathcal{ND}$ y $\mathcal{C}$: la última muestra la relación de dominancia entre dos conjuntos de soluciones, mientras que la primera cuenta el número de soluciones novedosas, perteneciente al primer conjunto que no descubre el segundo.

Los resultados de las métricas de comparación confirman nuestras afirmaciones anteriores. Con respecto a la métrica $\mathcal{C}$, como puede verse en la Tabla 5.3(a) y la Figura 5.20, vemos que APRIORI domina a algunas de las soluciones de GO-GPS-GA, y a su vez GO-GPS-GA domina a casi la mitad de las soluciones de APRIORI.
La métrica $\mathcal{ND}$ muestra en la Tabla 5.3(b) que GO-GPS-GA se comporta mejor que APRIORI en sentido que descubre 146 soluciones que APRIORI no descubre ni domina, mientras que APRIORI encuentra 8 soluciones que GO-GPS-GA no llega a obtener.

(a) Métrica $\mathcal{C}$

$\mathcal{C}(X', X'')$	APRIORI	GO-GPS-GA
APRIORI	-	0.06
GO-GPS-GA	0.43	-

(b) Métrica $\mathcal{ND}$

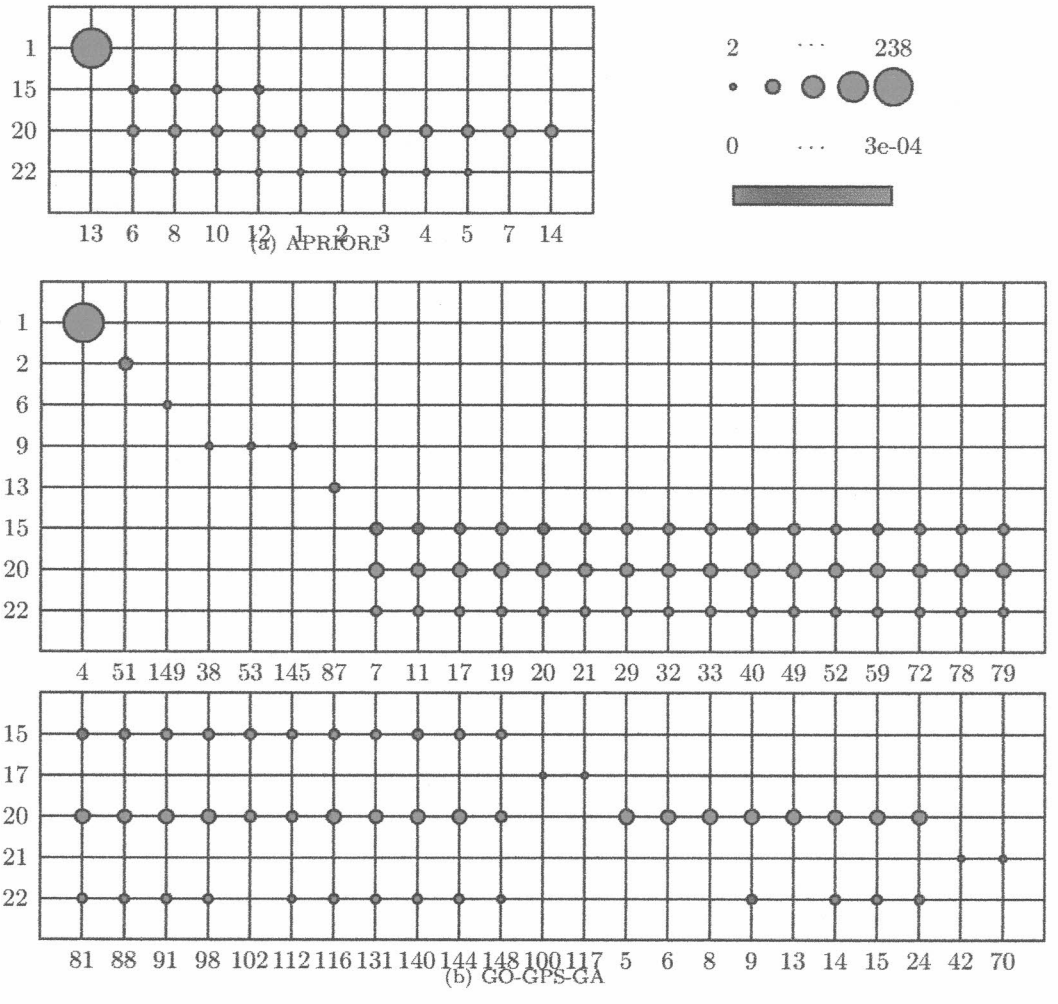
$\mathcal{ND}(X', X'')$	APRIORI	GO-GPS-GA
APRIORI	-	8
GO-GPS-GA	146	-

Tabla 5.4: Resultado de las métricas $\mathcal{C}$ y $\mathcal{ND}$ para los algoritmos APRIORI y GO-GPS-GA.

Desde el punto de vista de la biología, otro tipo de comparación que se puede realizar entre APRIORI y el método presentado en esta tesis es analizando como logran explicar los perfiles de expresión del problema biológico. En la Figura 5.21(a) se ilustra las intersecciones de APRIORI con respecto a los perfiles de expresión junto con las intersecciones de GO-GPS-GA (figura 5.21(b)).

APRIORI encuentra un menor número de clusters relevantes comparado con el método GO-GPS-GA (este hecho se produce porque la diversidad de las soluciones de GO-GPS-GA es mucho mayor). Más aún, todos ellos están incluidos en el conjunto de clusters obtenidos por nuestro método (clusters de expresión 1, 15, 20 y 22). Asimismo, los clusters 15, 20 y 22 contienen numerosos conjuntos en los clusters GO y en los clusters APRIORI que los intersecan. Sin embargo, las descripciones asociadas a ellos no son muy específicas, con solamente uno o dos términos GO en el segundo o tercer nivel de la jerarquía. El cluster 1 contiene únicamente una intersección en ambos algoritmos y se ha visto en la sección 5.5 que su única descripción está a nivel 1 en la jerarquía y es muy poco específica.

Estos grupos, a pesar de tener asociados información de *Gene Ontology* a niveles bajos, son descripciones válidas que cubren un amplio conjunto de genes diferentes.



(a) APRIORI

(b) GO-GPS-GA

Figura 5.21: Intersección de los perfiles de expresión y los clusters GO para APRIORI y GO-GPS-GA.

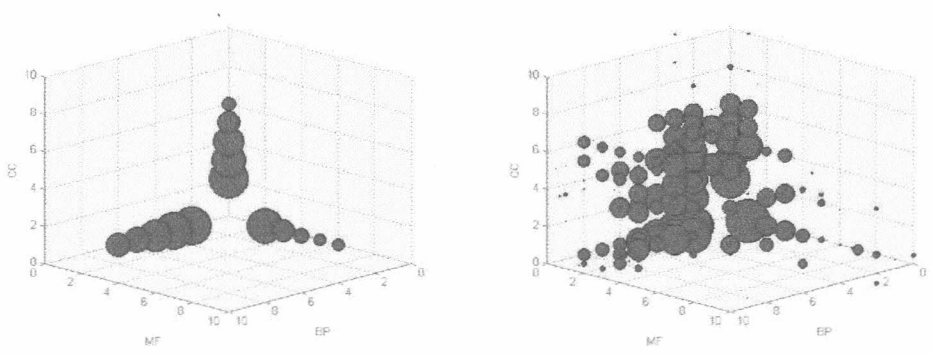
5.6.2. Comparación con FatiGO

FatiGO es una herramienta que utiliza la base de datos estructurada de *Gene Ontology* para realizar un análisis estadístico de uno o dos grupos de genes. Con un único grupo de genes de entrada, la herramienta realiza una búsqueda en diversas bases de datos para encontrar los términos GO de cada gen. Luego presenta la frecuencia de aparición de cada término en el grupo de genes mediante un histograma.

Existen dos aspectos de FatiGO que se han propuesto mejorar en el presente trabajo. Por un lado, no limitarse a un determinado nivel de la jerarquía GO en una determinada ejecución, y por otro lado, utilizar las tres ontologías al mismo tiempo. Con respecto a esto, GO-GPS-GA explora toda la jerarquía GO en sus tres ontologías y en todos sus niveles en una única ejecución.

Con el objeto de comparar los resultados obtenidos por FatiGO contra los obtenidos por GO-GPS-GA, se ha ejecutado FatiGO 15 veces, una por cada ontología y para cada nivel posible de 2 a 6. Los resultados obtenidos se presentan en la Figura 5.22. En dicha figura se pueden apreciar gráficamente las diferencias entre los clusters obtenidos por ambas herramientas. Los clusters de nuestro método obtienen descripciones de mayor calidad combinando términos GO de las tres ontologías. En el caso de FatiGO las bolas están distribuidas sobre los ejes (i.e. no hay combinación de las ontologías) mientras que en GO-GPS-GA se observan bolas distribuídas en todo el espacio de variables. Con respecto a la generalidad (i.e. tamaño de los clusters) se observa que GO-GPS-GA no disminuye esta propiedad en comparación con FatiGO (incluso hay esferas en GO-GPS-GA más grandes que en FatiGO).

Es importante recordar que cada cluster de FatiGO contiene una única descripción GO, mientras que los clusters de GO-GPS-GA pueden contener más de una. Por lo tanto, el poder de inferencia que se tendrá con los clusters de FatiGO es mucho menor que el que se tendrá con los resultados de GO-GPS-GA. Nuestro método obtiene un clustering con mejor diversidad y descripciones en cada cluster sin perder generalidad. Este hecho es importante puesto que permite explicar mejor los pérfiles de expresión como se analizará a continuación.



(a) (b)

Figura 5.22: Gráficos comparativos de los clusters obtenidos con FatiGO (a) y con GO-GPS-GA (b).

Desde el punto de vista biológico, los clusters obtenidos por FatiGO y GO-GPS-GA se pueden comparar analizando como logran explicar los perfiles de expresión. En las Figuras 5.23(a), 5.23(b) y 5.23(c) se presentan los resultados de la intersección para la ejecución de FatiGO a nivel 3 en las tres ontologías y en la Figura 5.23(d) los resultados para GO-GPS-GA. Se ha decidido ejecutar FatiGO a nivel 3 puesto que este es el nivel

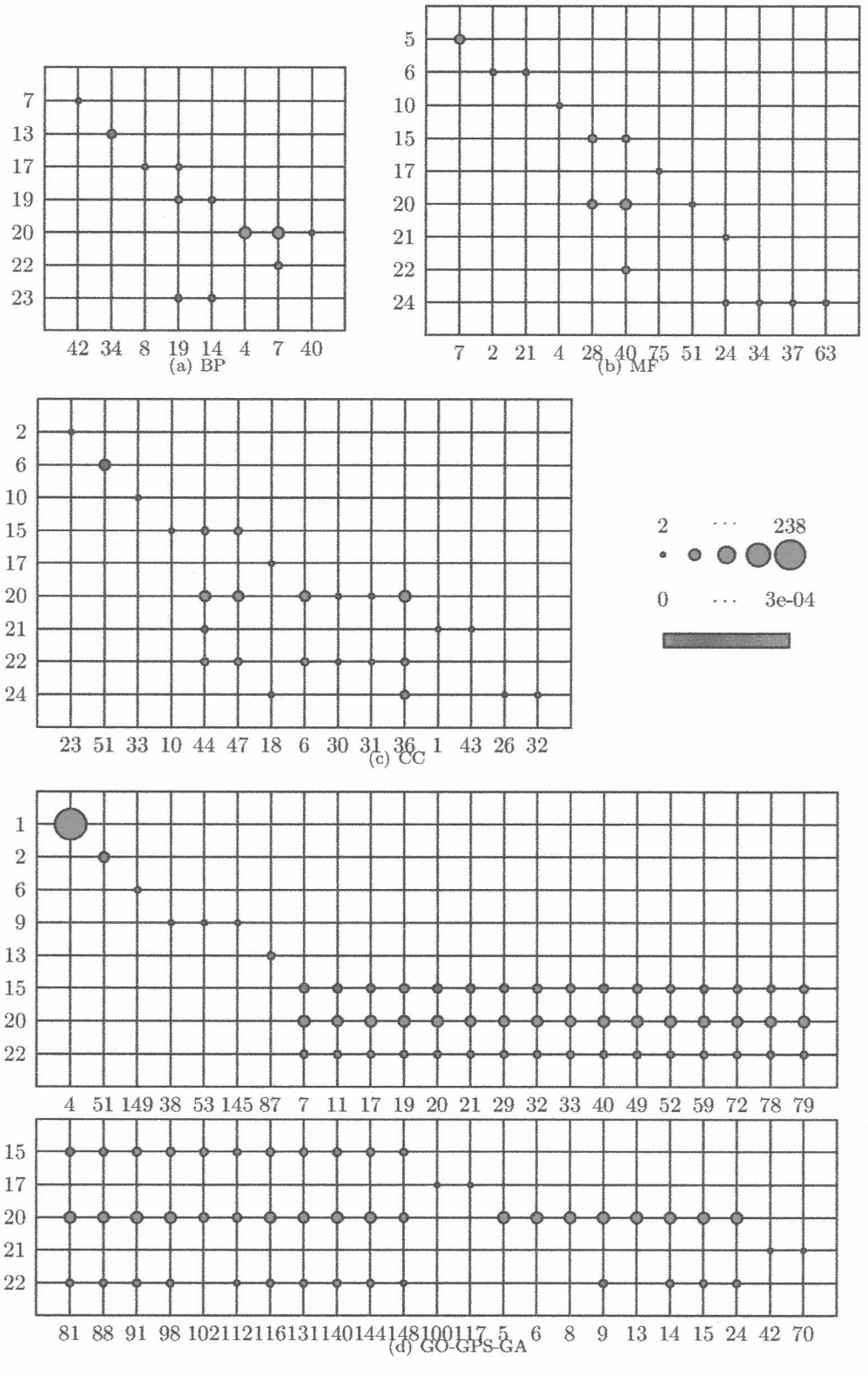


Figura 5.23: Intersección de los perfiles de expresión y los clusters de FatiGO y GO-GPS-GA. La ejecución de FatiGO se ha realizado a nivel 3 para las tres ontologías (BP, MF y CC)

sugerido por los autores para utilizar la herramienta [FASD04].

En las figuras se puede apreciar que hay perfiles con los cuales ambas herramientas tienen intersección, y perfiles para los cuales una de las herramientas tiene intersección y la otra no.

En el caso en que ambas herramientas tienen intersección (i.e. perfiles de expresión 6, 13, 15, 17, 20, 21, 22), GO-GPS-GA logra explicar los perfiles con información de GO más específica y por lo tanto de mejor calidad. Se tomarán dos ejemplos para mostrar este hecho, los perfiles de expresión 15 y 17. Con respecto al primer perfil de expresión, FatiGO consigue descripciones GO en las ontologías MF y CC a nivel 3, mientras que GO-GPS-GA consigue descripciones en BP a niveles 7, 8, 9, en MF a niveles 3, 4 y en CC a niveles 4, 5. como se puede apreciar en el apéndice A. Se observa además que gracias a la diversidad de soluciones de GO-GPS-GA, este perfil de expresión tiene intersección con numerosos clusters GO, esto permite obtiener descripciones de calidad superior a las obtenidas por FatiGO. Con respecto al perfil de expresión 17, FatiGO presenta descripciones para las tres ontologías a nivel 3, mientras que GO-GPS-GA obtiene también descripciones en las tres ontologías, pero a niveles 7 y 8 para BP, 4 para MF, 4 y 5 para CC (ver apéndice A). En ambos casos se puede ver que la calidad de las intersecciones, dada por el tamaño de los círculos y el color, se mantiene similar en ambas herramientas, con lo cual se puede concluir que GO-GPS-GA es más específico que FatiGO sin perder sensibilidad.

Por otro lado, existen dos perfiles de expresión que son explicados por GO-GPS-GA y no por FatiGO. Estos perfiles son el 1 y el 9. La descripción que encuentra GO-GPS-GA para el primer perfil es poco específica (a nivel 1) sin embargo, la cantidad de genes en la intersección es grande. Con respecto al perfil 9, el método encuentra descripciones en las ontologías MF y BP a niveles 5 y 7 respectivamente (ver apéndice A).

Se puede apreciar en la Figura 5.23 que FatiGO encuentra intersección con algunos pérfiles de expresión contra los cuales GO-GPS-GA no lo hace. Este hecho ocurre porque FatiGO, al realizar la búsqueda únicamente a un determinado nivel del grafo de GO, no implementa ningún criterio para discriminar soluciones en los grupos generados. En GO-GPS-GA es necesario aplicar un criterio de optimización y retornar únicamente las soluciones no dominadas, puesto que al realizar una búsqueda heurística en el grafo completo de GO, la cantidad de soluciones devueltas si no se aplicase esta discriminación sería demasiado grande.

Las características que diferencian al método GO-GPS-GA de otras técnicas que se pueden utilizar para analizar el mismo problema, ya sea técnicas computacionales de machine learning (e.g. APRIORI) como así también herramientas específicas para el análisis de grupos de genes utilizando *Gene Ontology* (e.g. FatiGO) son las siguientes:

- GO-GPS-GA difiere de los métodos de aprendizaje supervisados, puesto que para realizar su tarea no necesita ningún tipo de información brindada por expertos.
- GO-GPS-GA permite que un objeto pueda pertenecer a más de un cluster, a diferencia de otros métodos. Esto brinda mayor flexibilidad al momento de realizar las intersecciones de los clusters obtenidos contra los perfiles de expresión, permitiendo obtener mejores resultados.
- Los clusters obtenidos con GO-GPS-GA están acompañados de información sobre procesos biológicos, funciones moleculares y componentes celulares compartida por los genes. Esto facilita su interpretación y permite que sean utilizados para inferir características biológicas de los diferentes perfiles de expresión.

- GO-GPS-GA obtiene mejor diversidad que APRIORI. Esto brinda mayor flexibilidad al momento de realizar las intersecciones de los clusters obtenidos contra los perfiles de expresión, permitiendo obtener mejores resultados.
- GO-GPS-GA es una mejora con respecto a FatiGO puesto que, al analizar la jerarquía de *Gene Ontology* en su totalidad, presenta grupos de genes que comparten términos GO en diferentes ontologías y a diferentes niveles. GO-GPS-GA mejora la diversidad y las descripciones de los clusters, sin perder generalidad.

5.7. Observaciones finales

En este capítulo se ha presentado GO-GPS-GA, un método basado en la metodología *CC-EMO (Clustering conceptual basado en evolución multiobjetivo)* que realiza clustering conceptual en conjuntos de genes utilizando los mismos objetivos definidos en GO-GPS.

Se ha ejecutado el método con un subconjunto de 62 genes para comparar la calidad de los resultados del algoritmo genético GO-GPS-GA contra el algoritmo exhaustivo GO-GPS. Los resultados han sido los esperados, puesto que la aproximación al Pareto óptimo resultó satisfactoria, verificándose la misma tanto gráficamente como cuantitativamente.

Asimismo, se han analizado los resultados obtenidos con la aplicación de GO-GPS-GA al conjunto completo de 1776 genes y se han comparado con los resultados obtenidos por APRIORI y por FatiGO. En el caso de APRIORI, se ha observado que este método descubre pocas soluciones y las mismas resultan poco específicas en comparación con las soluciones obtenidas por GO-GPS-GA. En el caso de FatiGO, se ha visto que los grupos de genes obtenidos contienen información de *Gene Ontology* de una calidad inferior a las soluciones encontradas por GO-GPS-GA. Esto se debe a que este último método explora las tres ontologías de GO en todos sus niveles al mismo tiempo, mientras que FatiGO necesita que se defina al principio en qué nivel y con qué ontología trabajar.

GO-GPS-GA presenta diversas ventajas con respecto a otros métodos estudiados. GO-GPS-GA no necesita ningún tipo de información brindada por expertos (i.e. información supervizada). Esto es importante en problemas como el que se estudia en está tesis, donde hay un alto grado de incertidumbre. Por otro lado, GO-GPS-GA presenta mayor diversidad en las soluciones y mejor calidad en las descripciones de los clusters, lo cual brinda mayor flexibilidad al momento de realizar las intersecciones con los perfiles de expresión, permitiendo obtener mejores resultados. GO-GPS-GA encuentra un balance entre las propiedades deseables de un clustering conceptual, mejorando la diversidad y la calidad de las descripciones de los clusters, sin perder generalidad.

Capítulo 6

Conclusiones y trabajo futuro

En el presente trabajo de tesis se ha propuesto un método capaz de extraer conocimiento de bases de datos estructuradas, inspirado en la técnica de clustering conceptual y basado en técnicas de optimización multiobjetivo. El método ha sido diseñado para ser utilizado en el análisis de datos provenientes de experimentos biológicos de expresión genética, no obstante puede ser adaptado fácilmente para su aplicación a otros dominios.

Como se ha visto, existen varias herramientas que pueden ser utilizadas en este tipo de problemas. Se han considerado diversas técnicas tradicionales de data mining, pero todas ellas presentan inconvenientes a la hora de manejar datos estructurados, además de no producir información conceptual o cualitativa sobre los clusters obtenidos. Otro tipo de herramientas utilizan específicamente información de *Gene Ontology* para analizar conjuntos de genes, como por ejemplo FatiGO. Esta herramienta presenta dos aspectos donde puede ser mejorada. Por un lado, FatiGO no utiliza todos los niveles de la jerarquía al mismo tiempo y por otro lado, la herramienta no considera las ventajas que puede brindar la combinación de las tres ontologías para encontrar grupos de genes que compartan términos GO.

El método propuesto utiliza la base de datos estructurada de *Gene Ontology* para obtener clusters de genes junto con características biológicas compartidas entre estos genes. Estos grupos pueden ser utlizados por científicos para explicar perfiles de expresión en términos de los procesos biológicos, funciones moleculares y componentes celulares que compartidos por los genes de estos perfiles.

Existen características que permiten evaluar cualitativamente un clustering conceptual. Una propiedad deseable es que el clustering tenga la mayor cobertura con el mínimo número posible de clusters, por otro lado, es deseable obtener descripciones con varias características para cada cluster aumentando de esta manera el poder de inferencia. La tercer propiedad que se busca es obtener el mínimo solapamiento entre los clusters. Encontrar los objetivos para obtener un clustering conceptual de alta calidad es dificultoso puesto que estas tres propiedades son conflictivas (i.e. el aumento en una de ellas causa una disminución en las otras). Se ha presentado el algoritmo exhaustivo GO-GPS, el cual ha sido utilizado para estudiar diferentes funciones en el proceso de optimización, obteniendo finalmente tres objetivos: *sensitividad*, *complejidad* y *especificidad*. GO-GPS logra encontrar un balance entre las tres propiedades deseables del clustering conceptual utilizando estos tres objetivos, que luego han sido incorporados a GO-GPS-GA, el método propuesto basado en algoritmos evolutivos multiobjetivo.

Se ha evaluado el comportamiento de GO-GPS-GA con un subconjunto de 62 genes

para comparar la calidad de los resultados contra el algoritmo exhaustivo GO-GPS. Los resultados han sido los esperados, puesto que la aproximación al Pareto óptimo resultó satisfactoria.

El método GO-GPS-GA ha sido utilizado luego para analizar un nuevo experimento biológico que constituye un esfuerzo conjunto de varias instituciones médicas e investigadores por explicar el sistema inflamatorio en seres humanos. Este experimento consiste en varios estudios de Microarrays realizados en distintos instantes de tiempo a un grupo de humanos voluntarios tratados con endotoxina intravenosa comparados contra placebo. Los resultados del estudio han sido agrupados utilizando el algoritmo k-medias con el cual se obtienen 24 perfiles de expresión.

Se ha cruzado la información de los clusters obtenidos con GO-GPS-GA contra los perfiles de expresión logrando resultados satisfactorios. Se ha podido identificar con información de *Gene Ontology* el comportamiento de varios grupos de genes que se han expresado de la misma manera a través del tiempo. Los resultados obtenidos constituyen una fuente de información para investigaciones del área biológica que actualmente está siendo utilizada.

Los resultados obtenidos con la aplicación de GO-GPS-GA al experimento biológico se han comparado con los obtenidos por otros dos métodos, APRIORI y FatiGO. En el caso de APRIORI, se ha observado que este método descubre pocas soluciones y las mismas resultan poco específicas en comparación con las soluciones obtenidas por GO-GPS-GA. En el caso de FatiGO, se ha visto que los grupos de genes obtenidos contienen información de *Gene Ontology* de una calidad inferior a las soluciones encontradas por GO-GPS-GA.

A continuación se detallan las características que diferencian al método GO-GPS-GA de otras técnicas que se pueden utilizar para analizar el mismo problema, ya sea técnicas generales de data mining como así también herramientas específicas para el análisis de grupos de genes utilizando *Gene Ontology*:

- GO-GPS-GA difiere de los métodos de aprendizaje supervisados, puesto que para realizar su tarea no necesita ningún tipo de información brindada por expertos.
- GO-GPS-GA permite que un objeto pueda pertenecer a más de un cluster, a diferencia de otros métodos. Esto brinda mayor flexibilidad al momento de realizar las intersecciones de los clusters obtenidos contra los perfiles de expresión, permitiendo obtener mejores resultados.
- Los clusters obtenidos con GO-GPS-GA están acompañados de información sobre procesos biológicos, funciones moleculares y componentes celulares compartida por los genes. Esto facilita su interpretación y permite que sean utilizados para inferir características biológicas de los diferentes perfiles de expresión.
- GO-GPS-GA es una mejora con respecto a FatiGO puesto que, al analizar la jerarquía de *Gene Ontology* en su totalidad, presenta grupos de genes que comparten términos GO en diferentes ontologías y a diferentes niveles.
- Los objetivos de optimización utilizados tanto en GO-GPS como en GO-GPS-GA permiten encontrar un balance entre las tres propiedades deseables del clustering conceptual, mejorando la diversidad y la calidad de las descripciones de los clusters con respecto a otros métodos, sin perder generalidad.

Concluimos entonces que los objetivos primordiales han sido alcanzados. Por un lado, se ha logrado mejorar la herramienta FatiGO, puesto que el método presentado

explora la jerarquía de *Gene Ontology* en su totalidad en una única ejecución. Por otro lado, se han obtenido clusters de genes que comparten información sobre procesos biológicos, funciones moleculares y componentes celulares y que potencialmente pueden ser utilizados por científicos para explicar biológicamente perfiles de expresión genética.

Como trabajo futuro existen varias líneas de investigación a seguir.

Una línea de investigación es la adaptación y aplicación del método a otras áreas, entre las cuales se puede encontrar Economía, Internet (Web Ontologies), etc.

En referencia a las posibles aplicaciones de la metodología a otras áreas de la biología, existen diversos problemas que almacenan información en bases de datos estructuradas con los que se podría trabajar. Un ejemplo es la base de datos BIND (Biomolecular Interaction Network Database) [AAA+05], que acumula información sobre *interacciones proteína-proteína*. Esta base de datos es una colección de información sobre interacciones moleculares, cuyos contenidos incluyen información recuperada de la literatura científica y de datos obtenidos mediante experimentos biológicos de gran escala. En el repositorio BIND se acumulan asociaciones moleculares con tres clasificaciones: moléculas que se asocian unas con otras para formar interacciones, complejos moleculares que están conformados por una o más interacciones y caminos que están definidas por una secuencia específica de una o más interacciones. La aplicación de nuestra propuesta necesitaría un correcto modelado de la base de datos y, posiblemente, un estudio de funciones objetivos para la etapa de optimización.

Con respecto al método general y su aplicación al dominio de *Gene Ontology*, destacamos dos líneas de investigación posibles:

- Resulta de interés obtener el resultado del algoritmo exhaustivo con el conjunto completo de 1776 genes, por lo tanto, paralelizar el algoritmo exhaustivo es una posible tarea. Asimismo, el método GO-GPS-GA es un algoritmo genético y estos algoritmos resultan especialmente adaptables para su ejecución en paralelo. Existen varias maneras de paralelizar algoritmos genéticos y hay numerosos estudios sobre el tema para investigar y aplicar al método.

- Específicamente en el dominio de *Gene Ontology*, existe información que puede ser incorporada al método.
  El mapeo de un gen a un término GO puede estar basado en varios tipos de soporte, es decir, evidencia que sostiene la asociación. En GO existe un conjunto de *Códigos de Evidencia (Evidence Codes)* que se utilizan para clasificar cada mapeo. Algunos de estos códigos son:

  ```
  IMP: inferred from mutant phenotype
  IGI: inferred from genetic interaction
  IPI: inferred from physical interaction
  ISS: inferred from sequence similarity
  IDA: inferred from direct assay
  IEP: inferred from expression pattern
  IEA: inferred from electronic annotation
  TAS: traceable author statement
  NAS: non-traceable author statement
  ND: no biological data available
  IC: inferred by curator
  ```

  El mapeo más confiable es `TAS` y el menos confiable es `IEA`, puesto que ha sido deducido basado en métodos electrónicos. `IEA` es utilizado cuando ninguna

persona ha verificado la anotación para evaluar su exactitud o precisión. Esta información se podría utilizar para "medir" la confiabilidad de los clusters. Por ejemplo, si todos los genes de un cluster tienen mapeos con código de evidencia **TAS**, este grupo será más confiable comparado con otro que tenga todos sus mapeos **IEA**. El trabajo consistiría en incorporar esta información al método y utilizarla para ofrecer mayor confiabilidad en los resultados.

Apéndice A

Resultados completos

En el presente apéndice se presenta una tabla con todos los clusters obtenidos como resultado de la aplicación de GO-GPS-GA al conjunto completo de datos de 1776 genes. Recordemos que se han obtenido un total de 156 clusters como resultado.

Se presenta cada cluster con todos los términos GO que lo componen, junto con todos los niveles de la jerarquía donde cada uno de estos términos se encuentra.

cluster GO	BP	MF	CC
1	GO:0006917 induction of apoptosis (level: 7) GO:0045449 regulation of transcription (level: 8) GO:0042518 negative regulation of tyrosine phosphorylation of Stat3 protein (level: 11)	GO:0003674 molecular_function (level: 1)	GO:0005634 nucleus (level: 5) GO:0005829 cytosol (level: 5)
2	GO:0050875 cellular physiological process (level: 5)	GO:0046961 hydrogen-transporting ATPase activity ; rotational mechanism (level: 9)	GO:0005623 cell (level: 2)
3	GO:0007266 Rho protein signal transduction (level: 8) GO:0030036 actin cytoskeleton organization and biogenesis (level: 8) GO:0043123 positive regulation of I-kappaB kinase/NF-kappaB cascade (level: 9) GO:0042346 positive regulation of NF-kappaB-nucleus import (level: 8)	GO:0003674 molecular_function (level: 1)	
4		GO:0003674 molecular_function (level: 1)	
5	GO:0009987 cellular process (level: 2)		
6	GO:0050875 cellular physiological process (level: 5)	GO:0003674 molecular_function (level: 1)	
7			GO:0005622 intracellular (level: 4)
8	GO:0050875 cellular physiological process (level: 5)		

cluster GO	BP	MF	CC
9			GO:0005623 cell (level: 2)
10	GO:0007368 determination of left/right symmetry (level: 6) GO:0008285 negative regulation of cell proliferation (level: 7) GO:0042981 regulation of apoptosis (level: 9) GO:0046579 positive regulation of Ras protein signal transduction (level: 9) GO:0006355 regulation of transcription ; DNA-dependent (level: 6) GO:0030097 hemopoiesis (level: 4) GO:0016049 cell growth (level: 4) GO:0002011 morphogenesis of an epithelial sheet (level: 5)	GO:0005509 calcium ion binding (level: 5) GO:0004872 receptor activity (level: 4) GO:0005515 protein binding (level: 4)	GO:0005622 intracellular (level: 4) GO:0009986 cell surface (level: 3) GO:0005887 integral to plasma membrane (level: 6)
11			GO:0043229 intracellular organelle (level: 4)
12	GO:0050875 cellular physiological process (level: 5)	GO:0005488 binding (level: 2) GO:0046961 hydrogen-transporting ATPase activity ; rotational mechanism (level: 9)	GO:0043231 intracellular membrane-bound organelle (level: 4)
13	GO:0007582 physiological process (level: 3) GO:0009987 cellular process (level: 2)		

cluster GO	BP	MF	CC
14	GO:0050875 cellular physiological process (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0005623 cell (level: 2)
15	GO:0044237 cellular metabolism (level: 8)	GO:0003674 molecular_function (level: 1)	
16	GO:0006355 regulation of transcription ; DNA-dependent (level: 6) GO:0016573 histone acetylation (level: 12) GO:0006334 nucleosome assembly (level: 10) GO:0045941 positive regulation of transcription (level: 8) GO:0016481 negative regulation of transcription (level: 8)	GO:0003677 DNA binding (level: 4)	GO:0005634 nucleus (level: 5)
17	GO:0044237 cellular metabolism (level: 8)	GO:0003674 molecular_function (level: 1)	GO:0005622 intracellular (level: 4)
18		GO:0004871 signal transducer activity (level: 5)	
19	GO:0044237 cellular metabolism (level: 8)		
20	GO:0050875 cellular physiological process (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0005622 intracellular (level: 4)
21	GO:0009987 cellular process (level: 2)		GO:0043229 intracellular organelle (level: 4)
22	GO:0050875 cellular physiological process (level: 5)		GO:0005634 nucleus (level: 5)
23		GO:0003674 molecular_function (level: 1)	GO:0005859 muscle myosin (level: 6)

cluster GO	BP	MF	CC
24	GO:0050875 cellular physiological process (level: 5)		GO:0005623 cell (level: 2)
25	GO:0007275 development (level: 3)		
26	GO:0044237 cellular metabolism (level: 8)	GO:0003674 molecular_function (level: 1)	GO:0015008 ubiquinol-cytochrome-c reductase complex (sensu Eukaryota) (level: 10)
27	GO:0006810 transport (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0043229 intracellular organelle (level: 4)
28		GO:0004869 cysteine protease inhibitor activity (level: 6)	
29	GO:0050875 cellular physiological process (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0043229 intracellular organelle (level: 4)
30	GO:0008150 biological_process (level: 1)	GO:0003674 molecular_function (level: 1)	GO:0016020 membrane (level: 4)
31	GO:0006913 nucleocytoplasmic transport (level: 6) GO:0035067 negative regulation of histone acetylation (level: 14) GO:0006334 nucleosome assembly (level: 10)	GO:0003674 molecular_function (level: 1)	GO:0043229 intracellular organelle (level: 4)
32	GO:0050875 cellular physiological process (level: 5)		GO:0043229 intracellular organelle (level: 4)
33		GO:0003674 molecular_function (level: 1)	GO:0043229 intracellular organelle (level: 4)

cluster GO	BP	MF	CC
34	GO:0016573 histone acetylation (level: 12)	GO:0003674 molecular_function (level: 1)	GO:0005634 nucleus (level: 5)
35	GO:0007154 cell communication (level: 3)		
36	GO:0050875 cellular physiological process (level: 5)		GO:0030125 clathrin vesicle coat (level: 8)
37	GO:0006355 regulation of transcription ; DNA-dependent (level: 6)	GO:0042623 ATPase activity ; coupled (level: 4)	
38		GO:0008234 cysteine-type peptidase activity (level: 5)	
39	GO:0007253 cytoplasmic sequestering of NF-kappaB (level: 8)	GO:0003674 molecular_function (level: 1)	GO:0005737 cytoplasm (level: 5)
40	GO:0044237 cellular metabolism (level: 8)		GO:0005623 cell (level: 2)
41	GO:0050875 cellular physiological process (level: 5)	GO:0005488 binding (level: 2)	GO:0000228 nuclear chromosome (level: 6)
42	GO:0050875 cellular physiological process (level: 5)	GO:0003924 GTPase activity (level: 8)	GO:0043229 intracellular organelle (level: 4)
43		GO:0017111 nucleoside-triphosphatase activity (level: 7)	
44	GO:0050875 cellular physiological process (level: 5)	GO:0003677 DNA binding (level: 4)	GO:0016021 integral to membrane (level: 7)
45			GO:0043231 intracellular membrane-bound organelle (level: 4)
46			GO:0005764 lysosome (level: 7)
47	GO:0050875 cellular physiological process (level: 5)	GO:0042625 ATPase activity ; coupled to transmembrane movement of ions (level: 7)	

cluster GO	BP	MF	CC
48	GO:0044237 cellular metabolism (level: 8) GO:0007253 cytoplasmic sequestering of NF-kappaB (level: 8)		GO:0043231 intracellular membrane-bound organelle (level: 4)
49			GO:0005737 cytoplasm (level: 5)
50	GO:0007165 signal transduction (level: 5)	GO:0005488 binding (level: 2)	GO:0005623 cell (level: 2)
51			GO:0016020 membrane (level: 4)
52	GO:0044267 cellular protein metabolism (level: 7)		GO:0005623 cell (level: 2)
53	GO:0006917 induction of apoptosis (level: 7)	GO:0003674 molecular_function (level: 1)	
54	GO:0006355 regulation of transcription ; DNA-dependent (level: 6)	GO:0005515 protein binding (level: 4)	
55	GO:0050875 cellular physiological process (level: 5)	GO:0017111 nucleoside-triphosphatase activity (level: 7)	GO:0043229 intracellular organelle (level: 4)
56			GO:0016021 integral to membrane (level: 7)
57	GO:0006355 regulation of transcription ; DNA-dependent (level: 6) GO:0006334 nucleosome assembly (level: 10)	GO:0003677 DNA binding (level: 4)	GO:0005634 nucleus (level: 5)
58	GO:0008285 negative regulation of cell proliferation (level: 7) GO:0006436 tryptophanyl-tRNA aminoacylation (level: 10)		GO:0005737 cytoplasm (level: 5) GO:0005625 soluble fraction (level: 4)
59	GO:0044260 cellular macromolecule metabolism (level: 9)		

cluster GO	BP	MF	CC
60	GO:0000122 negative regulation of transcription from RNA polymerase II promoter (level: 10)	GO:0003674 molecular_function (level: 1)	GO:0043229 intracellular organelle (level: 4)
61	GO:0045449 regulation of transcription (level: 8)	GO:0003674 molecular_function (level: 1)	
62	GO:0008150 biological_process (level: 1)		GO:0016020 membrane (level: 4)
63	GO:0007275 development (level: 3) GO:0008284 positive regulation of cell proliferation (level: 7) GO:0051045 negative regulation of membrane protein ectodomain proteolysis (level: 11)		GO:0005578 extracellular matrix (sensu Metazoa) (level: 3)
64	GO:0007165 signal transduction (level: 5) GO:0045449 regulation of transcription (level: 8)	GO:0003674 molecular_function (level: 1)	
65		GO:0005515 protein binding (level: 4)	
66	GO:0050875 cellular physiological process (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0043231 intracellular membrane-bound organelle (level: 4)
67		GO:0005515 protein binding (level: 4)	GO:0005623 cell (level: 2)
68	GO:0007165 signal transduction (level: 5)		
69	GO:0006810 transport (level: 5)	GO:0042625 ATPase activity ; coupled to transmembrane movement of ions (level: 7)	GO:0005887 integral to plasma membrane (level: 6) GO:0005622 intracellular (level: 4)
70		GO:0003924 GTPase activity (level: 8)	
71	GO:0000018 regulation of DNA recombination (level: 9) GO:0045577 regulation of B-cell differentiation (level: 7)		GO:0016021 integral to membrane (level: 7)

cluster GO	BP	MF	CC
72	GO:0044237 cellular metabolism (level: 8)	GO:0003674 molecular_function (level: 1)	GO:0043229 intracellular organelle (level: 4)
73	GO:0050875 cellular physiological process (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0016020 membrane (level: 4)
74	GO:0007399 neurogenesis (level: 7) GO:0042981 regulation of apoptosis (level: 9) GO:0008635 caspase activation via cytochrome c (level: 9)	GO:0005515 protein binding (level: 4) GO:0000166 nucleotide binding (level: 5) GO:0008656 caspase activator activity (level: 4)	GO:0005622 intracellular (level: 4)
75	GO:0009987 cellular process (level: 2)		GO:0043231 intracellular membrane-bound organelle (level: 4)
76	GO:0007275 development (level: 3) GO:0044237 cellular metabolism (level: 8)	GO:0003674 molecular_function (level: 1)	
77	GO:0006418 tRNA aminoacylation for protein translation (level: 9)		GO:0005622 intracellular (level: 4)
78	GO:0044267 cellular protein metabolism (level: 7)	GO:0003674 molecular_function (level: 1)	GO:0043229 intracellular organelle (level: 4)
79	GO:0050875 cellular physiological process (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0005737 cytoplasm (level: 5)
80	GO:0007165 signal transduction (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0005623 cell (level: 2)
81	GO:0044237 cellular metabolism (level: 8)	GO:0003674 molecular_function (level: 1)	GO:0005623 cell (level: 2)
82		GO:0003674 molecular_function (level: 1)	GO:0005634 nucleus (level: 5)
83		GO:0005524 ATP binding (level: 6) GO:0008026 ATP-dependent helicase activity (level: 4) GO:0003723 RNA binding (level: 4)	

cluster GO	BP	MF	CC
84	GO:0006810 transport (level: 5)	GO:0003674 molecular_function (level: 1)	
85	GO:0044237 cellular metabolism (level: 8)	GO:0005488 binding (level: 2)	GO:0005634 nucleus (level: 5)
86	GO:0050875 cellular physiological process (level: 5)	GO:0005515 protein binding (level: 4)	
87		GO:0003674 molecular_function (level: 1)	GO:0016021 integral to membrane (level: 7)
88	GO:0044237 cellular metabolism (level: 8)		GO:0043229 intracellular organelle (level: 4)
89	GO:0006355 regulation of transcription ; DNA-dependent (level: 6)		GO:0043231 intracellular membrane-bound organelle (level: 4)
90	GO:0006260 DNA replication (level: 8)	GO:0003677 DNA binding (level: 4) GO:0008094 DNA-dependent ATPase activity (level: 5) GO:0005524 ATP binding (level: 6)	GO:0005634 nucleus (level: 5)
91		GO:0003674 molecular_function (level: 1)	GO:0005737 cytoplasm (level: 5)
92	GO:0007165 signal transduction (level: 5)	GO:0003723 RNA binding (level: 4)	
93	GO:0044237 cellular metabolism (level: 8)	GO:0003676 nucleic acid binding (level: 3)	GO:0005634 nucleus (level: 5)
94	GO:0006917 induction of apoptosis (level: 7) GO:0045449 regulation of transcription (level: 8)	GO:0003674 molecular_function (level: 1)	GO:0005634 nucleus (level: 5)
95	GO:0044237 cellular metabolism (level: 8)	GO:0003674 molecular_function (level: 1)	GO:0005737 cytoplasm (level: 5) GO:0005634 nucleus (level: 5)

cluster GO	BP	MF	CC
96	GO:0006355 regulation of transcription ; DNA-dependent (level: 6)	GO:0003676 nucleic acid binding (level: 3)	GO:0005634 nucleus (level: 5)
97	GO:0019219 regulation of nucleobase ; nucleoside nucleotide and nucleic acid metabolism (level: 8)		GO:0043229 intracellular organelle (level: 4)
98	GO:0050875 cellular physiological process (level: 5)		GO:0005737 cytoplasm (level: 5)
99	GO:0008152 metabolism (level: 4)	GO:0005515 protein binding (level: 4) GO:0008415 acyltransferase activity (level: 6)	GO:0005947 alpha-ketoglutarate dehydrogenase complex (sensu Eukaryota) (level: 8)
100	GO:0007253 cytoplasmic sequestering of NF-kappaB (level: 8)		GO:0005622 intracellular (level: 4)
101	GO:0044260 cellular macromolecule metabolism (level: 9) GO:0016481 negative regulation of transcription (level: 8)	GO:0003677 DNA binding (level: 4)	GO:0005634 nucleus (level: 5)
102	GO:0050875 cellular physiological process (level: 5)	GO:0003676 nucleic acid binding (level: 3)	
103	GO:0045449 regulation of transcription (level: 8)	GO:0003674 molecular_function (level: 1)	GO:0043229 intracellular organelle (level: 4)
104	GO:0006355 regulation of transcription ; DNA-dependent (level: 6)		
105	GO:0044237 cellular metabolism (level: 8)		GO:0005634 nucleus (level: 5)
106	GO:0006810 transport (level: 5)	GO:0005515 protein binding (level: 4)	
107	GO:0006355 regulation of transcription ; DNA-dependent (level: 6)		GO:0005634 nucleus (level: 5)

cluster GO	BP	MF	CC
108	GO:0006355 regulation of transcription ; DNA-dependent (level: 6)	GO:0005488 binding (level: 2)	GO:0005634 nucleus (level: 5)
109	GO:0006355 regulation of transcription ; DNA-dependent (level: 6)	GO:0003674 molecular_function (level: 1)	GO:0005634 nucleus (level: 5)
110		GO:0004871 signal transducer activity (level: 5)	GO:0043229 intracellular organelle (level: 4)
111	GO:0045449 regulation of transcription (level: 8)		GO:0043229 intracellular organelle (level: 4)
112		GO:0003723 RNA binding (level: 4)	
113	GO:0019219 regulation of nucleobase ; nucleoside ; nucleotide and nucleic acid metabolism (level: 8)		
114	GO:0006355 regulation of transcription ; DNA-dependent (level: 6)	GO:0003677 DNA binding (level: 4)	GO:0005634 nucleus (level: 5)
115	GO:0045449 regulation of transcription (level: 8)		
116	GO:0044267 cellular protein metabolism (level: 7)	GO:0003674 molecular_function (level: 1)	GO:0005623 cell (level: 2)
117	GO:0045449 regulation of transcription (level: 8) GO:0044267 cellular protein metabolism (level: 7)	GO:0003677 DNA binding (level: 4)	GO:0005634 nucleus (level: 5)
118	GO:0006355 regulation of transcription ; DNA-dependent (level: 6)	GO:0005488 binding (level: 2)	
119	GO:0050875 cellular physiological process (level: 5)	GO:0017111 nucleoside-triphosphatase activity (level: 7)	

cluster GO	BP	MF	CC
120	GO:0008150 biological_process (level: 1)		GO:0043231 intracellular membrane-bound organelle (level: 4)
121	GO:0050875 cellular physiological process (level: 5)	GO:0005515 protein binding (level: 4)	GO:0005623 cell (level: 2)
122	GO:0008150 biological_process (level: 1)	GO:0003674 molecular_function (level: 1)	GO:0016021 integral to membrane (level: 7)
123	GO:0006810 transport (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0005623 cell (level: 2)
124	GO:0000122 negative regulation of transcription from RNA polymerase II promoter (level: 10)	GO:0005488 binding (level: 2)	GO:0005634 nucleus (level: 5) GO:0005737 cytoplasm (level: 5)
125	GO:0044267 cellular protein metabolism (level: 7) GO:0006917 induction of apoptosis (level: 7)	GO:0004871 signal transducer activity (level: 5)	
126	GO:0044237 cellular metabolism (level: 8) GO:0006917 induction of apoptosis (level: 7)	GO:0003674 molecular_function (level: 1)	
127			GO:0005634 nucleus (level: 5)
128	GO:0050875 cellular physiological process (level: 5)		GO:0016020 membrane (level: 4)
129	GO:0006810 transport (level: 5)		
130	GO:0008285 negative regulation of cell proliferation (level: 7) GO:0030838 positive regulation of actin filament polymerization (level: 12) GO:0042110 T-cell activation (level: 11)		GO:0005623 cell (level: 2)
131	GO:0044267 cellular protein metabolism (level: 7)		GO:0043229 intracellular organelle (level: 4)

cluster GO	BP	MF	CC
132	GO:0009987 cellular process (level: 2)		GO:0005634 nucleus (level: 5)
133	GO:0050875 cellular physiological process (level: 5)		GO:0043231 intracellular membrane-bound organelle (level: 4)
134	GO:0007275 development (level: 3) GO:0050875 cellular physiological process (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0005634 nucleus (level: 5)
135	GO:0007275 development (level: 3)		GO:0005737 cytoplasm (level: 5)
136	GO:0045449 regulation of transcription (level: 8)	GO:0004871 signal transducer activity (level: 5)	
137	GO:0050875 cellular physiological process (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0005634 nucleus (level: 5)
138	GO:0008150 biological_process (level: 1)	GO:0005515 protein binding (level: 4)	GO:0005623 cell (level: 2)
139		GO:0017017 MAP kinase phosphatase activity (level: 9)	
140	GO:0044267 cellular protein metabolism (level: 7)	GO:0003674 molecular_function (level: 1)	
141	GO:0016481 negative regulation of transcription (level: 8)	GO:0003674 molecular_function (level: 1)	GO:0005634 nucleus (level: 5)
142	GO:0006355 regulation of transcription ; DNA-dependent (level: 6) GO:0044267 cellular protein metabolism (level: 7)	GO:0003674 molecular_function (level: 1)	
143	GO:0007154 cell communication (level: 3)		GO:0043229 intracellular organelle (level: 4)

cluster GO	BP	MF	CC
144	GO:0044267 cellular protein metabolism (level: 7)	GO:0003674 molecular_function (level: 1)	GO:0005622 intracellular (level: 4)
145	GO:0006917 induction of apoptosis (level: 7)	GO:0004871 signal transducer activity (level: 5)	
146	GO:0050875 cellular physiological process (level: 5)	GO:0042625 ATPase activity ; coupled to transmembrane movement of ions (level: 7)	GO:0043229 intracellular organelle (level: 4)
147	GO:0044237 cellular metabolism (level: 8) GO:0007154 cell communication (level: 3)		GO:0043229 intracellular organelle (level: 4)
148	GO:0009987 cellular process (level: 2)	GO:0003723 RNA binding (level: 4)	
149		GO:0005509 calcium ion binding (level: 5)	GO:0005623 cell (level: 2)
150	GO:0006810 transport (level: 5)		GO:0005623 cell (level: 2)
151	GO:0050875 cellular physiological process (level: 5)	GO:0003674 molecular_function (level: 1)	GO:0005764 lysosome (level: 7)
152	GO:0050875 cellular physiological process (level: 5)	GO:0003677 DNA binding (level: 4)	GO:0016020 membrane (level: 4)
153	GO:0042110 T-cell activation (level: 11) GO:0050875 cellular physiological process (level: 5)		GO:0005623 cell (level: 2)
154		GO:0005515 protein binding (level: 4)	GO:0043229 intracellular organelle (level: 4)
155	GO:0006917 induction of apoptosis (level: 7)	GO:0003674 molecular_function (level: 1)	GO:0043229 intracellular organelle (level: 4)
156		GO:0003674 molecular_function (level: 1)	GO:0016459 myosin (level: 7)

Bibliografía

[AAA+05] C. Alfarano, C.E. Andrade, K. Anthony, N. Bahroos, M. Bajec, K. Bantoft, D. Betel, B B. obechko, K. Boutilier, E. Burgess, K. Buzadzija, R. Cavero, C. D'Abreo, I. Donaldson, Dorair D. ajoo, M.J. Dumontier, M.R. Dumontier, V. Earles, R. Farrall, H. Feldman, E. Garderman, Gon Y. g, R. Gonzaga, V. Grytsan, E. Gryz, V. Gu, E. Haldorsen, A. Halupa, R. Haw, Hrvojic A., L. Hurrell, R. Isserlin, F. Jack, F. Juma, A. Khan, T. Kon, S. Konopinsky, V. Le, Lee E., S. Ling, M. Magidin, J. Moniakis, J. Montojo, S. Moore, B. Muskat, I.Ñg, Paraiso J.P., B. Parker, G. Pintilie, R. Pirone, J.J. Salama, S. Sgro, T. Shan, Y. Shu, J. Siew, Sk D. inner, K. Snyder, R. Stasiuk, D. Strumpf, B. Tuekam, S. Tao, Z. Wang, M. White, Willis R., C. Wolting, S. Wong, A. Wrong, C. Xin, R. Yao, B. Yates, S. Zhang, K. Zheng, Pawson T., B.F.F. Ouellette, and C.W.V. Hogue. The Biomolecular Interaction Network Database and related tools 2005 update. *Nucl. Acids Res.*, 33(suppl_1):D418–424, 2005.

[AHU82] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *Data Structures and Algorithms.* Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley, 1982.

[AIS93] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 1993.

[AJL+03] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts. and P. Walter. *Biología molecular de la célula. Cuarta Edición.* Omega, 2003.

[AK88] M. Amadasun and R. A. King. Low-level segmentation of multispectral images via agglomerative clustering of uniform neighbourhoods. *Pattern Recognition*, 21(3):261–268, 1988.

[AMI] AmiGO. http://www.godatabase.org/cgi-bin/amigo/go.cgi.

[BFM97] T. Back, D. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation.* IOP Publishing Ltd., Bristol, UK, 1997.

[BKML+03] D.A. Benson, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, and D.L. Wheeler. GenBank. *Nucleics Acids Research*, 31(1):23–27, 2003.

[BPKF98] W. Banzhaf, P. Nordin, R. Keller, and F. Francone. *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications.* Morgan Kaufmann, January 1998.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms.* MIT Press, Cambridge, Massachusetts, USA, 2001.

[Con00] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 24:25–29, 2000.

[CVL02] C. Coello, D. Van Veldhuizen, and G. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Genetic Algorithms and Evolutionary Computation. Klumer, 2002.

[CXR+05] S. E. Calvano, W. Xiao, D. R. Richards, R. M. Feliciano, H. V. Baker, R. J. Cho, R. O. Chen, B. H. Brownstein, J. P. Cobb, S. K. Tschoeke, C. Miller-Graziano, L. L. Moldawer, M.Ñ. Mindrinos, R. W. Davis, R. G. Tompkins, and S. F. Lowry. The inflammation and host response to injury large scale collaborative research program. a network-based analysis of systemic inflammation in humans. *Nature*, in press, 2005.

[DAPM00] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[Dar59] C. Darwin. *On The Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life.* John Murray, London, 1859.

[Deb01] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Inc., 2001. ISBN 047187339X.

[Dra03] S. Draghici. *Data Analysis Tools for DNA Microarrays.* Chapman and Hall/CRC, 2003.

[EZ05] C. Rubio Escudero and I. Zwir. Characterizing microarray analysis methods by reverse gene mapping gene expression profiles. *en preparación*, 2005.

[FASD04] R. Díaz Uriarte F. Al-Shahrour and J. Dopazo. Fatigo: a web tool for finding significant associations of gene ontology terms with groups of genes. *Bioinformatics*, 20:578–580, 2004.

[FAT] FatiGO. http://www.fatigo.org.

[Gol89] D. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley, 1989. ISBN 0-201-15767-5.

[GR87] D. Goldberg and J.J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings Second International Conference on Genetic Algorithm*, pages 41–49, 1987.

[HK00] J. Han and M. Kamber. *Data Mining: Concepts and Techniques.* The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2000.

[Jac12] P. Jaccard. The distribution of flora in the alpine zone. *The New Phytologist*, 11(2):37–50, 1912.

[JCH01] I. Jonyer, D. J. Cook, and L. B. Holder. Discovery and evaluation of graph-based hierarchical conceptual clusters. *Machine Learning Research*, 2:19–43, 2001.

[KAA+05] C. Kanz, P. Aldebert, N. Althorpe, W. Baker, A. Baldwin, K. Bates, P. Browne, A. van den Broek, M. Castro, G. Cochrane, K. Duggan, R. Eberhardt, N. Faruque, J. Gamble, F. García-Diez, N. Harte, T. Kulikova, Q. Lin, V. Lombard, R. Lopez, R. Mancuso, M. McHale, F.Ñardone, V. Silventoinen, S. Sobhany, P. Stoehr, M. Tuli, K. Tzouvara, R. Vaughan, D. Wu, W. Zhu, and R. Apweiler. The EMBL Nucleotide Sequence Database. *Nucleics Acids Research*, 33(suppl_1):D29–33, 2005.

[Koz92] J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, USA, 1992. ISBN 0-262-11170-5.

[LB95] C. Li and G. Biswas. Knowledge-based scientific discovery in geological databases. In *Proceedings of the First International Conference on Knowledge Discovery and DataMining*, pages 204–209, Montreal, Canada, August 20-21 1995.

[Lew01] B. Lewin. *Genes VII.* Marbán, 2001.

[MF00] Z. Michalewicz and D.B. Fogel. *How to solve it: modern heuristics.* Springer-Verlag New York, Inc., New York, NY, USA, 2000.

[Mit97] T. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

[Osm95] I. Osman. An introduction to meta-heuristics. *Operational Research Tutorial Papers Series, Annal Conference OR37 - Canterbury*, 1995.

[Ras92] E. Rasmussen. *Clustering algorithms.* Information Retrieval: Data Structures and Algorithms. W. B. Frakes and R. Baeza-Yates,Eds. Prentice-Hall, Inc., Upper Saddle Eds. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992. 419-442 pp.

[RR84] Michalski R. and Stepp R. *Learning from observation: Conceptual clustering In R. S. Michalski, J. G. Carbonell and T. M. Mitchel, editors.* Machine Learning: An Artificial Intelligence Approach. Springer, Berling, Heidelberg, 1984. 331-363 pp.

[RRF+89] R. P. Richardson, C. D. Rhyne, Y. Fong, D. G. Hesse, K. J. Tracey, M. A. Marano, S. F. Lowry, A. C. Antonacci, and S. E. Calvano. Peripheral blood leukocyte kinetics following in vivo lipopolysaccharide (lps) administration to normal human subjects. influence of elicited hormones and cytokines. *Ann.Surg.*, 210(2):239–245, 1989.

[RZCRE+] R. Romero-Zaliz, O. Cordón, C. Rubio-Escudero, I. Zwir, and J.P. Cobb. A multi-objective evolutionary conceptual clustering methodology for gene annotation from networking databases. Sometido.

[SD94] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[THC+99] S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.

[WF99] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, 1999.

[ZDT00] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical results. *Evolutionary Computation*, 8(2): 173–195, 2000.

[ZHG05] I. Zwir, H. Huang, and F.A. Groisman. Analysis of differentially-regulated genes within a regulatory network by gps genome navigation. *Bioinformatics*, 21:4073–4083, 2005.

[ZSK+05] I. Zwir, D. Shin, A. Kato, K.Ñishino. T. Latifi. F. Solomon, J. Hare, H. Huang, and E. Goisman. Dissecting the phop regulatory network of escherichia coli and salmonella enterica. In *PNAS*, USA, March 2005.

[ZT99] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.

[ZZR04] R. Romero Zaliz, I. Zwir, and E. Ruspini. *Applications of Multi-Objective Evolutionary Algorithms*, chapter Generalized Analysis of Promoters (GAP): A method for DNA sequence description. World Scientific, 2004. ISBN 981-256-106-4.