UNIVERSIDAD NACIONAL DEL SUR

TESIS DE DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

Desarrollo de Técnicas de Computación Evolutiva Multiobjetivo y Aprendizaje Automático para la Inferencia, Modelado y Simulación de Redes Regulatorias

Cristian Andrés Gallo

BAHÍA BLANCA ARGENTINA

2014

PREFACIO

Esta Tesis se presenta como parte de los requisitos para optar al grado Académico de Doctor en Ciencias de la Computación, de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otra. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el ámbito del Departamento de Ciencias e Ingeniería de la Computación durante el período comprendido entre el 22 de Septiembre de 2009 y el 02 de Diciembre de 2013, bajo la dirección del Dr. Ignacio Ponzoni, profesor del Departamento de Ciencias e Ingeniería de la Computación e investigador adjunto del CONICET, y de la Dra. Jessica A. Carballido, profesora del Departamento de Ciencias e Ingeniería de la Computación e investigadora asistente del CONICET.

Cristian Andrés Gallo
Bahía Blanca, 2 Diciembre de 2013

Universidad Nacional del Sur
Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el …/…/……., mereciendo la calificación de …… (……...)

Agradecimientos

Esta Tesis fue posible gracias al apoyo y aliento de muchas personas. Deseo expresar en primer término mi más profundo agradecimiento a Ignacio y a Jessica, quienes me iniciaron en el mundo de la investigación, me ayudaron con paciencia a caminar los primeros pasos y me acompañaron con su valiosísima amistad e inmensa sabiduría durante todo el recorrido de este sendero.

A Flavia, quien me acompaño con su amor incondicional y comprension durante los últimos años. Gracias por la paciencia y el apoyo cuando necesité de tiempo para este trabajo.

Al Consejo Nacional de Investigaciones Científicas y Técnicas y al Departamento de Ciencias e Ingeniería de la Computación, mi agradecimiento por haberme facilitado los medios para realizar este trabajo.

Quiero expresar mi mayor gratitud a mis padres, mis hermanos y mis amigos, quienes con su afecto y constante aliento hicieron posible que concretara esta tesis. A ellos mi mayor agradecimiento por el amor y apoyo brindados.

Y por último, a todas las personas que, durante estos años, de alguna u otra manera me acompañaron y ayudaron, me aportaron ideas y con quienes compartí horas de trabajo y desinteresadamente me brindaron su conocimiento en áreas ajenas a mi especialidad.

Resumen

Durante las últimas décadas el desarrollo de la bioinformática nos ha permitido lograr una mayor comprensión de los procesos biológicos que ocurren con nuestras células a nivel molecular. Al respecto, las mejoras e innovaciones en la tecnología continúan estimulando la mejora en la calidad de los datos biológicos que pueden ser obtenidos a nivel genómico. En tal sentido, grandes volúmenes de información pueden ser encontrados en formas de anotaciones o bases de datos computacionales. Estos conjuntos de datos, apropiadamente combinados, tienen el potencial de posibilitar descubrimientos novedosos que lleven a avances en campos tan relevantes para el desarrollo nacional como son la biotecnología o la medicina post-genómica.

En particular, esta tesis se centra en la investigación de técnicas de aprendizaje automático y computación evolutiva para la inferencia de redes regulatorias de genes a partir de datos de expresión de genes, a nivel de genomas completos. Una red regulatoria de genes es una colección de segmentos de ADN (ácido desoxirribonucleico) en una célula que interactúan unos con otros (indirectamente a través del producto de su expresión) y con otras sustancias en la célula, gobernando así las tasas de transcripción de los genes de la red en ARNm (ácido ribonucleico mensajero).

La principal contribución de esta tesis esta relacionada con el desarrollo de metodologías computacionales que asistan, a expertos en bioinformática, en la ingeniería inversa de las redes regulatorias de genes. En tal sentido, se desarrollaron algoritmos de computación evolutiva que permiten la identificación de grupos de genes co-expresados bajo ciertos subconjuntos de condiciones experimentales. Estos algoritmos se aplican sobre datos de expresión de genes, y optimizan características deseables desde el punto de vista biológico, posibilitando la obtención de relaciones de co-expresión relevantes. Tales algoritmos fueron cuidadosamente validados por medio de comparaciones con otras técnicas similares disponibles en la literatura, realizando estudios con datos reales y sintéticos a fin de mostrar la utilidad de la información extraída. Además, se desarrolló un algoritmo de inferencia que permite la extracción de potenciales relaciones causa-efecto entre genes, tanto simultáneas como también aquellas diferidas en el tiempo. Este algoritmo es una evolución de una técnica presentada con anterioridad, e incorpora características novedosas como la posibilidad de inferir reglas con múltiples retardos en el tiempo, a nivel genoma completo, e integrando múltiples conjuntos de datos. La técnica se validó mostrando su eficacia respecto de otros enfoques relevantes de la literatura. También se estudiaron los resultados obtenidos a partir de conjuntos de datos reales en términos de su relevancia biológica, exponiendo la viabilidad de la información inferida. Finalmente, estos algoritmos se integraron en una plataforma de software que facilita la utilización de estas técnicas permitiendo la inferencia, manipulación y visualización de redes regulatorias de genes.

Abstract

In recent decades, the development of bioinformatics has allowed us to achieve a greater understanding of the biological processes that occur at the molecular level in our cells. In this regard, the improvements and innovations in technology continue to boost the improvement in the quality of the biological data that can be obtained at the genomic level. In this regard, large volumes of information can be found in forms of ontology's or computer databases. These datasets, appropriately combined, have the potential to enable novel discoveries that lead to progress in relevant fields to national development such as biotechnology and post-genomic medicine.

In particular, this thesis focuses on the research of machine learning techniques and evolutionary computation for the inference of gene regulatory networks from gene expression data at genome-wide levels. A gene regulatory network is a collection of segments of DNA (deoxyribonucleic acid) in a cell which interact with each other (indirectly through their products of expression) and with other substances in the cell, thereby governing the rates of network genes transcription into mRNA (messenger ribonucleic acid).

The main contribution of this thesis is related to the development of computational methodologies to attend experts in bioinformatics in the reverse engineering of gene regulatory networks. In this sense, evolutionary algorithms that allow the identification of groups of co-expressed genes under certain subsets of experimental conditions were developed. These algorithms are applied to gene expression data, and optimize desirable characteristics from the biological point of view, allowing the inference of relevant co-expression relationships. Such algorithms were carefully validated by the comparison with other similar techniques available in the literature, conducting studies with real and synthetic data in order to show the usefulness of the information extracted. Furthermore, an inference algorithm that allows the extraction of potential cause-effect relationships between genes, both simultaneous and time-delayed, were developed. This algorithm is an evolution of a previous approach, and incorporates new features such as the ability to infer rules with multiple time delays, at genome-wide level, and integrating multiple datasets. The technique was validated by showing its effectiveness over other relevant approaches in the literature. The results obtained from real datasets were also studied in terms of their biological relevance by exposing the viability of the inferred information. Finally, these algorithms were integrated into a software platform that facilitates the use of these techniques allowing the inference, manipulation and visualization of gene regulatory networks.

Lista de Publicaciones

En Revistas Cientificas y en la serie Lecture Notes in Computer Science:

- **Gallo, C.A., Carballido, J.A., Ponzoni, I.** "Discovering Time-Lagged Rules from Microarray Data using Gene Profile Classifiers", ***BMC Bioinformatics.*** Vol. 12, No. 123 (**2011**). BioMedCentral. ISSN 1471-2105.
- **Gallo, C.A., Dussaut, J., Carballido, J.A., Ponzoni, I.** "BAT: A new Biclustering Analysis Toolbox", In: Ferreira, C.E.; Miyano, S.; Stadler, P.F. (Eds.): Advances in Bioinformatics and Computational Biology, 5th Brazilian Symposium on Bioinformatics, BSB 2010, Buzios, Rio de Janiero, Brazil, August 30-September 3, 2010, Proceedings. ***Lecture Notes in Computer Science,*** Vol. 6268, pp. 67-71. Springer-Verlag Berlin Heidelberg, (**2010**). ISSN: 0302-9743.
- **Gallo, C.A., Carballido, J.A., Ponzoni, I.** "Microarray Biclustering: A Novel Memetic Approach based on the PISA Platform", In: Pizzuti, C.; Ritchie, M.D.; Giacobini, M. (Eds.): Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, 7th European Conference, EvoBIO 2009, Tübingen, Germany, April 15-17, Proceedings. ***Lecture Notes in Computer Science***, Vol. 5483, pp. 44–55. Springer-Verlag Berlin Heidelberg, (**2009**). ISSN: 0302-9743.
- **Gallo, C.A., Carballido, J.A., Ponzoni, I.** "BiHEA: A Hybrid Evolutionary Approach for Microarray Biclustering", In: Guimarães, K.S.; Panchenko, A.; Przytycka, T.M. (Eds.): Advances in Bioinformatics and Computational Biology, 4th Brazilian Symposium on Bioinformatics, BSB 2009, Porto Alegre, Brazil, July 29-31, 2009, Proceedings. ***Lecture Notes in Computer Science***, Vol. 5676, pp. 36–47, 2009. Springer-Verlag Berlin Heidelberg, (**2009**). ISSN: 0302-9743.

En Capítulos de Libros con referato:

- **Gallo, C.A., Carballido, J.A., Ponzoni, I.** "Inference of Gene Regulatory Networks based on Association Rules", In: ELLOUMI, M; ZOMAYA, A.Y. (Eds.): Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data. Wiley (**2013**). ISBN: 978-1118132739.

En Actas de Congresos y Reuniones Científicas con comité de revisión:

- **Gallo, C.A., Carballido, J.A., Ponzoni, I.** "Inferring Time-Lagged Association Rules from Microarray Time-Series Data", ***SolBio 2010*** *(1st International Conference on Bioinformatics)*, 26 al 28 de Septiembre de 2010, Termas de Chillan, Chile.
- **Gallo, C.A., Carballido, J.A., Ponzoni, I.,** "Using multi-objective evolutionary computing for biclustering of gene expression data", trabajo completo aceptado en el ***VI ALIO/EURO Workshop on Applied Combinatorial Optimization***, 15 al 17 de Diciembre de 2008, Buenos Aires, Argentina.
- **Gallo, C.A., Carballido, J.A., Ponzoni, I.,** "A Hybridized Multiobjective Evolutionary Approach for Microarray Biclustering", trabajo completo aceptado en la ***CLEI 2008 (XXXIV Conferencia Latinoamérica de Informática),*** 8 al 12 de Septiembre de 2008, Santa Fe, Argentina.
- **Gallo, C.A., Carballido, J.A., Ponzoni, I.** "GeRNet: A Framework for Inference, Visualization and Manipulation of Gene Regulatory Networks based on Association Rules", ***CAB2C 2013 (VI Congreso Argentino de Bioinformática y Biología Computacional)***, 29 al 31 de Octubre de 2013, Rosario, Argentina.
- **Gallo, C.A., Carballido, J.A., Ponzoni, I.,** "Inferring Time-Lagged Association Rules from Microarray Time-Series Data", ***CAB2C 2011 (II Congreso Argentino de Bioinformática y Biología Computacional)***, 11-13 de mayo de 2011, Córdoba, Argentina.
- **Gallo, C.A., Dussaut, J.S., Carballido, J.A., Ponzoni, I.,** "A Microarray Biclustering Analysis Tool based on the BiHEA Algorithm", ***CAB2C 2010 (I Congreso Argentino de Bioinformática y Biología Computacional)***, 12-14 de mayo de 2010, Quilmes, Argentina.
- **Gallo, C.A., Maguitman, A.G., Carballido, J.A., Ponzoni, I.,** "Biclustering in Data Mining using a Memetic Multi-Objective Evolutionary Algorithm", trabajo completo aceptado en el ***CACIC '2008 (XIV Congreso Argentino de Ciencias de la Computación)***, 6 al 10 de Octubre de 2008, Chilecito, La Rioja, Argentina.

Índice General

Capítulo 1

Capítulo 2

Capítulo 3

Capítulo 4

Capítulo 5

Capítulo 6

Capítulo 7

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

1.1. Inferencia de Redes Regulatorias de Genes

El genoma codifica miles de genes cuyos productos permiten el desarrollo y el funcionamiento celular. La cantidad y los patrones temporales en los que estos productos aparecen en la célula son cruciales para los procesos de la vida. En este contexto, el advenimiento de nuevas tecnologías en el área de genómica, tales como los *microarrays*, han posibilitado la obtención de grandes volúmenes de información relativas al funcionamiento molecular de seres vivos (Sohler, 2006). Esto ha derivado en una verdadera revolución del conocimiento en el campo de la Biología Molecular, la cual ha dado origen a una nueva disciplina, conocida como Bioinformática. En la actualidad, gran parte de los esfuerzos llevados adelante en esta área apuntan a generar modelos computacionales que permitan extraer conocimiento a partir de la gran cantidad de datos biológicos disponibles actualmente (Schlitt y Brazma, 2007). Esto involucra un proceso de ingeniería inversa a fin de dilucidar las intricadas interacciones de los mecanismos biológicos. En este contexto, el uso de técnicas de aprendizaje automático (Bishop, 2006) y computación evolutiva (De Jong, 2006) está teniendo gran impacto en el diseño de modelos predictivos que faciliten el descubrimiento automatizado de nuevo conocimiento biológico (Pal *et al.*, 2006; Cios *et al.*, 2007; Handl *et al.*, 2007).

Uno de los principales problemas en Bioinformática consiste en el desarrollo de técnicas para la inferencia de redes regulatorias de genes (RRG), tema de vital y actual relevancia en el campo de la genómica funcional (Schlitt y Brazma, 2007). Una RRG es una colección de especies moleculares y sus interacciones, que conjuntamente controlan la abundancia de los productos de los genes. Numerosos procesos celulares están afectados por las redes regulatorias. Descubrir y estudiar la estructura de las RRGs de diferentes organismos resulta de fundamental importancia para avanzar en la comprensión del funcionamiento biológico de los seres vivos, y posee innumerables aplicaciones tanto en medicina (por ejemplo, en el desarrollo de nuevos medicamentos (Huang, 2002) o, en el descubrimiento de los mecanismos de enfermedades relacionadas con la disfunción del proceso regulatorio (Karlebach y Shamir, 2008)), como en biotecnología (por ejemplo, en el diseño de nuevas variantes de cultivos (Yamaguchi-Shinozaki y Shinozaki, 2006)).

Varios modelos computacionales han sido desarrollados para el análisis de las RRGs. Estos se pueden categorizar en tres clases. La primera, los modelos cualitativos, describen las redes

regulatorias en forma cualitativa, tal cual su nombre lo indica. Permiten al usuario obtener un entendimiento básico de las diferentes funcionalidades de una dada red bajo diferentes condiciones. Su naturaleza cualitativa hace que sean flexibles y fáciles de encajar en el fenómeno biológico, aunque solo pueden responden preguntas cualitativas. Para entender y manipular comportamientos que dependen de sincronizaciones finas y concentraciones moleculares exactas, se desarrolló una segunda clase de modelos, los modelos continuos. Por ejemplo, para simular los efectos de una restricción dietaria en células de levadura bajo diferentes concentraciones de nutrientes (Weindruch y Walford, 1988), el usuario debe recurrir a la resolución mas fina de los modelos continuos. Una tercera clase de modelos fue introducida siguiendo la observación de que la funcionalidad de regulación de la red se ve muchas veces afectada por el ruido. Dado que la mayoría de estos modelos tienen en cuenta interacciones individuales entre moléculas, son llamados modelos a nivel molecular. Estos modelos explican la relación entre la estocasticidad y la regulación de genes.

Los modelos cualitativos de RRG son decididamente atractivos debido a la complejidad en la dinámica molecular de las redes. Es mas, la mayoría de las redes de genes son difíciles de mapear precisamente por cualquier modelo matemático parsimonioso (Li *et al.*, 2006). En este contexto, los enfoques de minería de datos ofrecen una forma de identificar mecanismos regulatorios en forma cualitativa directamente de los datos de entrada, permitiendo la inferencia de modelos de RRG correspondientes a esta clase. En tal sentido, durante el último lustro surgieron diferentes métodos para efectuar la ingeniería inversa de RRGs cualitativas empleando técnicas de minería de datos (Schlitt y Brazma, 2007). Estos métodos utilizan datos de expresión genética, los cuales constituyen una medida de la abundancia de ARNm de un subconjunto (o totalidad) de los genes presentes en el genoma de un organismo. Dichas mediciones son usualmente obtenidas mediante experimentos en laboratorios con *microarrays*, los cuales permiten recolectar datos de expresión génica a gran escala. Sobre la base de esta información, los métodos de inteligencia computacional propuestos para reconstruir RRGs infieren potenciales asociaciones de regulación entre genes, mediante la búsqueda sistemática de correlaciones presentes en los datos. Este proceso de inferencia puede requerir efectuar primero una discretización de los datos, representando la evolución del valor de expresión de cada gen en términos de un conjunto finito de estados. La idea básica detrás de esta metodología es que los diferentes estados de un gene constituyen una medida de su grado de actividad dentro de cada muestra. De este modo, la identificación de un patrón de comportamiento similar (o contrapuesto) entre diferentes genes puede estar indicando la presencia de un potencial vinculo de regulación entre los mismos. Las asociaciones identificadas mediante estos métodos, pueden

luego ser expresadas como potenciales reglas de regulación, las cuales permitirán la posterior reconstrucción de la RRG estudiada.

Las técnicas basadas en reglas ofrecen varias ventajas cuando se realiza análisis dirigido por los datos. En primer lugar, la información inferida es suficiente para dilucidar la estructura relacional presente en las RRGs. Además son técnicas de modelo libre altamente abstractas, por lo que requieren la menor cantidad de datos en relación a otras mas complejas, con una importante habilidad para realizar inferencias (Karlebach y Shamir, 2008). Es mas, la simplicidad de estos enfoques permite la inferencia de modelos de gran tamaño con una alta velocidad de análisis.

Las técnicas de *clustering* fueron una de las primeras estrategias empleadas para analizar datos de expresión de genes (Spellman *et al.*, 1998; Jiang *et al.*, 2004; Madeira y Oliveira, 2004). Estos enfoques aproximan las redes regulatorias identificando grupos de genes co-expresados a lo largo de todas las condiciones experimentales. Una generalización de ese enfoque son los llamados *biclusters*, que identifican grupos de genes co-expresados sobre un subconjunto de las condiciones experimentales. Sin embargo, mediante el empleo de técnicas de *clustering* (*biclustering*) se asume que la co-expresión es equivalente a la regulación, lo cual implica una relación simétrica entre los genes que no siempre corresponde al fenómeno biológico (Ponzoni *et al.*, 2007). En consecuencia, estas técnicas solo permiten inferir un subconjunto de los tipos de interacciones biológicamente posibles entre genes.

Las redes booleanas fueron otras de las primeras técnicas empleadas para el análisis de datos de *microarrays* (Liang *et al.*, 1998; Akutsu *et al.*, 1999; Mehra *et al.*, 2004). Estos algoritmos utilizan variables booleanas para expresar el estado de un gen, activo (1) o inactivo (0), mientras que las interacciones son modeladas mediante funciones booleanas que calculan el estado de un gen a partir de la activación de otros genes. La sencillez de este enfoque constituye su principal ventaja pero también su mayor limitación. Las redes booleanas permiten analizar en forma eficiente redes regulatorias de gran dimensión, dado que se realizan fuertes suposiciones que simplifican la estructura y dinámica de los sistemas regulatorios genéticos. Por ejemplo, se asume que las transiciones entre estados ocurren sincrónicamente. Por ende, cuando las transiciones no suceden simultáneamente, lo cual es usual en la mayoría de los casos, ciertos comportamientos quedan fuera del modelado computacional. Finalmente, dependen de discretizaciones arbitrarias de los valores de expresión de genes (Soinov *et al.*, 2003), que imponen fuertes suposiciones y restricciones acerca del sistema biológico bajo estudio.

Las redes bayesianas proveen un enfoque probabilístico para abordar el modelado de RRGs. En este caso la representación es un grafo dirigido acíclico, donde cada nodo representa una variable (usualmente genes) y las aristas representan dependencias. Ejemplos de la aplicación de

redes bayesianas al modelado de RRGs pueden encontrarse en (Friedman, 2004; Pe'er, 2005). En general el uso de esta metodología para estudiar RRGs posee interesantes ventajas, tales como la capacidad de lidiar con aspectos estocásticos y con la presencia de ruido en los datos. A pesar del fuerte respaldo teórico detrás de estos enfoques, la explosión exponencial del espacio de parámetros requerido por estos modelos conjuntamente con la gran cantidad de datos necesarios para hacer inferencias confiables, reducen su capacidad para inferir RRGs complejas usando solo datos de expresión de genes. Es más, dado que son grafos dirigidos acíclicos, no pueden representar la autorregulación o la evolución temporal de la red de forma directa (Styczynski y Stephanopoulos, 2005).

Un aspecto que en general ha sido escasamente tratado en la inferencia de RRGs cualitativas es el caso de la regulación diferida en el tiempo (*time-lagged regulation*). Esta forma de regulación sucede cuando la relación causa-efecto de una acción regulatoria requiere varias unidades de tiempo para que efectivamente ocurra.

Como se ha mencionado en otros estudios (Silvescu y Honavar, 1997; Yeang y Jaakkola, 2003), la regulación diferida en el tiempo es un fenómeno común. Entonces, las regulaciones diferidas múltiples unidades de tiempo pueden ser consideradas como la norma, mientras que las asociaciones diferidas una sola unidad de tiempo o ninguna pueden considerarse como la excepción (Li *et al.*, 2006). Este tópico de la regulación de genes diferida en el tiempo es bien reconocida por varios autores (van Someren *et al.*, 2000; Soinov *et al.*, 2003; Bulashevska y Eils, 2005; Li *et al.*, 2006; Ponzoni *et al.*, 2007), ya que permite la inferencia de potenciales relaciones causa-efecto. Sin embargo, en muchos casos solo lidian con asociaciones de genes con a lo sumo una unidad de retardo de tiempo, debido a la complejidad inherente y al costo computacional involucrado.

Por último, una de las principales falencias de las metodologías descritas anteriormente, reconocida por la mayoría de los especialistas en el área, es la dificultad de lograr modelos completos de redes regulatorias a partir de un único proceso de inferencia (Pridgeon y Corne, 2004; Schlitt y Brazma, 2005; Schlitt y Brazma, 2006). En este sentido, existen trabajos (Lee *et al.*, 2004; Lee *et al.*, 2007) que realizan inferencia de asociaciones entre genes integrando múltiples técnicas de inferencia, e incluso múltiples fuentes de información. Sin embargo, las interacciones obtenidas solo modelan relaciones simétricas entre los genes y no consideran la regulación diferida en el tiempo, lo cual no siempre corresponde al fenómeno biológico (Li *et al.*, 2006), como se ha mencionado previamente.

1.2. Objetivos y Alcance

El objetivo general de esta tesis es diseñar nuevas técnicas computacionales que asistan, a expertos en bioinformática, en la obtención de nuevos conocimientos sobre el funcionamiento de los mecanismos existentes de regulación de genes en los organismos biológicos. Más específicamente, se busca desarrollar sistemas de software que asistan en la reconstrucción (o descubrimiento) de la estructura relacional presente en las redes regulatorias de genes. En tal sentido, dicho sistema deberá inferir potenciales redes regulatorias cualitativas a nivel de genoma completo mediante el empleo de datos de expresión de genes. Además, deberá permitir inferir los patrones temporales existentes en las relaciones entre genes e integrar, en la medida de lo posible, diferentes técnicas de inferencia.

De este modo, y como consecuencia de la identificación de esas características, se procedió al estudio de los métodos que podrían resultar interesantes para atacar la problemática que nos compete. Se realizo un análisis cuidadoso de los diferentes algoritmos de aprendizaje automático, como así también su aplicación juiciosa para datos de origen biológico.

Dentro de este análisis, se iniciaron investigaciones focalizándose en la inferencia de genes co-expresados sobre subconjuntos de condiciones experimentales. La motivación de tal enfoque surgió debido a que representan uno de los tipos de información cualitativa con el mayor nivel de abstracción capaz de ser extraída de datos de expresión de genes a nivel de genoma completo. En este sentido, se emplearon nociones de *biclustering* y de algoritmos genéticos meméticos multi-objetivo. Estas ultimas son técnicas especialmente aptas para aquellos problemas de optimización que, por diferentes razones, no pueden ser resueltos satisfactoriamente con los métodos clásicos. El resultado de tales investigaciones fue una metodología de inferencia que incorpora características novedosas, como la posibilidad de inferir genes co-expresados pero cuyo nivel de expresión se da de forma opuesta. A pesar de haber obtenido resultados importantes en el estudio mencionado previamente, ciertas características presentes en el problema de *biclustering* (como el solapamiento de *biclusters*) hacen que la utilización de algoritmos de optimización multi-objetivo de propósito general no sea óptima. La razón es la poca flexibilidad de estos algoritmos evolutivos para incorporar este tipo de características en el proceso de búsqueda. De este modo, se comenzó el desarrollo de un nuevo algoritmo evolutivo memético diseñado específicamente para el problema de *biclustering* de datos de expresión de genes. El resultado fue un método que incorpora mecanismos novedosos que evitan la perdida de buenas soluciones a través de las generaciones, manteniendo un bajo solapamiento entre los *biclusters* con un nivel de diversidad satisfactorio en el espacio genotípico. Estos buenos resultados llevaron al desarrollo de un software para el análisis de datos de *microarrays*, que permite la utilización del algoritmo de *biclustering* desarrollado en un entorno de interfaz

gráfica amigable con numerosas características interesantes para los usuarios biólogos. Todos los algoritmos desarrollados en esas investigaciones fueron evaluados mediante el empleo de conjuntos de datos reales y sintéticos. En tal sentido, se introdujo un marco de comparación para evaluar la calidad de algoritmos de *biclustering,* el cual permite una comparación justa de los métodos exponiendo su verdadero desempeño.

Si bien el análisis de co-expresión de genes representa un desafío importante, la información obtenida por medio de *clustering*/*biclustering* es en general insuficiente para reconstruir una red regulatoria de genes, como se menciono previamente. En este sentido, los métodos de aprendizaje automático que se enfocan en la extracción de reglas de asociación constituyen una herramienta capaz de inferir este tipo de interacciones cualitativas entre genes. Luego de un análisis minucioso del estado del arte de los enfoques de extracción de reglas a partir de datos de *microarrays*, se diseño un nuevo algoritmo tomando como base un método de extracción de reglas de asociación entre genes basado en optimización combinatorial previamente propuesto por nuestro grupo de investigación (Ponzoni *et al*., 2007). Esto conllevo a un rediseño total del método con el fin de superar sus limitaciones, resultando en un novedoso algoritmo que incorpora nuevas características como la inferencia de reglas de interacción con múltiples retardos de tiempo, la consideración de múltiples datos de serie de tiempo en la extracción de las reglas y varias mejoras sobre el proceso básico de inferencia. A fin de validar la nueva propuesta, se realizo un amplio análisis de su desempeño con datos reales, evaluando su rendimiento en comparación con otros algoritmos del estado del arte y analizando la relevancia biológica de los resultados obtenidos. Además, el nuevo método mostró su aplicabilidad en contextos de genoma completo, en donde la gran dimensionalidad en los genes torna prácticamente imposible la utilización de otros algoritmos relacionados.

El desarrollo del conjunto de metodologías propuestas para la inferencia de relaciones entre genes aportó herramientas que permiten dilucidar la estructura relacional de las RRGs. En tal sentido, los dos algoritmos principales desarrollados en esta tesis se integraron en un sistema de software conformando un marco de trabajo amigable al usuario, proveyendo la oportunidad para realizar ingeniería inversa de RRGs diferidas en el tiempo basadas en reglas de asociación, y a nivel de genoma completo.

1.3. Contenido y Estructura

La tesis está organizada en 8 capítulos. En el presente capitulo, se introdujo brevemente la problemática y se delinearon los principales objetivos. Los capítulos 2 y 3 introducen conceptos básicos de biología molecular y tecnología de *microarrays*, y están orientados a introducir al lector ajeno a estos conocimientos con los términos que serán utilizados en el resto de la tesis.

Luego, el capitulo 4 introduce la problemática de *biclustering*, presenta a la computación evolutiva y proporciona una revisión del estado del arte de las técnicas disponibles en la literatura para *biclustering*. A continuación se detallan, en el capitulo 5, las contribuciones de esta tesis relacionadas con el *biclustering* de datos de *microarrays*. En el capitulo 6 se introduce la problemática de inferencia de reglas de asociación entre genes conjuntamente con una revisión del estado del arte en esa rama de la bioinformática. El capitulo 7 esta destinado a presentar la contribución de esta tesis en lo relacionado a la inferencia de reglas de asociación entre genes presentando, además, la herramienta de software integradora. Finalmente, en el capítulo 8, se exponen las principales conclusiones de este trabajo y se enuncian lineamientos y sugerencias para futuras investigaciones y mejoras.

1.4. Sumario

En esta sección se introdujo al lector en los principales enfoques utilizados para realizar la inferencia de redes regulatorias de genes. Una vez descriptos brevemente tales enfoques, y habiendo introducido las principales cuestiones para abordar esta problemática, se detalló el principal objetivo de esta tesis. El mismo consistió en desarrollar algoritmos que asistan en la reconstrucción (o descubrimiento) de la estructura relacional presente en las redes regulatorias de genes. A su vez, se describió el alcance del trabajo aquí propuesto y desarrollado, y la importancia de las investigaciones realizadas. Finalmente se mostró la organización de esta tesis para un mejor entendimiento y seguimiento de la misma.

Capítulo 2

Conceptos Básicos de Biología Molecular

Para poder entender los métodos y conceptos discutidos en los siguientes capítulos, se asume que el lector esta familiarizado con los conceptos de biología molecular. Este capitulo esta destinado a proveer las bases para aquellos sin este conocimiento. Dado que será una presentación muy general, muchas excepciones y desviaciones de los principios generales no serán tratados aquí. Para una examinación en profundidad de la biología molecular, se sugiere la lectura de *Life: the Science of Biology* por William K. Purves *et al.* (2003), *Genes VIII* por Benjamin Lewin (2003) y *Molecular Biology* por Robert F. Weaver (2001).

2.1. Célula

El sentido literal de "biología" es "el estudio de todas las cosas vivas" (Standafer y Wahlgren, 2002). En el contexto biológico, las "cosas vivas" son definidas por la presencia de ciertas características. Algunas de estas cualidades incluyen la habilidad para crecer y desarrollarse, mantener la homeostasis interna, reproducirse, detectar y responder a estímulos, adquirir y liberar energía, e interactuar con su entorno así como entre ellas (Farabee, 2007). En resumen, las cosas vivas tienen una activa participación en su entorno, en contraposición al estatus inactivo de las cosas no vivas (Setubal y Meidanis, 1997).

El pensamiento paleontológico actual ubica al origen de la vida en la Tierra hace 3.5 billones de años, poco después (en términos geológicos) de que la Tierra misma se formara (hace mas de 4 billones de años) (Standafer y Wahlgren, 2002). A partir de las primeras formas de vida simples, la acción del proceso evolutivo durante billones de años resulto en el cambio y diversificación de especies, para que hoy encontremos tanto organismos altamente complejos como organismos muy simples.

Tanto los organismos complejos como los simples están hechos de células, que a su vez pueden descomponerse en organelas, organelas en moléculas, y así siguiendo a través de la disminución en la jerarquía del tamaño. La Teoría de las Células fue primeramente formalizada en 1839 por Theodor Schwann y Matthias Schleiden y fue subsecuentemente elaborada por Rudolph Virchow y otros biólogos. Esta teoría tiene tres partes: 1) todas las cosas vivas están compuestas de una o mas células; 2) las células son las unidades básicas de estructura y función en un organismo; 3) las células provienen solamente de la reproducción de células existentes (Standafer y Wahlgren, 2002).

Las dos clases básicas de células, llamadas células procariotas y células eucariotas, se distinguen por su tamaño y los tipos de estructuras internas o organelas que contienen. Las estructuralmente simples células procariotas están representadas por las bacterias y las cianobacterias. Todos los otros tipos de organismos - protistas, hongos, plantas y animales - consisten de células eucariotas que son estructuralmente mas complejas (Karp, 2002).

La estructura interna y funciones de las células eucariotas son mucho mas complejas que aquellas de las células procariotas. La figura 2.1a y 2.1b ilustran la estructura interna de una célula animal eucariota y de una célula procariota promedio, respectivamente. Ambas células eucariota y procariota contienen una región nuclear que contiene el material genético de la célula. El material genético de una célula procariota esta presente en un nucleoide, que es una región pobremente demarcada de la célula la cual no contienen ninguna membrana que la separe del citoplasma que la rodea. En contraste, las células eucariotas posen un núcleo, que es una región delimitada por una estructura membranosa compleja llamada envoltura nuclear. Esta diferencia en la estructura nuclear es la base para los términos procariota (pro=antes, cariota = núcleo) y eucariota (eu = verdadero, cariota = núcleo) (Karp, 2002).

Tanto las células procariotas como las células eucariotas comparten una química molecular similar. Las moléculas más importantes en la química de la vida son las proteínas y los ácidos nucleicos. Hablando en términos simples, las proteínas determinan que es y que hace un ser viviente en un sentido físico, mientras que los ácidos nucleicos son responsables de la codificación de la información genética y de la transferencia de esta información a la siguiente generación (Setubal y Meidanis, 1997). En las siguientes secciones, se brindara una descripción breve del estado actual del conocimiento respecto de estas dos moléculas.

2.2. Proteínas

Todos los organismos vivos están compuestos mayormente por proteínas, y su importancia fue bien establecida por el distinguido científico Russell Doolitle, quien escribió "nosotros somos nuestras proteínas". Hay muchos tipos de proteínas. Las proteínas estructurales forman parten de la estructura celular, las encimas catalizan casi todas las reacciones bioquímicas que ocurren en la célula, las proteínas regulatorias controlan la expresión de los genes o la actividad de otras proteínas, y las proteínas de transporte llevan otras moléculas a través de la membrana celular o alrededor del cuerpo (Amaratunga y Cabrera, 2003).

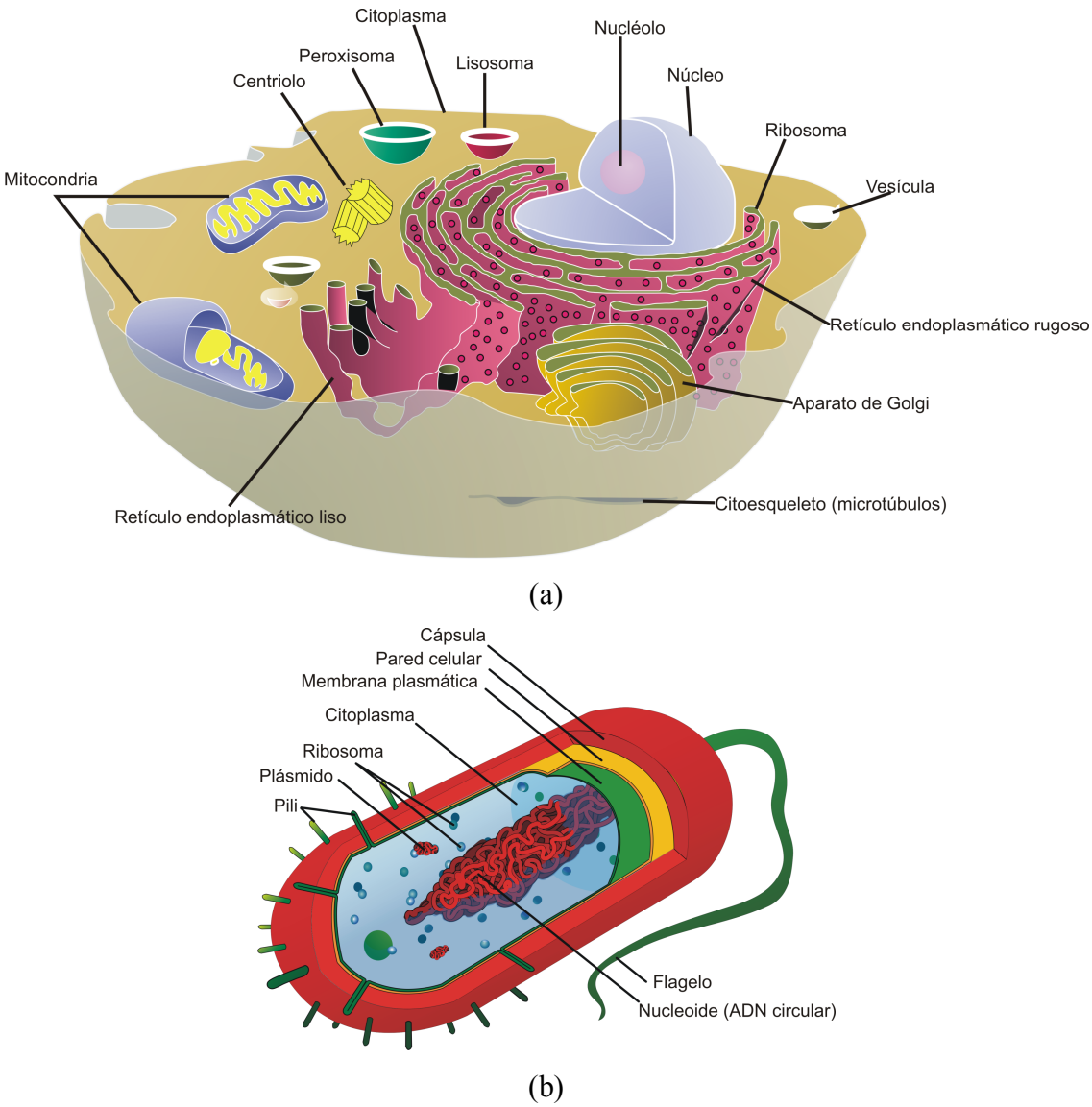


Figura 2.1. Estructura de una célula. Fig. 2.1a: Célula eucariota animal. Fig. 2.1b: célula procariota promedio.

Una proteína esta compuesta por una cadena de aminoácidos. La figura 2.2 ilustra la estructura de un aminoácido. Cada aminoácido esta organizado alrededor de un átomo central de carbono, conocido como el carbón alfa, o C_{α}. Entre los otros componentes de un aminoácido hay un átomo de hidrogeno, un grupo amino (NH_2), un grupo carboxilo (COOH), y una cadena lateral (Grupo R) (Purves *et al*., 2003). Es la cadena lateral la que distingue a un aminoácido de otro. En la naturaleza hay 20 aminoácidos, los cuales están listados en la tabla 2.1. Todas las cosas vivas (incluso los virus, que no cumplen completamente el criterio de "vivo") están hechos de varias combinaciones de los mismos 20 aminoácidos.

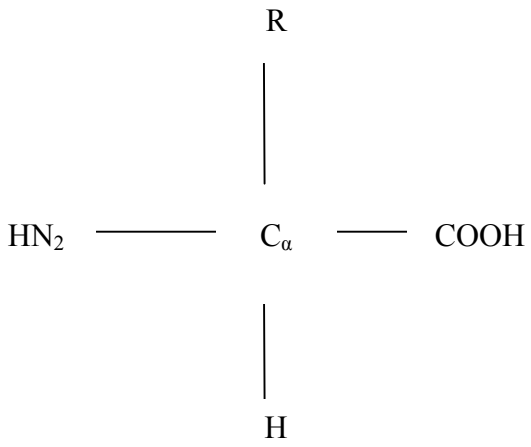


Figura 2.2. Estructura general de un aminoácido.

Estructuralmente, las proteínas son cadenas polipeptídicas, en las que los aminoácidos están ligados por una ligadura péptica. Una ligadura péptica se forma por la adición del extremo carboxilo de un aminoácido con el extremo amino del aminoácido adyacente, liberando una molécula de agua en el proceso. Como resultado de la perdida de agua, la cadena polipeptídica esta realmente hecha de residuos deshidratados de los aminoácidos originales. Entonces, generalmente se habla de una proteína como formada por un cierto numero de residuos mas que de aminoácidos. Las ligaduras pépticas forman la columna vertebral de cualquier proteína, que esta hecha de repeticiones de los bloques básicos - N - C_α - (CO) - (Setubal y Meidanis, 1997).

La secuencia de residuos de un polipéptido es llamada la estructura primaria de una proteína. Es mas, las proteínas se pliegan en tres dimensiones, resultando en estructuras secundarias, terciarias y cuaternarias. La estructura secundaria de una proteína esta formada a través de la tendencia del polipéptido a enrollarse o plegarse debido a la ligadura de H entre los grupos R. Esto resulta en estructuras "locales" como las hélices alfa y las hojas plegadas beta. Las estructuras terciarias son el resultado del empaquetado de la estructura secundaria en un nivel más global, debido a la atracción o repulsión entre los grupos R. Muchas proteínas están formadas por más de una cadena polipeptídica; la hemoglobina es un ejemplo de tal proteína. Estos polipéptidos conjuntamente empaquetados forman un nivel superior llamado estructura cuaternaria.

Tabla 2.1. Los 20 aminoácidos comúnmente encontrados en la naturaleza.

Aminoácido	Código de tres letras	Código de una letra
Alanina	Ala	A
Arginina	Arg	R
Asparagina	Asn	N
Ácido aspártico	Asp	D
Cisteína	Cys	C
Glutamina	Gln	Q
Ácido glutámic	Glu	E
Glicina	Gly	G
Histidina	His	H
Isoleucina	Ile	I
Leucina	Leu	L
Lisina	Lys	K
Metionina	Met	M
Fenilalanina	Phe	F
Prolina	Pro	P
Serina	Ser	S
Treonina	Thr	T
Triptófano	Trp	W
Tirosina	Tyr	Y
Valina	Val	V

La secuencia lineal de residuos (la estructura primaria) de una proteína determina su estructura tridimensional, y la forma tridimensional de una proteína determina su función. La razón fundamental es que una proteína plegada tiene una forma irregular con rincones y protuberancias que permiten un contacto cercano con otras moléculas (Setubal y Meidanis, 1997). Por ejemplo, las proteínas estructurales, como el colágeno, tienen una estructura regular y repetida. Estas realizan una variedad de funciones en los seres vivos; por ejemplo, forman los tendones, cuero y corneas de una vaca.

2.3. Acido Nucleico

El acido nucleico codifica la información necesaria para producir proteínas y es responsable de pasar esta "receta" a las subsiguientes generaciones (Setubal y Meidanis, 1997). Hay dos tipos básicos de acido nucleico: el ácido ribonucleico (ARN) y el ácido desoxirribonucleico (ADN). La secuencia polipeptídica que forma la estructura primaria de una proteína esta directamente relacionada con la información en la molécula de ARN, que a su vez, es una copia de la información en la molécula de ADN (ver la sección 2.4 para una descripción detallada de este proceso).

2.3.1. ADN

Una molécula de ADN consiste en dos hebras de moléculas simples. Cada hebra tiene una columna vertebral constituida por repeticiones de la misma unidad básica. La figura 2.3 (panel

derecho) ilustra la estructura de la columna vertebral del ADN. La unidad básica de ADN esta formada por una molécula de azúcar, 2'-desoxirribosa, ligada a un residuo fosfato. La molécula de azúcar contiene cinco átomos de carbono, etiquetados de 1' a 5' (ver figura 2.4 (izquierda)). La columna vertebral esta formada por una serie de ligaduras entre el átomo de carbono 3' de una molécula de azúcar, el residuo fosfato, y el átomo de carbono 5' de la siguiente molécula de azúcar. las hebras de ADN tienen una orientación determinada por la numeración de los átomos de carbono, que por convención, comienza en el extremo 5' y termina en el extremo 3'. Una secuencia de ADN de hebra simple es siempre escrita en esta dirección canónica 5' → 3' (Setubal y Meidanis, 1997).

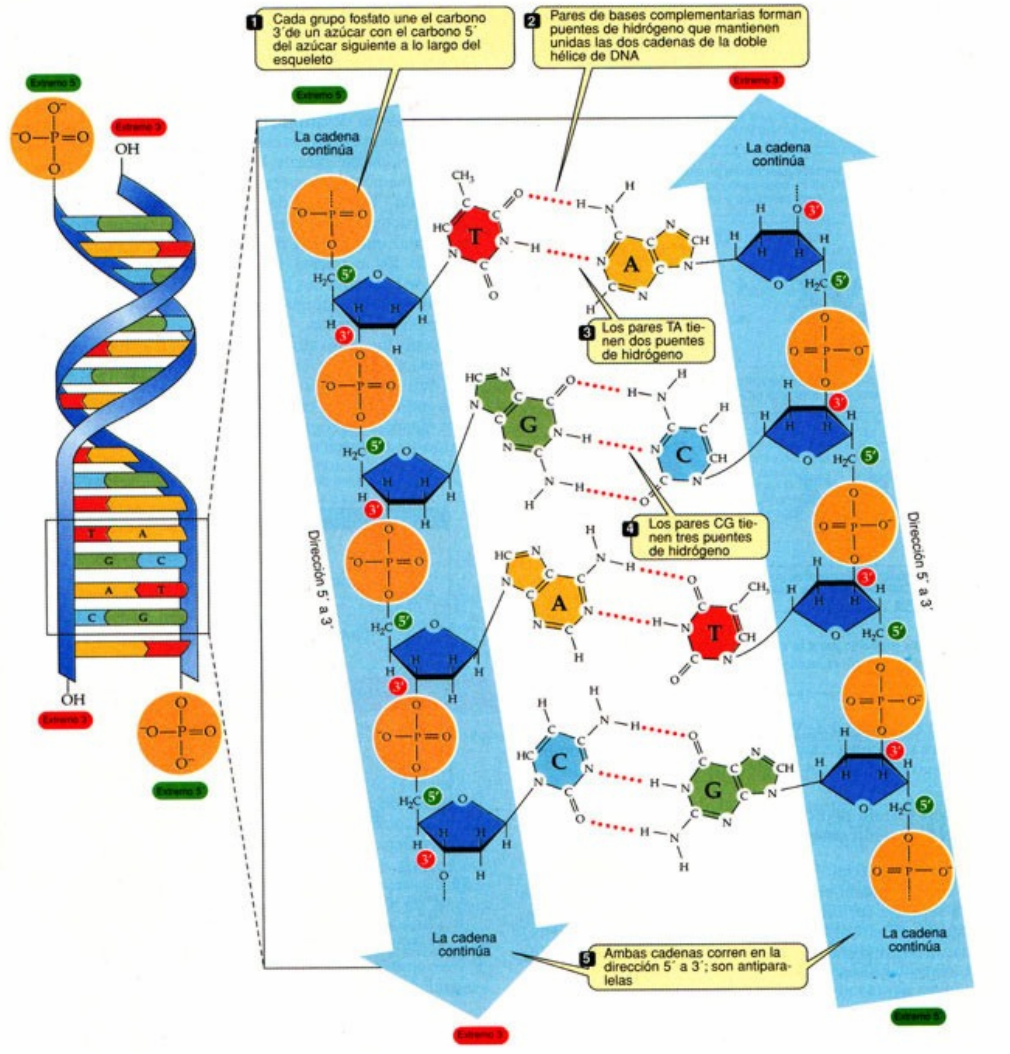


Figura 2.3. La estructura de doble hélice (izquierda) y la columna vertebral (derecha) del ADN. Fuente: Aidong Zhang (2006).

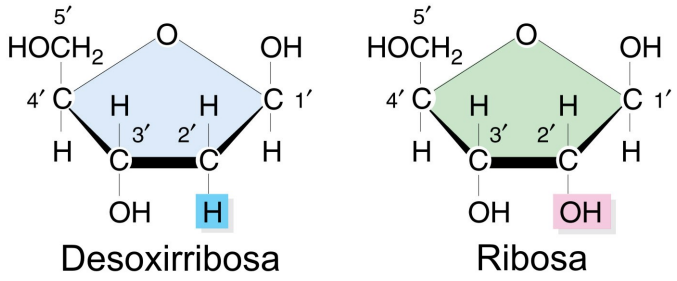


Figura 2.4. Estructura de la desoxirribosa (izquierda) y de la ribosa (derecha).

Las moléculas ligadas al átomo de carbono 1' son llamadas bases. Hay cuatro tipos de bases: adenina (A), guanina(G), citosina (C) y timina (T). Las bases que tienen dos anillos de átomos de carbono y nitrógeno, como la adenina y la guanina, son llamadas purinas. Las pirimidinas son las bases que tienen un solo anillo de átomos de carbono y nitrógeno, como la citosina y la timina (Standafer y Wahlgren, 2002). Dado que los nucleótidos están diferenciados solo por sus bases, se puede hacer referencia a que una molécula de ADN tiene tantas bases o nucleótidos, aunque bases o nucleótidos no sean sinónimos. En la naturaleza las moléculas de ADN son muy largas. En una célula humana, las moléculas de ADN tienen cientos de millones de nucleótidos (Setubal y Meidanis, 1997). A los fragmentos cortos de moléculas de ADN, típicamente de entre 5 y 50 pares de bases, se los llaman oligonucleótidos (o oligo).

Como se mencionó antes, las moléculas de ADN consisten en hebras dobles. Las hebras dobles están ligadas conjuntamente en una estructura de hélice (ver figura 2.3, panel izquierdo), con las bases en el centro y la unidades azúcar fosfato en los lados de la hélice. Esta estructura de doble hélice fue descubierta por James Watson y Francis Crick en 1953. Las bases en las dos hebras están apareadas acorde a la regla de apareo de bases complementarias (también llamada regla de apareo Watson-Crick): la adenina (base A) solo se liga a la timina (base T) y la guanina (base G) solo se liga a la citosina (C). Los pares formados se llaman pares de bases; ellos proveen la unidad de longitud mas usada en lo que se refiere a moléculas de ADN, y es abreviado con bp (*base pair*). Así, se puede decir por ejemplo que cierta pieza de ADN es 100000 bp de largo (Setubal y Meidanis, 1997). La fuerza que mantiene a los pares de bases unidos es un enlace débil de hidrogeno. Aunque cada enlace es débil, el efecto acumulativo de muchos de esos enlaces es suficientemente fuerte para mantener robustamente unidas a las dos hebras. Como resultado, el ADN es químicamente inerte y es en general un transportador estable de información genética (Amaratunga y Cabrera, 2003).

Tal como se mencionó arriba, cada hebra de ADN tiene una orientación de 5' a 3', indicada por la secuencia de sus átomos de carbono. En la estructura de doble hélice del ADN, cada hebra mantiene su propia orientación, con el extremo 5' de una hebra alineado con el extremo 3' de la otra hebra. En otras palabras, las dos hebras son anti paralelas. Como consecuencia, es posible

inferir la secuencia de una hebra si se conoce la secuencia de la otra por medio de una operación llamada complementación reversa. Por ejemplo, dada una hebra s=AGCTAAC en la dirección 5' → 3', primero se invierte s, obteniendo s'=CAATCGA, y luego se reemplaza cada base por su complemento, obteniendo s''=GTTAGCT, que es el complemento reverso de s (Setubal y Meidanis, 1997).

Esta complementación reversa es precisamente el mecanismo que permite al ADN en la célula ser replicado. Incluso antes del descubrimiento del ADN, se había reconocido que cualquier transportador de material hereditario debía ser capaz de replicarse a si mismo para que la información sea pasada de una generación a la siguiente. Sin embargo, el mecanismo de autorreplicación era desconocido. Cuando la estructura del ADN fue deducida, se comprendió que la estructura complementaria de la molécula de ADN permitiría la exacta auto replicación, cumpliendo con este requerimiento (Amaratunga y Cabrera, 2003).

2.3.2. ARN

Las moléculas de ARN son similares a las moléculas de ADN, con las siguientes diferencias básicas de composición y estructura (Setubal y Meidanis, 1997):

- El componente azúcar del ARN es la ribosa (ver figura 2.4 (derecha)) en vez de desoxirribosa.
- En el ARN, la timina (T) esta reemplazada por el uracilo (U) que también liga con la adenina.
- El ARN no forma una doble hélice. A veces hélices hibridas de ARN-ADN ocurren, o partes de una molécula de ARN pueden ligarse a otras partes de la misma molécula por complemento. La estructura en tres dimensiones del ARN es mucho mas variada que la del ADN.

El ADN y el ARN también difieren en que mientras que el ADN realiza esencialmente una función (que es la de codificar información), las células contienen una variedad de tipos de ARN, cada uno realizando diferentes funciones (Setubal y Meidanis, 1997). Esto será discutido con mas detalle mas adelante.

2.4. Dogma Central de la Biología Molecular

Las moléculas de ADN son responsables de codificar la información necesaria para construir cada proteína o molécula de ADN que se encuentran en un organismo. En esencia, al ADN a veces se lo refiere como el "plano de la vida". El flujo de información fluye desde el ADN vía el ARN y así a las proteínas, y es descrito por el llamado dogma central de la biología molecular, que incluye las cuatro etapas siguientes (Crick, 1970):

1. La información contenida en el ADN es duplicada por medio del proceso de replicación.
2. El ADN dirige la producción de ARN mensajero (ARNm) codificado a través del proceso llamado trascripción.
3. En las células eucariotas, el ARNm es luego procesado y migra del núcleo al citoplasma de la célula.
4. En la etapa final del proceso de transferencia de información, el ARNm transporta la información codificada a las estructuras sintetizadoras de proteínas llamadas ribosomas. A través de un proceso llamado traslación, los ribosomas usan esta información codificada para dirigir la síntesis de proteínas.

En esta sección, se describirán los mecanismos de codificación y como una proteína es construida a partir del ADN.

2.4.1. Genes y el Código Genético

Cada célula de un organismo tiene una o más moléculas de ADN. Cada molécula de ADN forma un cromosoma. El conjunto completo de cromosomas dentro de una célula es llamado genoma. El número de cromosomas en un genoma es característico de una especie particular. Por ejemplo, cada célula en los *Homo sapiens* (humanos modernos) tienen 46 cromosomas, mientras que este numero es de 8 en la *Drosophila melanogaster* (la mosca de la fruta) y 32 en la *Saccharomyces cerevisiae* (levadura).

Una molécula de ADN contiene ciertos segmentos continuos que codifican la información para construir proteínas. Sin embargo, algunas porciones de la molécula de ADN no contienen información codificada y son denominadas "ADN basura". Esto puede sonar como una denominación errada, ya que se ha sugerido que el ADN basura puede en realidad realizar alguna función importante y no reconocida. Un gen es un segmento continuo de ADN que contiene la información necesaria para construir una proteína o una molécula de ARN. En particular, un gen estructural es un gen cuya información codifica solo para proteínas. La longitud de los genes varían, pero los genes humanos tienen normalmente cerca de 10000bp. El punto de comienzo y de finalización de un gen pueden ser reconocidos por ciertos mecanismos celulares.

Como se describió en la Sección 2.2, una proteína esta compuesta de una cadena de aminoácidos. El mecanismo por el cual los genes especifican la secuencia de aminoácidos en una proteína se llama código genético. Para ser específicos, una tripleta de nucleótidos es usada para especificar cada aminoácido. Esa tripleta es llamada codón. La figura 2.5 ilustra la

correspondencia entre cada aminoácido y cada posible tripleta. En esta figura, las tripletas están denotadas usando bases de ARN en vez de bases de ADN, dado que es el ARN el que provee el link entre el ADN y la síntesis de proteínas. Esto será discutido con mas detalle en la Sección 2.4.2.

Tabla 2.2. Código genético mapeando aminoácidos en codones.

Aminoácido	Codones					
Alanina	GCA	GCC	CGC	GCU		
Arginina	AGA	AGG	CGA	CGC	CGG	CGU
Asparagina	GAC	GAU				
Ácido aspártico	AAC	AAU				
Cisteína	UCG	UGU				
Glutamina	CAA	CAG				
Ácido glutámico	GAA	GAG				
Glicina	GGA	GGC	GGG	GGU		
Histidina	CAC	CAU				
Isoleucina	AUA	AUC	AUU			
Leucina	UUA	UUG	CUA	CUC	CUG	CUU
Lisina	AAA	AAG				
Metionina	AUG					
Fenilalanina	UUC	UUU				
Prolina	CCA	CCC	CCG	CCU		
Serina	AGC	AGU	UCA	UCC	UCG	UCU
Treonina	ACA	ACC	ACG	ACU		
Triptófano	UGG					
Tirosina	UAC	UAU				
Valina	GUA	GUC	GUG	GUU		
Stop	UAA	UAG	UGA			

Dados los cuatro tipos de bases, el numero total de posibles combinaciones de tripletas con los nucleótidos es de 64. Sin embargo, estas 64 combinaciones pueden referir solo a 20 aminoácidos que realmente ocurren en la naturaleza. Entonces, hay redundancia en la codificación, y varias tripletas corresponden al mismo aminoácido. Por ejemplo, tanto AAG como AAA codifican para lisina (ver tabla 2.2). Es mas, tres de los posibles codones (UAA, UAG y UGA) no codifican a ningún aminoácido y son usados en cambio para señalizar el final de un gen. Esta redundancia es realmente una característica importante en el código genético, haciéndolo mas robusto en el caso de que pequeños errores ocurran en el proceso de transcripción. Esto será discutido en la siguiente sección.

2.4.2. Transcripción y Expresión de genes

La transcripción es el proceso de sintetizar ARN usando genes como molde. Un gen esta expresado cuando, a través del proceso de transcripción, su codificación es transferida a una molécula de ARN. Para iniciar el proceso de transcripción, la doble hélice del ADN se debe separar, comenzando en el sitio promotor del gen. El sitio promotor es una región del lado 5' de la hebra de ADN que indica que un gen esta próximo. El codón AUG, que codifica para metionina, también hace de señalización para el comienzo de un gen. Una vez que la doble hélice del ADN se ha abierto en el punto de comienzo, una hebra del ADN sirve como hebra molde. Una molécula de ARN es construida ligando entre si ribonucleótidos complementarios a

la hebra molde hasta que se llega a un codón STOP. El proceso de composición siempre construye moléculas de ARNm desde el extremo 5' al extremo 3', mientras que la hebra es leída desde 3' a 5'. Este ARN resultante es llamado ARN mensajero, o brevemente, ARNm. Dado que las dos hebras de la hélice de ADN original también son complementarias, esta nueva molécula de ARNm tendrá la misma secuencia de ribonucleótidos sin uso del la hebra de ADN, con la base U sustituyendo a T. La elección del molde a partir de las dos hebras disponibles en el ADN original varia de gen a gen, acorde a como es señalizado por la ubicación del sitio promotor de cada gen. Después del proceso de transcripción, el ARNm será transportado a las estructuras celulares llamadas ribosomas para guiar la manufactura de proteínas.

La transcripción como fue descripta arriba es válida para procariotas. Para eucariotas, muchos genes están compuestos por partes alternativas llamas intrones y exones. Después de la transcripción, los intrones son quitados del ARNm. Esto significa que solo los exones participan de la síntesis de proteínas. El *splicing* alternativo ocurre cuando el mismo ADN genómico puede dar como resultado 2 o mas moléculas de ARNm diferentes dependiendo de la elección alternativa de intrones y exones, resultando generalmente en la producción de diferentes proteínas (para mas detalles de los intrones, exones y *splicing* alternativo recurrir a (Lewin, 2003)). Debido a los cambios resultantes por el *splicing* de intrones y exones, el gen completo de los cromosomas es usualmente denominado ADN genómico, y la secuencia cortada conteniendo solo exones es llamada ADN complementario o ADNc (Setubal y Meidanis, 1997). El ADNc puede obtenerse por el proceso de transcripción inversa que transforma ARNm en ADN.

Con una secuencia de ADN o ARN, las bases pueden ser interpretadas de diferentes formas generando grupos de codones distintos. Por ejemplo, en la secuencia TAATCGAATGGGC, las bases adyacentes pueden agruparse en los codones TAA, TCG, AAT, GGG, omitiendo la C final. También es posible ignorar la T inicial, produciendo los codones ATT, CGA, ATG, GGC. Otra posible lectura de la secuencia podría resultar en los codones ATC, GAA, TGC, por medio de la omisión de las dos bases iniciales (TA) y las dos finales (GC). Un marco abierto de lectura (*open reading frame* o ORF) selecciona uno de estos enfoques para leer la secuencia de ADN e interpretar las bases en secuencias que comienzan en un codón de inicio, contienen un numero integral de codones, y no incluyen ningún codón STOP en la secuencia (Setubal y Meidanis, 1997).

Los científicos involucrados en la investigación de la expresión de genes usualmente encuentran mas fácil trabajar con marcadores de secuencia expresada (*expressed sequence tags* o ESTs). Un EST es una subsecuencia corta y única (de solo unos cientos de pares de bases en longitud), generada a partir de una secuencia de ADN de un gen, que actúa como marcador para

el gen. Una ventaja de las ESTs es que se pueden retraducir en código genético que codifica o expresa exones mas que incluir intrones u otro ADN sin codificación (Amaratunga y Cabrera, 2003).

2.4.3. Traslación y Síntesis de Proteínas

Una vez que el proceso de transcripción ha generado ARNm codificado apropiadamente, se inicia el proceso de traslación que sintetiza proteínas. Como se mencionó anteriormente, la síntesis de proteínas toma lugar dentro de las estructuras celulares llamadas ribosomas (ver figura 2.5). Otro tipo de ARN, ARN de transferencia o ARNt, hace la conexión entre un codón y el aminoácido correspondiente. Como se ilustra en la figura 2.6, cada molécula de ARNt tiene, en un lado un anticodón que tiene alta afinidad por un codón especifico y, en el otro extremo, un sitio de acoplamiento de aminoácido que se liga fácilmente al aminoácido correspondiente. A medida que el ARNm pasa a través del interior del ribosoma, un ARNt que machea el codón actual se ligara a el y traerá el aminoácido correspondiente. La fijación del aminoácido toma lugar justo al lado del aminoácido anterior en la cadena proteínica que esta siendo formada. Luego, una enzima cataliza la adición del aminoácido actual a la cadena proteínica, liberando el ARNt. De esta forma, la proteína es construida residuo por residuo. Cuando aparece un codón STOP, no hay ningún ARNt asociado a el, por lo que la síntesis termina. El ARNm es liberado y degradado en ribonucleótidos por mecanismos celulares que luego serán reciclados para fabricar otro ARN (Setubal y Meidanis, 1997).

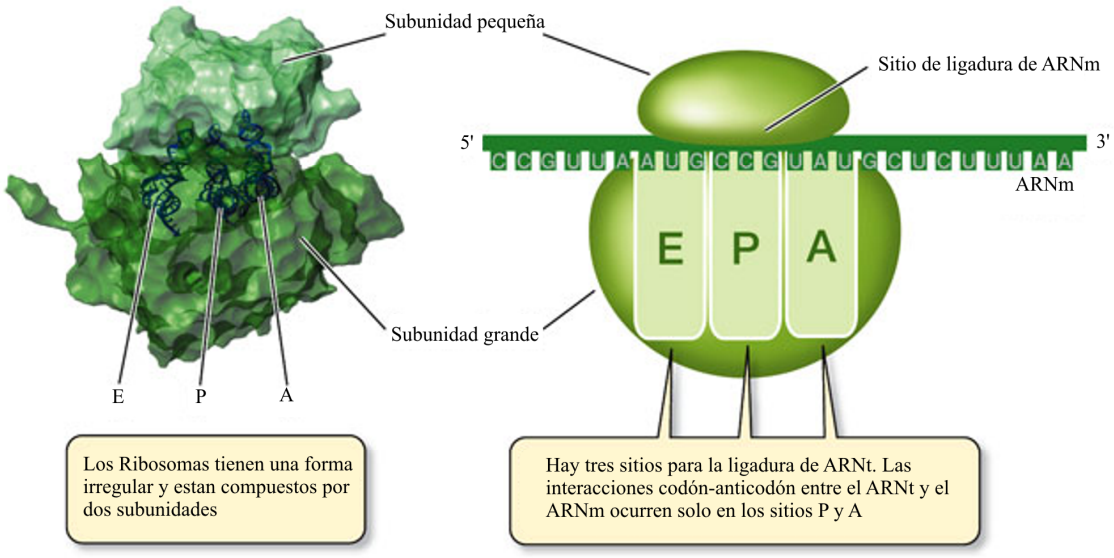


Figura 2.5. La estructura de un ribosoma. Fuente: Aidong Zhang (2006).

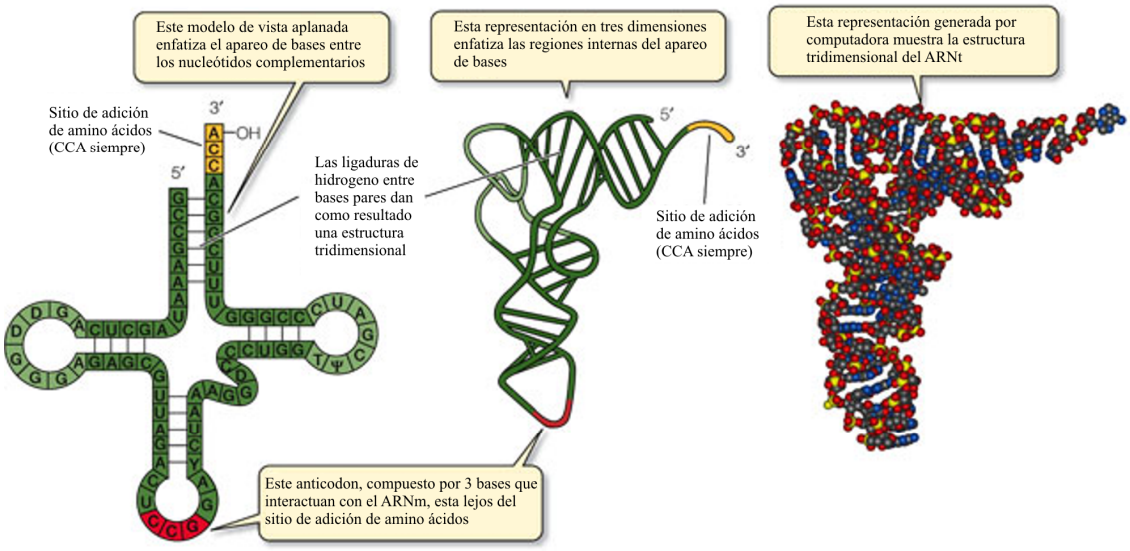


Figura 2.6. La estructura de el ARNt. Fuente: Aidong Zhang (2006).

2.5. La Regulación de la Expresión Génica

La regulación génica comprende todos aquellos procesos que afectan la acción de un gen, regulando sus productos funcionales. A continuación introduciremos brevemente como ocurre esa regulación tanto en los organismos procariotas como en los eucariotas, e introduciremos la noción de red regulatoria de genes.

2.5.1. Organismos Procariotas

En el curso de su larga historia evolutiva, *E. coli* y otro procariotas han desarrollado procedimientos que les permiten utilizar al máximo los nutrientes destinados al crecimiento celular. Si fuera posible decir que una célula bacteriana tiene un propósito o una función, aunque no lo es, ésta consistiría en crecer y multiplicarse lo mas rápidamente posible, y las bacterias son excelentes para lograrlo; un cultivo de células *E. coli*, por ejemplo, puede duplicar su numero cada 20 minutos.

Una razón para la efectividad de *E. coli* en el uso de nutrientes es su versatilidad; puede fabricar, por lo menos, 1700 enzimas y otras proteínas diferentes, lo cual la capacita para utilizar una amplia gama de nutrientes potenciales. Una segunda razón es que la célula es altamente eficiente en sus actividades de síntesis. No produce todas las proteínas posibles al mismo tiempo, sino solo cuando se precisan y en las cantidades necesarias. Por ejemplo, las células *E. coli* abastecidas con el disacárido lactosa como fuente de carbono y energía, requieren la enzima beta-galactosidasa para escindir ese disacárido. Las células que crecen en un medio con lactosa fabrican aproximadamente 3000 moléculas de beta-galactosidasa. Sin embargo, en ausencia de lactosa hay un promedio de una molécula de enzima por célula. En conclusión, la presencia de lactosa provoca la inducción de la producción de las moléculas de enzima necesarias para degradarla (figura 2.7a). Se dice, entonces, que esas enzimas son inducibles.

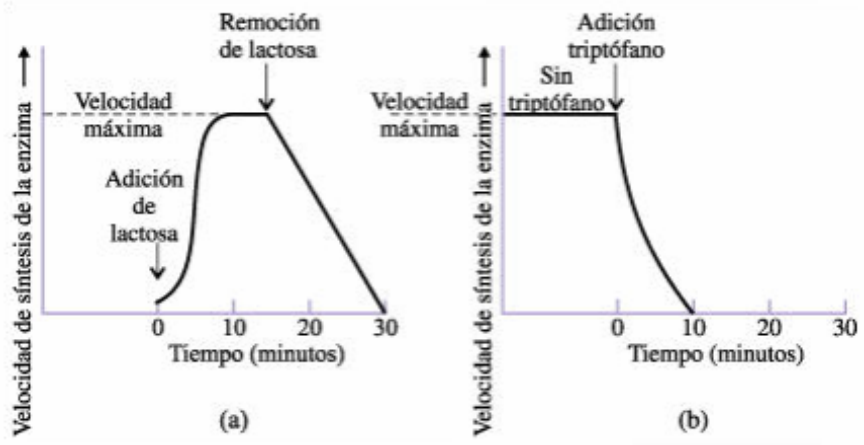


Figura 2.7. Enzimas inducibles y represibles. Fuente: Curtis *et al.* (2000).

Por el contrario, la presencia de un nutriente determinado puede inhibir la transcripción de un grupo de genes estructurales. *E. coli,* como otras bacterias, puede sintetizar uno de sus aminoácidos a partir de amoniaco y de una fuente de carbono. Los genes estructurales que codifican las enzimas necesarias para la biosíntesis del aminoácido triptófano, por ejemplo, están agrupados y se transcriben en una única molécula de ARNm. Este ARNm es producido continuamente por células en crecimiento si el triptófano no esta presente. En presencia de triptófano, se detiene la producción de las enzimas (figura 2.7b). Estas enzimas, cuya síntesis se reduce en presencia de los productos de las reacciones que catalizan, se denominan represibles.
En algunas ocasiones, aparecen mutantes de *E. coli* incapaces de regular la producción enzimática. Estas células, por ejemplo, producen beta-gatactosidasa aun en ausencia de lactosa, o fabrican las enzimas que sintetizan triptófano aun cuando el triptófano esta presente. Estos mutantes y otros similares están generalmente en desventaja, porque malgastan sus energías y recursos y, en consecuencia, las células normales de *E. coli* rápidamente los sobrepasan en numero.

Aunque la regulación de la síntesis de proteínas teóricamente podría ocurrir en muchos puntos del proceso biosintético, en los procariotas tiene lugar principalmente a nivel de la transcripción. La regulación implica interacciones entre el ambiente químico de la célula y las proteínas reguladoras especiales codificadas por genes reguladores. Estas proteínas pueden funcionar como controles negativos, reprimiendo la transcripción del ARNm, o como controles positivos, intensificándola. El hecho de que el ARNm se traduzca a proteína en forma tan inmediata (aun antes de que se haya completado la transcripción) y se degrade tan rápidamente, incrementa mas aun la eficiencia de esta estrategia de regulación.

La detección de los mutantes recién mencionados fue lo que condujo a nuestra comprensión actual de la regulación de la transcripción en los procariotas. Este conocimiento descansa en un modelo conocido como el modelo del operón. De acuerdo a este modelo, los grupos de genes que codifican proteínas con funciones relacionadas se disponen en unidades conocidas como operones. Un operón (figura 2.8) comprende el promotor, los genes estructurales y otra secuencia de ADN conocida como operador. El operador es una secuencia de nucleótidos situados entre el promotor y el gen o los genes estructurales.

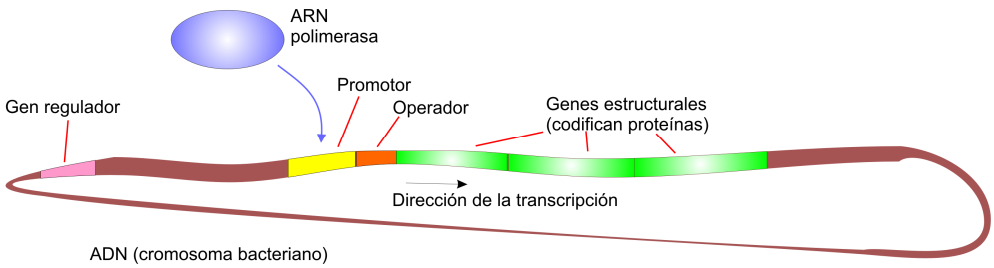


Figura 2.8. Representación esquemática de un operón.

La transcripción de los genes estructurales depende, frecuentemente, de la actividad de otro gen, el regulador, que puede estar localizado en cualquier lugar del cromosoma bacteriano. Este gen codifica una proteína llamada represor, que se une al operador. Cuando un represor está unido al operador, obstruye al promotor. En consecuencia, la ARN polimerasa no puede unirse a la molécula de ADN o, si se une, no puede comenzar su movimiento a lo largo de la molécula. El resultado, en cualquier caso, es el mismo: no hay transcripción del ARNm. Sin embargo, cuando se remueve el represor, la transcripción puede comenzar.

La capacidad del represor para unirse al operador y así bloquear la síntesis de proteínas depende, a su vez, de otra molécula que funciona como un efector. Según el tipo de operón, el efecto puede activar o bien inactivar al represor para ese operón en particular. El operón *lac* es un ejemplo en el que el efector funciona inactivando al represor. Cuando hay lactosa en el medio de cultivo, el primer paso en su metabolismo es la producción de un azúcar íntimamente relacionado, la alolactosa, que se une al represor y lo inactiva, separándolo del operador del operón *lac*. Como consecuencia de esto, la ARN polimerasa puede comenzar su movimiento a lo largo de la molécula de ADN, transcribiendo los genes estructurales del operón en moléculas de ARNm que luego serán traducidas a las enzimas de degradación de la lactosa (figura 2.9a). En el caso del operón triptófano (*trp*), la presencia del aminoácido activa al represor, que luego se uno al operador y bloquea al síntesis de las enzimas, que ya no son necesarias (figura 2.9b). Tanto la alolactosa como el triptófano, así como las moléculas que establecen interacciones con los represores de otros operones, son efectores alostéricos, que ejercen sus efectos causando un cambio en la configuración de la molécula del represor. Sin embargo, tanto los sistemas inducibles como los represibles son ejemplos de control negativo, dado que ambos involucran represores que interrumpen la transcripción.

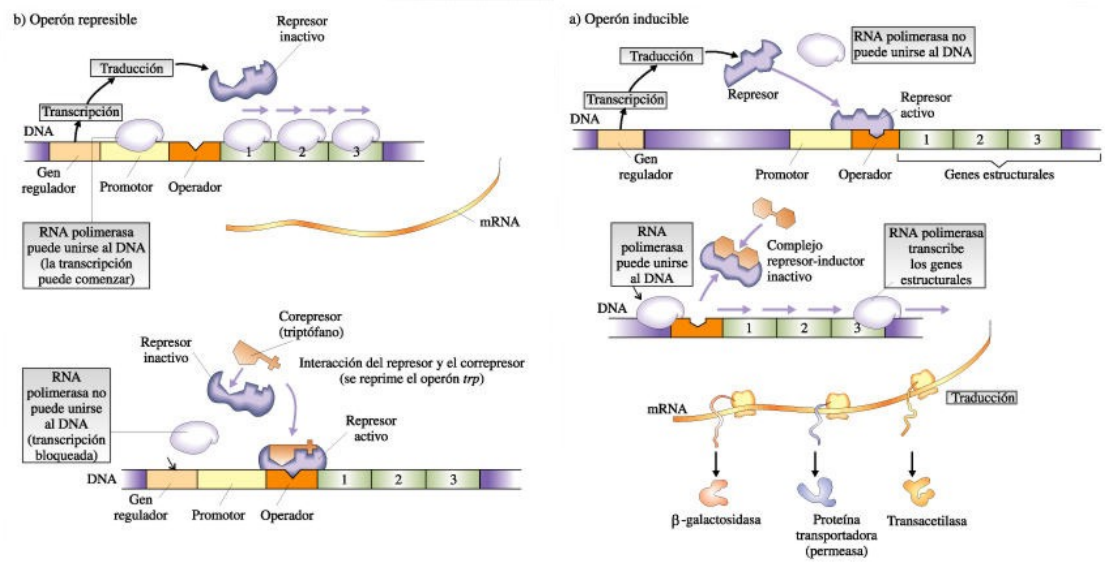


Figura 2.9. Operones inducibles y represibles. Fig. 2.9a: Operón inducible. Fig. 2.9b: Operón represible. Fuente: Curtis *et al.* (2000).

Un ejemplo de control positivo sobre el operón es el de una proteína reguladora, la proteína activadora de los genes catabólicos o CAP. Al igual que el operón mismo, el sistema CAP fue inicialmente investigado en relación con los metabolismos de la lactosa. Se sabe ahora que este sistema es uno de los activadores mas ampliamente usados por *E. coli* y que regula un gran numero de operones en respuesta a las condiciones nutricionales. La CAP se combina con una molécula conocida como AMP cíclica (cAMP), lo que le permite unirse a la región promotora de un operón. Solo entonces, cuando el complejo CAP-cAMP se une al promotor, la transcripción es máxima. Como hemos visto, el operón esta bajo control negativo del represor por lo que no hay transcripción a menos que se separe al represor. Pero también esta bajo el control positivo del complejo CAP-cAMP, que intensifica la transcripción cuando se une al operón.

El descubrimiento de este sistema de control fue una consecuencia de la observación de que *E. coli* no usa lactosa como fuente de energía si hay glucosa presente en el medio. En otras palabras, el operón *lac* permanece reprimido en presencia de glucosa, aunque haya lactosa en la célula. El intermediario de este proceso regulatorio es el cAMP. Cuando disminuye la reserva de glucosa en la célula, el nivel de cAMP aumenta, se forman mas complejos CAP-cAMP y estos quedan disponibles para unirse al operón *lac*. Así, se produce mayor cantidad de las proteínas codificadas por el operón *lac*, y se degrada la lactosa. El cAMP funciona, entonces, como una señal de "hambre". Indica la falta de glucosa y la necesidad de inducir la síntesis de proteínas que actúan en vías metabólicas por las que se obtiene energía a partir de fuentes de carbono alternativas.

Los mecanismo de control positivos y negativos son, por si mismos, ejemplos del modo por el cual la célula viva regula sus actividades bioquímicas. La manipulación de estos operones y sus sistemas de regulación es un componente esencial del ardid científico mediante el cual las células bacterianas son inducidas a sintetizar proteínas de mamíferos de importancia medica, como la somatostatina humana.

2.5.2. Organismos Eucariotas

La regulación de la expresión génica en los procariotas esta relacionada fundamentalmente con el ajuste fino de la maquinaria metabólica de la célula y es llevada a cabo por activación o represión de la expresión de ciertos genes, en respuesta a cambios en los nutrimentos disponibles en el ambiente. En los eucariotas, especialmente en los eucariotas multicelulares, los problemas de regulación son muy diferentes. Un organismo multicelular por lo común inicia la vida en forma de huevo fecundado, el cigoto. El cigoto se divide repetidamente por mitosis y citocinesis, produciendo muchas células. Estas células se diferencia transformándose, por

ejemplo, en células musculares, células nerviosas, células sanguíneas, células intestinales, etc. Cada tipo celular, cuando se diferencia, comienza a producir proteínas característicamente diferentes que lo distinguen de otros tipos de células, como ocurre con la hemoglobina en los glóbulos rojos de los mamíferos. Los genes que codifican estas moléculas de hemoglobina se expresan solamente en los glóbulos rojos inmaduros. A su vez, un mismo tipo celular puede producir variantes de las proteínas que sintetiza en distintas etapas del desarrollo del organismo. En las primeras etapas de la vida fetal, los glóbulos rojos sintetizan un tipo de hemoglobina fetal; los glóbulos rojos producidos en etapas posteriores contienen un segundo tipo de hemoglobina fetal; luego, cierto tiempo después del nacimiento, los glóbulos rojos comienzan a producir las cadenas alfa y beta características de la hemoglobina de los adultos. Así, los diferentes genes se expresan uno tras otro en una secuencia cuidadosamente controlada. Sin embargo, toda la información genética originalmente presente en el cigoto también esta presente en cada célula diploide del organismo. En otras palabras, los segmentos de ADN que codifican hemoglobina (tanto los tipos fetal como adulto) están presentes en las células epidérmicas, en las células cardiacas, en las células hepáticas, en las neuronas y, de hecho, en cada uno de los casi 200 tipos diferentes de células del cuerpo. Dado que cada tipo celular produce solamente sus proteínas características y no las proteínas características de otros tipos celulares, resulta claro que la diferenciación de las células de un organismo multicelular depende de la inactivación de ciertos grupos de genes y de la activación de otros.

La expresión de los genes puede ser regulada en varias de las etapas que conducen desde el ADN a las proteínas. Una célula eucariota regula las proteínas que sintetiza controlando: 1) el momento y la frecuencia con que un determinado gen es transcripto (control transcripcional); 2) el procesamiento del ARNm transcripto (control de procesamiento de ARNm); 3) las moléculas de ARNm que son exportadas del núcleo al citoplasma (control de transporte de ARNm); 4) los ARNm que son traducidos por los ribosomas en el citoplasma (control de traducción); 5) la vida media del ARNm (control de degradación del ARNm) y 6) la activación e inactivación de proteínas (control de la actividad de las proteínas). De todas estas etapas de regulación la primera es la que resulta mas económica para la célula, y es además en donde se pone el foco principal de esta tesis, por lo que será descrita brevemente en los siguientes párrafos.

La regulación de la transcripción en eucariotas es mas compleja que en los procariotas. A diferencia de los procariotas, los genes eucarióticos no están agrupados en operones, en los cuales dos o más genes estructurales se transcriben a una sola molécula de ARN (ARNm policistrónico). En los eucariotas, cada gen estructural se transcribe por separado en un ARNm monocisrónico y su transcripción se encuentra bajo controles separados. Además, para poder iniciar la transcripción, la ARN polimerasa requiere que un grupo de proteínas llamadas factores

generales de transcripción se ensamblen en la región promotora del gen que se va a transcribir. Esto permite la unión de la ARN polimerasa y la posterior transcripción.

Si bien en los procariotas también se da una regulación a este nivel, se hace cada vez mas claro que este nivel de control de la transcripción es mucho mas complejo en los eucariotas, en particular en los eucariotas multicelulares. En un organismo multicelular, un gen parece responder a la suma de muchas proteínas regulatorias diferentes, algunas de las cuales tienden a activar el gen y otras a desactivarlo. Por ejemplo, la unión de los factores generales de transcripción puede ser influenciada por otras proteínas regulatorias que se unen a otros sitios específicos en la molécula de ADN, distintos de la región promotora del gen; esto determina la velocidad con la que se inicia la transcripción. Los sitios en los cuales se unen esas proteínas regulatorias pueden estar a centenares o miles de pares de bases de distancia de la secuencia promotora en la que se une la ARN polimerasa. Entre estas secuencias regulatorias, se cuentan las secuencias denominadas "*enhancers*", a las que se unen las proteínas que activan la transcripción, y pueden encontrarse a miles de pares de bases de distancia del promotor. Hay muchos tipos de proteínas regulatorias, la mayor parte son activadoras de la transcripción y algunas funcionan como represoras. Tanto las proteínas regulatorias como las secuencias de ADN a las que se unen varían de gen a gen. A diferencia de los factores de transcripción, presentes en todas las células, distintos tipos celulares pueden tener distintas proteínas regulatorias que se encuentran en bajas cantidades. La expresión y la actividad de estas proteínas regulatorias responden a señales provenientes del medio externo o a señales internas de la célula. Estas proteínas tienen una región (dominio) de unión al ADN y otro dominio responsable de su función regulatoria que, por ejemplo, acelera o favorece la unión de los factores generales de transcripción al promotor.

2.5.3. Red Regulatoria de Genes Transcripcional

A cierto nivel, las células biológicas se puede considerar como "bolsas parcialmente mezcladas" de productos químicos biológicos. En el contexto de las redes regulatorias de genes, estos productos químicos son en su mayoría los ARNm y las proteínas que surgen de la expresión génica. Como hemos mencionado previamente, estos ARNm y proteínas interactúan entre sí con diferentes grados de especificidad. Algunos se difunden por toda la célula. Otros están ligados a las membranas celulares, interactuando con las moléculas en el medio. Y otros pasan a través de las membranas celulares y median en las señales de largo alcance para otras células en un organismo multicelular. Estas moléculas y sus interacciones constituyen una red regulatoria de genes. En forma general, una red regulatoria de genes es una colección de segmentos de ADN en una célula que interactúan entre sí (indirectamente a través de su ARNm

y proteínas) y con otras sustancias en la célula, con lo que regulan las tasas a las que los genes de la red se transcriben en ARNm. La figura 2.10 muestra la estructura general de una red regulatoria de genes a nivel transcripcional.

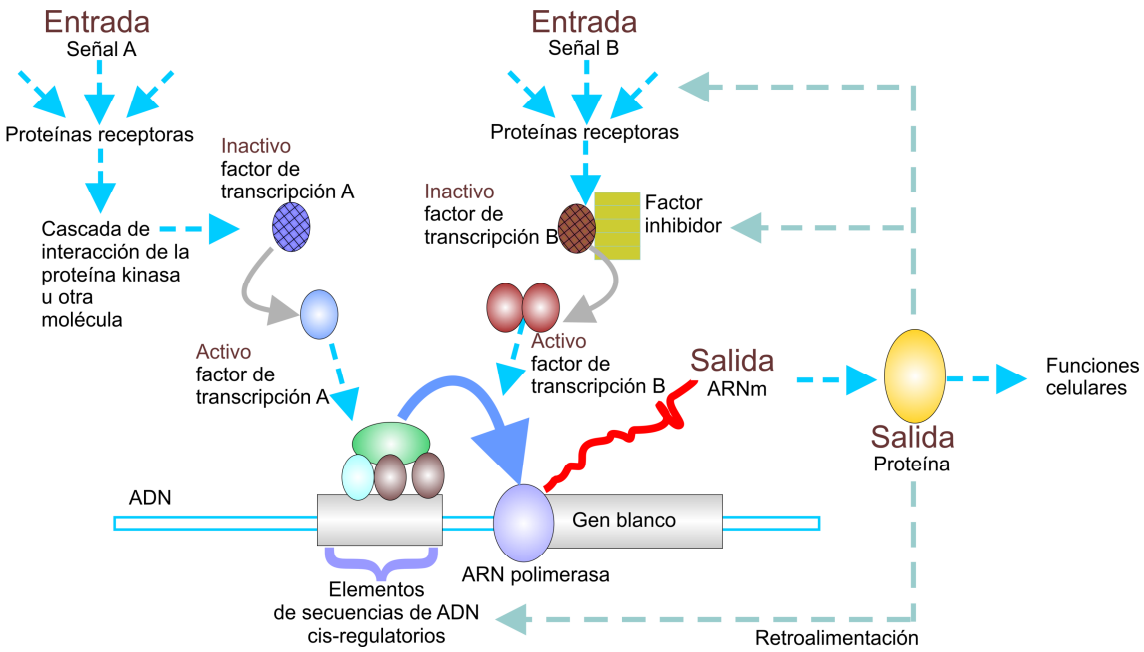


Figura 2.10. Esquema general de una red regulatoria de genes a nivel transcripcional.

2.6. Sumario

Este capitulo ha proporcionado una breve introducción a las características salientes de las células, proteínas, ADN, ARN, regulación génica y a otros conceptos biológicos necesarios para los capítulos siguientes. Esta presentación ha sido largamente extraída de una variedad de libros de texto estándares y sitios Web (Setubal y Meidanis, 1997; Curtis *et al*., 2000; Weaver, 2001; Purves et al., 2003; Zhang, 2006).

Capítulo 3

Microarrays

El reciente desarrollo de tecnologías de *microarrays* de alto rendimiento para ADN posibilita a los investigadores capturar "instantáneas" de las células a nivel de la transcripción en una escala de genoma completo. Los niveles de expresión de miles de genes pueden ser monitoreados usando un solo chip de *microarray*. Antes del advenimiento de estas tecnologías, la examinación de los niveles de expresión de los genes estaba limitada a una mucho menor escala por experimento (Mclachlan, 2004).

La tecnología moderna de *microarray* fue originada a partir del *Sounthern blot* (llamada así por el biólogo británico E. M. Southern), que fue el primer arreglo de material genético. Esta técnica emplea (radiactivamente) sondas etiquetas de ADN (o ARN) a hibridizar para identificar secuencias muy similares de ADN ubicadas en un filtro de nitrocelulosa llamado *blot*. El numero de hebras de ADN que hibridiza con una sonda da un estimado del numero de genes altamente relacionados en un organismo (Weaver, 2001). El *Northern blot* y el *Western blot* son técnicas similares en las cuales se emplean hebras de ARN y proteínas, respectivamente, en lugar de secuencias de ADN. El principio básico detrás de estas técnicas es que las hebras de ADN y ARN pueden ser etiquetadas para detección y luego usadas para sondear otras moléculas de acido nucleico que han sido ligadas a una superficie sólida.

En el libro *Analyzing Microarrays Gene Expression Data* (Maclachlan, 2004) se da una revisión breve de la historia de la tecnología de *microarrays*. En los 80s, un grupo liderado por R.P. Ekins en el *Department of Molecular Endocrinology* de la University *College*, Inglaterra, fue el primero en usar técnicas simples de *microspotting* en la manufactura de arreglos para estudios de inmunoensayos de alta sensibilidad (Ekins y Chu, 1999). Numerosos grupos de investigadores han promovido la tecnología introducida por Ekins y sus colegas. En los Estados Unidos, notables avances fueron logrados por Stephen P.A. Fodor y sus colegas en Affymetrix, Inc. (Santa Clara, California) (Fodor *et al.*, 1991), así como también por grupos en *Stanford University*, particularmente Patrick O. Brown, del *Department of Biochemistry and Biophysics* (Schema *et al.*, 1995). A Brown y sus colegas en Stanford se le atribuyen la creación del primer chip de *microarray* de ADN, mientras que Stephen Fodor y sus colegas en Affymetrix, Inc., crearon el primer chip patentado de *microarray* en oblea de ADN, el GeneChip. Numerosas entidades comerciales y grupos académicos han contribuido desde entonces con avances en la tecnología de *microarrays*.

En la forma más general, un arreglo de ADN es un chip hecho de una membrana de nylon, vidrio o plástico. Usualmente, el chip esta organizado en un patrón de grilla regular y contiene segmentos de hebras de ADN que son depositadas o sintetizadas en cada grilla individual. Una vez que el arreglo esta preparado, realizar el experimento de *microarray* requiere de tres etapas básicas: preparación de muestras y etiquetado, hibridación de muestras y lavado, y escaneo y procesamiento de las imágenes de *microarrays*.

En este capitulo, primero se introducirán y compararan varias tecnologías disponibles para la manufactura de *microarrays*, para luego describir los pasos involucrados en un experimento típico de *microarray*. Además, dado que los datos crudos resultantes de un experimento de *microarray* contienen ruido, valores faltantes y sesgo experimental, es necesario aplicarles técnicas de preprocesamiento y normalización antes de que cualquier análisis pueda realizarse. Estos tópicos también serán discutidos al final del capítulo.

3.1. Manufactura de Chips de Microarrays

Hay principalmente dos enfoques para la manufactura de chips de *microarrays*: deposición de fragmentos de ADN por medio de punteo robótico y síntesis *in situ* (Lander, 1999). La manufactura por deposición robótica puede hacerse por medio de deposición de clones ADNc amplificados mediante PCR o mediante la impresión de oligonucleicos previamente sintetizados. La fabricación *in situ* puede dividirse en fotolitografía, impresión por chorro de tinta y síntesis electroquímica (Draghici, 2003).

En esta sección se describirán los dos chips actualmente más usados, el *microarray* de ADNc y el Affymetrix GeneChip, como ejemplos para la manufactura de *microarrays* basada en deposición e *in situ* respectivamente.

3.1.1. Manufactura basada en deposición

La manufactura de arreglos basados en deposición involucra la consideración de 3 cuestiones: la selección de las "sondas"[1] de ADN, preparación de las sondas, y el proceso de impresión. Consideremos primero cuales "sondas" van a ser impresas en el arreglo. En muchos casos, estas son elegidas directamente de bases de datos incluyendo GenBank (Benson *et al*, 2002), dbEST (Boguski *et al*., 1993), y UniGene (Schuler *et al.*, 1996), recursos que son la espina dorsal de la tecnología de arreglos (Boguski *et al*., 1993; Duggan *et al.*, 1999). Adicionalmente se pueden usar cadenas completas de ADNc, colecciones de ADNc parcialmente secuenciados (o ESTs), o elecciones aleatorias de ADNc de cualquier librería de interés (Duggan *et al.*, 1999).

[1] A través de esta sección, se referirá al ADN del arreglo como sonda y al ADN etiquetado en la solución como molécula diana o blanco.

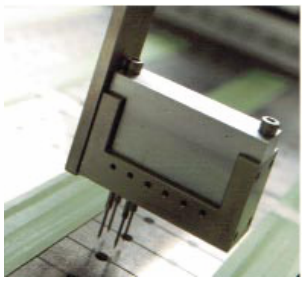
En el proceso de manufactura basada en deposición, la sondas de ADN son preparadas separadas del chip. Las sondas pueden ser productos de reacción en cadena de la polimerasa (PCR en inglés) o oligonucleótidos. La técnica de PCR fue desarrollada en 1983 por medio del trabajo de Kary B. Mullis y sus colegas en la Cetus Corporation en Emeeruville, Calirfornia (Mullis, 1990). Esta técnica crea millones de copias de fragmentos específicos de ADN a partir de una sola molécula de ADN. Luego de la amplificación, el producto resultante del PCR es parcialmente purificado por precipitación y/o filtración por gel para remover sustancia no deseadas como sales, detergentes, PCR *primers* y proteínas presentes en el coctel PCR (Duggan *et al.*, 1999). Alternativamente, las sondas de ADN pueden prepararse presitentizando oligonucleótidos de ADN para usar en el arreglo.

Una vez que las sondas de ADN están determinadas y preparadas, el proceso típico de impresión sigue cinco pasos básicos:

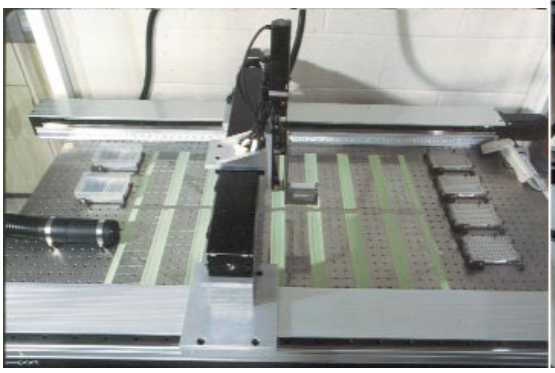
1. Pequeños pines robóticos se sumergen en los recipientes de las soluciones para recolectar el primer lote de ADN.
2. Los pines tocan la superficie de los arreglos para depositar el ADN. Usualmente, el ADN es depositado en un número variado de arreglos, dependiendo del número de arreglos a hacer y de la cantidad de liquido que los pines pueden mantener.
3. Posteriormente, los pines son lavados para remover cualquier solución residual y asegurar que no habrá contaminación en la próxima muestra.
4. Los pines se sumergen en el próximo conjunto de recipientes y se repite el proceso hasta que el arreglo está completo (Stekel, 2003).

La figura 3.1 ilustra algunos instrumentos empleados en la fabricación de *microarrays*.

(a)



(c)



(b)

(d)

Figura 3.1. Instrumentos para *microarrays*. (a) Robot de *microarrays* en la University of Pennsylvania. (b) Robot de *microarrays* en el Albert Einstein College of Medicine (AECOM). (c) Cuatro de los doce posibles pines en uso. (d) El scanner de laser de AECOM. Fuente: Cheung (1999).

3.1.2. Manufactura *In Situ*

Los arreglos sintetizados *in situ* son fundamentalmente diferentes de los arreglos basados en disposición en los siguientes aspectos (Draghici, 2003).

- **Selección de sondas.** La selección de sondas se realiza basándose solamente en la información de la secuencia. Entonces, cualquier sonda sintetizada en el arreglo es conocida. En contraste, con los arreglos ADNc, que lidian con etiquetas de secuencias expresadas, la función de la secuencia correspondiente a una dada ubicación en el arreglo es muchas veces desconocida. Además, dado que este método de selección evita duplicar secuencias idénticas entre los miembros de una misma familia de genes, esta técnica puede distinguir y monitorear cualitativamente genes altamente relacionados.
- **Preparación de las sondas.** Las sondas son sintetizadas fotoquímicamente base por base en la superficie del arreglo. No hay clonamiento ni proceso de PCR involucrado.

- **Proceso de impresión**. Dado que las sondas son sintetizadas en la superficie del arreglo, no se necesita el proceso de impresión. La eliminación del clonado, amplificación e impresión de ADN reduce muchas fuentes de ruido potencial en los sistemas de ADNc y entonces constituye un avance de este enfoque en la fabricación de arreglos.

La síntesis *in situ* toma lugar por la reacción covalente entre el 5' grupo hidroxilo del azúcar del último nucleótido a ser agregado en el arreglo y el grupo fosfato del siguiente nucleótido. Cada nucleótido añadido a la sonda oligonucleótida anclada al vidrio del chip, tiene un grupo de protección en su posición 5' para prevenir la adición de mas de una base durante el proceso de síntesis. Este grupo de protección es luego convertido a un grupo hidroxilo, usando acido o luz, antes de la siguiente ronda de síntesis (Stekel, 2003).

3.1.2.1. El GeneChip de Affymetrix

En Affymetrix, la construcción de arreglos de sondas de ADN de alta densidad se basa en la síntesis mediante luz directa usando dos técnicas: la fotolitografía (parecida a la tecnología usada para construir circuitos integrados de gran escala (VLSI)) y la síntesis en fase sólida de ADN (Lipshutz, 1999). Descripta en forma simple, la técnica de Affymetrix usa luz para convertir los grupos de protección del nucleótido terminal en un grupo hidroxilo al cual se le pueden adicionar nuevas bases. La luz es dirigida a través de una mascara de fotolitografía que permite a la luz pasar a áreas especificas del arreglo pero no a otras. De esta forma, una base específica puede ser añadida a una sonda de una ubicación específica. Una serie de estas máscaras permite la construcción de secuencias base por base (Draghici, 2003; Stekel, 2003) (ver figura 3.2).

Los arreglos de expresión de genes usan una estrategia de sondas coincidente/discordantes (Lipshutz *et al.*, 1999) (ver figura 3.3). Las sondas que coinciden exactamente con la secuencia diana (o blanco) son llamadas sondas de referencia. Para cada sonda de referencia, hay una sonda de discordancia correspondiente que contiene una alteración de un nucleótido en la posición de la base central. Estas dos sondas, la de referencia y la de discordancia, son siempre sintetizadas una adyacente a la otra para controlar diferencias espaciales en la hibridación. Para aumentar el nivel de confidencia en la detección de la señal, cada gen tiene varios pares de sondas de referencia/discordancia, donde cada par corresponde a varias partes del gen (Draghici, 2003). El uso las diferencias promedio de PM (*perfect match*) menos MM (*mismatch*) entre un conjunto de sondas para cada gen reduce altamente la contribución del entorno y de la hibridación cruzada, e incrementa la precisión cuantitativa y la reproducibilidad de las mediciones (Lipshutz *et al*, 1999).

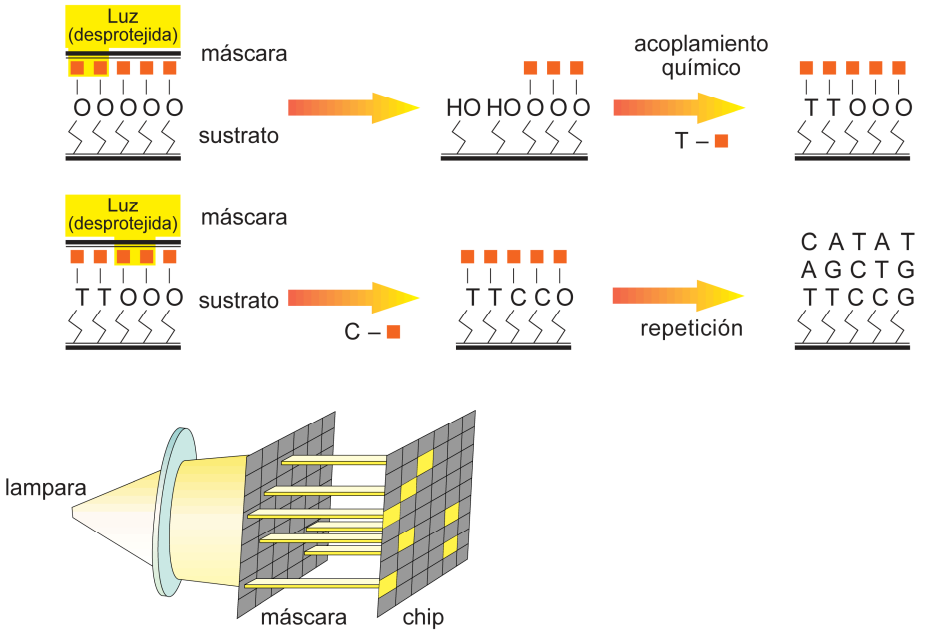


Figura 3.2. Síntesis de oligonucleótidos por medio de luz dirigida. Un sólido de soporte es preparado con un enlace molecular covalente terminal con un grupo protectivo fotolábil. La luz es dirigida a través de la máscara para desproteger y activar los sitios seleccionados, y haciendo que los nucleótidos protegidos se acoplen a estos sitios activados. Es proceso es repetido, activando diferentes sitios y acoplando diferentes bases permitiendo la construcción de sondas de ADN arbitrarias en cada sitio. Fuente: Lipshutz *et al*. (1999).

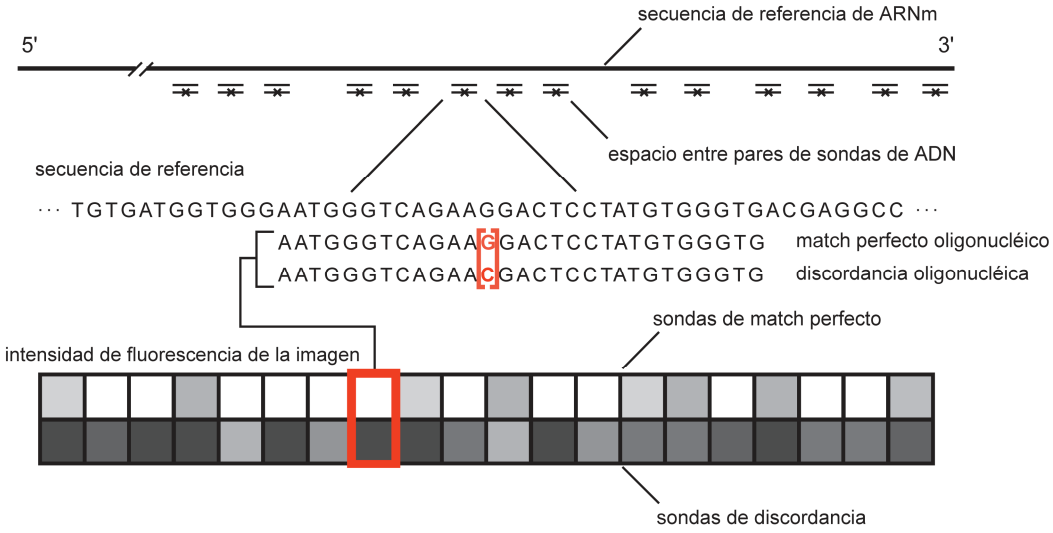


Figura 3.3. Sonda de expresion y diseño de arreglo de Affymetrix. Las sondas oligonucleótidas son elegidas basandose en criterios de unicidad y en reglas de composición. Para los organismos eucariotas, las sondas son eleguidas tipicamente a partir del lado 3' del gen o transcriptor para reducir los problemas que puedan ocurrir a partir del uso de ARNm parcialmente degradado. El uso de las diferencias promedio de PM menos MM entre un conjunto de sondas para cada gen reduce altamente la contribución del entorno y de la hibridación cruzada, e incrementa la precisión cuantitativa y la reproducibilidad de las mediciones. Fuente: Lipshutz *et al*. (1999).

3.2. Pasos en los experimentos de microarrays

Independientemente de la tecnología específica usada, un experimento de *microarray* consiste en tres pasos básicos: preparación de las muestras y etiquetado, hibridación de las muestras y lavado y, escaneo de las imágenes de *microarray* y procesamiento. En esta sección, los *microarrays* ADNc serán usados como base para la discusión general de estos pasos. En general, los experimentos basados en otras plataformas, como el GeneChip de Affymetrix, siguen principios similares a menos que se especifique lo contrario. El esquema general de un *microarray* de ADNc esta ilustrado por la figura 3.4.

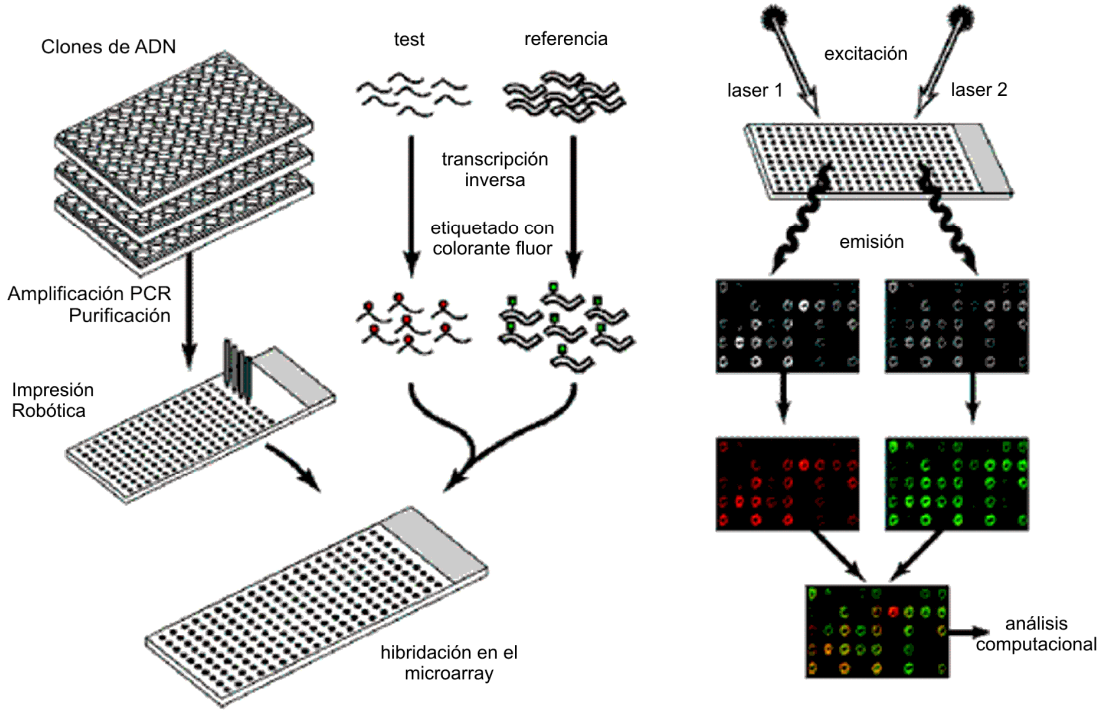


Figura 3.4. Esquema general de un *microarray* de ADNc. Fuente: Duggan *et al*. (1999).

3.2.1. Preparación de muestras y etiquetado

La preparación de muestras involucra la extracción y purificación de ARNm de tejidos de interés. Debido a un número de desafíos, este procedimiento puede ser muy variable (Amaratunga y Cabrera, 2003; Stekel, 2003). Primero, el ARNm diana típicamente forma parte de una pequeña fracción (menos del 3%) de todo el ARNm en la célula. Segundo, puede ser muy difícil aislar ARNm específico para el estudio a partir de un conjunto heterogéneo de células (por ejemplo, los tejidos enfermos contienen una mezcla de tejidos normales, células inflamadas, tejido necrótico y, en muestras de cáncer, áreas de diferentes grados). Finalmente, el ARNm capturado se degrada muy rápidamente. Para hacer frente a esta rápida degradación, el ARNm es usualmente trascripto a ADNc (para *microarrays* ADNc), que es mas estable, inmediatamente después de la extracción (Amaratunga y Cabrera, 2003).

Para permitir la detección de cuales ADNc están ligados al *microarray*, las muestras van a través de un proceso de etiquetado dependiente de la plataforma. Para la plataforma Affymetix, se contruye ARN complementario etiquetado con biotina para hibridizar con el GeneChip. Los protocolos son cuidadosamente definidos por Affymetrix para asegurar que cada laboratorio Affymetrix siga los mismos pasos. Así, los resultados experimentales de diferentes laboratorios Affymetrix son comparables entre si (Stekel, 2003).

La detección de ADN con *microarrays* de ADNc ha sido realizada anteriormente con ADN etiquetado radiactivamente, pero ahora es mas común el uso de colorantes que son fluorescentes cuando son expuestos a una longitud de onda especifica de luz. En los experimentos mas comunes, dos muestras son hibridizadas en el arreglo, cada una etiquetada con colorantes Cy3 o Cy5, que son excitados por láseres verdes y rojos, respectivamente (Amaratunga y Cabrera, 2003). Esto resulta en un *microarray* ADNc de dos canales y permite medir simultáneamente ambas muestras. También es posible que sean usadas más de dos muestras etiquetadas, produciendo un *microarray* de ADNc multicanal.

3.2.2. Hibridación

La hibridación es el paso en el cual las sondas de ADN en los *microarrays* y el ADN diana etiquetado (o ARN) forma cadenas heterogéneas de acuerdo a la regla de apareo de bases de Watson-Crick (Stekel, 2003). El principio esencial aquí es que una molécula de ADN de una sola hebra se ligara a otra molécula de ADN de una sola hebra con una secuencia de macheo precisa con mucho mas afinidad que a una secuencia con un macheo impreciso (Amaratunga y Cabrera, 2003). En realidad, sin embargo, la hibridación es un proceso complejo y un segmento de ADN podría también ligarse a una secuencia similar pero no idéntica a su diana complementaria, dando lugar al fenómeno llamado hibridación cruzada. Esto es influenciado por muchas condiciones, incluida la temperatura, humedad, concentración salina, volumen de la solución diana y operador de hibridación (Stekel, 2003).

La hibridación también puede ser realizada manualmente o por un robot. La hibridación robótica provee un mucho mejor control sobre la temperatura de la diana y de la filmina. El uso consistente de una sola estación de hibridación también reduce la variabilidad que emerge de múltiples hibridaciones y varios operadores (Sketel, 2003). Después de la hibridación, el *microarray* es removido de la cámara o estación y luego es lavado para eliminar cualquier exceso de muestras de etiquetado, para que solo el ADN complementario de las sondas permanezca ligado al arreglo. Finalmente, el *microarray* es secado usando una centrifugadora o por aplicación de aire limpio comprimido.

3.2.3. Escaneo de Imagen

Después de completar la hibridación, la superficie del arreglo hibridizado es escaneada para producir una imagen de *microarray*. Como se mencionó previamente, las muestras son etiquetadas con biotín o colorantes fluorescentes que emiten luz detectable cuando son estimuladas por láser. La luz emitida es capturada por un tubo foto multiplicador (*photo-multiplier tube* o PMT) en un escáner, y la intensidad es grabada. Muchos escáneres contienen uno o mas lásers que son enfocados en el arreglo (para los *microarrays* de dos canales, el escáner usa dos lásers) (Stekel, 2003).

Aunque el escáner se emplea para detectar la luz emitida por las hebras de ADN diana que están ligadas a su sonda complementaria, también puede capturar la luz de otras fuentes. Estas otras fuentes incluyen muestras de ADN etiquetado que se ha hibridizado accidentalmente a la filmina de vidrio, muestras etiquetadas residuales (sin lavar) que se han adherido al vidrio, varios químicos usados en el procesamiento de la filmina, e incluso la filmina misma. Esta luz capturada incidentalmente es llamada *background* (Amaratunga y Cabrera, 2003).

El escaneo de la salida de un chip Affymetrix es usualmente una imagen monocromática (ver figura 3.5). Con un *microarray* de dos canales, la salida es un par de imágenes monocromáticas. Cada imagen es de uno de los lásers en el escáner. Las dos imágenes monocromáticas son luego combinadas para crear las imágenes a color falsas de los *microarrays* (ver figura 3.6). Las imágenes monocromáticas y las de dos colores son usualmente almacenas en formato TIFF (tagged image file format).

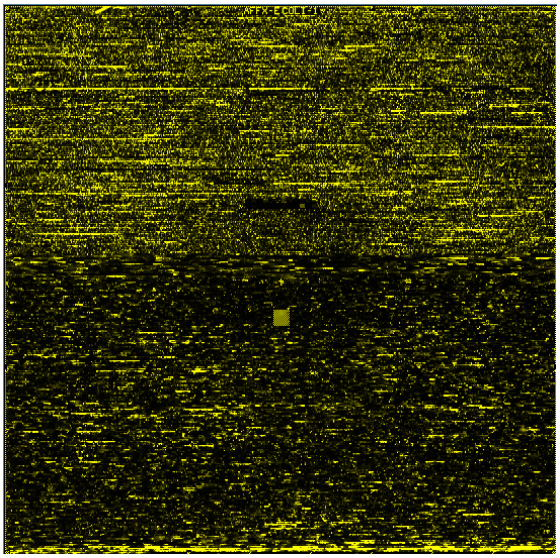


Figura 3.5. Ejemplo de una imagen Affymetrix. Fuente: http://arep.med.harvard.edu/rna_decay/.

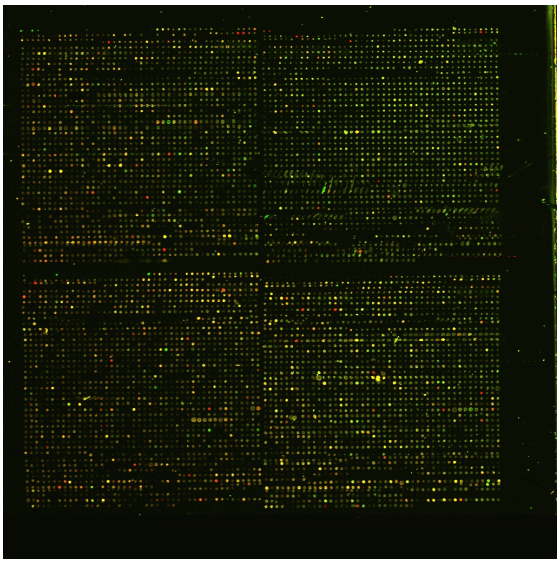


Figura 3.6. Imagen a color falsa resultante para un *microarray* de ADNc. Los puntos verdes corresponden a los puntos mas expresados en el canal uno. Los puntos rojos corresponden a aquellos mas expresados en el canal dos. Los puntos amarillos tienen un nivel de expresión similar en ambos canales. Los puntos negros tienen un nivel de expresión bajo en ambos canales. Fuente: http://genome-www.stanford.edu/cellcycle/.

3.3. Procesamiento de imagen

La imagen de *microarray* generada por el escáner conforma el dato crudo de un experimento de *microarray*. Antes del análisis de los datos, la imagen debe ser convertida del formato TIFF a información numérica que cuantifique la expresión de genes. La manera en la que esto se lleva a cabo tiene mucho impacto en la calidad de los datos resultantes y en el éxito de los análisis posteriores. Para la síntesis *in situ* de *microarrays*, tanto Affymetrix como Agilen han integrado adaptaciones de algoritmos de procesamiento de imágenes en sus paquetes de software, permitiendo a los usuarios finales generar directamente los datos de *microarray* cuantificados.
Para los *microarrays* de ADNc, las imágenes consisten en puntos (*spots*) ordenados en un patrón de grilla regular. El procesamiento de estas imágenes consiste de cuatro pasos básicos:

- **Identificación del punto.** El proceso de identificación de los puntos involucra localizar la posición de los puntos de señal individuales en una imagen y estimar su tamaño.
- **Segmentación de la imagen.** Este proceso involucra la descomposición de la imagen en dos conjuntos de regiones no solapadas. Específicamente, este paso involucra la diferenciación de aquellos píxeles que forman el punto y que deben incluirse en el calculo de la señal de los otros píxeles que forman parte del background o son ruido y que deben eliminarse.
- **Cuantificación del punto.** Después de que los píxeles pertenecientes a la señal y al *background* han sido distinguidos, este proceso involucra calcular la intensidad de cada

punto. Aquí, los valores de intensidad de cada píxel son combinados en un único valor representando el nivel de expresión de un gen depositado en un dado lugar.

- **Evaluación de la calidad del punto.** Este proceso involucra calcular alguna medida de control de calidad para evaluar la calidad tanto del arreglo entero como de los puntos individuales del arreglo. Estas mediciones pueden ayudar a los inspectores humanos en la determinación de la confiabilidad de los datos y en la identificación de aquellos puntos con valores de una calidad cuestionable.

Los datos numéricos obtenidos a partir de la imagen usualmente se pueden representar por una matriz de dos dimensiones $\boldsymbol{X}$, donde cada fila i en la matriz de datos corresponde a un gen, cada columna j corresponde a una condición experimental, y cada celda x_{ij} es un valor real representando el nivel de expresión del gen i bajo la condición j.

3.4. Limpieza, Preprocesamiento y Normalización de Datos de Microarray

3.4.1. Transformación de los datos

Es una práctica común transformar los datos de ADN de *microarrays* de intensidades crudas a intensidades logarítmicas antes de proceder con el análisis. Hay varios objetivos en esta transformación (Stekel, 2003):

- Debe haber una dispersión razonablemente uniforme de las características entre los rangos de intensidades.
- La variabilidad debe ser constante entre todos los niveles de intensidades.
- La distribución experimental de los errores debe ser aproximadamente cero.
- La forma de la distribución de las intensidades debe ser aproximadamente acampanada.
-

El análisis de datos de *microarray* típicamente usa logaritmos en base 2 (Stekel, 2003). En el procesamiento, los ratios crudos de las intensidades Cy5 y Cy3 son transformados en la diferencia entre los logaritmos de las intensidades de los canales Cy5 y Cy3. Entonces, los genes activos regulados con un *fold* de 2 corresponden a un ratio logarítmico de +1 mientras que los genes inhibidos regulados con un *fold* de 2 corresponden a un ratio logaritmo de -1. Los genes que no están diferencialmente expresados tienen un ratio logarítmico de 0. Estos ratios logarítmicos tienen una simetría natural que refleja la estructura biológica y no esta presente en las diferencias de los datos crudos.

3.4.2. Estimación de valores faltantes

Los experimentos de *microarrays* muchas veces generan conjuntos de datos con múltiples valores faltantes. Los valores faltantes ocurren por diversas razones, incluyendo resolución insuficiente, corrupción de imagen, o contaminación de la filmina por polvo o rayas. Los datos faltantes pueden también ocurrir sistemáticamente como resultado de los métodos robóticos empleados para generar los *microarrays* (Mclachlan, 2004). Desafortunadamente, muchos algoritmos para el análisis de expresión de genes requieren el conjunto de datos completo como entrada. Entonces, se necesitan métodos para estimar los datos faltantes antes de que estos algoritmos puedan ser aplicados.

Suponga que un conjunto de datos de *microarray* esta representado por una matriz donde cada fila corresponde a un gen y cada columna corresponde a una condición experimental. Un simple enfoque para imputar los datos faltantes es reemplazar cada entidad faltante con la expresión promedio sobre la fila (el método del promedio de fila). Este método no es óptimo dado que no toma en cuenta la correlación estructural completa del conjunto de datos. EnTroyanskaya *et al*. (2001), se proponen dos algoritmos mas complejos basados en los K vecinos mas cercanos (*KNN impute*) y en Descomposición en Valores Singulares (*SVD impute*). También evalúan el desempeño de estos dos algoritmos y del método del promedio de fila.

Imputación basada en los K vecinos más cercanos (KNN). En términos simples, el algoritmo de imputación KNN estima los valores faltantes eligiendo K genes con los perfiles de expresión mas similares al gen de interés. Suponga que, para el gen i, el valor de expresión x_{ij} es un dato faltante en el j-ésimo experimento. El algoritmo selecciona K genes con valores no faltantes para el experimento j que tienen el perfil de expresión mas cercano al gen i en los experimentos restantes. Luego, el promedio pesado de los valores de los K genes en el experimento j es usado como un estimador de x_{ij}. Métricas como la distancia Euclidiana o el coeficiente de correlación de Pearson pueden ser usadas para medir la similaridad entre los perfiles de genes. En Troyanskaya *et al*. (2001), los autores encontraron que la distancia Euclidiana fue una métrica suficientemente precisa para los datos transformados logaritmicamente. En el promedio pesado, la contribución de cada gen es pesada por la distancia o similaridad de su expresión para con el gen i.

Imputación basada de Descomposición en Valores Singulares (Singular Value Decomposition o SVD en ingles). Este método primero imputa todos los valores en la matriz $\boldsymbol{X}$ usando el método del promedio de fila. Luego se aplica la Descomposición en Valores Singulares para producir un conjunto de patrones mutualmente ortogonales de expresión llamados *eigengenes*. Estos eigengenes pueden combinarse linealmente para aproximar la

expresión de los genes en una matriz de datos de *microarray* $\boldsymbol{X}$ de $n \times m$, donde n y m son el numero de genes y experimentos, respectivamente.

Para ser específicos, la descomposición en valores singulares de $\boldsymbol{X}$ es:

$$X = U\Sigma V^T . \tag{3.1}$$

Las columnas de V^T forman los eigengenes de $\boldsymbol{X}^T\boldsymbol{X}$, cuya contribución a la expresión en el eigenespacio esta cuantificada por los correspondientes eigenvalores en la matriz diagonal Σ. Los k eigengenes mas significativos son seleccionados para formar la base del proceso de imputación. El valor de k usualmente se determina empíricamente. El proceso de imputación involucra una regresión de los valores faltantes x_{ij} respecto de los k eigengenes seleccionados (mientras se ignoran todos los valores correspondientes al experimento j). Esto es, el x_{ij} faltante es obtenido a partir de una combinación lineal de los k eigengenes pesados por los coeficientes de la regresión. Este proceso es iterado hasta que el cambio total en la matriz $\boldsymbol{X}$ converge a un valor arbitrario suficientemente pequeño.

En Troyanskaya *et al*. (2001), los autores compararon el rendimiento de la imputación por KNN y SVD, y el método del promedio de fila en términos de complejidad computacional y precisión de la estimación. Ellos concluyeron que aunque el método del promedio de fila es el más rápido, no funciona bien en términos de precisión, recomendando asi al método de imputación por KNN como el más robusto respecto del incremento en la fracción de datos perdidos. Sin embargo, también sugieren tener cuidado con la elaboración de conclusiones biológicas criticas a partir de datos que están parcialmente imputados; los datos estimados deben ser marcados en donde sea posible y su significancia en la formulación de conclusiones biológicas debe ser evaluada para evitar supuestos no garantizados.

3.4.3. Normalización de los datos

La complejidad del proceso de experimentación con *microarrays* muchas veces introduce un sesgo sistemático en la intensidad de las mediciones. Entre otras fuentes de variabilidad, el sesgo sistemático puede ser causado por la concentración y la cantidad de ADN puesto en los *microarrays*, desgaste en los equipos, cantidades de ARNm extraído de las muestras, sesgo en la transcripción inversa, falta de homogeneidad espacial entre las filminas, configuración del escáner, efectos de saturación, condiciones ambientales, etc. (Amaratunga y Cabrera, 2003).

Además, hay un sesgo en el colorante que esta presente en la mayoría de los experimentos multicanales. Generalmente, las intensidades del Cy5 (rojo) tienden a ser más altas que las intensidades del Cy3 (verde), pero la magnitud de la diferencia generalmente depende de la

intensidad global (Amaratunga y Cabrera, 2003). Las razones para el desbalance de los canales son estas (Stekel, 2003):

- Las etiquetas Cy3 y Cy5 pueden ser incorporadas diferenciadamente en las muestras de ADN con una frecuencia de ocurrencia variable.
- Los colorantes Cy3 y Cy5 pueden tener distinta respuesta de emisión para con la excitación del láser con una frecuencia de ocurrencia diferente.
- Las emisiones Cy3 y Cy5 pueden ser medidas diferencialmente por el tubo foto multiplicador a diferentes intensidades.
- Las intensidades Cy3 y Cy5 medidas en varias áreas del arreglo pueden diferir debido a alguna inclinación en el arreglo lo cual resulta en variaciones en el foco.

El propósito de la normalización es remover los efectos de cualquier fuente sistemática de variación lo más posible. Para los *microarrays* de Affymetrix, la normalización permite la comparación directa de los niveles individuales de expresión de los genes en un arreglo. Para un *microarray* multicanal (como los *microarrays* ADNc), la normalización puede ser aplicada para ajustar el sesgo entre los múltiples canales.

En general, los métodos de normalización pueden ser divididos en esquemas de normalización global y enfoques de normalización dependientes de la intensidad. Los primeros asumen que los puntos de intensidad de cada par de arreglos a ser normalizados están linealmente relacionados. Entonces, la falta de comparabilidad puede ser corregida ajustando cada punto individual de intensidad en el mismo arreglo o en el canal por medio de una cantidad idéntica llamada factor de normalización. Algunos ejemplos de este tipo de normalización son la estandarización y regresión lineal iterativa (Draghici, 2003). Por contraste, los métodos de normalización dependientes de la intensidad determinan el factor de normalización para diferentes puntos de acuerdo a su intensidad individual. Entonces, la normalización se basa en una función no lineal dependiente de la intensidad $X \rightarrow f(X)$ (Amaratunga y Cabrera, 2003). Algunos ejemplos de este tipo de normalización son el *Locally Weighted Linea Regression* (LOWESS) (Cleveland, 1979) y la normalización de la distribución (Bolstad *et al*, 2003).

3.5. Aplicaciones de Microarrays

Los *microarrays* han sido usados extensamente en la investigación biológica para resolver una amplia variedad de preguntas. Tal como es mencionado en Collins (1999), cuando son aplicados al análisis de expresión los *microarrays* permiten las mediciones de los niveles de ARNm para el conjunto completo de transcriptores de un organismo. Cuando son aplicados a

genotipado, estos ayudan en la posibilidad de determinar alelos de cientos de miles de locus a partir de cientos de muestras de ADN, permitiendo la contemplación de estudios de asociación de genoma completos para determinar la contribución genética en desordenes poligenéticos complejos. Es mas, la aplicación de *microarrays* a la detección de mutaciones de los genes relacionados con enfermedades con pronunciada heterogeneidad alélica probablemente mueva la posibilidad del testeo genético de la susceptibilidad a enfermedades de un individuo, o incluso de una población entera, al reino de la realidad práctica. En esta tesis nos enfocaremos en el análisis de datos de expresión de genes, específicamente en el análisis de genes con comportamiento similar y en la inferencia de redes regulatorias de genes.

3.6. Sumario

En este capitulo, hemos provisto con una descripción general de la generación y procesamiento de datos de expresión de genes de *microarrays*. Los materiales en este capitulo han sido largamente extraídos de varias publicaciones y sitios Web los cuales discuten estos tópicos (Amaratunga y Cabrera, 2003; Draghici, 2003; Stekel, 2003). Como hemos visto, la tecnología de *microarrays* ofrece una forma eficiente de medir los niveles de expresión de miles de genes en un solo experimento, entre diferentes condiciones experimentales y a lo largo del tiempo (DeRisi *et al.*, 1996; Heller *et al.*, 1997; Chen *et al.*, 1998;Browm *et al.*, 2000). El foco experimental puede incluir tipos de cáncer, organismos enfermos, o tejidos normales. Los arreglos son ahora comunes en la investigación biomédica para los perfiles de expresión de ARNm y son usados para explorar los patrones de expresión génica en la investigación clínica (Welford *et al*., 1998; Iyer *et al.*, 1999; Brazma y Vilo, 2000). La aplicación de esta tecnología en la investigación de los niveles de respuesta de los genes en los tratamientos con drogas, tiene el potencial de proveer una visión profunda dentro de la naturaleza de muchas enfermedades y una guía hacia el desarrollo de nuevas drogas.

Capítulo 4

Biclustering de Matrices de Expresión de Genes

Las matrices de expresión de genes han sido analizadas extensamente en sus dos dimensiones: la dimensión de los genes y la dimensión de las condiciones. Estos análisis corresponden, respectivamente, a analizar los patrones de expresión de los genes comparando las filas en la matriz y, a analizar los patrones de expresión de las condiciones comparando las columnas de la matriz.

Entre los objetivos perseguidos al analizar los datos de expresión de genes se incluyen:

1. Agrupar los genes acorde a su expresión bajo múltiples condiciones
2. Clasificación de un nuevo gen, dada la expresión de otros genes con clasificación conocida.
3. Agrupar condiciones basado en la expresión de cierto numero de genes
4. Clasificación de una nueva muestra, dada la expresión de los genes bajo esa condición experimental.

Las técnicas de agrupamiento (*clustering,* utilizaremos el término en inglés de aquí en adelante) pueden ser usadas para agrupar tanto genes como condiciones y, entonces, persiguen directamente los objetivos 1 y 3 mencionados antes e, indirectamente, los objetivos 2 y 4. Sin embargo, aplicar los algoritmos de *clustering* sobre los datos de expresión de genes conlleva dificultades significativas. Muchos patrones de activación son comunes a un grupo de genes solo en condiciones experimentales específicas. De hecho, el entendimiento general de los procesos celulares lleva a esperar que subconjuntos de genes estén co-regulados y co-expresados solo bajo ciertas condiciones experimentales, pero que se comporten casi independientemente bajo otras condiciones. El descubrimiento de esos patrones locales de expresión puede ser la llave para dilucidar muchas vías génicas(*pathways*) que no serian aparentes de otra manera. Entonces, es altamente deseable moverse mas allá del paradigma de *clustering* y desarrollar enfoques capaces de descubrir patrones locales en los datos de *microarrays* (Ben-Dor *et al.*, 2002).

El termino *biclustering* fue usado primero por (Cheng y Church, 2000) en el análisis de datos de expresión de genes. Se refiere a una clase de algoritmos de *clustering* que realizan agrupamiento simultáneo en las filas y columnas. Los algoritmos de *biclustering* también han sido propuestos y usados en otros campos de aplicación. Nombres como *co-clustering*,

agrupamiento bidimensional, agrupamiento subespacial, entre otros, a menudo son usados en la literatura para referir a la misma formulación del problema. Una de las primeras formulaciones de *biclustering* es el algoritmo de *clustering* directo introducido por (Hartigan, 1972), también conocido como *clustering* en bloque (Mirkin, 1996).

A partir de la introducción de este concepto, surgen naturalmente varias preguntas, tales como: ¿cuál es la diferencia entre *clustering* y *biclustering*? ¿por qué y cuando deberíamos usar *biclustering* en vez de *clustering*? El *clustering* puede ser aplicado a filas o a columnas de la matriz de datos, en forma separada. Por otro lado, el *biclustering* realiza *clustering* en estas dos dimensiones simultáneamente. Esto significa que el *clustering* deriva un modelo global mientras que el *biclustering* produce un modelo local. Cuando se utilizan los algoritmos de *clustering*, cada gen en un dado *cluster* de genes esta definido usando todas las condiciones. En forma similar, cada condición en un *cluster* de condiciones esta caracterizada por la actividad de todos los genes que pertenecen a la matriz de datos. Sin embargo, cada gen en un *bicluster* esta caracterizado solo por un subconjunto de condiciones y cada condición en el *bicluster* esta caracterizada solo por un subconjunto de genes. El objetivo en las técnicas de *biclustering* es entonces identificar subgrupos de genes y subgrupos de condiciones, mediante un *clustering* realizado simultáneamente tanto en las filas como en las columnas de la matriz de expresión de genes, en vez de hacerlo en estas dos dimensiones por separado. Se puede concluir que, a diferencia de los algoritmos de *clustering*, los algoritmos de *biclustering* identifican grupos de genes que muestran patrones de actividad similar bajo un subconjunto específico de condiciones experimentales. Entonces, las técnicas de *biclustering* son un enfoque clave a usar en cualquiera de las siguientes situaciones:

1. Solo un subconjunto de genes participan en un proceso celular de interés.
2. Un proceso celular de interés esta activo solo en un subconjunto de condiciones experimentales.
3. Un solo gen puede participar en múltiples procesos y mecanismos biológicos que pueden o no estar coactivos en todas las condiciones.

Adicionalmente, la robustez en los algoritmos de *biclustering* es especialmente relevante debido a dos características adicionales de los sistemas bajo estudio. La primera característica es la mera complejidad de los procesos de regulación de genes que requieren herramientas poderosas de análisis. La segunda característica es el nivel de ruido en los experimentos reales de expresión de genes, que hacen indispensable el uso de herramientas estadísticas inteligentes.

4.1. Definiciones y Formulación del Problema

De ahora en mas, se trabajara con una matriz $\boldsymbol{X}$ de datos de n por m, donde cada elemento x_{ij} será, en general, un dado valor real. En el caso de las matrices de datos de expresión de genes, x_{ij} representa el nivel de expresión del gen i bajo la condición j. Tal matriz $\boldsymbol{X}$ esta definida por su conjunto de filas, $I = \{i_1, i_2, ..., i_n\}$, y por su conjunto de columnas, $J = \{j_1, j_2, ..., j_m\}$. En ocasiones se usará (I, J) para denotar a la matriz $\boldsymbol{X}$. Considerando que $G \subseteq I$ y $C \subseteq J$ son subconjuntos de filas y columnas respectivamente, $\boldsymbol{X}_{GC} = (G, C)$ denota la submatriz de $\boldsymbol{X}$ que contiene solo elementos x_{ij} tal que $i \in G$ y $j \in C$.

Dada la matriz de datos $\boldsymbol{X}$, tal como fue definida anteriormente, definimos a un *cluster de filas* como un subconjunto de filas que exhiben comportamiento similar en todo el conjunto de columnas. Esto es, un *cluster* de filas $\boldsymbol{X}_{GJ} = (G, J)$ es un subconjunto de filas definido sobre todo el conjunto de columnas J, donde $G = \{i_1, i_2, ..., i_k\}$ es un subconjunto de filas ($G \subseteq I$ y $k \leq n$). Un *cluster* de filas (G, J) puede entonces ser definido como una submatriz k por m de la matriz $\boldsymbol{X}$.

Similarmente, un *cluster de columnas* es un subconjunto de columnas que exhiben comportamiento similar en todo el conjunto de filas. Un *cluster* de columnas $\boldsymbol{X}_{IC} = (I, C)$ es un subconjunto de columnas definido sobre todo el conjunto de filas I, donde $C=\{j_1, j_2, ..., j_s\}$ es un subconjunto de columnas ($C \subseteq J$ y $s \leq m$). Un *cluster* de columnas (I, C) pueden entonces ser definido como una submatriz n por s de la matriz $\boldsymbol{X}$.

Un *bicluster* es un subconjunto de filas que exhiben comportamiento similar en un subconjunto de columnas y viceversa. El *bicluster* $\boldsymbol{X}_{GC}=(G, C)$ es entonces un subconjunto de filas y un subconjunto de columnas donde $G=\{i_1, i_2, ..., i_k\}$ es un subconjunto de filas ($G \subseteq I$ y $k \leq n$), y $C=\{j_1, j_2, ..., j_s\}$ es un subconjunto de columnas ($C \subseteq J$ y $s \leq m$). Por ende, un *bicluster* (G, C) puede ser definido como una submatriz k por s de la matriz $\boldsymbol{X}$.

El problema específico abordado por los algoritmos de *biclustering* puede ser definido ahora. Dada una matriz de datos $\boldsymbol{X}$, queremos identificar un conjunto de *biclusters* $B_p=(G_p, C_p)$, tal que cada *bicluster* B_p es maximal respecto de su tamaño mientras que satisface algunas características especificas de homogeneidad. Aun cuando las características exactas de homogeneidad varíen de método a método, también es importante considerar que la varianza de cada fila en el *bicluster* sea relativamente alta, con el fin de capturar genes que exhiban tendencias coherentes fluctuantes bajo algún conjunto de condiciones, evitando así *biclusters* constantes triviales (Cheng y Church, 2000). En esta tesis, la homogeneidad $h(G,C)$ esta dada por la métrica residuo cuadrado medio, mientras que el tamaño del *bicluster* es representado por el numero de filas $f(G)$ y por el numero de columnas $g(C)$. La varianza $k(G, C)$ es la varianza de

filas (Cheng y Church, 2000). Así, nuestro problema de optimización puede ser definido como sigue:

maximizar

$$f(G)=|G|. \tag{4.1}$$

$$g(C)=|C|. \tag{4.2}$$

$$k(G,C)=\frac{\sum_{i\in G,j\in C}\left(x_{ij}-x_{iC}\right)^2}{|G|\cdot|C|}. \tag{4.3}$$

sujeto a

$$h(G,C)\le \delta. \tag{4.4}$$

con $(G,C)\in\nabla$, $\nabla=2^{\{1,\ldots,m\}}\times 2^{\{1,\ldots,n\}}$ siendo el conjunto de todos los posibles *biclusters*, donde

$$h(G,C)=\frac{1}{|G|\cdot|C|}\sum_{i\in G,j\in C}\left(x_{ij}-x_{iC}-x_{Gj}+x_{GC}\right)^2. \tag{4.5}$$

es el residuo cuadrado medio,

$$x_{iC}=\frac{1}{|C|}\sum_{j\in C}x_{ij},\quad x_{Gj}=\frac{1}{|G|}\sum_{i\in G}x_{ij}. \tag{4.6}$$

son el promedio de los valores de expresión de filas y columnas de (G,C) y

$$x_{GC}=\frac{1}{|G|\cdot|C|}\sum_{i\in G,j\in C}x_{ij}. \tag{4.7}$$

es el valor promedio de expresión sobre todas las celdas que están contenidas en el *bicluster* (G, C). El umbral δ definido por el usuario representa el valor máximo permitido de disimilaridad entre las celdas del *bicluster*. En otras palabras, el residuo cuantifica la diferencia entre el valor actual de un elemento x_{ij} y su valor esperado tal como es predicho por el correspondiente promedio de fila, promedio de columna, y promedio del *bicluster*. Si un *bicluster* tiene un residuo cuadrado medio menor que un dado valor δ, entonces llamaremos al *bicluster* como δ-*bicluster*.

4.1.1. Grafo Bipartito Pesado y Matrices de Datos.

Se puede establecer una conexión entre las matrices de datos y la teoría de grafos. Una matriz de datos puede verse como un grafo bipartito pesado. Un grafo (V, E), donde V es el conjunto de vértices y E es el conjunto de arcos, se dice que es bipartito si sus vértices pueden particionarse en dos conjuntos L y R tales que cada arco en E tiene un extremo en L y el otro en R: $V = L \cup R$. La matriz de datos $\boldsymbol{X}$ = (G, C) puede ser vista como un grafo bipartito pesado donde cada nodo $n_i \in L$ se corresponde con una fila y cada nodo $n_j \in R$ se corresponde con una columna. El arco entre el nodo n_i y el nodo n_j tiene un peso x_{ij}, denotando al elemento en la matriz que corresponde a la fila i y a la columna j (y denotando la fuerza en el nivel de

activación, en el caso de matrices de expresión de genes). Esta conexión entre matrices y teoría de grafo lleva a enfoques muy interesantes basados en algoritmos de grafos para el análisis de datos de expresión.

4.1.2. Complejidad del Problema.

Aunque la complejidad del problema de *biclustering* puede depender de la formulación exacta del problema y, específicamente, de la función de mérito usada para evaluar la calidad de un dado *bicluster*, todas las variantes interesantes de este problema son NP-completo. En su forma mas simple, la matriz de datos $\boldsymbol{X}$ es una matriz binaria, donde cada elemento x_{ij} es 0 o 1. Cuando este es el caso, un *bicluster* se corresponde con un

biclique en el grafo bipartito correspondiente. Así, encontrar el *bicluster* de mayor tamaño es equivalente a encontrar el *biclique* con mayor cantidad de arcos en un grafo bipartito, un problema conocido en ser NP-completo (Peeters, 2003). Casos mas completos, donde los valores numéricos de la matriz $\boldsymbol{X}$ son tomados en cuenta para computar la calidad de un *bicluster*, tienen una complejidad que es necesariamente no menor a este caso dado que, en general, también podrían ser usadas para resolver esta versión mas restrictiva del problema que es NP-completo. Es mas, el problema de encontrar el mayor δ-*bicluster* es NP-completo (Cheng y Church, 2000). La alta complejidad de este problema ha motivado a los investigadores a aplicar varias técnicas de aproximación para generar soluciones casi óptimas. En particular, los algoritmos evolutivos son muy apropiados para abordar esta clase de problemas (Bleuler *et al.*, 2004; Mitra y Banka, 2006; Divina y Aguilar-Ruiz, 2006) y serán introducidos brevemente a continuación.

4.2. Computación Evolutiva

La computación evolutiva es una rama en creciente desarrollo dentro de las ciencias de la computación. El enfoque engloba técnicas que simulan la evolución natural. Constituye un grupo de estrategias alternativas para abordar problemas complejos de búsqueda y aprendizaje a través de modelos computacionales de procesos evolutivos. Su filosofía parte de un hecho observado de la naturaleza: los organismos vivos tienen una destreza en la resolución de los problemas que se les presentan, y obtienen sus habilidades a través del mecanismo de la evolución natural.

La evolución se produce en casi todos los organismos, como consecuencia de dos procesos primarios: la selección natural y la reproducción sexual (cruzamiento). La evolución natural es un proceso de cambio sobre una población reproductiva que contiene variedades de individuos

con características heredadas y heredables. Los mismos difieren en su aptitud, la cual constituye el factor más importante al momento de obtener su éxito reproductivo.

Los orígenes de la teoría evolutiva datan del año 1859, cuando Darwin publica su libro "El origen de las especies". En el mundo científico surgieron entonces variadas polémicas por las revolucionarias teorías que allí se presentaban, como la idea de que las especies evolucionan acorde al medio, para adaptarse a éste. El universo dejaba de ser una creación de Dios estática y perfecta para transformarse en un conjunto de individuos en constante competición y evolución, con el fin de poder perpetuar su especie en el tiempo. La principal idea detrás de la teoría de Darwin consiste en que las especies se crean, evolucionan y desaparecen si no se adaptan; solo los mejores, los más aptos, los que mejor se acomoden en el medio sobreviven para perpetuar sus aptitudes.

Desde el punto de vista de la computación, se puede ver aquí un claro proceso de optimización. Se toman los individuos mejor adaptados (mejores soluciones temporales), se cruzan (mezclan), generando nuevos individuos (nuevas soluciones) que contendrán parte del código genético (información) de sus antecesores y, el promedio de adaptación de toda la población se mejora.

Los primeros trabajos en computación evolutiva aparecieron a finales de los años 50 con los trabajos de Bremermann (1958), Friedberg (1959) y otros. Sin embargo, el campo permaneció desconocido por tres décadas debido a la ausencia de una plataforma computacional poderosa y a los defectos metodológicos de los primeros métodos (Fogel *et al*., 1966). Los trabajos de Rechenberg (1973), Holland (1975) y Schwefel (1981) cambiaron lentamente el escenario. Tanto es así que hoy en día el incremento en producción científica desarrollada en torno a este tema es exponencial.

Los principales beneficios de la computación evolutiva consisten en que esta técnica brinda una importante ganancia de flexibilidad y adaptabilidad a distintos problemas, en combinación con un desempeño robusto y características de búsqueda global. Constituye un concepto general que se puede adaptar para la resolución de una gran variedad de situaciones, en especial problemas de optimización difíciles de abordar con otras técnicas. Aquellos problemas entre cuyas características se encuentran alta dimensionalidad, multi-modalidad, fuerte no linealidad, no diferenciabilidad, ruido y funciones dependientes del tiempo constituyen problemas difíciles (si no irresolubles), y es en esos casos en los que se recurre con éxito a los algoritmos evolutivos.

Los algoritmos evolutivos son todos los sistemas de resolución de problemas de optimización o búsqueda que emplean modelos computacionales de algún conocido mecanismo de evolución como elemento clave en su diseño e implementación. Dicho de otra forma, en

general, se denomina algoritmo evolutivo a cualquier procedimiento estocástico de búsqueda basado en los principios de la evolución. Para poder emular suficientemente el proceso de evolución ,el algoritmo evolutivo debe disponer de:

1. Una población de posibles soluciones debidamente representadas a través de individuos.
2. Un procedimiento de selección basado en la aptitud de los individuos.
3. Un procedimiento de transformación, esto es, de construcción de nuevas soluciones a partir de las disponibles actualmente.

Una vez que estos elementos han sido convenientemente especificados se implementa el algoritmo evolutivo siguiendo el siguiente esquema general:

t :=0;
Inicializar P(t);
Evaluar P(t)
Mientras no se termine hacer:
    P'(t) := variación [P(t)];
    evaluar [P'(t)];
    P(t+1) := seleccione[P'(t)$\cup$Q];
    t:=t+1;
Fin Mientras

Figura 4.1. Esquema general de un algoritmo evolutivo.

Existen tres tipos de algoritmos evolutivos: las estrategias evolutivas, la programación evolutiva y los algoritmos genéticos. Las estrategias evolutivas se basan en una técnica desarrollada en sus inicios por Rechenberg (1973) y Schwefel (1981), diseñada inicialmente con la meta de resolver problemas de optimización discretos y continuos, principalmente experimentales y considerados difíciles. Utiliza recombinación o cruzamiento y la operación de selección, ya sea determinística o probabilística; elimina las peores soluciones de la población y no genera copia de aquellos individuos con una aptitud por debajo de la aptitud promedio. Por su parte, la programación evolutiva fue introducida por Fogel *et al*. (1966). Inicialmente fue diseñada como un intento de crear inteligencia artificial. El procedimiento es muy similar a las estrategias evolutivas con la diferencia de que no emplea la recombinación. En cuanto a los algoritmos genéticos, estos modelan el proceso de evolución como una sucesión de frecuentes

cambios en los genes, con soluciones análogas a cromosomas. El espacio de soluciones posibles es explorado aplicando transformaciones a éstas soluciones candidatas tal y como se observa en los organismos vivientes: cruzamiento, mutación y selección. Los algoritmos genéticos constituyen el paradigma más completo de la computación evolutiva ya que en su filosofía resumen de modo natural todas las ideas fundamentales de dicho enfoque. Son muy flexibles ya que pueden adoptar con facilidad nuevas ideas, generales o específicas, que surjan dentro del campo de la computación evolutiva. Se pueden hibridizar con otros paradigmas y enfoques, aunque no tengan ninguna relación con la computación evolutiva. Por último, otra importante ventaja con respecto a las dos técnicas anteriormente presentadas consiste en que son el paradigma que cuenta con una mayor base teórica.

4.2.1. Algoritmos Genéticos Multi-Objetivo

Un problema de optimización multi-objetivo plantea la optimización (minimización o maximización) de un conjunto de funciones, habitualmente en conflicto entre sí. La existencia de múltiples funciones objetivo plantea una diferencia fundamental con un problema mono-objetivo: en general no existirá una única solución al problema, sino un conjunto de soluciones que plantearán diferentes compromisos entre los valores de las funciones a optimizar.

A continuación se presenta la formulación general de un problema de optimización multi-objetivo. Cabe mencionar que la mayor parte de los problemas de optimización subyacentes a problemas del mundo real tienen una formulación de este tipo, aunque en muchos casos son abordados siguiendo un enfoque mono-objetivo.

Maximizar/Minimizar

$$F(x) = (f_1(x), f_2(x), ..., f_M(x)) . \tag{4.8}$$

sujeto a

$$G(x) = (g_1(x), g_2(x), ..., g_s(x)) \geq 0 . \tag{4.9}$$

$$H(x) = (h_1(x), h_2(x), ..., h_R(x)) = 0 . \tag{4.10}$$

$$x_i^{(\inf)} \leq x_i \leq x_i^{(\sup)}, \; 1 \leq i \leq N . \tag{4.11}$$

La solución al problema de optimización multi-objetivo corresponderá a un vector de variables de decisión $x = (x_1, x_2, ..., x_N)$ que satisfaga las restricciones impuestas por las funciones G y H, ofreciendo valores que representen un compromiso adecuado para las funciones $f_1, f_2, ..., f_M$.

Considerando el caso de un problema de minimización de funciones, un punto x^* es "óptimo de Pareto" si para todo x en Ω (la región factible del problema), se cumple que $f_i(x) = f_i(x^*)$ $\forall i \in \{1..M\}$, o para al menos un valor de i se cumple que $f_i(x) > f_i(x^*)$. Esto significa que no

existe un vector factible que sea "mejor" que el óptimo de Pareto en alguna función objetivo sin que empeore los valores de alguna de las restantes funciones objetivo.

Asociada con la definición anterior, se introduce una relación de orden parcial denominada dominancia entre vectores solución del problema de optimización multi-objetivo. Un vector $w = (w_1, w_2, ..., w_N)$ domina a otro $v = (v_1, v_2, ..., v_N)$ si $w_i \leq v_i \ \forall i \in \{1..M\} \ \wedge \exists i \in \{1..M\} \mid w_i < v_i$. En este caso se nota $w \prec v$.

Dado que diferentes valores de las variables de decisión representan diferentes compromisos, la resolución de un problema de optimización multi-objetivo no se concentra necesariamente en hallar un único valor solución, sino que se plantea hallar un conjunto de soluciones no dominadas, de acuerdo a la definición presentada anteriormente.

El conjunto de soluciones óptimas al problema de optimización multi-objetivo se compone de los vectores factibles no dominados. Este conjunto se denomina conjunto óptimo de Pareto y está definido por $\boldsymbol{P^*} = \{ x \in \Omega / \neg \exists x' \in \Omega, f(x') \prec f(x) \}$. La región de puntos definida por el conjunto óptimo de Pareto en el espacio de valores de las funciones objetivo se conoce como frente de Pareto. Formalmente, el frente de Pareto está definido por $FP^* = \{u = (f_1(x), f_2(x), \ldots, f_M(x)) \mid x \in \boldsymbol{P^*}\}$.

4.2.1.1. Breve reseña histórica

La complejidad inherente a los problemas de optimización multi-objetivo plantea un difícil reto para su resolución mediante algoritmos exactos determinísticos a medida que crece la dimensión del espacio de soluciones. De este modo, las técnicas clásicas como los algoritmos enumerativos o los métodos exactos de búsqueda local, basados en gradientes o que utilizan las técnicas estándar de programación determinística – métodos golosos (*greedy*), técnicas de ramificación y poda (*branch & bound)*, etc. - si bien pueden ser aplicables para problemas de complejidad reducida, exigen un costo computacional excesivo para la resolución de problemas multi-objetivo complejos con aplicación en el mundo real (Fang *et al*., 1993).

En este contexto, las técnicas heurísticas estocásticas fueron propuestas como alternativas para la resolución de problemas de optimización multi-objetivo, para alcanzar soluciones aproximadas de buena calidad en tiempos de cómputo razonables. Enmarcados en esta categoría, las técnicas de computación evolutiva emergieron como métodos robustos y efectivos para la resolución de los problemas de optimización multi-objetivo y se popularizaron en la última década como consecuencia de su éxito. Los algoritmos genéticos para optimización multi-objetivo surgen como una extensión de los algoritmos genéticos para problemas mono-objetivo, utilizando fundamentalmente varios conceptos relacionados con el tratamiento de funciones “multimodales” por parte de los algoritmos genéticos mono-objetivo.

El concepto de *multimodalidad* se usa en diferentes contextos. Un problema multimodal es un problema que tiene varios máximos, todos ellos de la misma jerarquía, pero este término también se aplica a aquellos problemas que tienen varias soluciones posibles. En general, casi todo problema de búsqueda, formulado como un problema de optimización, suele tener varios máximos, pero uno de ellos es mejor que el resto, y se denomina máximo global. El resto son máximos locales; es decir, se puede definir una vecindad alrededor de ellos en la cual son máximos globales. Sin embargo, en muchos casos, todos los máximos globales tienen la misma jerarquía, ya sea por tener el mismo valor numérico, o bien porque se establezca un criterio de mejor tal que no se pueda decidir cuál de ellos es mejor que otro. Esto suele suceder en los problemas de optimización multi-objetivo, en los que las soluciones se califican con respecto a varias variables o criterios, y ningún criterio tiene preferencia sobre otro.

De acuerdo a (Coello Coello *et al*., 2002), la capacidad de los algoritmos genéticos para resolver problemas con múltiples objetivos fue sugerida en la década de 1960 por Rosenberg (1967), pero hasta mediados de la década de 1980 no se presentó la implementación de un algoritmo evolutivo para optimización multi-objetivo (Schaffer, 1984). A partir de la década de 1990 fueron realizadas una gran cantidad de propuestas de algoritmos genéticos (o evolutivos) multi-objetivo, formándose una comunidad de investigadores en el área que en la actualidad trabaja activamente. Dado que procesan en paralelo un conjunto de soluciones, los algoritmos genéticos tienen la potencialidad de tratar problemas con objetivos múltiples, hallando en cada ejecución un conjunto de soluciones aproximadas al frente de Pareto. Esto representa una importante ventaja respecto a los algoritmos tradicionales, que solamente generan una solución por ejecución. Complementariamente, los algoritmos genéticos tienen otras ventajas respecto a los algoritmos tradicionales, como ser menos sensibles a la forma o a la continuidad del frente de Pareto o permitir abordar problemas con espacio de soluciones de gran dimensión. El reporte técnico de (Coello Coello *et al*., 2002) denominó a esta rama de la investigación "Optimización Evolutiva Multi-objetivo".

4.2.1.2. Clasificación

Según (Coello Coello *et al*., 2002) pueden considerarse, en general, dos tipos principales de algoritmos evolutivos[2] multi-objetivo:

1. Los algoritmos que no incorporan el concepto de óptimo de Pareto en el mecanismo de selección del algoritmo evolutivo (p.ej., los que usan funciones de agregación lineales).

[2] En lo que resta de esta tesis, se hará referencia indistintamente a los algoritmos genéticos como algoritmos evolutivos, si bien se debe tener en mente que los primeros son en realidad un caso particular de los segundos.

2. Los algoritmos que jerarquizan a la población de acuerdo a si un individuo es o no dominado (usando el concepto de óptimo de Pareto).

Históricamente podemos considerar que ha habido dos generaciones de algoritmos evolutivos multi-objetivo:

1. *Primera Generación*: Caracterizada por el uso de jerarquización de Pareto y nichos. En esta generación se puede definir una nueva sub-caracterización en la que se diferencian las técnicas non-Pareto y las basadas en el óptimo Pareto. Dentro de las técnicas non-Pareto se encuentran por ejemplo las funciones de agregación.
2. *Segunda Generación*: Se introduce el concepto de elitismo en dos formas principales: usando selección y una población secundaria.

Tras gozar de mucho éxito por alrededor de 5 años, los algoritmos de primera generación comenzaron a caer en desuso. Desde finales de los 1990s los algoritmos evolutivos multi-objetivo que usan elitismo son vistos como el estado del arte en el área. Los algoritmos de segunda generación enfatizan la eficiencia computacional. Se busca vencer la complejidad de la jerarquización de Pareto ($O(MK^2)$, donde M es el número de funciones objetivo y K es el tamaño de la población, y de las técnicas tradicionales de nichos ($O(K^2)$). Algoritmos evolutivos como el *Non-dominated Sorting Genetic Algorithm-II* (*NSGA*-II) (Deb *et al*., 2000) y el *Strength Pareto Evolutionary Algorithm* 2 (*SPEA*-2) (Zitzler *et al*., 2002) se han convertido en enfoques estándares. Sin embargo, los algoritmos no basados en dominancia de Pareto siguen siendo usados en algunos dominios (p.ej., optimización combinatoria) con gran éxito (Grignon *et al.*, 1996; Arslan *et al.*., 1996; Eklund y Embrechts, 1999). Esto se debe a que son métodos eficientes y fáciles de implementar, apropiados para manejar pocos objetivos.

4.2.1.3. Funciones de agregación

Esta técnica se denomina "función de agregación" porque integra todos los objetivos en uno solo. Se puede utilizar suma, multiplicación o cualquier otra combinación de operaciones aritméticas. Está basada en un método de programación antiguo dado que puede derivarse de las condiciones de Kuhn y Tucker (1951) para soluciones no dominadas.

Un ejemplo de aplicación de este enfoque es mediante la suma de pesos con la forma:

$$\min \sum_{i=1}^{M} p_i f_i(\vec{x}) . \tag{4.12}$$

donde p_i es el coeficiente de peso representando la importancia relativa de la i-ésima función objetivo del problema. Generalmente se asume que:

$$\sum_{i=1}^{M} p_i = 1 . \tag{4.13}$$

Entre las principales ventajas de esta técnica se puede mencionar que resulta muy simple implementarla y su ejecución es muy eficiente. Como desventaja podemos encontrar que, en general, las combinaciones lineales de pesos no suelen funcionar en los casos en que el frente de Pareto es cóncavo, más allá de los pesos utilizados (Das y Patvardhan, 1998). Sin embargo, los pesos pueden ser generados de tal forma que el frente de Pareto sea rotado (Jin *et al*., 2001) y de esta forma, los frentes de Pareto pueden ser generados de manera eficiente.

4.3. Algoritmos para Biclustering

En esta sección se describirán brevemente los algoritmos de *biclustering* aplicados a datos de expresión de genes que a nuestro criterio resultaron más relevantes para el desarrollo de esta tesis. Primeramente se comenzaran describieron aquellos que siguen enfoques tradicionales de resolución, es decir, enfoques no evolutivos, para después describir a aquellos algoritmos de *biclustering* basados en computación evolutiva.

4.3.1. Algoritmos Tradicionales para *Biclustering*

Como se mencionó anteriormente, el primer enfoque en aplicar el concepto de *biclustering* a los datos de expresión de genes fue propuesto por Cheng y Church (2000). Dada una matriz de datos $\boldsymbol{X}$, y un valor de residuo cuadrado medio ($h(G, C)$) máximo aceptable (δ), la meta es encontrar el subconjunto de filas y el subconjunto de columnas con un $h(G, C)$ no mayor a δ. Con el fin de lograr este objetivo, en este trabajo se propusieron varias estrategias golosas (*greedy*) de adición y remoción de columnas y filas que son combinadas en un enfoque global. El borrado múltiple de nodos remueve todas las filas y columnas con un residuo de fila/columna superior a $\delta.\alpha$ en cada iteración, donde α es un parámetro introducido para el procedimiento de búsqueda local. El borrado simple de nodos remueve iterativamente la fila o la columna que da el mayor decrecimiento de $h(G, C)$. Finalmente, el método de adición de nodos añade filas y columnas que no incrementen el $h(G, C)$ actual del *bicluster*. Con el fin de encontrar un número dado de *biclusters*, el enfoque se ejecuta iterativamente en las filas y columnas restantes que no están presentes en los *bicluster* previamente obtenidos.

Otro método para *biclustering* es el *Iterative Signature Algorithm* (o ISA) (Ihmels *et al*., 2004). La novedad conceptual más importante de este algoritmo es el foco en la propiedad

deseada de los *biclusters* individuales co-regulados que serán extraídos de la matriz de datos de expresión. De acuerdo a los autores, este *bicluster* de transcripción consiste en todos lo genes que son similares cuando se los comparan sobre las condiciones, y todas las condiciones que son similares cuando se las comparan sobre los genes. Esta propiedad es referida como auto-consistencia. En este sentido, se propone la identificación de módulos (denominados como signaturas) por medio de la refinación iterativa de conjuntos de genes aleatorios dados como entrada, usando el algoritmo de signatura previamente introducido por los mismos autores. Así, los módulos de transcripción auto-consistentes emergen como puntos fijos de este algoritmo.

Bimax (Zimmermann *et al.*, 2006) es un algoritmo que consiste en el uso de una estrategia de dividir y conquistar con el fin de particionar a $\boldsymbol{X}$ en tres submatrices, una de las cuales contiene solo celdas con 0 y entonces puede ser ignorada en los pasos sucesivos. El procedimiento luego es aplicado recursivamente a las restantes submatrices U y V; la recursión termina cuando la matriz actual representa un *bicluster*, es decir, contiene solo 1s. Si U y V no comparten ninguna columna y fila de $\boldsymbol{X}$ entre si, entonces las dos matrices pueden ser procesadas independientemente. Sin embargo, si U y V tienen un conjunto de filas en común, entonces se debe tener especial cuidado en la generación de los *biclusters*. Una desventaja de este método es que solo funciona con matrices de datos binarios. Entonces, los resultados dependen fuertemente de la precisión del paso de discretización.

El método presentado por Ben-Dor *et al*. (2002), llamado *Order Preserving SubMatrix* (OPSM), define a un *bicluster* como una submatriz que preserva el orden. De acuerdo a su definición, un *bicluster* es un grupo de filas cuyos valores inducen un orden lineal entre un subconjunto de columnas. El trabajo se enfoca en el orden relativo de las columnas en el *bicluster* más que en la uniformidad de los valores reales de la matriz de datos. Más específicamente, se trata de identificar submatrices que preserven ese orden y que sean del mayor tamaño posible. Una submatriz preserva el orden si hay una permutación de sus columnas bajo la cual la secuencia de valores en cada fila es estrictamente creciente. De esta forma, en Ben-Dor *et al*. (2002) aspiran a encontrar un modelo completo con el mayor soporte estadísticamente significativo. En el caso de los datos de expresión, esta submatriz esta determinada por un subconjunto de genes y un subconjunto de condiciones, tal que, dentro del conjunto de condiciones, los niveles de expresión de todos los genes tienen el mismo ordenamiento lineal. Como resultado, se aborda la identificación y la evaluación estadística de patrones co-expresados para grandes conjuntos de genes considerando que, en general, los datos contienen varios de estos patrones.

4.3.2. Algoritmos Evolutivos para *Biclustering*

El primer método reportado que aborda el problema de *biclustering* de matrices de expresión de genes por medio de algoritmos evolutivos fue propuesto por Bleuler *et al*. (2004). En este trabajo se presentaron varias variantes, analizando el uso de un algoritmo evolutivo mono objetivo, un algoritmo evolutivo combinado con una estrategia de búsqueda local (Chen y Church, 2000) y esa misma búsqueda local por si sola. En el caso del algoritmo evolutivo, la novedad consiste en la forma de mantenimiento de diversidad que puede ser aplicada durante el proceso de selección. Para el caso del algoritmo evolutivo hibridizado con la búsqueda local, se considera el caso de que el nuevo individuo devuelto por el procedimiento de búsqueda local deba reemplazar al individuo original (enfoque Lamarckaniano) o no (enfoque Baldwiniano). En relación a la búsqueda local aplicada como estrategia por si sola, se propuso una nueva versión no determinística, donde la decisión en el curso de ejecución se realiza acorde a cierta probabilidad.

Respecto del algoritmo evolutivo, se adoptó una representación binaria para los individuos donde cada individuo representa a un *bicluster* específico. También se empleo un enfoque de mutación independiente de cada bit en el individuo y un operador de cruzamiento uniforme. Para la definición de la función de aptitud, se distinguieron dos casos: si el algoritmo evolutivo operaba solo o si trabajaba conjuntamente con la estrategia de búsqueda local. Para el primer caso se asigna un mejor valor de aptitud, obtenido a partir del tamaño del *bicluster*, a aquellos individuos que cumplen con la restricción del residuo. Si el *bicluster* tiene un residuo mayor a un dado umbral, llamado δ, entonces se le asigna un valor mayor a 1. Para el segundo caso, como la restricción del residuo es considerada por la estrategia de búsqueda local, solo se toma el tamaño como función de aptitud para los *biclusters*. Se emplearon dos conjuntos de datos en los experimentos: *Yeast* (Cho *et al.*, 1998) y *Arabidopsis Thaliana* (Menges *et al.*, 2003; Laule *et al.*, 2003). El estudio de los resultados esta organizado considerando si la meta es obtener un solo *bicluster* o un conjunto de *biclusters*. Para el análisis de un solo *bicluster*, la evaluación se enfoco en el tamaño de los *biclusters*, y el algoritmo que obtuvo el mejor resultado fue el algoritmo evolutivo combinado con el método de búsqueda local empleando una política de actualización (Lamarckaniano). Para el segundo caso de análisis, los resultado se compararon con respecto a la cobertura de la matriz de datos, y nuevamente el mismo algoritmo evolutivo hibrido obtuvo los mejores resultados.

Otro enfoque, llamado SEBI por *Sequential Evolutionary Biclustering*, fue propuesto posteriormente por Divina y Aguilar-Ruiz (2006). En este trabajo, se presenta un algoritmo evolutivo donde se representan a los *biclusters* por medio de cadenas binarias, al igual que en el enfoque anterior. La principal idea de esta técnica secuencial es que el algoritmo evolutivo es

ejecutado varias veces. De cada corrida, el algoritmo evolutivo retorna el mejor *bicluster* de acuerdo a su tamaño, varianza de filas y a un factor de solapamiento. Si el valor del residuo es menor a δ, entonces el *bicluster* es añadido a un archivo llamado *Results*. Si este es el caso, el método mantiene a los elementos del *bicluster* para usar esa información con el fin de minimizar el solapamiento en la siguiente ejecución del algoritmo evolutivo.

Respecto de los detalles del algoritmo evolutivo, la función de aptitud combina los objetivos anteriormente mencionados por medio de una función de agregación no-Pareto. Se emplea selección por torneo y se implementaron varias opciones para el operador de recombinación. Para los experimentos, el algoritmo evolutivo fue ejecutado sobre dos conjuntos de datos: *Yeast* (Cho *et al.*, 1998) y *Human B-cells* (Alizadeh *et al*., 2000). La comparación se realizo contra los *biclusters* encontrados por (Cheng y Church, 2000) respecto de la cobertura total de la matriz de expresión de genes. Para el conjunto de datos *Yeast*, SEBI obtuvo una cobertura del 38% mientras que el enfoque de Cheng y Church (2000) obtuvo una cobertura del 81%. Respecto del conjunto de datos *Human B-cells*, SEBI obtuvo un 34% mientras que Cheng y Church (2000) obtuvo un 37%. Los autores consideran que estos resultados pueden explicarse como una consecuencia del factor de solapamiento, dado que la consideración de este objetivo actúa naturalmente en detrimento de los otros.

Finalmente, en Mitra y Banka (2006) se presenta un algoritmo evolutivo multi-objetivo combinado con una estrategia de búsqueda local (Cheng y Chutch, 2000). Este método constituye el primer enfoque que implementa un algoritmo evolutivo multi-objetivo basado en la dominancia de Pareto para este problema. Los autores basan su trabajo en el NSGA-II, y buscan *biclusters* con máximo tamaño y homogeneidad. La representación es la misma que en los métodos introducidos previamente y se implementaron cruzamiento uniforme de 1 solo punto, mutación de un solo bit y selección por torneo. La estrategia de búsqueda local es aplicada a todos los individuos con un enfoque lamarkaniano, al comienzo de cada ciclo generacional. El método fue testeado en los datos de *microarray* de *Yeast* (Cho *et al.*, 1998) y *Humman B-cell* (Alizadeh *et al*., 2000). Para el análisis de los resultados, introdujeron una nueva métrica llamada índice de coherencia, definida como el ratio del residuo cuadrado medio respecto del tamaño del *bicluster*. Los *biclusters* fueron comparados con aquellos reportados por Cheng y Church (2000) y, en todos los cados, el algoritmo propuesto obtiene un mejor desempeño en términos del tamaño de los *biclusters*, mientras que satisface el criterio de homogeneidad en términos de δ. Sin embargo, respecto de la cobertura, el trabajo de Cheng y Church (2000) produce mejores resultados.

4.4. Sumario

En este capitulo introdujimos la problemática de *biclustering* de matrices de expresión de genes. Se definió y formulo formalmente al problema y se introdujeron varias nociones útiles. Posteriormente, se presento a la computación evolutiva multi-objetivo como una alternativa para resolver este problema. Finalmente, se enumeraron los principales algoritmos para *biclustering* de matrices de expresión de genes, haciendo especial hincapié en los métodos basados en computación evolutiva.

Capítulo 5

Algoritmos Evolutivos Multi-Objetivo para *Biclustering* de Matrices de Expresión de Genes

En este capitulo se presentan las principales contribuciones de esta tesis en torno al *biclustering* de datos de *microarrays*. Como se menciono en el capitulo anterior, el objetivo es identificar un conjunto de *biclusters*, tal que cada *bicluster* es maximal respecto de su tamaño mientras que satisface algunas características especificas de homogeneidad. Estas características, conflictivas unas con otras, son aptas para el modelado multi-objetivo. En particular, los algoritmos evolutivos son apropiados para abordar este tipo de problemas. A continuación se presentan dos enfoques basados en algoritmos evolutivos para el *biclustering* de datos de *microarrays*, seguidamente de una herramienta de software parar el análisis de datos de expresión de genes

5.1. Un Enfoque Memético Novedoso Basado en la Plataforma PISA

En esta sección proponemos un Algoritmo Evolutivo Multi-Objetivo (AEMO) memético implementado en el contexto de la plataforma PISA para abordar el problema de *biclustering* de datos de microarrays. Nuestra técnica hibridiza AEMOs tradicionales con una nueva versión de un procedimiento de búsqueda local muy conocido (Cheng y Church, 2000). A nuestro entender, esa metodología introduce dos características novedosas que nunca fueron abordadas, o que se consideraron parcialmente, en otras técnicas evolutivas diseñadas para esta instancia del problema. La primer contribución consiste en el diseño de la representación de individuos que contempla el mecanismo de regulación opuesta. La otra característica es la incorporación de un mecanismo que controla el equilibrio entre el tamaño y la varianza de filas de los *biclusters*.

5.1.1. Nuestra Propuesta

El objetivo de este estudio es usar un Algoritmo Evolutivo Multi-Objetivo (AEMO) para aproximar a la frontera de Pareto de *biclusters* de una dada matriz de expresión de genes, dado que este enfoque da el mejor equilibrio entre los objetivos que queremos optimizar (ver ecuaciones 4.1-4). Sin embargo, dado que la frontera de Pareto también incluye *biclusters* que no satisfacen la restricción de homogeneidad, se necesita guiar la búsqueda al área donde esta restricción es satisfecha. En este contexto, aplicamos una técnica de búsqueda local basada en el procedimiento de (Cheng y Church, 2000) después de cada generación, orientando así la exploración y acelerando la convergencia del AEMO por medio de la refinación de sus

cromosomas. Con el fin de tomar en cuenta las filas invertidas, hemos extendido la representación clásica de un *bicluster* y también hemos modificado los operadores genéticos. Así, nuestra propuesta realiza la búsqueda sobre un espacio de búsqueda del doble de tamaño respecto a los otros métodos evolutivos para *biclustering* que se encuentran en la literatura (Bleuler *et al*., 2004; Divina y Aguilar-Ruiz, 2006; Mitra y Banka, 2006). La importancia de considerar estas filas invertidas radica en que estas forman una imagen espejo del resto de las filas en el *bicluster*, y pueden entonces ser interpretadas como co-regulación opuesta (regulación inhibitoria) (Cheng y Church, 2000). De esta forma, nuestra propuesta es capaz de encontrar *biclusters* que los otros métodos evolutivos mencionados no pueden detectar.

Respecto de la implementación, la estrategia multi-objetivo fue construida en base a la plataforma llamada PISA (Bleuler *et al.*, 2003). PISA es una interfase basada en texto para algoritmos de búsqueda, separando el proceso de optimización en dos módulos. Un modulo contiene todas las partes que son especificas al problema de optimización (ej. evaluación de soluciones, representación del problema, y variación de soluciones) y es llamada *Variator*. El otro modulo contiene las partes de un proceso de optimización que son independientes al problema de optimización (mayormente el proceso de selección). Esta parte es llamada *Selector*. Estos dos módulos están implementados como programas separados que se comunican a través de archivos de texto.

Para este trabajo, hemos diseñado un *Variator* especifico para el *biclustering* de datos de *microarrays*, y lo hemos combinado con los *Selector* correspondientes a los algoritmos de optimización IBEA (Zitzler y Künzli, 2004), NSGAII (Deb *et al.*, 2000) y SPEA2 (Zitzler *et al.*, 2002). La razón para la selección de estos AEMOs es que son los optimizadores evolutivos mas recomendados en la literatura. De esta forma, evaluaremos al AEMO que exhibe el mejor rendimiento para este problema. En las siguientes subsecciones, describiremos las principales características del *Variator* implementado y como la búsqueda local es incorporada al proceso de búsqueda evolutivo.

5.1.1.1. Representación de individuos

Cada individuo representa un *bicluster*, que esta codificador por una cadena binaria de tamaño fijo construida por la concatenación de una cadena de bits para los genes con una cadena de bits para las condiciones. El individuo corresponde a una solución para el problema de la generación de los *biclusters* óptimos. Si una posición de la cadena (*locus*) está en 1, significa que la fila o columna relativa pertenecen al *bicluster* codificado. En el caso de que el *locus* este en 0 significa que no forma parte del *bicluster*. Para tener en consideración a la filas invertidas también se considera el agregado de valores negativos en la cadena para los genes. Esto es, un

locus de la cadena esta en -1 cuando la fila relativa inversa pertenece a la solución codificada. La figura 5.1 muestra un ejemplo de esta codificación para un individuo aleatorio.

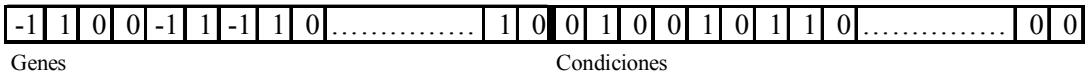


Figura 5.1. Un individuo codificado en forma de cadena binaria extendida representando a un *bicluster*.

5.1.1.2. Operadores genéticos

Es importante dar una descripción breve de los operadores genéticos, dado que tienen una influencia clave en como la búsqueda es realizada por el AEMO.

Mutación. Este operador esta implementado de la siguiente forma: primero se determina si el individuo tiene que mutar por medio de una probabilidad asignada al operador. Si este es el caso, una posición de la cadena es seleccionada al azar procediendo a alterar el *locus* en cuestión. Si la posición resultante es una columna, el *locus* correspondiente simplemente se complementa. Por otro lado si la posición resultante es una fila, entonces se tiene dos casos: si el *locus* es 0 entonces se pone en 1 y el signo se determina con una probabilidad de 0.5. Si el *locus* es 1 o -1, simplemente se cambia su valor a 0.

Recombinacion. Se implemento un operador de cruce de dos puntos con una pequeña restricción: uno de los puntos de corte es seleccionado al azar sobre las filas y el otro punto de corte es seleccionado al azar sobre las columnas. De esta forma, aseguramos que la recombinación se realiza tanto sobre los sub-espacios de los genes como de las condiciones. Así, cuando ambos hijos son obtenidos combinando cada una de las dos partes de los padres (i.e., los extremos y el centro), el individuo que es seleccionado para ser el único descendiente es el no dominado. Si ambos son no dominados, entonces uno de ellos es elegido al azar.

5.1.1.3. Función de aptitud multi-objetivo

Respecto de los objetivos a optimizar, hemos observado que es necesario generar conjuntos maximales de genes y condiciones mientras se mantiene la homogeneidad del *bicluster* con una varianza de fila relativamente alta. Estas características de los *biclusters*, que están en conflicto unas con otras, son muy adecuadas para el modelado multi-objetivo. En este contexto, hemos decidido optimizar los objetivos definidos en (ver ecuaciones 4.1-4): la cantidad de genes, la cantidad de condiciones, la varianza de fila y el residuo cuadrado medio. Los primeros tres objetivos son maximizados, mientras que el ultimo es minimizado.

5.1.1.4. Búsqueda local

Esta subsección describe el procedimiento de búsqueda local que hibridiza a los AEMOs seleccionados. La búsqueda local es aplicada en el modulo *Variator* a los *biclusters* que son seleccionados por el Selector como individuos resultantes de cada generación. Agregar la búsqueda local al *Variator* es la única forma de hibridizar un AEMO sin alterar los principios básicos de PISA (Bleuler *et al.*, 2003). El procedimiento *greedy* esta basado en el trabajo de (Cheng y Church, 2000), con algunas modificaciones introducidas a fin de considerar la varianza de filas y la eficiencia global de la propuesta. El algoritmo comienza con un dado *bicluster* (*G*, *C*). Los genes o condiciones que tengan un residuo cuadrado medio por arriba (o por debajo) de cierto umbral son selectivamente eliminados (o agregados) acorde al algoritmo 5.1.

Entrada: (*G*, *C*) *(un bicluster)*
Salida: (*G*, *C*)' *(un bicluster mejorado)*

Paso 1: *Computar* x_{iC}, x_{Gj}, x_{GC} *y* $h(G,C)$ *por medio de* (4.1-4).

Paso 2: *Si* $h(G,C) < \delta$ *ir al Paso 7.*

Paso 3: *Remover todos los genes* $i \in G$ *tal que* $\frac{1}{C}\sum_{j\in C}\left(x_{ij} - x_{iC} - x_{Gj} + x_{GC}\right)^2 > \alpha \cdot \delta$. *Recalcular todos los promedios y realizar la misma operación para las condiciones. La ecuación para las condiciones es análoga..*

Paso 4: *Recalcular* x_{iC}, x_{Gj}, x_{GC} *y* $h(G,C)$. *Si* $h(G,C) < \delta$ *ir al Paso 6.*

Paso 5: *Remover el nodo* i *(gene o condición) con el mayor* $d(i) = \frac{1}{C}\sum_{j\in C}\left(x_{ij} - x_{iC} - x_{Gj} + x_{GC}\right)^2$. *La ecuación para las condiciones es análoga. Ir al Paso 4.*

Paso 6: *Recalcular* x_{iC}, x_{Gj}, x_{GC} *y* $h(G,C)$.

Paso 7: *Agregar todas las condiciones* $j \notin C$ *tal que* $\frac{1}{G}\sum_{i\in G}\left(x_{ij} - x_{iC} - x_{Gj} + x_{GC}\right)^2 \le h(G,C)$

Paso 8: *Recalcular* x_{iC}, x_{Gj}, x_{GC}, $h(G,C)$ *y* $k(G,C)$.

Paso 9: *Agregar todos los genes* $i \notin G$ *(o su inverso) tal que* $\frac{1}{C}\sum_{j\in C}\left(x_{ij} - x_{iC} - x_{Gj} + x_{GC}\right)^2 \le h(G,C) \ \wedge \ k\left(G \cup \{i\}, C\right) \ge \mu \cdot k(G,C)$. *La ecuación para la fila inversa solo difiere en que el termino* x_{ij} *esta multiplicado por -1.*

Algoritmo 5.1. Búsqueda Local.

Las principales diferencias con la implementación de Cheng y Church (2000) son las siguientes:

- En el Paso 3 nosotros removemos múltiples nodos considerando un umbral diferente, $\alpha.\delta$ en vez de $\alpha.h(G, C)$. Como consecuencia, el paso 5 es realizado un número menor de veces con respecto a la propuesta original. Esto es útil debido a que, con un apropiado valor del parámetro α, el tiempo de CPU necesario para optimizar un *bicluster* es decrementado sin sufrir una perdida significativa en la precisión del algoritmo.
- En el Paso 7 se incorpora la varianza de filas, añadiendo aquellas filas que incrementan en una cierta proporción la varianza global de las filas del individuo.
- Finalmente, en los Pasos 7-9 el algoritmo original trata de agregar cada fila, cada columna y cada fila invertida en ese orden. En nuestro caso, primero se intenta agregar cada condición. Esto incrementa (en promedio) la cantidad de condiciones en el *bicluster* resultante dado que una columna, en general, tiene mayor probabilidad de ser insertada en la solución si esta contiene menos cantidad de filas.

Además de δ, hay dos parámetros adicionales a ser establecidos para este algoritmo. El parámetro α determina cuan frecuente se usa el borrado múltiple de genes y condiciones. Un α mayor lleva a menos nodos borrados en ese paso y entonces, en general, se requiere mayor tiempo de CPU. El otro parámetro es μ que establece la relación entre en numero de genes y la varianza de filas del *bicluster*. Un μ mas grande resulta en individuos con mayor varianza de filas y un menor tamaño. Si $\mu = 0$, este Paso resulta equivalente al de la propuesta original.

5.1.2. Marco Experimental y Resultados

Dos objetivos diferentes fueron establecidos para nuestro estudio. Primero necesitamos determinar cual de los AEMOs funciona mejor en este problema. Este análisis se realizara con las herramientas provistas por Knowles *et al.* (2005). Luego, se comparara el algoritmo evolutivo memético seleccionado con la propuesta de Mitra y Banka (2006) dado que, a nuestro entender, es el único método evolutivo multi-objetivo para *biclustering* de *microarrays* de la literatura.

5.1.2.1. Evaluación de la performance

Como se mencionó anteriormente, la elección del mejor AEMO mimético se basará en los resultados de las herramientas provista por Knowles *et al.* (2005), que son muy reconocidas en el área de la optimización multi-objetivo. Las métricas aplicadas en la evaluación de los AEMOs

son el *Ranking de Dominancia* (Knowles *et al.,* 2005), el *Indicador de Hipervolumen* I_H^- (Zitzler y Thiele, 1999), la versión multiplicativa del Indicador Epsilon Unario I_ε^1 (Zitzler *et al.*, 2003), y el *Indicador R2* I_{R2}^1 (Hansen y Jaszkiewicz, 1998). El Ranking de Dominancia es útil para evaluar la performance relativa de los optimizadores considerados, dado que depende del concepto de la dominancia de Pareto y de un procedimiento de rangos. El *Indicador de Calidad I* mide el número de objetivos que han sido alcanzados por el optimizador bajo consideración. Cada uno de los indicadores enfatiza un aspecto diferente en la preferencia a las soluciones obtenidas. Para mas detalles de las métricas anteriores se sugiere la lectura de Knowles *et al.* (2005).

Si se puede demostrar una diferencia significativa usando el *Ranking de Dominancia*, entonces el único propósito de los *Indicadores de Calidad* es el de caracterizar más las diferencias en los conjuntos de aproximación retornados por los algoritmos. Por otro lado, si no se pueden establecer diferencias significativas por medio del *Ranking de dominancia*, entonces los *Indicadores de Calidad* pueden ayudarnos en la decisión de cual de los optimizadores es mejor. Sin embargo, este último resultado no confirman que el método seleccionado genere mejores conjuntos de aproximación. Con el fin de poder realizar inferencias acerca de los resultados obtenidos con las métricas mencionadas, se aplicara el test de *kruskal-wallis* (Conover, 1999), dado que se testean mas de dos métodos (Knowles *et al.*, 2005).

Para este análisis, hemos usado dos conjuntos de datos de *microarrays*, los datos de expresión del ciclo celular de *Saccharomyces cerevisiae* tomados de Cho *et al.* (1998) y los datos de expresión de *Human B-cells* Lymphoma tomados de Alizadeh *et al.* (2000). Los datos de *Levadura* contienen 2.884 genes y 17 condiciones, y los valores de expresión denotan abundancia relativa de ARNm. Todos los valores son enteros que varían entre 0 y 600, reemplazando los valores perdidos por 0. El conjunto de datos de *Linfoma* contiene 4.026 genes y 96 condiciones. Los valores de expresión son números enteros que varían entre -750 y 650, donde los valores perdidos también fueron reemplazados por 0. Estos conjuntos de datos fueron usados directamente como en Cheng y Church (2000).

5.1.2.2. Primera fase experimental

Los tres AEMOs meméticos, IBEA, NSGA-II y SPEA2 han sido evaluados usando 50 ejecuciones y 75 generaciones sobre los dos conjuntos de datos. La tabla 5.1 resume los parámetros usados para esta comparación. Los valores fueron seleccionados acorde a algunas ejecuciones preliminares. En el caso de la búsqueda local, el parámetro δ fue establecido al mismo valor que en Cheng y Church (2000), μ fue fijado para el mejor compromiso entre

tamaño y varianza, y α fue determinado teniendo en cuenta la eficiencia general en cada conjunto de datos. Todas las ejecuciones fueron controladas por el modulo *Monitor* (Bleuler *et al.*, 2003). En el caso del algoritmo IBEA, se eligió el *Indicador Epsilon Aditivo*, y el resto de los parámetros se fijaron en los valores por defecto. Dado que la plataforma PISA asume que todos los objetivos se minimizan, los cuatro objetivos para nuestro enfoque (ver ecuaciones 4.1-4) fueron adaptados en concordancia. Para los parámetros de los indicadores, se mantuvieron los valores por defecto para los puntos nadir e ideal (apropiadamente extendidos para cuatro objetivos), dado que los objetivos son normalizados automáticamente por la herramienta al intervalo [1..2].

Tabla 5.1. Parámetros fijados para este estudio.

		Generaciones	Prob. Mutación	Prob. Cruzamiento	δ	α	μ
Nuestro Variator	*Yeast*	75	0.3	0.9	300	1.8	0.998
	Lymphoma				1200	1.5	0.999
PISA		α	μ	λ			
		100	50	50			

Respecto de los resultados experimentales, el test de *kruskal-wallis* no pudo detectar diferencias significativas en el *Ranking de Dominancia* de los tres AEMOs, asumiendo un nivel de significancia estadística de $\alpha = 0.05$. Esta situación es igual para los dos conjuntos de datos. Es mas, todos los resultados de las ejecuciones fueron asignados al mayor de los rangos, mostrando que ninguno de los AEMOs generan mejores conjuntos de aproximación respecto de los otros. Esto demuestra la alta influencia de la búsqueda local en el proceso de búsqueda y como ésta guía a los AEMOs a las mismas áreas en el espacio de búsqueda. La tabla 5.2 muestra, para el conjunto de datos de *Levadura*, los resultados del test de *kruskal-wallis* sobre los tres *Indicadores de Calidad*. Las tablas contienen, para cada par de Optimizadores O_R (fila) y O_C (columna), los valores *p* con respecto a la hipótesis alternativa de que el *Indicador de Calidad I* es significativamente mejor para O_R que para O_C. Para el conjunto de datos *Linfoma*, se encontraron diferencias pero ninguna de ellas fue estadísticamente significativa ($\alpha = 0.05$).

Tabla 5.2. Test de *Kruskal-wallis* sobre el *Indicador de Calidad* I_H^- (izquierda), I_ε^1 (centro) y I_{R2}^1 (derecha).

Indicador de Hypervolumen

	IBEA	SPEA2	NSGA-II
IBEA	-	1	0.99
SPEA2	1.50E-07	-	0.16
NSGA2	1.09E-05	0.84	-

Indicador Epsilon Multiplicativo

	IBEA	SPEA2	NSGA-II
IBEA	-	0.99	0.99
SPEA2	2.10E-04	-	0.61
NSGA2	2.10E-04	0.39	-

Indicador R2

	IBEA	SPEA2	NSGA-II
IBEA	-	1	0.99
SPEA2	8.00E-09	-	0.09
NSGA2	4.00E-06	0.91	-

Como se muestra en la tabla 5.2, tanto el SPEA2 como el NSGA-II superan en rendimiento al IBEA en los tres indicadores, pero las diferencias entre SPEA2 y NSGA-II no son

estadísticamente significativas. En vista de estos resultados, no se puede hacer ninguna aseveración respecto de cual de los AEMOs hibridizados funciona mejor en este contexto.

En este punto, vimos la necesidad de aplicar una estrategia *ad hoc* a fin de seleccionar uno de los algoritmos. La tabla 5.3 muestra el promedio de los objetivos para los *biclusters* encontrados por cada uno de los AEMOs meméticos ejecutados con los parámetros mostrados en la tabla 5.1. Es claro que IBEA obtiene *biclusters* de mayor tamaño (en promedio) con respecto a los obtenidos por Spea2 y NSGA-II. Es más, NSGA-II constituye el enfoque que obtiene los *biclusters* mas homogéneos y SPEA2 es el que obtiene la mejor relación entre residuo y varianza de filas. Este comportamiento se evidencia más para el conjunto de datos de *Levadura* que para el conjunto de datos de *Linfoma.* Es importante notar que, en general, los *biclusters* con mayor tamaño tienen un residuo mas alto y menor varianza de filas, mientras que los *bicluster* con residuo bajo presentan tamaños que tienden a ser menores, independientemente de la varianza de filas. Y la varianza de filas es al menos tan grande en valor como el residuo en todos los casos.

Tabla 5.3. Promedio de los valores de los objetivos de IBEA, SPEA2 y NSGA-II en los conjuntos de datos de *Levadura* (arriba) y de *Linfoma* (abajo).

Conjunto de datos de Levadura

	filas promedio	columnas promedio	residuo promedio	varianza promedio	tamaño promedio
IBEA	1047.63	12.52	261.61	296.35	13116.33
SPEA2	794.59	10.37	224.31	296.47	8239.898
NSGA-II	646.34	9.92	204.75	236.04	6411.693

Conjunto de datos de Linfoma

	filas promedio	columnas promedio	residuo promedio	varianza promedio	tamaño promedio
IBEA	655.93	60.71	1089.61	1135.93	39821.51
SPEA2	727.74	52.63	1048.91	1112.03	38300.96
NSGA-II	583.8	54.34	1046.68	1061.7	31723.69

Otra característica de los AEMOs que puede ayudar en la elección de un método esta constituida en cuan bien la búsqueda puede ser orientada por medio del parámetro μ en la búsqueda local. La figura 5.2 muestra, para el conjunto de datos de *Levadura*, el promedio de la varianza de filas (arriba) y el promedio del tamaño (abajo) para los *biclusters* obtenidos por cada método hibrido cuando el parámetro μ varia entre 0.99 y 1.7. Este umbral esta relacionado con los valores de μ que tienen mayor efecto sobre los resultados. Como se puede observar, tanto SPEA2 como NSGA-II se adaptan bien a ser guiados por el parámetro μ, dado que a medida que incrementamos μ, los *biclusters* resultantes tienen mayor varianza de fila en detrimento del

tamaño. En este sentido, el SPEA2 es el algoritmo con mejor desempeño. Por otro lado, el único efecto que se puede observar en el IBEA es la reducción del tamaño de los *biclusters*, dado que no podemos observar ningún efecto sobre la varianza de filas. Tal vez esto sea debido al hecho de que IBEA es un AEMO basado en indicadores, mientras que SPEA2 y NSGA-II son AEMOs basados en Pareto. Entonces, una conjetura es que los pequeños cambios introducidos por el parámetro μ en la población en cada generación no son bien percibidos por IBEA, debido probablemente a que el concepto de solución no dominada no esta soportado por el algoritmo. El comportamiento observado en el conjunto de datos *Linfoma* es similar.

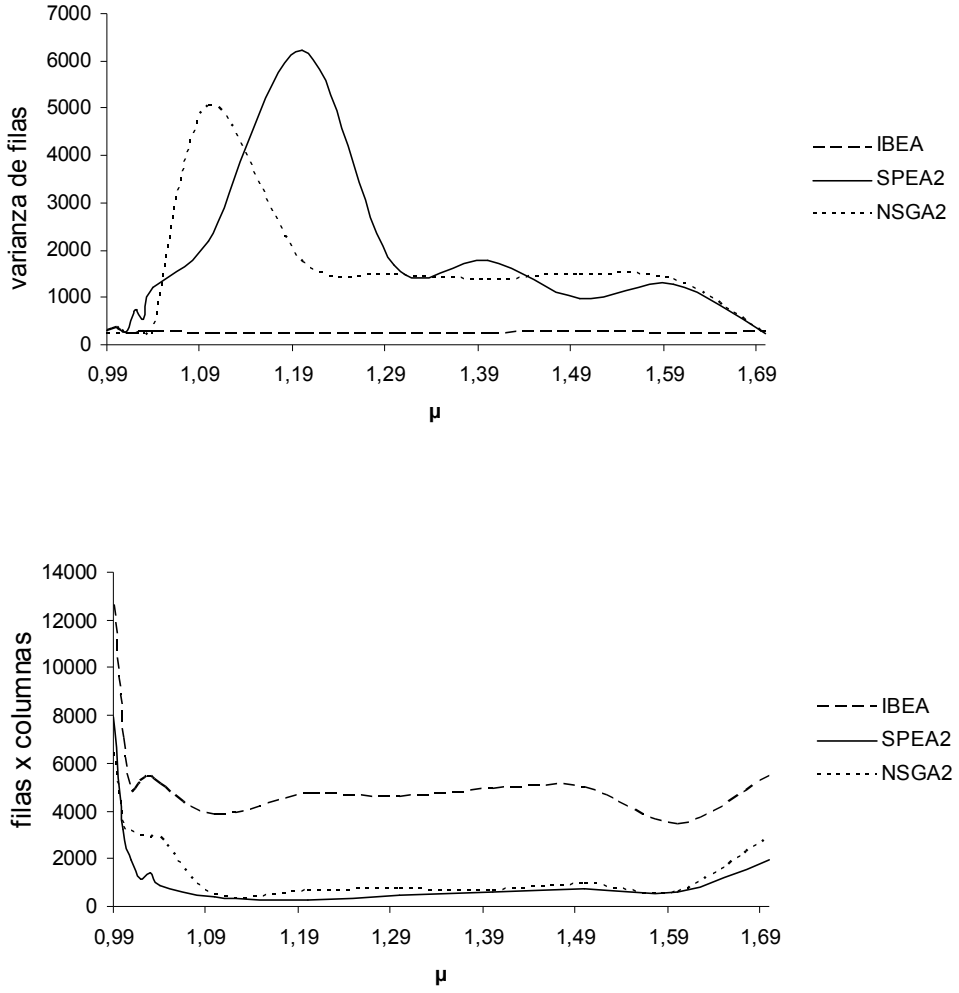


Figura 5.2. Promedio de varianza de filas (arriba) y promedio de tamaño (abajo) para IBEA, NSGA-II y SPEA2 en el conjunto de datos de *Levadura* cuando el parámetro μ varia entre 0.99 y 1.7.

Para el estudio comparativo de la siguiente subsección, hemos elegido al SPEA2 memético debido a que es mas sensible al parámetro μ que los otros, mientras que el tamaño promedio de los *biclusters* son similares a aquellos encontrados por el IBEA memético. Aunque IBEA puede encontrar *biclusters* mas grandes que SPEA2, no es sensible al parámetro μ.

Todas las pruebas se realizaron en un Sempron Móvil con 2GB de RAM. El tiempo de ejecución (en promedio) para el conjunto de datos de *Levadura* fue de 150 segundos mientras que para el conjunto de datos de *Linfoma* fue de 660 segundos. Dado que el tiempo de ejecución esta influenciado principalmente por el procedimiento de búsqueda local, los tres AEMOs obtuvieron resultados similares.

5.1.2.3. Segunda fase experimental

En esta subsección presentaremos una comparación entre el SPEA2 memético y el algoritmo propuesto por Mitra y Banka (2006). Para este análisis usamos los resultaos publicados por Mitra y Banka (2006) en su artículo. Los parámetros usados para el SPEA2 en este análisis son aquellos mostrados en la tabla 5.1. La tabla 5.4 muestra el promedio de los resultados de los valores de los objetivos en el conjunto de datos de *Levadura* para ambos enfoques. También son mostrados el tamaño del *bicluster* mas grande encontrando por cada método y la cobertura de la matriz de expresión de genes $\boldsymbol{X}$. La varianza de filas no fue reportada en Mitra y Banka (2006) por lo que es omitida en esta comparación. Como se puede observar, nuestra propuesta obtiene *biclusters* mas homogéneos (en promedio) mientras que los *biclusters* del algoritmo de Mitra y Banka son mas grandes en tamaño (en promedio). El *bicluster* más grande encontrado por cada uno de los dos métodos es similar en tamaño. Cuando consideramos la cobertura de $\boldsymbol{X}$, nuestra propuesta obtiene una cobertura significativamente mejor de las celdas respecto del algoritmo de Mitra y Banka (2006). Es importante remarcar que los *biclusters* encontrados por nuestro enfoque consideran también las filas invertidas; entonces, la búsqueda se lleva a cabo en un espacio del doble del tamaño de los otros métodos evolutivos para *biclustering* de *microarrays* que se encuentran en la literatura.

Tabla 5.4. Promedio de los valores de los objetivos de los *biclusters* encontrados en el conjunto de datos de *Levadura* por nuestro SPEA2 memético y por el enfoque de Mitra y Banka.

	filas promedio	columnas promedio	residuo promedio	tamaño promedio	tamaño bicluster mas grande	covertura de celdas
SPEA2 memético	794.59	10.37	224.31	8239.89	14602	72.50%
Algoritmo de Mitra y Banka (2006)	1095.43	9.29	234.87	10176.54	14828	51.34%

Respecto del conjunto de datos *Linfoma*, los resultados promedio de los objetivos para el algoritmo de Mitra y Banka (2006) no fueron reportados. Solo se muestra el *bicluster* mas grande obtenido y la cobertura promedio de $\boldsymbol{X}$. En este sentido, nuestra propuesta encuentra un *bicluster* que es mas grande que el reportado por Mitra y Banka (2006). Este *bicluster* tiene 1009 genes, 63 condiciones, un residuo cuadrado medio de 1181.06, una varianza de filas de

1295.05, y un tamaño de 63567; mientras que el *bicluster* mas grande reportado en Mitra y Banka (2006) tiene un tamaño de 37560. Podemos argumentar que, a nuestro entender, este *bicluster* es más grande que cualquier otro *bicluster* reportado por cualquier otro método de la literatura para este conjunto de datos. También, la cobertura de $\boldsymbol{X}$ lograda por nuestro SPEA2 memético es (en promedio) del 33.58% de celdas; significativamente mejor que el promedio de 20.96% obtenido por el algoritmo de Mitra y Banka (2006).

5.1.3. Conclusiones

En este apartado, hemos introducido un marco general multi-objetivo para *biclustering* de *microarrays* hibridizado por un procedimiento de búsqueda local que permite una sintonía fina de los resultados. En una primera fase experimental, hemos hibridizado y comparado tres AEMOs bien conocidos (IBEA, SPEA2 y NSGA-II) basados en la plataforma PISA, a fin de establecer cual de ellos obtiene el mejor resultado. Dado que no se obtuvo ningún resultado concluyente de esta evaluación, hemos elegido al SPEA2 debido a que es capaz de obtener *biclusters* relativamente grandes con una alta sensibilidad al parámetro μ propuesto. Luego, durante una segunda fase experimental, hemos mostrado que la calidad de los resultados del SPEA2 superan a los resultados reportados por Mitra y Banka (2006). La evaluación de los resultados fue llevada a cabo sobre dos conjuntos de expresión de genes para mostrar la efectividad del método propuesto.

Por otra parte, proveemos a los científicos biólogos con un parámetro extra para determinar que *bicluster* consideran mas relevantes, brindando la posibilidad de ajustar el tamaño y la varianza de filas de los *bicluster* encontrados. Es más, los enfoques evolutivos para *biclustering* que se encuentran en la literatura no consideran la inclusión de las filas invertidas, tal vez por razones de eficiencia dado que el espacio de búsqueda es duplicado. Sin embargo, estas filas invertidas son muy importantes debido a que pueden ser interpretadas como co-reguladas recibiendo regulación opuesta. En este contexto, también hemos demostrado que es posible tomar en cuenta a estas "filas extras" mejorando la calidad de los *biclusters* sin perdida de eficiencia.

5.2. BiHEA: Un Nuevo Enfoque Evolutivo Hibrido para Biclustering de Microarrays

A pesar de haber obtenido resultados importantes en el estudio mencionado previamente, ciertas características presentes en el problema de *biclustering* (como el solapamiento de *biclusters*) hacen que la utilización de algoritmos de optimización multi-objetivo de propósito general como los disponibles en PISA (Bleuler *et al*., 2003) no sea óptima. La razón es la poca

flexibilidad de estos algoritmos evolutivos para incorporar este tipo de características en el proceso de búsqueda. En este contexto, presentaremos un nuevo enfoque evolutivo multi-objetivo para *biclustering* de *microarrays*, llamado BiHEA (por *Biclustering via a Hybrid Evolutionary Algorithm*) que mezcla a un algoritmo evolutivo agregativo con ciertas características que mejoran sus capacidades naturales. Esta metodología introduce dos nuevas características que nunca fueron abordadas por otros algoritmos evolutivos diseñados para este problema. La primer contribución consiste en el diseño de un proceso de recuperación que extrae las mejores soluciones de todas las generaciones. La otra nueva característica es la incorporación de un procedimiento de elitismo que controla la diversidad en el espacio genotípico.

5.2.1. Nuestra propuesta

El objetivo de este estudio es generar *biclusters* casi óptimos de valores coherentes siguiendo un modelo aditivo, acorde a la clasificación dada por Madeira y Oliveira. (2004). Así el algoritmo evolutivo utilizado explora el espacio de búsqueda Ω. Sin embargo, los algoritmos evolutivos mono-objetivos y multi-objetivos no pueden generar soluciones satisfactorias sin una búsqueda local asociada (Bleuler *et al*., 2004; Divina y Aguilar-Ruiz, 2006; Mitra y Banka, 2006; Gallo *et al*., 2009a). Por tal motivo, volveremos a emplear la técnica de búsqueda local basa en el procedimiento de Cheng y Church (2000) para orientar la exploración y acelerar la convergencia del algoritmo evolutivo refinando los cromosomas. Además, dos mecanismos adicionales fueron incorporados en el proceso evolutivo a fin de evitar la perdida de buenas soluciones: un procedimiento de elitismo que mantiene los mejores *biclusters* así como también la diversidad en el espacio genotípico a través de las generaciones, y un proceso de recuperación que extrae las mejores soluciones de cada generación y luego copia esas soluciones a un archivo. Este archivo es el conjunto de *biclusters* retornado por el algoritmo. Aunque estos mecanismos parecen ser similares, hay varias diferencias entre ellos. El procedimiento de elitismo selecciona los *b* mejores *biclusters* que no se solapan en un determinado umbral, pasándolos a la siguiente generación. Estas soluciones pueden formar parte del proceso de selección en las generaciones futuras permitiendo la producción de nuevas soluciones basadas en ellas por medio del operador de recombinación. Sin embargo, debido a imperfecciones en el proceso de selección y de la función de aptitud, algunas buenas soluciones pueden perderse entre generaciones. Para lidiar con este inconveniente, hemos incorporado un archivo, que mantiene los mejores *biclusters* generados en todo el proceso evolutivo. Es importante remarcar que esta "meta" población no forma parte del proceso de selección, es decir, la evolución de la

población después de cada generación es monitoreada por el mecanismo de recuperación sin interferir en el proceso evolutivo.

5.2.1.1. Algoritmo principal

Como se menciono anteriormente, el bucle principal es un proceso evolutivo básico que incorpora la búsqueda local junto con los procesos de elitismo y de recuperación. El algoritmo 5.2 ilustra estos pasos.

Entrada:	*pop_size*	*(tamaño de la población)*
	max_gen	*(numero máximo de generaciones)*
	mut_prob	*(probabilidad de mutación)*
	δ	*(umbral de homogeneidad)*
	α	*(parámetro para la búsqueda local)*
	θ	*(grado de solapamiento en el proceso de recuperación)*
	$\boldsymbol{X}$	*(matriz de datos expresión de genes)*
Salida:	*arch*	*(conjunto de bicluster)*

Paso 1: *Inicialización. Cargar la matriz de datos* $\boldsymbol{X}$*. Generar una población aleatoria* P_0 *de tamaño pop_size. Generar una población vacía arch.*

Paso 2: *Bucle principal. Si max_gen es alcanzado, ir al Paso 9.*

Paso 3: *Selección. Realizar selección por torneo binario sobre* P_t *para llenar el pool de padres* Q_t *de tamaño pop_size.*

Paso 4: *Procedimiento de elitismo. Seleccionar a lo sumo los mejores pop_size/2 individuos de* P_t *que se solapen entre si en a lo sumo el 50% de celdas. Copiar esos individuos a* P_{t+1}*.*

Paso 5: *Descendencia. Generar los individuos restantes (al menos pop_size-pop_size/2) de* P_{t+1} *aplicando el operador de recombinación sobre dos padres seleccionados aleatoriamente de* Q_t*. Aplicar mutación uniforme sobre esos individuos.*

Paso 6: *Búsqueda local. Aplicar la optimización por búsqueda local a los individuos de* P_{t+1} *con un residuo cuadrado medio mayor a* δ*.*

Paso 7: *Procedimiento de recuperación. Para cada individuo* $I \in P_{t+1}$ *con residuo cuadrado medio menor a* δ*, intentar agregar I a arch de la siguiente forma: encontrar el individuo* $J \in$ *arch que comparta al menos el* θ *% de celdas y luego reemplazar J por I solo si I es mas grande que J. Si ningún J fue encontrado, agregar I a arch solo si el tamaño de arch es menor a pop_size. Descartar I en cualquier otro caso.*

Paso 8: *Fin del bucle. Ir al Paso 2.*

Paso 9: *Resultado. Retornar arch.*

Algoritmo 5.2. Algoritmo BiHEA.

En este punto, las diferencias entre el proceso de elitismo y el de recuperación deberían estar aclaradas. El umbral para el nivel de solapamiento en el proceso de elitismo, así como la

proporción de elitismo, fueron determinados empíricamente después de varias ejecuciones del algoritmo sobre diferentes conjuntos de datos. Es importante notar que, como consecuencia de un diseño cuidadoso del procedimiento de recuperación, y mediante una elección de valor adecuado para el parámetro θ, el conjunto de *biclusters* resultante estará ligeramente solapado en comparación con el alto grado de solapamiento presente en otros algoritmos evolutivos para *biclustering* (Bleuler *et al*., 2004; Mitra y Banka, 2006; Gallo *et al*., 2009a).

5.2.1.2. Representación de individuos

Cada individuo representa un *bicluster*, que esta codificado por una cadena binaria de tamaño fijo construida por medio de la concatenación de una subcadena de bits para los genes con otra subcadena de bits para las condiciones. El individuo constituye una solución para el problema de generación de biclusters óptimos. Si una posición esta en 1, la fila o columna relativa pertenece al *bicluster* codificado, de otra forma no pertenece. A diferencia de la representación empleada en la sección anterior, en este trabajo no se consideraran las filas invertidas. Sin embargo, la flexibilidad en la representación permite que estas puedan ser tenidas en cuenta sin mayores inconvenientes, si es que se consideran necesarias (véase sección 7.2.4). La figura 5.3 muestra un ejemplo de esta codificación.

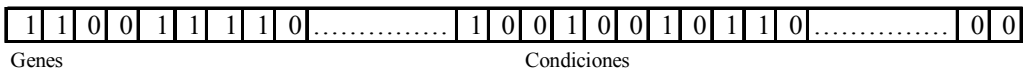


Figura 5.3. Codificación de un individuo representando a un *bicluster*.

5.2.1.3. Operadores genéticos

Después de algunas pruebas preliminares, decidimos aplicar mutación independiente de bits a ambas subcadenas de filas y columnas con un ratio de mutación que permita que el número de bits esperado a cambiar sea igual para ambas subcadenas. También se empleo un cruzamiento de dos puntos por sobre el de un punto debido a que este último puede evitar que sean cruzadas ciertas combinaciones de bits, especialmente en casos donde las diferencias entre las dimensiones de filas y columnas son notables. Así, un punto aleatorio es seleccionado sobre las filas y el otro punto aleatorio es seleccionado sobre las columnas, realizando la recombinación sobre ambos espacios de búsqueda. Luego, cuando ambos hijos son obtenidos combinando cada una de las dos partes de los padres, el individuo que es seleccionado para ser el descendiente es el más apto en términos de la función de aptitud.

5.2.1.4. Función de aptitud

Los objetivos a ser optimizados son los mencionados en el apartado 4.1, es decir, generar conjuntos máximos de genes y condiciones mientras se mantiene la "homogeneidad" del *bicluster* con una varianza de fila relativamente alta, tal y como fue establecido por las ecuaciones (4.1-4). Como se menciono anteriormente, estas características de los *biclusters* son conflictivas unas con otras por lo que son apropiadas para el modelado multi-objetivo. Sin embargo, y a diferencia del método basado en Pareto presentado anteriormente, aquí seguimos un enfoque agregativo. Así, la función de aptitud agregativa que incorpora estas características es presentada en la ecuación 5.1. En vista de que el procedimiento de búsqueda local garantiza la restricción del residuo (Cheng y Church, 2000), la principal razón para tener una consideración especial de los individuos con residuo por encima de δ es en la primer generación, donde los individuos en la población son generados aleatoriamente. Así, aquellas soluciones con menor residuo serán las más aptas. También es importante considerar que un individuo puede violar la restricción de residuo durante la creación de soluciones descendientes. Luego, dado que el operador de cruzamiento retorna el mejor de ambos hijos, los individuos con menor residuo son nuevamente preferidos en este caso, como se puede observar en la formulación de la función de aptitud (5.1).

$$aptitud\ \ (G,C) = \begin{cases} h(G,C) & si \quad h(G,C) > \delta \\ 1 - \dfrac{|G||C|}{mn} + \dfrac{h(G,C)}{\delta} + \dfrac{1}{k(G,C)} & si \quad h(G,C) <= \delta \wedge k(G,C) > 1 \\ 1 - \dfrac{|G||C|}{mn} + \dfrac{h(G,C)}{\delta} + 1 & si \quad cualquier \quad otro \quad caso \end{cases}. \tag{5.1}$$

Sin embargo, cuando los individuos satisfacen la restricción de homogeneidad, no se aplica la búsqueda local. De este modo, las mejoras en las soluciones solo dependen del proceso evolutivo y, luego la consideración de características de los *biclusters* como el tamaño, residuo cuadrado medio y varianza se tornan importantes. Como se menciono anteriormente, la ventaja practica en la consideración de la varianza de filas de un *bicluster* esta en evitar los *biclusters* constantes (Madeira y Oliveira, 2004), dado que pueden obtenerse trivialmente (Cheng y Church, 2000). Notar que la función de aptitud es minimizada en este caso.

5.2.1.5. Búsqueda local

La búsqueda local que hibridiza al proceso evolutivo ya fue descrita anteriormente. Tal y como fue mencionado, el procedimiento goloso esta basado en el trabajo de Cheng y Church (2000), con pequeños cambios que evitan la consideración de las filas invertidas, tal y como fue

aplicada en (Bleuler *et al*., 2004). El algoritmo comienza con un dado *bicluster* (*G*, *C*). Los genes o condiciones que tienen un residuo cuadrado medio superior (o inferior) a cierto umbral son selectivamente eliminados (o adicionados) acorde a la descripción dada en las secciones previas.

5.2.2. Marco Experimental y Resultados

Para este estudio se establecieron dos objetivos diferentes. Primero es necesario analizar la calidad de los resultados de BiHEA en la extracción de *biclusters* con valores coherentes siguiendo un modelo aditivo. Para este análisis, el nuevo método fue probado sobre matrices de datos sintéticos con diferentes grados de solapamiento y ruido. Posteriormente, los resultados fueron comparados con varios de los métodos para *biclustering* más relevantes. A pesar de que evaluar el rendimiento sobre datos sintéticos puede dar una visión precisa de la calidad de un método, puesto que los *biclusters* óptimos se conocen de antemano, cualquier escenario artificial inevitablemente esta sesgado por el modelo subyacente y solo refleja ciertos aspectos de la realidad biológica. Es por ello que en una segunda fase experimental, se analizó la relevancia biológica de los resultados de BiHEA sobre matrices de datos reales.

5.2.2.1. Evaluación del desempeño

A fin de evaluar el desempeño del método de *biclustering* propuesto sobre datos sintéticos, introduciremos la métrica general de pareo de *biclusters* (*general bicluster match score*), que esta basada en la métrica de pareo de genes propuesta por Zimmermann *et al.* (2006). Sean M_1 y M_2 dos conjuntos de *biclusters*. La métrica de pareo de *biclusters* de M_1 con respecto a M_2 esta dada por la ecuación 5.2, la cual refleja el promedio de los valores de coincidencia máximos para todos los *biclusters* en M_1 con respecto a los *biclusters* en M_2.

$$S^*(M_1, M_2) = \frac{1}{|M_1|} \sum_{(G_1,C_1)\in M_1} \max_{(G_2,C_2)\in M_2} \frac{|(G_1,C_1)\cap(G_2,C_2)|}{|(G_1,C_1)\cup(G_2,C_2)|}. \qquad (5.2)$$

En este caso, en vez de considerar solo los genes de los *biclusters* de cada conjunto (Zimmermann *et al.*, 2006), también se consideran las condiciones, es decir, se evalúa la cantidad de celdas de cada *bicluster*. Así, esta métrica es una medida más precisa que la presentada en (Zimmermann *et al.*, 2006) dado que evalúa a los *biclusters* completos y no a una parte de ellos. Ahora, sea M_{opt} el conjunto de los *biclusters* implantados (sintéticos) y sea M la salida de un método de *biclustering*. Definimos la precisión promedio de *biclusters* como $S^*(M, M_{opt})$, la cual refleja hasta que punto los *biclusters* generados por el método representan

verdaderos *biclusters*. En contraste, la cobertura promedio de *biclusters*, dada por $S^*(M_{opt}, M)$, cuantifica cuan bien cada uno de los verdaderos *biclusters* es recuperado por el algoritmo de *biclustering* bajo consideración. Ambas métricas toman el máximo valor de 1 cuando $M_{opt} = M$.

Respecto de los datos reales, dado que los *biclusters* óptimos son desconocidos, la métrica propuesta anteriormente no puede ser aplicada. Sin embargo, se ha tornado ampliamente disponible el conocimiento biológico en la forma de descripciones en leguaje natural de las funciones y procesos a los que los genes están relacionados. De este modo, y en forma similar a la idea perseguida en Tanay *et al.* (2002), Draghici *et al*. (2003) y Zimmermann *et al*. (2006), se investigará si los grupos de genes retornados por BiHEA muestran un enriquecimiento significativo respecto a alguna anotación ontológica de genes específica. Con ese fin hemos diseñado una métrica novedosa que permite medir el enriquecimiento en términos de función molecular y proceso biológico de los resultados de un dado método de *biclustering*. Sea M un conjunto de *biclusters*, GO una anotación y α un nivel de significancia estadística. El indicador de enriquecimiento total de M respecto de GO con un nivel de significancia estadística α esta dado por:

$$E^*(M, GO, \alpha) = \frac{1}{|M|} \sum_{(G,C)\in M} \frac{Maxenrichment(G, GO, \alpha)}{|G|} \sum_{(G,C)\in M} \frac{|G|}{n}. \qquad (5.3)$$

donde *Maxenrichment*(G, GO, α) es la máxima cantidad de genes de G que tienen la misma función molecular/proceso biológico bajo GO con un nivel de significancia estadística de α. En resumen, la métrica propuesta por la ecuación 5.3 mide el promedio de las proporciones de genes máximas de un conjunto de *biclusters* M con respecto a una anotación GO especifica, que tienen asociadas alguna función molecular/proceso biológico con un nivel de significancia estadística de α, ponderado por el promedio de genes de M. Es un indicador de la calidad de los resultados de un método de *clustering*/*biclustering* sobre datos reales, y puede usarse para comparar varios métodos, en donde a valores mas altos mejor.

5.2.2.2. Primera fase experimental: datos sintéticos

Preparación de los datos

El modelo artificial usado para generar datos sintéticos es similar a los enfoques propuestos por Ihmels *et al.* (2002) y Zimmermann *et al*. (2006). En este sentido, los *biclusters* representan módulos de transcripción, donde estos módulos están definidos por un conjunto de genes G regulados por un conjunto común de factores de transcripción, y un conjunto de condiciones C

donde estos factores de transcripción están activos. Es posible variar el grado de solapamiento de los *biclusters* implantados variando la cantidad de genes y condiciones que dos módulos tiene en común. Para esto, definimos el grado de solapamiento *d*, como un indicador de la cantidad máxima de celdas que dos módulos de transcripción pueden compartir. La cantidad total de celdas compartidas es en realidad d^2.

Este modelo permite la investigación de las capacidades de un método para recuperar grupos de genes y condiciones conocidos, mientras que al mismo tiempo, también posibilita estudiar aspectos como ruido y complejidad regulatoria (Zimmermann *et al.*, 2006). En este estudio trabajamos con conjunto de datos "pequeños", $n = 100$ y $m = 100$. Sin embargo, esto no restringe la generalidad de los resultados. En el cado de $d = 0$, implantamos 10 *biclusters* sin solapamiento entre ellos (de tamaño 10 filas por 10 columnas). Para cada $d > 0$, el tamaño de los *biclusters* artificiales fue incrementado en *d* filas y *d* columnas, a excepción del *bicluster* del extremo derecho, para el cual su tamaño se mantuvo sin cambios. Para todo $d > 1$, aparecen 18 *biclusters* adicionales como consecuencia del solapamiento de los módulos de transcripción implantados. Estos *biclusters* adicionales también son incluidos en nuestro estudio debido a que son igualmente apropiados para ser extraídos por un método de *biclustering*, aunque el grado de solapamiento es mayor al de los factores de transcripción artificiales. La figura 5.4 muestra el escenario anterior. Los valores de cada *bicluster* fueron determinados como sigue: para la primer fila, se incorporaron valores reales aleatorios entre 0 y 300 a partir de una distribución uniforme. Luego, para cada una de las restantes filas, se obtuvo un valor aleatorio entre 0 y 300 y se lo sumo a cada elemento de la primer fila. El resultado es un *bicluster* con valores coherentes que sigue un modelo aditivo (Madeira y Oliveira, 2004), con un residuo cuadrado medio igual a 0 y una varianza de filas mayor a 0. Las celdas restantes en la matriz de datos fueron llenadas con valores aleatorios reales entre 0 y 600.

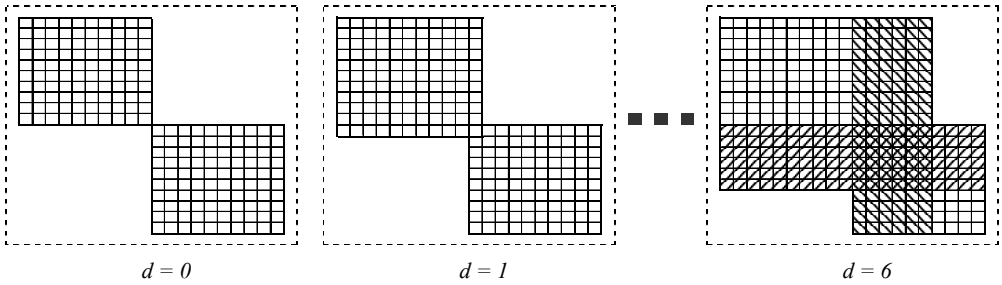


Figura 5.4. Grados de solapamiento entre los *biclusters* artificiales en relación a *d*. En $d = 6$, las líneas diagonales representan los *biclusters* extras generados por el solapamiento de los *biclusters* implantados.

Los conjuntos de datos sintéticos construidos siguiendo el procedimiento antes descrito son útiles para analizar el comportamiento de un método de *biclustering* en niveles crecientes de complejidad regulatoria. Sin embargo, estos conjuntos de datos representan un escenario ideal sin ruido, es decir, bastante lejos de los datos reales. Para hacer frente a esta cuestión, y en vista de que los escenarios reales tienen una alta complejidad regulatoria, se investigó el comportamiento de nuestra propuesta con $d = 6$ y con niveles incrementales de ruido. Para el modelo de ruido, los valores de expresión de los *biclusters* fueron alterados sumando a cada celda valores aleatorios obtenidos a partir de una distribución uniforme entre -k y k, con k variando entre 0 y 25 en incrementos de 5.

Resultados

A modo de referencia, varios algoritmos importantes de *biclustering* fueron ejecutados: BiMax (Zimmermann *et al.*, 2006), Cheng y Church (CC) (Cheng y Church, 2000), OPSM (Ben-Dor *et al.*, 2002), ISA (Ihmels *et al.*, 2004) y SPEA2^{LS} (Gallo *et al.*, 2009a). Para las primeras cuatro implementaciones, se uso la herramienta BicAT (Barlow *et al.*, 2006). Todos los parámetros para estos métodos fueron establecidos después de varias ejecuciones, a fin de obtener los mejores resultados de cada estrategia. Para el BiHEA, los parámetros fueron los siguientes: población = 200; generaciones = 100; $\delta = 300$; $\alpha = 1.2$; probabilidad de mutación = 0.3; y $\theta = 70$. Dado que el numero de *biclusters* generados varia fuertemente entre los métodos considerados, hemos aplicado un proceso de filtrado a la salida de cada algoritmo, similar al proceso de recuperación de nuestro método, para proveer una base común y realizar una comparación justa. El proceso de filtrado extrae, para cada uno de los conjuntos resultantes de *biclusters*, a lo sumo q de los *biclusters* mas grandes que se solapan en a lo sumo el $\theta = 70\%$ de las celdas. Para $d < 2$, q fue fijado en 10, y para el resto, q fue fijado en 28. Respecto de los resultados, en las figuras 5.5a y 5.5b se muestran la precisión y la cobertura promedio obtenida por los diferentes métodos de *biclustering* para los escenarios con grados incrementales de solapamiento. Similarmente, las figuras 5.5c y 5.5d muestran los resultados para los escenarios con niveles incrementales de ruido.

Como se puede observar, el algoritmo BiHEA supera a los métodos de referencia en todos los escenarios, tanto en términos de precisión promedio como de cobertura promedio de *biclusters*. A medida que el grado de solapamiento se incrementa, las figuras 5.5a y 5.5b muestran que los resultados obtenidos por nuestro método mejoran, alcanzando casi valores perfectos con $d = 6$. Esto puede explicarse en términos de la teoría de los esquemas básicos de los algoritmos genéticos, dado que a mayor grado de solapamiento, los esquemas útiles que comparten los *biclusters* óptimos son mayores en tamaño. Esta característica facilita la

construcción de soluciones que satisfacen la restricción de homogeneidad por medio del operador de cruzamiento. Esta observación debería ser cierta para la mayoría de los algoritmos evolutivos. Sin embargo, imperfecciones en los proceso de selección y en las funciones de aptitud pueden derivar en una perdida de esta ventaja, como aparentemente muestra el SPEA2^{LS}. Esto claramente ilustra la necesidad del proceso de recuperación introducido en el BiHEA.

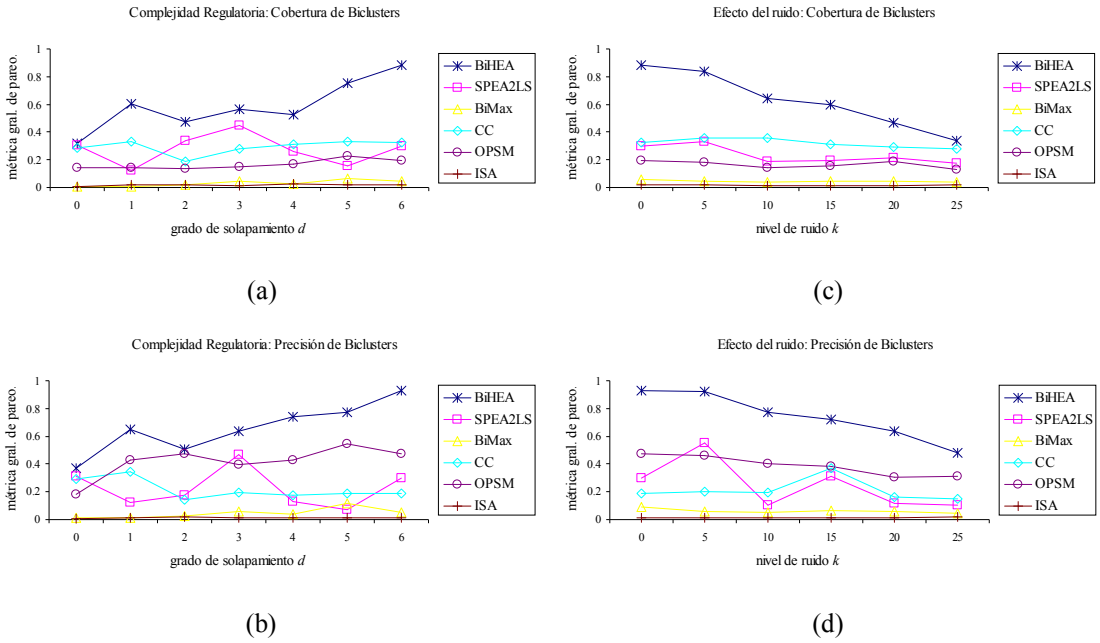


Figura 5.5. Resultados para los escenarios artificiales. Las figuras 5.5a y 5.5b muestran la precisión promedio y la cobertura promedio respectivamente en los escenarios de solapamiento. Las figuras 5.5c y 5.5d muestran la precisión promedio y la cobertura promedio respectivamente en los escenarios de ruido.

Respecto del los efectos del ruido, los resultados son los esperados. A medida que los niveles de ruido aumentan, la degradación de un *bicluster* perfecto aumenta el residuo cuadrado medio asociado a el y posiblemente, la restricción de homogeneidad ya no puede ser satisfecha por el *bicluster* completo. Los métodos de referencia OPSM, CC y SPEA2^{LS} muestran resultados similares, siendo OPSM el mas preciso y el que menos cobertura logra respecto de los otros dos enfoques. Sin embargo, estos métodos aparentar ser menos susceptivos al ruido que el BiHEA. Por otro lado, tanto BiMax e ISA no lograron obtener *biclusters* significativos, lo cual contrasta con las conclusiones publicadas en Zimmermann *et al.* (2006) donde ambos métodos alcanzan un rendimiento casi perfecto. No obstante, creemos que esto puede ser una consecuencia de la forma en que fueron construidos los datos sintéticos, dado a que en el caso de BiMax, el método de discretización pudo quizás ser incapaz de obtener una apropiada representación binaria de las matrices de dato. Por otro lado, la noción de similaridad de filas y columnas en el algoritmo ISA podría ser diferente a la empleada aquí. Sin embargo, los datos sintéticos usados en este trabajo fueron diseñados de la manera antes mencionada dado que,

acorde a nuestro conocimiento, representan matrices de expresión de datos de *microarrays* generales y relevantes que permiten una comparación justa en la evaluación de los algoritmos.

5.2.2.3. Segunda fase experimental: datos reales

En esta subsección, los resultados de BiHEA en matrices de expresión de datos de *microarray* reales serán analizadazos brevemente. Este estudio se enfocara en datos de cáncer de colon (Alon *et al.*, 1999) que consisten en una matriz de expresión de datos de 62 muestras de tejido de colon, 22 de las cuales son normales y 40 son tejidos tumorales. Este análisis se centra en los 2000 genes con la mayor intensidad mínima (Alon *et al.*, 1999). Para la experimentación, se realizo un análisis de los primeros 10 *biclusters* encontrados por BiHEA, CC, ISA, OPSM y SPEA2^{LS}. El algoritmo BiMax no fue incluido debido a que resulto imposible encontrar una adecuada parametrización para el paso de discretización. Los parámetros para los algoritmos fueron los mismos a los utilizados en la sección anterior, a excepción de los siguientes: $\delta = 150$; $\alpha = 2.0$, Toda la clasificación ontológica fue realizada con la herramienta Onto-Express (Draghici *et al*., 2003), aplicando una distribución hipergeométrica y referenciando los cálculos con los 2000 genes analizados.

Respecto de los resultados, la figura 5.6 muestra los valores logrados por los métodos anteriores en términos del indicador de enriquecimiento total (ecuación 5.3 con $\alpha = 0.05$) para el enriquecimiento con función molecular y proceso biológico. Es claro que BiHEA es el método que obtiene mejores resultados, dado que la calidad de estos superan a los resultados obtenidos por los algoritmos de referencia en términos del indicador de enriquecimiento total. Solo OPSM obtiene resultados cercanos, mientras que el rendimiento de los otros métodos es significativamente más bajo. Estos resultados son consistentes con los obtenidos sobre los datos sintéticos, mostrando la correctitud del modelo artificial elegido.

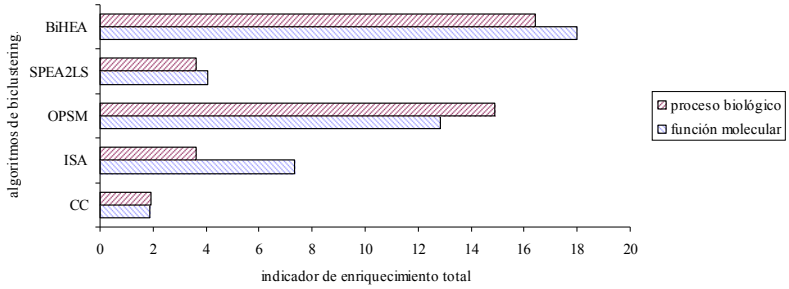


Figura 5.6. Indicador de enriquecimiento total del BiHEA, SPEA2^{LS}, OPSM, ISA y CC para el enriquecimiento con función molecular y proceso biológico de datos de cáncer de colon.

5.2.3. Conclusiones

En esta sección, hemos introducido un nuevo enfoque evolutivo memético para *biclustering* de *microarrays*, hibridizado con un procedimiento de búsqueda local. Este original algoritmo introduce dos características novedosas: la primera fue diseñada a fin de evitar la pérdida de buenas soluciones a través de las generaciones, mientras que mantiene un bajo grado de solapamiento entre los *biclusters* resultantes. La otra característica fue concebida para mantener un nivel de diversidad satisfactorio en el espacio genotípico.

En una primera fase experimental sobre datos sintéticos, los resultados obtenidos por nuestro método superan a los obtenidos por varios algoritmos de *biclustering* de la literatura, especialmente en el caso de *biclusters* coherentes con alto grado de solapamiento. Sin embargo, esto no puede considerarse como una desventaja debido a que, en general, la complejidad regulatoria de un organismo esta lejos del modelo de *biclusters* sin solapamiento. Además, se realizó un análisis sobre datos reales y, en términos de la métrica propuesta, la calidad de los resultados de BiHEA son claramente superiores a los resultados de los métodos de referencia. De hecho, esto muestra la correctitud en el diseño del modelo para construir los *biclusters*, es decir, *bicluster* coherentes que siguen un modelo aditivo. Aunque esto es consistente con los resultados obtenidos en datos sintéticos, se requiere un análisis extensivo sobre varios conjuntos de datos reales para poder confirmar estos resultados.

Finalmente, el marco para comparación de algoritmos de *biclustering* fue refinado mediante la introducción de dos nuevas métricas: la métrica general de pareo de *biclusters* S^* y el indicador de enriquecimiento total E^*. El primero es útil para testear sobre datos sintéticos debido a que los *biclusters* óptimos son conocidos de ante mano. El otro indicador puede ser usado para evaluar el rendimiento de varios métodos sobre datos reales en términos de alguna anotación ontológica de genes específica. Ambas métricas son indispensables en cualquier evaluación de calidad de algoritmos de *biclustering* debido a que proveen de un marco justo en el cual los métodos pueden ser comparados.

5.3. BAT: Una Nueva Herramienta para el Análisis de Biclustering

Como se ha mencionado a lo largo de este capitulo, varios algoritmos para *biclustering* han sido propuestos en la literatura (Cheng y Church, 2000; Bleuler *et al*., 2004; Mitra y Banka, 2006; Divina y Aguilar-Ruiz, 2006; DiMaggio *et al*., 2008). Estas técnicas varían desde enfoques simples *greedy* hasta algoritmos estocásticos evolutivos mucho más complejos. Sin embargo, hay casos en donde el software que implementa tales algoritmos es difícil de usar o no esta disponible. En este sentido, nuestra motivación en este apartado radica en presentar una nueva herramienta para el análisis de *biclustering*, llamada BAT (*Biclustering Analysis*

Toolbox), con el fin de lograr la usabilidad de un algoritmo de *biclustering* novedoso (Gallo *et al*., 2009b).

5.3.1. Principales Características de la Herramienta

BAT implementa al algoritmo BiHEA (Gallo *et al*., 2009b), presentado en la sección anterior, conjuntamente con varios medios para visualización y análisis de *biclustering*. Las principales características de la herramienta de software pueden resumirse como sigue:

Manejo de datos

La información, que consiste en la matriz de expresión de genes completa y en el conjunto de *biclusters* extraídos por BiHEA, se organiza en una estructura de lista que es mostrada en el panel derecho de la interfaz grafica de usuario (figura 5.7a). Esta estructura permite acceder a los conjuntos de datos y a los resultados para previsualizarlos de varias formas. Adicionalmente, provee información relacionada con el tamaño, el residuo cuadrado medio y la varianza de filas de los *biclusters*.

Considerando la naturaleza estocástica de los algoritmos evolutivos, el software es capaz de realizar varias ejecuciones secuencialmente, cada una con diferente semilla. En este caso, la columna en el panel de la izquierda llamada *trials* indica la cantidad de ejecuciones en las que apareció cada *bicluster*.

Preprocesamiento

El archivo de datos de entrada puede ser cualquier archivo en formato CSV (*Comma Separated Values*) que incluya anotaciones sobre los genes y condiciones, o también algún proyecto guardado previamente. La matriz de datos cargada puede luego ser transformada por medio de varios métodos, incluyendo normalización (log_2) y estandarización. Esta ultima técnica, cuando es aplicada sobre las filas, es particularmente útil cuando los investigadores no desean considerar a la varianza de filas como objetivo a optimizar.

Visualización

La matriz de datos de expresión puede mostrarse en tres formas diferentes: como un mapa de calor (*heatmap*, figura 5.7a), como una matriz numérica o en términos de la cobertura de los *biclusters* resultantes (figura 5.7b). Las anotaciones sobre las condiciones se muestran en el tope mientras que las anotaciones sobre los genes son listadas sobre el lado izquierdo. Adicionalmente, los *biclusters* pueden ser visualizados en forma de mapa de calor, como sub-matriz numérica, o como una colección de perfiles de expresión (figura 5.7c). Los perfiles de expresión muestran el comportamiento de aquellos genes que están agrupados en un *bicluster* en

el cual, para cada gen, una línea coloreada conecta los valores de expresión para las diferentes condiciones. Finalmente, en el caso de los mapas de calor, los colores pueden ser ajustados a fin de mejorar la visualización.

Posprocesamiento

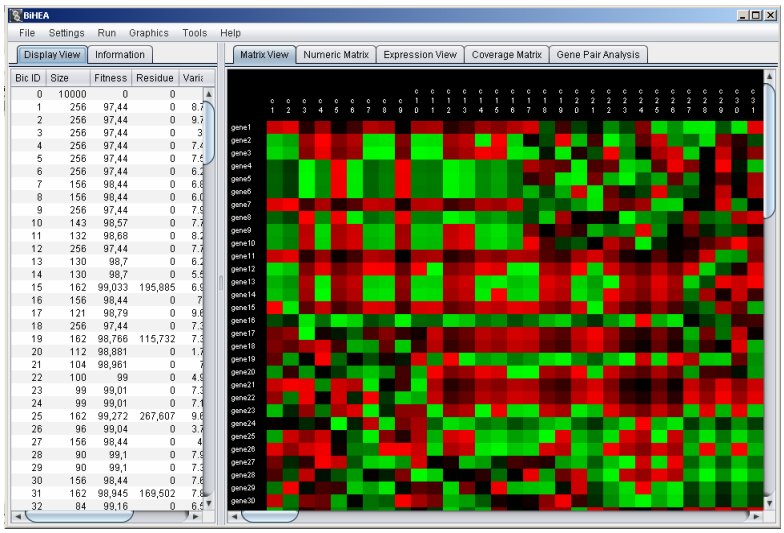
A fin de analizar los resultados, el software ofrece la posibilidad de realizar un análisis de pares de genes. Mas específicamente, para cada par de genes, se calcula la frecuencia con la cual esos genes aparecen juntos en el mismo *bicluster*. Este numero de co-ocurrencias detecta a aquellos genes que pueden estar funcionalmente relacionados. Además, como se menciono anteriormente, el grado de cobertura de los *biclusters* resultantes con respecto a la matriz de expresión de genes puede ser visualizado como una matriz en escala de grises (figura 5.7b). Esto es útil a fin de proveer una visión global de las áreas en la matriz de datos de expresión sobre las que están ubicados los *biclusters*.

Las figuras, gráficos y resultados pueden ser exportados de BAT para uso posterior. En el caso de las figuras y gráficos, también es posible ajustar la resolución en ppp de las imágenes exportadas acorde a las necesidades del usuario. Todo el trabajo realizado durante una sesión de BAT puede ser guardado como un proyecto y restaurado posteriormente.

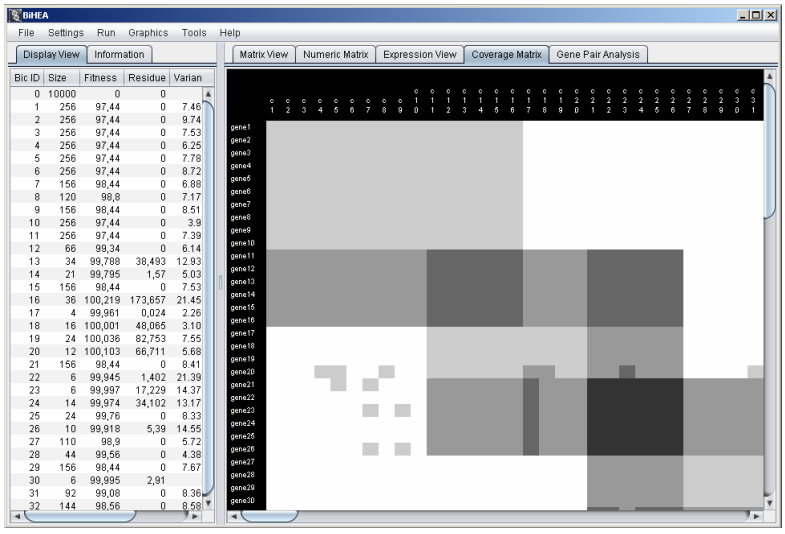
5.3.2. Conclusiones finales y discusión

En esta sección, hemos introducido a BAT: una herramienta de software de matrices de datos de expresión de genes que implementa al algoritmo BiHEA. Como frecuentemente ocurre en bioinformática cuando emerge un nuevo algoritmo, el software que lo implementa es difícil de encontrar y usar, si es que esta disponible. Este trabajo esta enfocado principalmente en esta cuestión, proveyendo de un marco completo de trabajo en el cual los biólogos pueden confiar a fin de realizar análisis de datos de expresión de genes a través de un algoritmo reciente.

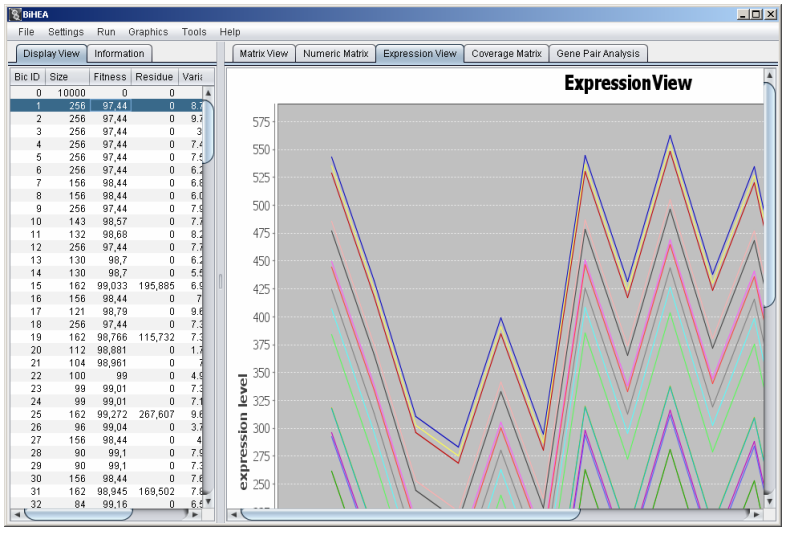
El software, código fuente, manuales y varios ejemplos están disponibles libremente en http://lidecc.cs.uns.edu.ar a fin de ofrecer todo el soporte necesario para el usuario. Adicionalmente, la implementación es independiente del sistema operativo, por lo que funcionará en la mayoría de las computaras personales.



(a)



(b)



(c)

Figura 5.7. Interfaz grafica de usuario del software BAT. Fig. 5.7a: Visualización del mapa de calor de una matriz de datos de expresión. Fig. 5.7b: Cobertura de los *biclusters* resultantes en una matriz de datos de expresión. Fig. 5.7c: Perfil de expresión de un *bicluster*.

5.4. Sumario

En este capitulo introdujimos las principales contribuciones es esta tesis en relación al *biclustering* de datos de expresión de genes. La primer contribución esta relacionada al empleo de algoritmos genéticos meméticos multi-objetivo basados en Pareto, lo que dio lugar al artículo publicado en Gallo *et al*. (2009a). La metodología de inferencia presentada en este trabajo incorpora características novedosas como la posibilidad de inferir genes co-expresados pero cuyo nivel de expresión se da de forma opuesta. También permite orientar la búsqueda genética en relación al compromiso existente entre los múltiples objetivos del algoritmo evolutivo por medio de un parámetro establecido previamente por el usuario. Además, el trabajo fue desarrollado bajo la plataforma PISA (Bleuler *et al*., 2003), lo que permitió la evaluación y comparación de varios algoritmos evolutivos muti-objetivo reconocidos y su correcta validación por medio de herramientas estadísticas propias de la optimización multi-objetivo (Knowles *et al*., 2005). Por ultimo, los resultados fueron comparados con los obtenidos por otro método perteneciente al estado del arte (Mitra y Banka, 2006), siendo los del algoritmo propuesto superadores en relación a este.

A pesar de haber obtenido resultados importantes en el estudio mencionado previamente, ciertas características presentes en el problema de *biclustering* (como el solapamiento de *biclusters*) hacen que la utilización de algoritmos de optimización multi-objetivo de propósito general como los disponibles en PISA (Bleuler *et al*., 2003) no sea óptima. La razón es la poca flexibilidad de estos algoritmos evolutivos para incorporar este tipo de características en el proceso de búsqueda. De este modo, se comenzó el desarrollo de un nuevo algoritmo evolutivo memético diseñado específicamente para el problema de *biclustering* de datos de expresión de genes, lo que dio lugar al articulo publicado en Gallo *et al*. (2009b). Este algoritmo, denominado BiHEA, incorpora mecanismos novedosos que evitan la perdida de buenas soluciones a través de las generaciones, manteniendo un bajo solapamiento entre los *biclusters* con un nivel de diversidad satisfactorio en el espacio genotípico. La evaluación del algoritmo BiHEA se realizó tanto con datos sintéticos como con datos reales, obteniendo resultados superadores en comparación a los obtenidos con el método presentado previamente (Gallo *et al*., 2009a) y en comparación a otros algoritmos (Cheng y Church, 2000; Ihmels *et al*., 2004; Zimmermann *et al*., 2006; Ben-Dor *et al*., 2002) pertenecientes al estado del arte de esta línea de investigación. Estos buenos resultados llevaron al desarrollo y publicación (Gallo *et al*., 2010a; Gallo *et al*., 2010b) de un software integrador para el análisis de datos de *microarrays*, que permite la utilización del algoritmo BiHEA en un entorno de interfaz gráfica amigable con numerosas características interesantes para los usuarios biólogos. Entre las principales aptitudes de esta herramienta podemos mencionar las facilidades brindadas para el manejo de datos, pre-

procesamiento y mecanismos de visualización de *microarrays* y resultados, y las herramientas de post-procesamiento que permiten realizar varios análisis sobre los resultados obtenidos de manera automática.

Capítulo 6

Inferencia de Redes Regulatorias de Genes Basadas en Reglas de Asociación

El modelado de redes de genes usa datos de perfiles de expresión de los mismos para describir el comportamiento fenotípico del sistema bajo estudio. A fin de reconstruir la red, el procedimiento involucra alterar la red de genes de alguna forma, observar la salida, y usar métodos computacionales para inferir los principios subyacentes de la red. En este contexto, los métodos de minería de datos son enfoques apropiados para realizar la ingeniería inversa y, en particular, estas estrategias de reconstrucción pueden ser beneficiadas por la aplicación de técnicas de extracción de reglas de asociación. Básicamente, una Regla de Asociación (RA) establece un link causal entre dos o mas variables, donde la semántica y la interpretación de la regla depende de los datos de entrada y de los mecanismos empleados para inferir la asociación. Las RAs han sido extensamente usadas para descubrir relaciones interesantes entre variables en grandes conjuntos de datos (Ceglar y Roddick, 2006). En bioinformática, estos métodos pueden usarse para revelar asociaciones biológicamente relevantes entre genes, en diversas condiciones experimentales u observaciones temporales, a partir de diferentes muestras de *microarrays* (Creighton y Hanash, 2003; Carmona-Saez *et al.*, 2006; Gallo *et al.*, 2011a).

Este capítulo se enfoca en la regulación de genes y en las formas en que los datos de transcriptoma pueden ser usados para descifrar las relaciones complejas entre los genes que conforman una red regulatoria de genes. En particular, se describen los principales tópicos que deben considerarse en el campo de la minería de RA para la ingeniería inversa de RRGs, y se presenta el estado del arte de las técnicas disponibles actualmente en la literatura. La organización del capítulo es como sigue. En la sección 6.2 se presentan y discuten conceptos centrales acerca del minado de RA para la reconstrucción de RRGs conjuntamente con otros aspectos relevantes. En la sección 6.3, se reseñan diferentes enfoques de minería de datos usados para la inferencia de RA. Finalmente, en la sección 6.4 se resumen las principales conclusiones y se presentan las observaciones finales.

6.2 Minería de Datos e Inferencia de RRGs basadas en RAs

Una RRG es un tipo de red regulatoria causal, mientras que otros tipos de redes incluyen a las redes de proteínas y procesos metabólicos (McShan *et al.*, 2004). Las RRGs tienen una estructura desordenadamente robusta como consecuencia de la evolución (Sterelny y Griffiths, 1999). Una RRG puede ser representada por un grafo dirigido (Silvescu y Honavar, 2001), en el

cual el conjunto de vértices N representa los genes y el conjunto de arcos E describe las relaciones regulatorias entre cada par de genes. Las RRGs pueden ser también modeladas como grafos no dirigidos (Toh y Horimoto, 2002), aunque la verdadera red regulatoria subyacente es representada mejor como grafo dirigido. Cada arco puede también estar decorado con información adicional, como el tipo de regulación (activación o inhibición) y/o el retraso en la regulación, entre otras. La figura 6.1 muestra un ejemplo de una RRG representada como grafo dirigido.

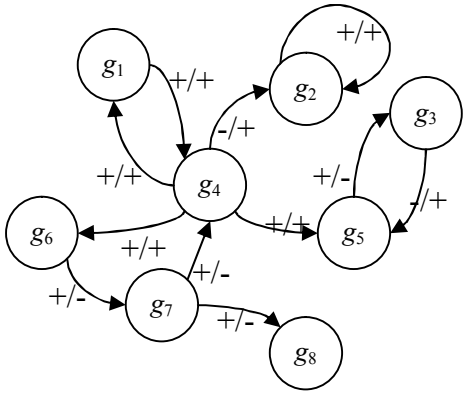


Figura 6.1. RRG representada como grafo dirigido. La dirección del arco indica el rol regulatorio (regulador o blanco) de los genes en cada interacción. Un símbolo + (-) a la izquierda de la etiqueta en el arco indica expresión (no expresión) del gen regulador, mientras que un símbolo + (-) en el lado derecho indica activación (inhibición) del gen blanco.

Otra forma común de representar a una RRG es por medio de una lista de RAs. Supongamos que tenemos un conjunto de genes I con n cantidad de genes. De esta manera, cada interacción entre genes es representada como una regla de la forma $g_r \rightarrow g_i$, donde g_r, $g_i \in I$, y g_r es el gen regulador mientras que g_i es el gen blanco. De manera similar a la representación de grafo, las RAs pueden contener información adicional respecto de la interacción entre los genes. Es más, dado un grafo que representa a una RRG, se puede obtener de forma directa una lista equivalente de RAs por medio del conjunto de arcos del grafo, con una RA por cada arco. Si consideramos la RRG de la figura 6.1, la siguiente lista de RAs representa la misma RRG: $+g_1 \rightarrow +g_4$, $+g_4 \rightarrow +g_1$, $-g_4 \rightarrow +g_2$, $+g_2 \rightarrow +g_2$, $+g_4 \rightarrow +g_5$, $+g_5 \rightarrow -g_3$, $-g_3 \rightarrow +g_5$, $+g_4 \rightarrow +g_6$, $+g_6 \rightarrow -g_7$, $+g_7 \rightarrow -g_4$, $+g_7 \rightarrow -g_8$. En el resto de este capítulo, haremos referencia indistintamente a arcos como a RAs, dado que ambos representan el mismo tipo de información.

Una RRG puede ser considerada como un sistema estocástico de componentes discretos. Sin embargo, modelar a una RRG de esta forma no es tratable en forma computacional (Fogelberg y Palade, 2009). Por esta razón, los sistemas estocásticos no serán considerados en este capítulo, y así, el principal foco estará puesto en los modelos discretos de RRG. En estos modelos, una

RRG es un conjunto de genes I y un conjunto de funciones F tal que hay una función por cada gen: $\forall g_i \in I,\ \exists f_i : f_i \in F$. Cada una de estas funciones toma como entrada todo o un subconjunto de I como parámetro, y la salida corresponde a un valor discreto de un dado conjunto de estados de genes. Usando este modelo, se pueden inferir y representar las características más importantes de la relaciones regulatorias.

La minería de datos puede ser usada tanto para inferir epístasis (determinar que genes interactúan) o para crear modelos explicatorios de una red. La epístasis se identifica tradicionalmente por medio de experimentos de letalidad sintética (Tong *et al*., 2001; Giaever *et al.*, 2002; Lum *et al.*, 2004) y Y2H (*yeast two-hybrid*). Los enfoques de minería de datos son necesarios en estos casos debido a que los datos contienen a menudo mucho ruido y, como sucede con Perl-3, los cambios fenotípicos pueden ser invisibles a menos que varios genes sean bloqueados. Con frecuencia es mejor inferir modelos explicativos de una red, con aplicaciones más útiles al entendimiento biológico, ingeniería genética y diseño farmacéutico. De esta forma, con el fin de inferir un modelo explicativo de una RRG con un enfoque de minería de datos, es necesario tomar en cuenta varias consideraciones. Primero, se necesitan responder algunas preguntas respecto del tipo de dato biológico a partir del cual será inferido el modelo: ¿Qué tipo de información representa? ¿Son datos de estados en equilibrio o series de tiempo? ¿Son los datos inherentemente ruidosos o no? Segundo, dado que el objetivo global es inferir una RRG con un enfoque de minería de datos, la discretización puede jugar un rol importante en todo el proceso: el algoritmo, ¿requiere de discretización de los datos? Y si esto es cierto, ¿cuántos estados son necesarios para representar el comportamiento subyacente de los genes? ¿Cómo pueden obtenerse estos estados? Además, hay varias otras características relevantes que deben considerarse: el patrón de cardinalidad de las asociaciones (cantidad de genes que pueden ser asociados mediante una regla), la manera en la que es modelado el comportamiento temporal (asociaciones diferidas en el tiempo), y como conciliar las reglas extraídas a partir de múltiples fuentes de datos. Estos tópicos pueden afectar la complejidad computacional del algoritmo de inferencia y la factibilidad del modelo inferido. Finalmente, la validación estadística y biológica de las RAs obtenidas es el paso más importante en la inferencia de una RRG, dado que determina la verosimilitud biológica del modelo inferido. La figura 6.2 resume gran parte de las cuestiones que deben considerarse a fin de inferir RRGs con enfoques de minería de datos. En las siguientes secciones se discutirán todos estos tópicos.

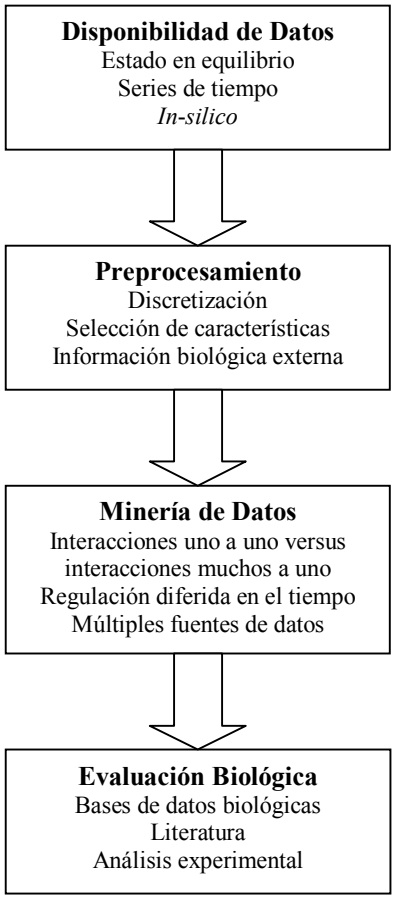


Figura 6.2. Resumen de varias cuestiones que se deben considerar a fin de inferir RRGs con enfoques de minería de datos.

6.2.1. Tipos de Datos Biológicos Usados para la Inferencia de RRG

Existen algunos tipos de datos biológicos disponibles para abordar el modelado de la red en bioinformática. El tipo de datos más ampliamente utilizado por las técnicas de minería de datos en la reconstrucción de RRGs es el de expresión génica o de expresión de genes. Como se ha mencionado en capítulos anteriores, los datos de expresión génica representan la actividad de cada gen $g_i \in N$, medida como la concentración de ARNm, ya que la actividad de transcripción no se puede medir directamente. Por lo tanto, dado que la interacción regulatoria o fenotípica de proteínas puede consumir algo de ARNm antes que se lleve a cabo la regulación del gen (Segal *et al*., 2005), esto podría parecer ser una medida inexacta de la actividad de los genes (Segal *et al.*, 2003). Aún más, una proteína puede unirse a una región promotora sin producir ningún efecto regulador (Hartemink *et al*., 2002). Además, la mayoría de los genes no están involucrados en la mayoría de los procesos celulares (Jiang *et al*., 2004). Esto implica que varios de los genes incluidos en la muestra podrían parecer variar aleatoriamente sus niveles de expresión. Sin embargo, si el conjunto de datos es comprensible y la única preocupación es la inferencia de relaciones regulatorias, estas influencias no son importantes (Fogelberg y Palade, 2009). El aumento en la cantidad de datos o la realización de inferencia dirigida puede evitar el

problema de los genes irrelevantes. Las influencias no génicas que no se modelan son análogas a las variables intermedias ocultas en las redes bayesianas (Hartmink *et al.*, 2002). Una influencia como esta no distorsiona la relación regulatoria o la exactitud en la predicción del modelo inferido (Fogelberg y Palade, 2009).

6.2.1.1. Tipos de datos de expresión

Hay dos tipos de datos de expresión génica: niveles de expresión en equilibrio (estado estacionario) que corresponden a una situación estática, y niveles de expresión de series de tiempo que se recogen durante una fase fenotípica como el ciclo celular (Spellman *et al*, 1998). Los datos de expresión génica son obtenidos generalmente por medio de *microarrays* o alguna tecnología similar. Como se mencionó con anterioridad, un *microarray* es un portaobjetos de pre-preparados, divididos en celdas. Cada celda está recubierta individualmente con un producto químico que emite fluorescencia cuando se mezcla con el ARNm generado por un sólo gen. El brillo de cada celda se utiliza como una medición del nivel de ARNm y por lo tanto del nivel de expresión génica.

En general, los datos de expresión en estado estacionario están representados por una matriz $\boldsymbol{X}'$ de $n \times m$, donde las filas representan los n genes de la muestra y las columnas representan las m condiciones experimentales diferentes (o réplicas o ambos). Las diferentes condiciones experimentales se refieren a los diferentes tejidos, temperaturas, compuestos químicos o cualquier otra situación que pueda producir un comportamiento regulatorio diferente entre los genes incluidos en la muestra. Cada elemento x_{ij} de $\boldsymbol{X}'$ contiene el valor de expresión del gen x_i en la muestra o condición experimental j. Por otro lado, la serie de tiempo codificada en el conjunto de datos de expresión de genes, está representada por medio de una matriz de datos de expresión génica, $\boldsymbol{X}'$, donde las filas y las columnas representan los genes y los puntos temporales, respectivamente. En este caso, los datos de series de tiempo se recogen mediante el uso de mutantes sensibles a la temperatura (o químicos) para pausar el proceso fenotípico mientras que se realiza un *microarray* en una muestra. Por lo tanto, las diferentes columnas representan los valores de expresión de cada gen muestreado en diferentes momentos bajo la misma condición experimental durante alguna fase fenotípica. Los intervalos de muestreo en el que se muestrean los genes son determinados por el investigador acorde a la naturaleza del estudio, y no se toman necesariamente en intervalos equidistantes.

Como hemos mencionado con anterioridad, los *microarrays* pueden contener tanto ruido biológico (Nykter *et al*., 2006) como ruido técnico (Jiang *et al.,* 2004). El primero de ellos es la incertidumbre biológica en forma de ruido intrínseco y extrínseco. El segundo es el ruido experimental debido al proceso complejo de medición, que va desde las condiciones de

hibridación a las técnicas de procesamiento de imágenes de *microarrays*. Sin embargo, la magnitud y el impacto del ruido es un tema muy debatido, y esto depende de la tecnología exacta utilizada para recoger las muestras. Investigaciones recientes (Klebanov y Yakovlev, 2007; Fogelberg y Palade, 2009) sostienen que tanto la magnitud como el impacto del ruido han sido "gravemente exagerados".

6.2.2. Discretización de Expresión de Genes

La discretización de datos, también conocida como *binning*, es una técnica usada con frecuencia en las ciencias de la computación y la estadística aplicada al análisis de datos biológicos. La discretización de los datos reales en un pequeño número de valores finitos es requerida a menudo por los algoritmos de aprendizaje automático (Dougherty *et al*., 1995), las aplicaciones de redes bayesianas (Friedman y Goldszmidt, 1996), y cualquier algoritmo de modelización mediante modelos de estados discretos. Una ventaja importante del uso de estados discretos es que en el proceso se absorbe una parte significativa del ruido.

No obstante, la selección de un enfoque razonable de discretización no es una tarea trivial. En general, los procesos de discretización implican pérdida de información, y diferentes estrategias pueden dar como resultado modelos de estado discreto distintos. Por lo tanto, la semántica biológica y la interpretación de los modelos resultantes difieren, incluso cuando los datos de valores reales subyacentes son siempre los mismos. Por esta razón, en la elección de un método de discretización, se debe tener en cuenta la naturaleza intrínseca de los datos biológicos, así como las características particulares del método computacional que hará uso de estos datos discretizados.

Se han propuesto varias técnicas de discretización en la literatura. La discretización binaria es la forma más sencilla de discretización de datos, que se utiliza, por ejemplo, para la construcción de modelos de redes booleanas para RRGs (Kauffman, 1969; Albert y Othmer, 2003). Los datos de expresión son discretizados en dos estados cualitativos como presente o ausente. Una desventaja obvia de la discretización binaria es que generalmente causa la pérdida de una gran cantidad de información. En tal sentido, se han desarrollado y estudiado modelos discretos y técnicas de modelización que permiten múltiples estados (Thieffry y Thomas, 1998; Laubenbacher y Stigler, 2004).

6.2.2.1. Problema de Discretización

Sea $\boldsymbol{X}'$ una matriz de expresión génica de n filas por m columnas, donde x'_{ij} representa el nivel de expresión del gen g_i bajo la condición j. La matriz $\boldsymbol{X}'$ está definida por su conjunto de filas, I, y su conjunto de columnas, J. Por otra parte, denotemos a x'_{IJ} como el valor medio en la

matriz de expresión $\boldsymbol{X'}$ y, x'_{iJ} y x'_{Ij} como el promedio de la fila i y de la condición j, respectivamente. Sea H_{IJ} el máximo valor en la matriz de expresión $\boldsymbol{X'}$, y h_{iJ} y h_{Ij} el valor máximo de la fila i y de la columna j, respectivamente. De la misma manera, sea M_{IJ} el valor de la mediana en la matriz de expresión $\boldsymbol{X'}$, y M_{iJ} y M_{Ij} el valor de la mediana de la fila i y la columna j, respectivamente.

Una matriz de datos $\boldsymbol{X'}$ discretizada es un mapeo en donde cada elemento en $\boldsymbol{X'}$ se asigna a un elemento de un alfabeto $\sum$, que consiste en un conjunto de símbolos diferentes que representan un nivel de activación distinto. En el caso más simple, $\sum$ puede contener sólo dos símbolos, un símbolo que se utiliza para la regulación (o activación) y otro símbolo para la no-regulación (o inhibición). En este caso, la matriz de expresión generalmente se transforma en una matriz binaria, donde 1 significa regulación y 0 significa que no hay regulación. Otra opción ampliamente utilizada es considerar un conjunto de tres símbolos de discretización, {-1, 1, 0}, lo que significa inhibición, activación o sin cambio, respectivamente. Sin embargo, los valores en la matriz $\boldsymbol{X'}$ pueden ser discretizados a un número arbitrario de símbolos. Después del proceso de discretización, la matriz $\boldsymbol{X'}$ se transforma en la matriz $\boldsymbol{X}$ y $x_{ij} \in \sum$ representa el valor discretizado del nivel de expresión del gen g_i bajo la condición j. Se han utilizado varias técnicas de discretización en el análisis de datos de expresión. De acuerdo con Madeira y Oliveira (2005), estas técnicas se pueden agrupar en dos categorías de alto nivel:

1. Discretización usando valores de expresión absolutos
2. Discretización usando variaciones de expresión entre puntos de tiempo

Los enfoques que pertenecen a la primer categoría se pueden utilizar en los datos de expresión, en general, y discretizan directamente los valores absolutos de expresión génica utilizando diferentes técnicas. El segundo conjunto de enfoques, sólo aplicable a los datos de expresión de series de tiempo, calcula las variaciones entre dos puntos de tiempo consecutivos y luego discretizan estas variaciones. En los apartados siguientes se detallarán los principales criterios de discretización.

6.2.2.2. Discretización Usando Valores Absolutos

Comencemos con la discretización usando el promedio y el desvío estándar.

Discretización usando el promedio y el desvío estándar

Un método de discretización directo discretiza la matriz de expresión génica $\boldsymbol{X'}$ con el valor promedio de expresión (Li *et al*, 2006.), o con el promedio junto con el desvío estándar de los

valores de expresión (Koyuturk *et al*, 2004; Madeira y Oliveira, 2005; Park y Szpankowski, 2005). Los valores límite para la discretización se pueden calcular utilizando todos los valores en la matriz de expresión, es decir, empleando el promedio general del nivel de expresión y su desvío estándar. Otra posibilidad es calcular el promedio y el desvío estándar para cada fila o cada columna en la matriz.

Cuando el objetivo es discretizar la matriz de expresión en una matriz binaria con dos símbolos, uno para la regulación y otro para no-regulación (por ejemplo, 1 y 0), generalmente se usa solo el valor promedio de expresión. La discretización puede ser calculada utilizando todos los valores de la matriz, por filas o por columnas, utilizando una de las siguientes expresiones:

$$x_{ij} = \begin{cases} 1 & si \quad x'_{ij} \geq x'_{IJ} \\ 0 & caso\ contrario \end{cases}. \tag{6.1}$$

$$x_{ij} = \begin{cases} 1 & si \quad x'_{ij} \geq x'_{iJ} \\ 0 & caso\ contrario \end{cases}. \tag{6.2}$$

$$x_{ij} = \begin{cases} 1 & si \quad x'_{ij} \geq x'_{Ij} \\ 0 & caso\ contrario \end{cases}. \tag{6.3}$$

Otra posibilidad es discretizar la matriz utilizando tres símbolos, por ejemplo -1, 0 y 1, que significan inhibición, activación o sin cambio. En este caso, el valor promedio de expresión normalmente se combina con su desvío estándar. Sea α el parámetro que se utiliza para ajustar el desvío estándar deseado a partir del promedio, y sean σ_{IJ}, σ_{iJ} y σ_{Ij} los desvíos estándar de los valores totales de la matriz, de la fila i y de la columna j, respectivamente. Entonces, la discretización se puede realizar utilizando una de las siguientes ecuaciones (Koyuturk *et al*, 2004; Parque Szpankowski, 2005; Madeira y Oliveira, 2005):

$$x_{ij} = \begin{cases} -1 & si \quad x'_{ij} < x'_{IJ} - \alpha\sigma_{IJ} \\ 1 & si \quad x'_{ij} > x'_{IJ} - \alpha\sigma_{IJ} \\ 0 & caso\ contrario \end{cases}. \tag{6.4}$$

$$x_{ij} = \begin{cases} -1 & si \quad x'_{ij} < x'_{iJ} - \alpha\sigma_{iJ} \\ 1 & si \quad x'_{ij} > x'_{iJ} - \alpha\sigma_{iJ} \\ 0 & caso\ contrario \end{cases}. \tag{6.5}$$

$$x_{ij} = \begin{cases} -1 & si \quad x'_{ij} < x'_{Ij} - \alpha\sigma_{Ij} \\ 1 & si \quad x'_{ij} > x'_{Ij} - \alpha\sigma_{Ij} \\ 0 & caso\ contrario \end{cases}. \tag{6.6}$$

Discretización basada en el principio de igual frecuencia

El proceso de discretización basado en el principio de igual frecuencia considera un número dado de símbolos en los que se pueden discretizar los valores de expresión. A continuación, los puntos de datos se dividen de tal manera que existe el mismo número de puntos de datos por símbolo, discretizando en consecuencia los valores de expresión para el símbolo correspondiente. Este proceso se puede aplicar a un número arbitrario de símbolos. Cuando sólo se consideran dos símbolos, el proceso es equivalente al procedimiento que se realiza para llevar a cabo la discretización de los valores por medio del valor de la mediana. Al igual que antes, el principio de igual frecuencia puede ser aplicado usando todos los valores de expresión en la matriz, los valores de expresión por fila o los valores de expresión por columna (Lonardi *et al.*, 2004).

Discretización basada en clustering

Otra técnica común de discretización para la matriz de expresión génica $\boldsymbol{X}'$ se basa en el *clustering* o agrupamiento (Jain y Dubes, 1988). En general, el algoritmo de *clustering* se aplica a cada fila (perfil de expresión génica) realizando, en consecuencia, la discretización respecto de la partición devuelta por el algoritmo. El empleo de *clustering* en la discretización permite la consideración de múltiples estados en forma directa, aunque puede introducir algún costo computacional adicional dependiendo de la técnica de *clustering* seleccionada.

Uno de los algoritmos de *clustering* más comunes es el de agrupamiento por *k-medias* desarrollado por MacQueen (1967). El objetivo del algoritmo de *k-medias* es reducir al mínimo la disimilitud en los elementos dentro de cada grupo, mientras maximiza este valor entre los elementos de diferentes agrupaciones. El algoritmo toma como entrada un conjunto de puntos S a ser particionado y un número entero k fijo, particionando a S en k subconjuntos eligiendo un conjunto de k de centroides de grupo. La elección de los centroides determina la estructura de la partición, ya que cada punto en S se asigna al centroide más cercano. A continuación, para cada grupo, los centroides son re-calculados en base a qué elementos están contenidos en el grupo. Estos pasos se repiten hasta que se logra la convergencia. Muchas de las aplicaciones del agrupamiento por *k-medias*, como el *MultiExperiment Viewer* (Saeed *et al*., 2003), comienzan computando una partición al azar de k grupos y calculando sus centroides. Como consecuencia de esto, se puede obtener una partición diferente de S cada vez que se ejecuta el algoritmo. Para el caso especial en que se consideran sólo 2 estados en la discretización ($k = 2$), la solución óptima se puede calcular de manera optimizada debido al ordenamiento total de los elementos en el perfil de expresión de genes (Gallo *et al.*, 2011a).

6.2.2.3. Discretización Usando Variaciones de Expresión Entre Puntos de Tiempo

Se han propuesto varias técnicas de discretización sobre la base de las transiciones en los estados de expresión entre puntos de tiempo sucesivos. Estas técnicas suelen tener en cuenta ya sea dos o tres estados, según lo indicado anteriormente. En general, la discretización de una matriz $\boldsymbol{X}'$ utilizando variaciones de expresión entre puntos de tiempo produce una matriz de datos discretos $\boldsymbol{X}$ con m - 1 muestras.

La discriminación de Transición de Estado (Möller-Levet *et al.*, 2003) es una técnica de discretización que utiliza dos símbolos. Después de la normalización de los datos de expresión de $\boldsymbol{X}'$ por medio de *z-score* (los perfiles de expresión se escalan a una media cero y a un desvío estándar de una unidad), cada perfil de expresión génica es discretizado utilizando dos transiciones de estado:

$$x_{ij} = \begin{cases} 1 & si \quad x'_{ij} - x'_{i(j-1)} \geq 0 \\ 0 & caso\ contrario \end{cases}. \tag{6.7}$$

Otra técnica de discretización puede llevarse a cabo computando las variaciones entre los instantes de tiempo sucesivos como antes, pero teniendo en cuenta que estas variaciones son significativas cuando se supera un umbral preestablecido (Ji y Tan, 2004; Ji y Tan, 2005). Durante el proceso de discretización, la matriz de expresión se transforma en una matriz $\boldsymbol{X}$ de $n \times (m - 1)$ que refleja la tendencia de cambio en el tiempo de cada valor de expresión de genes. Se pueden considerar un número arbitrario de tendencias cambiantes conduciendo la discretización de la matriz a un conjunto de símbolos Σ. Cuando se consideran tres posibles tendencias (Ji y Tan, 2004; Ji y Tan, 2005), un nivel de expresión puede aumentar desde el punto del tiempo t_i a t_{i+1}, puede disminuir, o pueden permanecer sin cambio. Estas tendencias son entonces discretizadas en tres símbolos (aumento, reducción o sin cambios, respectivamente). En este caso, y con este conjunto de símbolos, la matriz de datos discretos $\boldsymbol{X}$ se obtiene después de dos pasos. En el primer paso, la matriz de expresión $\boldsymbol{X}'$ se transforma en una matriz $\boldsymbol{X}''$ de variaciones de $n \times (m - 1)$ tal que:

$$x''_{ij} = \begin{cases} \dfrac{x'_{i(j-1)} - x'_{ij}}{\left|x'_{ij}\right|} & si \quad x'_{ij} \neq 0 \\ 1 & si \quad x'_{ij} = 0 \wedge x'_{i(j-1)} > 0 \\ -1 & si \quad x'_{ij} = 0 \wedge x'_{i(j-1)} < 0 \\ 0 & si \quad x'_{ij} = 0 \wedge x'_{i(j-1)} = 0 \end{cases}. \tag{6.8}$$

Una vez que se genera la matriz $\boldsymbol{X''}$, se obtiene la matriz de datos discretos final $\boldsymbol{X}$, también con n filas y (m-1) columnas, en un segundo paso agrupando los valores de la matriz transformada y teniendo en cuenta el umbral $t > 0$ como sigue:

$$x_{ij} = \begin{cases} 1 & si \quad x''_{ij} \geq t \\ -1 & si \quad x''_{ij} \leq -t \\ 0 & caso\ contrario \end{cases} . \tag{6.9}$$

6.2.3. Asociaciones Uno a Uno vs. Muchos a Uno

La aridad de las RAs inferibles con métodos de minería de datos tiene implicaciones biológicas y computacionales. Desde el punto de vista biológico, la estructura de la RRG parece no ser ni al azar ni rígidamente jerárquica, pero sí de escala libre. Esto significa que la distribución de probabilidad para el grado de salida, k_{out}, sigue una ley de potencia (Kauffman, 1991; Barabási y Oltvai, 2004). En otras palabras, la probabilidad de que un gen g_i regule otros k genes es $p(k) \approx k^{\lambda}$, donde, usualmente, $\lambda \in \{2, 3\}$. En el análisis de Kauffman (1991) con respecto a las redes de escala libre de operadores booleanos, se demostró que algunos sistemas muy desordenados espontáneamente "cristalizan" en un alto grado de orden, contribuyendo a la "capacidad de evolución" y adaptabilidad de las RRGs (Barabási y Oltvai, 2004).

En el lado computacional, estas distribuciones sobre k_{out} y k_{in} (grado de entrada) significan que se han hecho un número de suposiciones en investigaciones previas con el fin de simplificar el problema y hacer que sea más manejable. Por ejemplo, la distribución exponencial sobre k_{in} significa que la mayoría de los genes están regulados sólo por unos pocos otros genes. Por desgracia, este promedio no es un máximo. Esto significa que las técnicas que limitan estrictamente el k_{in} a una constante arbitraria (Silvescu y Honavar, 2001; Tegner *et al.*, 2003) podrían no ser capaces de deducir todas las redes, comprometiendo así su poder explicativo.

6.2.3.1. Funciones Regulatorias "Uno a Uno"

Las funciones de regulación "uno a uno" se refieren a un gen g_i (blanco) que sólo está regulado por un gen g_r (regulador), es decir, una relación apareada o por pares. En este caso, la función regulatoria, $f_i(r)$, puede ser más o menos lineal, sigmoidal o puede adoptar cualquier otra forma. Además, la fuerza del efecto de g_r sobre g_i puede variar de fuerte a débil. Sin embargo, esta última característica puede ser modelada sólo si se consideran varios estados discretos. Además, en el modelado de la función regulatoria se pueden considerar otros factores no génicos sobre g_i, denotados por ϕ_i. En esta situación, la función de regulación puede ser expresada como $g_i' = f_i\ (r,\ \phi_i)$ (véase por ejemplo el trabajo de Marnellos y Mjolsness (1998)).

Sin embargo, en la mayoría de las situaciones y por razones de simplicidad, se supone que $\frac{\delta f}{\delta \phi} = 0$.

La relación regulatoria entre el gen g_r y el gen g_i puede ser tanto de activación o inhibición. Aún más, el gen g_r puede tanto activar al gen g_i cuando esta sobre expresado e inhibir a g_i cuando no está expresado, y viceversa. Para el caso en donde aplica siempre la regulación opuesta (cuando el regulador está sobre expresado o cuando el regulador no está expresado), el proceso químico subyacente no está claro. Por otra parte, en los modelos inferidos de Perkins *et al*. (2006), este tipo de regulación no pudo ser verificada biológicamente. En cualquiera de los casos, este tipo de regulación es especialmente compleja y no es evolutivamente robusta, lo que significa que la ocurrencia de tal relación es poco probable en términos biológicos. Otras propiedades de los organismos también influyen en los tipos de relaciones regulatorias que pueden inferirse. Por ejemplo, los genes inhibidores son más comunes en organismos procariotas que en organismos eucariotas (Herrgard *et al*., 2003).

6.2.3.1. Funciones Regulatorias "Muchos a Uno"

Un gen g_i puede ser regulado por varios genes, en cuyo caso la función de regulación es por lo general más compleja. En particular, la regulación de genes de un organismo eucariota puede ser enormemente compleja (FitzGerald *et al.*, 2006), en el que la función de regulación puede ser una función definida por casos (Spirtes *et al*., 2003; Cui *et al*., 2005; Segal *et al*., 2005). La complejidad surge debido a la compleja regulación indirecta, de varios niveles y de varias etapas del proceso biológico subyacente a la regulación de genes. El proceso regulatorio fue introducido en el capitulo 2 y se detalla en trabajos como (Vohradský, 2001; de Jong, 2002; Driscoll y Gardner, 2006). Por último, algunas de las relaciones regulatorias lógicamente posibles parecen ser poco probables. Por ejemplo, parece que las relaciones de o-exclusivo son biológica y estadísticamente improbables (Liang *et al*., 1998).

6.2.4. Inferencia de RRGs a partir de Múltiples Fuentes de Datos

La mayoría de las primeras investigaciones en el aprendizaje automático de las redes de regulación transcripcional emplean únicamente datos de expresión de genes. Estudios recientes de simulación, sugieren que las redes regulatorias inferidas únicamente a partir de datos de expresión de genes pueden ser oscurecidas considerablemente por la recuperación de interacciones espurias, en el caso de que el número de observaciones sea pequeño (Husmeier, 2003). La integración de resultados de múltiples fuentes de datos (por ejemplo, secuencias de ADN, perfiles de expresión de genes y proteínas, interacciones proteína-proteína, información estructural de proteínas y datos de ligaduras de proteínas-ADN) podría superar este

inconveniente (Li *et al*., 2006). Sin embargo, hay varios problemas en la integración de diversos datos de genómica en los modelados de redes (Lee *et al*., 2007). En primer lugar, la inferencia a partir de múltiples fuentes de datos diferentes puede interferir en la inferencia de interacciones que sólo están presentes en ciertas condiciones experimentales. Además, los datos de genómica son heterogéneos en su sensibilidad y especificidad para las relaciones entre los genes. Por ejemplo, los métodos experimentales tales como la espectrometría de masas observan preferentemente proteínas abundantes, mientras que los métodos de genómica comparativa sólo se aplican a los genes evolutivamente conservados. El aumento en la sensibilidad de detección por lo general acarrea el costo de aumentar las identificaciones falsas positivas. Por lo tanto, el sesgo sistemático de cada método debe ser entendido y considerado en la integración de datos. Además, los conjuntos de datos de genómica varían mucho en cuanto a su utilidad para la reconstrucción de las redes de genes. Por lo tanto, se requieren métodos de evaluación comparativa robustos capaces de evaluar cada conjunto de datos permitiendo la comparación de sus méritos relativos. Por último, los conjuntos de datos están a menudo correlacionados, lo que complica la integración, ya que la correlación puede ser difícil de medir debido a la falta de completitud de los datos (un problema común) y a los sesgos en los muestreos.

Se han desarrollado en conjunto dos grandes enfoques relacionados en el aprendizaje de regulación transcripcional a partir de múltiples fuentes de datos. En un enfoque se utilizan varios tipos de datos para identificar conjuntos de genes que interactúan conjuntamente en la célula, o que están co-regulados en módulos (Segal *et al*., 2003; Bar-Joseph *et al*., 2003). En el otro enfoque, se utilizan varios tipos de datos para complementar los datos de expresión génica en el aprendizaje de redes regulatorias (Imoto *et al*., 2004; Bernard y Hartemink, 2004). En cuanto a estos últimos trabajos, Bernard y Hartemink (2004) presentaron un método para el aprendizaje de conjuntos de modelos dinámicos de redes regulatorias transcripcionales, a partir de datos de expresión de genes y de datos de ligadura de factores de transcripción, sobre la base de algoritmos de inferencia de redes bayesianas dinámicas. Los resultados obtenidos a partir del análisis de datos del ciclo celular de la levadura, demuestran que las redes regulatorias dinámicas obtenidas a partir de múltiples tipos de datos mediante este algoritmo de aprendizaje conjunto, son más precisas que las que se obtienen a partir de cada tipo de datos por sí solo. En Imoto *et al*. (2004) se propuso en cambio un método estadístico para la estimación de una red de genes basada en redes bayesianas, a partir de datos de expresión de genes conjuntamente con conocimiento biológico, incluyendo interacciones proteína-proteína, interacciones proteína-ADN, información de ligadura de factores de transcripción, literatura existente y otros. Una ventaja del método es que el equilibrio entre la información de *microarrays* y el conocimiento biológico se optimiza automáticamente por el criterio propuesto. Simulaciones de Monte Carlo

mostraron la eficacia del método propuesto, extrayendo más información a partir de datos de *microarrays* y estimando la red de genes con mayor precisión. En Yeang *et al*. (2004) se presentó un marco para inferir la regulación transcripcional. Los modelos desarrollados, llamados modelos físicos de red, son grafos anotados con interacciones moleculares. Los atributos en el modelo corresponden a propiedades verificables del sistema biológico subyacente, como la existencia de interacciones proteína-proteína y proteína-ADN, la direccionalidad de la transducción de señales en las interacciones proteína-proteína, los signos de los efectos inmediatos de estas interacciones, etc. Las posibles configuraciones de estas variables se ven limitadas por las fuentes de datos disponibles. La aplicación de este algoritmo en conjuntos de datos relacionados con los *pathways* de respuesta de la feromona en la levadura, demostró que el modelo derivado fue consistente con el conocimiento previo de dicho *pathway*.

6.2.5. Asociaciones Diferidas en el Tiempo a partir de Datos de Series de Tiempo

Otro aspecto importante a considerar cuando se trata la reconstrucción de RRG, está constituido por la forma en que se capturan los patrones temporales de una RRG. Como se menciona en Silvescu y Honavar (2001) y en Yeang y Jaakkola (2003), la regulación de genes diferida el tiempo es un fenómeno común. De este modo, las regulaciones diferidas múltiples unidades de tiempo pueden considerarse la norma, mientras que las asociaciones con regulación diferida en una sola unidad pueden considerarse como la excepción (Li *et al.*, 2006). Esto ocurre debido a que, dentro de los procedimientos de regulación, se producen varios eventos en diferentes etapas. Por lo general, el paso de la transcripción (del ADN a ARNm) es rápido, mientras que el tiempo para la traducción varía de proteína a proteína (Lewis, 1999). Además, la regulación proteína-ADN es un proceso de acumulación y el umbral es diferente para cada par de genes involucrados en la interacción (Lewis, 1999). Supongamos que g_r regula a g_i, el cambio de nivel de expresión de g_r puede afectar el nivel de expresión de g_i después de un cierto intervalo de tiempo. Por ejemplo, sobre la base de los conjuntos de datos de expresión de genes de la levadura de Spellman *et al*. (1998), el gen *MCM1* regula el gen *CLN3*, y, cada vez que el nivel de expresión de *MCM1* cambia, el nivel de expresión correspondiente de *CLN3* cambia 30 minutos más tarde. Además, los intervalos de retraso de tiempo en la regulación pueden variar para diferentes pares de genes. Por ejemplo, los genes *TNF*-α humano e *iNOS* están regulados por *AP-1* y *NF-κB1*. Sus retrasos en la expresión después de la activación de *AP-1* y *NF-κ B1* son tres y seis horas, respectivamente (Lee *et al.*, 2003). Se sabe, además, que debería haber un límite superior para el tiempo de retardo en una red de genes ya que la longitud de un ciclo celular es limitada. La regulación de los genes puede formar bucles de retroalimentación (por

ejemplo, $g_1 \rightarrow g_2 \rightarrow ... \rightarrow g_1$), que existen en muchos *pathways* metabólicos y son críticos en el mantenimiento de la estabilidad de una red de genes (Cinquin y Demongeot, 2002).

La variabilidad en el tiempo de los procesos biológicos complica aún más la inferencia de asociaciones de genes a partir de datos de series de tiempo. La velocidad a la que se desarrollan procesos similares subyacentes, tales como el ciclo celular, puede esperarse que difiera entre organismos, variantes genéticas, y condiciones ambientales. Por ejemplo, en Spellman *et al.* (1998) analizan datos de series de tiempo de la levadura durante el ciclo celular utilizando diferentes métodos para sincronizar las células. Es claro que las longitudes del ciclo a través de los diferentes experimentos varían considerablemente, y que las series comienzan y terminan en diferentes fases del ciclo celular. Esto complica la interpretación de los resultados si varios conjuntos de datos de series de tiempo se utilizan en la inferencia de las interacciones entre los genes (Gallo *et al*., 2011a). Por lo tanto, es necesario un procedimiento para alinear dichas series con el fin de hacerlas comparables, tales como la representación de perfiles de expresión de series de tiempo como curvas continuas, permitiendo la normalización de la velocidad a la que se considera cada muestra en cada uno de los conjuntos de datos.

6.2.6. Validación Biológica y Estadística de las RRGs Inferidas

Una vez que se obtuvo la RRG mediante un enfoque de minería de datos, es crucial validarla con el fin de determinar la exactitud y/o la viabilidad biológica de las RAs inferidas. El tipo de validación requerido depende del objetivo del estudio que se llevó a cabo. En primer lugar, es necesario separar los análisis en los que el objetivo es la evaluación del algoritmo de minería de datos, de aquellos en los que el objetivo corresponde a la identificación de algunas hipótesis prometedoras de nuevos conocimientos biológicos. En el primer caso, el análisis también depende del tipo de datos empleado en la inferencia. Los datos sintéticos permiten el uso de métricas bien conocidas de minería de datos, tales como precisión, sensibilidad y especificidad, porque las interacciones "verdaderas" entre los genes se conocen de antemano. Por lo tanto, es posible comparar varios algoritmos y determinar cuál de ellos reconstruye mejor una RRG sobre un determinado conjunto de datos sintéticos. Aunque los resultados no pueden ser concluyentes, ya que dependen del enfoque utilizado en la generación de los datos sintéticos, proporcionan información general sobre el comportamiento de cada método.

Cuando se emplean datos reales para evaluar un algoritmo, las interacciones reales entre genes que están presentes en los datos no se conocen de antemano. En general, sólo se dispone de conocimiento curado específico sobre las interacciones reales entre genes. Aún más, esas interacciones conocidas podrían no estar presentes en el conjunto de datos reales debido a condiciones experimentales específicas del experimento. Por lo tanto, las métricas mencionadas

anteriormente, en general, no son aplicables. Sin embargo, existen otros medios para evaluar el algoritmo cuando se utilizan datos reales. En primer lugar, existe el análisis regla por regla de la relevancia biológica de las relaciones obtenidas por el método. Esto se hace por medio de una búsqueda a través de la literatura, mirando la información biológica conocida de los genes bajo consideración y conjeturando sobre su posible relevancia biológica. Este enfoque es sano cuando se evalúa un único método, sin embargo, tiene inconvenientes que complican su aplicación en la mayoría de los escenarios. En primer lugar, sólo es aplicable cuando se evalúa un pequeño conjunto de reglas, ya que todo el proceso se lleva a cabo manualmente. Otra desventaja es que no puede ser utilizado para la comparación de varios métodos, debido a que la calidad de una regla está sesgada por el experto que la evalúa, y por lo tanto es imposible establecer un orden razonable de mérito para los algoritmos bajo consideración. Otro enfoque consiste en el empleo de bases de datos en línea de interacciones de genes para evaluar las RAs inferidas. Como ejemplo, en el caso de la levadura, hay dos bases de datos bien conocidas: *KEGG* (Kanehisa *et al*., 2006) y *Gene Ontology* (GO) (Dwight *et al*., 2002). KEGG es una colección de mapas de *pathways* extraída manualmente que representan el conocimiento en la regulación de la interacción molecular, y los *pathways* de interacción contienen información que es relevante para el ciclo celular de la levadura. Por lo tanto, si una RA extraída se corresponde con la información de regulación en KEGG, entonces la regla puede ser considerada como correcta. De la misma manera, la anotación GO es otra fuente de asociaciones potenciales para los genes de la levadura. Se pueden considerar los pares de genes que representan a todos los pares de genes que comparten términos de procesos biológicos entre niveles específicos de una anotación ontológica de genes, y utilizar ese conjunto como referencia en la evaluación. No obstante, como se dijo antes, debe quedar claro que si los conjuntos de datos no tienen correlaciones entre los genes implicados en interacciones conocidas importantes, ningún enfoque dirigido por datos podrá inferir tales relaciones.

Independientemente del tipo de datos empleado en el análisis, otra técnica común para evaluar el rendimiento de un algoritmo de minería de RAs es la validación cruzada (Geisser, 1993; Kohavi, 1995; Devijver y Kittler, 1982). La validación cruzada es una técnica ampliamente utilizada en minería de datos para la evaluación de cómo los resultados de un análisis estadístico se generalizan a un conjunto de datos independientes. Se utiliza principalmente en la reconstrucción RRG para estimar cuan preciso será el modelo predictivo en la práctica. Una ronda de validación cruzada consiste en dividir al conjunto de datos en subconjuntos complementarios, realizando el análisis en un subconjunto (llamado conjunto de entrenamiento), y validando el análisis con el otro subconjunto (llamado conjunto de validación o conjunto de pruebas). Para reducir la variabilidad, se realizan múltiples rondas de validación

cruzada utilizando diferentes particiones, y los resultados de la validación se promedian sobre las rondas. Hay algunos tipos comunes de validación cruzada, como la validación cruzada *K-fold* y la validación de sub-muestreo repetido aleatorio. La mayor desventaja de este tipo de validación en la minería de RAs entre genes es, en general, la cantidad reducida de muestras disponibles en los conjuntos de datos. Si una inferencia se lleva a cabo en un conjunto de datos con pocas muestras, la cantidad efectiva de muestras utilizadas en el proceso de inferencia es aún menor debido a la partición en conjuntos de entrenamiento y prueba, y por lo tanto, afecta negativamente a la predictibilidad del modelo resultante. Otra desventaja es cuando se emplean datos de series temporales para inferir reglas diferidas en el tiempo. La partición en conjuntos de entrenamiento y prueba no se puede realizar debido a que ambos conjuntos se referirán a períodos completamente diferentes de tiempo, convirtiéndose así en incomparables desde el punto de vista biológico.

Por último, si el objetivo global del análisis es la inferencia de nuevos conocimientos biológicos, es necesario aclarar que las reglas inferidas por cualquier método de minería de datos siempre representarán asociaciones regulatorias confidentes entre genes. Es decir, el método de extracción de reglas puede ser útil para la identificación de algunas hipótesis prometedoras en cuanto a la naturaleza de los experimentos analizados. Sin embargo, la corroboración por medio de experimentos biológicos será siempre obligatoria a fin de obtener nuevos conocimientos curados.

6.2.7. Ventajas y Limitaciones de la Inferencia de RRGs basada en RAs

Inferir RRGs con enfoques de minería de RAs presenta varias ventajas. En primer lugar, los modelos inferidos son muy abstractos y por lo tanto, requieren menos cantidad de datos que los modelos continuos (tales como las ecuaciones diferenciales ordinarias generales). Esto favorece su capacidad de realizar inferencias, ya que casi todos los datos de expresión de genes son adecuados para la extracción de RAs. Además, la simplicidad del modelo inferido permite la inferencia de modelos más grandes con una mayor velocidad de análisis, y también facilita la interpretación de los resultados.

Sin embargo, también hay varias desventajas en la inferencia de RAs. La más importante es que las RAs pueden mostrar sólo comportamiento dinámico cualitativo. Esto podría ser superado hasta cierto punto si se consideran varios estados para cada gen (más de dos). No obstante, esto también complica la inferencia, ya que requiere más datos para deducir las interacciones, y también exige más recursos computacionales debido al aumento del espacio de búsqueda. Además, dado que los modelos obtenidos son muy abstractos, el nivel de detalle que

puede ser modelado es muy limitado. Este problema afecta generalmente a su fidelidad en lo que respecta a la realidad biológica, y también limita su capacidad de modelado dinámico.

6.3. Técnicas para la Inferencia de RRGs Basadas en RA

Comencemos con los métodos basados en conjuntos de elementos frecuentes.

6.3.1. Métodos Basados en Conjuntos de Elementos Frecuentes

Los métodos basados en *conjuntos de elementos frecuentes* (*frequent itemset*) fueron originalmente desarrollados para encontrar asociaciones o relaciones de correlación interesantes entre datos de grandes bases de datos, como los registros de transacciones de negocios. El descubrimiento de RAs interesantes derivadas de los llamados *conjuntos de elementos frecuentes* es valioso en muchos procesos de toma de decisión de negocios, tales como diseño de catálogos, marketing cruzado, y análisis de pérdida-líder (Han y Kamber, 2000). Siguiendo las definiciones de Zhang (2006), sea $\Gamma = \{g_1, ..., g_n\}$ un conjunto de literales distintos, denominados elementos. Un conjunto $X \subseteq \Gamma$ con $|X| = k$ se llama *k-conjunto de elementos* o simplemente *conjunto de elementos*. Sea D un conjunto de *transacciones*, donde cada *transacción* T es un conjunto de elementos. Hay un único identificador asociado con cada transacción, su *identificación de transacción* (*TID*). Una transacción T contiene o soporta a un conjunto de elementos X, si $X \subseteq T$. Como se dijo anteriormente, una RA es una regla $X \rightarrow Y$, donde en este caso $X \subseteq \Gamma$, $Y \subseteq \Gamma$ y $X \cap Y = \varnothing$. El conjunto de elementos X tiene un *soporte o* en la base de datos de transacciones D si el $o\%$ de las transacciones T en D contiene X:

$$\text{supp}(X) = \frac{|\{T \mid X \subseteq T, T \in D\}|}{|D|}. \tag{6.10}$$

La regla $X \rightarrow Y$ tiene un soporte s en la base de datos de transacciones D si el $s\%$ de las transacciones T en D contiene $X \cup Y$, es decir:

$$\text{supp}(X \rightarrow Y) = \frac{|\{T \mid \{X \cup Y\} \subseteq T, T \in D\}|}{|D|}. \tag{6.11}$$

La regla $X \rightarrow Y$ tiene una confidencia c en la base de datos de transacciones D si el $c\%$ de las transacciones T en D que contienen a X también contienen a Y, es decir:

$$\text{conf}(X \rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}. \quad (6.12)$$

Note la diferencia en el significado de las métricas de soporte y confidencia. Mientras que el soporte de un conjunto de elementos o una regla indica la significación estadística del conjunto de elementos o de la regla, la confidencia es una medida de la fuerza de las reglas. Por lo general, sólo se consideran las reglas con valores de soporte y confidencia por encima de ciertos umbrales (*minsupport* y *minconf* respectivamente).

En esta formulación, el problema de la minería de reglas se puede descomponer en dos pasos: identificación de conjuntos de elementos frecuentes y generación de reglas. El primer paso implica la identificación de todos los conjuntos de elementos frecuentes $F = \{X \mid \text{supp}(X) \geq minsupp\}$. Una vez que se conoce el conjunto de todos los conjuntos de elementos frecuentes, así como su soporte, el segundo paso consiste en la derivación de la RAs a partir de F. Este procedimiento es muy simple: para cada $X \in F$, se comprueba la confidencia de todas las reglas posibles $X\text{-}Y \rightarrow Y$, donde $Y \subset X$ e $Y \neq \varnothing$, excluyendo a aquellas reglas cuya confidencia este por debajo de *minconf*.

El principal reto en minar RAs de esta manera se encuentra en el primer paso, la identificación de conjuntos de elementos frecuentes. Es intuitivamente obvio que un aumento lineal en $|I|$ resultará en un crecimiento exponencial del número de conjuntos de elementos que deben ser considerados. Afortunadamente, el soporte de conjuntos de elementos tiene la *propiedad de clausura descendiente*: todos los subconjuntos de un conjunto de elementos frecuentes deben ser frecuentes (Zhang, 2006). Como resultado, hay una frontera en la estructura reticular que separa los conjuntos de elementos frecuentes de los no frecuentes (Hipp, *et al.*, 2000), con los conjuntos de elementos frecuentes situados por encima de la frontera y los conjuntos de elementos no frecuentes situados por debajo. El principio básico radica en el empleo de esta frontera para podar el espacio de búsqueda eficientemente.

La mayoría de los métodos de extracción de conjuntos de elementos propuestos son una variante del algoritmo APRIORI (Agrawal *et al.*, 1993). El algoritmo APRIORI adopta un enfoque de búsqueda a lo ancho en la retícula de conjuntos de elementos, utilizando *k*-conjuntos de elementos para explorar (*k+1*)-conjuntos de elementos. El algoritmo analiza la base de datos en una primera iteración para contar las ocurrencias de cada elemento. A continuación, encuentra el *1*-conjunto de elementos frecuentes (denotado como L_1) con respecto a un umbral *minsupp* dado. Una iteración subsiguiente del algoritmo (por ejemplo, la iteración *k*) consta de dos fases. En primer lugar, el (k-1)-conjunto de elementos frecuentes L_{k-1} encontrado en la iteración (k-1)-ésima es usado para generar el conjunto de elementos candidatos C_k. En segundo

lugar, la base de datos es escaneada y los *k*-conjuntos de elementos de C_k son analizados. Si un *k*-conjunto de elementos X de C_K no es frecuente, este es eliminado de C_k. Los restantes *k*-conjuntos de elementos en C_k constituyen L_k y se utilizarán para la $(k+1)$-ésima iteración. Estas dos fases iteran hasta que el conjunto de *k*-conjuntos de elementos frecuentes L_k es vacío.

Los métodos basados en APRIORI muestran un buen rendimiento con conjuntos de datos dispersos, como los datos de "canasta de mercado", donde los patrones frecuentes son muy cortos. Sin embargo, con conjuntos de datos densos tales como los *microarrays*, donde hay muchos patrones frecuentes largos, estos métodos escalan pobremente y son a veces poco prácticos. Este inconveniente se debe al alto costo computacional requerido para la evaluación de los conjuntos de candidatos y de prueba utilizados por los enfoques basados en APRIORI. En este sentido, nuevos métodos como el FP-GROWTH (Han *et al.*, 2000) han surgido como una estrategia prometedora, simplificando el problema de encontrar patrones largos mediante la concatenación de patrones pequeños. De hecho, varios métodos han sido desarrollados sobre la base del FP-GROWTH (Ceglar y Roddick, 2006; Han *et al.*, 2007). La idea principal se basa en una estructura de árbol compacto denominado árbol FP, que es buscado en forma recursiva con el fin de enumerar todos los patrones frecuentes. El crecimiento de los patrones se logra mediante la concatenación de un patrón sufijo con el patrón frecuente generado a partir de un árbol condicional FP (por ejemplo, los patrones con longitud igual a 1 se utilizarán para la generación de los que tienen longitud igual a 2, y así sucesivamente). Incluso los métodos basados en árboles como el FP-GROWTH pueden encontrar algunas dificultades cuando se emplean con conjuntos de datos de gran dimensionalidad. Un patrón frecuente de tamaño *k* (número de elementos) implica la presencia de 2^k-2 patrones frecuentes adicionales, cada uno de los cuales se comprueba de forma explícita por tales métodos. Por lo tanto, los algoritmos que emplean heurísticas sofisticadas para minería de conjuntos de elementos frecuentes largos constituyen soluciones prácticas para el análisis de asociaciones de genes.

Actualmente existen dos alternativas para la minería de patrones largos. La primera es extraer solo conjuntos de elementos frecuentes maximales, como en MAXMINER (Bayardo, 1998) y GENMAX (Gouda y Zaki., 2005), que son típicamente órdenes de magnitud menores que todos los patrones frecuentes. Los conjuntos de elementos maximales son aquellos patrones frecuentes largos encontrados bajo cierto umbral de soporte. A pesar del hecho de que los patrones maximales ayudan a entender los conjuntos de elementos largos en dominios densos, es importante también tener en cuenta que conducen a la pérdida de información; debido a que el conteo de subconjuntos no está disponible, los conjuntos máximales no son adecuados para la generación de reglas. La segunda alternativa es la de minar solo conjuntos cerrados frecuentes como en CLOSE (Pasquier *et al.*, 1999), CLOSE + (Wang *et al.*, 2003) y CHARM (Zaki y

Hsiao, 2002). Los conjuntos cerrados son sin pérdida en el sentido de que pueden ser usados para determinar unívocamente el conjunto de todos los patrones frecuentes y sus frecuencias exactas. Un conjunto de elementos cerrado es un patrón frecuente que se ajusta a un umbral de soporte y no tiene ningún súper conjunto de patrones frecuentes con valor de soporte similar que lo contenga. Por otra parte, los algoritmos basados en CLOSE pueden manejar la redundancia de patrones, que es bastante común en la minería de asociaciones en bases de datos de alta dimensionalidad (Ceglar y Roddick, 2006; Han *et al.*, 2007). Sin embargo, incluso con esta estrategia, la alta dimensionalidad de *microarrays* sigue planteando grandes desafíos para estos métodos.

Es importante tener en cuenta que todos los métodos mencionados anteriormente emplean una combinación exponencial de todas las filas (es decir, genes) en la matriz de expresión de genes. Tal tamaño de espacio de búsqueda aumenta proporcionalmente con el número de genes. Por lo tanto, los métodos de minería de patrones frecuentes que no utilizan la generación de conjuntos de candidatos son generalmente más eficientes. El tipo de patrones que se encuentran también juega un papel importante en la fortaleza o debilidad de un método de minería de patrones frecuentes. Así, las estrategias de conjunto de elementos cerrados son más confiables para la minera de RAs de genes. De tal discusión general, se podría esperar que CLOSE+ sea el enfoque más adecuado para la minería de RA de genes. Sin embargo, el método no se ha aplicado a ningún tipo de dato de expresión de genes, aunque si se evaluó con éxito frente a su contraparte mediante el uso de otros conjuntos de datos de alta densidad (Alves *et al.*, 2009).

6.3.1.1. RA Diferidas en el Tiempo con Minería de Conjunto de Elementos Frecuentes

Actualmente existen dos alternativas para la minería de RAs diferidas en el tiempo con métodos basados en conjuntos de elementos frecuentes. La primera es extraer las reglas por medio de la aplicación del algoritmo APRIORI (o cualquier otro algoritmo de minería de conjuntos de elementos) en matrices de expresión de perfiles diferidos en el tiempo (o TdE por sus siglas en inglés) (Baralis *et al.*, 2008), similares a las utilizadas en Li *et al.* (2006). La TdE captura la regulación entre los genes en **W** unidades de tiempo. Consiste simplemente en una matriz de $n \times (\mathbf{W} - 1)$, en la que cada fila es una ventana de tiempo y las columnas contienen los **W** valores correspondientes para cada gen. Por ejemplo, tal como se reportó en Baralis *et al.* (2008), si se considera la matriz discreta de la tabla 6.1, se obtiene la matriz diferida en el tiempo de la tabla 6.2 en el caso de las regulaciones entre 2 instantes de tiempo (es decir, $\mathbf{W} = 2$)

.

Tabla 6.1. Una matriz discreta de datos de expresión.

Discrete Matrix						
gene	t_1	t_2	t_3	t_4	...	t_M
g_1	0	1	0	0	…	1
g_2	1	1	0	0	…	0
g_3	0	0	1	1	…	0
g_4	1	0	1	0	…	1
...	…	…	…	…	…	…
g_N	0	0	1	0	…	0

Tabla 6.2. Una matriz de expresión de perfiles diferidos en el tiempo.

Time-Delay Matrix								
	g_1+0	g_1+1	g_1+2	g_2+0	...	g_N+0	g_N+1	g_N+2
W_1	0	1	0	1	…	0	0	1
W_2	1	0	0	1	…	0	1	0
…	…	…	…	…	…	…	…	…
W_M	1	NA	NA	0	…	0	NA	NA

La otra alternativa es extender los conceptos de minería de conjuntos de elementos con el fin de tener en cuenta las reglas diferidas en el tiempo. En este sentido, Nam *et al*. (2009) desarrolló un método de minería de reglas de asociación temporales (TARM), basado en el algoritmo APRIORI, extendiendo los conceptos básicos de la siguiente manera: un elemento temporal es un elemento que tiene una estampilla de tiempo. Sea $\ddot{I}$ un conjunto no vacío de elementos temporales. Dado un $\ddot{I}$, un conjunto T de transacciones sobre $\ddot{I}$, y un número entero positivo *minsupport*, $\ddot{I}$ es un conjunto de elementos temporales frecuentes con respecto a T y *minsupport* si *supp* $(\ddot{I}) \geq$ *minsupport*. Un RA temporal ($X(w) \rightarrow Y$) es un par de conjuntos de elementos temporales disjuntos donde la estampilla de tiempo de cada elemento temporal en X está adelantada a las de todos los elementos temporales en Y, y donde w es el intervalo de dos estampillas de tiempo diferentes.

La figura 6.3 muestra un ejemplo, reportado en Nam *et al.* (2009), del proceso de minería de conjuntos de elementos temporales. Supongamos un proceso de discretización de tres estados como se muestra en la figura 6.3(a). Con el fin de encontrar los genes asociados temporalmente, se supone en primer lugar que todos los genes relacionados pueden tener diversos tamaños de retraso de tiempo transcripcional. Por lo tanto, el método busca los genes asociados en todos los posibles conjuntos de diferentes puntos de tiempo en los que el intervalo temporal varía de 0 a **W** (figura 6.3 (b)). Por ejemplo, el conjunto de transacciones temporal $t_0 + t_2 = [+g_{1L}, -g_{2L}, +g_{1R}, +g_{2R}, -g_{3R}]$ consta de genes activados o inhibidos en las estampillas de tiempo t_0 y t_2, con el tamaño de retardo de tiempo transcripcional $w = 2$. Es importante tener en cuenta que g_1 es activado tanto en t_0 como en t_2, pero se considera como dos genes diferentes: g_{1L} (g_1 en lado izquierdo) y g_{1R} (g_1 en el lado derecho). A continuación, la figura 6.3(c) indica los conjuntos de elementos temporales frecuentes extraídos con un umbral de soporte del 50%. Y, por último, se

encuentran dos RAs temporales con umbral de confianza del 50%, como se muestra en la figura 6.3(d). De esta manera, TARM puede encontrar varios tamaños de retardo de tiempo transcripcional entre los genes asociados, relaciones de activación e inhibición, y conjuntos de co-reguladores para los genes regulados.

	t_0	t_1	t_2	t_3	t_4	t_5	t_6
g_1	1	0	1	0	1	0	0
g_2	-1	-1	1	-1	1	0	1
g_3	0	0	-1	-1	0	-1	0

(a)

$t_0+t_2 = \{+g_{1L}, -g_{2L}, +g_{1R}, +g_{2R}, -g_{3R}\}$

$t_1+t_3 = \{-g_{2L}, -g_{2R}, -g_{3R}\}$

$t_2+t_4 = \{+g_{1L}, +g_{2L}, -g_{3L}, +g_{1R}, +g_{2R}\}$

$t_3+t_5 = \{-g_{2L}, -g_{3L}, -g_{3R}\}$

$t_4+t_6 = \{+g_{1L}, +g_{2L}, +g_{2R}\}$

(b)

$\{+g_{1L}\}$, $\{-g_{2L}\}$, $\{+g_{2R}\}$, $\{-g_{3R}\}$

$\{+g_{1L}, +g_{2R}\}$, $\{-g_{2L}, -g_{3R}\}$

(c)

$+g_1 \xrightarrow{2} +g_2$

$-g_2 \xrightarrow{2} -g_3$

(d)

Figura 6.3. Una ilustración del proceso de minería de RA temporales con retraso transcripcional $w = 2$, soporte ≥ 50% y confidencia ≥ 50%. Fig. 6.3a: Datos de series de tiempo discretizados, con 3 genes y 6 puntos de tiempo. Fig. 6.3b: Conjuntos de transacciones temporales, con retraso de tiempo transcripcional $w = 2$. Fig. 6.3c: Conjuntos de elementos temporales frecuentes, con soporte = 50%. Fig. 6.3d: RA temporales, con confidencia = 50%.

6.3.2. Enfoques Basados en Árboles de Clasificación y Regresión

Un árbol de decisión es una herramienta de soporte que utiliza un grafo en forma de árbol o modelo de decisiones y sus posibles consecuencias, incluyendo resultados de azar, costos de recursos y utilidad. Los árboles de decisión son de uso común en la investigación de operaciones, específicamente en el análisis de decisiones, para ayudar a identificar la estrategia

con mayores probabilidades de alcanzar una cierta meta. En la minería de RAs, un árbol de decisión es un árbol con raíz en el que los nodos que no son hojas se etiquetan con genes explicadores, los arcos que salen de los nodos que no son hojas se marcan con posibles características de los genes explicadores, y las hojas del árbol se etiquetan con los estados de los genes predichos. Hay dos tipos de árboles de decisión: árboles de clasificación y árboles de regresión (Breiman *et al*., 1984). Los primeros son aquellos cuyos resultados son las clases a las que pertenecen los datos, mientras que los segundos son aquellos cuyos resultados pueden ser considerados como números reales. Un ejemplo de un árbol de decisión para la clasificación del gen de levadura *CLN2* se muestra en la figura 6.4. Cada camino desde el nodo raíz a un nodo hoja en el árbol representa una regla que define un estado en la predicción del gen, a través de los niveles de expresión de los genes explicadores. De ello se desprende que cada árbol de decisión es equivalente a una lista de reglas de decisión. Este método de representación permite la descomposición de los árboles de decisión, de estructura compleja, a una representación simple y compacta de RAs, que se puede comparar de forma independiente con el conocimiento existente. De esta manera, el árbol de decisión de la figura 6.4 se puede representar por medio de la siguiente lista de RA:

- *-SWI5^+CDC28^+CLN1 → -CLN2*
- *-SWI5^+CDC28^-CLN1→+CLN2*
- *-SWI5^-CDC28 → +CLN2*
- *+SWI5^-CLB1 → +CLN2*
- *+SWI5^+CLB1 → -CLN2*

donde el símbolo ^ representa el *AND* lógico.

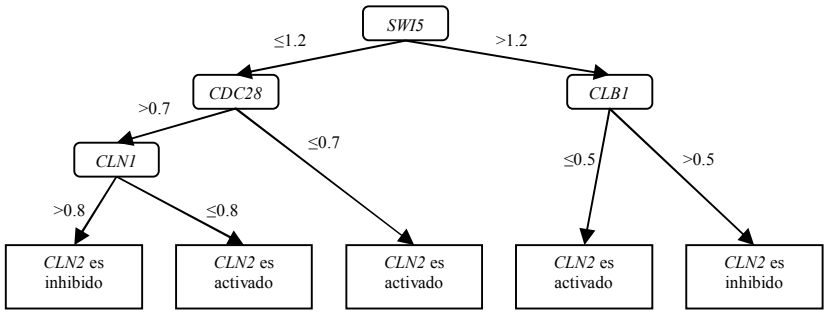


Figura 6.4. Un posible árbol de clasificación para el gen *CLN2* de *S. cerevisiae. CLN2* es el gen blanco; *SWI5, CLB1, CDC28* y *CLN1* son los genes reguladores. Los umbrales de expresión de los respectivos genes explicadores están marcados los arcos.

En el contexto de la minería de RAs por medio de árboles de decisión, las funciones que determinan los estados de los genes blanco a partir de datos se llaman clasificadores, mientras que los algoritmos de construcción de tales clasificadores en base a datos con estados conocidos son llamados algoritmos inductores, ensambladores o de inducción. Cada perfil de expresión con un estado conocido de un gen predicho se llama ejemplo o instancia. El conjunto de ejemplos que se utilizan para la creación del clasificador es el conjunto de entrenamiento. Si un subconjunto de los ejemplos se separa del conjunto de entrenamiento, y se lo utiliza para la estimación de la precisión de clasificación, se lo denomina conjunto de prueba. De este modo, la inferencia de un árbol de decisión para la minería de RAs puede ser considerada como un problema de clasificación estándar de la siguiente manera. Sea $Y = \{y_1, ..., y_n\}$ el conjunto de todos los perfiles de expresión de la muestra, y sea $Y_{/i} = \{y_{1/i}, ..., y_{n/i}\}$ el conjunto de los perfiles de expresión de las muestras parciales para un gen g_i determinado de la matriz $\boldsymbol{X}'$. Vamos a definir un clasificador C, como una función que mapea un vector y a un valor discreto s. A veces, en el contexto de la clasificación, el vector y es llamado el vector característica, mientras que s es una etiqueta. El subconjunto de vectores y con las etiquetas correctamente asignadas se denomina conjunto de datos, D, para un problema de clasificación particular. Un algoritmo de inducción mapea un conjunto de datos D en un clasificador C. Por lo tanto, a fin de resolver el problema descrito anteriormente, es necesario definir los conjuntos de datos y, a continuación, elegir algoritmos de inducción apropiados. Más específicamente, tomemos como objetivo predecir el estado del gen g_i a partir de una matriz $\boldsymbol{X}'$. Un algoritmo de inducción I mapea el conjunto de datos $D_i = (Y_{/i}, s_i)$ en el clasificador C_i (el índice i de D y C se utiliza para enfatizar que corresponden al gen g_i.). Para el conjunto de datos D_i dado, es necesario crear un clasificador que prediga correctamente el estado del gen g_i, esto es, $I(D_i, y_{j/i}) = C_i(y_{j/i}) = s_{ij}$. Por lo tanto, para este problema, el gen g_i predicho y los genes explicadores pertenecen a la misma muestra j.

Como parte del problema de clasificación, es necesario encontrar los genes que son relevantes para la predicción de un gen particular. Esto se conoce como el problema de selección de características. En general, se han presentado en la literatura dos tipos de métodos para la selección de características: métodos de filtrado y de envoltura (Kohavi, 1995; Witten y Frank, 1999). En el enfoque de filtrado, el conjunto de características se filtra para encontrar el subconjunto "más prometedor" mediante la evaluación de una función objetivo antes de ejecutar el algoritmo de inducción. El punto débil de este enfoque es que no se tienen en cuenta las propiedades del algoritmo de inducción en particular. En el enfoque de envoltura, el algoritmo de selección utiliza el algoritmo de inducción en sí para evaluar la función objetivo. El enfoque de envoltura de Kohavi (1995) tuvo mejor rendimiento que el método de filtrado para muchos

conjuntos de datos reales y artificiales. La idea del algoritmo de envoltura es ajustar los parámetros de un algoritmo de inducción considerándolo como una caja negra, a fin de optimizar alguna función objetivo (por ejemplo, la exactitud de un clasificador). El conjunto de atributos relevantes para la clasificación se pueden considerar como parámetros de un algoritmo de inducción. La elección de los parámetros que maximizan la función objetivo da una lista de características "buenas". Los detalles del algoritmo de selección de características pueden ser consultados en Kohavi (1995). Las reglas de clasificación inferidas de esta manera asumen que para realizar predicciones exactas son suficientes un número limitado de genes reguladores.

Soinov *et al.* (2003) fueron los primeros autores que se acercaron a la tarea de inferir las RAs por medio de árboles de decisión. Ellos emplearon algoritmos de clasificación para datos continuos, en los que la discretización forma parte del algoritmo. Esto les permitió encontrar umbrales de abundancia de los genes reguladores, que son específicos para diferentes interacciones entre los genes de la red, y suficientes para la conmutación del gen blanco de un estado a otro. De esta manera, cada gen tiene su propio umbral de discretización, único para las señales de entrada. Se utilizaron dos tipos de algoritmos de inducción. El primero explota el enfoque de envoltura para la selección de características (Kohavi, 1995). Se llama C4.5, por Quinlan (1992), con envolturas de Kohavi (1995). El segundo es el algoritmo C4.5 en sí. El C4.5 es un algoritmo que construye el modelo de clasificación inductivamente, generalizando la información a partir de ejemplos clasificados correctamente. Este ha demostrado ser un algoritmo de buen rendimiento para una gran variedad de conjuntos de datos.

Un ejemplo publicado más recientemente es el trabajo de Huynh-Thu *et al*. (2010) en el que se emplean árboles de regresión para resolver el problema. La idea básica de este método es descomponer la predicción de una red de regulación entre p genes en p problemas de regresión diferentes. En cada uno de los problemas de regresión, el patrón de expresión de uno de los genes regulados se predice a partir de los patrones de expresión de todos los otros genes. La diferencia entre este método y el enfoque de Soinov *et al.* (2003) radica en el uso de árboles de regresión en lugar de árboles de clasificación. De esta forma, se comparan dos métodos de ensamblado basados en árboles fundamentados en la aleatoriedad, llamados *Random Forests* (Breiman, 2001) y *Extra-Trees* (Geurts *et al*., 2006). En un ensamblado *Random Forest*, cada árbol se construye en una retroalimentación de muestras a partir del ejemplo de aprendizaje original y, en cada nodo de prueba, se seleccionan K atributos al azar entre todos los atributos candidatos antes de determinar la mejor división. Por otro lado, en el método *Extra-Trees*, cada árbol se construye a partir del ejemplo de aprendizaje original y en cada nodo de prueba, la mejor división se determina entre K divisiones aleatorias, y cada una es determinada por una selección aleatoria de una entrada (sin reemplazo) y un umbral.

6.3.2.1. RAs Diferidas en el Tiempo con Árboles de Decisión

El marco de trabajo mencionado anteriormente para la inferencia de RAs mediante árboles de decisión no tiene en cuenta las posibles interacciones diferidas en el tiempo. En Soinov *et al.* (2003) se introdujo una definición ampliada para el problema de interacciones diferidas una única unidad de tiempo. Esta formulación es muy similar a la anterior, excepto que el conjunto de datos es ahora $D_i = (Y'_{/i}, s'_i)$, donde $Y'_{/i} = \{y_{1/i},\ldots,y_{n-1/i}\}$ y $s'_i = (s_{i_2}, \ldots, s_{i_n})$. El clasificador C_i se dice que clasifica correctamente al gen g_i para la muestra j, si $C_i(y_{j/i}) = s_{i(j+1)}$. Tenga en cuenta que, en el caso de este problema, los genes reguladores pertenecen a la muestra que precede a la muestra del gen blanco g_i. Esta formulación puede ser generalizada a cualquier retardo de tiempo del efecto entre los genes reguladores y el gen blanco.

Otro enfoque fue propuesto por Li *et al.* (2006). En este trabajo se introduce un método que permite que la expresión de un gen blanco en el tiempo t+1 interactúe con otros genes en tiempos {t, t-1, ..., t-(**W**-1)}. Para cada gen blanco, se construye su perfil de expresión diferido en el tiempo. Luego, se utiliza un árbol de decisión para descubrir las regulaciones diferidas en el tiempo que modulan las actividades del gen regulado (véase la figura 6.5 para un ejemplo).

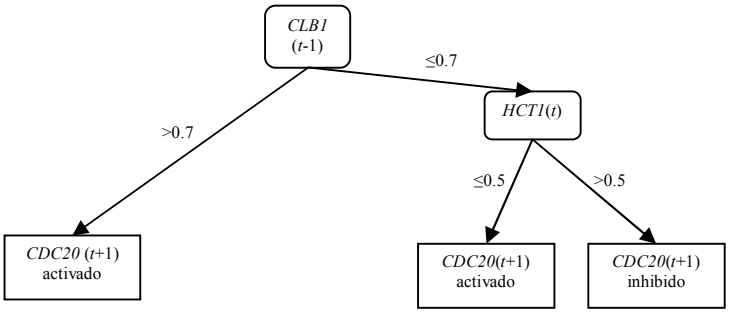


Figura 6.5. Un posible árbol de clasificación para el gen *CDC20* de *S. cerevisiae. CDC20* es el gen regulado. *CLB1* y *HTC1* son los genes reguladores. Los umbrales de expresión de los respectivos genes explicadores están marcados en los arcos.

Un perfil de expresión de gen diferido en el tiempo (*time delayed gene expression profile* o TdE) es una matriz de (m - **W**) × (n × **W**), donde cada bloque de **W**-columnas en las n×**W** columnas representa las actividades de cada uno de los n genes reguladores en los puntos de tiempo t, t-1, ..., t-(**W**-1), por lo que cada fila es un vector de dimensión (n × **W**). A medida que el valor de t cambia de **W** a m-1 (la ventana de tiempo se mueve desde el primer punto de tiempo hasta el punto de tiempo m-**W**), se producen m-**W** vectores o muestras. A continuación, es necesario establecer el fenotipo correspondiente (etiqueta) para cada muestra, que fue determinado por los estados del gen blanco g_i. Por último, los datos completos de los perfiles de expresión de genes diferidos en el tiempo para el gen blanco se denotan por D_i = (TdE,C_i),

donde C_i es un vector columna de los estados para el gen g_i. La matriz D_i = (TdE,C_i) para el gen objetivo g_i se da en la tabla 6.3.

Tabla 6.3. Matriz D_i=(TdE,C_i) para el gen blanco g_i. Los genes g_1, .., g_n son los posibles genes reguladores a ser evaluados. Los valores d_{kl} son las transcripciones temporales para esos genes. C_i denota el vector fenotipo (estado) para el gen blanco g_i en el punto de tiempo (t+1, ..., m).

Gen\t+1	g_1				...	g_n				C_i
	t+(**W**-1)	...	t-1	t		t+(**W**-1)	...	t-1	t	
W+1	d_{11}	...	$d_{1(\mathbf{W}-1)}$	$d_{1\mathbf{W}}$	...	d_{n1}	...	$d_{n(\mathbf{W}-1)}$	$d_{n\mathbf{W}}$	$C_{i(\mathbf{W}+1)}$
W+2	d_{12}	...	$d_{1\mathbf{W}}$	$d_{1(t+1)}$	...	d_{n2}	...	$d_{n\mathbf{W}}$	$d_{n(t+1)}$	$C_{i(\mathbf{W}+2)}$
...	...	...	...	...	...	...	...	...	...	...
W+(m-**W**)	$d_{1(m-\mathbf{W})}$	...	$d_{1(m-2)}$	$d_{1(m-1)}$	...	$d_{n(m-\mathbf{W})}$	...	$d_{n(m-2)}$	$d_{n(m-1)}$	C_{im}

6.3.3 Redes Bayesianas

Una red bayesiana es una representación de una distribución de probabilidad conjunta como un grafo acíclico dirigido (GAD) (Imoto *et al*., 2002; Friedman *et al.*, 2000). Los vértices de un GAD corresponden a variables aleatorias [V_1, ..., V_n] y los arcos o aristas corresponden a dependencias padre-hijo entre las variables. Las variables aleatorias pueden ser de valores discretos o continuos. En el contexto de las RRGs, V_i representa el nivel de expresión del gen g_i, y los arcos del GAD representan las relaciones entre los genes. De este modo, una red bayesiana puede ser representada por una lista de RAs que corresponden a las dependencias padre-hijo entre las variables. La distribución de probabilidad conjunta puede ser escrita en forma de producto simple:

$$P[V_1,...,V_n] = \prod_{i-1}^{n} P[V_i \,|\, P_a(V_i)]. \tag{6.13}$$

Las redes bayesianas tienen una serie de características que las convierten en candidatas atractivas para el modelado de los datos de expresión de genes: son adecuadas para manejar datos con ruido o con datos faltantes, para manejar variables ocultas tales como los niveles de proteína que pueden tener un efecto sobre los niveles de estado estacionario de ARNm, para describir procesos que interactúan localmente y para hacer inferencias causales a partir de los modelos derivados. Friedman *et al*. (2000) propuso modelar una red de genes como una red bayesiana: cada gen es un vértice y cada relación de regulación es un arco en la red bayesiana. Como es técnicamente difícil aprender una red dispersa, en Friedman *et al*. (2000) se propuso un algoritmo de dos pasos llamado algoritmo de candidato disperso, para aprender la estructura y los parámetros: para cada gen, (1) se seleccionan algunos padres candidatos que tienen alguna probabilidad de ser los padres del gen blanco; (2) se calcula la puntuación bayesiana para cada posible subconjunto del conjunto candidato de padres, y se busca la mejor combinación. En el primer paso se aplica un método general usando correlación por pares, como *Mutual*

Information (MI), para encontrar los genes con alta dependencia con los genes blanco. Sin embargo, algunas dependencias no se pueden medir por MI. Por lo tanto, se generan unos padres "débiles". Los padres débiles son padres de un gen blanco, pero no tienen una alta dependencia con el mismo. También se emplea la divergencia *Kullback-Leibler* (KL) en el trabajo, mejorando iterativamente las dependencias entre pares de genes, utilizando la red aprendida como conocimiento previo en el proceso de aprendizaje iterativo. El segundo paso se puede hacer mediante algún método heurístico tal como *hill climbing* (Friedman *et al*., 2000). Friedman demostró que los resultados obtenidos por el algoritmo propuesto son biológicamente significativos analizándolos con un conjunto de métricas estadísticas: prueba de robustez, relación de orden y relación de Markov entre otras. Desde entonces, se han propuesto muchas obras basadas en el marco de la red bayesiana, y se han obtenido resultados biológicamente relevantes. En Hartemink *et al*. (2001) se extendió el trabajo de Friedman *et al*. (2000) mediante la adición de estas anotaciones a los arcos: +, - o +/- que representan regulación positiva, negativa o desconocida. En Beal *et al*. (2005) propusieron incluir los genes no medidos como factores ocultos para aprender una red de genes, mediante la implementación de los *State-Space Models* (SSM). Lee y Lee (2005) plantearon un enfoque de aprendizaje modular sobre la base de la suposición de que la mayoría de los genes son propensos a estar relacionado con otros genes en los mismos módulos biológicos, en lugar de con los genes en diferentes módulos. En su trabajo, propusieron la búsqueda de módulos superpuestos en los genes, y el aprendizaje de las sub-redes en módulos con una red bayesiana. En Zhou *et al*. (2004) presentaron la construcción de RRGs probabilísticas que hacen hincapié en la topología de red utilizando una técnica de salto de cadena de Markov reversible. Finalmente, Rogers y Girolami (2005) propusieron inferir redes regulatorias por el método de regresión bayesiana, que trabaja directamente con variables continuas.

Las redes bayesianas tienen el inconveniente de excluir los aspectos dinámicos de la regulación génica, ya que deben ser grafos acíclicos. Hasta cierto punto, esto puede ser superado a través de generalizaciones como las Redes Bayesianas Dinámicas (RBD), que permiten relaciones de retroalimentación entre los genes en una red. Una RBD es una red bayesiana que ha sido temporalmente "desenrollada". Típicamente, las variables son vistas como entidades cuyo valor cambia con el tiempo. Sin embargo, si las variables se consideran como constantes (como en el modelo oculto de Markov), es posible representar, por ejemplo, a g_i en el instante t y a g_i en el instante t+1 usando dos variables diferentes, por ejemplo, g_i^t y g_i^{t+1}. Asumiendo que las dependencias condicionales no pueden apuntar hacia atrás o "lateralmente" en el tiempo, esto significa que el grafo debe ser acíclico, incluso si g_i se autorregula. Además, asumiendo que las dependencias condicionales son constantes en el tiempo y que la distribución conjunta a priori

es la misma que la distribución conjunta temporal (Friedman, 1998), la red sólo necesita ser "desenrollada" para un único instante de tiempo. Un ejemplo se proporciona en la figura 6.6.

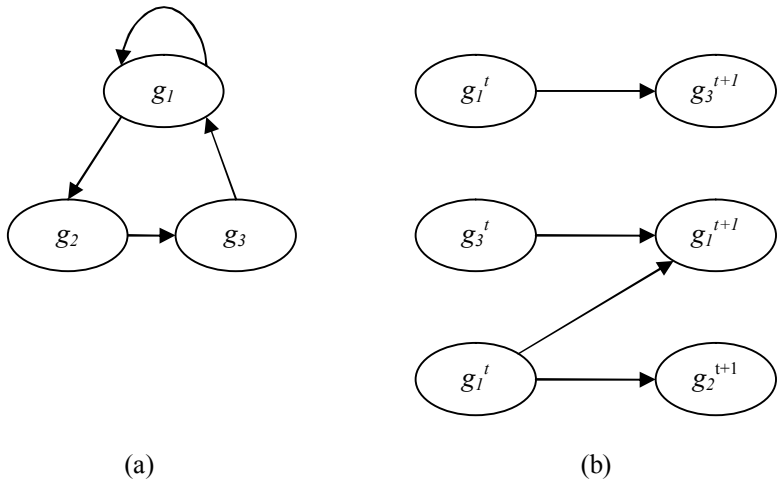


Figura 6.6. Una Red Bayesiana cíclica y su RBD acíclica equivalente. Fig 6.6a: Red Bayesiana cíclica imposible de factorizar. Fig 6.6b: RBD acíclica y equivalente.

Murphy y Mian (1999) y Gransson y Koski (2002) utilizaron las RBD para modelar redes de genes. En este modelo, como se dijo antes, un gen en un punto de tiempo se regula por su padre en el punto de tiempo anterior. Por lo tanto, la limitación de que la red bayesiana debe ser acíclica se supera con las RBD. En Murphy y Mian (1999) se da un reporte completo sobre la aplicación de RBD en el aprendizaje de redes de genes. Imoto *et al*. (2002) y Kim *et al*. (2004) extienden más allá a las Redes Bayesianas y a las RBD mediante la integración de regresión no paramétrica en los modelos, para que los métodos puedan utilizar valores continuos de expresión de genes en lugar de valores discretos, como en los enfoques generales de redes bayesianas. Sus propuestas son capaces de capturar relaciones no lineales entre los genes. En Yu *et al.* (2004) se presentó una métrica de influencia para medir la magnitud de la fuerza regulatoria de los arcos. Esta métrica es útil para eliminar los falsos positivos, así como para distinguir la regulación positiva o negativa de los arcos. Con más y más trabajos utilizando redes bayesianas como marco para abordar el problema de la reconstrucción de redes de genes (Hasty *et al.*, 2001; Pe'er *et al.*, 2001; Segal *et al.*, 2003; Friedman, 2004; Bulashevska y Eils, 2005), se está convirtiendo a la red bayesiana en un método ampliamente utilizado en esta área de la bioinformática.

La comprensión de las relaciones causales en una red es crucial en la tarea de determinar el impacto de las intervenciones en el nivel genético y, en llevar a cabo el razonamiento contra fáctico que conduce a la búsqueda de "causas". En general, las relaciones de dependencia en las redes bayesianas no dan inferencias causales únicas. Hay varios grafos que producen la misma

distribución conjunta. Las mediciones de la expresión de genes, en la ausencia de intervenciones, son insuficientes para determinar en forma inequívoca los mecanismos causales subyacentes. Recientemente, algunos estudios proporcionan métodos para inferir únicamente mecanismos causales para ciertos casos de redes bayesianas, sobre la base de datos de perturbación (Guelzim *et al.*, 2002; Yoo y Cooper, 2002; Xing y van der Laang, 2005). Sin embargo, la mayoría de las investigaciones sobre ingeniería inversa de RRGs por cualquiera de los modelos descritos en esta tesis no toman en consideración el aspecto "causal" de las relaciones génicas. En los laboratorios, el aprendizaje de las relaciones causales entre los genes se puede hacer por la anulación de todos los posibles subconjuntos de genes de un conjunto dado, y estudiar el impacto sobre los otros genes en el conjunto. Esto no es a menudo posible cuando el número de genes en el conjunto es más que un puñado. Un enfoque alternativo es utilizar los datos de expresión de genes de series de tiempo. Por desgracia, estos datos sólo se pueden obtener para células de organismos particulares, tales como la levadura. Para los tejidos humanos, los datos de expresión génica de alta densidad por lo general solo están disponibles para estado estacionario. Por lo tanto, cómo hacer para inferir relaciones causales entre los genes de datos en estado estacionario es una cuestión abierta para los investigadores de este campo.

Variaciones modernas de redes bayesianas también han añadido nuevas funcionalidades a estas, particularmente las redes bayesianas difusas. Estas van desde técnicas especialmente diseñadas para reducir la complejidad en la propagación de creencias de las Redes Bayesianas Hibridas (RBH) con aproximaciones difusas, a formalizaciones más generales (Akutsu *et al.*, 2000; Simon *et al.*, 2001; Sivakumar *et al.*, 2003; Husmeier, 2003). Estas formalizaciones generales permiten a las variables en las redes bayesianas tomar estados difusos, con todas las ventajas en robustez, comprensibilidad y reducción de dimensionalidad que esto proporciona (Friedman, 1998; Smolen *et al.*, 2000).

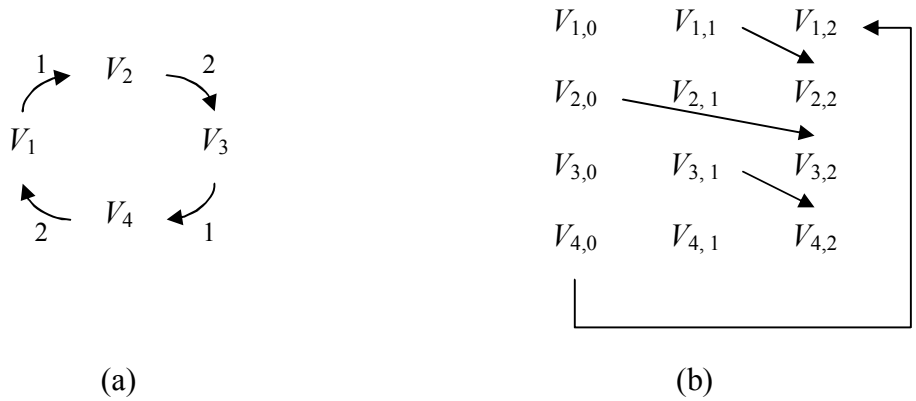
6.3.3.1 RRGs Diferidas en el Tiempo con Redes Bayesianas

En Tiefei (2005) se propone una red bayesiana diferida en el tiempo para modelar RRGs, pudiendo capturar relaciones con retrasos en la regulación de varios instantes de tiempo, así como también descubrir bucles directos que se extienden sobre al menos un instante de tiempo. La red bayesiana diferida en el tiempo se define de la siguiente manera: sea **W** el retardo máximo permitido para cada regulación. Y sea $T = <G, \theta, \delta>$, donde $G = < V, E >$ es un grafo dirigido, $V = \{V_1, V_2, \ldots, V_n\}$ es el conjunto de variables de G, y E es el conjunto de aristas dirigidas de G. Cada variable V_i representa un gen, y cada arista (V_i, V_j) representa el proceso de regulación de V_i a V_j. Para cada arista $(V_i, V_j) \in E$, $\delta(V_i, V_j)$ representa el retardo de tiempo único para la arista (V_i, V_j). Nótese que $\delta(V_i, V_j)$ es un entero y $0 \leq \delta(V_i, V_j) \leq \mathbf{W}$. θ es el conjunto de

parámetros de G que almacena la distribución de probabilidad condicional $P(V_i \mid P_a(V_i))$ para cada $V_i \in V$, donde $P_a(V_i)$ es el conjunto de padres de V_i en G. Un ciclo está permitido si al menos una de sus aristas tiene un retardo de tiempo mayor a 1. La figura 6.7a muestra un ejemplo de un ciclo con cuatro genes en una red bayesiana diferida en el tiempo.

Para modelar las redes bayesianas diferidas en el tiempo, se puede establecer una relación entre estas y las redes bayesiana tradicionales (Tiefei, 2005). Dado un tiempo de retardo máximo **W**, una variable en un intervalo de tiempo sólo puede verse afectada por variables en el segmento de tiempo actual y por variables de los **W** instantes de tiempo anteriores. Para cada variable V_i, sea $V_{i,0}$, $V_{i,1}$,..., $V_{i,\mathbf{W}\text{-}1}$, $V_{i,\mathbf{W}}$ sus estados en los **W** instantes de tiempo anteriores y en el instante actual. Aprender si la arista (V_j,V_i) tiene un retardo de tiempo w es equivalente a aprender si $(V_{j,\mathbf{W}\text{-}w},V_{i,\mathbf{W}})$ es un arco. La transformación formal se describe como sigue: dada una red bayesiana diferida en el tiempo $T = <G,\theta,\delta>$, donde $G = <V, E>$, con un tiempo de retardo máximo **W**, T se puede representar mediante una red tradicional $U = <H , \theta'>$ tal que: $H = <V',E'>$, donde V' es el conjunto de vértices y E' es el conjunto de aristas y, $V' = \{V_{i,t} \mid V_i \in V, t = 0, 1, ..., \mathbf{W}\}$. Por lo tanto, cada vértice $V_i \in V$ se transforma en **W** + 1 vértices $\{V_{i,0}, ..., V_{i,\mathbf{W}}\}$. Sea una variable $V_i \in V$, con $P_a(V_i) = \{V_{i_1}, ..., V_{i_s}\}$ siendo el conjunto de padres de V_i en G. En H, la variable $V_{i,\mathbf{W}}$ tiene s padres $V_{i_1,(\mathbf{W}\text{-}w_1)}$, $V_{i_2,(\mathbf{W}\text{-}w_2)}$, ..., $V_{i_s,(\mathbf{W}\text{-}w_s)}$ donde w_j es el retardo de tiempo $\delta(V_i,V_{i_j})$ asociado con la arista entre V_i y V_{i_j}. En el conjunto de parámetros θ', la distribución de probabilidad condicional $P(V_{i,\mathbf{W}} \mid V_{i_1,(\mathbf{W}\text{-}w_1)}, V_{i_2,(\mathbf{W}\text{-}w_2)}, ..., V_{i_s,(\mathbf{W}\text{-}w_s)})$ de U es la misma que la distribución de probabilidad condicional $P(V_i \mid V_{i_1}, ..., V_{i_s})$ de T.

La figura 6.7 muestra un ejemplo de la transformación reportado en Tiefei (2005). Se puede verificar fácilmente que la red transformada U es un grafo acíclico dirigido y que la red U contiene todos los parámetros de T. Una vez que la red de U es aprendida, los parámetros de la red de T se pueden recuperar fácilmente. Además, si el tiempo de retardo de **W** es 0, la Red Bayesiana diferida en el tiempo es de hecho una Red Bayesiana tradicional, y si **W** es 1 la Red Bayesiana diferida en el tiempo es una Red Bayesiana dinámica. Un trabajo relacionado con este enfoque es el modelo *k*-RBD. *k*-RBD fue propuesto por Boyen *et al*. (1999) para encontrar variables ocultas en una red. Aunque *k*-RBD no se utilizó para el aprendizaje de relaciones causales como es el caso en una red de genes, se puede extender a aprender la estructura de una red de genes, permitiendo más de una arista con diferentes retardos de tiempo entre el gen g_i y el gen g_j.



(a) (b)

Figura 6.7. Ejemplo de una transformación de red. Fig. 6.7a: La red diferida en el tiempo contiene cuatro variables y cuatro aristas. El entero en cada arista indica el retardo en el tiempo de la regulación, y el máximo retardo de tiempo k se asume en 2. Esta red tiene un ciclo: $V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_1$. Fig. 6.7b: La red transformada contiene 12 variables y cuatro aristas. Cada variable V_i es transformada en tres variables: $V_{i,0}$, $V_{i,1}$ y $V_{i,2}$. La arista (V_i,V_j), con retardo de tiempo w, se transforma en la arista $(V_{i,\mathbf{w}\text{-}w},V_{\mathrm{j},\mathbf{w}})$. Luego de la transformación, no existen ciclos.

6.3.4 Redes Booleanas

Los modelos de redes booleanas, introducidos originalmente en Kauffman (1990) y en Kauffman (1993), pueden proporcionar información útil en la dinámica de la red a nivel grueso. En una red booleana, una entidad puede alcanzar dos niveles alternativos: activo (1) o inactivo (0). Por ejemplo, un gen puede ser descrito como expresado o no expresado en cualquier momento. El nivel de cada entidad se actualiza de acuerdo a los niveles de varias entidades, a través de una función booleana específica. El vector de ceros y unos que describe los niveles de todas las entidades se llama estado del sistema, o estado global. Se supone que cambia sincrónicamente, de tal manera que en cada instante de tiempo, el nivel de cada entidad se determina de acuerdo a los niveles de sus reguladores en el instante de tiempo anterior acorde a la función de regulación. La figura 6.8 es un ejemplo de una red booleana. En muchos casos, no se han establecido las relaciones regulatorias entre los componentes de una red y, por lo tanto, deben ser derivados a partir de datos experimentales. Para cualquier entidad bajo un modelo de red booleana, tanto sus reguladores como la función regulatoria (siendo esta consistente con un conjunto de perfiles de expresión de genes) se pueden encontrar de manera eficiente, a condición de que el número de los reguladores de cada entidad no sobrepase un límite establecido (Lahdesmaki *et al.*, 2003).

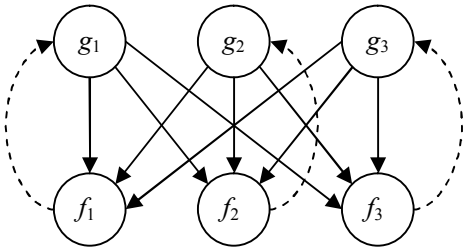


Figura 6.8. Una red booleana. Por claridad, cada $f \in F$ ha sido puesta dentro de un nodo. Normalmente las funciones están implícitas en los arcos la red. $f_1 = \neg g_1 \wedge g_2 \wedge g_3, f_2 = \neg g_1 \vee g_2 \wedge g_3, f_3 = g_1 \vee g_3$.

La redes booleanas no modelan correctamente la dinámica de un factor de transcripción que inhibe su propia expresión debido al limitado nivel de detalle del modelo (Kauffman *et al.*, 2003). Otro problema es el alto costo computacional asociado a analizar la dinámica de las redes de gran tamaño, ya que el número de estados globales es exponencial en el número de entidades. Sin embargo, cuando el número de entidades es pequeño y sólo está disponible conocimiento cualitativo, las redes booleanas pueden proporcionar información importante, como la existencia y naturaleza de los estados estacionarios o la robustez de la red. Por otra parte, para el modelado de sistemas de regulación génica a gran escala, las redes booleanas pueden representar la única alternativa práctica (Smolen *et al*., 2000).

Los datos de *microarrays* exhiben incertidumbre en varios niveles, como se dijo antes. En primer lugar, existe incertidumbre biológica en forma de ruido intrínseco y extrínseco. En segundo lugar, hay ruido experimental debido al proceso complejo de medición, que va desde las condiciones de hibridación a las técnicas de procesamiento de imágenes de *microarrays*. En tercer lugar, es posible que estén interactuando variables latentes, tales como proteínas, diversas condiciones ambientales, u otros genes que no se miden, proporcionando otras fuentes de variabilidad en las mediciones. Para hacer frente a la incertidumbre, en Shmulevich *et al*. (2002) se introdujeron las Redes Booleanas Probabilísticas (RBP) mediante la asociación de varios predictores con cada gen blanco. Si el gen objetivo g_i tiene $l(i)$ funciones de predicción asociadas, $f_1^{(i)}$, $f_2^{(i)}$, ..., $f_{(i)}^{(i)}$, luego en cada punto en el tiempo t se selecciona una de estas funciones para formar la regla de transición para g_i en el tiempo t+1. Claramente, si $l(i) = 1$ para todo $i = 1,2, ..., N$, la RBP se reduce simplemente a una red booleana estándar. El bloque de construcción básico de una RBP se muestra en la figura 6.9. El diagrama de cableado para toda la RBP consiste en n de tales bloques de construcción. Conceptualmente, el predictor probabilístico de cada gen blanco puede ser pensado como un interruptor al azar, en donde en cada punto de tiempo de la red, la función $f_k^{(i)}$ se elige con probabilidad $c_k^{(i)}$ para predecir el gen g_i. Una manera de asignar estas probabilidades es emplear el Coeficiente de Determinación

(CoD) (Shmulevich *et al.*, 2002), normalizado de tal manera que $\sum_{k=1}^{l(i)} c_k^{(i)} = 1$. Es decir, $c_k^{(i)} = \theta_k^{(i)} / \sum_{j=1}^{l(i)} \theta_k^{(i)}$, donde $\theta_k^{(i}$ es el CoD para el gen blanco g_i relativo a los genes utilizados como entradas al predictor $f_k^{(i)}$.

En el contexto de las RBP, en Hashimoto *et al*. (2004) desarrollaron un método para hacer crecer una red a partir de un número más pequeño de genes de interés, o genes semilla. El algoritmo propuesto es flexible y permite varias opciones de diseño respecto a la forma de proceder, como la métrica de fuerza de conexión entre los genes, el protocolo de búsqueda, y las condiciones de detención. Como ejemplo, el CoD (Shmulevich *et al*., 2002) se puede asignar como función para la medición de la fuerza. La identificación de los genes semilla de interés es un paso crítico en este algoritmo, por lo que generalmente se seleccionan con la ayuda de conocimientos biológicos previos.

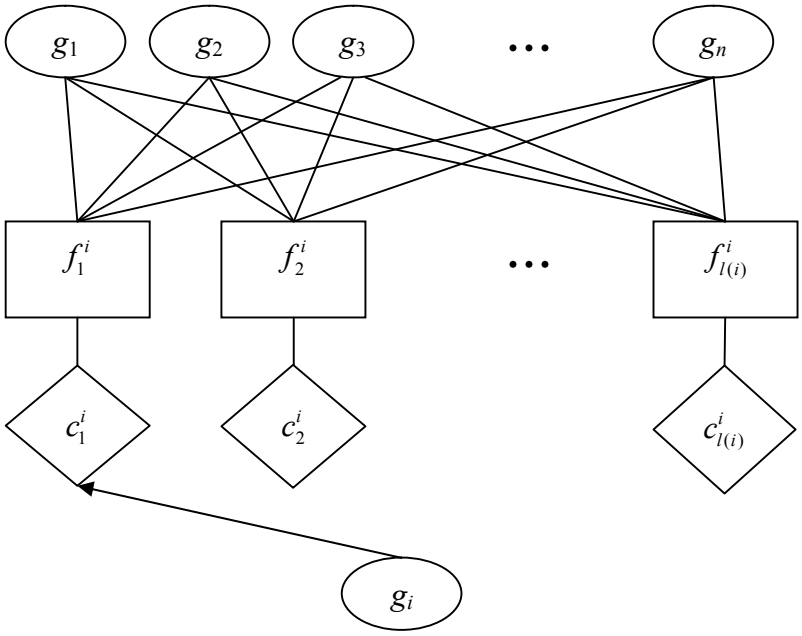


Figura 6.9. Un bloque de construcción básico de una RBP.

Aunque bueno en abstraer la incertidumbre en el sistema biológico, el modelo de RBP falla al describir el determinismo específico del contexto de los sistemas regulatorios. Este contexto puede ser definido como una determinada condición en la que un número limitado de genes están estrechamente regulados entre sí para un mecanismo celular específico o una tarea específica. Esta tarea específica puede ser una etapa de desarrollo diferente, o función específica del tejido, lo que resulta en un tipo de célula específica. El cambio de este contexto resultará en el cambio del conjunto de genes que están interactuando intensamente, y probablemente también cambie su conectividad y relaciones. Diferentes contextos biológicos también pueden correlacionar con diferentes enfermedades o pueden ser una de las razones por la cual un cierto

grupo de pacientes responden a una terapia, mientras que otros no. En Li *et al.* (2004) desarrollaron un modelo de red booleana sensible al contexto (cBN) para describir el comportamiento de los sistemas celulares. Una cBN puede ser considerada como una RBP restringida, donde la restricción es la manera de asignar la probabilidad para el modelo. La inferencia de reglas se basa en la suposición de que las reglas inferidas y las observaciones son consistentes dentro de un contexto dado. La figura 6.10 muestra un ejemplo de cBN que contiene dos contextos y quince genes.

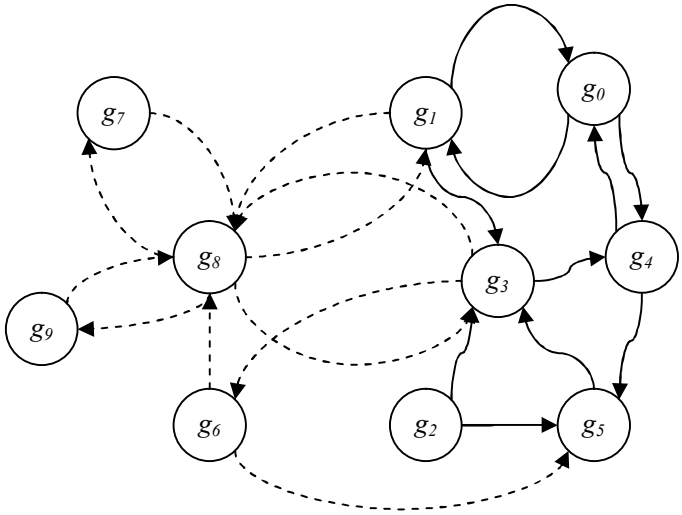


Figura 6.10. Un ejemplo de una cBN con dos contextos.

6.3.4.1 RRGs diferidas en el tiempo con Redes Booleanas

Con el fin de hacer frente al retardo en la regulación génica, Silvescu y Honavar (2001) propusieron un algoritmo que utiliza los datos de series de tiempo para encontrar redes booleanas temporales (*Temporal Boolean Networks* o TBoN). Las TBoN se desarrollaron para modelar retrasos regulatorios, que pueden ocurrir debido a la falta de genes intermedios y retrasos espaciales o bioquímicos entre la transcripción y la regulación, como se dijo antes. Un ejemplo de una red booleana temporal se presenta en la figura 6.11.

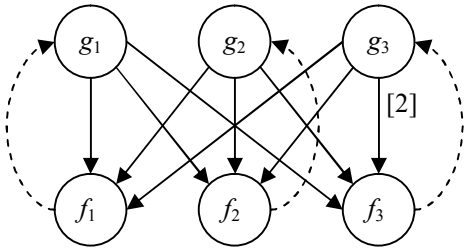


Figura 6.11. Una red booleana temporal. Las funciones regulatorias son como en la figura 6.8, pero los retardos se muestran entre corchetes entre los genes y las funciones. El retardo por defecto si no hay ninguna anotación presente se asume en 0.

Una TBoN es muy similar a una red booleana normal, excepto que las funciones $f \in F$ pueden referir a niveles de expresión génica ya ocurridos. En lugar de depender sólo de t para inferir t+1, los parámetros de f_i se pueden anotar con un número entero representado un retraso temporal. Por ejemplo: $g_3 = f_3 (g_1, g_2, g_3) = g_1^{0} \vee g_3^{-2}$ significa que g_3 se expresa en t+1 si g_1 se expresó en t o si g_3 se expresó en el tiempo t-2. Las TBoN también pueden ser reformuladas y deducirse como árboles de decisión.

6.3.5 Otras Técnicas

En los últimos años, se han propuesto otros métodos que no entran en la clasificación anterior. Sin embargo, estos métodos también pueden utilizarse para inferir RAs de genes a partir de datos de *microarray*. Estas técnicas se describen brevemente en los apartados siguientes.

6.3.5.1 Clustering

Uno de los principales problemas que dificultan la investigación en la reconstrucción de redes de genes es el problema de la dimensión, es decir, hay muchos genes con muy pocas replicas. Un enfoque útil es agrupar genes con patrones de expresión similares en grupos, y luego inferir las relaciones regulatorias entre los grupos (Tiefei, 2005). Los investigadores creen que los genes con patrones similares de expresión tienen funciones similares o están involucrados en los mismos eventos biológicos (Wahde y Hertz, 2000). Actualmente, se utilizan varios métodos de *clustering* para este propósito. Diferentes métodos de *clustering* pueden generar resultados muy diferentes. Cada combinación de métrica de distancia y de algoritmo de *clustering* tiende a enfatizar un tipo diferente de regularidad en los datos. No existe un criterio único para la elección del mejor método de *clustering*. Cómo elegir el método depende del énfasis particular deseado.

Una vez encontrados los grupos de genes, también hay varios métodos para encontrar las interacciones entre ellos. En Chen *et al.* (1999) redujeron 3.131 genes de levadura en 308 grupos por medio de *clustering* de enlace promedio. Luego, utilizaron un algoritmo de enfriamiento simulado para optimizar una red cualitativa basada en la sincronización de los picos en los datos. En Wahde y Hertz (2000) agruparon 65 genes de conjuntos de datos correspondientes a médula espinal e hipocampo de rata en cuatro "señales" utilizando el algoritmo de agrupamiento jerárquico de Fitch y Margoliash (1967). Luego, mediante un algoritmo genético, se construyó una red neuronal recurrente de tiempo continuo de cuatro nodos. En Someren *et al.* (2000) se redujeron 2.467 genes de levadura en t-1 grupos representando a cada grupo por medio de un gen "prototipo" calculado a partir del grupo correspondiente. A continuación, se generó un

modelo lineal de los genes prototipo mediante regresión lineal. En Toh y Horimoto (2002) se propuso promediar los valores de expresión génica de cada grupo, y luego descubrir las relaciones regulatorias por medio del Modelado Gráfico Gaussiano. Finalmente, en Guthke *et al.* (2005) se propuso agrupar los genes en clases eligiendo genes representativos de cada uno. Luego, modelaron las conexiones entre los genes representativos mediante ecuaciones diferenciales.

6.3.5.2. Métodos de relaciones por pares.

Los métodos por pares tratan de descubrir las relaciones entre los genes por medio de comparaciones entre pares de genes. No tienen en cuenta las interacciones donde la expresión de un gen se logra mediante los efectos combinados de múltiples genes. En Arkin *et al.* (1997) se propuso la Construcción de la Métrica de Correlación (CMC). La CMC calcula la magnitud de los pares de genes por correlación cruzada. Se construye una matriz de distancia para cada par de genes mediante la comparación de sus similitudes con otros genes. A continuación, se elabora un diagrama para resumir la fuerza de interacción y predecir las conexiones mecánicas entre los genes. En Chen *et al*. (1999) se propuso usar redes de activación/inhibición para encontrar la regulación en función de si los picos de una señal preceden picos en otra señal, agrupando los genes con perfiles de expresión similares. A continuación, se genera un prototipo para cada grupo de genes por promedio de los valores de expresión de los genes en el grupo. Cada prototipo representa un grupo de genes con patrones de expresión similares y se representa como una serie de picos. Las correlaciones entre los pares de prototipos se calculan para determinar el tipo de relaciones regulatorias (activación, inhibición o incomparable) y medir la fuerza de la relación regulatoria entre dos prototipos. Por último, la matriz de regulación es generada siguiendo esos resultados. En Ponzoni *et al*. (2007) se propuso un algoritmo de aprendizaje automático llamado GRNCOP basado en optimización combinatoria que no asume ninguna discretización de valores de expresión de genes arbitraria ni uniforme. Los umbrales se calculan de forma dinámica mediante la aplicación de las mismas técnicas de discretización de atributos de valores continuos utilizados para los algoritmos de clasificación basado en árboles de decisión. Entonces, cada posible par de genes es evaluado obteniendo una RA con una precisión particular, sobre la base de una función objetivo. Finalmente, sólo se informan las relaciones que alcanzaron un valor de precisión superior a un umbral preseleccionado.

6.3.5.3 Métodos de Maquinas de soporte vectorial

Los métodos de máquinas de soporte vectorial (*Support Vector Machine* o SVM) han atraído un gran interés en la comunidad bioinformática en los últimos años debido a su buen rendimiento en la predicción para diversas tareas. Se basan en los principios de la teoría de

aprendizaje estadístico (Schölkopf y Smola, 2002). La idea es construir un hiperplano óptimo entre dos clases +1 y -1 tales que el margen, es decir, la distancia del hiperplano hasta el punto más cercano a él, se maximiza. Para permitir la clasificación no lineal, se emplean las llamadas funciones núcleo, que pueden ser pensadas como métricas especiales de similaridad. Estas mapean implícitamente los datos originales en algún espacio de alta dimensionalidad, en el que es posible encontrar al hiperplano óptimo. Como ejemplo, suponga que se consideran los núcleos lineales $k(x,x') = \langle x,x' \rangle$ así como los núcleos polinomiales de grado 2 $k(x,x') = \langle x,x' \rangle^2$, donde x y x' son los niveles de expresión de todos los genes, excepto del gen g_i en la muestra j. El núcleo polinomial calcula implícitamente todos los productos por pares entre los niveles de expresión de dos genes. De esta manera, pueden ser capturadas tanto dependencias lineales como también no lineales entre las expresiones de los genes. Además de una función núcleo, se debe fijar un parámetro de margen suave C. En Guyon *et al.* (2002) se propone un algoritmo llamado RFE capaz de determinar, para cada gen g_i, los genes que se adaptan mejor para predecir su estado. Este algoritmo elimina sucesivamente el gen que influye en menor medida en el tamaño del margen. La terminación de este procedimiento se puede determinar mediante validación cruzada (*10-fold cross-validation*).

6.4 Sumario

El desarrollo de métodos computacionales para la modelación de RRGs es un tema de investigación "caliente". La principal contribución de este capitulo, publicado en Gallo *et al*. (2013a), es la extensa revisión de una familia específica de algoritmos para extraer RAs entre genes. La ingeniería inversa de RRGs a partir de RAs tiene una ventaja metodológica importante: permite la reconstrucción de redes de modelo libre. En otras palabras, estas técnicas no requieren, en general, ninguna restricción o conocimiento previo sobre las relaciones estructurales de la red, ni hacen suposiciones relacionadas con los principios fisicoquímicos que rigen las interacciones entre genes. Estos métodos sólo necesitan información de expresión de genes como fuente de datos para el proceso de inferencia.

Todas las técnicas examinadas proceden de diversos enfoques de minería de datos, pero la mayoría de ellas comparten aspectos comunes, como algunos pasos de pre-procesamiento. En particular, la discretización de los datos de expresión génica constituye un punto central para estos métodos, con importantes implicaciones semánticas. Como se ha descrito, existen varios algoritmos para hacer frente a este problema, que van desde discretizaciones arbitrarias simplistas a métodos adaptativos elaborados. Por otra parte, también se presentaron aspectos adicionales complejos, que surgen como parte de este paso de pre-procesamiento cuando los estados de transición se modelan a partir de datos de series temporales.

En cuanto a las metodologías de inferencia, se ilustró una amplia variedad de técnicas, como métodos basados en conjuntos de elementos frecuentes, árboles de clasificación y regresión, redes bayesianas, redes booleanas, máquinas de soporte vectorial, enfoques de agrupamiento y algunos algoritmos por pares. Para la mayoría de estos enfoques de minería de datos, se revisaron varios algoritmos, haciendo hincapié en sus ventajas y limitaciones.

Otro punto relevante es la inferencia de asociaciones temporales entre los genes. Este punto fue abordado de manera transversal a lo largo del capítulo, ilustrando de que manera los diferentes métodos de minería de datos consideran este tipo de reglas diferidas en el tiempo. También se analizaron en detalle temas adicionales como el modelado de la cardinalidad de las reglas, la validación estadística y biológica de la red, y la extracción de asociaciones a partir de múltiples fuentes de datos.

Capítulo 7

Inferencia de Reglas Diferidas en el Tiempo a Partir de Datos de *Microarray* Usando Clasificadores de Perfiles de Expresión

En este capítulo, se presenta un nuevo método de aprendizaje automático para la inferencia de reglas de asociación diferidas en el tiempo, a partir de datos de expresión de genes en series de tiempo. Las relaciones descubiertas, que representan interacciones potenciales entre genes, se pueden usar para predecir los estados de expresión de un gen en términos de los valores de expresión génica de otros genes y, de esta manera, puede entonces reconstruirse una RRG putativa mediante la aplicación y la combinación de estas reglas. El enfoque ofrece varias características relevantes y distintivas en relación con la mayoría de los métodos existentes. En primer lugar, el criterio de discretización utilizado en este trabajo para los valores de expresión de los genes no es ni arbitrario ni uniforme. En segundo lugar, se pueden inferir reglas con múltiples retrasos de tiempo. Además, los resultados pueden ser interpretados fácilmente ya que las reglas se derivan de esquemas que clasifican los diferentes estados de regulación. Asimismo, el algoritmo puede inferir las relaciones entre los genes de forma automática a partir de múltiples datos de series de tiempo de *microarrays*. Por último, el nuevo método es capaz de procesar datos a gran escala con el fin de realizar estudios a nivel de genomas completos. El resto del capítulo se organiza como sigue: en el próximo apartado, se presenta la metodología utilizada y las principales características del nuevo algoritmo, y se describen dos fases experimentales. Seguidamente, se describe una herramienta de software llamada GeRNet que integra al algoritmo descrito en este capitulo conjuntamente con el algoritmo BiHEA, y que incluye características para la visualización y manipulación de datos y de RRG.

7.1. GRNCOP2

Como se discutió en el capítulo anterior, se han propuesto muchas técnicas de inteligencia artificial y estadística para reconstruir las RRGs a partir de datos de expresión de genes. Entre estas se encuentra el trabajo presentado por Ponzoni *et al.* (2007), en donde se propone un algoritmo de aprendizaje automático, llamado GRNCOP, basado en optimización combinatorial que no asume ninguna discretización arbitraria ni uniforme de los valores de expresión de los genes. Los umbrales son calculados dinámicamente por medio de la aplicación de las mismas técnicas de discretización de atributos continuos usadas en los algoritmos de clasificación basados en árboles de decisión. Sin embargo, este proceso de discretización es empleado solo

para los genes reguladores, dado que los umbrales de discretización para los genes regulados son calculados mediante el valor de expresión promedio. Otra limitación es que solo es capaz que inferir reglas con a lo sumo una unidad de retardo de tiempo. GRNCOP es el ancestro del enfoque propuesto en este capítulo, llamado GRNCOP2, en el cual todas las limitaciones aquí mencionadas son superadas y en el que además se incorporan nuevas características.

Por otra parte, con la excepción de los métodos de *clustering*, la gran mayoría de los algoritmos de minería de datos basados en RAs para datos de expresión de genes se han evaluado solamente con conjuntos de datos muy reducidos. A pesar de que inferir RAs sobre una pequeña cantidad de datos que puede dar una idea aproximada de la eficacia de un método, en cualquier escenario real el gran tamaño y la cantidad de datos disponibles imponen otro desafío en la reconstrucción de RRGs que muy pocos autores han considerado: el problema de la escalabilidad (Alves *et al.*, 2010). Esta cuestión representa una de las debilidades más importantes de los métodos de inferencia basados en RAs, debido a la falta de pruebas de que realmente pueden funcionar e inferir relaciones relevantes con grandes conjuntos de datos, impidiendo así su aplicación en cualquier estudio complejo. En este contexto, el algoritmo presentado en el presente trabajo exhibe la mayor parte de las características deseables mencionadas previamente, y además, trata con éxito los principales inconvenientes detectados en los métodos existentes.

7.1.1. Definiciones

Siguiendo el esquema adoptado en el contexto de esta tesis, la serie de tiempo codificada en el conjunto de datos de expresión de genes está representada por medio de una matriz de datos de expresión génica, X, donde las filas y las columnas representan los genes y los puntos temporales, respectivamente. De esta manera, cada elemento x_{ij} de X contiene el valor de la expresión del gen i (g_i) en el punto de tiempo (muestra o condición experimental) j. Aunque los valores de expresión de los genes pertenecen a un intervalo continuo de los números reales, es posible definir un conjunto de estados de expresión finitos para cada gen por medio de un procedimiento de discretización, tal y como se mencionó en el capítulo anterior. En general, se requiere tal procedimiento con el fin de codificar las entradas para cualquier proceso de optimización combinatorial o método de aprendizaje automático. En el método presentado en este capítulo se trabaja con dos estados para cada gen: activo, cuando el gen se expresa con un valor superior a un umbral de discretización específico, e inactivo o inhibido, cuando el gen se expresa con un valor inferior o igual a un umbral específico de discretización. Por lo tanto, el proceso de inferencia requiere la definición de umbrales de discretización con el fin de inferir relaciones regulatorias putativas entre los genes. Estos "umbrales de

discretización" tradicionalmente han sido estimados como valores estáticos únicos para todos los genes en estudio. Por ejemplo, como se mencionó antes, se han aplicado métodos *ad hoc* basados en los valores promedio de expresión. Sin embargo, un esquema más significativo desde el punto de vista biológico debe modelar el hecho de que un gen en realidad puede tener diferentes umbrales de discretización en relación con diferentes genes en la RRG (Soinov et al., 2003; Ponzoni *et al.*, 2007). Por ejemplo, con respecto a la red de regulación en estudio en el trabajo presentado en este capítulo, que se corresponde con el organismo *Saccharomyces cerevisiae*, los genes *CLB2* y *SWI5* son potencialmente activados por el gen *CLB1*, pero sus respectivos umbrales de activación son diferentes. Por lo tanto, un problema fundamental consiste en la estimación de los umbrales de regulación para cada gen en relación con cada gen potencialmente regulado, pudiendo reflejar interacciones significativas entre ellos con un mayor nivel de precisión.

En este sentido, se definen dos tipos diferentes de discretizaciones. En términos generales, la primera es para establecer el estado de cada gen blanco, y se llama Umbral de Discretización del Blanco (*Target Discretization* Threshold o TDT). La segunda es para evaluar la interacción potencial entre cada par de genes y se calcula de manera adaptativa entre pares de genes. Esta última discretización se llama Umbral de Regulación Relativo (*Relative Regulation Threshold o* RRT).

En este punto, nuestra hipótesis se fija como sigue: las reglas - potenciales relaciones regulatorias - se pueden inferir con precisión a partir de datos de series temporales de expresión de genes para revelar cómo el estado presente y futuro de un gen puede verse afectado por los valores de expresión génica de otros genes, teniendo en cuenta su RRT. Asimismo, consideramos que las reglas diferidas en el tiempo representan la situación en la que el estado de un g_i en un punto de tiempo j depende de los valores de expresión de otros genes en el punto temporal j - w , donde w es un número entero no negativo que representa el tiempo de retardo en la relación. La sintaxis de las reglas es: <*simbolo*> $<g_r>$ $\rightarrow w$ <*simbolo*> $<g_i>$ donde g_r y g_i representan el gen regulador y el gen blanco, respectivamente. Un símbolo +(-) en el lado izquierdo de la regla indica que el gen g_r está por encima (por debajo) de algún RRT con respecto al g_i, mientras que un símbolo +(-) en el lado derecho de la regla indica activación (inhibición) del gen g_i, dependiendo de su TDT. Por ejemplo, la regla +/- *CLB1* $3\rightarrow$ +/- *CLB5* denota que si *CLB1* está por encima de su RRT en relación a *CLB5*, $t_{\mathrm{CLB1,CLB5}}$, en un punto de tiempo j, entonces *CLB5* estará expresado en el punto de tiempo j+3 y, si *CLB1* está por debajo o iguala a $t_{\mathrm{CLB1,CLB5}}$ en una muestra j, entonces *CLB5* estará inhibido o sin expresión en la muestra j+3 . Los tipos de reglas obtenidos a través de este esquema son similares a los estudiados en el capítulo anterior. Este esquema permite representar explícitamente reglas tanto

simultáneas como diferidas en el tiempo en cualquier unidad de tiempo, que constituyen el tipo de reglas que GNRCOP2 es capaz de inferir.

GRNCOP2 infiere las reglas de asociación descritas anteriormente mediante la exploración de las posibles combinaciones de interacción entre cada par de genes. En este sentido se asumen seis casos particulares, que están representados por los números enteros no nulos entre -3 y 3, y un caso especial que indica la ausencia de cualquier relación representado por el número 0. Todos estos casos se describen en la tabla 7.1.

Tabla 7.1. Tipos de reglas inferidas por GRNCOP2, donde *w* denota el retardo de tiempo en la regulación.

Tipo de regla	Regla diferidad en el tiempo asociada
-3	$+ g_r \; w \rightarrow - g_i$
-2	$- g_r \; w \rightarrow + g_i$
-1	$+/- g_r \; w \rightarrow -/+ g_i$
0	g_r no interactua con g_i
1	$+/- g_r \; w \rightarrow +/- g_i$
2	$+ g_r \; w \rightarrow + g_i$
3	$- g_r \; w \rightarrow - g_i$

En términos matemáticos, la inferencia de las reglas para reconstruir una RRG puede expresarse como el siguiente problema de optimización:

$$\bigcup_{i=1}^{n} \max_{\bar{\pi}_i \in P} \sigma^*\left(\bar{\pi}_i, \delta\left(X, i\right)\right). \tag{7.1}$$

sujeto a:

- n es el número de genes en el conjunto de datos de *microarrays*.
- m es el número de puntos de tiempo o muestras en el conjunto de datos de *microarrays*.
- $\boldsymbol{X} \in \Re^{n \times m}$ es la matriz con los datos de expresión.
- $\boldsymbol{P}$ es el espacio de todos los vectores v de dimensión n tal que $v(r) \in \{-3, -2, -1, 0, 1, 2, 3\} \; \forall r, \; r = 1..n$.
- $\delta(\boldsymbol{X}, i)$ es la función de discretización tal que $\delta(\boldsymbol{X}, i) = D_i$ y $D_i \in \{-1,1\}^{n \times m}$.
- $\bar{\pi}_i \in \boldsymbol{P}$ es un clasificador para D_i.
- $\sigma^*(\bar{\pi}_i, D_i)$ es una función general de evaluación de $\bar{\pi}_i$ como clasificador de D_i.

A partir de ahora, el símbolo Π^w indica el conjunto de clasificadores óptimos, $\Pi^w = \{\overline{\pi}_1, \overline{\pi}_2, ..., \overline{\pi}_n\}$, para un tiempo de retardo *w* dado. Es importante notar que el problema de optimización general es el mismo para todas las reglas diferidas en el tiempo. La única diferencia radica en la definición de la función δ ***(X, i)***, debido a que las reglas con retardo están basadas en discretizaciones de valores de expresión de ***X*** que consideran el desplazamiento temporal requerido.

7.1.2. Algoritmo

Aunque las ideas básicas detrás de GRNCOP, más concretamente, los umbrales de regulación adaptativos y la optimización combinatorial de clasificadores de reglas, permanecen en GRNCOP2, el nuevo método constituye una evolución importante del algoritmo anterior, debido a los retos que imponen las mejoras propuestas. La figura 7.1 muestra una representación abstracta del algoritmo. El proceso de aprendizaje automático utilizado para obtener las reglas realiza iterativamente la búsqueda a través de todos los conjuntos de datos en todos los tiempos de retardo requeridos. El algoritmo recibe como entrada **K** conjuntos de datos de series de tiempo y devuelve un arreglo **Π** de dimensión **W** que contiene, en cada posición *w*, el conjunto de reglas obtenidas con un tiempo de retardo *w*. En la próxima sección se explicarán en detalle las principales características del procedimiento.

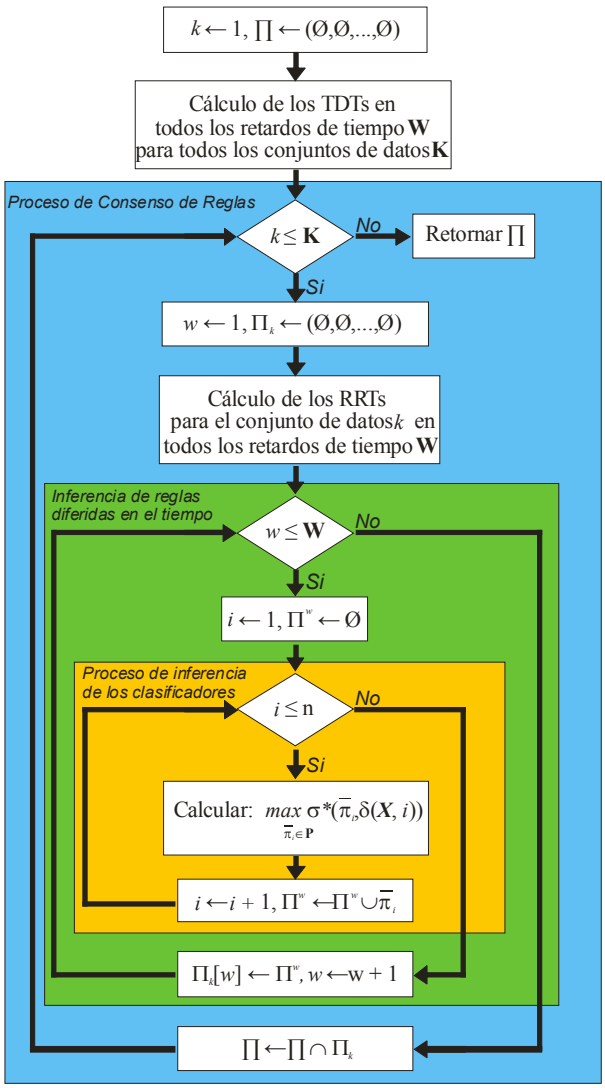


Figura 7.1. Esquema general del algoritmo GRNCOP2.

7.1.2.1. Una técnica de discretización mejorada para los genes blanco

Con el fin de obtener los TDTs de los genes blanco, GRNCOP2 emplea una técnica que es capaz de inferir los estados del gen de una manera más precisa, en comparación con el valor de la expresión media utilizada en GRNCOP. En términos matemáticos, el procedimiento para la discretización de un gen g_i se puede definir de la siguiente manera:

$$\min_{S_1, S_2 \subset S} (\mathrm{var}(S_1) + \mathrm{var}(S_2)) . \tag{7.2}$$

Sujeto a:

- S es el conjunto de muestras para g_i,
- $S_1 \cap S_2 = \emptyset$, $S_1 \cup S_2 = S$, $|S_1|>1$ y $|S_2|>1$.
- $var(S_1)$ y $var(S_2)$ son la varianza de S_1 y S_2 respectivamente.
- S_1 y S_2 representas los dos estados de expresión para g_i.

Básicamente, el procedimiento divide las muestras de g_i en los dos conjuntos que tienen la menor suma de sus varianzas. La cardinalidad de S_1 y S_2 debe ser mayor que uno con el fin de evitar los efectos de un posible valor atípico en las muestras, ya que es improbable que un gen esté claramente expresado o inhibido en un solo punto de tiempo. Por lo tanto, cuando las muestras de g_i se separan en una partición que viola esta restricción, el gen g_i ya no se considera en el proceso de inferencia para el conjunto de datos actual. Otro enfoque podría haber sido el de excluir el punto de tiempo conflictivo en la búsqueda de la partición. Sin embargo, esto puede llevar a la misma situación descrita anteriormente, reintroduciendo el problema que supuestamente debía ser corregido.

Esta técnica es, en esencia, un procedimiento de *clustering* similar a un *k*-medias con $k = 2$. Sin embargo, como el número de grupos es 2 y los elementos de S tienen un orden total, el problema puede ser resuelto de manera óptima y eficiente a través del siguiente procedimiento determinista:

- Ordenar los elementos de S en un arreglo L.
- Buscar el elemento e tal que $var(L[1..e]) + var(L[e+1 ..|S|])$ sea el mínimo.
- Retornar $(L[e]+L[e+1])/2$ como el TDT para g_i.

Es importante remarcar que, acorde a la figura 7.1, los valores TDT son calculados para cada conjunto de datos en forma separada.

7.1.2.2. Proceso de consenso de reglas

En esencia, el bucle principal del algoritmo aplica el mismo método de inferencia a los **K** conjuntos de datos dados como entrada, y luego devuelve la intersección de los resultados en cada conjunto de datos. El objetivo de este procedimiento, incorporado en GRNCOP2, es evaluar automáticamente las reglas obtenidas por el algoritmo a través de diferentes conjuntos de datos, aumentando así el grado de evidencias requerido en las potenciales relaciones regulatorias a ser retornadas. La intersección de las reglas obtenidas a partir de dos conjuntos de datos k_1 y k_2 se define como sigue:

$$\Pi_{k_1} \cap \Pi_{k_2} = (\Pi^1_{k_1} \cap \Pi^1_{k_2}, \Pi^2_{k_1} \cap \Pi^2_{k_2}, ..., \Pi^W_{k_1} \cap \Pi^W_{k_2}). \quad (7.3)$$

donde:

- **W** es el máximo retardo de tiempo establecido por el investigador.
- Π_{k_1} y Π_{k_2} son las reglas obtenidas a partir de los conjuntos de datos k_1 y k_2 respectivamente en todos los retardos de tiempo **W**.
- $\Pi_{k_1}^{w}$ y $\Pi_{k_2}^{w}$ son las reglas obtenidas a partir de los conjuntos de datos k_1 y k_2 respectivamente en un retardo de tiempo *w*.

Básicamente, la intersección de los resultados es la intersección de cada componente (tiempo de retardo *w*) de los conjuntos de reglas obtenidos a partir de los **K** conjuntos de datos. En un escenario ideal, todos los conjuntos de datos de series de tiempo deberían tener la misma frecuencia de muestreo en los puntos de tiempo, pudiendo establecer de esta manera una correspondencia directa entre el tiempo de retardo *w* y los puntos de tiempo de los conjuntos de datos. Sin embargo, esto está lejos de cualquier situación real debido a la limitada disponibilidad de réplicas para el mismo experimento. Es más, los conjuntos de datos de series de tiempo de *microarrays* podrían haber sido obtenidos bajo condiciones experimentales completamente diferentes, por lo que las frecuencias de muestreo de cada uno de ellos podrían ser incomparables debido a posibles retrasos en el proceso de regulación introducidos por estas condiciones experimentales. No obstante, este tipo de proceso de consenso para las reglas fue realizado manualmente por otros autores (Soinov *et al.*, 2003; Li *et al.*, 2006; Ponzoni *el at.*, 2007) sin ningún tipo de re-muestreo o integración de los datos. Esto lleva a la interpretación de un retardo de tiempo *w* como una unidad abstracta que denota una posible relación futura entre los genes que participan en la regla, introduciendo la noción de antes, igual y después de un tiempo, pero no asumiendo cuándo ocurrirá exactamente.

Este tipo de proceso de consenso no limita el número de datos de *microarrays* empleados en el proceso de inferencia. Por lo tanto, puede surgir la siguiente pregunta: ¿es necesario evaluar las reglas en todos los conjuntos de datos de *microarrays*? Y una respuesta clara es no. De esta manera, hemos introducido un parámetro en el proceso de consenso, denominado Precisión de Consenso de Reglas (*Rule Consensus Accuray* o RCA), que especifica la proporción mínima de conjuntos de datos en los que una regla debe "predecir bien" para ser devuelta por el algoritmo como una potencial relación. Este parámetro no impone ningún orden de importancia entre los conjuntos de datos y, por tanto, todos ellos tienen el mismo peso en el proceso de consenso. De este modo, por ejemplo, si el algoritmo se ejecuta con 10 conjuntos de datos de series de tiempo y el parámetro RCA se establece en 0.60, significa que las reglas devueltas por el algoritmo

"predicen bien" en al menos 6 de los conjuntos de datos, no importa cuáles. En este sentido, y con el fin de determinar el mejor valor para este parámetro, los investigadores deberán tener en cuenta el número de conjuntos de datos disponibles y, siguiendo el ejemplo anterior, contestar una pregunta como ésta: ¿es suficiente evidencia de una relación regulatoria viable que una regla sea soportada por al menos 6 de los 10 conjuntos de datos (RCA = 0.60)? La respuesta dependerá naturalmente de la naturaleza biológica de los experimentos y del criterio del investigador.

7.1.2.3. Umbrales de regulación relativos

Durante la discretización, los números reales correspondientes a los valores de expresión génica, que se mantienen en la matriz ***X***, son mapeados a valores de -1 y 1 mediante la función δ ***(X, i)***. La principal cuestión en este punto es cómo definir los RRT para cada gen en relación con los otros. Un enfoque tradicional consiste en utilizar el valor promedio de expresión de un gen g_r en su correspondiente conjunto de muestras en ***X***. Esta solución es fácil de implementar, pero representa una fuerte simplificación de la realidad, ya que supone un umbral de regulación putativo único para cada gen con respecto a los otros. En este sentido, es bien sabido que el valor de expresión requerido por un gen g_r para activar (o inhibir) a un gen g_{i1} puede no necesariamente ser el mismo valor requerido por el mismo g_r para activar (o inhibir) a otro gen g_{i2}. Por esta razón, en GRNCOP2 se aplica una política de selección de umbrales más flexible y dinámica que calcula un umbral de regulación específico para cada par de genes, tal y como se empleó previamente en GRNCOP (Ponzoni *et al.,* 2007).

En esencia, GRNCOP2 considera cada valor de expresión de un gen g_r en ***X*** como su umbral potencial de discretización en relación con un gen g_i ya discretizado. Así, se genera una partición del conjunto de muestra de ***X*** en dos subgrupos, llamados ***Do*** y ***Up*** (por *Downregulated* y *Upregulated* respectivamente), para cada g_r y cada umbral candidato t. ***Do*** contiene todas las muestras en las que g_r tiene un valor de expresión menor o igual a t, mientras que ***Up*** contiene todas las muestras en las que g_r tiene un valor de expresión mayor que t. De ese modo, ***Do*** y ***Up*** representan una partición del conjunto de muestras de g_r en el que tiene valores iguales a -1 y 1, respectivamente, sobre la base de t. Aquí, t representa el umbral de regulación/discretización candidato para g_r en relación a g_i.

El siguiente paso consiste en realizar el cálculo de la entropía de partición, que es un indicador estadístico de la calidad de un umbral t como valor de discretización para g_r con respecto a g_i. Para ilustrar mejor este concepto, supongamos que estamos tratando de inferir los posibles reguladores de g_i (ya discretizado utilizando su TDT). Entonces, para cada g_r (potencial regulador de g_i), GRNCOP2 selecciona como su RRT, el umbral t que minimiza la entropía de

partición mediante el uso de la ecuación 7.4 que se muestra a continuación. En términos numéricos, la entropía de partición es 0 cuando todas las muestras cumplen con el mismo tipo de regla (situación ideal desde el punto de vista predictivo) y la entropía de partición es 1 cuando las muestras pertenecen a los dos escenarios de regulación en la misma proporción (50 por ciento y 50 por ciento). Entonces, cuando el valor de entropía de partición asociado con una discretización se aproxima a 0, el umbral que genera esta discretización representa una solución mejor. Por lo tanto, dicho umbral permite detectar de manera óptima posibles relaciones significativas entre g_i y g_r en términos del tipo de regla (ver tabla 7.1). El cálculo de la entropía se basa en las definiciones dadas en Mitchel (1997) y las entropías para ***Do*** y ***Up*** se basan en los valores discretizados de g_i obtenidos con su TDT correspondiente. La ecuación de la entropía de partición fue definida y utilizada previamente por Kohani (1995) de la siguiente manera:

$$PEntropy(g_r, t; X_S) = \frac{|Do|}{|X_S|} Entropy(Do) + \frac{|Up|}{|X_S|} Entropy(Up) . \quad (7.4)$$

donde:

- g_r identifica al gen bajo consideración (potencial regulador).
- t es el umbral de partición.
- $\boldsymbol{X_S}$ es el conjunto de muestras de g_r correspondiente a la serie de tiempo $\boldsymbol{X}$.
- ***Do*** es el subconjunto de $\boldsymbol{X}$ con las muestras donde el valor de expresión de g_r es menor o igual a t.
- ***Up*** es el subconjunto de $\boldsymbol{X}$ con las muestras donde el valor de expresión de g_r es mayor a t.

Después de ese paso, para cada posible g_i, la función δ ***(X, i)*** mapea los correspondientes valores de expresión de genes en $\boldsymbol{X}$ a la matriz discreta D_i utilizando los umbrales calculados anteriormente. De este modo, cada g_i en la matriz $\boldsymbol{X}$ original está asociado con una matriz discreta D_i. Sin embargo, a diferencia de GRNCOP y debido a razones de eficiencia, la matriz D_i no se calcula en GRNCOP2. En su lugar, cada elemento de D_i se calcula por demanda por medio de un acceso indirecto a la matriz $\boldsymbol{X}$, a través del umbral de regulación específico para g_r con respecto a g_i, en el tiempo de retardo w correspondiente y conjunto de datos k, mejorando así el tiempo computacional necesario para su ejecución.

7.1.2.4. Inferencia diferida en el tiempo

El proceso de inferencia diferida en el tiempo se representa como el bucle del medio en la figura 7.1. Simplemente consiste en la búsqueda de reglas en forma iterativa en cada retardo *w* empleando el proceso de inferencia de los clasificadores. Sin embargo, hay varios aspectos fundamentales de este proceso que son importantes destacar. Para cada matriz discreta D_i, hay que considerar el tiempo de retardo actual *w*, es decir, el desplazamiento temporal para el vector que codifica los valores de expresión de g_i. De este modo, con el fin de evaluar todos los posibles reguladores de g_i, es necesario remover los primeros *w* puntos de tiempo de g_i, los últimos *w* puntos de tiempo del resto de los genes (los que actuarán como posibles reguladores para el g_i) y, posteriormente, realinear las muestras. Así, a medida que el valor de *w* aumenta, menos son las muestras a partir de las cuales se pueden inferir las reglas. Esto limita el valor máximo de **W** a *m* - 4, siendo *m* el número de puntos de tiempo del conjunto de datos que tiene la menor cantidad de muestras. Esta limitación se debe a los TDTs empleados en este trabajo, ya que requieren un mínimo de 4 muestras para determinar los estados de g_i. En el caso extremo de que **W** se establezca en el máximo valor posible, la matriz discreta D_i tendrá sólo cuatro puntos de tiempo para ese conjunto de datos, lo que aumenta la posibilidad de inferir reglas por casualidad debido al bajo número de muestras. Sin embargo, es un requisito del investigador establecer el mejor valor para **W**, dependiendo de la cantidad de puntos de tiempo de los conjuntos de datos disponibles y en la probabilidad de que tales eventos puedan ocurrir en términos biológicos. En este sentido, supongamos que disponemos de **K** conjuntos de datos con m_k puntos de tiempo muestreados a un intervalo de tiempo Δt_k cada uno. De este modo, si la hipótesis con respecto a la naturaleza de los experimentos biológicos es que pueden ocurrir eventos de regulación con a lo sumo Δt_H unidades de retraso de tiempo, entonces **W** se puede calcular de la siguiente manera:

$$\mathbf{W} = \begin{cases} \left\lceil \Delta t_H \Big/ \min_{1 \le k \le \mathbf{K}} (\Delta t_k) \right\rceil & si \quad \left\lceil \Delta t_H \Big/ \min_{1 \le k \le \mathbf{K}} (\Delta t_k) \right\rceil \le \min_{1 \le k \le \mathbf{K}} (m_k) - 4 \\ \min_{1 \le k \le \mathbf{K}} (m_k) - 4 & si \quad caso \quad contrario \end{cases} . \qquad (7.5)$$

La ecuación anterior es muy simple y representa la relación entre la frecuencia del máximo retardo de regulación dado por hipótesis y el menor tiempo de muestreo de los conjuntos de datos, limitando el valor a la ventana de tiempo máxima posible para los **K** conjuntos de datos.

Adicionalmente, el proceso de discretización también debe considerar el tiempo de retardo. En el caso de la discretización de los genes blanco, los TDTs se calculan al principio del

algoritmo, para todos los retardos de tiempo **W** y para todos los conjuntos de datos **K**. Esto implica que los primeros *w* elementos de las muestras, en particular cuando $w \geq 1$, se omiten en el cálculo de los TDTs debido al desplazamiento temporal necesario para inferir las reglas diferidas en el tiempo. En el mismo sentido y en el caso de la política de discretización empleada para los potenciales genes reguladores, los RRT se calculan omitiendo los últimos *w* elementos de las muestras. Sin embargo, es necesario calcular los RRT al inicio del proceso de consenso de reglas con el fin de reducir la cantidad de espacio requerido para todas las combinaciones posibles de g_r, g_i, retardo de tiempo *w* y conjunto de datos *k*.

7.1.2.5. Proceso de inferencia de los clasificadores

Como fue definido en la ecuación 7.1, el problema de optimización consiste en encontrar un conjunto de $\overline{\pi}_i$ óptimos que definan reglas potenciales entre g_i y los otros genes (potenciales reguladores). Básicamente, $\overline{\pi}_i$ es un vector que representa el conjunto de los posibles reguladores de g_i. Cada componente *r* del vector tiene un valor entero entre -3 y 3, que representa uno de los siete casos regulatorios que se muestran en la tabla 7.1. Por lo tanto, $\overline{\pi}_i(r)$ indica el caso de regulación detectado entre g_r y g_i, en otras palabras, $\overline{\pi}_i$ es un clasificador de perfiles de genes que representa a los potenciales reguladores de g_i junto con las características de estas relaciones potenciales.

Teniendo en cuenta que las reglas inferidas por GRNCOP2 son cualitativas de a pares de genes, los componentes de $\overline{\pi}_i$ pueden asumirse como independientes entre sí desde el punto de vista de la optimización. Por lo tanto, el clasificador óptimo correspondiente a una matriz discreta D_i se puede calcular siguiendo una filosofía *greedy* mediante un enfoque constructivo, maximizando una función de rendimiento en cada componente $\overline{\pi}_i(r)$. En nuestro proceso de inferencia de los clasificadores, el proceso de optimización para $\overline{\pi}_i$, tal como se introdujo en la ecuación 7.1, se lleva a cabo de la siguiente manera:

$$\max_{\overline{\pi}_i \in P} \sigma^*\left(\overline{\pi}_i, D_i\right) \equiv \max_{c \in \{-3,-2,-1,1,2,3\}} \sigma(\overline{\pi}_i(r), D_i, c) \quad \forall r, \quad r = 1..n. \qquad (7.6)$$

donde:

- $c \in \{-3,-2,-1,1,2,3\}$ es uno de los casos regulatorios de la tabla 7.1.

Note que la definición de $\sigma^*(\overline{\pi}_i, D_i)$ no es necesaria debido a la suposición de independencia entre los componentes de $\overline{\pi}_i$. En este trabajo, se utiliza la siguiente función de rendimiento para optimizar la *r*-ésima componente de $\overline{\pi}_i$:

$$\sigma(\overline{\pi}_i(r), D_i, c) = \left(\frac{TP_c}{TP_c + FP_c} \right) \times \left(\frac{TN_c}{TN_c + FN_c} \right). \quad (7.7)$$

donde:

- ***TP**c* (Positivos verdaderos para el tipo de regla *c*) es el número de casos positivos de D_i correctamente clasificados por $\overline{\pi}_i(r)$ cuando se considera una regla de tipo *c*.
- ***FN**c* (Negativos falsos para el tipo de regla *c*) es el número de casos positivos de D_i clasificados incorrectamente por $\overline{\pi}_i(r)$ cuando se considera una regla de tipo *c*.
- ***TN**c* (Negativos verdaderos para el tipo de regla *c*) es el número de casos negativos de D_i correctamente clasificados por $\overline{\pi}_i(r)$ cuando se considera una regla de tipo *c*.
- ***FP**c* (Positivos falsos para el tipo de regla *c*) es el número de casos negativos de D_i clasificados incorrectamente por $\overline{\pi}_i(r)$ cuando se considera una regla de tipo *c*.

En la formulación anterior, el primer factor es el valor de predicción positiva, mientras que el segundo factor es el valor de predicción negativa. Ambos factores generan valores entre 0 y 1 y, en consecuencia, $\sigma(\overline{\pi}_i(r), D_i, c)$ está siempre en este rango. El mejor escenario para una interacción potencial entre g_i y g_r es cuando $\sigma(\overline{\pi}_i(r), D_i, c) = 1$, ya que representa la situación en la que todos los estados de expresión se clasifican correctamente, mientras que $\sigma(\overline{\pi}_i(r), D_i, c) = 0$ se refiere al caso opuesto. Se debe tener en cuenta que no se considera a $c = 0$ en la maximización de la función de rendimiento, ya que los valores de ***TP**c*, ***TN**c*, ***FP**c* y ***FN**c* no pueden ser determinados para ese caso. La principal diferencia entre esta función de rendimiento y la fórmula empleada en Ponzoni *et al.* (2007) es que la primera se centra en la precisión de las reglas definidas en la tabla 7.1, mientras que la empleada en Ponzoni *et al.* (2007) se focaliza en la especificidad y sensibilidad de tales reglas.

En la práctica, se establece un umbral (llamado parámetro *Precisión*) con el fin de retornar las reglas que alcancen una puntuación por encima de ese valor específico. Este valor actúa como punto de corte para los componentes, descartando aquellas reglas que no predicen bien de acuerdo con el valor máximo de la ecuación 7.7. Las reglas desechadas se consideran como reglas de tipo 0 de acuerdo a la tabla 7.1. Para los casos 1 y -1, la función de aptitud de la ecuación 7.7 se aplica como se indica, difiriendo únicamente en la forma en que se consideran los casos positivos y negativos. Para los casos 2, 3, -2 y -3, se emplea sólo uno de los factores de la función de aptitud (el que corresponde al tipo de regla) y, con el fin de evitar las reglas que superan al parámetro *Precisión* con pocos ***TP*** (***TN***) (en relación con el número de muestras), se define un parámetro adicional denominado porcentaje de cobertura de la muestra (*Sample*

Coverage Percentage o SCP). Este parámetro establece el porcentaje mínimo de ***TP*** (***TN***) que las reglas de los casos 2, 3, -2 y -3 necesitan lograr para no ser descartadas por el algoritmo. Ambos parámetros (*Precisión* y *SCP*) también se utilizan en GRNCOP. Dado que GRNCOP2 evalúa automáticamente las reglas en múltiples conjuntos de datos de *microarrays*, la precisión asignada a cada regla que supera al proceso de consenso es el valor mínimo alcanzado en todos los conjuntos de datos, es decir, el enfoque más conservador.

En cuanto a la mejor configuración para estos parámetros, varios autores (Soinov *et al.*, 2003; Bulashevska y Eils, 2005; Li *et al*., 2006; Ponzoni *et al*., 2007) consideran que las relaciones confiables entre genes son aquellas con una precisión superior al 0.70. En este sentido, y como regla general, un valor del parámetro *Precisión* de 0.75 debería ser suficiente para obtener relaciones regulatorias de alta calidad entre los genes en términos de la ecuación 7.7. En el caso del parámetro *SCP*, las reglas de los casos -2, 2, -3 y 3 son más probables de ser obtenidas por casualidad. De este modo, la única manera de garantizar reglas confiables para estos casos es estableciendo el valor del parámetro *SCP* cerca del valor máximo (es decir, 1). Si el parámetro *SCP* se establece en 1, entonces ninguna regla de estos casos es devuelta por el algoritmo.

Como se indicó anteriormente, GRNCOP2 calcula $\overline{\pi}_i$ utilizando el mismo enfoque constructivo empleado en GRNCOP (Ponzoni *et al.*, 2007), que explora todas las posibles combinaciones de valores de los componentes $\overline{\pi}_i(r)$. En resumen, GRNCOP2 calcula la función de desempeño definida en la ecuación 7.7 para cada caso posible de interacción (codificado por los valores que oscilan entre -3 y 3) y asigna el tipo de regla *c* que lo maximiza a $\overline{\pi}_i(r)$. Después de repetir esto para cada $\overline{\pi}_i(r)$, con $r = 1..n$, el $\overline{\pi}_i$ resultante es el clasificador óptimo de perfiles del gen g_i. De este modo, para un conjunto de datos de expresión de genes de *n* genes y *m* muestras, la complejidad computacional del proceso de inferencia de clasificadores es del $O(m.n^2)$ en el peor de los casos. Si se considera todo el algoritmo de inferencia, el tiempo requerido para inferir las reglas diferidas en el tiempo en **K** conjuntos de datos para **W** retardos de tiempo es del $O(\mathbf{K}.\mathbf{W}.m.n^2)$. Aunque a primera vista el orden de ejecución del algoritmo parece ser considerablemente alto en términos de complejidad computacional, se puede optimizar de manera eficiente a fin de realizar estudios a nivel de genoma completo, como se demostrará en las siguientes secciones.

7.1.3. Pruebas

Dos objetivos diferentes fueron divisados para el estudio de evaluación del nuevo algoritmo. En primer lugar, es importante analizar la calidad de los resultados de GRNCOP2 con respecto a la versión anterior (Ponzoni *et al.*, 2007) y con respecto a otros enfoques relacionados

disponibles en la literatura (Soinov *et al*., 2003; Bulashevska y Eils, 2005; Li *et al*., 2006). Para este análisis, GRNCOP2 fue probado usando los mismos 20 genes de levadura seleccionados por Soinov *et al*. (2003), Bulashevska y Eils (2005), Li *et al*. (2006) y Ponzoni *et al*. (2007), a partir de los datos de *microarrays* de Segal *et al.* (2003), con el fin de lograr una comparación justa. Sin embargo, a pesar de que la utilización de estos conjuntos de datos reducidos puede dar una visión adecuada del desempeño del método en comparación con otros, el problema de escalabilidad impone otro gran reto (Alves *et al.*, 2010). Para ello, en una segunda fase experimental se llevó a cabo la evaluación del desempeño de GRNCOP2 con un estudio a nivel de genoma completo para el organismo *Saccharomyces cerevisiae*.

7.1.3.1. Evaluación del rendimiento

A fin de medir la calidad de los resultados de un algoritmo de minería de reglas de genes, la técnica más frecuentemente utilizada es el análisis regla por regla de la relevancia biológica de las relaciones obtenidas. Esto se hace por medio de una búsqueda a través de la literatura, examinando interacciones biológicas conocidas entre los genes bajo consideración. Este enfoque resulta sano cuando se evalúa un único método, sin embargo, tiene inconvenientes que hacen casi imposible su aplicación en la mayoría de los escenarios. En primer lugar, sólo es aplicable cuando se evalúa un pequeño conjunto de reglas, ya que todo el proceso se lleva a cabo manualmente. Otra desventaja es que no puede ser utilizado para la comparación de varios métodos, ya que la calidad de una regla está sesgada por el experto que evalúa, y por lo tanto es imposible establecer un orden de mérito razonable para los algoritmos bajo consideración. No decimos que el uso de este proceso de evaluación para métodos de minería de reglas de genes sea inadecuado; solo nos limitamos a decir que tiene que ser utilizado como complemento de alguna otra técnica que permita la evaluación y la comparación de los diferentes enfoques de manera rápida, directa y objetiva.

En este contexto, varios análisis complejos de asociaciones entre genes están disponibles en diferentes bases de datos para la levadura (Stapley y Benoit, 2000; Jenssen *et al*., 2001; Dwight *et al.*, 2002; Lee *et al*., 2004; Lee *et al.*, 2007). Estos estudios pueden utilizarse para evaluar automáticamente la calidad de los resultados obtenidos por un algoritmo midiendo varias métricas de minería de datos bien conocidas, tales como precisión, sensibilidad y especificidad. En cuanto a Yeastnet v.2 (Lee *et al*., 2007), se reportaron 102.803 vínculos entre 5.483 genes de la levadura como posibles asociaciones entre pares de genes, asignando un valor de puntuación para cada asociación (con las asociaciones más fuertes obteniendo puntuaciones más altas). De la misma manera, las anotaciones de Ontologia de Genes (*Gene Ontology* o GO) (Dwight *et al.*, 2002) es otra fuente de posibles asociaciones de genes. En Lee *et al.*, (2007) se usa como

conjunto de referencia 66.174 pares de genes que representan a todos los pares de genes que comparten algún término de proceso biológico entre los niveles 2-10 de una anotación Ontológica de Genes (descargado de *Saccharomyces cerevisiae Genome Database* (SGD) (Dwight *et al.*, 2002)). Además, el enfoque de co-citación (Stapley y Benoit, 2000; Jenssen *et al.*, 2001; Lee *et al*., 2004) ofrece otra fuente de información independiente para analizar los resultados de los algoritmos de minería de reglas de genes. En este caso, en Lee *et al*., (2004) se analizó un conjunto de 29.135 resúmenes *Medline* que incluían la palabra "*Saccharomyces cerevisiae*" para coincidencias perfectas a cualesquiera de los nombres normalizados o nombres comunes (o sus sinónimos) de 5.794 genes de levadura. Es ese trabajo se reportó un conjunto de 29.483 pares de genes con un valor de puntuación para cada asociación (con las asociaciones más fuertes obteniendo puntuaciones más altas).

Por lo tanto, la idea principal para el marco de evaluación de los métodos consiste en medir la precisión, la sensibilidad y la especificidad alcanzada en relación con cada uno de los estudios mencionados anteriormente. Además, en el caso de Lee *et al*. (2007) y Lee *et al.* (2004), se evalúa también la *puntuación* medida como el promedio de los valores de puntuación de las reglas encontradas por un método. Sin embargo, ya que este tipo de información no tiene en cuenta el tiempo de retardo en las reglas inferidas por los métodos o la dirección de la interacción real (ninguno de los genes están anotados como regulador o blanco), se debe introducir una convención a fin de hacer una comparación equitativa entre los diferentes algoritmos. En este sentido, los resultados de un algoritmo se transformarán a fin de representar el mismo tipo de información de los conjuntos de referencia. En otras palabras, sólo el conjunto de interacciones gen-gen será considerado para la medición, dejando de lado las nociones de tiempo retardo y regulador-blanco de las reglas que se infieren. Esto evita la validación repetida de múltiples reglas (debido a los diferentes tiempos de retraso o enlaces simétricos) por la misma asociación en los conjuntos de referencia, una situación que podría producir una comparación injusta. No obstante, en la comparación entre los algoritmos se considerarán solo intervalos temporales de inferencia iguales.

7.1.3.2. Estudio comparativo

En esta sección, GRNCOP2 será comparado con algunos métodos representativos de aprendizaje automático que están presentes en la literatura. La eficacia predictiva fue probada usando los datos de *microarrays* publicados en Segal *et al.* (2003), basados en los datos de Spellman *et al.* (1998), y que también incluyen datos de cultivos de células de *Saccharomyces cerevisiae* (Yeang y Jaakkola, 2003). Estos conjuntos de datos se sincronizaron con tres métodos diferentes: cdc15, cdc28 y factor-alfa, y se tomaron muestras a intervalos de 10 minutos, 10

minutos y 7 minutos, respectivamente. Por lo tanto, los correspondientes conjuntos de datos de expresión génica pueden ser considerados como estadísticamente independientes (van Someren *et al.*, 2000). Para el análisis realizado en esta sección, fueron utilizados los siguientes 20 genes a fin de estar en concordancia con los estudios de Soinov *et al.* (2003), Bulashevska *et al.* (2005), Li *et al.* (2006) y Ponzoni *et al.* (2007): *CLN1-3*, *CLB1-2*, *CLB4-6*, *MCM1*, *SIC1*, *CDC28*, *CDC53*, *MBP1*, *CDC34*, *SWI4-6*, *SKP1*, *CDC20* y *HCT1*. Sólo se consideraron las mediciones equidistantes adyacentes en las mismas unidades de tiempo con el objetivo de facilitar la interpretación de las reglas diferidas en el tiempo. Así, varios puntos de tiempo para el conjunto de datos cdc15 fueron truncados resultando en un total de 15, 17 y 18 puntos de tiempo disponibles para los datos cdc15, cdc28 y factor-alfa respectivamente.

En este estudio comparativo, la precisión, especificidad y sensibilidad se calcularon en relación al espacio de búsqueda reducido determinado por los 20 genes. La tabla 7.2 muestra las características de este espacio de búsqueda, que consiste en 190 posibles interacciones gen-gen (que no consideran las reglas diferidas en el tiempo y los vínculos simétricos con el fin de coincidir con el tipo de información de conjuntos de referencia). Es importante tener en cuenta que, en este caso, la precisión de las 190 combinaciones posibles de pares de genes en los estudios de referencia determina la probabilidad de seleccionar aleatoriamente un par de genes válidos para estos conjuntos. En otras palabras, los valores de precisión y de puntuación de las columnas de la tabla 7.2 serían los valores esperados si se seleccionaran conjuntos aleatorios (distribuidos uniformemente) de pares de genes. De este modo, es muy importante que los algoritmos obtengan valores por encima de estos números.

Tabla 7.2. Características de las 190 posibles interacciones gen-gen.

	Yeastnet		**Co-citation**		**GO**	numero de posibles associaciones
	precisión	*puntuación*	*precisión*	*puntuación*	*precisión*	
Todas las combinaciones gen-gen	51.58%	1.53033843	43.68%	1.3487118	45.26%	190

Debido a las diferencias en la disponibilidad de los métodos y en los resultados reportados por Soinov *et al.* (2003), Bulashevska *et al.* (2005), Li *et al.* (2006) y Ponzoni *et al.* (2007), este análisis se lleva a cabo en tres etapas diferentes. En la subsección A, se presenta una evaluación de las mejoras de GNRCOP2 con respecto a la versión anterior. La subsección B corresponde a la comparación con los otros métodos seleccionados disponibles en la literatura (Soinov *et al.*, 2003; Bulashevska *et al.*, 2005; Li *et al.*, 2006). Por último, en el apartado C, se discute la importancia biológica de las reglas encontradas por GNRCOP2 para esta instancia reducida del problema.

A. GRNCOP2 versus GRNCOP

El análisis de las mejoras de GRNCOP2 sobre GRNCOP (Ponzoni *et al.*, 2007) se realizó para los 20 genes de la levadura previamente mencionados, en los conjuntos de datos cdc15, cdc28 y factor-alfa. A fin de realizar una comparación justa, ambos algoritmos emplearon el proceso de consenso de reglas descrito anteriormente con el *RCA* establecido en 1. De esta manera, se realizaron varias corridas de cada algoritmo variando el parámetro *Precisión* entre 0.60 y 0.90 con incrementos de 0.05, y el parámetro *SCP* entre 0.60 y 0.95 con incrementos de 0.05. Esto resultó en un total de 56 ejecuciones para cada método, midiendo al conjunto de asociaciones obtenidas en cada ejecución en términos de las métricas de *precisión*, *sensibilidad*, *especificidad* y *puntuación* previamente definidas. El valor del parámetro *Precisión* en 0,95 fue omitido ya que ambos algoritmos fueron incapaces de encontrar alguna regla con esta configuración. También es importante señalar que el foco fue puesto en reglas simultáneas y con un retardo de tiempo de una unidad (es decir, **W** = 1 en el caso de GRNCOP2), ya GRNCOP no fue diseñado para buscar reglas con múltiples retrasos de tiempo (Ponzoni *et al.*, 2007). Los resultados promedio de las 56 ejecuciones en términos de *precisión*, *especificidad* y *sensibilidad* en Yeasnet (Lee *et al.*, 2007), GO (Dwight *et al.*, 2002) y Co- citación (Lee *et al.*, 2004) se muestran en la tabla 7.3, junto con la *puntuación* promedio en el caso de Lee *et al.* (2004) y Lee *et al.* (2007).

Tabla 7.3. *Precisión*, *sensibilidad*, *especificidad* y *puntuación* promedio obtenidas por GRNCOP2 y GRNCOP para las 56 ejecuciones. La *precisión* y la *puntuación* de la selección aleatoria también están incluidas. Las puntuaciones en negrita denotan los mejores valores.

		GRNCOP2	GRNCOP	ALEATORIO
Yeastnet	*precisión* promedio	**84.50%**	76.69%	51.58%
	sensibilidad promedio	16.25%	**28.13%**	-
	especificidad promedio	**94.66%**	82.43%	-
	puntuación promedio	**2.79**	2.49	1.53033843
Co-citación	*precisión* promedio	**84.13%**	74.86%	43.68%
	sensibilidad promedio	19.02%	**30.46%**	-
	especificidad promedio	**95.28%**	82.76%	-
	puntuación promedio	**2.91**	2.50	1.3487118
GO	*precisión* promedio	**70.73%**	52.25%	45.26%
	sensibilidad promedio	13.93%	**22.55%**	-
	especificidad promedio	**91.48%**	76.60%	-
número promedio de asociaciones		20.84	**43.73**	-

Como se puede observar, GRNCOP2 supera (en promedio) a GRNCOP en varias de las métricas propuestas, mientras que ambos algoritmos obtienen valores significativamente por encima de la selección al azar, como se esperaba. En particular, aunque GRNCOP2 es, en promedio, más preciso y más específico que GRNCOP, este último recupera en promedio un mayor número de las "interacciones relevantes" (es decir, es más sensible). Estos resultados pueden explicarse por el hecho de que en realidad GRNCOP recupera en promedio el doble de la cantidad de las asociaciones obtenidas por GRNCOP2. Sin embargo, ya que los valores de la tabla 7.3 representan el promedio de las 56 ejecuciones, la imagen real puede ser interpretada de forma errónea. Por lo tanto, con el fin de establecer correctamente el comportamiento de cada algoritmo, se realizaron varios gráficos. Las figuras 7.2a a 7.2e representan las métricas de *precisión* y *puntuación* obtenidas por ambos algoritmos en cada una de las 56 ejecuciones con respecto al porcentaje de cobertura del espacio combinatorial de búsqueda (*Coverage Percentage of the Combinatorial Search Space* o *CP- CSS*), es decir, el porcentaje de asociaciones devueltas por los métodos en relación a todas las posibles combinaciones de genes por pares (véase la tabla 7.2).

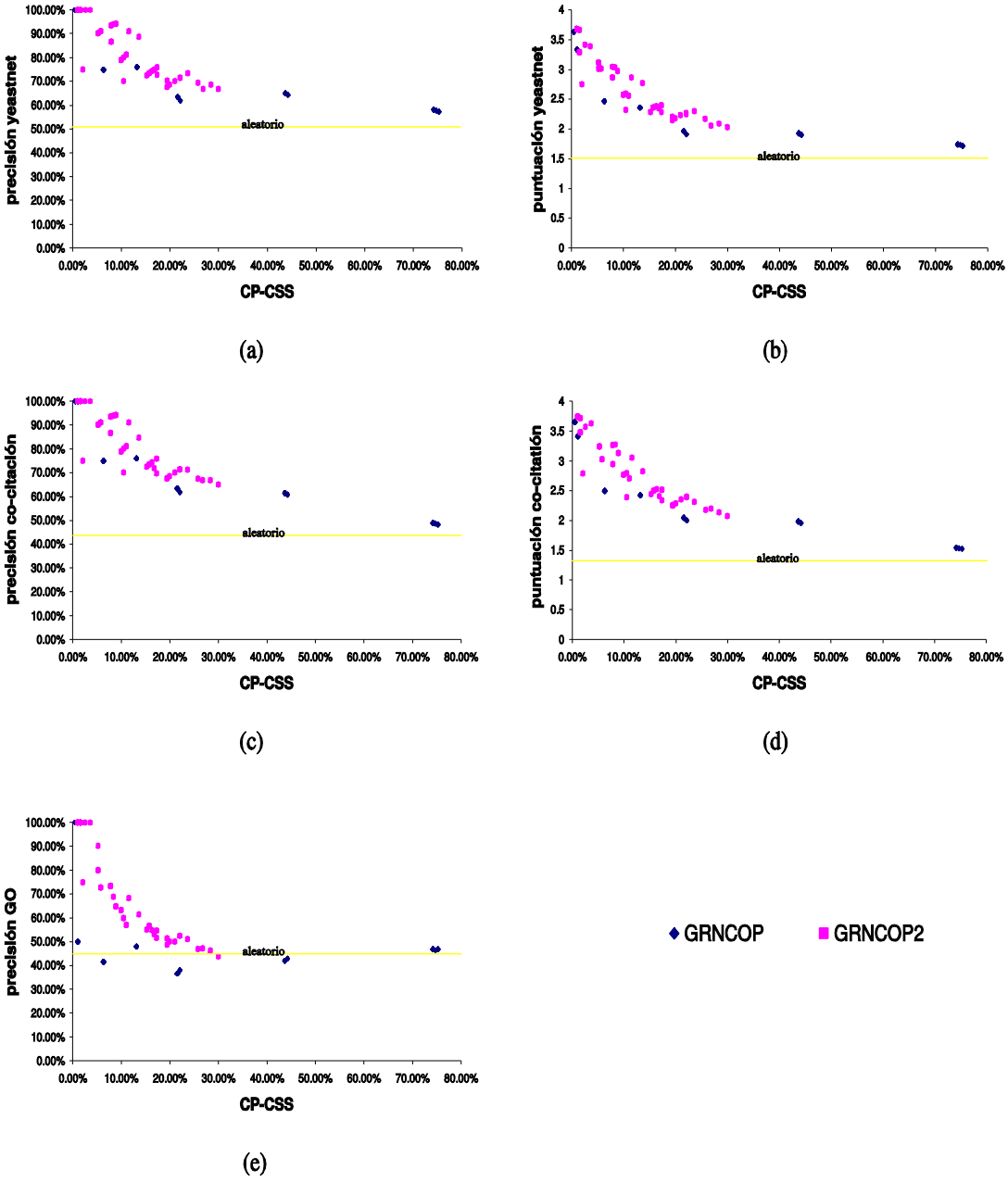


Figura 7.2. Valores de las métricas *precisión* y *puntuación* alcanzados por GRNCOP2 y GRNCOP en cada una de las 56 ejecuciones con respecto al *CP-CSS*. Fig. 7.2a: *precisión* yeastnet. Fig. 7.2b: *puntuación* yeastnet. Fig. 7.2c: *precisión* co-citación. Fig. 7.2d: *puntuación* co-citación. Fig. 7.2e: *precisión* GO.

Se pueden deducir tres observaciones importantes de estas figuras. La primera es que, en varios casos, especialmente en valores bajos del parámetro *Precisión*, GRNCOP devuelve casi el 80 % de todas las posibles combinaciones de pares de genes. De esta manera, su rendimiento disminuye casi al nivel de una selección aleatoria debido a la cantidad excesivamente grande de asociaciones encontradas. Esto explica los valores altos en la sensibilidad promedio y número de asociaciones promedio mostrados en la tabla 7.3. Es más, este comportamiento no es deseable en lo absoluto, ya que puede limitar su aplicabilidad en contextos de genoma completo, donde el número de posibles combinaciones de asociaciones alcanza muy alta dimensionalidad. La

segunda observación, pero no menos importante, es que para la misma cantidad de asociaciones devueltas por los algoritmos, las interacciones encontradas por GRNCOP2 parecen ser (en general) más precisas y con puntuaciones más altas que las encontradas por GRNCOP. Esto es particularmente relevante, ya que este comportamiento evidencia las mejoras conseguidas por las modificaciones incluidas en el algoritmo de inferencia anteriormente detalladas. La tercera observación tiene que ver con las diferentes formas en la distribución de los puntos en las figuras de ambos métodos. Junto con el elevado número de asociaciones mencionado anteriormente, parece que GRNCOP tiene menos variación en los valores de *precisión* y de *puntuación* obtenidos en relación a los logrados por GRNCOP2. Sin embargo, esto puede ser explicado por el hecho de que GRNCOP es casi insensible a las variaciones de su parámetro *SCP* en los valores empleados en esta comparación. Esto está relacionado con el valor de expresión promedio empleado por GRNCOP para la discretización de los genes blanco. En general, un valor promedio del perfil de expresión de un gen tenderá a dividir las muestras en dos particiones de aproximadamente el mismo tamaño (excepto en la presencia de pocas muestras con elevado valor absoluto relativo a las otras muestras). Por lo tanto, sólo un pequeño número de reglas de los casos -3, -2, 2 y 3 van a satisfacer el umbral de *SCP*, en otras palabras, GRNCOP requiere valores incluso inferiores del parámetro *SCP* para obtener reglas de estos tipos. Es más, esta situación aumenta la probabilidad de encontrar estos tipos de reglas por casualidad, dado que hay un menor número de muestras para el proceso de inferencia. Sin embargo, estas observaciones no invalidan las conclusiones con respecto a las mejoras de GRNCOP2 sobre GRNCOP, ya que se ha observado que, a valores más bajos del parámetro *SCP*, ambos algoritmos tienden a rendir peor en términos de los indicadores propuestos.

Por último, también es importante analizar el comportamiento de los dos algoritmos en relación con las métricas de *sensibilidad* y *especificidad*. Las figuras 7.3a a 7.3c muestran la *sensibilidad* frente a la *especificidad* de los algoritmos en relación a los tres conjuntos de referencia para las 56 ejecuciones. Es fácil observar que, en general, GRNCOP2 es superior a GRNCOP, ya que en los mismos niveles de *sensibilidad* (*especificidad*), la *especificidad* (*sensibilidad*) alcanzada por el primero es más alta. Por otra parte, aunque GRNCOP es capaz de recuperar casi el 80% de las asociaciones pertinentes en todos los casos, esto es debido a la gran cantidad de interacciones devueltas por el algoritmo, como se discutió anteriormente. Por lo tanto, los resultados muestran que, en este caso de estudio, GRNCOP2 se comporta mejor que GRNCOP, y que las modificaciones propuestas en la nueva metodología realmente mejoran el proceso de inferencia ya que los resultados parecen ser más relevante en términos de las métricas de *precisión*, *sensibilidad*, *especificidad* y de *puntuación*.

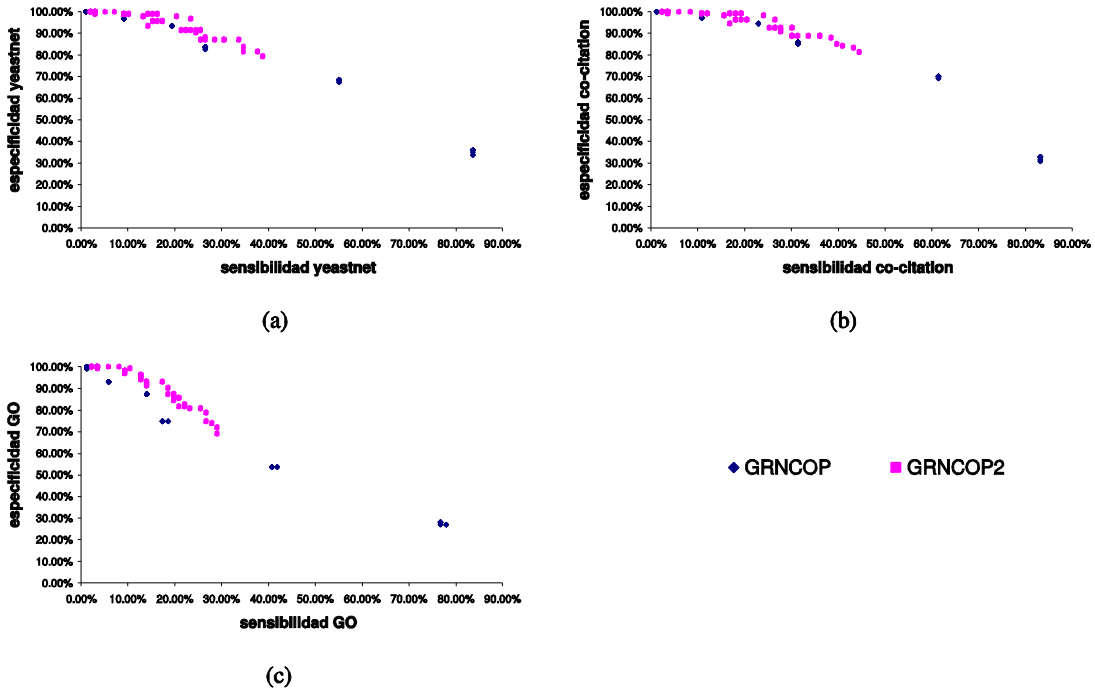


Figura 7.3. Valores se *sensibilidad* y *especificidad* alcanzados por GRNCOP2 y GRNCOP en los tres conjuntos de referencia para las 56 ejecuciones. Fig. 7.3a: *sensibilidad* versus *especificidad* respecto del conjunto yeastnet. Fig. 7.3b: *sensibilidad* versus *especificidad* respecto del conjunto co-citación. Fig. 7.3c: *sensibilidad* versus *especificidad* respecto del conjunto GO.

B. GRNCOP2 versus otros métodos basados en reglas de asociación

En esta sección, el desempeño de GRNCOP2 se comparara con otros algoritmos descritos en la literatura en términos de las métricas propuestas. Esta comparación se limita conforme a los resultados reportados por Soinov *et al.* (2003), Bulashevska y Eils (2005) y Li *et al*. (2006) debido a la falta de disponibilidad de los algoritmos. Para hacer una comparación justa con estos métodos de acuerdo al proceso de consenso de reglas de GRNCOP2 con el parámetro *RCA* = 1, las reglas encontradas por los tres enfoques se filtraron en concordancia con la precisión informada. De esta manera, sólo las reglas que alcanzaron una precisión de al menos 0.75 en los tres conjuntos de datos (cdc15, cdc28 y factor alfa) se seleccionaron para este estudio. En el caso de Bulashevska y Eils (2005) y Soinov *et al.* (2003), la comparación se lleva a cabo sólo con las reglas simultáneas, ya que las reglas con retardo de una unidad, en Bulashevska y Eils (2005) se validaron con un solo conjunto de datos (cdc15), y en Soinov *et al.* (2003) los valores de precisión no se informan para esas reglas. Por lo tanto, la tabla 7.4 muestra los resultados de las métricas propuestas para las reglas simultáneas de Bulashevska y Eils (2005), Soinov *et al.* (2003) y GRNCOP2 ejecutado en los tres conjuntos de datos con una *Precisión* de 0.75, un *SCP* de 0.95, un *RCA* de 1 y con **W** = 0. La tabla 7.5 muestra los resultados obtenidos con las métricas propuestas para las reglas con retrasos de tiempo del 1 al 5 de Li *et al*. (2006), y para GRNCOP2 con la misma parametrización anterior, excepto para **W** que se estableció en 5 y, a

continuación, se eliminaron las reglas simultáneas a fin de hacer la comparación. Esta parametrización se ha establecido como sigue: **W** = 0 denota que GRNCOP2 sólo llevará a cabo la búsqueda de las reglas simultáneas, en el caso de **W** = 5, denota que la búsqueda se llevará a cabo en cinco unidades de retardo de tiempo, incluyendo reglas simultáneas; *RCA* = 1 dice que las reglas deben "predecir bien" en todos los conjuntos de datos; *SCP* = 0.95 intenta obtener reglas de los casos -3, -2, 2 y 3 con altas tasas de **TP** (**TN**), y la *Precisión* = 0.75 intenta representar el mismo nivel de exactitud de los otros métodos, aunque esto no es necesariamente cierto debido a los diferentes criterios empleados en cada algoritmo para la evaluación de las reglas. Tenga en cuenta que los valores de las métricas de la tabla 7.4 y la tabla 7.5 se calcularon teniendo en cuenta sólo los conjuntos de interacciones gen a gen, como se explicó anteriormente.

Tabla 7.4. *Precisión*, *sensibilidad*, *especificidad* y *puntuación* promedio obtenidas por GRNCOP2, Soinov *el al.* y, Bulashevska y Eils para reglas simultaneas con una *Precisión* = 0.75. La *precisión* y la *puntuación* de la selección aleatoria también están incluidas. Las puntuaciones en negrita denotan los mejores valores.

		GRNCOP2	Soinov *et al.*	Bulashevska and Eils	ALEATORIO
Yeastnet	*precisión*	**93.33%**	50.00%	88.89%	51.58%
	sensibilidad	**14.29%**	3.06%	8.16%	-
	especificidad	**98.91%**	96.74%	**98.91%**	-
	puntuación	**3.04**	1.84	2.77	1.530338427
Co-citation	*precisión*	**93.33%**	50.00%	88.89%	43.68%
	sensibilidad	**16.87%**	3.61%	9.64%	-
	especificidad	**99.07%**	97.20%	**99.07%**	-
	puntuación	**3.26**	1.85	2.84	1.348711804
GO	*precisión*	**73.33%**	50.00%	55.56%	45.26%
	sensibilidad	**12.79%**	3.49%	5.81%	-
	especificidad	96.15%	**97.12%**	96.15%	-
número de asociaciones		**15.00**	6.00	9.00	-

Tabla 7.5. *Precisión*, *sensibilidad*, *especificidad* y *puntuación* promedio obtenidas por GRNCOP2 y Li *et al*. para reglas con retardo de tiempo de 1 a 5 unidades y con una *Precisión* = 0.75. La *precisión* y la *puntuación* de la selección aleatoria también están incluidas. Las puntuaciones en negrita denotan los mejores valores.

		GRNCOP2	Li *et al.*	ALEATORIO
Yeastnet	*precisión*	**100.00%**	**100.00%**	51.58%
	sensibilidad	**10.20%**	5.10%	-
	especificidad	**100.00%**	**100.00%**	-
	puntuación	**3.03**	2.89	1.530338427
Co-citation	*precisión*	**100.00%**	**100.00%**	43.68%
	sensibilidad	**12.05%**	6.02%	-
	especificidad	**100.00%**	**100.00%**	-
	puntuación	**3.37**	2.99	1.348711804
GO	*precisión*	**90.00%**	80.00%	45.26%
	sensibilidad	**10.47%**	4.65%	-
	especificidad	**99.04%**	**99.04%**	-
número de asociaciones		**10.00**	5.00	-

Como se puede observar, GRNCOP2 se desempeña igual o mejor en este nivel de *Precisión* en casi todas las métricas propuestas. Las diferencias con respecto a los métodos de referencia

son más evidentes en el caso de las reglas simultáneas (véase la tabla 7.4) que en el caso de las reglas con retardados de tiempo (véase la tabla 7.5). Sin embargo, en este último escenario, GRNCOP2 es capaz de recuperar el doble de interacciones relevantes (ver los valores de *sensibilidad*) que Li *et al.* (2006) en el mismo nivel de *precisión*. Aunque estos resultados no son concluyentes en la determinación del mejor método, ya que se limitan a un solo caso de estudio en un solo nivel de Precisión, proporcionan información con respecto al desempeño real del enfoque propuesto. En este sentido, estas observaciones indican claramente que GRNCOP2 es un método capaz de inferir interacciones importantes con altos niveles de *precisión* que otros métodos de la literatura no son capaces de encontrar.

C. Relevancia biológica de los resultados

Las reglas obtenidas por GRNCOP2 con *Precisión* = 0.75, *SCP* = 0.95, *RCA* = 1 y **W** = 5, para los conjuntos de datos cdc15, cdc28 y factor alfa, se resumen en la tabla 7.6. Los resultados representados en esta tabla son, de hecho, los mismos que los empleados en el estudio comparativo de la subsección anterior. La única diferencia radica en que, en este caso, el tiempo de retardo y los enlaces simétricos de las interacciones se mantienen, dado que no se lleva a cabo la evaluación con los conjuntos de referencia. Las tres últimas columnas indican relaciones de interacción que también fueron inferidas por los otros métodos, utilizando los mismos tres conjuntos de datos y el mismo nivel de *Precisión*, como se ha descrito anteriormente. Sólo reglas de los casos 1 y -1 fueron inferidas debido al alto valor de *SCP* empleado. Es necesario tener en cuenta que múltiples retrasos de tiempo permiten el descubrimiento de interacciones adicionales que no son visibles en GRNCOP. Además, ninguna de las reglas diferidas en el tiempo encontradas por GRNCOP2 fueron encontradas por Li *et al*. (2006), omitiendo por lo tanto la columna correspondiente en la tabla 7.6. Este hecho podría estar relacionado con los diferentes procesos de discretization, ya que en Li *et al.* (2006) se empleó un umbral fijo (cero) para determinar los estados de todos los genes en todos los retrasos de tiempo.

Tabla 7.6. Reglas inferidas por GRNCOP2 usando los 20 genes *cyclins* de los conjuntos de datos de Segal *et al.* (2003). Las ultimas tres columnas indican si las reglas fueron encontradas por alguno de los otros métodos. Las reglas que fueron encontradas completas por los otros métodos están representadas por *; + para los casos en que fueron encontradas solo en la regulación positiva y - para el caso en que fueron encontradas solo en la regulación negativa.

Rules					Rule found by		
					GRNCOP	Soinov et. al.	Bulashevska and Eils
+/-	*CLB1*	*0* →	+/-	*CLB2*	*	-	+
+/-	*CLB1*	*3* →	+/-	*CLB5*			
+/-	*CLB1*	*3* →	+/-	*CLB6*			
+/-	*CLB1*	*0* →	-/+	*CLN2*	*		
+/-	*CLB1*	*0* →	+/-	*SWI5*	*		
+/-	*CLB2*	*0* →	+/-	*CDC20*			
+/-	*CLB2*	*0* →	+/-	*CLB1*	*	*	+
+/-	*CLB2*	*3* →	+/-	*CLB5*			
+/-	*CLB2*	*3* →	+/-	*CLB6*			
+/-	*CLB2*	*0* →	-/+	*CLN2*			
+/-	*CLB2*	*0* →	+/-	*SWI5*	*		
+/-	*CLB5*	*4* →	+/-	*CLB2*			
+/-	*CLB5*	*0* →	+/-	*CLB6*	*	+	
+/-	*CLB5*	*3* →	-/+	*CLB6*			
+/-	*CLB6*	*0* →	-/+	*CLB1*			+
+/-	*CLB6*	*0* →	+/-	*CLB5*	*	+	
+/-	*CLB6*	*3* →	-/+	*CLB5*			
+/-	*CLB6*	*0* →	+/-	*CLN2*	*		
+/-	*CLB6*	*1* →	+/-	*CLN2*			
+/-	*CLN1*	*0* →	+/-	*CLB6*	*		
+/-	*CLN1*	*2* →	-/+	*CLB6*			
+/-	*CLN1*	*0* →	+/-	*CLN2*	*		
+/-	*CLN2*	*4* →	+/-	*CLB2*			
+/-	*CLN2*	*0* →	+/-	*CLB5*	*		
+/-	*CLN2*	*0* →	+/-	*CLB6*	*		
+/-	*CLN2*	*2* →	-/+	*CLB6*			
+/-	*CLN2*	*3* →	-/+	*CLB6*			
+/-	*SIC1*	*0* →	+/-	*CLB5*			
+/-	*SWI4*	*0* →	+/-	*CLB5*			
+/-	*SWI4*	*4* →	-/+	*CLB5*			
+/-	*SWI5*	*0* →	+/-	*CDC20*	*		
+/-	*SWI5*	*0* →	+/-	*CLB2*	*		+
+/-	*SWI5*	*3* →	+/-	*CLB6*			

La relevancia biológica de las reglas inferidas se estimó mediante el análisis de si tales relaciones reflejan propiedades funcionales clave relativas a las diferentes fases del ciclo celular: G1, S, G2, M, M/G1. Los genes *CLN1* y *CLN2* transcriben los G1-ciclines, mientras que *CLB5* y *CLB6* transcriben los B-ciclines. Ellos comparten un patrón de expresión similar y alcanzan sus niveles de expresión más altos durante la fase G1, que se puede verificar en los datos experimentales analizados en Kuhne y Linder (1993), Chen *et al*. (2000) y Hwang *et al*. (1998). Este conocimiento es consistente con las reglas: *+/-CLB6 0→ +/-CLB5*, *+/-CLB6 0→ +/-CLN2*, *+/-CLN2 0→ +/-CLB5*, *+/-CLB5 0→ +/-CLB6*, *+/-CLN1 0→ +/-CLB6*, *+/-CLN2 0→ +/-CLB6*, *+/-CLN1 0→ +/- CLN2* y *+/- CLB6 1→ +/-CLN2*. Estas reglas también son coherentes con algunas observaciones sobre la redundancia funcional parcial existente entre *CLB5*, *CLN1* y *CLN2*, que ha sido informada por Epstein y Cross (1992) y Levine *et al*. (1996). En particular, la relación diferida en una sola unidad de tiempo indicada por la regla *+/-CLB6 1→ +/-CLN2*, detectada sólo por GRNCOP2, puede ser explicada en términos de la progresión de las concentraciones de ARNm de los genes *CLB5*, *CLB6* y *CLN2* al comienzo del ciclo celular de la

levadura, tal como se detalla en el modelo molecular de la levadura presentado por Chen *et al*. (2000).

CLB1 y *CLB2* son ciclines específicos de la fase G2, y existe evidencia biológica de que están co-expresados en este proceso (Althoefer *et al.*, 1995). El gen *SWI5* es un factor de transcripción cuya activación se produce durante la fase G2. Estos hechos justifican las siguientes reglas: +/-*CLB2* $0\rightarrow$ +/-*CLB1*, +/-*CLB1* $0\rightarrow$ +/-*CLB2*, +/-*CLB1* $0\rightarrow$ +/-*SWI5*, +/-*CLB2* $0\rightarrow$ +/-*SWI5*, +/-*SWI5* $0\rightarrow$ +/-*CLB2*, que están soportadas adicionalmente por la evidencia biológica presentada por Koranda *et al*. (2000). En particular, la regla +/-*SWI5* $0\rightarrow$ +/-*CLB2* sólo fue descubierta por el algoritmo GRNCOP2. Además, la transcripción de *SWI5* se activa más tarde en la fase S, y su pico de concentración de ARNm se produce durante la fase G2 (Loy *et al*., 1999); mientras que *CLB6* está activo en la fase G1 del ciclo celular. Esta información es consistente con la regla diferida en el tiempo: +/-*SWI5* $3\rightarrow$ +/-*CLB6*.

También es bien sabido que en la levadura en gemación, los G1-ciclines, tales como *CLN1* y *CLN2*, se expresan en las fases G1 y S, mientras que los ciclines mitóticos como *CLB1* y *CLB2* se expresan en las fases G2 y M. En Amon *et al*. (1993) encontraron que las *CLBs* juegan un papel central en la transición de la fase S a la fase G2, mostrando evidencia de que los *CLBs* reprimen a los *CLNs*. Esta regulación negativa de los *CLNs* puede ocurrir a través del factor de transcripción *SWI4*, porque los *CLBs* son necesarios para la represión en G2 de los genes *SCB*-regulados como *CLN1* y *CLN2*. Por otra parte, Andrews y Measday (1998) presentan evidencia de que los complejos Ciclin/*CDK* (*CDC28*/*CLN1* y *CDC28*/*CLN2*) regulan la proteólisis *CLB*. Estos datos son consistentes con las relaciones inhibidoras inferidas entre los genes específicos a G1 y a G2: +/-*CLB1* $0\rightarrow$ -/+*CLN2*, +/-*CLB6* $0\rightarrow$ -/+*CLB1* y +/-*CLB2* $0\rightarrow$ -/+*CLN2*. En particular, la última regla sólo es inferida por GRNCOP2. Se remite al lector a Althoefer *et al.* (1995), Loy *et al.* (1999) y Schneider *et al.* (1998) para obtener más información sobre la importancia biológica de estas asociaciones.

Con respecto a *SIC1*, es bien sabido que este gen es un inhibidor de los complejos *CLB*, y que está activo durante la fase G1 - junto con *CLB5* y *CLB6* - inhibiendo a *CLB1* y a *CLB2* (Toyn *et al.*, 1997). Este conocimiento valida la nueva regla +/-*SIC1* $0\rightarrow$ +/-*CLB5* inferida por GRNCOP2. *CDC20* y *SWI5* se transcriben más tarde en la fase S/G2 (Hwang *et al.*, 1998), lo que explica la asociación representada por la regla +/-*SWI5* $0\rightarrow$ +/-*CDC20*. Esta regla no fue detectada por los métodos comparados con GRNCOP2. Printz *et al.* (1998) presentó pruebas de que *CLB2* estimula la síntesis de *CDC20*. Esta característica es capturada por la regla +/-*CLB2* $0\rightarrow$ +/-*CDC20*.

La proteína *SWI4* es un componente del complejo SBF, que controla la expresión de genes durante la fase G1 (Igual *et al.*, 1997). Esto está de acuerdo con el papel de activador de *SWI4*

en los genes expresados en la fase G1, como se representa por la regla: +/-*SWI4 0*→ +/-*CLB5*. Estas observaciones ofrecen evidencia de la importancia biológica de las reglas de asociación inferidas por GRNCOP2.

Por último, el comportamiento opuesto entre los genes específicos de G1 y de G2 - como se evidencia por las reglas obtenidas a partir del análisis de los puntos de tiempo simultáneos de los datos de *microarrays* - se convierten en patrones de activación similares cuando se considera algún tiempo de retardo, como consecuencia de la comparación de patrones en diferentes fases celulares. En otras palabras, si GRNCOP2 relaciona el comportamiento de un gen G1-ciclin en la fase G1 con el comportamiento de un gen G2-ciclin en la fase G2 se infiere una correlación positiva. Este es el caso de las siguientes reglas: +/-*CLB1 3*→ +/-*CLB5*, +/-*CLB1 3*→ +/-*CLB6*, +/-*CLB2 3*→ +/-*CLB5*, +/-*CLB2 3*→ +/-*CLB6*, +/-*CLB5 4*→ +/-*CLB2* y +/-*CLN2 4*→ +/-*CLB2*. De manera similar, cuando GRNCOP2 compara los patrones de activación de genes con niveles de expresión altos durante la fase G1 en contraste con el patrón de expresión de estos mismos genes durante la fase G2, algunas relaciones opuestas y lógicas pueden surgir: +/-*CLB5 3*→ -/+*CLB6*, +/-*CLB6 3*→ -/+*CLB5*, +/-*CLN1 2*→ -/+*CLB6*, +/-*CLN2 2*→ -/+*CLB6*, +/-*CLN2 3*→ -/+*CLB6* and +/-*SWI4 4*→ -/+*CLB5*. Tomemos por ejemplo la regla +/-*SWI4 4*→ -/+*CLB5* que tiene una interacción contradictoria con la regla +/-*SWI4 0*→ +/-*CLB5*. La figura 7.4 muestra los perfiles de expresión reales y discretizados de los dos genes con 0 (izquierda) y 4 (derecha) unidades de retardo de tiempo para el conjunto de datos cdc15. Como se puede observar, las dos reglas son perfectamente válidas desde el punto de vista algorítmico y, a priori, igualmente probables en términos biológicos. Por lo tanto, en tales casos de interacciones contradictorias, se requiere un análisis más profundo a fin de establecer la relación real entre los genes. No obstante, es importante tener en cuenta que la inferencia de estas interacciones contradictorias diferidas en el tiempo pueden ayudar en el análisis del patrón de comportamiento dinámico de la activación y la represión de los genes a lo largo de las diferentes fases del ciclo celular, y pueden ayudar en la identificación de las transiciones de fase en los datos.

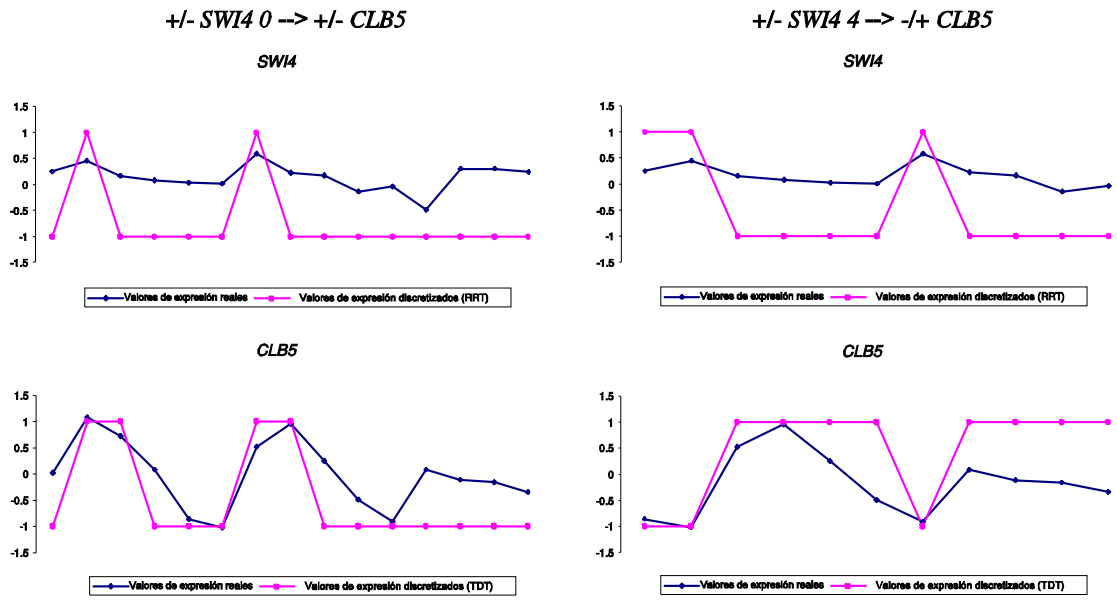


Figura 7.4. Perfiles de expresión reales y discretizados de los genes *SWI4* y *CLB5* con 0 (izquierda) y 4 (derecha) unidades de retardo para el conjunto de datos cdc15.

Dejando de lado el análisis anterior, es necesario aclarar que no pretendemos que las reglas inferidas por GRNCOP2 siempre representen asociaciones regulatorias confiables entre los genes. Creemos que nuestro enfoque de extracción de reglas puede ser útil para la identificación de algunas hipótesis prometedoras, cuya corroboración por medio de experimentos biológicos será siempre obligatoria a fin de obtener nuevos conocimientos curados. Además de esto, debe quedar claro que importantes interacciones conocidas no serán encontradas por GRNCOP2 (y por cualquier otro enfoque manejado por los datos) si los datos de *microarrays* no tiene correlaciones entre los genes implicados en este tipo de relaciones en los lapsos de tiempo que se analizan.

7.1.3.3. Estudio a nivel de genoma completo

El objetivo de este análisis es mostrar la utilidad y la capacidad de GRNCOP2 en estudios a nivel de genoma completo. Para dar cuenta de esto, hemos aplicado el algoritmo propuesto a varios conjuntos de datos de *microarrays* de series de tiempo (Spellman *et al.*, 1998; Ronen y Botstein, 2006; Lai *et al.*, 2005; Pramila *et al.*, 2002; Pramila *et al.*, 2006; Sapra et al., 2004) para el organismo *Saccharomyces cerevisiae*, descargados de la base de datos *Gene Expression Omnibus* (GEO) (Barrett y Edgar, 2006), y de otras fuentes (Spellman *et al.*, 1998). La lista completa de fuentes se resume en la tabla 7.7.

Tabla 7.7. Lista de los conjuntos de datos empleados en el estudio a nivel de genoma completo. Algunos conjuntos de datos fueron separados en dos conjuntos diferentes basándose en las condiciones experimentales descritas para cada uno.

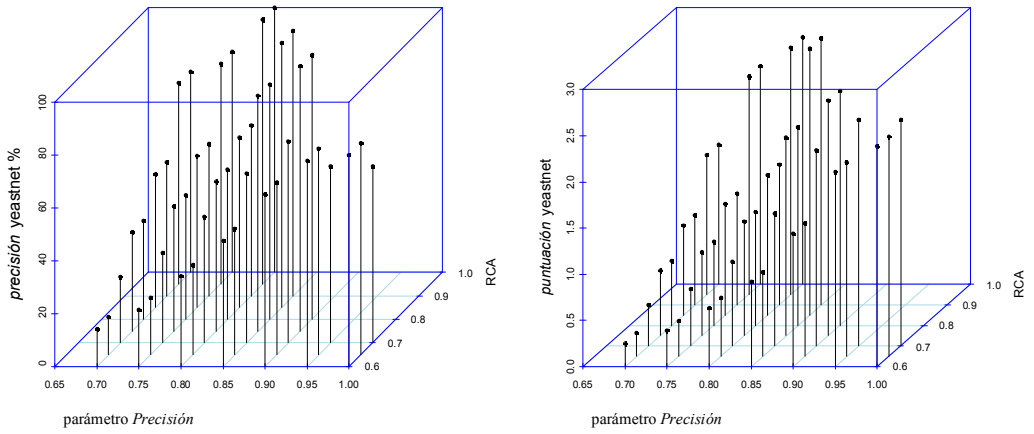
Conjuntos de datos de series de tiempo de Microarrays	Referencia	cantidad de muestras
GDS1752_d1	Ronen y Botstein, 2006	12
GDS1752_d2		14
GDS2003_d1	Lai *et al*., 2005	15
GDS2003_d2		15
GDS2347	Pramila *et al., 2002*	13
GDS2350_d1	Pramila *et al*., 2006	25
GDS2350_d2		25
GDS759	Sapra *et al*., 2004	24
ELUTRIATION	Spellman *et al*., 1998	14
ALPHA FACTOR		18
CDC15		24
CDC28		17

Con el fin de realizar inferencias de reglas a partir de estos conjuntos de datos, se realizaron algunos pasos previos. Dado que la lista de genes reportados en cada conjunto de datos difiere ligeramente de los otros conjuntos de datos, hemos seleccionado solo los genes que se han medido en todos los estudios. Es más, esta lista fue filtrada de acuerdo a los genes de las bases de datos de referencia descritas anteriormente. Esto dio lugar a una lista final de 5.245 genes de levadura sobre la cual se enfocó este estudio. Además, las muestras de algunos conjuntos de datos (Ronen y Botstein, 2006; Lai *et al*., 2005; Pramila *et al.*, 2006) se separaron en dos conjuntos diferentes basados en las condiciones experimentales descritas para cada uno, lo que resultó en 12 conjuntos de datos diferentes. Se utilizó todo el conjunto de muestras para este análisis. Por último, se estimaron los valores perdidos utilizando un método bayesiano de estimación de valores perdidos (Oba *et al.*, 2003). Es necesario aclarar que a pesar del hecho de que los conjuntos de datos en realidad tienen diferentes frecuencias de muestreo, no se realizó ninguna normalización de estas frecuencias. Por lo tanto, las reglas diferidas en el tiempo se deben interpretar como se discutió previamente en la sección 7.1.2.2.

Para este análisis, se realizaron 63 ejecuciones del algoritmo GRNCOP2, que resultaron de la variación del parámetro *Precisión* entre 0.70 y 1 con incrementos de 0.05 y de la variación del parámetro *RCA* entre 0.60 y 1 con incrementos de 0.05. Además, sólo las reglas con una retardo de hasta 4 unidades de tiempo (**W** = 4) fueron inferidas ya que consideramos que este valor es apropiado (en cuanto a su magnitud) para evaluar la escalabilidad del algoritmo a nivel del genoma completo. Sin embargo, a fin de obtener relaciones diferidas en el tiempo significativas entre los genes, se anima a los investigadores a seguir la recomendación dada en la ecuación 7.5 teniendo en cuenta sus hipótesis acerca de las regulaciones diferidas en el tiempo que pueden

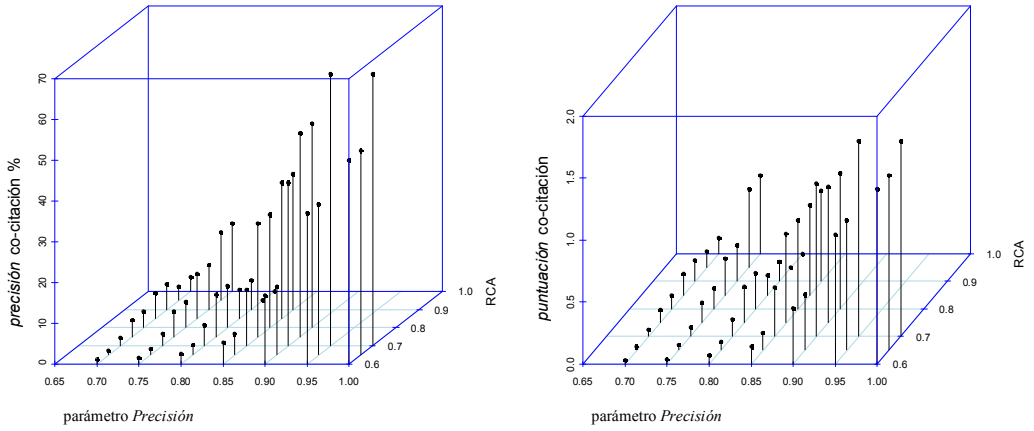
estar presentes en los experimentos. El parámetro *SCP* se fijó en 0.95 siguiendo el criterio sugerido dado que el objetivo es analizar el comportamiento del algoritmo variando la proporción de conjuntos de datos que soportan las reglas. Cada corrida llevó 30 min. de ejecución en un procesador de seis núcleos con 8 GB de RAM.

En cuanto a los resultados, la figura 7.5 muestra las métricas *precisión* y *puntuación* sobre los conjuntos de referencia y el número de asociaciones alcanzados por GRNCOP2 en cada ejecución. Los puntos de la esquina superior derecha de las figuras (donde la *Precisión* y el parámetro *RCA* se acercan a 1) se han omitido ya que el algoritmo no pudo obtener ninguna regla con esos valores en los parámetros. Como se puede observar, a medida que los valores de los parámetros *Precisión* y *RCA* aumentan, los valores de *precisión* y *puntuación* obtenidos por el algoritmo mejoran (véanse las figuras 7.5a a 7.5e). Esto es importante por el hecho de que el método muestra un comportamiento adecuado del proceso de consenso de reglas y de la función objetivo (ecuación 7.7), ya que la importancia del conjunto de reglas está directamente relacionada con los valores de esos parámetros. Por otro lado, el número de interacciones también disminuye considerablemente (véase la figura 7.5f). Aún más, si consideramos la métrica *sensibilidad*, GRNCOP2 sólo es capaz de recuperar a lo sumo el 4.69%, 1.23% y 1.85% de las interacciones en los conjuntos de referencia Yeastnet, Co-citación y GO, respectivamente, valores que disminuyen con la reducción del número de asociaciones. A pesar de que el método parece lograr un bajo rendimiento en cuanto a la métrica *sensibilidad*, se debe tener en cuenta la escala real del estudio a nivel de genoma completo realizado aquí, ya que sólo el 0.70%, el 0.20% y el 0.45 % de todas las posibles interacciones de pares de genes pertenecen a los conjuntos de referencia Yeastnet, Co-citación y GO, respectivamente. Es más, estos conjuntos de referencia se obtuvieron empleando diferentes fuentes de información, por lo que no es incluso realista esperar que puedan ser recuperados utilizando sólo estos datos de *microarrays*, especialmente si la información no está presente en los datos de expresión génica. Se debe considerar también que en la discusión anterior, la métrica *especificidad* se omitió debido a la gran cantidad de ***TN***s que los tres conjuntos de referencia imponen, haciendo que el algoritmo obtenga siempre un valor por encima del 99 % en esta métrica.



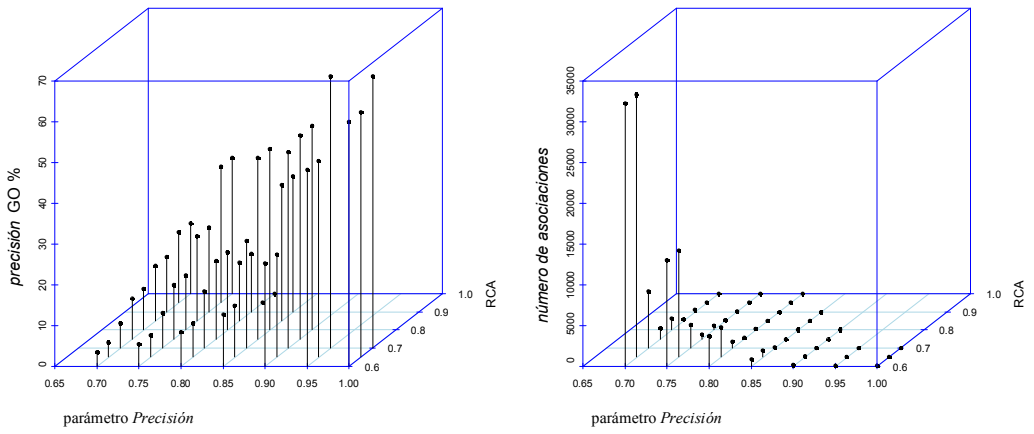
(a)

(b)



(c)

(d)



(e)

(f)

Figura 7.5. Valores de las métricas *precisión* y *puntuación* alcanzados por GRNCOP2 con los parámetros *Precisión* y *RCA* variando desde 0.70 a 1 y desde 0.60 a 1 respectivamente, con el parámetro *SCP* fijo en 0.95, y con **W** = 4. También se muestra el número de asociaciones. Fig. 7.5a: *precisión* yeastnet. Fig. 7.5b: *puntuación* yeastnet. Fig. 7.5c: *precisión* co-citación. Fig. 7.5d: *puntuación* co-citación. Fig. 7.5e: *precisión* GO. Fig. 7.5f: número de asociaciones.

Sin embargo, el análisis anterior no dice nada acerca de la naturaleza biológica de la RRG obtenida de estos conjuntos de datos. Por lo tanto, se realizó un análisis más profundo con el objetivo de descubrir el conocimiento real recuperado por GRNCOP2. La figura 7.6 representa la RRG obtenida en una de las ejecuciones anteriores, la que corresponde a un valor de *Precisión* y *RCA* de 0.75 y 0.75 respectivamente. Esta RRG consta de 352 genes y 559 reglas. Los genes se agruparon en función de su conectividad con el fin de mejorar la visualización.

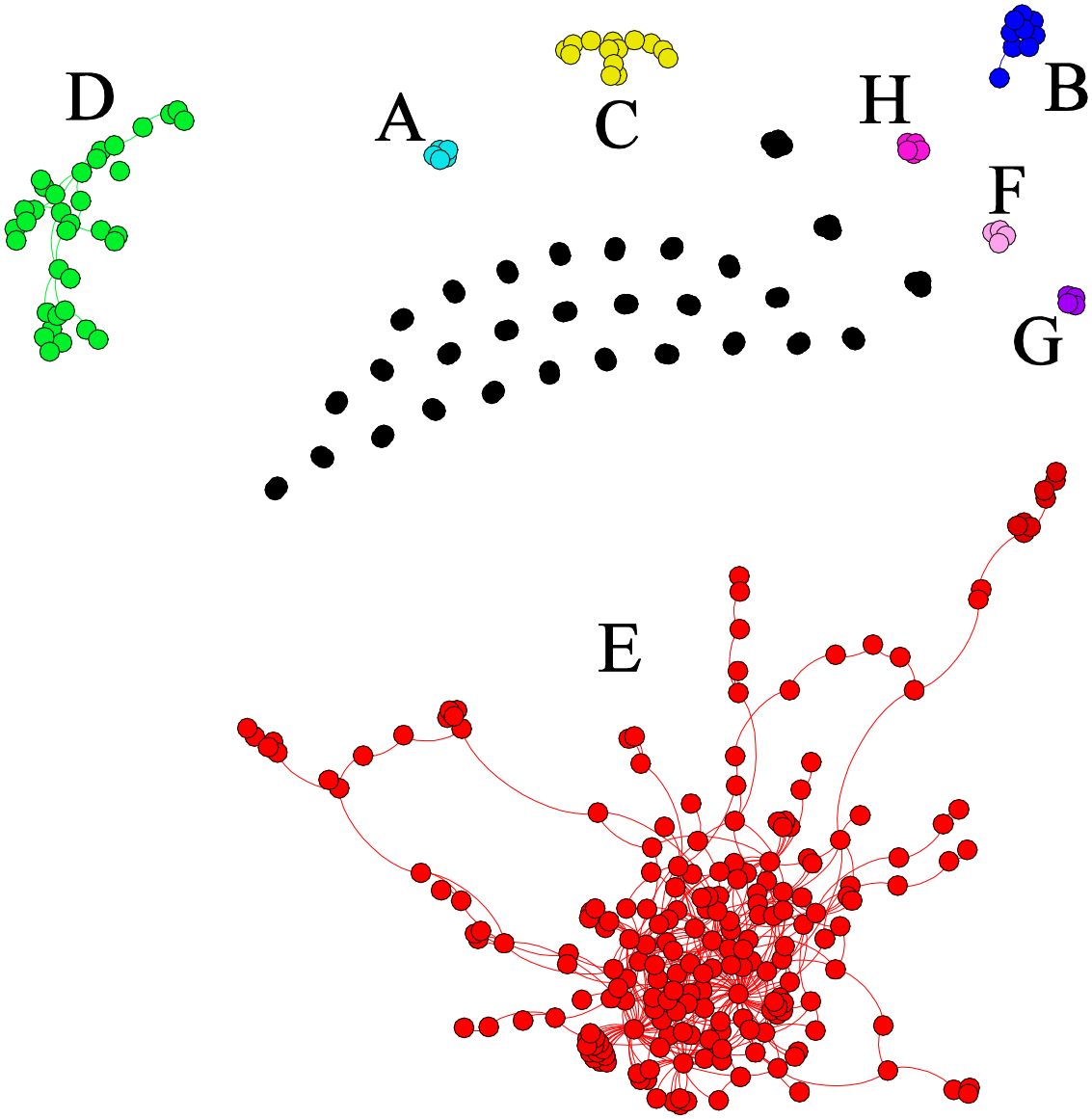


Figura 7.6. Red regulatoria de genes reconstruida con *Precision* = 0.75, *RCA* = 0.75, *SCP* = 0.95 y **W** = 4.

Es fácil concluir que la RRG resultante no es un grafo totalmente conexo. En cambio, se pueden identificar visualmente varias sub-redes junto con otras reglas que están absolutamente desconectadas. Por lo tanto, la siguiente pregunta puede surgir: ¿es posible que los genes que componen cada sub-red puedan estar relacionados de alguna manera? Esta pregunta será

respondida realizando un análisis ontológico sobre estos grupos de genes. Por lo tanto, se examinaron el proceso biológico, la función molecular y los componentes celulares de las ocho sub-redes más grandes usando Onto-Express (Draghici *et al.*, 2003), asumiendo una distribución hiper-geométrica y haciendo referencia a los cálculos con los 5245 genes analizados. Estos resultados se muestran en la tabla 7.8, junto con los valores obtenidos cuando se considera toda la RRG.

Como se muestra en la tabla 7.8, todas las sub-redes logran un relativamente alto grado de enriquecimiento ontológico no trivial en al menos una de las categorías, y estos resultados son estadísticamente significativos a un nivel de $\alpha = 0.01$. Por otra parte, la proporción de enriquecimiento de genes de cada grupo es mayor que la proporción de enriquecimiento de genes de toda la RRG. Estos resultados demuestran que los genes de cada sub-red están altamente relacionados unos con otros, relaciones que son establecidas directamente o indirectamente a través de las reglas descubiertas por GRNCOP2.

Tabla 7.8. Las ocho sub-redes más grandes, con su respectivo enriquecimiento ontológico, para las categorías de proceso biológico, función molecular y componente celular. La columna *anotación* denota la anotación más común para los genes en las sub-redes, mientas que la columna *porcentaje* es el porcentaje de genes con respecto al número de genes en la sub-red que recibió tal anotación. El *valor p corregido* es la significancia estadística de la anotación. Finalmente, las categorías y valores en negrita remarcan los casos donde las anotaciones fueron estadísticamente significativas a un nivel de α del 0.01.

	Proceso Biológico			**Función Molecular**			**Componente Celular**			
	anotación	*porcentaje*	*valor p correjido*	*anotación*	*porcentaje*	*valor p correjido*	*anotación*	*porcentaje*	*valor p correjido*	*numero de genes*
A	**translation**	**100%**	**0**	**structural constituent of ribosome**	**100%**	**0**	**ribosome**	**100%**	**0**	5
B	**chromatin assemby or disassembly**	**88.89%**	**0**	**DNA binding**	**88.89%**	**0**	**nucleosome**	**88.89%**	**0**	9
C	**cell cycle**	**50%**	**5.80E-04**	DNA binding	35.71%	0.06779	nucleus	71.43%	0.04154	14
D	**DNA replication**	**37.14%**	**0**	**DNA binding**	**40%**	**0**	**nucleus**	**71.43%**	**3.40E-04**	35
E	**ribosome biogenesis**	**53.74%**	**0**	molecular function	38.79%	0.02142	**nucleus**	**62.62%**	**0**	214
F	**cell division**	**75.00%**	**0.00247**	molecular function	50.00%	0.35995	cytoplasm	50.00%	0.73076	4
G	**methionine biosynthetic process**	**100.00%**	**0**	transferase activity	75.00%	0.01018	**cytoplasm**	**100.00%**	**0**	4
H	biological process	60.00%	0.18427	**nucleic acid binding**	**80.00%**	**3.00E-05**	cellular component	60.00%	0.04812	5
Toda	**ribosome biogenesis**	**35.51%**	**0**	molecular function	33.24%	0.31227	**nucleus**	**57.39%**	**0**	352

7.1.4. Conclusiones

En este trabajo, presentamos un algoritmo de optimización combinatorial de modelo libre llamado GRNCOP2, diseñado para la inferencia de RRGs. Aunque las ideas básicas detrás su antecesor GRNCOP permanecen en GRNCOP2 (es decir, los umbrales adaptativos de regulación y la optimización combinatorial de los clasificadores de reglas), el método presentado en esta sección es un nuevo algoritmo que constituye una evolución relevante del método anterior, debido a los desafíos que imponen las mejoras propuestas. El nuevo algoritmo

incorpora novedosas características tales como inferencia de reglas con múltiples retrasos de tiempo y en un número ilimitado de conjuntos de datos de series de tiempo, e incluye mejoras sobre todo el proceso de inferencia. Esta última característica se demuestra por el hecho de que los resultados obtenidos por GRNCOP2 son significativamente mejores que los resultados obtenidos por la versión anterior. Además, la relevancia del nuevo método se hizo más evidente ya que los puntajes obtenidos por GRNCOP2 fueron superiores a los obtenidos por otros algoritmos relacionados en términos de las métricas propuestas. Asimismo, las relaciones inferidas por GRNCOP2 demostraron ser biológicamente relevantes. Es más, GRNCOP2 fue capaz de obtener nuevas posibles interacciones entre genes, en consonancia con conocimientos biológicos previos, que no fueron descubiertas por ninguno de los otros métodos.

Por otro lado, también se evaluó la capacidad de GRNCOP2 para realizar estudios a nivel de genomas completos. En este sentido, se realizó un análisis sobre varios conjuntos de datos de series de tiempo de genoma completo, para los que se discutió el buen funcionamiento del algoritmo en términos de las métricas propuestas. Además, con la realización de un análisis ontológico se mostró que los resultados son significativos en términos biológicos, ya que se encontró que los genes de las sub-redes descubiertas están altamente relacionados en términos estadísticos.

Sin embargo, este estudio no afirma que el enfoque de aprendizaje automático basado en datos propuesto en este trabajo sea suficiente para inferir redes de regulación biológicamente significativas. No obstante, esta herramienta puede ofrecer evidencia significativa necesaria para ayudar a los científicos en la exploración e identificación de asociaciones biológicamente relevantes, cuya evaluación por experimentos biológicos es una condición para alcanzar nuevos conocimientos curados.

7.2. GeRNet: Una Plataforma Integradora para la Inferencia de Redes Regulatorias de Genes

En esta sección, introduciremos una herramienta de software llamada GeRNet (por *Gene Regulatory Networks*) para la inferencia de RRGs basadas en reglas de asociación a partir de múltiples conjuntos de datos de series de tiempo (o estados en equilibrio) de *microarrays*. El software integra al algoritmo GNRCOP2, introducido en la sección anterior, para la inferencia de reglas de asociación diferidas en el tiempo. También incorpora al algoritmo de *biclustering* BiHEA, presentado en el capítulo 5, para la inferencia de relaciones adicionales que podrían no ser capturadas por GRNCOP2, mejorando así las capacidades globales de inferencia. Además, el software posee varias cualidades para el manejo de datos, pre-procesamiento y visualización y manipulación de resultados, las cuales serán descritas a continuación.

7.2.1. Manejo de Datos

El software permite elegir varios conjuntos de datos de expresión de genes para realizar la inferencia de la RRG. Estos datos se organizan en una estructura tabular en el panel derecho de la interfaz de usuario (figura 7.7). La vista muestra cada archivo de datos en una lengüeta diferente indicando si corresponde a un conjunto de datos de series de tiempo o de estados en equilibrio. Además, se puede elegir para los datos entre una visualización en forma de mapa de calor (figura 7.7) o de matriz numérica (figura 7.8). En la vista como mapa de calor, el software permite la elección de los colores que representan los valores por encima y por debajo del promedio en el mapa. En el modo de vista numérica, los valores de conjunto de datos pueden modificarse manualmente haciendo doble clic en la celda correspondiente. Además, es posible estimar los valores faltantes en todos los conjuntos de datos con solo presionar el botón "*Estimate Missing Values*". Ambas vistas permiten hacer zoom sobre los conjuntos de datos con varias escalas para mejorar la visualización.

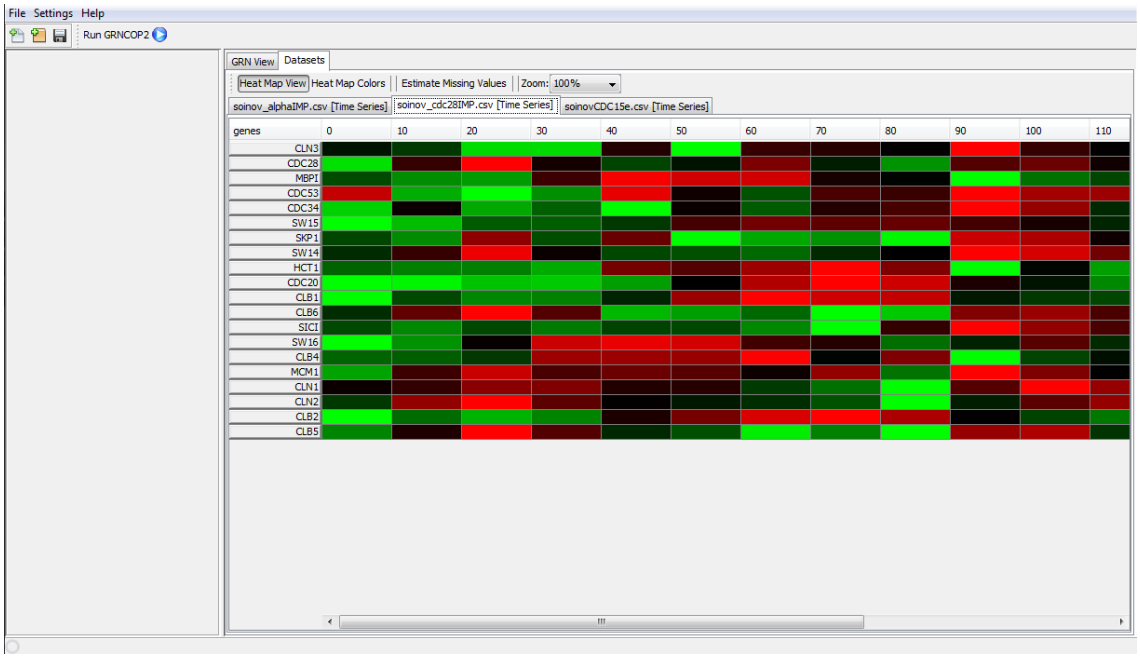
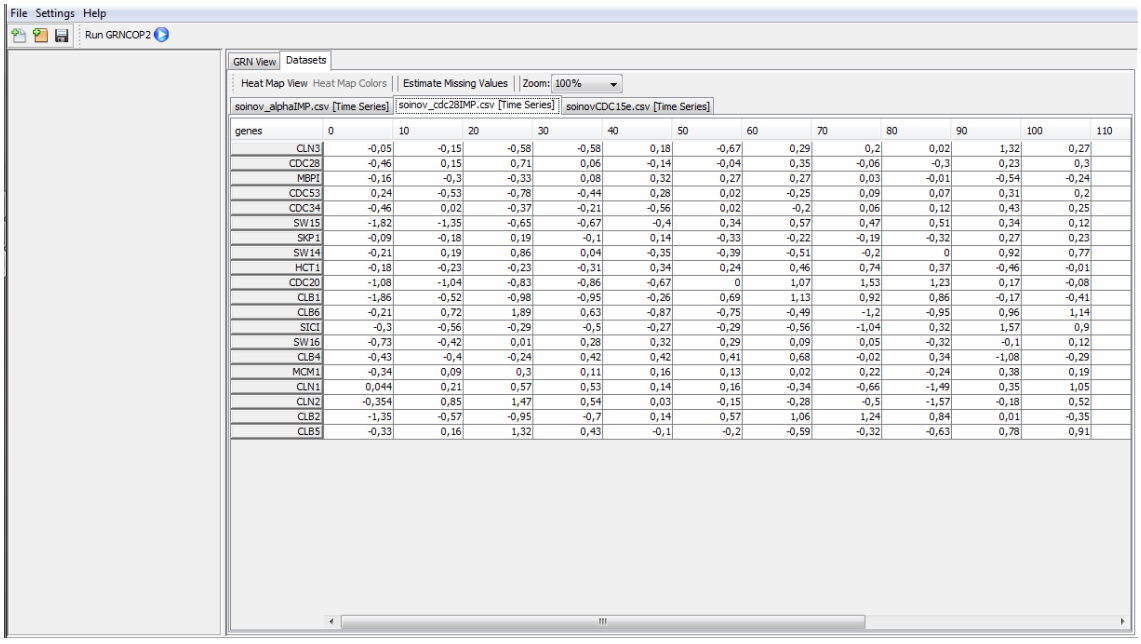


Figura 7.7. Vista de los conjuntos de datos de expresión en forma de mapa de calor.



genes	0	10	20	30	40	50	60	70	80	90	100	110
CLN3	-0,05	-0,15	-0,58	-0,58	0,18	-0,67	0,29	0,2	0,02	1,32	0,27	
CDC28	-0,46	0,15	0,71	0,06	-0,14	-0,04	0,35	-0,06	-0,3	0,23	0,3	
MBPI	-0,16	-0,3	-0,33	0,08	0,32	0,27	0,27	0,03	-0,01	-0,54	-0,24	
CDC53	0,24	-0,53	-0,78	-0,44	0,28	0,02	-0,25	0,09	0,07	0,31	0,2	
CDC34	-0,46	0,02	-0,37	-0,21	-0,56	0,02	-0,2	0,06	0,12	0,43	0,25	
SW15	-1,82	-1,35	-0,65	-0,67	-0,4	0,34	0,57	0,47	0,51	0,34	0,12	
SKP1	-0,09	-0,18	0,19	-0,1	0,14	-0,33	-0,22	-0,19	-0,32	0,27	0,23	
SW14	-0,21	0,19	0,86	0,04	-0,35	-0,39	-0,51	-0,2	0	0,92	0,77	
HCT1	-0,18	-0,23	-0,23	-0,31	0,34	0,24	0,46	0,74	0,37	-0,46	-0,01	
CDC20	-1,08	-1,04	-0,83	-0,86	-0,67	0	1,07	1,53	1,23	0,17	-0,08	
CLB1	-1,86	-0,52	-0,98	-0,95	-0,26	0,69	1,13	0,92	0,86	-0,17	-0,41	
CLB6	-0,21	0,72	1,89	0,63	-0,87	-0,75	-0,49	-1,2	-0,95	0,96	1,14	
SIC1	-0,3	-0,56	-0,29	-0,5	-0,27	-0,29	-0,56	-1,04	0,32	1,57	0,9	
SW16	-0,73	-0,42	0,01	0,28	0,32	0,29	0,09	0,05	-0,32	-0,1	0,12	
CLB4	-0,43	-0,4	-0,24	0,42	0,42	0,41	0,68	-0,02	0,34	-1,08	-0,29	
MCM1	-0,34	0,09	0,3	0,11	0,16	0,13	0,02	0,22	-0,24	0,38	0,19	
CLN1	0,044	0,21	0,57	0,53	0,14	0,16	-0,34	-0,66	-1,49	0,35	1,05	
CLN2	-0,354	0,85	1,47	0,54	0,03	-0,15	-0,28	-0,5	-1,57	-0,18	0,52	
CLB2	-1,35	-0,57	-0,95	-0,7	0,14	0,57	1,06	1,24	0,84	0,01	-0,35	
CLB5	-0,33	0,16	1,32	0,43	-0,1	-0,2	-0,59	-0,32	-0,63	0,78	0,91	

Figura 7.8. Vista de los conjuntos de datos de expresión en forma de matriz numérica.

7.2.2. Inferencia de RRGs

A fin de poder ejecutar al algoritmo GRNCOP2, es necesario tener al menos un conjunto de datos cargado en el software, ya sea por medio de la creación de un nuevo proyecto o abriendo uno existente. Una vez que el algoritmo fue puesto a correr, aparecerá una barra de progreso indicando el avance de la búsqueda. Los resultados de la búsqueda dependen de los tipos de datos cargados y de los parámetros elegidos. El tipo de los conjuntos de datos determina primariamente el retardo de las reglas que pueden ser inferidas. Si todos los conjuntos de datos son de tipo serie de tiempo, entonces el máximo retardo de tiempo posible está determinado por el conjunto de datos con menor cantidad de muestras menos 4 (Gallo *et al.*, 2011a). Si todos los conjuntos de datos son de estados en equilibrio, entonces solo se podrán inferir reglas simultáneas. En cualquier otro caso, cuando haya conjuntos de datos de tipos mixtos (series de tiempo y estados en equilibrio), la reglas simultáneas serán inferidas con todos los datos, mientras que las reglas diferidas en el tiempo serán inferidas solo a partir de los datos de series de tiempo.

7.2.3. Visualización y Manipulación de la RRG

Una vez que GRNCOP2 termina la búsqueda, la RRG asociada a las reglas inferidas es automáticamente dibujada en la pestaña "*GRN View*" (figura 7.9). Además, aparecerá una tabla conteniendo todas las reglas listadas en el lado derecho de la ventana principal. Cada gen en la pestaña *GRN View* está representado como un vértice en forma de círculo verde, mientras que las reglas que los relacionan son arcos dirigidos dibujados acorde al siguiente esquema:

- +/- g_r $d \rightarrow$ +/- g_i está representada por un arco dirigido de g_r a g_i y significa que cuando g_r está sobre expresado (sub expresado), g_i es activado (inhibido) con un retardo de d unidades de tiempo.
- + g_r $d \rightarrow$ + g_i está representada por un arco dirigido de g_r a g_i y significa que cuando g_r está sobre expresado, g_i es activado con un retardo de d unidades de tiempo.
- - g_r $d \rightarrow$ - g_i está representada por un arco dirigido de g_r a g_i y significa que cuando g_r está sub expresado, g_i es inhibido con un retardo de d unidades de tiempo.
- +/- g_r $d \rightarrow$ -/+ g_i está representada por un arco dirigido de g_r a g_i y significa que cuando g_r está sobre expresado (sub expresado), g_i es inhibido (activado) con un retardo de d unidades de tiempo.
- + g_r $d \rightarrow$ - g_i está representada por un arco dirigido  de g_r a g_i y significa que cuando g_r está sobre expresado, g_i es inhibido con un retardo de d unidades de tiempo.
- - g_r $d \rightarrow$ - g_i está representada por un arco dirigido  de g_r a g_i y significa que cuando g_r está sub expresado, g_i es activado con un retardo de d unidades de tiempo.

Solo los genes que tienen al menos una regla asociada son dibujados en la RRG.

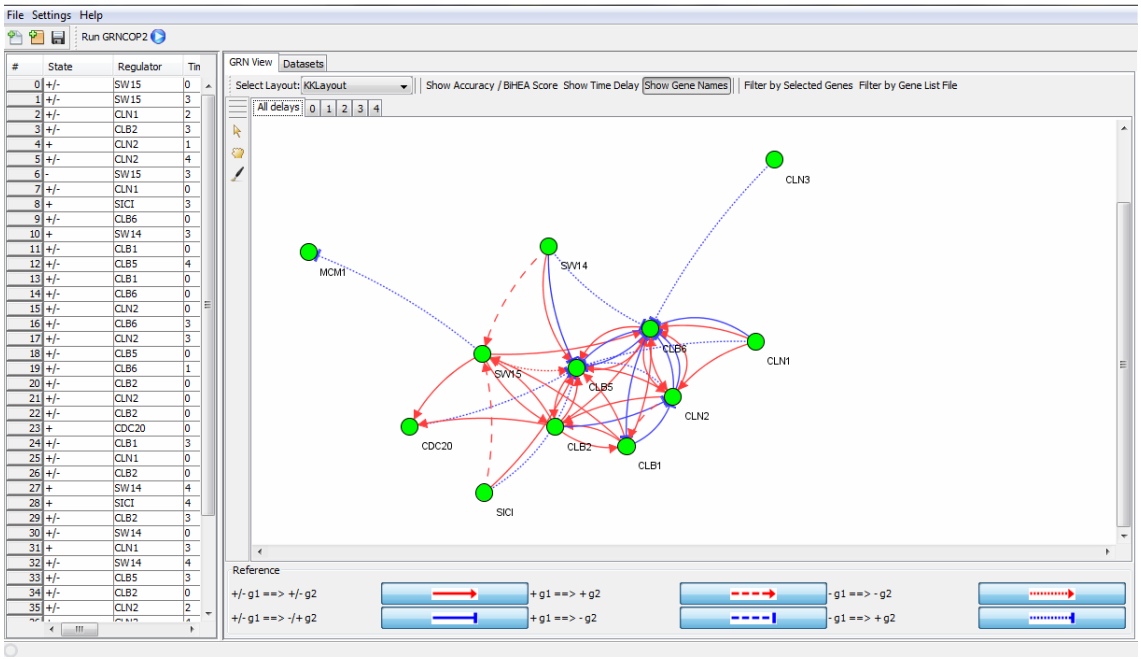


Figura 7.9. Vista de la RRG inferida.

La visualización de RRG ofrece varias vistas de la red. Contiene una pestaña llamada "*All delays*" que muestra la red completa con todas las reglas diferidas en el tiempo, y también contiene una pestaña por retardo de tiempo que muestra solo las reglas para ese retardo de tiempo específico (figura 7.10).

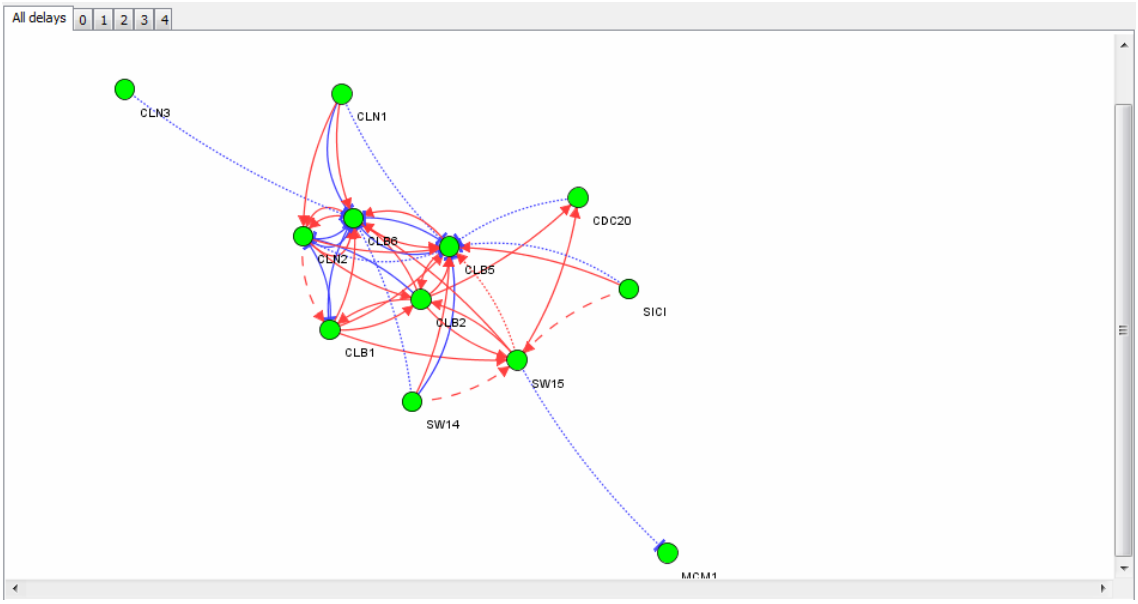


Figura 7.10. Vista de la RRG inferida en múltiples retardos de tiempo.

Debajo de la vista del grafo de la red hay un panel de referencia que contiene indicaciones del significado de cada arco (figura 7.11). También se permite ocultar cualquier tipo de interacción haciendo clic en los botones que contienen al arco.

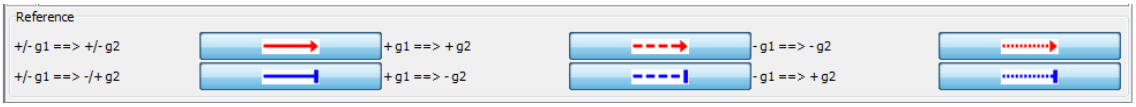


Figura 7.11. Panel de control de visualización de interacciones.

Arriba de la vista del grafo de red hay una barra de opciones dirigida a personalizar aún más la vista actual de la red (figura 7.12). En ella se pueden elegir entre varios *layouts* que automáticamente reorganizan los genes acorde a algún criterio específico. También se pueden añadir o eliminar varios tipos de etiquetas para los vértices y los arcos del grafo, tales como la precisión y el tiempo de retardo de las reglas, y los nombres de los genes. Además, se permite ocultar genes acorde a alguna selección específica en el grafo, o por medio de algún archivo conteniendo una lista específica de genes.

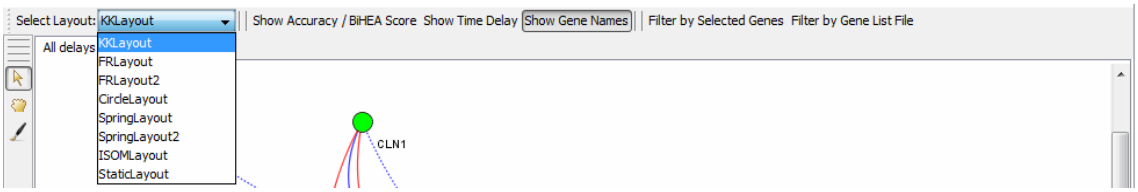


Figura 7.12. Panel de control de etiquetas, *layouts* y filtros.

En el lado izquierdo de la visualización de la RRG hay una barra de herramientas que permite interactuar con la red (figura 7.12). Contiene tres herramientas: la flecha , la mano y la birome :

- Herramienta de flecha : Esta herramienta pone a la visualización en modo selección, permitiendo elegir tanto genes como interacciones de múltiples formas.
- Herramienta de mano : Esta herramienta pone a la visualización en modo transformación, permitiendo trasladar, rotar o inclinar a la RRG.
- Herramienta de birome : Esta herramienta pone a la visualización en modo edición, permitiendo añadir genes o interacciones nuevas a la RRG.

Además, con cualquier herramienta se permite escalar la visualización con la rueda del *mouse* o hacer clic derecho para desplegar un menú contextual con varias opciones (figura 7.13), entre ellas la de emplear al algoritmo BiHEA para buscar nuevas interacciones y la de exportar la visualización. Estas últimas opciones serán descritas en las siguientes secciones.

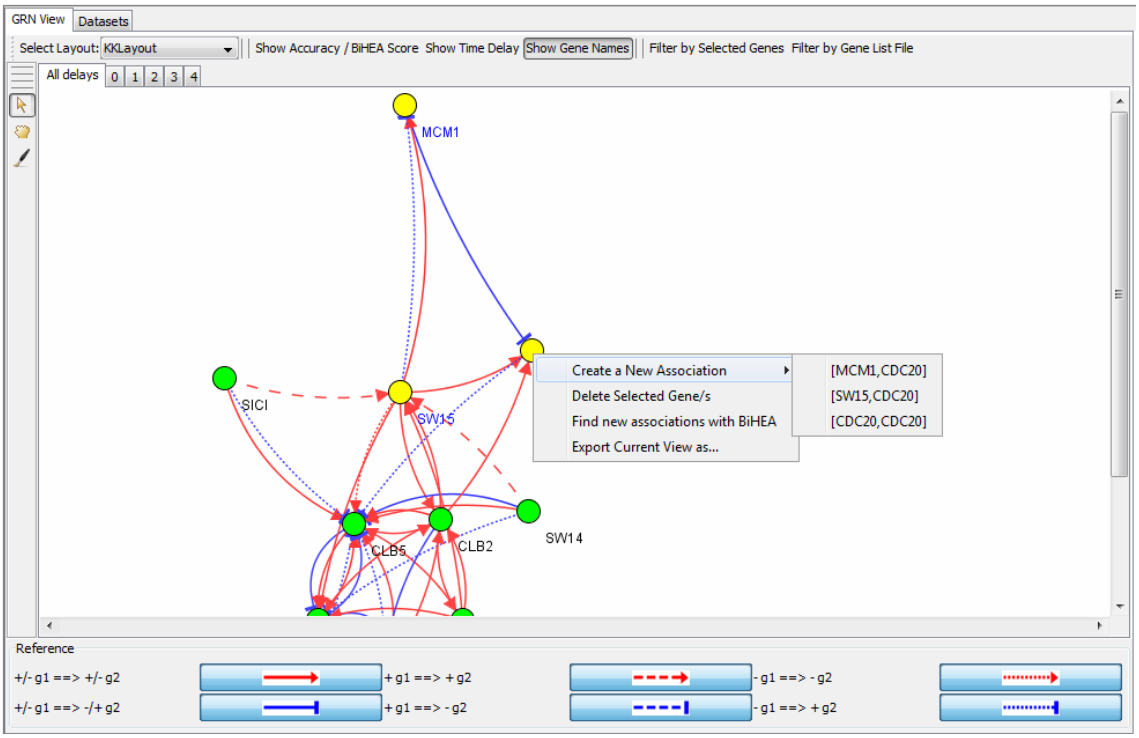


Figura 7.13. Menú contextual con varias opciones.

7.2.4. Integración con BiHEA

El software integra al algoritmo BiHEA para permitir buscar nuevas interacciones que podrían no ser descubiertas por el algoritmo GRNCOP2. Dado que BiHEA solo encuentra subconjuntos de genes que están relacionados en un subconjunto de condiciones (ver capítulo 5

para más detalles), se necesita de algún mecanismo adicional para permitir recuperar interacciones que se comparen a las reglas inferidas por GRNCOP2 y así, añadirlas a la RRG.

La integración con el algoritmo BiHEA involucra varios pasos. Primero, el usuario debe elegir en la pestaña *GRN View* cuáles serán los genes para los cuales se buscaran nuevas interacciones utilizando este método. Estos genes actuarán como reguladores potenciales para los otros genes en el conjunto de datos. Luego, se puede proceder a iniciar la búsqueda eligiendo la opción del menú contextual "*Find new associations with BiHEA*" (figura 7.13).

La búsqueda está relacionada al actual tiempo de retardo mostrado en la red, es decir, si el tiempo de retardo mostrado es 1, entonces el algoritmo buscará reglas con un tiempo de retardo de una unidad. Si todos los tiempos de retardo son visualizados (es decir, esta seleccionada la pestaña "*All delays*"), entonces aparecerá un cuadro de diálogo permitiendo al usuario elegir un tiempo de retardo específico. Este tiempo de retardo afecta a la forma en que los datos son tomados por el algoritmo, dado que los genes elegidos para actuar como potenciales reguladores deben ser desplazados en los datos con respecto a los otros genes acorde al tiempo de retardo elegido. Como ejemplo, suponga un conjunto de datos k_1 con 5 genes y 6 muestras. Si los genes que están seleccionados son el gen g_2 y el gen g_4, y el tiempo de tardo es 1, entonces la matriz sobre la cual se realizará la búsqueda es la k_1' (figura 7.14).

k_1

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
g_1	0.64	0.19	0.69	0.47	0.88	0.30	0.40	0.99	0.87	0.10
g_2	0.83	0.71	0.81	0.54	0.52	0.18	0.03	0.61	0.22	0.20
g_3	0.63	0.25	0.27	0.46	0.18	0.05	0.41	0.35	0.09	0.75
g_4	0.01	0.45	0.87	0.62	0.60	0.65	0.03	0.44	0.83	0.46
g_5	0.24	0.08	0.41	0.17	0.82	0.41	0.15	0.28	0.89	0.43

⇩

k_1'

		s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
g_1		0.64	0.19	0.69	0.47	0.88	0.30	0.40	0.99	0.87	0.10
g_2	0.83	0.71	0.81	0.54	0.52	0.18	0.03	0.61	0.22	0.20	
g_3		0.63	0.25	0.27	0.46	0.18	0.05	0.41	0.35	0.09	0.75
g_4	0.01	0.45	0.87	0.62	0.60	0.65	0.03	0.44	0.83	0.46	
g_5		0.24	0.08	0.41	0.17	0.82	0.41	0.15	0.28	0.89	0.43

Figura 7.14. Transformación de la matriz de datos tomada como entrada por BiHEA con desplazamientos por el tiempo de retardo.

El siguiente paso realiza la búsqueda de *biclusters* con el algoritmo BiHEA en cada uno de los conjuntos de datos separadamente. Una vez que todas las búsquedas finalizaron, se realiza un análisis de pares para cada uno de los genes seleccionados. Para hacer la explicación más simple, tomemos el ejemplo anterior. Si g_2 y g_4 son los genes seleccionados, entonces el análisis

de pares computa para cada par de genes (g_2, g_1), (g_2, g_3), (g_2, g_4), (g_2, g_5) y (g_4, g_1), (g_4, g_2), (g_4, g_3), (g_4, g_5), el número de *biclusters* de k_1' que contienen a los dos genes del par. Luego, esos valores son promediados con los valores de los otros conjuntos de datos para obtener la puntuación final del BiHEA (BiHEA *Score*) para cada posible interacción.

Finalmente, cuando culmina todo el proceso descrito anteriormente, aparece un cuadro de dialogo (ver figura 7.15) mostrando los 50 mejores BiHEA *Scores* conjuntamente con el par de genes de la interacción, permitiendo al usuario elegir las asociaciones deseadas y descartar el resto. Las asociaciones elegidas son automáticamente incluidas en la RRG.

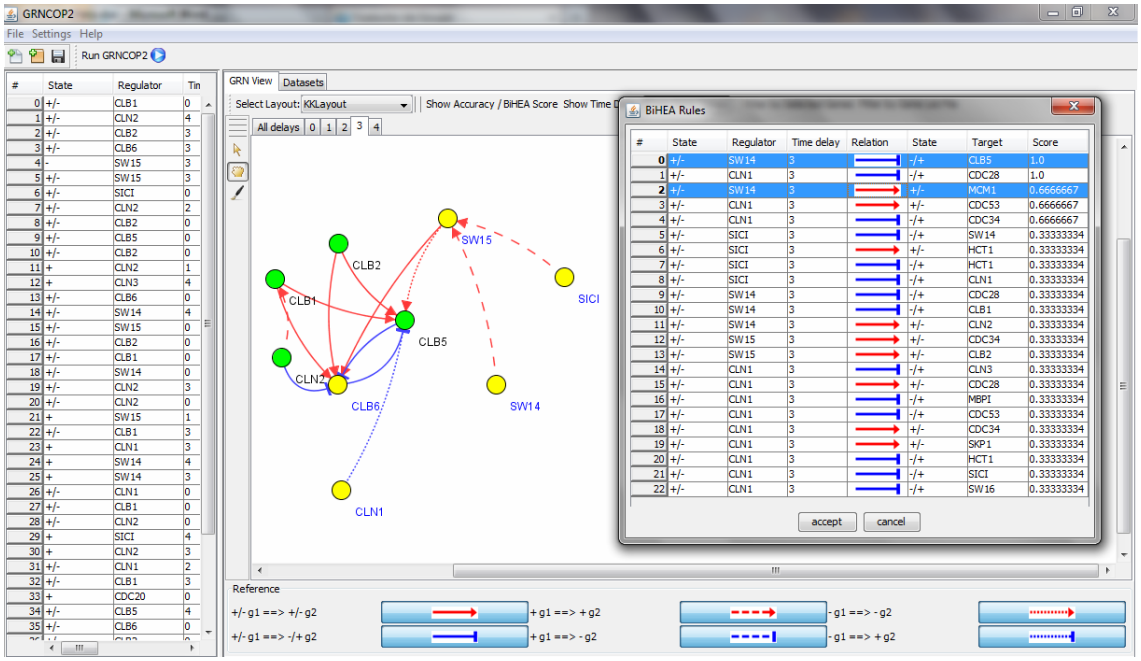


Figura 7.15. Resultados obtenidos por el algoritmo BiHEA.

Si la búsqueda se realiza para un retardo de tiempo igual a 0, es decir, para reglas simultáneas sin tiempo de retardo, entonces cada par de genes se muestra dos veces, una para cada uno de los genes actuando como posible regulador. Esto es porque a priori no se puede asumir causalidad en la relación dado que la única información disponible conocida es la co-expresión de los dos genes. Por otro lado, si la búsqueda se realiza para retardos de tiempos mayores a 0, es decir, para reglas diferidas en el tiempo, entonces se puede asumir que la expresión (o inhibición) del gen blanco es debido a la expresión (o inhibición) directa o indirecta del gen regulador en la muestra previa (Gallo *et al.*, 2013a). La regulación opuesta también es inferida, dado que para este desarrollo se extendió apropiadamente al algoritmo BiHEA para que construya los *biclusters* considerando tanto perfiles de expresión similares como perfiles de expresión opuestos de cada gen.

Con el fin de diferenciar la puntuación otorgada por el algoritmo BiHEA (BiHEA *Score*) a cada regla con la puntuación otorgada por el algoritmo GRNCOP2 (*Precisión*), dado que ambos

valores tienen diferentes interpretaciones, los valores obtenidos por el algoritmo BiHEA contienen las letras BS como posfijo del valor.

7.2.5 Exportación de Resultados

Finalmente, en esta sección describiremos brevemente los métodos para exportar los resultados. Hay dos tipos de resultados que pueden ser exportados por el software: las reglas inferidas por los algoritmos y la vista gráfica de la red correspondiente a esas reglas. Las reglas son exportadas como un archivo separado por comas (CSV) mediante la opción correspondiente en el menú principal. La vista gráfica puede ser exportada tanto desde el menú principal como desde el menú contextual del grafo (figura 7.13). Al seleccionar esta opción, aparece un cuadro de diálogo que permite elegir el nombre del archivo de salida y el formato en el cual se desea guardarlo. Hay varios formatos disponibles, entre los cuales están PDF, EPS, JPG, PNG y muchos otros más, conjuntamente con opciones para cada uno de ellos. La imagen guardada es la vista actual de la red, es decir, la red que actualmente se está visualizando en la pantalla, por lo que la misma no necesariamente representa a la red completa.

7.2.6 Conclusiones

En esta sección hemos introducido una plataforma de software llamada GeRNet. El software representa un marco de trabajo amigable al usuario que integra dos algoritmos recientemente publicados, proveyendo la oportunidad para realizar ingeniería inversa de redes regulatorias de genes diferidas en el tiempo basadas en reglas de asociación. También ofrece varias características para manejo de datos, visualización y manipulación de RRGs, creando un entorno bioinformático completo para el biólogo investigador. Sin embargo, algunas características deseables como la integración con conocimiento biológico externo de bases de datos como KEGG y *Gene Ontology* son dejadas para versiones futuras.

7.3. Sumario

En este capitulo introdujimos las principales contribuciones es esta tesis en relación a la inferencia de RA a partir de datos de expresión de genes. Como hemos extensamente mencionado, los métodos de minería de datos que se enfocan en la extracción de reglas de asociación constituyen una herramienta capaz de inferir este tipo de interacciones cualitativas entre genes. En este sentido, se diseño un nuevo algoritmo tomando como base un método de extracción de reglas de asociación entre genes previamente propuesto por nuestro grupo de investigación (Ponzoni *et al*., 2007). Esto conllevo a un rediseño total del método con el fin de superar sus limitaciones, incorporando nuevas características como la inferencia de reglas de interacción con múltiple retardo de tiempo, la consideración de múltiples datos de expresión de

genes en la extracción de las reglas, y varias mejoras sobre el proceso básico de inferencia. El resultado de estas investigaciones se encuentra publicado en Gallo *et al.* (2011a). También se presento en carácter de difusión en Gallo *et al*, (2010c) y en Gallo *et al*. (2011b). La contribución mas importante a destacar consiste en que el nuevo algoritmo de extracción de reglas supera tanto a su predecesor (Ponzoni *et al.*, 2007), como a otros métodos del estado del arte (Soinov *et al.*, 2003; Bulashevska y Eils, 2005; Li *et al*., 2006) en lo referente a calidad de resultados y a las posibilidades que brinda en cuanto a la información inferida. Además, se mostró su aplicabilidad en contextos de genoma completo, en donde la gran dimensionalidad en los genes torna prácticamente imposible la aplicación de otros algoritmos (Gallo *et al.*, 2011a).

Finalmente, introdujimos una plataforma de software llamada GeRNet (Gallo *et al*., 2013b), que representa un marco de trabajo amigable al usuario integrando dos de los algoritmos desarrollados en esta tesis. También ofrece varias características para manejo de datos, visualización y manipulación de RRG. De este modo, esta herramienta provee de un entorno bioinformático completo para el biólogo investigador, brindándole la oportunidad de realizar ingeniería inversa de redes regulatorias de genes diferidas en el tiempo basadas en reglas de asociación.

Capítulo 8

Conclusiones

El objetivo general de esta tesis consistió en diseñar nuevas técnicas computacionales para asistir, a expertos en bioinformática, en la obtención de nuevos conocimientos sobre el funcionamiento de los mecanismos existentes de regulación de genes en los organismos biológicos. Más específicamente, se buscó desarrollar metodologias computacionales que asistan en la reconstrucción (o descubrimiento) de la estructura relacional presente en las redes regulatorias de genes.

La redes regulatorias de genes gobiernan los niveles de expresión de los genes, permitiendo la supervivencia celular y afectando a numerosos procesos celulares. De este modo, la cantidad y los patrones temporales en los que estos productos aparecen en la célula son cruciales para los procesos de la vida. La introducción de nuevas tecnologías en los métodos experimentales, como los *microarrays*, han permitido estudios a gran escala que permiten medir paralelamente la expresión de miles de genes a nivel de genomas completos, en un dado instante de tiempo, para un dado conjuntos de condiciones experimentales, y para varias células o tejidos de interés. Esta tecnología introdujo una variedad de cuestiones en el análisis de datos que no estaba presente en la biología molecular tradicional.

En tal sentido, se han propuesto en los últimos años varias técnicas estadísticas y de inteligencia artificial para llevar a cabo la ingeniería inversa de las redes regulatorias de genes, a partir del monitoreo y análisis de los datos de expresión. Estas técnicas varían desde los modelos de redes booleanas simples hasta modelos continuos a nivel molecular. Esta tesis se centro en los enfoques de modelo libre, ya que son decididamente atractivos debido a las complejidades en la dinámica molecular de las redes. Estos métodos ofrecen una forma de identificar los mecanismos regulatorios directamente a partir de los datos de entrada sin ningún modelo subyacente. Además, ofrecen varias ventajas cuando se realiza análisis dirigido por los datos. Estas técnicas son altamente abstractas requiriendo la menor cantidad de datos en relación a otras más complejas, con una importante habilidad para realizar inferencias. La simplicidad de estos enfoques permiten la inferencia de modelos de gran tamaño con una alta velocidad de análisis. Aunque, por otro lado, solo pueden lidiar con aspectos dinámicos cualitativos.

Además, en el desarrollo de esta tesis se abordaron varios aspectos claves de la inferencia de redes regulatorias de genes. En primer lugar, se trataron las cuestiones relacionados con la co-expresión de genes bajo subconjuntos de condiciones experimentales. Seguidamente, se abordó la problemática de la inferencia de relaciones diferidas en el tiempo, tema clave que ayuda en el

descubrimiento de potenciales relaciones causales entre genes. También se atendieron cuestiones de integración de datos, integración de metodologías de inferencia y análisis de datos a nivel de genomas completos. Finalmente, se abordó en forma transversal un tema de vital importancia en la inferencia de redes de genes como lo es la validación. En tal sentido, se propusieron diferentes métricas realizando estudios con datos reales, así como también en menor medida con datos sintéticos.

A continuación se presentan y discuten las principales contribuciones de esta tesis, analizando las implicancias de las distintas propuestas. Además, se introducen recomendaciones y lineamientos generales para futuras investigaciones en el tema, así como también sugerencias sobre extensiones naturales del trabajo desarrollado en esta tesis.

8.1. Resumen de las Contribuciones

En primera instancia, nuestro enfoque estuvo centrado en relación a la inferencia de genes co-expresados a partir de datos de expresión de genes en *microarrays*. En este sentido, se emplearon nociones de *biclustering* y de algoritmos genéticos meméticos multi-objetivo basados es Pareto, desarrollando una metodología de inferencia que incorporó características novedosas. La primera de ellas esta relacionada con la posibilidad de inferir genes co-expresados pero cuyo nivel de expresión se da en forma opuesta, demostrando que es posible tomar en cuenta a estas "filas extras" mejorando la calidad de los *biclusters* sin perdida de eficiencia. Otra característica es el mecanismo que permite orientar la búsqueda genética en relación al compromiso existente entre los múltiples objetivos del algoritmo evolutivo, mediante un parámetro establecido previamente por el usuario. Esto provee a los científicos biólogos con un parámetro extra para determinar que *bicluster* consideran mas relevantes, brindando la posibilidad de ajustar el tamaño y la varianza de filas de los *biclusters* encontrados. Además, el trabajo fue desarrollado bajo la plataforma PISA, lo que permitió la evaluación y comparación de varios algoritmos evolutivos muti-objetivo reconocidos y su correcta validación por medio de herramientas estadísticas propias de la optimización multi-objetivo. Por ultimo, los resultados fueron comparados con los obtenidos por otro método perteneciente al estado del arte, siendo los del algoritmo propuesto superadores en relación a este. Toda la evaluación de los resultados fue llevada a cabo sobre dos conjuntos de datos reales de expresión de genes, a fin de mostrar la efectividad del enfoque propuesto.

A pesar de haber obtenido resultados importantes en el estudio mencionado previamente, ciertas características presentes en el problema de *biclustering*, como el solapamiento de *biclusters*, hacen que la utilización de algoritmos de optimización multi-objetivo de propósito general como los disponibles en PISA no sea óptima. La razón es la poca flexibilidad de estos

algoritmos evolutivos para incorporar este tipo de características en el proceso de búsqueda. De este modo, se desarrolló un nuevo algoritmo evolutivo memético diseñado específicamente para el problema de *biclustering* de datos de expresión de genes. Este algoritmo, al cual se lo denominó BiHEA, introduce dos mecanismos novedosos: el primero fue diseñado a fin de evitar la pérdida de buenas soluciones a través de las generaciones, mientras que mantiene un bajo grado de solapamiento entre los *biclusters* resultantes. El otro mecanismo fue concebido para mantener un nivel de diversidad satisfactorio en el espacio genotípico. La evaluación del algoritmo BiHEA se realizó tanto con datos sintéticos como con datos reales. En una primera fase experimental sobre datos sintéticos, los resultados obtenidos por nuestro método superaron a los obtenidos por varios algoritmos de *biclustering* de la literatura, especialmente en el caso de *biclusters* coherentes con alto grado de solapamiento. Sin embargo, esto no puede considerarse como una desventaja debido a que como hemos visto, en general, la complejidad regulatoria de un organismo esta lejos del modelo de *biclusters* sin solapamiento. Además, se realizó un análisis sobre datos reales y, en términos de la métrica propuesta, la calidad de los resultados de BiHEA son claramente superiores a los resultados de los métodos de referencia. De hecho, esto muestra la correctitud en el diseño del modelo para construir los *biclusters*, es decir, *bicluster* coherentes que siguen un modelo aditivo.

Estos buenos resultados llevaron al desarrollo de un software integrador para el análisis de datos de *microarrays,* llamado BAT, que permite la utilización del algoritmo BiHEA en un entorno de interfaz gráfica amigable con numerosas características interesantes para los usuarios biólogos. Entre las principales aptitudes de esta herramienta podemos mencionar las facilidades brindadas para el manejo de datos de *microarrays*, las herramientas de pre-procesamiento, los mecanismos de visualización tanto de datos de *microarrays* como de resultados, y las herramientas de post-procesamiento que permiten realizar varios análisis sobre los resultados obtenidos de manera automática.

Si bien el análisis de co-expresión de genes representa un desafío importante, la información obtenida por medio de *biclustering* es en general insuficiente para reconstruir una red regulatoria de genes. Esto es debido a que solo se infieren asociaciones no causales entre genes sin considerar, además, la posibilidad de que la interacción tenga un cierto retardo, debido a que los procesos bioquímicos no ocurren de forma instantánea. En tal sentido, se realizo una extensa revisión de una familia específica de algoritmos para extraer reglas de asociaciones entre genes, ya que la ingeniería inversa de redes regulatorias de genes a partir de reglas de asociación tiene una ventaja metodológica importante: permite la reconstrucción de redes de modelo libre. En otras palabras, estas técnicas no requieren, en general, ninguna restricción o conocimiento previo sobre las relaciones estructurales de la red, ni hacen suposiciones relacionadas con los

principios fisicoquímicos que rigen las interacciones entre genes. Estos métodos sólo necesitan información de expresión de genes como fuente de datos para el proceso de inferencia. En cuanto a las metodologías de inferencia revisadas, se ilustró una amplia variedad de técnicas, como métodos basados en conjuntos de elementos frecuentes, árboles de clasificación y regresión, redes bayesianas, redes booleanas, máquinas de soporte vectorial, enfoques de *clustering* y algunos algoritmos por pares. Para la mayoría de estos enfoques de minería de datos, se revisaron varios algoritmos, haciendo hincapié en sus ventajas y limitaciones. Finalmente, otro punto relevante que se trató en esta contribución es la inferencia de asociaciones temporales entre los genes. Este punto fue abordado de manera transversal a lo largo de la presentación, ilustrando de que manera los diferentes métodos de minería de datos consideran este tipo de reglas diferidas en el tiempo. También se analizaron en detalle temas adicionales como el modelado de la cardinalidad de las reglas, la validación estadística y biológica de la red, y la extracción de asociaciones a partir de múltiples fuentes de datos.

Luego de ese análisis minucioso, se diseño un nuevo algoritmo, llamado GRNCOP2, tomando como base un método de extracción de reglas de asociación entre genes previamente propuesto por nuestro grupo de investigación (GRNCOP). Esto conllevo a un rediseño total del algoritmo, manteniendo solo las ideas relacionadas con los umbrales adaptativos de regulación y la optimización combinatorial de los clasificadores de reglas. Así, el nuevo algoritmo constituyó una evolución relevante del método anterior, debido a los desafíos que impusieron las mejoras propuestas. El nuevo enfoque incorporó características novedosas tales como la inferencia de reglas con múltiples retrasos de tiempo y en un número ilimitado de conjuntos de datos de series de tiempo, además de mejoras sobre todo el proceso de inferencia. Esta última característica quedo demostrada por el hecho de que los resultados obtenidos por GRNCOP2 son significativamente mejores que los resultados obtenidos por la versión anterior. Además, la relevancia del nuevo método se hizo más evidente ya que los puntajes obtenidos por GRNCOP2 fueron superiores a los obtenidos por otros algoritmos relacionados en términos de las métricas propuestas. En adición, las relaciones inferidas por GRNCOP2 demostraron ser biológicamente relevantes. Es mas, GRNCOP2 fue capaz de obtener nuevas posibles interacciones entre genes, en consonancia con conocimientos biológicos previos, que no fueron descubiertas por los otros métodos. Por otro lado, también se evaluó la capacidad de GRNCOP2 para realizar estudios a nivel de genomas completos. En este sentido, se realizó un análisis sobre varios conjuntos de datos de series de tiempo de genomas completos, para los que se discutió el buen funcionamiento del algoritmo en términos de las métricas propuestas. Además, con la realización de un análisis ontológico se mostró que los resultados fueron significativos en

términos biológicos, ya que se encontró que los genes de las sub-redes descubiertas están altamente relacionados en términos estadísticos.

Como ultima contribución al trabajo desarrollado en esta tesis, podemos mencionar a la plataforma de software llamada GeRNet, que representa un marco de trabajo amigable al usuario integrando dos algoritmos recientemente publicados, proveyendo la oportunidad para realizar ingeniería inversa de redes regulatorias de genes diferidas en el tiempo basadas en reglas de asociación. También ofrece varias características para manejo de datos, visualización y manipulación de redes regulatorias de genes, creando un entorno bioinformático completo para el biólogo investigador.

Por último, una pregunta importante probablemente permanezca en la mente del lector: ¿qué tan verosímiles pueden ser estos modelos regulatorios discretos? En este punto, como corolario de esta tesis, es importante señalar que en cualquier metodología de modelado de redes, se sabe y acepta que un modelo describe sólo algunas de las propiedades del sistema biológico real, y hace caso omiso a muchas otras. En otras palabras, un modelo hace hincapié en aspectos particulares de la realidad, dejando de lado los detalles que no siempre son relevantes para el propósito del estudio. En este contexto, los algoritmos de inferencia desarrollados en esta tesis constituyen una valiosa herramienta para entender y descubrir posibles relaciones ocultas entre los genes, y la discretización de los valores de expresión génica en estados permite al modelador centrase en los patrones de información pertinentes. Por lo tanto, la visión discreta de los datos puede ayudar a capturar el comportamiento de los genes de una manera más sencilla de interpretar. Sin embargo, una reconstrucción realista de los complejos mecanismos de regulación que se producen en la célula requerirá atacar el problema desde diferentes perspectivas, y con métodos computacionales complementarios. Finalmente, la validación biológica definitiva de cada nueva asociación será siempre necesaria con el fin de obtener una red viable y confidente.

8.2. Investigaciones Futuras

Como extensión natural a las investigaciones realizadas en esta tesis, se proyecta ver la posibilidad de continuar mejorando los métodos desarrollados, incorporando nuevas metodologías que ayuden a refinar la información extraída y a incorporar nuevo conocimiento en los modelos inferidos. Una de las características a mejorar está relacionada con la integración de datos de serie de tiempo de distinto intervalo temporal entre muestras. Los trabajos desarrollados no consideran la presencia de esta característica en los datos, lo que dificulta la interpretación de las reglas diferidas en el tiempo cuando estas se extraen a partir de múltiples conjuntos de datos.

Por otro lado, los algoritmos desarrollados solo infieren relaciones de co-expresión entre un conjunto de genes, o reglas entre un gen regulador y un gen blanco. Si bien es sabido que la cantidad de genes reguladores que pueden afectar a un dado gen blanco esta acotada, este tipo de interacciones son altamente probables en términos biológicos, sobre todo en organismos eucariotas. Estas relaciones que involucran a múltiples genes reguladores son solo inferidas de forma parcial por las estrategias desarrolladas, ya que la información inferida no permite determinar si todos los genes reguladores son "necesarios" (relación de *AND* lógico) para el cambio en el estado del gen blanco.

Finalmente, se prevé explorar la integración en el proceso de inferencia de conocimiento biológico externo correspondiente a bases de datos como KEGG y *Gene Ontology*, e incluso integrar otras fuentes de datos biológicos que no necesariamente representen niveles de expresión de genes, a fin de refinar aun más los conocimientos extraídos.

Referencias

Agrawal, R. and Srikant, R. Fast Algorithms for Mining Association Rules. In*: Proc. VLDB Conf,* 1994.

Agrawal, R., Imielinski, T. and Swami, A. Mining association rules between sets of items in large databases. In: *Proceedings of the 1993 ACM-SIGMOD International Conference on Management of Data,. Washington, DC, USA: ACM Press*, 207-216, 1993.

Akutsu, T., Miyano, S. and Buhara, S. Identification of Genetic Networks from a Small Number of Gene Expression Patterns under the Boolean Network Model. *Proc. Pacific Symp. Biocomputing Jan*., 4:17-28, 1998.

Akutsu, T., Miyano, S. and Kuhara, S. Algorithms For Inferring Qualitative Models of Biological Networks. *In: Pacific Symposium on Biocomputing (PSB)*, 293-304, 2000.

Alizadeh, A.A., et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503-511, 2000.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D. and Levine, A. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci*, 96:6745-6750, 1999.

Althoefer, H., Schleiffer, A., Wassmann, K., Nordheim, A. and Ammerer, G. McmI Is Required to Coordinate G2-Specific Transcription in Saccharomyces cerevisiae. *Molecular and Cellular Biology,* 15:5917-5928, 1995.

Alves, R., Rodriguez-Baena, D.S. and Aguilar-Ruiz, J.S. Gene association analysis: a survey of frequent pattern mining from gene expression data. *Brief Bioinform*, 11(2):210-24, 2010.

Amaratunga, D. and Cabrera, J. *Exploration and analysis of DNA microarray and protein array data.* Wiley Series in Probability and Statistics, 2003.

Amon, A., Tyers, M., Futcher, B. and Nasmyth, K. Mechanisms that Help the Yeast Cell Cycle Clock Tick: G2 Cyclins Transcriptionally Activate G2 Cyclins and Repress G1 Cyclins. *Cell,* 74:993-1007, 1993.

Andrews, B. and Measday, V. The Cyclin Family of Budding Yeast: Abundant Use of a Good Idea. *Trends in Genetics,* 14:66-72, 1998.

Arkin, A., Shen, P. and Ross, J. A Test Case of Correlation Metric Construction of A Reaction Pathway from Measurements. *Science*, 277:1275-1279, 1997.

Arslan, T., Horrocks, D.H. and Ozdemir, E. Structural Synthesis of Cell-based VLSI Circuits using a Multi-Objective Genetic Algorithm, *IEE Electronic Letters*, 32(7):651-652, 1996.

Barabasi, A.L. and Oltvai, Z.N. Network biology: Understanding the cell's functional organisation. *Nat. Rev. Genetics*, 5(2):101-113, 2004.

Baralis, E., Bruno, G. and Ficarra, E. Temporal association rules for gene regulatory networks. In: *Proceedings of the 4th International IEEE Conference Sep,* 2-7, 2008.

Bar-Josephm, Z., Gerber, G.K., Lee, T.I., Rinaldi, N.J., Yoo, J.Y., Robert, F., Gordon, D.B., Fraenkel, E., Jaakkola, T.S., Young, R.A. and Gifford, D.K. Computational discovery of gene modules and regulatory networks. *Nat Biotechnol*, 21:1337-42, 2003.

Barkow, S., Bleuler, S., Prelic, A., Zimmermann, P., and Zitzler, E. BicAT: a biclustering analysis toolbox. *Bioinformatics*, 22(10):1282-1283, 2006.

Barrett, T., Edgar, R. Gene expression omnibus: microarray data storage, submission, retrieval, and analysis. *Methods Enzymol,* 411:352-369, 2006.

Bayardo, R.J. Efficiently mining long patterns from databases. In: *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data. Seattle, Washington, USA: ACM Press,* 88-93, 1998.

Beal, M.J., Falciani, F., Ghahramani, Z., Rangel, C. and Wild, D.L. A Bayesian Approach to Reconstructing Genetic Regulatory Networks with Hidden Factors. *Bioinformatics*, 21(3):349-356, 2005.

Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J.F. and Gandrillon, O. Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human sage data. *Genome Biology*, 3(12), 2002.

Ben-Dor, A., Chor, B., Karp, R., and Yakhini, Z. Discovering Local Structure in Gene Expression Data: The Order-Preserving Submatrix Problem. *Proc. Sixth Int'l Conf. Computational Biology (RECOMB '02)*, 49-57, 2002.

Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Rapp, B.A., and Wheeler, D.L. Genbank. *Nucl. Acids Res*., 30:17-20, 2002.

Bernard, A. and Hartemink, A.J. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. *Pac Symp Biocomput*, 459-70, 2005.

Bleuler, S., Laumanns, M., Thiele, L. and Zitzler, E. PISA - A Platform and Programming Language Independent Interface for Search Algorithms. *In: Proceeding of Evolutionary Multi-Criterion Optimization*, 494-508, 2003.

Bleuler, S., Prelic, A. and Zitzler, E. An EA framework for biclustering of gene expression data. *In: Proceeding of Congress on Evolutionary Computation*, 166-173, 2004.

Boguski, M.S., Lowe, T.M., Tolstoshev, C.M. dbEST-database for "expressed sequence tags". *Nat. Genet*., 4(4):332-333. Aug 1993.

Bolstad, B.M., Irizarry, R.A., Astrand, M. and Speed, T.P. A comparison of normalization methods for high desity oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185-193, 2003.

Boyen, X., Friedman, N., Koller and D. Discovering the Hidden Structure of Complex Dynamic Systems. In: *Uncertainty in Artificial Intelligence*, 1999.

Brazma, A. and Vilo, J. Minireview. Gene epxression data analysis. *Federation of European Biochemical societies*, 280:17-24, 2000.

Breiman, L. Random forests. *Machine Learning*, 45:5-32, 2001.

Breiman, L., Friedman, J.H., Olsen, R.A. and Stone, C.J. *Classification and Regression Trees*. Wadsworth International (California), 1984.

Bremermann, H.J. The Evolution of Intelligence. The nervous system as a model of its environment. *Technical report, no.1, contract no. 477(17), Dept. Mathematics, Univ. Washington, Seattle, July*, 1958.

Brown, M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., Ares, M.Jr. and Haussler, D. Kwonledge-based analysis of microarray gene expression data using support vector machines. *Proc. Natl. Acad. Sci.*, 97(1):262-267, 2000.

Bulashevska, S. and Eils, R. Inferring Genetic Regulatory Logic from Expression Data. *Bioinformatics,* 21:2706-2713, 2005.

Carmona-Saez, P., Chagoyen, M., Rodriguez, A., Trelles, O., Carazo, J.M. and Pascual-Montano, A. Integrated analysis of gene expression by association rules discovery. *BMC Bioinformatics*, 7:54, 2006.

Carmona-Saez, P., Chagoyen, M., Rodriguez, A., Trelles, O., Carazo, J.M. and Pascual-Montano, A. Integrated analysis of gene expression by association rules discovery. *BMC Bioinformatics*, 7:54, 2006.

Ceglar, A. and Roddick, J.F. Association mining. *ACM Comput Surv*, 38:2, 2006.

Chen, J.J., Wu, R., Yang, P.C, Huang, J.Y., Sher, Y.P., Han, M.H., Kao, W.C., Lee, P.J., Chiu, T.F., Chang, F., Chu, Y.W., Wu, C.W. and Peck, K. Profiling expression patterns and isolating differentially expressed genes by cDNA microarray system with colorimetry detection. *Genomics*, 51:313-324, 1998.

Chen, K.C., Csikasz-Nagy, A., Gyorffy, B., Val, J., Novak, B. and Tyson, J.J. Kinetic Analysis of a Molecular Model of the Budding Yeast Cell Cycle. *Molecular Biology of the Cell,* 11:369-391, 2000.

Chen, T., Filkov, V. and Skiena, S.S. Identifying Gene Regulatory Networks from Exprimental Data. In: *International Conference on Research in Computational Molecular Biology (RECOMB),* 94-103, 1999.

Cheng, Y. and Church, G.M. Biclustering of Expression Data. *Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology (ISMB '00)*, 93-103, 2000.

Cheung, V.G., Morley, M., Aguilar, F., Massini, A., Kucherlapati, R. and Childs, G. Making and reading microarrays. *Nature Genetics*, 21(Suppl.):15-19, 1999.

Cho, K.H., Choo, S.M., Jung, S.H., Kim, J.R., Choi, H.S. and Kim, J. Reverse engineering of gene regulatory networks. *IET Syst Biol*, 1:149-63, 2007.

Cho, R., et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, 2(1):65-73, 1998.

Cinquin, O. and Demongeot, J. Positive and Negative Feedback: Striking a Balance Between Necessary Antagonists. *Journal of Theoritical Biology*, 216:229-241, 2002.

Cleveland, W.S. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Stat. Assoc.*, 74:829-836, 1979.

Coello Coello, C.A., Van Veldhuizen, D. and Lamont, G. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.

Collins, F.S. Microarrays and macroconsequences, *Nature genetics*, 21(1):2, 1999.

Conover, W. *Practical Nonparametric Statistics*. John Wiley & Sons, 1999.

Creighton, C. and Hanash S. Mining gene expression databases for association rules. *Bioinformatics*, 19:79-86, 2003.

Crick, F. Central dogma of molecular biology. *Nature*, 227(5258):561-563, Aug. 1970.

Csete, M.E. and Doyle, J,C. Reverse engineering of biological complexity. *Science* , 295:1664-9, 2002.

Cui, Q., Liu, B., Jiang, T. and Ma, S. Characterizing the dynamic connectivity between genes by variable parameter regression and kalman filtering based on temporal gene expression data. *Bioinf.*, 21(8):1538-1541, 2005.

Curtis, H., Barnes, H.S. and Schnek, A. *Biologia*. Médica Panamericana, 2000.

Das, D.B. and Patvardhan, C. New Multi-objective Stochastic Search Technique for Economic Load Dispatch. *IEEE Proceedings on Generation, Transmission and Distribution*, 145(6):747-752, 1998.

de Jong, H. Modeling and simulation of genetic regulatory systems: A literature review. *J. Comp. Bio.*, 9(1):67-103, 2002.

Deb, K., Agraval, S., Pratap, A. and Meyarivan, T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *In: M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo Guervós, H-P. Schwefel (Eds.): PPSN VI. LNCS*, 1917:849-858, 2000.

DeRisi, J., Penland, L., Brown, P.O., Bittner, M.L, Meltzer, P.S., Ray, M., Chen, Y., Su, Y.A. and Trent, J.M. Use of cDNA microarray to analyse gene expression patterns in human cancer. *Nature Generics*, 14:457-460, 1996.

Devijver, P.A. and Kittler, J. *Pattern Recognition: A Statistical Approach*, Prentice-Hall, London, 1982.

Dimitrova, E.S., Licona, M.P., McGee, J. and Laubenbacher, R. Discretization of Time Series Data. *J Comput Biol*. 17(6):853-68, 2005.

Divina, F. and Aguilar-Ruiz, J.S. Biclustering of Expression Data with Evolutionary Computation. *IEEE Trans. Knowl. Data Eng*, 18(5):590-602, 2006.

Dougherty, J., Kohavi, R. and Sahami, M. Supervised and unsupervised discrimination of continuous Features. In Prieditis, A. and Russell, S. (eds.), *Machine learning: Proceedings of the 12th International Conference, Morgan Kauffman, San Francisco, CA*, 1995.

Draghici, S. *Data analysis tools for DNA microarrays*. Chapman & Hall/CRC, 2003.

Draghici, S., Khatri, P., Bhavsar, P., Shah, A., Krawetz, S. and Tainsky, M. Onto-Tools, the toolkit of the modern biologist: Onto-Express, Onto-Compare, Onto-Design, and Onto-Translate. *Nuc Acids Res*, 31(13):3775-3781, 2003.

Driscoll, M.E. and Gardner, T.S. Identification and control of gene networks in living organisms via supervised and unsupervised learning. *J. Process Control*, 16(3):303–311, 06.

Duggan, D.J., Bittner, M., Chen, Y., Meltzer, P. and Trent, J.M. Expression profiling using cDNA microarrays. *Nat. Genet*., 21(1 Suppl):10-14, 1999.

Dwight, S.S., Harris, M.A., Dolinski, K., Ball, C.A., Binkley, G., Christie, K.R., Fisk, D.G., Issel-Tarver, L., Schroeder, M., Sherlock, G., Sethuraman, A., Weng, S., Botstein, D. and Cherry, J.M. Saccharomyces Genome Database (SGD) provides secondary gene annotation using the Gene Ontology (GO). *Nucleic Acids Res,* 30:69-72, 2002.

Ekins, R.P. and Chu, R.W. Microarrays: their origins and applications. *Trends in Biotechnology*, 17:217-218, 1999.

Eklund, N.H. and Embrechts, M.J. GA-Based Multi-Objective Optimization of Visible Spectra for Lamp Design. In Cihan H. Dagli, Anna L. Buczak, Joydeep Ghosh, Mark J. Embrechts, and Okan Ersoy, editors*, Smart Engineering System Design: Neural Networks,*

Fuzzy Logic, Evolutionary Programming, Data Mining and Complex Systems, 451-456, 1999.

Epstein, C.B. and Cross, F.R. CLB5: A Novel B Cyclin from Budding Yeast with a Role in S Phase. *Genes and Development,* 6:1695-1706, 1992.

Erdal, S., Ozturk, O., Armbruster, D., Ferhatosmanoglu, H. and Ray, W.C. A time series analysis of microarray data. In: *Proceeding of the 4rd IEEE Symposium on Bioinformatics and Bioengineering*, 366-374, 2004.

Fang, H.L., Ross, P. and Corne, D. A Promising Genetic Algorithm Approach to Job-Shop Scheduling, Rescheduling and Open Shop Scheduling Problems. In Forrest S. editor. *Proceedings of the Fifth International Reference on Genetic Algorithms, San Mateo: Morgan Kaufmann Publishers*, 375-382, 1993,

Farabee, M.J. *On-Line Biology Book*. http://www.emc.maricopa.edu/faculty/farabee/BIOBK/BioBookTOC.html

Fitch, W.M. and Margoliash, E. Construction of phylogenetic trees. *Science*, 155:279-284, 1967.

FitzGerald, P.C., Sturgill, D., Shyakhtenko, A., Oliver, B. and Vinson, C. Comparative genomics of drosophila and human core promoters. *Genome Bio*., 7:R53, 2006.

Fodor, S.P., Read, J.L., Pirrung, M.C., Stryer, L., Lu, A.T. and Solas, D. Light-directed, spatially addressable parallel chemical synthesis. *Science*, 251 (4995):767-773, 1991.

Fogel, L.J., Owens, A.J. and Walsh, M.J. *Artificial Intelligence through Simulated Evolution*. Wiley, 1966.

Fogelberg, C. and Palade, V. Machine Learning and Genetic Regulatory Networks: A Review and a Roadmap. In: Abraham, A., Hassanien, A.E., Vasilakos, A., Pedrycz, W., Herrera, F., Siarry, P., de Carvalho, A. and Engelbrecht, A.P. (Eds.), *Foundations of Computational Intelligence. Springer Berlin Heidelberg, Heidelberg*, 3-34, 2009.

Friedberg, R.M. A Learning Machine. *IBM Journal, Research and Development*, 3(2):183-191, 1959.

Friedman, N. and Goldszmidt M. Discretization of continuous attributes while learning Bayesian networks. In Saitta, L. (ed.), *Proc. Of the 13th International Conference on Machine Learning, Morgan Kauffman, San Francisco, CA*, 157-165, 1996.

Friedman, N. Inferring Cellular Networks Using Probabilistic Graphical Models. *Science*, 303(6):799-805, 2004.

Friedman, N. The Bayesian Structure EM Algorithm. In: *Uncertainty in Artificial Intelligence*, 129-138, 1998.

Friedman, N., Linial, M., Nachman, I. and Pe'er, D. Using Bayesian networks to analyze expression data. *J Comput Biol*, 7:601-20, 2000.

Gallo, C., Carballido, J.A. and Ponzoni, I. Microarray Biclustering: A Novel Memetic Approach Based on the PISA Platform, *Lecture Notes in Computer Scienc*e, 5483:44-55, 2009a.

Gallo, C.A., Carballido, J.A. and Ponzoni, I. BiHEA: A Hybrid Evolutionary Approach for Microarray Biclustering. *Lecture Notes in Computer Science*, 5676:36-47, 2009b.

Gallo, C.A., Carballido, J.A. and Ponzoni, I. Discovering Time-lagged rules from microarray data using gene profile classifiers. *BMC Bioinformatics*, 12(123):1-21, 2011a.

Gallo, C.A., Carballido, J.A., Ponzoni, I. GeRNet: A Framework for Inference, Visualization and Manipulation of Gene Regulatory Networks based on Association Rules. *VI Congreso Argentino de Bioinformática y Biología Computacional, 29-31 de octubre de 2013, Rosario, Argentina*, 2013b.

Gallo, C.A., Carballido, J.A., Ponzoni, I. Inference of gene regulatory networks based on association rules. In: Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data. *Wiley*, 2013a

Gallo, C.A., Carballido, J.A., Ponzoni, I. Inferring Time-Lagged Association Rules from Microarray Time-Series Data. *SolBio 2010 (1st International Conference on Bioinformatics), 26 al 28 de Septiembre de 2010, Termas de Chillan, Chile*, 2010c.

Gallo, C.A., Carballido, J.A., Ponzoni, I. Inferring Time-Lagged Association Rules from Microarray Time-Series Data. *II Congreso Argentino de Bioinformática y Biología Computacional, 11-13 de mayo de 2011, Córdoba, Argentina*, 2011b.

Gallo, C.A., Dussaut, J., Carballido, J.A., Ponzoni, I. BAT: A new Biclustering Analysis Toolbox, In: *Ferreira, C.E.; Miyano, S.; Stadler, P.F. (eds.) BSB 2010. Lecture Notes in Computer Science*, 6268:67-71, 2010a.

Gallo, C.A., Dussaut, J.S., Carballido, J.A., Ponzoni, I. A Microarray Biclustering Analysis Tool based on the BiHEA Algorithm. *I Congreso Argentino de Bioinformática y Biología Computacional, 12-14 de mayo de 2010, Quilmes, Argentina*, 2010b.

Geisser, S. *Predictive Inference*. Chapman and Hall (New York), 1993.

Geurts, P., Ernst, D., Wehenkel, L. Extremely randomized trees. *Machine Learning,* 36:3-42, 2006.

Giaever, G., Chu, A.M. and Ni, L. *et at*. Functional profiling of the Saccharomyces cerevisiae genome. *Nat.*, 418(6896):387-91, 2002.

Gouda, K. and Zaki, M.J. GenMax: an efficient algorithm for mining maximal frequent itemsets. *Data Min Knowl Discov*, 11:223-42, 2005.

Gransson, L. and Koski, T. Using a Dynamic Bayesian Network to Learn Genetic Interactions. *Technical Report, Graduate School of Biomedical Research, Linkoping University,* 2002.

Grignon, P., Wodziack, J. and Fadel, G.M. Bi-Objective Optimization of Components Packing using a Genetic Algorithm. In *NASA/AIAA/ISSMO Multidisciplinary Design and Optimization Conference*, 352-362, 1996.

Guelzim, N., Bottani, S., Bourgine, P. and Kepes, F. Topological and causal structure of the yeast transcriptional regulatory network. *Nat Genet*, 31:60-3, 2002.

Guthke, R., Moller, U., Hoffmann, M., Thies, F. and Topfer, S. Dynamic Network Reconstruction from Gene Expression Data Applied to Immune Response During Bacterial Infectioin. *Bioinformatics*, 21(8):1626-1634, 2005.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46:389-422, 2002.

Han, J. and Kamber, M. *Data Mining: Concept and Techniques*. The morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2000.

Han, J., Cheng, H., Xin, D. and Yan, X. Frequent pattern mining: current status and future directions. *DataMin Knowl Discov*, 15:55-86, 2007.

Han, J., Pei, J., Yin, Y. Mining frequent patterns without candidate generation. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. Dallas, Texas, USA: ACM Press*, 1–12, 2000.

Hansen, M. and Jaszkiewicz A. Evaluating the quality of approximations to the non-dominated set. *Technical University of Denmark*, 1998.

Hartemink, A. Principled computational methods for the validation and discovery of genetic regulatory networks. *Massachusetts Institute of Technology, Ph. D. dissertation*, 2001.

Hartemink, A.J., Gifford, D.K., Jaakkola, T.S. and Young, R.A. Combining location and expression data for principled discovery of genetic regulatory network models. *Pacific Symp. on Biocomp.*, 437-449, 2002.

Hartemink, A.J., Gifford, D.K., Jaakkola, T.S. and Young, R.A. Using Graphical Models and Genomic Expression Data to Statistically Validate Models of Genetic Regulatory Networks. *Proceedings of the Pacific Symposium on Biocomputing'01,* 422-433, 2001.

Hartigan, J.A. Direct Clustering of a Data Matrix. *J. Am. Statistical Assoc,* 67(337):123-129, 1972.

Hashimoto, R., Kim, S., Shmulevich, I., Zhang, W., Bittner, M.L. and Dougherty, E.R. Growing genetic regulatory networks from seed genes. *Bioinformatics*, 20:1241-1247, 2004.

Hasty, J., McMillen, D., Isaacs, F. and Collins, J.J. Computational Studies of Gene Regulatory Networks: In Numero Molecular Biology. *Nature Reviews Genetics*, 2(4):268-279, 2001.

Heller, R.A., Schema, M., Chai, A., Shalon, D., Bedilion, T., Gilmore, J., Woolley, D.E. and Davis, R.W. Discovery and analysis of inflammatory disease-related genes using cDNA microarrays. *Proc. Natl. Acad. Sci. USA*, 94:2150-2155, 1997.

Herrgård, M.J., Covert, M.W. and Palsson, B. Reconciling gene expression data with known genome-scale regulatory network structures. *Genome Research*, 13(11):2423-2434, 2003.

Hipp, J., Gntzer, U. and Nakhaeizadeh, G. Algorithms for association rule mining a general survey and comparison. *ACM SIGKDD Exploration*, 2(Issue 1):58-64, 2000.

Holland, J.H. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1975.

Husmeier, D. Sensitivity and Specificity of Inferring Genetic Regulatory Interactions From Microarray Experiments with Dynamic Bayesian Networks. *Bioinformatics*, 19(17):2271-2282, 2003.

Hwang, L.H., Lau, L.F., Smith, D.L., Mistrot, C.A., Hardwick, K.G., Hwang, E.S., Amon, A. and Murray, A.W. Budding Yeast CDC20: A Target of the Spindle Checkpoint. *Science,* 279:1041-1044, 1998.

Igual, J.C., Toone, W.M. and Johnston, L.H. A Genetic Screen Reveals a Role for the Late G1-Specific Transcription Factor Swi4p in Diverse Cellular Functions Including Cytokinesis. J. *Cell Science,* 110:1647-1654, 1997.

Ihmels, J., Bergmann, S. and Barkai, N. Defining transcription modules using large-scale gene expression data. *Bioinformatics*, 20(13):1993-2003, 2004.

Ihmels, J., Friedlander G., Bergmann S., Sariq, O., Ziv, Y. and Barkai N. Revealing modular organization in the yeast transcriptional network. *Nat. Genet*., 31:370-377, 2002.

Imoto, S., Goto, T. and Miyano, S. Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression. *Pac Symp Biocomput*, 175-86, 2002.

Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S. and Miyano, S. Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. *J Bioinform Comput Biol*, 2:77-98, 2004.

Iyer, V.R., Eisen, M.B., Ross, D.T, Schuler, G., Moore, T., Lee, J.C.F., Trent, J.M., Staudt, L.M., Hudson, Jr.J., Boguski, M.S., Lashkari, D., Shalon, D., Botstein, D. and Brown, P.O. The transcriptional program in the response of human fibroblast to serum. *Science*, 283:83-87, 1999.

Jain, A. and Dubes, R. *Algorithms for clustering data*. Prentice Hall. 58-89, 1988.

Jenssen, T.K., Laegreid, A., Komorowski, J. and Hovig, E. A literature network of human genes for high-throughput analysis of gene expression. *Nat Genet*, 28:21–28, 2001.

Ji, L. and Tan, K Identifying time-lagged gene clusters using gene expression data. *Bioinformatics*, 21(4):509-516, 2005.

Ji, L. and Tan, K. Mining gene expression data for positive and negative co-regulated gene clusters. *Bioinformatics*, 20(16):2711-2718, 2004.

Jiang, D., Tang, C. and Zhang, A. Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1370-1386, 2004.

Jin, Y., Olhofer, M. and Sendhoff, B. Evolutionary Dynamic Weighted Aggregation (EDWA): Why does it work and how?. *Proceedings of Genetic and Evolutionary Computation Conference, San Francisco*, USA, 2001.

Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K.F., Itoh, M., Kawashima, S., Katayama, T., Araki, M. and Hirakawa, M. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res*, D354-357, 2006.

Karlebach, G. and Shamir, R. Modelling and analysis of gene regulatory networks. *Nat Rev Mol Cell Biol,* 9:770-780, 2008.

Karp, G. *Cell and Molecular Biology; Concepts and Experiments*. John Wiley & Sons Inc, 2002.

Kauffman, S., Peterson, C., Samuelsson, B. and Troein, C. Random Boolean network models and the yeast transcriptional network. *Proc. Natl Acad. Sci. USA 100*, 14796-14799, 2003.

Kauffman, S.A, Requirements for Evolvability in Complex Systems: Orderly Dynamics and Frozen Components. *Physica D*, 42:135-152, 1990.

Kauffman, S.A. Antichaos and adaptation. *Scientific American*, 265(2):78-84, 1991.

Kauffman, S.A. Metabolic stability and epigenesist in randomly constructed genetic nets. *J. Theor. Biol*. 22:437-467. 1969.

Kauffman, S.A. *The Origins of Order, Self-Organization and Selection in Evolution*. New York, 1993.

Kim, S., Imoto, S. and Miyano, S. Dynamic Bayesian Network and Nonparametric Regression for Nonlinear Modeling of Gene Networks from Time Series Gene Expression Data. *Biosystems*, 75:57-65, 2004.

Kitano, H. Perspectives on systems biology. *New Generat Comput*, 18:199-216, 2000.

Klebanov, L. and Yakovlev, A. How high is the level of technical noise in microarray data? *Bio. Direct*, 2:9, 2007.

Knowles, J., Thiele, L. and Zitzler, E. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. *TIK Computer Engineering and Networks Laboratory*, 2005.

Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence,* 2(12):1137-1143, 1995.

Kohavi, R:.Wrappers for performance enhancement and oblivious decision graphs. *PhD thesis, Stanford University, Computer Science Department*, 1995.

Koranda, M., Schleiffer, A., Endler, L. and Ammerer, G. Forkhead-Like Transcription Factors Recruit NddI to the Chromatin of G2/M-Specific Promoters. *Nature,* 406:94-98, 2000.

Koyuturk, M., Szpankowski, W. and Grama, A. Biclustering gene-feature matrices for statistically significant dense patterns. In: *Proceedings of the 8th Annual International Conference on Research in Computational Molecular Biology*, 480-484, 2004.

Kuhn, H.W. and Tucker, A.W. Nonlinear Programming. *Proc. Of Sec. Berk. Symp. On Math Stat. And Prob*, 481-492, 1951.

Kuhne, C. and Linder, P. A New Pair of B-Type Cyclins from Saccharomyces cerevisiae that Function Early in the Cell Cycle. *European Molecular Biology Organization J*., 12:3437-3447, 1993.

Kwon, A., Hoos, H. and Ng, R. Inference of transcriptional regulation relationships from gene expression data. *Bioinformatics*, 19(8):905-912, 2003.

Lähdesmäki, H., Shmulevich, I. and Yli-Harja, O. On learning gene regulatory networks under the Boolean network model. *Machine Learning,* 52:147-167, 2003.

Lai, L.C., Kosorukoff, A.L., Burke, P.V. and Kwast, K.E. Dynamical remodeling of the transcriptome during short-term anaerobiosis in Saccharomyces cerevisiae: differential response and role of Msn2 and/or Msn4 and other factors in galactose and glucose media. *Mol Cell Biol,* 25(Suppl 10):4075-91, 2005.

Lander, E.S. Array of hope. *Nat Genet*., 21 (1 Suppl):3-4, Jan 1999.

Laubenbacher, R. and Stigler, B. A computational algebra approach to the reverse engineering of gene regulatory networks. *J. Theor. Biol.*, 229:523-537, 2004.

Laule, O., et al. Crosstalk between cytosolic and plastidial pathways of isoprenoid bio systhesis in arabidopsis thaliana. *PNAS,* 100(11):6866-6871, 2003.

Lee, A.K., Sung, S.H., Kim, Y.C. and Kim, S.G. Inhibition of Lipopolysaccharide-Inducible Nitric Oxide Synthase TNF-α and COX-2 Expression by Sauchinone Effects on NF-κB1 Phosphorylation, C/EBP and AP-1 Activation. *British Journal of Pharmacology*, 139:11-20, 2003.

Lee, I., Date, S.V., Adai, A.T. and Marcotte, E.M. A probabilistic functional network of yeast genes. *Science*, 306:1555–1558, 2004.

Lee, I., Li, Z. and Marcotte, E.M. An improved, bias-reduced probabilistic functional gene network of baker's yeast, Saccharomyces cerevisiae. *PLoS ONE*, 2(Suppl 10):e988, 2007.

Lee, P.H. and Lee, D. Modularized Learning of Genetic Interaction Networks from Biological Annotations and mRNA Expression Data. *Bioinformatics*, 21(11):2739-2747, 2005.

Levine, K., Huang, K. and Cross, F.R. Saccharomyces cerevisiae G1 Cyclins Differ in Their Intrinsic Functional Specificities. *Molecular and Cellular Biology*, 16:6794-6803, 1996.

Lewin, B. *Genes VIII*. Prentice Hall, 2003.

Lewin, B. *Genes*. Oxford University Press, 7 edition, 1999.

Li, H., Whitmore, J., Suh, E., Bittner, M. and Kim, S. Learning context-sensitive Boolean network from steadystate observations and its analysis. *Research in Computational Molecular Biology* , 2004.

Li, H., Xuan, J., Wang, Y. and Zhan, M. Inferring regulatory networks. *Front Biosci*, 13:263-275, 2008.

Li, J., Li, X., Su, H., Chen, H. and Galbraith, D.W. A framework of integrating gene relations from heterogeneous data sources: an experiment on Arabidopsis thaliana. *Bioinformatics*, 22:2037-2043, 2006.

Li, X., Rao, S., Jiang, W., Li, C., Xiao, Y., Guo, Z., Zhang, Q., Wang, L., Du, L., Li, J., Li, L., Zhang, T. and Wang, Q.K. Discovery of Time-Delayed Gene Regulatory Networks based on temporal gene expression profiling. *BMC Bioinformatics,* 7:26, 2006.

Liang, S., Fuhrman, S. and Somogyi, R. REVEAL, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures. *Proc. Pacific Symp. Biocomputing Jan.*, 3:18-29, 1998.

Lipshutz, R.J., Fodor, S.P.A., Gingeras, T.R. and Lockhart, D.J. High density synthetic oligonucletide arrays. *Nature genetics*, 21(1):20-24, 1999.

Lonardi, S., Szpankowski, W. and Yang, Q. Finding biclusters by random projections. In: *Proceedings of the 15th Annual Symposium on Combinatorial Pattern Matching, Springer*, 102-116, 2004.

Loy, C.J., Lydall, D. and Surana, U. NDDI, a High-Dosage Suppressor of cdc28-I N, Is Essential for Expression of a Subset of Late-S-Phase-Specific Genes in S. cerevisiae. *Molecular and Cellular Biology,* 19:3312-3327, 1999.

Lum, P.Y., Armour, C.D., Stepaniants, S.B., Cavet, G., Wolf, M.K., Butler, J.S., Hinshaw, J.C., Garnier, P., Prestwich, G.D., Leonardson, A., Garrett-Engele, P., Rush, C.M., Bard, M., Schimmack, G., Phillips, J.W., Roberts, C. J. and Shoemaker, D.D. Discovering modes of action for therapeutic compounds using a genome-wide screen of yeast heterozygotes. *Cell*, 116(1):121-137, 2004.

MacQueen, J. Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium of Mathematical Statistics and Probability,* 1:281-297, 1967.

Madeira, S.C. and Oliveira, A.L. A linear time algorithm for biclustering time series expression data. In: *Proceedings of 5th Workshop on Algorithms in Bioinformatics*, 2005.

Madeira, S.C. and Oliveira, A.L. An Evaluation of Discretization Methods for Non-Supervised Analysis of Time-Series Gene Expression Data. *INESC-ID Technical Report*, 2005.

Madeira, S.C. and Oliveira, A.L. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans. on Comp. Biology and Bioinformatics*, 1:24–45, 2004.

Marnellos, G. and Mjolsness, E. A gene network approach to modeling early neurogenesis in drosophila. In: *Pacific Symp. on Biocomp.*, 3:30-41, 1998.

Mclachlan, G.J. *Analyzing Microarrays Gene Expression Data*. Willey, 2004.

McShan, D.C., Updadhayaya, M. and Shah, I. Symbolic inference of xenobiotic metabolism. *World Scientific*, 545-556, 2004.

Mehra, S., Hu, W.S. and Karypis, G. G: A Boolean Algorithm for Reconstructing the Structure of Regulatory Networks. *Metabolic Eng,* 6:326-339, 2004.

Menges, M., Hennig, L., Gruissem, W., Murray, J. Genome-wide gene expression in an Arabidopsis cell suspension. *Plant Mol. Biol,* 53(4):423-442, 2003.

Mirkin, B. *Math. Classification and Clustering*. Springer, 1996.

Mitchel, T. *Machine Learning*. WCB/McGraw-Hill, 1997.

Mitra, S. and Banka, H. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognit*, 39:2464-2477, 2006.

Mollr-Levet, C., Cho, S. and Wolkenhauer, O. Microarray data clustering based on temporal variation: Fcv and tsd preclustering. *Applied Bioinformatics*, 2(1):35-45, 2003.

Mullis, K.B. The unusual origin of the polimerase chain reaction. Scientific American, 256:56-65, 1990.

Murphy, K. and Mian, S. Modelling gene expression data using dynamic Bayesian networks. In: *Division of Computer Science, University of California, Berkeley*, 1999.

Nam, H., Lee, K. and Lee, D. Identification of temporal association rules from time series microarray data sets. *BMC Bioinformatics*, 10:(Suppl 3)**:**S6, 2009.

Nykter, M., Aho, T., Ahdesmäki, M., Ruusuvuori, P., Lehmussola, A. and Yli-Harja, O. Simulation of microarray data with realistic characteristics. *Bioinf*., 7:349, 2006.

Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K. and Ishii, S. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics,* 19(Suppl 16):2088-2096, 2003.

Park, G. and Szpankowski, W. Analysis of biclusters with applications to gene expression data. In: *Proceedings of the 1st International Conference on Analysis of Algorithms,* 2005.

Pasquier, N., Bastide, Y., Taouil, R. and Lakhal, L. Efficient mining of association rules using closed itemset lattices. *Inf Syst*, 24:25-46, 1999.

Pe'er, D., Regev, A., Elidan, G. and Friedman, N. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(Suppl 1):S215-24, 2001.

Peeters, R. The Maximum Edge Biclique Problem is NPComplete. *Discrete Applied Math*, 131(3):651-654, 2003.

Pensa, R.G., Leschi, C., Besson, J. and Boulicaut, J. Assessment of discretization techniques for relevant pattern discovery from gene expression data. In: *4th Workshop on Data Mining in Bioinformatics*, 2004.

Perkins1, T.J., Jaeger, J. and Reinitz, J. Reverse engineering the gap gene network of drosophila melanogaster. *PLoS Comp. Bio*., 2(5):e51, 2006.

Ponzoni, I., Azuaje, F., Augusto, J. and Glass, D. Inferring Adaptive Regulation Thresholds and Association Rules from Gene Expression Data through Combinatorial Optimization Learning. *IEEE/ACM Trans. on Comp. Biology and Bioinformatics*, 4(Suppl 4):624-634, 2007.

Pramila, T., Miles, S., GuhaThakurta, D., Jemiolo, D. and Breeden, L.L. Conserved homeodomain proteins interact with MADS box protein Mcm1 to restrict ECB-dependent transcription to the M/G1 phase of the cell cycle. *Genes Dev,*16(Suppl 23):3034-45, 2002.

Pramila, T., Wu, W., Miles, S., Noble, W.S. and Breeden, L.L. The Forkhead transcription factor Hcm1 regulates chromosome segregation genes and fills the S-phase gap in the transcriptional circuitry of the cell cycle. *Genes Dev,* 20(Suppl 16):2266-2278, 2006.

Pridgeon, C. and Corne, D. Genetic Network Reverse-Engineering and Network Size; Can We Identify Large GRNs?. In: *Proc. 2004 IEEE Symp. Computational Intelligence in Bioinformatics and Computational Biology,* 32-36, 2004.

Prinz, S., Hwang, E.S., Visintin, R. and Amon, A. The Regulation of Cdc20 Proteolysis Reveals a Role for the APC Components Cdc23 and Cdc27 during S Phase and Early Mitosis. *Current Biology,* 8:750-760, 1998.

Purves, W.K., Sadava, D., Orians, G.H and Heller, H.C. *Life: the Science of Biology*. Sinauer Associates Inc. and W.H.FREEMAN and Company, 7 edition, 2003.

Quinlan, J.R. C4.5: *Programs for Machine Learning*. Morgan Kaufmann, 1992.

Rechenberg, R. I. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog, 1973.

Rogers, S. and Girolami, M. A Bayesian Regression Approach to the Inference of Regulatory Netowkrs from Gene Expression Data. *Bioinformatics*, 21(14):3131-3137, 2005.

Ronen, M. and Botstein, D. Transcriptional response of steady-state yeast cultures to transient perturbations in carbon source. *Proc Natl Acad Sci,*103(Suppl 2):389-394, 2006.

Rosenberg R.S. *Simulation of Genetic Populations with Biochemical Properties*. PhD thesis, University of Michigan, Ann Harbor, Michigan, 1967.

Saeed, A., Sharov, V., White, J., Li, J., Liang, W., Bhagabati, N., Braisted, J., Klapa, M., Currier, T., Thiagarajan, M., Sturn, A., Snuffin, M., Rezantsev, A., Popov, D., Ryltsov, A., Kostukovich, E., Borisovsky, I., Liu, Z., Vinsavich, A., Trush, V. and Quackenbush, J. TM4: a free, open-source system for microarray data management and analysis. *BioTechniques*, 34(2):374-378, 2003.

Sapra, A.K., Arava, Y., Khandelia, P. and Vijayraghavan, U. Genome-wide analysis of pre-mRNA splicing: intron features govern the requirement for the second-step factor, Prp17 in Saccharomyces cerevisiae and Schizosaccharomyces pombe. *J Biol Chem,* 279(Suppl 50):52437-46, 2004.

Schaffer D. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University. 1984.

Schema, M., Shalon, D., Davis, R.W. and Brown, P.O. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467-470, 1995.

Schneider, B., Patton, E., Lanker, S., Mendenhall, M., Wittenberg, C., Futcher, B. and Tyers, M. Yeast GI Cyclins Are Instable in GI Phase. *Nature,* 395:86-89, 1998.

Schölkopf, B. and Smola, A.J. *Learning with Kernels*. MIT Press, Cambridge, MA., 2002.

Schuler, G.D., Boguski, M.S., Stewart, E.A., Stein, L.D., Gyapay, G., Rice, K., White, R.E., Rodriguez-Tom, P., Aggarwal, A., Bajorek, E. A Gene Map of the Human Genome. *Science*, 274(5278):540-546, Oct. 1996.

Schwefel, H.P. *Numerical Optimization of Computer Models*. Wiley, 1981.

Segal, E., Friedman, N., Kaminski, N., Regev, A. and Koller, D. From signatures to models: Understanding cancer using microarrays. *Nat. Genetics*, 37:S38-S45, 2005.

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D. and Friedman, N. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genetics*, 34(2):166-176, 2003.

Setubal, J.C. and Meidanis, J. *Introduction to Computational Molecular Biology*. PWS Publishing, 1997.

Shmulevich, I., Dougherty, E.R., Kim, S. and Zhang, W. Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18:261-74, 2002.

Silvescu, A. and Honavar, V. Temporal Boolean network models of genetic networks and their inference from gene expression time series. *Complex Sys.*, 13:54-70, 2001.

Silvescu, A. and Honavar, V. Temporal boolean network models of genetic networks and their inference from gene expression time series. *Complex Systems,* 11:61-78, 1997.

Simon, I., Barnett, J., Hannett, N., Harbison, C.T., Rinaldi, N.J., Volkert, T.L., Wyrick, J.J., Zeitlinger, J., Gifford, D.K., Jaakkola, T.S. and Young, R.A. Serial Regulation of Transcriptional Regulators in the Yeast Cell Cycle. *Cell*, 106(6):697-708, 2001.

Sivakumar, K., Chen, R. and Kargupta, H. Learning Bayesian Network Structure from Distributed Data. *In SIAM International Data Mining Conference*, 284-288, 2003.

Smolen, P., Baxter, D.A. and Byrne, J.H. Modeling Transcriptional Control in Gene Networks - Methods, Recent Results and Future Directions. *Bulletin of Mathematical Biology*, 62:247-292, 2000.

Soinov, L.A., Krestyaninova, M.A. and Brazma1, A. Towards reconstruction of gene networks from expression data by supervised learning. *Genome Biol*, 4(1):R6, 2003.

Someren, E.P.V., Wessels, L.F.A. and Reinders, M.J.T. Linear Modeling of Genetic Networks from Experimental Data. In: *International Conference on Intelligent Systems for Molecular Biology (ISMB),* 355-366, 2000.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Bio. of the Cell*, 9(12):3273-3297, 1998.

Standafer, E. and Wahlgren, W. *Modern Biology*. Holt, Rinehart and Winston, 2002.

Stapley, B.J. and Benoit, G. Biobibliometrics: information retrieval and visualization from co-occurrences of gene names in Medline abstracts. *Pac Symp Biocomput*, 529–540, 2000.

Stekel, D. *Microarray bioinformatics*. Cambridge University Press, 2003.

Sterelny, K. and Griffiths, P.E. *Sex and death : An Introduction to philosophy of bio. Science and Its Conceptual Foundations series*. University of Chicago Press, Chicago, IL, 1999.

Styczynski, M.P. and Stephanopoulos, G. Overview of Computational Methods for the Inference of Gene Regulatory Networks. *Computers and Chemical Eng*., 29:519-534, 2005.

Tanay, A., Sharan, R. and Shamir, R. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(Suppl. 1):S136--S144, 2002.

Tegner, J., Yeung, M.K., Hasty, J. and Collins, J.J. Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling. *Proc. of the National Academy of Sciences, USA*, 100(10):5944-5949, 2003.

Tiefei, L. Learning Gene Network Using Bayesian Network Framework. *PhD thesis, National University of Singapore,* 2005.

Toh, H. and Horimoto, K. Inference of A genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics*, 18(2):287–297, 2002.

Tong, A.H., Evangelista, M., Parsons, A.B., Xu, H., Bader, G.D., Page, N., Robinson, M., Raghibizadeh, S., Hogue, CW., Bussey, H., Andrews, B., Tyers, M. and Boone, C. Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, 294(5550):2364-2368, 2001.

Toyn, J.H., Johnson, A.L., Donovan, J.D., Toone, W.M., Johnston, L.H. The Swi5 Transcription Factor of Saccharomyces cerevisiae Has a Role in Exit from Mitosis through Induction of the Cdk-Inhibitor SicI in Telophase. *Genetics,* 145:85-96, 1997.

Troyanskaya, O., Cantor M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. and Altman R. Missing Value Estimation methods for DNA microarrays. Bioinformatics, 17(6):520-525, 2001.

van Someren, E.P., Wessels, L.F. and Reinders, M.J. Linear modeling of genetic networks from experimental data. *Proc Int Conf Intell Syst Mol Biol*, 8:355-366, 2000.

Vohradský, J. Neural network model of gene expression. *FASEB Journal*, 15:846-854, 2001.

Wahde, M. and Hertz, J. Course-Grained Reverse Engineering of Genetic Regulatory Networks. *Biosystems*, 55:129-136, 2000.

Wang, J., Han, J. and Pei, J. CLOSET+: searching for the best strategies for mining frequent closed itemsets. *ACM Press*, 236-245, 2003.

Weaver, R.F. *Molecular Biology*. McGraw-Hill, 2001.

Weindruch, R. and Walford, R. L. *The Retardation Of Aging And Disease By Dietary Restriction*. Thomas, Springfield, 1988.

Welford, S.M., Gregg, J., Chen, E., Garrison, D., Sorensen, P.H, Denny, C.T. and Nelson, S.F. Detection of differentially expressed genes in prinmary tumor tissues using representational defferences analysis coupled to microarray hybridization. *Nucl. Acids Res.*, 26:3059-3065, 1998.

Wen, X., Fuhrman, S., Michaels, G.S., Carr, D.B., Smith, S., Barker, J.L. and Somogyi, R. Large-scale temporal gene expression mapping of central nervous system development. *Proc. Natl. Acad. Sci.*, 95(1):334-339, 1998.

Witten, I. and Frank, E. *Data Mining - Practical Machine Learning Tools and Techniques with JAVA Implementations*. San Francisco, CA: Morgan Kaufmann, 1999.

Xing, B. and van der Laan, M.J. A causal inference approach for constructing transcriptional regulatory networks. *Bioinformatics*, 21:4007-4013, 2005.

Yeang, C.H. and Jaakkola, T. Physical Network Models and Multi-Source Data Integration. *Proc. Seventh Ann. Int'l Conf. Research in Computational Molecular Biology Apr.*, 312-321, 2003.

Yeang, C.H., Ideker, T. and Jaakkola, T. Physical network models. *J Comput Biol*, 11:243-62, 2004.

Yeang, CH. and Jaakkola, T. Time series analysis of gene expression and location Data. In: *Third IEEE Symposium on BioInformatics and BioEngineering (BIBE'03) Bethesda, Maryland: Institute of Electrical and Electronics Engineers, Inc.,* 305-312, 2003.

Yoo, C. and Cooper, G.F. Discovery of gene-regulation pathways using local causal search. *Proc AMIA Symp*, 914-918, 2002.

Yu, J., Smith, V.A., Wang, P.P., Haremink, A.J. and Jarvis, E.D. Advances to Bayesian Network Inference for Generating Causal Networks from Observational Biological Data. *Bioinformatics*, 20(18):3594-3603, 2004.

Zaki, M.J. and Hsiao, C.J. CHARM: an efficient algorithm for closed itemset mining. *SIAM Press*, 457-473, 2002.

Zamani, Z., Hajihosseini, A., Masoudi-Nejad, A. Computational Methodologies for Analyzing, Modeling and Controlling Gene Regulatory Networks. *Biomedical Engineering and Computational Biology,* 2:47-62, 2010.

Zhang, A. Advanced Analysis of Gene Expression Microarray. *World Scientific press*, 2006.

Zhou, X.B., Wang, X.D., Pal, R., Ivanov, I., Bittner, M. and Dougherty, E.R. A Bayesian Connectivity-Based Approach to Constructing Probabilistic Gene Regulatory Networks. *Bioinformatics*, 20(17):2918-2927, 2004.

Zimmermann, P., Wille, A., Buhlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E., Prelic, A. and Bleuler, S. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics,* 22(9):1122-1129, 2006.

Zitzler, E. and Künzli, S. Indicator-Based Selection in Multiobjective Search. *In: X. Yao, E. K. Burke, J. Lozano, J. Smith, J. Merelo Guervós, J. Bullinaria, J. Rowe, P. Tiño, A. Kabán, H-P. Schwefel (Eds.): PPSN VIII. LNCS*, 3242:832-842, 2004.

Zitzler, E. and Thiele, L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Trans. Evol. Comput*, 3(4):257-271, 1999.

Zitzler, E., Laumanns, M. and Thiele, L. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *In Giannakoglou, Tsahalis, Periaux, Papailiou, and Fogarty (eds), Evolutionary Methods for Design, Optimisations and Control*, 19-26, 2002.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. and Grunert da Fonseca, V. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Trans. Evol. Comput*, 7(2):117-132, 2003.

Zou, M. and Conzen, S.D. A New Dynamic Bayesian Network (DBN) Approach for Identifying Gene Regulatory Networks from Time Course Microarray Data. *Bioinformatics*, 21:71-79, 2005.