EXTRACCIÓN DE PATRONES DE COMPORTAMIENTO EN DATOS DE EXPRESIÓN GENÓMICA

Beatriz Pontes
Dto. Lenguajes y Sistemas Informáticos, Universidad de Sevilla, bepontes@lsi.us.es

Raúl Giráldez
Escuela Politécnica Superior, Universidad Pablo de Olavide, rgirroj@upo.es

Eduardo F. Camacho
Dto. Ingeniería de Sistemas y Automática, Universidad de Sevilla, eduardo@esi.us.es

Resumen

Los algoritmos de biclustering persiguen obtener subconjuntos de genes que se expresan de una manera similar frente a un subconjunto de condiciones. Resulta necesario por tanto poder determinar la calidad de los biclusters obtenidos. En este artículo se presenta una técnica basada en programación lineal para la extracción de patrones de desplazamiento en biclusters, pudiendo de esta manera dar una medida de cómo se ajustan los genes de dichas submatrices a un patrón de comportamiento. Los resultados obtenidos son comparados con los que se obtienen utilizando computación evolutiva.

Palabras clave: Biclustering, Datos de Expresión Genómica, Programación Lineal.

1. INTRODUCCIÓN

La tecnología MicroArray (biochip) [17, 18] es una de las diferentes aproximaciones al análisis comparativo de patrones de expresión de ADN o ARN, cuyo fin es colocar en una micromatriz cada uno de los genes de un genoma cuyos niveles de expresión pueden ser cuantificados [3]. Para ello se sintetiza el material genético y se insertan de forma automática en una capa de cristal, silicio o plástico, colocándose en unas casillas que actúan a modo de tubo de ensayo. Después se hibrida y se elimina todas las cadenas que no se han unido mediante lavados (sólo las moléculas que hibridan permanecerán en el biochip), y se procede al revelado mediante un escáner óptico o con microscopia láser confocal. De esta manera se obtiene una matriz de expresión donde las columnas serán genes y las filas experimentos. Los datos son posteriormente normalizados y transformados para disminuir las variaciones y hacer los cálculos posteriores más sencillos.

Las técnicas de clustering [4] permiten extraer conocimiento a partir de los MicroArrays. Consisten en agrupar los datos en conjuntos disjuntos, llamados clusters, de forma que los genes o condiciones (según se aplique por filas o por columnas) que estén en un mismo cluster sean muy similares. Para ello se sirven de algoritmos o fórmulas matemáticas que permitan determinar la similaridad entre los datos.

Sin embargo, muchos patrones de actividad de grupos de genes sólo se presentan bajo un determinado conjunto de condiciones experimentales, mientras que bajo otras condiciones estos genes pueden comportarse de forma independiente [14, 19]. El descubrimiento de estos patrones locales de comportamiento puede ser la clave para descubrir relaciones entre genes o condiciones que no podrían ser obtenidas mediante técnicas de clustering.

Las técnicas de biclustering [13] consisten en obtener submatrices a partir de un MicroArray, donde tanto el conjunto de genes como de condiciones seleccionado sean subconjuntos de la matriz original. De esta manera se aborda la extracción de datos con respecto a las dos dimensiones simultáneamente [7]. Además, a diferencia de las técnicas de clustering, se tiene en cuenta que pueden haber genes o condiciones no contemplados en ninguno de los biclusters obtenidos, así como que un mismo gen o condición puede formar parte de varios biclusters.

Existen métodos muy variados para abordar el biclustering en la literatura. En [7] se describe un método basado en recorrer el espacio de búsqueda mediante la adición y eliminación de filas y columnas. Otros trabajos utilizan algoritmos evolutivos [5, 10] o de enfriamiento simulado [6].

Los genes contenidos en un bicluster deben presentar un comportamiento similar (se dice

que están co-expresados frente al mismo conjunto de condiciones), pero esto no implica que sus valores de expresión sean similares. De esta manera podemos hablar de patrones de expresión [1], que se clasifican en dos tipos, patrones de desplazamiento y de escalado. A cada bicluster podremos asociarle un patrón de comportamiento que representará el comportamiento de todos los genes que se incluyan en dicha submatriz.

El objetivo de este trabajo es comparar el patrón de desplazamiento obtenido para cada bicluster mediante dos técnicas diferentes, algoritmos evolutivos [16] y programación lineal. Dicha comparación nos permitirá valorar los resultados obtenidos mediante computación evolutiva en trabajos previos, para posteriormente poder realizar la búsqueda de patrones más complejos difícilmente abordables mediante programación lineal. Los resultados muestran que ambas técnicas son equiparables en cuanto a la calidad de los patrones obtenidos.

La estructura de este artículo es la siguiente: en el apartado 2 se describen los distintos tipos de patrones presentes en un bicluster, el apartado 3 muestra el método utilizado para extraer dichos patrones mediante programación lineal, los resultados obtenidos mediante esta técnica y su comparación con los de un algoritmo evolutivo son presentados en la sección 4. Por último, el apartado 5 resume las principales conclusiones.

2. PATRONES DE COMPORTAMIENTO

Todos los genes pertenecientes a un determinado bicluster deben seguir un mismo patrón de comportamiento. Estos patrones fueron ya nombrados en [7], encontrándose formalmente descritos en [1], donde se definen dos tipos de patrones.

Sea $\mathcal{M}$ un MicroArray representado por una matriz de números reales, tal que se componga de N filas (condiciones) y M columnas (genes), cuyos elementos se denominen v_{ij}, y sea $\mathcal{B} \subseteq \mathcal{M}$ un bicluster formado por $n \leq N$ condiciones y $m \leq M$ genes, cuyos elementos vengan dados por w_{ij}, podemos decir que dicho bicluster sigue un patrón de desplazamiento cuando sus valores pueden ser obtenidos según la siguiente expresión:

$$w_{ij} = \pi_j + \beta_i + \xi_{ij} \tag{1}$$

donde π_j es un valor fijo para el gen j^{th}, β_i es el valor de desplazamiento para la condición i^{th}, y ξ_{ij} es el error cometido por el patrón para el valor w_{ij}.

De una manera análoga podemos decir que un bicluster sigue un patrón de escalado cuando sus valores siguen la expresión:

$$w_{ij} = \pi_j \times \alpha_i + \xi_{ij} \tag{2}$$

donde hemos cambiado el valor aditivo β de la ecuación anterior por un valor multiplicativo α, que representa el factor de escalado para cada una de las condiciones.

Ambos tipos de patrones pueden combinarse, existiendo en los biclusters los dos comportamientos, pudiendo escribirse los datos como:

$$w_{ij} = \pi_j \times \alpha_i + \beta_i + \xi_{ij} \tag{3}$$

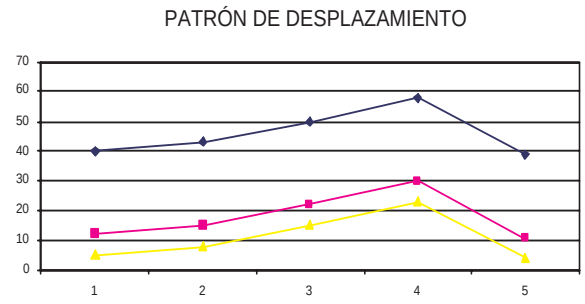
En todos los casos, diremos que un bicluster es *perfecto* cuando todos los valores ξ_{ij} sean igual a 0. La figura 1 muestra dos biclusters perfectos (de desplazamiento el de arriba y de escalado el de abajo). Cada una de las líneas presentes en las gráficas se corresponden con un gen del bicluster, representándose en el eje de abcisas las condiciones. En el primer caso todos los genes tienen la misma forma y pendiente en todos sus tramos, mientras que en el caso del escalado, los genes presentan la misma forma pero sus niveles de expresión varían de una manera multiplicativa, como se puede ver en la figura.

3. PATRONES DE DESPLAZAMIENTO CON PROGRAMACIÓN LINEAL

El objetivo es buscar el patrón de desplazamiento que mejor se ajuste a un bicluster dado. En esta sección se explica cómo extraer el patrón de desplazamiento que minimice el error cometido con respecto a todos los genes, utilizando para ello métodos de programación lineal [11].

Para abordar el problema desde este punto de vista es necesario hacer un planteamiento del mismo de manera que podamos resolverlo como un sistema de ecuaciones lineales. Partiendo de la descomposición de los datos del bicluster de la manera en que aparece en la fórmula 1 (patrón de desplazamiento), es posible expresar el problema matricialmente de la siguiente forma:

$$\mathcal{E} = \mathcal{W} - \mathcal{X} \times \mathcal{O} \tag{4}$$



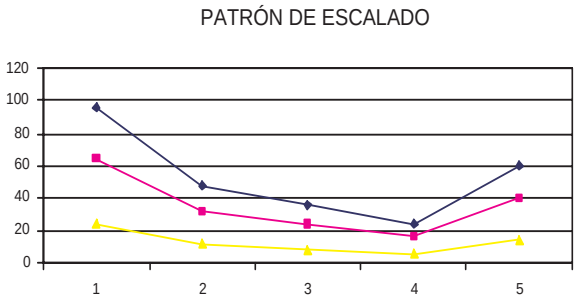


Figura 1: Biclusters con patrones de desplazamiento y escalado perfectos

donde $\mathcal{W}$ representa la matriz de todos los valores w_{ij} ordenados vectorialmente por columnas, esto es, los i primeros elementos corresponderán a los valores del primer gen (o columna) del bicluster para todas las condiciones. $\mathcal{E}$ es un vector formado por todos los errores ξ_{ij} ordenados de la misma manera que el vector $\mathcal{W}$ de los valores originales. $\mathcal{O}$ representa el vector de parámetros, que contendrá todos los valores π_j y β_i organizados vectorialmente. Por último, $\mathcal{X}$ se corresponde con una matriz con las dimensiones apropiadas que permita que se realice el producto, y que contendrá únicamente los valores 0 o 1 situados de tal manera que el producto de esta matriz por el vector de parámetros sea una suma de dos elementos, un valor constante β_i para cada condición y el valor típico π_j de cada gen, cumpliendo así la estructura de desplazamiento que se especifica en la sección 2.

A continuación se muestra un ejemplo de cómo obtener el vector de los valores de error según la ecuación 4. Supongamos que la matriz original o bicluster de partida viene dado según los siguientes datos, que se corresponden con la primera gráfica de la figura 1:

$$\mathcal{W} = \begin{pmatrix} 40 & 12 & 5 \\ 43 & 15 & 8 \\ 50 & 22 & 15 \\ 58 & 30 & 23 \\ 39 & 11 & 4 \end{pmatrix}$$

El objetivo es calcular los valores π y β que minimicen el error cometido al aproximar los datos oginales por la suma de ambos. Para ello es posible calcular el vector de errores $\mathcal{E}$ como el resultado de la siguiente expresión:

$$\begin{pmatrix} 40 \\ 43 \\ 50 \\ 58 \\ 39 \\ 12 \\ 5 \\ 22 \\ 30 \\ 11 \\ 5 \\ 8 \\ 15 \\ 23 \\ 4 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{pmatrix}$$

El sistema de ecuaciones lineales resultante de igualar la expresión de la ecuación 4 a el vector nulo solamente tendrá solución cuando se esté aplicando a un bicluster perfecto. En otro caso se buscará la solución mediante el método de mínimos cuadrados [9], asegurando de esta manera que se obtienen los parámetros que minimizan la suma de los errores al cuadrado, $\sum \xi_{ij}^2$. Debido a la naturaleza de los datos que componen los biclusters, que suelen expresarse mediante número reales, existen infinitas soluciones de descomposición de los datos de la matriz que minimizan el error cometido.

Debido a las características de la matriz $\mathcal{X}$, es posible que el método de mínimos cuadrados no produzca solución al existir una matriz que sea singular, y que por tanto no tenga inversa. En estas ocasiones el resultado es expresado en términos de la pseudoinversa de dicha matriz [12].

En el siguiente apartado se muestran los resultados obtenidos mediante este procedimiento así como una comparación con los resultados tras aplicar un algoritmo evolutivo desarrollado en trabajos anteriores.

4. RESULTADOS Y COMPARATIVA

Para la obtención de los resultados mediante programación lineal se ha utilizado en entorno de programación MATLAB [15]. La búsqueda de patrones de desplazamiento se ha realizado

sobre biclusters obtenidos a partir de datos reales por medio de un algoritmo evolutivo [10]. Los MicroArrays de partida son detallados en [2, 8], y se corresponden con datos reales del ciclo celular de la levadura (*Saccharomyces cerevisiae*), y de células humanas respectivamente.

Diferentes tipos de biclusters han sido evaluados mediante la herramienta presentada, variando éstos en el número de genes y condiciones así como el grado de similitud existente entre sus genes. Los patrones obtenidos para algunos biclusters de ambas bases de datos se representan en las figuras 2 (levadura) y 3 (humanos). Estos biclusters han sido elegidos para ser representados ya que son algunos de los que aparecen en un trabajo anterior que obtiene los patrones mediante un algoritmo evolutivo [16].

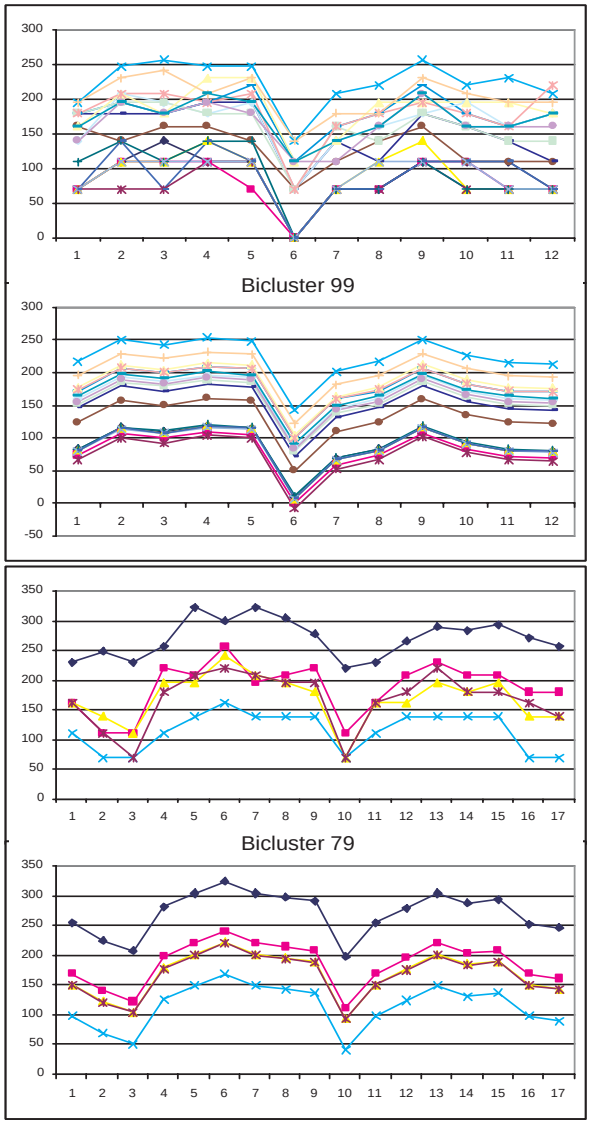


Figura 2: Dos biclusters y sus patrones para la levadura

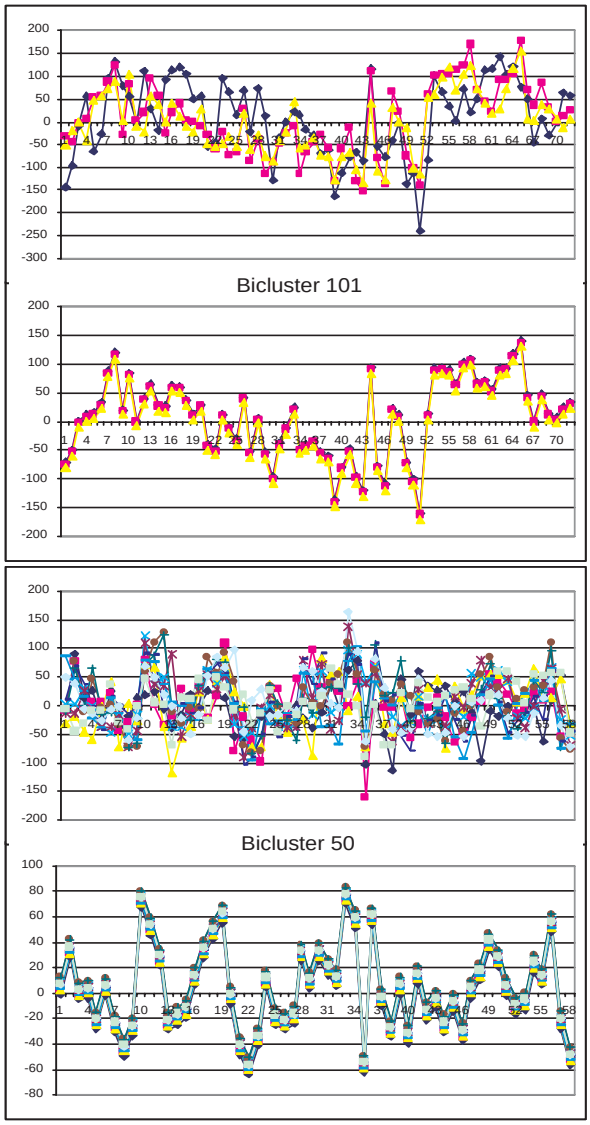


Figura 3: Dos biclusters y sus patrones para datos humanos

En ambas gráficas se presentan dos figuras por cada bicluster. La primera de ellas se corresponde con el bicluster original, obtenido de [10], mientras que la segunda representa el patrón obtenido para ese bicluster, mediante el método anteriormente presentado, y escalado a los rangos de cada uno de los genes, para poder realizar una mejor comparativa.

Como se puede apreciar en las figuras, la principal diferencia entre los patrones obtenidos en el caso de la levadura y de células humanas es el número de condiciones presentes en cada bicluster. Mientras que para el primer caso los biclusters se componen de 15 condiciones aproximadamente, para datos humanos éste número se sitúa alrededor de 65 condiciones. Esta diferencia implica que el error cometido sea mayor, ya que el

NB	error AE	error PL	#cond	#genes
1	11.5	11.53	17	82
41	11.09	11.23	17	20
43	11.21	11.46	16	13
48	11.01	11.24	17	9
61	11.45	11.62	15	7
64	10.83	11.32	14	9
66	10.16	10.84	16	7
68	11.81	11.99	15	8
79	11.98	12.3	17	5
95	11.68	11.85	14	8
98	10.66	11.35	16	5
99	11.39	11.56	12	18

Cuadro 1: Errores cometidos por ambos algoritmos sobre biclusters de la levadura

NB	error AE	error PL	#cond	#genes
1	26.87	26.73	56	37
31	27.4	26.64	65	14
50	26.4	26.52	58	11
54	27.31	27.28	47	9
72	27.08	27.33	60	5
74	26.94	27.01	39	14
84	27	27.34	41	8
89	26.93	27.32	49	6
93	25.63	25.85	52	7
97	26.99	27.25	62	6
100	27.29	27.45	52	8
101	24.77	26.77	72	3

Cuadro 2: Errores cometidos por ambos algoritmos sobre biclusters de células humanas

patrón deberá buscar un comportamiento similar frente a un mayor número de datos, y por tanto deberá ajustarse a un mayor número de puntos.

A continuación se presentan dos tablas que contienen información acerca de los errores cometidos por ambos métodos (programación lineal y algoritmos evolutivos), para los biclusters más relevantes de los dos conjuntos de datos que se han analizado, células de levadura y humanas. La primera de las columnas indica el número del bicluster según la nomenclatura utilizada en [10]. Las columnas segunda y tercera muestran el error cometido según se haya utilizado un algoritmo evolutivo, detallado en [16], o el método aquí presentado. Las dos últimas columnas representan el número de condiciones y genes presentes en ese bicluster, respectivamente. Hay que tener en cuenta que para poder comparar los resultados de ambos métodos, y debido a que los errores evaluados mediante el algoritmo evolutivo vienen dados en términos del error absoluto cometido, hemos calculado también el error absoluto cometido al obtener los patrones de ésta segunda manera, a partir de los resultados.

Como puede verse en dichas tablas, no existen diferencias significativas entre los errores cometidos por ambos métodos. En la mayoría de los casos, el algoritmo evolutivo presenta resultados sensiblemente mejores. Sin embargo, y como ya se ha dicho anteriormente, los dos métodos mencionados utilizan distintas funciones de minimización (error absoluto o error cuadrático), justificando así que los resultados obtenidos mediante mínimos cuadrados sean sensiblemente superiores a aquellos que resultan del algoritmo evolutivo. Cabe destacar que, aunque los resultados en términos del error cometido sean muy similares en ambos casos, los valores π y β mencionados en la sección 2 y obtenidos por ambos métodos no coinciden. Esto es debido a que según la formulación del problema, existen infinitas combinaciones de dichos valores que minimicen el error cometido.

Es importante destacar que el propósito de este artículo es corroborar, mediante un método analítico y determinista, la validez de los resultados cometidos por el algoritmo evolutivo presentado en trabajos anteriores [16]. Es evidente que el coste cometido por un procedimiento evolutivo es mucho mayor, pero ésto se ve justificado por ser un primer paso hacia la búsqueda de patrones de desplazamiento y escalado conjuntamente (ver sección 2), que no podría ser lograda mediante métodos de programación lineal.

5. CONCLUSIONES

Las técnicas de biclustering se aplican sobre datos de expresión genómica para agrupar genes y condiciones de un MicroArray simultáneamente. Los genes que componen un mismo bicluster se caracterizan por tener un comportamiento similar a lo largo de las condiciones presentes. Dicho comportamiento viene dado por un patrón que tenga la representación gráfica que más se asemeje a la de todos los genes por separado. Un caso particular de patrón es el de desplazamiento, que se caracteriza por considerar que las gráficas de todos los genes tienen la misma similitud y pendientes. En este artículo hemos presentado una herramienta basada en programación lineal capaz de obtener el patrón de desplazamiento que más se aproxima a todos los genes de un bicluster. Los resultados obtenidos confirman la validez de una herramienta evolutiva presentada en trabajos anteriores encargada de realizar la misma tarea. Dicha herramienta evolutiva es necesaria ya que será ampliada para poder realizar la búsqueda de patrones de desplazamiento y

escalado simultáneamente, no pudiendo hacerse ésta mediante programación lineal.

Referencias

[1] J. S. Aguilar-Ruiz. Shifting and scaling patterns from gene expression data. *Bioinformatics*, 21:3840–3845, 2005.

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

[3] P. Barrero. Aplicaciones de la técnica de microarrays en ciencias biomédicas: presente y futuro. *Química viva*, 3(4):1–10, 2005.

[4] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–297, 1999.

[5] S. Bleuler, A. Prelić, and E. Zitzler. An ea framework for biclustering of gene expression data. pages 166–173, Piscataway, NJ, 2000.

[6] K. Bryan, P. Cunningham, and N. Bolshakova. Biclustering of expression data using simulated annealing. In *18th IEEE Symposium on Computer-Based Medical Systems*, pages 383–388, Dublin, Ireland, 2005.

[7] Y. Cheng and G. M. Church. Biclustering of expression data. In *In Proceedings of the 8th International Conference on Intellingent Systemns for Molecular Biology*, pages 93–103, La Jolla, CA, 2000.

[8] R. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. Wolfsberg, A. Gabrielian, D. Landsman, D. Lockhart, and R. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2:65–73, 1998.

[9] B. P. Demidowitsch, Maron, and Schuwalowa. *Métodos numéricos de análisis*. Paraninfo, 1978.

[10] F. Divina and J. S. Aguilar-Ruiz. Evolutionary biclustering of microarray data. *Lecture Notes on Computer Science*, 3449:1–10, 2005.

[11] M. A. Goberna. *Optimización Lineal. Teoría, Métodos y Modelos*. McGraw-Hill, 2004.

[12] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins University Press, Third Edition.

[13] J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[14] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1370–1386, 2004.

[15] J. H. Mathews and K. D. Fink. *Numerical Methods using MATLAB*. Prentice Hall, 2000.

[16] B. Pontes, R. Giráldez, and J. Aguilar. Shifting patterns discovery in microarrays with evolutionary algorithms. In *10th International conference on Knowledge-based & Intelligent Information & Engineering Systems (KES-06), In Press*, 2006.

[17] M. Schena, D. Shalon, R. Davis, and P. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 70:467–470, 1995.

[18] D. Shalon, S. Smith, and P. Brown. A dna microarray system for analyzing complex dna samples using two-color fluorescent probe hybridization. *Genome Res*, 6:639–645, 1996.

[19] A. Tanay, R. Sharan, and R. Shamir. Biclustering algorithms: A survey. In *In Handbook of Computational Molecular Biology, Edited by Srinivas Aluru, Chapman & Hall/CRC, Computer and Information Science Series*, 2005.