Centro de Investigación Científica y de Educación Superior de Ensenada

Maestría en Ciencias de la Computación

Reconocimiento de Patrones

------------------------------------------------------------------------------------------------------------------------------

Biclustering para datos de Expresión Genética

utilizando Optimización Multiobjetivo MOEA/D y posterior análisis mediante SOM y PCA.

------------------------------------------------------------------------------------------------------------------------------

Estudiantes:

* César Miguel Valdez Córdova
* Luis Enrique García Hernández

Profesor:

* Dr. Hugo H. Hidalgo Silva

Fecha: 20 de abril de 2018

Resumen

Una de áreas de la ciencia beneficiadas por los avances en la capacidad de cómputo y desarrollo de herramientas de medición es la Genómica. Entre los experimentos que han tenido gran popularidad son los Microarreglos de ADN. Dicha técnica permite medir los niveles de expresión génica de cientos o miles de genes bajo distintas condiciones experimentales. Los resultados obtenidos se organizan en una matriz, llamada Matriz de Expresión (*Expression Matrix*) en el que cada uno de los elementos representan el nivel de expresión de un gen bajo una condición experimental específica. Para la interpretación funcional del gran conjunto de datos generados a través de Microarreglos de ADN, se necesita un desarrollo paralelo de métodos computacionales. Sobre estos conjuntos de datos, los algoritmos de construcción de biclusters tratan de identificar asociaciones de genes y condiciones experimentales, donde los genes exhiben una alta correlación para cada condición dada. En el presente artículo describimos la utilización del algoritmo genético multi-objetivo, MOEA/D, para la identificación de biclusters de interés en datos de Microarreglos de ADN. Se realizó un posterior análisis de los resultados mediante técnicas de análisis de componentes principales (PCA) y mapa auto-organizado (SOM).

# 

Contenido

[Biclustering para datos de Expresión Genética 1](#_Toc511840822)

[utilizando Optimización Multiobjetivo MOEA/D y posterior análisis mediante SOM y PCA. 1](#_Toc511840823)

[Resumen 1](#_Toc511840824)

[1. Introducción 2](#_Toc511840825)

[1.1. Biclustering: un problema NP difícil 3](#_Toc511840826)

[1.2. Complejidad del problema 4](#_Toc511840827)

[1.3. Biclustering para datos de expresión génica. 4](#_Toc511840828)

[2. Materiales y Métodos 7](#_Toc511840829)

[2.1. Matrices de Expresión utilizadas. 7](#_Toc511840830)

[2.2. Algoritmos Evolutivos Multi-Objetivo 8](#_Toc511840831)

[2.3. MOEA/D 9](#_Toc511840832)

[3. Resultados 13](#_Toc511840833)

[4. Conclusiones 16](#_Toc511840834)

[5. Referencias 16](#_Toc511840835)

1. Introducción

La Avances tecnológicos han impulsado la creación de experimentos de mayor complejidad y precisión; una de las tantas áreas de la ciencia beneficiadas por los avances en la capacidad de cómputo y desarrollo de herramientas de medición es la Genómica. Uno de los experimentos que ha tenido gran popularidad son los Microarreglos de ADN. Dicha técnica permite medir los niveles de expresión génica de cientos o miles de genes bajo distintas condiciones experimentales. Los resultados obtenidos se organizan en una matriz, llamada Matriz de Expresión (*Expression Matrix*) en el que cada uno de los elementos representa el nivel de expresión de un gen bajo una condición experimental específica.

Para el análisis de la Matriz de Expresión se han utilizado técnicas de aprendizaje máquina y de clustering. Este último busca encontrar grupos de genes que presentan variaciones de niveles de expresión génica similares bajo todas las condiciones experimentales. Si un par de genes muestran una tendencia similar a través de todas las muestras, estos pudieran reflejar algún tipo de interacción o relación entre sus funciones. Pero no necesariamente los genes se tienen que relacionar entre todo el grupo de condiciones, pudiera ser que su semejanza se muestre en un subconjunto de condiciones, razón por la que el clustering debería aplicarse de forma simultánea entre genes y condiciones. Para esto es usada la técnica de Biclustering, ver figura 1. La técnica fue por primera vez introducida en la década de los 70’s por Hartigan, y la primera vez que fue utilizada en el contexto de análisis de datos de expresión génica fue por Cheng y Church en el año 2000 [1]

* 1. Biclustering: un problema NP difícil
     1. **Minería de datos y biclustering**

Cómo habíamos mencionado antes, avances tecnológicos han hecho posible una mejor medición, captura y almacenaje de datos. El reto al que nos enfrentamos con esta cantidad de información es el llamado “*data avalanche*” en donde tenemos una gran cantidad de datos sobre múltiples variables de un fenómeno complejo, pero aún no podemos determinar cuáles son los parámetros que describen el estado actual del fenómeno. Este tipo de sistemas complejos son fáciles de encontrar en ´áreas como biología, ecología, sociología, economía [2].

Dentro de las ciencias computacionales, en los campos de minería de datos y aprendizaje de máquina, se han desarrollado una gran variedad de técnicas y enfoques para resolver este problema; en dónde no sólo utilizan métodos estadísticos, sino también utilizan métodos asociados con optimización, métodos algebraicos y redes neuronales [2].

En minería de datos se busca revelar perfiles de similitud entre los datos, descartando aquellos que son irrelevantes para el estado del fenómeno estudiado. Para encontrar patrones, a veces se busca particionar en muestras de datos para un criterio de similitud. Esta tarea se denomina clustering. Dependiendo del tipo de datos (numéricos, categóricos o binarios, es que se selecciona una técnica (k-medias, *self-organizing maps* (SOM), etc.) [2].

Sin embargo, existe la posibilidad de no sólo analizar la similitud entre las muestras bajo todos sus atributos, sino también encontrar similitudes entre atributos. Se pudiera esperar que, por un par de muestras asociadas dentro de un clúster, la asociación sea inducida por los componentes de la muestra. Por lo tanto, si realizamos una asociación por componentes para realizar un clúster pudiéramos encapsular un grupo de muestras. Tal par de clusters, los llamamos bicluster y el problema de particionar datos en biclusters es llamado biclustering [2]. En la figura 1 podemos observar la diferencia entre el método de clustering y biclustering.

.

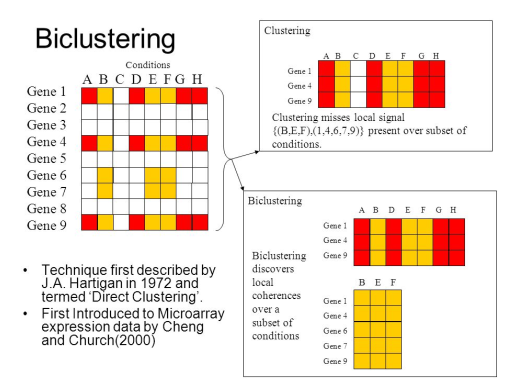


Fig. 1: Comparación entre el método de clustering y biclustering.

* + 1. **Definición de Biclustering**

Suponiendo un conjunto de datos de n muestras y atributos en una matriz   
donde el valor de es la expresión de la i-ésima atributo en la j-ésima muestra. Una herramienta muy utilizada para la visualización de conjuntos de datos son los heatmaps. Un heatmap es una cuadrícula rectangular compuesta de pixeles que corresponden a cada valor del conjunto de datos. El color del pixel entra en un rango de verde (o azul) para denotar los valores menores, mientras que se utiliza un color rojo para valores de mayor intensidad. Esta herramienta hace que la observación de un posible patrón sea más sencilla.

* 1. Complejidad del problema

El biclustering es un problema NP-difícil (Demostrado en el 2002 Tanay et al.), y es por ello que la mayoría de los métodos utilizados se basan en procedimientos de optimización. El que sea NP-difícil implica que una búsqueda exhaustiva en el espacio de decisión no sea factible, pero aplicando una medida de calidad a una solución candidata, el uso de una meta-heurística para la solución del problema parece apropiado. Las meta-heurísticas hacen pocas o ninguna suposición sobre el problema que se desea optimizar y son capaces de hacer una búsqueda en un espacio de soluciones candidatas grande de manera iterativa, en donde cada iteración trata de mejorar la mejor solución candidata encontrada al momento en función de la medida de calidad preestablecida en el algoritmo. Es importante recordar que el uso de las meta-heurísticas no nos garantiza que la solución óptima será encontrada [3].

* 1. Biclustering para datos de expresión génica.

La aplicación de biclustering en minería de datos en la rama de la Biología son variadas, por ejemplo, Busygin et al. [4] señala que la demanda por métodos para el análisis de datos en ciencias de la vida surge con dos factores importantes. Primero, el proyecto de genoma humano y de otros seres vivos; y segundo, el uso de microarreglos de ADN. Por otro lado, Madeira et al. [3] menciona que muchos de los datos de expresión génica están relacionados con estudios de cáncer. Algunos datos son de tejidos cancerosos en distintas etapas de la enfermedad. Otras analizan muestras de distintos individuos que padecen de diferentes tipos de cáncer y algunas otras bases de datos se tienen muestras de individuos enfermos de un cáncer específico mezcladas con muestras de individuos sanos.

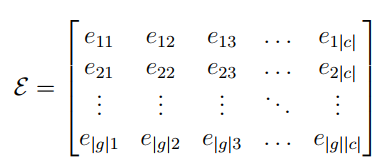
La aplicación de biclustering desea encontrar el poder hacer inferencias sobre el tipo de genes que se presentan en diferentes condiciones y comprender más sobre su función dentro del desarrollo de la enfermedad. También se han analizado datos en el que las condiciones son diferentes etapas del tratamiento de la enfermedad, y encontrar una relación entre el efecto positivo del tratamiento y los genes. También se han hecho estudios en el área de nutrición para identificar subconjuntos de alimentos con otros subconjuntos de sus atributos. En nuestro caso nos enfocaremos en, la utilización de biclustering como técnica de análisis de datos, para analizar Matrices de Expresión.

En una reciente revisión del estado del arte sobre aplicaciones de Biclustering, elaborada por Xie et al. [4] se explica que durante los últimos 17 años se desarrolló una cantidad considerable de métodos biclustering. SAMBA, ISA, BIMAX, QUBIC y FABIA son algunos algoritmos populares para uso general. CCC-biclustering y LateBiclustering están diseñados para el análisis de datos temporales, y BicPAM, BicNET y MCbiclust son tres herramientas recientes. Además, varias herramientas (paquetes R, servidores web, etc.) se han desarrollado para facilitar a los usuarios con un fondo computacional limitado. GEMS es un servidor web para minería de expresión genética basada en un paradigma de muestreo de Gibbs, y biclust y QUBICR son dos paquetes R que integran múltiples algoritmos existentes, funciones de preprocesamiento de datos, su interpretación y visualización de los resultados.

En su artículo Xie et al. [4] señalan que la aplicación de biclustering no ha progresado en paralelo con el diseño de algoritmos. Considerando todas las publicaciones relacionadas con biclustering, la porción de la aplicación los estudios han sido mucho más bajos que los estudios de desarrollo de algoritmos del año 2000-17. Esta situación está afectada por múltiples factores. Primero, hay una brecha entre la herramienta desarrollo y la comprensión de nuevas biotecnologías y propiedades de datos correspondientes. Por ejemplo, en microarreglos los datos reflejan la expresión génica absoluta con valores continuos de intensidad de fluorescencia, mientras que los datos de ARN-seq miden el nivel de expresión relativa utilizando discreto, positivo y cuentas de lectura muy asimétricas. Además, hay abundantes ceros en los datos de expresión de genes basados en RNA-seq, como no todos los genes se expresan bajo una condición experimental específica, lo cual es particularmente cierto en los datos de RNA de célula única (scRNA-Seq). Por lo tanto, algoritmos diseñados y evaluados en el uso de datos de microarreglos pueden no ser adecuado para su aplicación directa a los datos de RNA-seq. Los datos de RNA-seq y scRNA-seq necesitan el diseño de algoritmo específicos y desarrollo de herramientas. Sin embargo, contrariamente a el hecho de que RNA-seq se está volviendo más y más popular, pocos los algoritmos de biclustering están explícitamente diseñados para datos de RNA-seq. En segundo lugar, existe una brecha de conocimiento para aplicar herramientas biclustering y elegir el acompañamiento apropiado herramientas analíticas para análisis de datos específicos. Por lo general, Biclustering no es una herramienta de análisis de datos, por sí solo. En cambio, se conecta con otros procesos de anotación de resultados, programas de visualización (por ejemplo, Cytoscape) y métodos estadísticos (por ejemplo, análisis de componentes principales y análisis de regresión), para derivar una interpretación más completa. Vale la pena señalar que la integración orgánica de un algoritmo biclustering y las herramientas apropiadas que lo acompañan en una tubería no son triviales. La construcción de una tubería unificada requiere una comprensión más profunda de los diseños de algoritmos subyacentes, las entradas de datos y los resultados esperados.

A continuación, definimos la notación de bicluster utilizada comúnmente en el análisis de expresión génica.

Sea *E* un bicluster que consiste de un conjunto *g* de genes con cardinalidad *|g|* y un conjunto de condiciones *c* con cardinalidad *|c|*, y sea el nivel de expresión del gen *i* bajo la condición *j*, entonces el bicluster se representa como:



En el artículo publicado por Pontes et al.[3] clasifican a los algoritmos de Biclustering para expresión génica en dos. Una clase se compone de aquellos que se basan en una medida de calidad del bicluster para dirigir la búsqueda del algoritmo y una segunda clase con aquellos que no dirigen su búsqueda con evaluaciones de calidad del bicluster. Para el propósito de nuestro proyecto, estamos interesados en la primera clase de algoritmos antes mencionada. En la figura 2 se muestran los algoritmos encontrados por Pontes et al.[3] dentro de dicha clase.

Nuestro proyecto se enfoca en las meta-heurística bio-inspiradas y en particular aquellas que resuelven el problema de biclustering como un problema de optimización multi-objetivo (MOP).

Las primeras referencias utilizadas para nuestro estudio fueron las publicaciones elaboradas por Mitra & Banka [5] en 2016 y el trabajo de Brizuela et al.[6] en 2013. El objetivo que se nos propuso fue implementar el algoritmo evolutivo multi-objetivo MOEA/D, para la identificación de biclusters significativos en datos de microarreglos.

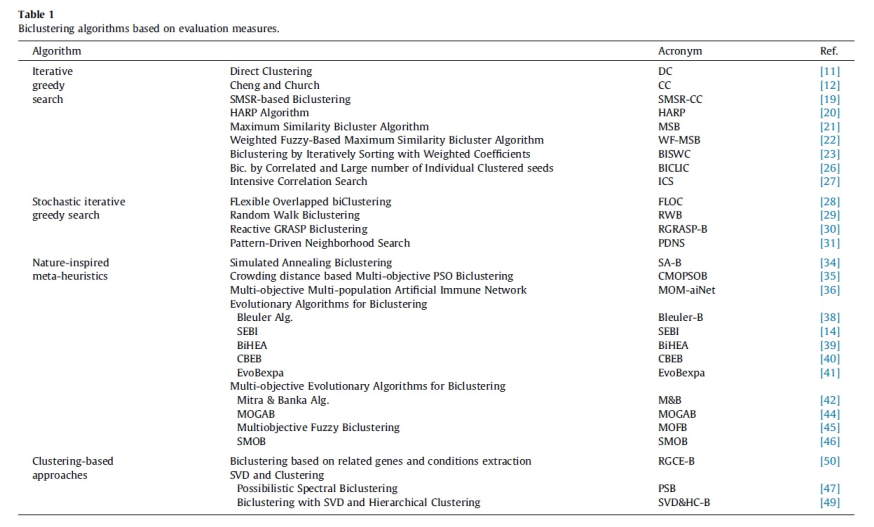


Fig. 2: Clasificación de algoritmos que usan una métrica de evaluación de los biclusters.

1. Materiales y Métodos
   1. Matrices de Expresión utilizadas.

Con el objetivo de validar que nuestra implementación de MOEA/D fuera correcta hemos realizado pruebas con varios conjuntos de datos de expresión genética de referencia, ellos son:

* Levadura Saccharomyces Cerevisiae (con 2884 genes, 17 condiciones y 34 valores nulos, ver en: <http://arep.med.harvard.edu/biclustering/yeast.matrix> )
* Linfoma de células B humanas (4026 genes y 96 condiciones, con 12.3% de valores perdidos, ver en: <http://arep.med.harvard.edu/biclustering/lymphoma.matrix>)
* Datos de expresión de genes de 14 tipos de cáncer. (propuesto por Ramaswamy et. al.[7]) (ver en: <https://github.com/probml/pmtk3/tree/master/bigData/14cancer> )

La disponibilidad de literatura sobre el rendimiento de los algoritmos relacionados en estos conjuntos de datos, es la causa principal de su selección en este estudio. El problema sugiere, el tamaño de un bicluster extraído debe ser lo más grande posible mientras satisface un criterio de homogeneidad. Comparamos los resultados obtenidos al ejecutar los dos primeros conjuntos de datos con los obtenidos por el algoritmo eMOGB implementado por Mitra & Banka [5] en 2016. Estos datos también fueron analizados la primera vez que un enfoque de biclustering se empleó para matrices de expresión por Cheng & Church; por lo que muchos algoritmos de biclustering (incluyendo a los algoritmos evolutivos multiobjetivo) que se encuentran en la literatura, comparan sus resultados analizando estas matrices.

* 1. Algoritmos Evolutivos Multi-Objetivo

El objetivo de los algoritmos evolutivos multi-objetivo (MOEAs) es encontrar un conjunto representativo de las soluciones del óptimo de Pareto. Los MOEAs son una meta-heurística basada en una población de soluciones candidatas y de forma iterativa buscan mejorar los individuos hasta llegar a un conjunto final de soluciones.

Características generales:

* **Fitness:** El objetivo principal es guiar la búsqueda del algoritmo a un conjunto óptimo de Pareto. Existen evaluaciones que implementan el concepto de dominancia.
* **Preservación de la diversidad:** Los algoritmos buscan generar diversos frentes de Pareto pertenecientes al espacio de decisión.
* **Elitismo:** El uso de soluciones elite agiliza el desempeño del algoritmo.

Para nuestro proyecto se utilizaron 2 funciones objetivo, en donde buscamos maximizar el tamaño del bicluster y minimizar el MSR (*Mean Squared Residue*). El MSR es una medida que evalúa la coherencia entre genes y condiciones; fue planteada por primera vez como medida de evaluación a biclusters por Cheng & Church en el año 2000 [10]. Estas dos funciones se encuentran en conflicto ya que entre más grande sea el bicluster, es más probable que aumente el valor de MSR.

Funciones objetivo:

**Definición 2** Función objetivo 1:

(1)

**Definición 3** Función objetivo 2:

Medida de Homogeneidad *Mean Squared Residue* (MSR)

(2)

El límite δ es impuesto para rechazar cualquier bicluster que lo supere.

(3)

Sea el promedio de los elementos de la i-ésima fila del bicluster.

(4)

Sea el promedio de los elementos de la j-ésima fila del bicluster.

(5)

Sea el promedio de **todos** los elementos del bicluster.

(6)

* 1. MOEA/D

El desarrollo de algoritmos evolutivos para la solución de problemas de optimización multiobjetivo, ha tenido progresos considerables en los últimos años [3]. Para nuestra implementación analizaremos el algoritmo MOEA/D publicado por Zhang y Li en 2007 [11], tomando en cuenta algunas consideraciones al momento de seleccionar a los padres, como fue publicado también por Zhang y Li en 2009, como el MOEA/D-DE [12].

Un problema de optimización multiobjetivo (MOP) puede ser definido como:

Maximizar sujeto a

donde es el espacio de la variable de decisión, consiste de funciones objetivo y es llamado el espacio objetivo. Es usual que las funciones objetivo se encuentren en conflicto mutuo, debido a esto, es crucial hacer un balance entre estos. El mejor balance entre objetivos se define en términos de *optimalidad de Pareto*.

Sean se dice que domina a si y solo si para cada y para al menos un índice . El punto es Pareto optimo si no existe otro punto tal que domina a . es llamado el vector objetivo Pareto óptimo [11].

En este trabajo se implementó, en lenguaje C++, un algoritmo evolutivo multiobjetivo basado en descomposición llamado MOEA/D. Dicho algoritmo descompone el MOP en problemas de optimización mono-objetivo y los resuelve simultáneamente manteniendo una población de soluciones. En cada generación, la población se compone de la mejor solución encontrada hasta ese punto para cada subproblema. La relación de vecindario entre estos subproblemas se define en función de las distancias entre sus vectores de coeficientes de agregación. Las soluciones óptimas para dos subproblemas vecinos deberían ser muy similares.

Existen muchos enfoques para convertir el problema de aproximación del Frente Pareto en una cantidad de problemas de optimización mono-objetivo, en este trabajo se utilizó la descomposición de Tchebycheff:

minimizar

sujeto a:

donde es el punto de referencia, es decir, .

**El algoritmo implementado se describe a continuación:**

**1 inicialización**

**1.1** Definir EP = 0

**1.2** Calcular la distancia Euclidiana entre dos vectores de pesos y luego calcular los vectores de peso más cercanos a cada vector de peso.

En este proyecto de distribuyeron los vectores de peso de manera uniforme, dado que tenemos dos objetivos (tamaño del bicluster y MSR), los vectores de pesos serán bidimensionales y deberán cumplir la restricción . Por lo tanto, los vectores de pesos se distribuyen de la siguiente manera:

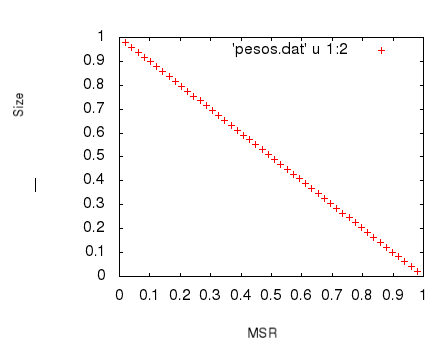
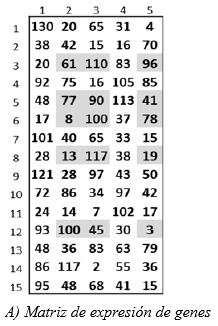
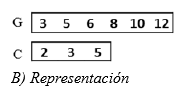
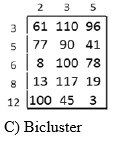


Fig. 5: Distribución uniforme de los pesos.

**1.3** Generar una población inicial de manera aleatoria.

Un individuo se representa mediante un bicluster por medio de dos secuencias de enteros, uno para los genes(G) y otra para las condiciones (C). Si la secuencia de genes tiene un valor indica que el gen es parate del bicluster, lo mismo aplica para secuencia de condiciones. De esta manera se logra tener individuos de tamaño variable. Un ejemplo de esta representación del bicluster se muestra a continuación:

**1.4** Inicializar .

Esta inicialización se hace seleccionando el mejor MSR y mejor tamaño que se hayan generado en la población inicial.

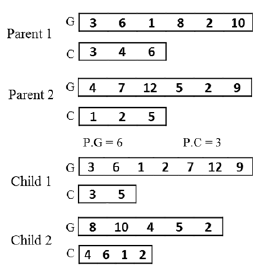
**2.** **Actualización**

For , do

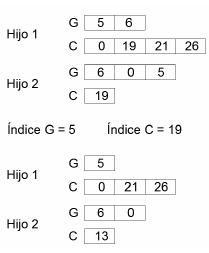
**2.1** Reproducción:

**2.1.1** Selección de padres: En este paso se seleccionan dos individuos que serán los padres, esta selección se realiza seleccionando los padres del vecindario del subproblema o del vecindario de cualquier otro subproblema bajo cierta probabilidad, sesgando esta probabilidad a seleccionar padres del mismo vecindario. Esto se hace con el fin de mantener diversidad en la población [6].

**2.1.2** Cruzamiento: La cruza se realiza con probabilidad a los individuos seleccionados en el paso 2.1.1. Para este proceso se toman los dos padres generados y se generan dos nuevos biclusters (hijos). Se seleccionan dos puntos de cruza de manera aleatoria del padre 1, uno para la secuencia de genes y otro para la secuencia de condiciones. Los puntos de cruza seleccionados contienen los alelos que trabajan como pivotes, P.G y P.C, para la secuencia de genes y condiciones respectivamente. El hijo 1 toma del padre 1 los alelos que sean menores o iguales al pivote, mientras que el hijo 2 recibe alelos del padre 1 que sean mayores al pivote. El hijo 1 es completado con los alelos del padre 2 que sean mayores al pivote, mientras que el hijo 2 es completado con los alelos del padre 2 menores o iguales que el pivote. De esta manera se garantiza que no aparezcan alelos repetidos en la descendencia. En la siguiente figura se muestra un ejemplo de cruza [12].



**2.1.3** Mutación: La muta se realiza con probabilidad . Se toma un índice de manera aleatoria de la matriz de datos, uno para los genes y otro para las condiciones. Si este índice ya está dentro del arreglo se elimina, si no, se agrega. Si al eliminar, ya sea un gen o una condición del arreglo, hace que quedé un conjunto vacío, se toma un valor de manera aleatoria y se agrega al arreglo. En la siguiente figura se muestra un ejemplo de esta rutina.



**2.2** Mejora: En este paso se selecciona uno de los hijos generados, esto se hace seleccionado aquel hijo que domine al otro, si ninguno de los dos lo hace se toma uno de manera aleatoria. Se revisa que el MSR del hijo seleccionado no sobrepase un valor establecido, si este no cumple con la condición, se genera un hijo nuevo utilizando los mismos padres. Este proceso se repite hasta lograr generar un hijo que cumpla con la restricción.

**2.3** Actualización de : Este vector se actualiza comparando su valor con el del hijo previamente generado, si el hijo mejora a alguno de sus elementos, se actualiza con este valor.

**2.4** Actualización de vecindario: Para esto se hace uso del enfoque de Tchebycheff. Para cada índice , si , entonces establecer donde es el hijo generado. Esto se hace con el objetivo de ir minimizando la diferencia entre vectores solución y el vector . Esta actualización se limitó un valor establecido para evitar que una buena solución conduzca al algoritmo a estancarse en un óptimo local.

**2.5** Actualizar EP: En este paso de actualiza el frente Pareto bajo el concepto de dominancia, un individuo I domina a un individuo j si se cumple cualquiera de las dos condiciones siguientes:

* El MSR de la solución I es menor o igual que el MSR de la solución j, y el tamaño de la solución I es mayor que la solución j.
* El tamaño de la solución I es mayor o igual al tamaño de la solución j, y el MSR de la solución I es menor que el de la solución j.

Bajo este concepto, el frente Pareto se actualiza de la siguiente manera:

* Se remueven de frente aquellos vectores dominados por .
* Se agrega al frente si no existen vectores en el frente que lo dominen.

**2.6** Criterio de parada. El algoritmo se detiene al cumplirse un cierto número de generaciones.

1. Resultados

El código fuente, los casos de prueba y la documentación del proyecto están colocados en Github en la url pública <https://github.com/legarcia2904/biclustering> e incluye:

* El reporte del proyecto en formato editable.
* Código fuente, así como una breve explicación de los detalles de compilación y un ejemplo de línea de comando para su ejecución.
* Biblioteca de casos de prueba, con una breve explicación en formato Jupiter Notebook del procesamiento realizado a los datos originales.

Las pruebas fueron realizadas en un equipo con las siguientes características:

Tabla 1: Características del equipo utilizado.

|  |  |
| --- | --- |
| Sistema operativo | GNU/Linux Ubuntu 16.04 |
| RAM | 8 Gb |
| Procesador | Intel i5 1.6 GHz |
| Lenguaje de Programación | C++ v11 |

Los parámetros del algoritmo MOEA/D que mejores resultados dieron fueron los mostrados en la siguiente tabla.

Tabla 2: Parámetros del algoritmo MOEAD/D

|  |  |
| --- | --- |
| **Parámetro** | **Valor** |
| Tamaño de población | 300 |
| Generaciones | 400 |
| Tamaño de vecindario | 30 |
| Límite de soluciones actualizadas | 5 |
|  | 200 |
| Probabilidad de cruzamiento | 1.0 |
| Probabilidad de mutación | 0.4 |
| Probabilidad de mutación de genes | 0.8 |
| Probabilidad de mutación de condiciones | 0.2 |

Comprobamos que nuestra implementación de MOEA/D generaba resultados similares al algoritmo eMOGB implementado por Mitra & Banka [5] utilizando en ambos casos los datos de microarreglos de referencia, como se puede observar en la figura 3.

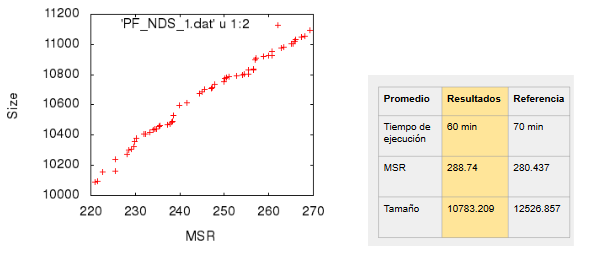


Fig. 3: Validación del algoritmo con datos de levadura (microarreglos).

Para el análisis del algoritmo se realizaron 30 ejecuciones bajo la configuración mostrada en la Tabla 2. En la figura 5 se muestra el Frente Pareto para una de las corridas, donde se tomaron aquellos datos con un tamaño de bicluster superior a 9000 y un MSR mayor a 150.

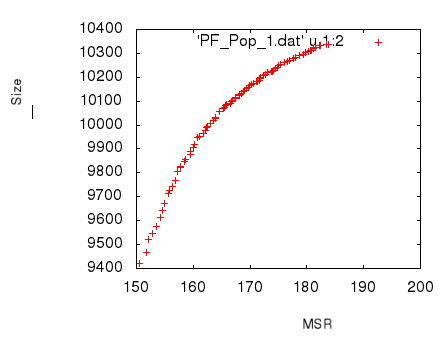


Fig. 6: Frente Pareto de una de las 30 corridas.

Para tener una métrica sobre la calidad de la solución se empleó una tasa CI (*Consistency Index*), el cual representa la relación entre el MSR de un bicluster y su tamaño. Esta relación indica que tan bien se cumplen los dos requisitos de bicluster (los niveles de expresión de genes son similares en un rango de condiciones, es decir, un MSR bajo, y el tamaño es lo más grande posible). De aquí que, un bicluster se considera mejor ya que su valor de CI es más pequeño.

Para cada corrida se calcularon las siguientes variables:

* MSR promedio
* Tamaño promedio del bicluster
* CI promedio

En la tabla 3 se muestran los valores promedio de las 30 corridas.

Tabla 3: Valores promedio de las 30 ejecuciones

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Promedio** | **Promedios máximos** | **Promedios mínimos** |
| MSR | 135.28 | 193.665 | 42.3912 |
| Tamaño | 8358.29 | 10332.432 | 3088.142 |
| CI | 0.0162 | 0.0210 | 0.013 |
| Tiempo de ejecución | 35.23 min | 42.27 min | 31.06 min |

1. Conclusiones

Como parte del proyecto realizamos un proceso de adecuación de datos de RNA-Seq, así como la implementación y validación del algoritmo genético multi-objetivo: MOEA/D para la identificación de biclusters de interés en datos de RNA-Seq.

Durante la validación inicial de nuestra implementación con datos de microarreglos, en relación a los resultados antes reportados en la literatura [6], nuestros valores promedios logran superar a los obtenidos por el algoritmo MOEA de Mitra & Banka en 2006 sobre el conjunto de datos de Levadura Saccharomyces Cerevisiae (con un δ = 300).

Durante el proceso de desarrollo del proyecto pudimos constatar lo sensible que son los algoritmos evolutivos al realizar cambios en la selección de padres y de los descendientes. El primer cambio favorable que se observó durante las primeras pruebas, fue permitir que la selección de padres se realizará también con individuos fuera del vecindario, aunque la probabilidad de esta selección fuera baja (de 0.1) sin duda ayudo a que el algoritmo no se estancara y permitiera explorar aún más es espacio de decisión. Esta modificación al MOEA/D fue tomada de un rediseño más reciente que tuvo el algoritmo en el 2009 llamado MOEA/D-DE [12]. Otro de los cambios más importantes realizados fue el criterio para seleccionar a uno de los dos hijos obtenidos por los operadores genéticos; ya que en un inicio se seleccionaba de manera aleatoria, metodología que se modificó en donde ahora se hacía la selección bajo un criterio de dominancia.

En trabajos futuros se propone validar los biclusters obtenidos, para ver si forman conjuntos validos de expresión, respecto a los reportados por la comunidad biológica, a través de múltiples conjuntos de datos. Además, extender el MOEA/D a una herramienta que funcione para predecir redes de expresión y/o conjuntos de genes *de novo.*

1. Referencias

[1] B. Pontes, R. Giráldez, y J. S. Aguilar-Ruiz, «Biclustering on expression data: A review», *J. Biomed. Inform.*, vol. 57, pp. 163-180, 2015.

[2] S. Busygin, O. Prokopyev, y P. M. Pardalos, «Biclustering in data mining», *Comput. Oper. Res.*, vol. 35, n.o 9, pp. 2964-2987, 2008.

[3] S. C. Madeira y A. L. Oliveira, «Biclustering algorithms for biological data analysis: a survey», *IEEEACM Trans. Comput. Biol. Bioinforma. TCBB*, vol. 1, n.o 1, pp. 24-45, 2004.

[4] J. Xie, A. Ma, A. Fennell, Q. Ma, y J. Zhao, «It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data», *Brief. Bioinform.*, 2018.

[5] S. Mitra y H. Banka, «Multi-objective evolutionary biclustering of gene expression data», *Pattern Recognit.*, vol. 39, n.o 12, pp. 2464-2477, 2006.

[6] C. A. Brizuela, J. E. Luna-Taylor, I. Martinez-Perez, H. A. Guillen, D. O. Rodriguez, y A. Beltran-Verdugo, «Improving an evolutionary multi-objective algorithm for the biclustering of gene expression data», en *Evolutionary Computation (CEC), 2013 IEEE Congress on*, 2013, pp. 221-228.

[7] S. Ramaswamy *et al.*, «Multiclass cancer diagnosis using tumor gene expression signatures», *Proc. Natl. Acad. Sci.*, vol. 98, n.o 26, pp. 15149-15154, 2001.

[8] L. Collado-Torres *et al.*, «Reproducible RNA-seq analysis using recount2», *Nat. Biotechnol.*, vol. 35, n.o 4, p. 319, 2017.

[9] M. I. Love, W. Huber, y S. Anders, «Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2», *Genome Biol.*, vol. 15, n.o 12, p. 550, 2014.

[10] Y. Cheng y G. M. Church, «Biclustering of expression data.», en *Ismb*, 2000, vol. 8, pp. 93-103.

[11] Q. Zhang y H. Li, «MOEA/D: A multiobjective evolutionary algorithm based on decomposition», *IEEE Trans. Evol. Comput.*, vol. 11, n.o 6, pp. 712-731, 2007.

[12] H. Li y Q. Zhang, «Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II», *IEEE Trans. Evol. Comput.*, vol. 13, n.o 2, pp. 284-302, 2009.

[13] K. Mehlhorn, S. Näher, M. Seel, y C. Uhrig, «The LEDA user manual», *Max Plank Inst. Saarbr. Ger.*, 1999.