Análisis de estabilidad de soluciones de Clustering en bioinformática

David Campo y Anibal Rindisbacher

UTN Facultad Regional Santa Fe, Lavaise 610 - S3004EWB Santa Fe,
Página Web: http://www.frsf.utn.edu.ar/
{dncampo; anibal784}@gmail.com

Resumen Este trabajo se realizó con el objetivo de estudiar cómo hacer análisis de estabilidad sobre los algoritmos clásicos de segmentación, en particular en cuanto a cómo medir la estabilidad de grupos solapados, en el contexto de un proyecto final de carrera. El punto de partida fue el estudio de los algoritmos k-medias y mapas auto-organizativos; así como los índices para comparar soluciones de agrupamiento: Fowlkes-Mallows ($\mathcal{FM}$) y Maximum-Match ($\mathcal{MM}$). Luego se realizó un caso de estudio aplicando los algoritmos e índices estudiados. Por último se hace una valoración de los resultados alcanzados.

Keywords: Mapa auto-organizativo, análisis de estabilidad, algoritmos de segmentación, Fowlkes-Mallow, Maximum Match

1. Introducción

El procesamiento y descubrimiento de relaciones en la enorme cantidad de datos que deben analizarse en ciertas áreas de la bioinformática representan actualmente grandes desafíos. Descubrir patrones ocultos en los datos de expresión génica en microarreglos y datos de perfiles metabólicos de plantas de interés económico para la agrobiotecnología, es actualmente un reto ya que el empleo de algún tipo de algoritmo para reconocimiento de patrones sobre estos datos se ve entorpecido por la llamada maldición de la dimensionalidad. Esto pone en evidencia la necesidad de desarrollar nuevas técnicas tendientes a superar las limitaciones de las existentes, principalmente basadas en métodos estadísticos [8].

Por ejemplo, los mapas auto-organizativos (SOM) han probado, recientemente, ser adecuados para las tareas de agrupamiento y visualización de relaciones [12]. Si un SOM es alimentado con datos transcripcionales y de perfiles metabólicos, el mapa de características resultante puede mostrar neuronas activadas por genes co-expresados y metabolitos co-acumulados, mostrando relaciones previamente desconocidas.

Esto se denomina clustering o descubrimiento de clases, en el cual se exploran los datos desde el punto de vista de la existencia o no de relaciones y mecanismos desconocidos y se formulan hipótesis que expliquen estos mecanismos. Por

ejemplo, el algoritmo de agrupación jerárquica es un método determinista basado en una matriz de distancias que ha sido aplicado para esta tarea. En este algoritmo se establecen pequeños grupos de genes/condiciones que tienen un patrón de expresión común y posteriormente construye un dendograma de forma secuencial. Este algoritmo permite inferir un árbol para los genes sobre la base de una matriz de distancia, que luego es podado y a partir de las ramas de este árbol se pueden detectar grupos con características comunes y definir clases que identifiquen a estos grupos. En cuanto a los algoritmos de tipo no-jerárquicos, se comienza a calcular las distancias a partir de un numero pretendido de grupos y se van colocando de forma iterativa los genes en los diferentes grupos hasta minimizar la dispersión interna de cada uno. El algoritmo mas representativo de este tipo de agrupación es k-medias [19][18][20].

En todos estos casos, los datos son analizados bajo la premisa de que genes que se comporten de forma similar pueden ser parte de redes de regulación comunes. Este principio se denomina "guilt-by-association" y postula que un conjunto de genes involucrados en un proceso biológico están co-regulados (y por lo tanto co-expresados) bajo el control de una misma red de regulación. De esta forma, si un gen desconocido está co-expresado con genes conocidos en un determinado proceso biológico, este gen desconocido estará también probablemente involucrado en la misma vía metabólica [8].

Recientemente se han propuesto métodos del tipo *muestreo y agrupamiento*, para analizar la estabilidad de las soluciones encontradas con algoritmos de clustering. Para ello, todo el conjunto de datos es agrupado tomando éste como el agrupamiento de referencia, luego, en el paso de muestreo, se toma una submuestra del conjunto total de datos, para posteriormente, en el paso de agrupamiento, aplicar el algoritmo de clustering sobre dicha submuestra. Para cada agrupamiento encontrado, se calcula la similaridad con el agrupamiento de referencia. Cuando la estructura de los datos es bien representada, la partición de la muestra será muy similar a las particiones de las submuestras. Actualmente existen trabajos publicados que hacen uso de éstos métodos para el análisis de estabilidad de soluciones, por ejemplo Ben-Hur y Guyon [3], realizan el estudio sobre Agrupamiento Jerárquico, o Kuncheva L.I [10] sobre k-medias. Sin embargo, ninguno de éstos métodos han sido aún adaptados y aplicados para SOM.

El presente trabajo está estructurado de la siguiente forma:

Seccion 2: se introducen el conjunto de datos utilizado junto con los algoritmos de segmentación.

Sección 3: se presenta el algoritmo para análisis de estabilidad junto con las métricas, así como también se propone la modificación para se aplicados sobre clusters solapados.

Sección 4: se muestran los resultados obtenidos.

Sección 5: presentación de las conclusiones a partir de los logros obtenidos en el desarrollo del trabajo.

2. Materiales y métodos

2.1. Datos

En esta etapa se presenta *Solanum lycopersicum*, el conjunto de datos utilizado para la aplicación de los algoritmos de agrupamiento: k-medias y SOM.

Solanum lycopersicum Estos datos corresponden al análisis de perfiles metabólicos y transcripcionales de líneas de introgresión (ILs) de *Solanum lycopersicum* (fruto del tomate). Las ILs poseen, en ciertos segmentos de sus cromosomas, porciones introgresadas de una especie salvaje (*Solanum pennelli*). El uso de las ILs permite el estudio y creación de nuevas variedades de especies de tomate, para, por ejemplo, mejorar alguna característica de interés comercial. Este conjunto de datos posee los datos de expresión de 70 metabolitos y 1159 genes, haciendo un total de 1229 datos. Cada uno de estos datos posee 21 dimensiones (características o mediciones de interés, un valor para cada IL)[16].

2.2. Algoritmos de segmentación

Los algoritmos de segmentación (también conocidos como algoritmos de agrupamiento o, en inglés, *clustering*) pertenecen al grupo de métodos de minería de datos definido como no supervisados. El objetivo del clustering no es clasificar, estimar o predecir una variable; sino entender la estructura macroscópica y relaciones entre objetos, considerando las maneras en las que estos son similares y diferentes [15][11]. En otras palabras, se enfoca en segmentar el conjunto completo de datos en subgrupos homogéneos. A los objetos que se parecen en cuanto a cierta similaridad dada, se los agrupa en lo que se llama cluster. Un buen cluster tenderá a maximizar la similitud de los registros que agrupa, mientras que a la vez minimizará dicha semejanza entre objetos de distintos clusters.

A continuación se describirán los algoritmos de clustering más usados hoy en día en bioinformática [14][4][1] y aplicados en este trabajo.

***k*-medias**

El algoritmo k-medias, perteneciente al grupo de los llamados algoritmos particionales, es uno de los más populares y extendidos [19]. La idea detrás de este algoritmo puede describirse como sigue: sea J una función de optimización y sea $x_i \in \Re^d, i = 1, \ldots, N$ un conjunto de datos, el algoritmo tratará de distribuir cada uno de los N puntos en k clusters o particiones $\{C_1, \ldots, C_k\}$, sujeto a la restricción de optimizar un criterio predefinido en J [20]. El objetivo de la función J es minimizar la diferencia entre los patrones de un grupo a la vez que maximiza la diferencia entre datos de clusters diferentes. El algoritmo puede dividirse en 3 etapas:

- Etapa I "Inicialización": existen varias estrategias posibles pero por lo general se inicializa eligiendo $k \in \Re^d$ puntos de forma aleatoria. Otras estrategias pueden incluir elegir los k puntos como solución previa de un subconjunto del conjunto de datos, sea tanto aplicando k-medias u otro algoritmo [19].

- Etapa II "Funcionamiento principal": el cual se puede describir en 2 subetapas: asignación de datos, donde cada registro o punto es asignado al cluster cuyo centroide presenta la mejor similaridad, entendiéndose como mejor el valor que más se ajusta al objetivo de la función de optimización. En este trabajo se utilizará como medida de similaridad la distancia Euclídea, debido a que es la métrica más utilizada en la mayoría de los estudios de bioinformática [14][6][13]. La otra subetapa consiste en el recálculo de centroides dado que, mientras el algoritmo no converja, los puntos están constantemente cambiando de cluster y los centroides de los mismos deben ser recalculados. Para esto se utiliza, por lo general, la media aritmética de todos los puntos pertenecientes al cluster.
- Etapa III "Criterio de convergencia": El algoritmo termina cuando los patrones no cambian de cluster. Puede demostrarse que el algoritmo alcanza convergencia en un número finito de pasos [19].

Un aspecto a considerar es que se deberían realizar varias corridas del algoritmo, debido a que éste puede producir distintos valores dada su aleatoriedad en la inicialización; terminándose por elegir el que produzca el agrupamiento más adecuado dado algún criterio, por ejemplo, mayor cohesión interna de los grupos.

Mapas auto-organizativos (SOM)

Los mapas auto-organizativos (SOM de su nombre en inglés *Self-Organizing Maps*) fueron introducidos por Kohonen en 1982 [9]. Ellos representan un tipo especial de red neuronal que utiliza aprendizaje competitivo, el cual se basa en la idea de unidades (neuronas) que compiten de alguna forma para responder a un conjunto de entradas. Los nodos en la capa de entrada aceptan patrones de entrada y están completamente conectados con los nodos de la capa "competitiva" [11]. SOM consiste en una grilla de neuronas. Cada neurona j es representada por un vector prototipo (o vector de peso sináptico) $\mathbf{w_j} = [\mathbf{w_{j1}}, \ldots, \mathbf{w_{jd}}]$, donde d es la dimensión del vector de entrada. Las neuronas están relacionadas con sus neuronas adyacentes por una relación de vecindad [7].

El principal objetivo de SOM es representar complejos patrones de entrada con vectores prototipos que pueden ser visualizados en una estructura de red de dos dimensiones, mientras preservan la relación de proximidad de los datos originales lo más posible [20].

La red es entrenada iterativamente. En cada paso del entrenamiento, un vector muestra x es tomado al azar del conjunto de datos de entrada. Su similaridad (o distancia) a los vectores prototipos son computadas; todas las neuronas compiten entre sí y sólo una, la más cercana al dato de entrada [1], es activada. Cuando es determinada la neurona ganadora, SOM actualiza no sólo el vector de la neurona ganadora sino también el conjunto de vectores de peso que son vecinas a la misma. Por ello, SOM estructura los nodos de salida (neuronas) tal que los más cercanos durante el entrenamiento son más similares entre sí [11].

[1] en inglés se la llama Best-Matching unit (BMU)

En un SOM, cada neurona es un cluster. Sin embargo, debido a su naturaleza los datos de neuronas adyacentes son más similares entre sí que los de neuronas no adyacentes. Además, cada neurona tiene, al menos, otras 2 neuronas vecinas. Por ejemplo, cualquier neurona que no esté en un borde o esquina del mapa tendrá, como mínimo, 4 vecinas. Esto se denomina vecindad de Von Neumann[2] (la cual se denominará V_n de ahora en más). Tomando esto en cuenta, cada neurona junto con sus vecinas (de acuerdo a la V_n elegida) puede considerarse que pasa a formar parte de varios clusters que se solapan, teniendo a la neurona de interés como centro o punto de unión. Esto será explicado con mayor detalle cuando se aplique el análisis de estabilidad a un SOM.

3. Análisis de estabilidad en Clustering

3.1. Medidas para análisis de estabilidad

Los algoritmos de agrupamiento presentan como desventaja el hecho de que siempre encuentran grupos de datos, incluso cuando éstos no existen. Además, cuando se hace clustering no supervisado, no se sabe qué tan estables son los resultados obtenidos, es decir; no se cuenta con métodos computacionales para saber si los grupos encontrados son reales o no. Es por ello que ha surgido en los últimos años lo que se ha denominado análisis de estabilidad de soluciones de clustering, entendiendo por estabilidad a la tendencia de un modelo de clustering para producir repetidamente agrupamientos similares, desde la misma fuente de datos.

Cuando se hace análisis de estabilidad en clustering, se habla de *estructuras naturales*, las cuales podrían definirse como un grupo de objetos que se pueden inferir de los datos y no son obtenidos como el producto artificial de un algoritmo concreto. Con esto se quiere decir que las estructuras naturales existen y son independientes del algoritmo utilizado para detectarlas. Si bien no hay un acuerdo en cuanto a su definición, hay trabajos que relacionan este concepto con las soluciones de agrupamientos altamente estables bajo perturbaciones de los datos [2].

Uno de ellos es el propuesto por Ben Hur y Guyon [3], en el cual se utiliza un algoritmo denominado de *Muestreo y agrupamiento*, definido en el algoritmo 1. La etapa de muestreo se utiliza para tomar versiones perturbadas de los datos[3], a los que luego en la etapa de agrupamiento se le aplica un algoritmo de segmentación [2] y luego se les mide su similaridad según una determinada métrica. En este enfoque se asume que si un problema tiene estructuras naturales es posible encontrarlas como parte de soluciones de agrupamiento que resultan de versiones perturbadas de los datos. Dicho de otro modo, se puede asumir que si se obtienen repetidamente los mismos grupos al variar ligeramente los datos, dichas soluciones no deberían ser un artefacto del algoritmo de segmentación utilizado.

[2] http://mathworld.wolfram.com/vonNeumannNeighborhood.html
[3] al tomar un subconjunto de los datos se entiende como una forma de pertubación

Algoritmo 1: Exploración de datos basado en el concepto de Muestreo y Agrupamiento

Data:
Data: conjunto de datos
K_{max}: máximo número de grupos
Rep: número de repeticiones del procedimiento de muestreo
Result:
$S(i,k)$: lista con *Rep* similaridades para cada k, donde $i = 1, 2, \ldots, Rep$ y $k = 1, 2, \ldots, K_{max}$

```
1  begin
2      f ← 0,8. Fracción de los datos a considerar en cada partición.
3      for k ← 1 hasta K_max do
4          for i ← 1 hasta Rep do
5              sub_1 ← tomar una fracción f de Data
6              sub_2 ← tomar una fracción f de Data
7              L_1 ← cluster(sub_1, k). Buscar k grupos en la muestra 1.
8              L_2 ← cluster(sub_2, k)
9              interseccin = sub_1 ∩ sub_2
10             S(i,k) ← s(L_1(interseccin), L_2(interseccin)). Computar la
               similaridad entre las etiquetas de los objetos que forman parte de la
               intersección sub_1 y sub_2.
11         end
12     end
13 end
```

En [3] los autores evalúan las distintas soluciones obtenidas midiendo su similaridad mediante el denominado índice Fowlkes-Mallows ($\mathcal{FM}$) para diferente número de grupos, realizando un barrido de diferentes valores de k. En base a los valores que toma $\mathcal{FM}$ se propone elegir la solución estable (mayor valor de $\mathcal{FM}$) que posea la mayor cantidad de grupos.

Matriz de similaridad o contingencia Antes de explicar cada uno de los índices utilizados en este trabajo, es necesario presentar la matriz de similaridad, ya que todas las métricas estudiadas basan su cálculo en ella. Llamaremos $C = \{C_1, \ldots, C_k\}$ y $C' = \{C'_1, \ldots, C'_l\}$, con posiblemente $k \neq l$, a los agrupamientos obtenidos de un conjunto de datos. La matriz de *similaridad* $M(C, C')$ es una matriz de $k \times l$ elementos, donde el ij-ésimo elemento de la matriz (m_{ij}) es igual al número de elementos en la intersección de los grupos C_i y C'_j, tal que

$$m_{ij} = |C_i \cap C'_j|, 1 \leq i \leq k, 1 \leq j \leq l. \tag{1}$$

$\mathcal{FM}$ El índice de Fowlkes and Mallows ($\mathcal{FM}$) fue inicialmente introducido como una medida para la comparación de agrupamiento jerárquico. De todas formas puede usarse para un agrupamiento particional, como lo es el algoritmo k-medias.

El objetivo es cuantificar cuan similares son dos soluciones de clustering. En la ecuación 2 se define el índice Fowlkes-Mallows

$$\mathcal{FM}(C,C') = \frac{\sum_{i=1}^{k}\sum_{j=1}^{l} m_{ij}^2 - n}{\sqrt{(\sum_i |C_i|^2 - n)(\sum_j |C'_j|^2 - n)}} = \frac{n_{11}}{\sqrt{(n_{11}+n_{10})(n_{11}+n_{01})}}, \tag{2}$$

donde n es la cantidad total de datos, cada término n_{ab} $a, b = 0$ ó 1; es el resultado de comparar cada par de datos de una solución contra el mismo par en la otra solución, llegando a una cantidad de $\binom{n}{2}$ comparaciones, y en el cual:

- n_{11}: cantidad de pares que están en el mismo cluster bajo C y C',
- n_{10}: cantidad pares que están en el mismo cluster bajo C, pero en diferentes clusters en C',
- n_{01}: cantidad pares que están en distintos clusters en C, pero en el mismo cluster en C',

$\mathcal{FM}$ puede tomar valores entre 0 y 1. $\mathcal{FM}$= 1 cuando $M(C,C')$ tiene exactamente k celdas no vacías, lo cual sucede cuando k grupos en cada agrupamiento se corresponden exactamente. $\mathcal{FM}$= 0 cuando cada m_{ij} de la matriz de similaridad es 0 ó 1, ésto significa que cada par de objetos que aparecen en el mismo grupo en C son asignados a diferente grupo en C' [5].

Esta métrica tiene la indeseable propiedad de reflejar un alto valor del índice en agrupamientos con bajo número de k y además trabaja sobre la hipótesis de que los clusters son independientes y de tamaño fijo [17].

Algoritmo 2: Algoritmo para calcular $\mathcal{MM}$.

```
Data:
    C, C': soluciones de clustering
Result:
    M: valor de la métrica MM
1  begin
2      resultado ← 0
3      n ← cantidad de datos
4      M ← matriz de contingencia entre C y C'
5      repeat
6          i, j ← buscar índices de máximo valor en M
7          resultado ← resultado + M_{i,j}
8          Borrar fila i y columna j de M
9      until min(filas(M), columnas(M)) > 0;
10     resultado ← resultado / n
11 end
```

$\mathcal{MM}$ El índice Maximum Match ($\mathcal{MM}$) es una medida simétrica presentada por Meila y Heckerman, usada para la comparación de algoritmos. La idea detrás de esta medida es comparar resultados de agrupamientos contra una solución de agrupamiento considerada óptima [17].

La métrica puede ser descrita como sigue: buscar la mayor entrada m_{ij} en la *matriz de similaridad* $M(C, C')$ y asociarlo a los clusters correspondientes C_i y C'_j (ese es el par de grupos con el mayor valor absoluto de solapamiento). Luego, eliminar la fila i y columna j de la matriz y repetir este paso hasta que la matriz tenga tamaño 0. Para finalizar, se deben sumar los valores obtenidos y dividirlos por la cantidad total de elementos. En el algortimo 4 se describe en pseudocódigo su cálculo.

Está métrica está basada en el concepto de tasa de clasificación de aciertos, dada una cierta partición de referencia. Es por esto que los valores mínimos y máximos de este índice pueden variar entre 0 y 1.

3.2. Clusters solapados

En la sección 3.1 se definió matriz de similaridad. Esta definición no tiene en cuenta el concepto de clusters que se solapan, es decir, grupos que comparten datos en alguna medida. Es por ello que ha debido re-definirse el cálculo de la misma para los propósitos de este trabajo. Esto fue necesario, en particular, para poder tener en cuenta la vecindad (V_n) que puede definirse para las neuronas de los mapas auto-organizativos, la cual puede llevar a considerar a un grupo de neuronas como un sólo cluster, la cual solapa su contenido con otro grupo de neuronas (otro cluster). Esto se explica a continuación a través de un ejemplo.

Si se supone que existe un agrupamiento $C = \{C_1, C_2, \ldots, C_n\}$, y existen como mínimo dos clusters C_i y C_j, tal que $C_i \cap C_j \neq \emptyset$, entonces ambos clusters se encuentran solapados. Esto no puede pasar en k-medias pero sí en SOM, al considerar vecindades (V_n) para las neuronas del mapa.

Para SOM, se ha definido una función $vecindad(x, V_n)$ que recibe los siguientes argumentos: x que es la neurona de la cual se quiere conocer sus vecinas, y V_n es la vecindad de Von Neumann a considerar a partir de x. Esta función devolverá un conjunto de datos contenidos en las neuronas vecinas a x, que se encuentran a una distancia $\leq V_n$. En la figura 1 puede observarse un ejemplo de la función de *vecindad*. El círculo azul, centro, representa la neurona a la cual se le desea encontrar sus vecinas y está a una distancia $V_n = 0$ de sí misma. Mientras que los puntos verdes y rojos se encuentran a una distancia $V_n = 1$ y $V_n = 2$, respectivamente, de la neurona del centro. Si se considera $C_i =$ neuronas a una distancia $V_n = 1$ de la neurona azul; y $C_j =$ neuronas a una distancia $V_n = 2$ de la neurona azul, resulta evidente el solapamiento existente entre ambos clusters.

Teniendo en cuenta esto, la nueva forma de calcular la matriz de similaridad para SOM con vecindades es:

$$m_{ij} = |vecindad(i, V_n) \cap vecindad(j, V'_n)|, 1 \leq i \leq k, 1 \leq j \leq l. \quad (3)$$

Una de las formas de etiquetar las neuronas del SOM numerar con números consecutivos de arriba hacia abajo y de izquierda a derecha; tal como se muestra

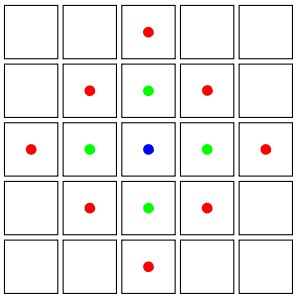

Figura 1: Ejemplo de la función $vecindad(n, V_n)$: neuronas a una distancia $\leq V_n = 1$ (puntos verdes) y $\leq V_n = 2$ (puntos verdes y rojos), respecto de la neurona central (punto azul)

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Figura 2: Numeración de las neuronas en un SOM

en la figura 2. Luego, teniendo como referencia un mapa cuadrado de 4×4, el modo en que quedarían los clusters si se quisiera conformar un cluster con las neuronas vecinas a la neurona 1 del mapa de referencia sería: $vecindad(1, V_n = 1) = \{C_1 \cup C_2 \cup C_5\}$; o bien si se quisiera tomar como un mismo grupo a la neurona 11 del mapa comparativo y sus vecinas con radio de vecindad 1: $vecindad(11, V'_n = 1) = \{C'_7 \cup C'_{10} \cup C'_{11} \cup C'_{12} \cup C'_{15}\}$. Esto puedo comprobarse observando la figura 2.

4. Resultados

Para el análisis de la base de datos del *Solanum lycopersicum* se ha utilizando un rango de tamaños de k en función de la relación con la cantidad patrones. Es decir, no usar una cantidad muy pequeña de clusters, de modo tal que queden agrupaciones con una cantidad muy alta de datos en cada una; y tampoco usar un número muy alto de grupos que haga que cada dato quede agrupado prácticamente solo o de a pares. Es por eso que, teniendo en cuenta la cantidad total de registros del conjunto de datos *Solanum lycopersicum* se decidió explorar el rango $k = n \times n, n = 10, \ldots, 25$.

Para no basarse en una única solución, ya que uno de los algoritmos de segmentación utilizados presenta una fuerte dependencia de su inicialización aleatoria, y para evitar soluciones no representativas, éste proceso se ha repetido 50 veces y en las figuras se informa el resultado promedio de los valores obtenidos en los índices para dichas repeticiones.

4.1. Aplicación del índice $\mathcal{FM}$

En la figura 3 se muestra el resultado del índice $\mathcal{FM}$ utilizando k-medias y SOM como algoritmos de agrupamiento.

Puede observarse como el valor del índice para k-medias aumenta a medida que aumenta la cantidad de clusters, independientemente del muestreo. Debido a que el algoritmo k-medias se inicializa de forma aleatoria, dos ejecuciones del mismo pueden generar dos soluciones de clustering distintas. Es por ello que cabe aclarar que al trabajar con el 100 % de los datos, no se obtiene el máximo valor de $\mathcal{FM}$, 1,00. Por el contrario, el índice tiene una tendencia opuesta cuando se lo aplica al agrupamiento generado por SOM y el 80 % de los datos, mientras que utilizando el 100 % el valor del índice arroja siempre 1,00. Esto se corresponde con el funcionamiento de SOM ya que, utilizando inicialización PCA y manteniendo tanto el mismo conjunto de datos como los parámetros del algoritmo, en cada corrida se genera exactamente el mismo mapa.

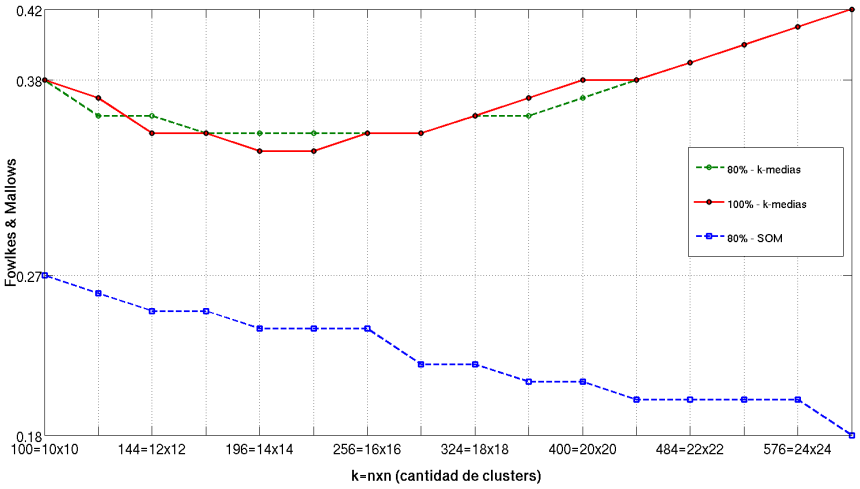

Figura 3: Resultados del índice $\mathcal{FM}$ para k-medias y SOM para el conjunto completo de datos y un muestreo del 80 %

Utilizando ahora la nueva forma de calcular la matriz de contingencia considerando vecindad, explicado en la sección 3, en la figura 4 se muestra la aplica-

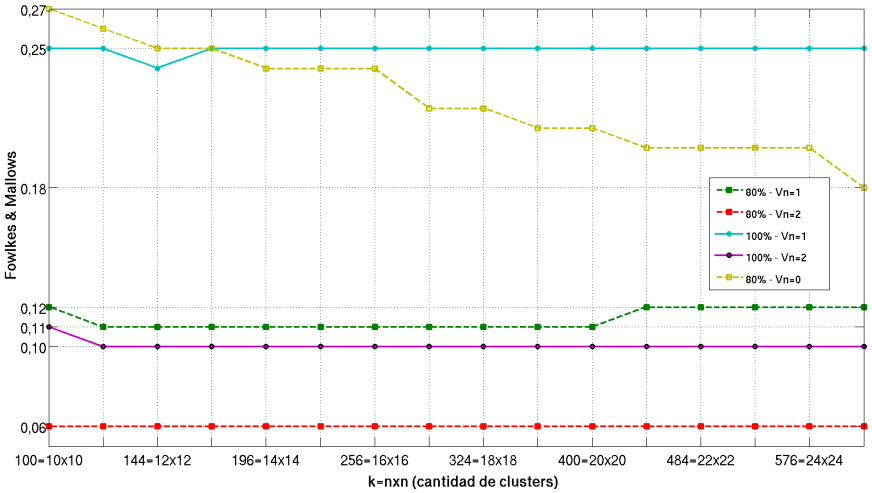

Figura 4: Resultados del índice $\mathcal{FM}$ utilizando vecindad para SOM para el conjunto completo de datos y un muestreo del 80 %

ción del índice $\mathcal{FM}$ entre un mapa de referencia (sin vecindad: $V_n = 0$) y otro comparativo (con vecindades), para diferentes tamaños de mapas y diferentes muestreos. Al igual que en la figura 3, puede observarse como el valor del índice disminuye a medida que aumentan la cantidad de clusters y a medida que se aumenta la vecindad.

Si ambos mapas son similares podría esperarse que los patrones de alguna neurona del mapa de referencia se agrupen en neuronas vecinas del mapa comparativo. Es por esto que al considerar vecindad, y tomar una neurona y sus vecinas como cluster del mapa comparativo se espera agrupar más patrones y aumentar la probabilidad de que un cluster del mapa comparativo se asemeje más, en cuanto a contenido, a uno del mapa de referencia.

Analizando, por ejemplo, la neurona 22, elegida aleatoriamente; de un mapa de 22×22 denominado de referencia, se observa que en el mapa comparativo los datos fueron agrupados en las neuronas $439, 440, 462$ y 483. Tal como se muestra en en el cuadro 1. En la figura 5 puede observarse que si bien los datos se dispersaron en neuronas diferentes, dichas neuronas se encuentran cercanas unas a otras, y si, por ejemplo, se considera vecindad 2 y la neurona 461 como centro, todos los patrones fueron agrupados. Éste agrupamiento fue escogido al azar, sin embargo, el comportamiento se repite a lo largo de la mayoría de las neuronas.

Neurona mapa de referencia	**Patrones**	**Neuronas mapa comparativo**	**Patrones**
22	LE16014	439	LE16014
	LE16A07 trehalose	440	LE16A07 ascorbate trehalose
	LE23D18	462	LE23D18
	LE26M24 maltose ornithine	483	LE26M24 maltose ornithine

Cuadro 1: Detalle de neurona 22 en el mapa de referencia y cómo se agruparon sus patrones en el mapa comparativo.

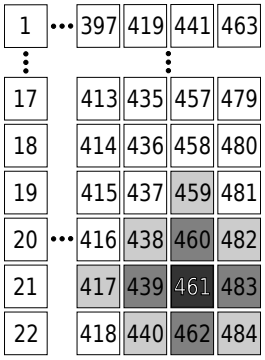

Figura 5: Mapa de 22×22 tomando a la neurona 461 como centro y vecindad $V_n = 2$

4.2. Aplicación del índice $\mathcal{MM}$

En la figura 6 pueden verse los resultados de aplicar el índice $\mathcal{MM}$ con los mismos datos que para $\mathcal{FM}$, y solamente se reportan los resultados para el 80 % de los datos totales en las comparaciones, dado que para el 100 % el valor es siempre 1,00. Puede verse cómo cuando no se toma vecindad ($V_n = 0$,) el índice tiene un cierto valor, y al tomar en cuenta las vecindades se observa que el valor se incrementa notoriamente.

Hasta ahora, se han presentado los resultados de aplicar el análisis de estabilidad muestreando los datos, sin embargo se ha decidido observar los resultados al cambiar otro parámetro del algoritmo, en éste caso el tamaño de mapa. Los resultados obtenidos pueden observase en la cuadro 2. En dicho cuadro podemos observar que tanto el índice $\mathcal{FM}$ como $\mathcal{MM}$ se mantienen relativamente constantes cuando se varían los tamaños de mapas y se mantiene la vecindad, pero poseen tendencias opuestas cuando se considera mayor vecindad; observándose una disminución en el primero y un aumento en el segundo.

Habiendo modificado el cálculo de la matriz para considerar vecindad, se pudo observar que el índice $\mathcal{FM}$ no refleja la situación proyectada sobre el conjunto

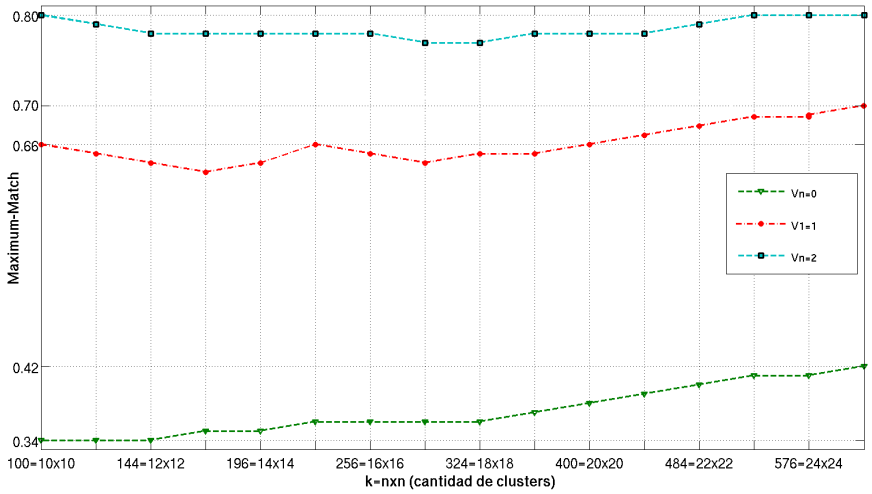

Figura 6: Resultados del índice $\mathcal{FM}$ utilizando vecindad para SOM para el conjunto completo de datos y un muestreo del 80 %

SOM referencia	**SOM comparativo**	$\mathcal{FM}$			$\mathcal{MM}$		
		$V_n = 0$	$V_n = 1$	$V_n = 2$	$V_n = 0$	$V_n = 1$	$V_n = 2$
15×15	13×13	$0,33$	$0,14$	$0,07$	$0,44$	$0,72$	$0,82$
	14×14	$0,34$	$0,14$	$0,07$	$0,45$	$0,77$	$0,87$
	16×16	$0,34$	$0,15$	$0,08$	$0,47$	$0,80$	$0,90$
	17×17	$0,31$	$0,15$	$0,07$	$0,44$	$0,76$	$0,87$

Cuadro 2: Índices $\mathcal{FM}$ y $\mathcal{MM}$ para mapas de referencia de 15×15 para el conjunto de datos *Solanum lycopersicum*

de datos estudiado. En éste trabajo se presentó únicamente como ejemplo la neurona 22, mientras que el índice $\mathcal{MM}$ si lo hace.

5. Conclusiones

En este trabajo se realiza un estudio de los principales algoritmos de segmentación utilizados en bioinformática, como son k-medias y mapas auto- organizativos. Se buscan y analizan métricas para evaluar la similaridad de soluciones de clustering y se las utiliza, junto con el algoritmo de muestreo y agrupamiento, para realizar un análisis de estabilidad sobre un caso de estudio: *Solanum lycopersicum*. A su vez se propone una modificación al cálculo de la matriz de contingencia para poder considerar clusters solapados en la aplicación de los índices estudiados.

Referencias

1. Bandyopadhyay, S.; Bhattacharyya, M. A biologically inspired measure for coexpression analysis. IEEE/ACM Trans. Comput. Biology Bioinform., vol. 8, no. 4, pp. 929-942 (2011)
2. Bayá, A. Aplicación de algoritmos no supervisados a datos biológicos" , tesis doctoral Doctorado en Ingeniería, Universidad Nacional de Rosario (Marzo 2011).
3. Ben-Hur, A. y Guyon, I. Detecting stable clusters using principal component analysis, en M. Brownstein y A. Khodursky, eds., "In Methods in Molecular Biology", Ed. Humana press, pp. 159-182. (2003).
4. Datta, S.; Datta, S. Evaluation of clustering algorithms for gene expression data. BMC Bioinformatics, vol. 7, pp. S17+ (2006).
5. Fowlkes, E.; Mallows, C. A method for comparing two hierarchical clusterings, Ed. J. Am. Stat. Assoc, Vol. 78, pp. 553-569 (1983).
6. Handl, J.; Knowles, J.; Kell, D. Çomputational cluster validation in post-genomic data analysis". Bioinformatics, vol 21, no. 15, pp. 3201-3212 (2005).
7. Juha V.; Esa A. Clustering of the Self-Organizing Map. IEEE Transactions on Neural Networks, Vol.11 No 3 (2000).
8. Kelemen, A.; Abraham, A.; Chen, Y. (Eds.), Computational Intelligence in Bioinformatics, Series: Studies in Computational Intelligence, Vol. 94, XVI, 26 pp. 104. (2008).
9. Kohonen, T. Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43:59-69 (1982).
10. Kuncheva, L. Evaluation of stability ok k-means cluster ensembles with respect to random initialization. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, pp. 1798-1808 (2006)
11. Milone, D.; Stegmayer, G.; Gerard, M.; Kamenetzky, L.; López, M.; Carrari, F.; Chapter 14 "Analysis and Integration of Biological Data: A Data Mining Approach using Neural Networks" in "Knowledge Discovery Practices and Emerging Applications of Data Mining: Trends and New Domains". Ed. IGI Global, pp. 287-314 (2011)
12. Milone, D.; Stegmayer, G; Kamenetzky, L.; López, M.; Min Lee, Je; Giovannoni, James J.; Carrari, F. "*omeSOM: a software for clustering and visualization of transcriptional and metabolite data mined from interspecific crosses of crop plants", BMC Bioinformatics 11:438. (2010).
13. Pihur, V.; Datta, S.; Datta, S. Weighted rank aggregation of cluster validation measures: a Monte Carlo cross-entropy approach. Bioinformatics, vol. 23, no. 13, pp. 1607-1615 (2007).
14. Rubel, O.; Weber, G.; Huang, M.; Bethel, E.; Biggin, M; Fowlkes, C.; Hendriks, C.; Keranen, S.; Eisen, M.; Knowles, D.; Malik, J.; Hagen, H.; Hamann, B. Ïntegrating data clustering and visualization for the analysis of 3d gene expression data". IEEE/ACM Trans. Comput. Biol. Bioinformatics, Vol 7, pp. 64-79 (2010).
15. Skillicorn David. Understanding complex datasets. Data mining with matrix decompositions. Ed. Chapman & Hall / CRC (2007).
16. Stegmayer G., Milone D., Kamenetzky L., López M., Carrari F., Neural Network Model for Integration and Visualization of Introgressed Genome and Metabolite Data, IEEE International Joint Conference on Neural Networks (IJCNN), Atlanta, EEUU, pp. 2983 – 2989, Junio (2009).
17. Wagner, S.; Wagner, D. Comparing Clusterings - An Overview. (2007).

18. Witten, I.; Frank, E.; Hall, M. Data Mining: Practical Machine Learning Tools and Techniques (Third Edition). Ed. ELSEVIER (2011).
19. Wu, X.; Kumar, V.; The top ten algorithms in data mining. Ed. Chapman & Hall / CRC (2009).
20. Xu, R.; Wunsch, D. Clustering. IEEE Press Series on Computational Intelligence. Ed. Wiley (2009).