Creación de una tipología de genes mediante técnicas de Data Mining para *Drosophila*

Ramón Álvarez[1] Flavio Pazos[2] Adrián Valentín [2]
Curso de data Mining-2012,Instituto Pasteur
[1]IESTA(Instituto de Estadística)
[2]IIBCE (Instituto de Investigaciones Biológicas Clemente Estable)

IESTA

Montevideo,16 de Setiembre

Esquema de la Presentación

1. Presentación del problema
2. Anlisis Previo
3. Algoritmos Usados
   - Método Jerárquico
   - Método K-means
   - Método PAM
   - Metodo SILUETA
4. Resultados preliminares
5. Análisis de los clusters en referencia a BP
6. Futuros pasos

Presentación del problema

Posee varias características que lo han convertido en un organismo modelo clásico en la biología moderna.

- breve ciclo de vida
- fácil y barato para mantener grandes poblaciones en el laboratorio
- genoma totalmente secuenciado (aproximadamente 15.016 genes)
- reducido número de cromosomas (4 pares)
- aproximadamente el $61\,\%$ de los genes de enfermedades humanas que se conocen tienen una contrapartida identificable en Drosophila
- $50\,\%$ de las secuencias proteícas de Drosophila tiene análogos en los mamíferos.

Ciclo de vida en la mosca

Embrión: 12 muestras cada 2 hs (0-2 hs ….. 22-24 hs)

Larva: 6 muestras: L1, L2, L3 12 hr post-molt, L3 dark blue gut, L3 light blue gut, L3 clear gut

Pupa: 6 muestras: WPP, WPP + 12 hr, WPP + 24 hr, WPP + 2 days, WPP + 3 days, WPP + 4 days

Adulto: 3 muestras: 1 día, 5 días y 30 días, diferenciadas por sexo

Para este trabajo usaremos solo las datos que cubren el desarrollo (embrión, larva y pupa)

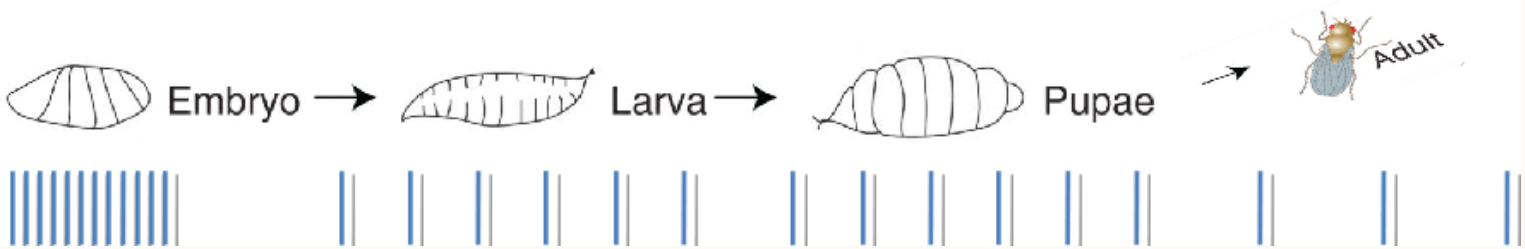


Figura: distintos niveles de expresión a lo largo de ciclo de vida

Matriz de Datos(1)

Se tiene una matriz $Z_{i,k}$ que corresponde a un total de 13677 genes y que se puede particionar en $Z_{i,k} = [I_{i,k}|E^t_{i,k}|S_{i,k}|F_{i,k}|]$

- $I_{i,k}$ Variables de identificación
- $E^t_{i,k}$ Variables de Niveles de expresión cada 2 horas en las 3 etapas: Emb rión,Larva,Pupa
- $S_{i,k}$ Scores de trayectoria con referencia BP para los 3 etapas
- $F_{i,k}$ Clasificaciones de los genes de acuerdo a ....

Transformación de la Matriz de Datos

Los datos de niveles de expresión se transforman relativizando el nivel de $E^t_{i,k}$ contra el nivel máximo individual

$$Exp^t_{i,k} = \frac{E^t_{i,k}}{max(E^{\cdot\cdot}_{i,k})} \tag{1}$$

$$Exp^t_{i,k} \in [0;1] \tag{2}$$

Perfiles de los genes

```
> summary(cbind(nivel.e.min,nivel.e,nivel.e.max,nivel.e.sum,creci.expresion))
  nivel.e.min           nivel.e          nivel.e.max        nivel.e.sum
 Min.   :   0.0000   Min.   :   0.000   Min.   :    0.00   Min.   :     0.00
 1st Qu.:   0.0000   1st Qu.:   3.213   1st Qu.:   13.57   1st Qu.:    77.12
 Median :   0.2989   Median :  11.349   Median :   38.88   Median :   272.39
 Mean   :  11.3861   Mean   :  46.031   Mean   :  174.47   Mean   :  1104.73
 3rd Qu.:   5.3483   3rd Qu.:  29.001   3rd Qu.:   91.87   3rd Qu.:   696.02
 Max.   :1895.9400   Max.   :5993.770   Max.   :20189.00   Max.   :143850.47
 creci.expresion
 Min.   :    0.0
 1st Qu.:    2.0
 Median :   16.0
 Mean   :  134.9
 3rd Qu.:   53.0
 Max.   :20189.0
```

Ejemplo de niveles de expresión transformados

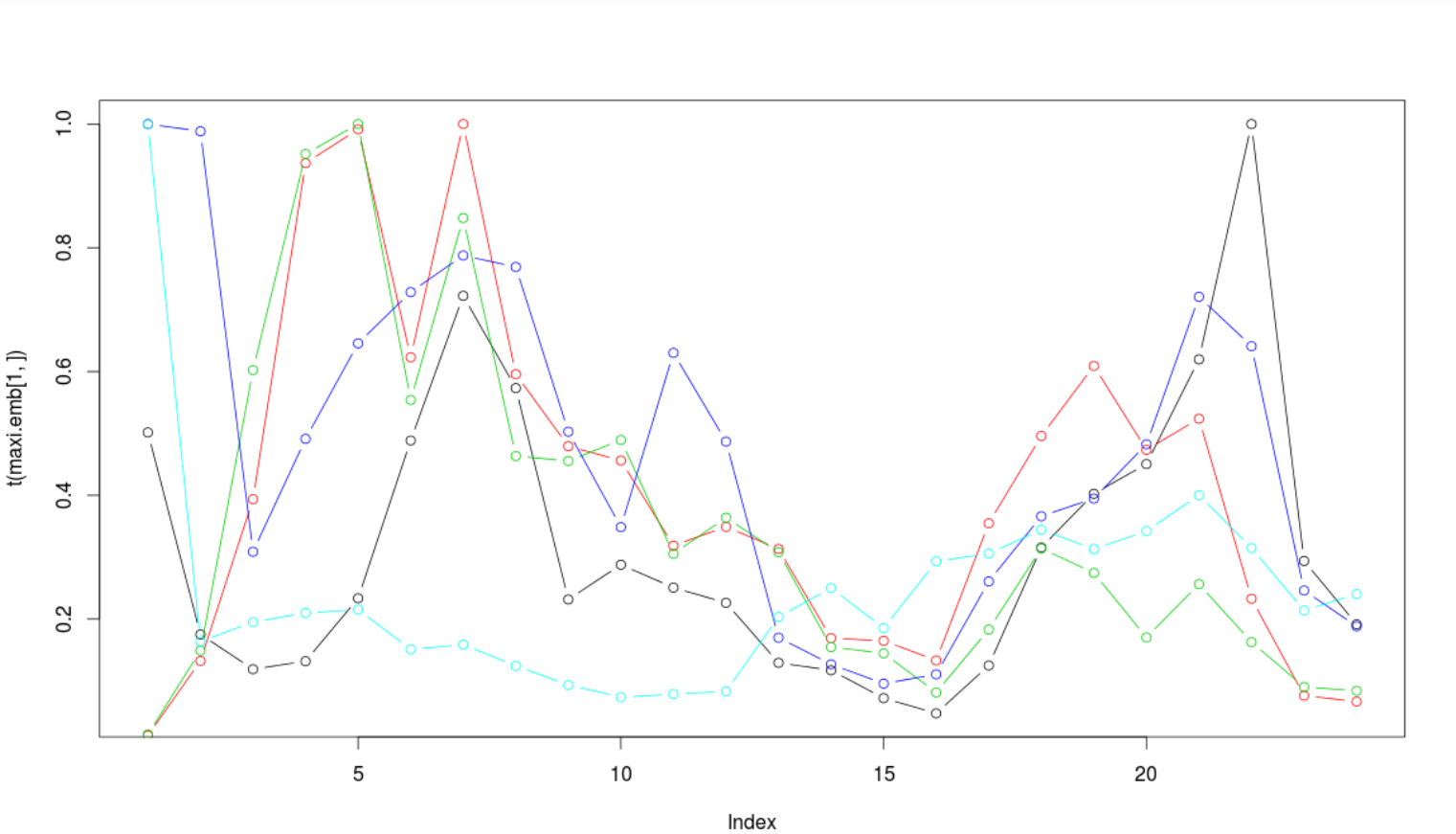


Figura: niveles de expresión transformados

Diferentes Algoritmos y librerías

Para los cálculos hechos para los diferentes algoritmos se usa el sistema R [1] y las librerías que se detallan a continuación

- Algoritmos Jerárquicos-library(fastcluster)
- Algoritmos NO Jerárquicos-library(cluster)- **K-means**,**Pam**,**Clara**,**Fanny**
- library(clv)-Homegeneidad de los grupos

Método jerárquicos-Algoritmo de WARD

Los métodos jerárquicos se caracterizan por generar una serie de particiones encajadas y requieren la definición de una distancia. Inicialmente, cada objeto se le asigna a su propio grupo, y entonces los algoritmos proceden iterativamente, en cada etapa unen los dos grupos más similares, continuando hasta que sólo quede un solo grupo.[2]

Método jerárquicos-Algoritmo de WARD

Los métodos jerárquicos se caracterizan por generar una serie de particiones encajadas y requieren la definición de una distancia. Inicialmente, cada objeto se le asigna a su propio grupo, y entonces los algoritmos proceden iterativamente, en cada etapa unen los dos grupos más similares, continuando hasta que sólo quede un solo grupo.[2]

$$T = W + B \tag{3}$$

Donde T es la matriz de varianzas y covarianzas del total, W la matriz de varianzas y covarianzas dentro de los grupos y B la matriz de varianzas y covarianzas entre grupos. En este caso para determinar con que cantidad de grupos trabajar, existen varias reglas de detención, de las cuales se presentan algunas

Método jerárquico-Reglas de Detención

R cuadrado: Establece la relación entre la variación explicada y la variación total, donde la variación explicada la representa la estructura de grupos hallada en cada nivel.

$$R^2 = 1 - \frac{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \sum_{j=1}^{J} (x_{(ij(k))} - \bar{x}_{kj})^2}{\sum_{i=1}^{I} \sum_{j=1}^{J} (x_{(ij)} - \bar{x}_j)^2} \tag{4}$$

Método jerárquico-Reglas de Detención

R cuadrado: Establece la relación entre la variación explicada y la variación total, donde la variación explicada la representa la estructura de grupos hallada en cada nivel.

$$R^2 = 1 - \frac{\sum_{k=1}^{K}\sum_{i=1}^{n_k}\sum_{j=1}^{J}(x_{(ij(k))} - \bar{x}_{kj})^2}{\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{(ij)} - \bar{x}_j)^2} \tag{4}$$

Regla de Calinski (llamada Pseudo F):se busca máximos locales

$$pseudo\ F = \frac{R^2/(k-1)}{(1-R^2)/(n-k)} \tag{5}$$

Método jerárquico-Reglas de Detención

R cuadrado: Establece la relación entre la variación explicada y la variación total, donde la variación explicada la representa la estructura de grupos hallada en cada nivel.

$$R^2 = 1 - \frac{\sum_{k=1}^{K}\sum_{i=1}^{n_k}\sum_{j=1}^{J}(x_{(ij(k))} - \bar{x}_{kj})^2}{\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{(ij)} - \bar{x}_j)^2} \tag{4}$$

Regla de Calinski (llamada Pseudo F):se busca máximos locales

$$pseudo\ F = \frac{R^2/(k-1)}{(1-R^2)/(n-k)} \tag{5}$$

Test de Duda-Hart (pseudo t^2):disminución en la suma de cuadrados residuales (variación intragrupos, o variación en los grupos)

$$pseudo\ t^2 = \frac{trW_G + trW_L}{trW_{GL}} \tag{6}$$

K-MEANS(1)

A partir de un conjunto de de n observaciones $(x_1, x_2, ..., x_n)$, que se puede considerar un vector p, el método de k-means buscar encontrar una partición de los n individuos en k subconjuntos con $k \leq n$,de manera de minimizar la suma de cuadrados intraclase (SCIC) de los subconjuntos S_i:

K-MEANS(1)

A partir de un conjunto de de n observaciones $(x_1, x_2, ..., x_n)$, que se puede considerar un vector p, el método de k-means buscar encontrar una partición de los n individuos en k subconjuntos con $k \leq n$,de manera de minimizar la suma de cuadrados intraclase (SCIC) de los subconjuntos S_i:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \mu_i \right\|^2 \tag{7}$$

considerando μ como el centroide de los puntos en el grupo S_i

K-MEANS(1)

A partir de un conjunto de de n observaciones $(x_1, x_2, ..., x_n)$, que se puede considerar un vector p, el método de k-means buscar encontrar una partición de los n individuos en k subconjuntos con $k \leq n$,de manera de minimizar la suma de cuadrados intraclase (SCIC) de los subconjuntos S_i:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \mu_i \right\|^2 \tag{7}$$

considerando μ como el centroide de los puntos en el grupo S_i

Al inicio, todos los centros de los conglomerados están en la media de las celdas de Voronoi (que se puede interpretar como el conjunto de puntos de los datos que están más cerca del centro de ese grupo que de cualquier otro grupo).

K-MEANS(2)

El algoritmo funciona de la siguiente manera:

1. Se eligen en forma aleatoria los centros iniciales. Queda entonces la siguiente secuencia $m_1.m_2,...,m_k$ de k centros

K-Means(2)

El algoritmo funciona de la siguiente manera:

1. Se eligen en forma aleatoria los centros iniciales. Queda entonces la siguiente secuencia $m_1.m_2,...,m_k$ de k centros
2. Se asigna cada observación al cluster con la media mas próxima, es decir que la partición queda determinada por el diagrama de Voronoi que se generó con las medias iniciales

K-MEANS(2)

El algoritmo funciona de la siguiente manera:

1. Se eligen en forma aleatoria los centros iniciales. Queda entonces la siguiente secuencia $m_1.m_2,...,m_k$ de k centros
2. Se asigna cada observación al cluster con la media mas próxima, es decir que la partición queda determinada por el diagrama de Voronoi que se generó con las medias iniciales
3. Se calcula los S_i de la siguiente manera

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\| \leq \left\| x_p - m_j^{(t)} \right\| \ \forall \ 1 \leq j \leq k \right\} \tag{8}$$

donde cada uno de los x_p queda asignado a uno de los $S_i^{(t)}$.

K-MEANS(2)

El algoritmo funciona de la siguiente manera:

1. Se eligen en forma aleatoria los centros iniciales. Queda entonces la siguiente secuencia $m_1.m_2,...,m_k$ de k centros
2. Se asigna cada observación al cluster con la media mas próxima, es decir que la partición queda determinada por el diagrama de Voronoi que se generó con las medias iniciales
3. Se calcula los S_i de la siguiente manera

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\| \leq \left\| x_p - m_j^{(t)} \right\| \ \forall \ 1 \leq j \leq k \right\} \tag{8}$$

   donde cada uno de los x_p queda asignado a uno de los $S_i^{(t)}$.
4. El algoritmo se actualiza calculando las nuevas medias del grupo

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j \tag{9}$$

K-MEANS(2)

El algoritmo funciona de la siguiente manera:

1. Se eligen en forma aleatoria los centros iniciales. Queda entonces la siguiente secuencia $m_1.m_2,...,m_k$ de k centros
2. Se asigna cada observación al cluster con la media mas próxima, es decir que la partición queda determinada por el diagrama de Voronoi que se generó con las medias iniciales
3. Se calcula los S_i de la siguiente manera

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\| \leq \left\| x_p - m_j^{(t)} \right\| \ \forall \ 1 \leq j \leq k \right\} \tag{8}$$

   donde cada uno de los x_p queda asignado a uno de los $S_i^{(t)}$.
4. El algoritmo se actualiza calculando las nuevas medias del grupo

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j \tag{9}$$

5. El algoritmo se detiene luego que al reasignar alguna observación a otro grupo no hay cambios menores a una tolerancia prefijada en la (SCIC)

K-MEDOIDES MÉTODO PAM

El algoritmo PAM (partition around medoids), se basa en la búsqueda de k objetos representativos o medoides entre las observaciones del conjunto de datos, de manera que representen adecuadamente la estructura de los datos. Un *medoide* se podría definir como el objeto perteneciente a un cluster o conglomerado, cuyo promedio de disimilaridad a todos los objetos en el conglomerado es mínima, es decir, que se puede considerar como el punto más céntrico de la agrupación considerada.

K-MEDOIDES MÉTODO PAM

El algoritmo PAM (partition around medoids), se basa en la búsqueda de k objetos representativos o medoides entre las observaciones del conjunto de datos, de manera que representen adecuadamente la estructura de los datos. Un *medoide* se podría definir como el objeto perteneciente a un cluster o conglomerado, cuyo promedio de disimilaridad a todos los objetos en el conglomerado es mínima, es decir, que se puede considerar como el punto más céntrico de la agrupación considerada.

1. Inicialización: Selección al azar de k de los n puntos de datos como los candidatos a medoides *fase de construcción*).

K-MEDOIDES MÉTODO PAM

El algoritmo PAM (partition around medoids), se basa en la búsqueda de k objetos representativos o medoides entre las observaciones del conjunto de datos, de manera que representen adecuadamente la estructura de los datos. Un *medoide* se podría definir como el objeto perteneciente a un cluster o conglomerado, cuyo promedio de disimilaridad a todos los objetos en el conglomerado es mínima, es decir, que se puede considerar como el punto más céntrico de la agrupación considerada.

1. Inicialización: Selección al azar de k de los n puntos de datos como los candidatos a medoides *fase de construcción*).
2. Se asigna cada observación al cluster con el medoide mas próximo, dependiendo de la distancia elegida (euclidiana, Manhattan o Minkowski). Luego se encuentra un mínimo local para la función objetivo, es decir, una solución de tal manera el cambio de observación con un medoide haga que la función objetivo decrezca (esto se denomina la fase de intercambio).

K-MEDOIDES MÉTODO PAM

El algoritmo PAM (partition around medoids), se basa en la búsqueda de k objetos representativos o medoides entre las observaciones del conjunto de datos, de manera que representen adecuadamente la estructura de los datos. Un *medoide* se podría definir como el objeto perteneciente a un cluster o conglomerado, cuyo promedio de disimilaridad a todos los objetos en el conglomerado es mínima, es decir, que se puede considerar como el punto más céntrico de la agrupación considerada.

1. Inicialización: Selección al azar de k de los n puntos de datos como los candidatos a medoides *fase de construcción*).
2. Se asigna cada observación al cluster con el medoide mas próximo, dependiendo de la distancia elegida (euclidiana, Manhattan o Minkowski). Luego se encuentra un mínimo local para la función objetivo, es decir, una solución de tal manera el cambio de observación con un medoide haga que la función objetivo decrezca (esto se denomina la fase de intercambio).
3. Se repiten los pasos anteriores hasta que los medoides queden estables (es decir que haya cambios en los medoides).

Clusters Difusos (Fanny)

- A diferencia de los métodos anteriores donde se obtiene una partición en k subconjuntos , es decir $N = \bigcup_1^n x_i = \bigcup_1^k \bigcup_{j=1}^{j=n_k} c_j$, donde c_j son los clusters determinados mediante algún algoritmo, en el agrupamiento *difuso* (fuzzy), cada observación se 'extiende' a través de los distintos grupos.

Clusters Difusos (Fanny)

- A diferencia de los métodos anteriores donde se obtiene una partición en k subconjuntos , es decir $N = \bigcup_1^n x_i = \bigcup_1^k \bigcup_{j=1}^{j=n_k} c_j$, donde c_j son los clusters determinados mediante algún algoritmo, en el agrupamiento *difuso* (fuzzy), cada observación se 'extiende' a través de los distintos grupos.
- En este método se puede establecer: m parámetro de incertidumbre (fuzziness parameter), v_k es el centro del cluster c y u_{ik} el grado de pertenencia del individuo i al grupo k.

Clusters Difusos (Fanny)

- A diferencia de los métodos anteriores donde se obtiene una partición en k subconjuntos , es decir $N = \bigcup_1^n x_i = \bigcup_1^k \bigcup_{j=1}^{j=n_k} c_j$, donde c_j son los clusters determinados mediante algún algoritmo, en el agrupamiento *difuso* (fuzzy), cada observación se 'extiende' a través de los distintos grupos.
- En este método se puede establecer: m parámetro de incertidumbre (fuzziness parameter), v_k es el centro del cluster c y u_{ik} el grado de pertenencia del individuo i al grupo k.
- Si se tiene n el número de observaciones, k el número de clusters, r es el parámetro de ajuste del modelo y $d(i,j)$ la disimilaridad entre las observaciones i y j, el método *fanny* busca minimizar la función objetivo [3]

Clusters Difusos (Fanny)

- A diferencia de los métodos anteriores donde se obtiene una partición en k subconjuntos , es decir $N = \bigcup_1^n x_i = \bigcup_1^k \bigcup_{j=1}^{j=n_k} c_j$, donde c_j son los clusters determinados mediante algún algoritmo, en el agrupamiento *difuso* (fuzzy), cada observación se 'extiende' a través de los distintos grupos.
- En este método se puede establecer: m parámetro de incertidumbre (fuzziness parameter), v_k es el centro del cluster c y u_{ik} el grado de pertenencia del individuo i al grupo k.
- Si se tiene n el número de observaciones, k el número de clusters, r es el parámetro de ajuste del modelo y $d(i,j)$ la disimilaridad entre las observaciones i y j, el método *fanny* busca minimizar la función objetivo [3]

$$\sum_{v=1}^{v=n} \frac{\sum_1^i \sum_1^j u_{i,v}^r u_{j,v}^r d(i,j)}{2\sum_1^n u_{j,}^r} \tag{10}$$

Graficos de SILUETA

- El método *Silueta* en realidad no es método de clustering en si mismo sino un método de interpretación y validación del número de conglomerados o cluster hallado[4].

Graficos de SILUETA

- El método *Silueta* en realidad no es método de clustering en si mismo sino un método de interpretación y validación del número de conglomerados o cluster hallado[4].
- Tiene la ventaja que puede ser utilizada para datos que hayan sido clasificados a través de cualquier método como por ejemplo *k-medias* o *k-medoides* .

GRAFICOS DE SILUETA

- El método *Silueta* en realidad no es método de clustering en si mismo sino un método de interpretación y validación del número de conglomerados o cluster hallado[4].
- Tiene la ventaja que puede ser utilizada para datos que hayan sido clasificados a través de cualquier método como por ejemplo *k-medias* o *k-medoides* .
- Para cada observación i, $a(i)$ es la disimilaridad promedio de i con todos los demás observaciones dentro del mismo grupo. Para los demás clusters C se define $d(i,C)$ como la disimilaridad promedio de i con los datos de C. Se repite el procedimiento para cada grupo del cual el i no es miembro y se determina $b(i) := min_C d(i,C)$, que representa la mínima disimilaridad promedio de i con cualquier otro grupo, lo que representa la disimilaridad entre i y los clusters vecinos.

GRAFICOS DE SILUETA

- El método *Silueta* en realidad no es método de clustering en si mismo sino un método de interpretación y validación del número de conglomerados o cluster hallado[4].
- Tiene la ventaja que puede ser utilizada para datos que hayan sido clasificados a través de cualquier método como por ejemplo *k-medias* o *k-medoides* .
- Para cada observación i, $a(i)$ es la disimilaridad promedio de i con todos los demás observaciones dentro del mismo grupo. Para los demás clusters C se define $d(i,C)$ como la disimilaridad promedio de i con los datos de C. Se repite el procedimiento para cada grupo del cual el i no es miembro y se determina $b(i) := min_C d(i,C)$, que representa la mínima disimilaridad promedio de i con cualquier otro grupo, lo que representa la disimilaridad entre i y los clusters vecinos.
- Se define entonces el estadístico $S(i)(silueta)$ como

$$S(i) = \frac{(b(i) - a(i))}{max[(b(i) - a(i))]} \tag{11}$$

K-MEANS CON 5 GRUPOS

```
> t(cl$centers)
             1          2         3          4         5
E2  0.09451729 0.04549325 0.7005092 0.09126290 0.6224316
E4  0.06976785 0.03663106 0.6669122 0.06006982 0.4853914
E6  0.05200680 0.04144900 0.6622698 0.06126588 0.5117242
E8  0.04874375 0.03778847 0.5982512 0.07166238 0.5595062
E10 0.05924937 0.04415054 0.5481369 0.09046413 0.5966156
E12 0.07304989 0.05128013 0.4372425 0.15618972 0.6530265
E14 0.09475699 0.06028291 0.4013951 0.20763792 0.6458198
E16 0.10864135 0.05242655 0.2827742 0.30099194 0.5200064
E18 0.12262361 0.05739138 0.2323882 0.43902292 0.4894492
E20 0.12002702 0.05750560 0.1758907 0.58814960 0.4372801
E22 0.07951006 0.04914118 0.2300487 0.63396929 0.4261991
E24 0.09446943 0.06095481 0.2321860 0.67976249 0.4520072
L1  0.09380911 0.04950302 0.2014166 0.57615186 0.4526143
L2  0.14961218 0.04372531 0.1751300 0.41449886 0.4525066
L3  0.15759944 0.05075910 0.1420976 0.37630754 0.3767204
L4  0.17088339 0.14060169 0.1475242 0.28223697 0.3495223
L5  0.15614521 0.26413651 0.2103130 0.24859360 0.4394517
L6  0.15464978 0.34773477 0.2588040 0.24250181 0.5332572
P1  0.14566715 0.33641448 0.2656134 0.21647214 0.5444819
P2  0.15892571 0.55961192 0.2858562 0.24666518 0.5280605
P3  0.11275222 0.76608002 0.3330679 0.23786608 0.5857101
P4  0.17567576 0.85030185 0.2505225 0.30594208 0.5451092
P5  0.17230391 0.73836655 0.1600613 0.34498603 0.4314000
P6  0.25267440 0.59334493 0.1444121 0.38311299 0.3641153
```

Método K-Means

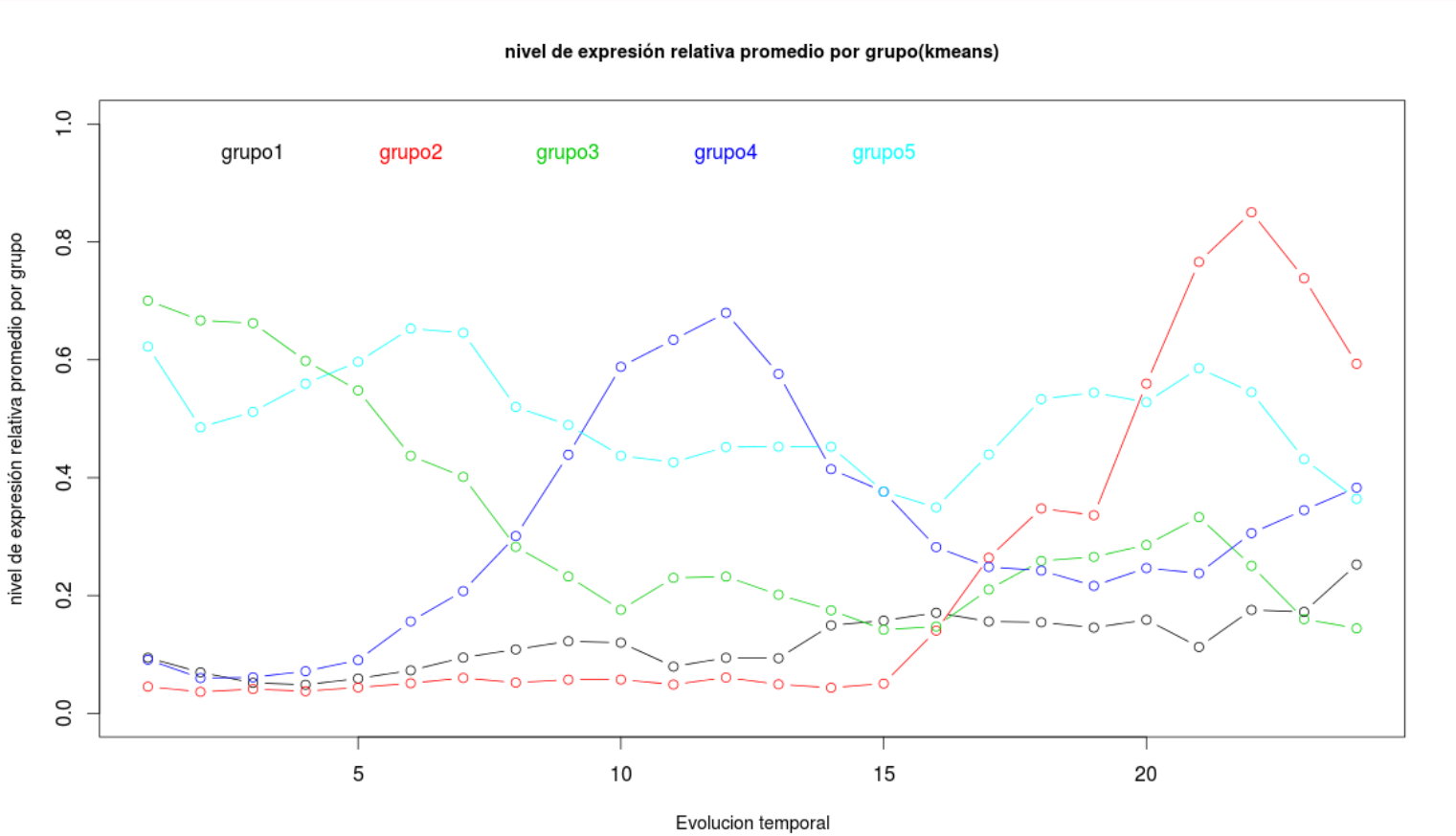


Figura: Nivel de Expresión de los centroides en **K-means**

Variación de Var Intra-grupos e Inter-grupos K-Means

$traza(W_k)$	$\frac{traza(W_k)}{traza(B_k)}$	$\frac{det(W_k)}{det(T_k)}$	corrida	Nro de clusters
20771.2	0.3389	0.1870	1	2
14296.3	0.9453	0.0132	2	5
11490.8	1.4203	0.0009	3	10
10155.4	1.7385	0.0001	4	15
9204.0	2.0216	2.8637 e-05	5	20
7861.5	2.5376	1.6415 e-06	6	30
7212.3	2.8560	3.0336 e-07	7	40
6675.1	3.1663	6.8733 e-08	8	50
5898.8	3.7147	6.3202 e-09	9	75
5415.3	4.1356	1.2348 e-09	10	100

Método clara con 5 grupos

```
> t(expresion.clara.5$medoids)
           [,1]      [,2]      [,3] [,4]         [,5]
E2  0.004706330 0.5762622 1.0000000    0 0.0000000000
E4  0.007375511 1.0000000 0.2402822    0 0.0000000000
E6  0.019229513 0.7442305 0.4548643    0 0.0002285595
E8  0.042960548 0.7654104 0.3984432    0 0.0000000000
E10 0.106036657 0.6337475 0.3553435    0 0.0000000000
E12 0.587024631 0.6002185 0.2889077    0 0.0009665157
E14 0.521030477 0.4367079 0.3649157    0 0.0043963804
E16 0.502898161 0.4162153 0.2395671    0 0.0082322747
E18 0.530844548 0.3258754 0.3042037    0 0.0327255605
E20 0.550832965 0.2758905 0.2469958    0 0.0398652184
E22 0.403868167 0.3811598 0.2554960    0 0.0335287032
E24 0.453457043 0.2811900 0.3137398    0 0.0399245372
L1  0.419139262 0.2167176 0.2841627    0 0.0127725850
L2  0.152228916 0.1981361 0.3051957    0 0.0047534937
L3  0.076363686 0.1357463 0.1913885    0 0.0037532743
L4  0.085443982 0.1514864 0.2508042    0 0.0916014008
L5  0.120850796 0.2120001 0.3650808    0 0.2294640196
L6  0.198965236 0.2242569 0.5017734    0 0.3564509720
P1  0.180286578 0.2504778 0.4537056    0 0.2816811273
P2  0.496540905 0.2700856 0.2980829    0 0.5032908898
P3  0.714320243 0.3289870 0.3436349    0 0.8200317912
P4  1.000000000 0.2053624 0.4065804    0 1.0000000000
P5  0.405396371 0.1534872 0.3651877    0 0.9004785244
P6  0.136037523 0.1720257 0.2157419    0 0.6244411697
```

Método clara con 5 grupos

```
Objective function:  1.123279
Clustering vector:   int [1:13677] 1 2 2 2 3 4 1 1 3 1 1 1 1 2 2 1 4 4 ...
Cluster sizes:       1889 3088 3399 2945 2356
Best sample:
 [1]   831  1246  1281  1941  1947  2012  2110  2568  2825  3006  4120
[12]  4209  4375  4485  4553  5054  5195  5533  5618  6011  6385  6605
[23]  6683  6996  7128  7513  8008  8207  8326  8470  8862  9025  9205
[34]  9438  9856 10082 10700 10754 10865 11240 12178 12192 12507 12793
[45] 12963 13050 13055 13172 13402 13462
```

Método Clara en 2 etapas

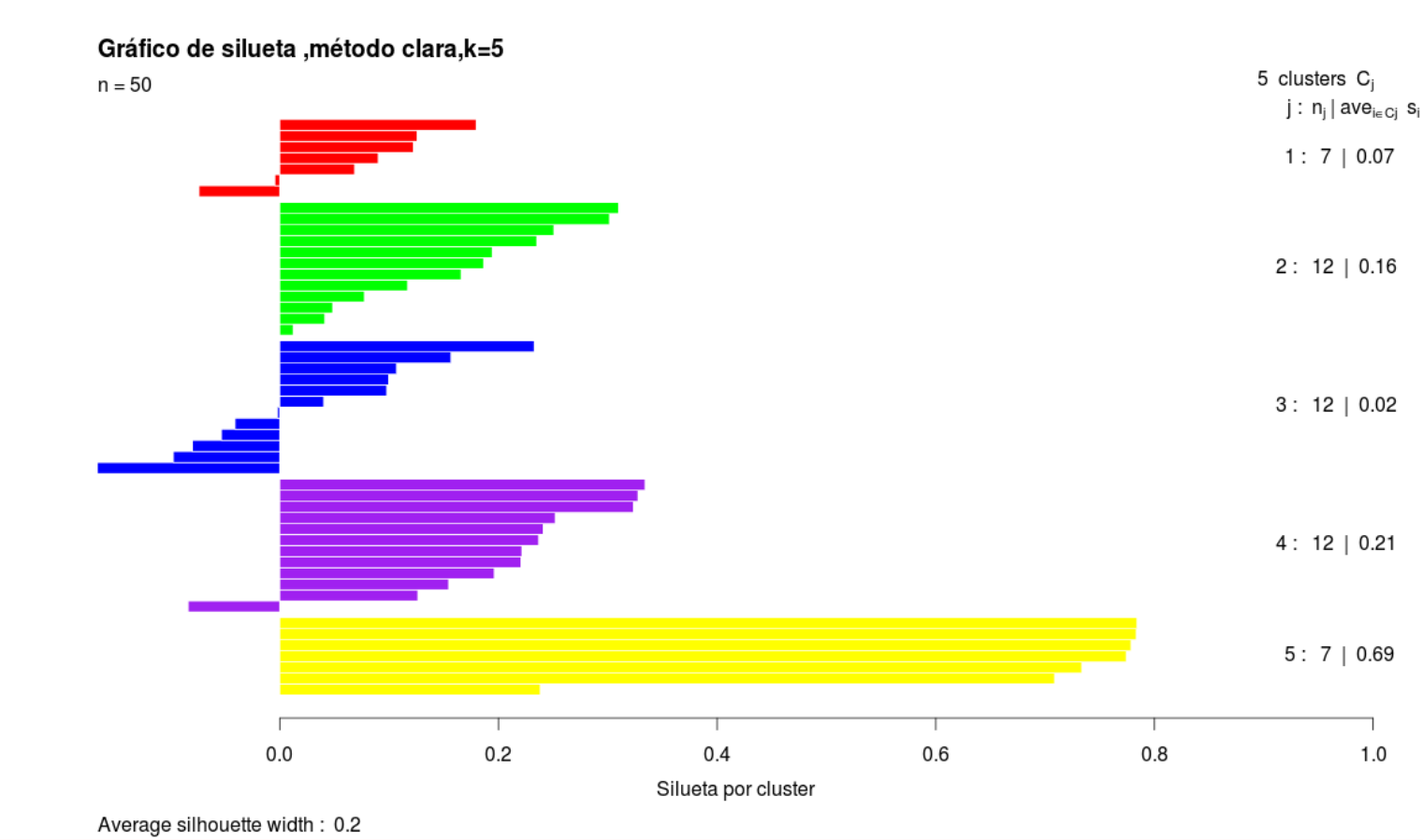


Figura: Gráfico de silueta

ESTABILIDAD DE LOS GRUPOS ENCONTRADOS

Teniendo en cuenta que los metodos son con arranmque aleatorio se hacen 10 corridas de cada uno

```
nSim <-10 # nro de simulaciones
nCl <- 5 # = nro de grupos
set.seed(123)# (para poder hacer reproducibles las corridas)
cl.kmeans <- matrix(NA,nrow(maxi.emb), nSim)
for(i in 1:nSim)
  cl.kmeans[,i] <- kmeans(maxi.emb, nCl)$cluster
cl.kmeans<-as.data.frame(cl.kmeans)
colnames(cl.kmeans)<-c("corrida1","corrida2","corrida3","corrida4","corrida5","corrida6","corrid
head(cl.kmeans,50)
tcl.kmeans <- t(apply(cl.kmeans,1, tabulate, nbins = nCl))
tcl.kmeans[1:50,]

> table(cl.kmeans$corrida1,cl.kmeans$corrida2)

      1    2    3    4    5
  1   1    0    0 2985    0
  2   0 2398    0    0    1
  3 3049    0   11    0    0
  4   0    0    0    0 2384
  5   0    0 2844    0    4
```

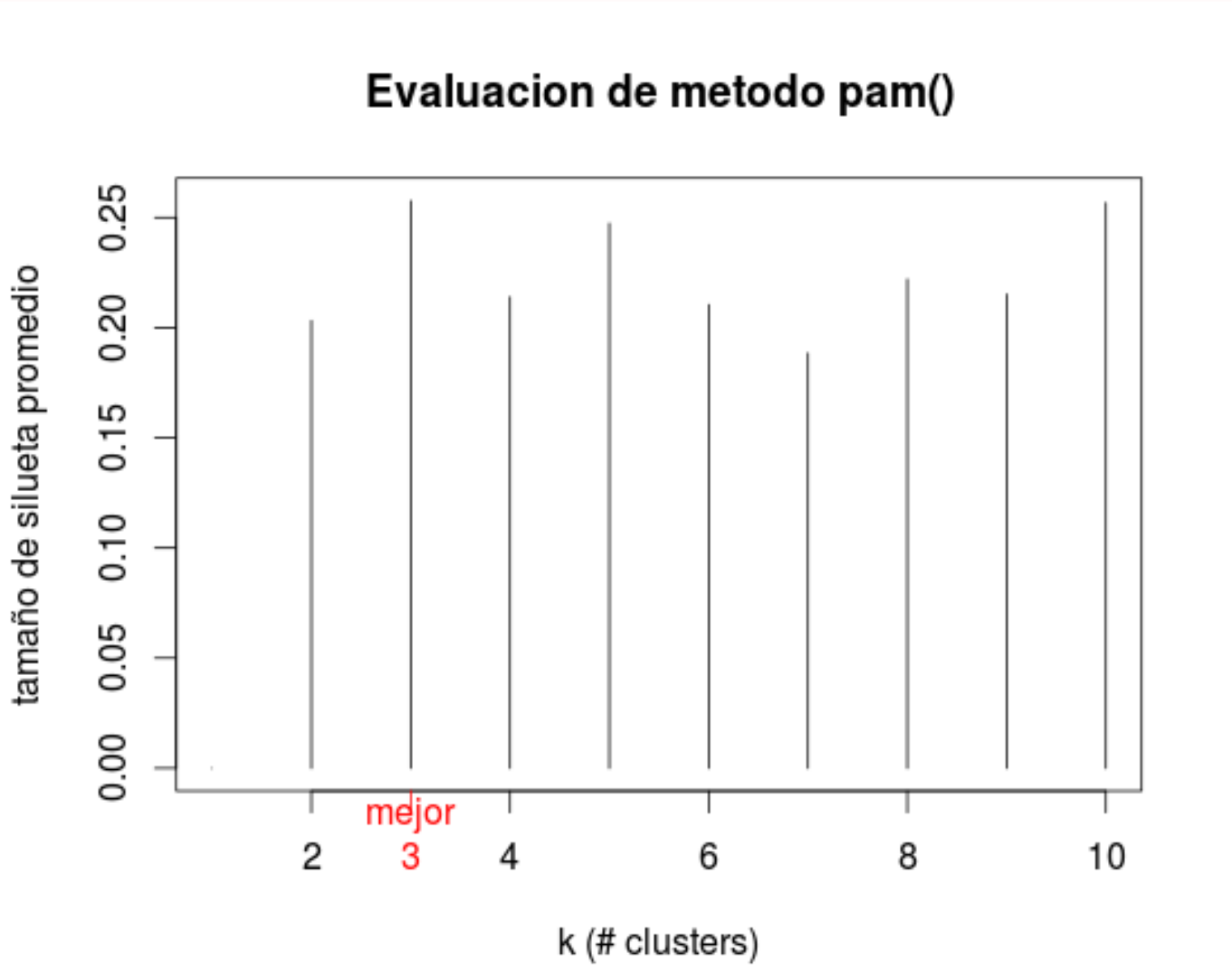
Solución óptima para Clara

```
k# = nro.de clases

asw <- numeric(10)
set.seed(123)# (reproducibilidad)
for (k in 2:10)
  asw[k] <- clara(maxi.emb, k, medoids.x = TRUE,rngR=FALSE) $ silinfo $ avg.width
k.best <- which.max(asw)

> k.best
[1] 3
> asw
 [1] 0.0000000 0.2031964 0.2575950 0.2139665 0.2473610 0.2104562
 [7] 0.1884837 0.2219976 0.2151947 0.2567135
```

Número óptimo de clusters para Clara



Descripción para método Clara

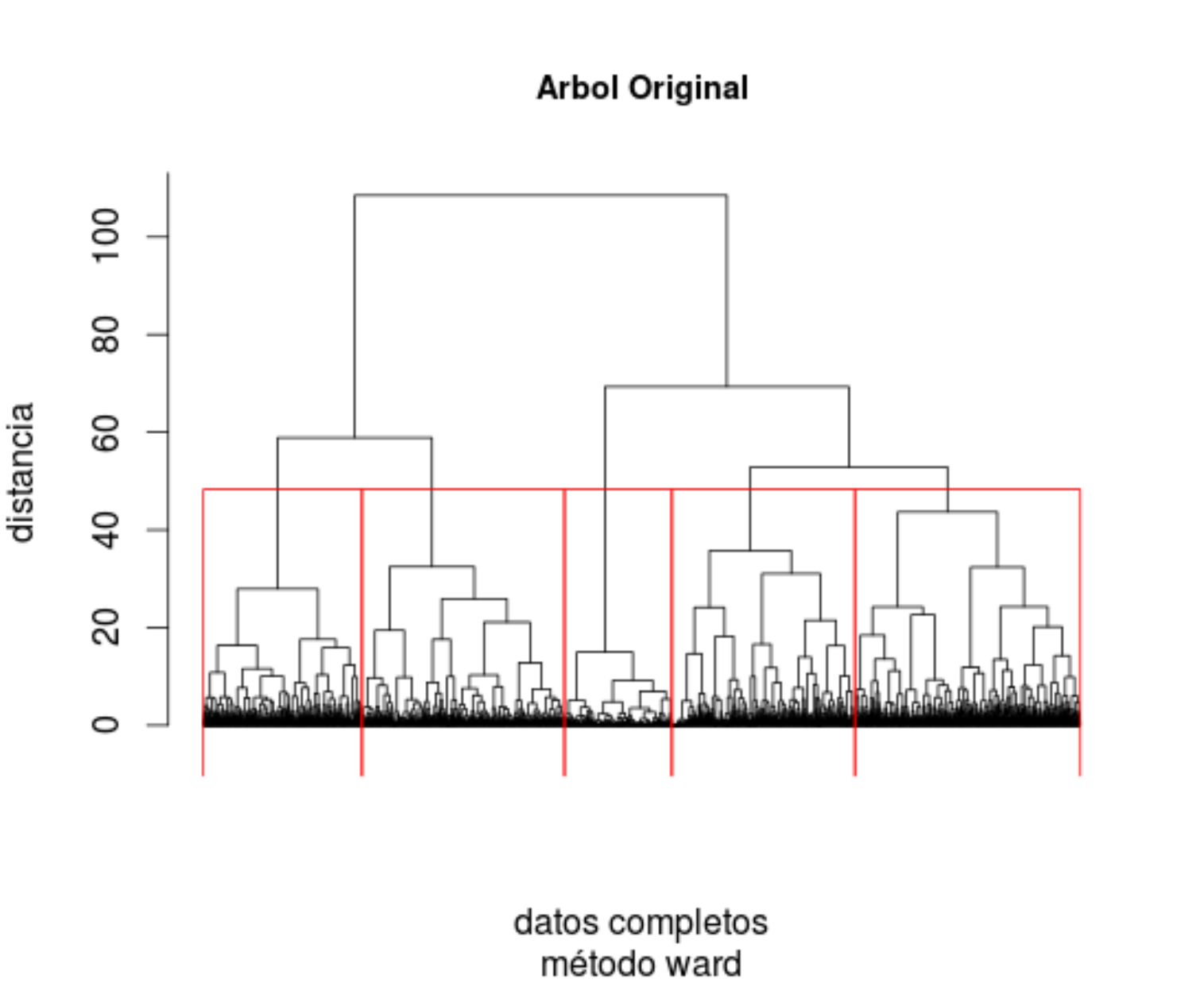
```
> tabla3
$mean
 [1] 0.3308607 0.2824126 0.2840332 0.2801176 0.2833196 0.2859754 0.2917405 0.2569282 0.2672754
[10] 0.2692187 0.2767020 0.2958745 0.2689039 0.2459926 0.2191415 0.2169538 0.2623146 0.3059721
[19] 0.3013491 0.3497658 0.3969341 0.4103881 0.3529065 0.3341495

$cluster.center
         [,1]       [,2]       [,3]       [,4]       [,5]       [,6]       [,7]       [,8]
c1 0.09451729 0.06976785 0.05200680 0.04874375 0.05924937 0.07304989 0.09475699 0.10864135
c2 0.04549325 0.03663106 0.04144900 0.03778847 0.04415054 0.05128013 0.06028291 0.05242655
c3 0.70050922 0.66691225 0.66226979 0.59825118 0.54813687 0.43724248 0.40139505 0.28277424
c4 0.09126290 0.06006982 0.06126588 0.07166238 0.09046413 0.15618972 0.20763792 0.30099194
c5 0.62243161 0.48539143 0.51172425 0.55950624 0.59661560 0.65302645 0.64581976 0.52000635
         [,9]      [,10]      [,11]      [,12]      [,13]      [,14]      [,15]      [,16]      [,17]
c1 0.12262361 0.1200270 0.07951006 0.09446943 0.09380911 0.14961218 0.1575994 0.1708834 0.1561452
c2 0.05739138 0.0575056 0.04914118 0.06095481 0.04950302 0.04372531 0.0507591 0.1406017 0.2641365
c3 0.23238819 0.1758907 0.23004873 0.23218602 0.20141664 0.17512999 0.1420976 0.1475242 0.2103130
c4 0.43902292 0.5881496 0.63396929 0.67976249 0.57615186 0.41449886 0.3763075 0.2822370 0.2485936
c5 0.48944920 0.4372801 0.42619911 0.45200719 0.45261426 0.45250663 0.3767204 0.3495223 0.4394517
       [,18]      [,19]      [,20]      [,21]      [,22]      [,23]      [,24]
c1 0.1546498 0.1456671 0.1589257 0.1127522 0.1756758 0.1723039 0.2526744
c2 0.3477348 0.3364145 0.5596119 0.7660800 0.8503019 0.7383666 0.5933449
c3 0.2588040 0.2656134 0.2858562 0.3330679 0.2505225 0.1600613 0.1444121
c4 0.2425018 0.2164721 0.2466652 0.2378661 0.3059421 0.3449860 0.3831130
c5 0.5332572 0.5444819 0.5280605 0.5857101 0.5451092 0.4314000 0.3641153
```

Método jerárquico ward con 5 grupos

```
for(k in 1:5){ cent <- rbind(cent, colMeans(maxi.emb[memb == k, , drop = FALSE]))
  grupo<-rbind(grupo,k)}
> cent
              E2         E4         E6         E8        E10
[1,] 0.104186393 0.08625390 0.10405491 0.13880906 0.17567603
[2,] 0.669524845 0.58537701 0.58972912 0.60391991 0.62981467
[3,] 0.733994896 0.62047027 0.59982363 0.53542734 0.47576472
[4,] 0.060279755 0.04440080 0.04655982 0.04626153 0.06002041
[5,] 0.007071319 0.01455049 0.01924650 0.01529095 0.01484259
             E12        E14        E16        E18        E20
[1,] 0.25520575 0.31688639 0.35016141 0.43330219 0.50280958
[2,] 0.65551065 0.64080137 0.50013940 0.47510409 0.41945719
[3,] 0.37422840 0.33433727 0.24888495 0.20894284 0.16560328
[4,] 0.06511271 0.07401467 0.08402987 0.09679386 0.11595819
[5,] 0.01441965 0.01382759 0.01162494 0.01218524 0.01346264
             E22        E24          L1          L2         L3
[1,] 0.47371448 0.51949067 0.424267184 0.328623848 0.2735078
[2,] 0.43998036 0.46254708 0.460648760 0.466914264 0.3851489
[3,] 0.21744510 0.21484331 0.201169583 0.172433591 0.1424597
[4,] 0.11387972 0.13136414 0.138372285 0.172893743 0.2102607
[5,] 0.01109779 0.01334663 0.009585933 0.009294139 0.0193506
             L4         L5         L6         P1         P2
[1,] 0.2181592 0.2337815 0.2593071 0.2500349 0.3082395
[2,] 0.3349469 0.4099919 0.4960294 0.5074718 0.4864539
[3,] 0.1438051 0.1960112 0.2338685 0.2343956 0.2585374
[4,] 0.2539018 0.2516447 0.2649756 0.2610922 0.2730075
[5,] 0.1149511 0.2471658 0.3289017 0.2993109 0.5378820
             P3         P4         P5         P6
[1,] 0.3363655 0.3855998 0.3328261 0.3047153
[2,] 0.5431816 0.5078973 0.4122054 0.3506405
[3,] 0.2995425 0.2308052 0.1560547 0.1470940
[4,] 0.2206758 0.2592585 0.2584848 0.3560343
[5,] 0.7922241 0.9148043 0.8397939 0.6871284
```

Dendrograma para método Ward



Comparación de los 3 métodos de clusters

```
> addmargins(table(datos$cl_kmeans_5,datos$memb))
          1     2     3     4     5   Sum
  1     616     0   380  1985     5  2986
  2     295     2    20   413  1669  2399
  3     157   360  2504    39     0  3060
  4    1895   130    46   313     0  2384
  5     552  1983   208   105     0  2848
  Sum  3515  2475  3158  2855  1674 13677
> addmargins(table(datos$cl_kmeans_5,datos$cl_clara_5))
          1     2     3     4     5   Sum
  1     154     4   238  2441   149  2986
  2     137     0    98     1  2163  2399
  3      19  2065   932    43     1  3060
  4    1233    19   630   460    42  2384
  5     346  1000  1501     0     1  2848
  Sum  1889  3088  3399  2945  2356 13677
> addmargins(table(datos$cl_clara_5,datos$memb))
          1     2     3     4     5   Sum
  1    1685    86     1   117     0  1889
  2     174  1146  1766     2     0  3088
  3     642  1243  1055   459     0  3399
  4     831     0   329  1785     0  2945
  5     183     0     7   492  1674  2356
  Sum  3515  2475  3158  2855  1674 13677
```

Características generales de los clusters

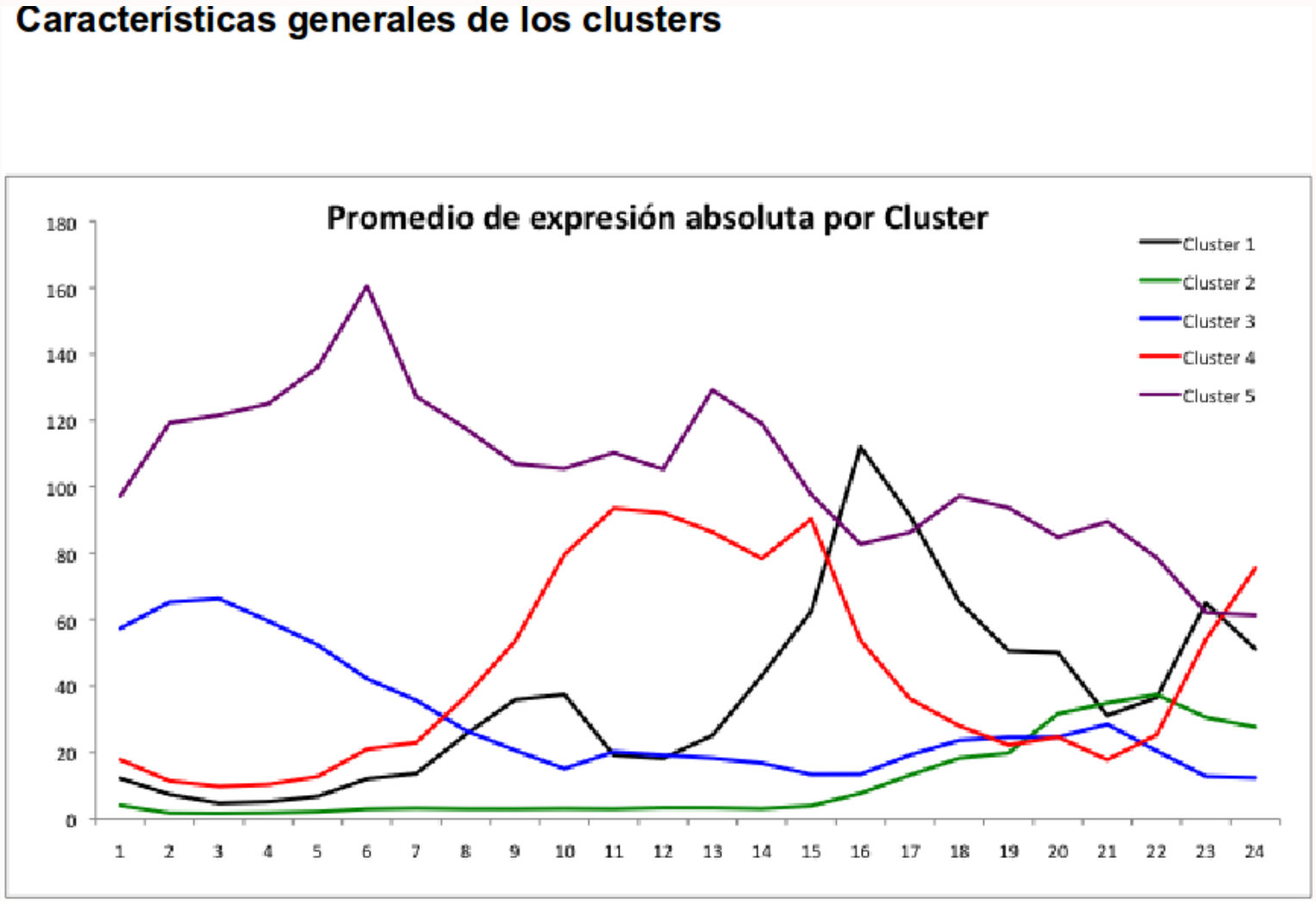


FIGURA: Caracterización de los clusters de acuerdo a su dinámica

Un gen tomado como referencia

- Se considera el gen de BP (BruchPilot) el cual se considera la referencia
- Para cada gen se recontruye la secuencia que sigue en referencia a BP: si en cada t al cambio que este gen i tiene concide con **BP** la secuencia es 1 y 0 de los contrario
- Cálculo del score Brp: un punto por cada paso en el que haya coincidencia con el comportamiento de Brp

SCORE DE BP PARA MÉTODO K-MEANS

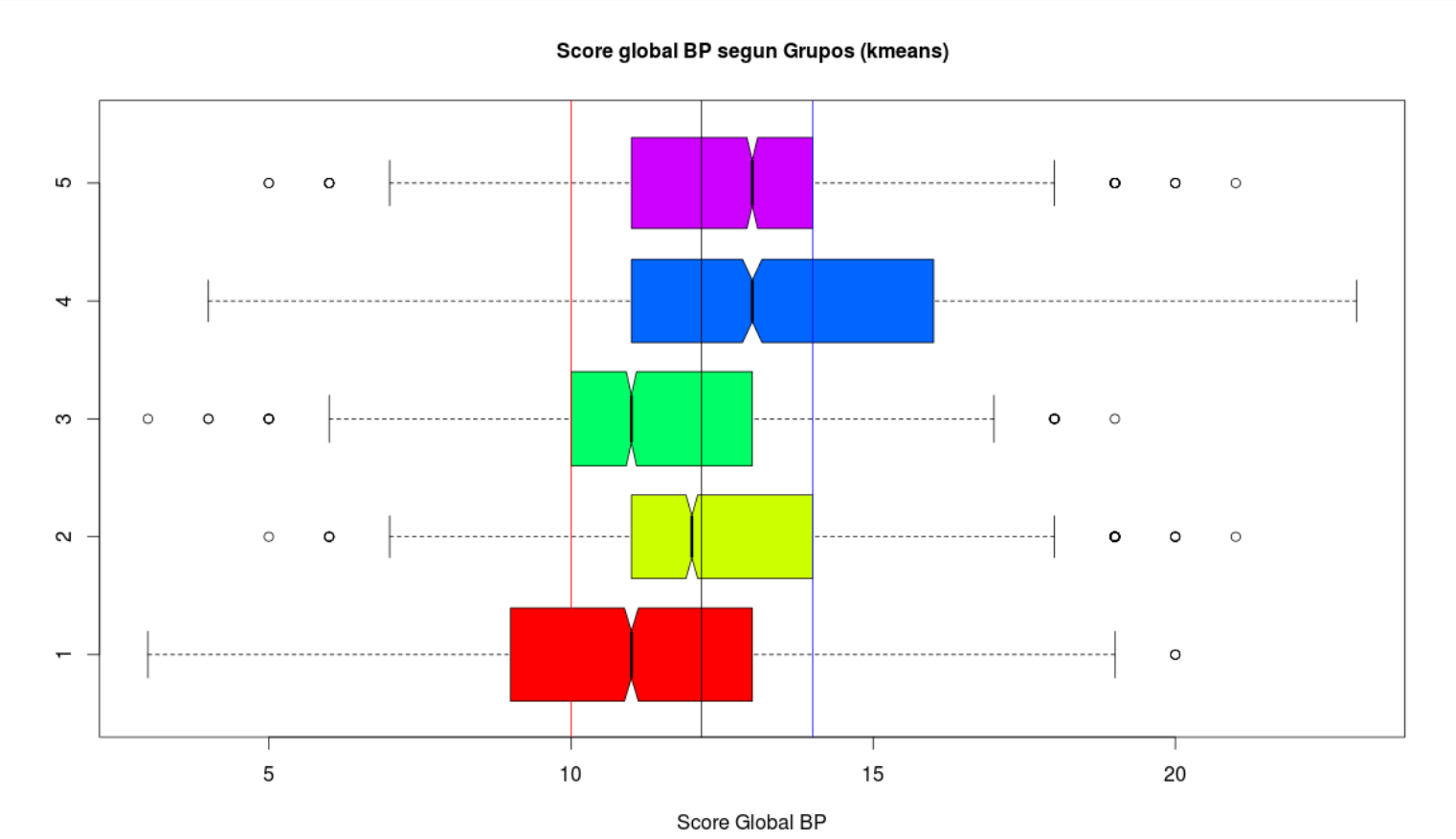


FIGURA: Score de BP para método **k-means**

Score de BP para método Clara

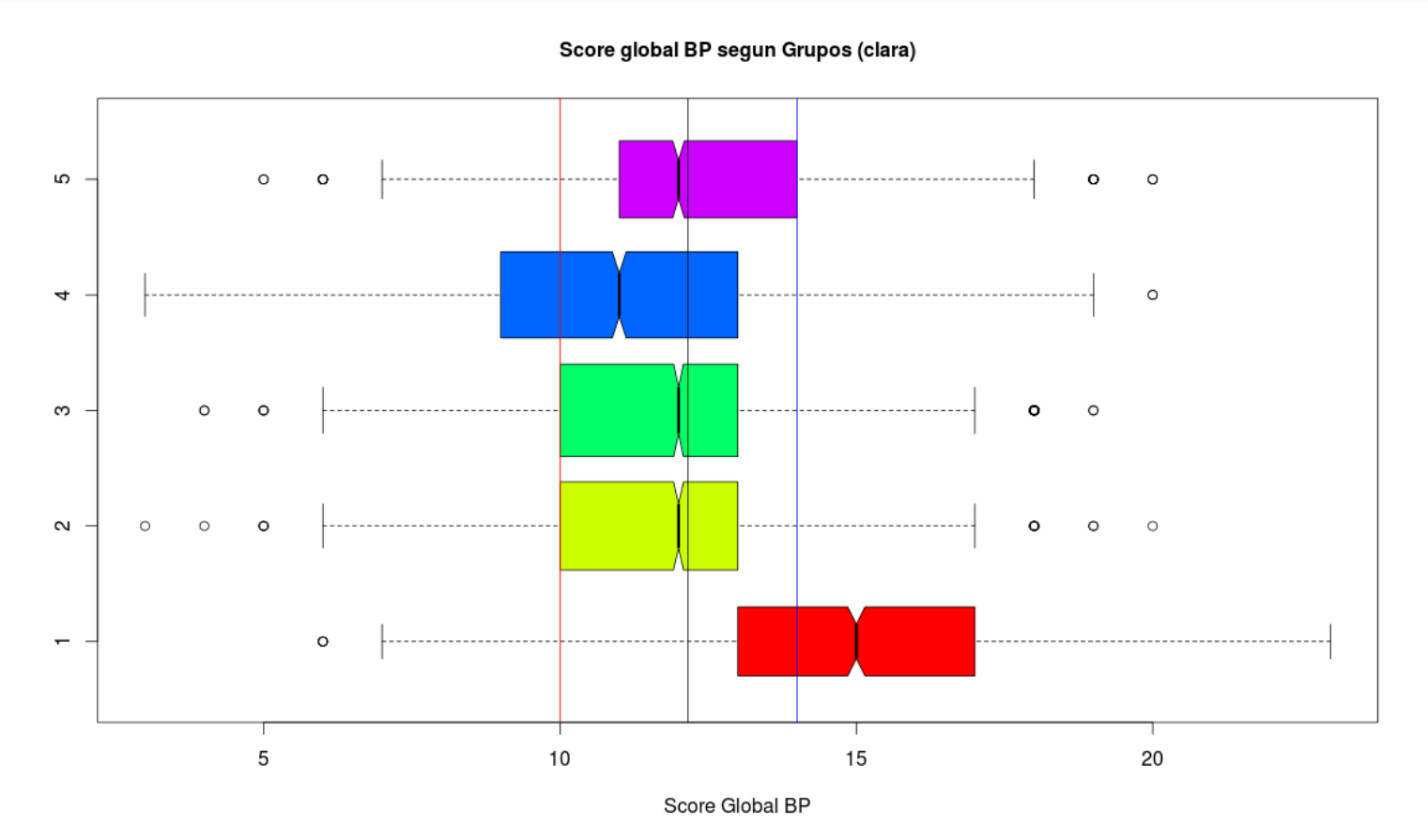


Figura: Score de BP para método **Clara**

Score de BP para método Ward

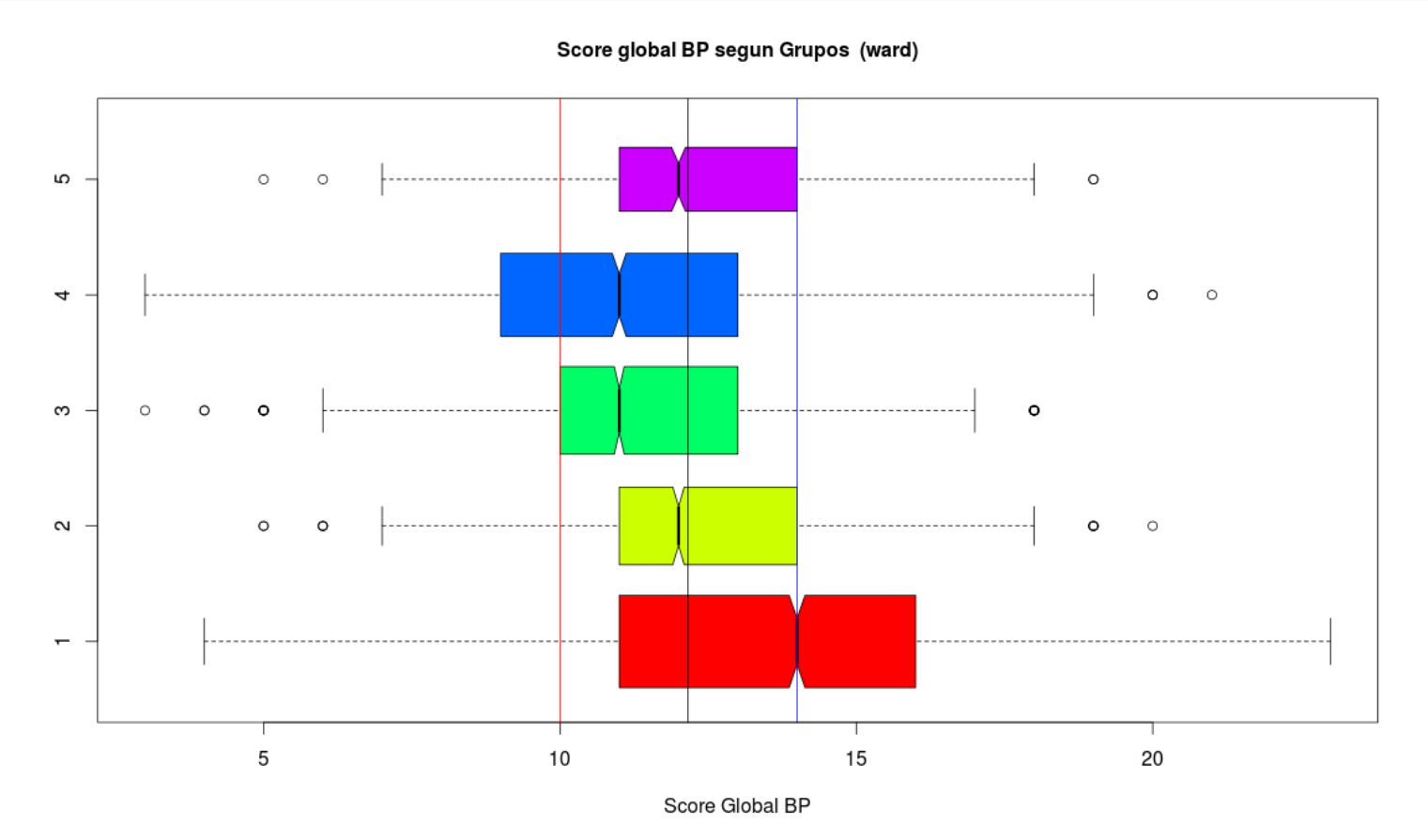


Figura: Score de BP para método **Ward**

Distribución de los grupos de genes seguun sobre expresión por tejidos

Distribución de los genes con sobre expresión por tejidos

	SNC	FAT	MIDGUT	SG	CAR
Cluster 1	89	237	233	154	260
Cluster 2	63	109	25	36	45
Cluster 3	179	36	22	50	23
Cluster 4	312	118	299	77	180
Cluster 5	151	93	43	155	31
Total	794	593	622	472	539

Cantidad de genes de cada cluster que se sobe expresan en cada tejido

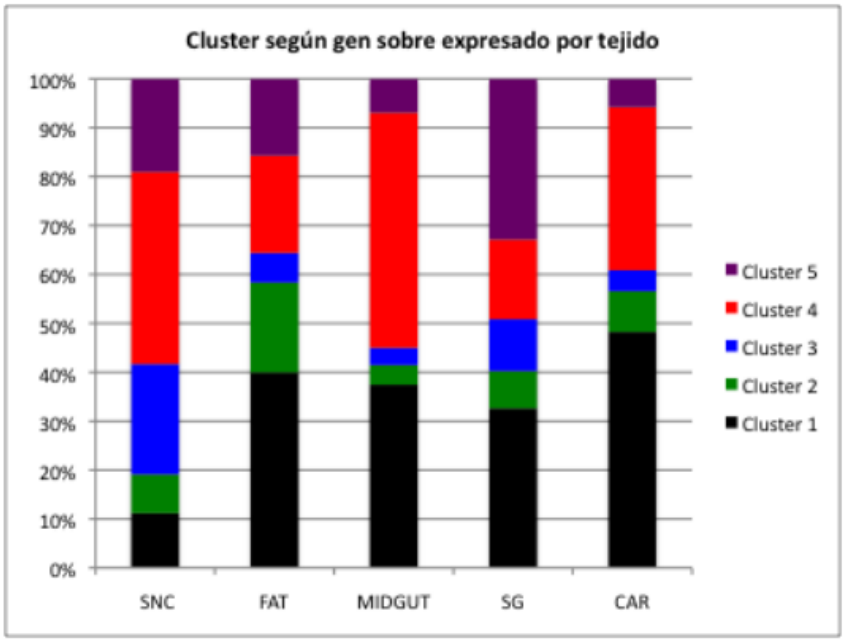


FIGURA: Dendrograma para método **Ward**

Varias alternativas metodológicas

- Trabajar con datos acumulados en los niveles de expresión
- Probar con datos funcionales considerando lso 24 nodos de cada trayectoria por gen en los niveles de expresión

R Development Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria; 2012.
ISBN 3-900051-07-0.
Available from: http://www.R-project.org/.

Blanco J.
Introducción al Análisis Multivariado:Teoría y aplicaciones a la realidad latinoamericana.
IESTA UdlR, editor. IESTA, Universidad de la República; 2006.

Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K. cluster: Cluster Analysis Basics and Extensions; 2012.
R package version 1.14.2 — For new features, see the 'Changelog' file (in the package source).

Kaufman L, Rousseeuw PJ.
Finding Groups in Data: An Introduction to Cluster Analysis.
New York: John Wiley; 1990.