

# Detección de Comunidades II

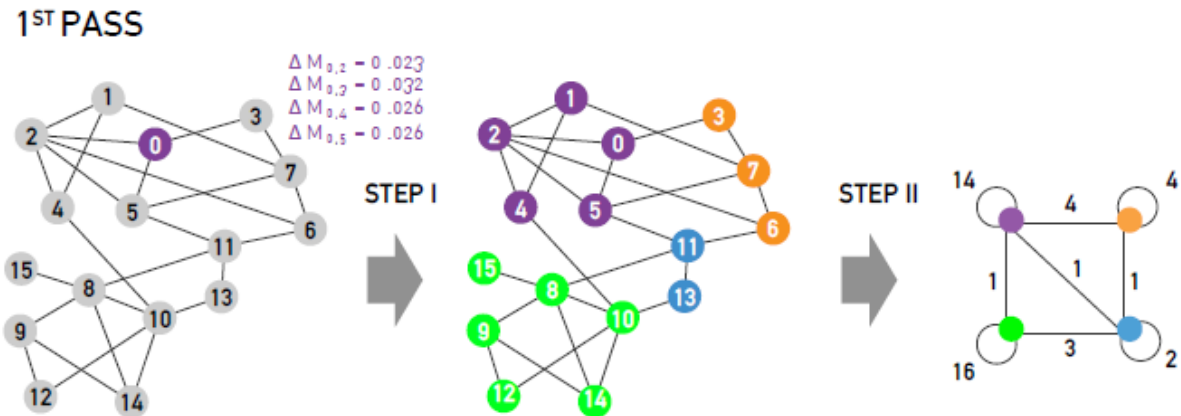
- Louvain, Infomap
- Cfinder, Link CLustering
- Testing / Verifying communities
- MsigDB - GO – HyoperG
- GO-dist

# Algoritmo Louvain

Algoritmo de reconocimiento de comunidades para redes pesadas, de complejidad computacional  $o(L)$ .



Modularidad  $M$  optimizada en pasadas de dos pasos



## Paso 1

Se optimiza  $M$  con cambios locales, tratando de unir un nodo con sus vecinos.

Se elige el cambio de mayor  $\Delta M$  (si el cambio es positivo) Se repite esto para cada nodo de la red

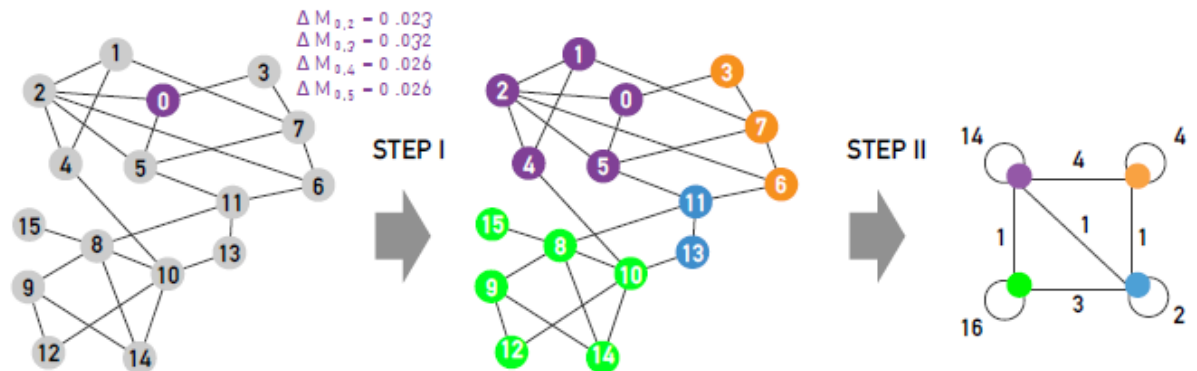
# Algoritmo Louvain

Algoritmo de reconocimiento de comunidades para redes pesadas, de complejidad computacional  $o(L)$ .



Modularidad  $M$  optimizada en pasadas de dos pasos

1<sup>ST</sup> PASS



Paso 1

Se optimiza  $M$  con cambios locales, tratando de unir un nodo con sus vecinos. Se elige el cambio de mayor  $\Delta M$  (si el cambio es positivo) Se repite esto para cada nodo de la red

Paso 2

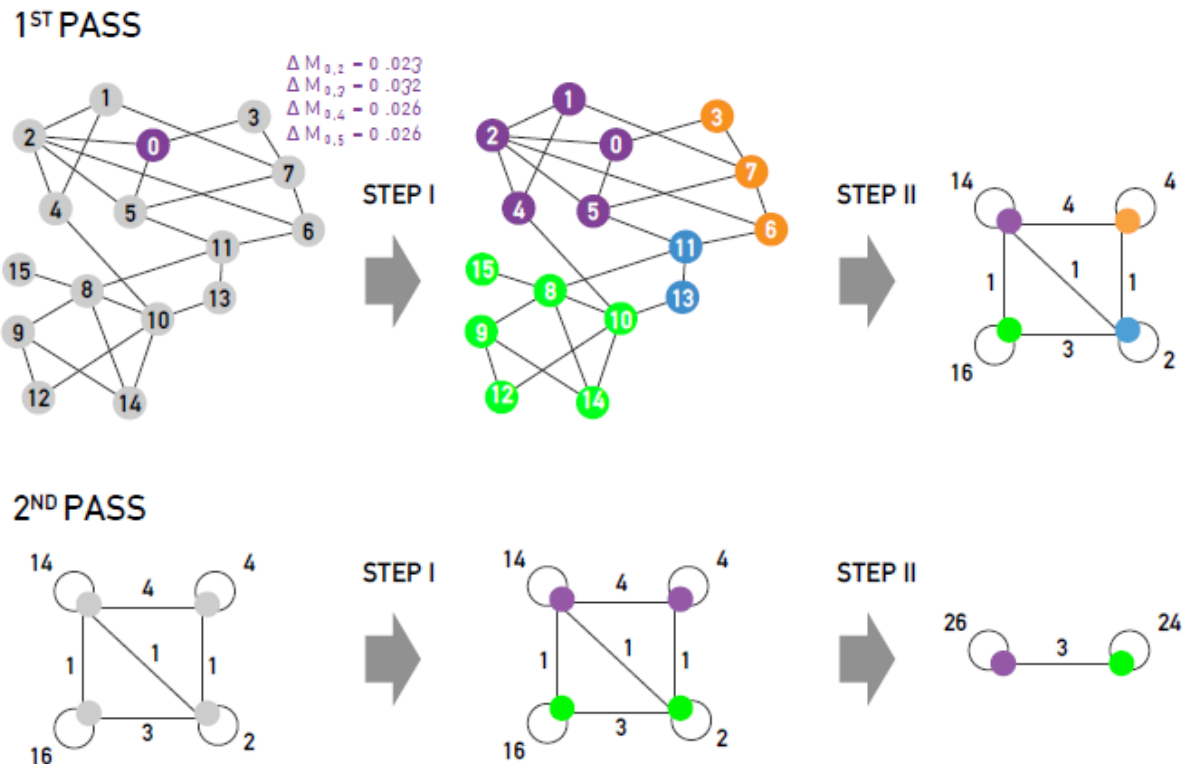
Se arma una nueva red donde cada nodo es una comunidad encontrada en el Paso 1. Se generan auto-enlaces que corresponden a lazos intra-comunidad.

# Algoritmo Louvain

Algoritmo de reconocimiento de comunidades para redes pesadas, de complejidad computacional  $o(L)$ .



Modularidad  $M$  optimizada en pasadas de dos pasos



Se repite hasta no poder obtener incremento de modularidad

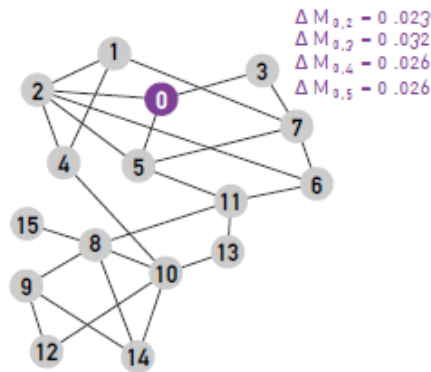
# Algoritmo Louvain

Algoritmo de reconocimiento de comunidades para redes pesadas, de complejidad computacional  $O(L)$ .



Modularidad  $M$  optimizada en pasadas de dos pasos

1<sup>ST</sup> PASS



La variación de modularidad al mover un nodo aislado  $i$  dentro de la comunidad C se puede computar **eficientemente** de manera local

$$M = \sum_r \left( \frac{L_r}{L} - \left( \frac{k_r}{2L} \right)^2 \right)$$

suma de pesos de nodo- $i$  a nodos del cluster

Cambio de  $M$ :

nodo- $i$  anexo a comunidad C

$$\Delta M = \left[ \frac{\Sigma_{in} + 2k_{i,in}}{2W} - \left( \frac{\Sigma_{tot} + k_i}{2W} \right)^2 \right] - \left[ \frac{\Sigma_{in}}{2W} - \left( \frac{\Sigma_{tot}}{2W} \right)^2 - \left( \frac{k_i}{2W} \right)^2 \right]$$

suma de pesos enlaces C

suma de pesos enlaces intra-C

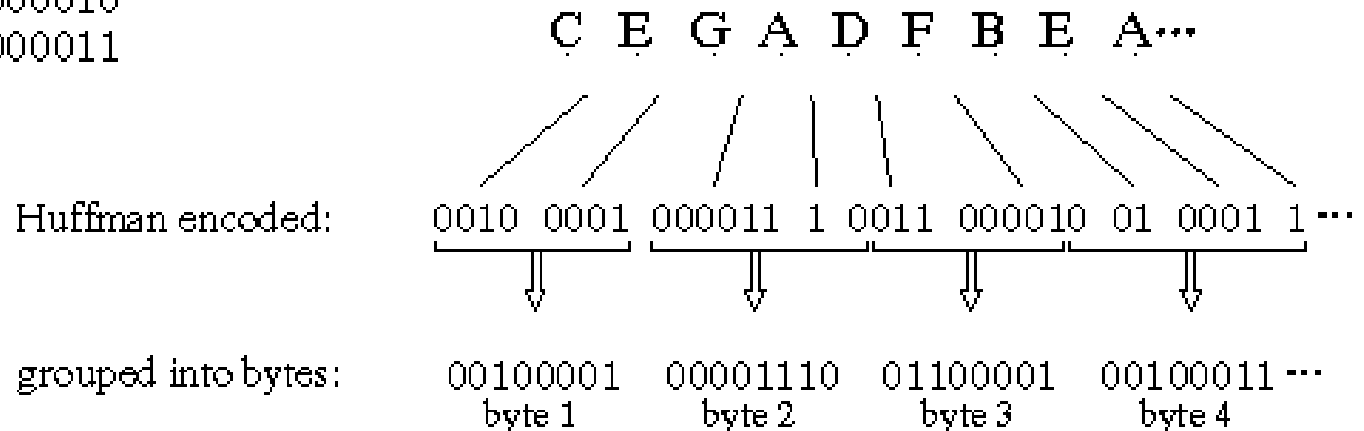
# Infomap

Existe una **dualidad** entre

- problema de **compresión** de datos
- detectar **estructura y patrones** en los mismos.

Example Encoding Table

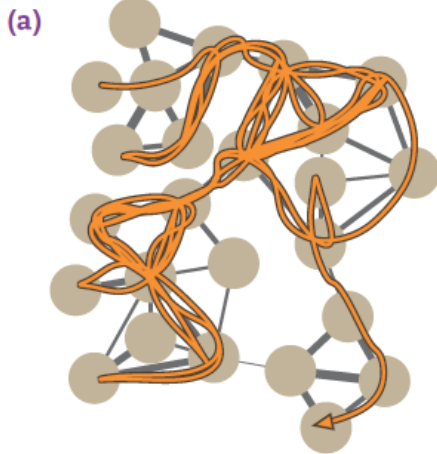
letter	probability	Huffman code
A	.154	1
B	.110	01
C	.072	0010
D	.063	0011
E	.059	0001
F	.015	000010
G	.011	000011



# Infomap

Dualidad compresion/patrones

Caminatas aleatorias como indicador de estructura de la red

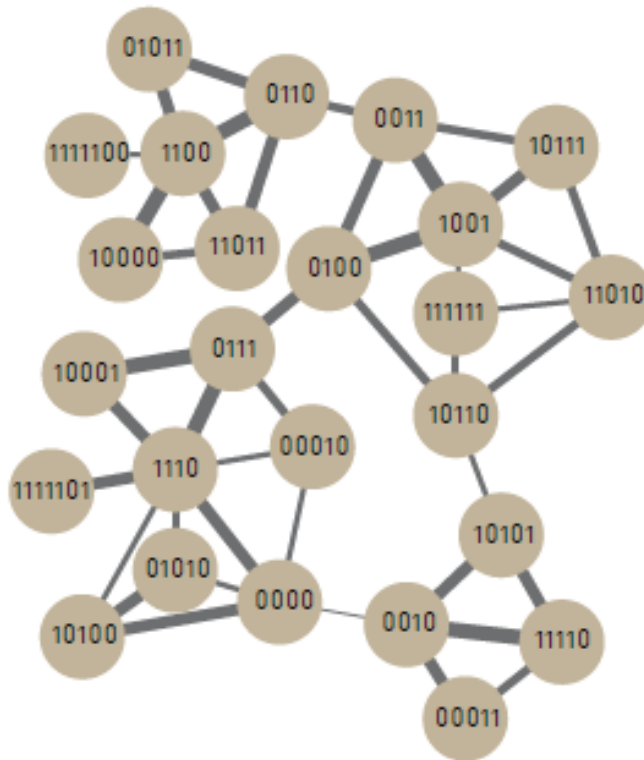
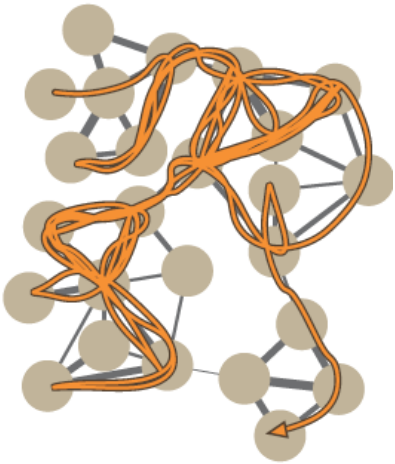


## Ingredientes

1. Map Equation: Dada una **particion** de la red en comunidades, la *map equation* es la manera de cuantificar que tan eficiente es la descripcion de una caminata aleatoria sobre la red
2. Minimizacion de la Map Equation frente al conunto de todas las posibles **particiones** de a red en comunidades



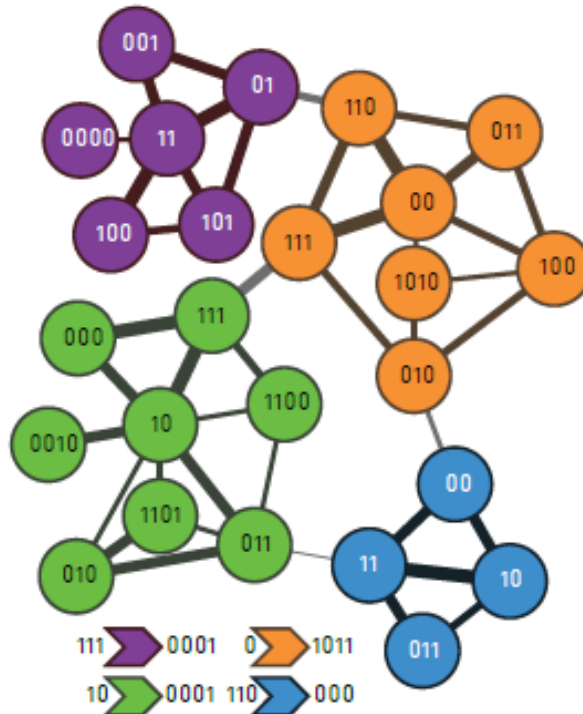
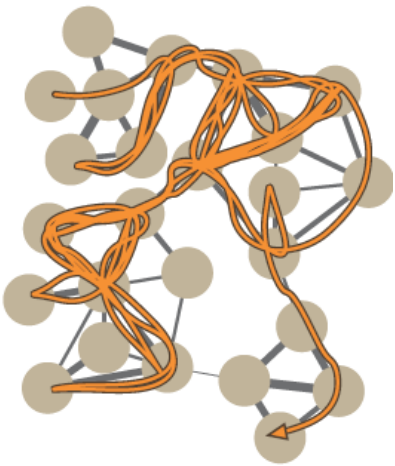
# Infomap



```
1111100 1100 0110 11011 10000 11011 0110 0011 10111 1001 0011
1001 0100 0111 10001 1110 0111 10001 0111 1110 0000 1110 10001
0111 1110 0111 1110 1111101 1110 0000 10100 0000 1110 10001 0111
0100 10110 11010 10111 1001 0100 1001 10111 1001 0100 1001 0100
0011 0100 0011 0110 11011 0110 0011 0100 1001 10111 0011 0100
0111 10001 1110 10001 0111 0100 10110 111111 10110 10101 11110
00011
```

- Para poder describir la trayectoria de 70 pasos, necesito etiquetar a cada uno de los 25 nodos.
- Naivamente...necesito:  $\log_2(25) \cdot 70 = 325$  bits
- Código de **compresión de Huffman** asigna nombres usando la probabilidad de visita de un caminante aleatorio a cada nodo- $i$  ( $\sim \sum_j A_{ij}$ ).
- En el ejemplo esto resulta en una descripción de 314 bits para la caminata de 70 pasos que comienza en el 1111100 y finaliza en el 00011

# Infomap

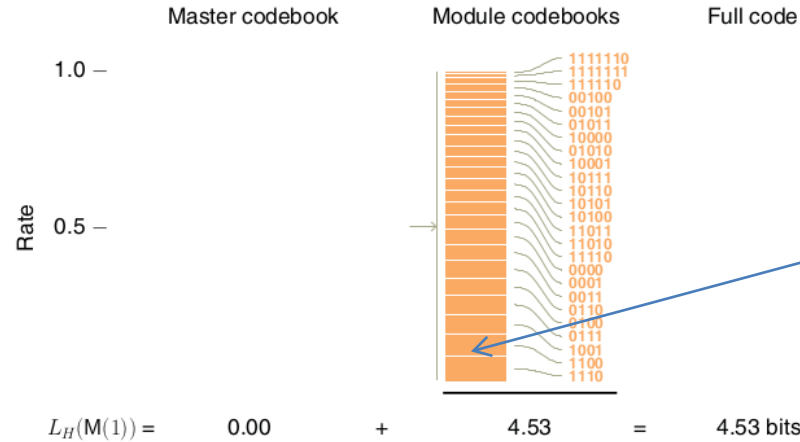
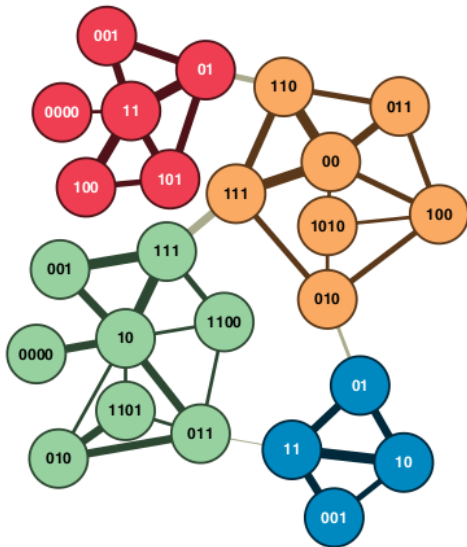
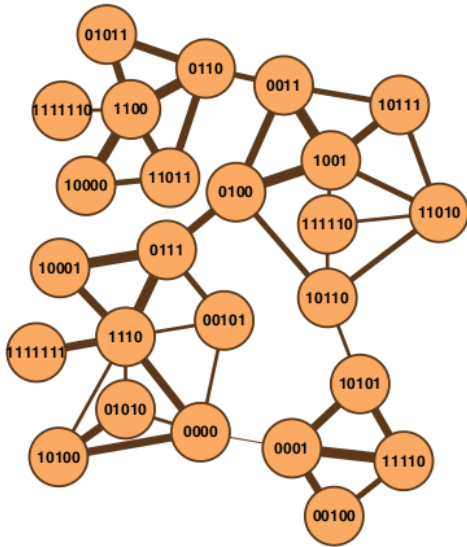


111 0000 11 01 101 100 101 01 0001 0 110 011 00 110 00 111 1011 10  
111 000 10 111 000 111 10 011 10 000 111 10 111 10 0010 10 011 010  
011 10 000 111 0001 0 111 010 100 011 00 111 00 011 00 111 00 111  
110 111 110 1011 111 01 101 01 0001 0 110 111 00 011 110 111 1011  
10 111 000 10 000 111 0001 0 111 010 1010 010 1011 110 00 10 011

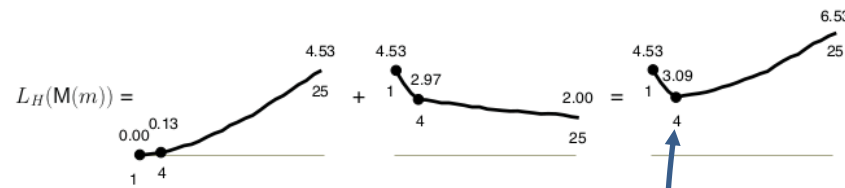
- Infomap propone un etiquetado de 2 niveles:
  - **Index code-book:** códigos asignado a entrada y salida de una comunidad
  - **Module code-book:** código interno de cada nodo dentro de su comunidad.
- Con esta estrategia se pueden reusar nombres (en particular nombres cortos) y se suelen alcanzar descripciones más cortas que usando Huffman
- La descripción de la misma caminata ahora usa 243bits (< 314bits)
- Uso Huffman para *index* y para *module code-books* por separado

Pensar: por qué encontrar una **descripción óptima** (en particular el **index code-book**) en este esquema de compresión es encontrar **buenos clusters** (!)

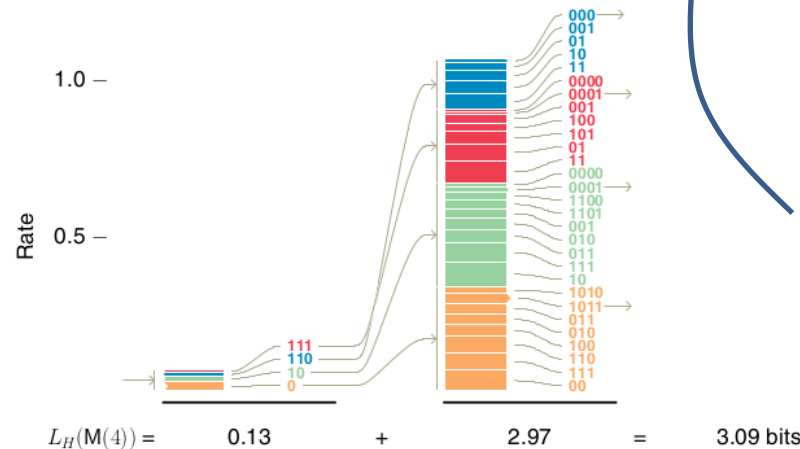
# Descripcion en 2 niveles...



Altura proporcional a la frecuencia de visita en un random walk



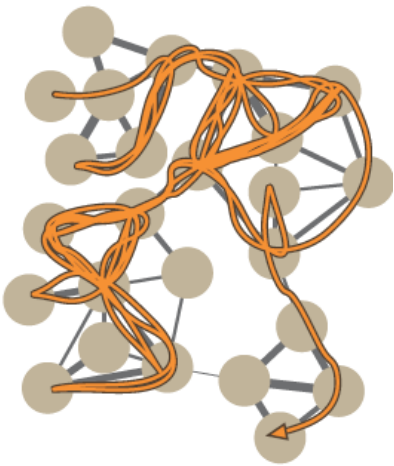
Description length para numero variables de clusters



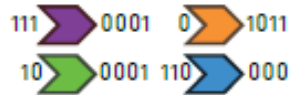
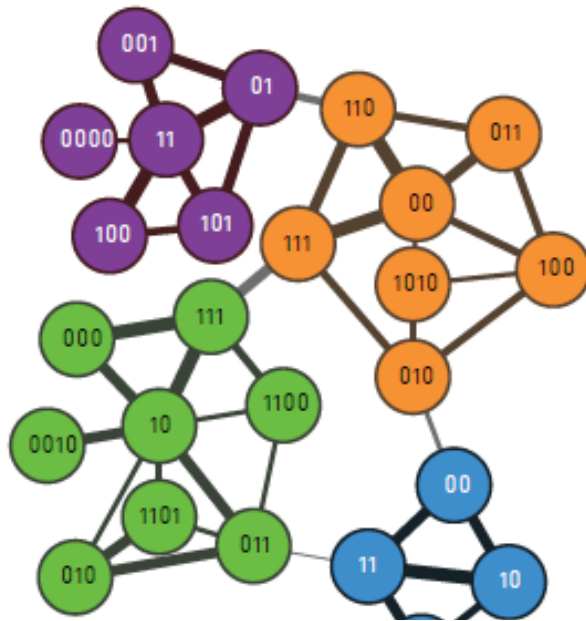
Partición optima

# Shannon y la *map equation*

- Para una partición  $M$  de  $n$  nodos en  $m$  módulos queremos calcular una cota mínima para el tamaño de la descripción de caminatas sobre la red  $L(M)$
- Shannon: La longitud media de etiquetas de un código para describir los  $n$  estados de una variable aleatoria  $X$ , que ocurren con frecuencias  $p_i$ , no puede ser inferior a la entropía de la fuente  $H(X) = -\sum_{i=1}^n p_i \log_2 p_i$



# Infomap



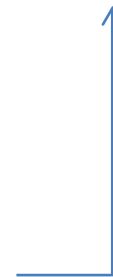
111 000 11 01 101 100 101 01 0001 0 110 011 01  
 111 000 10 111 000 111 10 011 10 000 111 10 11  
 011 10 000 111 0001 0 111 010 100 011 00 111 0  
 110 111 110 1011 111 01 101 01 0001 0 110 111  
 10 111 000 10 000 111 0001 0 111 010 1010 010 1

Bits necesarios en promedio para describir movimientos **entre** comunidades.

Lo estimo a partir de la entropía (i.e. de las freqs de aparición) de las variables del *index code-book* (codigos de salida)

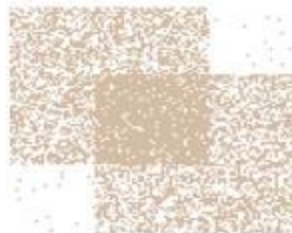
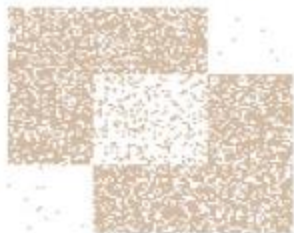
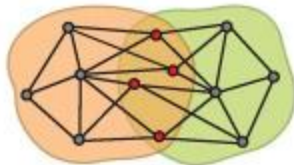
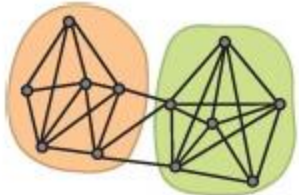
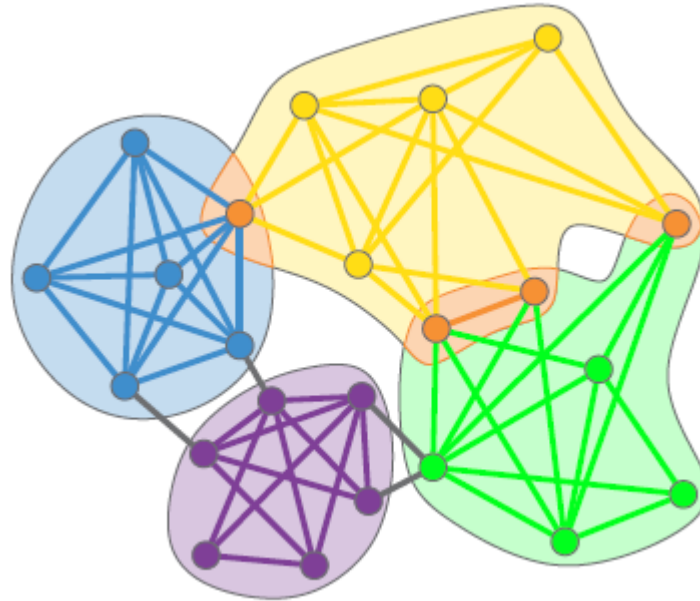
La **descripción óptima** en terminos de longitud de descripción (i.e. **buenos clusters**) se logra minimizando la Map equation:

$$\mathcal{L} = qH(Q) + \sum_{c=1}^{n_c} p_c^c H(P_c)$$



Bits necesarios en promedio para describir movimientos **dentro** de comunidades

# Comunidades solapadas

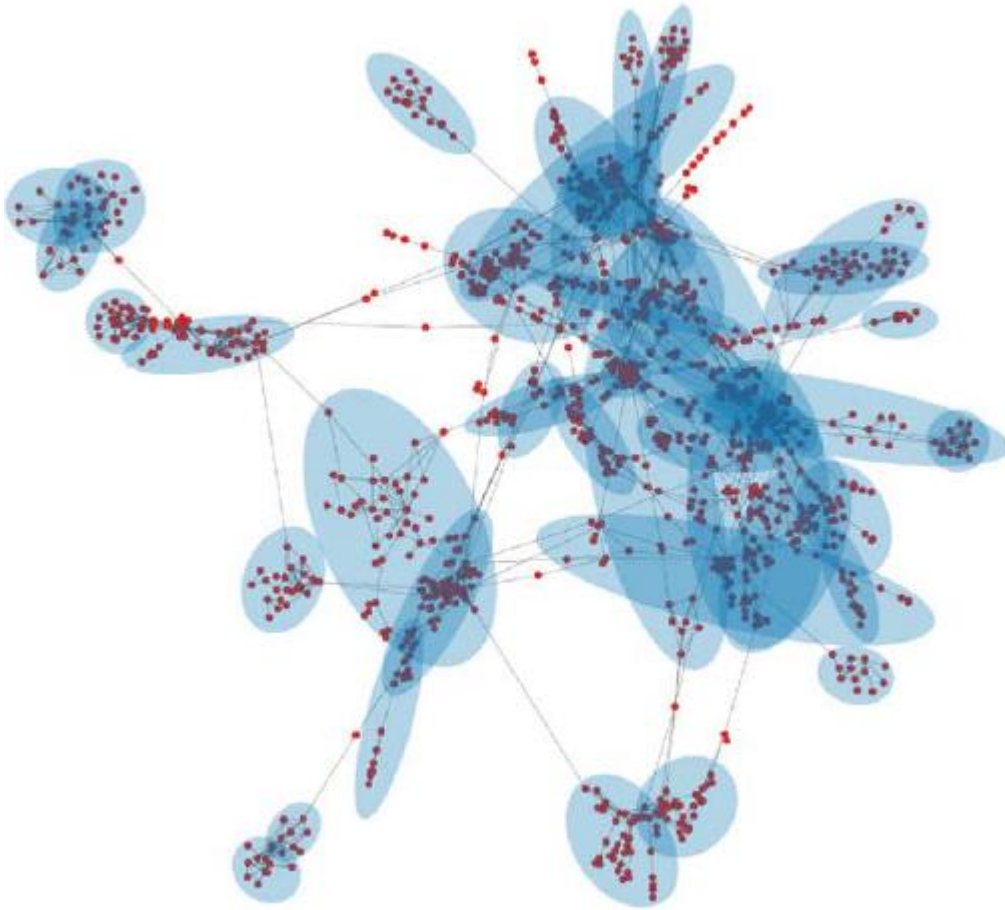


Diferentes posibilidades

- Sin overlap
- Overlap poco conectado
- Overlap muy conectado



# Comunidades solapadas

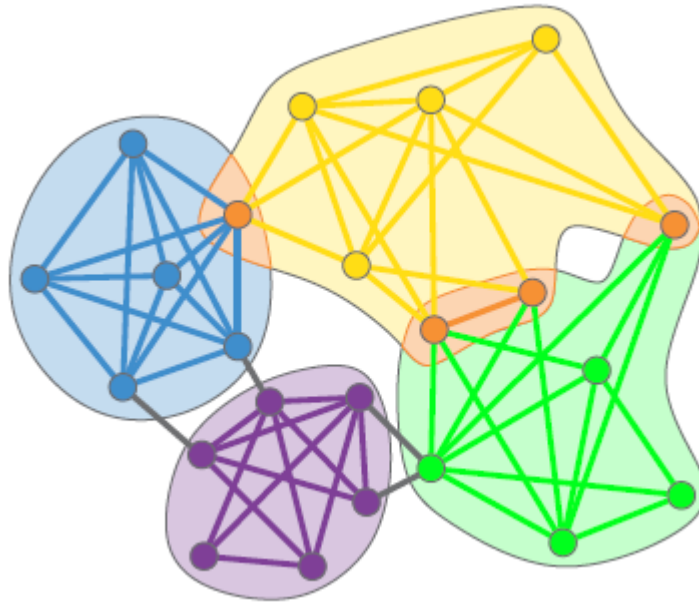


Comunidades AGM de una red de interacción de proteínas.

Comunidades solapadas pueden dar cuenta de múltiples funciones que pueda llevar adelante un determinado producto génico en diferentes contextos



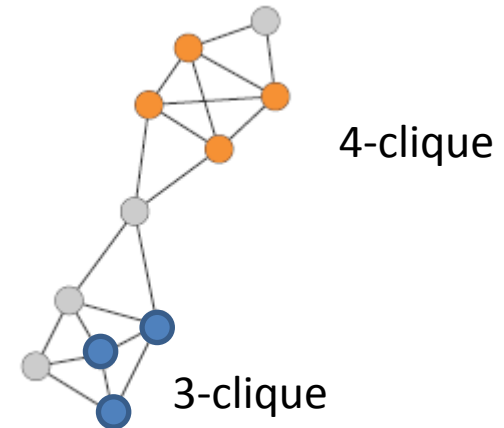
# Comunidades solapadas



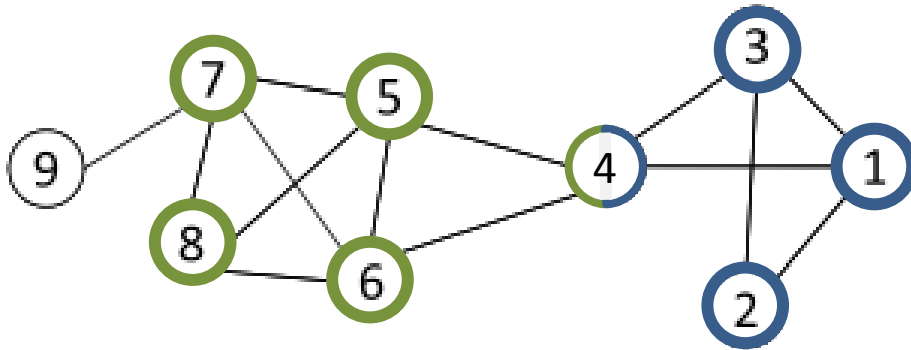
- Cfinder: Percolación de k-cliques
- Link Clustering
- Affiliation Graph Model

# Método de percolación de cliques

- Vimos que un clique implica una noción fuerte de comunidad
- Se los suele usar como **estructuras semilla** a partir de las cuales definir comunidades más grandes
- MPC es una metodología que permite buscar comunidades solapadas.
  1. Se elige un valor para  $k$
  2. Se encuentran todos los  $k$ -cliques de la red
  3. Se construye un **grafo de cliques**, donde dos  $k$ -cliques estarán vinculados si comparten  $k-1$  nodos.
  4. Cada componente conexa de este grafo forma una comunidad

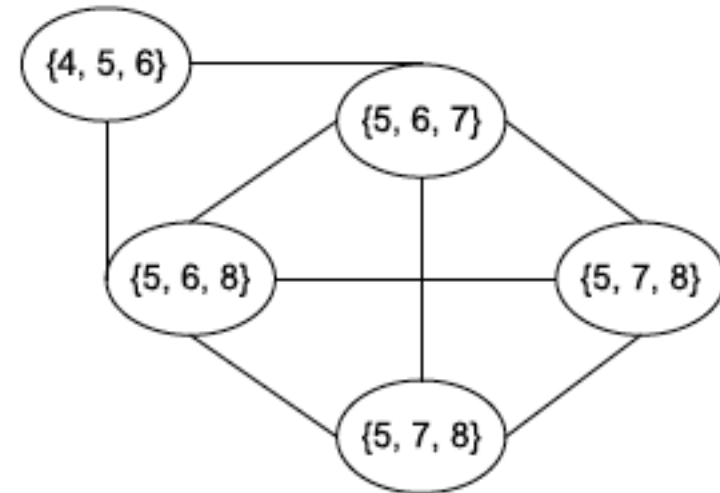


# Ejemplo de MPC



**3-Cliques:**

$\{1, 2, 3\}$ ,  $\{1, 3, 4\}$ ,  $\{4, 5, 6\}$ ,  
 $\{5, 6, 7\}$ ,  $\{5, 6, 8\}$ ,  $\{5, 7, 8\}$ ,  
 $\{6, 7, 8\}$

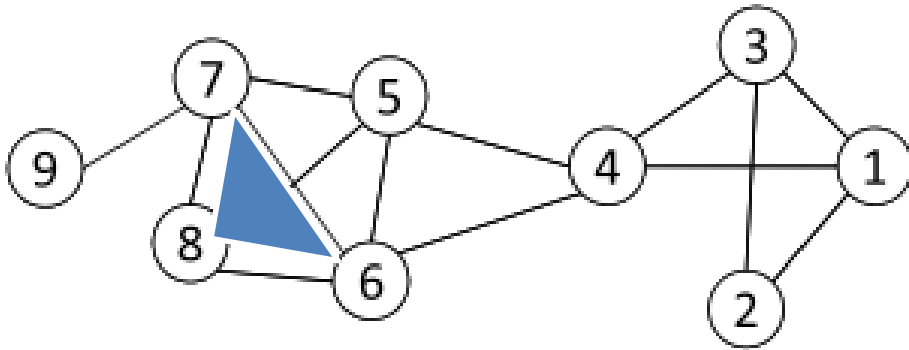


**Comunidades:**

$\{1, 2, 3, \underline{4}\}$   
 $\{\underline{4}, 5, 6, 7, 8\}$

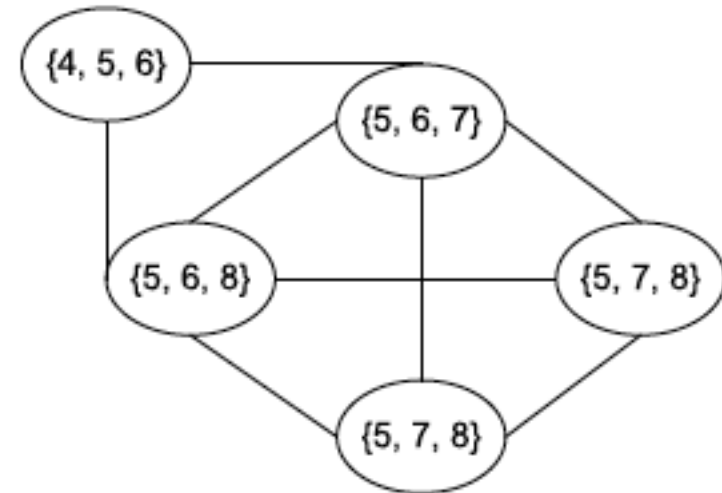


# Ejemplo de MPC: rolling k-cliques



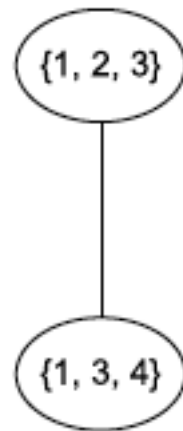
**3-Cliques:**

$\{1, 2, 3\}$ ,  $\{1, 3, 4\}$ ,  $\{4, 5, 6\}$ ,  
 $\{5, 6, 7\}$ ,  $\{5, 6, 8\}$ ,  $\{5, 7, 8\}$ ,  
 $\{6, 7, 8\}$

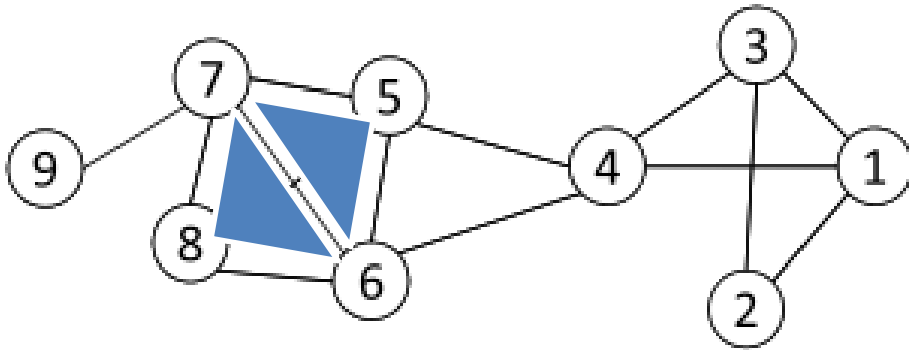


Comunidades:

$\{1, 2, 3, \underline{4}\}$   
 $\{\underline{4}, 5, 6, 7, 8\}$

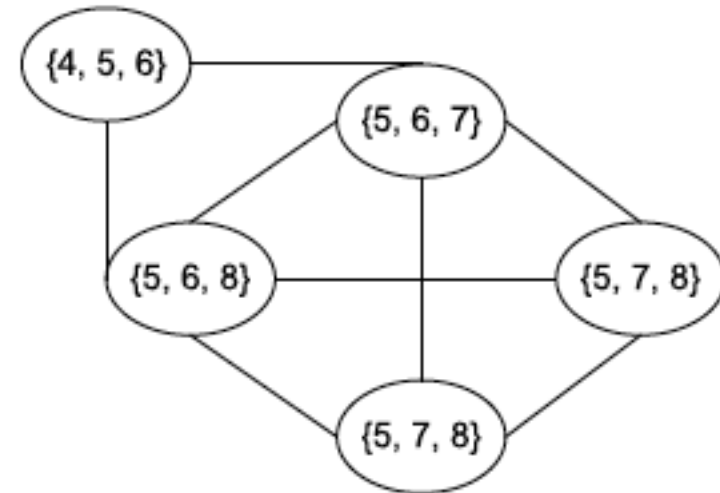


# Ejemplo de MPC: rolling k-cliques



**3-Cliques:**

$\{1, 2, 3\}$ ,  $\{1, 3, 4\}$ ,  $\{4, 5, 6\}$ ,  
 $\{5, 6, 7\}$ ,  $\{5, 6, 8\}$ ,  $\{5, 7, 8\}$ ,  
 $\{6, 7, 8\}$

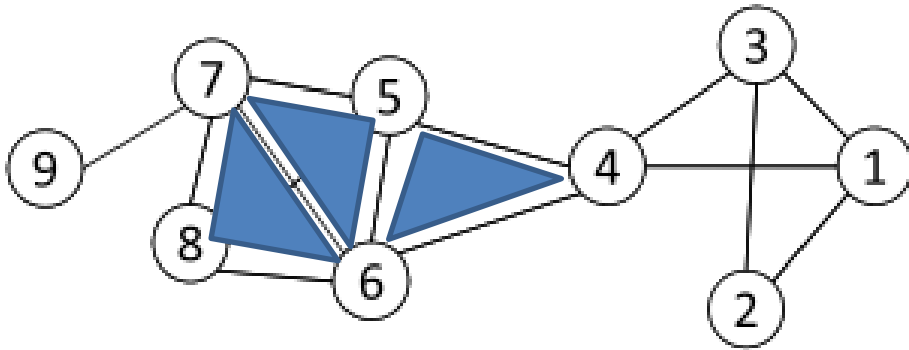


Comunidades:

$\{1, 2, 3, \underline{4}\}$   
 $\{\underline{4}, 5, 6, 7, 8\}$



# Ejemplo de MPC: rolling k-cliques

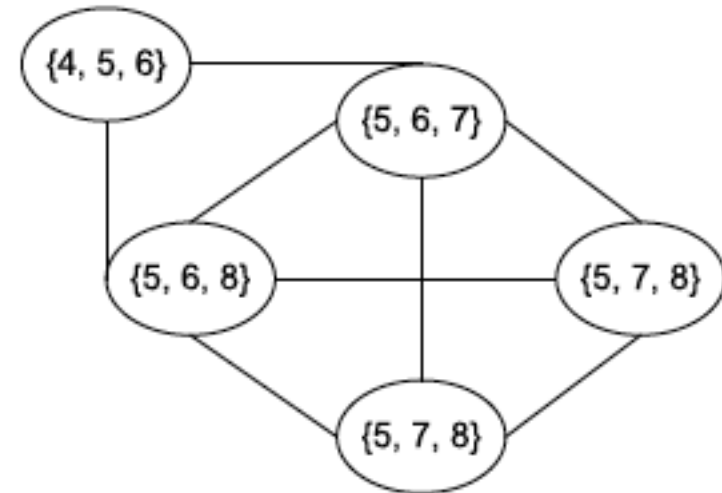


**3-Cliques:**

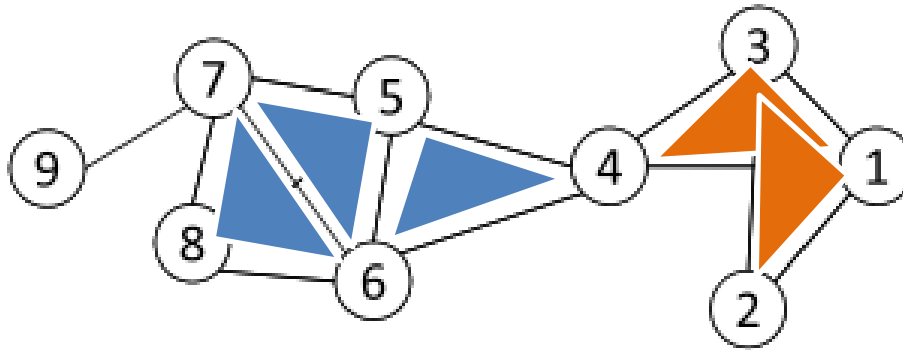
$\{1, 2, 3\}$ ,  $\{1, 3, 4\}$ ,  $\{4, 5, 6\}$ ,  
 $\{5, 6, 7\}$ ,  $\{5, 6, 8\}$ ,  $\{5, 7, 8\}$ ,  
 $\{6, 7, 8\}$

Comunidades:

$\{1, 2, 3, \underline{4}\}$   
 $\{\underline{4}, 5, 6, 7, 8\}$



# Ejemplo de MPC: rolling k-cliques

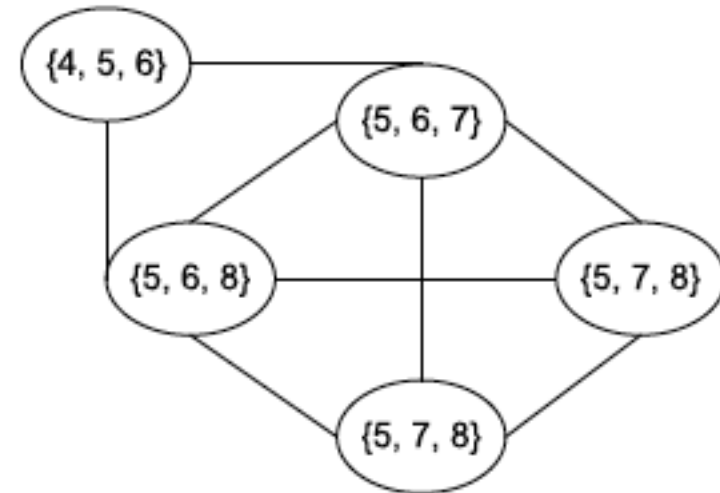


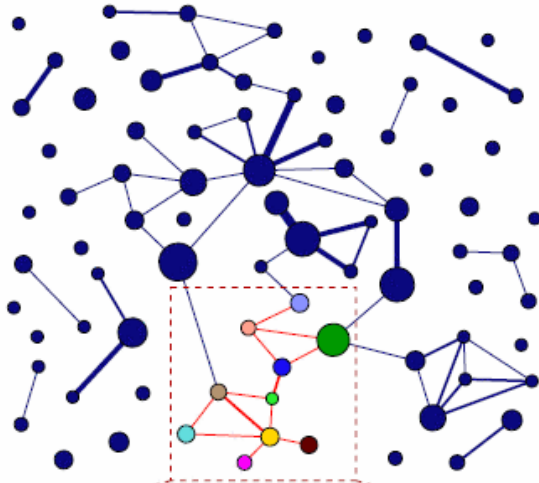
**3-Cliques:**

$\{1, 2, 3\}$ ,  $\{1, 3, 4\}$ ,  $\{4, 5, 6\}$ ,  
 $\{5, 6, 7\}$ ,  $\{5, 6, 8\}$ ,  $\{5, 7, 8\}$ ,  
 $\{6, 7, 8\}$

Comunidades:

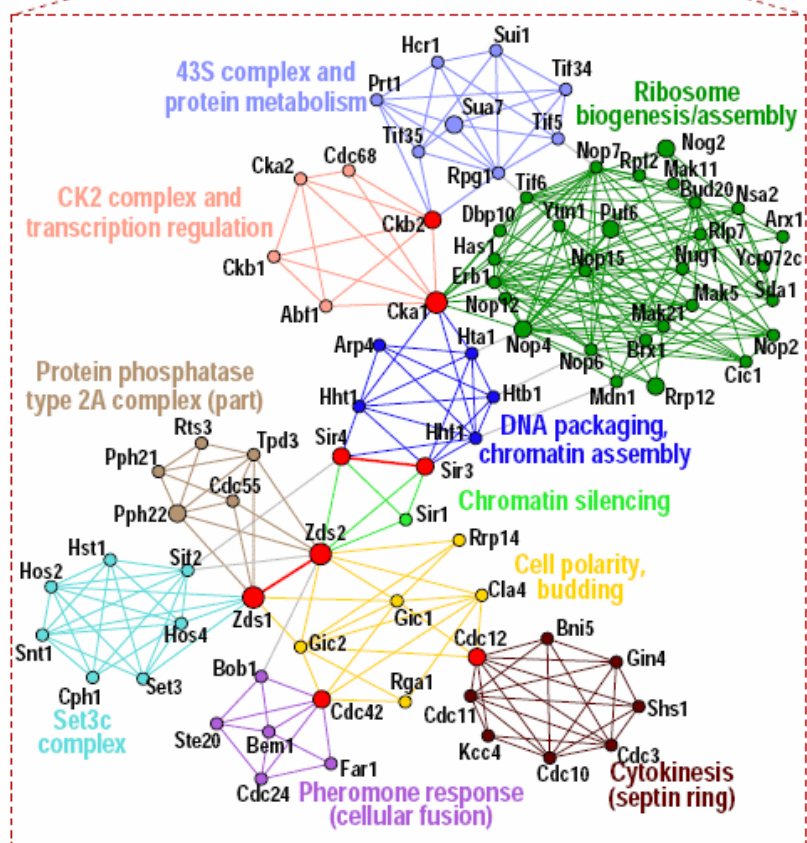
$\{1, 2, 3, \underline{4}\}$   
 $\{\underline{4}, 5, 6, 7, 8\}$





Red de 82 comunidades (MPC,  $k=4$ ) de la red de interacción de proteínas DIP para *S. cerevisiae*

Área de círculos y ancho de líneas proporcionales al tamaño de comunidad y overlap respectivamente

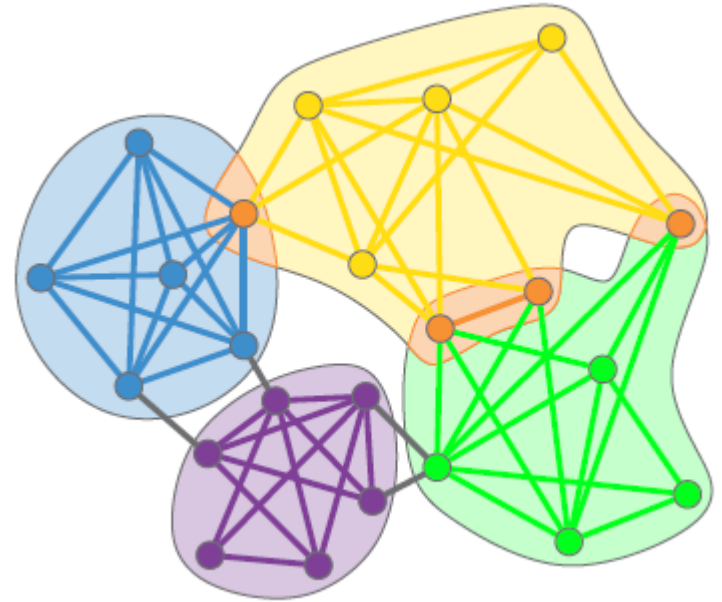


En el zoom se han coloreado las comunidades y los nodos que pertenecen a más de una comunidad aparecen en rojo



# Agrupamiento de enlaces

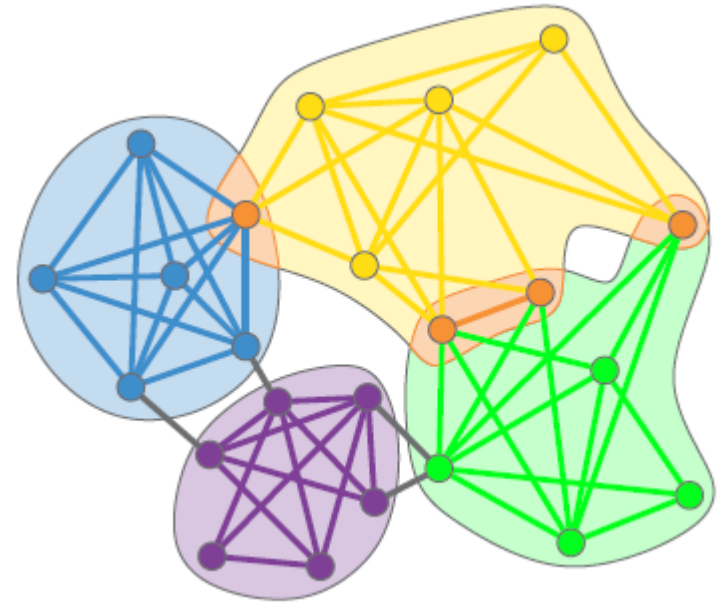
- Nodos pueden pertenecer a más de una comunidad
- Los enlaces capturan la naturaleza de la interacción entre nodos por lo que tienden a ser más específicos.
  - Redes Sociales: vínculos de familia, trabajo, hobby, club, etc
  - Redes biológicas: interacción de lugar a función biológica.



Idea: **agrupar enlaces** de manera jerárquica.  
Sólo es necesario saber como medir **similitud entre enlaces**

# Agrupamiento de enlaces

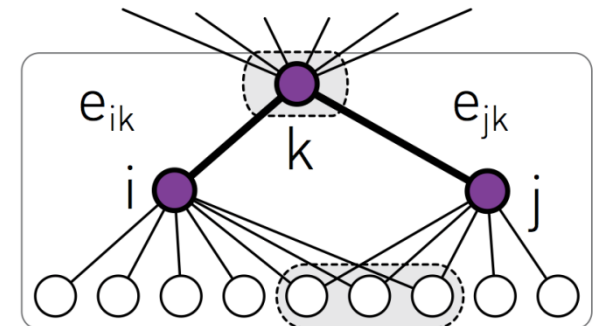
- Nodos pueden pertenecer a más de una comunidad
- Los enlaces capturan la naturaleza de la interacción entre nodos por lo que tienden a ser más específicos.
  - Redes Sociales: vínculos de familia, trabajo, hobby, club, etc
  - Redes biológicas: interacción de lugar a función biológica.



Idea: **agrupar enlaces** de manera jerárquica.  
Sólo es necesario saber como medir **similitud entre enlaces**

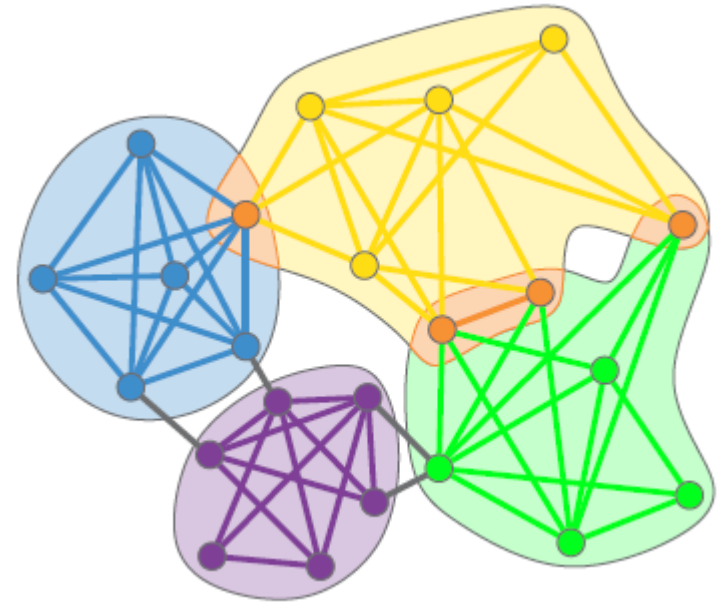
$$S(e_{ik}, e_{jk}) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|}$$

$n_+(i)$ : cjto de vecinos del nodo- $i$  U nodo- $i$



# Agrupamiento de enlaces

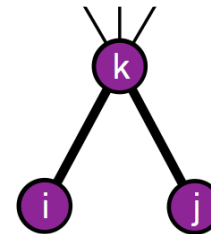
- Nodos pueden pertenecer a más de una comunidad
- Los enlaces capturan la naturaleza de la interacción entre nodos por lo que tienden a ser más específicos.
  - Redes Sociales: vínculos de familia, trabajo, hobby, club, etc
  - Redes biológicas: interacción de lugar a función biológica.



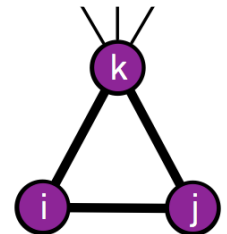
Idea: **agrupar enlaces** de manera jerárquica.  
Sólo es necesario saber como medir **similitud entre enlaces**

$$S(e_{ik}, e_{jk}) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|}$$

$n_+(i)$ : cjto de vecinos del nodo- $i$  U nodo- $i$

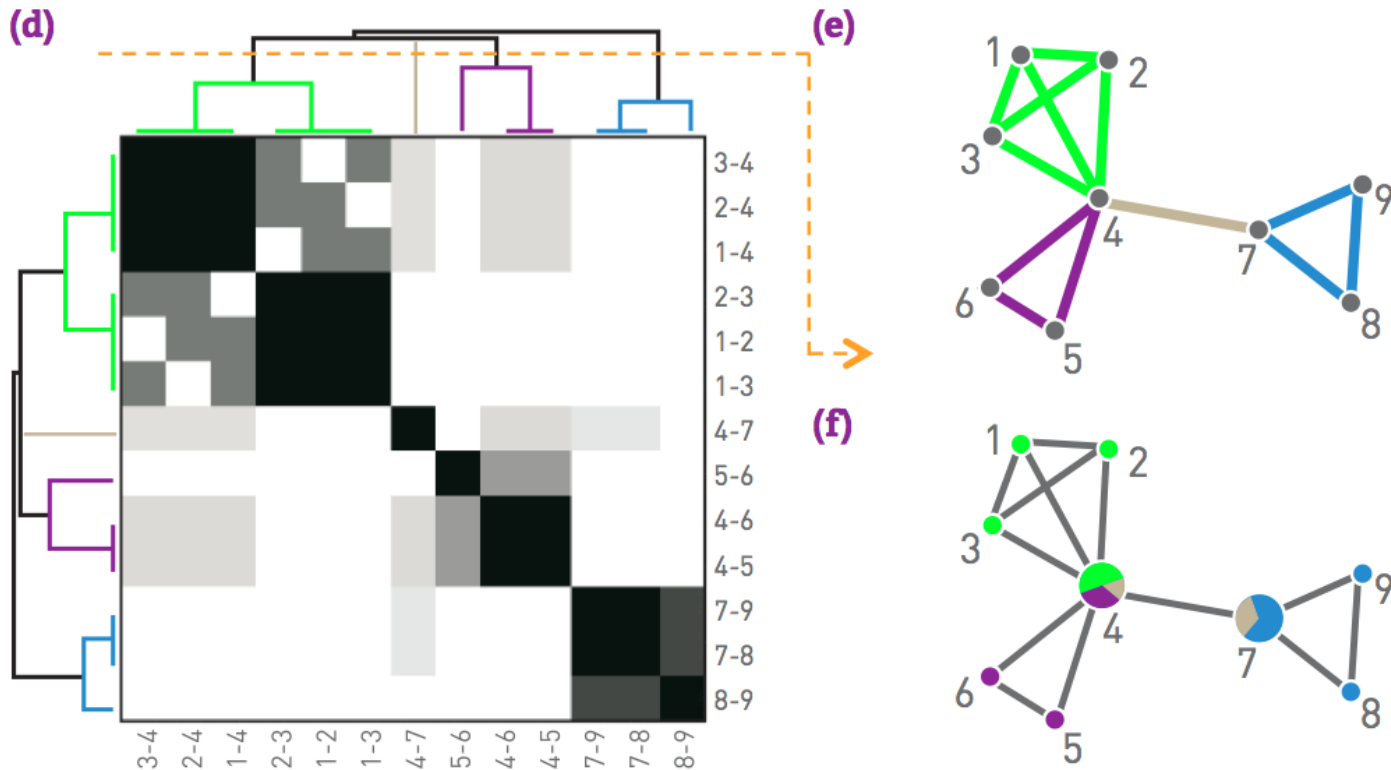


$$S(e_{ik}, e_{jk}) = \frac{1}{3}$$



$$S(e_{ik}, e_{jk}) = 1$$

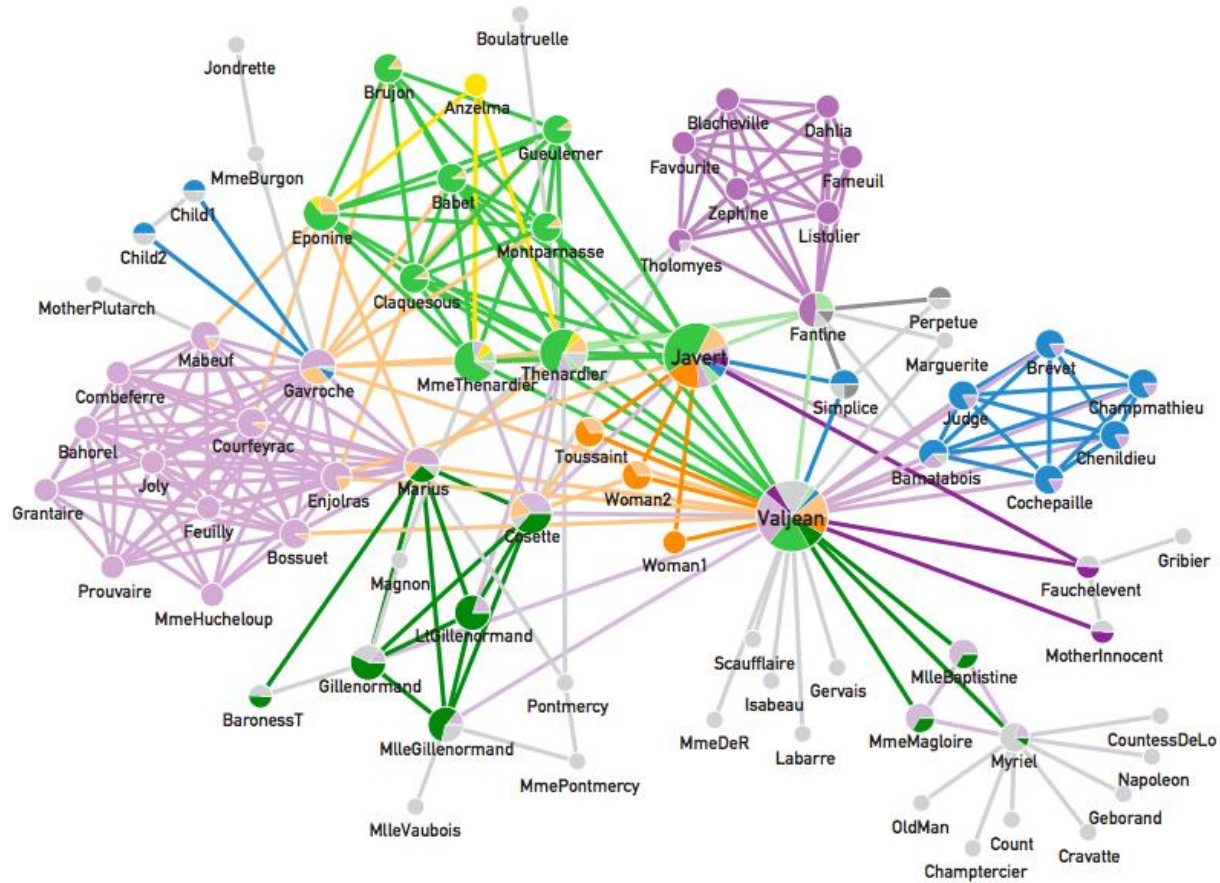
# Agrupamiento de enlaces



$$S(e_{34}, e_{24}) = \frac{|n_+(3) \cap n_+(2)|}{|n_+(3) \cup n_+(2)|} = \frac{|\{1,2,4,3\} \cap \{1,3,4,2\}|}{|\{1,2,4,3\} \cup \{1,3,4,2\}|} = 1$$

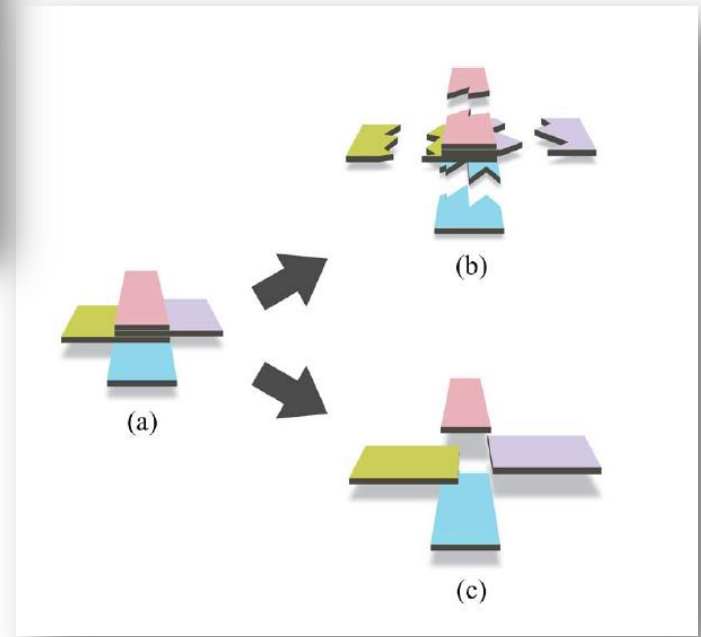
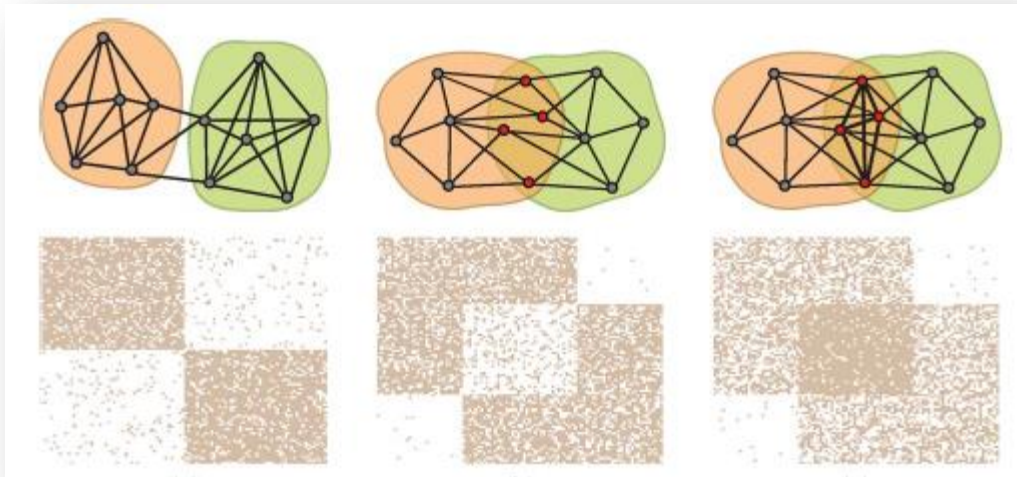
$$S(e_{54}, e_{24}) = \frac{|n_+(5) \cap n_+(2)|}{|n_+(5) \cup n_+(2)|} = \frac{|\{6,4,5\} \cap \{1,3,4,2\}|}{|\{6,4,5\} \cup \{1,3,4,2\}|} = \frac{1}{6}$$

# Agrupamiento de enlaces

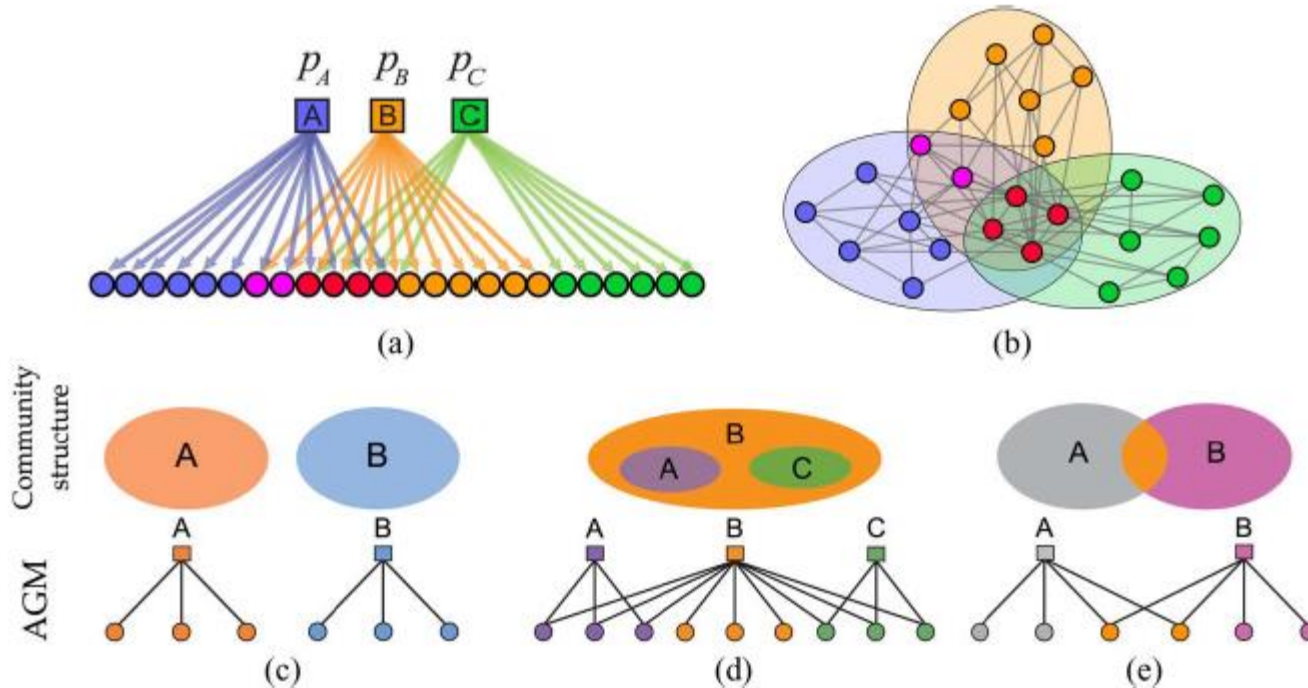


Nota: En general este enfoque anda bien cuando el overlap entre comunidades es poco denso. Otra alternativa interesante es el Affiliation Graph Model de Leskovec et al

# Affiliation Graph Model



# Affiliation Graph Model



Modela la red a partir de una estructura de red de filiación subyacente  
Estima el **conjunto de filiaciones**:  $\{F_A, F_B, \dots, F_M\}$  y las probabilidades de **intraconectividad**  $\{p_A, p_B, \dots, p_M\}$  respectivas

# Evaluando Particiones

- Medidas internas:
  - Idea general, cuantificar nivel de compacidad/separación de grupos
    - Modularidad
    - Silhouette
    - ....



# Evaluando Particiones

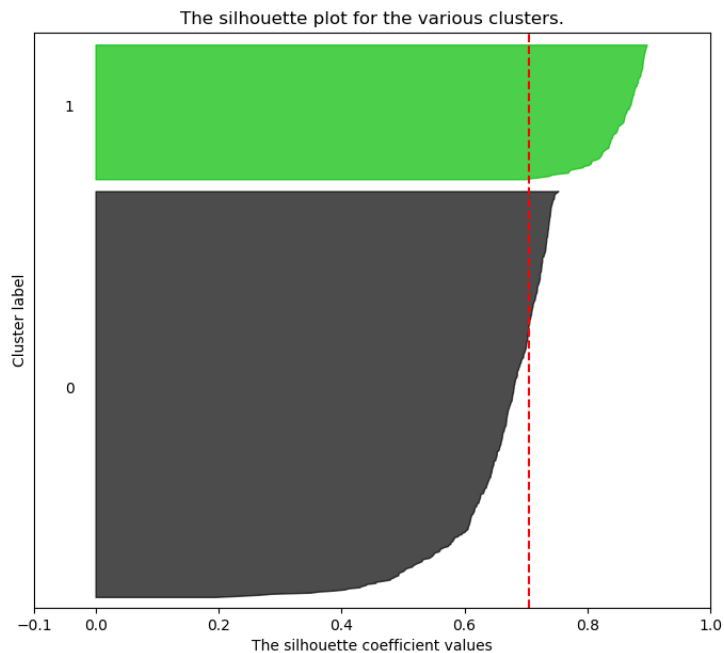
## Compacidad/Separación de grupos : Silhouette

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$a(i)$ : distancia media del nodo- $i$  con el resto de su cluster

$b(i)$ : mín distancia del nodo- $i$  con lgun elemento de otro cluster

**Silhouette analysis for KMeans clustering on sample data with n\_clusters = 2**



# Evaluando Particiones

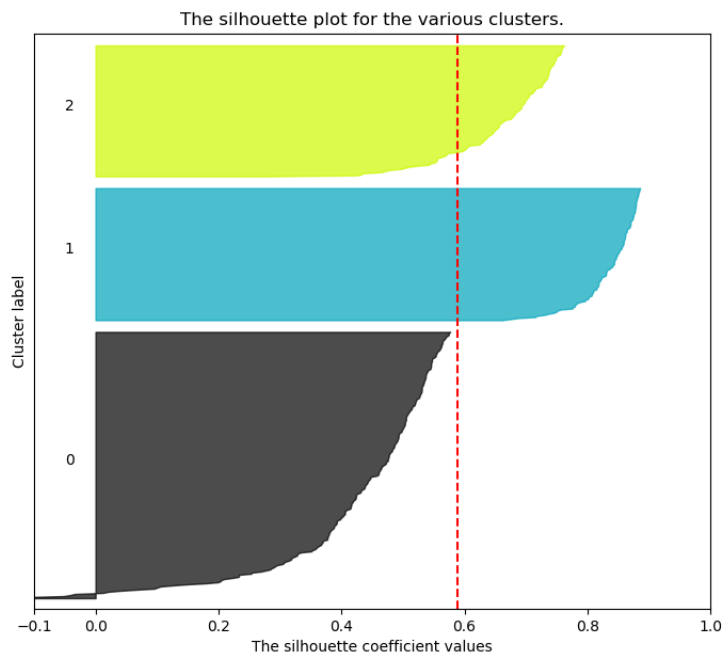
## Compacidad/Separación de grupos : Silhouette

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$a(i)$ : distancia media del nodo- $i$  con el resto de su cluster

$b(i)$ : mín distancia del nodo- $i$  con lgun elemento de otro cluster

**Silhouette analysis for KMeans clustering on sample data with n\_clusters = 3**



# Evaluando Particiones

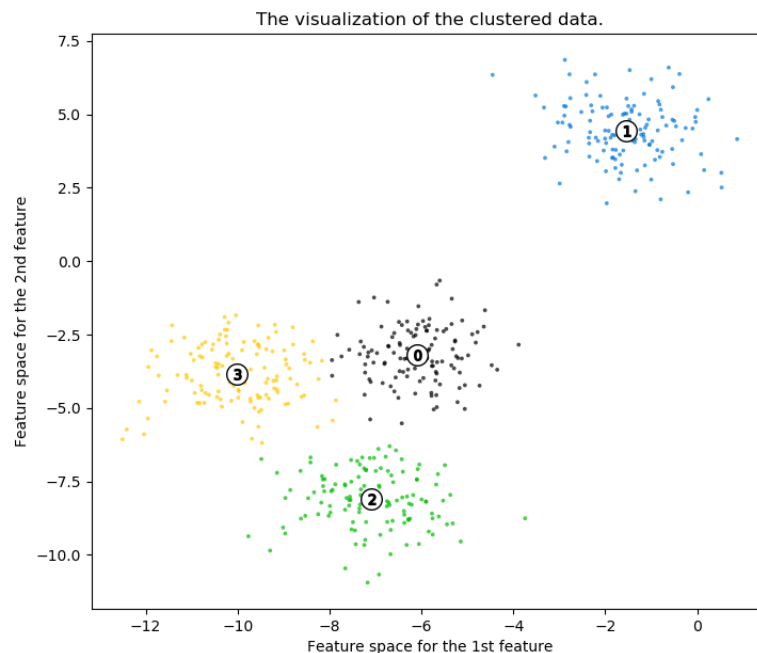
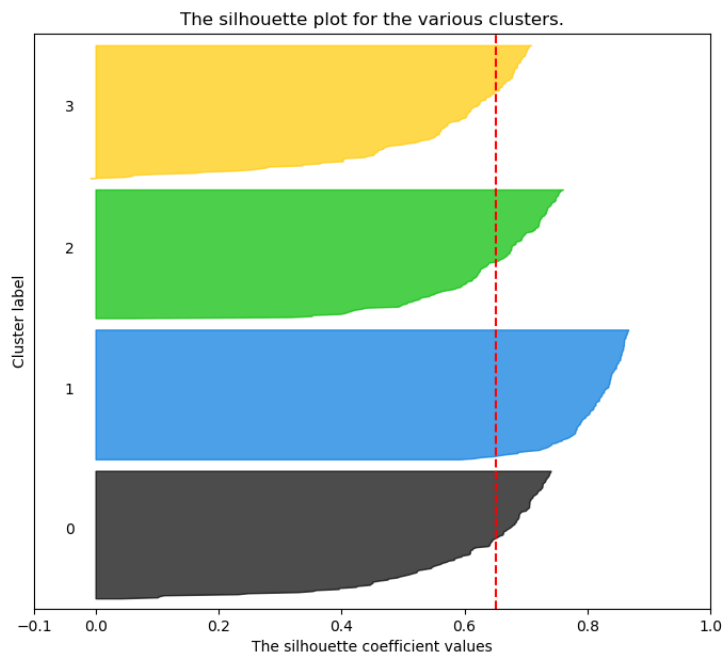
## Compacidad/Separación de grupos : Silhouette

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$a(i)$ : distancia media del nodo- $i$  con el resto de su cluster

$b(i)$ : mín distancia del nodo- $i$  con lgun elemento de otro cluster

**Silhouette analysis for KMeans clustering on sample data with n\_clusters = 4**



# Evaluando Particiones

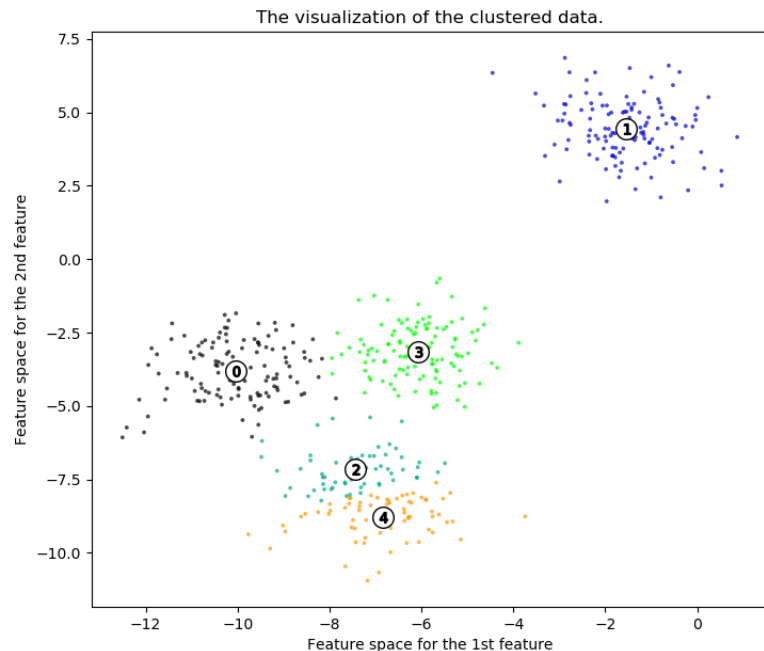
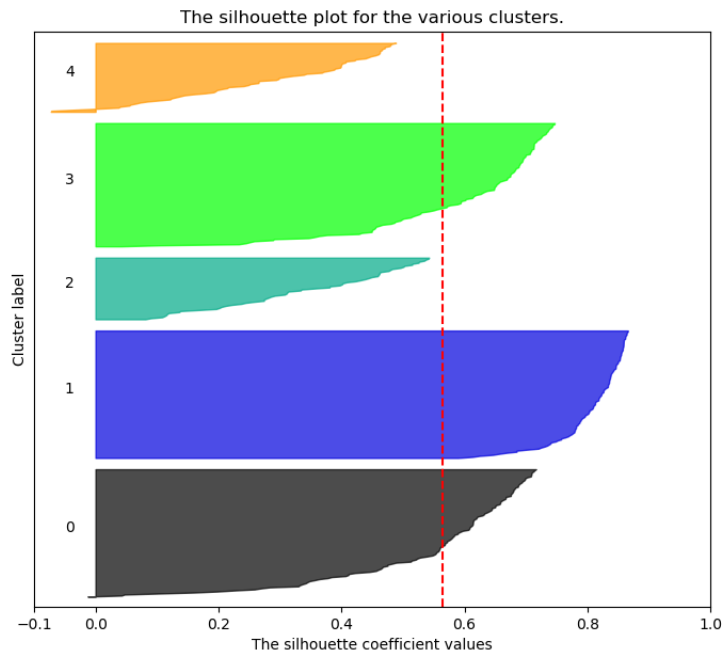
## Compacidad/Separación de grupos : Silhouette

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$a(i)$ : distancia media del nodo- $i$  con el resto de su cluster

$b(i)$ : mín distancia del nodo- $i$  con lgun elemento de otro cluster

**Silhouette analysis for KMeans clustering on sample data with n\_clusters = 5**



# Evaluando Particiones

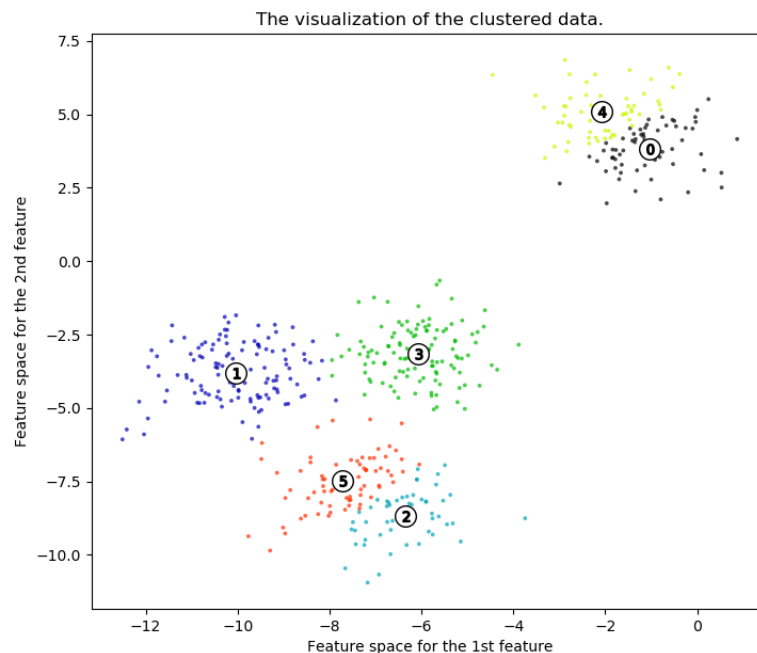
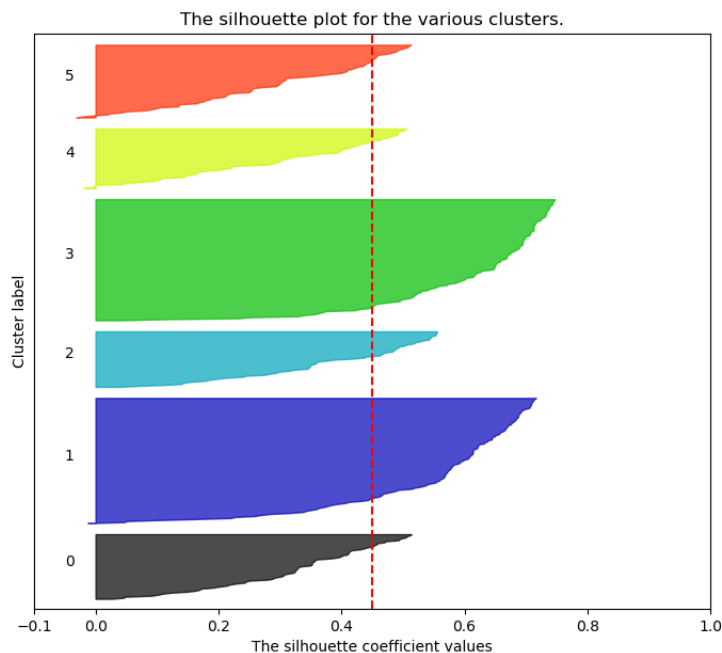
## Compacidad/Separación de grupos : Silhouette

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

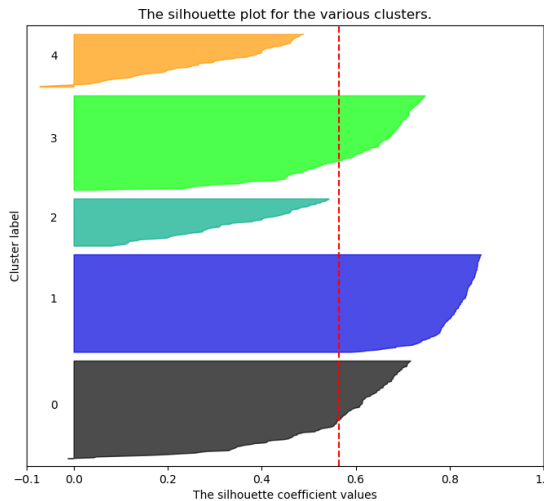
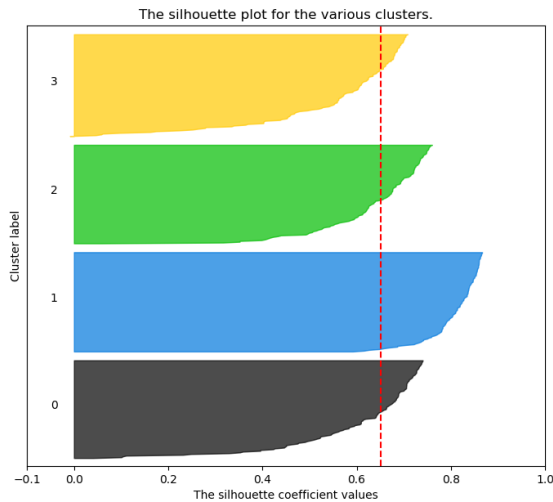
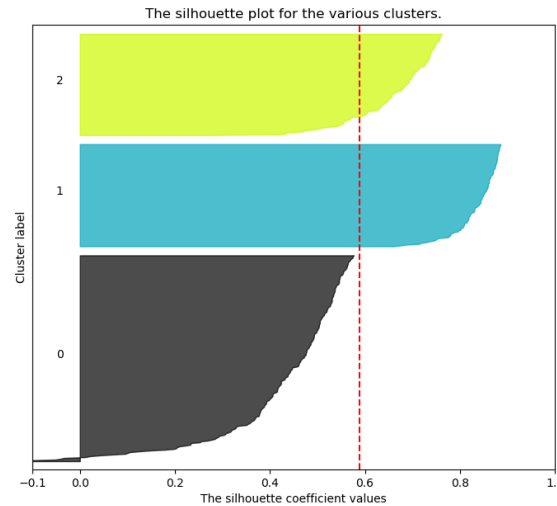
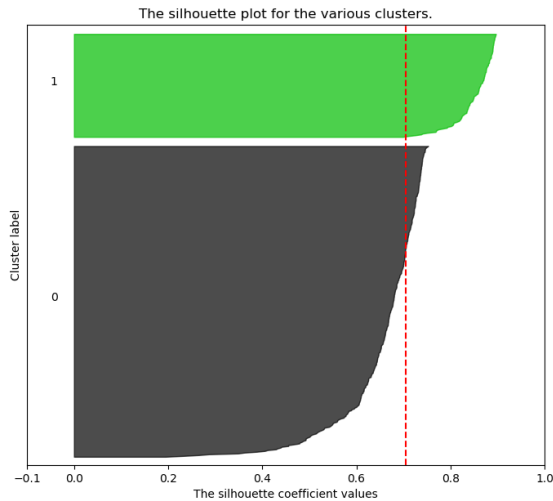
a(i): distancia media del nodo-i con el resto de su cluster

b(i): mín distancia del nodo-i con lgun elemento de otro cluster

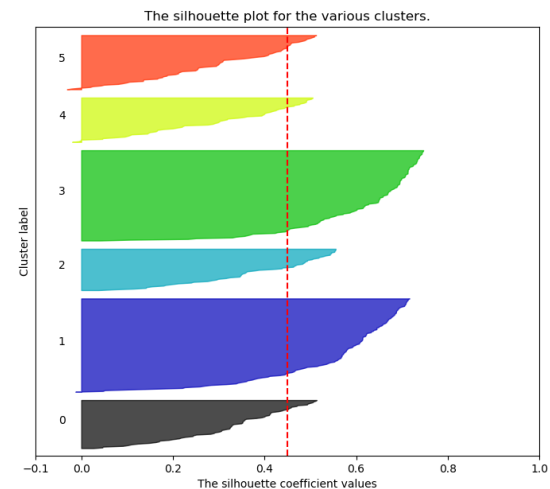
**Silhouette analysis for KMeans clustering on sample data with n\_clusters = 6**



# Evaluando Particiones



- Altura de stacks idea de asimetría en tamaño de clusters
- Buen clustering = alto silhouette
- Particiones  $N_{clus}=3,5,6$  tienen clusters con Silhouette score por debajo de la media



# Evaluando Particiones

Compacidad/Separación de grupos : Silhouette

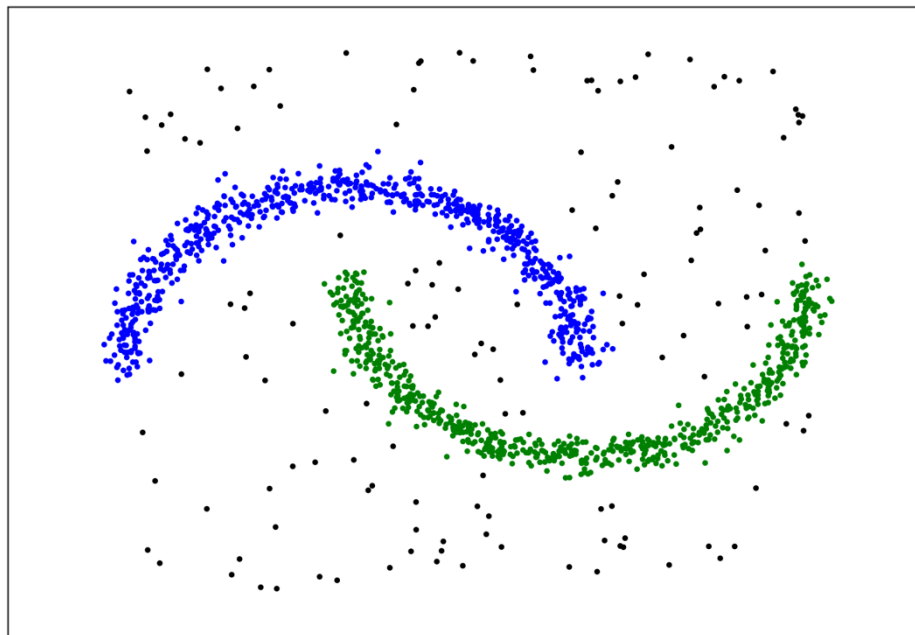
**OJO** con esto!!!

Los datos no tienen la culpa....

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$a(i)$ : distancia media del nodo- $i$  con el resto de su cluster

$b(i)$ : mínima distancia del nodo- $i$  con lgun elemento de otro cluster



# Evaluando Particiones

- En ausencia de información externa
  - Medidas internas
    - Modularidad
    - Silhouette
    - ...
- Con **información externa**
  - Partición de referencia disponible
    - Información mútua
    - Precisión de la asignación de grupos
  - Conocimiento externo
    - Coherencia de grupos



# Cuantificando resultados contra partición de referencia



Agrupamiento **Real**

Resultado de Clustering

- El nro de comunidades puede no ser el mismo
- Puede no haber una correspondencia clara en la composición de clusters

Cómo cuantificar acuerdo/desacuerdo?

# Prueba/Teoría de la Información

- Consideremos a la asignación en clusters como una variable aleatoria que se le asigna a cada nodo

$$p(C_1) = \frac{N_{C_1}}{\sum_C N_C} : \text{probabilidad de que un nodo elegido al azar pertenezca a la comunidad } C_1 \text{ de una dada partición}$$

- Si tengo dos particiones alternativas, puedo computar la probabilidad conjunta

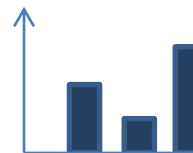
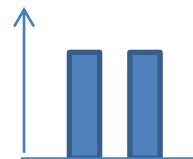
$$p(C_1, C_2) = \frac{N_{C_1, C_2}}{\sum_{C_1, C_2} N_{C_1, C_2}} : \text{probabilidad } \mathbf{conjunta} \text{ de que un nodo elegido al azar pertenezca a la comunidad } C_1 \text{ de la primera partición y a la } C_2 \text{ de la segunda}$$

Partición a: [1, 1, 1, 2, 2, 2]

Partición b: [1, 2, 1, 3, 3, 3]



$N_{C_1}$



$N_{C_1, C_2}$  Matriz de coocurrencia

	j=1	j=2	j=3
i=1	2	1	0
i=2	0	0	3

# Prueba/Teoría de la Información

- Consideremos a la asignación en clusters como una variable aleatoria que se le asigna a cada nodo

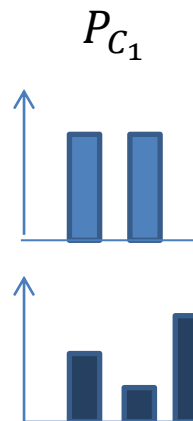
$$p(C_1) = \frac{N_{C_1}}{\sum_C N_C} : \text{probabilidad de que un nodo elegido al azar pertenezca a la comunidad } C_1 \text{ de una dada partición}$$

- Si tengo dos particiones alternativas, puedo computar la probabilidad conjunta

$$p(C_1, C_2) = \frac{N_{C_1 C_2}}{\sum_{C_1, C_2} N_{C_1 C_2}} : \text{probabilidad } \mathbf{conjunta} \text{ de que un nodo elegido al azar pertenezca a la comunidad } C_1 \text{ de la primera partición y a la } C_2 \text{ de la segunda}$$

Partición a: [1, 1, 1, 2, 2, 2]

Partición b: [1, 2, 1, 3, 3, 3]



$P_{C_1 C_2} \quad (P_{C_1} P_{C_2})$

	j=1	j=2	j=3
i=1	1/3 (1/6)	1/6(1/12)	0 (1/4)
i=2	0 (1/6)	0 (1/12)	1/4 (1/4)

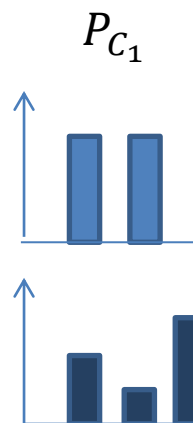
# Información Mútua

- Si dos particiones son similares obtengo mucha información de un agrupamiento conociendo el otro.
- Enfoque de teoría de la información: la **información mútua** entre dos variables aleatorias cuantifica la cantidad de información (en bits) que obtengo de una variable, a partir de la otra

$$I(\{C_1\}, \{C_2\}) = \sum_{C_1} \sum_{C_2} p(C_1, C_2) \log \frac{p(C_1, C_2)}{p(C_1)p(C_2)} \leftarrow \begin{array}{l} \text{probabilidad conjunta si} \\ \text{asumimos } \mathbf{independencia} \\ \text{entre particiones} \end{array}$$

Partición a: [1, 1, 1, 2, 2, 2]

Partición b: [1, 2, 1, 3, 3, 3]



	$j=1$	$j=2$	$j=3$
$i=1$	1/3 (1/6)	1/6 (1/12)	0 (1/4)
$i=2$	0 (1/6)	0 (1/12)	1/4 (1/4)

$$I(\{C_1\}, \{C_2\}) = \sum_{C_1} \sum_{C_2} p(C_1, C_2) \log \frac{p(C_1, C_2)}{p(C_1)p(C_2)}$$

# Información Mútua

- Versión normalizada de información mútua

$$I_n = \frac{2 I(\{C_1\}, \{C_2\})}{H(\{C_1\}) + H(\{C_2\})} = 0.8278$$

$$0 < I_n < 1$$

Entropía de Shannon:

cuanta información obtengo al conocer la realización de una variable aleatoria

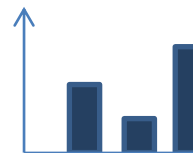
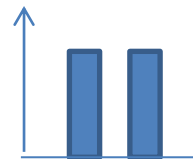
$$H(\{C_1\}) = - \sum_{C_1} p_{C_1} \log p_{C_1} \\ = - \sum_{C_1} \frac{N_{C_1}}{\sum_C N_C} \log \frac{N_{C_1}}{\sum_C N_C}$$

Partición a: [1, 1, 1, 2, 2, 2]

Partición b: [1, 2, 1, 3, 3, 3]



$P_{C_1}$



$P_{C_1 C_2} (P_{C_1} P_{C_2})$

	j=1	j=2	j=3
i=1	1/3 (1/6)	1/6 (1/12)	0 (1/4)
i=2	0 (1/6)	0 (1/12)	1/4 (1/4)

# Precisión

- Consideramos todos los posibles pares de nodos y evaluamos si residen en la misma comunidad que en la partición de referencia luego de la detección
- Habra un **error** si
  - Dos nodos que pertenecen a la **misma** comunidad de referencia son asignados a comunidades **diferentes**
  - Dos nodos de **diferentes** comunidades de referencia son asignados a una **misma** comunidad
- Para cuantificar esto armamos: **matriz confusión**

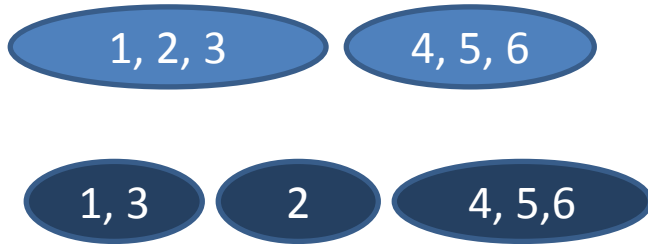
		<u>Referencia</u>	
		C(i)=C(j)	C(i)≠C(j)
<u>clustering</u>	C(i)=C(j)	a	b
	C(i)≠C(j)	c	d

$$precisión = \frac{a + d}{a + b + c + d} = \frac{a + d}{n(n - 1)/2}$$

# Precisión

Partición Ref: [1, 1, 1, 2, 2, 2]

Partición : [1, 2, 1, 3, 3, 3]



clustering

Referencia

	C(i)=C(j)	C(i)≠C(j)
C(i)=C(j)	4	0
C(i)≠C(j)	2	9

$$precisión = \frac{a + d}{a + b + c + d} = \frac{a + d}{n(n - 1)/2}$$

$$precisión = 0.87$$

# Detección de Comunidades

Basados en noción de  
Distancia/Similaridad

- Agrupamiento Jerárquico
- k-means
- PAM
- ...

Basados en optimización de  
figura de mérito

- Newman-Girvan
- fast greedy
- Louvain
- Infomap

Detección comunidades  
solapadas

- Percolación de cliques
- Agrupamiento de enlaces
- Affinity Graph Model

## Validación

---

En ausencia de información externa

- Medidas internas
  - Modularidad
  - Silhouette
  - ...

Con **información externa**

- Partición de referencia disponible
  - Información mútua
  - Precisión de la asignación de grupos

Conocimiento externo

- Coherencia de grupos