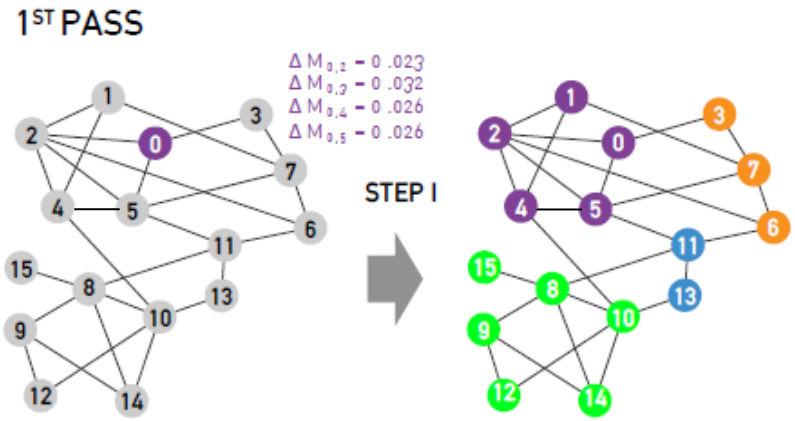
Detección de Comunidades II

- 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

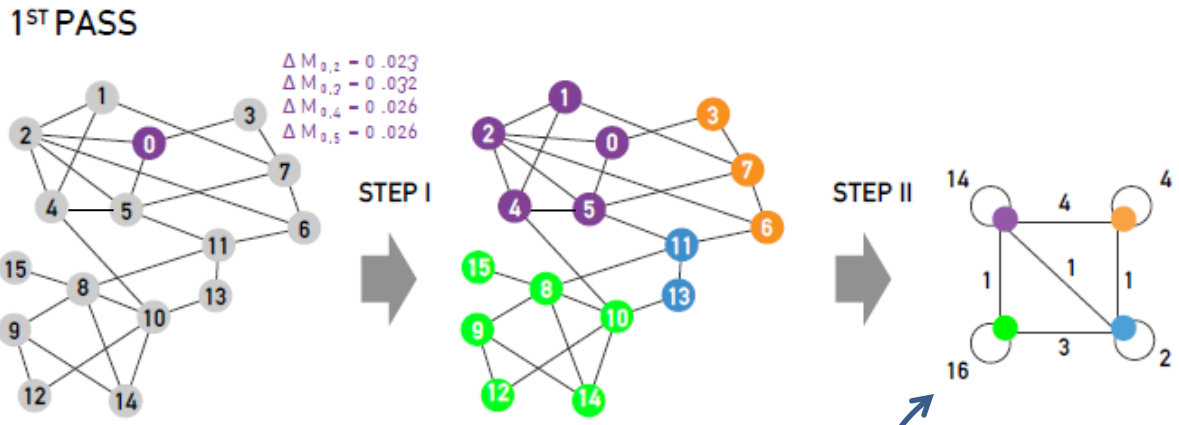
Se optimiza M con cambios locales, tratando de unir un nodo con sus vecinos. Se elige el cambio de mayor ΔM (si el cambio es positivo) Se repite esto para cada nodo de la red

Blondel, J.Stat.Mech 2008

Algoritmo Louvain

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

Modularidad **M** optimizada en pasadas de dos pasos



$$k_{verde} = 2L_{verde}$$

Paso 1

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

Paso 2

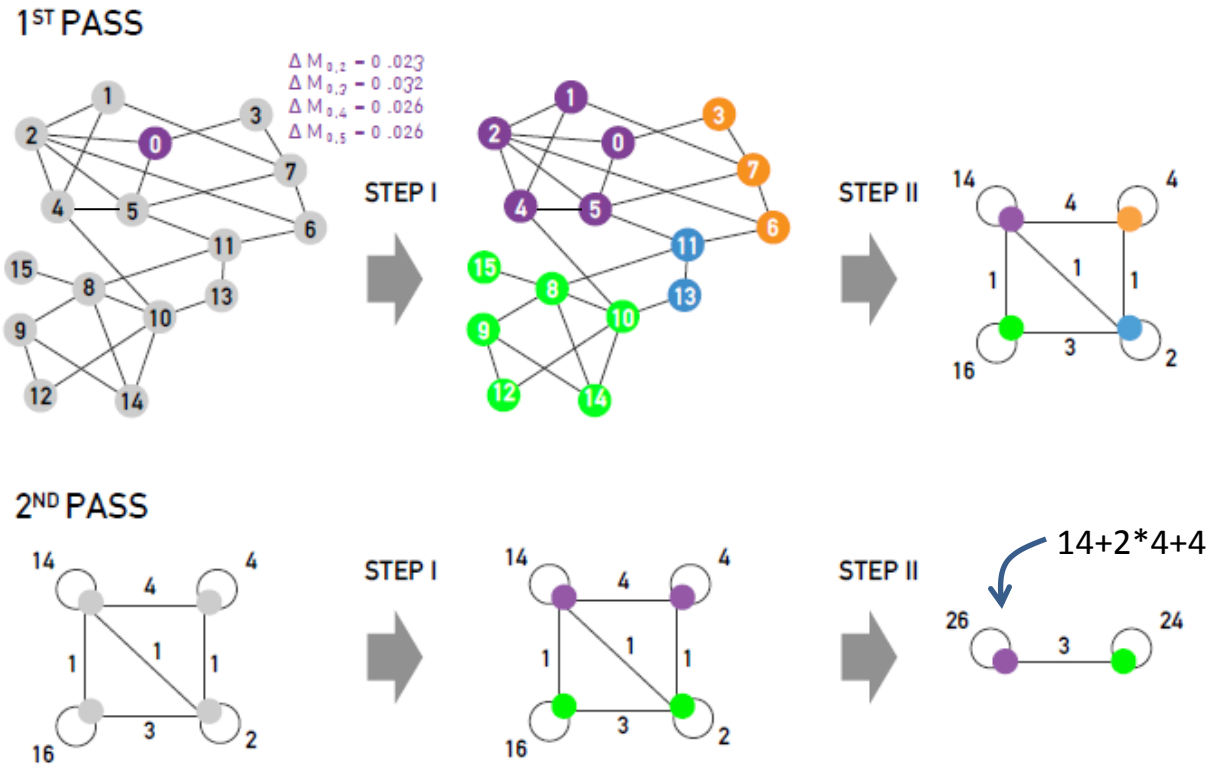
Se arma una nueva red donde cada nodo es una comunidad encontrada en el Paso 1. Se generan auto-enlaces que corresponden a lazos intra-comunidad.

Blondel, J.Stat.Mech 2008

Algoritmo Louvain

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

Modularidad **M** optimizada en pasadas de dos pasos



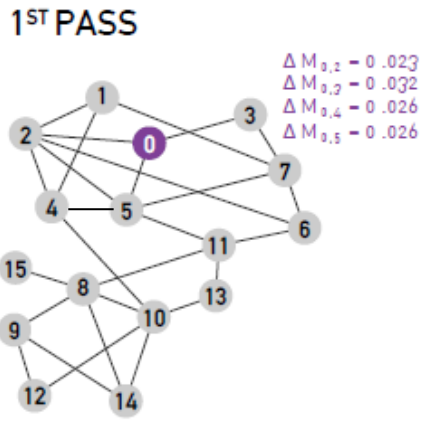
Se repite hasta no poder obtener incremento de modularidad

Blondel, J.Stat.Mech 2008

Algoritmo Louvain

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

Modularidad **M** optimizada en pasadas de dos pasos



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

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

Cambio de M:
nodo-i anexado a comunidad C

suma de pesos de nodo-i a nodos del cluster

suma de pesos enlaces 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 intra-C

Infomap

Existe una **dualidad** entre

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

C E G A D F B E A…

01000011 01000101 01000111 01000001 01000100 01000110 01000010 01000101 01000001

9 letras (ASCII characters) -> 9 bytes (72 bits)

Se puede hacer mejor?

Huffman code:

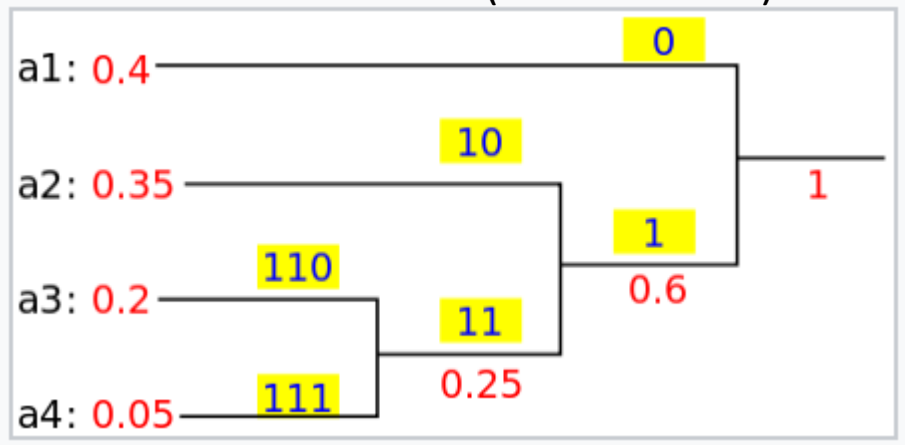
- Optimally efficient code
- Message dependant
- Prefix-free

Huffman trees

Una fuente aleatoria genera 4 simbolos diferentes: $\{a_1, a_2, a_3, a_4\}$ con probabilidades $p_{a_1} = 0.4, p_{a_2} = 0.35, p_{a_3} = 0.2, p_{a_4} = 0.04$
cuantos bits/simbolo necesito para transmitir un mensaje con esta fuente?

En principio 2 bits por simbolo (para codificar 4 simbolos)….pero se puede hacer mejor

Arbol binario (Huffman tree)



Symbol	Code	fixed
a1	0	00
a2	10	01
a3	110	10
a4	111	11

Una *frase* generada por esta fuente tendra en promedio 40% de a1, 35% de a2, etc…

$$0.4 * 1 + 0.35 * 2 + 0.2 * 3 + 0.05 * 3 = 1.85 \quad \text{bits/caracter}$$

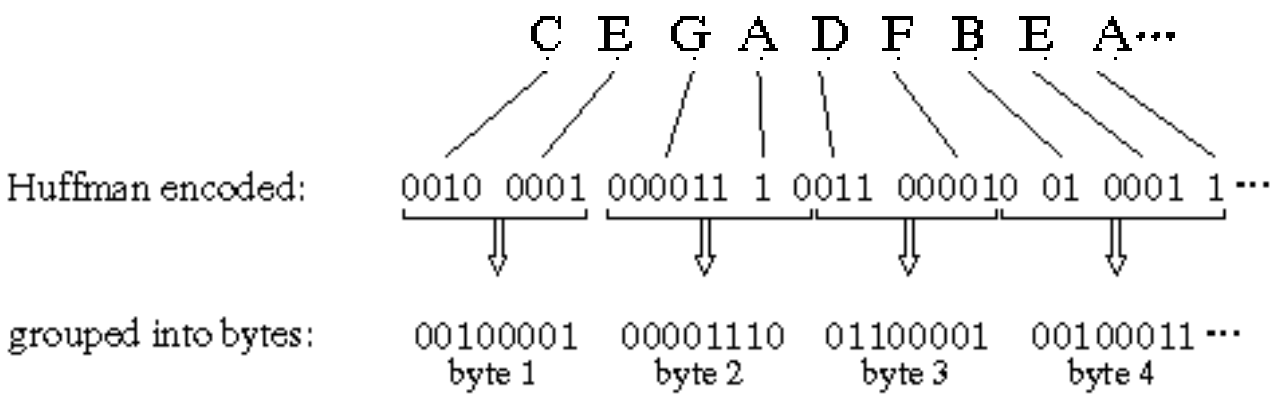
Infomap

Existe una **dualidad** entre

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

9 letras (ASCII characters) -> 9 bytes (72 bits)

Se puede hacer mejor?



Huffman code:

- Optimally efficient code
- Message dependant
- Prefix-free

Example Encoding Table

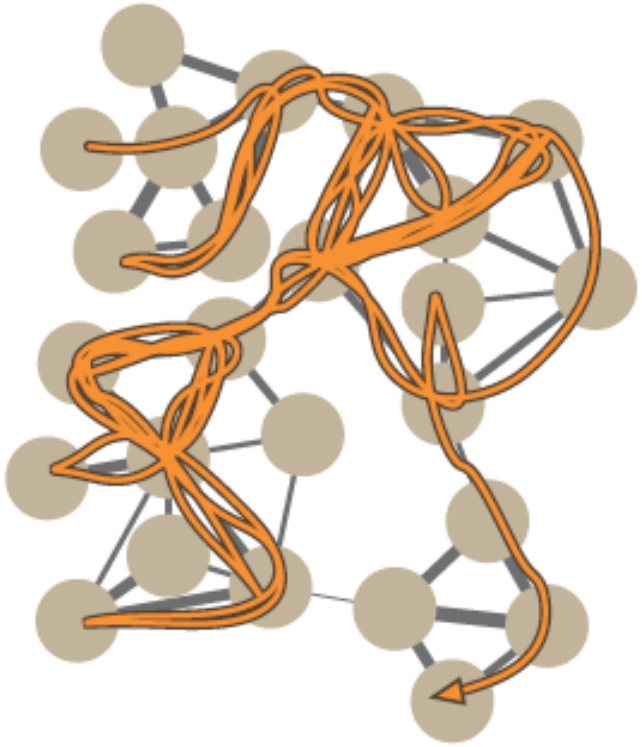
letter	probability	Huffman code
A	.154	1
B	.110	01
C	.072	0010
D	.063	0011
E	.059	0001
F	.015	000010
G	.011	000011

Manera **eficiente** de asignar etiquetas refleja regularidades estadisticas de los datos

Infomap

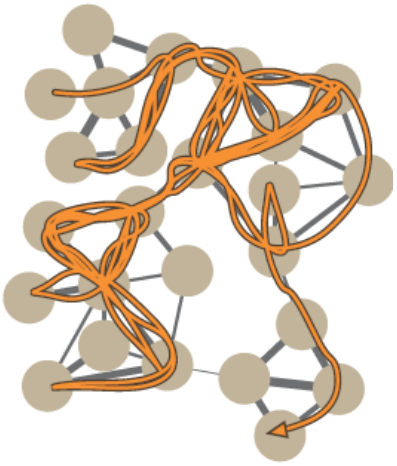
Existe una **dualidad** entre

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

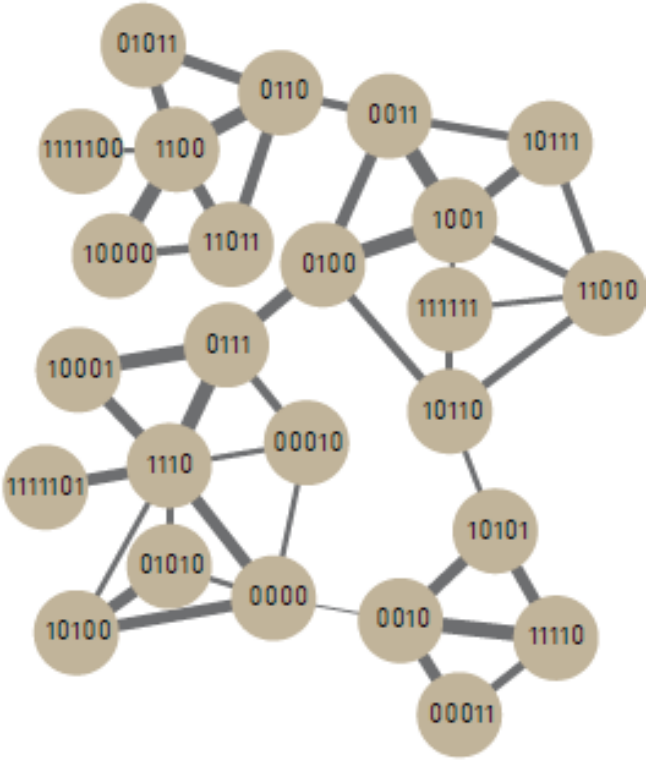


Caminatas aleatorias como indicador de estructura de la red

- Exploracion de la red
- Largos tiempos de transito intra-modular



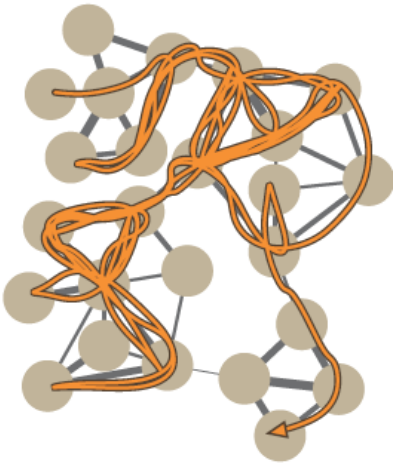
Infomap



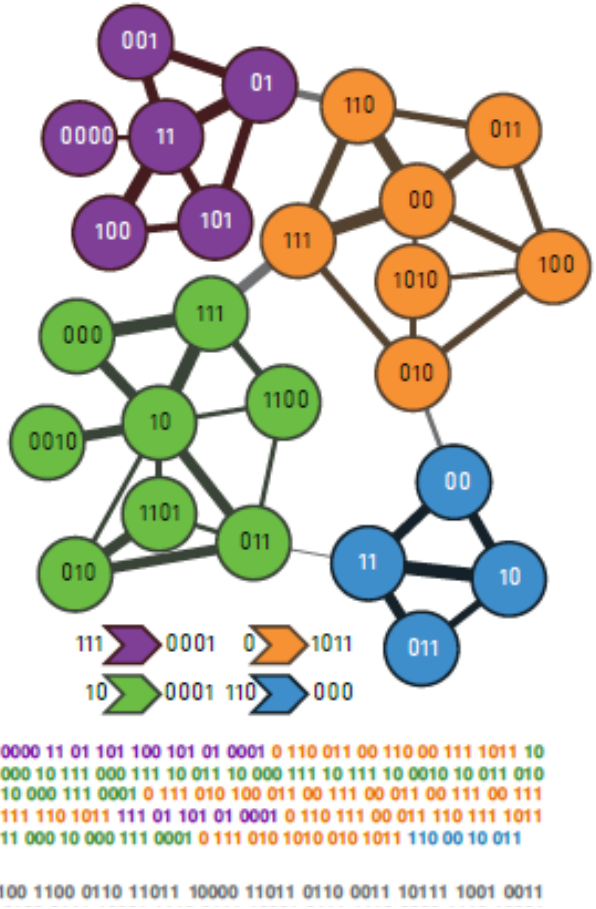
1111100 1100 0110 11011 10000 11011 0110 0011 10111 1001 0011 1001 0100 0111 10001 1110 0111 10001 0111 1110 0000 1110 10001 0111 1110 0111 1110 1111101 1110 0000 10100 0000 1110 10001 0111 0100 10110 11010 10111 1001 0100 1001 10111 1001 0100 1001 0100 0011 0100 0011 0110 11011 0110 0011 0100 1001 10111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

- Para poder describir la trayectoria de 71 pasos, necesito etiquetar a cada uno de los 25 nodos.
- Naifmente (codigo uniforme): $\log_2(25)*71=330$ bits

  Tamanio medio de etiquetas (medido en 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 71 pasos que comienza en el 1111100 y finaliza en el 00011



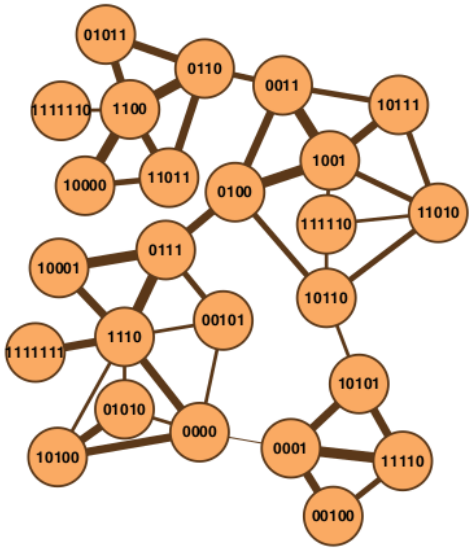
Infomap

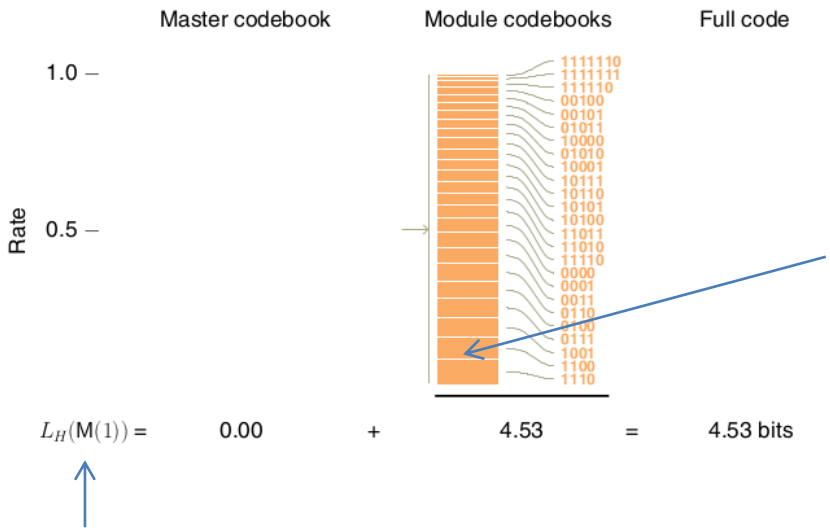


- Infomap propone un etiquetado de 2 niveles:
  - **Index code-book**: códigos asignado a entrada y salida de una comunidad
  - **Module code-book**: codigo interno de cada nodo dentro de su comunidad.
- Con esta estrategia, **si el etiquetado resuena con la estructura modular de la red**. se pueden 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

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

Cuantificando que tan buena es una descripcion

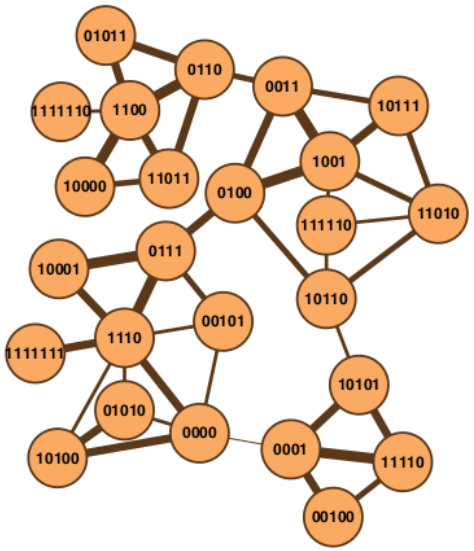


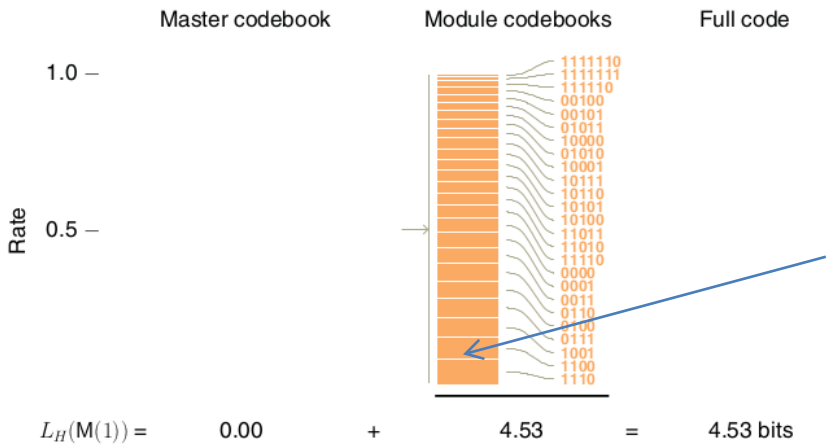


Altura proporcional a la frecuencia de visita en un random walk

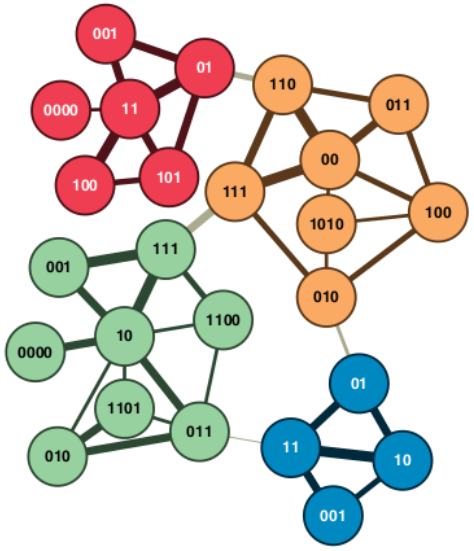
Per-step Code length: $\sum \; p_i \; length_i$

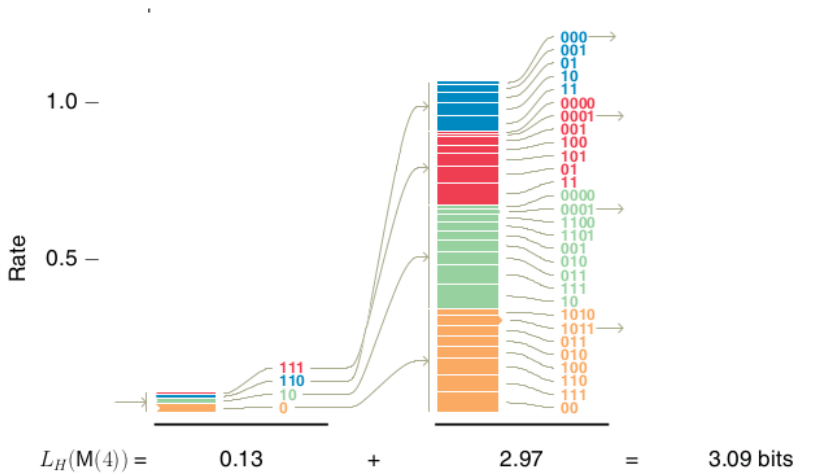
Cuantificando que tan buena es una descripcion



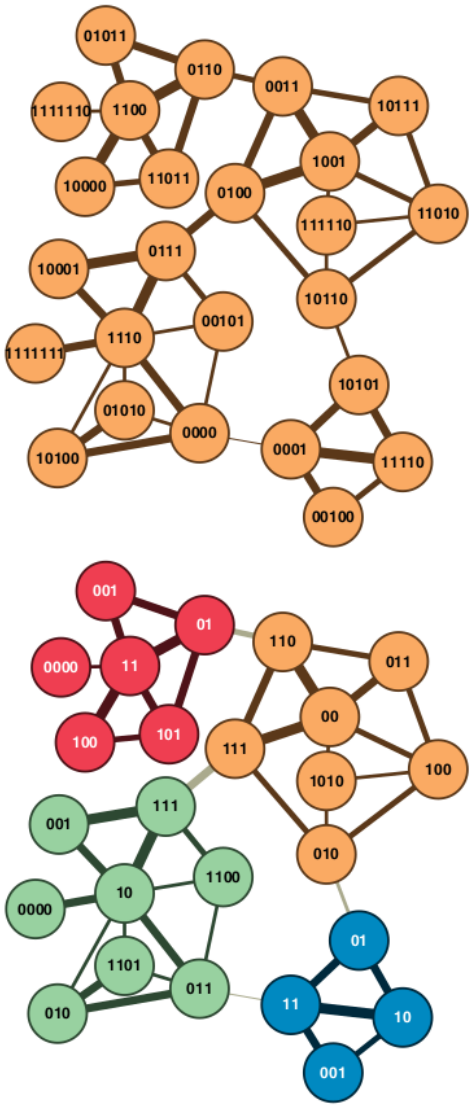


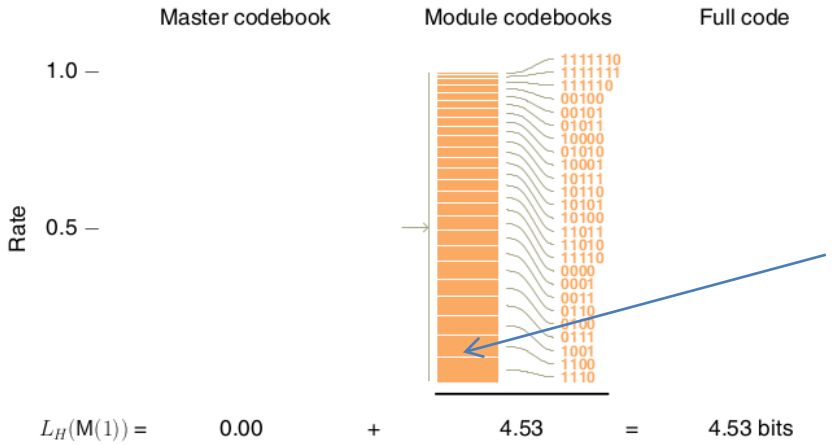
Altura proporcional a la frecuencia de visita en un random walk



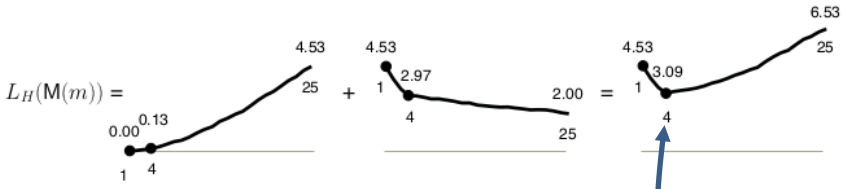


Cuantificando que tan buena es una descripcion

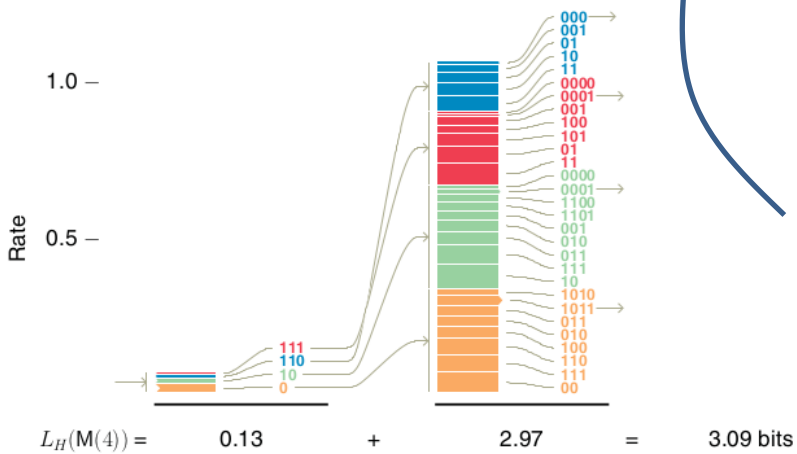




Altura proporcional a la frecuencia de visita en un random walk



Description length para numero variables de clusters de particiones “optimas"



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 source coding theorem: 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 $H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$

Cota para la longitud media de mis etiquetas de 2-niveles

Map equation:

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

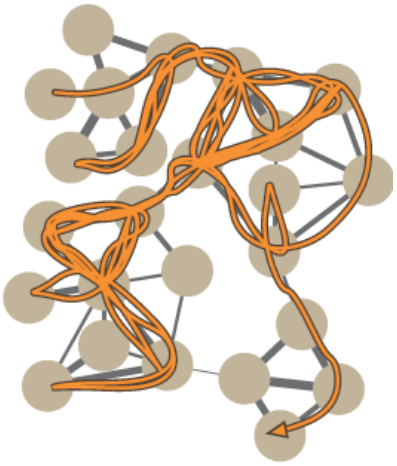
q:prob de cambiar modulos

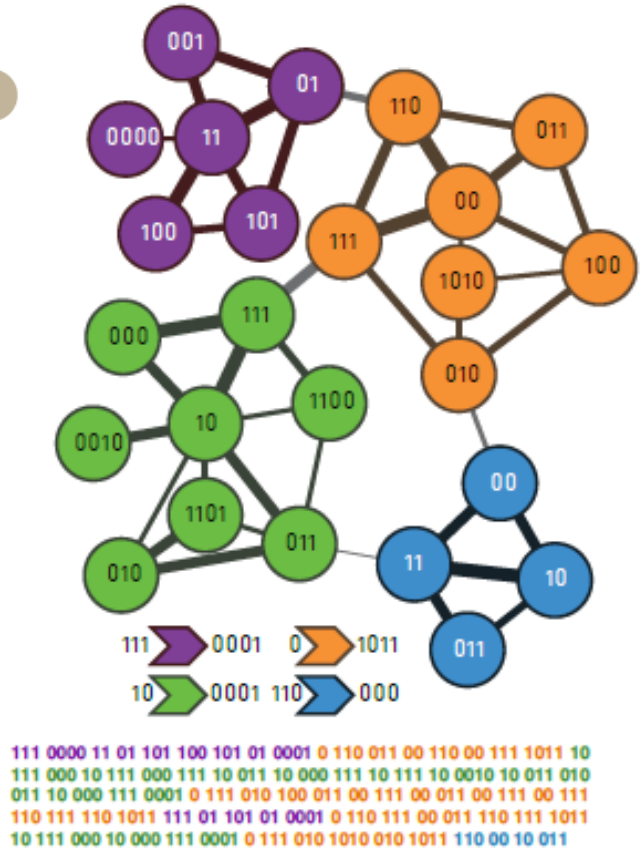
p_o^c: fracc tiempo que paso en modulo C

Long.media de etiquetas master

Long.media de etiquetas modulos

Infomap





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

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

Long.media de etiquetas master

Long.media de etiquetas modulos

q:prob de cambiar modulos

p_o^c: fracc tiempo que paso en modulo C

www.mapequation.org

Rosvall & Bergstrom PNAS 2008 Maps of random walks on complex networks reveal community structure
Rosvall, Axelsson, Bergstrom, Eur.Phys.Journal 2009, The map equation

Infomap details I

$$L(\mathsf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathcal{P}^i).$$

$$H(\mathcal{Q}) = -\sum_{i=1}^{m} \frac{q_{i\curvearrowright}}{\sum_{j=1}^{m} q_{j\curvearrowright}} \log\left(\frac{q_{i\curvearrowright}}{\sum_{j=1}^{m} q_{j\curvearrowright}}\right)$$

$$H(\mathcal{P}^i) = -\frac{q_{i\curvearrowright}}{q_{i\curvearrowright} + \sum_{\beta\in i} p_\beta} \log\left(\frac{q_{i\curvearrowright}}{q_{i\curvearrowright} + \sum_{\beta\in i} p_\beta}\right) - \sum_{\alpha\in i} \frac{p_\alpha}{q_{i\curvearrowright} + \sum_{\beta\in i} p_\beta} \log\left(\frac{p_\alpha}{q_{i\curvearrowright} + \sum_{\beta\in i} p_\beta}\right).$$

i: clusters
α: nodos

$$L(\mathsf{M}) = \left(\sum_{i=1}^{m} q_{i\curvearrowright}\right) \log\left(\sum_{i=1}^{m} q_{i\curvearrowright}\right) - 2\sum_{i=1}^{m} q_{i\curvearrowright} \log\left(q_{i\curvearrowright}\right) - \sum_{\alpha=1}^{n} p_\alpha \log\left(p_\alpha\right) + \sum_{i=1}^{m} \left(q_{i\curvearrowright} + \sum_{\alpha\in i} p_\alpha\right) \log\left(q_{i\curvearrowright} + \sum_{\alpha\in i} p_\alpha\right).$$

Independiente de la partcion

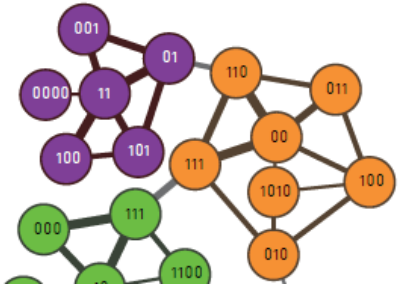
Para conocer L(M) solo es necesario estimar

$$q_{i\curvearrowright} \qquad \sum_{\alpha\in i} p_\alpha$$

Infomap details II

Para obtener L(M) solo es necesario estimar $q_{i\curvearrowright} \quad \sum_{\alpha \in i} p_\alpha$

Para redes **no dirigidas**:



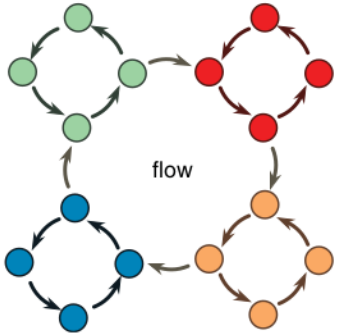
$$w_{\curvearrowright} = \sum_{i=1}^{m} w_{i\curvearrowright} \qquad w_i = \sum_{\alpha \in i} w_\alpha$$

Fraccion de pesos salientes de un cluster

Fraccion de pesos de un cluster

$$L(\mathrm{M}) = w_{\curvearrowright} \log\left(w_{\curvearrowright}\right) - 2\sum_{i=1} w_{i\curvearrowright} \log\left(w_{i\curvearrowright}\right)$$
$$- \sum_{\alpha=1}^{n} w_\alpha \log \qquad) + \sum_{i=1}^{m} \left(w_{i\curvearrowright} + w_i\right) \log\left(w_{i\curvearrowright} + w_i\right).$$

Infomap details III



Para obtener L(M) solo es necesario estimar $q_{i\curvearrowright}$ $\sum_{\alpha\in i} p_\alpha$

Para redes **dirigidas**: metodo de potencias (iterativo)

Modelo ***random surfer.***

En cada iteración

- Con prob $1-\tau$ se desplaza por algún out-link del nodo actual α al nodo β con prob $w_{\alpha\beta}$
- Con prob τ se *teletransporta* a cualquier otro nodo de la red

- ✓ Se comienza con prob uniforme sobre los nodos: $p_\alpha = \frac{1}{n}$

En cada iteracion:

- ✓ Se distribuye una fraccion $1-\tau$ dela prob de cada nodo hacia nodos vecinos segun $w_{\alpha\beta}$
- ✓ El resto, se distribuye uniformemente entre todos los nodos de la red
- ✓ Salgo de la iteracion si la estimacion de p_α se estabiliza

$$q_{i\curvearrowright} = \tau \frac{n-n_i}{n} \sum_{\alpha\in i} p_\alpha + (1-\tau) \sum_{\alpha\in i} \sum_{\beta\notin i} p_\alpha w_{\alpha\beta},$$

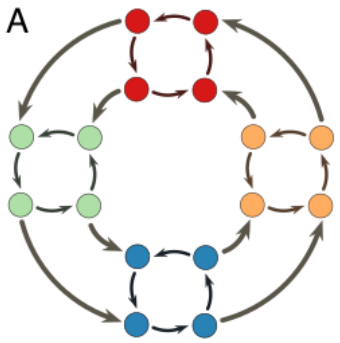
Infomap y Modularidad

$$Q = \sum_{i=1}^{m} \frac{w_{ii}}{w} - \frac{w_i^{\text{in}} w_i^{\text{out}}}{w^2}.$$

Soluciones que minimizan **L**

Soluciones que maximizan Q

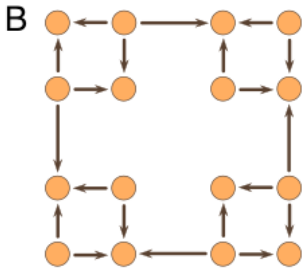
Soluciones maximizan el tiempo de transito intra-cluster de un random walker



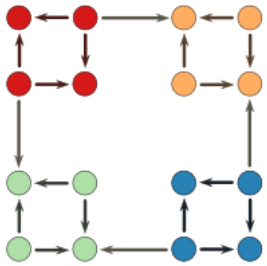
Map equation L = 2.67 bits/step
Modularity Q = 0.25

Map equation L = 4.13 bits/step
Modularity Q = 0.50

Soluciones maximizan links intra-clusters

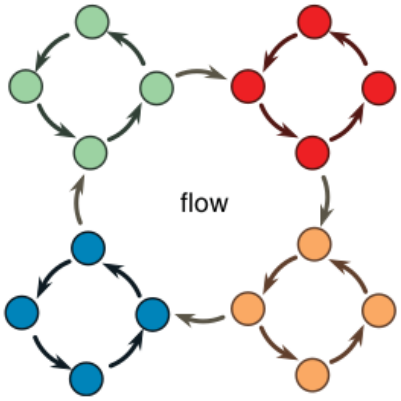


Map equation L = 2.73 bits/step
Modularity Q = 0.00

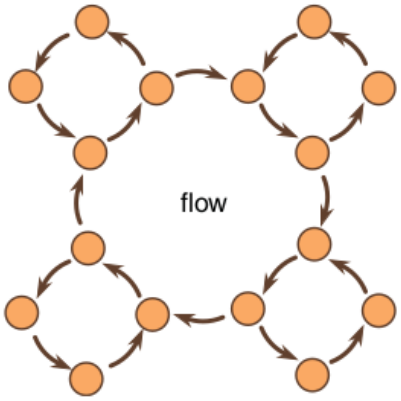


Map equation L = 4.68 bits/step
Modularity Q = 0.56

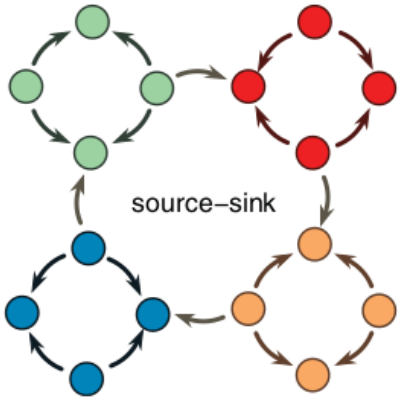
Infomap y Modularidad



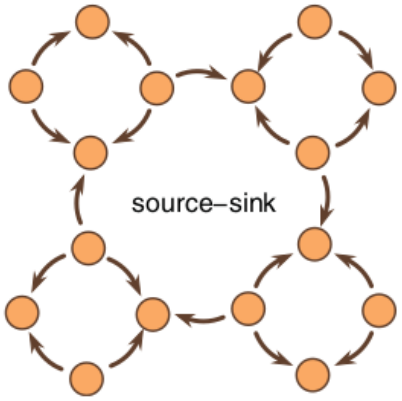
$L = \mathbf{3.33}$ $Q = \mathbf{0.55}$



$L = 3.94$ $Q = 0.00$



$L = 4.58$ $Q = \mathbf{0.55}$



$L = \mathbf{3.93}$ $Q = 0.00$

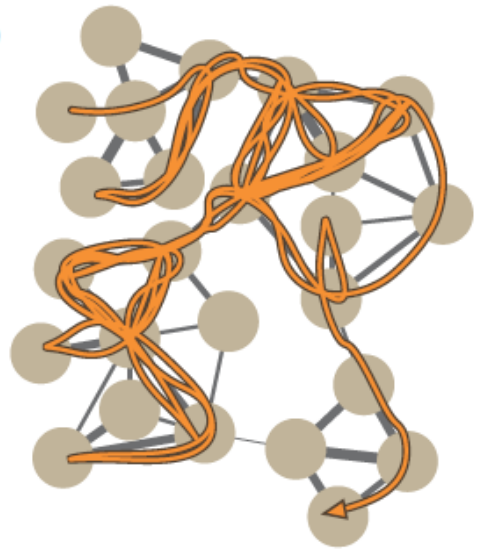
Modularidad para redes pesadas dirigidas

$$Q = \sum_{i=1}^{m} \frac{w_{ii}}{w} - \frac{w_i^{\text{in}} w_i^{\text{out}}}{w^2}.$$

Infomap

Existe una **dualidad** entre

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



Caminatas aleatorias como indicador de estructura de la red

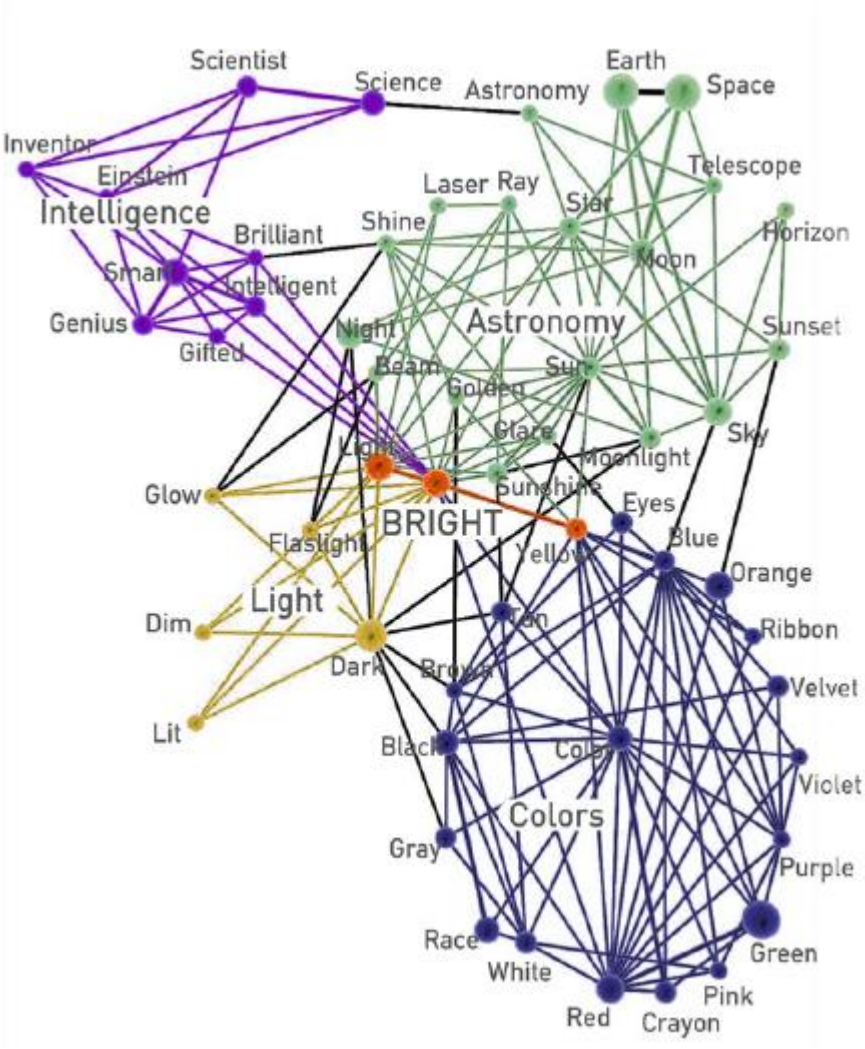
Ingredientes

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

www.mapequation.org

Rosvall & Bergstrom PNAS 2008 Maps of random walks on complex networks reveal community structure

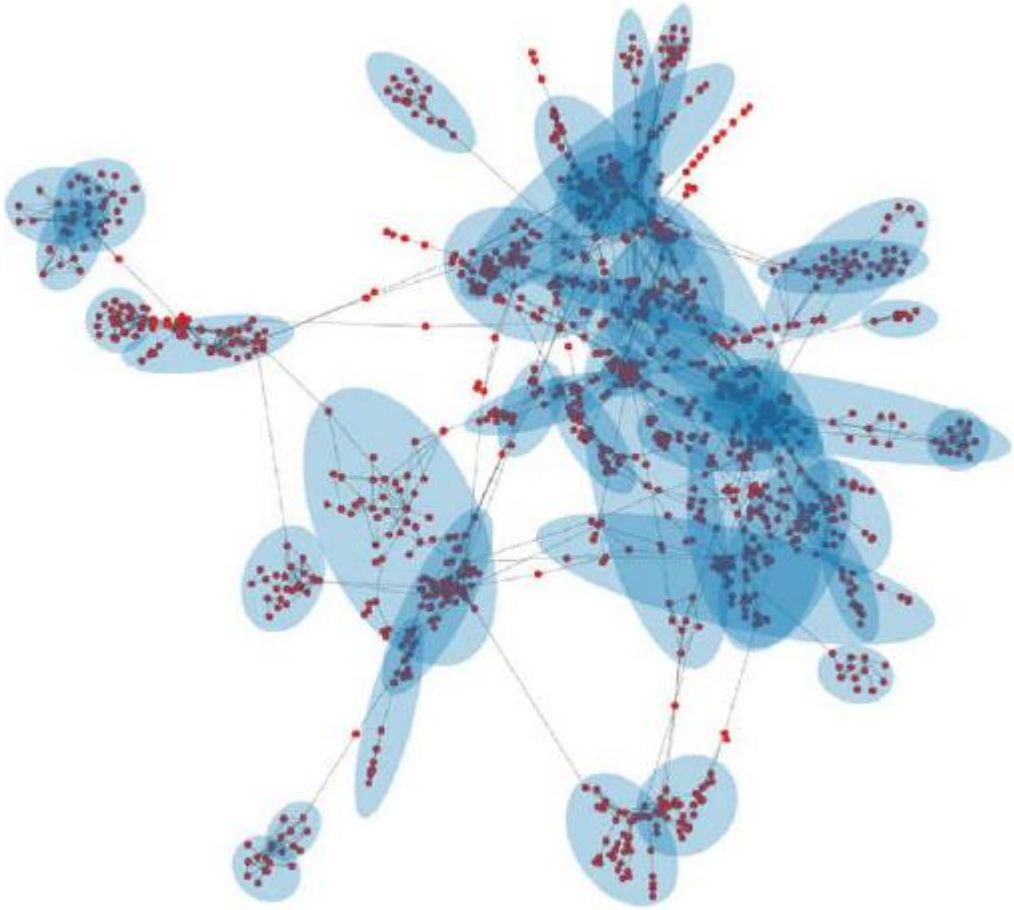
Comunidades solapadas



Red ego para la palabra **bright** en la *South Florida Free Association Network* en la que dos palabras estan conectadas si poseen un significado relacionado.

La pertenencia **simultanea** de **b*right*** a diferentes comunidades (CFinder) da cuenta de los diferentes significados de la palabra.

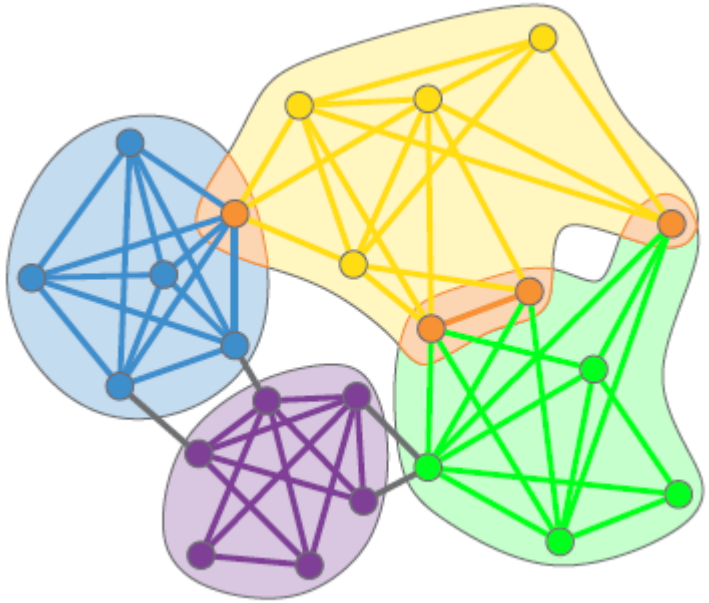
Comunidades solapadas



Comunidades AGM de una red de interacción de proteinas.

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

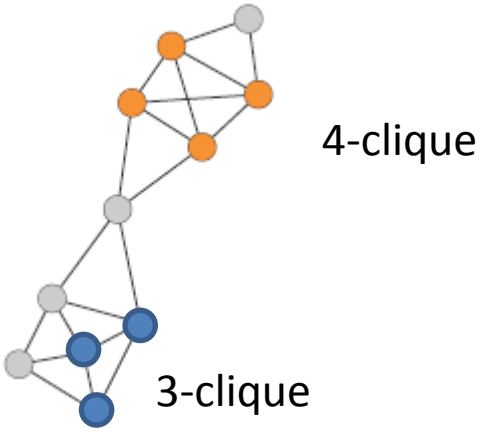
Comunidades solapadas



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

Método de percolación de cliques

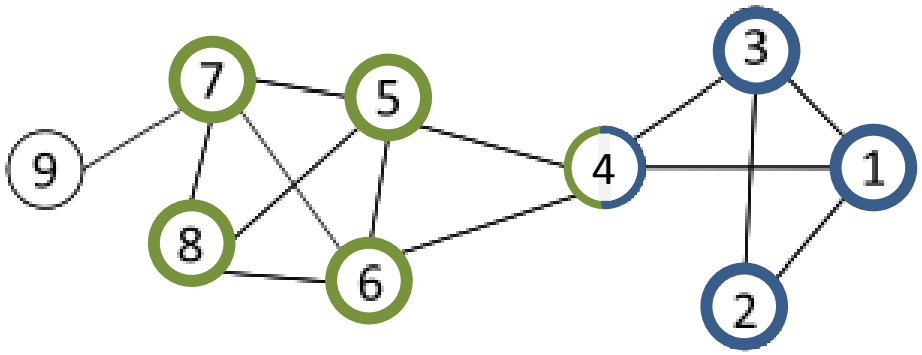
- Vimos que un clique implica una noción fuerte de comunidad
- Se los suele usar como **estructuras semilla** a partir de las cuales definir comunidades más grandes
- MPC es una metodología que permite busacar comunidades solapadas.



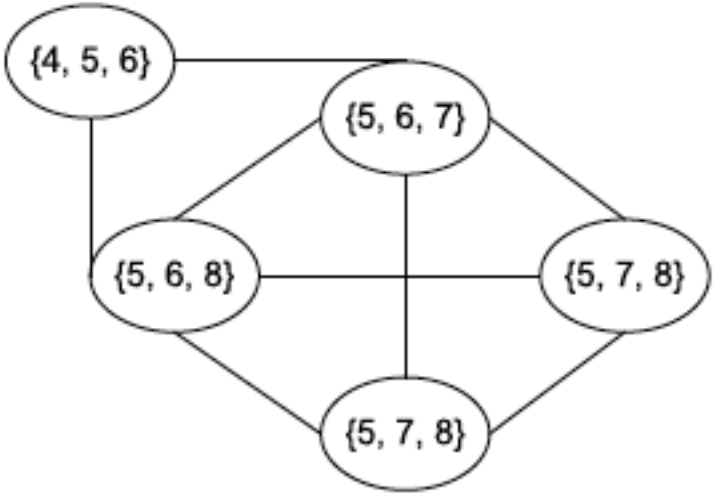
1. Se elige un valor para k
2. Se encuentran todos los k-cliques de la red
3. Se construye un **grafo de cliques**, donde dos k-cliques estarán vinculados si comparten k-1 nodos.
4. Cada componente conexa de este grafo forma una comunidad

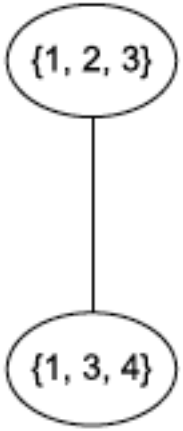
Palla 2005 Nature , Uncovering the overlapping community structure of complex networks in nature and society

Ejemplo de MPC



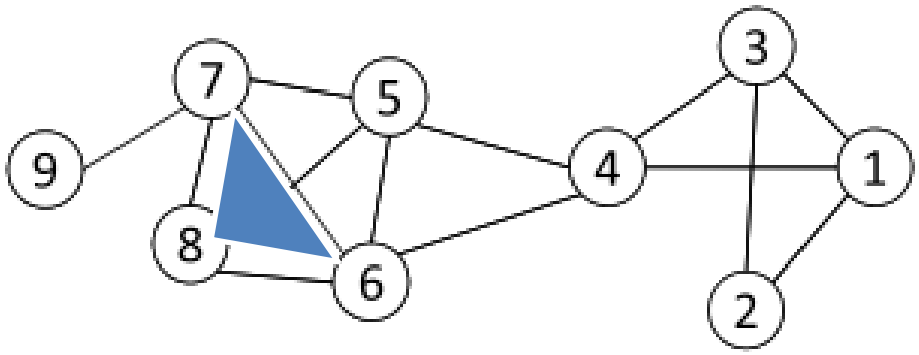
3-Cliques:
{1, 2, 3}, {1, 3, 4}, {4, 5, 6}, {5, 6, 7}, {5, 6, 8}, {5, 7, 8}, {6, 7, 8}



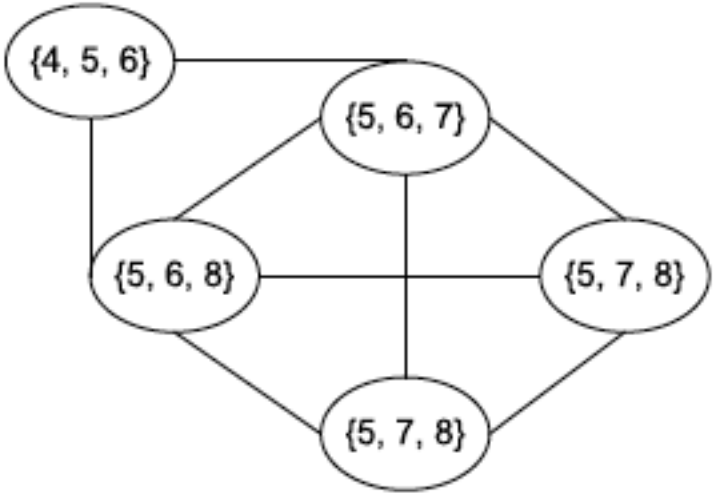


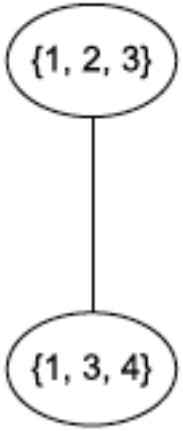
Comunidades:
{1, 2, 3, 4}
{4, 5, 6, 7, 8}

Ejemplo de MPC: rolling k-cliques



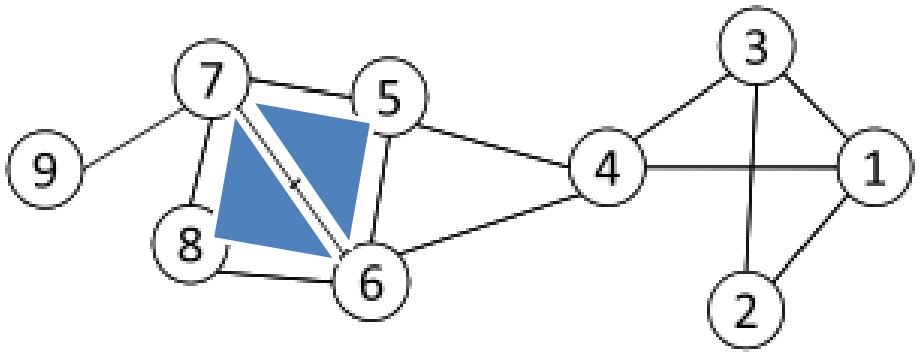
3-Cliques:
{1, 2, 3}, {1, 3, 4}, {4, 5, 6}, {5, 6, 7}, {5, 6, 8}, {5, 7, 8}, {6, 7, 8}



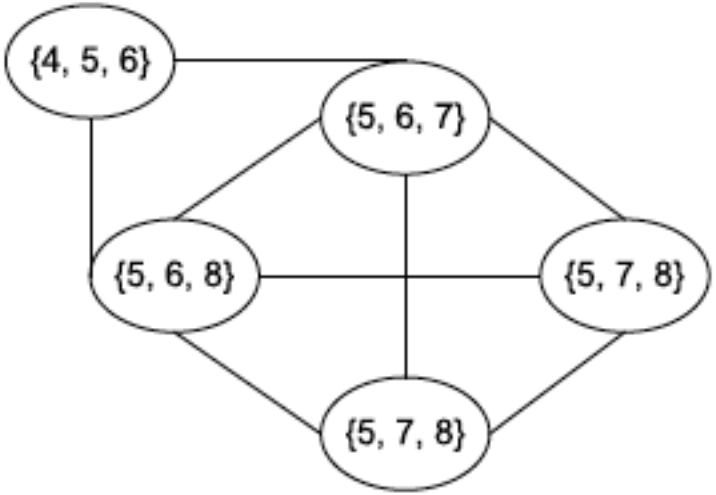


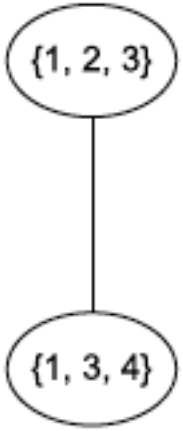
Comunidades:
{1, 2, 3, 4}
{4, 5, 6, 7, 8}

Ejemplo de MPC: rolling k-cliques



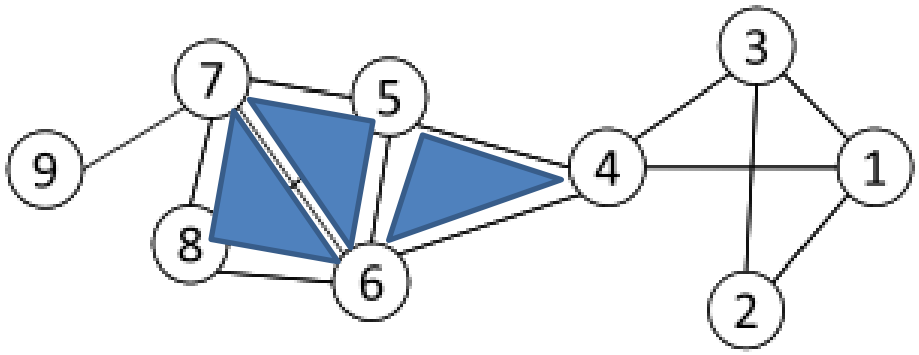
3-Cliques:
{1, 2, 3}, {1, 3, 4}, {4, 5, 6}, {5, 6, 7}, {5, 6, 8}, {5, 7, 8}, {6, 7, 8}



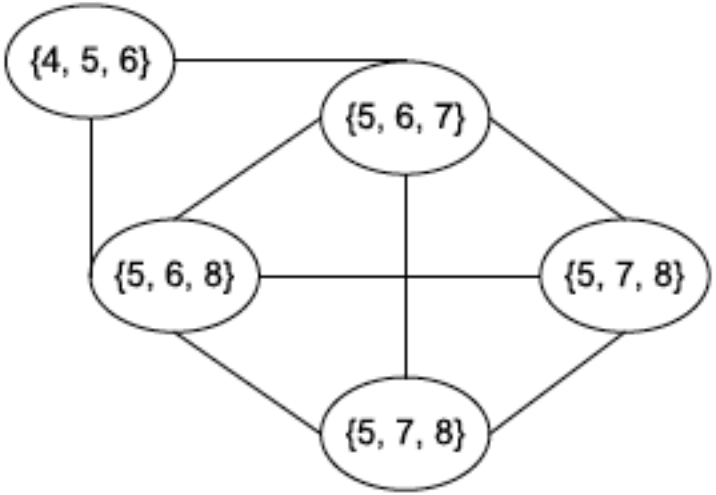


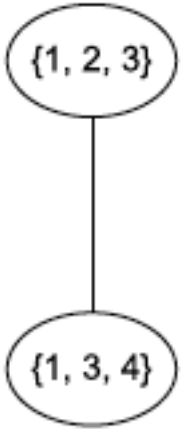
Comunidades:
{1, 2, 3, 4}
{4, 5, 6, 7, 8}

Ejemplo de MPC: rolling k-cliques



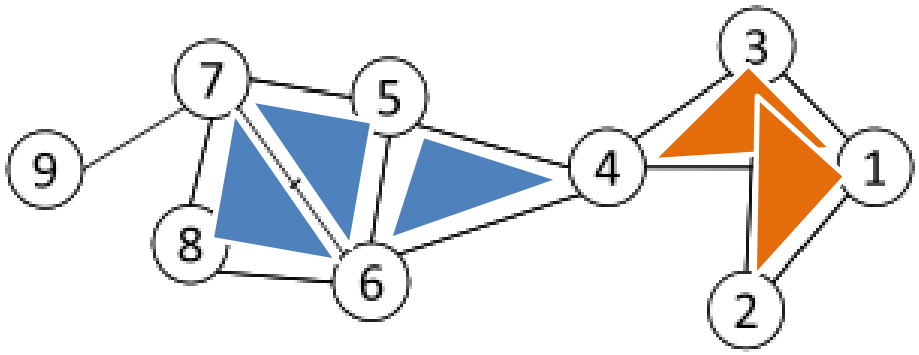
3-Cliques:
{1, 2, 3}, {1, 3, 4}, {4, 5, 6}, {5, 6, 7}, {5, 6, 8}, {5, 7, 8}, {6, 7, 8}



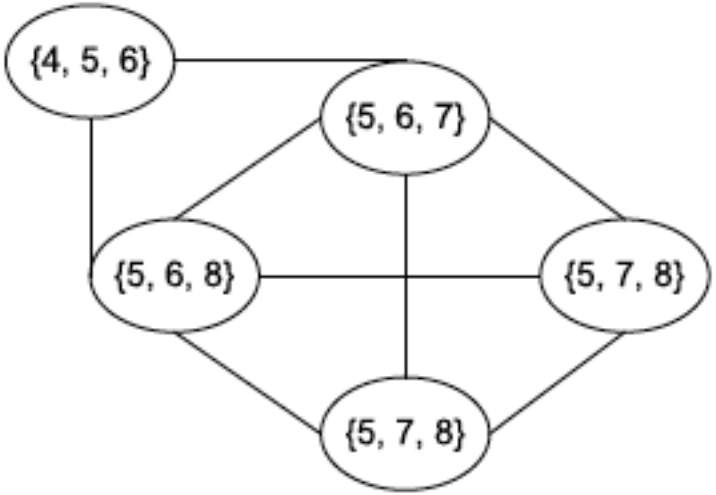


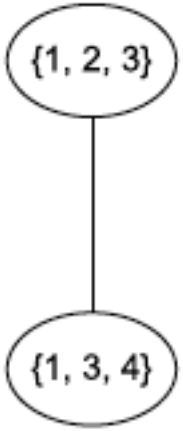
Comunidades:
{1, 2, 3, 4}
{4, 5, 6, 7, 8}

Ejemplo de MPC: rolling k-cliques

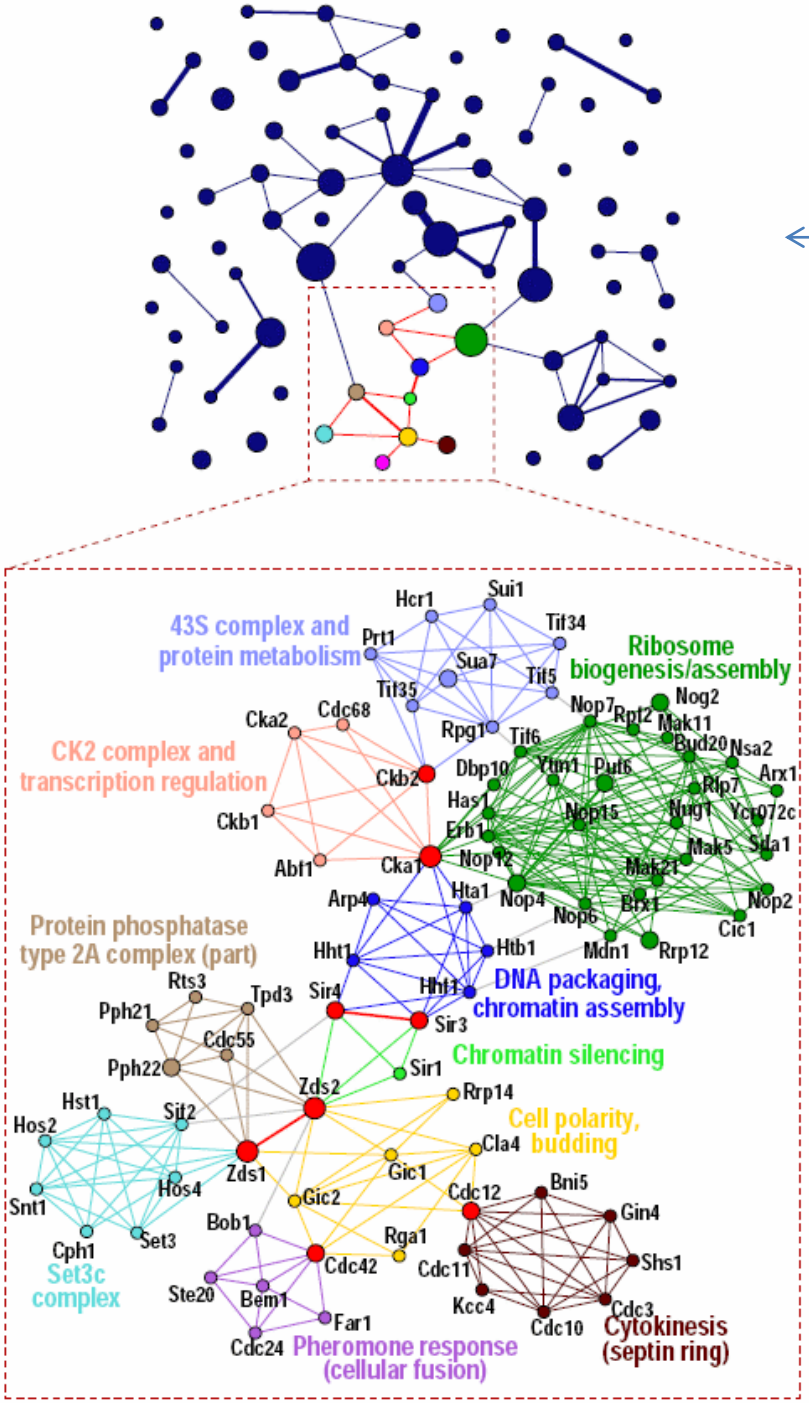


3-Cliques:
{1, 2, 3}, {1, 3, 4}, {4, 5, 6}, {5, 6, 7}, {5, 6, 8}, {5, 7, 8}, {6, 7, 8}





Comunidades:
{1, 2, 3, 4}
{4, 5, 6, 7, 8}



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

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

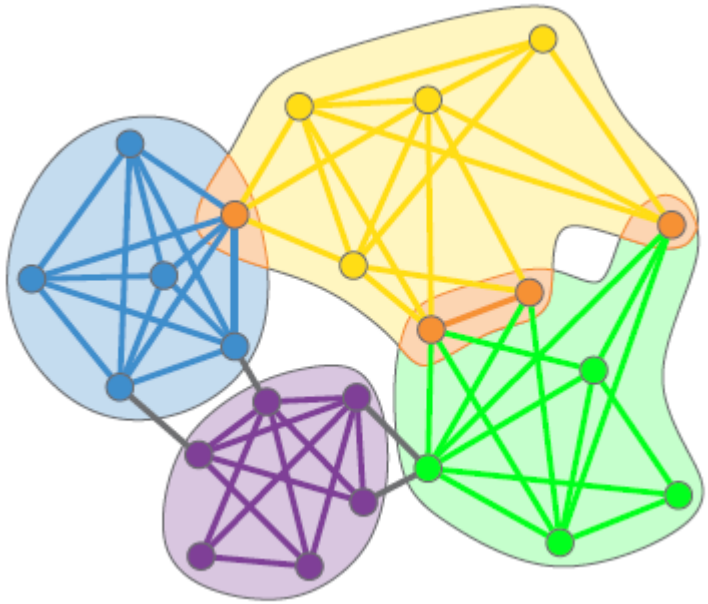
En el zoom se han coloreado las comunidades y los nodos que pertenecen a más de una comunidad aparecen en rojo

Adamcsek 2006 Bioinformatics

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: vinculos de familia, trabajo, hoby, club, etc
  - Redes biológicas: interacción de lugar a función biológica.

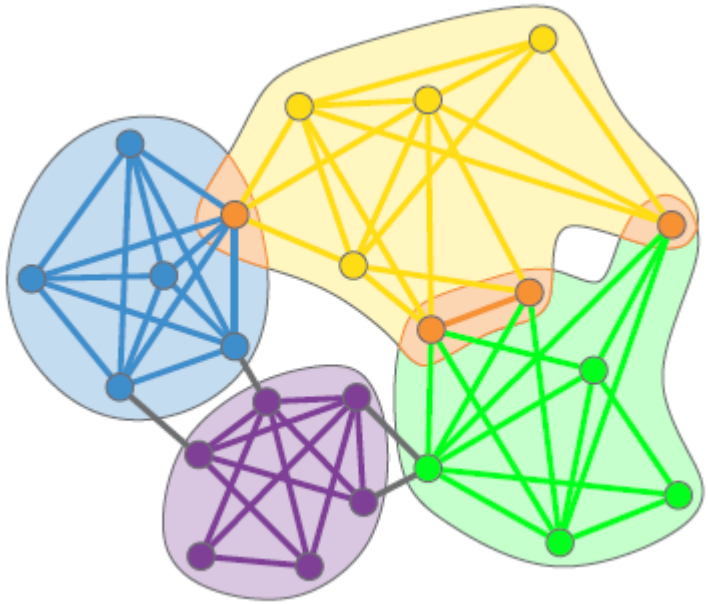


Idea: **agrupar enlaces** de manera jerárquica.
Sólo es necesario saber como medir **similitud entre enlaces**

Ahn, Bragow and Lehmann, *Nature* 466 (2010).

Agrupamiento de enlaces

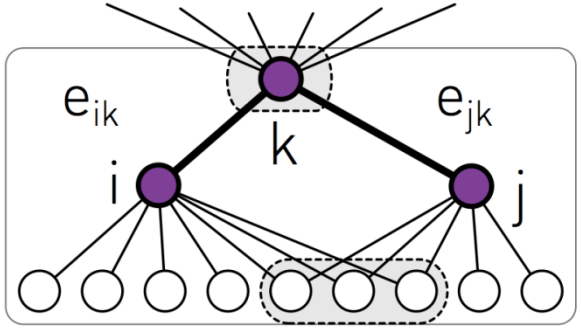
- Nodos pueden pertenecer a más de una comunidad
- Los enlaces capturan la naturaleza de la interacción entre nodos por lo que tienden a ser más específicos.
  - Redes Sociales: vinculos de familia, trabajo, hoby, club, etc
  - Redes biológicas: interacción de lugar a función biológica.



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

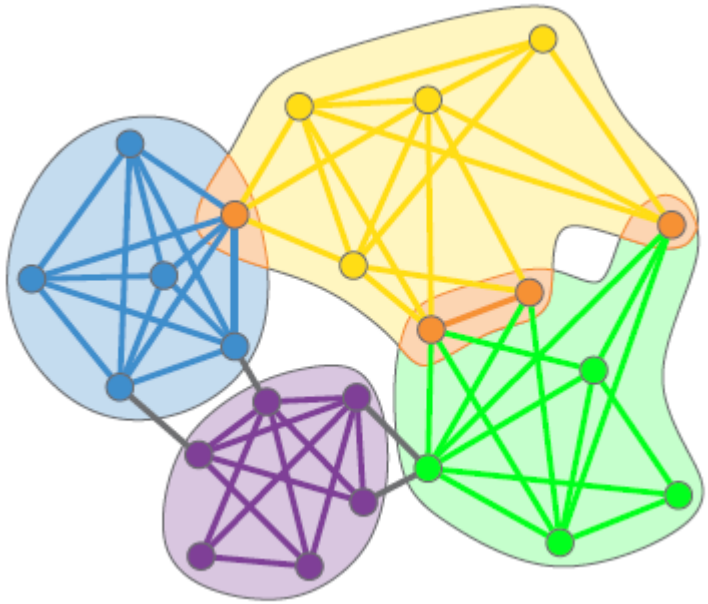
$$S(e_{ik}, e_{jk}) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|}$$

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



Agrupamiento de enlaces

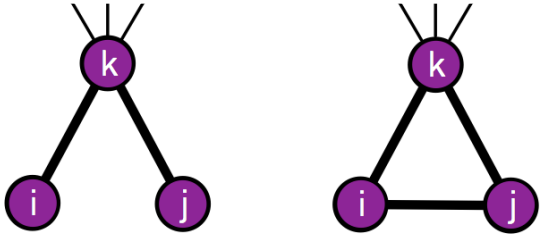
- Nodos pueden pertenecer a más de una comunidad
- Los enlaces capturan la naturaleza de la interacción entre nodos por lo que tienden a ser más específicos.
  - Redes Sociales: vinculos de familia, trabajo, hoby, club, etc
  - Redes biológicas: interacción de lugar a función biológica.



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

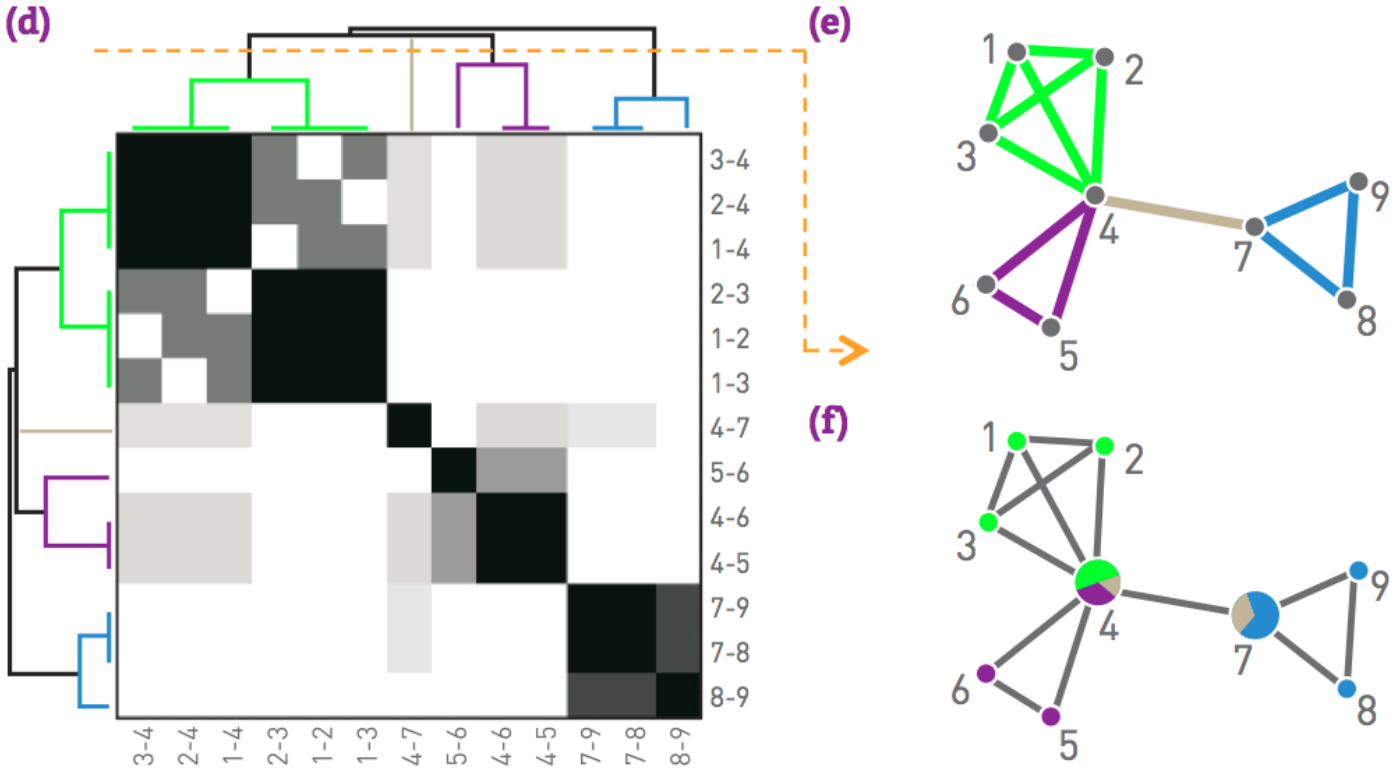
$$S(e_{ik}, e_{jk}) = \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|}$$

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



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

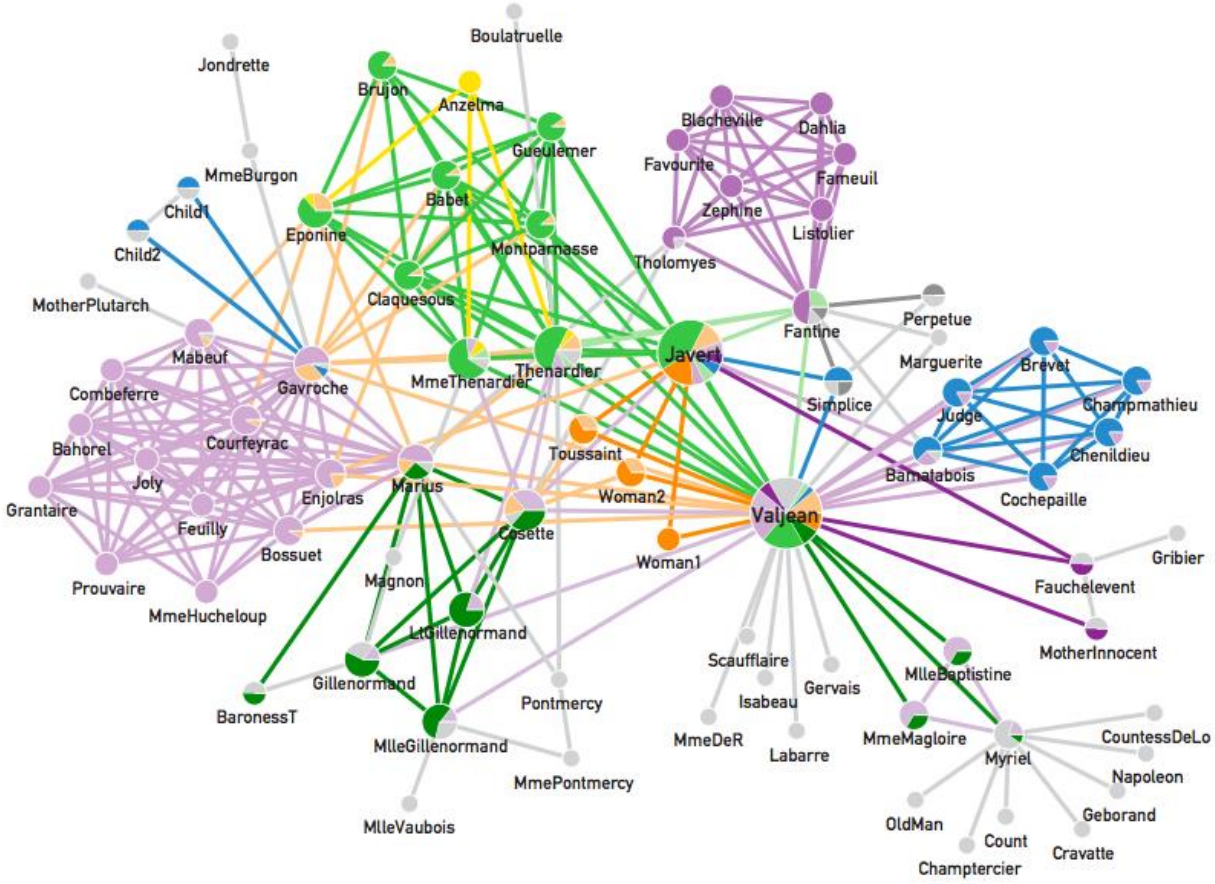
Agrupamiento de enlaces



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

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

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

Evaluando Particiones

- Medidas internas:

  Idea general, cuantificar nivel de compacidad/separación de grupos

  - Modularidad
  - Silhouette
  - ….

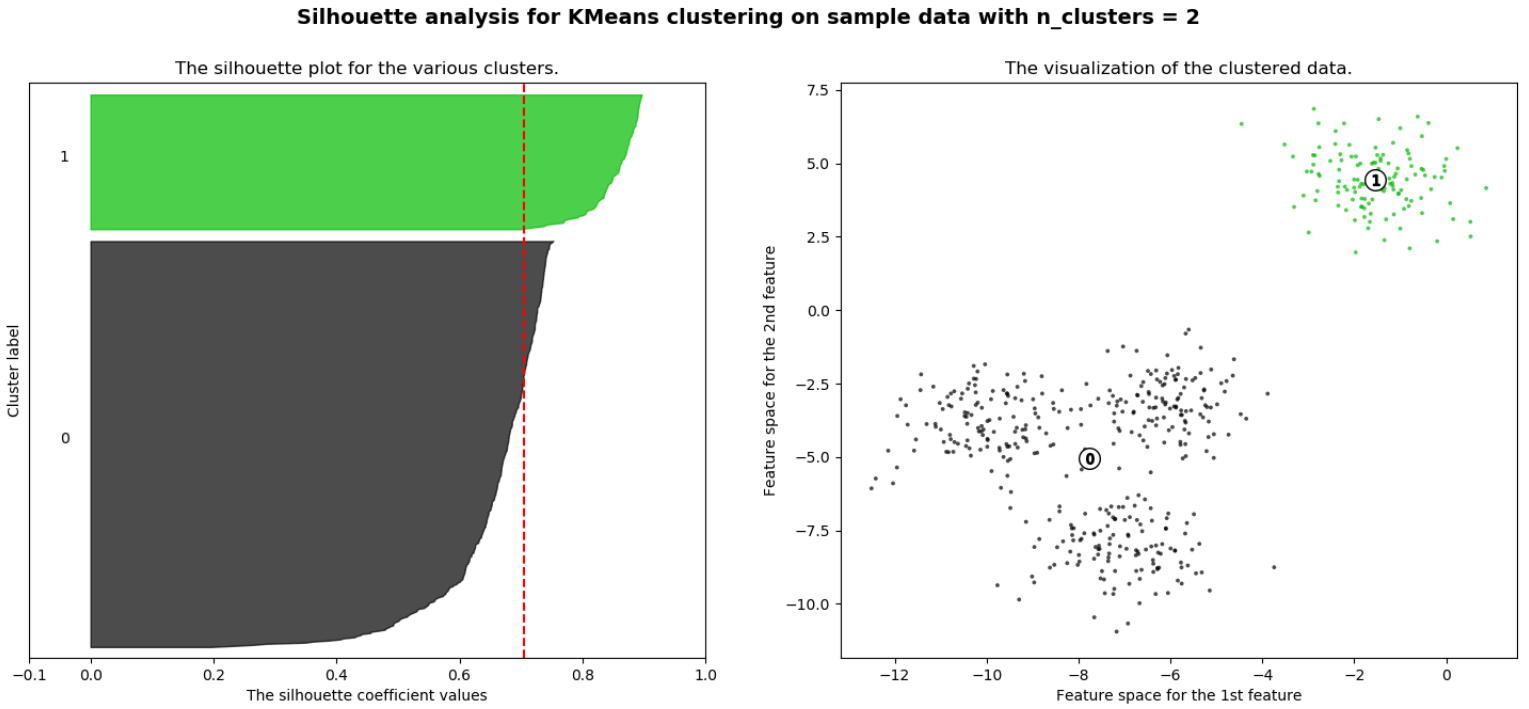
Evaluando Particiones

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



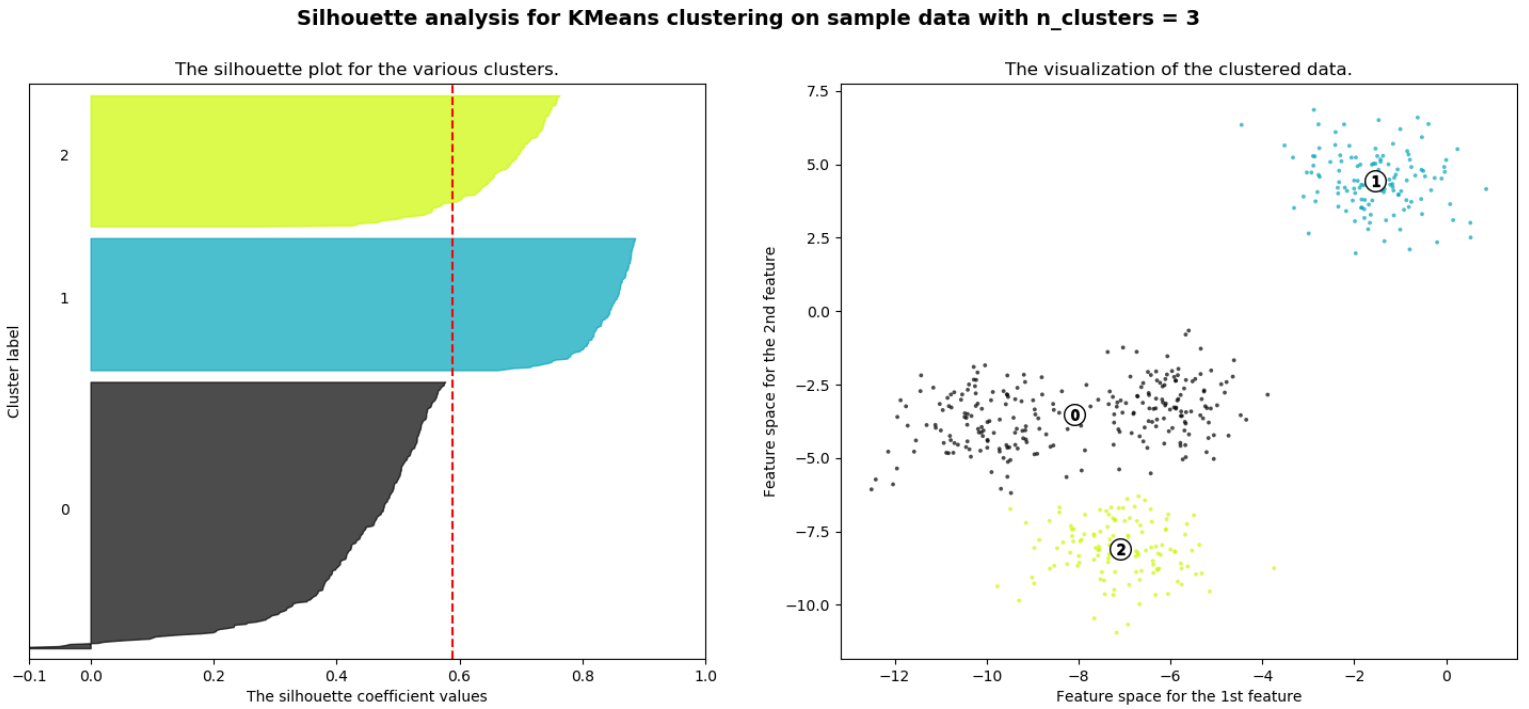
Evaluando Particiones

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



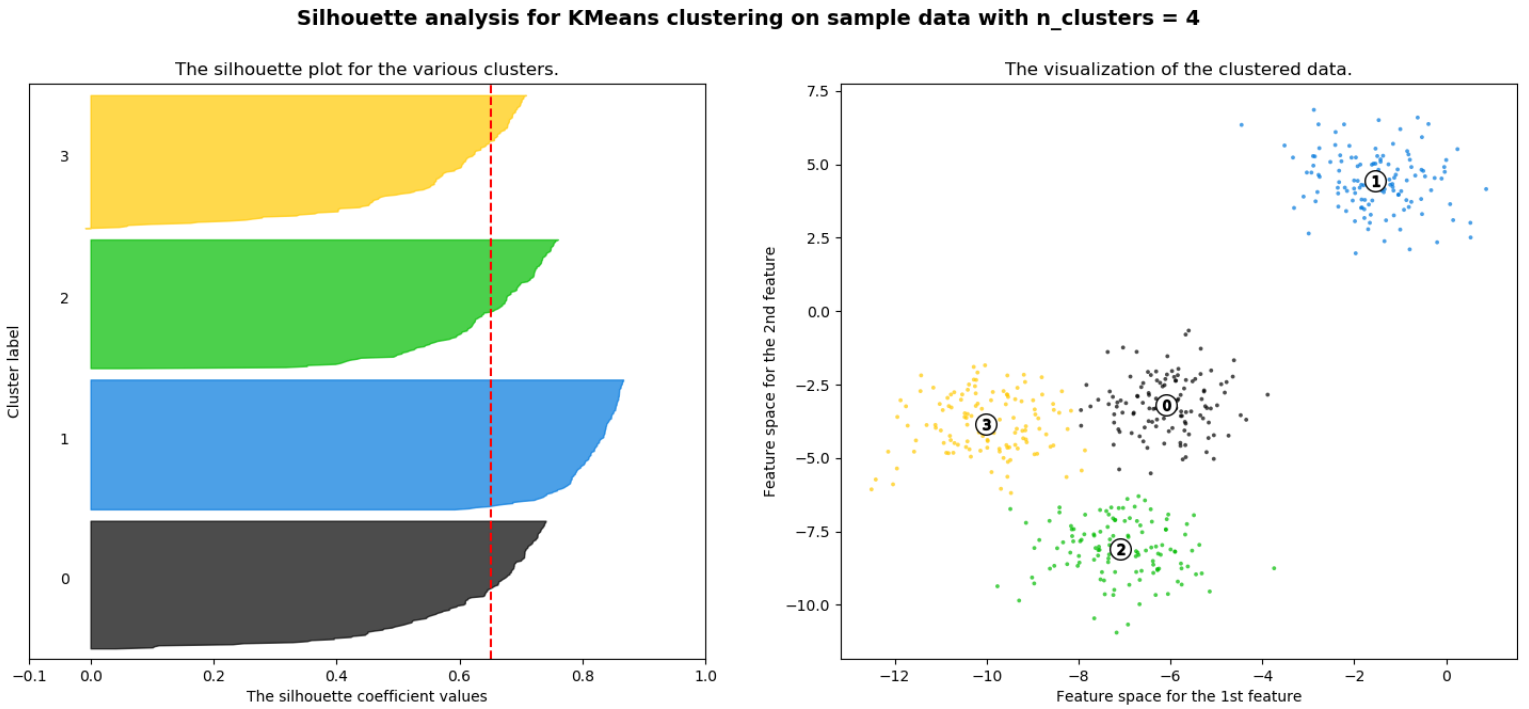
Evaluando Particiones

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



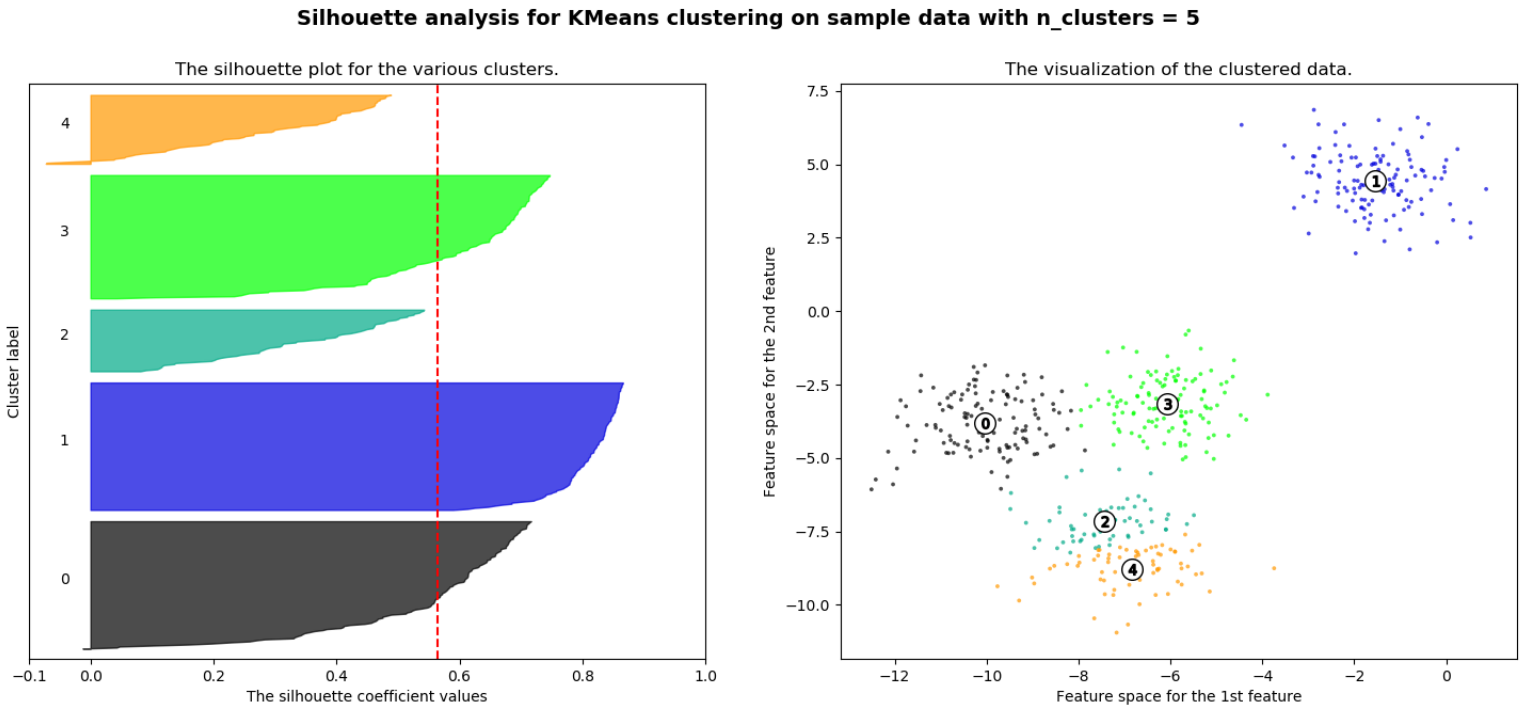
Evaluando Particiones

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



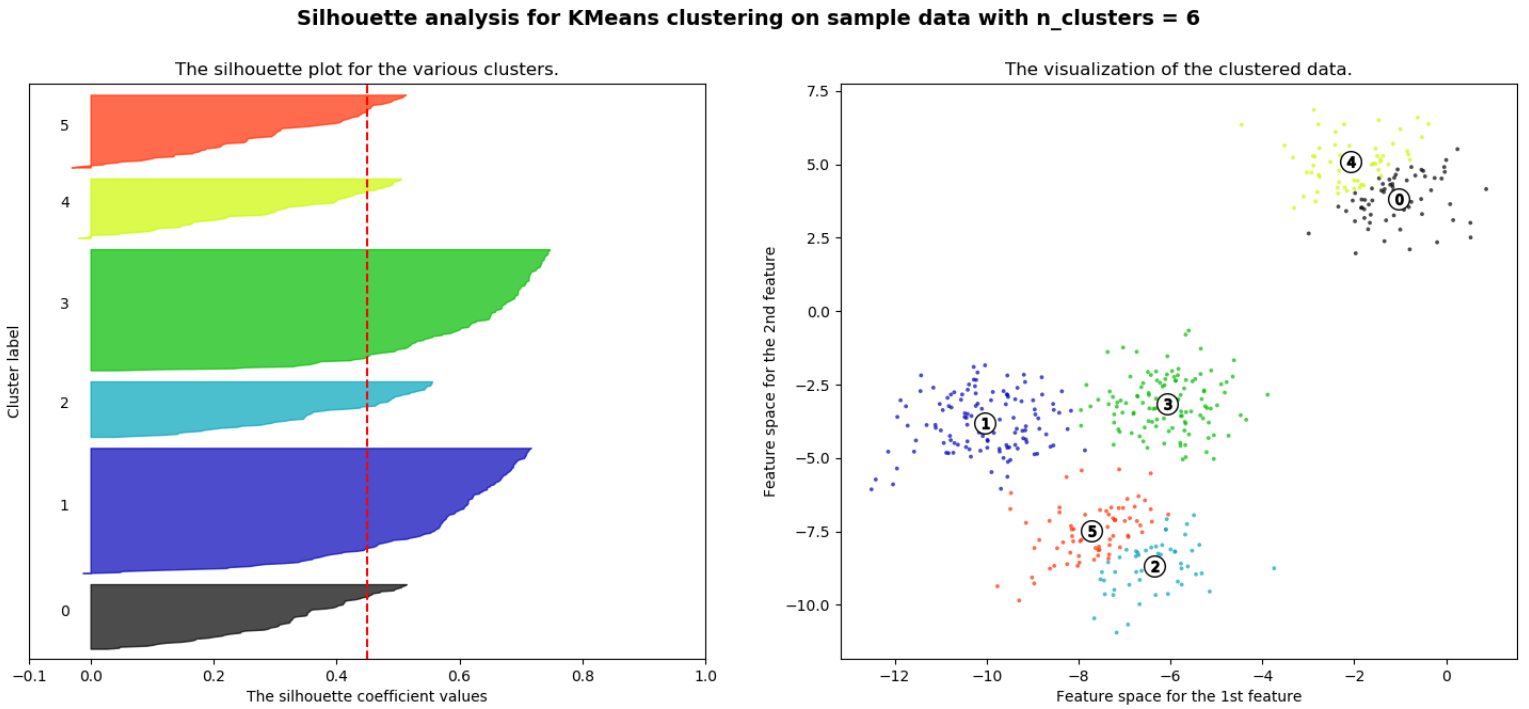
Evaluando Particiones

Compacidad/Separación de grupos : Silhouette

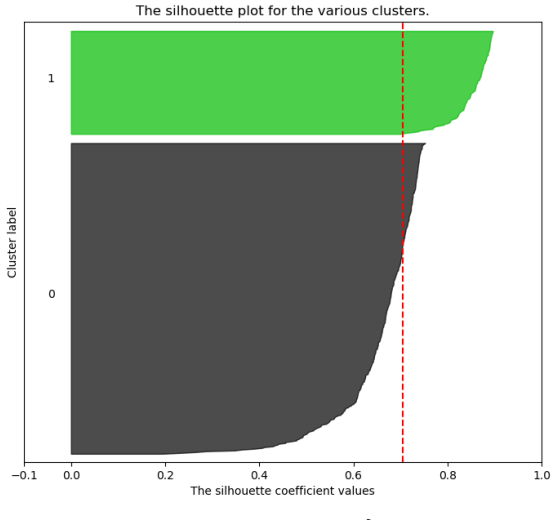
$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

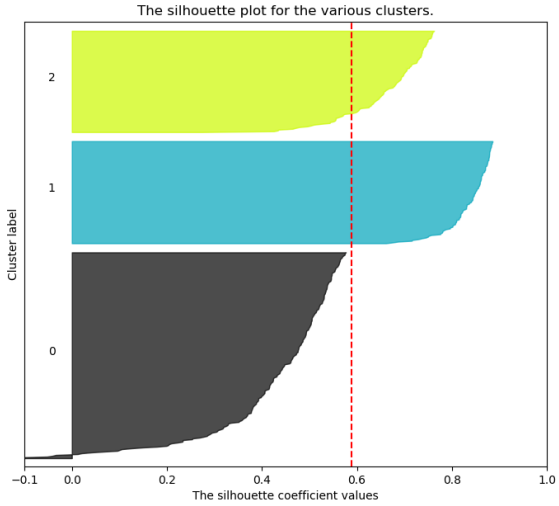
a(i): distancia media del nodo-i con el resto de su cluster

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

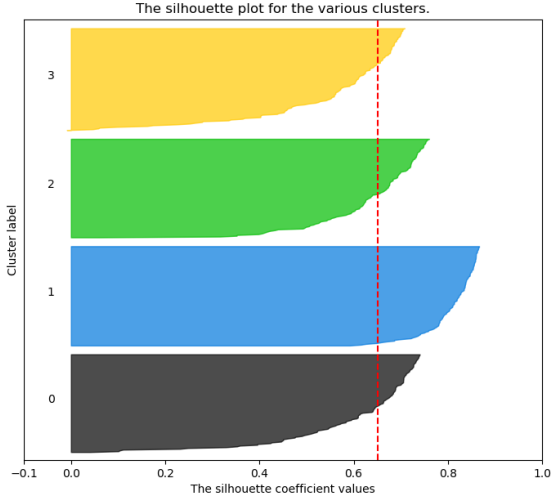


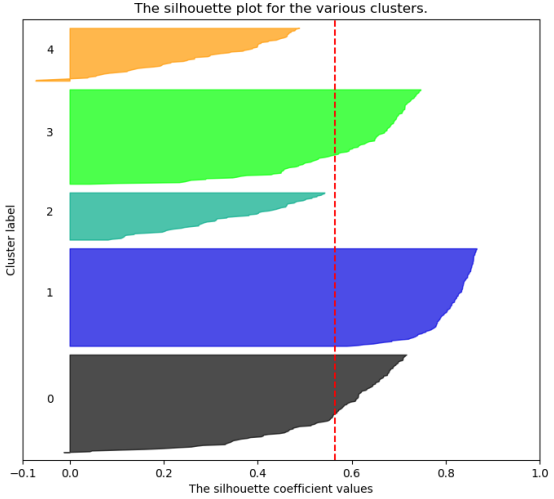
Evaluando Particiones

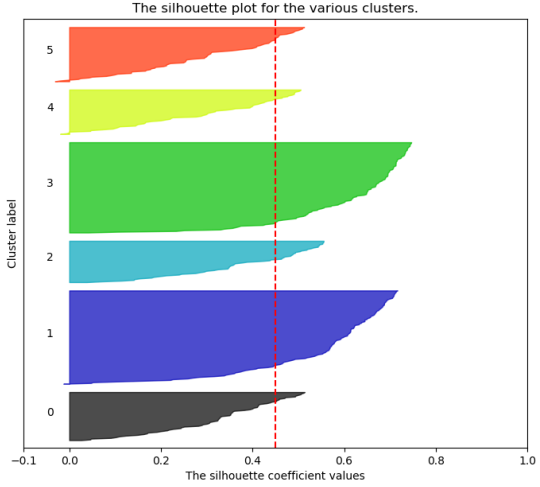




- Altura de stacks idea de asimetria en tamaño de clusters
- Buen clustering = alto silhouette
- Particiones Nclus=3,5,6 tienen clusters con Silhouette score por debajo de la media







Evaluando Particiones

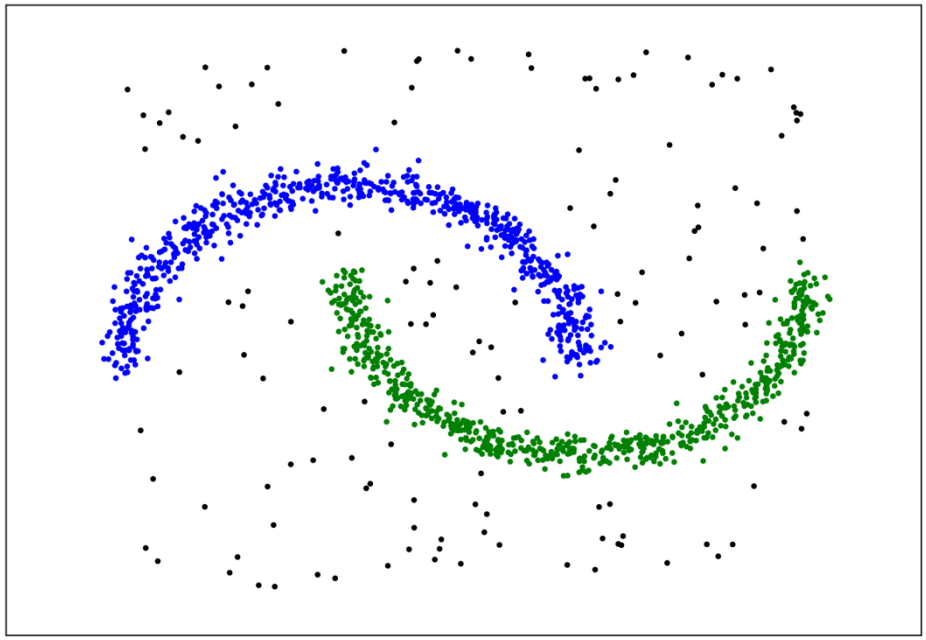
Compacidad/Separación de grupos : Silhouette

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

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

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

OJO con esto!!!
Los datos no tienen la culpa….



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

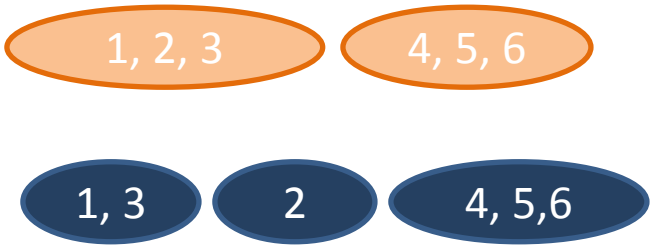
Cómo cuantificar acuerdo/desacuerdo?

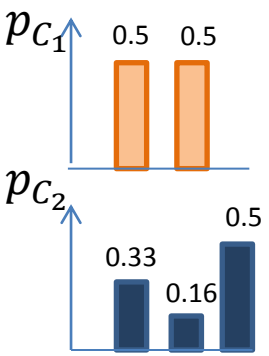
Proba/Teoría de la Información

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

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

Partición a: [1, 1, 1, 2, 2, 2]
Partición b: [1, 2, 1, 3, 3, 3]





Prob conjunta p_{C_1,C_2}

	j=1	j=2	j=3
i=1	2/6	1/6	0
i=2	0	0	3/6

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

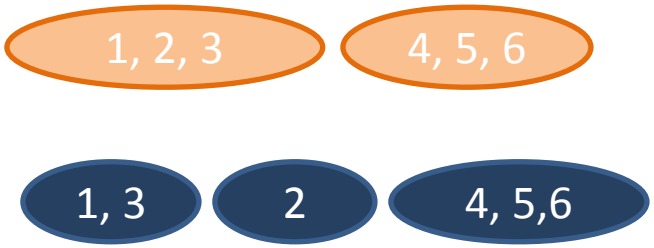
$$p(C_1, C_2) = \frac{N_{C_1C_2}}{\sum_{C_1,C_2} N_{C_1C_2}}$$: probabilidad **conjunta** de que un nodo elegido al azar pertenezca a la comunidad C_1 de la primera partición y a la C_2 de la segunda

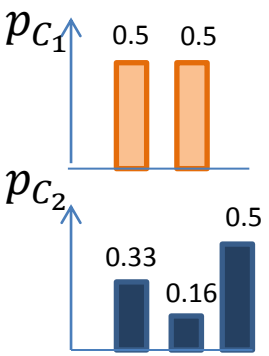
Proba/Teoría de la Información

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

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

Partición a: [1, 1, 1, 2, 2, 2]
Partición b: [1, 2, 1, 3, 3, 3]





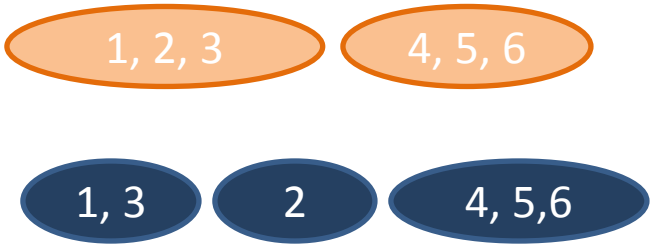
Prob conjunta $p_{C_1C_2} \neq (p_{C_1} * p_{C_2})$

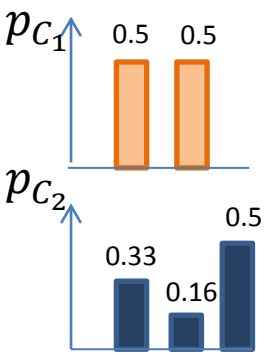
	j=1	j=2	j=3
i=1	2/6	1/6	0
i=2	0	0	3/6

	j=1	j=2	j=3
i=1	1/6	1/12	1/4
i=2	1/6	1/12	1/4

Que tan coherentes son dos particiones?

Partición a: [1, 1, 1, 2, 2, 2]
Partición b: [1, 2, 1, 3, 3, 3]





$p_{C_1C_2}$

	j=1	j=2	j=3
i=1	2/6	1/6	0
i=2	0	0	3/6

$p_{C_1}p_{C_2}$

	j=1	j=2	j=3
i=1	1/6	1/12	1/4
i=2	1/6	1/12	1/4

Informacion Mutua:

$$I(\{C_1\},\{C_2\}) = \sum_{C_1}\sum_{C_2} p(C_1C_2)\log\frac{p(C_1,C_2)}{p(C_1)p(C_2)}$$

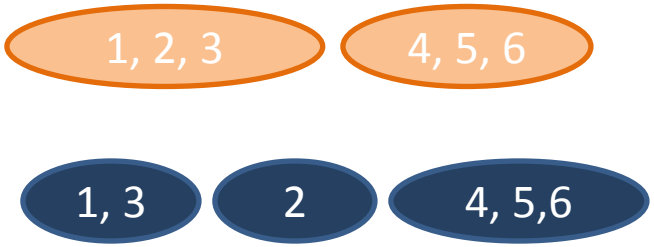
Cotas para la IM:

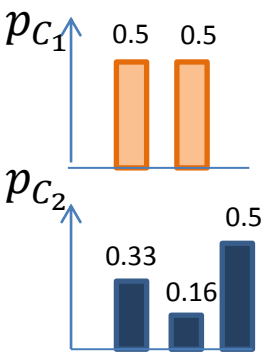
- Sea $\{C_2\}$ igual $\{C_1\}$: $p(C_1C_2) = p_{C_1}\delta_{C_1C_2}$

$$I(\{C_1\},\{C_2 = C_1\}) = \sum_{C_1}\sum_{C_2} p_{C_1}\delta_{C_1C_2}\log\frac{p_{C_1}\delta_{C_1C_2}}{p_{C_1}p_{C_2}} = \sum_{C_1} p_{C_1}\sum_{C_2}\delta_{C_1C_2}\log\frac{1}{p_{C_2}} = \sum_{C_1} p_{C_1}\log\frac{1}{p_{C_1}}$$

Que tan coherentes son dos particiones?

Partición a: [1, 1, 1, 2, 2, 2]
Partición b: [1, 2, 1, 3, 3, 3]





$p_{C_1C_2}$

	j=1	j=2	j=3
i=1	2/6	1/6	0
i=2	0	0	3/6

$p_{C_1}p_{C_2}$

	j=1	j=2	j=3
i=1	1/6	1/12	1/4
i=2	1/6	1/12	1/4

Informacion Mutua:

$$I(\{C_1\},\{C_2\}) = \sum_{C_1}\sum_{C_2} p(C_1C_2)\log\frac{p(C_1,C_2)}{p(C_1)p(C_2)}$$

Cotas para la IM:

- Sea $\{C_2\}$ igual $\{C_1\}$: $p(C_1C_2) = p_{C_1}\delta_{C_1C_2}$

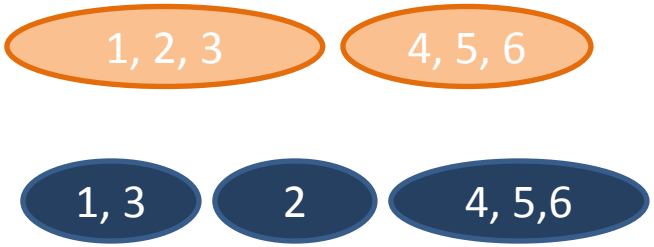
$$I(\{C_1\},\{C_2 = C_1\}) = -\sum_{C_1} p_{C_1}\log p_{C_1} = H(\{C_1\})$$

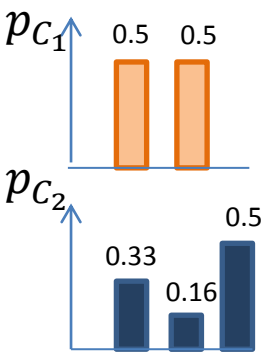
Entropía de Shanon:

cuanta información obtengo al conocer la realización de una variable aleatoria

Que tan coherentes son dos particiones?

Partición a: [1, 1, 1, 2, 2, 2]
Partición b: [1, 2, 1, 3, 3, 3]





$p_{C_1C_2}$

	j=1	j=2	j=3
i=1	2/6	1/6	0
i=2	0	0	3/6

$p_{C_1}p_{C_2}$

	j=1	j=2	j=3
i=1	1/6	1/12	1/4
i=2	1/6	1/12	1/4

Informacion Mutua:

$$I(\{C_1\},\{C_2\}) = \sum_{C_1}\sum_{C_2} p(C_1C_2)\log\frac{p(C_1,C_2)}{p(C_1)p(C_2)}$$

Cotas para la IM:

- Sea $\{C_2\}$ igual $\{C_1\}$: $p(C_1C_2) = p_{C_1}\delta_{C_1C_2}$

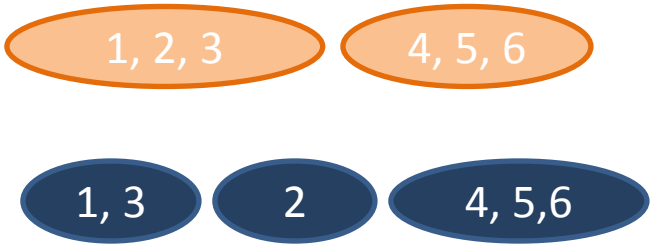
$$I(\{C_1\},\{C_2 = C_1\}) = -\sum_{C_1} p_{C_1}\log p_{C_1} = H(\{C_1\})$$

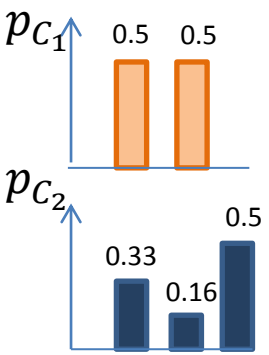
- Sea $\{C_1\}$ igual $\{C_2\}$: $p(C_1C_2) = p_{C_2}\delta_{C_1C_2}$

$$I(\{C_1 = C_2\},\{C_2\}) = -\sum_{C_2} p_{C_2}\log p_{C_2} = H(\{C_2\})$$

Que tan coherentes son dos particiones?

Partición a: [1, 1, 1, 2, 2, 2]
Partición b: [1, 2, 1, 3, 3, 3]





$p_{C_1C_2}$

	j=1	j=2	j=3
i=1	2/6	1/6	0
i=2	0	0	3/6

$p_{C_1}p_{C_2}$

	j=1	j=2	j=3
i=1	1/6	1/12	1/4
i=2	1/6	1/12	1/4

Informacion Mutua:

$$I(\{C_1\},\{C_2\}) = \sum_{C_1}\sum_{C_2} p(C_1C_2)\log\frac{p(C_1,C_2)}{p(C_1)p(C_2)}$$

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

$$0 < I_n < 1$$

Precisión

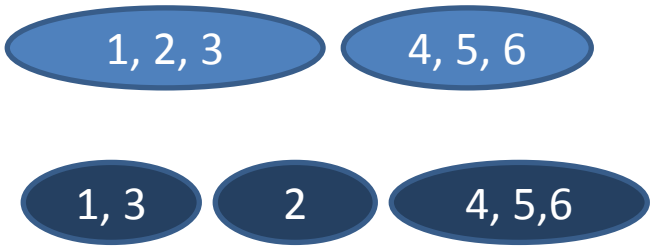
- Consideramos todos los posibles pares de nodos y evaluamos si residen en la misma comunidad que en la partición de referencia luego de la detección
- Habra un error *si*
  - Dos nodos que pertenecen a la misma comunidad de referencia son asignados a comunidades diferentes
  - Dos nodos de diferentes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

clustering	Referencia C(i)=C(j)	Referencia C(i)≠C(j)
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)	Referencia 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